VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ
FACULTY OF BUSINESS AND MANAGEMENT

HODNOTA PRO VLASTNÍKY A JEJÍ GENERÁTORY V PROSTŘEDÍ MALÝCH A STŘEDNÍCH PODNIKŮ

SHAREHOLDERS' VALUE AND ITS GENERATORS IN THE ENVIRONMENT OF SMALL AND MEDIUM-SIZED ENTERPRISES

HABILITAČNÍ PRÁCE V OBORU EKONOMIKA A MANAGEMENT
HABILITATION THESIS IN THE FIELD ECONOMICS AND MANAGEMENT

BRNO 2021

Abstrakt

Práce se zabývá predikcí kategorizace malých a středních podniků v České republice, která je stanovena na základě ekonomické přidané hodnoty. Do provedených analýz v rámci testovaných modelů vstupovaly účetní závěrky a další údaje o více jak 25 tis. podnicích. Data z účetních závěrek byla dále upravována a celkově bylo stanoveno celkem 15 prediktorů, kde 5 prediktorů bylo kategoriálních a zbytek numerických.

Vstupní data byla za období 2013-2017. Nad uvedeným datovým souborem proběhlo několik analýz strojového učení bez učitele a s učitelem. U metod bez učitele byly použity Principal coponent analytsis, k-means clustering, Gaussian mixture models, Hierarchical clustering a Kohonenovy sítě. V případě metod s učitelem byly provedeny analýzy Nejbližší soused, Stromová klasifikace, Naive Bayes klasifikace, Discriminant analýza, Multiclass support machine a dopředné neuronové sítě. Výsledky jednotlivých metod byly optimalizovány vlastními i automatickými algoritmy. Pro lepší interpretovatelnost výsledků byla data vizualizována.

Za nejvhodnější model lze označit ten, který dosahuje schopnosti klasifikace podniků do čtyř kategorií ve výši 45 % na kontrolní množině podniků. Jedná se o model založený na stromové struktuře. Při zjednodušení zatřídění do dvou kategorií dokáže model predikovat kladný výsledek hospodaření nad úrovní bezrizikových výnosů s pravděpodobností přesahující 80 %. Zjednodušený rozhodovací model je přílohou práce a může být na něj navázáno softwarem pro testování podniků či využit pro tvorbou vnitropodnikových cílů.

Klíčová slova

Predikce, Ekonomická přidaná hodnota, Principal coponent analytsis, k-means clustering, Gaussian mixture models, Hierarchical clustering, Kohonenovy sítě, Nejbližší soused, Stromová klasifikace, Naive Bayes klasifikace, Discriminant analýza, Multiclass support machine, dopředné neuronové sítě.

Abstract

The work deals with forecasting a category of enterprises determined based on economic value added. The analysis included financial statements and other data about more than 25,000 enterprises. The financial statements data were further processed and a total of 15 predictors were determined, out of which 5 were categorical predictors, while the remaining 10 predictors were numerical.

The above dataset was subjected to several analysis of machine learning both without and without a supervision. In the case of methods without a supervision, Principal component analysis, k-means clustering, Gaussian mixture models, Hierarchical clustering and Kohonen networks were included. The methods with a supervision included Nearest neighbour, Tree algorithm, Naïve Bayes classification, Discriminant analysis, Multiclass support machine, and Feed-forward neural networks. The results of the individual methods were optimized using own and automatic algorithm. For better interpretation of the results obtained, the data were visualised.

The most suitable model is capable to classify the enterprises into four categories of 45 % on the control set of enterprises. It is a model based on tree structure. By simplifying the classification to two categories, the model is able to forecast a positive economic result above the risk-free yield with a probability exceeding 80 %. The simplified decision-making model is attached to the work and can be followed by software or creation of enterprise internal goals.

Key Words

Prediction, Economic value added, Principal component analysis, k-means clustering, Gaussian mixture models, Hierarchical clustering, Kohonen networks, Nearest neighbour, Tree clustering, Naive Bayes, Discriminant analysis, Multiclass support machine, Feed-forward neural networks.

Čestné prohlášení

Prohlašuji, že předložená habilitační práce s názvem *Hodnota pro vlastníky a její generátory v prostředí malých a středních podniků* je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a oprávek souvisejících s právem autorským).

V Českých Budějovicích 27. října 2021

.....

Ing. Vojtěch Stehel, MBA, PhD.

Poděkování

Velmi děkuji prof. Ing. Janu Váchalovi, CSc., a doc. Ing. Jarmila Strakové, Ph.D. za poskytnutí cenných rad a za řadu zajímavých námětů, které usnadnily řešení habilitační práce. V neposlední řadě bych rád poděkoval své rodině za podporu a trpělivost při zpracování této práce.

Obsah

1 Úvod

Hlavním cíle podniku je podle Kislingerové (2007) růst hodnoty pro akcionáře. Ten můžeme změřit pomocí řady ukazatelů. Mezi těmito ukazateli (Total shareholder value, market value added, free cash flow to equity) zaujímá zajímavou pozici ukazatel ekonomické přidané hodnoty ve variantě ekvity. Jeho hodnota spočívá především v tom, že poměřuje přínos pro akcionáře odvíjející se od rizika, které svou investicí podstupuje. Avšak ekonomická přidaná hodnota, ať už dle Brealey, Mayerse a Allen (2013), Kislingerové (2007) či Neumaierových (2003), je vypočítávána vždy z pohledu akcionáře. Jedná se tak o výpočet zhodnocení vkladu vlastníka, který však neodráží skutečnou tvorbu hodnoty podnikem. Podle teorie výrobních faktorů (Wöhe, 1995) tvoří hodnoty především řídící práce, výkonná práce, dlouhodobý majetek a materiál. Neexistuje tak přímé spojení mezi ukazatelem hodnoty pro akcionáře a výrobními faktory (generátory hodnoty). Avšak dle manažerských teorií vlastníci vyžadují od managementu naplnění především cílů vlastníku podniku. Hlavním úkolem je najít nástroj řízení podniku, jímž by bylo možné dekomponovat cíl růstu hodnoty do dílčích aktivit podniku.

Vzhledem k tomu, že neexistuje přímá souvislost, hledáme kauzální vztah vstupních veličin, tedy generátorů hodnoty a cíle vlastníků, potažmo podniku. Pokud takový vztah najdeme budeme schopni predikovat výsledek na základě určité kombinace vstupů případně budeme kombinovat vstupy, tak abychom dosáhli tíženého cíle.

Dle Vochozky (2019) nejlepších výsledků pro predikci podniků dosahují neuronové sítě. Neuronové sítě a oblast strojového učení jsou ve společnosti skloňovány častěji, než tomu bylo v minulosti. Již se nejedná pouze o teoretickou vědní disciplínu, ale o vědní disciplínu s mimořádně silným aplikačním potenciálem. Existuje minimum oborů, které lidstvo, v takto rozsáhlém směru, ovlivňují a budou ovlivňovat v budoucnu.

Schopnost neuronových sítí a dalších metod strojového učení se učit na základě vzorů a tyto pak dále aplikovat jim dává možnost řídit automobily, překládat texty, psát zprávy, řídit výrobu složitého procesu a mnoho dalšího. Všechny zmíněné činnosti přitom neuronové sítě začínají zvládat lépe než člověk. Díky tomu je například nehodovost aut řízených pomocí neuronových sítí v současné době řádově menší než v případě lidí. Důvodů je celá řada. Mimo jiné je to i schopnost analyzovat a učit se z množství dat, které by člověk nebyl schopen vstřebat za lidský život. V důsledku toho se neuronová síť může poučit z extrémních situací, které se vyskytují s minimálními pravděpodobnostmi.

Dopad na společnost mají neuronové sítě mimořádný, ten se však bude ještě prohlubovat. Již nyní si díky neuronovým sítím snadno přeložíme článek na internetu z cizího jazyka do naší mateřštiny. V momentě, kdy ale ve velkém začnou být využívány autonomní vozidla, dojde k nejenom zvyšování efektivity dopravních společností, ale také k značnému socio-demografickému efektu, který bude ovlivňovat společnost. Postupné vymizení povolání řidiče přitom nebude ojedinělým důsledkem, naopak se bude jednat o řadu dalších profesí ve výrobních firmách, administrativě atd.

Existuje přitom řada důvodů, které povedou k dalšímu rozvoji neuronových sítí a jejich aplikací. Kromě obchodní války mezi USA, Čínou a dalšími zeměmi je zde především samotný fakt lidské neefektivnosti, kdy člověk sice dokáže kontrolovat kvalitu výrobků, ale musí spát, jíst apod. V důsledku toho dříve či později bude docházet k rychlejšímu bodu zvratu při realizaci investice do těchto technologií.

Samozřejmě i neuronové sítě mají svá omezení, a proto se nemusíme obávat toho, že by nás alespoň v nejbližších desetiletích umělá inteligence zcela nahradila. Na druhou stranu se bude muset lidská společnost v určitých směrech transformovat a zohlednit tak moderní vývoj. Toto se týká i obchodních společností. Jejich řízení musí být efektivnější, než tomu bylo v minulosti a musí využívat těchto moderních nástrojů. V opačném případě doje k růstu konkurence, která dříve či později tuto technologii začne využívat.

2 Cíl práce

Cílem práce je provedení kauzální analýzy generátorů hodnoty a EVA ve variantě ekvity na souboru dat malých a středních podniků působících v České republice v letech 2013 až 2017 a vývoj modelu pro řízení hodnoty pro vlastníky malých a středních podniků.

Mimo hlavního cíle podniku byly stanoveny tyto výzkumné otázky:

- Ovlivňuje místo podnikání v České republice výsledek hospodaření?
- Dosahují vyšších hodnot EVA podniky s více zaměstnanci?
- Existuje podstatný rozdíl mezi zaměřením podniku a tvorbou hodnoty EVA?

3 Literární rešerše

Literární rešerše umožní provést ucelený pohled na danou problematiku. Tento pohled se bude skládat z ekonomické části, která se vymezí na základní principy a motivaci pro další výzkum. Druhá část nazvaná strojové učení a neuronové sítě se bude věnovat metodám, které budou klíčové pro posouzení výsledků výzkumu a stanovení závěrů.

3.1 Cíle podniku

Existuje mnoho pohledů na hlavní cíl podniku. Často je uváděn pohled dosažení zisku (Brealey, Myers and Allen, 2013). Tento přístup se ale s ohledem na odvod daní částečně transformoval do zvýšení hodnoty pro vlastníky (Kislingerová, 2013). Řada podniků ale může mít i jiné cíle, zejména pokud jsou vlastněny specifickou skupinou osob (například městem). Cíl v takovém případě může být poskytování služeb, krátkodobé zajištění zaměstnanosti apod. a zisk může být až druhotným faktorem. V některých případech se s tímto přístupem můžeme setkat i u rodinných podniků, které v době krize jinak přistupují k propouštění, neboť jsou zde i velké sociální vazby, které mohou mít v některých případech významnější dopad do lidského života než čistě pohled krátkodobé ekonomické ztráty. Z určitého pohledu bychom mohli konstatovat, že pokud bychom se na podnik dívali pohledem samotného podniku nikoli vlastníků, tak je jeho cílem přežít (Klieštik, Vrbka a Rowland, 2018) i za cenu krátkodobé ekonomické ztráty.

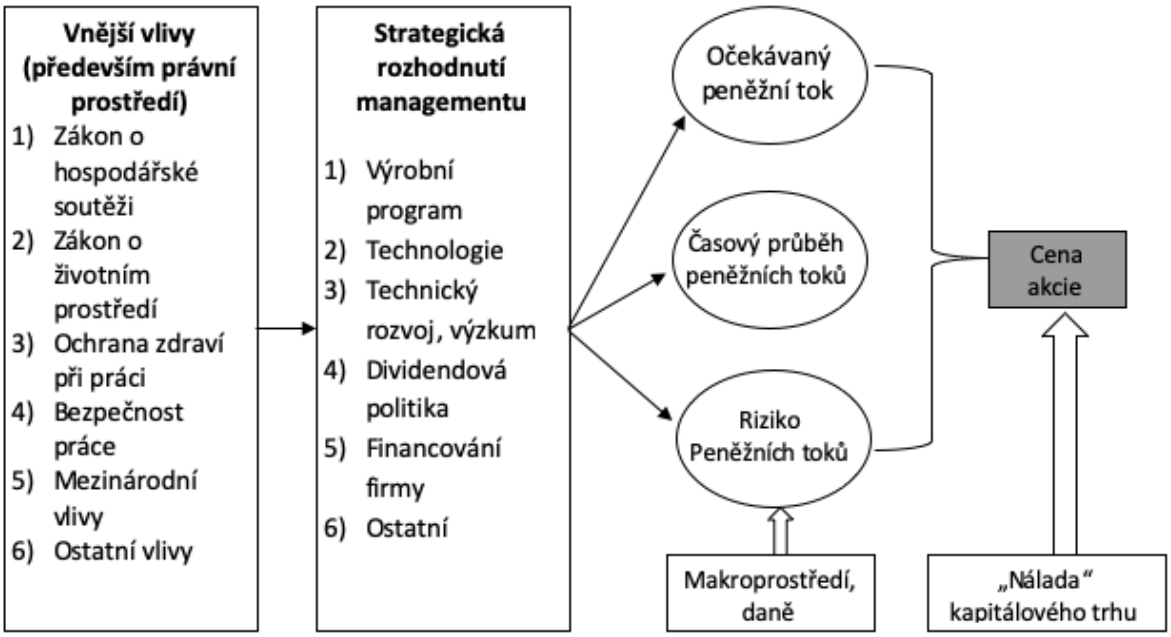
Každý z výše uvedených přístupů má svůj smysl, logiku a limity. Z určitého pohledu se zároveň výše uvedené pohledy prolínají právě v dosahování dlouhodobého a udržitelného zisku. Podmínka dlouhodobosti nám vymezuje limity v tom, že manažeři nemohou krátkodobě rozprodat veškerý hmotný majetek, aby měli mimořádný výsledek hospodaření, protože v následujícím období bude výkonnost jistě klesat, pokud podnik neskončí úplně.

Dlouhodobý pohled v sobě rovněž skrývá implicitně vyjádřenou možnost krátkodobého negativního vývoje, který může nastat v situaci, kdy by ekonomické zákonitosti mohly vést k propuštění nejbližšího příbuzného. Toto ale v konečném důsledku může mít negativní sociální vliv, který může překonat krátkodobou ztrátu.

Pokud bude podnik dlouhodobě dosahovat zisku, který bude podložen financemi (nebude se jednat jen o virtuální zisk, jako tomu bylo například u společnosti SAZKA před jejím krachem), bude tento podnik žít. Samozřejmě i tato má své další předpoklady.

Z jiného pohledu na danou problematiku poukazuje Kislingerová (2007), která předpokládá, že krátkodobé a dlouhodobé cíle se integrují v podobě hodnoty akcie, přičemž komplexnost tohoto ukazatele je vidět na následujícím obrázku. Z uvedeného obrázku je patrné, o jak komplexní ukazatel se jedná. Jeho komplexnost však v sobě zahrnuje i negativa. Například nálada na trhu ovlivňuje cenu akcie, a i když management společnosti postupuje ve všech bodech správně, tak tento postup může být paradoxně negativně hodnocen. Druhou a z pohledu cíle práce zásadnější povahou je nutnost dostatečné likvidity a velikosti podniku. Akciová firma, která nebude dostatečně likvidní z pohledu jejího obchodování na finančním trhu, nemusí být ovlivněna níže uvedenými faktory. V neposlední řadě toto měřítko nelze použít pro malé a střední firmy, které nejsou na finančních trzích obchodovány a často ani nemají povahu akciové společnosti.

Obrázek 1: Faktory ovlivňující cenu akcií



Zdroj: Kislingerová 2007.

3.2 Ekonomická přidaná hodnota ve variantě entity

V rešerši se dále zaměříme na klasický přístup, kdy budeme posuzovat výkonnost podniku, která vice či méně ovlivňuje hodnotu podniku. Hodnota podniku může být stanovena řadou metod (Mařík 2011). Jednou z velmi často používaných metod je EVA (ekonomická přidaná hodnota – Vochozka, 2011). Výše ukazatele EVA v sobě zohledňuje nejenom dosažený finanční výsledek, ale rovněž i riziko (Kislingerová, 2007) s jakým bylo tohoto výsledku

dosaženo (Neumaierová, 1998 a 2003). V literatuře je výpočet tohoto ukazatele realizován několika způsoby. Podle Kislingerové (2007).

$$EVA = NOPAT - C \cdot WACC \quad (1)$$

kde *NOPAT* je Net operating profit after taxes – čistý provozní zisk,

EVA – ekonomická přidaná hodnota,

C – kapitál poskytnutý za úplatu, (základní kapitál + dlouhodobé bankovní úvěry),

WACC – vážené průměrné náklady na kapitál.

Velmi významné je zde pojetí veličiny NOPAT. Nemělo by se zde jednat pouze o klasický provozní zisk, jak je to chybně používáno v řadě znaleckých posudků (osobní zkušenost autora práce), ale o provozní zisk, který je očištěn od jakýchkoli mimořádných vlivů, jako je prodej dlouhodobých aktiv či některé finanční operace, které nesouvisí s hlavní činností podniku (Kislingerová, 2007). Tento vzorec lze pak rozepsat jako:

$$EVA = EBIT \cdot (1 - t) - C \cdot WACC \quad (2)$$

kde *EBIT* – zisk před úroky a zdaněním,

t – sazba daně z příjmů.

WACC (vážené průměrné náklady na kapitál) představují náklady na celkově investovaný kapitál. Tyto náklady zohledňují jak náklady na ušlý zisk v podobě nákladů na vlastní kapitál, tak i náklady na cizí kapitál. Váhu zde představuje množství jednotlivých složek kapitálu. Ukazatel WACC se pak vypočítají jako:

$$WACC = \frac{E}{C} * r_e + \frac{D}{C} * r_d * (1 - t) \quad (3)$$

kde *E* je základní kapitál,

D – dlouhodobé bankovní úvěry,

r_e – alternativní náklady na základní kapitál**,**

r_d – náklady na cizí kapitál.

3.2.1 CAPM model

Stanovení hodnoty r_e, jež představuje alternativní náklady na vlastní kapitál, je často velmi obtížné, neboť se jedná o ušlý zisk za předpokladu stejné rizikovosti. K výpočtu alternativních nákladů na vlastní kapitál se používá model CAPM, který se vypočítá jako:

$$r_e = r_f + \beta(r_m - r_f) \tag{4}$$

kde r_e je očekávaná míra výnosnosti podílu,

r_f – bezrizikový výnos (stanovuje se na úrovni úrokové míry státních dluhopisů),

β – systematické riziko (riziko plynoucí z vývoje ekonomiky),

$(r_m - r_f)$ – prémie za riziko,

Model vychází z principu, který je zobrazen na následujícím obrázku. Jedná se o regresní model výkonnosti jednotlivých akciových titulů, kde křivka představuje ideální portfolio (investici). Křivka má rostoucí gradient, neboť za zvyšující se riziko (osa x) by měl následovat vyšší zisk (osa y). Koeficient beta pak představuje směrnici dané přímky, tedy to, jak moc je změna rizika spojená se změnou výnosu. Riziko je zde měřeno rozptylem. Z těchto důvodů se β vypočte jako:

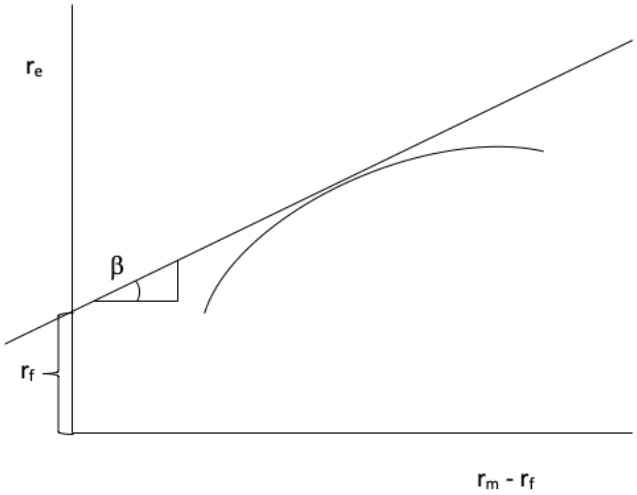
$$\beta = \frac{COV(r_i, r_m)}{\sigma_m^2} \tag{5}$$

Kde r_i je výnosnost i-té akcie,

r_m – výnosnost akciového indexu,

σ – směrodatná odchylka.

Obrázek 2: CAPM model



Zdroj: Vlastní tvorba dle Brealey, Myers, Allen (2013).

Nedostatky modelu jsou následující:

- předpoklad informační symetrie,
- předpoklad normálního rozdělení rizika,
- předpoklad racionálního investora,
- předpoklad nekonečně mnoho aktiv a bezrizikové investice.

Kromě výše uvedeného je zřejmé, že model pracuje s akciovým trhem a s aktivy, které jsou na tomto trhu likvidní a lze u nich měřit výkonnost a rizikovost. K výše uvedeným nedostatkům, které jistě nejsou kompletním výčtem, by autor práce proto doplnil i prakticky minimální využitelnost pro malé a střední podniky.

3.3 Ratingový model INFA

Alternativním modelem je ratingový model vyvinutý manžely Neumaierovými (2003). Jedná se o model, u kterého se dle ekonomické teorie analyzují alternativní náklady na kapitál. Jinými slovy – ekonomická přidaná hodnota vzniká tam, kde výnosnost přesahuje alternativní náklady na kapitál. Tato metoda je využívána i Ministerstvem průmyslu a obchodu ČR (2021) a má následující podobu.

3.3.1 Ekonomická přidaná hodnota pro vlastníky

V první fázi se vypočítají náklady na vlastní kapitál dle následujícího vzorce:

$$r_e = \frac{WACC * \frac{C}{A} - (1-t) * \frac{r_d}{D} * \left(\frac{C}{A} * \frac{E}{A}\right)}{\frac{E}{A}} \quad (6)$$

Kde *A* představuje celková aktiva,

E je vlastní kapitál,

D – dlouhodobé závazky,

r_d – náklady na cizí kapitál,

WACC – průměrné vážené náklady na kapitál.

Kde průměrné vážené náklady na kapitál se vypočítají jako:

$$WACC = r_f + r_{LA} + r_{business} + r_{FinStab} \quad (7)$$

Kde $r_{business}$ je podnikatelské riziko,

$r_{FinStab}$ – riziko finanční stability,

r_f – bezriziková sazba,

r_{LA} – riziko spojené s kapitálovou strukturou.

Ekonomická přidaná hodnota se nakonec vypočte jako:

$$EVA = (ROE - r_e) \cdot E \quad (8)$$

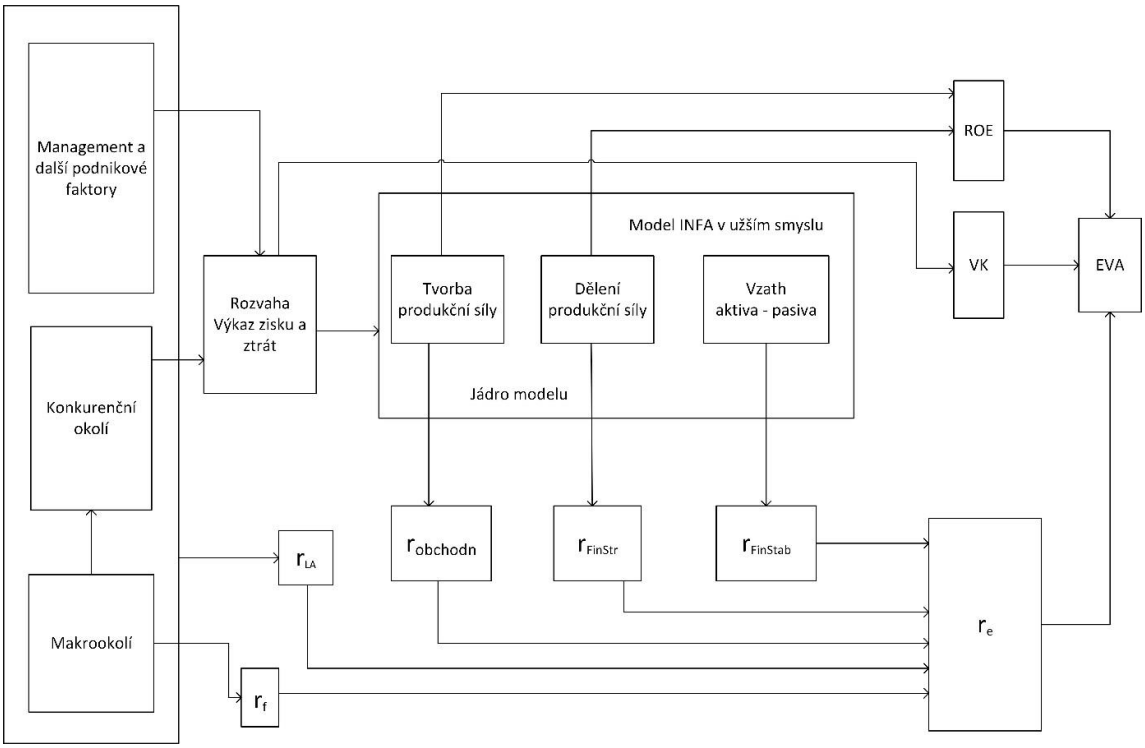
Kde *ROE* je návratnost vlastního kapitálu,

r_e – alternativní náklady na kapitál,

E – vlastní kapitál.

Princip výpočtu metody EVA dle ratingového modelu, který je využívaný MPO je vidět na následujícím obrázku. Z obrázku je patrný vliv jednotlivých složek na výsledek v podobě ekonomické přidané hodnoty. Ze schématu je zřejmé, že vše vychází z mikro a makro prostředí, kde mikroprostředím se rozumí prostředí, které bezprostředně ovlivňují osoby, které podnik řídí, a makrookolí je naopak podnikem neovlivnitelné. Makrookolí ovlivňuje přímo bezrizikovou sazbu, která vstupuje do výpočtu nákladů na vlastní kapitál (r_e). Společně s mikrookolím jsou ovlivňovány další faktory, které definují podnik, nebo konkurenční prostředí. Každý z těchto faktorů se do určité míry podílí na tvorbě hodnoty.

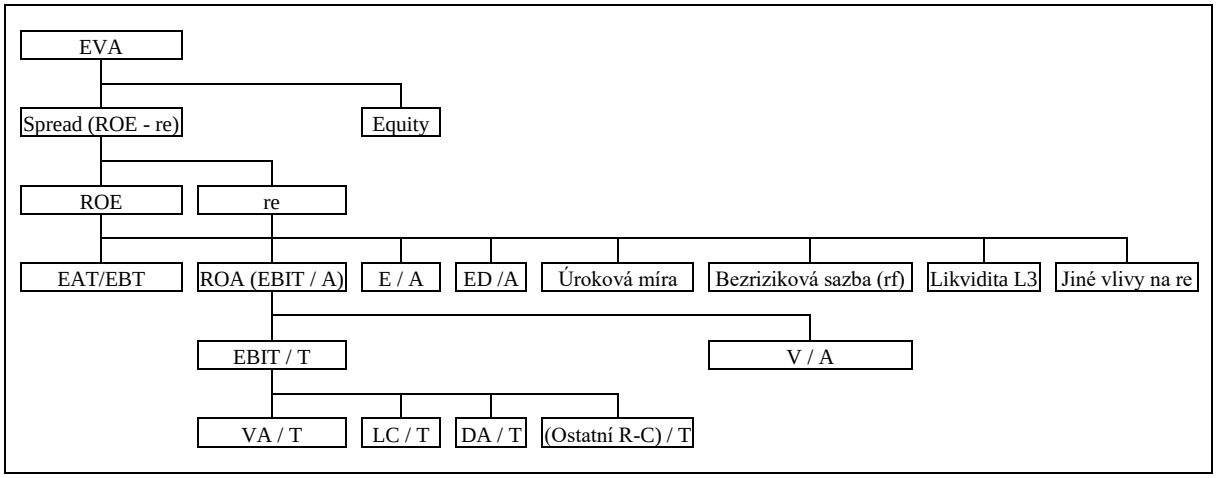
Obrázek 3: Princip EVA dle INFA



Zdroj: Neumaierová, I. (2003).

Výše uvedené schéma je principiální, a nelze proto z něho odvodit, jak například jednotlivé složky rozhodování manažerů ovlivňují EVA. K posouzení tohoto vlivu v obecné rovině bude s největší pravděpodobností vždy existovat nedostatek dat (obchodní tajemství). Vliv jednotlivých složek účetní závěrky je však možné snadno analyzovat, neboť každá společnost v ČR má povinnost zveřejňovat svoje účetní závěrky. Výsledek dekompozice parametru EVA je vidět na následujícím obrázku.

Obrázek 4: Rozklad hodnoty EVA na dílčí komponenty



Zdroj: Ministerstvo průmyslu a obchodu 2021.

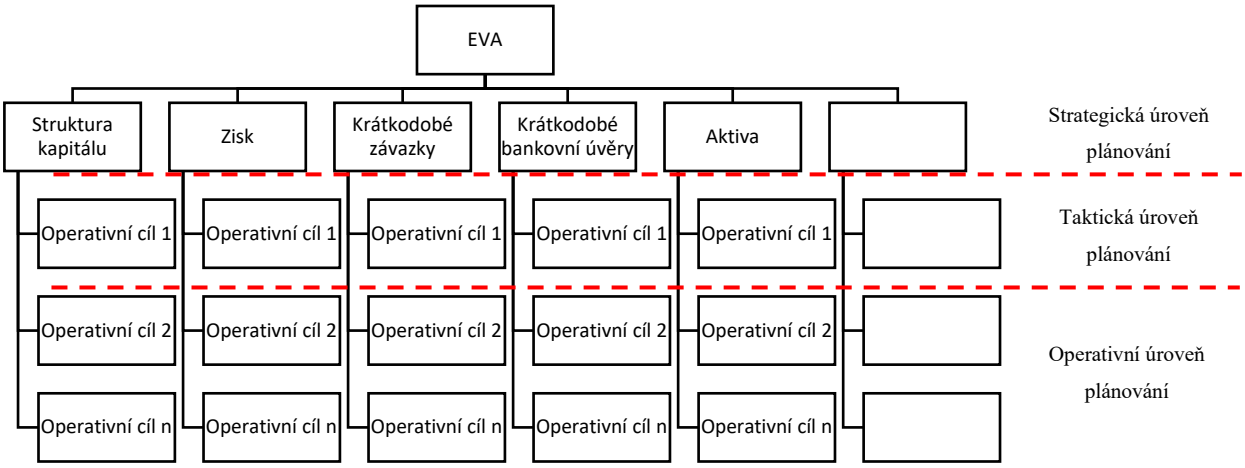
Velkou výhodou výše uvedeného modelu je možnost analyzovat, jak dané vlivy určují hodnotu EVA. Díky tomu byl i vyvinut benchmarkingový systém na základě kterého mohou jednotlivé podniky analyzovat svoji výkonnost a porovnávat se tak se zbytkem trhu v dané oblasti podnikání (sekce NACE). Aplikace přehledně a snadno spočítá, jak si podnik stojí a výsledek zobrazí dle příkladu níže:

Obrázek 4 analýza EVA za pomocí metody INFA.

Zdroj: Vlastní tvorba dle – Ministerstvo průmyslu a obchodu 2021.

Zelené složky představují pozitivní vliv na hodnotu EVA. Žluté složky mají neutrální vliv a červené složky mají negativní vliv na hodnotu EVA. Díky tomuto pohledu existuje pro podniky metodika, která jim umožňuje srovnat se s konkurencí v dané oblasti podnikání. V důsledku této skutečnosti je možné následně řídit dílčí cíle podniku takovým způsobem, aby postupně společnost dosahovala požadovaných výsledků EVA. Toto řízení lze realizovat dle rozpadu dílčích komponent, které je zobrazené na následujícím obrázku.

Obrázek 5: Řízení hodnoty na základě dekompozice cílů



Zdroj: Vlastní tvorba dle Stehel a Vochozka (2014).

3.4 Komparace používaných modelů

Při porovnání obou běžně používaných metod lze konstatovat že metoda CAPM je globálnější z pohledu celkového použití. Je parametrizovaná na větší společnosti a celosvětově byla podrobena mnohem větší diskuzi ve vědecké komunitě (Roll, 1977). Tato metoda však v sobě skrývá řadu nedostatků, díky kterým by její aplikace na malé a střední podniky nebyla příliš smysluplná.

Oproti tomu metoda INFA(Neumaierová a Neumaier, 2006) poskytuje výsledky, které jsou mnohem lépe interpretovatelné. Díky tomu je jejich využití pro běžného manažera je daleko snadnější. Metoda byla navíc vyvinuta v lokálních podmínkách, a proto v sobě bude již implicitně zahrnovat řadu skrytých vzorců typických pro Českou republiku.

Především při ocenění podniků se využívají nevíce výše uvedené metody (Mařík, 2011). Využití CAPM metody pro malý podnik na obci s pár obyvateli je zcela zjevně chybný postup, který dle názoru autora nelze ani korigovat tzv. expertními odhady v posudcích, jak se často děje, neboť tyto expertní odhady nejsou nikterak ověřitelné. Metoda INFA je oproti tomu mnohem více realisticky využitelná v Českém prostředí, ale oproti předchozí metodě zde stále chybí empirický výzkum, který by ve větší míře a s ohledem na další kategorie podniku, než na kterých byla metoda vyvinuta, ověřil, jak daná metoda funguje z hlediska determinace výsledků pro vlastníky. S ohledem na výše uvedený stav poznání toto bude hlavním zaměřením práce,

jejíž výzkumný cíl, respektive úkol bude zkoumán moderními metodami, které budou popsány níže.

3.5 Strojové učení a neuronové sítě

3.5.1 Bez učitele

Princip metod strojového učení bez učitele je založen na hledání skrytých zákonitostí a vztahů mezi daty (Mehryar, Rostamizadeh, Talwalkar, 2012). Daty je zde myšlena sada prediktorů, které nějakým způsobem ovlivňují výsledek. Například množství přihrávek a hodů na koš souvisí s tím, zdali je hráč útočník, nebo obránce. Lze předpokládat, že obránci ve shodné výkonnostní skupině budou mít podobné charakteristiky. Tímto dochází k přirozené segmentaci mezi jednotlivými skupinami hráčů.

V běžných úlohách se setkáváme s tím, že data mají různou váhu pro predikci. Jinými slovy jsou některá data velmi významná a jiná naopak mohou být zcela bezvýznamná. Čím více je prediktorů tím složitější jsou veškeré výpočty a tím může být i výsledek méně zobecňující. Je proto důležité redukovat celkové množství prediktorů. Pro redukci prediktorů se používají metody Multidimensional scaling (Mead, 1992) a Principal Component Analysis

3.5.1.1 Multidimensional scaling

Tato metoda nejdříve vypočítá párové vzdálenosti mezi jednotlivými měřeními (Seber, 1984). Vzdálenosti přitom mohou být různého druhu. Matlab může pracovat s následujícími verzemi (Matlab-pdist, 2021):

Euclidean distance

$$d_{st}^2 = (x_s - x_t)(x_s - x_t)' \tag{9}$$

Kde x_s a x_t jsou vektory v matici o rozměrech m x n

Standardized Euclidean distance

$$d_{st}^2 = (x_s - x_t)V^{-1}(x_s - x_t)' \tag{10}$$

Kde V je diagonální matice, kde každý j-tý faktor $(S_{(j)})^2$ představuje měřítko pro každou dimenzi.

Mahalanobis distance

$$d_{st}^2 = (x_s - x_t)C^{-1}(x_s - x_t)' \tag{11}$$

Kde C je kovariační matice.

City block distance

$$d_{st} = \sum_{j=1}^{n} |x_{sj} - x_{tj}| \tag{12}$$

Minkowski distance

$$d_{st} = \sqrt[p]{\sum_{j=1}^{n} |x_{sj} - x_{tj}|^p} \tag{13}$$

Chebychev distance

$$d_{st} = max_j\{|x_{sj} - x_{tj}|\} \tag{14}$$

Cosine distance

$$d_{st} = 1 - \frac{x_s x_t'}{\sqrt{(x_s x_s')(x_t x_t')}} \tag{15}$$

Correlation distance

$$d_{st} = 1 - \frac{(x_s - \bar{x_s})(x_t - \bar{x_t})'}{\sqrt{(x_s - \bar{x_s})(x_s - \bar{x_s})'}\sqrt{(x_t - \bar{x_t})(x_t - \bar{x_t})'}} \tag{16}$$

Kde $\bar{x_s} = \frac{1}{n}\sum_j x_{sj}$ a obdobně v případě $\bar{x_t} = \frac{1}{n}\sum_j x_{tj}$

Pro výpočet se použije příkaz:

```
d = pdist(measurements,distance)
```

Párové vzdálenosti představují vektor o velikosti všech hodnot nad diagonálou v matici porovnání vzdáleností. Pro lepší představu lze uvést příklad:

Prvním příkazem provedeme náhodné 4 pozorování o 3 proměnných

```
X = rand(4,3)

X =

   0.9572   0.4218   0.6557

   0.4854   0.9157   0.0357

   0.8003   0.7922   0.8491

   0.1419   0.9595   0.9340
```

V druhém příkazu provedeme párové porovnání.

```
D = pdist(X)

D =

   0.9225   0.4464   1.0155   0.8809   0.9627   0.6846
```

Třetí příkaz je uveden pouze pro lepší představu, jaké vzdálenosti předchozí vektor D vyjadřuje. Barevné hodnoty korespondují s předchozím příkazem:

```
Z = squareform(D)

Z =

         0   0.9225   0.4464   1.0155

   0.9225         0   0.8809   0.9627

   0.4464   0.8809         0   0.6846

   1.0155   0.9627   0.6846         0
```

Výstup předchozí funkce (pdist(X)) se následně využije pro hodnocení významnosti jednotlivých prediktorů.

```
[x,e] = cmdscale(D)
```

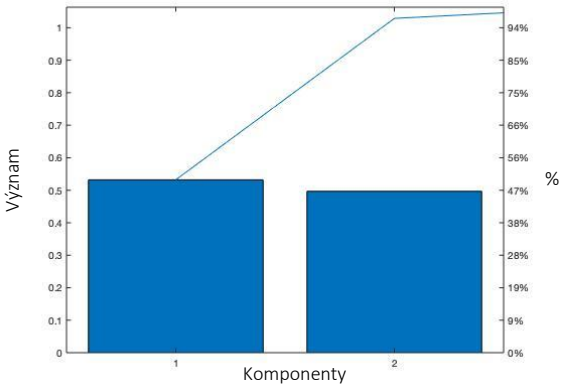
Proměnná x nám zredukuje matici Z na méně dimenzionální prostor, ve kterém je možné dosáhnout párových vzdáleností. Proměnná e představuje významnost parametru, pomocí kterého lze dosáhnout získání párových vzdáleností. Pokud jsou některé hodnoty parametru e vyšší než jiné, může dojít k redukci proměnných. V našem náhodně vygenerovaném případě

byly první dva prediktory významně důležitější než zbytek. Toto lze zobrazit a vyjádřit pomocí příkazu

```
pareto(e)
```

Výsledek je vidět na následujícím obrázku.

Obrázek 6: Významnost parametrů

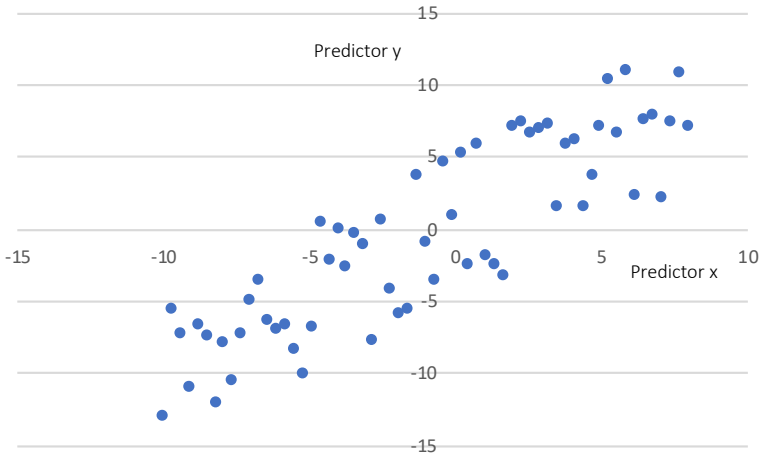


Zdroj: Vlastní tvorba dle Matlab documentation 2021.

3.5.1.2 Principal Component Analysis

Alternativním postupem k redukci prediktorů je Principal coponent analytsis. Tato metoda byla vyvinuta už začátkem minulého století (Pearson, 1901), následně byla rozvinuta a pojmenována (Hotelling, 1933) a slouží k dekorelizaci prediktorů. V některých případech je uváděná jako Karhunen-Loèveho transformace (Soummer, Pueyo, a Larkin, 2012), Hotellingova transformace, nebo jako singulární rozklad. V principu jde o přepsání dat v jiném souřadnicovém sytému. Důležitá je však normalizace dat, bez které by metoda poskytovala chybné výsledky (Abdi a Williams, 2010). Dále je důležité skóre komponenty, neboť ve většině případů má každá komponenta zcela jiný význam a v praxi tak může dojít k transformaci úlohy do méně dimenzionálního prostoru (Shaw, 2003). Pro lepší pochopení předpokládejme, že máme data o dvou proměnných, která jsou zobrazena na obrázku níže.

Obrázek 7: PCA – základní data



Zdroj: Stehel et al. 2021.

Pokud na tato data použijeme metodu PCA, měli bychom použít následující příkaz:

```
[pcs,scrs,~,~,pexp] = pca(Z)
```

První proměnná PCS odpovídá vymezení novému prostoru, výsledek je následující:

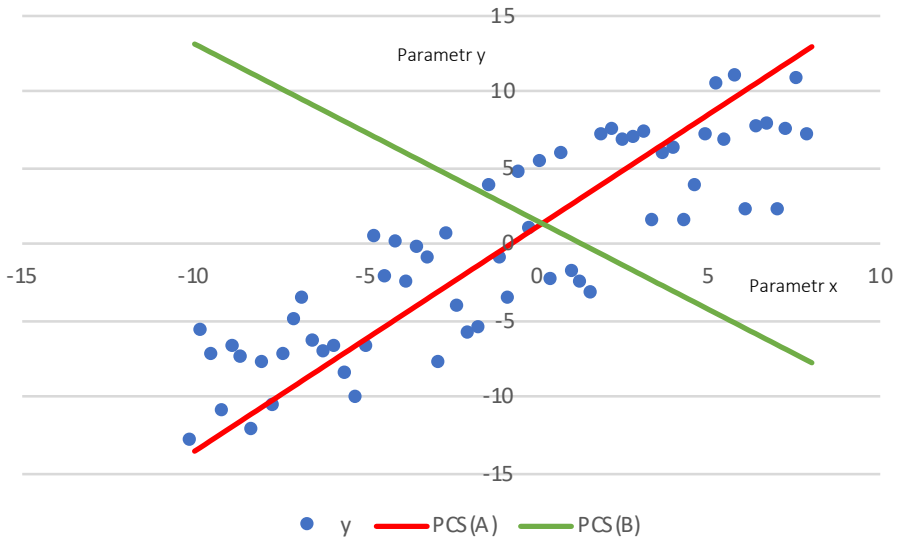
```
pcs =

   0.6223    0.7828

   0.7828   -0.6223
```

Čísla reprezentují směrnicové vyjádření vektoru přestavující novou dimenzi podle vzorce:

$$y = kx + q \qquad (17)$$

kde q je reprezentováno prvním číslem výsledku a k je reprezentováno druhým číslem sloupcového vektoru. Pokud dimenze zobrazíme získáme následující graf.

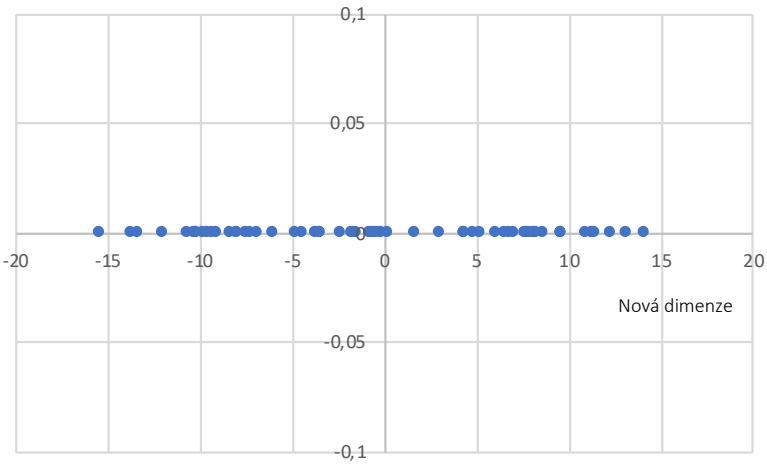
Obrázek 8: PCA – data se zobrazenými dimenzemi



Zdroj: Stehel et al. 2021.

Proměnná scrs představuje jednotlivé vzdálenosti bodů v novém prostoru a proměnná pexp představuje významnost kritéria. V našem případě představuje první dimenze 93 % významu predikce. Je tedy možné zobrazit výsledek rozdělení množiny bez druhé dimenze. Výsledek je možné vidět na následujícím obrázku.

Obrázek 9: PCA – první dimenze



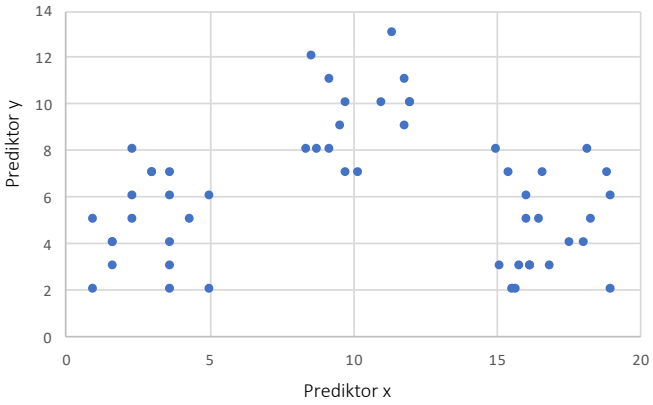
Zdroj: Stehel et al. 2021.

3.5.1.3 k-Means Clustering

Princip této metody je založený na tom, že vytvoříme několik typických reprezentantů dané množiny pozorování (Kriegel, Schubert, Zimek, 2016). Metoda se používá, jak pro učení

s učitelem, tak pro učení bez učitele (Coates, Andrew, 2012). Často se používá i v kombinaci s jinými metodami jako PCA (Zha et al., 2001). Pokud máme 3 shluky, pak pomocí této metody můžeme vytvořit 3 typické reprezentanty dané množiny, které budou typickými zástupci daných shluků a data z dané množiny je budou obklopovat. Příklad těchto shluků je vidět na následujícím obrázku. První metodu použil MacQueen (1967).

Obrázek 10: Příklad k-Means Clustering – základní data

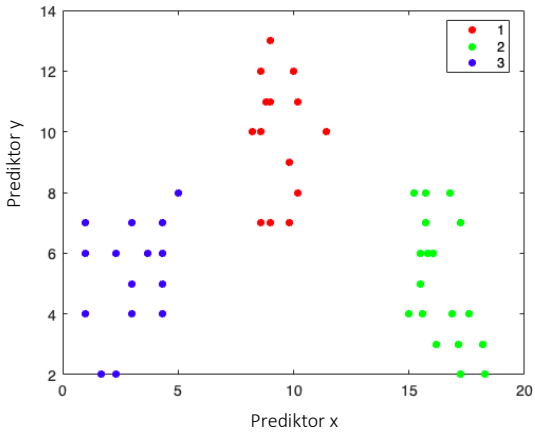


Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Pokud na shluky použijeme následující příkaz, tak získáme výstup zobrazený na následujícím obrázku.

```
g = kmeans(x,3);

gscatter(x(:,1),x(:,2),g)
```

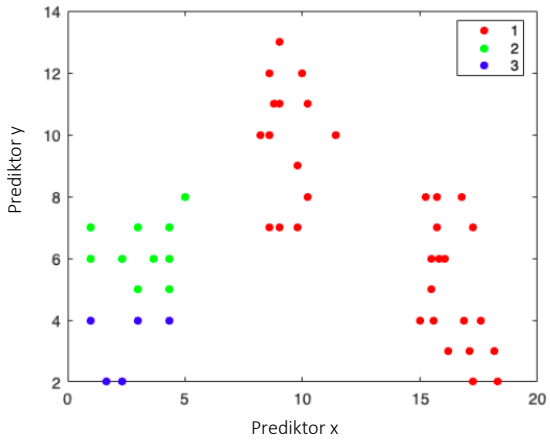
Obrázek 11: Rozdělení dat pomocí k-Means Clustering do shluků



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Nevýhoda této metody spočívá v počátečních souřadnicích typyckých reprezentantů, pro jejichž rozmístění se používá nejčastěji náhodné umístění (Hamerly a Elkan, 2002). Pokud jsou zvoleny špatně, může výsledek vyhodnotit střed mezi dvěma shluky jako střed samotné skupiny. Metoda totiž postupně interakčně nastavuje hodnotu typického reprezentanta dané skupiny a může skončit v situaci, která není z hlediska věcného posouzení zcela ideální (Pelleg, Moore, 1999). Běžně používanou metodou je náhodné určení rozmístění reprezentantů (Hamerly, Elkan, 2002). Výsledek chybně nastavených počátečních podmínek je vidět na následujícím obrázku.

Obrázek 12: Rozdělení dat do clusterů se špatnými inicačními podmínkami



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Aby se této skutečnosti předešlo, lze příkaz použít několikrát a následně vyhodnotit, jaký typ reprezentace je pro shluky nejlepší. K toumto se použije následující příkaz, kde červeně jsou vyznačeny nové parametry, které znamenají 5x opakování příkazu a následné porovnání výsledku:

```
g = kmeans(Z,3,'Replicates',5);
```

Pokud bychom navíc příkaz modifikovali tak, že výstup uložíme do dvou proměnných, pak druhá proměnná bude reprezentovat souřadnice typických zástupců shluků.

```
[g, C] = kmeans(Z,3,'Replicates',5);
```

V našem případě je výsledek:

```
C =

   9.3200   9.8667
```

```
16.5015   4.9500

 2.7772   5.1111
```

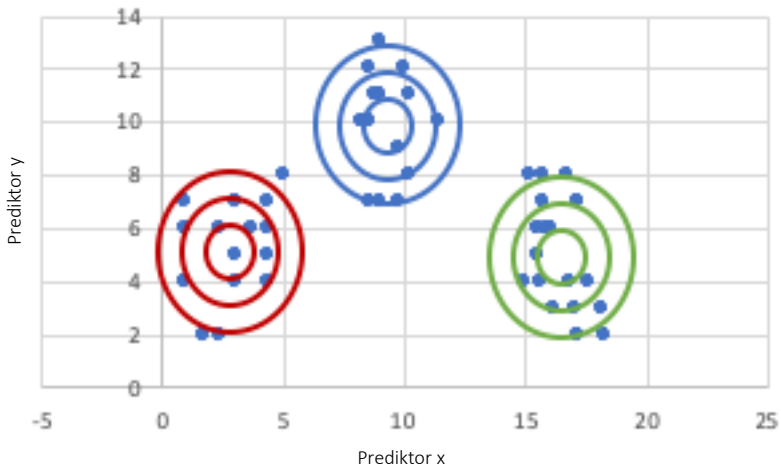
Jak je patrné ze souřadnic tří centrojdů (typický zástupce množiny), je výsledek opět rozložen dle množin, jak je patrné z vizuálního porovnání, které je možné, protože se jedná jen o dvourozměrný problém. V případě vícerozměrného shlukování, kde by bylo více nezávislých významných prediktorů, by tato vizualizace nebyla možná. Princip metody by však byl stále shodný.

3.5.1.4 Gaussian Mixture Models

Tato metoda je velmi podobná předchozí metodě (Press et al., 2007). Opět zde hledáme jednotlivé shluky. Na rozdíl od předchozího případu zde zohledňujeme pravděpodobnost, s jakou jednotlivá měření patří k určité množině. Toto je výhodné zejména pro hraniční hodnoty, kde můžeme snadno porovnávat, s jakou přesností a vypovídající schopností nám model poskytuje korektní data, která je nutné ověřovat dodatečnými metodami. Principy metody se zabýval již Pearson v 19. století (Améndola et al., 2015), ale až s příchodem moderní výpočetní techniky našla metoda uplatnění v řadě vědeckých oborů (Titterington, Smith, Makov, 1985).

Obrázek níže vyjadřuje pravděpodobnost začlenění do dané množiny. Čím je prvek vzdálenější, tím je pravděpodobnost nižší. Jak už název napovídá, model pracuje s Gaussovo rozdělením pravděpodobnosti.

Obrázek 13: Rozdělení pravděpodobnosti Gausian mixture model



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Pokud na data použijeme příkaz

```
mdl = fitgmdist(Z,3)
```

Získáme výstup

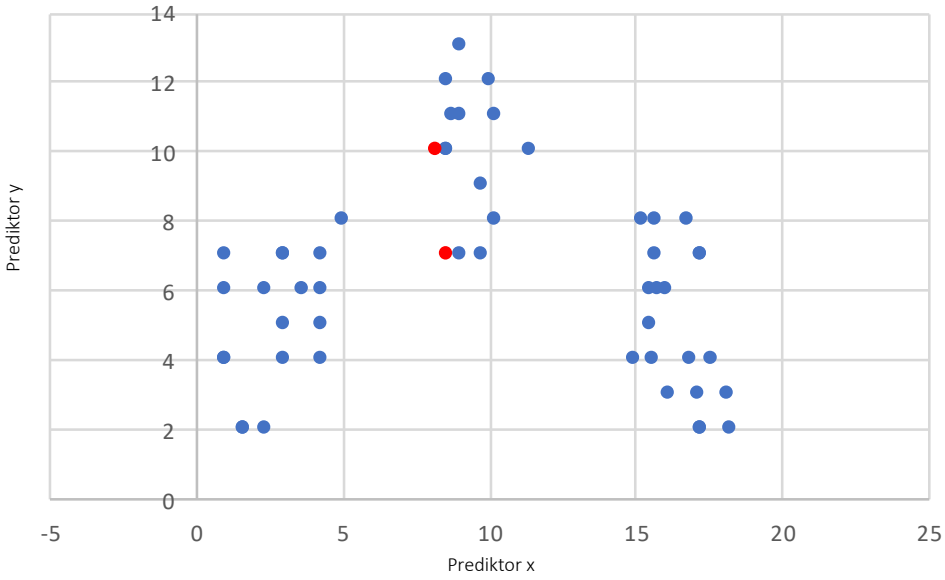
```
Gaussian mixture distribution with 3 components in 2 dimensions

Component 1:

Mixing proportion: 0.339633

Mean:    2.7774    5.1112

Component 2:

Mixing proportion: 0.377359

Mean:   16.5015    4.9500

Component 3:

Mixing proportion: 0.283008

Mean:    9.3200    9.8667
```

Z výstupu je patrné, jaká je střední hodnota daného shluku (hodnoty přibližně odpovídají typickému reprezentantu v předchozí metodě) a podíl zastoupení na celkové množině. V proměnné mdl jsou ovšem uloženy další hodnoty, jako je například směrodatná odchylka. Pokud budeme chtít získat začlenění jednotlivých měření do kategorií, pak musíme použít příkaz:

```
[g,~,p] = cluster(mdl,Z);
```

Do proměnné g se uloží rozdělení měření dle kategorií. Do proměnné p se uloží pravděpodobnost pro každé měření a danou kategorii. V našem případě byly téměř všechny měření klasifikovány na 100 % s výjimkou 2 měření, které byly klasifikovány na 99,98 % do kategorie 1 a na 0,0002% do kategorie 2.Tato měření jsou vyznačena na následujícím obrázku oranžovou barvou.

Obrázek 14: Klasifikace hraničních pozorování - Gausian mixture model

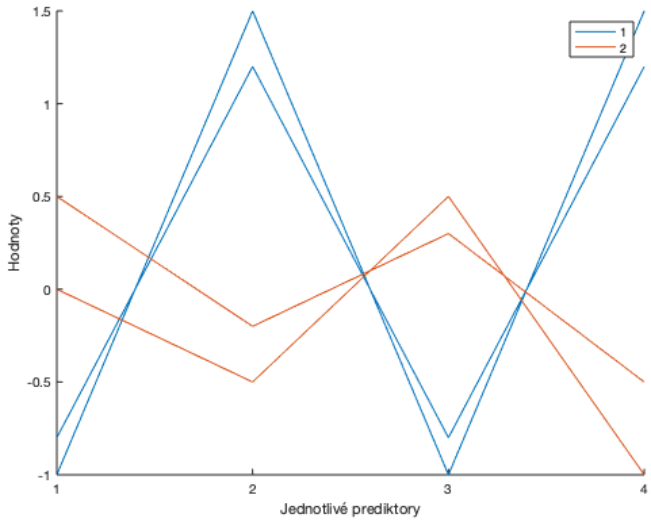


Zdroj: Vlastní tvorba dle Matlab documentation 2021.

3.5.2 Interpretace výsledků

Interpretace výsledků je ve vícerozměrných problémech složitá (Moustafa a Wegman, 2002). Pro tyto účely lze porovnat jednotlivé vlastnosti měřených objektů. Na ose x jsou vyneseny jednotlivé vlastnosti. Osa y obsahuje typické hodnoty dané vlastnosti (znormované). Příklad takového grafu je uveden na následujícím obrázku. Z tohoto obrázku je patrné, že v případě vlastnosti x3 jsou hodnoty skupiny 2 velmi nízké, zatímco v případě skupiny 1 jsou naopak velmi vysoké. Rovněž je patrné, že tato vlastnost bude dobře segmentovat jednotlivé skupiny na rozdíl od vlastnosti x2, kde se některé hodnoty napříč segmenty velmi přibližují a v rámci jednoho segmentu naopak poměrně vzdalují.

Obrázek 15: Paralel coordination příklad



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Pokud bychom chtěli zobrazit podobný graf v Matlabu, měli bychom použít příkaz:

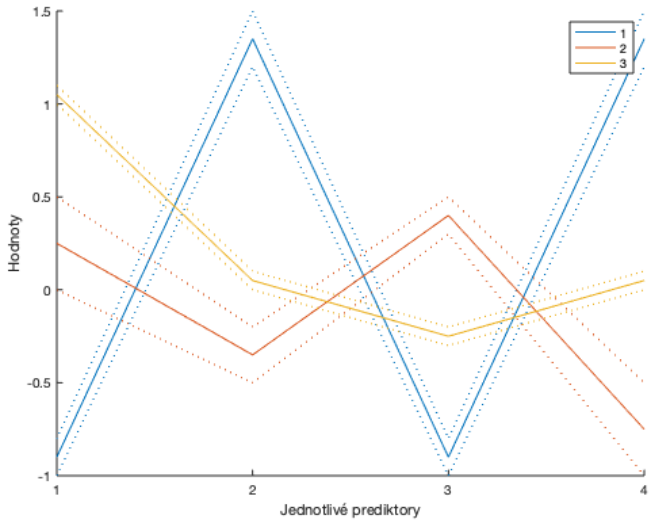
```
parallelcoords(X,'Group',grp)
```

Existuje ale i rozšíření daného příkazu, které nám zobrazí i další měření, respektive okraje prvního quartilu. Rozšíření příkazu vypadá následujícím způsobem

```
parallelcoords(C,'Group',grp,'Quantile',0.25)
```

Pro jiný typ úlohy by výsledek mohl vypadat, jak je zobrazeno na následujícím obrázku.

Obrázek 16: Zobrazení prvního quantilu



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

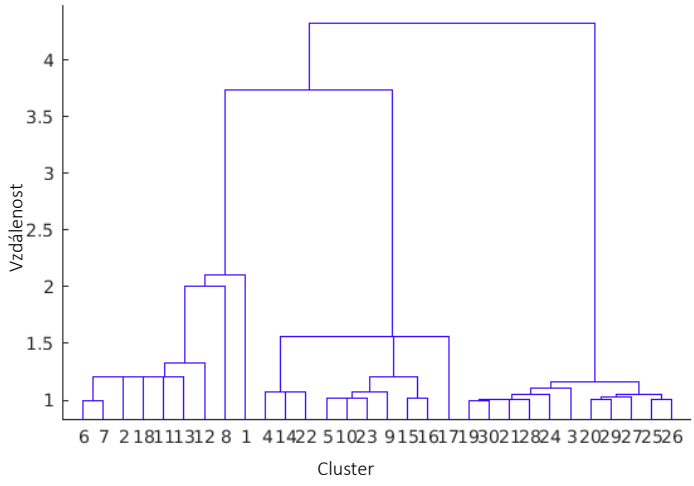
3.5.3 Hierarchical Clustering

Při učení bez učitele je obtížné (zejména u mnohodimenzionálních problémů) určit, kolik shluků chceme vytvořit. Jednotlivé determinanty mohou segmentovat skupiny do menších celků, které jsou pro nás příliš podrobné a naopak (Kaufman a Roussew, 1990). Díky této skutečnosti byla vyvinuta metoda, která hierarchicky klasifikuje jednotlivé shluky (Rokach a Oded, 2005). Nejdříve je porovnává na nižší úrovni a posléze shlukuje stále více obecněji až vytvoří výslednou celkovou množinu (Székely a Rizzo, 2005). Stejně jako v předchozích případech se pracuje se vzdáleností mezi jednotlivými prediktory, které mohou mít různé varianty (SAS 2019) a proto je důležitá normalizace dat. V Matlabu k vytvoření modelu použijeme funci linkage

```
K = linkage(Z);

dendrogram(K)
```

Další funkce dendogram pak vykreslí hierarchické rozdělení, jak je vidět na následujícím obrázku. Na obrázku osa x představuje jednotlivá měření, zatímco osa y představuje relativní vzdálenost mezi skupinami. Z obrázku je tedy zřejmé, že vzdálenosti mezi jednotlivými měřeními jsou relativně malé mezi 3 hlavními skupinami. Mezi těmito skupinami jsou ale vzdálenosti následně relativně veliké.

Obrázek 17: Příklad dendrogramu



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Rozdělení do skupin pak lze provést pomocí příkazu

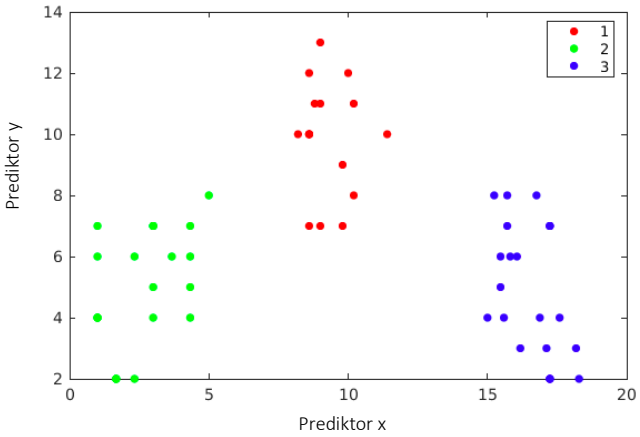
```
grp = cluster(K,'maxclust',3);
```

Poslední parametr vyjadřuje počet segmentů, do kterých chceme měření rozdělit. Pokud toto rozdělení vykreslíme pomocí příkazu

```
gscatter(X(:,1),X(:,2),grp)
```

získáme výsledek, který odpovídá i logice rozdělení dle předchozích metod. Výsledek tohoto zobrazení je vidět na následujícím obrázku, kde je množina bodů rozdělena do tří kategorií podle proměnných x a y.

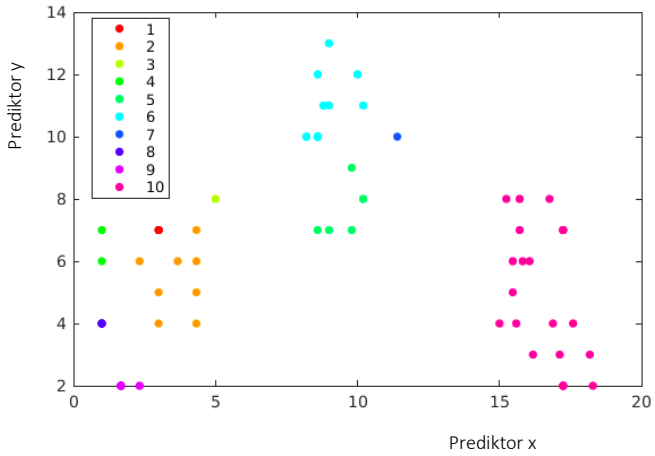
Obrázek 18: Rozdělení bodů do množin dle stromové klasifikace



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Pokud bychom použili v příkazu cluster parametr segmentu 10 a následně nechali výsledek vykreslit, tak se nám zobrazí graf, který je vidět na následujícím obrázku. Tento obrázek odpovídá dendrogramu viz obrázek výše. Třetí skupina má k sobě relativně blízko z pohledu hierarchické relativní vzdálenosti. Naopak první skupina má mezi některými měřeními relativně větší vzdálenosti. Toto vede k tomu, že první skupina je rozdělena na nejvíce segmentů, následuje druhá skupina rozdělená na 3 segmenty a poslední 3. skupina, která je v celku. Opticky by se mohlo zdát, že 1. a 3. skupina jsou si podobné co do vzdálenosti mezi jednotlivými body. Pokud ale vezmeme v úvahu celkovou vzdálenost bodu od počátku, tak v případě první skupiny existují body, které jsou od sebe vzdáleny ve stovkách procent své vzdálenosti od středu. Opačně je tomu u vzdálenější 3. skupiny. Zde relativní vzdálenosti mezi body nedosahují více jak desítek procent. Tuto skutečnost potvrzuje i 2 skupina, která je mezi oběma zmiňovanými skupinami. Relativní vzdálenosti jsou menší než v případě první skupiny, ale větší než v případě 3. skupiny. Toto odpovídá i relativní vzdálenosti skupiny od středu.

Obrázek 19: Rozdělení množiny do více skupin



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Kvalitu zobrazení vzdáleností můžeme ověřit za pomocí vypočítání párových vzdáleností mezi jednotlivými měřeními a následném provedení korelace výstupů. V případě, že je výsledek blízký hodnotě 1 je řešení velmi kvalitní.

```
Y = pdist(Z);

c = cophenet(Z,Y)

c =

   0.8352
```

V našem případě je výsledek přes 80 %.

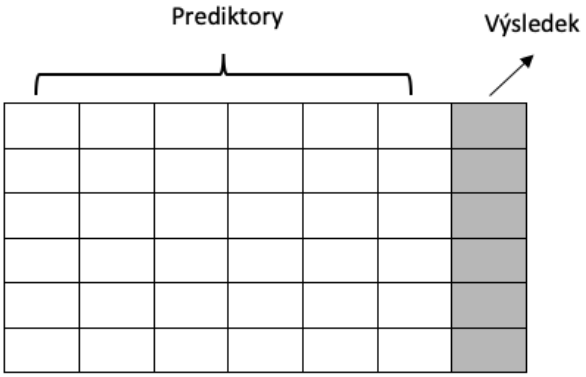
3.5.4 S učitelem

V případě učení bez učitele (Duda et al., 2001) jsme měli sadu dat a sledovali jsme jejich přirozené zákonitosti mezi jednotlivými charakteristikami (Hinton et al., 1999). Nebyl zde žádný vzor, a tak jsme data vyhodnocovali pouze na základě charakteristik (Bousquet, et al., 2004). V případě učení s učitelem (Russell a Norvig, 2010) je situace odlišná. Data jsou determinována výsledkem, kterého chceme dosáhnout. Nehledáme tak primárně přirozené zákonitosti mezi skupinami charakteristik, ale hledáme, co ovlivňuje výsledek, kterého potřebujeme dosáhnout (Mohri, Rostamizadeh, Talwalkar, 2012). Příklad struktury dat je zobrazena na následujícím obrázku.

Aplikace tohoto přístupu je možné vidět v řadě případů – například při měření na výrobní lince. Měření mohou spočívat od kvality materiálu, přes dobu výroby dílčí komponenty až po teplotu

okolí. Výsledek je nějaká součástka, která prošla kontrolou, nebo neprošla kontrolou. Řešený model tak může informovat, jaké skutečnosti mohou mít vliv na chybu ve výrobě. K těmto účelům se využívá řada metod, které jsou popsány níže (Smith a Martinez, 2011).

Obrázek 20: Struktura dat



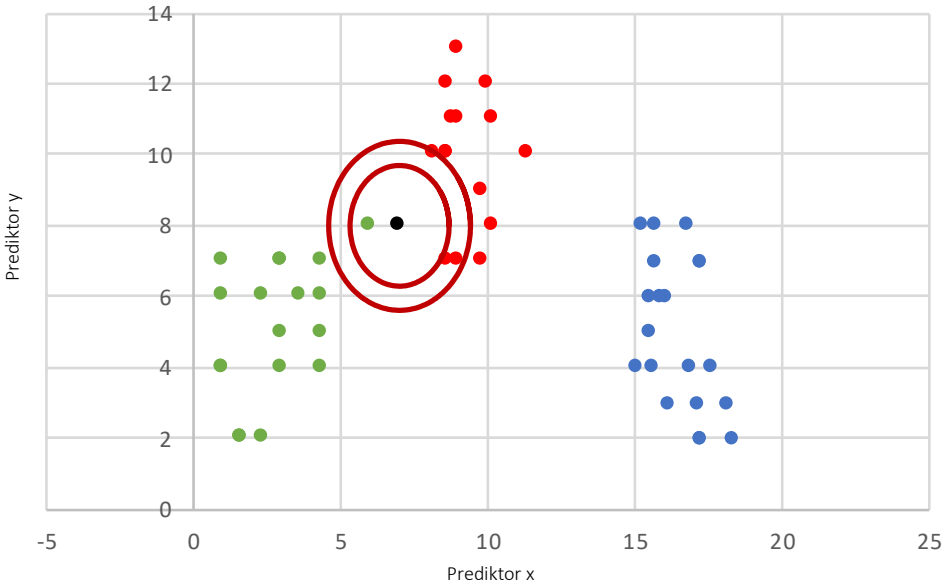
Zdroj: Vlastní tvorba dle Matlab documentation 2021.

3.5.4.1 Nejbližší soused

Metoda nejbližšího souseda se používá pro segmentování nebo regresi dat (Cover a Hart ,1967). Základní myšlenka spočívá v nalezení nejbližších sousedů (Beyer et al., 1999) a zařazení sledovaného objektu do kategorie objektů, které jsou mu nejblíže (Hall a Samworth, 2008). Obrázek níže zobrazuje situaci, kdy dochází ke klasifikaci neznámého objektu. Pokud je při klasifikaci nastavena pouze hodnota 1, pak se vyhledá pouze nejbližší zařazený soused a podle jeho kategorie se přiřadí i kategorie sledovanému objektu. V praxi je bohužel rozdělení dat někdy nejednoznačné a data se tak mohou překrývat nebo se v nich mohou objevovat šumy (Coomans a Massart, 1982). Pokud by prvním sousedem byl zrovna objekt, který představuje šum, mohlo by dojít k chybné klasifikaci. Z tohoto důvodu se využívá parametru, který určuje, kolik nejbližších sousedů se vyhledá, a výsledná kategorie je přiřazena podle objektů s nejvyšší shodou.

Princip této metody je zobrazen na následujícím obrázku. Pokud bychom v tomto případě použili pouze nejbližšího souseda, pak by byl klasifikovaný objekt zařazen do zelené kategorie. V případě, že model rozšíříme o druhou kružnici, tak již převládají červení sousedé a objekt tak bude klasifikován jako člen červené skupiny.

Obrázek 21: KNN - princip metody



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Pro klasifikaci využijeme příkaz:

```
mdl = fitcknn(dataTrain,'group','NumNeighbors',3)
```

Pro predikci pak

```
pg3 = predict(mdl,dataTest);
```

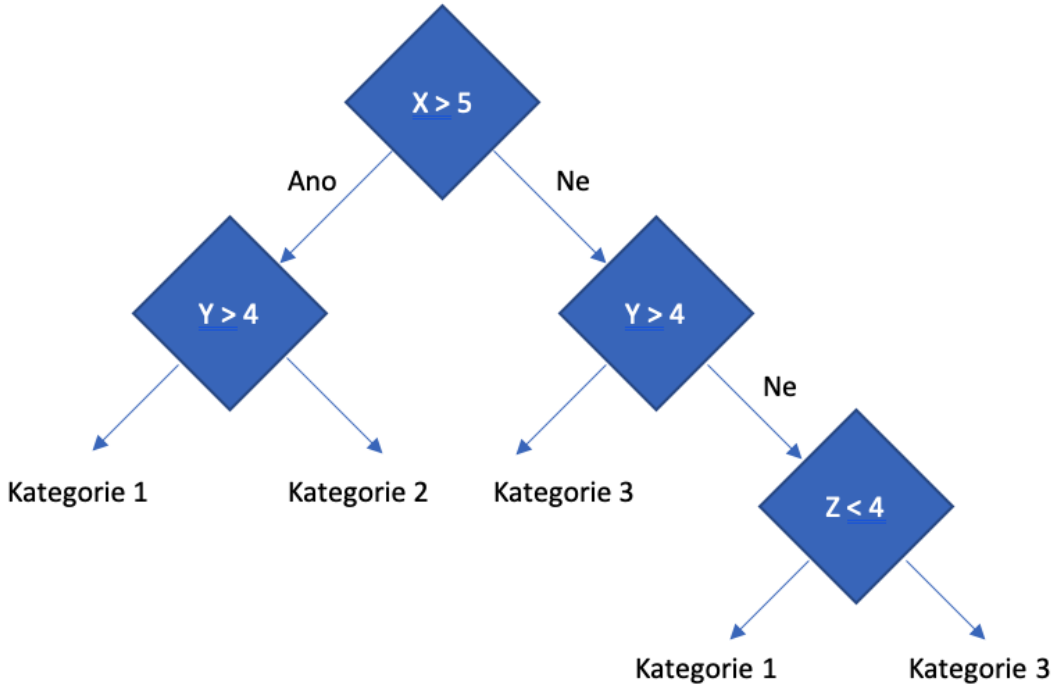
3.5.4.2 Klasifikační stromy

Klasifikační stromy jsou dalším typem algoritmu, který je schopen přiřadit sledovaný objekt do určitého shluku (Kamiński, Jakubczyk, Szufel, 2017). Na základě vstupních proměnných klasifikační strom rozdělí data na největší možné množiny, které posléze dále dělí (Utgoff, 1989). Následné přiřazení proměnné do určitého shluku je velmi rychlé, neboť se vyhodnotí jen několik podmínek (Quinlan, 1987).

Nevýhoda této metody spočívá v tom, že snadno může dojít k situaci, kdy model bude skvěle fungovat na trénovacích datech, ale nebude funkční pro testovací data. Důvodem je, že v reálu existuje řada šumů mezi hlavními kategoriemi. Pokud budeme model trénovat tak, že rozhodovací algoritmus bude 100% funkční na trénovacích datech, pak s největší pravděpodobností nebude funkční na testovacích datech. Obecně je proto lepší redukovat počet rozhodovacích kritérií. Díky tomu bude model lépe zobecnitelný a v praxi bude poskytovat lepší výsledky (Karimi a Hamilton, 2011).

Princip modelu je zobrazen na následujícím obrázku. Každé modré pole představuje rozhodnutí určitého prediktoru a podle tohoto prediktoru jsou pak jednotlivé objekty klasifikovány do určitých kategorií (v našem případě se jedná o 3 kategorie).

Obrázek 22: Rozhodovací strom



Zdroj: Vlastní tvorba.

Pro vlastní trénování dat se v Matlabu využije funkce

```
treeModel = fitctree(tableData, 'ResponseVariable');
```

Následně je možné strom zobrazit pomocí příkazu:

```
view(treeModel,'mode','graph');
```

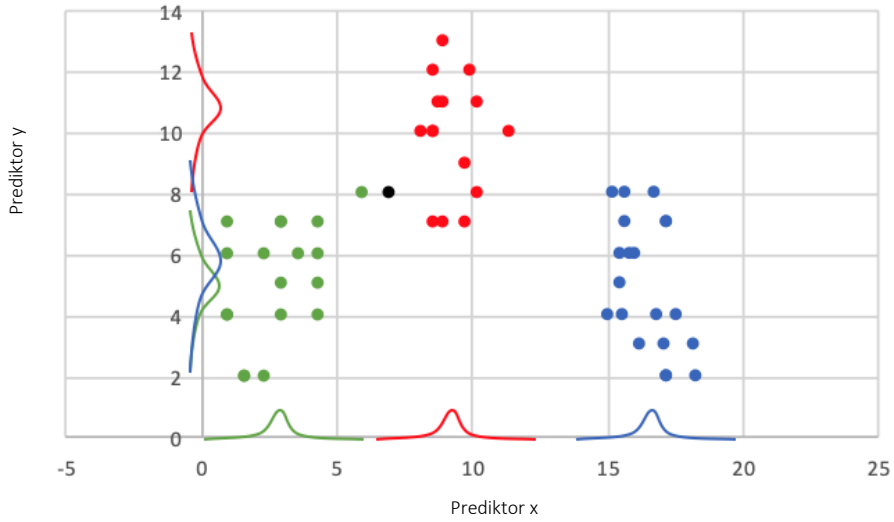
3.5.4.3 Naive Bayes klasifikace

Předchozí metody nebraly v úvahu rozdělení pravděpodobnosti výskytu proměnné ve skupině. Díky tomu byly oba modely snadno náchylné na šumy. Metoda Naive Bayes Classification je založená na tom, že sleduje rozložení pravděpodobnosti pro každý prediktor a předpokládá, že tyto prediktory jsou na sobě nezávislé (Rennie et al., 2003). Pokud chceme následně přiřadit jednotlivé měření do určité skupiny, pak pro každou skupinu vypočteme, s jakou

pravděpodobností je daný subjekt členem určité skupiny. Díky tomu je možné snadno určit, jak kvalitní je predikce pro dané měření (Rish, 2001).

Princip metody je zobrazen na následujícím obrázku. Zde je patrné, že existují tři skupiny. Každá skupina má rozložení pravděpodobnosti pro své zástupce na ose x (prediktor 1) a na ose y (prediktor 2). Objekt, který je následně analyzován a klasifikován, je podroben testu výpočtu pravděpodobnosti pro všechny skupiny. Již z pozorování je patrné, že zelená skupina není pro daný bod relevantní. Otázkou je však přiřazení ke žluté skupině. Na základě rozložení pravděpodobností je zřejmé, že zařazení k modré skupině je 1,6 x pravděpodobnější.

Obrázek 23: NB - princip metody



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

V Matlabu využijeme k vytvoření modelu příkaz

```
mdlNB = fitcnb(dataTrain,'group');
```

Pro predikci pak využijeme funkci

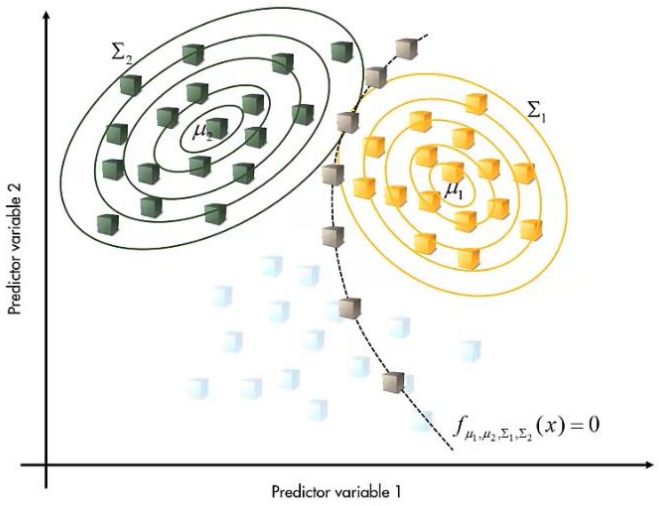
```
predictedGroups = predict(mdlNB,dataTest);
```

3.5.4.4 Discriminant analýza

Diskriminant analysis opět počítá rozdělení pravděpodobností jako tomu bylo v předchozí metodě. Na rozdíl od Naive Bayes Classification však nepředpokládá nezávislost jednotlivých prediktorů, naopak počítá vícerozměrné rozložení bodů (McLachlan, 2004). Mezi jednotlivými množinami pak vytváří hranice, ve kterých se hodnota pravděpodobnosti mezi jednou a druhou

množinou rovná (William, 1980). Výhoda této metody spočívá ve velké robustnosti oproti šumům (Demir a Ozmehmet, 2005). Toto pramení z principu sledování rozdělení pravděpodobnosti. Robustnost je ještě větší díky mnohadimenzionálnímu pohledu na rozložení bodů. Nevýhodou je větší obtížnost výpočtu při vícerozměrných problémech. Princip metody je zobrazen na následujícím obrázku.

Obrázek 24: Diskriminační analýza princip metody



Zdroj: Matlabacademy 2021.

Vytvoření modelu je možné udělat základním způsobem, který je lineární. V takovém případě budou hranice mezi skupinami lineární. Výsledný výpočet je pak rychlejší, neboť se předpokládá, že kovariace na hranicích jsou shodné. V takovém případě se použije příkaz

```
daModel = fitcdiscr(dataTrain,'response');
```

Pokud opustíme předchozí předpoklad o rovnosti kovariací vytvoříme kvadratické hranice mezi jednotlivými třídami. Toto bude přesnější a bude to více odpovídat obrázku výše. Bude to však rovněž náročnější na časové zpracování výpočtu:

```
daModel = fitcdiscr(dataTrain,'response','DiscrimType','quadratic');
```

Predikce se následně provede stejně jako v předchozích případech pomocí funkce

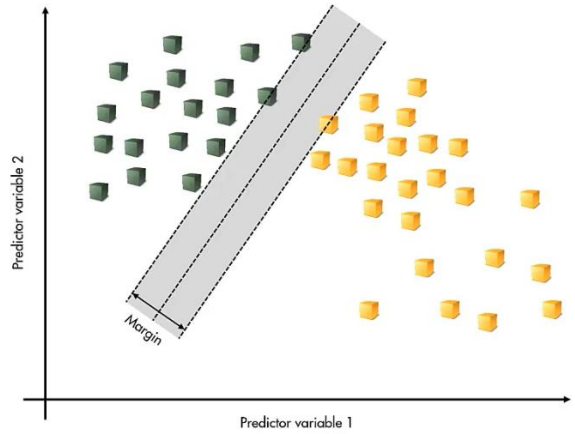
```
predictedGroups = predict(daModel,dataTest);
```

3.5.4.5 Support Vector Machines

Support Vector Machines je metoda, která rozdělí data na 2 segmenty. Základní myšlenka spočívá v rozdělení množin tak, aby dělící vektor byl co nejvíce vzdálený od nejbližších subjektů (Cortes a Vapnik, 1995). Tento princip může být výhodnější tam, kde je velký rozptyl

nebo velké množství odlehlých hodnot v rámci jedné skupiny (Boser, Guyon, Vapnik, 1992). Tyto odlehlé hodnoty by mohly vést k vytvoření posunutých hranic vlivem těchto objektů. Metoda SVM je pak vhodnější pro použití, neboť tyto odlehlé hodnoty nemají na výsledek vliv (Christianini a Shawe-Taylor, 2000). Princip je zobrazen na následujícím obrázku.

Obrázek 25: Princip metody SVM



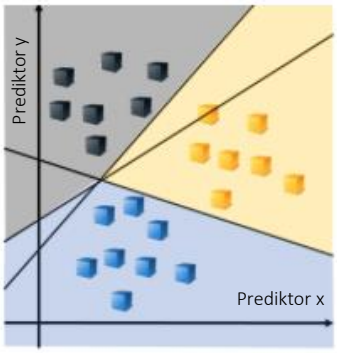
Zdroj: Matlabacademy 2021.

Vytvoření modelu probíhá za pomocí následujícího příkazu.

```
svmModel = fitcsvm(tableData, 'ResponseVariable')
```

Nevýhodou SVM metody je, že metoda umí rozdělit pouze dvě množiny. Pokud bychom potřebovali rozdělit více množin, je nutné využít Multiclass SVM. Princip metody je stále shodný jako u klasické SVM. Metoda tedy vytvoří jednotlivé vektory pro rozčlenění dat mezi páry shluků. Následně jsou výsledné vektory složeny do jednoho celku, který je již schopen klasifikovat celé řešení. Toto je zobrazeno na následujícím obrázku.

Obrázek 26: Princip MSVM – agregace



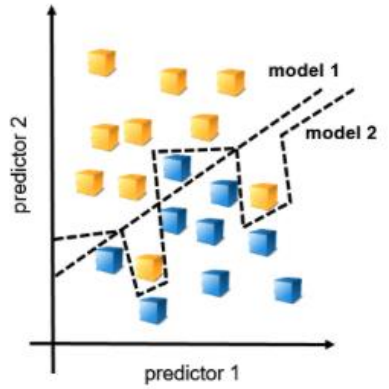
Zdroj: Matlabacademy 2021 - upraveno.

Pro vytvoření MSVM použijeme následující příkazy:

```
template = templateSVM('PropertyName',PropertyValue)

ecocModel = fitcecoc(dataTrain,'y','Learners',template)
```

3.5.5 Hodnocení výsledků

Interpretace výsledků a hodnocení výkonnosti modelu je velmi důležitý krok. Pomocí dat a nastavení parametru učení můžeme dojít u stejné metody k různým modelům, které budou predikovat hodnoty s odlišnou kvalitou výstupů. Příklad různých modelů je zobrazen na následujícím obrázku.



Zdroj: Matlabacademy 2021.

Model 1 je jednoduchou přímkou dělící data na dvě kategorie žluté a modré barvy. Toto rozdělení není dokonalé, neboť dva žluté objekty pod přímkou by byly chybně klasifikovány jako modré a jeden modrý objekt nad přímkou by byl naopak klasifikován jako žlutý. Toto může být způsobeno šumy, nebo může jít o nedostatečně natrénovaný model, který nezohledňuje všechny prediktory. Oproti tomu model 2 klasifikuje všechny hodnoty správně. Zdánlivě je tedy model 2 lepší. Ve skutečnosti ale může model fungovat jen na datech, které jsou mu poskytnuty pro vytvoření modelu. Pokud má ale model fungovat i pro další klasifikaci objektů, tak může poskytovat horší predikce než model 1. Tomuto efektu se říká přetrénování a jedná se o častou chybu začátečníků, kteří se snaží vytrénovat dokonalý model, který v praxi zcela selhává.

Situace může být i zcela opačná a model 2 může být kvalitnější. Aby bylo možné posoudit, jak je model kvalitní, rozdělují se data na trénovací a testovací. Rozdělení lze provést na základě příkazů

```
cvpt = cvpartition(groupData.group,'holdout',0.35);

dataTrain = groupData(training(cvpt),:);

dataTest = groupData(test(cvpt),:);
```

Jak již název napovídá, na trénovačích datech se model vytvoří a na testovacích datech se ověří jeho reálná výkonnost. Následně je možné poměrově posoudit počet správně a špatně zařazených objektů. K tomuto účelu jsou v Matlabu vytvořeny dvě funkce

```
trainErr = resubLoss(mdl)
```

Tato funkce vrátí informaci o chybně zařazených proměnných na trénovačích datech. Principiálně je tato hodnota nižší než v případě dalšího příkladu, kterým je

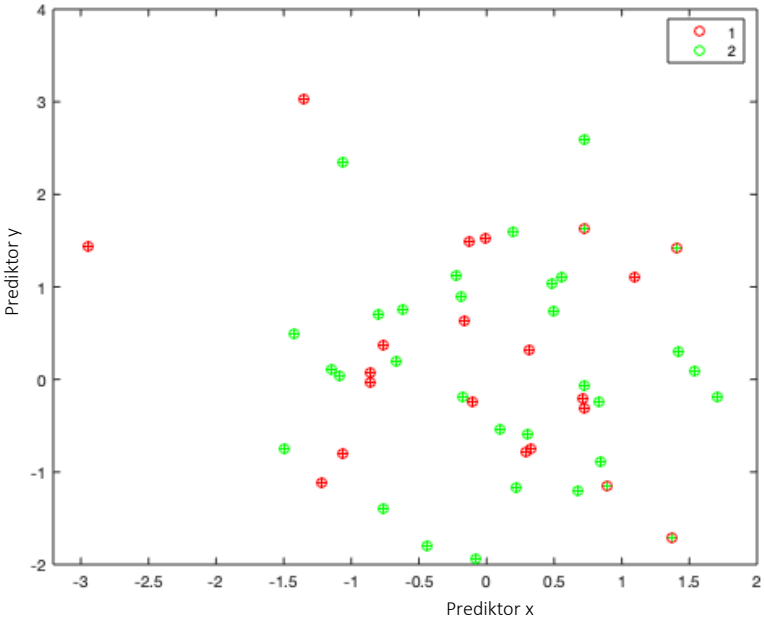
```
testErr = loss(mdl,testResponse)
```

Tento příkaz vrátí informaci o chybně zařazených proměnných na testovacích datech.

Další možností, jak ověřit, zdali je naše predikce kvalitní, je vykreslení dvou grafů přes sebe. První graf bude vykreslen na základě naměřených hodnot. Tyto hodnoty budou zobrazeny křížkem. Dále budou vykreslena predikovaná data, která budou vykreslena kolečkem. Tam kde se budou barvy shodovat, došlo ke korektní predikci. Objekty, kde kolečko chybí, jsou trénovací data a objekty, kde jsou barvy křížku a kolečka odlišné, jsou chybné klasifikace. Pro tyto účely využijeme následující kód. Výsledek je zobrazen na následujícím obrázku.

```
plotGroup(groupData,'x')

hold on

plotGroup(dataTest,predictedGroups,'o')

hold off
```

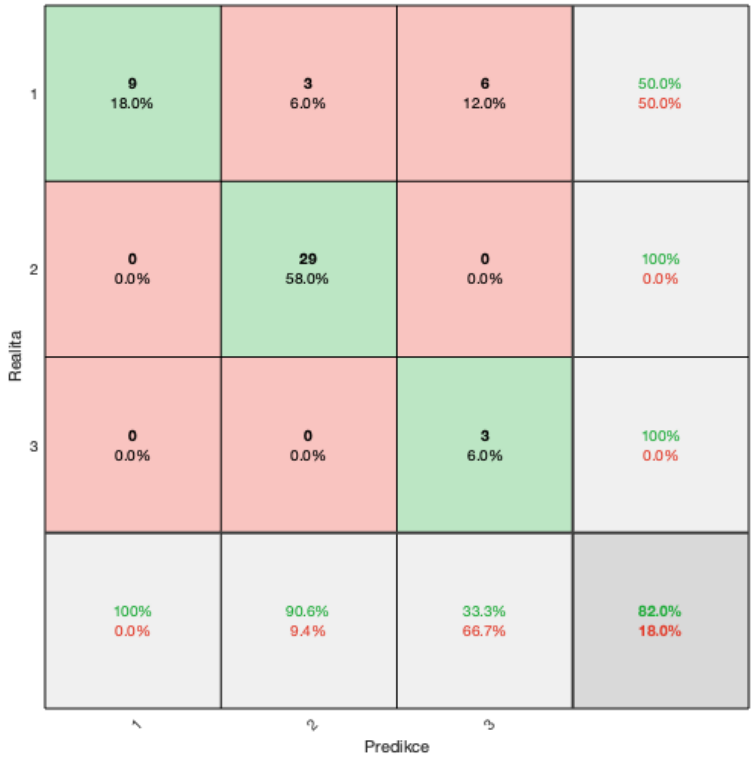
Obrázek 27: Vizualizace predikce



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Jednou z nejčastěji používaných metod je konfuzní matice. Tato matice nám ukazuje, jak dobře jsme zatřídili jednotlivá pozorování. Díky metodě zobrazení je možné porovnat kvalitu predikce i pro samostatné kategorie. Toto je důležité zejména v situacích, kdy váha chybně zařazené predikce nemusí být pro různé kategorie shodná. Například pokud budeme predikovat riziko infarktu a model zařadí zdravého člověka jako rizikového, tak chyba bude mít menší váhu než v případě, kdy bude člověk s vysokým rizikem zařazen jako zdravý. Příklad konfuzní matice je zobrazen na následujícím obrázku.

Obrázek 28: Konfuzni matice



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Konfusní matici vytvoříme příkazem:

```
[cm,grp] = confusionmat(yObserved,yPred)
```

V takovém případě jsou na ose y zobrazeny skutečné hodnoty a na ose x predikované hodnoty. Jinými slovy – v předchozím případě byl první prvek správně zařazen 9x a 3x byl zaražen jako prvek 2 a 6x byl chybně zařazen jako prvek 3.

3.5.6 Regrese

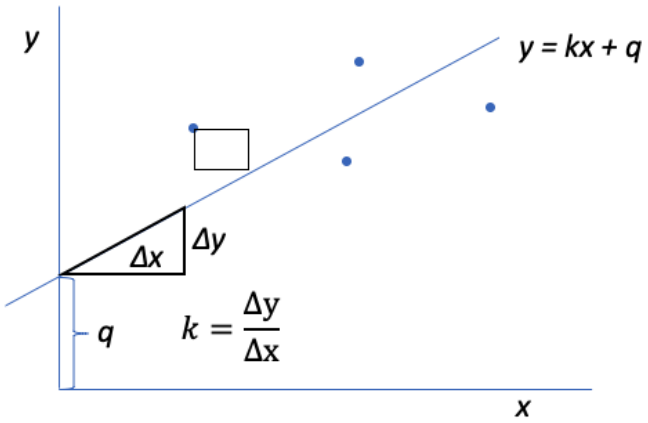
V předchozích dvou kapitolách jsme se věnovali analýze skrytých vzorců, které vedou ke klasifikaci, tedy rozčlenění dat, do určitých tříd. Další typ úloh může být spojen s hledáním číselné hodnoty dané veličiny za určitých podmínek (Freedman, 2009). Jako příklad můžeme sledovat dojezd elektromobilu v závislosti na rychlosti jízdy, kapacitě baterie, terénu apod. Základní princip regresní metody (Fisher, 1954), kterou je možné pro tento typ úlohy použít, je zobrazen na následujícím obrázku. Na tomto obrázku je vidět, že existuje několik objektů (modré tečky). Objekty jsou zobrazeny v souřadnicovém systému, kde x je nezávislá proměnná a y je závislá proměnná (Mogull, 2004). Pro příklad můžeme uvést, že náklady budou záviset na výši tržeb. Náklad bude tedy závislá proměnná a tržby nezávislá proměnná. Na základě

historických dat můžeme tyto hodnoty (výši nákladů a výnosů) zanést do grafu. Pokud budeme chtít v budoucnu odhadnout výši nákladů na základě naplánovaných tržeb, můžeme využít právě regrese (Freedman, 2009), která nám určí, jak y závisí na x. V případě lineárního modelu, který je zobrazen níže, je výsledkem přímka ve směrnicovém tvaru.

$$y = kx + q \tag{188}$$

Kde y je závislá proměnná,
k – směrnice přímky,
x – nezávislá proměnná,
q – parametr odpovídající posunu křivky po ose y.

Obrázek 29: Příklad regresní křivky



Zdroj: Vlastní tvorba.

Výpočet parametrů přímky, která vyjadřuje funkční závislost, je na základě minimalizace hodnoty obsahu čtverců, které jsou vyneseny nad jednotlivými body od přímky (jak je naznačeno na obrázku výše v případě prvního bodu). Jednotlivé parametry se vypočítají na základě vzorců.

$$k = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2} \tag{19}$$

$$q = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{n\sum x_i^2 - (\sum x_i)^2} \tag{20}$$

Kde x_i a y_i představují jednotlivé body (objekty).

Regrese nemusí být pouze lineární (Rao, Toutenburg et al., 2008), ale data zle proložit i jinými křivkami. O tom, zdali je regrese lineární nebo nelineární, rozhoduje výpočet parametrů k a q (Hastie, Tibshirani, Friedman, 2009).

V obecné rovině je vzorec pro lineární regresi

$$y = \sum k_i f_i(x_1, x_2) \tag{21}$$

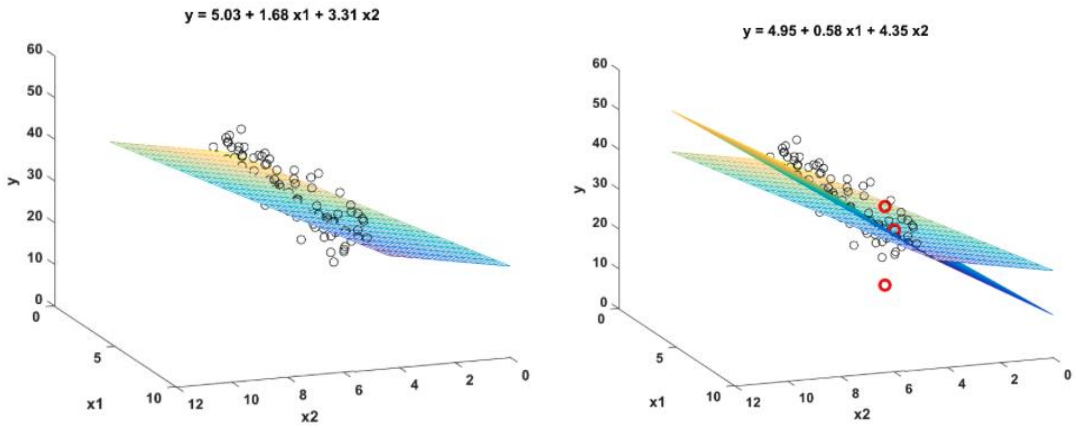
Kde x_1, a x_2 jsou nezávislými proměnnými ovlivňující závislou proměnnou y. V Matlabu vytvoříme lineární regresi za pomocí příkazu

```
mdl = fitlm(data,modelspec)
```

Kde data musí být uspořádána v tabulce, přičemž posledním sloupcem je závislá proměnná. V proměnné modelspec můžeme definovat, jakou funkci použijeme pro predikci ('linear', 'quadratic', …).

Pokud budeme tvořit regresní model, který bude mít více prediktorů, můžeme se dostat do problémů s tím, že model bude zbytečně komplikovaný a výrazně ovlivnitelný šumem. Obrázek níže demonstruje, jak se změní regresní funkce v případě, že se změní několik málo proměnných ve vícerozměrném modelu. V levé části je původní model, v pravé části je nový model se zvýrazněnými změnami.

Obrázek 30: Změna parametrů regrese při změně několika málo dat



Zdroj: Matlabacademy 2021.

3.5.7 Neparametrické metody

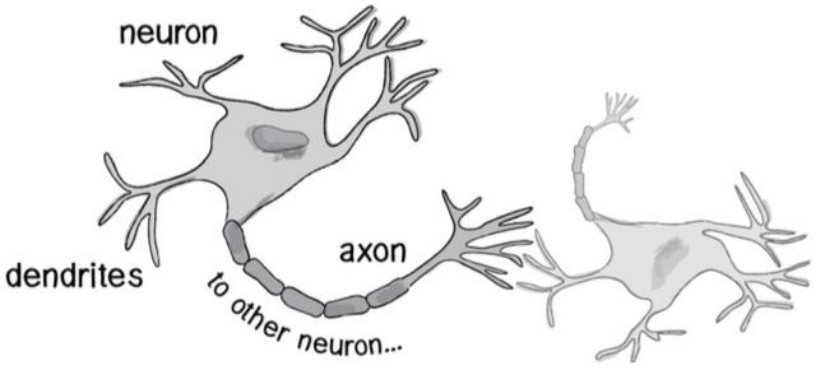
Parametrické modely jsou založeny na vzorci, díky kterému je možné predikovat a snadno interpretovat výsledek. Rovněž je možné provést citlivostní analýzu změny dopadu jednotlivých složek na výslednou hodnotu. Nevýhodou těchto modelů je, že v případě mnoha dimenzionálních úloh je obtížné nalézt vhodné vzorce, které by měly odpovídající výsledky. Z výše uvedených důvodů existují i metody neparametrické, které mohou u mnoha nezávislých proměnných, jež determinují výsledek, lépe predikovat výstup, ale na druhou stranu zde není přesný vzorec jejich chování. Díky tomu je výstup hůře interpretovatelný a rovněž citlivostní analýza je složitější. Jako neparametrické modely lze použít metody SVM a Tree, které byly popsány výše.

3.6 Neuronové sítě

Neuronová síť je algoritmus, který je schopen zpracování dat (Šťastný, 2014). Systém zpracování je inspirován biologickými neuronovými sítěmi (Widrow et al., 2013). Díky tomuto přístupu je možné řešit formy zpracování dat, které by byly pro počítače mimořádně složité. Nejčastější využití neuronových sítí je v oblasti rozpoznávání obrazu (Abbod, 2007), akustice (Sak, Senior, Beaufays, 2014), řízení automobilů (Zissis a Dimitrios, 2015), chemii (Balabin a Lomakina, 2009) predikce časových řad (Miljanovic, 2012), detekce anomálií (Ghosh a Reilly, 1994), v lékařství (Fiala, Karhan, Ptáček 2014) apod.

Základy neuronových sítí byly položeny v polovině minulého století vědci Warrenem S. McCullochem a Walterem Pittsem (1943). Ve své práci popsali, jak jedna buňka zpracovává signály a generuje výstup. Takto navržený model se později doplňoval a rozšiřoval až do podoby, kdy neuronové sítě přestaly být teorií s minimální aplikací a přetvořily se do systémů, které nás obklopují v běžném životě (Google překladač, burzovní zprávy apod.). S ohledem na jejich robustnost lze předpokládat, že jejich využití dále poroste (Simkanič, R. 2016).

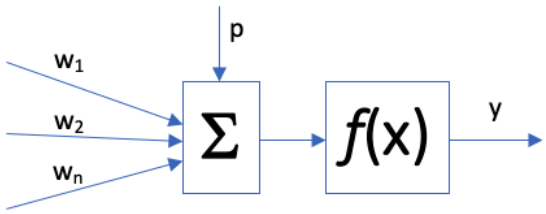
Obrázek 31: Neuron



Zdroj: Simkanič 2016.

Schéma umělého neuronu je vidět na následujícím obrázku 31. Podobně jako v případě biologického neuronu je zde několik vstupů. Vstupy mohou být podměty z vnějšího okolí nebo výstupy jiných neuronů. Tyto vstupy mají různou váhu w. Váha je klíčovou vlastností vstupní informace a vypovídá o důležitosti daného vstupu. Vstupy jsou zpracovány agregační a aktivační funkcí, která může mít několik podob viz dále. Výstupem je pak signál Y.

Obrázek 32: Model neuronu – matematický princip



Zdroj: vlastní tvorba dle Šnorek a Jiřina (1998).

$$y = f\left(\sum_{i=1}^{n} w_i x_i + p\right) \tag{22}$$

Kde x_i – jsou vstupy neuronu a počet těchto neuronů je n,

w_i – synaptické váhy,

p – práh neuronu,

f(x) – aktivační funkce neuronu.

Agregační funkce (Šíma a Neruda, 1996), jak již název napovídá, zpracovává vstupní signály. Nejčastěji se jedná o Lineární basickou funkci (LBF):

$$u(t) = \sum_{i=1}^{n} w_i(t)x_i(t) \tag{23}$$

Nebo o radiální basickou funkci (RBF):

$$u(t) = \sqrt{\sum_{i=1}^{n} \left(w_i(t)x_i(t)\right)^2} \tag{24}$$

Aktivační funkce převádí hodnotu vypočtenou agregační funkcí na výstupní signál. V tomto případě existuje několik různých aktivačních signálů:

- Skoková

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \tag{25}$$

- Lineární

$$f(x) = kx + q \tag{26}$$

- Sigmoidální funkce

$$f(x) = \frac{1}{1 + e^{kx}} \tag{27}$$

- Hyperbolicko-tangenciální funkce

$$f(x) = \tanh(kx) \tag{28}$$

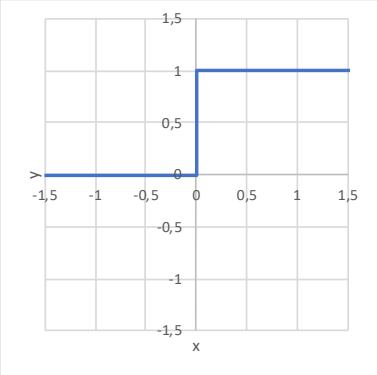
Kde k je parametr

q – posun křivky po y

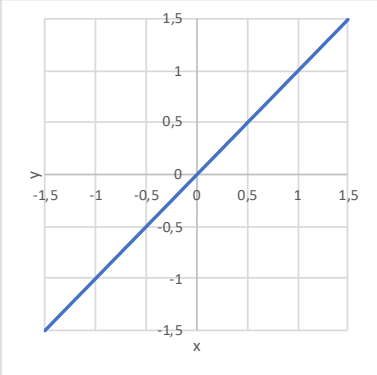
Průběhy funkcí jsou zobrazeny na následujícím obrázku. S výjimkou sigmoidální funkce byly hodnoty k i q rovny 0. V případě sigmoidální funkce byl parametr k roven 10.

Obrázek 33: Typy funkcí

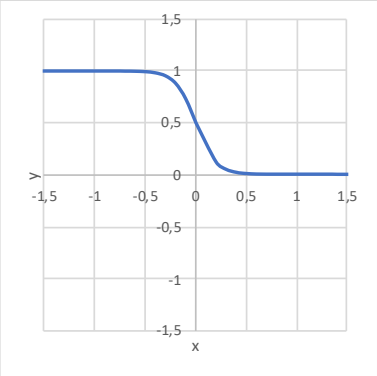
Skoková funkce



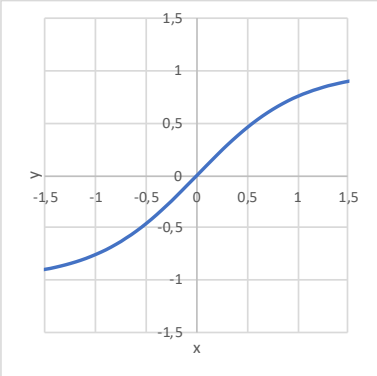
Lineární funkce



Sigmoidální funkce



Hyperbolicko-tangenciální funkce



Zdroj: Vlastní tvorba dle Volná, 2008.

Stejně jako pro předchozí algoritmy je proces učení pro neuronovou síť klíčovou vlastností. Na základě vstupních dat pak dochází k procesu učení a tím i k hledání vnitřních skrytých mechanizmů, které transformují vstupy na výstupy. Podle toho, zdali jsou k dispozici výstupní data určujeme, zdali se jedná o:

- učení s učitelem (Ojha, Abraham, Snášel, 2017),
- učení bez učitele (Haykin, 1999),
- zpětnovazebné učení (Reinforcement Learning - Kaelbling, Littman, Moore, (1996)).

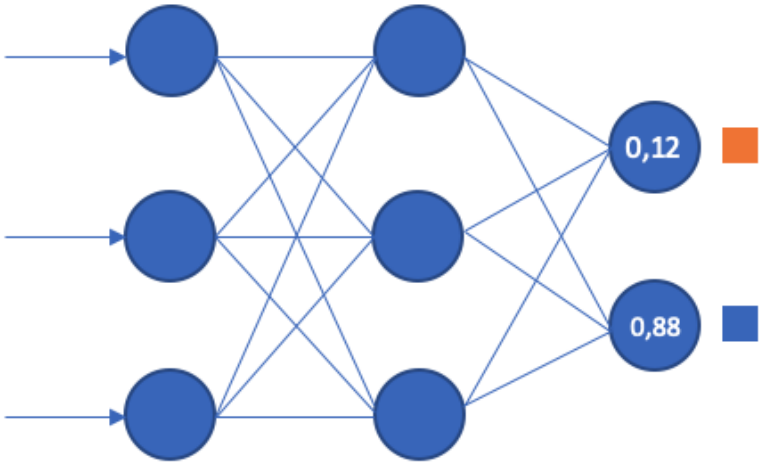
Jednotlivé neuronové sítě můžeme rozlišovat podle různých charakteristik. Z pohledu průchodu signálu rozlišujeme sítě na:

- dopředné (Šnorek, Jiřina, 1998),
- rekurentní (Vondrák, 2000).

3.6.1 Dopředné vícevrstvé sítě

Jedná se o klasický typ neuronové sítě (Volná, 2008), který je možné použít pro klasifikaci nebo regresi dat (Malý, 2007). Tato síť obsahuje standardně jednu vstupní vrstvu, jednu výstupní vrstvu a jednu nebo více skrytých vrstev (Jiřina, 2008). Každý neuron následné vrstvy je spojen se všemi neurony předchozí vrstvy (Malý, 2007). Spojení však obsahují určitou váhu, která určuje význam signálu pro daný neuron. Princip sítě je zobrazen na následujícím obrázku.

Obrázek 34: Vícevrstvá neuronová síť



Zdroj: Vlastní tvorba dle Matlabacademy 2021.

Na obrázku je vidět situace, kdy se snažíme identifikovat neznámý objekt. Tento objekt má tři prediktory (vlastnosti). První vrstva je vstupní vrstvou, kde počet neuronů odpovídá počtu prediktorů. Druhá vrstva obsahuje shodně 3 neurony. Počet neuronů ve skryté vrstvě však může být i výrazně vyšší bez ohledu na počet vstupních signálů. Po skryté vrstvě následuje výstupní vrstva, která obsahuje 2 neurony, což odpovídá počtu tříd, do kterých chceme klasifikovat výstupy (Ciresan, Meier, Schmidhuber, 2012). V našem případě tedy existují dva shluky. V uvedeném příkladu je zřejmé, že neznámý identifikovaný objekt patří do množiny modrých objektů.

Pokud bychom chtěli řešit regresní úlohu – výstupem je číslo (například dojezd elektromobilu), byl by na výstupu pouze jeden neuron. Ostatní body neuronové sítě by byly v tomto případě obdobné.

Při trénování neuronových sítí je velmi důležité, aby bylo dosáhnuto optima z hlediska dosaženého výsledku, který bude dále zobecnitelný, a predikce nových objektů budou relativně

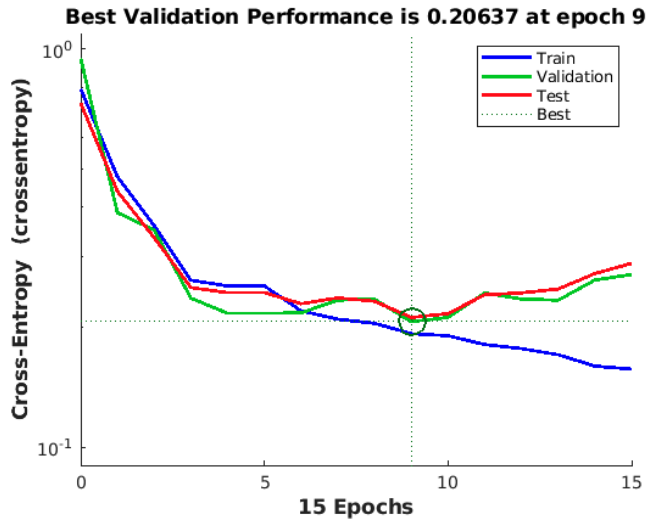
kvalitní. Pokud však dojde k přetrénování, pak sice bude síť fungovat skvěle pro trénovací data, ale velmi špatně pro testovací data. Je proto důležité zastavit trénování sítě ve správný okamžik. Aby k tomu mohlo dojít, rozdělují se data na trénovací, ověřovací a testovací. Rozdělení množiny dat provedeme v Matlabu pomocí příkazů:

```
net.divideParam.trainRatio = 70/100;

net.divideParam.valRatio = 15/100;

net.divideParam.testRatio = 15/100;
```

Na trénovačích datech se vytváří model. Na ověřovacích datech se testuje, zdali nedochází k přetrénování. Pokud vše proběhne jsou následně použita data testovací pro ověření kvality natrénovaného modelu.

Princip trénování je zobrazen na následujícím obrázku. Modrá čára vyjadřuje průběh trénování neuronové sítě. Pomocí algoritmu učení sítě dochází k posupnému přepočítávání vah tak, že dochází ke snižování hodnoty chyby téměř k nule. V určitý okamžik, který je vyznačen vertikální čárou však začíná být model přetrénovaný. V tento moment začíná stoupat počet chybně klasifikovaných vzorů u ověřovací množiny (červená čára).

Obrázek 35: Proces trénování



Zdroj: Vlastní tvorba dle Matlabacademy 2021.

Síť o třech skrytých neuronech vytvoříme v Matlabu za pomocí příkazu:

```
net = patternnet(3)
```

Následně síť natrénujeme pomocí příkazu

```
[net,tr] = train(net,X',targets');
```

Výstupem je samotná síť, která je uložená v proměnné net, a dále proměnná tr, která obsahuje informace o trénovacích, ověřovacích a testovacích datech. Rozpoznání nových vzorů provedeme pomocí příkazu

```
preds = net(Xnew');
```

Pokud bychom chtěli upravit počty neuronů v jednotlivých vrstvách, tak můžeme použít příkaz, ve kterém v matici vyjádříme počet neuronů pro každou vrstvu:

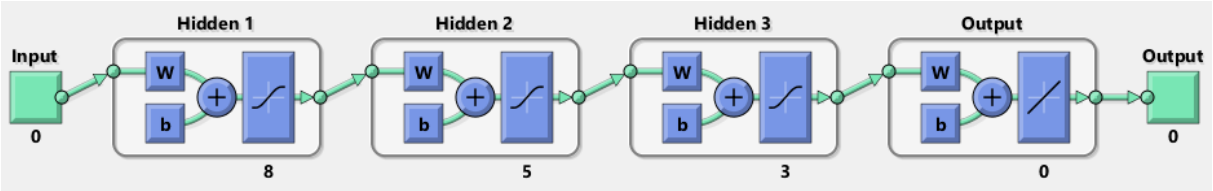
```
net = fitnet([8 5 3])
```

Blokové schéma neuronové sítě je možné zobrazit pomocí příkazu

```
view(net)
```

Výsledek je vidět na následujícím obrázku. V tomto případě jsou to 3 skryté vrstvy o 8, 5 a 3 neuronech.

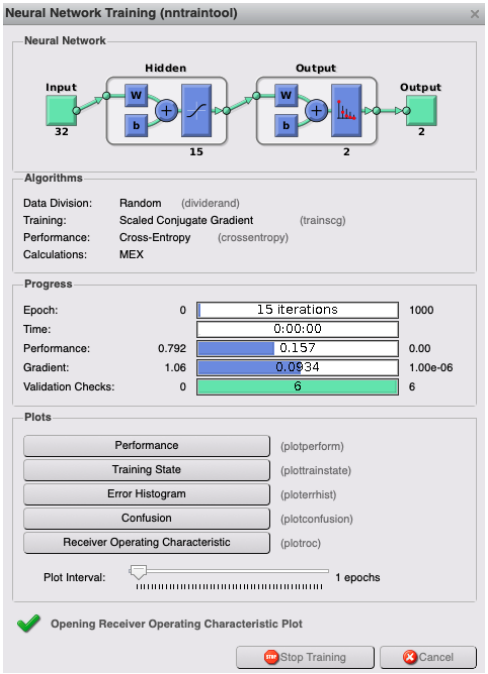
Obrázek 36: Schéma sítě



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Při trénování neuronové sítě se automaticky zobrazí dialogové okno, které informuje o průběhu trénování. V okně se ukáže počet interakcí, schéma sítě, výkonnost a další parametry. Příklad dialogového okna je zobrazen na dalším obrázku.

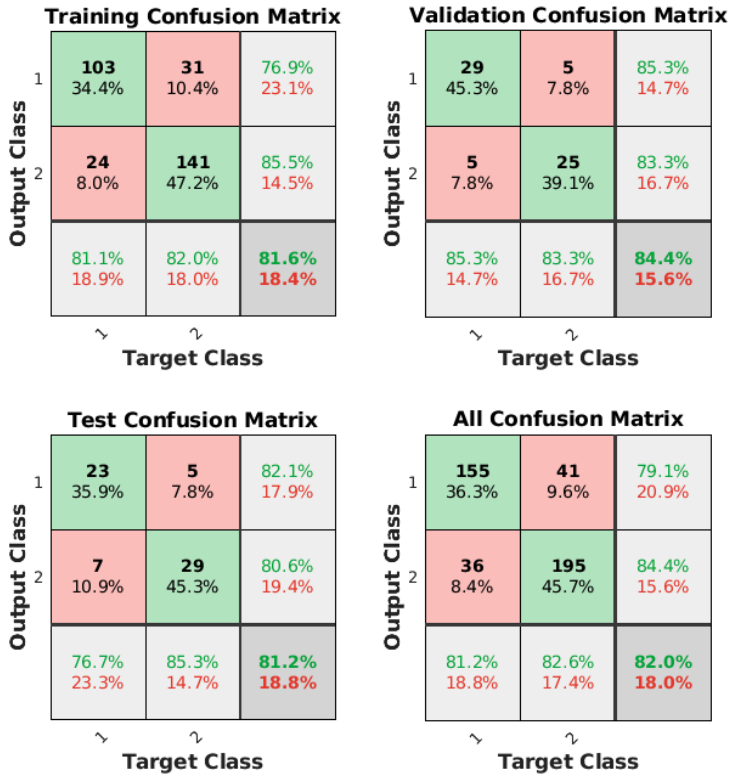
Obrázek 37: Proces trénování



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

V oblasti plots je možné zobrazit nejrůznější grafy, které vyjadřují výkonnost neuronové sítě. Nejpoužívanější je Performance, která odpovídá obrázku 37. Dále je jsou důležité konfuzní matice. Ty se vykreslí dle obrázku níže. Zde jsou zobrazeny konfuzní matice dle logiky předchozích metod. Matice jsou ale zobrazeny pro každý z typů dat.

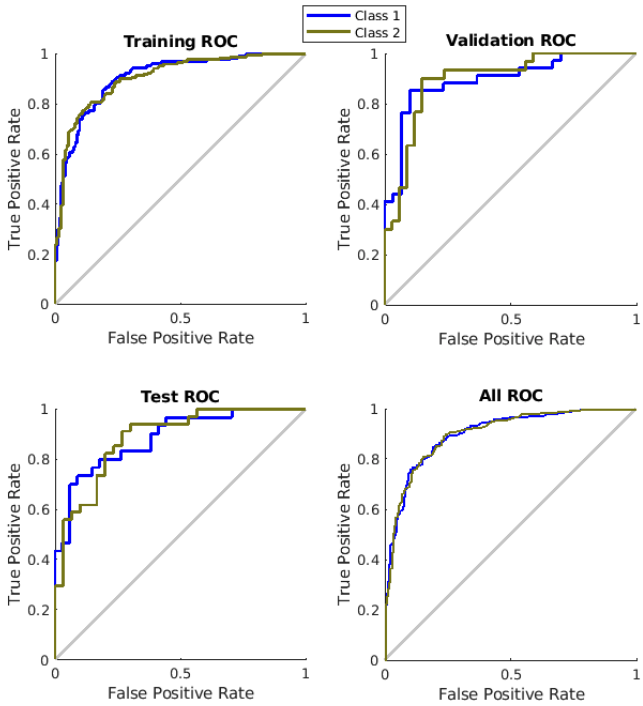
Obrázek 38: Konfuzní matice



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

V neposlední řadě je možné zobrazit ROC křivku, ze které je patrné, jaký je poměr mezi špatně a správně zařazenými daty. Čím více je křivka prohnutá směrem k hodnotě [0; 1], tím je klasifikace kvalitnější. Příklad je zobrazen na následujícím obrázku:

Obrázek 39: ROC křivka



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

3.6.2 Sebeorganizující se (Kohonenovy) mapy

Kohonenovy mapy (Kohonen, 1982) jsou typem neuronových sítí (Vojáček, 2006), které se učí bez učitele (Roman, 2019). Často jsou tyto sítě využívány pro redukci proměnných a vizualizaci segmentace sledovaných objektů (Kohonen, 1989). Cílem je, aby podobné části sítě reagovali obdobně na určité vstupní vzorce, tak jako tomu je částečně v případě lidského mozku při zpracování zvukových či obrazových vjemů (Haykin, 1999).

Princip spočívá v náhodném rozmístění neuronů v prostoru případně v rovnoměrném rozmístění v prostoru (Buhmann a Kuhnel, 1992). Následně se vybere náhodně objekt a k němu se počítá vzdálenost všech neuronů podle vzorce:

$$u(t) = \sum_{i=1}^{n} \left(x_i(t) - w_{ij}(t) \right)^2 \qquad (29)$$

Kde $w_{ij}(t)$ představuje pozici neuronu (váhu),

x_i (t) – představuje vybraný objekt.

Neuron, který je k objektu nejblíže se vybere a označí za BMU (best matching unit). Následně se upraví váha (pozice) vybraného neuronu a neuronů, se kterými přímo souvisí (Kohonen, 2001). Úprava pozice je směrem k vybranému objektu.

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t)\eta(t)\left[x_i(t) - w_{ij}(t)\right] \quad (30)$$

Kde $w_{ij}(t)$ představuje pozici neuronu (váhu),

$\alpha(t)$ – parametr omezení učení, který se postupně snižuje,

$\eta(t)$ – je omezení kvůli vzdálenosti nejbližšího souseda,

$x_i(t)$ – představuje vybraný objekt.

V Matlabu vytvoříme Kohonenovu mapu pomocí příkazu

```
net = selforgmap([10,10]);
```

Následně určíme, kolik epoch bude použito pro trénování (kolik kroků).

```
net.epochs = 1000;
```

V dalším kroku dojde k trénování sítě.

```
net = train(net,X);
```

V případě, že budeme predikovat další objekty, použijeme příkaz

```
preds = net(XNew);
```

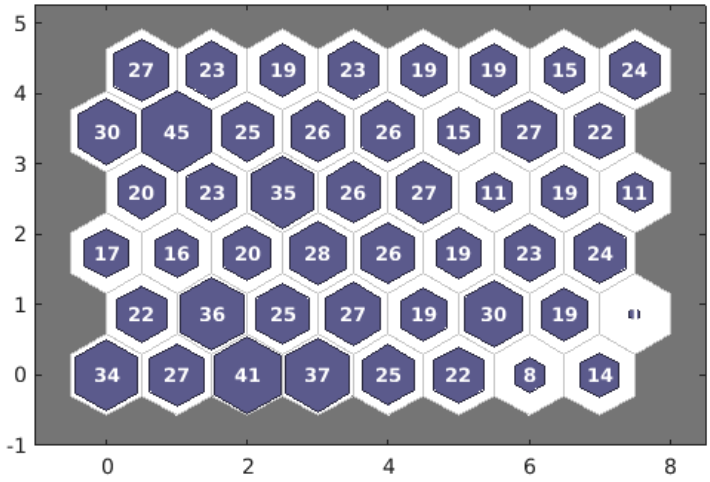
Výsledkem je matice, kde řádek představuje odpovídající neuron a sloupec představuje sledovaný objekt. Pro lepší interpretovatelnost výsledku je možné dohledat index odpovídajícího neuronu za pomocí příkazu

```
predindex = vec2ind(preds);
```

Výsledky Kohonenovy mapy je možné zobrazit řadou způsobů. První z nich je vidět na následujícím obrázku. Jedná se o histogram, kde každé pole je reprezentované jedním neuronem. Číslo v poli představuje počet objektů, které se v jeho okolí nachází. Tento výstup se zobrazí za pomocí příkazu:

```
plotsomhits(nn,inputs)
```

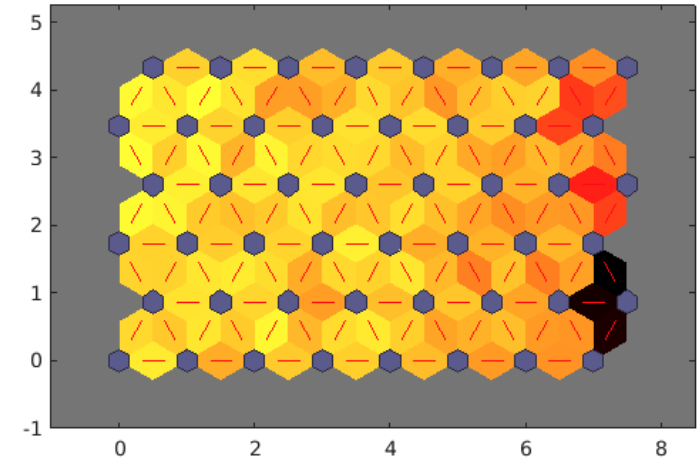
Obrázek 40: Počet dat v jednotlivých clusterech



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

Další možností je zobrazení jednotlivých vzdáleností mezi neurony. Neurony jsou v tomto případě zobrazeny jako modré hexagony. Váhy jsou naznačeny červenou čárou. Podbarvení váhy je od žluté po černou a vyjadřuje vzdálenost mezi neurony. Na níže uvedeném obrázku je vidět, že neuron v předposledním řádku a posledním sloupci je od ostatních poměrně vzdálený a reprezentuje tak množinu vzdálených prvků, kterými mohou být například šumy.

Obrázek 41: Vzdálenost mezi sousedy



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

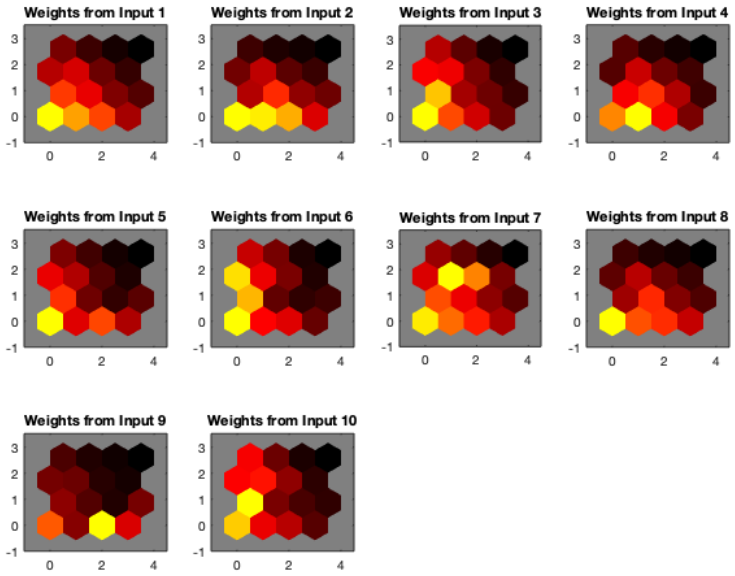
Dalším užitečným nástrojem pro analýzu vstupních prediktorů a sítě je příkaz

```
plotsomplanes(net)
```

Tento příkaz zobrazí barevné schéma, které vyjadřuje propojení vstupního prediktoru s konkrétním neuronem. Silně negativní vyzby jsou zobrazeny černě, nulové propojení je

červené a silně pozitivní propojení je žluté. Čím více jsou si prediktory podobné, tím více lze předpokládat, že jsou korelovány.

Obrázek 42: Váhy prediktorů k jednotlivým neuronům



Zdroj: Vlastní tvorba dle Matlab documentation 2021.

4 Metodika

Data: Datový soubor obsahoval po vygenerování setu z databáze Albertina společnosti Bisnode celkem 42592 datových řádků. Přitom každý řádek obsahoval:

1. Identifikaci firmy: název firmy, IČO, obec kraj, velikost obce,
2. Údaje o firmě: NACE, počet zaměstnanců, kód NACE5A, M_NACE, OKEČ5A, rok účetní závěrky
3. Výkazy účetní závěrky za daný rok působení firmy: rozvahu, výkaz zisků a ztrát, výkaz o peněžních tocích.
4. Vybrané ukazatele rentability, aktivity, likvidity, zadluženosti, produktivity a další.

Postup přípravy dat (MS EXCEL):

1. Výpočet EBIT (přičtením daně a úroků k EAT).
2. Výpočet ROA (EBIT/Aktiva).
3. Výpočet ROE (EAT/Vlastní kapitál).
4. Výpočet EVA Equity (dle metodiky manželů Neumaierových – MPO).
5. V souboru byly ponechány podniky splňující všechny následující podmínky:
    a. Počet zaměstnanců: 0 – 499,
    b. s kladnými aktivy,
    c. s kladným dlouhodobým majetkem,
    d. s kladným dlouhodobým finančním majetkem,
    e. s kladným dlouhodobým nehmotným majetkem,
    f. s kladným oběžným majetkem,
    g. s kladnými zásobami,
    h. s kladnými dlouhodobými pohledávkami,
    i. s kladnými krátkodobými pohledávkami,
    j. s kladnými pohledávkami z obchodních vztahů,
    k. s kladnými pohledávkami k přidruženým společnostem,
    l. s kladným základním kapitálem,
    m. s kladnými rezervními fondy,
    n. s kladenými rezervami,
    o. s kladnými penězi,
    p. s kladnými tržbami za zboží,

q. s kladným spotřebovaným materiálem,
r. s kladnou výkonovou spotřebou,
s. s kladnými výkony,
t. s kladnými náklady na zboží,
u. s kladnými odpisy,
v. s kladnými tržbami z prodeje dlouhodobého majetku,
w. s kladnými tržbami z prodeje materiálu,
x. s kladnou zůstatkovou cenou prodaného dlouhodobého majetku,
y. s kladnými nákladovými úroky,
z. se mzdovými náklady vyššími než 120 tis. kč za rok,
aa. s ROA v intervalu (-100 %, +100 %),
bb. s ROE v intervalu (-100 %, +100 %),
cc. s alternativními náklady na vlastní kapitál v intervalu (0 %, +100 %),
dd. s tržbami za zboží a za vlastní výkony v součtu alespoň v částce 120 tis. Kč za rok.

6. Kódování velikosti společnosti podle počtu zaměstnanců:

Původní hodnota	Kód
0	0
1-5	1
6-9	2
10-19	3
20-24	4
25-49	5
50-99	6
100-199	7
200-249	8
250-499	9

7. Kódy NACE 5A byly změněny pouze na sekce CZ-NACE (tedy pouze na první písmeno klasifikace)

Úpravou se zmenšil datový soubor ze 42592 řádků na 29611 řádků:

Rok	Původní datový soubor	Upravený datový soubor
2013	7 976	5 705
2014	8 059	5 492
2015	8 046	5 449
2016	8 803	5 982
2017	9 708	6 983
Celkem	42 592	29 611

Výsledný soubor zároveň obsahuje kompletní účetní závěrku s některými dopočtenými daty, které jsou uvedeny výše. Tyto údaje jsou tak prediktory, kterých je více jak 100. Z těchto důvodů bude výsledný soubor podniků zredukován na hlavní komponenty a dále bude podle metodiky Neumaierových stanovena kategorie podniku dle schématu níže.

- Podniky tvořící hodnotu (kladná hodnota EVA) – roe > re.
- Podniky s kladným ziskem se zápornou hodnotou EVA, které však překonávají bezrizikovou sazbu rf – re > ROE > rf.
- Podniky s kladným ziskem kde ROE nedosahuje ani bezrizikové sazby – re > rf > ROE > 0.
- Podniky se záporným ziskem.

Tyto údaje pak budou vstupovat do dalších analýz. Hlavními komponentami jsou:

- Kraj.
- Velikost obce.
- Počet zaměstnanců.
- Sekce NACE.
- Rok účetní závěrky.
- Aktiva celkem - tis. Kč.
- Dlouhodobý majetek - tis. Kč.
- Oběžná aktiva - tis. Kč.
- Vlastní kapitál - tis. Kč.
- Cizí zdroje - tis. Kč.
- Krátkodobé závazky.
- Osobní náklady - tis. Kč.
- Odpisy dlouhodobého nehmotného a hmotného majetku - tis. Kč.
- Provozní výsledek hospodaření - tis. Kč.
- Nákladové úroky - tis. Kč.
- Finanční výsledek hospodaření - tis. Kč.
- Výsledek hospodaření za účetní období (+/-) - tis. Kč.
- Daň z příjmů za běžnou a mimořádnou činnost.
- Obrat.
- Kategorie podniku.

5 Výsledky

5.1 Predikce výkonnosti podniků dle metodiky INFA

5.1.1 Analýza vstupních dat

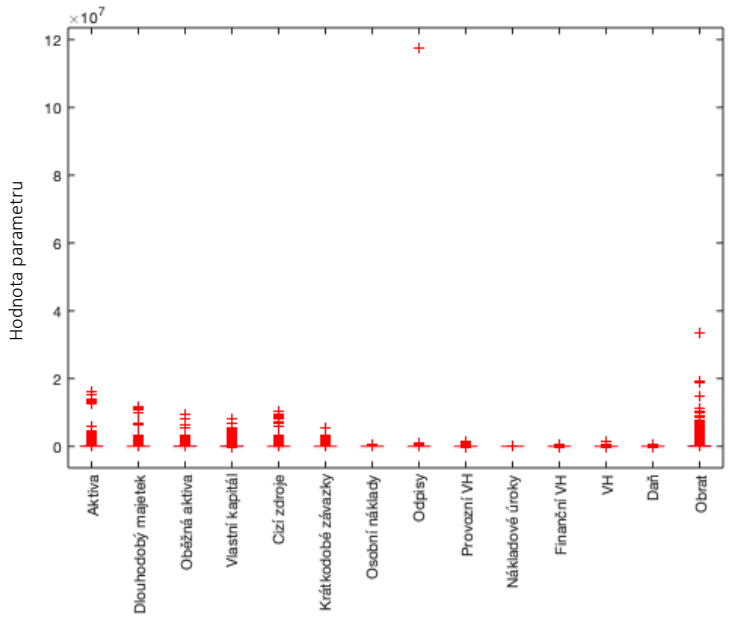
5.1.1.1 Popisná analýza

V první fázi provedeme analýzu souboru dat, abychom mohli analyzovat, jestli data není nutné ještě nějakým způsobem upravit. Nejdříve posoudíme numerická data. Jejich charakteristiku je možné provést za pomocí krabicového grafu, který zobrazíme pomocí příkazů:

```
boxplot(table2array(import(:,6:end-1, 'labels', import.Properties.VariableNames(6:end-1));

xtickangle(90);

ylabel("Hodnota parametru");
```

Příkaz *boxplot* pracuje pouze s číselnou maticí, proto je nutné nejdříve převést tabulku na matici za pomocí příkazu *table2array*. Ostatní příkazy a parametry pouze určují popis obrázku pro snadnější interpretaci. Výsledek je vidět na obrázku 43. Z obrázku je patrné, že data trpí extrémním množstvím odlehlých hodnot. Tyto hodnoty, které jsou v grafu zobrazeny červenou značkou +, prakticky znemožňují zobrazit střední hodnotu a rozptyl. Navíc je zcela zřejmé, že v některých případech půjde o evidentní chybu v datech. Toto je například patrné v případě odpisů., kdy by jedna firma měla mít odpisy ve stovkách miliard.

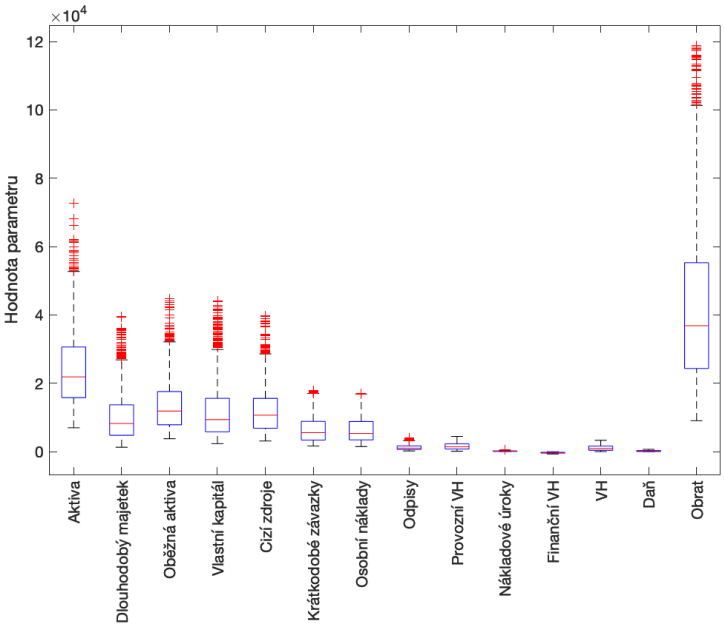
Obrázek 43: Charakteristika numerických prediktorů souboru



Zdroj: Vlastní tvorba.

Za pomocí příkazu *rmoutliers* lze odlehlé hodnoty eliminovat. V takovém případě je vidět výsledek na následujícím obrázku. Zde je již patrné, jakých hodnot z hlediska statistické charakteristiky souboru jednotlivé složky účetní závěrky dosahují. Z výsledku je patrné, že největšího rozptylu dosahují položky obratů a celkových aktiv. Naopak nákladové úroky, daně a finančních výsledek hospodaření patří k nejméně variabilním prediktorům.

Obrázek 44: Statistická charakteristika souboru s redukcí odlehlých hodnot



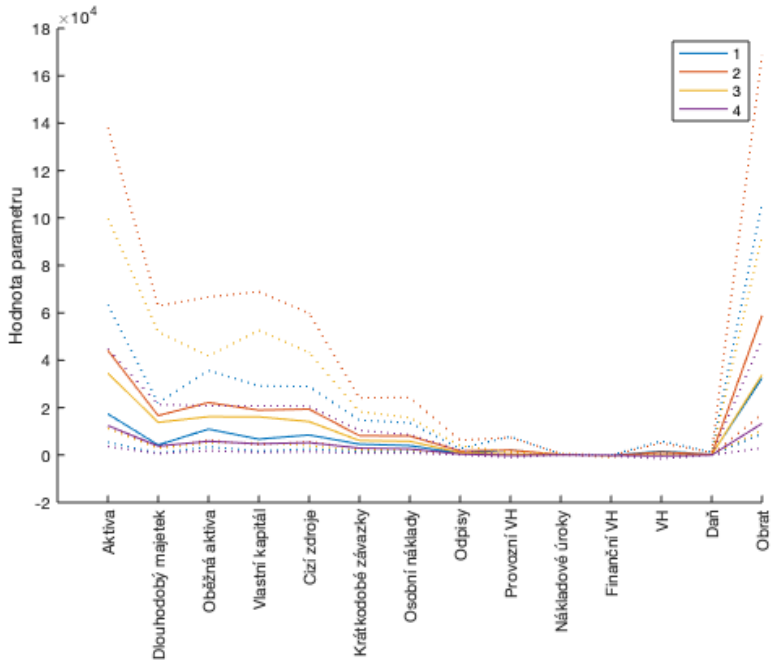
Zdroj: Vlastní tvorba.

Při posuzování souboru nás zajímá nejenom charakteristika variability a střední hodnoty, ale rovněž, jaké charakteristiky mají jednotlivé prediktory ve vztahu k výsledku. V našem případě je jako výsledek chápána kategorie podniku dle stanovené metodiky. Vizualizaci této skutečnosti lze provést za pomocí příkazu:

```
parallelcoords(table2array(import(:,6:end-1)),'Group',import.Kategorie_podniku,'Quantile',0.25,
'LineWidth',1, 'labels', import.Properties.VariableNames(6:end-1))
```

Příkaz parallelcoords vytvoří podobně jako předchozí boxplot graf, který obsahuje střední hodnotu a při tomto nastavení 1 quartil. Ostatní parametry jen zlepšují čitelnost grafu. Výsledek je vidět na následujícím obrázku. První quartil pro danou kategorii je zobrazen tečkovanou čárou. Legenda zobrazuje jednotlivé kategorie dle metodiky. Z grafu je patrné, že například podniky, které mají kladnou hodnotu EVA (kategorie 1 zobrazená modrou čárou), nemají příliš vysokou hodnotu aktiv. Pokud se však podíváme pouze na parametr aktiv, tak zjistíme že velmi blízkou jsou však i podniky, které jsou naopak ve 4. kategorii a tudíž dosáhly ztráty. Nejvyšší hodnotu mají podniky v kategorii 2, které mají sice kladný výsledek hospodaření, ale zápornou hodnotu EVA. Jejich výsledek však překonal bezrizikovou sazbu. Podobné pořadí se nese napříč prvními sedmi prediktory.

Obrázek 45: Charakteristika importovaného souboru dat (parallelcoords)



Zdroj: Vlastní tvorba.

Pro řadu analýz je nezbytné data znormovat. Důvodem je skutečnost, že například některé metody jako PCA sledují vzdálenosti mezi jednotlivými vektory a přirozená rozdílnost velikosti

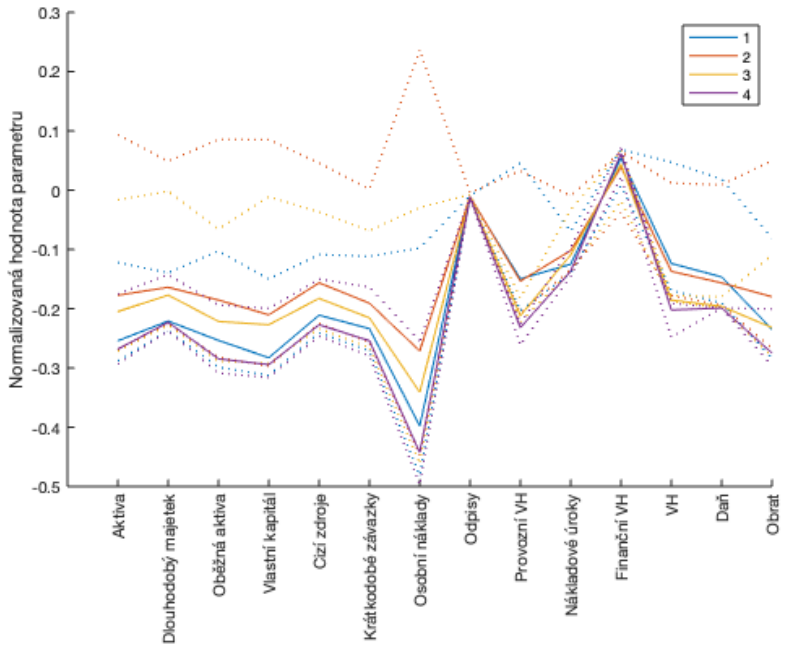
jednotlivých složek by vedla k tomu, že by charakteristiky menších složek nebyly zaneseny. V našem případě by tak nebyl zohledněn finanční výsledek hospodaření apod.

Normalizaci dat provedeme za pomocí příkazu *normalize*. Celý příkaz pak bude vypadat následovně:

```
parallelcoords(normalize(table2array(import(:,6:end-1))), 'Group', import.Kategorie_podniku, 'Quantile', 0.25, 'LineWidth',1, 'labels', import.Properties.VariableNames(6:end-1))
```

Výsledek je zobrazen na následujícím obrázku. Na obrázku je překvapivá hodnota vztahující se k odpisům. Tato položka se tváří, jako by zde nebyl prakticky žádný rozdíl mezi jednotlivými charakteristikami. Výsledek však může úzce souviset s chybovou hodnotou, která byla objevena za pomoci krabicového grafu.

Obrázek 46: Normalizovaná data s extrémními hodnotami



Zdroj: Vlastní tvorba.

Před odstraněním extrémní hodnoty z tabulky si nejdříve zobrazíme celý záznam, abychom mohli posoudit, zdali se skutečně jedná o chybu dat. Toto provedeme za pomocí příkazu

```
import(import.Odpisy > 100000000, :)
```

Hodnota 100 000 000 byla odhadnuta na základě zobrazení v krabicovém grafu s odlehlými hodnotami. Výsledek příkazu je vidět v následující tabulce.

Tabulka 1: Charakteristika podniku s extrémní hodnotou (tis. Kč)

Parametr	Hodnota
Kraj	Zlínský kraj
Velikost obce	[04] 5 000 - 9 999 obyv.
Počet zaměstnanců	3
Sekce NACE	Sekce I
Rok účetní závěrky	2017
Aktiva	59437
Dlouhodobý majetek	53182
Oběžná aktiva	5804
Vlastní kapitál	38346
Cizí zdroje	21080
Krátkodobé závazky	3719
Osobní náklady	2989
Odpisy	117601176
Provozní VH	-2358
Nákladové úroky	2265
Finanční VH	-2051
VH	-4277
Daň	-132
Obrat	7770
Kategorie podniku	4

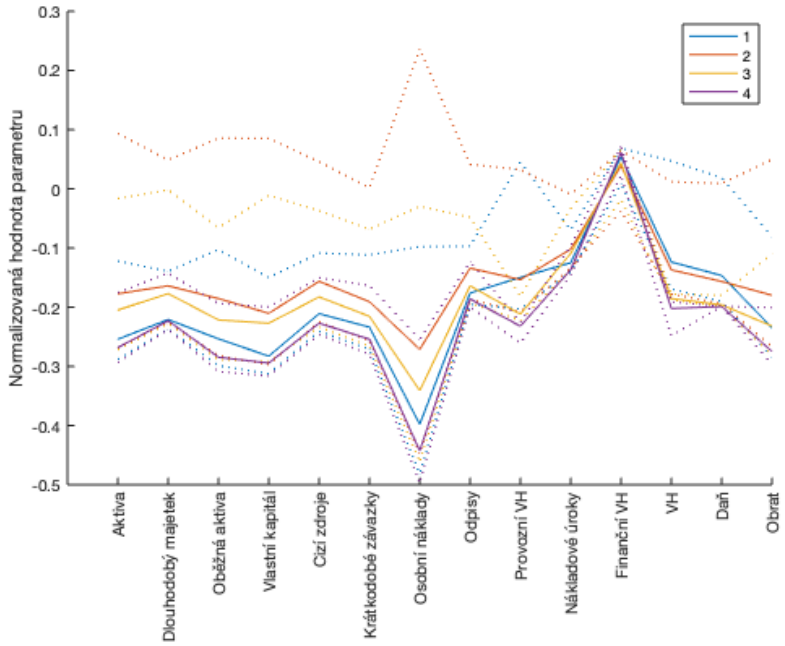
Zdroj: Vlastní tvorba.

Z uvedených záznamů (zejména obratu a výsledku hospodaření) je patrné, že firma nemůže mít odpisy převyšující 117 mild. Kč. Z těchto důvodů záznam z dalších analýz vyloučíme za pomocí příkazu:

```
import(import.Odpisy > 10000000, :) = [];
```

Následně opět zobrazíme znormované hodnoty prediktorů. Výsledek je vidět na následujícím obrázku.

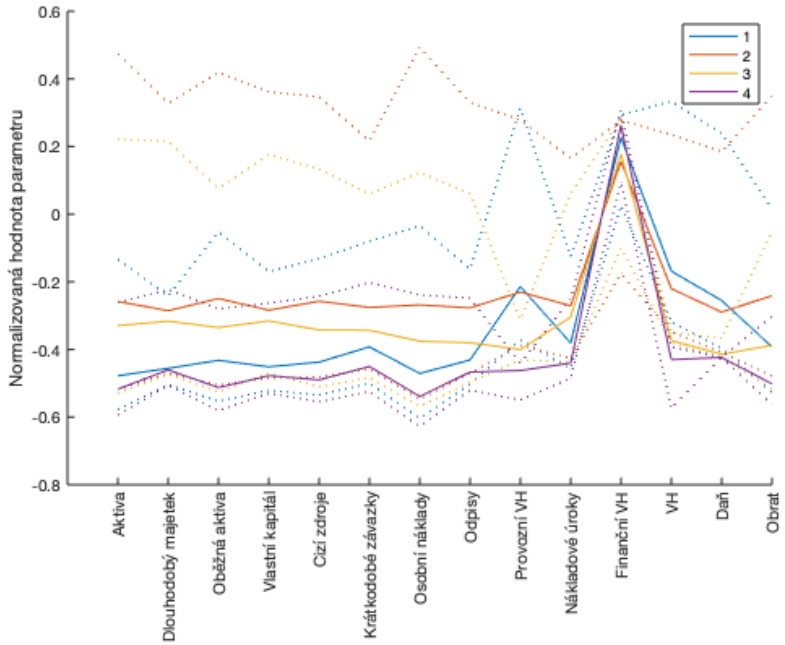
Obrázek 47: Normalizovaná analýza prediktorů s ohledem na kategorizaci podniku bez extrémní hodnoty



Zdroj: Vlastní tvorba.

Z obrázku je patrné, že odstranění jedné chyby má poměrně značný vliv na výslednou charakteristiku při normování. Z těchto důvodů využijeme opět funkci *rmoutliers*, abychom odstranili i zbylé extrémní hodnoty. Při použití funkce bylo odstraněno celkem 4,1 % záznamů. Výsledek zobrazení je pak vidět na následujícím obrázku. Takto upravená data budou následně vstupem pro další analýzy. Nyní jsou již patrnější charakteristiky jednotlivých složek účetní závěrky připadající na kategorie podniku. Z těchto charakteristik vyplývá, že nejlepší podniky jsou podniky s menšími aktivy a dalšími složkami rozvahy. Dosahují však nejvyšších provozních zisků. Jejich obrat je na úrovni podniků v kategorii 3. Oproti tomu podniky v kategorie 2 jsou největší z pohledu aktiv i jejich dalších složek a rovněž i z pohledu obratu. V případě podniků v kategorii 4 je možné pozorovat, velmi podobné charakteristiky jako v případě kategorie 1. Rozdíl je patrný především v oběžných aktivech a pak dále v obratu, provozním výsledku hospodaření, výsledku hospodaření a daních. Tyto odlišné položky jsou vždy vyšší v případě podniku v kategorii 1.

Obrázek 48: Normalizovaná analýza prediktorů s ohledem na kategorizaci podniku bez extrémní hodnoty

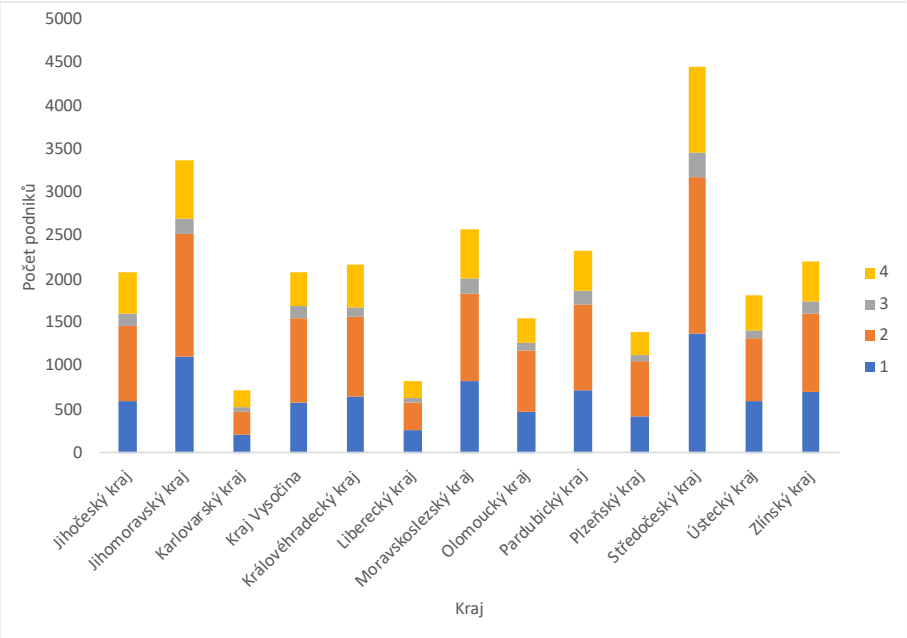


Zdroj: Vlastní tvorba.

Pro predikci kategorie podniku budou využity i další informace o podniku, které však budou kategoriální. Jedná se o soubor 5 proměnných spojených s krajem, velikostí obce, rokem účetní závěrky, počtem zaměstnanců a předmětem podnikání.

Počet podniků dle kategorie podle krajů je zobrazen na následujícím obrázku. Z obrázku vyplývá, že nejvíce podniků je ze Středočeského kraje (okolo 4 500) a dále pak z Jihomoravského kraje (okolo 3 500). Nejméně podniků je naopak v kraji Karlovarském (přes 700) a Libereckém (přes 800).

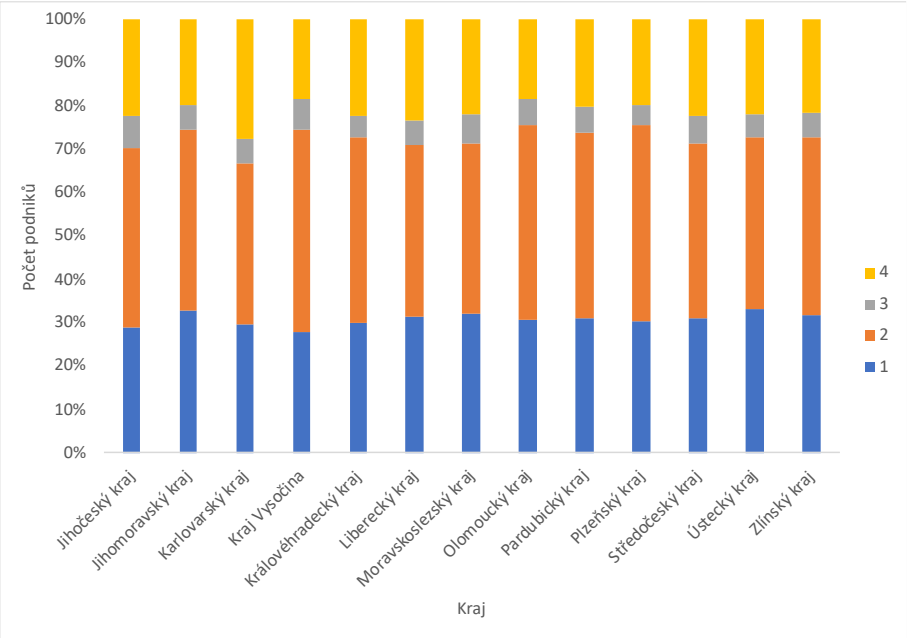
Obrázek 49: Počet podniků podle kategorie podle krajů



Zdroj: Vlastní tvorba.

Z pohledu tvorby hodnoty je na předchozím obrázku patrné, že rozložení kategorií přibližně odpovídá četnosti. Jinými slovy nejvíce podniků tvořící hodnotu je opět ve Středočeském kraji (cca 1380) a Jihomoravském kraji (cca 1100). Obdobně je to mu i naopak. Pokud bychom rozložení firem provedli z pohledu procent, tak uvidíme výsledek na následujícím obrázku. Z obrázku je patrné, že procentní podíl podniků, které tvoří v hodnotu, v jednotlivých krajích příliš nekolísá a pohybuje se na úrovni 30 %. Nejlepší procentní podíl má Ústecký kraj (33 %) společně s Jihomoravským krajem (32 %). Naopak nejhorší procentní podíl firem, které jsou ve ztrátě, má Karlovarský kraj (27, 5 %) a Liberecký kraj (22 %). Z obrázku rovněž vyplývá, že největší složkou jsou firmy, které sice dosahují kladného zisku, který převyšuje bezrizikovou sazbu, ale rovněž se jedná o firmy, které mají zápornou hodnotu ekonomické přidané hodnoty. Jedná se přibližně o 40 % podniků. Naopak nejmenší složkou v analyzovaných datech jsou podniky, které mají kladnou hodnotu zisku, která je nižší než bezriziková sazba. Toto je způsobeno především relativně stabilním prostředím a díky tomu nízké hodnotě bezrizikové sazby. Napříč kraji se jedná o cca 5 až 7% podíl.

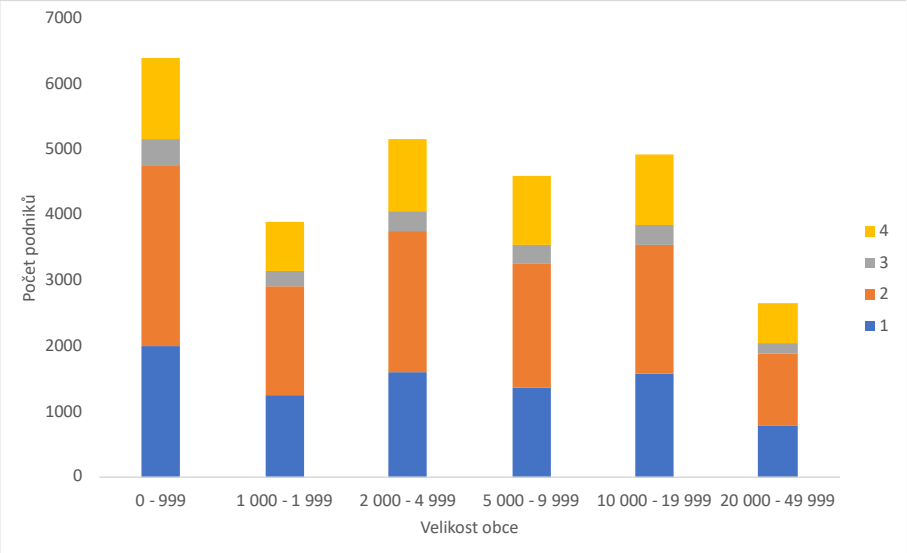
Obrázek 50: Počet podniků v krajích podle kategorií – procentní rozdělení



Zdroj: Vlastní tvorba.

Nejen kraj tvoří podnikatelské prostředí, ale i konkrétní obec, ve které se firma nachází. Pokud zobrazíme počet podniků dle kategorie a velikosti obce zobrazí se nám obrázek 52. Z obrázku vyplývá, že nevíce jsou zastoupeny podniky, které se nachází na území obce do 1 000 obyvatel. Naopak nejmenší zastoupení mají podniky v obcích v rozmezí 20 až 50 tis. obyvatel. Podniky v obci nad 50 tis. obyvatel byly v souladu s metodikou vyloučeny z analýzy.

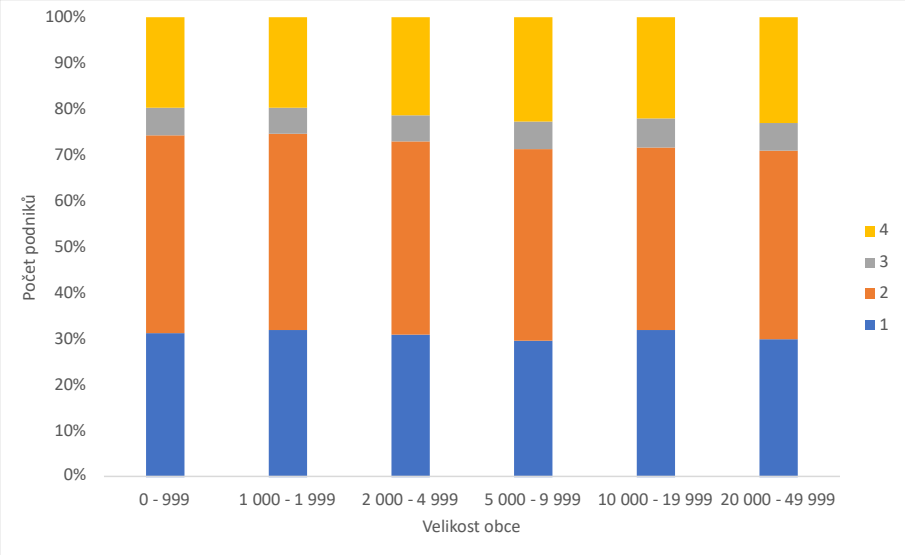
Obrázek 51: Počty podniků dle kategorie a velikosti obce



Zdroj: Vlastní tvorba.

Abychom mohli snadněji porovnat vliv velikosti obce na přidanou hodnotu, změníme graf na procentní rozložení. Výsledek je vidět na následujícím obrázku. Z výsledku je patrné, že velikost obce má relativně zanedbatelný vliv na výslednou kategorizaci. Tento pohled je samozřejmě velmi zjednodušený a při analýze křížových prediktorů může vliv velikosti obce hrát svojí roli. Jinými slovy – velikost obce může hrát roli třeba jen u podnikání v maloobchodu, ale nehraje roli v chemickém průmyslu. Tyto skryté zákonitosti, kterých je velké množství, by měly být odhaleny pomocí metod strojového učení, a proto i velikost obce bude zahrnuta jako jeden z prediktorů.

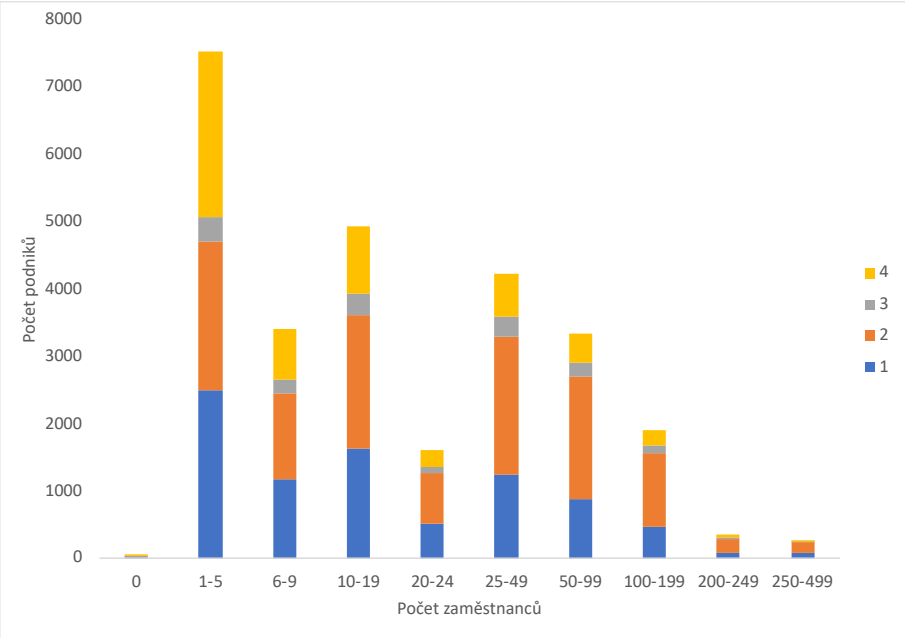
Obrázek 52: Počty podniků v procentech dle velikosti obce



Zdroj: Vlastní tvorba.

Další kategoriální proměnnou je počet podniků podle počtu zaměstnanců. Při zobrazení rozložení (obrázek 54) vidíme, že v analyzovaném soboru jsou nevíce zastoupeny podniky do 5 zaměstnanců a dále pak podniky od 10 do 20 zaměstnanců. Nejméně jsou pak zahrnuty podniky, které mají více jak 200, respektive 250 zaměstnanců, za předpokladu, že nepočítáme podniky, které vykazují 0 zaměstnanců, což dává ekonomicky smysl pouze v případě, že jednatel nemá pracovní smlouvu, nebo firma jen uvedla chybný údaj ve statistickém šetření.

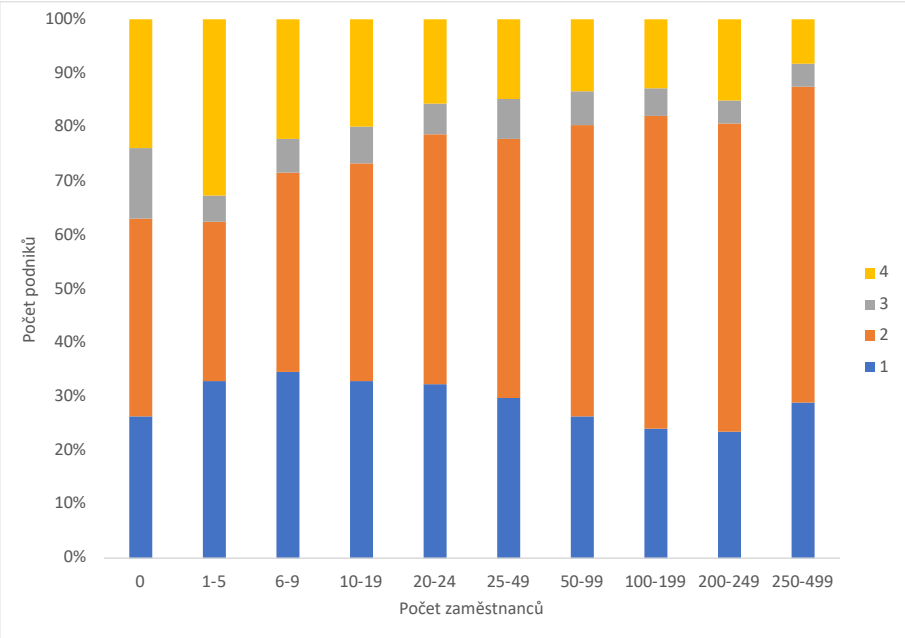
Obrázek 53: Počet podniků podle kategorie a počtu zaměstnanců



Zdroj: Vlastní tvorba.

Abychom mohli posoudit vliv počtu zaměstnanců na přidanou hodnotu, znormujeme předchozí kategorie. Výsledek je vidět na obrázku 55. Z obrázku je patrný zajímavý trend, který v případě podniků tvořících hodnotu (kategorie 1) vypadá jako graf funkce sinus. Procentní zastoupení podniků tvořících hodnotu tedy nejdříve stoupá do kategorie 6 až 10 zaměstnanců (v této kategorii dosahuje přibližně 35 %) a dále pak klesá do kategorie 200 až 250 zaměstnanců (v této kategorii dosahuje přibližně 25 %) a nakonec mírně stoupne na úroveň téměř 29 %. Dále je možné spatřit trend, který naznačuje, že s počtem zaměstnanců klesá podíl firem, které dosáhly ztráty. Tento podíl se pohybuje od 10 % u kategorie 250 až 500 zaměstnanců až do 30 % u kategorie 1 až 5 zaměstnanců. Výjimku tvoří kategorie podniků, které nevykazují zaměstnance. Z výše uvedeného rovněž vyplývá, že s počtem zaměstnanců roste podíl firem, které dosahují kladného zisku. Z pohledu dat je zajímavý i výsledek firem vykazující zisk pod bezrizikovou sazbou, které jsou v případě kategorie s nulovým počtem zaměstnanců jsou neobvykle vysoké. Toto do určité míry může souviset s daňovou optimalizací a úvahou firem v rovině vykáži zisk, aby bylo menší riziko kontroly z finančního úřadu, avšak tento zisk nevykáži příliš vysoký s ohledem na odvody daní. Samozřejmě, že podobnou úvahu mohou mít i firmy v dalších kategoriích, avšak s vyšším počtem zaměstnanců je realizace podobného postupu výrazně složitější.

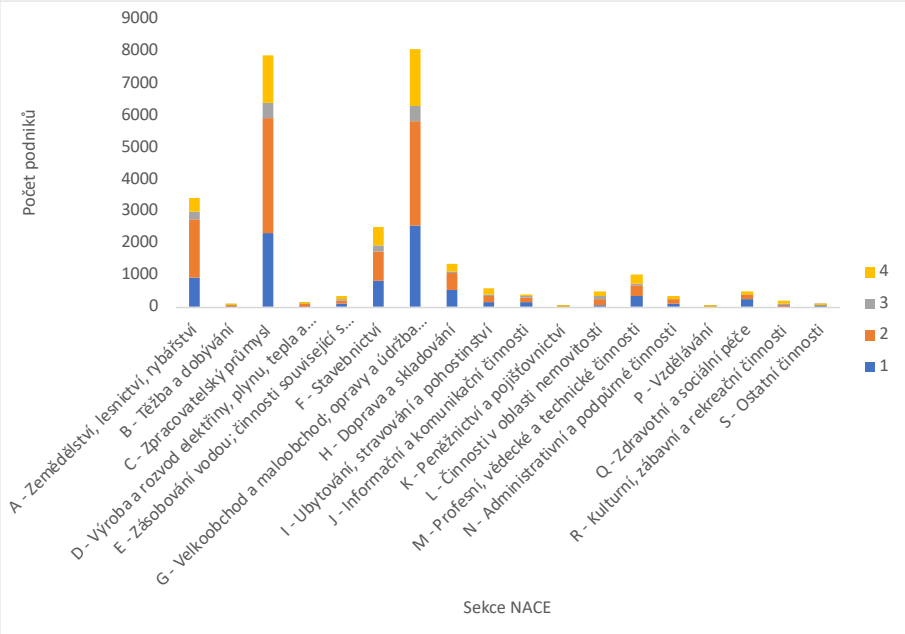
Obrázek 54: Počet podniků podle kategorie a počtu zaměstnanců v procentech



Zdroj: Vlastní tvorba.

Do analýzy byly zahrnuta i data o hlavní sekci klasifikace NACE, ve které firmy působí. Přehled počtu firem podle sekce a sledované kategorie je vidět na následujícím obrázku. Z tohoto rozdělení vyplývá, že byly nejvíce zahrnuty podniky v oblasti obchodu, ve zpracovatelském průmyslu, zemědělství a stavebnictví.

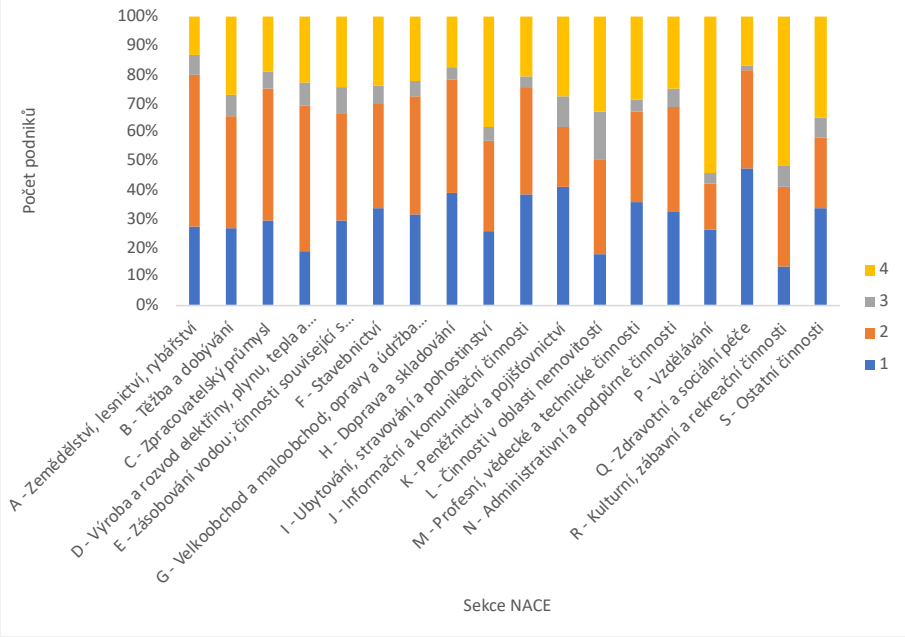
Obrázek 55: Počet podniků podle kategorie a sekce NACE



Zdroj: Vlastní tvorba.

Z vyrovnaného grafu, který je zobrazen na obrázku (56) je patrné, že sekce NACE má daleko větší vliv než předchozí kategoriální proměnné. Například informační a komunikační firmy nebo firmy v oblasti peněžnictví a pojišťovnictví mají počet firem v kategorii tvořících hodnotu okolo 40 %. Oproti tomu firmy v oblasti nemovitostí mají počet firem v této kategorii nejméně (přibližně 2x méně než nejlepší kategorie). Toto ovšem v tomto případě odpovídá odlišnému charakteru podnikání, kdy developer je často ve ztrátě do doby prodejů, nebo pronájmů objektů, které realizuje. Zajímavý je i výsledek firmem ve zdravotní a sociální péči, které dosahují nejvyššího procentního podílu firem v kategorii tvořících hodnotu (přes 47 %). Dále pak je zajímavý výsledek zemědělských podniků, které mají nejmenší podíl firem, které dosáhly ztráty. Z opačného konce firmy v oblasti vzdělávání nebo kulturní a rekreační činnosti mají podíl ztrátových firem více jak 50 %.

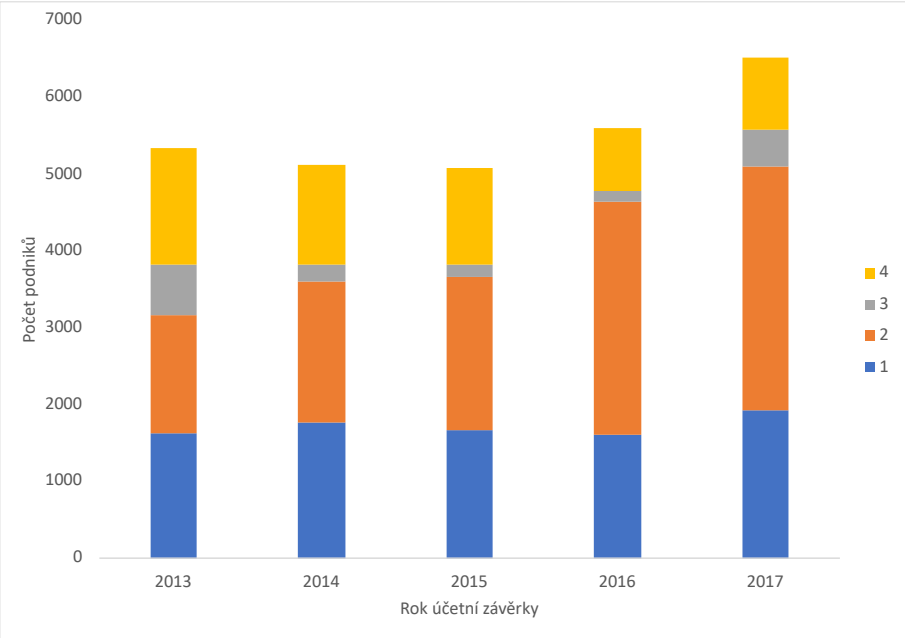
Obrázek 56: Počet podniků podle kategorie a sekce NACE v procentech



Zdroj: Vlastní tvorba.

Poslední kategoriální proměnnou je rok účetní závěrky. Tato kategoriální informace může být důležitá s ohledem na výkonnost celkové ekonomiky. Rovněž řada legislativních opatření a dotací či dalších podpůrných či regulatorních opatření může mít krátkodobý nebo dlouhodobý vliv na výkonnost jednotlivých podniků nebo podniků v určité kategorii (velikost, sekce NACE apod.). Počet podniků v jednotlivých letech je zobrazen na následujícím obrázku. Z obrázku je patrné přibližně rovnoměrné zastoupení podniků v úrovni přibližně od 5 000 do 6 000. Díky přibližné vyrovnanosti množství firem lze předběžně tvrdit, že postupně roste počet firem, které dosahují kladného zisku.

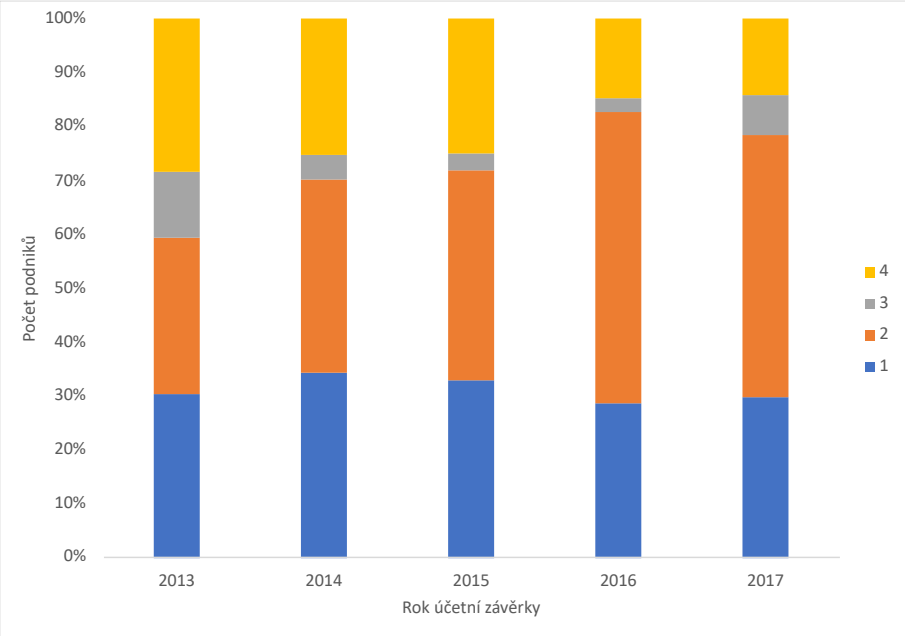
Obrázek 57: Počet podniků podle kategorie a roku účetní závěrky



Zdroj: Vlastní tvorba.

Přesnější analýzu je však možné provést až po vyrovnání souboru, která je zobrazena na následujícím obrázku. Toto vyrovnání potvrdilo předpoklad, že v letech roste podíl firem, které dosahují kladného zisku. Neroste však podíl firem, zařazených do kategorie tvořící kladnou ekonomickou přidanou hodnotu.

Obrázek 58: Počet podniků podle kategorie a roku účetní závěrky v procentech



Zdroj: Vlastní tvorba.

5.1.1.2 Principal component analýza

Principal component analýza slouží k redukci prediktorů, aniž by došlo ke ztrátě informací. Díky tomu je možné snadněji vizualizovat analyzovaná data. Rovněž je možné využít principů strojového učení na menším množství proměnných, díky čemuž dochází k redukci doby učení, což pro některé aplikace využívané v reálném čase (řízení robotů) může být velmi podstatné. V některých případech může využití analýzy sloužit k tvorbě robustnějších modelů, které bude snadněji možné generalizovat, neboť bude složitěji docházet k jejich přetrénování, a naopak bude docházet k odstraňování okrajových prediktorů, které mohou působit jako šumy. V našem případě nejdříve analýzu využijeme pro vizualizaci dat. Později budeme analyzovat robustnost modelů, pokud budou trénovány na výsledcích analýzy.

Analýzu spustíme za pomocí příkazu

```
[dimenze, skore, ~, ~, procenta] = pca(import2{:, 6:end-1});
```

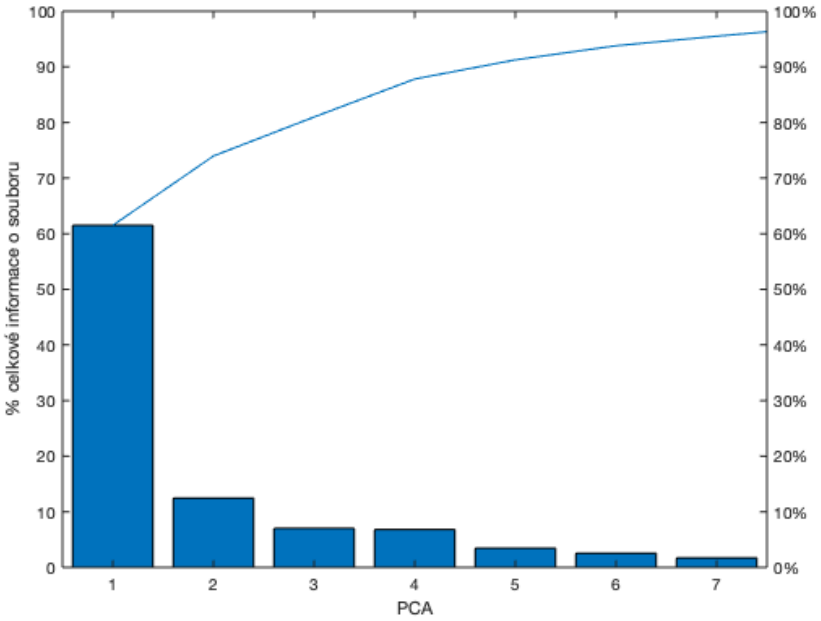
Kde dimenze jsou souřadnice nového prostoru,
skore – představuje nové souřadnice pro jednotlivé záznamy v novém prostoru,
procenta – určuje významnost jednotlivých komponent pro složení celku.

Významnost složek je možné zobrazit pomocí příkazu parreto. Konkrétně následujícím způsobem:

```
pareto(procenta);
xlabel("PCA");
ylabel("% celkové informace o souboru");
```

Výsledek je zobrazen na následujícím obrázku. Z obrázku je pak patrné, že první složka PCA představuje více jak 60 % informací o souboru. Další složka více jak 10 % a dále pak význam každé složky klesá. Jinými slovy, pokud bychom pro predikci použili pouze první 3 složky PCA, pak v nich budeme mít obsaženo téměř 80 % informací o sledovaném souboru, respektive 70 %, pokud bychom použili pouze 2 složky. Toto může být pro řadu aplikací zcela dostačující. V jiných případech ale mohou chybět potřebné informace.

Obrázek 59: Analýza významnosti složek metody PCA pro charakterizaci souboru



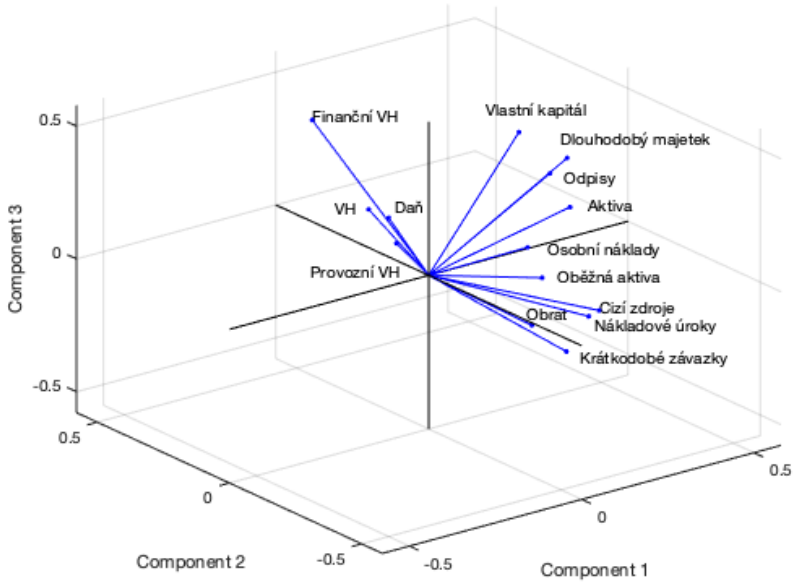
Zdroj: Stehel et al. 2021.

V každé komponentě je určitým způsobem zahrnuta informace z prediktorů původního prostoru. Toto je možné analyzovat za pomocí funkce biplot. Konkrétně:

```
biplot(dimenze(:,1:3), 'varlabels',import.Properties.VariableNames(6:19));
```

Parametr dimenze(:,1:3) vyjadřuje, že budou zahrnuty pouze 3 komponenty. Výsledný graf bude tedy trojrozměrný a je zobrazen na následujícím obrázku. Interpretace trojrozměrného grafu v této podobě je poněkud horší, neboť bez možnosti rotace je obtížně viditelné, jak jednotlivý vektor vytyčuje prostor mezi jednotlivými dimenzemi.

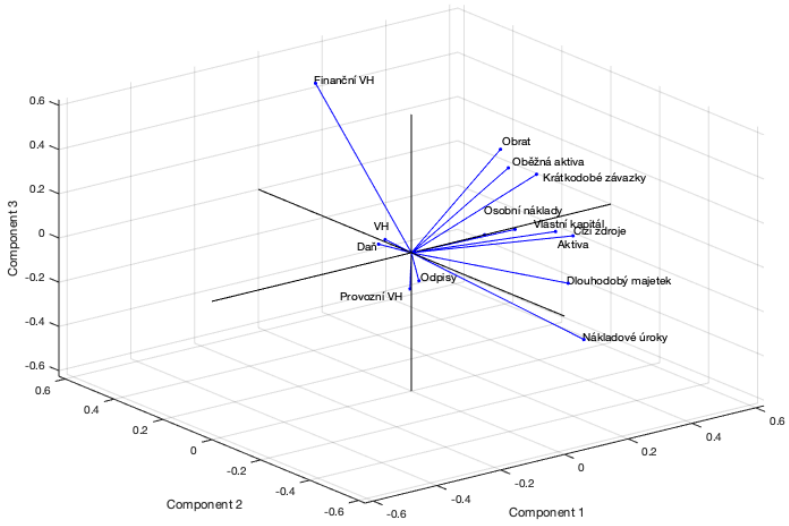
Obrázek 60: Zapojení jednotlivých prediktorů do PCA – normovaná data s odstraněnými šumy



Zdroj: Vlastní tvorba.

Výhoda trojrozměrného zobrazení je, že můžeme porovnat, jak by výsledek PCA analýzy dopadl, kdyby nedošlo k redukci šumů za pomocí příkazu rmoutliers, jak bylo popsáno výše. Výsledek je zobrazen na následujícím obrázku. Z porovnání je patrné, že některé složky by nabyly řádově většího významu v porovnání s ostatními. Například je toto vidět u finančního výsledku hospodaření. U některých složek je pak zcela změněn vektor z pohledu směru i významu – například odpisy. Při porovnání obou grafů je zřejmé, že po odstranění šumu (obrázek 62) je více kompaktní a jednotlivé složky se co do významu zapojují více z hlediska charakteristiky souboru. Rovněž úpravou některých vektorů dojde k zobrazení přirozených vazeb, které jsou logické ze své podstaty. Například odpisy mají téměř totožný vektor (zjistili bychom při rotaci) s dlouhodobým majetkem. Odlišná je však jeho významnost pro jednotlivé komponenty

Obrázek 61: Zapojení jednotlivých prediktorů do PCA – normovaná data bez odstranění šumů



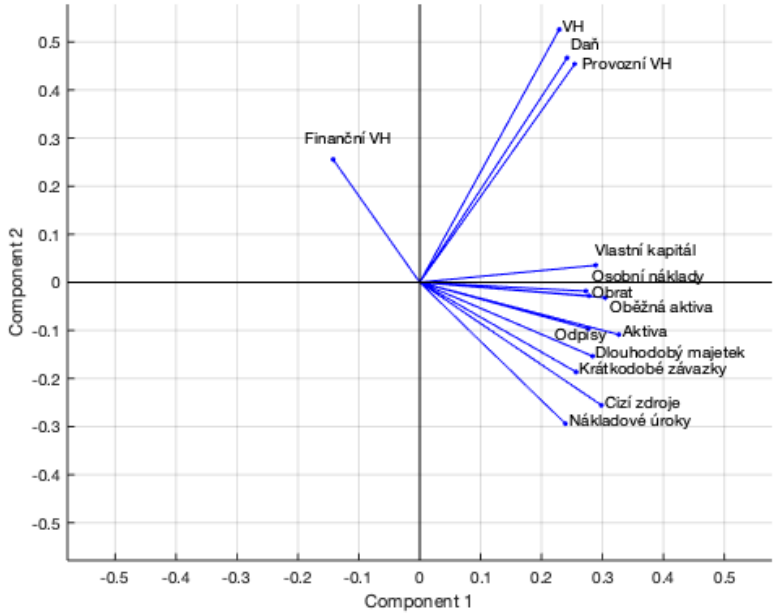
Zdroj: Vlastní tvorba.

Jelikož je interpretace trojrozměrného grafu složitější bez možnosti grafem otáčet a dále díky tomu, že více jak 70 % informací je již obsaženo v prvních dvou složkách PCA budeme dále data vizualizovat za pomoci prvních dvou složek. Význam jednotlivých komponent, které jsme zobrazili za pomocí příkazu:

```
biplot(dimenze(:,1:2), 'varlabels',import.Properties.VariableNames(6:19));
```

je vidět na následujícím obrázku. Je-li vektor v první kvadrantu znamená to, že je pozitivně korelován pro obě komponenty PCA. V našem případě se jedná například o vlastní kapitál. Zasahuje-li vektor do 2. kvadrantu, je negativně korelován s první komponentou, ale pozitivně korelován s druhou komponentou. V našem případě se jedná o finanční výsledek hospodaření. Spadá-li vektor do 4. kvadrantu je pozitivně korelován s první komponentou, ale negativně korelován s druhou komponentou. Jedná se například o cizí zdroje.

Obrázek 62: Zapojení složek prediktorů do PCA pro 2 komponenty



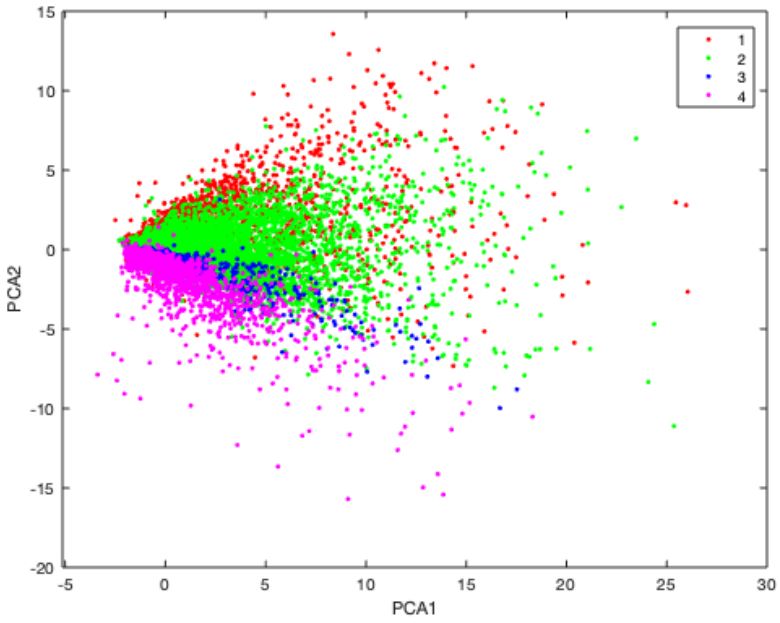
Zdroj: Stehel et al. 2021.

Díky PCA můžeme vizualizovat rozložení podniku ve dvourozměrném prostoru podle jednotlivých sledovaných kategorií. Toto provedeme za pomocí příkazu

```
gscatter(skore(:, 1), skore(:, 2), import2.Kategorie_podniku, 'rgcm');

xlabel("PCA1");

ylabel("PCA2");
```

Výsledek je vidět na následujícím obrázku. Z obrázku je patrné, že podniky jsou v určitém shluku, který nemá přesné hranice. Na druhou stranu ale lze vizuálně rámcově oddělit jednotlivé kategorie podniku. Mělo by tedy být snadno možné najít za pomocí metod strojového učení charakteristicky, za pomoci kterých zatřídíme jednotlivé vzory s vysokou přesností. Bohužel toto tvrzení není zcela správné, neboť data podniků byla zatřízena na základě hospodářského výsledku. Metody strojového učení by tak uvedený skrytý vzorec našly a zatřiďovaly by podniku právě podle těchto prediktorů. Z tohoto důvodu musíme data ještě upravit a odstranit tyto prediktory.

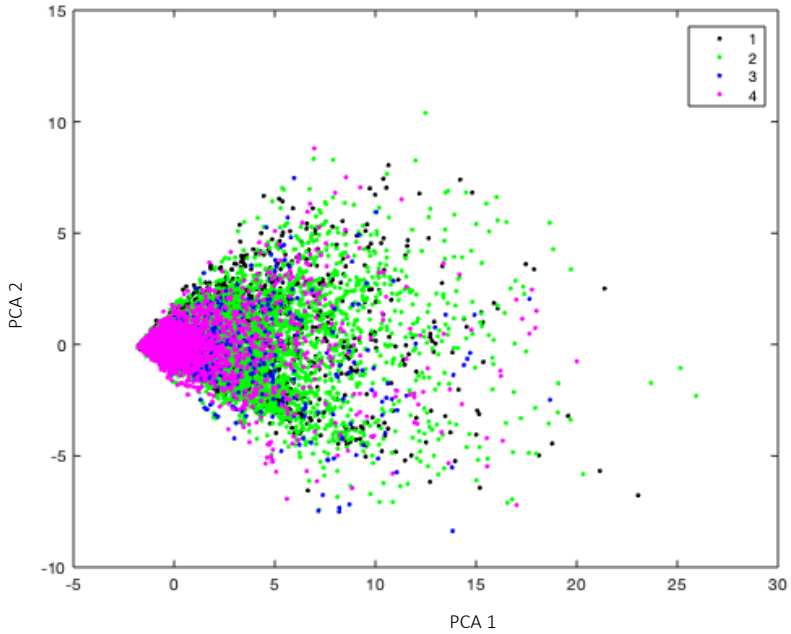
Obrázek 63: Vizualizace rozdělení podniků za pomocí metody PCA



Zdroj: Vlastní tvorba.

Po odstranění prediktorů spojených s finančním, provozním, celkovým hospodářským výsledkem a daněmi získáme vizualizaci, která na rozdíl od předchozí verze není vůbec průkazná a je vidět na následujícím obrázku.

Obrázek 64: Vizualizace podniků bez dat o HV



Zdroj: Vlastní tvorba.

Přesto, že data bez hospodářských výsledků a daní nevykazují zjevnou separaci, lze předpokládat, že za pomocí kategoriálních proměnných bude možné tuto separaci provést. Rovněž je možné, že pokud budeme modely trénovat na úplné množině, tak budou existovat vlastnosti modelů, které umožní odseparovat jednotlivé kategorie velmi dobře. Toto bude předmětem dalšího zkoumání.

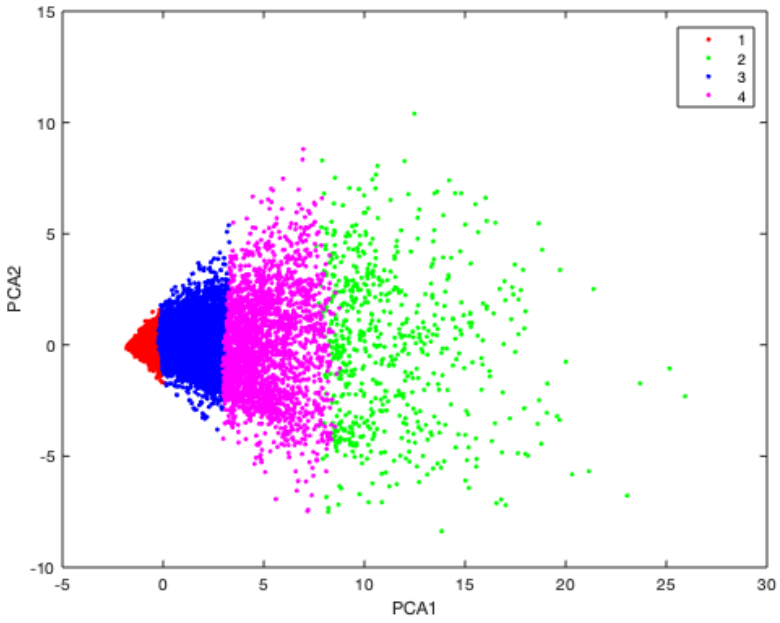
5.1.1.3 K-Means clustering

Tato metoda rozdělí sledovanou množinu do několika skupin. Dělení do skupin ale bude probíhat bez učitele, tedy pro model nebudou známy informace o výstupních kategoriích podniku. Tato metoda zkoumá přirozené zákonitosti mezi daty a dělí tak model podle nejvýraznějších odlišností. Metodu provedeme pomocí příkazu

```
g = kmeans(import3{:, 6:end-1},4,'Replicates',5);

gscatter(skore(:, 1), skore(:, 2), g, 'rgbm');

xlabel("PCA1");

ylabel("PCA2");
```

Příkaz nejdříve převede tabulku na numerickou matici a dále rozdělí data na 4 skupiny. Jelikož je metoda do určité míry závislá na inicializačních podmínkách, které jsou náhodné, opakujeme metodu 5x a dále pracujeme jen s nejlepším výsledkem rozdělení. Příkaz gscatter vykreslí do dvojrozměrného prostoru podniky, kde souřadnice x představují první komponentu PCA a souřadnice y druhou komponentu PCA. Rozdělení do skupiny je provedeno pomocí barev, které reprezentuje proměnná g. Výsledek je zobrazen na obrázku 65. Z obrázku je patrné, že dělení do skupin je zcela odlišné od skupin, které sledujeme (obrázek porovnáváme s obrázkem Vizualizace podniků bez dat o HV). Dělení je prakticky provedeno v určitých hodnotách první komponenty PCA. Toto však pro naši analýzu nebude představovat přínos.

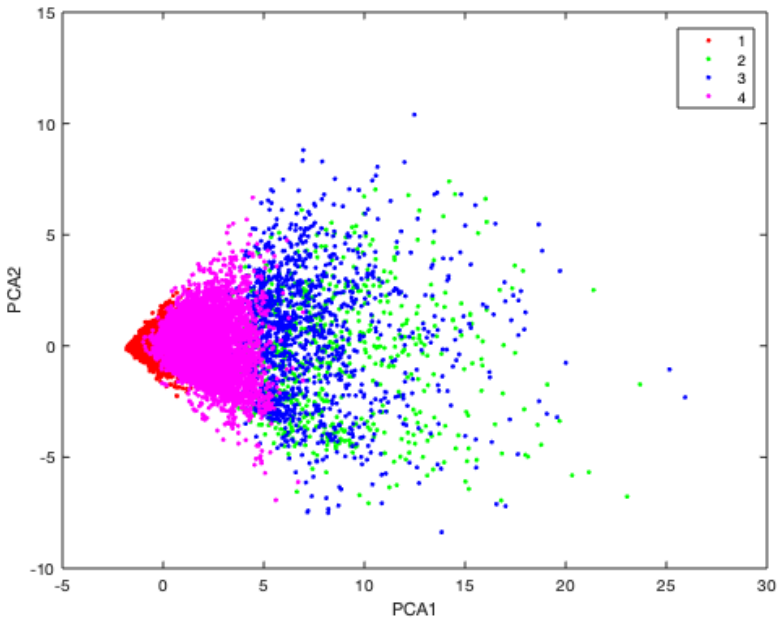
Obrázek 65: Dělení dat do skupin za pomocí metod k-menas bez dat o HV a daních



Zdroj: Vlastní tvorba.

Pro úplnost provedeme rozdělení do skupin ještě jednou se stejnými parametry. V tomto případě však nevynecháme údaje o hospodářských výsledcích a daních. Výsledek pak porovnáme s obrázkem 64. Vizualizace podniků bez dat o HV. Nově vygenerovaný graf je vidět na následujícím obrázku.

Obrázek 66: Dělení dat do skupin za pomocí metod k-menas s daty o HV a daních



Zdroj: Vlastní tvorba.

Nově vytvořený obrázek 66 je mnohem podobnější obrázku 64 Vizualizace podniků bez dat o HV. Z tohoto důvodu lze usoudit, že data o HV a daních mají vliv na přirozené rozdělení podniků do skupin pomocí této metody. Bez těchto informací ale tato metoda nebude příliš využitelná pro samotnou klasifikaci. Toto neznamená, že by daná metoda nefungovala na datech dobře. Jenom dochází k tomu, že některé charakteristiky podniků jsou pro metodu významnější, a tak probíhá klasifikace na základě těchto charakteristik, které mohou být využity v jiných analýzách. Může se například jednat o velikost podniku, jeho rizikovost (likviditu) apod. Zkoumání rozdělení do skupin je nad rámec cíle práce, a proto se jím nebudeme dále zabývat.

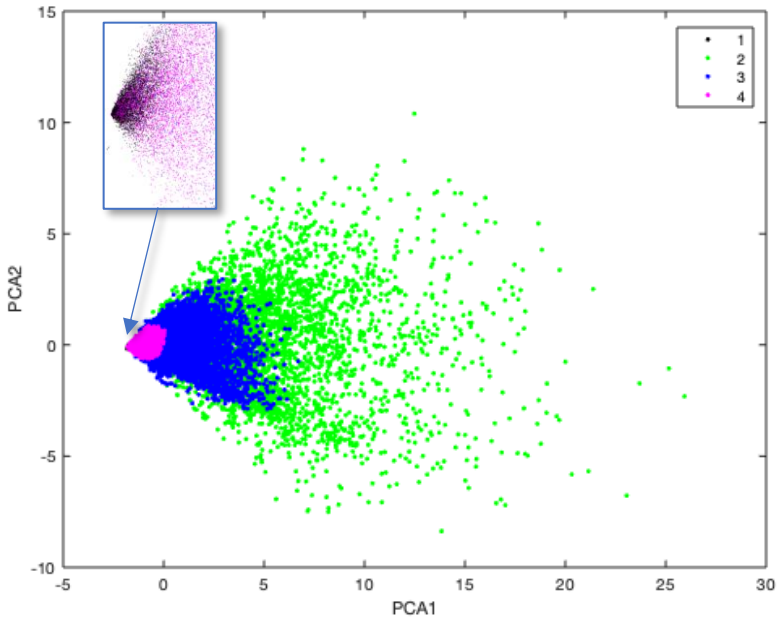
5.1.1.4 Gaussian Mixture Models

Gaussian mixture model funguje podobně jako model k-means s tím rozdílem, že dokáže určit pravděpodobnost zatříděné množiny. Zohlednění míry pravděpodobnosti může mít vliv na celkové dělení clusterů, a proto metodu provedeme za pomocí následujících příkazů:

```
gm = fitgmdist(table2array(import3(:, 6:end-1)),4);

[g,~,p] = cluster(gm,table2array(import3(:, 6:end-1)));

gscatter(skore(:, 1), skore(:, 2), g, 'kgbm');
```

První příkaz fitgmdist vytvoří model a druhý příkaz z vytvořeného modelu vytvoří skupiny, které uloží do vektoru g. Zároveň vytvoří matici s pravděpodobností začlenění jednotlivých podniků do dané skupiny. Poslední příkaz pak vše vizualizuje, jak tomu bylo v předchozích případech. Výsledný graf je zobrazen na následujícím obrázku. Z obrázku je patrné, že metoda rozdělila data odlišně. Opět se jedná o data bez HV a daní. Rozdělení dat daleko více zohledňuje i další komponenty, a proto zatřízení není s tak ostrými hranicemi, jako tomu bylo v předchozím případě. Z důvodů množství dat došlo k překrytí první kategorie podniku (reprezentuje černá barva) čtvrtou kategorií podniku (růžová barva). Aby nedošlo ke ztrátě vypovídající schopnosti grafu byl obrázek doplněn minigrafem, který zvětšuje pomyslný střed nejkoncentrovanějších dat. Z tohoto minigrafu je patrné i zastoupení první kategorie podniku. Z dat je zřejmé, že i v tomto případě nebude přirozené dělení příliš využitelné pro klasifikaci podniků, jako tomu bylo v předchozím případě.

Obrázek 67: Rozdělení dat na základě Gaussian mixture modelu



Zdroj: Vlastní tvorba.

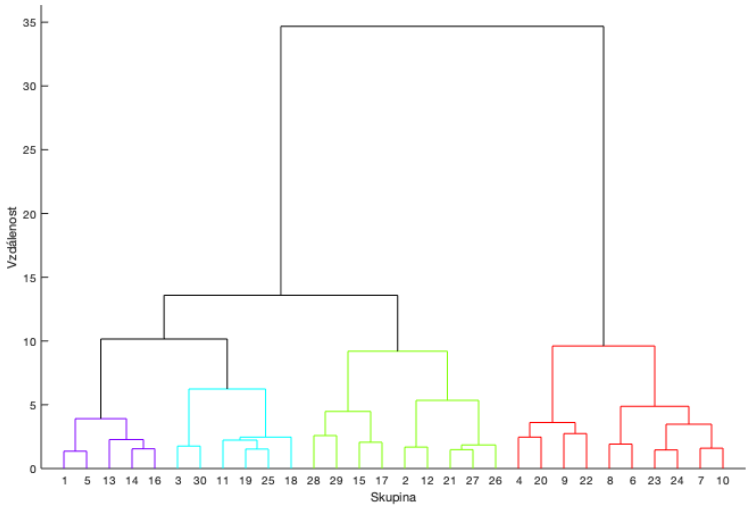
5.1.1.5 Hierarchické dělení (Hierarchical Clustering)

Poslední metoda, která bude použita pro analýzu přirozených charakteristik dat, je hierarchické dělení. Tato metoda porovnává vzdálenosti mezi jednotlivými množinami a na základě těchto vzdáleností rozděluje data do hierarchického stromu. Metodu provedeme pomocí příkazů:

```
Z = linkage(import4{:, 6:end-1},'ward','cosine');

xlabel("Skupina");

ylabel("Vzdálenost");

c = cluster(Z,'Maxclust',4);

dendrogram(Z, 'ColorThreshold', 10)
```

Příkaz *linkage* vytvoří model stromové struktury. Tento model do 4 úrovně následně rozdělí podniky za pomocí příkazu *cluster*. Následně je vše vykresleno pomocí příkazu *dendrogram*. Ostatní parametry slouží pro interpretovatelnost výsledků, které jsou vidět na následujícím obrázku. Na obrázku jsou vidět vzdálenosti mezi jednotlivými skupinami (osa y). Z obrázku je patrné, že největší vzdálenosti jsou mezi prvními dvěma skupinami. Následně je relativně velká vzdálenost mezi 2. a 3. skupinou. Rozdíl vzdáleností mezi 4., 5. a 6. skupinou je však relativně malý.

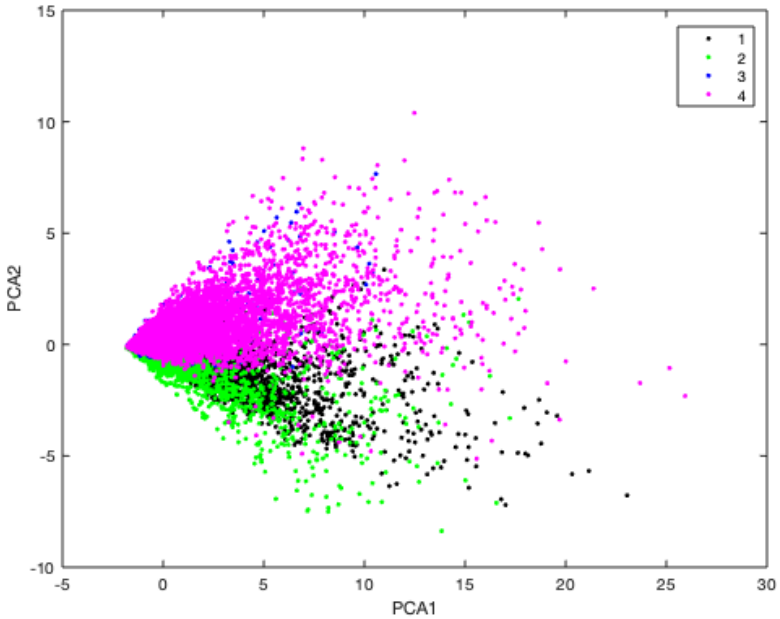
Obrázek 68: Stromová struktura rozdělení dat



Zdroj: Vlastní tvorba.

Zatřízení jednotlivých skupin bylo provedeno v předchozím příkaze cluster. Nyní proto můžeme vizualizovat data pomocí příkazu gscatter, jako tomu bylo v předchozích případech. Výsledek je vidět na následujícím obrázku. Na obrázku opět dochází z důvodu rozsahu k překryvu modré barvy barvou růžovou. Na rozdíl od předchozího případu však tento překryv je po celé délce x grafu, a tudíž by samostatné zobrazení nepomohlo lepší vizualizaci dat.

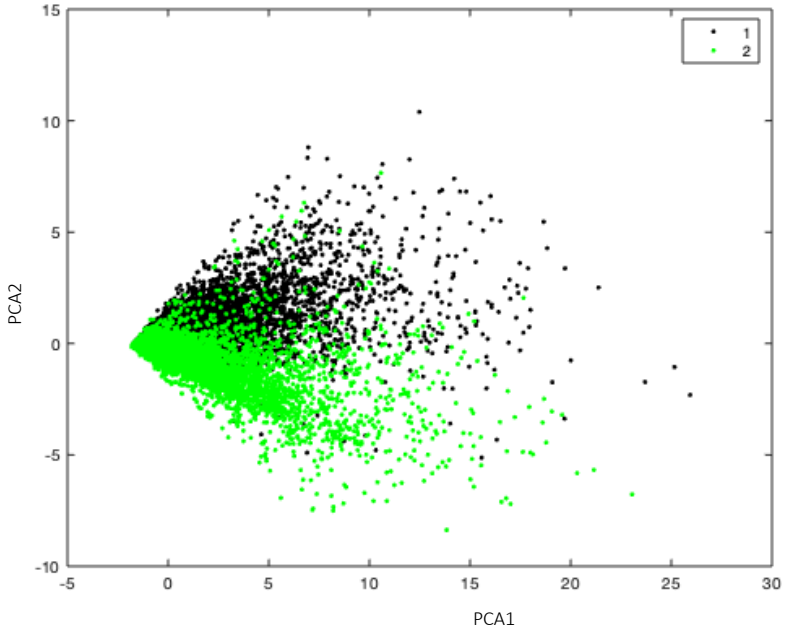
Obrázek 69: Rozdělení dat pomocí hierarchického členění



Zdroj: Vlastní tvorba.

Z obrázku 69 je dále patrné, že rozdělení dat oproti předchozím metodám jakoby mnohem více zohledňuje druhou komponentu. Termín „jakoby“ je použit, protože metody jsou na sobě samozřejmě nezávislé, pouze opticky lze spatřit, že metody zohlednily shodné skryté zákonitosti v datech. Pokud však data porovnáme s metodou PCA obrázku 64, tak stále musíme konstatovat, že tato metoda nebude sama o sobě schopna snadno klasifikovat data. Rozdělení do různých skupin tak opět proběhlo na základě jiných parametrů. Jelikož je vzdálenost mezi podniky pro 1. dvě skupiny veliká zobrazíme vizuálně i tento výsledek, který je vidět na obrázku 70. Z obrázku je patné, že první dvě skupiny se poměrně hodně překrývají, což nebylo vidět z předchozí vizualizace. Z tohoto tedy vyplývá, že hieratické členění ani nenachází hlavní rozdílnost, jako je tomu u druhé komponenty PCA. Předchozí tvrzení tedy zřejmě není správné. Dále tento překryv naopak více odpovídá zobrazení obrázku 63 a tudíž je možné, že stromová struktura bude sloužit k optimálnímu zatřídění dat. Určitě to ale nebude možné použít pouze při analýze dat bez učitele. Naopak se budou muset použít metody s učitelem.

Obrázek 70: Zobrazení dat na základě rozdělení do 2 skupin za pomocí hierarchického členění



Zdroj: Vlastní tvorba.

5.1.2 Nejbližší soused – KNN (Nearest Neighbor Classification)

Metoda je založena, jak název napovídá, na posuzování nového vzoru na základě již zatřízených vzorů (Altman, 1992). Podle nejbližších vzorů se pak určí, do jaké skupiny podnik patří. Tato metoda je rychlá a spolehlivá, pokud nejsou data příliš zatížená šumem. Model vytvoříme na základě příkazů níže. V první fázi nejdříve rozdělíme data na trénovací a testovací. Na rozdíl

od neuronových sítí zde nemusíme dělit data ještě na validační. Rozdělení provedeme následujícími příkazy:

```
c = cvpartition(import4.Kategorie_podniku,'HoldOut',0.4);

testIndex = test(c);

trainingIndex=training(c);

testTable = import4(testIndex, :);

trainTable = import4(trainingIndex, :);
```

První příkaz *cvpartition* rozdělí množinu na 2, kde jedna bude mít zastoupení 60 % a druhá 40 % v souboru. Druhý a třetí příkaz (*test, training*) získají indexy pro jednotlivé podniky, které budou zahrnuty do množiny trénovačích nebo testovacích dat, přičemž trénovací data představují 60 %. Následně se vše uloží do samostatných tabulek za pomocí běžného indexování v Matlabu.

Samotný model vytvoříme na základě příkazu

```
mdlKnn = fitcknn(trainTable(:, 6:end), 'Kategorie_podniku', 'NumNeighbors',3);
```

V tomto případě jsme metodu použili tak, že vstupem je tabulka (může být i matice), kde poslední sloupec (Kategorie_podniku) představuje výstup, podle kterého se má model učit. Modle bude klasifikovat nová data na základě tří nejbližších sousedů. Výsledné zatřízení bude provedeno na základě vyššího počtu ze 3 sousedů, kteří budou ve stejné kategorii. Pokud budou 3 nejbližší sousedé z různých kategorií, dojde k rozhodnutí na základě nejbližšího souseda.

Pro analýzu úspěšnosti modelu použijeme následující funkce:

```
resubLossKnn = resubLoss(mdlKnn)

lossKnn = loss(mdlKnn, testTable(:, 6:end))
```

První funkce nám říká, k jakému množství chybně zatřízených množin dojde v případě trénovací množiny. Druhý příkaz toto samé hodnotí na testovací množině. V našem případě je výsledek

```
resubLossKnn = 0.3259

lossKnn = 0.5734
```

Uvedené výsledky ukazují, že model na trénovačích datech správně zatřídí téměř 70 % dat. V případě testovacích dat je to ale méně jak 50 %. Predikci nových dat provedeme za pomocí příkazu

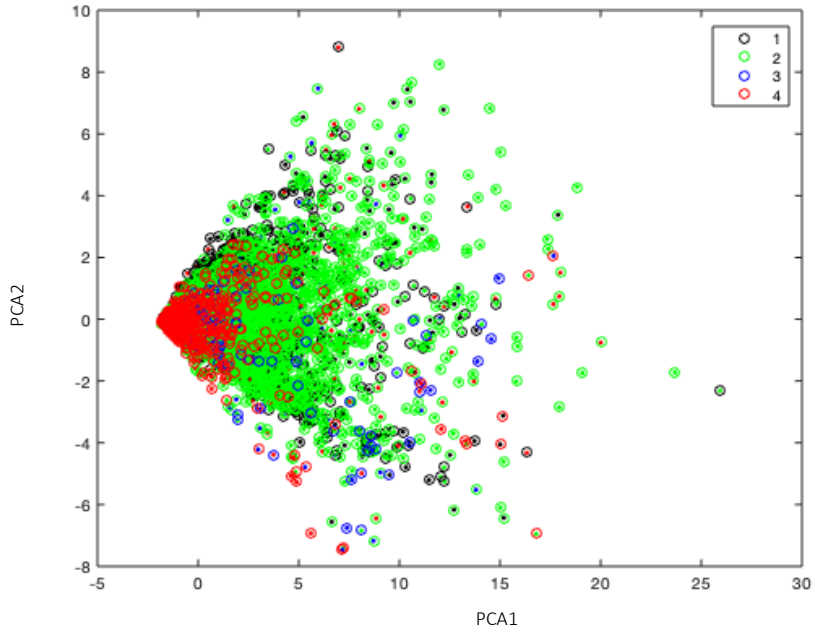
```
predikceKnn = predict(mdlKnn, testTable(:, 6:end));
```

Následně pak můžeme vykreslit graf rozdělení dat na základě reality a na základě predikce. Tyto grafy prolneme přes sebe s tím, že v případě predikce budou značkou dat kroužky ve stejné barvě. Tam kde bude odpovídat barva bodu s barvou kroužku model zatřídil podnik správně. V opačném případě došlo k chybnému zatřízení. Tyto úkony provedeme následujícími příkazy:

```
gscatter(skore(testIndex, 1), skore(testIndex, 2), testTable.Kategorie_podniku);

hold on

gscatter(skore(testIndex, 1), skore(testIndex, 2), predikceKnn, 'rgcm', 'o');

xlabel("PCA1");

ylabel("PCA2");
```

Funkce příkazu gscatter se nemění jako v předchozích případech. Nově však přibyl příkaz hold on, který umožní prolnout grafy přes sebe. Parametry 'rgcm' a 'o' pak určují barevnost a dále značku pro predikované hodnoty. Výsledek je vidět na následujícím obrázku.

Obrázek 71: Vizualizace zatřídění dat KNN



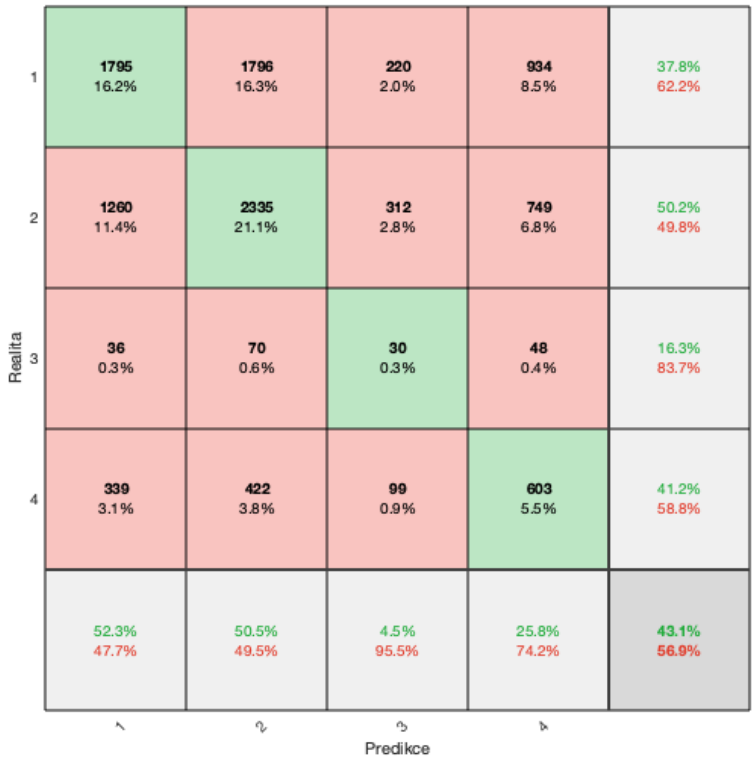
Zdroj: Vlastní tvorba.

Předchozí vizualizace je s ohledem na množství a koncentraci dat obtížně interpretovatelná. Z tohoto důvodu bude využita další metoda v podobě konfusní matice. Tuto zobrazíme za pomocí příkazů:

```
plotconfusion(testTable.Kategorie_podniku, predikceKnn);

xlabel("Predikce");

ylabel("Realita");
```

Výsledek je vidět na následujícím obrázku. Z výsledku je zřejmé, že model relativně dobře zatřiďuje první dvě množiny, tedy že podnik dosáhne kladné hodnoty EVA, nebo že dosáhne kladného výsledku hospodaření převyšujícího bezrizikovou sazbu. Model naopak selhává při zatřiďování podniků, které budou mít ještě kladný výsledek hospodaření, který bude zároveň nižší než bezriziková sazba. V případě podniků, které budou v dalším roce ztrátové, je zatřízení na úrovni náhodné procházky a model tak má spolehlivost v tomto případě přibližně 25 %.

Obrázek 72: Konfusní matice KNN



Zdroj: Vlastní tvorba.

V předchozím případě jsme použili model s nastavením, které znamenalo, že nová klasifikace probíhá na základě tří nejbližších sousedů. Toto může být nejlepší nastavení pro model, ale rovněž může být zcela chybné a při jiných hodnotách tohoto parametru můžeme teoreticky

dosáhnout lepších výsledků. V tomto ohledu tedy provedeme optimalizaci hyperparametrů modelu za pomocí cyklu *for*, který bude ukládat výsledky jednotlivých testů do samostatného vektoru. Konkrétně optimalizaci hyperparametrů provedeme pomocí příkazů

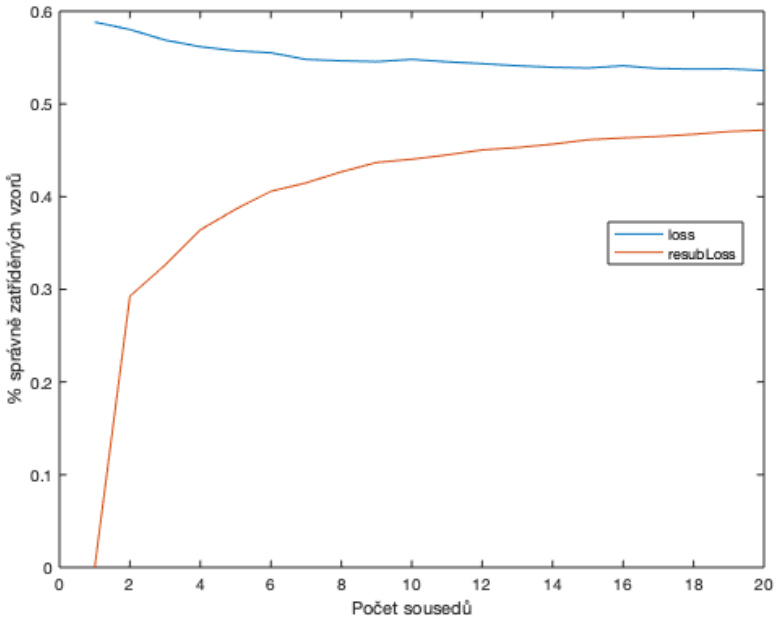
```
for i = 1:20

mdlKnn = fitcknn(trainTable(:, 6:end), 'Kategorie_podniku', 'NumNeighbors',i);

lossKnn(1, i) = loss(mdlKnn, testTable(:, 6:end));

resubLossKnn(1, i) = resubLoss(mdlKnn);

end
```

Příkaz for je cyklus, který proběhne dle nastavených parametrů 20x. Pokaždé se bude inkrementovat (zvyšovat) hodnota parametru o 1. Tato proměnná se bude dosazovat při trénování do modelu a bude se tak zvyšovat v průběhu každého cyklu počet sousedů, na základě kterého se vypočítá zatřídění nových vzorů. Následně proběhne validace modelu jako v předchozím případě. Jediný rozdíl bude spočívat v tom, že se data budou postupně ukládat do vektoru pro pozdější analýzu. Po provedení cyklu dojde k zobrazení dat pomocí následujícího příkazů:

```
plot(lossKnn);

hold on

plot(resubLossKnn);

xlabel("Počet sousedů");

ylabel("% správně zatříděných vzorů");

legend("loss", "resubLoss");
```

Příkazy zobrazí grafy na základě uložených dat z předchozího cyklu. Zároveň graf popíší pro snadnou interpretovatelnost. Výsledek je vidět na následujícím obrázku. Oranžová čára reprezentuje chybovost zatříděných dat na trénovacím modelu. Z principu metody je zřejmé, že při jednom sousedovi bude chybovost na trénovacích datech nulová. Oproti tomu chybovost na testovacích datech je nejvyšší a pohybuje se o kolo 60 %. Hodnota chyby u trénovacích dat postupně roste, oproti tomu hodnota chyby u testovacích dat naopak postupně klesá. Od zhruba 10 sousedů je zřejmé, že model dosáhl svých maxim a téměř se nezlepšuje. Další optimalizace proto bude probíhat na této hodnotě.

Obrázek 73: Optimalizace hyperparametrů KNN – počet sousedů



Zdroj: Vlastní tvorba.

Další možností optimalizace je určení, jakým způsobem bude probíhat určení vzdálenosti. V této oblasti umožňuje příkaz varianty, které jsou patrné z kódu níže. Nastavení všech těchto parametrů opět provedeme pomocí cyklu for a následujících příkazů:

```
parametr = ["cityblock" "chebychev" "correlation" "cosine" "euclidean" "hamming" "jaccard" "mahalanobis" "minkowski" "seuclidean" "spearman"]

for i = 1:11

mdlKnn = fitcknn(trainTable(:, 6:end), 'Kategorie_podniku', 'NumNeighbors', 10, 'Distance', parametr(i));

lossKnnP(1, i) = loss(mdlKnn, testTable(:, 6:end));

resubLossKnnP(1, i) = resubLoss(mdlKnn);

end
```

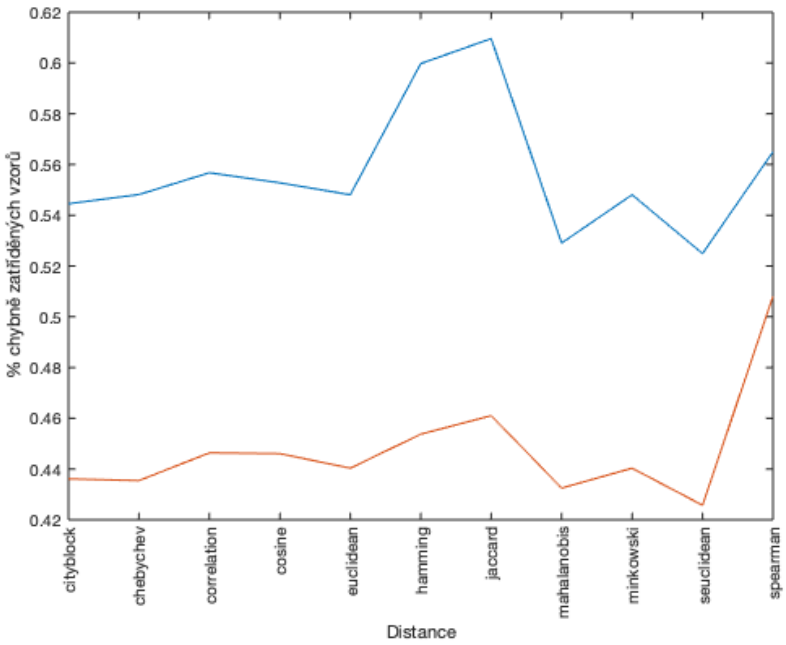
Oproti předchozímu případu jsme si nejdříve vytvořili pole s textem. Následně jsme upravili počet cyklů příkazu for, aby odpovídal počtu parametrů, které budeme testovat. Následně jsme provedli trénování modelu s tím, že u specifického parametru, který určuje vlastnost distance jsme provedli odkaz na pole, u kterého se postupně zvětšuje index. Následně vše ukládáme do proměnných, abychom mohli vše analyzovat a zobrazit jako v předchozím případě. Toto provedeme následujícími příkazy:

```
plot(lossKnnP);
```

```
hold on

plot(resubLossKnnP);

xlabel("Distance");

xticklabels(parametr);

ylabel("% správně zatříděných vzorů");

legend("loss", "resubLoss");
```

Příkazy jsou velmi podobné předchozímu zobrazení. Je zde však rozdíl v proměnných, které obsahují údaje o optimalizaci hyperparametrů vztahujících se k parametru Distance. Dále je pak mírně odlišný způsob popisu grafu s ohledem na to, že na ose x jsou jednotlivá specifikace parametru. Výsledek je vidět na následujícím obrázku, ze kterého je patrné, že jednotlivé specifikace nemají příliš rozdílný vliv na výsledek. Základní specifikace euclidean dosahuje téměř nejlepších výsledků. Všechny ostatní metody se v chybně zařazených vzorech pohybují mezi 44 % až 50 % u trénovacích dat a od 50 do 60 % chybně zařazených dat u testovací množiny. Jelikož defaultní nastavení je jedno z nejlepších budeme další optimalizaci provádět při tomto nastavení.

Obrázek 74: Optimalizace hyperparametrů KNN – metoda vzdálenosti



Zdroj: Vlastní tvorba.

Dalším optimalizovaným parametr bude ScoreTransform. Tento parametr pracuje již s vypočtenými vzdálenostmi a dále je upravuje, aby určil podle počtu sousedů výsledné zatřídění. Parametr umožňuje varianty, které jsou uvedeny v následující tabulce.

Obrázek 75: tabulka variant parametrů ScoreTrasform

Hodnota	Popis	
'doublelogit'	$\frac{1}{1 + e^{-2x}}$	(31)
'invlogit'	$\log\left(\frac{x}{(1 - x)}\right)$	(32)
'ismax'	Nastaví skóre pro třídu s největším skóre na 1 a nastaví skóre pro všechny ostatní třídy na 0	
'logit'	$\frac{1}{1 + e^{-x}}$	(33)
'none' or 'identity'	x (bez transformace)	
'sign'	–1 pro $x < 0$ 0 pro $x = 0$ 1 pro $x > 0$	(34)
'symmetric'	$2x - 1$	
'symmetricismax'	Nastaví skóre pro třídu s největším skóre na 1 a nastaví skóre pro všechny ostatní třídy na –1	
'symmetriclogit'	$\frac{2}{1 + e^{-x}} - 1$	(35)

Zdroj: Matlab documentation 2021.

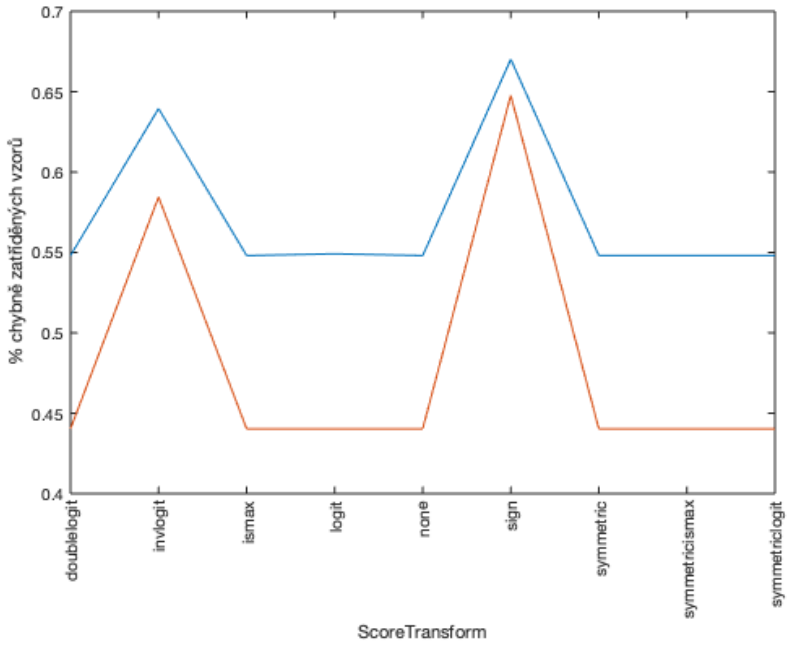
Samotnou optimalizaci provedeme za pomocí následujícího kódu

```
parametr2 = ["doublelogit" "invlogit" "ismax" "logit" "none" "sign" "symmetric" "symmetricismax" "symmetriclogit"]

for i = 1:9

mdlKnn = fitcknn(trainTable(:, 6:end), 'Kategorie_podniku', 'NumNeighbors', 10, 'ScoreTransform', parametr2(i));

lossKnnP2(1, i) = loss(mdlKnn, testTable(:, 6:end));

resubLossKnnP2(1, i) = resubLoss(mdlKnn);

end
```

Oproti předchozímu případu je zde jiný výčet textového pole. Dále je upraven počet cyklů, které budou realizovány, na 9. Ostatní parametry zůstávají shodné, pouze výsledek je uložen do odlišných proměnných. Tyto následně zobrazíme shodnými příkazy, jako tomu bylo v předchozím případě. Výsledek je vidět na následujícím obrázku. Stejně jako v předchozím případě je vidět, že optimalizace pro daná data není příliš účinná a standardní nastavení

dosahuje nejlepších výsledků, stejně jako 7 dalších metod. Další dvě metody naopak výsledek zhoršily jak v případě trénovacích, tak v případě testovací množiny.

Obrázek 76: Optimalizace hyperparametrů KNN – ScoreTransform



Zdroj: Vlastní tvorba.

5.1.3 Stromová klasifikace (Classification trees)

Další metodou, kterou použijeme pro klasifikaci je stromová klasifikace (classification trees). Tato metoda je velmi využitelná, neboť je snadno analyzovatelné, jak se počítač rozhoduje. Nenastává tak efekt černé skříňky, který je problematický zejména u konvolučních neuronových sítí. Nevýhoda metody spočívá zejména v tom, že je velmi závislá na předložených datech. Z tohoto důvodu existují určité optimalizace, které tento nedostatek odstraňují a kterým se budeme věnovat dále.

V první fázi budou data rozdělena dle stejného klíče jako tomu bylo v případě metody KNN viz výše. Dále dojde k natrénování modelu a základnímu ověření následujícími příkazy:

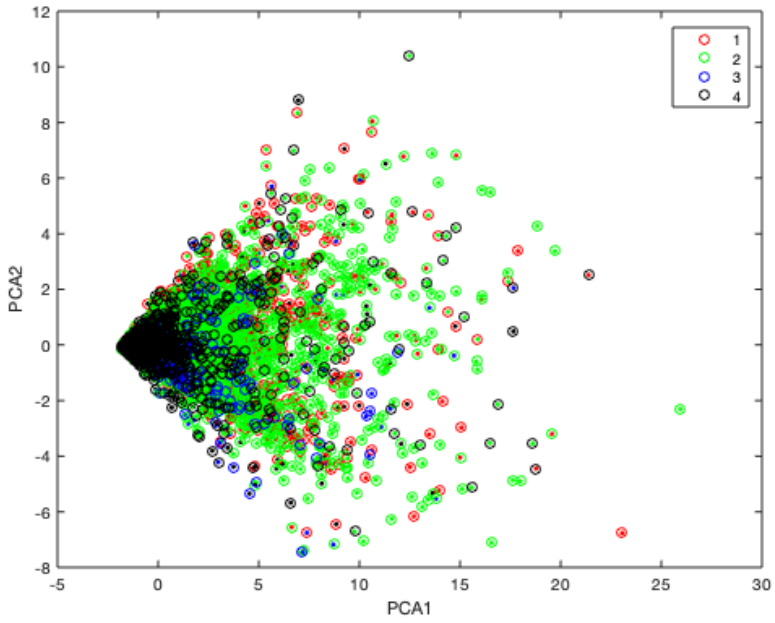
```
mdlTree = fitctree(trainTable, 'Kategorie_podniku');

resubLossTree = resubLoss(mdlTree)

lossTree = loss(mdlTree, testTable)

predikceTree = predict(mdlTree, testTable);
```

První příkaz *fitctree* natrénuje model. Na rozdíl od metody KNN zde není redukce vstupu pouze na numerické proměnné, ale jsou zahrnuty i kategoriální proměnné. Redukce proměnných byla provedena v předchozím případě zápisem *trainTable(:, 6:end)*, který vyjadřuje, že se do modelu zahrnou data až od 6. sloupce. Prvních 5 sloupců v tabulce tedy obsahuje kategoriální proměnné. Predikce nových dat z testovací tabulky je následně analyzována a porovnána se skutečnými daty. Toto porovnání je provedeno výše uvedeným příkazem *loss*. Jako doplněk je analyzován i počet špatně zatříděných vzorů na trénovacích datech za pomocí příkazu resubLoss. Výsledkem příkazů je v tomto případě:

```
resubLossTree = 0.1526

lossTree = 0.5512
```

Model je tedy velmi spolehlivý na trénovacích datech, ale bohužel již méně spolehlivý na testovacích datech, kde správně zatřídí méně jak polovinu vzorů. Vizuálně výsledek třídění realizujeme stejně jako v předchozím případě za pomocí příkazů gscatter, které překreslíme přes sebe. Přičemž originální data budou zobrazena tečkou a predikovaná data budou reprezentována kolečkem. Při správném zatřízení bude barva korespondovat. V opačném případě došlo k chybnému zatřízení. Příkazy jsou stejné jako v předchozím případě, proto je zde nebudeme uvádět. Výsledek je vidět na následujícím obrázku. Na obrázku je vidět, že například firma zcela vpravo je ve skutečnosti v druhé kategorii podniků (záporná hodnota EVA, ale tvoří zisk nad úrovní bezrizikové sazby). Tento podnik byl správně klasifikován i tímto modelem. Stejně tomu je v případě druhého podniku, který leží o jednu úroveň blíže. Podnik byl správně zařazen, přičemž i v realitě spadá do kategorie 1 (kladná hodnota EVA). Naopak chybně klasifikovaný je podnik nejvýše v grafu. Tento podnik je v realitě ve 2. kategorii, ale model jej zařadil do 4. kategorie.

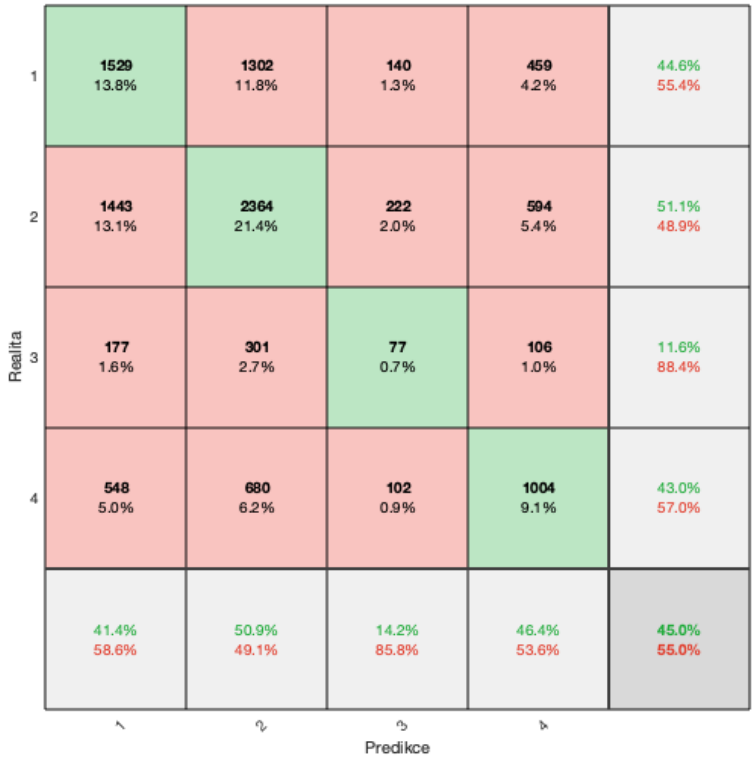
Obrázek 77: Vizuální analýza spolehlivosti stromového modelu



Zdroj: Vlastní tvorba.

Pro lepší analýzu zobrazíme i konfusní matici. Opět jsou zde shodné příkazy, proto je nebudeme vypisovat. Výsledek je vidět na následujícím obrázku. Model je nejspolehlivější při zatřiďování druhé kategorie podniku. Při tomto zatřiďování dosahuje spolehlivosti přes 50 %. Dále pak klasifikuje s menší přesností podniky v 1. skupině a 4. skupině. Minimální spolehlivost má model při zatřiďování 3. kategorie. Tato kategorie je obecně věcně i technicky problematická, a proto s ní měl problém i předchozí model. Věcný problém spočívá v predikování toho, že firma dosáhne minimálního, avšak kladného výsledku. Technický problém spočívá v relativně nižší množině trénovacích a testovacích dat. Porovnáme-li oba modely při základním nastavení, pak model Tree má lepší schopnost zatřízení tohoto vzoru.

Obrázek 78: Konfusní matice – stromový model



Zdroj: Vlastní tvorba

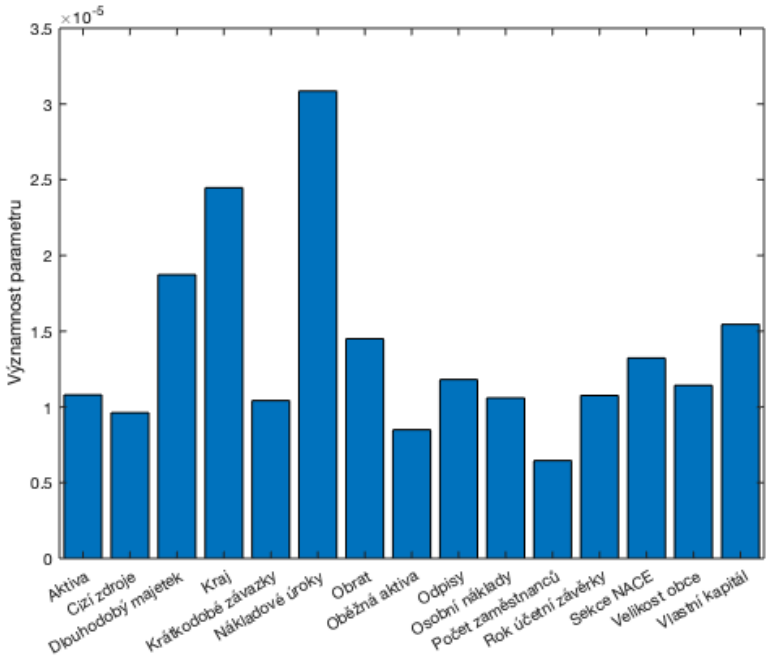
5.1.3.1 Významnost kritérií

V případě stromové klasifikace lze snadno určit jaké prediktory hrají významnou roli při rozhodování. Toto je možné provést za pomocí následujících příkazů:

```
p = predictorImportance(mdlTree);

kategorie = categorical(trainTable.Properties.VariableNames(1:end-1));

bar(kategorie, p);
```

Výsledek je vidět na následujícím obrázku. Na uvedeném obrázku je vidět, že model má nejdůležitější kritérium velikost nákladových úroků. Toto může dávat i věcný smysl, neboť řada společností může dosahovat ztráty při financování velkých projektů z důvodu vysokých úroků (zejména developerské společnosti). Dalším kritériem je kraj, celková velikost dlouhodobého majetku, vlastní kapitál apod. Nejmenší vliv přikládá model počtu zaměstnanců a oběžnému majetku.

Obrázek 79: Významnost parametru pro rozhodování stromu



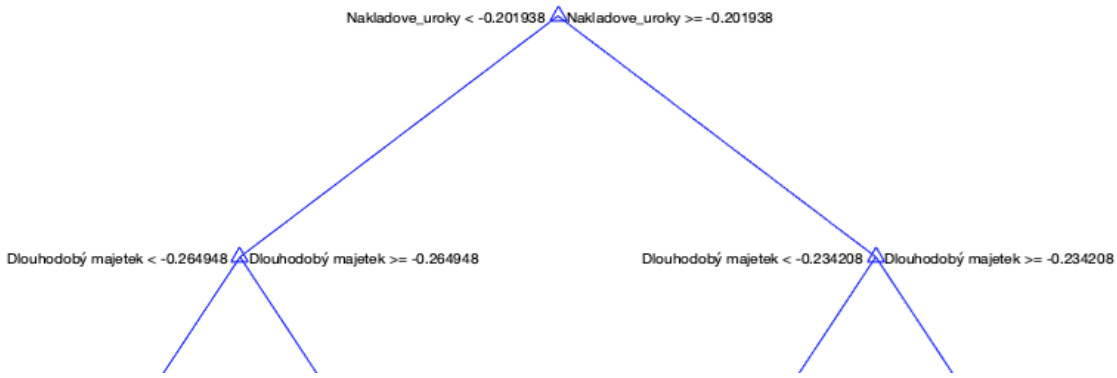
Zdroj: Vlastní tvorba.

Jak bylo popsáno výše stromová struktura je výhodná, neboť umožňuje snadno analyzovat, jak se počítač rozhoduje při zatřiďování vzoru. Toto je možné posoudit za pomocí následujících příkazů:

```
mdlTree = fitctree(trainTable, 'Kategorie_podniku', 'MaxNumSplits',7);

view(mdlTree,'mode','graph');
```

Pro účely zobrazení byl modelu upraven parametr *'MaxNumSplits',7*, který následně určuje velikost rozhodovacího stromu. Kdybychom použili celou šíři, tak by nebylo možné strom zobrazit v tištěné podobě, neboť obsahuje okolo 80 pater. Příkaz *view* následně rozhodovací strom vykreslí. Výsledek je vidět na následujícím obrázku. V první fázi rozhodování algoritmus posoudí výši nákladových úroků. Jsou-li nákladové úroky vyšší jak přibližně -0,486 (pracujeme s normalizovanými daty), vstoupí algoritmus do pravé větve. Zde následně posuzuje výši dlouhodobého majetku. Rozhodovací hladina je zde přibližně -0,41. Je-li dlouhodobý majetek menší, je podnik zatřízen do 1. kategorie podniků. Je-li dlouhodobý majetek vyšší, je zatřízen do 2. skupiny podniků. V případě, že by byly nákladové úroky nižší než přibližně 0,486, rozhodoval by se algoritmus v další fázi na základě vlastního kapitálu. Je-li vlastní kapitál menší jak přibližně -0,546, je podnik zařazen do 1. skupiny, v opačném případě je zařazen do 4. skupiny. Algoritmus již dále nepokračuje, neboť byl pro účely publikace omezen do této úrovně členění. Pro celkové zatřízení, které jsme provedli na datech, má strom přibližně 80 pater.

Obrázek 80: Rozhodování stromového modelu



Zdroj: Vlastní tvorba.

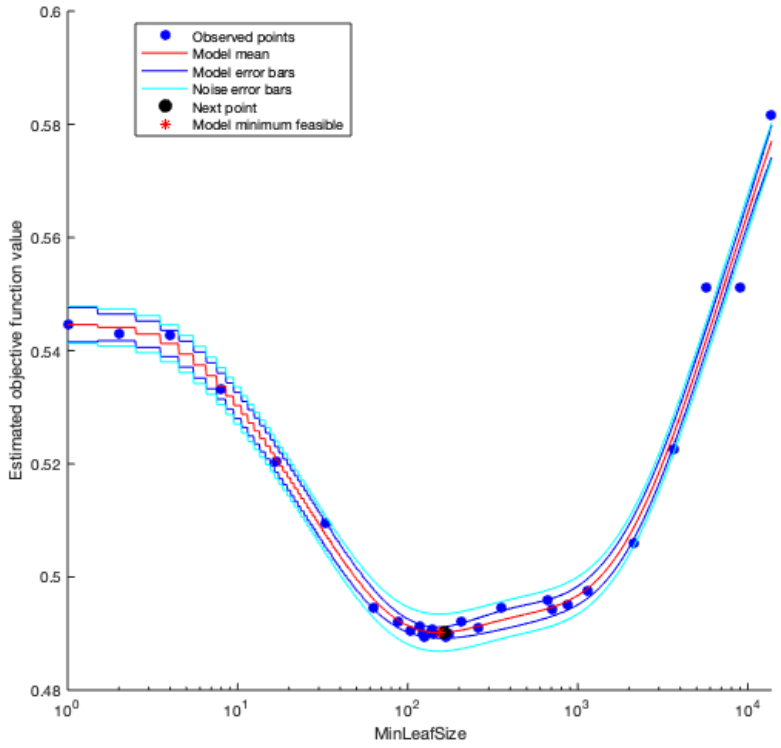
5.1.3.2 Optimalizace hyperparametrů

I v tomto případě je velké množství parametrů, které mohou mít vliv na výkonnost modelu. Optimalizaci hyperparametrů provedeme nejdříve za pomocí vlastního nastavení v Matlabu za pomocí příkazu:

```
mdlTree = fitctree(trainTable, 'Kategorie_podniku', 'OptimizeHyperparameters','auto');
```

Výstupem je tabulka, která je uvedena spolu s dalšími výstupy jako příloha 1 této práce. Dále pak jsou výsledkem 2 grafy. První je zobrazen na následujícím obrázku. Obrázek vyjadřuje, jak se mění výkonnost modelu s ohledem na nastavované parametry. Především se upravuje parametr představujících členění uzlu do dalších skupin (MinLeafSize). Z obrázku je patrné, že nejdříve hodnota klesá a přibližně v úrovni okolo 100 bodů začíná dosahovat svého minima. Od úrovně 1 000 pak naopak začne chybovost poměrně stoupat. Minimum na grafu je vyznačeno červenou hvězdičkou a jedná se o výsledný model, který je pak následně použit pro predikci či další zpracování. Ověření je prováděno i pomocí křížového ověření (cross validace) dat na úrovni 10 kategorií. Jinými slovy – došlo k rozdělení dat do 10 skupin trénovacích a testovacích dat. Výsledná chyba pak byla stanovena s ohledem na chybu v každé ze skupin.

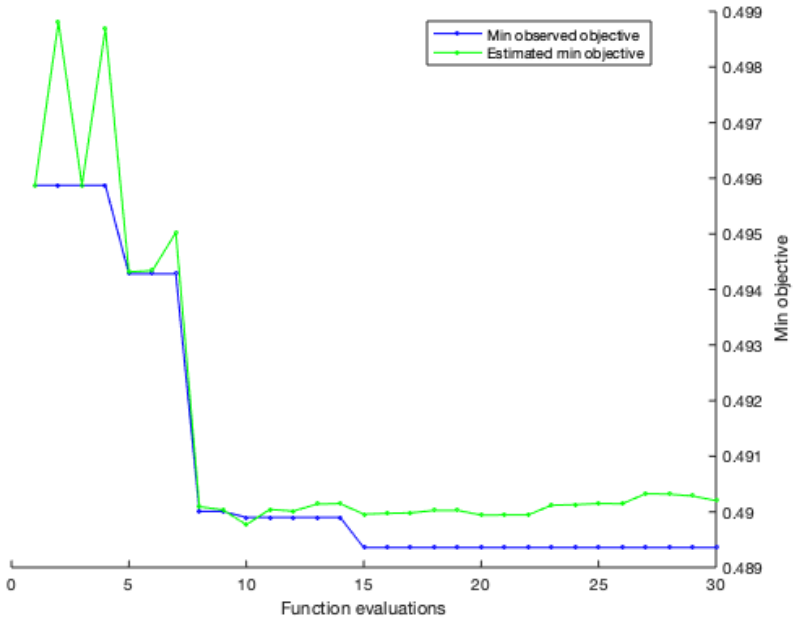
Obrázek 81: Model funkce – optimalizace hyperparametrů stromová klasifikace



Zdroj: Vlastní tvorba.

Druhý graf přestavuje výkonnost modelu, kterou sledujeme s ohledem na počet testovaných proměnných. Graf je vidět na obrázku č. 82. Graf prakticky ukazuje při kolika výpočtech začíná výsledný model dosahovat minimální chybovosti. V našem případě je vidět, že první 4 náhodné testování nepřinášely zlepšení výkonnosti modelu. Následovalo mírné zlepšení, které po dobu dalších 2 testování zůstalo na lokálním minimu. Následovalo opět zlepšení, které vydrželo s mírným skokem téměř 7 dalších výpočtů. Nakonec došlo k finálnímu výpočtu globálního minima. Model s ohledem na náročnost některých výpočtů pracuje i s odhady výkonnosti modelů, což je vidět ze zelené křivky obrázku. U obrázku je důležité si všimnout osy y, která je v rozsahu 0,489 až 0,499. Přesto, že na grafu mohou změny působit díky měřítku jako zásadní, tak ve skutečnosti se jedná jen o kosmetické změny, které mírně zlepšují model, ale nepřestavují zásadní význam.

Obrázek 82: Počet funkcí vs výkonnost modelu – stromová klasifikace



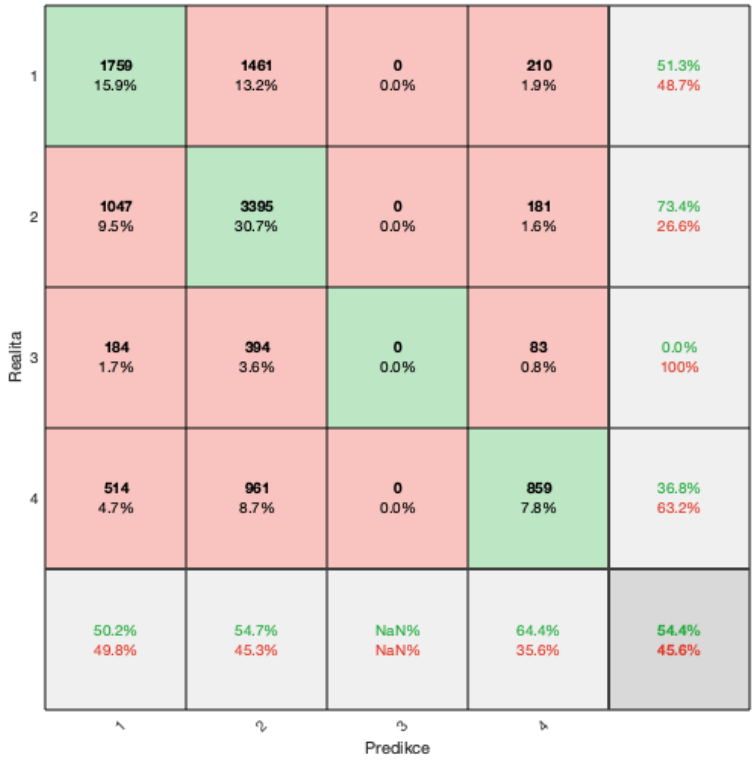
Zdroj: Vlastní tvorba.

U optimalizovaného modelu provedeme predikci za pomocí příkazu:

```
predikceTree = predict(mdlTree, testTable);
```

a dále zobrazíme konfuzní matici, která je vidět na následujícím obrázku. Z obrázku je patrných několik skutečností. V první řadě je model výkonnější než model trénovaný s defaultním nastavením. Model dokonce překročil hodnotu 50% úspěšnosti, což je velmi úspěšné s ohledem na princip úlohy a skutečnost, že klasifikuje 4 kategorie. Velkým nedostatkem je ale skutečnost, že model prakticky vynechává zatřízení 3. kategorie. Toto je pro statistiku úspěšnosti modelu velmi dobré, protože 3. kategorie podniku není v souboru příliš zastoupená. Pro lepší vysvětlení si lze představit úlohu, kde bude binární klasifikace, přičemž trénovací i testovací množina bude obsahovat jen 10 % vzorů druhé kategorie. Pokud by výsledný model tvrdil, že testovaný vzor spadá do 1. kategorie, pak bude mít úspěšnost 90 %, což se může zdát jako velmi dobrý výsledek, ale bez jakéhokoli smyslu z pohledu predikce. V našem případě je výsledek obdobný, jen je klasifikace členěná do 4 oblastí a model ignoruje 3. kategorii podniků. Naopak velmi pozitivní je, že model dokáže dobře zatřídit 1. a 2. kategorii podniků.

Obrázek 83: Konfuzní matice – optimalizace hyperparametrů stromové klasifikace



Zdroj: Vlastní tvorba.

V případě optimalizace hyperparametrů může dojít ke změně nejrůznějších parametrů modelů. Abychom porovnali, k jakým úpravám došlo, zobrazíme si model v Matlabu. Výsledky porovnání jsou vidět v následující tabulce. Z tabulky vyplývá, že se modely liší ve dvou parametrech. Jde o parametry MinParent a MinLeaf.

Tabulka 2: Parametry modelů

Hyperparametr	Defeaultní Model	Optimalizovaný model
SplitCriterion	'gdi'	'gdi'
MinParent	282	10
MinLeaf	141	1
MaxSplits	16 573	16 573
NVarToSample	'all'	'all'
MergeLeaves	'on'	'on'
Prune	'on'	'on'
PruneCriterion	'error'	'error'
QEToler	[]	[]
NSurrogate	0	0
MaxCat	10	10
AlgCat	'auto'	'auto'
PredictorSelection	'allsplits'	'allsplits'

Hyperparametr	Defeaultní Model	Optimalizovaný model
UseChisqTest	1	1
Stream	[]	[]
Reproducible	0	0
Version	2	2
Method	'Tree'	'Tree'
Type	'classification'	'classification'

Zdroj: Vlastní tvorba.

Jelikož optimalizace hyperparametrů nevzala v úvahu úpravu některých parametrů, provedeme toto opět za pomoci vlastního cyklu. V první fázi budeme nastavovat hyperparametry zpracování kategoriálních proměnných. Matlab v nápovědě k této funkci uvádí tyto tyto způsoby:

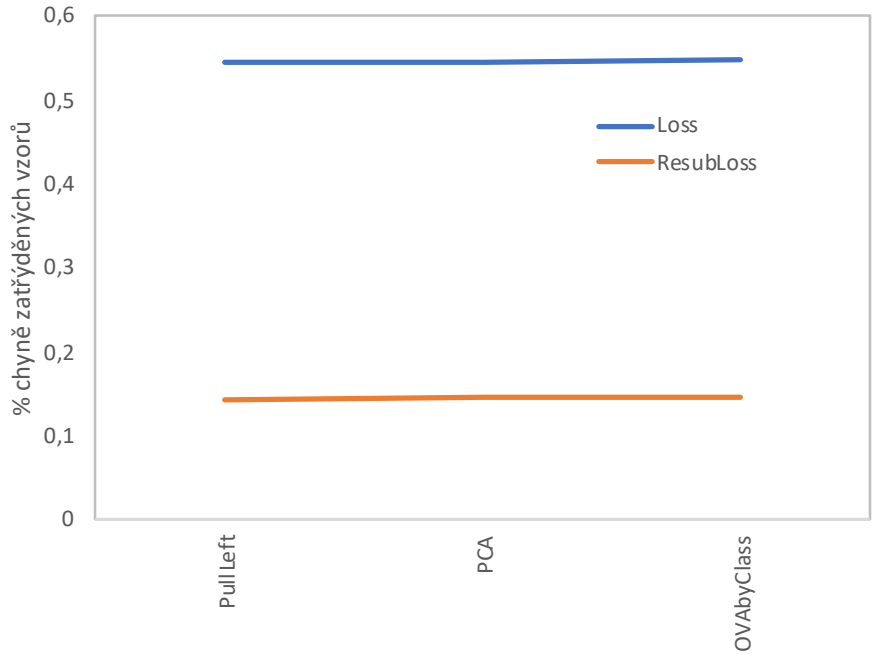
- Exact – vezme v úvahu všechny možnosti dle vzorce $2^{C-1} - 1$
- PullLeft – algoritmus začne se všemi kategoriemi C na pravé větvi a zváží přesunutí každé kategorie do levé větve, aby dosáhl minimální chyby kategorie podniku. Ze sekvence pak vybere rozdělení, které má nejnižší chybu.
- PCA – vypočte skóre pro každou kategorii mezi první hlavní složkou PCA a váženou kovarianční matici a vektorem pravděpodobností třídy pro tuto kategorii. Následně zpracuje výstup v redukované podobě C – 1
- OVAbyClass – začněte se všemi kategoriemi C na pravé větvi. U každé třídy je kategorie na základě jejich pravděpodobnosti pro danou třídu podniku. U první třídy zváží přemístění každé kategorie na levou větev v pořadí, přičemž při každém tahu zaznamenává míru chyby. Toto opakuje dokud není chyba minimální.

Při textech se první parametr ukázal jako natolik složitý, že výstup nebyl ani po několika dnech spočítán, dokud nedošlo k pádu systému. Z těchto důvodů byl parametr z testů vyřazen a dále byly využity ostatní parametry. Ty byly propočítány na základě následujícího cyklu:

```
parametr = ["PullLeft" "PCA" "OVAbyClass"]

for i = 1:3

mdlTree = fitctree(trainTable, 'Kategorie_podniku', 'AlgorithmForCategorical', parametr(i));

lossTreeP(1, i) = loss(mdlTree, testTable);

resubLossTreeP(1, i) = resubLoss(mdlTree);

end
```

První příkaz pouze nastavuje kategoriální proměnnou. Dále následuje cyklus for, který proběhne 3x, což je počet měněných parametrů. V cyklu dojde k natrénování modelu za pomoci funkce fitctree a uložení chyby na trénovací množině a testovací množině. Chyby se ukládají postupně do vektorů, tak aby bylo možné je následně zobrazit. Výsledek zobrazení je vidět na následujícím obrázku. Z něho je patrné, že ani jeden parametr neměl vliv na výsledný výpočet chyby.

Obrázek 84: Optimalizace hyperparametrů – stromové učení, kategorické proměnné



Zdroj: Vlastní tvorba.

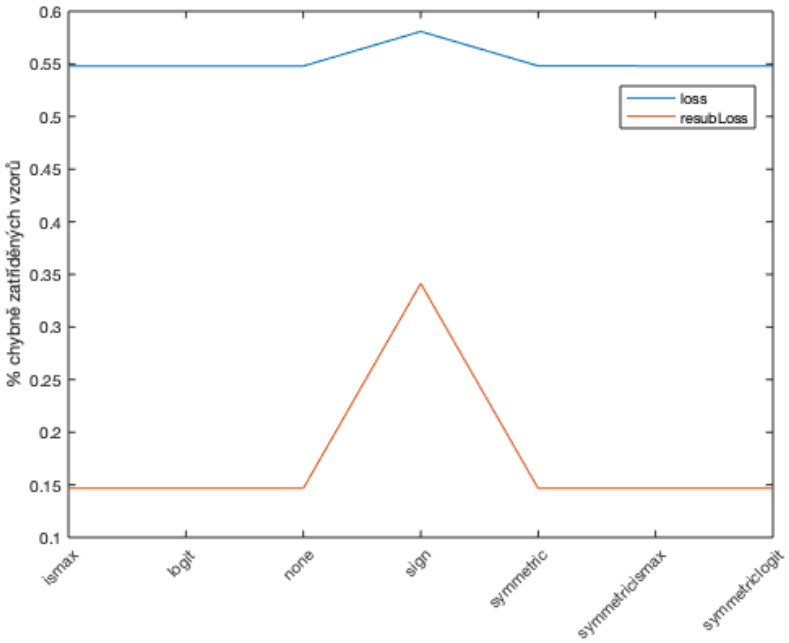
V další fázi došlo k analýze výpočtu vlivu transformace vzdáleností, které jsou shodné jako v případě KNN algoritmu. Opět došlo k trénování a ukládání modelu na základě těchto příkazů:

```
parametr2 = ["doublelogit" "invlogit" "ismax" "logit" "none" "sign" "symmetric" "symmetricismax" "symmetriclogit"]

for i = 1:9

mdlTree = fitctree(trainTable, 'Kategorie_podniku', 'ScoreTransform', parametr2(i));

lossTreeP2(1, i) = loss(mdlTree, testTable);

resubLossTreeP2(1, i) = resubLoss(mdlTree);

end
```

Výsledky chyb byly opět uloženy do vektorů, které je možné vykreslit pro analýzu. Výsledek je vidět na následujícím obrázku. Z něho je opět zřejmé, že pouze jeden parametr měl vliv na

výsledek. Jedná se o parametr sign, který způsobil zhoršení výsledků zejména na trénovací množině dat.

Obrázek 85: Optimalizace hyperparametrů – stromová kategorizace, vzdálenost



Zdroj: Vlastní tvorba.

5.1.3.3 Redukce vlivu rozdělení množin – generování lesa

Jak bylo popsáno výše, stromová klasifikace je velmi závislá na poskytnutých datech. Výsledný strom tak může klasifikovat velmi dobře v rámci trénování, ale může být špatně generalizovatelný. Této skutečnosti odpovídají i výsledky, kdy na trénovacích datech měl model chybovost do 20 %, ale na testovacích datech převyšovala tato chybovost 50 %. Nevýhoda této vlastnosti stromové klasifikace se dá odstranit za pomocí metody, kdy se natrénuje několik stromů. Tyto stromy se následně použijí pro rozhodování o zatřízení modelu. Dojde tak k odstranění lokálního zaměření stromu s ohledem na trénovací data. Natrénování modelu provedeme příkazem

```
mdlETree = fitensemble(trainTable,'Kategorie_podniku','AdaBoostM2',10,'Tree');
```

V rámci příkazu jsme uvedli parametr 10 stromů, které se vytvořily v rámci modelu. Pokud u tohoto modelu počítáme chybovost pomocí příkazů:

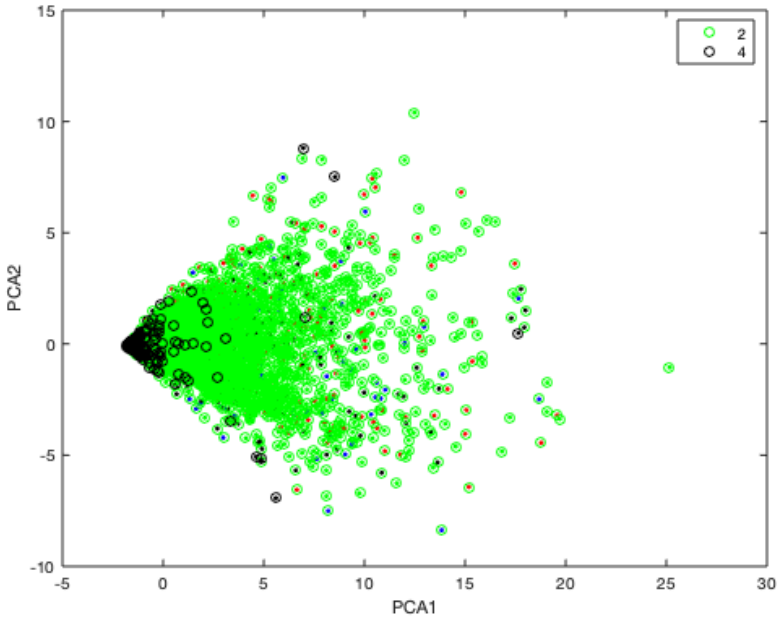
```
lossETree = resubLoss(mdlETree)

lossETree = loss(mdlETree, testTable)
```

Získáme výsledky:

```
resubLossETree = 0.5350

lossETree = 0.5332
```

Z výsledků je patrné, že jak na trénovacích, tak na testovacích datech byla dosažena přibližně podobná chyba. Toto odpovídá přechozímu tvrzení. Míra chyby na testovacích datech však odpovídá předchozím výsledkům. Pro analýzu vykreslíme nejdříve graf kategorií podniků a graf predikce. Kde bude barva souhlasit, je zatřízení provedeno správně. Výsledek je vidět na následujícím obrázku.

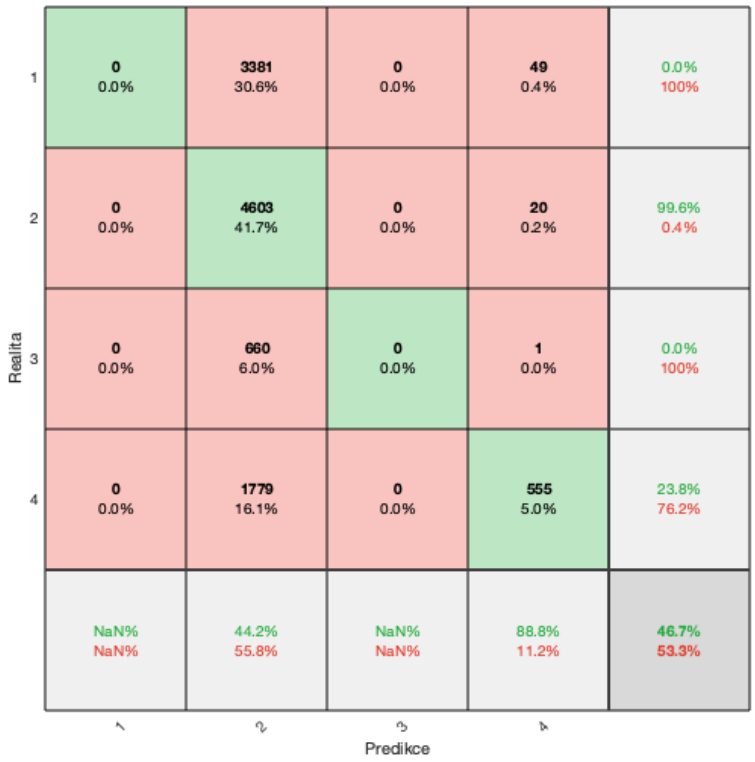
Obrázek 86: Porovnání predikovaného zatřídění a reality u hyperparametru optimalizovaného lesa



Zdroj: Vlastní tvorba.

Z obrázku je zřejmé, že výsledný les zatřiďuje pouze do 2 kategorií podniků. Toto tvrzení ověříme na konfuzní matici, která je vidět na následujícím obrázku. Z obrázku je patrné, že tento model má celkovou úspěšnost zatřízení přes 46 %. Tohoto výsledku však dosahuje tím, že zatřídí vzory pouze do 2 a 4. kategorie. Jinými slovy zcela ignoruje méně četné skupiny podniků. Toto vede k tomu, že v celkovém důsledku je relativně úspěšný, ale prakticky nepoužitelný. Model se prakticky chová, jako by byl přetrénovaný u neuronové sítě.

Obrázek 87: Konfuzní matice - stromová klasifikace při 10 stromech



Zdroj: Vlastní tvorba.

Parametr 10 mohl způsobit špatné výsledky modelu, stejně jako metoda AdaBoostM2. Provedeme proto optimalizaci hyperparametru, kde budeme tyto hodnoty upravovat. Toto provedeme za pomocí následujícího kódu:

```
parametr = ["AdaBoostM2" "LPBoost" "RUSBoost" "TotalBoost"]
for j = 1:4
    for i = 1:30
        mdlETree = fitensemble(trainTable,'Kategorie_podniku',parametr(j) ,i,'Tree');
        lossETree(j, i) = loss(mdlETree, testTable);
        resubLossETree(j, i) = resubLoss(mdlETree);
    end
end
```

Kód je složen ze dvou cyklů. První cyklus dosazuje parametr metody. Těmi jsou:

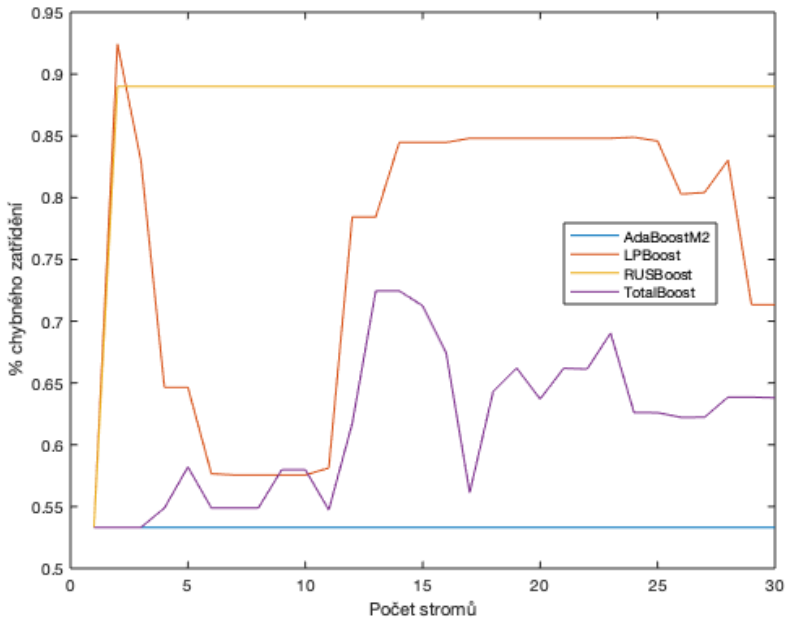
- AdaBoostM2 – Adaptive Boosting for Multiclass Classification (Freund, a Schapire, 1997),

- LPBoost – Linear Programming Boosting (Warmuth, Liao a Ratsch, 2006),
- RUSBoost – Random Undersampling Boosting (Seiffert et al, 2008),
- TotalBoost – Totally Corrective Boosting (Warmuth, Liao a Ratsch, 2006).

Druhý vnořený cyklus mění počet stromů, které budou vytvářeny. Pro každý model se vypočítá procento chybně zatřízených vzorů na trénovací a testovací množině. Výsledek testu chybně zatřízených vzorů se ukládá do matice, která bude mít 4 řádky a 30 sloupců.

Pro přehlednost jsme zobrazili pouze matici s chybně zatřízenými vzory na testovací množině. Ta je vidět na následujícím obrázku. Grafické zobrazení matice s chybně zatřízenými vzory na trénovací množině je uvedeno v příloze, stejně jako samotné matice. Z obrázku níže je patrné, že v případě metody AdaBoostM2 a RUSBoost nemá počet stromů přílišný vliv na kvalitu modelu (v případě RUSBoost s výjimkou prvního kroku cyklu – počet stromů 2). V případě zbylých dvou metod jsou rozdíly poměrně velké a kolísají od 60 do 85 % chybně zatřízených vzorů. Nejlepších výsledků je přitom dosaženo přibližně při počtu stromů <10. Celkově však má nejlepší výsledky metoda AdaBoostM2, kterou jsme použili jako základní metodu. Chybovost této metody je okolo 54 %.

Obrázek 88: Optimalizace hyperparametrů – stromová klasifikace – les



Zdroj: Vlastní tvorba.

5.1.4 Naive Bayes klasifikace

Další metodou, která bude použita pro analýzu Naive Bayes klasifikace. Jak bylo výše uvedeno tato metoda zohledňuje rozložení pravděpodobnosti a pracuje jak s kategorickými, tak i s numerickými prediktory. Základní model vytvoříme na základě následujících příkazů:

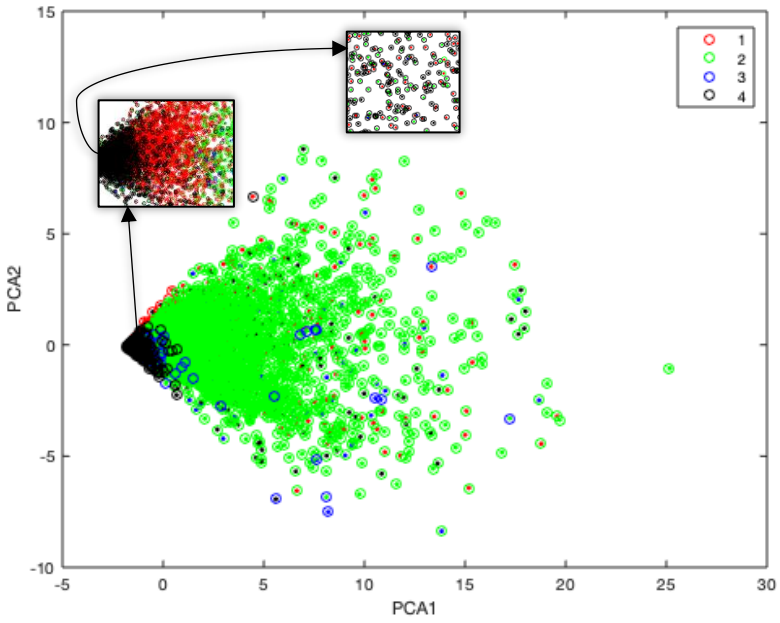
```
dists = [ repmat({'mvmn'},1,5) repmat({'kernel'},1,10) ];

mdlNb = fitcnb(trainTable, 'Kategorie_podniku','Distribution',dists);

resubLossNb = resubLoss(mdlNb)

lossNb = loss(mdlNb, testTable)

predikceNb = predict(mdlNb, testTable)
```

První příkaz uloží rozdělení proměnných na kategoriální (v našem případě prvních 5) a numerické (v našem případě následujících 10 proměnných). Další příkaz vytvoří model, u kterého se opět spočítá procento chybně zatřízených vzorů na trénovací a testovací množině a vytvoří predikce zatřízení na trénovací množině. Chybovost modelu při tomto nastavení vychází následovně:

```
resubLossNb = 0.5447

lossNb = 0.5831
```

Výsledná chyba základního modelu přesahuje 58 %. Díky provedení predikce modelu může stejně jako v předchozích případech vizualizovat zatřídění za pomocí příkazu gscatter. Výsledek je vidět na následujícím obrázku. S ohledem na množství dat, které se vzájemně překrývají a zejména v počátku zhoršují čitelnost, bylo provedeno dvojnásobné zvětšení detailu. Z uvedeného grafu je patrné, že pro zatřízení jsou použity všechny 4 kategorie podniku. Stejně jako v předchozích modelech i zde dominuje 2. kategorie. První a 4. kategorie je zatřízena především na pomyslném středu souboru podniků. Odlehlejší hodnoty, které nespadají do 2. kategorie, jsou téměř vždy klasifikovány špatně, jako podniky z kategorie 2.

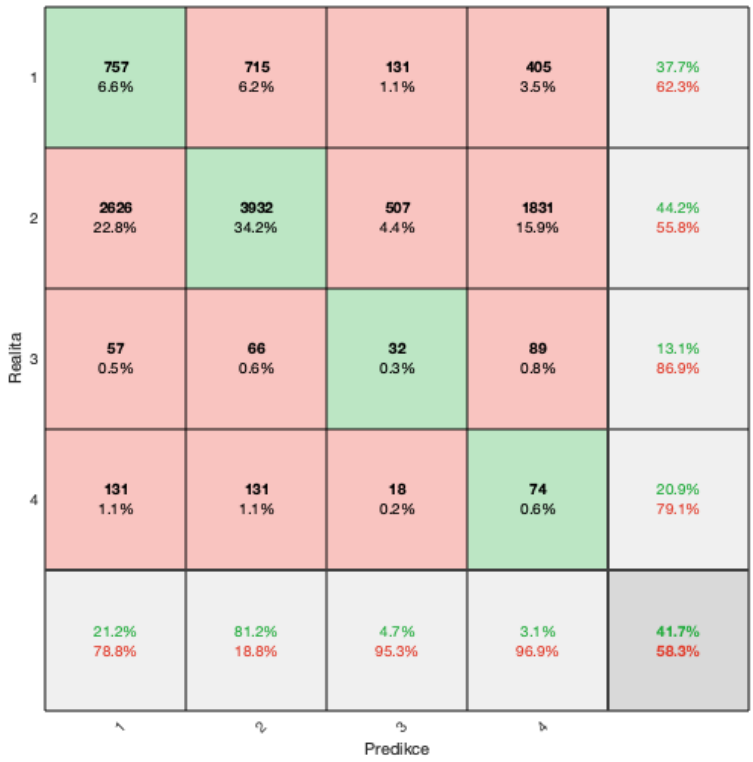
Obrázek 89: Vizualizace zatřízení NB základního modelu



Zdroj: Vlastní tvorba.

Pro podrobnější analýzu zobrazíme konfuzní matici. Tato matice je zobrazena na následujícím obrázku.

Obrázek 90: Konfuzní matice základního modelu NB



Zdroj: Vlastní tvorba.

Z uvedené konfuzní matice vyplývá, že model je schopen predikovat s relativně vyšší mírou úspěšnosti zatřízení do 1. a 2. kategorie. V případě první kategorie je model úspěšný přibližně ze 37, 7 %. V případě 2. kategorie je úspěšnost nad 44 %. Naopak velmi nízkou úspěšnost dosahuje model při zatřízení 3. (13 %) a 4. kategorie (přibližně 21 %).

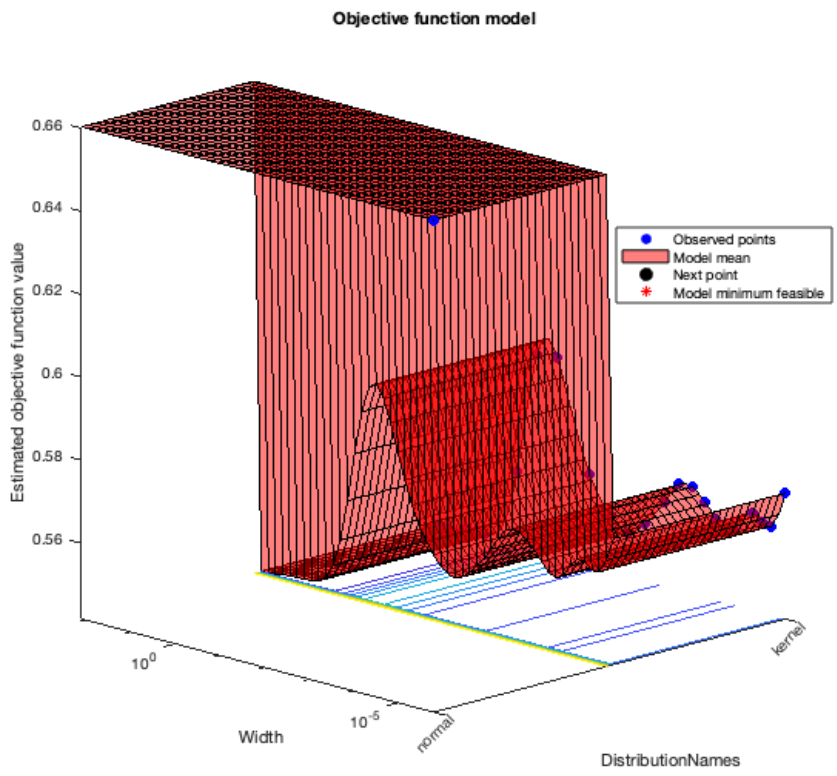
5.1.4.1 Optimalizace Hyperparametrů

Stejně jako předchozí modely i tento umožňují optimalizaci nastavení hyperparametrů modelu. Toto provedeme následujícím příkazem:

```
mdlNb = fitcnb(trainTable, 'Kategorie_podniku','Distribution',dists 'OptimizeHyperparameters','auto');
```

Celý výstup optimalizace modelu je vidět v příloze 1. Na obrázku 91 je zobrazen průběh funkce z pohledu optimalizace a dosažených výsledků. Výsledná úspěšnost modelu je vynesena na ose z. Na ose x je testován model z pohledu parametru Width, která definuje jak „hladká“ bude funkce použitá pro výpočet. Princip je podobný jako v případě klouzavého průměru. Na ose y jsou vyneseny parametry v podobě funkcí, které jsou použity pro výpočet. Jedná se pouze o dvě možnosti (normal a kernel), přičemž varianta normal není pro výpočet evidentně optimální.

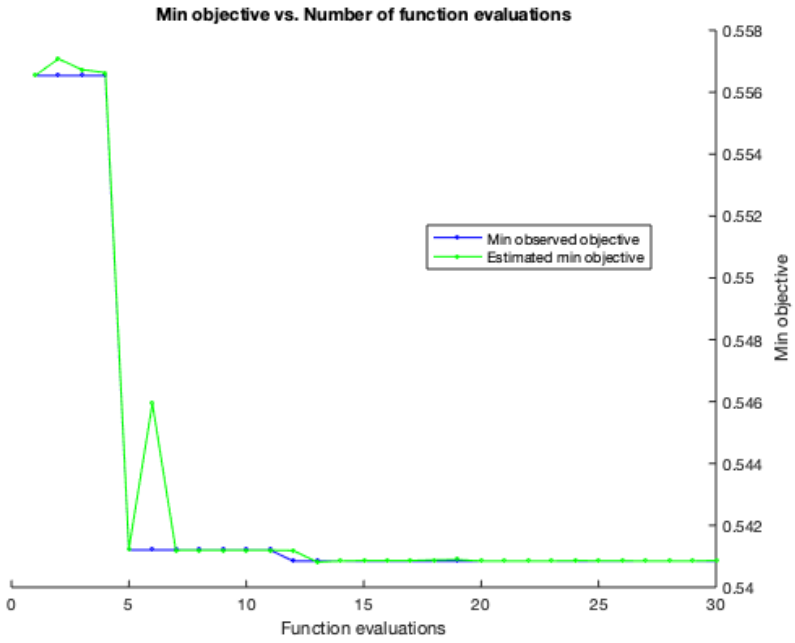
Obrázek 91: Průběh funkce NB při optimalizaci hyperparametrů



Zdroj: Vlastní tvorba.

Při optimalizaci modelu došlo ke skokovému zlepšení při testu 5. funkce. Další zlepšení modelu byla spíše minimální. Rovněž celkové zlepšení modelu není příliš významné a pohybuje se nad jedním procentem. Toto je graficky zobrazeno na následujícím obrázku.

Obrázek 92: Optimalizace hyperparametrů – zlepšení modelu s ohledem na počet testování



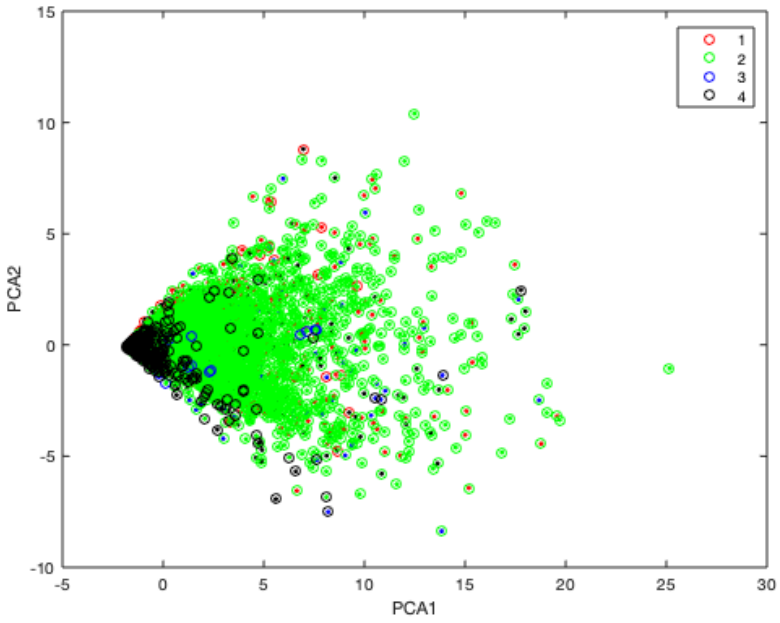
Zdroj: Vlastní tvorba.

Optimalizovaný model dosahuje následujících výsledků při testu na trénovacích a testovacích datech:

```
lossNb = loss(mdlNb, testTable)

lossNb = 0.5488

resubLossNb = resubLoss(mdlNb);

resubLossNb = 0.5324
```

Z pohledu celkového výsledku na testovacích datech došlo ke zlepšení přibližně o 4 %. Pokud provedeme predikci zatřízení na testovacích datech a výsledek vizualizujeme jako v předchozím případě, získáme obrázek 94. Při porovnání s obrázkem 90 je vidět minimální rozdíl. Stále dominuje 2. kategorie, která je ale na rozdíl od předchozího modelu více doplněna dalšími kategoriemi (zejména 4.).

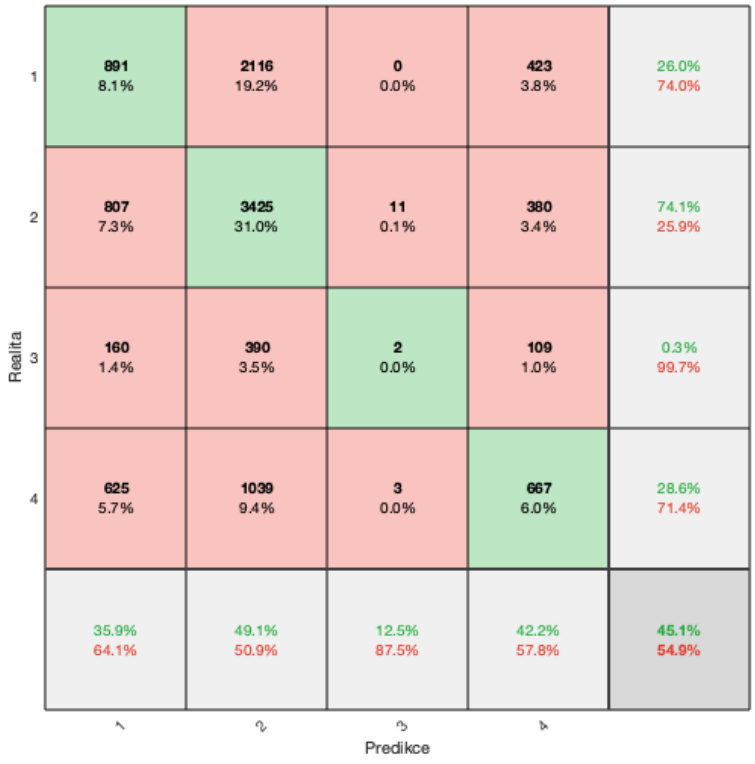
Obrázek 93: Vizualizace zatřízení optimalizovaného modelu



Zdroj: Vlastní tvorba.

Podrobnější analýza zatřízení testovacích dat je vidět na konfuzní matici níže.

Obrázek 94: Konfuzní matice optimalizovaného modelu – NB



Zdroj: Vlastní tvorba.

Z uvedeného výsledku je patrné, že optimalizovaný model zlepšil predikční schopnost pro 2. (více jak 70 %) a 4. kategorii (28,6 %). Naopak došlo ke zhoršení predikční schopnosti v případě 1. kategorie (na 26 %) a v případě 3. kategorie (0,3 %).

Optimalizace hyperparametrů neupravovala metodu formy výpočtu. Toto je možné provést na základě následujících parametrů:

- Box (uniform) $f(x) = 0{,}5I\{|x| \leq 1\}$
- Epanechnikov $f(x) = 0.75(1 - x^2)I\{|x| \leq 1\}$
- Gaussian $f(x) = \left(\frac{1}{\sqrt{2\pi}}\right)^{-0{,}5x^2}$
- Triangular $f(x) = (1 - |x|)I\{|x| \leq 1\}$

Jelikož by některá z metod mohla mít vliv na výslednou kvalitu modelu, provedeme optimalizaci na základě vlastního cyklu za pomocí následujících příkazů:

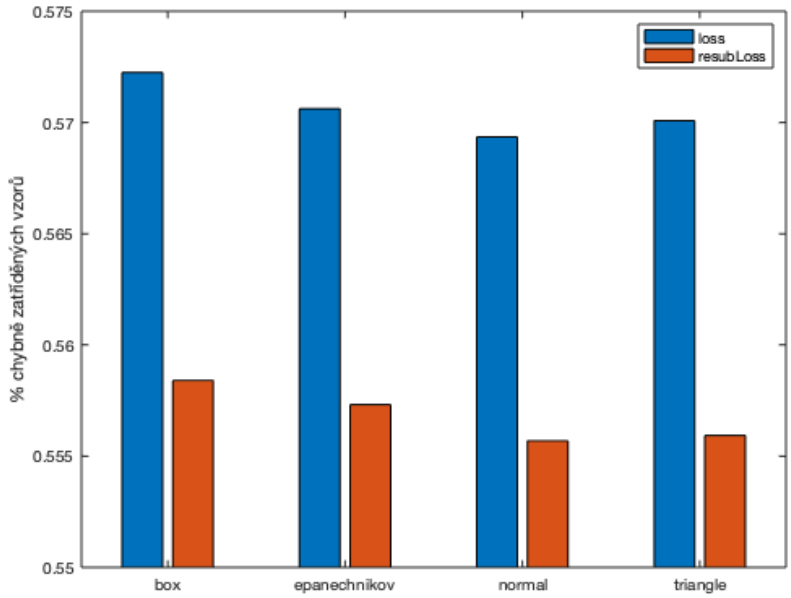
```
parametr = ["box" "epanechnikov" "normal" "triangle"]

for i = 1:4

    mdlNb = fitcnb(trainTable, 'Kategorie_podniku','Distribution',dists, 'Kernel', parametr(i));

    lossNbP(1, i) = loss(mdlNb, testTable);

    resubLossNbP(1, i) = resubLoss(mdlNb);

end
```

Nejdříve uložíme hodnoty možných parametrů do vektoru. Následně provedeme cyklus for, který se bude opakovat dle počtu parametrů. Z kódu je zřejmé, že optimalizujeme základní model. Výsledky pak ukládáme do vektorů proměnných, které reprezentují chybu zatřídění na trénovacích a testovacích datech. Vše následně vizualizujeme za pomocí následujícího kódu:

```
loss = [lossNbP; resubLossNbP]

bar(loss')

xticklabels(parametr);

ylabel("% chybně zatříděných vzorů");

legend("loss", "resubLoss");

ylim([0.55 0.575])
```

V tomto případě proměnné obsahující informaci o chybovosti modelu spojíme do matice a vykreslíme za pomocí sloupcového grafu. Abychom lépe odlišili drobné změny, omezíme dimenze osy y za pomocí příkazu ylim. Výsledek je vidět na následujícím obrázku. Z výsledků je patrné, že nejmenší chybovost vykazuje metoda normal. Rozdíly mezi metodami jsou však téměř neznatelné a pohybují se pod úrovní 0,05 %.

Obrázek 95: Optimalizace hyperparametrů za pomocí vlastní ho cyklu – metoda výpočtu



Zdroj: Vlastní tvorba.

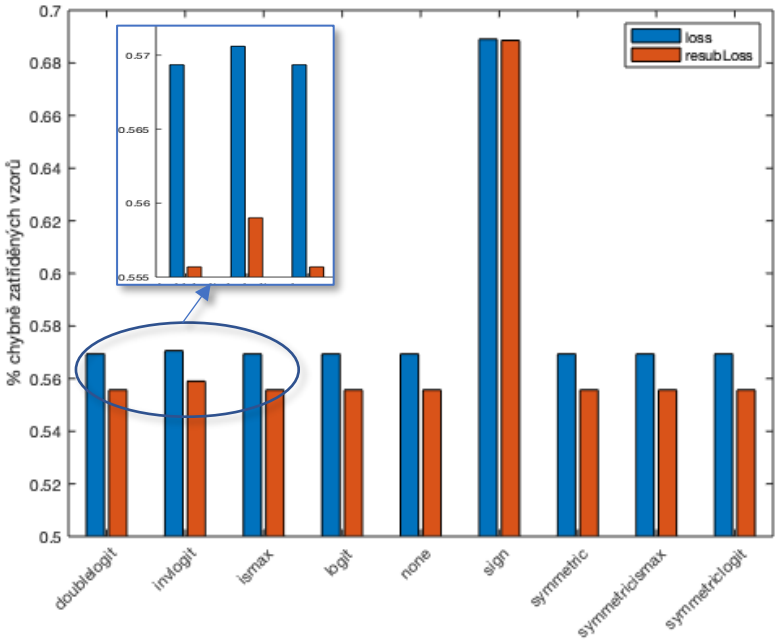
Další parametr, který by mohl mít vliv na výsledný model, je 'ScoreTransform'. Jeho možné varianty jsou shodné jako v předchozích případech. Model optimalizujeme za pomocí následujícího kódu:

```
parametr = ["doublelogit" "invlogit" "ismax" "logit" "none" "sign" "symmetric" "symmetricismax" "symmetriclogit"]

for i = 1:9

    mdlNb = fitcnb(trainTable, 'Kategorie_podniku','Distribution',dists, 'ScoreTransform', parametr(i));

    lossNbP(1, i) = loss(mdlNb, testTable);

    resubLossNbP(1, i) = resubLoss(mdlNb);

end
```

Význam kódu je shodný jako v předchozím modelu, jen s rozdílem samotného trénování modelu, které je specifické s ohledem na optimalizaci hyperparametrů. Výsledek je zobrazen

na následujícím obrázku. Z výsledků je patrné, že mezi metodami není příliš rozdíl. Pouze metoda Sign vykazovala výrazně horší výsledky. S ohledem na nutné měřítko osy y bylo provedeno zvětšení části grafu, kde k mírným rozdílům došlo (první tři metody).

Obrázek 96: Scoretransform – NB – optimalizace vlastním cyklem



Zdroj: Vlastní tvorba.

5.1.5 Discriminant Analysis

Čtvrtou metodou, kterou použijeme pro analýzu dat je discriminant analýza (Klecka a William, 1980). Tato analýza by opět měla zohledňovat rozložení pravděpodobnosti jednotlivých prediktorů v prostoru. Model vytvoříme zadáním příkazu:

```
mdlDa = fitcdiscr(trainTable(:, 6:end) , 'Kategorie_podniku');
```

Z uvedeného příkazu je zřejmé, že vstupem do analýzy jsou numerické prediktory. Jsou tedy vynechány kategoriální prediktory, které jsou v tabulce zastoupeny v prvních 5 sloupcích. Kvalitu modelu nejdříve otestujeme standardními testy:

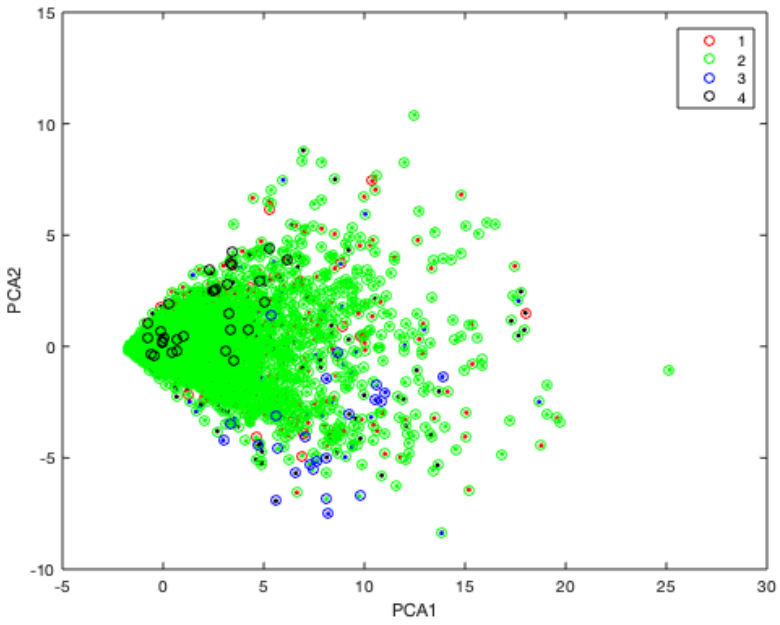
```
resubLossDa = resubLoss(mdlDa)

resubLossDa = 0.5791

lossDa = loss(mdlDa, testTable(:, 6:end));

lossDa = 0.5862
```

Stejně jako v předchozích případech příkaz resubLoss podává informaci o chybně zatřízených podnicích na trénovacích podnicích, tak i loss udává informaci o chybně zatřízených podnicích na testovací sadě dat. Predikci zatřízení testovacích dat provedeme za pomocí příkazu:

```
predikceDa = predict(mdlDa, testTable(:, 6:end));
```

Následně vše zobrazíme jako tomu bylo v předchozím případě za pomocí příkazu gscatter. Výsledek je vidět na následujícím obrázku. Z obrázku je patrné, že zcela převládá 2. kategorie podniku. Rovněž je zřejmé, že model využívá pro klasifikaci všechny 4. kategorie. Na první pohled vypadá relativně dobře zatřízení odlehlých hodnot kategorie 3 a naopak velmi špatně zatřízení odlehlých hodnot kategorie 1 a 4, které jsou klasifikovány jako podniky 2. kategorie.

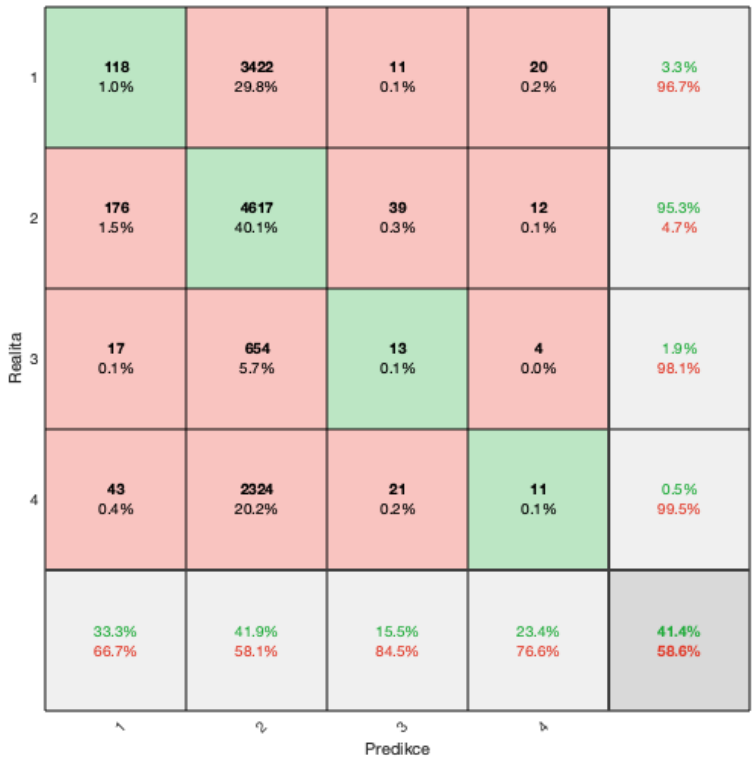
Obrázek 97: Vizualizace discriminant analýzy – základní model



Zdroj: Vlastní tvorba.

Podrobnější analýzu provedeme zobrazením konfuzní matice. Ta je zobrazena na následujícím obrázku. Z konfuzní matice vyplývá, že model prakticky u více jak 90 % zatřiďuje podniky do nejčetnější skupiny, a tedy do 2. kategorie podniku. Tím, že je kategorie nejčetnější pak dosahuje relativně vysoké spolehlivosti, která je mírně nad 40 %. V ostatních případech je míra úspěšnosti pod 3,5 % respektive pod 2 a 0,5 %. Takto je model prakticky nevyužitelný.

Obrázek 98: Konfuzní matice – diskriminant analýza – základní model



Zdroj: Vlastní tvorba.

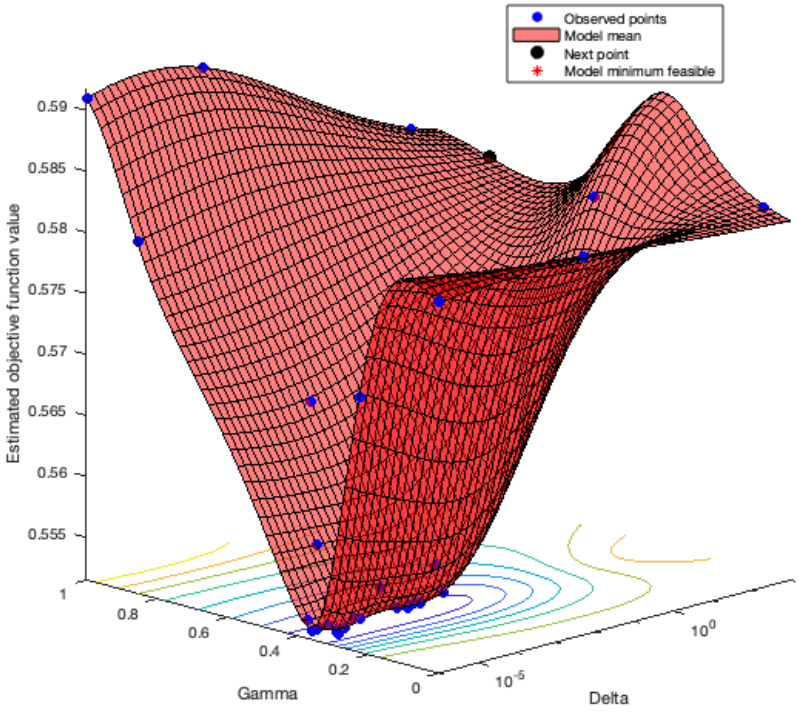
5.1.5.1.1 Optimalizace hyperparametrů

Stejně jako v předchozích případech provedeme optimalizaci hyperparametrů. Pro tento model využijeme následující příkaz:

```
mdlDa = fitcdiscr(trainTable(:, 6:end), 'Kategorie_podniku', 'OptimizeHyperparameters','auto');
```

Kompletní výstup příkazu je uveden v příloze 2. Na obrázku 99 je vidět funkce diskriminační analýzy s ohledem na nastavené parametry a testování funkce. Na ose z je uvedena chyba zatřízení podniků. Osa x představuje gamu a osa y představuje deltu. Pro model se jeví jako klíčové především nastavení parametru gama, jehož optimum je přibližně na úrovni 0,4. Naopak hodnota parametru delta nemá příliš významný vliv na výsledek modelu.

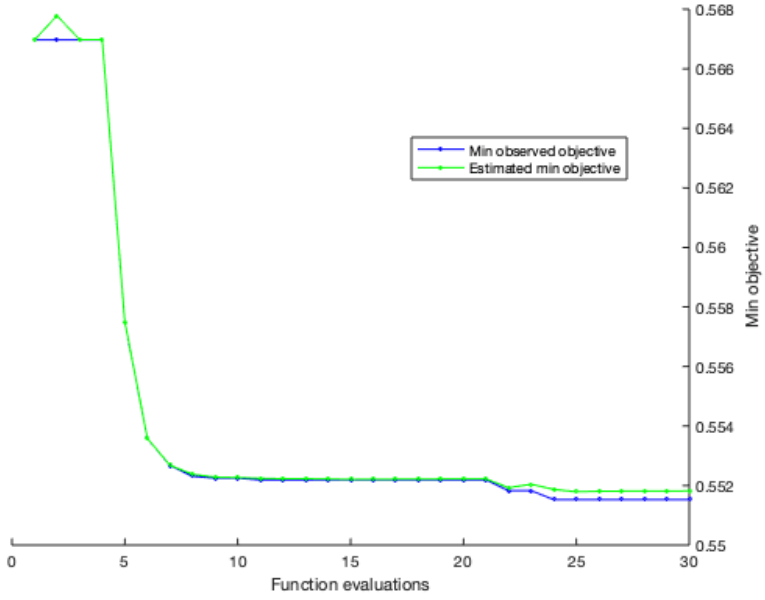
Obrázek 99: Průběh funkce discriminant analýzy s ohledem na optimalizované hyperparametry



Zdroj: Vlastní tvorba.

Druhým výstupem je míra optimalizace s ohledem na počet testovaných funkcí. Výsledek je vidět na následujícím obrázku. Z obrázku je patrné, že model se dostal do optimalizované podoby přibližně po 24 testech. Nejzásadnější změna však byla po prvních 7 testech.

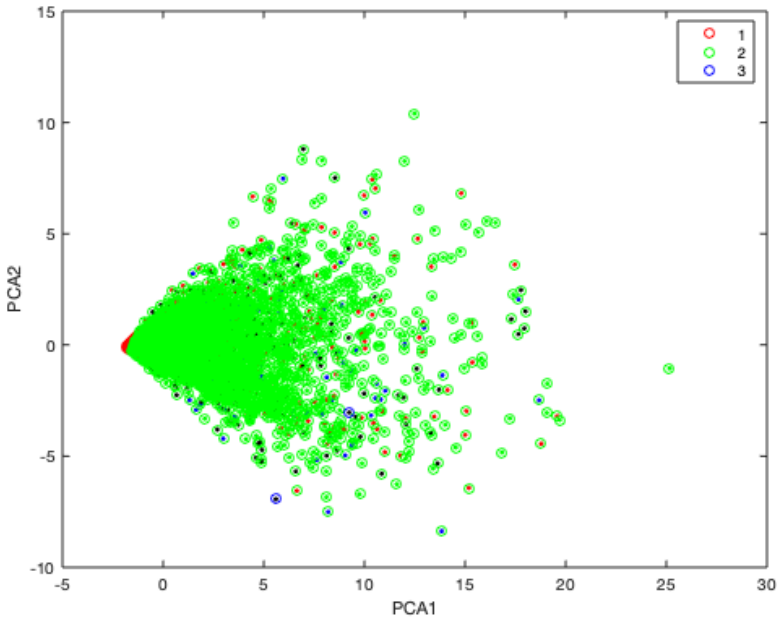
Obrázek 100: Počet testování a optimalizace funkce



Zdroj: Vlastní tvorba.

Stejně jako v předchozím případě provedeme predikci a vizualizaci zatřídění modelu na testovacích datech. Výsledek je zobrazen na následujícím obrázku. Z obrázku je zřejmé, že je zcela vynechána kategorie 4, neboť není obsažena v legendě. Všechny podniky v této kategorii jsou tedy nepochybně zařazeny špatně. Dále je vidět ještě výraznější dominance 2. kategorie podniků na grafu. Kategorie 1 a 3 se v grafu téměř nevyskytují.

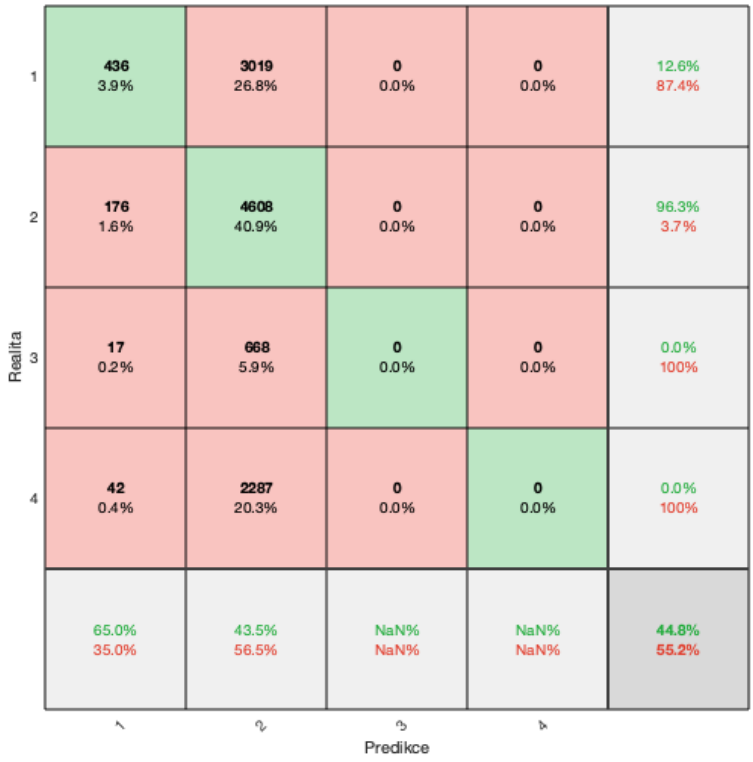
Obrázek 101: Hyperparametry optimalizace – discriminant analýza



Zdroj: Vlastní tvorba.

Při zobrazení konfuzní matice vidíme na obrázku 102, že model nepredikuje žádný podnik do kategorie 4, jak jsme předpokládali, dle výše uvedené analýzy. Dále je z matice zřejmé, že model v případě 3. kategorie zatřiďuje minimální počet proměnných a prakticky má téměř nulovou úspěšnost. Naopak velmi vzrostla úspěšnost pro zatřízení první kategorie. Toto následně vedlo k celkovému zlepšení modelu, jehož úspěšnost se pohybuje nad úrovní 44 %.

Obrázek 102: Konfuzní matice - da po hyperparametrech



Zdroj: Vlastní tvorba.

Optimalizace hyperparametrů nevzala v úvahu nastavování parametru „DiscrimType“ a rovněž neupravovala nastavení parametru „ScoreTransform“. Provedeme proto optimalizaci za pomocí vlastního kódu. ScoreTransform más dhodnémá shodné parametry, jako v předchozích modelech, proto jej zde již nebudeme blíže specifikovat. DiscrimType může nabývat následujících hodnot:

- Llinear – Všechnyvšechny třídy mají stejnou kovarianční matici.
- Quadratic – Kovariančníkovarianční matice se mohou mezi třídami lišit.
- Diaglinear – Všechnyvšechny třídy mají stejnou diagonální kovarianční matici.
- Diagquadratic – Kovariančníkovarianční matice jsou diagonální a mohou se mezi třídami lišit.
- Pseudolinear – Všechnyvšechny třídy mají stejnou kovarianční matici. Software invertuje kovarianční matici pomocí pseudo inverze.
- Pseudoquadratic – Kovariančníkovarianční matice se mohou mezi třídami lišit. Software invertuje kovarianční matici pomocí pseudo inverze.

Optimalizaci provedeme za pomocí následujícího kódu:

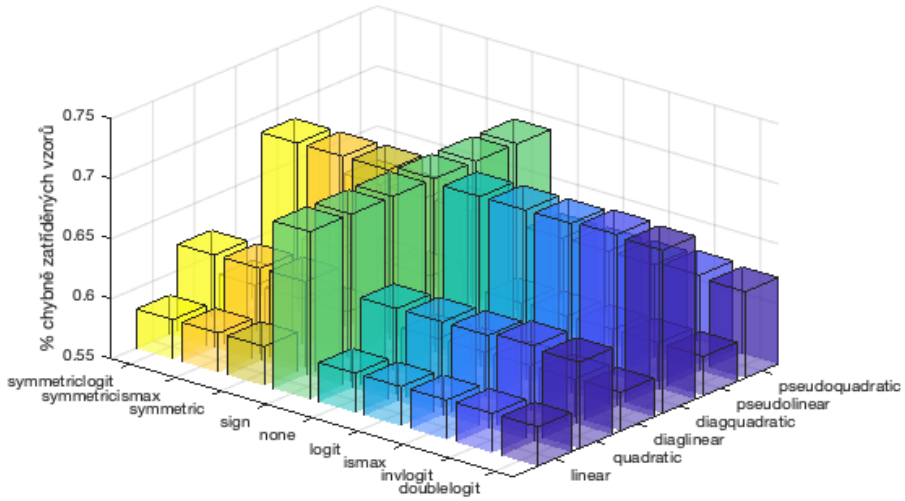
```
parametr2 = ["linear" "quadratic" "diaglinear" "diagquadratic" "pseudolinear" "pseudoquadratic"]

parametr = ["doublelogit" "invlogit" "ismax" "logit" "none" "sign" "symmetric" "symmetricismax" "symmetriclogit"]

for i = 1:9

  for j = 1:6

    mdlDa = fitcdiscr(trainTable(:, 6:end), 'Kategorie_podniku', 'ScoreTransform', parametr(i), 'DiscrimType', parametr2(j));

    lossDaP(j, i) = loss(mdlDa, testTable(:, 6:end));

    resubLossDaP(j, i) = resubLoss(mdlDa);

  end

end
```

Kód nejdříve do proměnné parametr a parametr2 uloží možné hodnoty jednotlivých hyperparametrů jako vektoru. Následně provede cyklus, jehož počet opakování odpovídá délce vektoru parametr, což je v našem případě 9. Při každém jednotlivém cyklu se provede další cyklus, který se bude opakovat 6x, což odpovídá délce parametru 2. Tímto dojde ke vzájemné kombinaci všech možných nastavení. Výsledky kvality natrénovaného modelu se ukládají do dvou matic, z nichž lossDaP(j, i) reprezentuje chyby predikce zařazení podniků na testovacích datech a resubLossDaP(j, i) reprezentuje chyby zařazení podniků na trénovacích datech. Kompletní výsledky jsou v příloze 3. Níže uvádíme grafický výstup matice lossDaP(j, i). Grafické zobrazení provedeme pomocí následujících příkazů:

```
b = bar3(resubLossDaP)

set(b,'FaceAlpha',0.50)

xticklabels(parametr);

yticklabels(parametr2);

zlabel("% chybně zatříděných vzorů");

zlim([0.55 0.75])
```

První příkaz nejdříve vykreslí trojrozměrný sloupcový graf z matice, kde dimenze x odpovídá indexu j a dimenze y odpovídá indexu i. Hodnota v matici pak odpovídá dimenzi z. Druhý příkaz nastavuje průhlednost pro lepší čitelnost 3D grafu. Následuje popis jednotlivých os a omezení rozsahu dimenze z pro lepší čitelnost rozdílů.

Obrázek 103: Optimalizace hyperparametrů DA – vlastní kód



Zdroj: Vlastní tvorba.

Z obrázku vyplývá, že nejhorších výsledků dosahuje nastavení *sign* v případě parametru *ScoreTransform*. Bez ohledu na další parametry byl výsledek na úrovni 69 % špatně zatřízených dat. Podobně tomu bylo u nastavení *diagquadratic* v rámci parametru *DiscrimType.* I zde docházelo ke stejně vysoké chybě bez ohledu na další nastavení parametrů. Nejlepší výsledky pak dosáhly parametry *linear* a *pseudolinear* v rámci parametru *DiscrimType*. V obou případech byly výsledky ve výši 58 % špatně zatřízených dat. Výsledky se neměnily při jiných hodnotách *ScoreTransform,* pokud nepočítáme nastavení *sign* viz výše.

5.1.6 Multiclass Support Vector Machines

Poslední metodou, kterou použijeme z metod strojového učení, před použitím neuronových sítí je Muticlass support vector machines. Klasickou metodu Support vector machines bohužel nemůžeme použít, neboť klasifikujeme podniky do více kategorií. Model vytvoříme za pomocí následujícího příkazu:

```
mdlSvm = fitcecoc(trainTable, 'Kategorie_podniku');
```

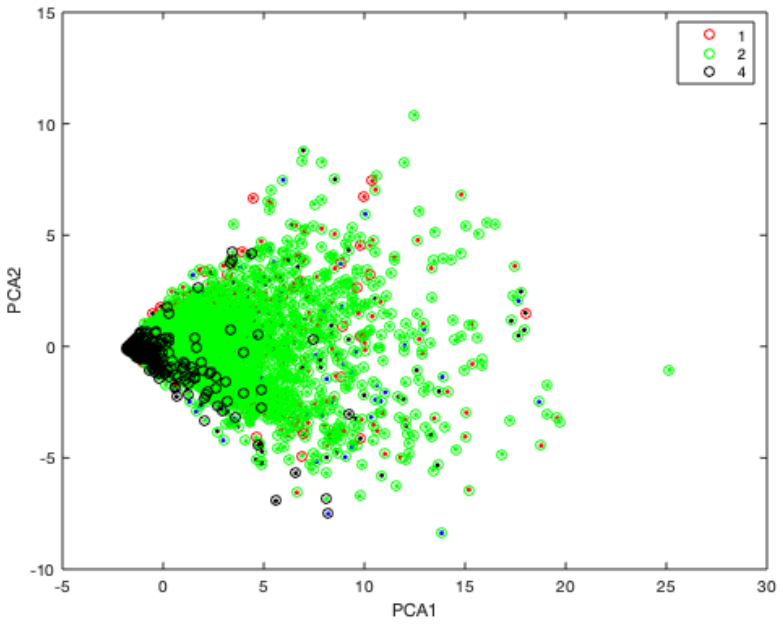
Základní model má následující výsledky zatřízení na trénovacích a testovacích datech:

```
resubLossSvm = resubLoss(mdlSvm);
```

```
resubLossSvm = 0.5565

lossSvm = loss(mdlSvm, testTable);

lossSvm = 0.5677
```

Výsledný model má tedy přibližně 55 % chybu při zatřízení kategorie podniku na trénovacích datech a o jedno procento horší chybu na testovacích datech. Vizualizaci provedeme opět funkcí gscatter jako v předchozích případech na testovacích datech. Výsledek je zobrazen na následujícím obrázku. Z obrázku je patrné, že dominantně je zastoupená 2. kategorie podniku. Velmi četně je rovněž zastoupená 4. kategorie podniku. Naopak zde zcela chybí 3. kategorie.

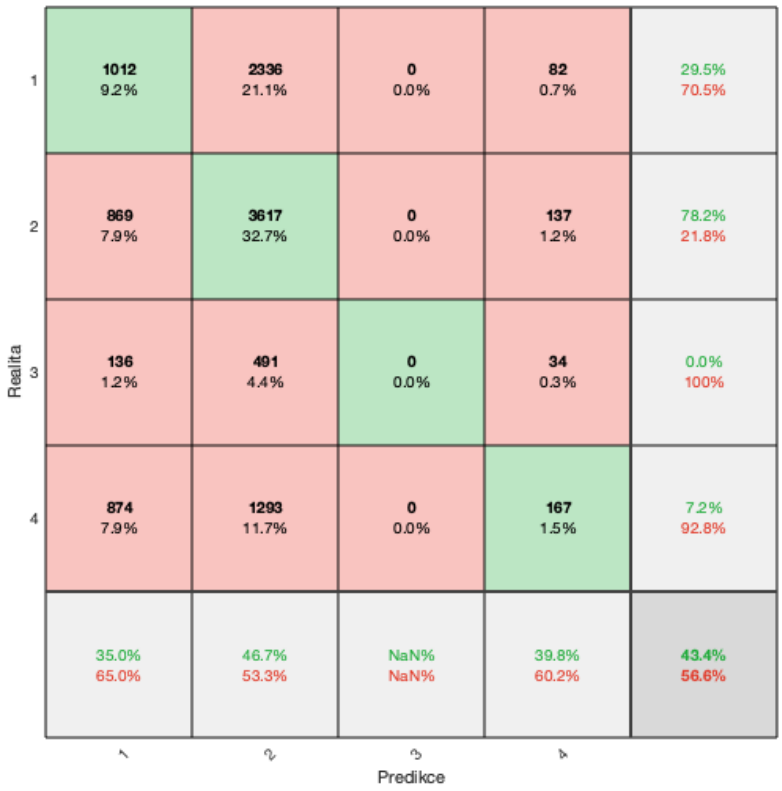
Obrázek 104: Vizualizace základního modelu SVM



Zdroj: Vlastní tvorba.

Podrobnější analýzu budeme moct provést pomocí konfuzní matice, kterou zobrazíme jako v předchozím případě pomocí funkce plotconfusion. Výsledek je vidět na následujícím obrázku. Na obrázku je vidět, že se potvrdil předpoklad o chybějící 3. kategorii. Model nepredikuje žádný podnik do této kategorie. Z podniků, které jsou ve skutečnosti ve 2. kategorii správně zatřídí do 2. kategorie více jak 78 %. V případě 1. kategorie je úspěšnost nad 29 % a pro 4. kategorii se pohybuje pod 8 %. V případě poslední, tedy 3. kategorie je úspěšnost 0, jak bylo popsáno výše.

Obrázek 105: SVM základní model - konfuzní matice



Zdroj: Vlastní tvorba.

5.1.6.1 Optimalizace hyperparametrů

I u tohoto modelu je možné použít optimalizaci hyperparametrů. Toto lze provést za pomocí příkazu:

```
mdlSvm = fitcecoc(trainTable, 'Kategorie_podniku', 'OptimizeHyperparameters','auto');
```

Model však bohužel na několika různých počítačích a verzích matlabu i po několika dnech nepřinesl žádné výsledky. Z těchto důvodů provedeme optimalizaci hyperparametrů pomocí vlastního kódu. Parametr, který můžeme upravovat, se nazývá *Coding* a může nabývat následujících hodnot.

- Allpairs
- Binarycomplete
- Denserandom
- Onevsall
- Ordinal

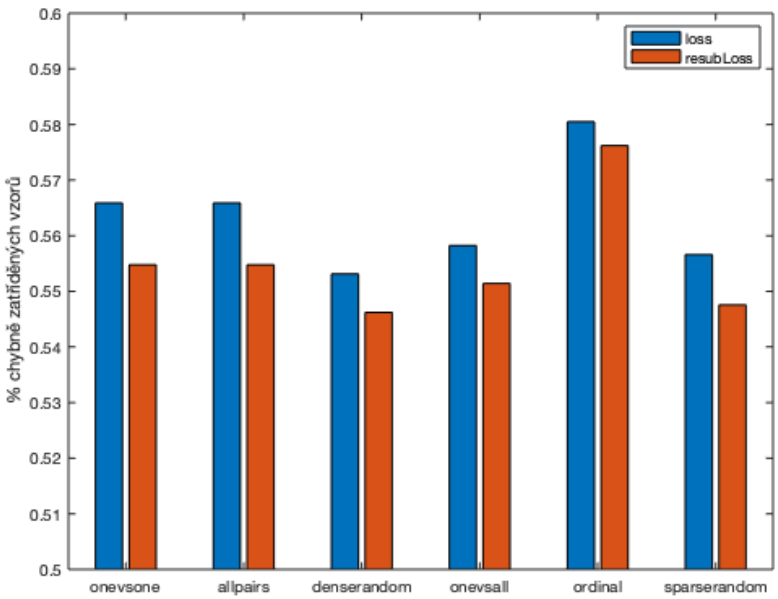
- Sparserandom
- Ternarycomplete

Samotnou optimalizaci provedeme za pomoci následujícího kódu.

```
parametr = ["onevsone" "allpairs" "denserandom" "onevsall" "ordinal" "sparserandom"]
for j = 1:6
   mdlSvm = fitcecoc(trainTable, 'Kategorie_podniku', 'Coding', parametr2(j));
   resubLossSvm(j) = resubLoss(mdlSvm);
   lossSvm(j) = loss(mdlSvm, testTable);
end
```

Kód nejdříve uloží možné varianty do proměnné parametr. Následně provede cyklus for v počtu opakování rovnající se počtu možných variant parametru. V každém cyklu se natrénuje model, u něhož se ověří chybovost na trénovacích a testovacích datech. Vše se uloží pro budoucí analýzu. Výsledek uložených hodnot jsme zobrazili jako v předchozím případě, což je vidět na následujícím obrázku. Z grafu je patrné, že model dosahuje mírně odlišných výsledků na základě nastavených parametrů. Nejlépe jsou na tom Denseradom a Sparserandom. Naopak nejhůře dopadl parametr ordinal.

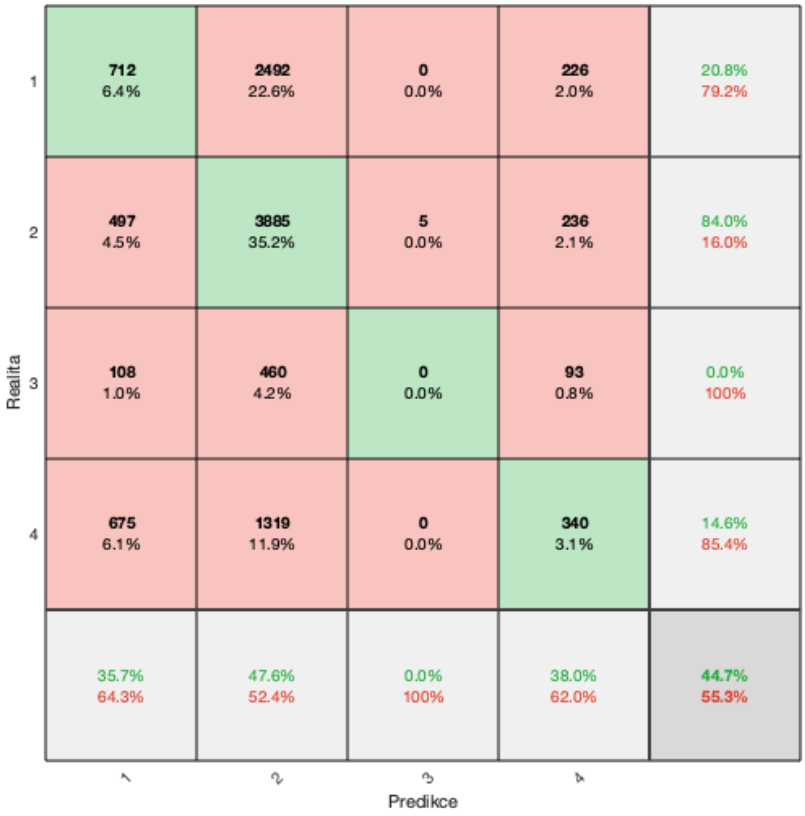
Obrázek 106: Výsledky optimalizace hyperparametru – SVM



Zdroj: Vlastní tvorba.

Abychom mohli analyzovat výsledky zobrazíme i konfusní matici nejlepšího výsledku. Ta je zobrazena na následujícím obrázku. Na rozdíl od předchozí situace model predikuje všechny 4 kategorie podniku. Bohužel 3. kategorii zcela chybně, a proto je výsledná úspěšnost pro zatřízení podniků v této kategorii 0. Model však zlepšil svojí predikční schopnost oproti základnímu modelu v druhé kategorii, kde dosáhl dokonce více jak 84% úspěšnosti. Rovněž úspěšnost při zatřízení 4. kategorie výrazně stoupla na 14,6 %. Naopak klesla úspěšnost zatřízení 1. modelu, kde se pohybujeme mírně nad 20 %.

Obrázek 107: Konfuzní matice po optimalizaci hyperparametru - SVM



Zdroj: Vlastní tvorba.

5.1.7 Samoorganizující se mapy (Kohonenovy sítě)

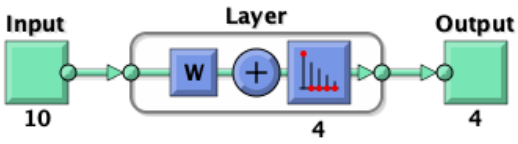
Samoorganizující se mapy (Kohonenovy sítě) jsou neuronové sítě, které dokáží vizualizovat n dimenzionální úlohu do 2 nebo 1-dimenzionálního stavu. Prakticky se může jednat o alternativu k PCA. Velmi důležité je, že tato síť hledá podobné charakteristiky a vytváří určité shluky (clustery) s určitou podobností. Díky tomu můžeme analyzovat, jaké je zastoupení podniků v jednotlivých clusterech s ohledem na náš cíl. Lze však předpokládat, že zatřízení nebude pro náš cíl ideální při malém počtu clusterů, neboť se jedná o učení bez učitele a síť tak neví, jaké

výsledky nás zajímají. Nejdříve se pokusíme o rozdělení souboru do čtyř skupin. Toto provedeme pomocí následujících příkazů

```
net = selforgmap([2,2]);

X = trainTable(:, 6:end-1)‘;

net = train(net,X);
```

Kód nejdříve vytvoří síť, která bude obsahovat 4 clustery. Dále jsme trénovací data uložili do proměnné X. Jedná se o numerická data, což je patrné ze zápisu „6:end-1“. Data byla transponována (řádky převedeny na sloupce a naopak), neboť toto je předpokládaný vstup do neuronové sítě. Nakonec trénujeme model na uvedených datech. Neuronová síť má strukturu, která je zobrazena na následujícím obrázku. Z něho je zřejmé, že vstupem je 10 prediktorů, který každý vstupuje do samostatného neuronu a výstupem jsou 4 neurony.

Obrázek 108: Samoorganizující se mapa 2x2



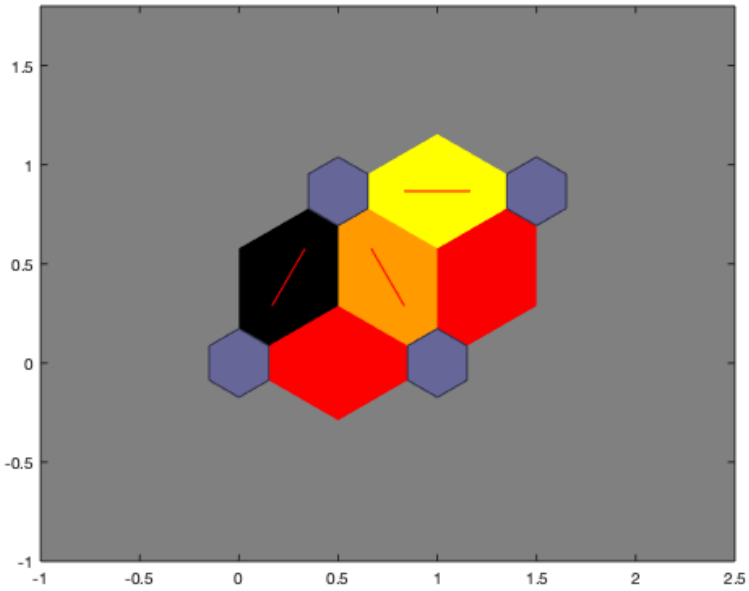
Zdroj: Vlastní tvorba.

Pomocí příkazu

```
plotsomnd(net)
```

zobrazíme vzdálenost mezi jednotlivými clustery. Výsledek je vidět na následujícím obrázku. Žlutá barva představuje nejmenší vzdálenost. Dále následuje oranžová barva, přes červenou až po černou. Z výsledku tedy vyplývá, že cluster nevýše vlevo je velmi vzdálený od clusteru nejníže vlevo. Naopak vzdálenost mezi nejvyššími clustery je relativně malá. Vzdálenost mezi clustery zcela vpravo je menší než mezi nejvzdálenějšími clutery, ale zároveň větší než v případě nejbližších clusterů. Dimenze x a y nemají pro interpretaci žádný význam, proto v obrázku nejsou popsány.

Obrázek 109: Vzdálenosti mezi jednotlivými clustery



Zdroj: Vlastní tvorba.

Další příkaz

```
plotsomhits(net,X);
```

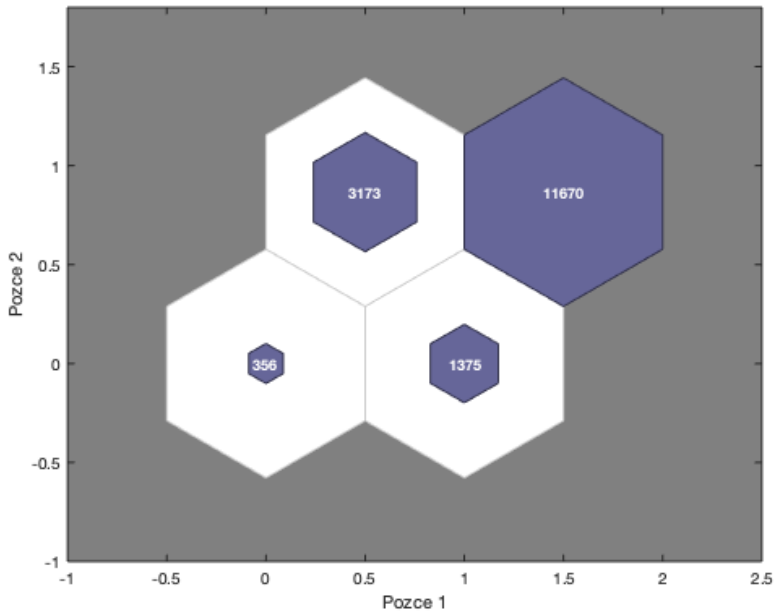
nám zobrazí počet podniků, které byly do každého clusteru zařazeny. Výsledek je zobrazen na následujícím obrázku. Z tohoto obrázku vyplývá, že neuronová síť v drtivé většině zatřídila podniky do clusteru, který je vpravo nejvýše. Z rozdělení počtu dat je rovněž zřejmé, že nemůže odpovídat potřebám kategorizace, které úloha sleduje. Toto je zřejmé, neboť počet podniků v nečetnější kategorii je výrazně nižší. Pro ověření použijeme příkaz:

```
>> countcats(trainTable.Kategorie_podniku)
```

Výsledek je:

- 5358 podniků pro kategorii 1,
- 7267 podniků pro kategorii 2,
- 1034 podniků pro kategorii 3,
- 3596 podniků pro kategorii 4.

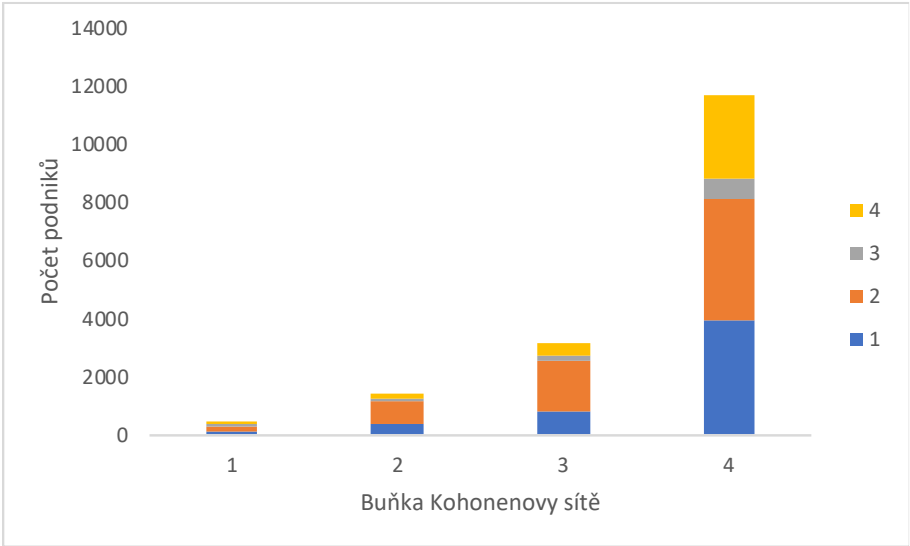
Obrázek 110: Počet podniků v jednotlivých clusterech



Zdroj: Vlastní tvorba.

Jelikož je zřejmé, že v nevýše položeném clusteru vpravo je počet podniků, které se musí skládat z několika kategorií, provedeme analýzu, zdali i v dalších clusterech jsou podniky složené z těchto skupin, neboť v dalších případech to nutně nemusí platit. Výsledek je vidět na následujícím obrázku. Obrázek odpovídá co do výše počtu podniků pro jednotlivé clustery. Číslování je přitom zleva odzdola směrem doprava a následně v novém řádku opět stejným způsobem.

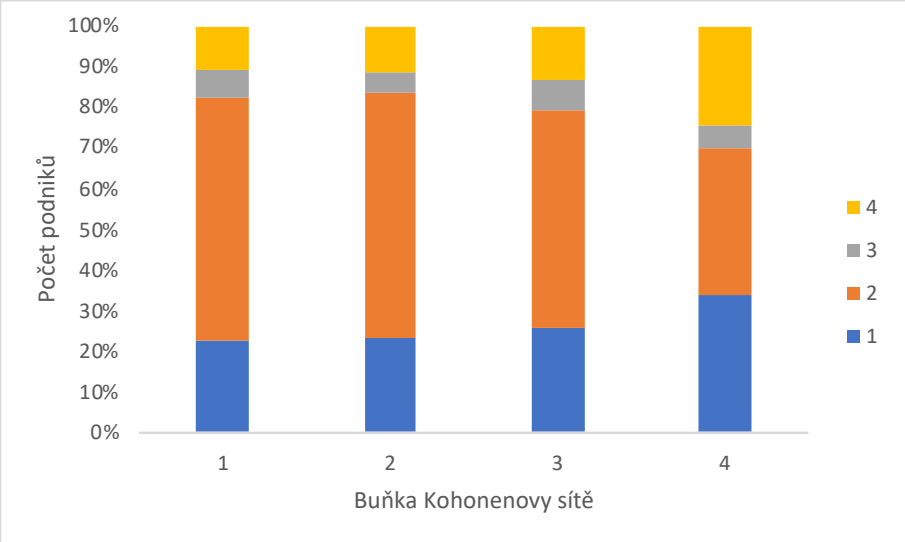
Obrázek 111: Počet podniků v jednotlivých clusterech v síti 2x2



Zdroj: Vlastní tvorba.

Z výše uvedeného grafu nevyplývá, zda nějaký cluster Kohonenovy sítě nezatřiďuje přirozeně správně některou skupinu podniků. Abychom mohli analyzovat tuto skutečnost, provedeme normalizaci dat, která je zobrazena na následujícím obrázku. Z obrázku je patrné, že sice v posledním clusteru stoupá procentuální podíl kategorie 1 a 4, ale celkově se nejedná o významné rozdíly, které by mohli pomoci lepší predikci dat.

Obrázek 112: Procentuální počet podniků v jednotlivých clusterech Kohonenovy sítě 2x2



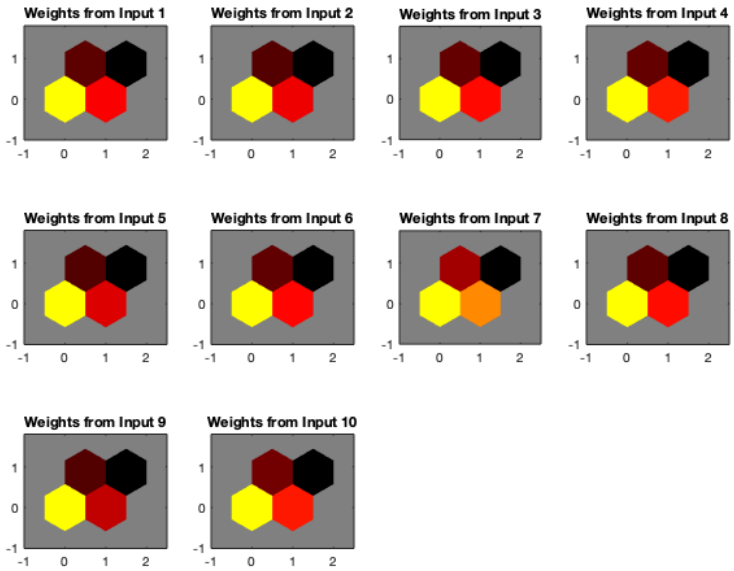
Zdroj: Vlastní tvorba.

Výslednou síť budeme analyzovat i z pohledu jednotlivých prediktorů. Toto provedeme za pomocí příkazu

```
plotsomplanes(net)
```

Graf zobrazuje pro každý prediktor, jaký má s daným clusterem propojení. Pro negativní propojení je barva černá, pro pozitivní propojení je barva žlutá a nulová váha je červená. Ideální stav představuje, pokud jednotlivé prediktory mění svoje váhy propojující je s jednotlivými neurony představujícími clustery. Toto bohužel na obrázku níže není vidět. Prakticky se jednotlivé prediktory propojují podobnými váhami a rozdělení podniků do jednotlivých kategorií je tak velmi slabé z pohledu vytyčení přesných hranic, které jsou odlišeny jasnou korelací.

Obrázek 113: Váhy propojení pro jednotlivé prediktory a clustery

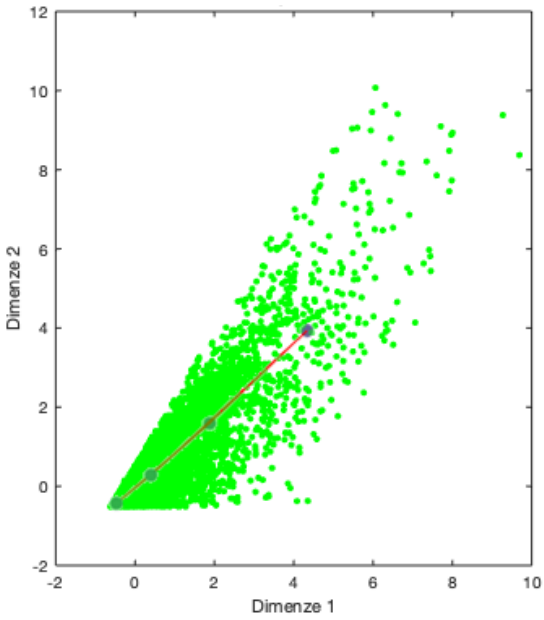


Zdroj: Vlastní tvorba.

Poslední analýza pro danou neuronovou síť bude zobrazení rozložení jednotlivých neuronů v rámci podniků. Toto provedeme následujícími příkazy

```
plotsompos(net)

plotsompos(net, X)
```

Obrázek 114: Rozložení neuronů v datech



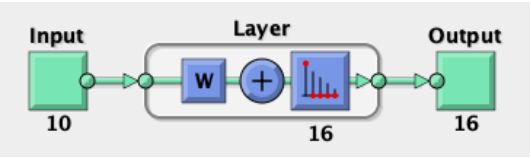
Zdroj: Vlastní tvorba.

Bohužel grafický výstup nepodporuje přílišnou optimalizaci zobrazení, a proto vykreslíme dva grafy, které následně prolneme. Výsledek je vidět na obrázku 115. Výsledná kategorizace spíše odpovídá kategorizaci na straně 68. a bohužel je pro náš účel nevyužitelná. V další fázi proto použijeme síť, která bude obsahovat 16 clusterů. Síť natrénujeme stejným způsobem jako v předchozím případě a zobrazíme následujícím příkazem:

```
view(net)
```

Výsledek je vidět na následujícím obrázku. Z obrázku je patrné, že počet prediktorů zůstal stejný. Změnil se však počet výstupních neuronů, který odpovídá matici 4x4.

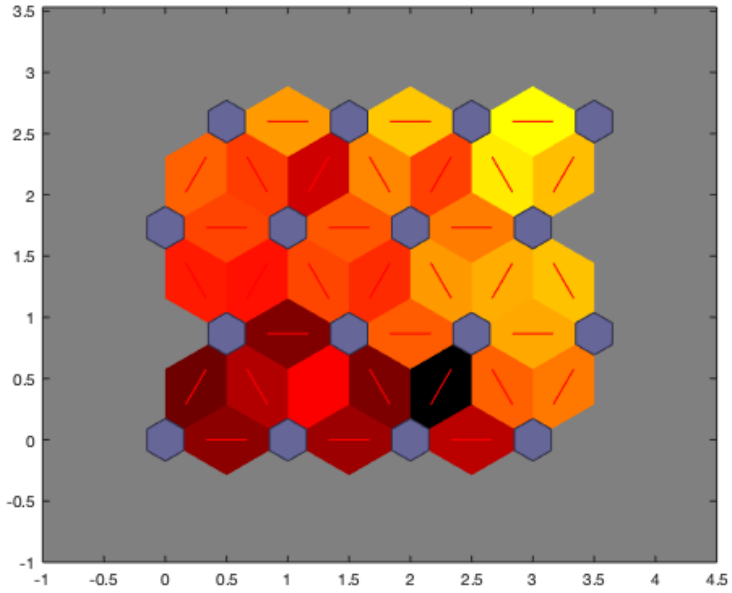
Obrázek 115: Schéma sítě 4x4



Zdroj: Vlastní tvorba.

Nejdříve provedeme analýzu z pohledu vzdáleností jednotlivých clusterů (neuronů) mezi sebou. Toto provedeme stejně jako v předchozím případě za pomoci příkazu *plotsomnd*. Výsledek je vidět na následujícím obrázku.

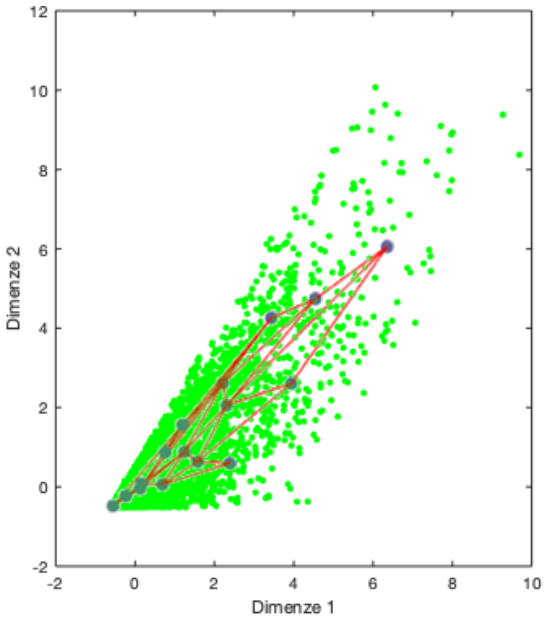
Obrázek 116: Vzdálenosti mezi jednotlivými neurony – síť 4x4



Zdroj: Vlastní tvorba.

Z obrázku je patrné, že zejména neurony vlevo dole mají větší vzdálenosti mezi nejbližšími sousedy, než je tomu v případě neuronů vpravo nahoře. Výsledné rozložení naznačuje, že dolní dolní neurony budou spíše reprezentovat odlehlejší data. Celkové rozložení neuronů lze na datech opět zobrazit stejně jako v předchozí síti za pomocí příkazu plotsompos. Výsledek je vidět na následujícím obrázku. Rozložení neuronu odpovídá předchozímu schématickému obrázku, neboť jsou zde na počátku grafu neurony, které mají velmi blízko ke svému sousedovi, ale jsou zde i neurony, které jsou od sousedů velmi vzdáleny (v právo nahoře).

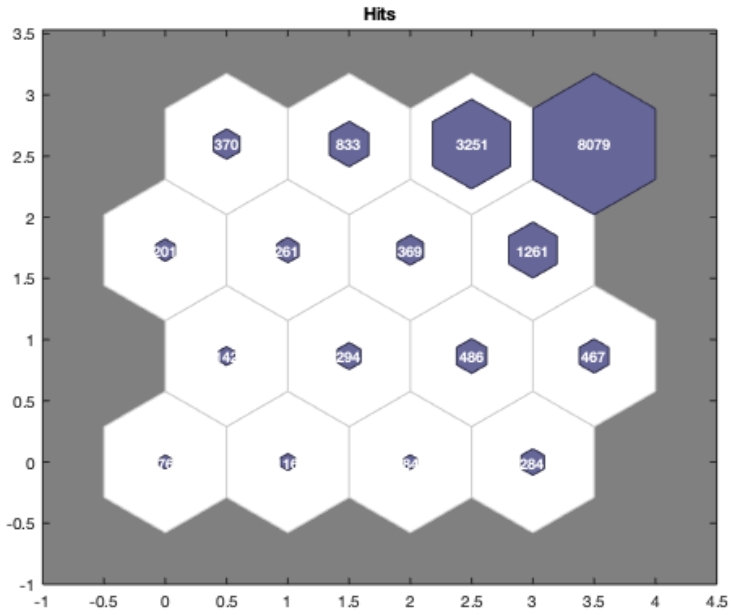
Obrázek 117: Rozložení neuronů pro síť 4x4



Zdroj: Vlastní tvorba.

Stejně jako v předchozím případě za pomocí příkazu *plotsomhits* zobrazíme, jakou množinu podniků reprezentují jednotlivé neurony. Výsledek je vidět na následujícím obrázku. Z něho je zřejmé, že rozdělení dat není rovnoměrné a jeden neuron pokrývá téměř polovinu podniků. Zbytek se pak dělí mezi ostatní neurony. Velikost plochy vybarveného clusteru vyjadřuje počet počet podniků. Bohužel grafický výstup dle nápovědy neumožňuje příliš optimalizačních parametrů, a proto některé číselné hodnoty jsou méně čitelné. Pro rámcovou představu rozdělení dat by však měl být graf dostatečně reprezentativní. Rovněž by mělo být zřejmé, že neurony jsou rozděleny tak, že pravý horní neuron představuje dolní levý neuron na předchozím obrázku. Zde je největší počet podniků a rovněž jsou zde vzdálenosti k sousedům velmi krátké, což odpovídá obrázku 118.

Obrázek 118: Počet podniků, které reprezentují jednotlivé neurony



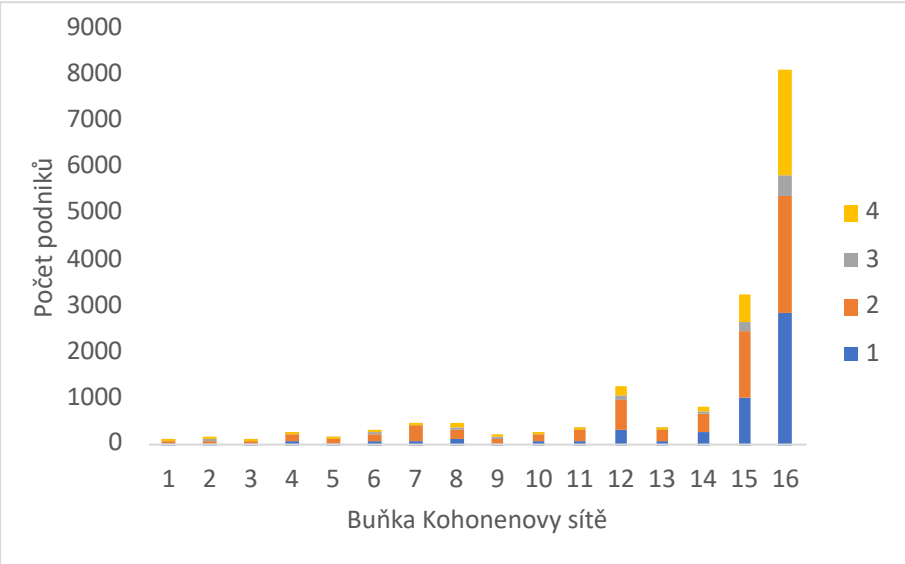
Zdroj: Vlastní tvorba.

Výše uvedený obrázek nám však neříká nic o zastoupení jednotlivých kategorií podniků v jednotlivých clusterech. Abychom mohli provést tuto analýzu musíme využít následujících příkazů:

```
Y = vec2ind(net(X))

Y = Y';

Y = categorical(Y);

Z = [Y, trainTable.Kategorie_podniku];
```

První příkaz převede kategorie do podoby čísla. Jedná se prakticky o opak příkazu dummyvar, který bude využit později pro dopředné neuronové sítě. Následuje transpozice tabulky a zaměnění tak sloupců a řádků. V neposlední řadě se hodnoty převedou na kategoriální a následně je možné je spojit do jedné matice. Vizualizovaný výsledek je vidět na následujícím obrázku. Z celkových hodnot jednotlivých podniků je patrné, o jaké neurony se jedná s ohledem na předchozí grafy. Rovněž je zřejmé, že většina neuronů v sobě obsahuje podniky z různých kategorií.

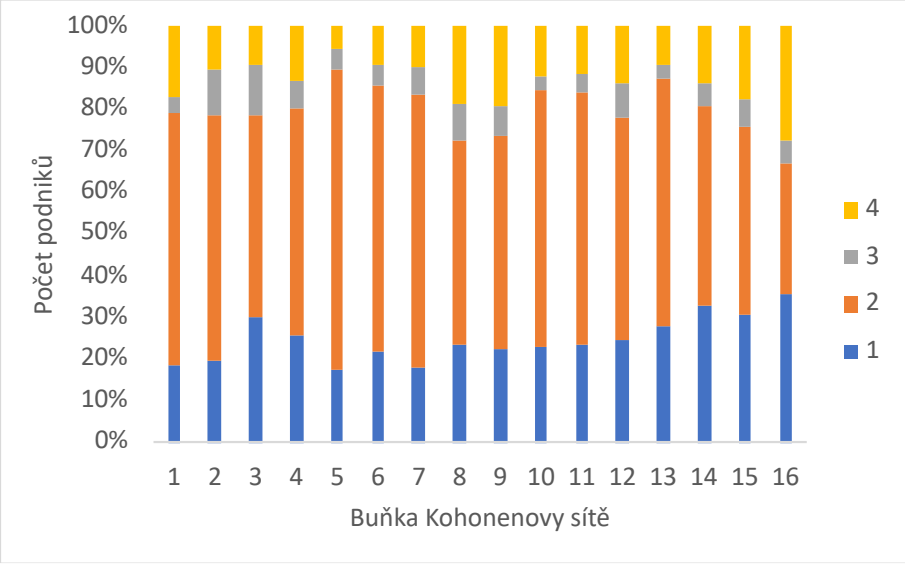
Obrázek 119: Reprezentace jednotlivých neuronů sítě 4x4



Zdroj: Vlastní tvorba.

Předchozí zobrazení nám nic neříká o tom, jak dobře zatřiďuje neuronová síť jednotlivé kategorie podniku a zda náhodou nedochází k rozpoznání skrytých zákonitostí, které by mohly být využity pro naši klasifikaci. Z těchto důvodů provedeme normalizaci dat a převod na procenta pro každý jednotlivý neuron. Výsled je zobrazen na následujícím obrázku. Z obrázku je zřejmé, že žádný z uvedených clusterů nedokázal dobře zatřídit nějakou konkrétní kategorii. Zajímavý je však trend spočívající v tom, že když se blížíme pomyslnému počátku, tak stoupá procentuální zastoupení podniků 1 a 4. kategorie.

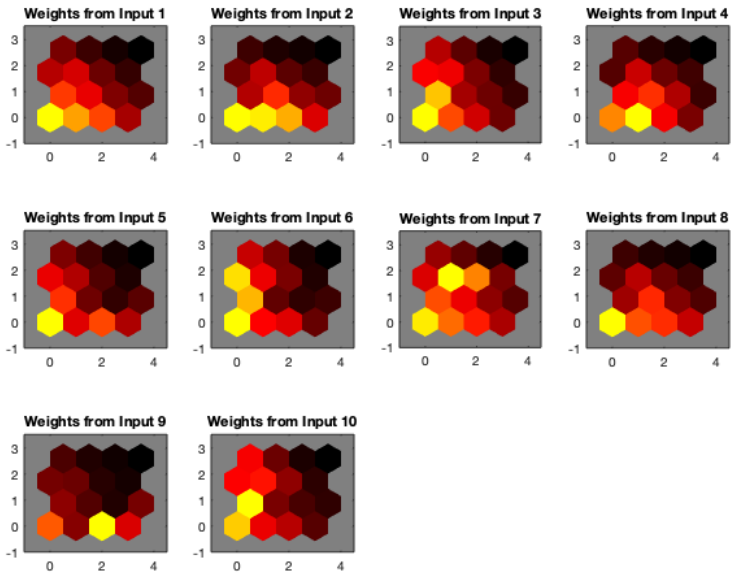
Obrázek 120: Normalizované rozložení podniků v jednotlivých clusterech



Zdroj: Vlastní tvorba.

Poslední analýzou pro daný model bude posouzení vah propojení jednotlivých prediktorů s daným neuronem. Toto provedeme jako v předchozím případě za pomocí příkazu plotsomplanes, který je vidět na následujícím obrázku.

Obrázek 121: Váhy jednotlivých neuronů vztahující se k prediktorům u sítě 4x4



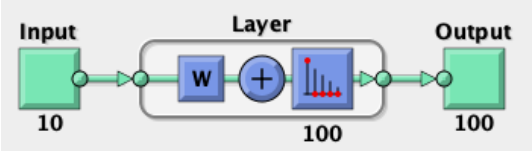
Zdroj: Vlastní tvorba.

S ohledem na výše uvedené bude jako poslední analýza provedeno posouzení sítě o 100 clusterech. Tuto síť vytvoříme za pomocí příkazů:

```
net = selforgmap([10,10]);

net = train(net,X);
```

Schéma sítě je vidět na následujícím obrázku. Síť má stále 10 prediktorů, ale počet výstupů je nyní 100 neuronů. Jedná se tak o mapu 10x10 clusterů.

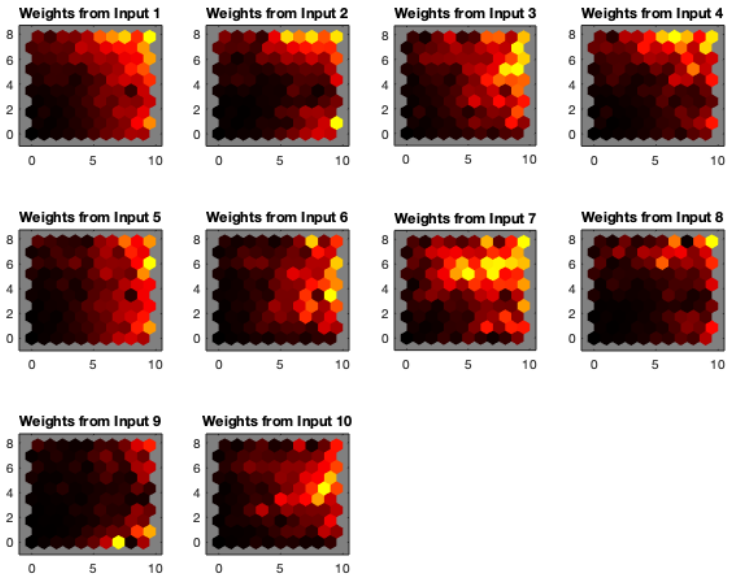
Obrázek 122: Schéma sítě 10 x 10



Zdroj: Vlastní tvorba.

Jako první oproti předchozím dvěma případům provedeme analýzu vah pro jednotlivé prediktory. Toto je zobrazeno na následujícím obrázku. Na rozdíl od předchozích případů jsou zde již patrnější rozdíly mezi jednotlivými prediktory. Zejména prediktor 7, který odpovídá osobním nákladům, má výrazně odlišné charakteristiky než zbytek prediktorů. Na druhou stranu většina prediktorů má v levé části negativní hodnoty vah pro neurony, zatím co v pravé části naopak mírně pozitivní nebo nulový. Toto opět potvrzuje reaktivně slabé hranice z pohledu zatřízení jednotlivých vzorů.

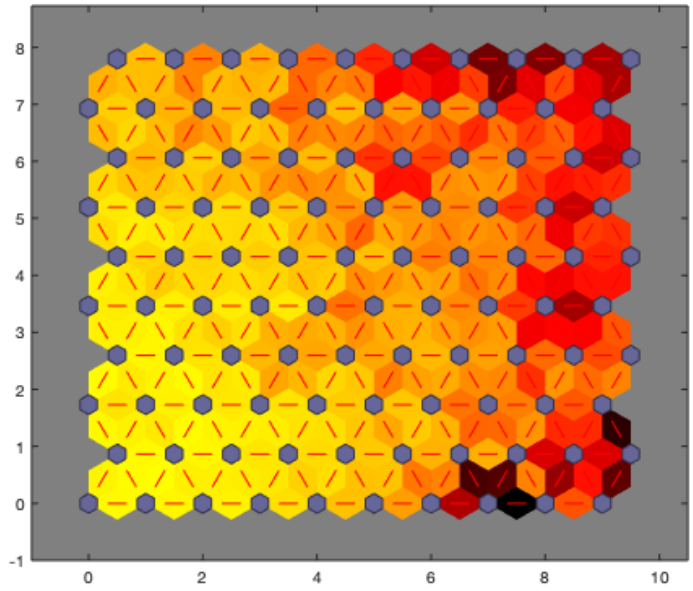
Obrázek 123: Váhy pro jednotlivé prediktory – síť 10x10



Zdroj: Vlastní tvorba.

Dále provedeme analýzu uspořádání jednotlivých neuronů, které reprezentují clustery. Mapa vzájemných vzdáleností, se zobrazí stejně jako v předchozích případech pomocí příkazu *plotsomnd.* Výsledek je vidět na následujícím obrázku. Z něho je patrné, že se neurony dělí na tři skupiny. První skupina (dolní levý roh) je skupina neuronů, které k sobě mají relativně velmi blízko. Oproti tomu pravý horní i dolní roh mají ke svým sousedům velké vzdálenosti. Poslední skupina je ve středu grafu a představuje neurony, které k sobě mají blíže než předchozí skupina, ale jsou více vzdáleny než v případě první skupiny.

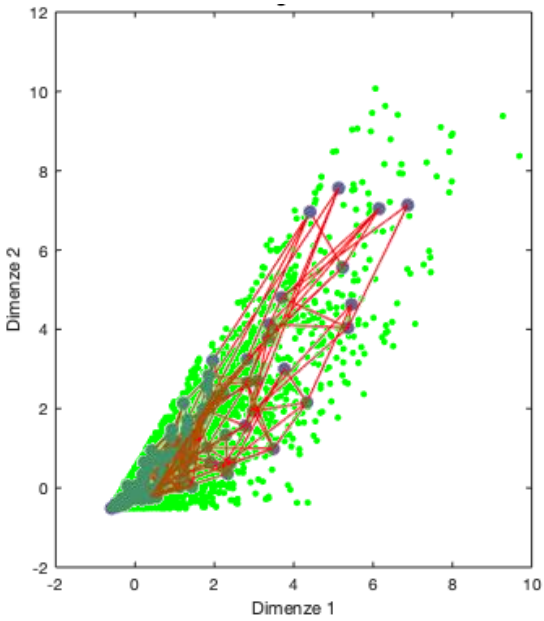
Obrázek 124: Vzdálenost k nebližšímu sousedovi



Zdroj: Vlastní tvorba.

Pokud zobrazíme rozložení neuronů v prostoru reprezentujícím podniky vidíme potvrzení předchozích tvrzení. Výsledek je vidět na následujícím obrázku. Na tomto obrázku jsou dole vpravo neurony velmi hustě uspořádány. Důvodem je velká koncentrace podniků. Naopak vpravo nahoře je uspořádání mnohem řidší a vzdálenosti mezi jednotlivými neurony postupně narůstají.

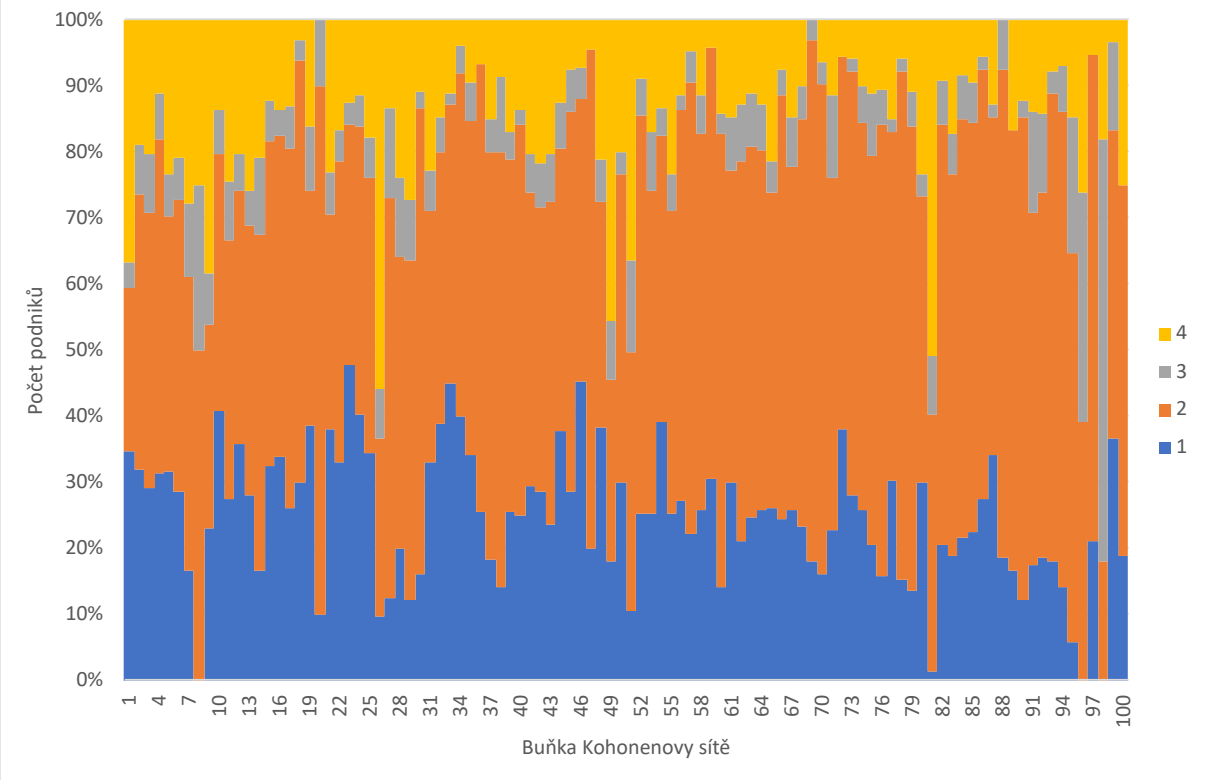
Obrázek 125: Neuronová síť v datech o podnicích



Zdroj: Vlastní tvorba.

Nakonec provedeme analýzu složení zástupců jednotlivých skupin podniků v jednotlivých clusterech. Výsledek je vidět na následujícím obrázku. Tabulka je přílohou tohoto dokumentu. Z grafu je zřejmé, že s výjimkou 3 clusterů se vždy v jednotlivých buňkách nachází různorodé zastoupení podniků. Tři zmiňované clustery neobsahují kategorii 1 podniků. Dále některé buňky neobsahují 3. nebo 4. kategorii podniků, a naopak všechny obsahují určitý podíl 2. skupiny. Síť by tak mohla vyloučit některé podniky při kombinované analýze. Pokud se však na jednotlivé clustery podíváme blíže zjistíme, že v samotném clusteru je velmi málo podniků. Z tohoto pohledu lze předpokládat, že se jedná spíše o nedostatek dat pro takto rozsáhlou síť než o její klasifikační schopnost s ohledem na cíle analýzy. Z důvodu rozsahu práce je zobrazen pouze relativní podíl podniků. Z přílohy je pak patrný absolutní počet pro každý cluster. Stejně jako v předchozích případech jsou zde clustery, které osahují větší zastoupení podniků (přes 2 nebo 3 tisíce podniků), ale také clustery, které obsahují pouze 10 podniků. U clusterů, kde je velký počet podniků je zastoupení podniků relativně rovnoměrné s ohledem na absolutní četnost. Naopak u clusterů s menším počtem podniků některé skupiny chybí, viz výše.

Obrázek 126: Rozložení jednotlivých skupin podniků



Zdroj: Vlastní tvorba.

5.1.8 Dopředné sítě (Feed Forward Networks)

Poslední metodou, kterou využijeme pro analýzu dat budou dopředné neuronové sítě. Bude se jednat o systém učení s učitelem. Síť s deseti skrytými neurony vytvoříme na základě následujícího příkazu:

```
net = patternnet(10);
```

U tohoto typu neuronových sítí je důležité sledovat, aby nedošlo k přeučení sítě, čímž by se sice síť naučila velmi dobře na trénovacích datech, ale zcela pak selhala v případě testovacích dat. K tomuto účelu se používá ještě jedna skupina dat, která se nazývá validační. Data rozdělíme na základě zadání následujících příkazů.

```
net.divideParam.trainRatio = 70/100;

net.divideParam.valRatio = 15/100;

net.divideParam.testRatio = 15/100;
```

Parametr 70 a 15 určuje procentuální zastoupení množiny v souboru. V našem případě jsme tedy rozdělili data na 70 % dat, na kterých budeme síť učit. Dále pak 15 % dat validačních a 15 % dat testovacích.

Před samotným učením sítě je ještě nutné vytvořit vzory, pomocí kterých se bude síť učit. Prakticky se jedná v našem případě o kategorie podniku. Toto provedeme následujícími příkazy.

```
vystup = trainTable.Kategorie_podniku;

vystup = dummyvar(vystup);
```

První příkaz vytvoří ze sloupce tabulky obsahující informace o kategoriích podniku. Dále je nutné tyto kategoriální informace převést na binární rozdělení za pomocí příkazu dummyvar. Binárním rozdělením je myšlen převod, který je vidět na následujícím obrázku:

Obrázek 127: Převod proměnné za pomocí příkazu dummyvar

Původní verze
1
3
4
2

Nová verze			
1	0	0	0
0	0	1	0
0	0	0	1
0	1	0	0

Zdroj: Vlastní tvorba.

Následně provedeme trénování sítě za pomocí následujícího příkazu.

```
[net,tr] = train(net,X,vystup);
```

Schéma sítě je vidět na následujícím obrázku. Ze schématu vyplývá, že síť má 10 vstupních neuronů, jednu skrytou vrstvu, která má 10 neuronů, a 4 výstupní neurony, které představují určení kategorie. Deset vstupních neuronů představuje 10 numerických prediktorů. Síť totiž standardně nepracuje s kategoriálními proměnnými. Toto omezení lze obejít a bude předmětem další analýzy.

Obrázek 128: Schéma dopředené sítě o 10 neuronech

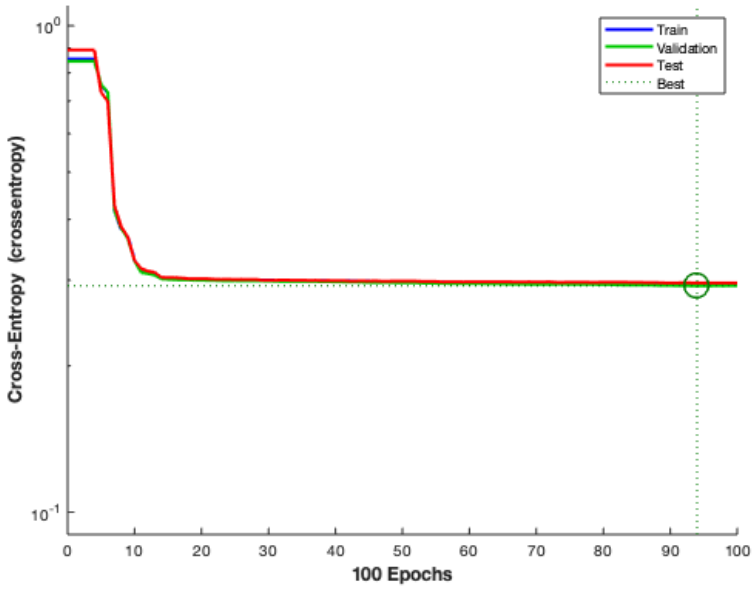
Zdroj: Vlastní tvorba.

Průběh trénování sítě je vidět na následujícím obrázku a zobrazí se pomocí příkazu

```
plotperform(tr)
```

Na obrázku je vidět, že síť upravila velikost vah více jak 90x než došla k optimálnímu výsledku. K zásadním změnám z pohledu míry chyby však došlo po prvních 10 epochách trénování.

Obrázek 129: Průběh trénování



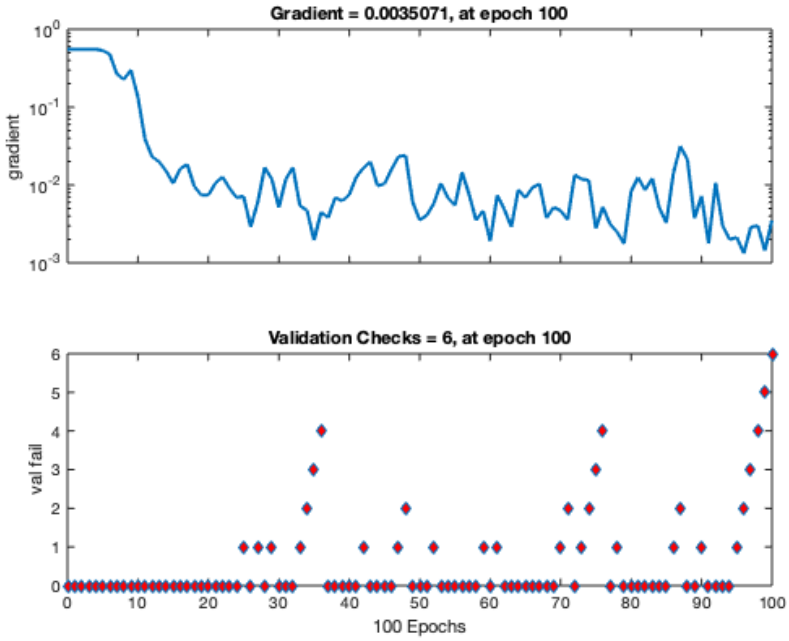
Zdroj: Vlastní tvorba.

Další charakteristiku průběhu trénování získáme za pomocí příkazu

```
plottrainstate(tr);
```

Výsledek je vidět na následujícím obrázku. Horní část představuje, jak se mění gradient s počtem trénovacích epoch. Z průběhu je patrné, že po prvních 10 trénování byla hodnota gradientu v setinách a mírně oscilovala. Svého extrému dosáhla po 94 trénováních, kdy gradient dosáhl hodnoty 3,5 tisíciny, tedy téměř nuly. Jedná se tedy o lokální extrém hledané funkce. Spodní část vyjadřuje míru chyby na validačních datech. Pokud dojde k selhání 6x po sobě, Matlab automaticky ukončí proces trénování.

Obrázek 130: Změna gradientu a chyba validačních dat



Zdroj: Vlastní tvorba.

Další informaci o neuronové síti nám zobrazí histogram chyb. Ten je možné zobrazit pomocí příkazů:

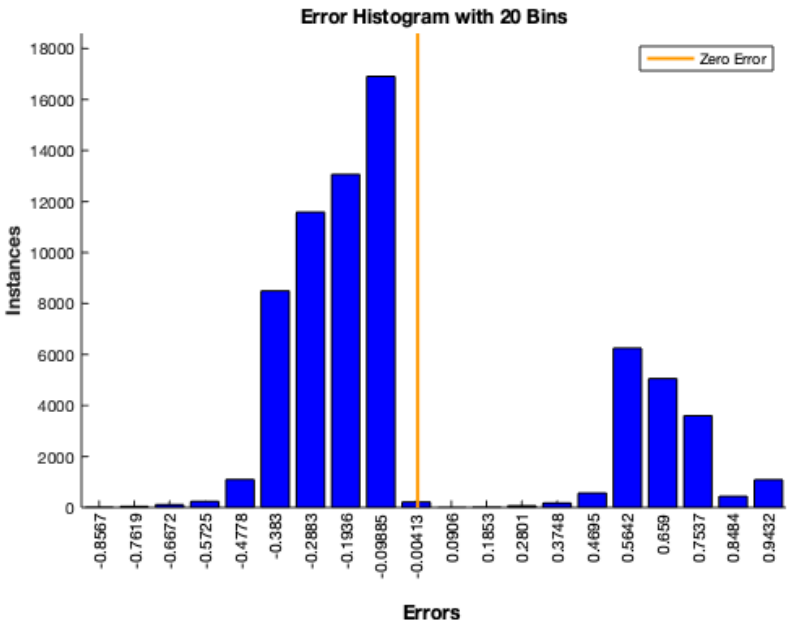
```
y = net(X);

e = vystup' - y;

ploterrhist(e);
```

Výše uvedený zdrojový kód nejdříve zatřídí vzory v množině, kterou jsme použili na trénovací, validační a testovací data. Dále odečte hodnoty reálného zatřízení od predikovaného a výsledek uloží do proměnné e. Tuto následně vykreslíme pomocí příkazu

```
ploterrhist(e,'bins',20)
```

Výsledný graf je vidět na následujícím obrázku a prakticky se jedná o stejný typ grafu, jako kdybychom vykreslili klasický histogram. S ohledem na to, že data mohou nabývat hodnoty 1 a 0, je stupnice x rozdělená na 20 kategorií od -1 do 1. Hodnoty -1 by mohla síť dosáhnout v případě, že podnik v dané kategorii není, ale síť jej tam zcela jednoznačně zařadila. Výstup sítě však není zcela jednoznačný a tomu odpovídá i zobrazení histogramu. Největší počet chyb je okolo 0. Což značí, že je chyba velmi malá a podnik tak v dané kategorii není a je predikována mírná pravděpodobnost jeho začlenění, nebo tam podnik je a je predikováno začlenění, které není tak jednoznačné. Zcela se stejnou logikou pak lze pohlížet i na chybu okolo hodnoty 1 a níže. Zde podnik v kategorii je (hodnota 1), ale síť predikuje téměř nulovou pravděpodobnost jeho zařazení do dané kategorie. Výše uvedený graf tedy odpovídá množství kategorií, které zatřiďujeme a způsobu predikce neuronové sítě. Pokud by bylo třízení lepší a jednoznačnější byly by sloupce menší. Pokud bychom zobrazili graf z interaktivního rozhraní, byly by jednotlivé sloupce rozděleny na validační, testovací a trénovací data. Jelikož je jejich zastoupení poměrně rovnoměrné, tak je zde dále nebudeme uvádět.

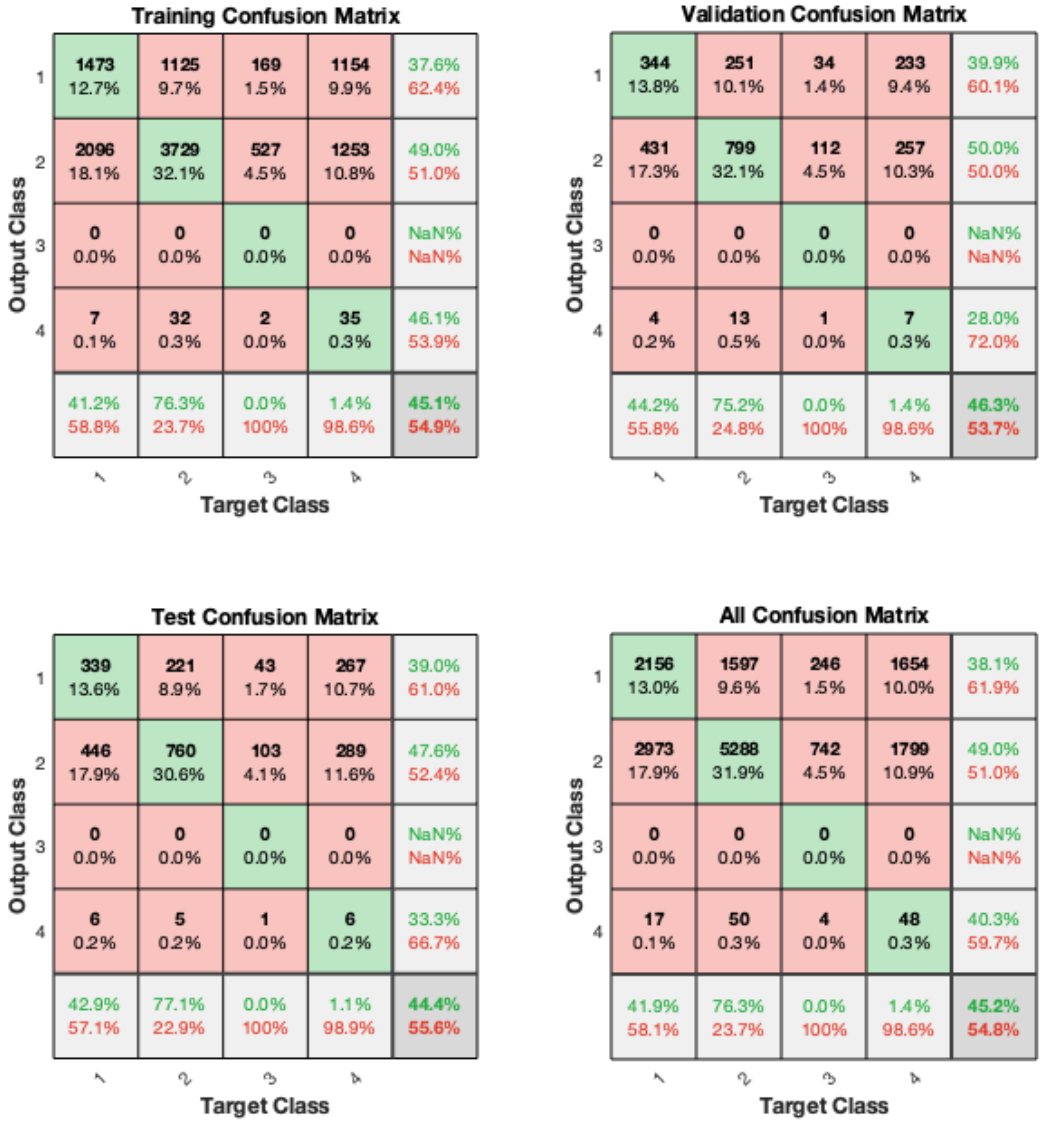
Obrázek 131: Histogram chyb



Zdroj: Vlastní tvorba.

Kvalitu predikce provedeme za pomocí zobrazení konfuzních matic. Ty zobrazíme z interaktivního panelu při trénování sítě. Výsledek je vidět na následujícím obrázku. Na obrázku jsou vidět 4 konfuzní matice. Každá je pro jednotlivé skupiny dat a dále je zde jedna výsledná statistika na celou množinou dat. Z jednotlivých konfuzních matic je zřejmé, že model se natrénoval tím způsobem, že predikuje relativně dobře 2. množinu. Úspěšnost zatřízení je přes 47 %. Dále relativně dobře 1. skupinu podniků. Zde model dosahuje úspěšnosti nad 37 %. V neposlední řadě i 4. skupinu podniků, kde model dosahuje celkové úspěšnosti nad 40 %. Model stejně jako některé předchozí modely vynechává 3. skupinu podniků.

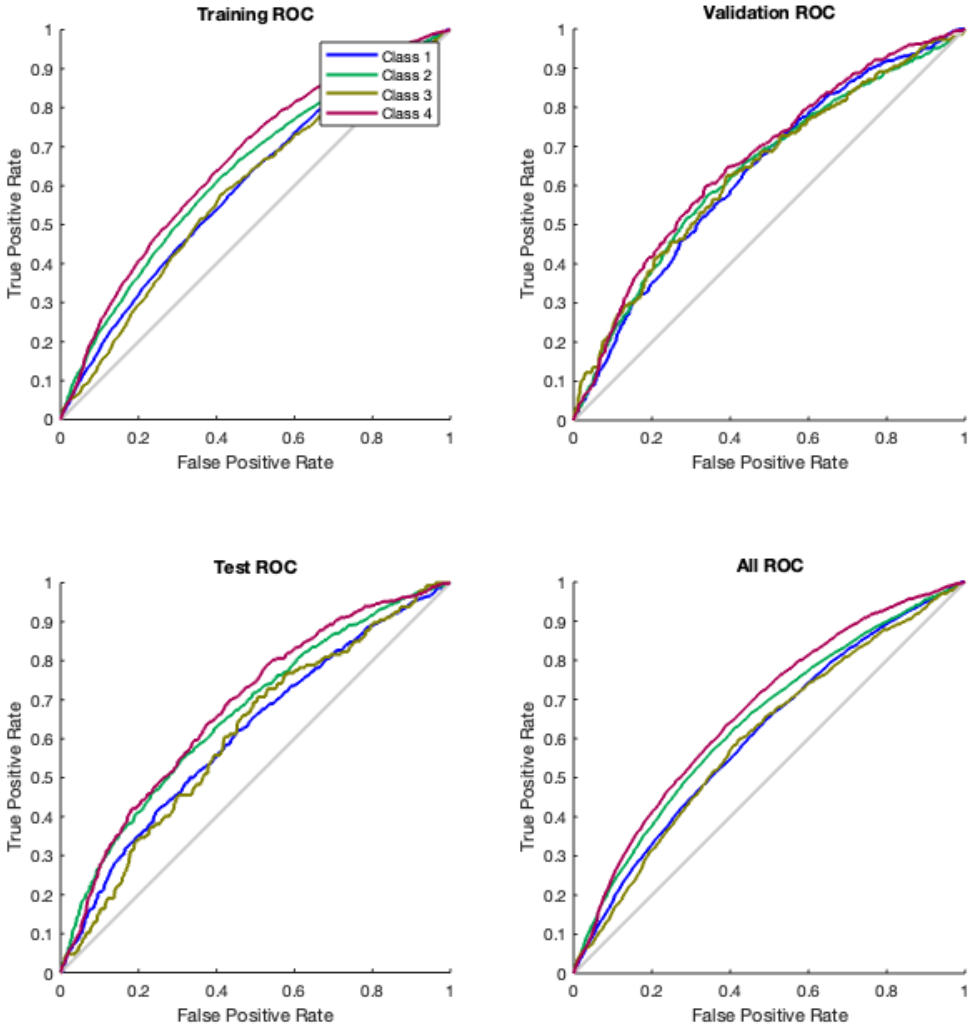
Obrázek 132: Konfuzní matice pro neuronovou síť



Zdroj: Vlastní tvorba.

Pro analýzu výkonnosti neuronových sítí se využívá i ROC (receiver operating characteristic) křivka. Výsledné křivky jsou zobrazeny na následujícím obrázku a byly opět vygenerovány z interaktivního prostředí při trénování modelu. Tato porovnává výkonost modelu s náhodným zatřízením. Náhodné zatřízení je reprezentováno přímkou o úhlu 45°. Hodnota nad touto křivkou je pozitivní. Čím více je hodnota nad touto křivkou blíže 1 tím je třízení jednoznačnější a lepší. V našem případě dosahujeme hodnot ve všech případech nad úrovní náhodného třídění. Křivka však není příliš vzdálená od přímky, a proto je kvalita modelu relativně nižší. S ohledem na výsledky konfuzní matice je však výsledek obdobný ve srovnání s ostatními modely.

Obrázek 133: ROC křivky pro model neuronové sítě



Zdroj: Vlastní tvorba.

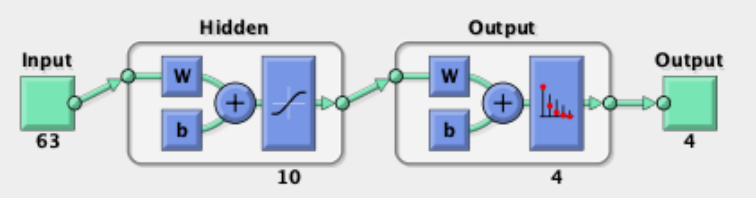
Nyní provedeme optimalizaci modelu a začleníme do vstupů i kategoriální proměnné. Abychom toto mohli udělat provedeme následující příkazy:

```
X2 = X;

for i = 1:5

    X2 = [dummyvar(trainTable{:, i}), X2];

End
```

Nejdříve si numerické prediktory uložíme do nové proměnné X2. Dále provedeme cyklus for, který proběhne 5x. Hodnota počtu opakování odpovídá počtu kategoriálních proměnných, které jsou obsaženy v trénovací tabulce (prvních 5 sloupců). Tyto kategoriální proměnné převedeme pomocí funkce dummyvar na matici, kde sloupce reprezentují jednotlivé kategorie, přičemž podnik do dané kategorie může patřit (hodnota 1), nebo ne (hodnota 0). Výslednou matici připojíme ke stávajícím datům a celý proces opakujeme.

V dalším kroku provedeme trénování neuronové sítě stejně jako tomu bylo v předchozím případě. Rovněž provedeme zobrazení schématu sítě. Ta je vidět na následujícím obrázku. Z obrázku je zřejmé, že vstupní vrstva obsahuje 63 neuronů, což odpovídá počtu prediktorů. Hodnota poměrně vzrostla oproti předchozím 10 neuronům. Toto způsobilo množství různých kategorií, kde každá kategorie vstupuje do samostatného neuronu.

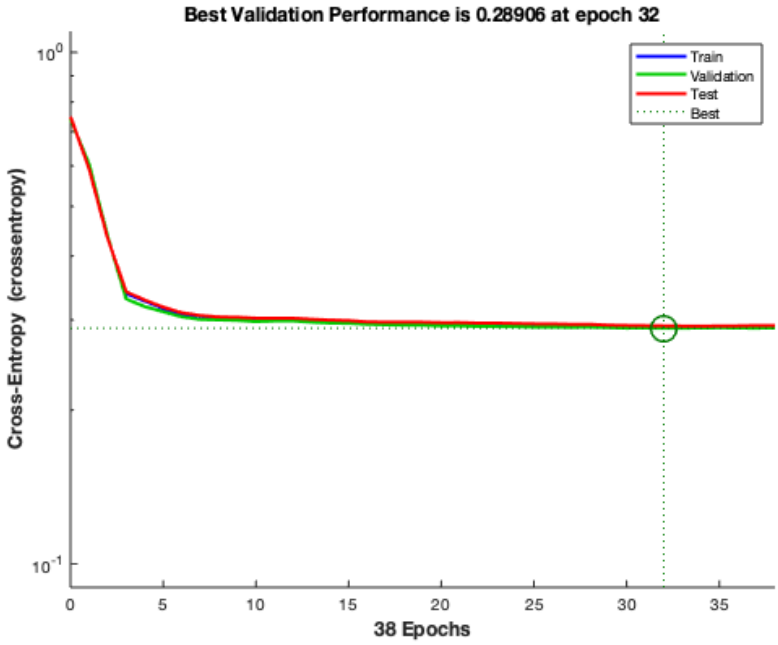
Obrázek 134: Schéma neuronové sítě s kategoriálními proměnnými



Zdroj: Vlastní tvorba.

Stejně jako v předchozím případě dále provedeme trénování neuronové sítě. Průběh trénování je zobrazen na následujícím obrázku. Z něho je patrné, že proces trénování se zkrátil téměř o více jak 2 třetiny. S ohledem náhodnosti inicializačních podmínek však tato změna nemusí souviset s počtem přidaných vstupních neuronů. Průběh trénování je však velmi podobný předchozí síti.

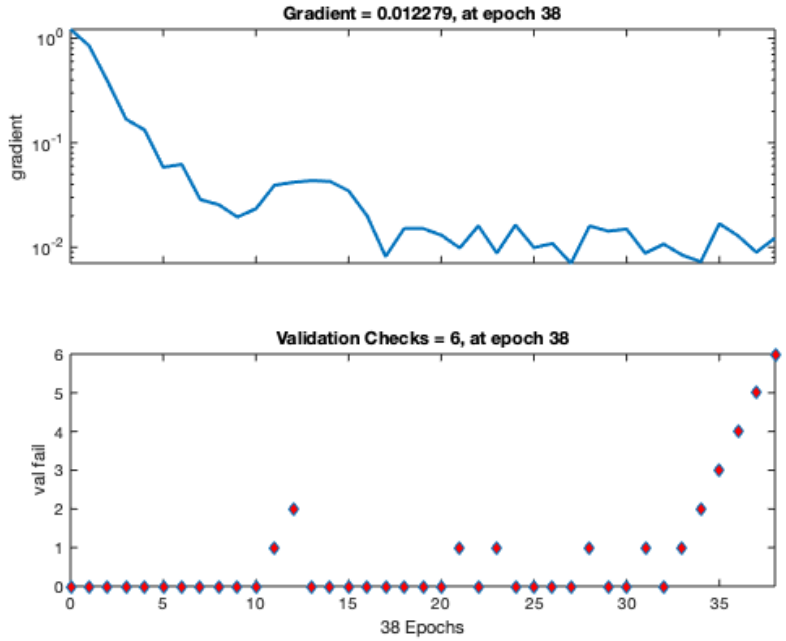
Obrázek 135: Průběh trénování neuronové sítě s kategoriálními prediktory



Zdroj: Vlastní tvorba.

Dalším informaci o průběhu trénování získáme zobrazením grafu za pomocí funkce plottrainstate. Výsledek je zobrazen na obrázku níže.

Obrázek 136: Průběh vývoje gradientu a chyba validačních dat

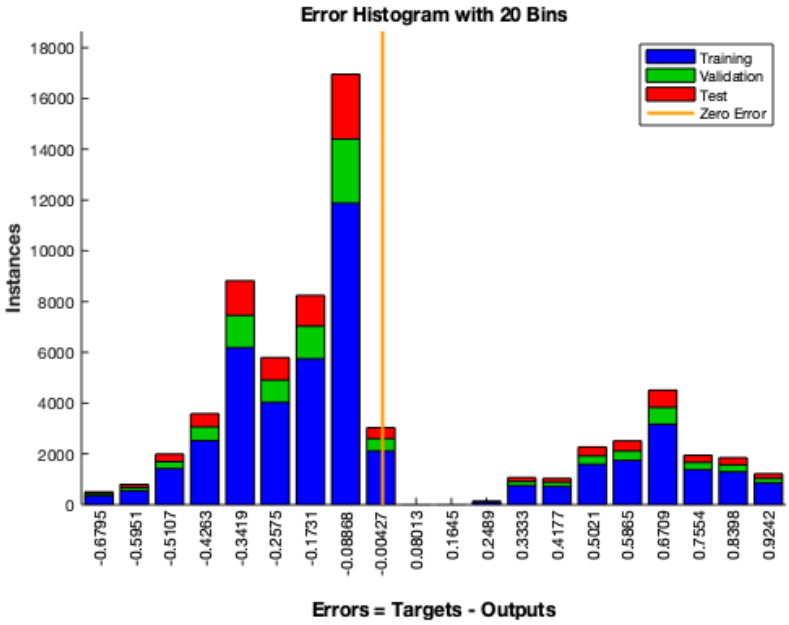


Zdroj: Vlastní tvorba.

Z něho je zřejmý vývoj gradientu hledané funkce neuronové sítě. Stejně jako v předchozím případě dochází k postupnému snižování hodnoty gradientu a jeho postupné oscilaci. Gradient dosáhne svého optima na úrovni 0,012, což je mírně vyšší hodnota, než byla v předchozím případě. Důvodem ukončení procesu učení je množství chyb, kterých síť dosáhla 6x za sebou. Toto reprezentuje spodní část grafu.

Nyní zobrazíme i histogram chyb. Na rozdíl od předchozího případu jej však zobrazíme z interaktivního rozhraní trénování sítě. Výsledek je zobrazen na obrázku níže. Z obrázku je patrné obdobné rozložení jako v předchozím případě. Opět je zde velké množství chyb, které jsou v hodnotách od 0 níže a od 1 níže. Toto stejně jako v předchozím případě ukazuje na nejednoznačnost zatřízení dat do jednotlivých kategorií, které provádí neuronová síť.

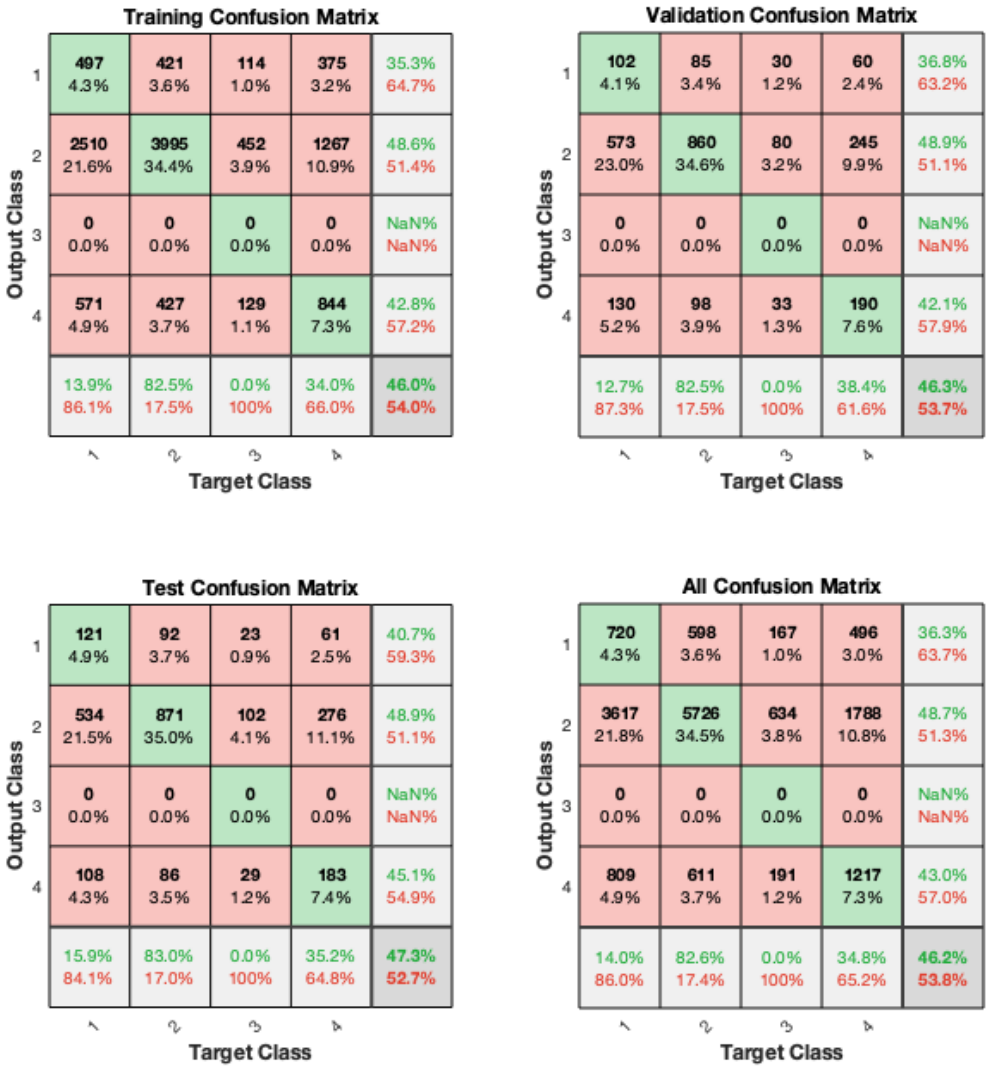
Obrázek 137: Histogram chyb



Zdroj: Vlastní tvorba.

Zobrazením konfuzních matic budeme moci porovnat, jestli došlo ke zlepšení výkonnosti modelu. Matice jsou zobrazeny na následujícím obrázku. Z něho je patrné, že v celkovém důsledku došlo ke zlepšení predikčních schopností modelu o cca 1 %. Toto především díky lepší schopnosti predikovat podniky ve 4. kategorii. Stále však přetrvává stav, že síť zcela ignoruje 3. kategorii podniků.

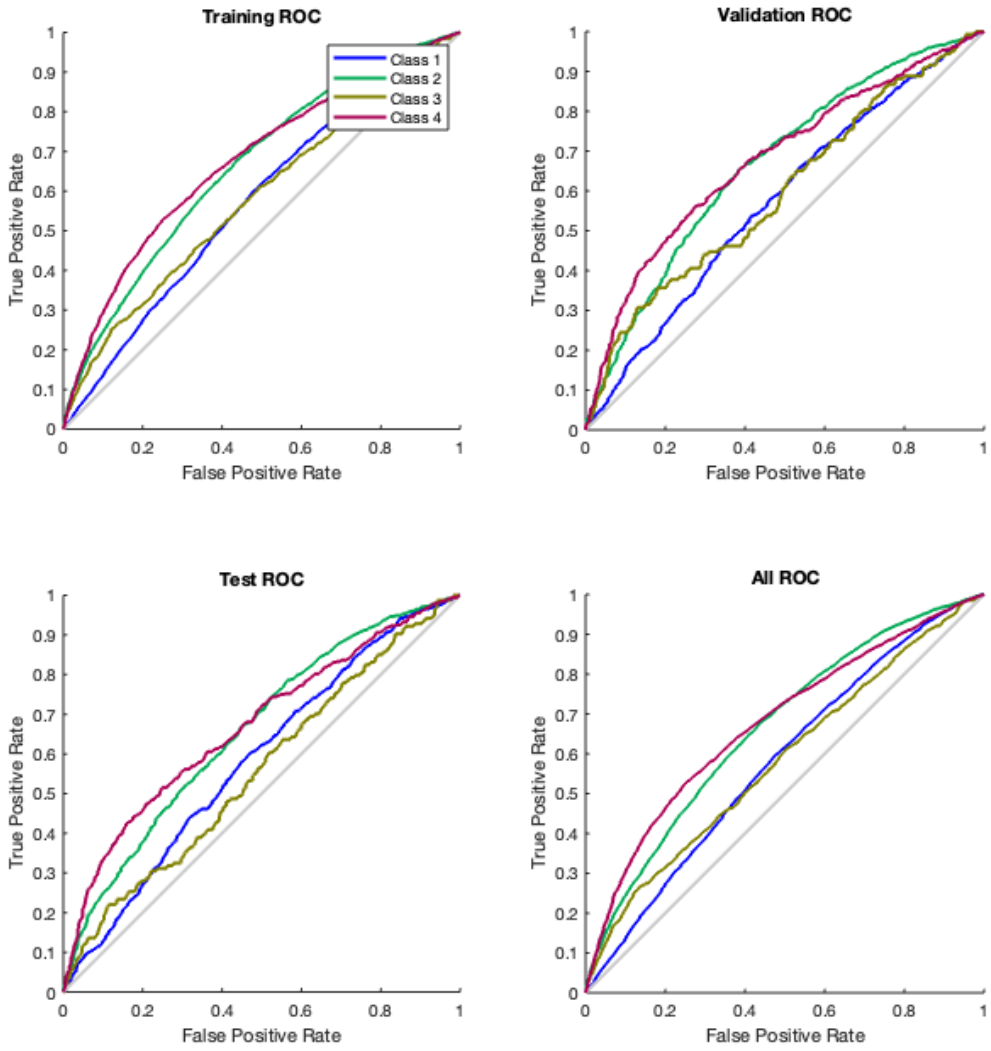
Obrázek 138: Kofuzní matice pro neuronovou síť s kategoriálními prediktory



Zdroj: Vlastní torba.

Nakonec pro tento model zobrazíme i ROC křivky pro jednotlivé kategorie dat (trénovací, testovací a validační). Výsledek je vidět na následujícím obrázku. Z obrázků je patrné, že ve všech případech je predikce lepší než náhodná procházka. Zahnutí grafu nad přímkou je relativně malé, a proto model nedosahuje příliš dobrých výsledků, což odpovídá i výsledkům konfuzních matic viz výše.

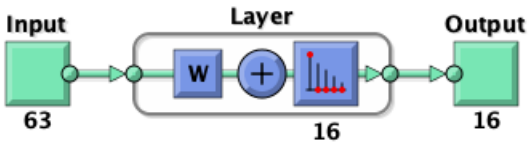
Obrázek 139: ROC křivka pro model neuronové sítě s kategoriálními prediktory



Zdroj: Vlastní tvorba.

Jelikož jsme pro neuronovou síť upravili data takovým způsobem, abychom je mohli využít pro predikci (převod pomocí proměnné dummyvar), využijeme této datové sady a provedeme opět výpočet neuronové sítě bez učitele. Výsledná síť bude mít 16 clusterů a její schéma je zobrazeno na následujícím obrázku.

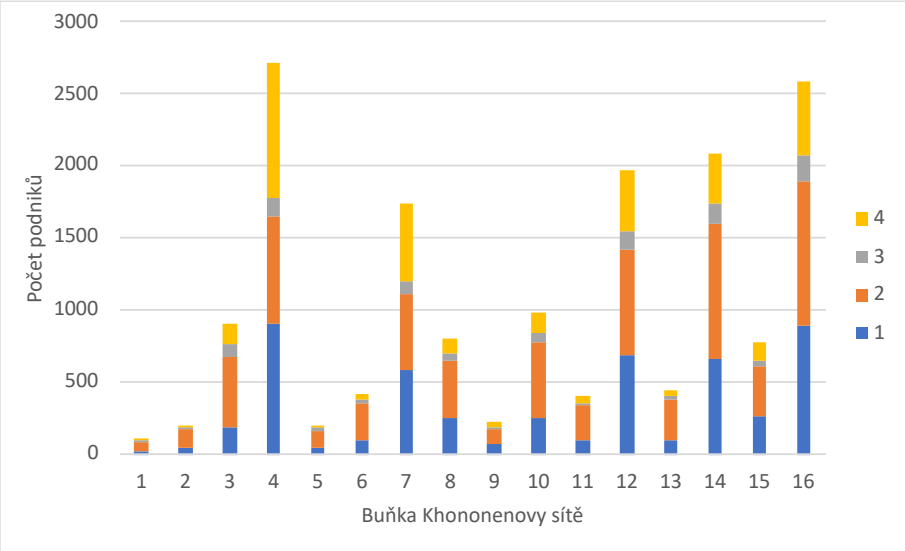
Obrázek 140: Schéma Kohonenovy sítě s kategoriálními prediktory



Zdroj: Vlastní tvorba.

Stejně jako v předchozích případech zobrazíme, jak jednotlivé clustery reprezentují zastoupení kategorií podniku. Výsledek je vidět na následujícím obrázku. Na obrázku je vidět, že na rozdíl od předchozích případů, je Kohonenova síť rozdělena mnohem pravidelněji. Už zde není jednoznačně dominující cluster, který obsahoval řádově více podniků, než tomu bylo u dalších. Dále je zde vidět několik clusterů, které jsou relativně malé co do počtu podniků, a naopak několik, které jsou ve skupině větších clusterů.

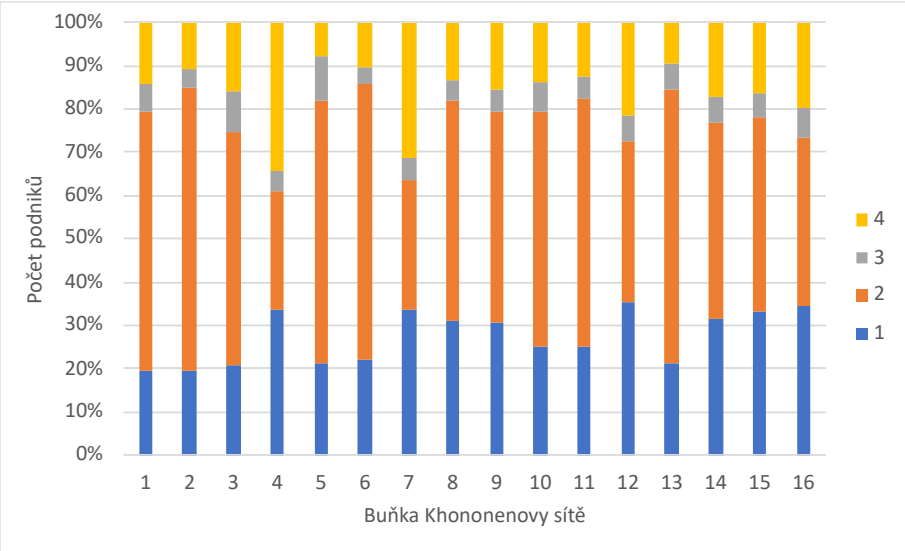
Obrázek 141: Kohonenova síť s numerickými prediktory – počet podniků v jednotlivých clusterech



Zdroj: Vlastní tvorba.

Abychom mohli analyzovat, jestli se v nějakém clusteru nezobrazuje více podniků určité kategorie, provedeme převod počtu podniků na procenta v daném clusteru. Výsledek je vidět na následujícím obrázku. Z něho je zřejmé, že žádný neuron nevymezuje svůj prostor, tak aby v něm byla zastoupená některá z kategorií podniku více než jiná. Jinými slovy zde stále přetrvává problém, že Kohonenova síť se samoorganizuje na základě jiných vlastností podniků a pro účely analýzy není příliš využitelná.

Obrázek 142: Procentní podíl zastoupení podniků v Kohonenově síti s kategoriálními prediktory



Zdroj: Vlastní tvorba.

6 Diskuze výsledků

Předmětem kapitoly je kritická analýza výstupů z testovaných modelů z pohledu na stanovené cíle řešení. Součástí příslušných komentářů je i komentář na navazující problematiky související s hlavním cílem.

První výzkumná otázka byla zaměřena na vazbu mezi místem podnikání v České republice a výsledkem hospodaření. Analýza včetně výstupů je uvedena na str. 65 až 67. Z této analýzy vyplynulo, že místo podnikání ovlivňuje počet podniků, které dosahují kladných hodnot EVA, obdobně se liší i další sledované kategorie. Míra ovlivnění je na druhé straně relativně malá.

Další otázka byla cílena na vztah mezi výší hodnot EVA a počtem zaměstnanců u příslušné skupiny MSP. Výsledky z provedených analýz jsou uvedeny na straně 68 až 69. Z této analýzy vyplynulo, že vyšší počet zaměstnanců u malých a středních podniků nemá přímý vliv na vyšší hodnotu EVA.

Poslední stanovenou otázkou bylo, zdali existuje podstatný rozdíl mezi zaměřením podniku a tvorbou hodnoty EVA. Z analýzy, která je prezentována na straně 69 až 70 vyplynula výrazná sektorová diferenciace z hlediska tvorby přidané hodnoty.

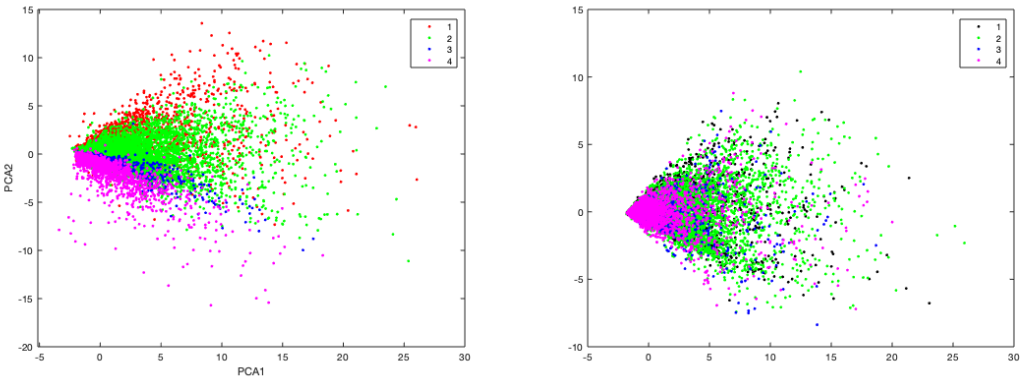
6.1 PCA

Principal component analitics byla provedena na normalizované sadě dat, která obsahovala (první fáze) a později neobsahovala (druhá fáze) informace o hospodaření podniků. Vzhledem k tomu, že informace o hospodaření podniků byla klíčová pro jejich segmentaci dle metody INFA došlo v první fázi k rozdělení dat v prostoru takovým způsobem, kdy bylo snadné definovat klíčové shluky i přesto, že se nejedná o metodu učení s učitelem. V druhé fázi, tedy při odstranění těchto prediktorů, se jednotlivé množiny navzájem překrývaly. Porovnání obou výsledků je patrno z obr. 145. V levé části obrázku jsou ve výsledcích obsaženy údaje o hospodaření podniku, pravá část pak tyto údaje nevstupovaly do provedené analýzy.

Přesto, že v prvním případě je množina snadno rozdělitelná, a tedy i predikovatelná, mělo smysl věnovat se predikci dat na množině, která nebude obsahovat informace o hospodářském výsledku. Pokud by totiž informace v množině byla, pak by osoba, která potřebuje znát výsledek zatřídění podniku, mohla hodnotu vypočítat ze známých vzorců a nemusela by využít klasifikačních metod. Rovněž by tento přístup nebyl v souladu se stanoveným cílem práce.

Metoda PCA byla dále použita pro vizualizaci výsledků dalších metod s ohledem na redukci n dimenzionální úlohy.

Obrázek 143: Porovnání výsledků PCA

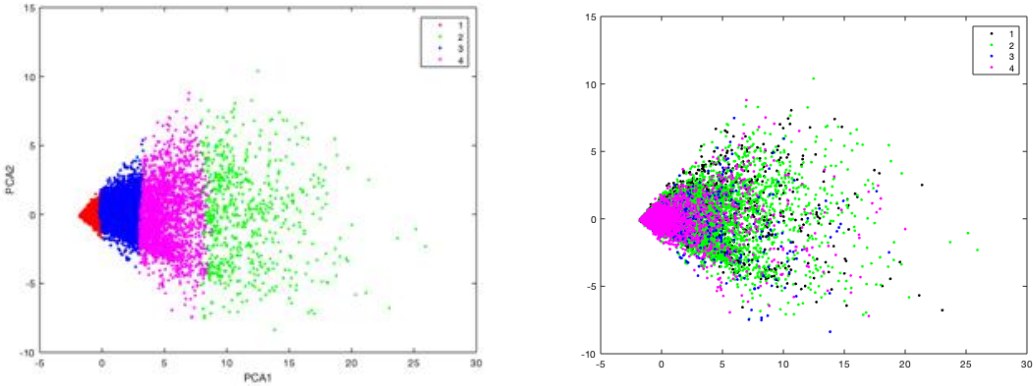


Zdroj: Vlastní tvorba.

6.2 K-Means clustering

Metoda k-means clusterin byla opět využita jako metoda bez učitele. Výsledky rozdělení jsou patrny z obr. 146, kde v levé části je vidět dělení do segmentů dle metody a vpravo data, reprezentující reálné členění. Z uvedeného obrázku je patrné, že členění zde probíhá na základě zcela jiných vlastností, a proto je s ohledem na cíle práce nevyužitelné.

Obrázek 144: Výsledky K-Menas clustering

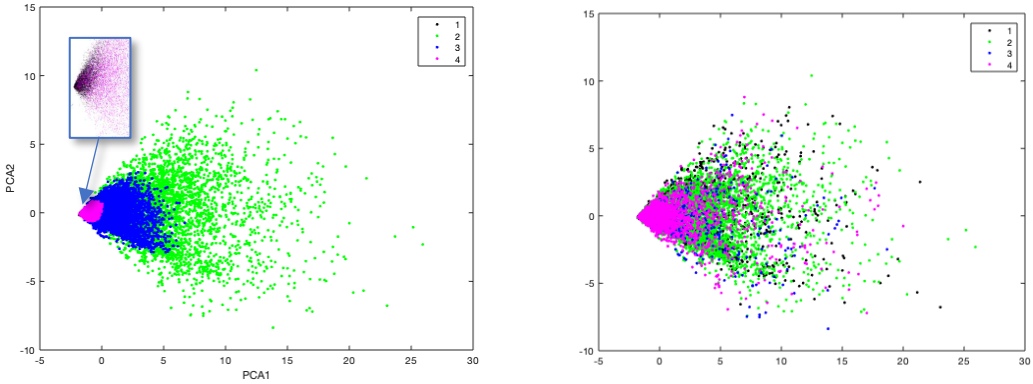


Zdroj: Vlastní tvorba.

6.3 Gaussian Mixture Models

V případě využití Gaussian mixture modelu se získané výstupy jeví na první pohled výrazně pozitivnější a průkaznější (viz levá část obrázku). Při podrobnější analýze se však tato kvalita výsledků neprokázala, neboť uvedená data se významně překrývala. Díky tomu i rozdělení podniků pomocí této metody nebylo příliš využitelné s ohledem na cíl práce.

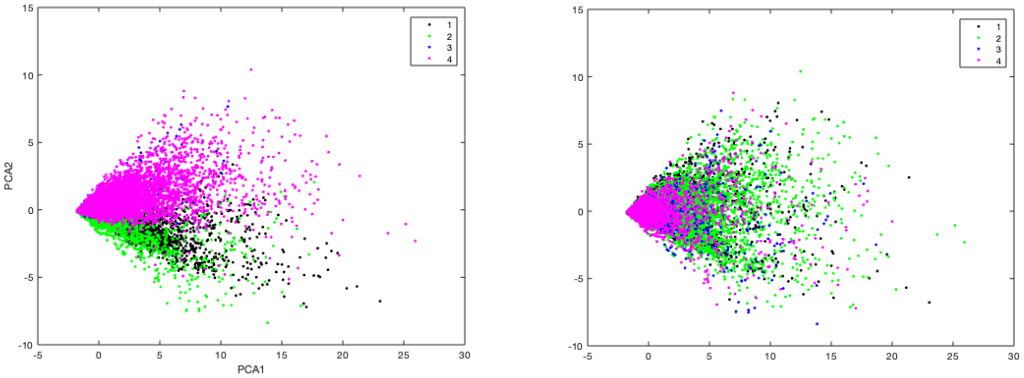
Obrázek 145: Gaussian mixture model – výsledky



6.4 Hierarchické členění

Hierarchické členění (obr. 148) přineslo příznivý a progresivní výsledek v mnohém shodný s výsledky metody PCA s daty o hospodářském výsledku. Jelikož tato data nemohla být využita pro predikci, tak i výsledky z této analýzy nelze využít pro snadnou klasifikaci z důvodu stanovených cílů práce.

Obrázek 146: Výsledek hierarchického členění

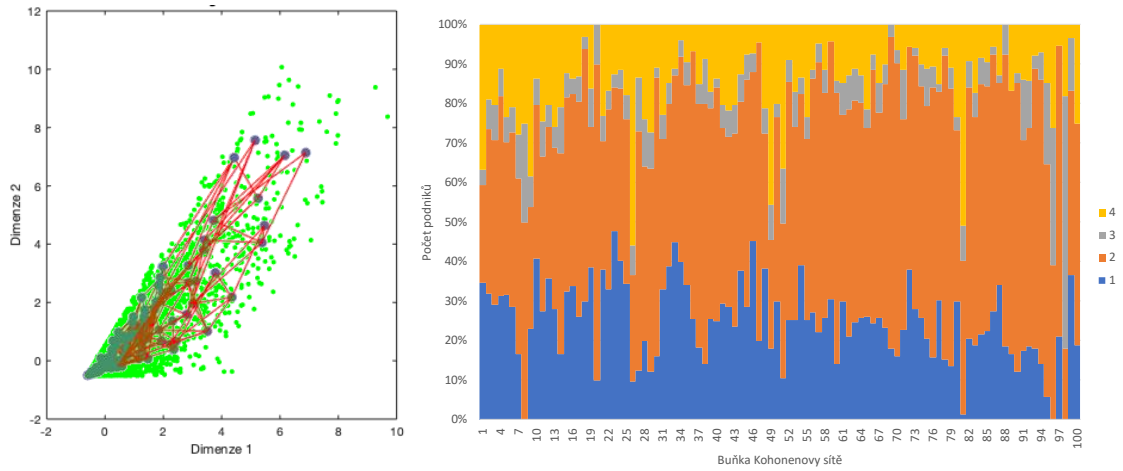


Zdroj: Vlastní tvorba.

6.5 Kohonenovy sítě

Další metodou, která byla využita pro klasifikaci bez učitele, byla Kohonenova síť. Výsledek ve variantě 100 neuronů (síť 10x10) je patrný z obr. 149 vlevo. Výstup Matlabu nám jednoduše neumožňuje rozložení neuronů probarvit jednotlivými clustery, proto jsme pro analýzu využili další graf, který je vidět na obrázku vpravo. Z obrázku je patrné, že žádný z neuronů nereprezentuje nějakou sledovanou skupinu podniků. V případě, že nějaká skupina podniků v daném clusteru chybí, je to způsobeno nedostatkem predikovaných dat (počet zatřízení menší jak 100). I v tomto případě není tato metoda využitelná pro naplnění stanovených cílů práce.

Obrázek 147: Neuronová síť v datech o podnicích



Zdroj: Vlastní tvorba.

6.6 Souhrnné výsledky učení s učitelem

Souhrnné výsledky jednotlivých algoritmů, které byly použity pro predikci, jsou uvedeny v tab. 3. Z tabulky je zřejmé, že úspěšnost predikce se pohybovala od 40 do 54 %. Data v tabulce představují procento správně zatříděných dat na dané množině. Z tabulky je rovněž patrné, že optimalizace hyperparametrů nejrůznějšími (vlastními i automatickými způsoby) nepřinesla zásadní vliv na výsledek. V řadě případů je zřejmé, že výsledná predikce kategorizace podniků dle hodnoty má lepší procento, než je tomu u jiných metod, ale metoda sama o sobě zcela ignoruje predikci 3 a v některých případech i 4. kategorie podniků. S ohledem na cíl je nesporné, že tento typ predikce pro nás není prakticky využitelný. Z hlediska našich potřeb jsou využitelné ty metody, které predikují podniky ve všech kategoriích. Z těchto metod nejlépe vychází metoda stromové klasifikace, která dosáhla 45 % úspěšnosti.

Tabulka 3: Souhrnné výsledky metod učení s učitelem

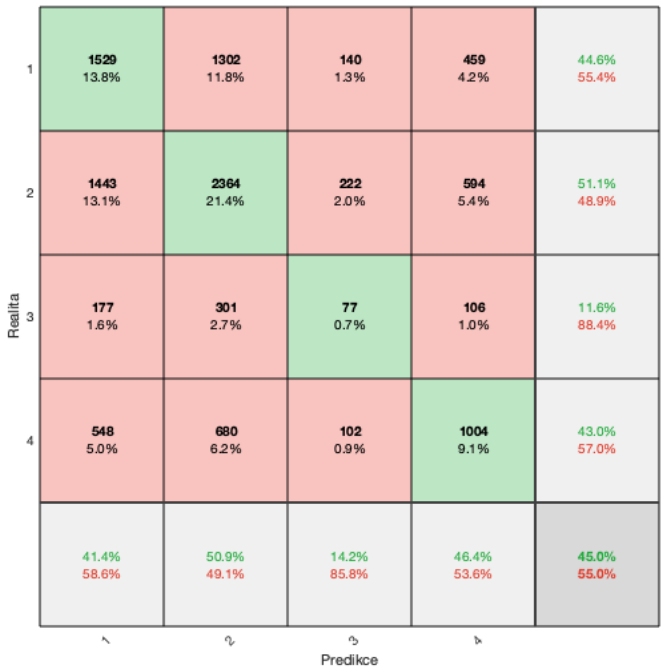
Kategorie podniku	1	2	3	4	Celková úspěšnost
Knn	37,8	50,2	16,3	41,2	43,1
Tree	44,6	51,1	11,6	43	45,0
Tree – opt	51,3	73,4	0	36,8	54,4
NB	37,7	44,2	13,1	20,9	41,7
NB – opt	26	74,1	0,3	28,6	45,1
DA	3,3	95,3	1,9	0,5	41,4
DA – opt	12,6	96,3	0	0	44,8
SVM	29,5	78,2	0	7,2	43,4
SVM - opt	20,8	84	0	14,6	44,7
NN	38,1	49	0	40	45,2
NN - klas	36,3	48,7	0	43	46,2
Průměr	**33,7**	**65,5**	**3,9**	**25,1**	**45,0**
Max	**51,3**	**95,3**	**16,3**	**43,0**	**54,4**
Min	**3,3**	**44,2**	**0,0**	**0,0**	**41,4**

Zdroj: Vlastní tvorba.

Procento 45 % úspěšnosti modelu by mohlo působit negativně z pohledu dosažených výsledků. Je však třeba si uvědomit, že jsou zařazovány 4 kategorie podniku. Z pohledu náhodného zatřídění by tak měla být úspěšnost okolo 25 %. Model tak dosahuje prokazatelně řádově větší přesnosti. Tato přesnost není zajištěna zacílením na nejčetnější množinu, což je velmi důležitý parametr s ohledem na ostatní modely. Dále je třeba zohlednit cíle vlastníků a manažerů malých a středních podniků. Tito manažeři a vlastníci velmi často ani neznají ekonomickou přidanou hodnotu, nebo se jí v praxi neřídí. Klíčové pro ně je, aby podnik dlouhodobě prosperoval a dosahoval „zisku“ nad úrovní bezrizikových investic. Zisk je zde uveden v uvozovkách, neboť musíme vzít v úvahu častou optimalizaci hospodaření podniku z pohledu daňových odvodů. Tato skutečnost je v našich podmínkách velmi frekventovaná a prakticky představuje jednu z nejzásadnější limity prováděných analýz.

Pro lepší interpretovatelnost výsledků modelu stromu je vhodné použít zobrazení v kompletní konfuzní matici (obr. 148). Z obrázku můžeme vysledovat, že pokud model určí, že se podnik bude nacházet v kategorii 1 (44,6% pravděpodobnost) nebo v kategorii 2. (51,1% pravděpodobnost), bude podnik z více jak 80% pravděpodobností dosahovat zisku nad úrovní bezrizikové sazby r_f. Z této hodnoty jasně vyplývá, že výsledky výzkumu mohou být použity pro model benchmakringu a následně v rámci nastavení vnitropodnikových cílů.

Obrázek 148: Konfusní matice – stromový model



Zdroj: Vlastní tvorba

Ve stromové analýze, stejně jako v jiných analýzách, byly využity prediktory, které mohou manažeři ovlivnit. Stejně tak byly využity i prediktory, které jsou dány okolím (velikost obce, kraj, …). Tyto skutečnosti manažer téměř nemůže ovlivnit, ale může s nimi pracovat z pohledu diverzifikace portfolia, neboť je zřejmé, že některé typy podnikání mají v určitých krajích lepší podmínky pro další rozvoj (pravděpodobně snadnější dodavatelsko-odběratelská struktura, přístup úřadů, …). Cílem práce nebylo analyzovat o jaké faktory se jedná v rámci vnějšího prostředí, neboť toto by vyžadovalo samostatnou studii. Cílem bylo naopak analyzovat kauzální závislosti, které jsou patrné z grafu významnosti jednotlivých prediktorů, které jsou zobrazeny v případě stromu na obrázku 79.

S ohledem na cíl práce je model základní stromové klasifikace hlavním výsledkem této práce. Model v Matlabu dokáže predikovat úspěšnost pohledu, a tedy i jeho zatřídění na základě (prediktorů) výše uvedených dat. Celkový strom je příliš rozsáhlý, aby bylo možné jej snadno písemně interpretovat. Aby však výstupem práce nebylo pouze teoretické konstatování o úspěšnosti modelu, uvádíme v příloze 6 zjednodušený model v tabulkové formě, která ukazuje, za jakých podmínek jsou podniky do jednotlivých kategorií řazené. Jednotlivé složky modelu, které mohou podniky ovlivnit, pak mohou být předmětem vnitropodnikových cílů. Při jejich realizaci pak podnik může dosahovat ekonomické přidané hodnoty s vyšší mírou pravděpodobnosti.

7 Přínos práce

7.1 Přínosy v teoretické oblasti

O novosti a reprezentativnosti v teoretické oblasti poznání způsobů kategorizace podniků na základě generování přidané hodnoty u MSP lze hovořit z pohledu rozsahu testovaného souboru (více jak 25 tis. malých a středních podniků v České republice za období 5 let). Podniky byly analyzovány z pohledu ekonomické přidané hodnoty ve variantě ekvity. Předmětem analýzy byl vztah mezi řadou prediktorů (15 položek vyjmenovaných v metodice) a kategorií podniku, která byla stanovena na základě dosažených výsledků. Z nově získaných výsledků je zřejmé, že uvedené prediktory ovlivňují prokazatelně výsledky podniků. Přínos pro rozvoj teoretického poznání v řešené problematice lze shrnout do následujících bodů:

- Návrh postupu a vlastní realizace rozsáhlé analýzy souboru dat malých a středních podniků v ČR s ohledem na jejich dosažené výsledky dle ekonomické přidané hodnoty s návrhem na formu interpretace výsledků (grafická i textová).
- Zmapování aktuálních poznatků o oblasti ekonomické přidané hodnoty.
- Konstrukce nového modelu pro predikci zatřídění podniků do příslušných kategorií na základě výsledků ekonomické přidané hodnoty.

7.2 Přínosy v pedagogické oblasti

Využití výstupů a poznatků získaných z řešení v rámci pedagogické oblasti:

- Koncepce nového předmětu „Strojové učení a neuronové sítě“ v rámci akreditace profesně zaměřeného (SP) Ekonomika podniku a to jak v oblasti přednášek, tak praktických cvičení (viz příloha – anotace předmětů).

- Dílčí výstupy z řešení byly (budou) uplatněny v předmětech:
  - Metodika odborné práce (příloha anotace předmětů),
  - Finance podniku I a II (příloha anotace předmětů)
  - a Controlling (příloha anotace předmětů).

- Výuka v rámci předmětů programů BBA a MBA.
- Zadávání témat bakalářských a diplomových prací.
- Zadávání témat v rámci SVOČ.

- Realizace kurzů a školení v rámci CŽV.
- Řešení smluvního výzkumu.
- Poradenská a konzultační činnost.

7.3 Přínosy pro podnikovou praxi

Praktický přínos realizovaného výzkumu:

- Konstrukce a ověření nového, snadno aplikovatelného modelu v podnikové praxi.
- Využití modelu pro tvorbu jednoduchého softwaru v Matlabu, Excelu nebo jiném programu pro klasifikaci MSP z hlediska generování přidané hodnoty.
- Využití modelu jako součásti benchmarkingového systému v podniku a jeho následné využití pro tvorbu vnitropodnikových cílů a jejich průběžnou inovaci dle dosahovaných výsledků a ekonomického cyklu.
- Využití modelu jako významného nástroje, resp. podkladu pro finanční řízení podniků.
- Výstupy z modelu mohou sloužit jako zdrojový informační materiál pro podnikový controlling a jeho budoucí směřování.
- Návrh základních rozhodovacích schémat nejprogresivnějšího modelu umožní managementu podniků posoudit, zdali je pro ně model smysluplný, a zdali je vhodné upravit některé podnikové cíle.
- Získané výstupy a poznatky budou uplatněny v rámci řešení následujících projektů TA ČR:
    - Projekt TREND „Optimalizace zakázkové kusové výroby v reálném čase využitím IoT a digitálních technologií, FW01010460, VUSTE-APIS, s.r.o. řešitel, VŠTE v Českých Budějovicích spoluřešitel, doba řešení 2020–2023.
    - Projekt TA ČR Éta „Stabilizace a rozvoj MSP ve venkovském prostoru“ TL01000349, Vysoká škola technická a ekonomická v Českých Budějovicích, oblast řešení finanční řízení MSP, doba řešení 2018–2021.
    - Projekt TA ČR Éta „Digitální transformace pro inovace obchodních modelů v malých a středních podnicích v České Republice“ TL02000215, VUT v Brně, Fakulta podnikatelská řešitel, VŠTE v Českých Budějovicích spoluřešitel, doba řešení 2019 – 2022

Výsledky budou konzultovány s HK pro Jihočeský kraj jako inovační nástroj řešení ekonomické udržitelnosti hospodaření MSP.

7.4 Limity studie

Limity výzkumu spočívají především v oblasti vstupních dat, viz kap. 5. Je zřejmé, že podkladová data jsou zatížená informačními šumy, které je nezbytné a možné redukovat, jak již bylo dokumentováno v předložené práci. I přes tuto skutečnost lze do určité míry algoritmy s danými chybami automaticky eliminovat. Za zásadní nedostatek v podobě vstupních dat lze spatřovat skutečnost, že řada malých a středních podniků nevyplňuje výkazy účetní závěrky. Zejména lze předpokládat, že se bude jednat o podniky s horšími ekonomickými výsledky.

Další limitou, kterou je nutné vzít v úvahu u malých a středních podniků, je daňová optimalizace, která se u menších podniků provede snadněji s větším dopadem na hospodářský výsledek. S největší pravděpodobností existuje řada podniků, které splňují cíle vlastníků i manažerů, ale z pohledu hospodářského výsledku se nachází pod úrovní bezrizikové sazby.

Limitace spočívá i v provedených analýzách, které sledují závislosti mezi prediktory a kategorií podniku. Neproběhla však další analýza spočívající v příčinném ověření výsledných závislostí. Toto je dáno zejména rozsahem, kdy výsledky pouze jedné analýzy mohou být předmětem zkoumání dalších několika podobně rozsáhlých výzkumných aktivit.

7.5 Koncepce směřování další vědecké činnosti

Další směřování výzkumných aktivit bude zaměřeno do třech základních oblastí, a to na odstranění výše uvedených limitů dosavadního výzkumu, na pokračování výzkumných aktivit při změně ekonomického cyklu a na dořešení výzkumných problematik s možností jejich přímé transformace do podnikové praxe.

První oblast – eliminace dosavadních výzkumných limitů spočívá v:

- dopracování metodiky sběru a přípravy vstupních dat pro příslušné analytické metody,
- návrhu forem automatické eliminace informačních šumů ve vstupních datech,
- zajištění maximální věrohodnosti vstupních údajů vytvořením stabilního testovacího souboru MSP vykazujícího kompletní finanční výkaznictví,
- eliminaci zkreslení ekonomických výsledků MSP v důsledku daňové optimalizace,
- návrhu postupu pro datamining pro odstraňování nedostatků v datové základně.

Druhá oblast – pokračování výzkumných aktivit za účelem prohloubení teoretických i praktických poznatků

- Potřebnost a aktuálnost řešené problematiky, jakou je generování přidané hodnoty včetně predikce vývoje bude narůstat zejména se změnou ekonomického cyklu. Zpracování a návrh obecně platných analýz, metod a modelů, které by umožňovaly její sledování, hodnocení a vykazování jak za určité období, tak v průběhu výrobního procesu či služeb. Lze předpokládat, že toto se stane v budoucnu jedním z nástrojů tvorby konkurenceschopnosti podniků a jeho udržitelnosti.
- Pokračování ve výzkumu uvedené problematiky s ohledem na potřebu získání výstupů z testování modelů s vyšší mírou citlivosti, relevantnosti a vypovídací schopnosti pro podnikovou sféru.

Třetí oblast – dořešení výzkumných problematik s možností jejich přímé transformace do podnikové praxe.

- V průběhu řešení autor práce navázal spolupráci s řadou podniků především z Jihočeského regionu, kdy ze strany podniků narůstá zájem o prakticky využitelné výstupy zejména v oblasti klasifikačních a regresních modelů využitelných při řízení podnikových procesů. Začátek výzkumu se předpokládá počátkem roku 2020.
- Mezi další problematiky, které budou v rámci spolupráce s podnikovou praxí řešeny, náleží:
  - projekce aplikovatelného modelu jako základu softwaru zaměřeného na generování přidané hodnoty,
  - navržení systému podnikového benchmarkingu pro tvorbu a kontrolu vnitropodnikových cílů,
  - tvorba manuálů pro využití výstupů z řešení pro oblast finančního řízení a podnikového controllingu.

8 Závěr

Předložená práce je zaměřena na problematiku, která stále silněji rezonuje v podnikové sféře. Možnost sledování, zejména pak hodnocení přidané hodnoty v rámci podniku, či v průběhu jednotlivých podnikových procesů se nesporně stane v horizontu několika let výraznou konkurenční výhodou a zdrojem ekonomické prosperity a udržitelnosti podniků. Toto konstatování má obecnou platnost, u malých a středních podniků, které tvoří základnu každé vyspělé národní ekonomiky, to platí dvojnásob. Ne jinak je tomu v České republice. Výzkumné zaměření práce reagovalo na podnikovou poptávku tohoto velikostního segmentu naší podnikové sféry, které se projevilo při řešení třech podpořených projektů v rámci TAČR cílených na MSP z hlediska jejich dalšího rozvoje. Zejména první podpořený projekt přímo obsahuje řešenou problematiku, je zacílen na měření tvorby přidané hodnoty v podnikových procesech ve strojírenství zaměřených na zakázkovou výrobu. Výstupy teoretické i metodologické prezentované v předložené práci budou přímo implementovány do metodiky řešení výše uvedeného projektu a bude tak umožněno praktické ověření získaných výstupů z habilitační práce v jednom sektoru MSP v ČR.

Při respektování zásad vědecké etiky a respektu ke stávajícímu poznání se lze domnívat, že práce přináší nový pohled a poznatky v oblasti sledování a hodnocení přidané hodnoty v podniku, na možnost predikce její tvorby jak za podnik jako celek, tak i za podnikové procesy a na postupy kategorizace podniků dle tvorby jejich přidané hodnoty. Nové poznání v této oblasti vychází ze souboru dat malých a středních podniků působících v České republice v letech 2013 až 2017. Testovací soubor zahrnoval více jak 25 tis. účetních závěrek malých a středních podniků a podle provedené rešerše je svým rozsahem výjimečný a reprezentativní. Data byla zpracována celou řadou metod strojového učení s učitelem i bez učitele. V průběhu zpracování dat byly jednotlivé metody podrobeny kritické diskuzi a docházelo k jejich optimalizaci za pomocí automatických i vlastních nástrojů. Veškeré modely byly vytvářeny na datech přibližně 17 tis. podniků a následně byly tyto modely testovány na kontrolní množině podniků, které nebyly zahrnuty do tvorby modelu. Pro výsledné posouzení kvality modelu byly vzaty v úvahu především výsledky na kontrolní množině podniků.

Součástí výstupů z řešení spadajících do teoretického i praktického poznání byla identifikace limit výzkumu v této oblasti, kdy mezi základní limity náleží kvalita vstupních dat, jejich zkreslování či nevykazování finančních výkazů včetně případné daňové optimalizace. Bez tohoto poznání by byl další výzkum zatížen nepřesnostmi a metodickými chybami.

Naznačeno je i dalších směřování navazujících výzkumných aktivit. Ty budou cíleny jak na zkvalitnění metodologie výzkumu včetně kvality vstupních údajů, tak i do teoretické oblasti a uživatelské sféry.

Při posouzení naplnění cílů práce lze uvést příklady z oblasti nového teoretického poznání, ale i výstupy, které mohou být přímo implementovány v podnikové praxi. Jako příklady lze uvést:

- Zmapování současného stavu stav poznání v oblasti vymezení a stanovení ekonomické přidané hodnoty prostřednictvím podrobné literární rešerše.
- Návrh nového modelu pro predikci zatřídění podniků do příslušných kategorií na základě výsledků ekonomické přidané hodnoty.
- Vtvoření modelu vhodného pro tvorbu jednoduchého softwaru pro testování MSP z hlediska generování jejich přidané hodnoty.
- Získání podkladů pro tvorbu manuálů pro MSP v oblasti finančního řízení, podnikového controllingu a benchmarkingového podnikového systému.

Mezi dílčí výstupy, které budou využity v dalších výzkumných aktivitách lze jako příklady uvést:

- Využití metody učení bez učitele nepředstavuje pro daný typ úlohy přílišné uplatnění.
- Jednotlivé testované modely dokáží segmentovat podniky, ale tato segmentace příliš nerozlišuje cíle sledované analýzou.
- Metoda učení s učitelem poskytovala relevantní výsledky, kdy jednotlivé modely řadily podniky do čtyř kategorií.
- Úspěšnost modelů se pohybovala mírně pod úrovní 50 %.
- Za relevantní modely bylo možné považovat pouze ty, které řadily podniky do všech kategorií, neboť některé kategorie nebyly početně vyrovnané.

Z hlediska souhrnného závěru pro podnikovou praxi výsledný model kategorizace podniků z pohledu běžného vlastníka malých a středních podniků předpokládá kladný výsledek hospodaření a musí být vyšší než bezrizikové zhodnocení jeho prostředků. V takovém případě nejúspěšnější model dokáže predikovat, že podnik dosáhne kladného výsledku hospodaření s více jak 80% pravděpodobností. Nejúspěšnější model je založen na jednoduchém rozhodovacím stromu, který je ve zjednodušené podobě přílohou této práce. Pomocí tohoto modelu může vzniknout program v Matlabu nebo v Excelu, který bude sloužit jako

bechmarking pro řízení podniků. Tento benchmarking samozřejmě může být dobrý vodítkem pro srovnání nebo stanovení nových podnikových cílů.

Je skutečností, že řada podnikatelských subjektů hledá zázračné recepty a metody pro zajištění svého růstu, stability a konkurenční výhody a přitom opomíjí, podceňují či neovládají metody sledování a hodnocení tvorby přidané hodnoty ve svém podniku. Zde musí sehrát jednu z rozhodujících rolí výzkumné organizace včetně výzkumných škol a novým poznáním v této oblasti přispět ke zvýšení odborné ekonomické gramotnosti u manažerů podniků.

Autor habilitační práce chce jejím předložením přispět svými výstupy a dosaženým poznáním k zahájení systémového řešení uvedené problematiky a k otevření vědecké diskuse k problematice, kterou lze považovat za jednu z limitujících při rozvoji malého a středního podnikání v České republice.

9 Zdroje

Abbod, M. F., 2007. Application of Artificial Intelligence to the Management of Urological Cancer. The Journal of Urology. 178 (4): 1150–1156. doi:10.1016/j.juro.2007.05.122. PMID 17698099

Abdi, H., Williams, L.J., 2010. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics. 2 (4): 433–459. arXiv:1108.4372. doi:10.1002/wics.101.

Altman, N. S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression (PDF). The American Statistician. 46 (3): 175–185.

Améndola, C., et al., 2015. Moment varieties of Gaussian mixtures. Journal of Algebraic Statistics. 7. arXiv:1510.04654. Bibcode:2015arXiv151004654A. doi:10.18409/jas.v7i1.42

Beyer, K., et al., 1999. When is nearest neighbor meaningful?. Database Theory—ICDT'99, 217-235

Boser, B. E., Guyon, I. M., Vapnik, V. N., 1992. A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory – COLT '92. p. 144. CiteSeerX 10.1.1.21.3818. doi:10.1145/130385.130401. ISBN 978-0897914970.

Bousquet, O., von Luxburg, U., Raetsch, G., eds., 2004. Advanced Lectures on Machine Learning. Springer-Verlag. ISBN 978-3540231226.

Brealey, R. A, S. C. Myers, F. Allen, 2013. Principles of corporate finance. 11th ed. New York: McGraw-Hill Irwin, p. cm. ISBN 00-780-3476-0.

Buhmann, J., Kuhnel, H., 1992. Unsupervised and supervised data clustering with competitive neural networks. [Proceedings 1992] IJCNN International Joint Conference on Neural Networks. 4. IEEE. pp. 796–801. doi:10.1109/ijcnn.1992.227220. ISBN 0780305590

Ciresan, D., Meier, U., Schmidhuber, J., 2012. Multi-column deep neural networks for image classification. IEEE Conference on Computer Vision and Pattern Recognition. pp. 3642–3649.

Coates, A., Ng, A. Y., 2012. Learning feature representations with k-means (PDF). In Montavon, G., Orr, G. B., Müller, K.-R. (eds.). Neural Networks: Tricks of the Trade. Springer.

Cortes, C., Vapnik, V. N., 1995. Support-vector networks (PDF). Machine Learning. 20 (3): 273–297. CiteSeerX 10.1.1.15.9362. doi:10.1007/BF00994018.

Cover T.M., Hart P.E. , 1967. Nearest neighbor pattern classification (PDF). IEEE Transactions on Information Theory. 13 (1): 21–27. CiteSeerX 10.1.1.68.2616.

Coomans D., D.L. Massart, 1982. Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-Nearest neighbour classification by using alternative voting rules. Analytica Chimica Acta. 136: 15–27. doi:10.1016/S0003-2670(01)95359-0

Freedman D. A. , 2009. Statistical Models: Theory and Practice. Cambridge University Press. ISBN 978-1-139-47731-4.

Demir, G. K., Ozmehmet, K., 2005. Online Local Learning Algorithms for Linear Discriminant Analysis. Pattern Recogn. Lett. 26 (4): 421–431. doi:10.1016/j.patrec.2004.08.005. ISSN 0167-8655.

Duda, R. O., Hart, P. E., Stork, D. G., 2001. Unsupervised Learning and Clustering. Pattern classification (2nd ed.). Wiley. ISBN 0-471-05669-3.

Fiala P., Karhan P., Ptáček J., 2014. Neuronové sítě a možnosti jejich využití. [cit. 2021-10-12] Dostupné online z: http://www.csfm.cz/userfiles/file/Udalosti_2014/KRF2014/fiala.pdf

Freund, Y. a R. E. Schapire, 1997. *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. J. of Computer and System Sciences, Vol. 55, pp. 119–139.

Ghosh and Reilly, Credit card fraud detection with a neural-network, 1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences, Wailea, HI, USA, 1994, pp. 621-630. doi: 10.1109/HICSS.1994.323314

Hall P, Park B.U., Samworth R.J., 2008. Choice of neighbor order in nearest-neighbor classification. Annals of Statistics. 36 (5): 2135–2152. arXiv:0810.5276. doi:10.1214/07-AOS537

Hamerly, G., Elkan, C., 2002. Alternatives to the k-means algorithm that find better clusterings (PDF). Proceedings of the eleventh international conference on Information and knowledge management (CIKM).

Hastie, T., Tibshirani, R., Friedman, J. H., 2009. The Elements of Statistical Learning (second ed.). Springer-Verlag. ISBN 978-0-387-84858-7. Archived from the original on 2009-11-10.

Haykin, S., 1999. 9. Self-organizing maps. Neural networks - A comprehensive foundation (2nd ed.). Prentice-Hall. ISBN 978-0-13-908385-3.

Hinton, G, Sejnowski, T. J., eds., 1999. Unsupervised Learning: Foundations of Neural Computation. MIT Press. ISBN 0-262-58168-X

Hotelling, H., 1933. Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24, 417–441

Christianini, N., and J. C. Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge, UK: Cambridge University Press, 2000.

Jiřina, M., 2008. Jak na neuronové sítě v programu STATISTICA Neuronové sítě. 2. vyd. Praha: StatSoft. ISBN 978-80-904033-1-4.

Karimi K., H.J. Hamilton, 2011, Generation and Interpretation of Temporal Decision Rules, International Journal of Computer Information Systems and Industrial Management Applications, Volume 3

Kaelbling, L. P., Littman, M. L., Moore, A. W., 1996. Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research. 4: 237–285. arXiv:cs/9605103. doi:10.1613/jair.301. Archived from the original on 2001-11-20.

Kamiński, B., Jakubczyk, M., Szufel, P., 2017. A framework for sensitivity analysis of decision trees. Central European Journal of Operations Research. 26 (1): 135–159. doi:10.1007/s10100-017-0479-6. PMC 5767274. PMID 29375266

Kaufman, L., Roussew, P. J., 1990. Finding Groups in Data - An Introduction to Cluster Analysis. A Wiley-Science Publication John Wiley & Sons.

Kislingerová, E. 2007. Manažerské finance (Managerial Finance). Second revised and expanded edition. Prague: C. H. Beck. ISBN 978-80-7179-903-0.

Klecka, W. R., 1980. Discriminant analysis. Quantitative Applications in the Social Sciences Series, No. 19. Thousand Oaks, CA: Sage Publications.

Klieštik, T., J. Vrbka a Z. Rowland, 2018. Bankruptcy prediction in Visegrad group countries using multiple discriminant analysis. Equilibrium-Quarterly Journal of Economics and Economic Policy, Torun, Polsko: Inst Economic Research-Poland, 2018, roč. 13, č. 3, s. 569-593. ISSN 2353-3293. doi:10.24136/eq.2018.028.

Kohonen, T., 1982. Self-Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics. 43 (1): 59–69. doi:10.1007/bf00337288.

Kohonen, T., 1989. Self-organizing and associative memory. (3rd ed.), Berlin: Springer-Verlag.

Kohonen, T., 2001. Self-Organizing Maps. Third, Extended Edition. Springer Series in Information Sciences vol. 30, Berlin, Germany: Springer-Verlag, ISBN 978-3-540-67921-9

Kriegel, H-P., Schubert, E., Zimek, A., 2016. The (black) art of runtime evaluation: Are we comparing algorithms or implementations?. Knowledge and Information Systems. 52 (2): 341–378. doi:10.1007/s10115-016-1004-2. ISSN 0219-1377.

Smith M.R., Martinez, T,. 2011. Improving Classification Accuracy by Identifying and Removing Instances that Should Be Misclassified. Proceedings of International Joint Conference on Neural Networks (IJCNN 2011). pp. 2690–2697. CiteSeerX 10.1.1.221.1371. doi:10.1109/IJCNN.2011.6033571

MacQueen, J. B., 1967. Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1. University of California Press. pp. 281–297.

Malý, M., 2007. Vícevrstvé dopředné neuronové sítě: úvod do teorie a aplikací. Ústí nad Labem: Univerzita J.E. Purkyně, Přírodovědecká fakulta, 2007. ISBN 978-80-7044-915-8.

Mařík, M, 2011. Metody oceňování podniku: proces ocenění - základní metody a postupy. 3., upr. a rozš. vyd. Praha: Ekopress, 2011. ISBN 978-80-86929-67-5.

Matlab documentation [online]. USA: The MathWorks, 2021 [cit. 2021-10-12]. Dostupné z: https://www.mathworks.com/help/matlab/index.html

Matlab-pdist. MathWorks [online]. Uniet States: The MathWorks, ©1994-2021 [cit. 2021-08-08]. Dostupné z: https://www.mathworks.com/help/stats/pdist.html

Matlabacademy: Machine Learning with MATLAB [online]. USA: The MathWorks, 2021 [cit. 2021-10-12]. Dostupné z: https://matlabacademy.mathworks.com

McCulloch, W., W. Pitts, 1943. A Logical Calculus of Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics. 5 (4): 115–133. doi:10.1007/BF02478259

McLachlan, G. J., 2004. Discriminant Analysis and Statistical Pattern Recognition. Wiley Interscience. ISBN 978-0-471-69115-0. MR 1190469

Mead, A., 1992. *Review of the Development of Multidimensional Scaling Methods*. Journal of the Royal Statistical Society. Series D (The Statistician). 41 (1): 27–39. JSTOR 234863

Mehryar M., A., Rostamizadeh, A., Talwalkar, 2012) Foundations of Machine Learning, The MIT Press ISBN 9780262018258.

Miljanovic, M., 2012. Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction (PDF). Indian Journal of Computer and Engineering. 3 (1).

Ministerstvo průmyslu a obchodu 2021. [online]. [cit. 2021-10-12]. Available from WWW: https://www.mpo.cz/cz/rozcestnik/analyticke-materialy-a-statistiky/analyticke-materialy/

Mogull, R. G., 2004. Second-Semester Applied Statistics. Kendall/Hunt Publishing Company. p. 59. ISBN 978-0-7575-1181-3.

Moustafa, R., Wegman, E., J., 2002. On Some Generalizations of Parallel Coordinate Plots (PDF). Seeing a Million, A Data Visualization Workshop, Rain Am Lech (nr.), Germany. Archived from the original (PDF) on 2013-12-24.

Neumaierová, I., 1998. Řízení hodnoty (Value Management). Prague: University of Economics in Prague. Faculty of Business Administration, 1998. 137 pp ISBN 80-7079-921-8.

Neumaierová, I., 2003. Aplikace řízení hodnoty (Value Management Application). Prague: University of Economics in Prague. Faculty of Business Administration. 95 pp ISBN 80-245-0536-3.

Neumaierová, I., Neumaier, I., 2006. Proč se ujal index IN a nikoli pyramidový systém ukazatelů INFA [online]. [cit. 2021-10-12].

Ojha, V. K., Abraham, A., Snášel, V., 2017. Metaheuristic design of feedforward neural networks: A review of two decades of research. Engineering Applications of Artificial Intelligence. 60: 97–116. arXiv:1705.05584. Bibcode:2017arXiv170505584O. doi:10.1016/j.engappai.2017.01.013.

Pearson, K., 1901. *On Lines and Planes of Closest Fit to Systems of Points in Space*. Philosophical Magazine. 2 (11): 559–572. doi:10.1080/14786440109462720.

Pelleg, D., Moore, A., 1999. Accelerating exact k -means algorithms with geometric reasoning. Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '99. San Diego, California, United States: ACM Press: 277–281. doi:10.1145/312129.312248. ISBN 9781581131437.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., 2007. Section 16.1. Gaussian Mixture Models and k-Means Clustering. Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York (NY): Cambridge University Press. ISBN 978-0-521-88068-8.

Quinlan, J. R., 1987. Simplifying decision trees. International Journal of Man-Machine Studies. 27 (3): 221–234. CiteSeerX 10.1.1.18.4267. doi:10.1016/S0020-7373(87)80053-6.

Simkanič, R., 2016, Deep Learning v analýze obrazu, VŠB – Technická univerzita Ostrava Fakulta elektrotechniky a informatiky

Rao, C. R., Toutenburg, H., et al., 2008. Linear Models: Least Squares and Alternatives. Springer Series in Statistics (3rd ed.). Berlin: Springer. ISBN 978-3-540-74226-5.

Rennie, J., Shih, L., Teevan, J., Karger, D., 2003. Tackling the poor assumptions of Naive Bayes classifiers

Rish, I., 2001. An empirical study of the naive Bayes classifier (PDF). IJCAI Workshop on Empirical Methods in AI.

Rokach, L., Maimon O., 2005. Clustering methods. Data mining and knowledge discovery handbook. Springer US.

Roll, R., 1977. A Critique of the Asset Pricing Theory's Tests. Journal of Financial Economics. 4 (2): 129–176. doi:10.1016/0304-405X(77)90009-5

Balabin R. M. a Lomakina E. I., 2009. Neural network approach to quantum-chemistry data: Accurate prediction of density functional theory energies. J. Chem. Phys. 131 (7): 074104. Bibcode:2009JChPh.131g4104B. doi:10.1063/1.3206326. PMID 19708729.

Roman, V., 2019. Unsupervised Machine Learning: Clustering Analysis. Medium. Retrieved 2019-10-01.

Ronald A. Fisher, 1954. Statistical Methods for Research Workers (Twelfth ed.). Edinburgh: Oliver and Boyd. ISBN 978-0-05-002170-5.

Russell S. J. a P. Norvig, 2010. Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall ISBN 9780136042594.

Sak, H., S., Andrew, B., Francoise , 2014. Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling (PDF). Archived from the original (PDF) on 24 April 2018.

SAS Institute, 2019. The DISTANCE Procedure: Proximity Measures. SAS/STAT 9.2 Users Guide.. Retrieved 2009-04-26. dostupné online: https://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_distance_sect016.htm

Seber, G. A. F, 1984. *Multivariate Observations*. Hoboken, NJ: John Wiley & Sons, Inc.

Seiffert, C., T. Khoshgoftaar, J. Hulse, and A. Napolitano, 2008. *RUSBoost: Improving clasification performance when training data is skewed*. 19th International Conference on Pattern Recognition, pp. 1–4.

Shaw P.J.A., 2003 Multivariate statistics for the Environmental Sciences, Hodder-Arnold. ISBN 0-340-80763-6.

Soummer, R., Pueyo, L., Larkin, J., 2012. Detection and Characterization of Exoplanets and Disks Using Projections on Karhunen-Loève Eigenimages. The Astrophysical Journal Letters. 755 (2): L28. arXiv:1207.4197. Bibcode:2012ApJ...755L..28S. doi:10.1088/2041-8205/755/2/L28

Stehel, V., J. Horak A T. Krulicky, 2021. Business performance assessment of small and medium-sized enterprises: Evidence from the Czech Republic. Problems and Perspectives in Management, 2021, roč. 19, č. 3, s. 430-439. ISSN 17277051

Stehel, V. a M. Vochozka 2014. Company Management by Using INFA Method. In 12th International Academic Conference : sborník příspěvků. 1. vyd. Praha: International Institute of Social and Economic Sciences (IISES), 2014. s. 1132-1144, 13 s. ISBN 978-80-87927-04-5.

Székely, G. J., Rizzo, M. L., 2005. Hierarchical clustering via Joint Between-Within Distances: Extending Ward's Minimum Variance Method. Journal of Classification. 22 (2): 151–183. doi:10.1007/s00357-005-0012-9.

Šíma, J. a Neruda, J. 1996. Teoretické otázky neuronových sítí. Matfyzpress, Praha.

Šnorek M., Jiřina M., 1998. Neuronové sítě a neuropočítače. ČVUT, Praha, ISBN 80–01–01455–X.

Šťastný, P., 2014. Rozpoznávání objektů pomocí neuronových sítí [online]. Brno, 2014 [cit. 2020-01-05]. Dostupné z: <https://is.muni.cz/th/rr91b/>. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Jiří Hřebíček.

Titterington, D., Smith, A., Makov, U., 1985. Statistical Analysis of Finite Mixture Distributions. Wiley. ISBN 978-0-471-90763-3.

Utgoff, P. E., 1989. Incremental induction of decision trees. Machine learning, 4(2), 161-186. doi:10.1023/A:1022699900025

Vochozka, M. a J. Horák, 2019. Comparison of neural networks and regression time series when estimating the copper price development. In Ashmarina, S., Vochozka, M.. Sustainable Growth and Development of Economic Systems: Contradictions in the Era of Digitalization and Globalization, In: Contributions to Economics. Cham, Switzerland: Springer. s. 169-181, 13 s. ISBN 978-3-030-11753-5.

Vochozka, M., 2011. Metody komplexního hodnocení podniku. 1. vyd. Praha: Grada Publishing, 2011. 246 s. Finanční řízení. ISBN 978-80-247-3647-1.

Vojáček, A., 2006. Samoučící se neuronová síť - SOM, Kohonenovy mapy [online]. [cit. 2021-10-12]. Dostupné z: https://www.kiv.zcu.cz/studies/predmety/uir/NS/Samouc_NN2.pdf

Volná, E., 2008 [online]. *Neuronové sítě 1*. Ostravská univerzita v Ostravě, [cit. 2021-10-12] Dostupné z: http://www1.osu.cz/~volna/Neuronove_site_skripta.pdf

Vondrák, I., 2000 Umělá inteligence a neuronové sítě. 2. vyd. Ostrava: Vysoká škola báňská - Technická univerzita. ISBN 80-7078-949-2.

Warmuth, M., J. Liao, a G. Ratsch, 2006. *Totally corrective boosting algorithms that maximize the margin*. Proc. 23rd Int'l. Conf. on Machine Learning, ACM, New York, pp. 1001–1008.

Widrow, B., et al. , 2013. The no-prop algorithm: A new learning algorithm for multilayer neural networks. Neural Networks. 37: 182–188. doi:10.1016/j.neunet.2012.09.020. PMID 23140797

Wöhe, G. 1995. Úvod do podnikového hospodářství. Praha C. H. Beck. ISBN 80-7179-014-1

Zha, H., Ding, C., Gu, M., He, X., Simon, H. D., 2001. Spectral Relaxation for k-means Clustering (PDF). Neural Information Processing Systems Vol.14 (NIPS 2001): 1057–1064.

Zissis, D., 2015. A cloud based architecture capable of perceiving and predicting multiple vessel behaviour. Applied Soft Computing. 35: 652–661. doi:10.1016/j.asoc.2015.07.002

10 Seznam zkratek a symbolů

CAPM	Capital Asset Pricing Model
DA	diskriminační analýza
EVA	Economic Value Added
HV	Hospodářský výsledek
KNN	k-nearest neighbors
MSVM	Multi Support Vector Machines
NB	Naive Bayes klasifikace
PCA	Principal coponent analytsis
ROC	Receiver Operating Characteristic
SVM	Support Vector Machines

11 Seznam obrázků

12 Seznam tabulek

13 Seznam příloh

Příloha 1 – Optimalizace hyperparametrů stromové klasifikace

|=====|=======|=========|=========|==========|=========|=============|

| Iter | Eval | Objective | Objective | BestSoFar | BestSoFar | MinLeafSize |

| | result | | runtime | (observed) | (estim.) | |

|=====|=======|=========|=========|==========|=========|=============|

| 1 | Best | 0.49587 | 1.2547 | 0.49587 | 0.49587 | 666 |

| 2 | Accept | 0.54464 | 4.4913 | 0.49587 | 0.49881 | 1 |

| 3 | Accept | 0.52035 | 2.0829 | 0.49587 | 0.49588 | 17 |

| 4 | Accept | 0.55108 | 0.25483 | 0.49587 | 0.49869 | 9072 |

| 5 | Best | 0.49428 | 0.81652 | 0.49428 | 0.49432 | 700 |

| 6 | Accept | 0.49768 | 0.64904 | 0.49428 | 0.49434 | 1153 |

| 7 | Accept | 0.49515 | 0.73782 | 0.49428 | 0.49502 | 868 |

| 8 | Best | 0.49001 | 1.1247 | 0.49001 | 0.49009 | 176 |

| 9 | Accept | 0.49037 | 1.2776 | 0.49001 | 0.49004 | 104 |

| 10 | Best | 0.4899 | 1.2076 | 0.4899 | 0.48977 | 141 |

| 11 | Accept | 0.49062 | 1.1934 | 0.4899 | 0.49004 | 140 |

| 12 | Accept | 0.4899 | 1.1969 | 0.4899 | 0.49001 | 141 |

| 13 | Accept | 0.49062 | 1.2568 | 0.4899 | 0.49014 | 140 |

| 14 | Accept | 0.492 | 1.0987 | 0.4899 | 0.49015 | 206 |

| 15 | Best | 0.48936 | 1.7578 | 0.48936 | 0.48996 | 126 |

| 16 | Accept | 0.52259 | 0.50201 | 0.48936 | 0.48997 | 3679 |

| 17 | Accept | 0.54276 | 3.9138 | 0.48936 | 0.48998 | 4 |

| 18 | Accept | 0.58157 | 0.17322 | 0.48936 | 0.49003 | 13803 |

| 19 | Accept | 0.4945 | 1.5354 | 0.48936 | 0.49003 | 63 |

| 20 | Accept | 0.48986 | 1.3086 | 0.48936 | 0.48994 | 122 |

|=====|

| Iter | Eval | Objective | Objective | BestSoFar | BestSoFar | MinLeafSize |

| | result | | runtime | (observed) | (estim.) | |

|=====|

| 21 | Accept | 0.50601 | 0.57273 | 0.48936 | 0.48994 | 2113 |

| 22 | Accept | 0.50945 | 2.2442 | 0.48936 | 0.48994 | 33 |

| 23 | Accept | 0.49138 | 1.3053 | 0.48936 | 0.49012 | 117 |

| 24 | Accept | 0.53331 | 2.9516 | 0.48936 | 0.49013 | 8 |

| 25 | Accept | 0.54308 | 4.2116 | 0.48936 | 0.49015 | 2 |

| 26 | Accept | 0.49457 | 1.0123 | 0.48936 | 0.49014 | 355 |

| 27 | Accept | 0.55108 | 0.2824 | 0.48936 | 0.49033 | 5674 |

| 28 | Accept | 0.49222 | 1.3046 | 0.48936 | 0.49032 | 88 |

| 29 | Accept | 0.49099 | 1.0434 | 0.48936 | 0.49028 | 258 |

| 30 | Accept | 0.48936 | 1.1633 | 0.48936 | 0.4902 | 167 |

_____

Optimization completed.

MaxObjectiveEvaluations of 30 reached.

Total function evaluations: 30

Total elapsed time: 67.7621 seconds.

Total objective function evaluation time: 43.9252

Best observed feasible point:

MinLeafSize
126

Observed objective function value = 0.48936

Estimated objective function value = 0.4902

Function evaluation time = 1.7578

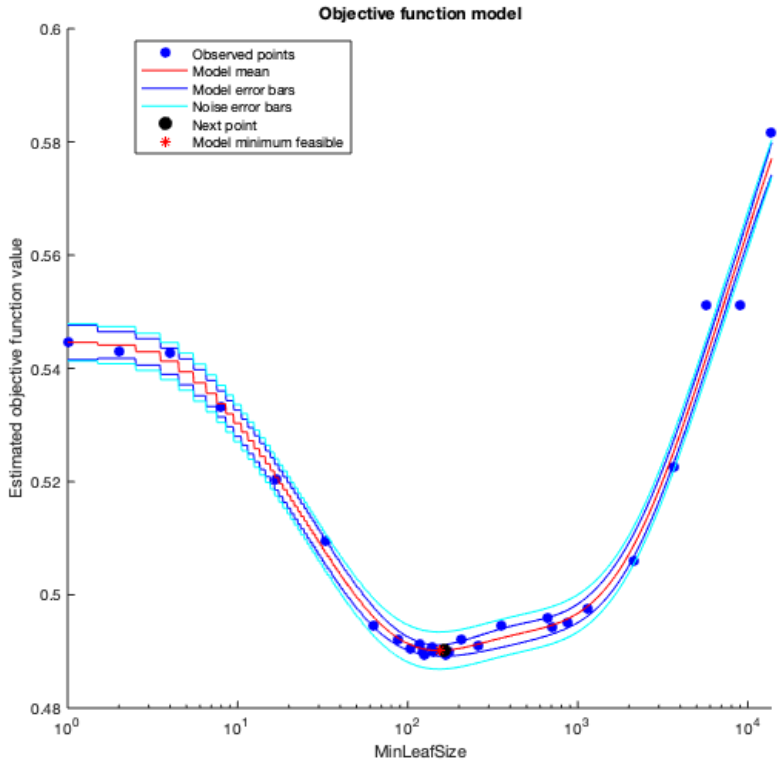
Best estimated feasible point (according to models):

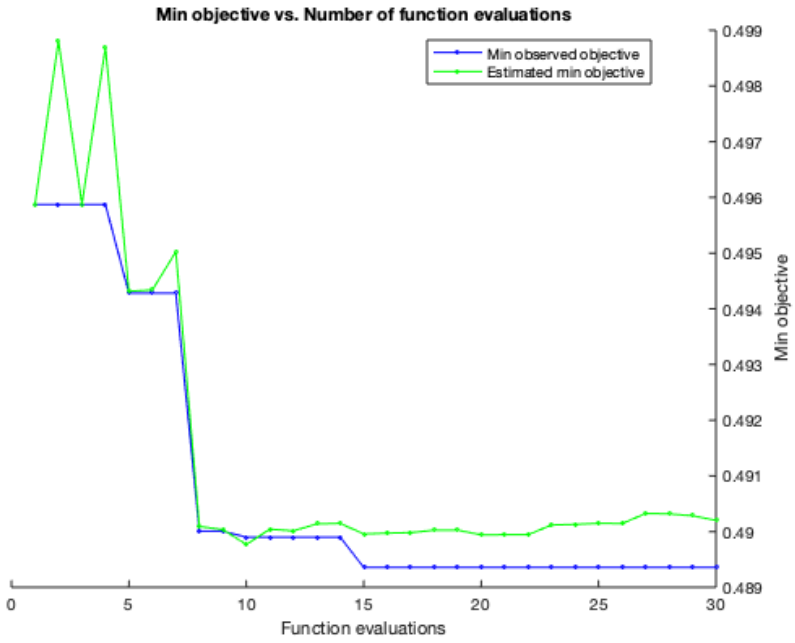
MinLeafSize

141

Estimated objective function value = 0.4902

Estimated function evaluation time = 1.2624




Min objective vs. Number of function evaluations
Min observed objective
Estimated min objective
Min objective
Function evaluations
0.499
0.498
0.497
0.496
0.495
0.494
0.493
0.492
0.491
0.49
0.489
0
5
10
15
20
25
30

Příloha 2 – Optimalizace hyperparametrů Naive Bayes

|=====|=======|=========|=========|=========|=========|=========|=========|

| Iter | Eval | Objective | Objective | BestSoFar | BestSoFar | Distribution-| Width |

| | result | | runtime | (observed) | (estim.) | Names | |

|=====|=======|=========|=========|=========|=========|=========|=========|

| 1 | Best | 0.55653 | 70.138 | 0.55653 | 0.55653 | kernel | 1.8458 |

| 2 | Accept | 0.56335 | 2.8875 | 0.55653 | 0.55708 | kernel | 7.4042e-05 |

| 3 | Accept | 0.65923 | 0.30221 | 0.55653 | 0.55673 | normal | - |

| 4 | Accept | 0.56414 | 2.8646 | 0.55653 | 0.55663 | kernel | 0.0001024 |

| 5 | Best | 0.54121 | 69.161 | 0.54121 | 0.54124 | kernel | 123.88 |

| 6 | Accept | 0.57258 | 2.2465 | 0.54121 | 0.54594 | kernel | 1.6011e-06 |

| 7 | Accept | 0.54121 | 59.739 | 0.54121 | 0.5412 | kernel | 122.22 |

| 8 | Accept | 0.54121 | 59.831 | 0.54121 | 0.5412 | kernel | 123.9 |

| 9 | Accept | 0.54121 | 60.83 | 0.54121 | 0.5412 | kernel | 123.69 |

| 10 | Accept | 0.54121 | 64.149 | 0.54121 | 0.5412 | kernel | 123.85 |

| 11 | Accept | 0.55285 | 10.223 | 0.54121 | 0.5412 | kernel | 0.0063435 |

| 12 | Best | 0.54085 | 74.082 | 0.54085 | 0.54119 | kernel | 55.359 |

| 13 | Accept | 0.54091 | 64.643 | 0.54085 | 0.54082 | kernel | 73.54 |

| 14 | Accept | 0.54091 | 62.936 | 0.54085 | 0.54085 | kernel | 73.104 |

| 15 | Accept | 0.54091 | 66.242 | 0.54085 | 0.54086 | kernel | 73.264 |

| 16 | Accept | 0.56353 | 2.2227 | 0.54085 | 0.54086 | kernel | 7.7582e-06 |

| 17 | Accept | 0.55834 | 4.5039 | 0.54085 | 0.54086 | kernel | 0.0010595 |

| 18 | Accept | 0.56613 | 28.13 | 0.54085 | 0.54088 | kernel | 0.046977 |

| 19 | Accept | 0.54133 | 68.553 | 0.54085 | 0.54089 | kernel | 27.759 |

| 20 | Accept | 0.54085 | 67.544 | 0.54085 | 0.54085 | kernel | 51.242 |

|=====|

| Iter | Eval | Objective | Objective | BestSoFar | BestSoFar | Distribution-| Width |

| | result | | runtime | (observed) | (estim.) | Names | |

|=====|

| 21 | Accept | 0.54085 | 66.781 | 0.54085 | 0.54085 | kernel | 52.294 |

| 22 | Accept | 0.5607 | 2.6736 | 0.54085 | 0.54085 | kernel | 2.0493e-05 |

| 23 | Accept | 0.5543 | 5.7856 | 0.54085 | 0.54085 | kernel | 0.0026134 |

| 24 | Accept | 0.59231 | 46.837 | 0.54085 | 0.54086 | kernel | 0.27361 |

| 25 | Accept | 0.54199 | 65.157 | 0.54085 | 0.54086 | kernel | 8.3405 |

| 26 | Accept | 0.54187 | 69.384 | 0.54085 | 0.54085 | kernel | 15.114 |

| 27 | Accept | 0.54085 | 59.587 | 0.54085 | 0.54085 | kernel | 50.391 |

| 28 | Accept | 0.56558 | 3.6688 | 0.54085 | 0.54085 | kernel | 0.0003455 |

| 29 | Accept | 0.56227 | 1.9657 | 0.54085 | 0.54085 | kernel | 3.2581e-06 |

| 30 | Accept | 0.54115 | 59.753 | 0.54085 | 0.54086 | kernel | 43.482 |

_____

Optimization completed.

MaxObjectiveEvaluations of 30 reached.

Total function evaluations: 30

Total elapsed time: 1260.1798 seconds.

Total objective function evaluation time: 1222.8222

Best observed feasible point:

DistributionNames	Width
kernel	55.359

Observed objective function value = 0.54085

Estimated objective function value = 0.54086

Function evaluation time = 74.0821

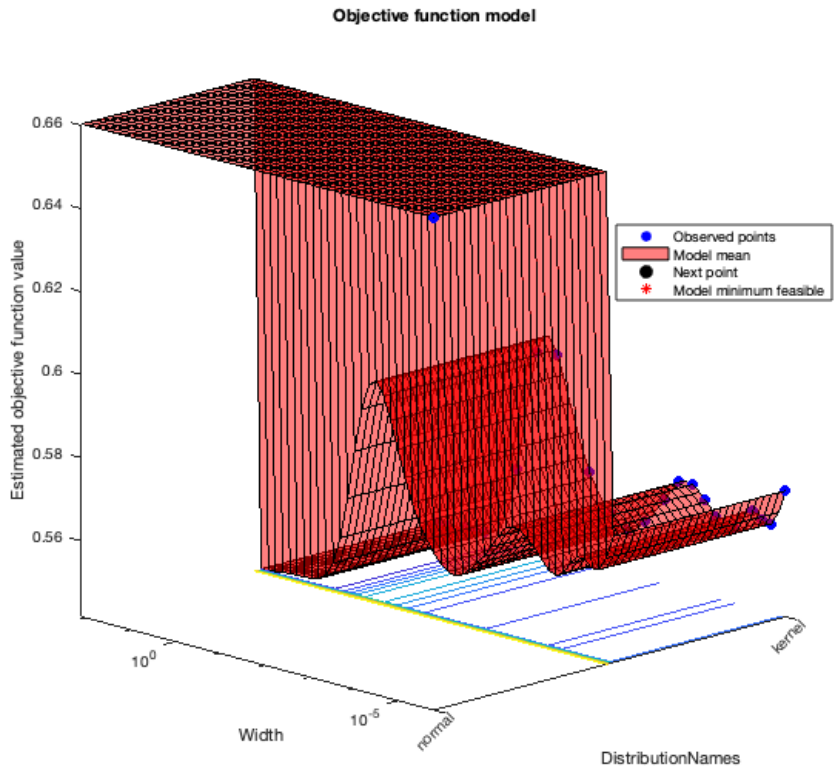
Best estimated feasible point (according to models):

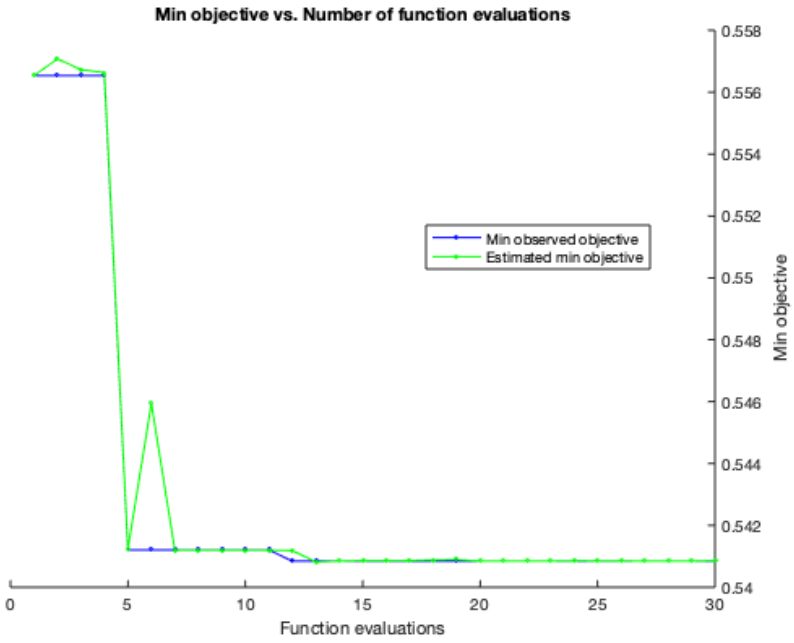
DistributionNames	Width

kernel 55.359

Estimated objective function value = 0.54086

Estimated function evaluation time = 65.2999




Min objective vs. Number of function evaluations
Min observed objective
Estimated min objective
Min objective
Function evaluations
0.558
0.556
0.554
0.552
0.55
0.548
0.546
0.544
0.542
0.54
0
5
10
15
20
25
30

Příloha 3 – Optimalizace hyperparametrů Discriminant analýzy

|===
==================================|

| Iter | Eval | Objective | Objective | BestSoFar | BestSoFar | Delta | Gamma |

| | result | | runtime | (observed) | (estim.) | | |

|===
==================================|

| 1 | Best | 0.56697 | 2.3392 | 0.56697 | 0.56697 | 0.00023248 | 0.62811 |

| 2 | Accept | 0.58157 | 0.38869 | 0.56697 | 0.56778 | 819.54 | 0.068357 |

| 3 | Accept | 0.58157 | 0.27475 | 0.56697 | 0.56697 | 155.34 | 0.98755 |

| 4 | Accept | 0.5823 | 0.25004 | 0.56697 | 0.56697 | 0.0063927 | 0.012775 |

| 5 | Best | 0.5575 | 0.41814 | 0.5575 | 0.5575 | 1.2556e-05 | 0.46867 |

| 6 | Best | 0.55358 | 0.22189 | 0.55358 | 0.55358 | 0.00013674 | 0.40189 |

| 7 | Best | 0.55267 | 0.29824 | 0.55267 | 0.55268 | 0.0018118 | 0.35008 |

| 8 | Best | 0.55231 | 0.18784 | 0.55231 | 0.55238 | 1.2768e-05 | 0.3647 |

| 9 | Best | 0.55225 | 0.25748 | 0.55225 | 0.55228 | 1.3858e-06 | 0.36028 |

| 10 | Accept | 0.55225 | 0.27579 | 0.55225 | 0.55227 | 1.4074e-06 | 0.36049 |

| 11 | Best | 0.55219 | 0.16698 | 0.55219 | 0.55224 | 1.4005e-06 | 0.36186 |

| 12 | Accept | 0.55219 | 0.24748 | 0.55219 | 0.55223 | 1.1502e-06 | 0.365 |

| 13 | Accept | 0.58031 | 0.29088 | 0.55219 | 0.55223 | 1.0647e-06 | 0.85619 |

| 14 | Accept | 0.5724 | 0.27052 | 0.55219 | 0.55222 | 1.0339e-06 | 0.22611 |

| 15 | Accept | 0.58157 | 0.20862 | 0.55219 | 0.55221 | 19.813 | 0.36806 |

| 16 | Best | 0.55219 | 0.31148 | 0.55219 | 0.55221 | 0.00023533 | 0.35211 |

| 17 | Accept | 0.55279 | 0.25127 | 0.55219 | 0.55221 | 1.4018e-06 | 0.38822 |

| 18 | Accept | 0.55219 | 0.19868 | 0.55219 | 0.55222 | 0.00045451 | 0.36444 |

| 19 | Accept | 0.55225 | 0.30734 | 0.55219 | 0.55222 | 2.9112e-06 | 0.36774 |

| 20 | Accept | 0.55231 | 0.20588 | 0.55219 | 0.55222 | 0.00014907 | 0.36108 |

|=====|

| Iter | Eval | Objective | Objective | BestSoFar | BestSoFar | Delta | Gamma |

| | result | | runtime | (observed) | (estim.) | | |

|=====|

| 21 | Accept | 0.55231 | 0.29808 | 0.55219 | 0.55222 | 0.00051279 | 0.35264 |

| 22 | Best | 0.55183 | 0.20807 | 0.55183 | 0.55193 | 6.3027e-06 | 0.35398 |

| 23 | Accept | 0.55255 | 0.264 | 0.55183 | 0.55205 | 1.1538e-05 | 0.34133 |

| 24 | Best | 0.55153 | 0.17611 | 0.55153 | 0.55187 | 4.2572e-06 | 0.35633 |

| 25 | Accept | 0.55165 | 0.21811 | 0.55153 | 0.55179 | 4.2084e-06 | 0.35686 |

| 26 | Accept | 0.55189 | 0.17036 | 0.55153 | 0.55181 | 3.8681e-06 | 0.35325 |

| 27 | Accept | 0.55418 | 0.18584 | 0.55153 | 0.55181 | 0.0039843 | 0.40925 |

| 28 | Accept | 0.59099 | 0.17139 | 0.55153 | 0.55182 | 0.00093599 | 0.99998 |

| 29 | Accept | 0.58194 | 0.44059 | 0.55153 | 0.55182 | 1.0039e-06 | 0.0019131 |

| 30 | Accept | 0.59093 | 0.22996 | 0.55153 | 0.55182 | 1.0086e-06 | 0.99918 |

_____

Optimization completed.

MaxObjectiveEvaluations of 30 reached.

Total function evaluations: 30

Total elapsed time: 49.1446 seconds.

Total objective function evaluation time: 9.7337

Best observed feasible point:

Delta	Gamma
4.2572e-06	0.35633

Observed objective function value = 0.55153

Estimated objective function value = 0.55182

Function evaluation time = 0.17611

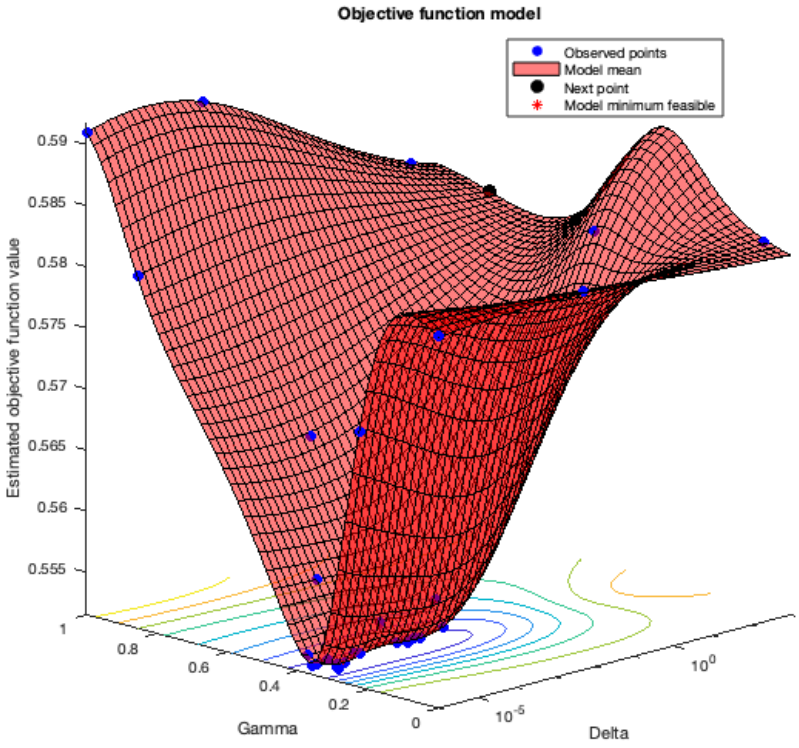
Best estimated feasible point (according to models):

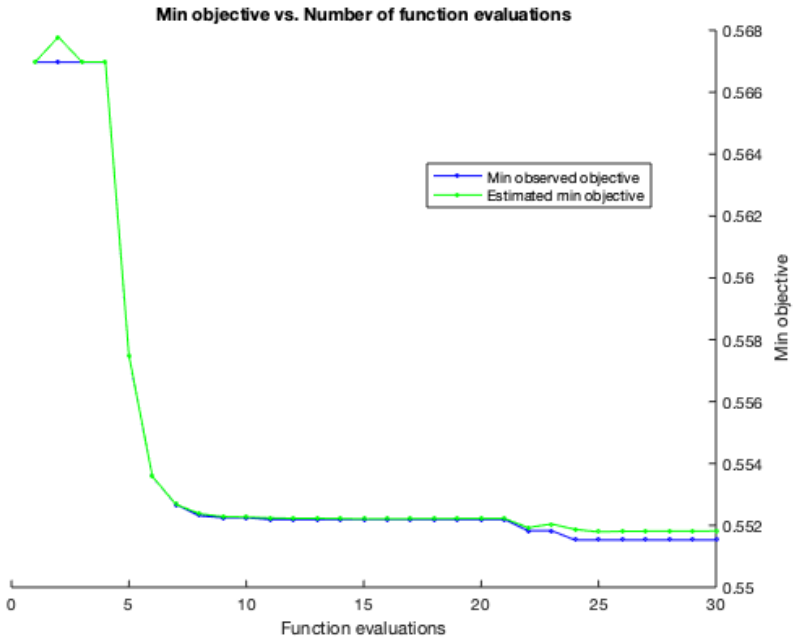
Delta	Gamma

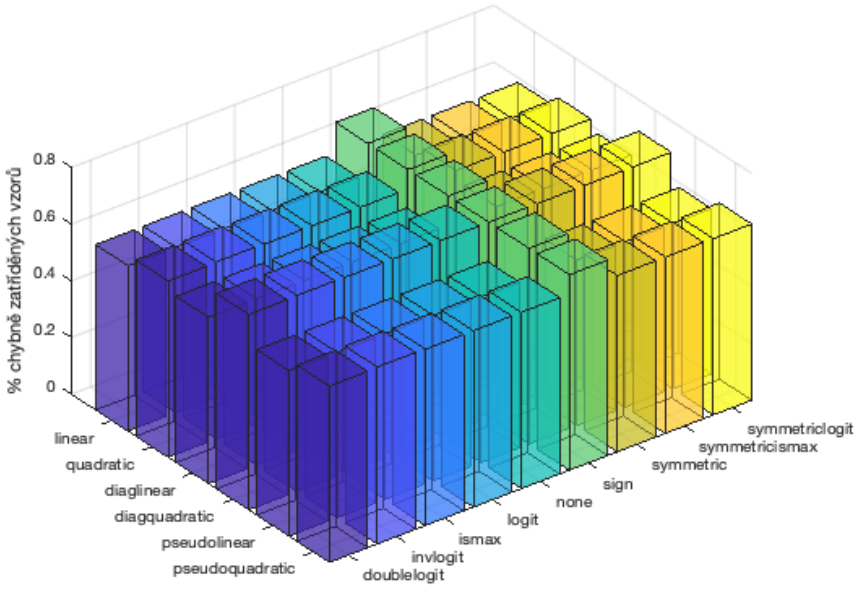
4.2084e-06 0.35686

Estimated objective function value = 0.55182

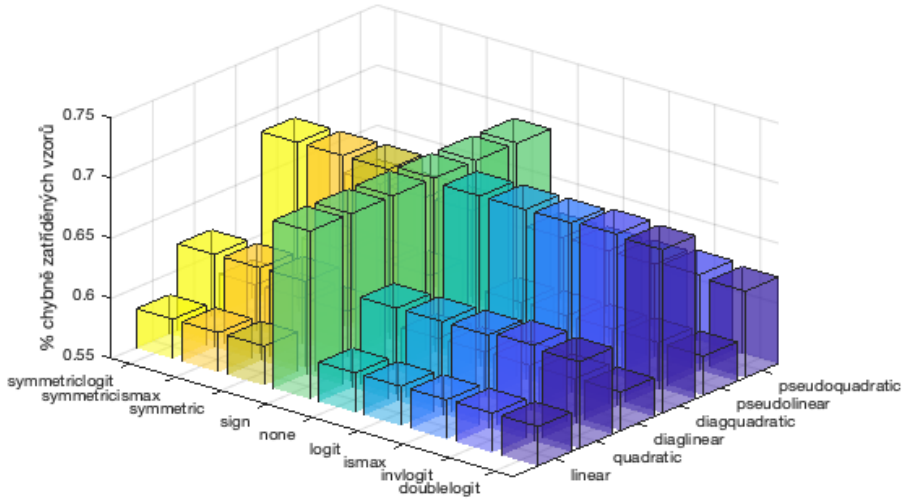
Estimated function evaluation time = 0.23061







0,58	0,58	0,58	0,58	0,58	0,69	0,58	0,58	0,58
0,62	0,62	0,62	0,62	0,62	0,69	0,62	0,62	0,62
0,59	0,59	0,59	0,59	0,59	0,69	0,59	0,59	0,59
0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69
0,58	0,58	0,58	0,58	0,58	0,69	0,58	0,58	0,58
0,62	0,62	0,62	0,62	0,62	0,69	0,62	0,62	0,62



0,58	0,58	0,58	0,58	0,58	0,69	0,58	0,58	0,58
0,62	0,62	0,62	0,62	0,62	0,69	0,62	0,62	0,62
0,58	0,58	0,58	0,58	0,58	0,69	0,58	0,58	0,58
0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,69
0,58	0,58	0,58	0,58	0,58	0,69	0,58	0,58	0,58
0,62	0,62	0,62	0,62	0,62	0,69	0,62	0,62	0,62

Příloha 5 – Kohonenova síť o 100 buňkách

Buňka Kohonenovy sítě	1	2	3	4	Celkový součet
1	1112	795	129	1174	3210
2	219	287	51	130	687
3	79	113	24	55	271
4	85	138	19	30	272
5	35	43	7	26	111
6	22	34	5	16	77
7	6	16	4	10	36
8		2	1	1	4
9	3	4	1	5	13
10	18	17	3	6	44
11	126	178	41	112	457
12	252	269	39	144	704
13	78	113	14	72	277
14	20	61	14	25	120
15	37	56	7	14	114
16	27	39	3	11	80
17	12	25	3	6	46
18	29	62	3	3	97
19	12	11	3	5	31
20	1	8	1		10
21	771	657	128	467	2023
22	85	118	12	43	258
23	184	140	13	48	385
24	99	107	12	28	246
25	52	63	9	27	151
26	5	14	4	29	52
27	11	54	12	12	89
28	5	11	3	6	25
29	4	17	3	9	33
30	6	26	1	4	37
31	208	239	37	144	628
32	131	138	17	50	336
33	53	50	2	13	118
34	20	26	2	2	50
35	29	43	5	8	85
36	15	40		4	59
37	11	37	3	9	60
38	5	23	4	3	35
39	12	25	2	8	47
40	11	26	1	6	44
41	89	134	18	61	302

42	92	139	22	70	323
43	50	103	15	43	211
44	64	72	12	21	169
45	53	106	12	14	185
46	19	18	2	3	42
47	9	34		2	45
48	18	16	3	10	47
49	2	3	1	5	11
50	9	14	1	6	30
51	19	71	25	66	181
52	40	95	9	14	158
53	40	78	14	27	159
54	29	32	3	10	74
55	27	49	6	25	107
56	24	52	2	10	88
57	14	43	3	3	63
58	9	20	2	4	35
59	7	15		1	23
60	5	24	1	5	35
61	47	74	13	23	157
62	46	125	19	28	218
63	40	91	13	18	162
64	30	63	8	15	116
65	11	20	2	9	42
66	13	34	2	4	53
67	14	28	4	8	54
68	14	37	3	6	60
69	6	26	1		33
70	5	23	1	2	31
71	24	56	13	12	105
72	40	59		6	105
73	29	66	2	6	103
74	18	41	4	7	70
75	13	37	6	7	63
76	3	13	1	2	19
77	16	28	1	8	53
78	8	40	1	3	52
79	5	26	2	4	37
80	9	13	1	7	30
81	1	26	6	34	67
82	40	123	13	18	194
83	12	37	4	11	64
84	26	76	8	10	120
85	19	52	5	8	84
86	29	68	2	6	105

87	16	24	1	6	47
88	10	40	4		54
89	5	20		5	30
90	5	30	1	5	41
91	15	46	13	12	86
92	17	51	11	13	92
93	23	90	4	10	127
94	8	41	4	4	57
95	2	20	7	5	34
96		9	8	6	23
97	4	14		1	19
98		2	7	2	11
99	11	14	4	1	30
100	3	9		4	16
Celkový součet	5146	6935	992	3501	16574

Příloha 6 – Rozhodování stromové struktury – zjednodušený model

Příloha 7 – anotace předmětů

Název předmětu v ČJ

Strojové učení a neuronové sítě

Název předmětu v AJ

Machine Learning and Neural Network

Forma: prezenční nebo kombinovaná, popř. pro obě formy?

Prezenční

Garant předmětu (min. doktor)

Ing. Vojtěch Stehel, MBA, PhD.

Vyučující

Ing. Vojtěch Stehel, MBA, PhD.

Předpoklady ČJ

Základy práce s programem Matlab, nebo ochota se je v prvních týdnech doučit (cca 2 až 4 h samostudia).

Předpoklady AJ

Basics of working with Matlab, or willingness to learn them in the first weeks.

Cíle předmětu ČJ

Student se seznámí s nejběžnějšími algoritmy pro strojové učení. Tyto algoritmy dokáže optimalizovat a prakticky použít ve svém oboru.

Cíle předmětu AJ

Students will learn the most common algorithms for machine learning. He is able to optimize these algorithms and apply them in practice.

Výstupy z učení ČJ

Student zná běžně používané algoritmy pro strojové učení včetně základních neuronových sítí. Student umí algoritmy prakticky použít v aplikaci na svůj obor. Student rovněž umí výsledky optimalizovat.
Student je schopen porozumět principu strojového učení, chybám, které mohou vzniknout při kódování a interpretaci výsledků.

Výstupy z učení AJ

The student knows commonly used algorithms for machine learning including basic neural networks.

Students can practically use algorithms in their application. The student can also optimize the results.

The student is able to understand the principle of machine learning, errors that can occur in coding and interpretation of results.

Osnova ČJ

1. Strojové učení – úvod do problematiky
2. Získání a příprava dat
3. Regrese a klasifikace
4. Nejbližší soused
5. Naive Bayes Classification
6. Discriminant Analysis
7. Support Vector Machines
8. Stromy
9. Gaussian Process Regression
10. Zlepšení prediktivní schopnosti modelu
11. Neuronové sítě
12. Samoorganizující se mapy a dopředné sítě
13. Hluboké učení

Osnova AJ

1. Machine learning - introduction
2. Data acquisition and preparation
3. Regression and classification
4. Nearest Neighbor
5. Naive Bayes Classification
6. Discriminant Analysis
7. Support Vector Machines
8. Trees

9. Gaussian Process Regression
10. Improving Predictive Models
11. Neural network
12. Self-Organizing Maps and Feed-Forward Networks
13. Deep Learning

Literatura:

Kvasnička, V. - Beňušková, L. - Pospíchal, J. - Farkaš, I. - Tiňo, P. - Kráľ, A.:Úvod do teórie neurónových sietí. IRIS, Bratislava 1997.

Šíma, J. Generalized back propagation for interval training patterns, Neural Network World 2 (1992), 167-173.

Šíma, J. - Neruda, J.: Teoretické otázky neuronových sítí. Matfyzpress, Praha 1996.

Podmínky testu:

100 % seminární práce

Název předmětu v ČJ

Controlling

Název předmětu v AJ

Controlling

Forma: prezenční nebo kombinovaná, popř. pro obě formy?

Prezenční, kombinovaná

Garant předmětu (min. doktor)

Ing. Vojtěch Stehel, MBA, PhD. (přednášející)

Vyučující

Ing. Simona Hašková, Ph.D. (cvičící)
Ing. Pavel Rousek, Ph.D. (cvičící)
Ing. Vojtěch Stehel, MBA, PhD. (cvičící)

Cíle předmětu ČJ

Cílem předmětu je seznámit studenta s funkcí controllingu v podniku a jeho významem pro podniky. Přednášky řeší nejen strategické a operativní uchopení controllingu v podniku, ale v přiměřené míře detailu se dotknou i controllingu základních a vybraných sekundárních procesů v podniku. V neposlední řadě ozřejmí vznik krizí podniku, jejich řešení a především řešení odchylek podniku od stanoveného plánu.

Cíle předmětu AJ

The aim of the course is to acquaint students with the function of controlling in the company and its importance for businesses. The lectures address not only the strategic and operational grasp of controlling in the company, but in a reasonable degree of detail they will also touch on the controlling of basic and selected secondary processes in the company. Last but not least, it clarifies the origin of the company's crises, their solutions and, above all, the solution of the company's deviations from the set plan.

Výstupy z učení ČJ

Po úspěšném absolvování předmětu student: • 1. rozumí roli controllingu při řízení podniku, • 2. sbírá data pro účely controllingu z korektních zdrojů a v korektní formě, • 3. rozumí rozdílu mezi operativním a strategickým controllingem, • 4. rozumí controllingu základních

podnikových procesů, • 5. rozumí controllingu vybraných sekundárních procesů podniků, • 6. rozumí řízení odchylek podniku od stanoveného plánu, • 7. umí sestavit plány vybraných sekundárních procesů podniku, • 8. umí sestavit plány primárních procesů podniku, • 9. aplikuje controlling nejvyššího cíle podniku – řízení hodnoty podniku, • 10. řídí odchylky od stanoveného plánu.

Výstupy z učení AJ

Upon successful completion of the course, the student: • 1. understands the role of controlling in the management of the company, • 2. collects data for the purposes of controlling from correct sources and in the correct form, • 3. understands the difference between operational and strategic controlling, • 4. understands the controlling of basic business processes, • 5. understands the controlling of selected secondary processes of companies, • 6. means the management of the company's deviations from the established plan, • 7. can compile plans of selected secondary processes of the company, • 8. can draw up plans for the company's primary processes, • 9. applies controlling the highest goal of the company - value management the company, • 10. manages deviations from the set plan.

Osnova ČJ

Přednášky:

1. Vymezení základních pojmů – kontrola a controlling, základní zdroje dat. Procesní pohled na řízení podniku.
2. Strategický a operativní controlling.
3. Operativní marketingový controlling a jeho nástroje.
4. Obchodní controlling a jeho nástroje.
5. Výrobní controlling – controlling kvality TQM/EFQM.
6. Strategický finanční controlling – hodnota firmy – pohled akcionáře.
7. Finanční controlling – výkazy účetní závěrky, benchmarking.
8. Finanční controlling – kalkulace.
9. Finanční controlling – kalkulace/spotřeba výrobních faktorů – materiálu, dlouhodobého majetku.
10. Finanční controlling – kalkulace moderními metodami ABC.
11. Personální controlling.
12. Controlling při řízení inovací a výzkumu.
13. Odchylky a jejich řízení – krize podniku.

Semináře:

1. Zdroje dat pro controlling – interní a externí zdroje dat.
2. Case study strategického a operativního controllingu.
3. Příprava marketingového plánu podniku.
4. Příprava obchodního plánu podniku.
5. Příprava výrobního plánu podniku.
6. Stanovení parametrů pro řízení hodnoty podniku.

7. Realizace finančního benchmarkingu podniku.
8. Příprava kalkulací výrobků I.
9. Příprava kalkulací výrobků II.
10. Příprava kalkulací výrobků III.
11. Příprava plánu lidských zdrojů.
12. Příprava plánu vývoje a výzkumu.
13. Case study – řízení odchylek.

Osnova AJ

Lectures:

1. Definition of basic terms - control and controlling, basic data sources. Process view of business management.
2. Strategic and operational controlling.
3. Operational marketing controlling and its tools.
4. Business controlling and its tools.
5. Production controlling - TQM / EFQM quality controlling.
6. Strategic financial controlling - company value - shareholder view.
7. Financial controlling - financial statements, benchmarking.
8. Financial controlling - calculation.
9. Financial controlling - calculation / consumption of production factors - material, fixed assets.
10. Financial controlling - calculation by modern ABC methods.
11. Personnel controlling.
12. Controlling in the management of innovation and research.
13. Deviations and their management - business crisis.

Seminars:

1. Data sources for controlling - internal and external data sources.
2. Case study of strategic and operational controlling.
3. Preparation of the company's marketing plan.
4. Preparation of the business plan of the company.
5. Preparation of the company's production plan.
6. Determination of parameters for business value management.
7. Implementation of financial benchmarking of the company.
8. Preparation of product calculations I.
9. Preparation of product calculations II.
10. Preparation of product calculations III.
11. Preparation of human resources plan.
12. Preparation of development and research plan.
13. Case study - deviation management.

Literatura:

povinná literatura

- LAZAR, J., 2012. Manažerské účetnictví a controlling. Praha: Grada. ISBN 978-80-247-4133-8.
- ŠOLJAKOVÁ, L., J. FIBÍROVÁ a J. WAGNER, 2013. Manažerské účetnictví I.: případové studie a příklady. Praha: Oeconomica. ISBN 978- 80-245-1952-4.
- VOCHOZKA, M. et al., 2016. Controlling. 2. vyd. České Budějovice: Vysoká škola technická a ekonomická v Českých Budějovicích. ISBN 978- 80-7468-110-3.
- WALTHER, L. M., 2017. Managerial accounting. [s. l.]: CreateSpace Independent Publishing Platform. ISBN 978-1548394325.
- BREALEY, R. A., S. C. MYERS a F. ALLEN, 2014. Principles of corporate finance. 11. ed., global ed. New York: McGraw-Hill Education. ISBN 978-0-07-715156-0.

doporučená literatura

- VÁCHAL, J. et al., 2013. Podnikové řízení. Praha: Grada. ISBN 978-80- 247-4642-5.
- VOCHOZKA, M. et al., 2012. Podniková ekonomika. Praha: Grada. ISBN 978-80-247-4372-1.

Podmínky testu:

Hodnocení předmětu se skládá z průběžného hodnocení (30 – 0 bodů) a z písemné zkoušky (70 – 0 bodů). Celková klasifikace je součtem bodů z průběžného hodnocení a písemné zkoušky. Celková klasifikace předmětu, tj. body z písemné zkoušky (70 – 0) + body z průběžného hodnocení (30 – 0 bodů): A 100 – 90, B 89,99 – 84, C 83,99 – 77, D 76,99 – 73, E 72,99 – 70, FX 69,99 – 30, F 29,99 – 0. Student prezenční formy studia je povinen na kontaktní výuce, tj. vše kromě přednášek, splnit povinnou 70% účast. Pokud účast nebude splněná, bude student automaticky klasifikován „F“

Název předmětu v ČJ

Metodika odborné práce

Název předmětu v AJ

Technical thesis methodology

Forma: prezenční nebo kombinovaná, popř. pro obě formy?

Prezenční, kombinovaná

Garant předmětu (min. doktor)

prof. Ing. Marek Vochozka, MBA, Ph.D., dr. h.c.

Vyučující

Ing. Jakub Horák, MBA (cvičící)
Ing. Jana Janíková, MBA (cvičící)
Ing. Jiří Kučera (cvičící)
Ing. Vojtěch Stehel, MBA, PhD. (přednášející, cvičící)
prof. Ing. Marek Vochozka, MBA, Ph.D., dr. h.c. (přednášející, cvičící)

Cíle předmětu ČJ

Cílem předmětu je získání odborných znalostí a praktických dovedností v oblasti přípravy, zpracování, prezentace a obhajoby odborných textů.

Cíle předmětu AJ

The objective of the course is to gain professional knowledge and practical skills in the field of preparation, writing, presentation, and defence of professional texts.

Výstupy z učení ČJ

Po úspěšném absolvování předmětu student: • 1. vybere vhodné téma odborného textu. • 2. vymezí cíl své práce. • 3. provede kvalitní literární rešerši. • 4. vymezí hypotézy. • 5. definuje výzkumné otázky. • 6. zpracuje metodiku (vhodně zvolí základní vědecké metody) vedoucí k potvrzení, či vyvrácení hypotéz, k odpovědím na výzkumné otázky a ke splnění cíle práce. • 7. vhodně prezentuje výsledky své práce. • 8. provede diskusi výsledků. • 9. vyhodnotí výsledky práce. • 10. uplatní pravidla formální úpravy odborného textu. • 11. řádně prezentuje svou odbornou práci.

Výstupy z učení AJ

Upon successful completion of the course, the student is able to: • 1. select a suitable topic of the text. • 2. specify the objective of the text. • 3. do quality literary research. • 4. formulate hypotheses. • 5. define research questions. • 6. choose a methodology (suitable basic scientific methods) to confirm or reject the hypotheses, find answers to research questions, and to achieve the objective of the work. • 7. present the results of his/her work. • 8. discuss the results achieved. • 9. evaluate the results achieved. • 10. apply the formatting rules for professional texts. • 11. properly present his/her work.

Osnova ČJ

Přednášky:

1. Úvod do předmětu (zaměření předmětu, způsob výuky, zakončení, výběr tématu práce, registrace do NTK). Základy a principy výzkumné a tvůrčí práce. Autorská práva a plagiátorství.
2. Specifika jednotlivých typů odborných prací. Práce se zdroji (hodnotná literatura, zdroje, práce knihovny VŠTE, vyhledávání v databázích).
3. Název a úvod práce.
4. Rešerše v první fázi.
5. Rešerše ve druhé fázi.
6. Hypotézy nebo výzkumné otázky.
7. Metodika v bodech.
8. Metodika v textu.
9. Výsledky.
10. Diskuse výsledků – formulace základních zjištění.
11. Diskuse výsledků – dokončení.
12. Závěr.
13. Finalizace textu, příprava prezentace.

Semináře

1. Výběr tématu práce (odborného textu). Registrace do Národní technické knihovny. Základy a principy výzkumné a tvůrčí práce. Autorská práva a plagiátorství.
2. Analýza jednotlivých typů odborných prací. Prezentace knihovny VŠTE, vyhledávání v databázích.
3. Název a úvod práce.
4. Rešerše v první fázi.
5. Rešerše ve druhé fázi.
6. Hypotézy nebo výzkumné otázky.
7. Metodika v bodech.
8. Metodika v textu.
9. Výsledky.
10. Diskuse výsledků – formulace základních zjištění.
11. Diskuse výsledků – dokončení.

12. Závěr.
13. Finalizace textu, příprava prezentace

Osnova AJ

Lectures:

1. Introduction in the course (the focus, teaching methods, completion of the course, selection of a topic, registration in National Library of Technology). Basics and principles of research and creative work. Copyright and plagiarism.
2. Specificities of individual types of professional texts. Working with resources (valuable literature, resources, VSTE library, searching in databases).
3. Title and introduction.
4. Literary research – first phase.
5. Literary research – second phase.
6. Hypotheses or research questions.
7. Methodology in points.
8. Methodology in the text.
9. Results.
10. Discussion of results – formulation of basic findings.
11. Discussion of results – completion.
12. Conclusion.
13. Text finalization, preparation of presentation.

Seminars

1. Selection of a topic (of the professional text). Registration in National Library of Technology. Basics and principles of research and creative work. Copyright and plagiarism.
2. Analysis of individual types of professional texts. VŠTE library, searching in databases.
3. Title and introduction.
4. Literary research – first phase.
5. Literary research – second phase.
6. Hypotheses or research questions.
7. Methodology in points.
8. Methodology in the text.
9. Results.
10. Discussion of results – formulation of basic findings.
11. Discussion of results – completion.
12. Conclusion.
13. Text finalization, preparation of presentation.

Literatura:

povinná literatura

- VOCHOZKA, M. et al., 2016. Metodika odborné práce. 2. dopl. a rozš. vyd. České Budějovice: Vysoká škola technická a ekonomická v Českých Budějovicích. ISBN 978-80-7468-108-0.
- BHATTACHERJEE, A., 2012. Social Science Research: Principles, Methods, and Practices. 2nd edition. Tampa: University of South Florida. ISBN 978-1475146127.
- BAILEY, S., 2011. Academic Writing. A Handbook for International Students. Third edition. London, New York: Routledge. ISBN 978-0- 203-83165-6.

doporučená literatura

- KAPOUNOVÁ, J. a P. KAPOUN, 2017. Bakalářská a diplomová práce: od zadání po obhajobu. Praha: Grada. ISBN 978-80-271-0079-8.
- PAULOVČÁKOVÁ, L. et al., 2015. Jak vypracovat bakalářskou a diplomovou práci. 6. aktualiz. vyd. Praha: Univerzita Jana Amose Komenského Praha. ISBN 978-80-7452-106-5.
- ÚŘAD VLÁDY ČR, 2017. Metodika hodnocení výzkumných organizací a hodnocení programů účelové podpory výzkumu, vývoje a inovací. In: Rada pro výzkum, vývoj a inovace [online]. Praha: Rada pro výzkum, vývoj a inovace, 2017, [cit. 2017-02-02].
- ŠANDEROVÁ, J., 2005. Jak číst a psát odborný text ve společenských vědách. Několik zásad pro začátečníky. Praha: Sociologické nakladatelství. ISBN 80-86429-40-7.

Podmínky testu:

Pro úspěšné splnění předmětu je nutné dosáhnout v součtu za seminární práci a její prezentaci 70 %.
Celková klasifikace předmětu, tj. body za závěrečné hodnocení (100 - 0): A 100 – 90, B 89,99 – 84, C 83,99 – 77, D 76,99 – 73, E 72,99 – 70, FX 69,99 – 30, F 29,99 – 0.

Název předmětu v ČJ

Finance podniku I

Název předmětu v AJ

Corporate finance I

Forma: prezenční nebo kombinovaná, popř. pro obě formy?

Prezenční, kombinovaná

Garant předmětu (min. doktor)

Ing. Simona Hašková, Ph.D.

Vyučující

Ing. Taťána Hajdíková, Ph.D. (cvičící)
Ing. Simona Hašková, Ph.D. (přednášející, cvičící)
Ing. Vojtěch Stehel, MBA, PhD. (přednášející, cvičící)
Ing. Jakub Horák, MBA (cvičící)
Ing. Jiří Kučera (cvičící)
Ing. Pavel Rousek, Ph.D. (cvičící)

Předpoklady ČJ

absolvovaný nebo souběžně studovaný předmět Metodika odborné práce

Předpoklady AJ

completed or in parallel studied subject BPE_MOP

Cíle předmětu ČJ

Student se naučí pracovat s výstupními daty z controllingu a jiných podpůrných činností, plně chápe význam dat a dokáže je přetvořit v podklady rozhodování. Absolvent předmětu porozumí finančnímu vyjádření vztahů v podniku a ve vztahu k okolí. Rozumí jeho majetkové, kapitálové a personální struktuře podniku.

Cíle předmětu AJ

The student will learn to work with output data from controlling and other supporting activities, fully understands the meaning of data and can transform them into decision-making materials. The graduate of the course will understand the financial expression of relationships in the

company and in relation to the environment. Understands its property, capital and personnel structure of the company.

Výstupy z učení ČJ

Po úspěšném absolvování předmětu student: • 1. rozumí úloze finančního manažera v podniku, • 2. optimalizuje peněžní tok, tok zásob a pohledávek podniku, • 3. hodnotí investiční alternativy z pohledu jejich finančního dopadu, • 4. chápe strategické i taktické finanční rozhodování, • 5. optimalizuje kapitálovou a majetkovou strukturu podniku.

Výstupy z učení AJ

Upon successful completion of the course, the student: • 1. understands the role of the financial manager in the company, • 2. optimizes the cash flow, flow of inventories and receivables of the company, • 3. evaluate investment alternatives in terms of their financial impact, • 4. understands strategic and tactical financial decision-making, • 5. optimizes the capital and property structure of the company.

Osnova ČJ

Přednášky:

1. Úloha finančního manažera v organizaci. Vazba controllingu a financí podniku. Práce s daty.
2. Časová hodnota peněz, vztah rizika a výnosů.
3. Řízení zásob.
4. Řízení hotovosti, řízení peněžního toku.
5. Řízení pohledávek.
6. Dlouhodobý majetek a investiční rozhodování – statické metody.
7. Dlouhodobý majetek a investiční rozhodování – dynamické metody.
8. Finanční dopad získávání nových zaměstnanců. Finanční dopad vzdělávání a rozvoje stávajících zaměstnanců.
9. Nákladové modely.
10. Financování vlastním kapitálem.
11. Financování cizím kapitálem.
12. Strategické finanční rozhodování a optimalizace kapitálové struktury podniku.
13. Finanční a kapitálové trhy.

Semináře:

1. Úloha finančního manažera v organizaci. Vazba controllingu a financí podniku. Práce s daty.
2. Časová hodnota peněz, vztah rizika a výnosů.
3. Řízení zásob.
4. Řízení hotovosti, řízení peněžního toku.
5. Řízení pohledávek.
6. Dlouhodobý majetek a investiční rozhodování – statické metody.

7. Dlouhodobý majetek a investiční rozhodování – dynamické metody.
8. Finanční dopad získávání nových zaměstnanců. Finanční dopad vzdělávání a rozvoje stávajících zaměstnanců.
9. Nákladové modely.
10. Financování vlastním kapitálem.
11. Financování cizím kapitálem.
12. Strategické finanční rozhodování a optimalizace kapitálové struktury podniku.
13. Finanční a kapitálové trhy.

Osnova AJ

Lectures:

1. The role of the financial manager in the organization. The connection between controlling and company finance. Work with data.
2. Time value of money, the relationship between risk and return.
3. Inventory management.
4. Cash management, cash flow management.
5. Receivables management.
6. Fixed assets and investment decisions - static methods.
7. Fixed assets and investment decisions - dynamic methods.
8. Financial impact of recruiting new employees. Financial impact of training and development of existing employees.
9. Cost models.
10. Equity financing.
11. Foreign capital financing.
12. Strategic financial decision-making and optimization of the company's capital structure. • 13. Financial and capital markets.

Seminars:

1. The role of the financial manager in the organization. The connection between controlling and company finance. Work with data.
2. Time value of money, the relationship between risk and return.
3. Inventory management.
4. Cash management, cash flow management.
5. Receivables management.
6. Fixed assets and investment decisions - static methods.
7. Fixed assets and investment decisions - dynamic methods.
8. Financial impact of recruiting new employees. Financial impact of training and development of existing employees.
9. Cost models.
10. Equity financing.
11. Foreign capital financing.
12. Strategic financial decision-making and optimization of the company's capital structure.

13. Financial and capital markets.

Literatura:

povinná literatura

- HAŠKOVÁ, S. a M. VOCHOZKA, 2018. Finance podniku I. České Budějovice: Vysoká škola technická a ekonomická v Českých Budějovicích. ISBN 978-80-7468-128-8.
- KISLINGEROVÁ, E. et al., 2010. Manažerské finance. 3. vyd. Praha: C.H. Beck. ISBN 978-80-7400-194-9.
- SCHOLLEOVÁ, H. a P. ŠTAMFESTOVÁ, 2015. Finance podniku. Sbírka řešených příkladů a otázek. Praha: Grada. ISBN 978-80-247-55441.
- BREALEY, R. A., S. C. MYERS a F. ALLEN, 2014. Principles of corporate finance. 11. ed., global ed. New York: McGraw-Hill Education. ISBN 978-0-07-715156-0.
- ROSS, S., R. WESTERFIELD a B. JORDAN, 2017. Essentials of corporate finance. 9 vyd. [s. l.]: McGraw-Hill Education. ISBN 9781259277214.

doporučená literatura

- KALOUDA, F., 2017. Finanční analýza a řízení podniku. 3. rozš. vyd. Plzeň: Vydavatelství a nakladatelství Aleš Čeněk, s.r.o. ISBN 978-80-7380646-0.
- MAREK, P., 2006. Studijní průvodce financemi podniku. Praha: Ekopress. ISBN 80-86119-37-8.
- REŽŇÁKOVÁ, M., 2012. Efektivní financování rozvoje podnikání. Praha: Grada. ISBN 978-80-247-1835-4.

Podmínky testu:

Pro úspěšné splnění předmětu je nutné v součtu dosáhnout z průběžného a závěrečného hodnocení minimálně 70 % za níže stanovených podmínek. V průběžném hodnocení lze získat 30 bodů tj. 30 %. V průběžném testu lze získat 15 bonusových bodů. V závěrečném hodnocení lze celkem získat 70 bodů tj. 70 %. Seminární práce přitom respektuje strukturu a náležitosti odborného textu (požadavky jsou stanoveny v rámci předmětu BPE_MOP a pokyny vyučujícího). Celková klasifikace předmětu, tj. body za závěrečné hodnocení (70 - 0) + body z průběžného hodnocení (30 - 0): A 100 – 90, B 89,99 – 84, C 83,99 – 77, D 76,99 – 73, E 72,99 – 70, FX 69,99 – 30, F 29,99 – 0. Student prezenční formy studia je povinen na kontaktní výuce, tj. vše kromě přednášek, splnit povinnou 70% účast. Pokud účast nebude splněná, bude student automaticky klasifikován „F“.

Název předmětu v ČJ

Finance podniku II

Název předmětu v AJ

Corporate finance II

Forma: prezenční nebo kombinovaná, popř. pro obě formy?

Prezenční, kombinovaná

Garant předmětu (min. doktor)

Ing. Jaromír Vrbka, MBA, PhD. (přednášející)

Vyučující

Ing. Simona Hašková, Ph.D. (cvičící)
Ing. Vojtěch Stehel, MBA, PhD. (přednášející, cvičící)
Ing. Tomáš Krulický, MBA (cvičící)
Ing. Pavel Rousek, Ph.D. (cvičící)

Cíle předmětu ČJ

Student rozšiřuje základní znalosti procesů podnikových financí na úroveň pochopení specializovanějších aktivit směřujících k usměrňování těchto dějů. Absolvent předmětu dokáže plánovat aktivity tak, aby optimalizoval podnikové procesy. Východiskem pro plánování je znalost finančních a kapitálových zdrojů podniku a metod hodnocení podniku tak, aby vše směřovalo k maximální shodě s potřebami vlastníka a akcionáře.

Cíle předmětu AJ

The student extends basic knowledge of business finance processes to the level of understanding of more specialized activities aimed at directing these processes. A graduate student can plan activities to optimize business processes. The starting point for planning is the knowledge of the financial and capital resources of the enterprise and the methods of the company's valuation so that everything is directed to the maximum match with the needs of the owner and the shareholder.

Výstupy z učení ČJ

Po úspěšném absolvování předmětu student: • 1. chápe hlavní cíle podniku v podobě růstu hodnoty pro akcionáře, • 2. určí hodnotu podniku pomocí výnosových metod oceňování podniku, • 3. chápe riziko jako součást činnosti finančního manažera, • 4. identifikuje riziko a navrhuje způsoby jeho eliminace, • 5. zná metody komplexního hodnocení podniku, • 6. určí, zda se podnik nachází v dobré finanční kondici, či nikoliv, • 7. rozumí leasingu a franchisingu jako novým metodám financování činnosti podniku, • 8. zná nové metody práce s pohledávkami – faktoring a forfaiting, • 9. aplikuje dividendovou politiku podniku, • 10. rozumí životnímu cyklu podniku, • 11. rozumí rozdílnému financování podniku v různých fázích jeho života, • 12. sestaví dlouhodobý finanční plán, • 13. rozumí vztahu dlouhodobého a krátkodobého finančního plánu, • 14. sestaví krátkodobý finanční plán.

Výstupy z učení AJ

Upon successful completion of the course, the student: • 1. understands the company's core business goals by increasing shareholder value, • 2.identifies the value of a business using revenue pricing methods, • 3. understands the risk as part of the financial manager's activity, • 4. identifies risk and suggests ways to eliminate it, • 5. knows the methods of a comprehensive business evaluation, • 6. determins whether a business is in good financial or not, • 7. understands leasing and franchising as new methods of financing the company's business, • 8. knows new ways of working with receivables - factoring and forfeiting, • 9. applies the company's dividend policy, • 10. understands the life cycle of the company, • 11. understands the different financing of the company at various stages of its life, • 12. draws up a long-term financial plan, • 13. understands the relationship of the long-term and short-term financial plan, • 14. draws up a short-term financial plan.

Osnova ČJ

Přednášky:

1. Hodnota podniku pro investory, akcionáře, věřitele.
2. Riziko a možnosti jeho eliminace.
3. Metody hodnocení podniku – finanční analýza.
4. Metody hodnocení podniku – bonitní modely.
5. Metody hodnocení podniku – bankrotní modely a ostatní modely.
6. Leasing. Franchising.
7. Faktoring. Forfaiting.
8. Zisk jako nástroj refinancování činnosti podniku. Dividendová politika.
9. Mimořádné financování – financování při zakládání, rozšiřování, sanaci, fúzi a zániku I.
10. Mimořádné financování – financování při zakládání, rozšiřování, sanaci, fúzi a zániku II.
11. Finanční plánování – metody sestavování plánu, kontrola plnění plánu.
12. Krátkodobý a dlouhodobý finanční plán.
13. Teoretické poznatky Financí podniku II. versus finanční řízení v praxi.

Semináře:

1. Hodnota podniku pro investory, akcionáře, věřitele.
2. Riziko a možnosti jeho eliminace.
3. Metody hodnocení podniku – finanční analýza.
4. Metody hodnocení podniku – bonitní modely.
5. Metody hodnocení podniku – bankrotní modely a ostatní modely.
6. Leasing. Franchising.
7. Faktoring. Forfaiting.
8. Zisk jako nástroj refinancování činnosti podniku. Dividendová politika.
9. Mimořádné financování – financování při zakládání, rozšiřování, sanaci, fúzi a zániku I.
10. Mimořádné financování – financování při zakládání, rozšiřování, sanaci, fúzi a zániku II.
11. Finanční plánování – metody sestavování plánu, kontrola plnění plánu.
12. Krátkodobý a dlouhodobý finanční plán.
13. Teoretické poznatky Financí podniku II. versus finanční řízení v praxi.

Osnova AJ

Lectures:

1. Business value for investors, shareholders, creditors.
2. Risk and possibilities of its elimination.
3. Company evaluation methods - financial analysis.
4. Business valuation methods - credit models.
5. Business evaluation methods - bankruptcy models and other models.
6. Leasing. Franchising.
7. Factoring. Forfaiting.
8. Profit as an instrument of refinancing the business. Dividend policy.
9. Exceptional financing - financing in setting up, dissemination, rehabilitation, merger and dissolution I.
10. Exceptional financing - financing in setting up, dissemination, rehabilitation, merger and demise II.
11. Financial Planning - Planning methods, control of plan implementation.
12. Short-term and long-term financial plan.
13. What a graduate of Finance Finance knows and what he / she must complete.

Seminars:

1. Business value for investors, shareholders, creditors.
2. Risk and possibilities of its elimination.
3. Company evaluation methods - financial analysis.
4. Business valuation methods - credit models.
5. Business evaluation methods - bankruptcy models and other models.
6. Leasing. Franchising.
7. Factoring. Forfaiting.

8. Profit as an instrument of refinancing the business. Dividend policy.
9. Exceptional financing - financing in setting up, dissemination, rehabilitation, merger and dissolution I.
10. Exceptional financing - financing in setting up, dissemination, rehabilitation, merger and demise II.
11. Financial Planning - Planning methods, control of plan implementation.
12. Short-term and long-term financial plan.
13. What a graduate of Finance Finance knows and what he / she must complete.

Literatura:

povinná literatura

- HNILICA, J. a J. FOTR, 2009. Aplikovaná analýza rizika ve finančním managementu a investičním rozhodování. Praha: Grada. ISBN 978-80-247- 2560-4.
- KALOUDA, F., 2017. Finanční analýza a řízení podniku. 3. rozš. vyd. Plzeň: Vydavatelství a nakladatelství Aleš Čeněk. ISBN 978-80-7380-646- 0 .
- KISLINGEROVÁ, E. et al., 2010. Manažerské finance. 3. vyd. Praha: C.H. Beck. ISBN 978-80-7400-194-9.
- BREALEY, R. A., S. C. MYERS a F. ALLEN, 2014. Principles of corporate finance. 11. ed., global ed. New York: McGraw-Hill Education. ISBN 978-0-07-715156-0.
- HOWARD, M., 2007. Accounting and business valuation methods: How to interpret IFRS accounts. [s. l]: CIMA Publishing. ISBN 978-80-750684682

doporučená literatura

- KUBÍČKOVÁ, D. a I. JINDŘICHOVSKÁ, 2015. Finanční analýza a hodnocení výkonnosti firmy. Praha: C. H. Beck. ISBN 978-80-7400-538-1.
- MEJSTŘÍK, M., M. PEČENÁ a P. TEPLÝ, 2014. Bankovnictví v teorii a praxi: Banking in theory and practice. Praha: Karolinum. ISBN 978-80- 246-2870-7.
- SCHOLLEOVÁ, H. a P. ŠTAMFESTOVÁ, 2015. Finance podniku. Sbírka řešených příkladů a otázek. Praha: Grada. ISBN 978-80-247-55441.
- VOCHOZKA, M., 2011. Metody komplexního hodnocení podniku. Praha: Grada. ISBN 978-802-4736-471.

Podmínky testu:

Pro úspěšné splnění předmětu je nutné v součtu dosáhnout z průběžného a závěrečného hodnocení minimálně 70 % za níže stanovených podmínek. V průběžném hodnocení lze získat 30 bodů tj. 30 %. V závěrečném hodnocení lze celkem získat 70 bodů tj. 70 %. Celková klasifikace předmětu, tj. body za závěrečné hodnocení (70 - 0) + body z průběžného hodnocení (30 - 0): A 100 – 90, B 89,99 – 84, C 83,99 – 77, D 76,99 – 73, E 72,99 – 70, FX 69,99 – 30, F 29,99 – 0. Student prezenční formy studia je povinen na kontaktní výuce, tj. vše kromě přednášek, splnit povinnou 70% účast. Pokud účast nebude splněná, bude student automaticky klasifikován „F“.