VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

PREDIKTIVNÍ MODELOVÁNÍ V JAZYCE PYTHON

PREDICTIVE MODELLING WITH PYTHON

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JAN DUDA**
AUTHOR

VEDOUCÍ PRÁCE **Doc. Ing. JAROSLAV ZENDULKA, CSc.**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2018/2019

Zadání bakalářské práce

Student: **Duda Jan**
Program: Informační technologie
Název: **Prediktivní modelování v jazyce Python**
Predictive Modelling with Python
Kategorie: Data mining

Zadání:

1. Seznamte se se základy získávání znalostí z dat (data mining).
2. Seznamte se s dostupnými prostředky na podporu dolování z dat v jazyce Python.
3. Po dohodě s vedoucím zvolte vhodnou případovou studii zaměřenou na uplatnění metod predikce.
4. Podrobněji se seznamte s přístupy a algoritmy pro řešení zvolené úlohy a s potřebnými knihovnami jazyka Python.
5. Po dohodě s vedoucím naprogramujte řešení úlohy tak, aby demonstrovalo využití prostředků jazyka Python pro predikci.
6. Na základě získaných zkušeností zhodnoťte vhodnost použití jazyka Python pro řešení podobných úloh.

Literatura:

- Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Third Edition. Morgan Kaufmann Publishers, 2012, 703 p., ISBN 978-0-12-381479-1. (Kapitola 1 a další informace relevantní ke zvolené úloze a použitým datům).
- Layton, R.: Learning Data Mining with Python: Use Python to manipulate data and build predictive models. 2nd Edition. Packt Publishing, 2017, 358 p., ISBN 978-1787126787.
- Nielsen, F.A.: Data Mining with Python. 2015. 101 p. Dostupné na http://www.freetechbooks.com/data-mining-with-python-working-draft-t1159.html.
- Torgo, L.: Data Mining with R. Learning with Case Studies. Chapman & Hall/CRC Press. 2011. 289 p.
- KDnuggets Datasets for Data Mining and Data Science. Dostupné na https://www.kdnuggets.com/datasets/index.html.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Zendulka Jaroslav, doc. Ing., CSc.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 5. března 2019

Abstrakt

Cílem této bakalářské práce je seznámení s oborem dolování dat a procesu získávání dat z databází. Uvádí nejdůležitější postupy prováděné při dolování. Následně jsou jednotlivé techniky použity v případové studii implementované v jazyce Python. Ta se zaměřuje na predikci indexu S&P 500, který má reprezentovat vývoj akciových trhů na americké burze. Je využito klasifikačních i regresních modelů. Pro vyhodnocení úspěšnosti modelů je využito experimentální metody Monte Carlo.

Abstract

The main goal of this bachelor thesis is get to know with the data mining and its domain, also with the Knowledge discovery in databases process. It shows the most importnant approaches, which are implemented in Python language afterwards. The case study contains the prediction of index S&P 500 describing stock market developments on the US stock exchange. Both classification and regression models are used for the forecasting. Model evaluation is reached by the Monte Carlo experimental method.

Klíčová slova

Dolování dat, predikce, strojové učení, jazyk Python, klasifikace, regrese, technické identifikátory, neuronové sítě, SVM, MARS, finanční analýza, index S&P 500, časové řady, Monte Carlo

Keywords

Data mining, prediction, machine learning, Python language, classification, regression, technical indicators, neural networks, SVM, MARS, financial analysis, index S&P 500, time series, Monte Carlo

Citace

DUDA, Jan. *Prediktivní modelování v jazyce Python.* Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. Ing. Jaroslav Zendulka, CSc.

Prediktivní modelování v jazyce Python

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Doc. Ing. Jaroslava Zendulky, CSc. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Duda
14. května 2019

Poděkování

Rád bych tímto chtěl poděkovat vedoucímu práce panu Doc. Ing. Jaroslavu Zendulkovi, Csc. za veškeré odborné rady, postřehy a přívětivý přístup poskytnutý při tvorbě této práce. Rovněž bych chtěl poděkovat Bc. Otu Duškovi za podporu během studia a Bc. Evě Kučerové za motivování a jazykovou korekturu.

Obsah

Kapitola 1

Úvod

Od konce 20. století, kdy se internet začínal stávat dostupnějším pro čím dál tím větší počet domácností, se lidstvo dostalo do stádia, kde prakticky kdokoliv ve vyspělých státech má možnost internet využívat.

Rovněž vývoj techniky v nejrůznějších oblastech dospěl do fáze, kdy počítače a stroje během svého používání generují obrovské množství dat, které je skladováno ve většině případů bez jakéhokoliv dalšího využití, přestože mohou nabídnout mnohdy cenné informace, které lze použít k dalším vylepšením, změně obchodní strategie, vynálezu dalších léčebných metod a mnohem více.

Za účelem získání těchto informací byla vyvinuta v informačních technologiích oblast s názvem získávání znalostí z dat (anglicky knowledge discovery in databases, přeneseně data mining). Pomocí matematických modelů, statistických postupů, lidského myšlení, faktů a výpočetních možností dnešních počítačů jsme schopni z velkého množství dat získat informace, které jsou pro člověka pouhým pozorováním skryté. Získávání znalostí samo o sobě se těší v posledních letech velkému rozmachu a jeho možnosti se zvětšují.

Prediktivní modelování, jakožto zadání této diplomové práce, je podmnožinou výše zmíněné oblasti. Jak již název napovídá, jedná se o vytváření modelu za pomocí historických a aktuálních dat, který je schopný předpovědět vývoj sledovaného problému v následujícím časovém období.

Úvodní kapitoly bakalářské práce představí teoretickou podstatu dolování dat a hlavní proces získávání znalostí z databází, který se pak aplikuje na různé typy dolovacích úloh. Následně jsou představeny v kapitole 5 prostředky programovacího jazyka Python, který je použitý pro implementaci případové studie.

Ta se zabývá doménou finanční analýzy, konkrétně se zaměřuje na predikování vývoje akciových trhů reprezentovaných indexem S&P 500. Za pomocí historických dat budou prakticky v jazyce Python představeny způsoby dolování ve snaze získat aktuální informace o vývoji trhu. Případovou studii popisují kapitoly 6 a 8, mezi které je vložena kapitola obsahující teoretické informace k přístupům použitých pro predikci tohoto indexu. V kapitole 9 jsou nastíněny plusy a minusy jazyka Python při použití v rámci dolování dat.

Kapitola 2

Dolování dat

Tato kapitola představí obecný pojem dolování dat a uvede hlavní souvislosti s tímto odvětvím. Představí, v jakých doménách se dolování dat prosazuje, a jiné využívané obory při analýze. Poslední sekce popisuje stěžejní aspekty, kterým se čelí při řešení těchto úloh.

2.1 Definování pojmu dolování dat

Výraz dolování dat (možná přesněji dolování z dat, z anglického data mining) se snaží o nadnesené přenesení pojmu těžby zlata (gold mining) do sféry informačních technologií [9]. Jedná se tedy o získání užitečných informací či nalezení vzorů a modelů z velkého množství dat (typicky databází). Tyto informace nejsou na první pohled postřehnutelné, proto proces získávání zahrnuje sofistikované postupy a matematické modely, aby se dospělo k dostatečně kvalitním a užitečným výsledkům.

2.2 Vědní oblasti využívané dolováním

Ukazuje se čím dál tím častěji, že oblast dolování dat je velice komplexním oborem. Dobrý analytik v této sféře musí zvládat minimálně základy z dalších oborů.

Data pro dolování budou převážně uložena v **databázích**. Tudíž bez vytváření SQL dotazů nad databázemi se dolování z dat neobejde. V jiných případech se data budou nacházet na webových stránkách. Pro potřeby získání dat z webů se v poslední době rozvíjí oblast zvaná **web scraping**, tedy extrakce dat z webových stránek.

U zkoumání vstupních dat, ale především u vyhodnocování výsledků, je nutná precizní **vizualizace dat**. Knihovny **programovacích jazyků** (další z potřebných dovedností) pro vykreslování grafů a statistik jsou již na vysoké úrovni, není ovšem jednoduché na závěr bádání prezentovat své výsledky klientům a laickým uživatelům, aby závěry byly správným způsobem pochopeny.

Matematické a statistické postupy jsou nezbytnou součástí celého procesu získávání dat z databází. V mnoha případech se nejedná o triviální záležitosti.

Stěžejní bod samotného procesu, tedy krok, kdy z již zpracovaných dat je nutné nalézt užitečné informace, zahrnuje a prolíná velké množství dalších odvětví. Využívá **algoritmizaci** jak ze **strojového učení**, tak z **umělé inteligence**.

2.3 Kde nachází dolování své uplatnění

Dolování dat je v poslední době považováno za nejrychleji rozvíjející se technologii v oblasti **podnikání**. Společnosti mnohdy prahnou po získání co největšího množství informací o svých zákaznících, aby jim své produkty mohly vytvářet na míru. Tyto informace mohou být získávána mimo jiné i za pomoci skenerů čárových kódů, RFID identifikátorů atd. Předmětem zájmu jsou data i pro pojišťovny, supermarkety a mobilní operátory [5]. Taktéž analyzují jejich chování na internetu za účelem reklamy, která je vhodná právě pro daného uživatele (např. analýza obsahu nákupního košíku) [27].

Nejen získávání dat o zákaznících, ale i analýzy pro budoucí vývoj firemních subjektů či jejich řízení nabízí technologie dolování dat. Návrh efektivního rozložení skladu není výjimkou [21].

V **ekonomické sféře** mluvíme o finančních analýzách, rizika pro poskytování půjček, detekci podvodných aktivit a především predikování vývoje cash flow, cen, hodnot společností nebo kompletních akciových trhů [27]. Tímto oborem se zabývá i případová studie uvedená v kapitole 8.

Jako nepostradatelné se ukazuje dolování dat ve **výzkumné činnosti** - např. pochopení vztahů v přírodě, analýza klimatického systému Země, předpovídání počasí. V **medicíně** dopomáhá k identifikaci pacientů nakaženými nejrůznějšími chorobami.

Ve Spojených státech amerických naopak v roce 2016 přinesl pozdvižení model předpovídající potencionální nebezpečí pro okolí u osob podezřelých z trestného činu. Tento model vzniklý za pomoci strojového učení v minulém století používala americká policie. Ukázal se jako diskriminující pro osoby tmavé pleti a poukázal na jeden z problému dolování [19].

2.4 Zavádějící pojmy a rozdíly mezi nimi

S postupným vývojem snahy o zlepšení a vyvinutí oblasti pro získávání informací bylo uměle vytvořeno mnoho spolu souvisejících pojmů. To vedlo do stádia, kde se mnoho pojmů prolíná či rozdíly mezi nimi jsou minimální. To vede ke složitějšímu pochopení hlavních myšlenek. Proto zde budou vymezeny podrobněji problematické pojmy.

KDD - Knowledge Discovery Data

Význam pojmu dolování z dat je v literatuře a ve webových článcích často zavádějící. Pravdou je, že dolování je pouze jedním krokem v procesu KDD (Knowledge Discovery in Databases, česky získávání znalostí z databází) [6]. V tomto kroku se již pracuje se zpracovanými, transformovanými daty, používají se algoritmy pro nalezení užitečných informací a vzorů v datech [7].

Nicméně postupem času z praktických důvodů začal být pojem dolování z dat mylně používán pro kompletní KDD proces, tedy i s předzpracováním dat a závěrečnou prezentací výsledků. **Kvůli zjednodušení a rozšířenosti jak anglického pojmu data mining, tak i jeho českého překladu dolování z dat, budu ve své práci používat výrazy dolování z dat a proces KDD jako synonyma, i když se jedná o nepřesný překlad**.

Proces získávání znalostí z databází je popsán detailněji v následující kapitole (3). Obecně popisuje tento koncept a dopomůže k lepšímu pochopení odlišností těchto výrazů.

Strojové učení v souvislosti s dolováním dat

Machine Learning (strojové učení) je jedním z důležitých odvětví, které je dolováním využíváno. Cílem strojového učení je automaticky se naučit a rozpoznat vzory a na základě dat poté vytvářet inteligentní rozhodnutí. Zkoumá se především, jak toho dosáhnout. Jsou využívány již známé algoritmy a iterativní proces ke zlepšování rozhodnutí v reálném čase, vše bez jakékoliv lidské interakce či specifického programování.

Dolování především analyzuje obrovské množství dat z různých zdrojů k získání informací, objevuje nové závislosti, ale neřídí žádné procesy či akce. Jedná se především o bádání, lidský faktor je nutností, vstupní data jsou nestrukturované a nepředpracované.

Oba termíny se ale z velké části prolínají a využívají se navzájem.

2.5 Klíčové aspekty ovlivňující řešení dolovací úlohy

V průběhu procesu získávání cenných informací z dat se nezřídka objeví méně či více problémů, které mohou výrazně ovlivnit přesnost výsledných znalostí. Proto je dobré mít na paměti, na co se zaměřit anebo čeho se vyvarovat při dolovacích úlohách, aby byly výsledky práce použitelné.

Hlavní požadované vlastnosti

Existují minimálně dvě hlavní metriky, které určují, zda koncept zkoumání a následné aplikování bude pro danou konkrétní úlohu vhodný.

První z nich je **efektivita**. Řešení musí být schopno pracovat rychle v závislosti na databázích, se kterými ve většině případů bude pracovat. Do výsledků hodnocení, zda je řešení efektivní, velkou měrou zasahují použité algoritmy a struktury, se kterými koncept pracuje. Pokud se jedná o „malé“ databáze, nejedná se většinou o zásadní problémy, pokud je dolování a změny prováděné na databázi pomalé, v případě „obrovských“ databázích se již jedná o klíčový aspekt.

A druhou je **škálovatelnost**. Vznikající modely ve velké míře ovlivňuje objem dat použitý v trénovacích a testovacích množinách. Pokud model vytváří správná rozhodnutí u velkého objemu hodnot, nemusí tomu tak být u malé testovací množiny. To platí samozřejmě i opačným směrem. V praxi se tato škálovatelnost někdy nedaří splnit, nicméně je možné použití paralelizmu, případně vzorkování (sampling).

Další aspekty

V průběhu zkoumání úloh se objevují i další klíčové body, se kterými je nutné se vypořádat.

Především se jedná o **odlišnosti ve vstupních datech** (overfitting). Pokud se data pro učení a vytváření modelu liší od dat, na které je poté model aplikován, je zřejmé, že to bude mít vliv i na přesnost výsledků. Správný výběr dat, který bude vhodným reprezentativním vzorkem, je nutností.

Problematika **odlehlých objektů** (outliers) je poměrně známá. Vznikají buď pomocí **chybných vstupních dat**, nebo zkrátka tyto extrémní hodnoty byly naměřeny. I přes nízký počet těchto objektů se mohou zásadně promítnout do výsledného obrazu. Je na zvážení každého analytika, jak se s nimi vypořádá. Nabízí se odstranění těchto záznamů, nahrazení průměrnými hodnotami, případně pracování s nimi beze změny. Chybná data vznikají díky poškození souborů, manuálnímu zadávání atd.

Interakce lidského faktoru může hrát velkou roli. Techničtí experti zpravidla neznají podrobně zkoumaná data, naopak uživatelé a pracovníci s touto znalostí nediskutují s technickými experty danou úlohu. Vede to opět k horším výsledkům.

Vizualizace výsledků již byla výše zmíněna několikrát. Jedná se o účinný nástroj jak u poznávání vstupních dat, tak u interpretace výsledků. Může v mnoha ohledech usnadnit práci, ale také být zavádějící a přivést ke skutečnostem, které nemají pro danou úlohu vliv.

Databáze velkých nadnárodních společností jsou enormně velké, což může vést k **velké dimenzi** dat. Jsou obsažena i data, která nejsou pro dolovací úlohu potřeba a pouze zkreslují výsledný model.

Odvětví dolování z dat naráží na **vládní vyhlášky**, ale taky s **nevstřícností korporátních firem**. Mnohdy nejsou velmi užitečná data z těchto důvodů předána analytikům a vědcům díky obavám z porušení zákonů či úniku citlivých informací (o osobách, firemních strategiích atd.).

Kapitola 3

Proces získávání znalostí z databází

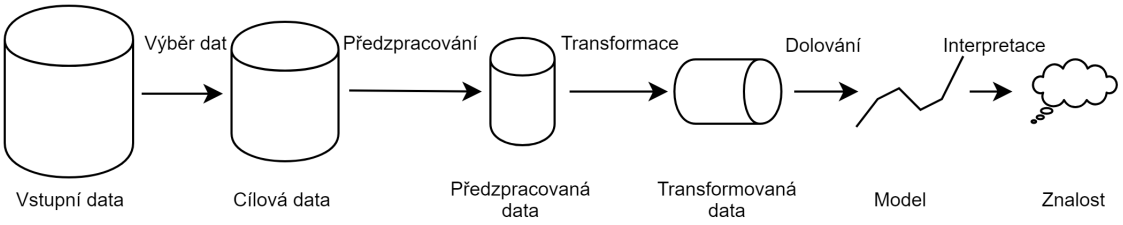
V této kapitole bude podrobněji popsán proces získávání znalostí z databází a kroky předcházející samotnému dolování, tedy načtení úvodní datové množiny, její hlubší analýza a úpravy nutné pro co nejpřesnější výsledky při dolování.

Stěžejní informace této kapitoly byly převzaty z [4], [27].

Definice 3.1 *„Získávání znalostí z databází (KDD, Knowledge discovery in databases) je proces hledání užitečných informací a vzorů v datech“ [6].*

V definici výše je potřeba zdůraznit slovo *užitečné*, což v mnoha případech je nejobtížnější část dolování dat. Hledané informace nejsou triviální, tedy nejsou dosažitelné pouhým zadáním databázového dotazu či pomocí sumarizujících statistik.

Celkový proces je komplexní, zahrnující posloupnost několika kroků. Graficky je znázorněn na obrázku 3.1. Často dochází k opakování některých kroků **v určitých iteracích**. Vstupem do procesu je obvykle velké množství dat (často z více zdrojů), výstupem užitečné informace požadované uživateli. Pro zachování kvalitních a užitečných informací bývá v rámci procesu nutná kooperace uživatelů, kteří znají dopodrobna obsah vstupních dat, a technických expertů pracujících na dolování.



Obrázek 3.1: Proces získávání znalostí z databází. Převzato z [7]

Základní kroky procesu získávání znalostí z databází:

1. **Výběr dat:** V tomto kroku jsou načtena data obvykle z více zdrojů. Probíhá načtení z různých databází, souborů, pamětí senzorů apod. Poté přichází na řadu zkoumání dat.

2. **Předzpracování dat:** Jedná se o velice důležitý krok. Data po načtení nebývají ve stavu, který by byl ihned použitelný pro dolování. Načtená data mohou obsahovat nesprávné či chybějící hodnoty, které bývají nahrazeny vhodnými hodnotami ze statistických postupů, případně mohou být chybné záznamy vymazány úplně. Zkoumání se také zaměřuje na data výrazně se odlišující od dat ostatních - tzv. odlehlé objekty.

3. **Transformace dat:** Zde probíhá konverze dat do formátu vhodného pro dolování a také konverze dat z různých zdrojů. Transformace také zahrnuje redukci dat, kde je rozhodnuto o odstranění atributů, které dle expertů v žádném případě nesouvisejí s hledanými informacemi, naopak by dolování mohly negativně ovlivnit. Dále mohou být atributy substituovány, příp. transformovány pomocí matematické funkce.

4. **Dolování z dat:** Hlavní krok procesu - jsou aplikovány nejrůznější algoritmy, vytvářeny modely a následné generovány požadované výsledky dolování.

5. **Vyhodnocení a prezentace výsledků:** Vizualizace výsledků stejně jako zkoumání vstupních dat u předzpracování jsou velice důležité. Samotnou vizualizaci bych označil jako další obor související s dolováním, jelikož analytici musí přesvědčit uživatele o pravdivosti svých výsledků.

Jednotlivé kroky procesu získávání znalostí z databází se opět skládají z dalších dílčích úloh a vypořádávají se s různými problémy. Zároveň příprava dat zpravidla zabere mnohem delší časový úsek než samotné dolování. Proto níže budou představeny přípravné fáze dat podrobněji.

Problematika dolovací fáze a různé typy úloh, se kterými se analytik vypořádává, jsou popsány v kapitole 4. Predikování spojitých hodnot, na které se zaměřuje tato bakalářská práce, je teoreticky popsáno níže (kap. 7).

3.1 Výběr dat

Vůbec prvním krokem procesu je načtení datových objektů. Objekty obsahují údaje o jednotlivých vzorcích (záznamech). Jsou popisovány množstvím atributů (proměnných, vlastností) zachycujících základní charakteristiku vzorku. Atributy dělíme na kategorické (kvalitativní) - pohlaví, barva očí, hodnocení tvrdosti minerálů, a numerické (kvantitativní) - částka, počet obyvatel. Také mohou být atributy děleny pomocí počtu nabývajících hodnot na diskrétní (konečná množina hodnot) a spojité (nekonečná množina hodnot).

Jakákoliv množina dat může být popsána pomocí tří základních charakteristik [21]. Jsou pak určující pro výběr dolovacích technik použitých v dalších krocích procesu dolování.

- **Dimenzionalita** - neboli také rozměr datového souboru. Jedná se tedy o počet atributů popisujících jednotlivý záznam. V případě vysoce dimenzionálních množin je často nutná redukce dat, jelikož atributy mezi sebou bývají redundantní.
- **Řídkost** (sparsity) - některé atributy nabývají hodnot pouze 0 a 1 (případně jakékoliv nenulové číslo), častokrát tehdy, když určují výskyt v dané kategorii. Důležité jsou pouze nenulové hodnoty, kterých je minimum. Nulové hodnoty nemusí být ukládány. Tímto ušetříme výpočetní čas a místo v úložišti.
- **Rezoluce hodnot** - pokud jsou data načítána z více zdrojů, atributy mohou být sice stejné, ale jednotky se mohou lišit. Je nutné je sjednotit, aby případné vzory ukryté v datech nevymizely.

Po fázi načtení a seznámení se s daty přichází na řadu předzpracování.

3.2 Předzpracování

Část předzpracování dat se zaměřuje především na dva základní problémy. Za prvé musí být data dostupná v takovém formátu, aby s nimi dokázaly pracovat dolovací algoritmy. A za druhé je nutné, aby vedla k co nejlepšímu výkonu a kvalitě modelů [2].

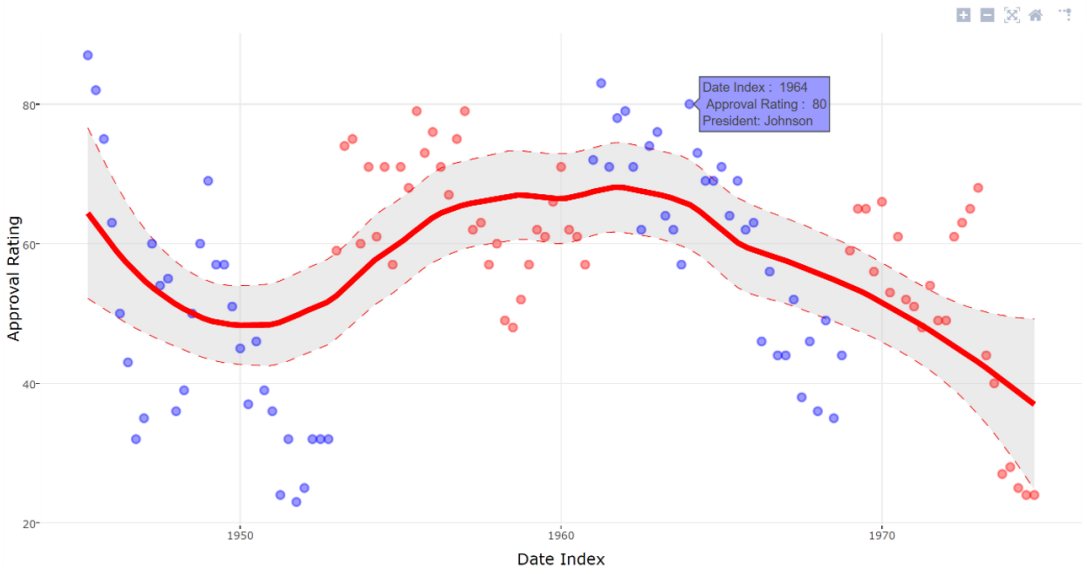
Čištění dat

Požadované vlastnosti vedoucí k dobré **kvalitě dat** jsou následující: **přesnost**, **kompletnost** a **konzistence** [9]. Tyto vlastnosti obvykle v prvotní fázi nesplňují data velkých databází a datových skladů. Zašuměné hodnoty jsou zpravidla na první pohled nesprávné, u nekonzistentních dat se mohou objevit redundantní atributy, případně hodnoty některých atributů jsou zapsány v jiných formátech. Tyto chyby mohou být způsobeny chybou lidského faktoru, poruše při sběru dat nebo právě použití více datových zdrojů [27]. Nedostatky se řeší v rámci **čištění a transformace dat**.

Metody čistění dat pro chybějící hodnoty:

- **Odstranění záznamu nebo atributu**
- **Manuální nahrazení** - nutná znalost domény
- **Automatické nahrazení** - např. konstantou, průměrem, průměrem v rámci příbuzných záznamů, interpolace, použitím algoritmů pro klasifikaci (predikci)

Přístupy pro nakládání se zašuměnými daty využívají již metody používané při fázi dolování. Toto je jedna z příčin, proč proces získávání znalostí z databází pracuje v iteracích. Pro vyhlazení zašuměných dat se používá např. **plnění** (binning), **regrese**, **shlukování** [27], podrobněji v 4. Dále se využívají také numerické metody z matematických a statistických analýz, příkladem může být aproximační křivka spline (na obrázku 3.2).



Obrázek 3.2: Znázornění vyhlazení zašuměných dat a nahrazení technikou Spline. Zdroj: https://appsource.microsoft.com/en-us/product/power-bi-visuals/WA104380860

3.3 Transformace

Rovněž techniky transformace slouží k přesnějším výsledkům dolovacích modelů. Některé zdroje uvádějí, že transformace je součást předzpracování dat. Cíl obou kroků je stejný. Ve své práci jsem se však rozhodl použít rozdělení dle Margaret H. Dunham [4]. Samozřejmě se ale metody obou kroků prolínají.

Do kroku transformace mohou být zahrnuty nejrůznější techniky. Ty nejdůležitější jsou uvedeny níže.

Integrace dat

Nesrovnalosti schémat dat z různých zdrojů řeší integrace dat. Popis schémat může být zprostředkován za pomocí metadat.

Definice 3.2 *Metadata jsou jednoduše definovaná jako data o datech. Jsou určená k popisu dat, se kterými jsou spjaty. [25].*

Obsahem těchto metadat jsou jména atributů a datový typ, případně přípustné rozsahy hodnot.

Především různé formáty či unikátnost dat jsou body, na které je nutné si dát pozor. Automatizované nástroje v této oblasti jsou omezené, proto častokrát musí být záznamy upraveny ručně [27].

Agregace

Někdy méně znamená více. Zkombinování dvou či více objektů do jednoho může být někdy na škodu, někdy naopak může dopomoci k výrazně lepším výsledkům. Menší množství

záznamů rovněž znamená menší nároky na paměť a výpočetní čas [21]. Např. výpočet standardní odchylky pro jednotlivé měsíce datové množiny je zašumělý, při agregaci na celé roky se objevuje pravidelná logaritmická funkce.

Pro agregaci spojitých atributů se často využívá suma či průměr. U těch kategorických mohou být data vynechána.

Normalizace

Dolovací metody normalizaci často vyžadují. Především je to nutné u algoritmů klasifikace založené na vzdálenosti a neuronových sítích. Nejinak tomu je i v případové studii - 8. Data jsou namapována do specifického rozsahu, vycentrována typicky do hodnoty 0. Zachovává se poměr mezi původními hodnotami [27].

Vytvoření nových popisných atributů a jejich selekce

Jsou případy, kdy zdrojová data obsahují i stovky atributů. Mnoho z nich nemusí být vůbec relevantní pro řešení dané úlohy. U některých úloh je důležité vytvoření nových vlastních popisných atributů (features). V případové studii této práce predikce probíhá výhradně na základě nově vytvořených atributů.

Vyhodnocování nejdůležitějších atributů a zároveň vyloučení nepoužitelných lze v literatuře najít pod pojmem **analýza relevance** (relevance analysis) [9]. Pro výběr nejvhodnějších atributů lze využít heuristické techniky (postupný dopředný výběr, zpětná postupná eliminace atd.) nebo algoritmy rozhodovacích stromů [27].

Diskretizace a binarizace

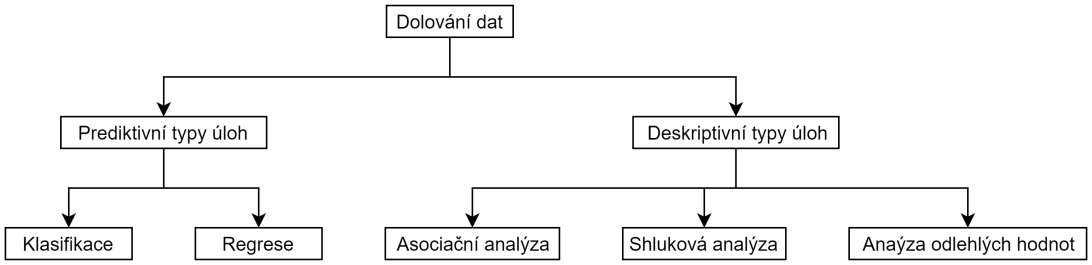
Opět se jedná o metody, díky kterým dosahují klasifikační algoritmy přesnějších výsledků. Diskretizace provádí transformaci spojitých atributů na diskrétní (např. ceny nemovitostí se podle nějakých intervalů převedou na hodnoty levný, průměrný, drahý nebo na číselnou podobu 1 - 3). U binarizace jsou převáděny jak hodnoty spojité, tak i diskrétní na nulovou a nenulovou hodnotu [21]. Typicky se jedná o atributy ve formátu ano/ne na formát 0/1.

Kapitola 4

Typy dolovacích úloh

Není překvapením, že existují různé typy dat a repozitářů, v návaznosti na to se objevuje i velké množství dolovacích úloh (data mining functionalities). Ty se liší kromě vstupních dat i typy modelů a vzorů, které v datech vyhledávají. Systémy musí být také dostatečně univerzální, jelikož v praxi se často stává, že uživatelé neví, jaký model a jakými způsoby daný model tvořit [27].

V následujících podkapitolách bude nastíněna většina typů dolovacích úloh a jejich příklady. Všechny tyto úlohy se pak provádějí v kroku dolování procesu KDD (kapitola 3) [27]. Rozdělení přehledně znázorňuje obrázek 4.1.



Obrázek 4.1: Typy dolovacích úloh. Převzato z [4]

Jak již obrázek 4.1 napovídá, úlohy jsou podle účelu dolování rozděleny na prediktivní a deskriptivní. V následující sekci jsou uvedeny ty prediktivní. Druhý typ úloh popisující vlastnosti a vztahy na existujících datech naleznete v sekci 4.2.

4.1 Prediktivní typy úloh

Cílem těchto úloh je predikování hodnot konkrétního atributu na základě hodnot jiných atributů [21]. Může se jednat buď o situaci, kdy díky datům z jiných zdrojů je vypočtena požadovaná informace, která v jiném zdroji chybí, nebo předpovědět budoucí chování pomocí analýzy historických dat [27].

Predikovaný atribut je často označován jako **cílová nebo závislá proměnná** (target or dependent variable), zatímco atributy použité pro predikci jsou nazývány jako **vysvětlující nebo nezávislé proměnné** (explanatory or independent variables) [21].

V některých publikacích i díky nešťastnému používání slov jsou popisy pojmů klasifikace, regrese a predikce zmatené, zavádějící (např. [4]). V této práci budu dle svých zkušeností odlišovat tyto tři termíny takto:

- **Klasifikace** - slouží k **predikci** hodnot kategorických (diskrétních, neuspořádaných) [27]
- **Regrese** - slouží k **predikci** hodnot spojitých [27]
- **Predikce** - zobecnění obou termínů - zahrnuje jak klasifikaci, tak regresi [9]

Modely se na určitém vzorku dat naučí vlastnosti a vztahy tohoto vzorku, které pak se snaží aplikovat na jiných datech, kde už model nemá k dispozici cílovou proměnnou, naopak se ji snaží predikovat.

Fáze predikce

Prediktivní typy úloh využívají metodu **strojového učení s učitelem** (supervised learning), tudíž **předpokladem jsou vstupní data se známými hodnotami cílové proměnné**. Při tvorbě každého modelu, který posléze bude sloužit k predikování hodnot, se prochází dvěma hlavními fázemi, typicky opakovaně v cyklu.

Vstupní data se známou hodnotou cílového atributu jsou rozdělena na trénovací, testovací a v některých případech i na validační množinu.

První část se nazývá **fáze učení**. Z **trénovací množiny** model poznává strukturu dat a „učí se“ vztahy mezi nimi. Na základě těchto dat je pak vytvořen model.

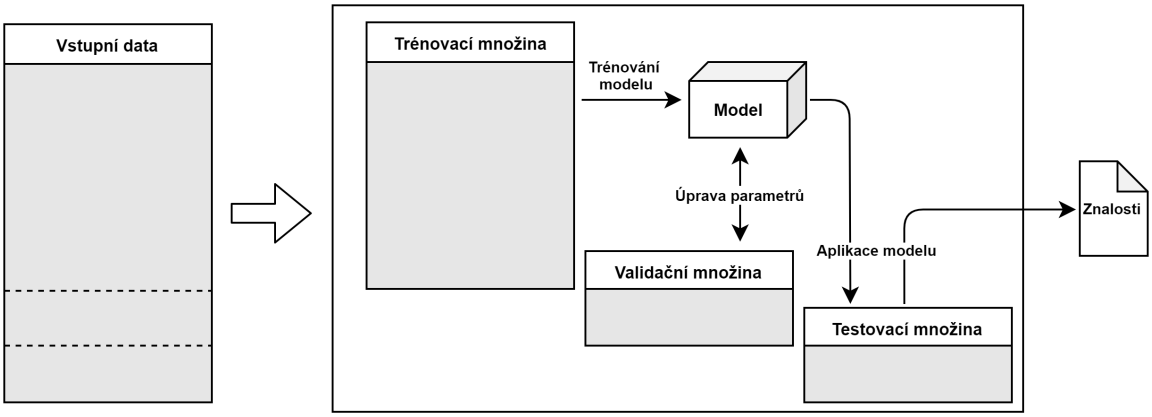
Vytvořený model vyhodnotí hledaný atribut na datech v **testovací sadě** bez jeho znalosti během **fáze testování**. Hodnoty jsou pak porovnány se skutečnými výsledky. Podle dosaženého skóre přesnosti poté analytik vyhodnotí, zda je model použitelný, případně použije model jiný.

Je nutné, aby data byla rozdělena rovnoměrně. Data v trénovací množině by měla být co možná nejreprezentativnější, aby vzniklá tzv. **klasifikační pravidla** byla co nejvíc komplexní [27]. Proto existují přístupy, kdy klasifikátor v iteracích zvolí vždy jiné trénovací a testovací množiny, aby následně mohl vybrat pro finální natrénování ty množiny, které prokazovaly nejlepší výsledky na testovacím souboru dat (např. křížová validace).

Každý vzniklý model má spoustu parametrů, které může analytik nastavit. Jedním z nich je **relativní chyba** („procentuální podíl chybně predikovaných instancí vůči mohutnosti celé uvažované množiny instancí“) [26]. Ta musí být nastavena vyváženě, jinak je model sice přesný na testovacích datech, ale klasifikační pravidla jsou pak moc konkrétní a nemohou být použita na jiných datech. Dojde k tzv. **přeučení modelu** (overfitting), které již bylo zmíněné v 2.5.

Tím se zabývá **validační množina** sloužící k nejvhodnějšímu nastavení parametrů pro daný problém. Není ale nutností. Modely si mohou upravovat své parametry již v rámci trénovací množiny. Validační množina se používá např. u prořezávání již hotového stromu [26].

Pro lepší pochopení vztahy dělení vstupních dat pro predikci popisuje i obrázek 4.2.



Obrázek 4.2: Rozdělení dat v případě prediktivních typů úloh

Klasifikace

Cílem v typické klasifikační úloze je rozdělení záznamů dat do jednotlivých skupin (tříd - classes). Důležitou vlastností je, že **cílová proměnná** je **diskrétního**[1] charakteru. Typicky budou data rozdělena do tříd podle vypočtené hodnoty cílové proměnné.

Vytvořený model může být sestaven za pomocí mnoha technik (klasifikační pravidla, rozhodovací stromy, Naive Bayes Classifier, neuronové sítě aj.). Typickým příkladem klasifikace je rozdělení klientů banky do třídy těch, kterým může být případně poskytnut úvěr, a do třídy opačné. Klasifikací je i predikování záplav na základě aktuálního stavu řek.

Zvláštním případem mohou být úlohy, kde důležitou roli hraje čas. Data obsahují typicky časový údaj a hodnoty atributů v daném čase. Jedná se o **analýzu časových řad** (time series analysis). Zde na rozdíl od jiných klasifikačních úloh jsou data seřazena a sousední hodnoty mají mezi sebou určitý vztah.

Regrese

Cílem regresní úlohy je **predikce hodnot spojitého atributu** na základě historických a aktuálních dat. Jednou z technik zpravidla používaných u regrese je regresní analýza. Jedná se o systém metod a funkcí, ze kterých jsou vybrány ty nejlépe popisující vývoj daného atributu [4].

Příkladem může být výpočet nové akční ceny pro výrobek v nákupním centru na základě aktuálního stavu ve skladu, poptávce a požadovaného zisku.

Stejně jako u klasifikace i zde se můžeme setkat s analýzou časových řad.

Zde je zmíněna klasifikace a regrese jen okrajově. Jelikož se jedná o hlavní téma této práce, je tato problematika dopodrobna popsána v kapitole 7.

4.2 Deskriptivní typy úloh

V těchto úlohách jsou vytvářeny deskriptivní modely, které hledají v datech vzory (korelace, trendy, anomálie) a vzájemné souvislosti. Tyto úlohy se často snaží vysvětlit vztahy a zákony z přírody a vyžadují následně důkladné zpracování výsledků [21]. Popisují do hloubky již

[1] diskrétní cílová proměnná - hledaný atribut dolování je výčtového typu, opakem je spojitý atribut u regrese [21]

existující data, nejedná se o predikování dat do budoucna, jak je to u prediktivních úloh (viz kapitola 4.1).

Asociační analýza

V rámci této analýzy je vytvářen model identifikující specifické druhy asociací dat. Nalezené vzory jsou typicky reprezentovány formou asociačních pravidel jako na příkladu 4.1. Výsledná pravidla se nemusí shodovat s realitou, tedy mohou spojovat objekty a jejich vlastnosti, které spolu v reálném světě v žádném případě nesouvisí. Nicméně přesto v rámci dané úlohy může být takové pravidlo stěžejní [4].

$$sportovec(X,'ano') \wedge bydliste(X,'Ostrava') \Rightarrow oblibenyKlub(X,'BanikOstrava') \\ [podpora = 2\%, spolehlivost = 65\%] \tag{4.1}$$

Pokud se v pravidle nachází více než jeden predikát, jedná se o **multidimenzionální asociační pravidlo**. V případě jednoho je **jednodimenzionální** [27].

Z důvodu exponenciálního růstu prohledávaného prostoru je cílem nalézt nejzajímavější vzory efektivní cestou [21].

Analýza nákupního košíku v internetových obchodech je velice známým příkladem této analýzy. Pokud uživatel přidá do košíku nějakou položku, jsou mu nabízeny i další produkty, které patrně s již vloženým produktem souvisejí (tzn. byly řadou zákazníků vloženy také).

Shluková analýza

Shluková analýza (anglicky clustering) se podobá klasifikaci, na rozdíl od ní ale nejsou předdefinované třídy, do kterých by měla být data rozdělena. V řeči strojového učení se jedná o učení bez učitele (unsupervised learning) [4]. Objekty patřící do jednoho shluku jsou si mnohem více podobné než objekty nacházející se ve dvou odlišných shlucích. Shluky mohou nebo nemusí být disjunktní.

Příkladem shlukové analýzy může být např. přiřazení novinových článků do jednotlivých shluků podle vyskytujících se slov a jejich počtu. Výsledné shluky by obsahovaly pouze články stejných témat (ekonomie, zdravotní péče) [21].

Analýza odlehlých hodnot

Takřka opačný cíl oproti shlukové analýze má **analýza odlehlých hodnot** (outliers analysis), také nazývaná jako analýza anomálií (anomaly analysis). Zde se jedná o nalezení hodnot, které se výrazně liší od chování drtivé většiny dat. Hledání probíhá pomocí statistických testů (distribuční funkce, pravděpodobnostní modely), měřením vzdáleností nebo metod zaměřených na měření hustoty [9].

Rozsah použití těchto typů úloh je velký, od informačních technologií (detekce útoků na server) přes bankovnictví (podvodné použití odcizených kreditních karet) až po zdravotnictví (detekování chorob na základě testů).

V těchto případech se klade velký důraz na vysokou míru detekce odlehlých hodnot a minimální míru chybovosti [21]. Pokud je vzato jako příklad detekování chorob, je nutné odhalit jejich maximální procento (mohou být pro neléčené pacienty smrtelné) a zároveň chybné určení choroby u zdravého pacienta způsobí, že následná vyšetření mohou být pro člověka zátěží.

Kapitola 5

Podpora dolování dat v jazyce Python

Obsahem této kapitoly bude představení nástrojů určených k praktickému využití v rámci dolování dat v jazyce Python. S rostoucí poptávkou po analyticích (v USA až o 28% do roku 2020, [8]) v této oblasti rostou i možnosti použití programovacích jazyků pro tyto úlohy. Mezi nejvíce využívané jazyky pro dolování dat patří dvojice Python a R. S menším odstupem následuje Java, SQL, Julia a Scala [1].

Jelikož případová studie této práce byla vypracována v jazyce Python, bude zde představen jazyk jako takový, jeho moduly pro analýzu dat a webová aplikace Jupyter Notebook.

5.1 Jazyk Python

Rychle se rozvíjející, dynamicky interpretovaný, objektově orientovaný, vysokoúrovňový skriptovací jazyk - Python[1].

Byl vytvořen jako open source projekt v roce 1991 Guidem van Rossumem, který název převzal z anglického televizního seriálu Monty Python's Flying Circus [20].

O téměř 30 let později se podle TIOBE indexu založeném na počtu zadaných dotazů skrze nejrozšířenější internetové vyhledávače jazyk Python pravidelně umísťuje mezi první pěticí všech programovacích jazyků [23].

Stal se oblíbeným především díky svému rozsahu použití a přehledné syntaxi, která je taky způsobena raritním odsazováním pomocí mezer oproti složeným závorkám v jiných jazycích. Je dostupný na běžných platformách, ve většině distribucí systému Linux ho lze nalézt jako součást základní instalace.

Díky rychlému vývoji se dnes (v roce 2019) lze setkat s verzemi 2.x a 3.x, které nejsou mezi sebou kompatibilní. Python 2.x v roce 2020 přestane být podporován. Z verze 3.x byla poslední vydaná verze 3.7.2 v prosinci roku 2018.

V jazyce Python bylo implementováno mnoho projektů z nejrůznějších IT oblastí, např. frameworky Django a Pyramid pro vývoj webů, nástroje pro vývoj software, hry a webové služby (Battlefield, Dropbox, UBER, Pinterest) [20].

Jak bylo zmíněno výše, Python je open source projekt. To umožnilo vývojářské komunitě vytvářet své moduly a volně je nabízet ostatním. Tyto moduly jsou jednoduše dostupné přes repozitář PyPI (Python Package Index), kde jsou k dispozici i nástroje pro potřeby dolování, které dělají Python vhodný pro tuto oblast.

[1] https://www.python.org/

5.2 Jupyter Notebook

Jupyter Notebook[2] je užitečný nástroj sloužící především k přehledné prezentaci výsledků a sdílení kódů mezi osobami. Webová aplikace s konzolí Pythonu, ale i s podporou dalších jazyků, kde lze kombinovat bloky kódu, výstupy z konzole a případně i přidávat vysvětlivky v LaTeX. Výsledky studií lze jednodušeji prezentovat zákazníkům, kteří nejsou IT specialisté, a mohou být z Jupyter Notebook exportovány do formátů PDF, HTML aj.

5.3 NumPy

Jedná se o balík nástrojů společně s balíkem SciPy[3] pro práci s numerickými daty a vědecké výpočty v Pythonu. Základním kamenem NumPy[4] je datový objekt homogenního multidimenzionálního pole `ndarray`. Jeho velikost je pevně daná již při inicializaci, na rozdíl od jiných polí používaných v Pythonu [16].

Současně balík obsahuje i mnoho metod pro práci s tímto datovým typem, které se jednoduše a intuitivně dají použít na všechny členy ndarray. Mohou to být jednoduché matematické operace, ale také práce s polynomy, vzorkováním, lineární algebrou a také diskrétní Fourierova transformace. Zároveň jsou důležité metody pro generování, řazení, hledání či práci se soubory.

Většina ostatních modulů pracujících s maticemi či tabulkami je postavena na NumPy nebo podporují ndarray objekt.

5.4 Pandas

Na modulu NumPy je postaven i Pandas[5]. Tento populární balík pro analytiky nabízí účinné a flexibilní datové struktury, které usnadňují manipulaci s daty a analýzu. Podobně jak NumPy má i Pandas svůj datový objekt - `DataFrame`. Jedná se prakticky o ekvivalent tabulky. Má integrované indexování a je složen ze `Series` představující sloupec tabulky, jak je znázorněné na obrázku 5.1.

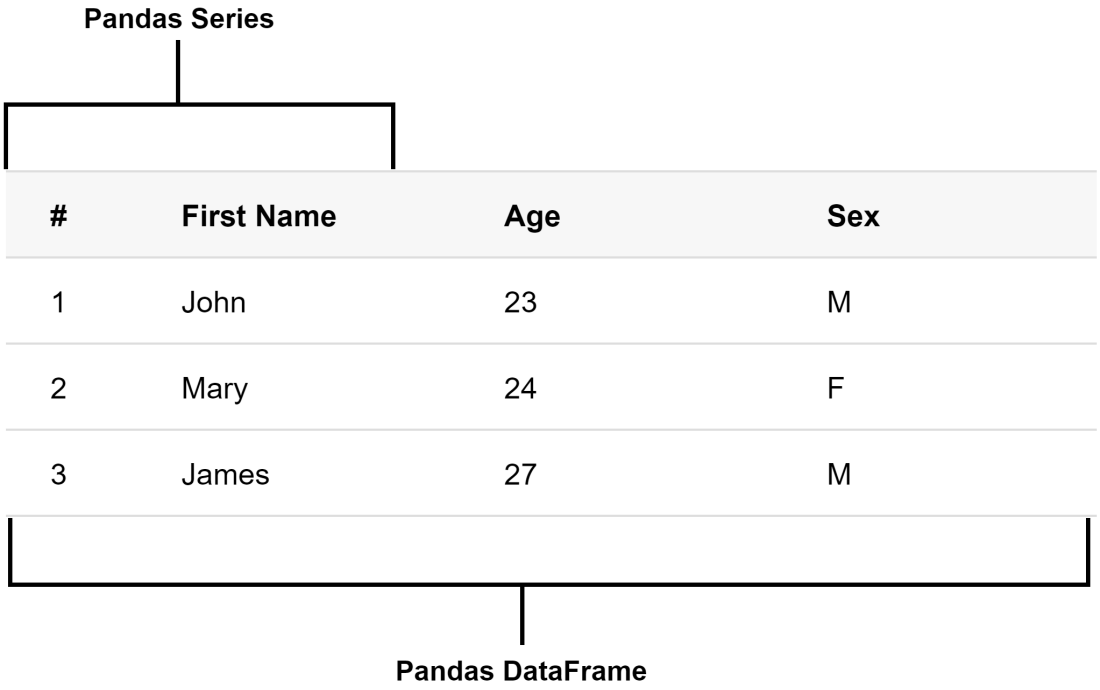
Mezi přednosti tohoto modulu patří především načítání a zapisování dat z, resp. do textových souborů, formátů tabulkového editoru Microsoft Excel, případně databází. Dále poskytuje nástroje pro předzpracování dat jako je práce s chybějícími daty, transformace, vytváření podmnožin, indexování, seskupování a tvorba sumarizujících statistik.

5.5 Scikit-learn

Pokud by měl být zmíněn jeden modul, bez kterého by se v jazyce Python dolování dat neobešlo, byl by to Scikit-learn[6]. Nabízí mnoho principů z oblasti strojového učení.

- **Repozitář cvičných či reálných datových množin**
- **Metody předzpracování dat a transformací** - extrakce atributů, náhodná projekce, aproximace, zřetězení více nástrojů, normalizace

[2] https://jupyter.org/
[3] https://www.scipy.org/
[4] https://www.numpy.org/
[5] https://pandas.pydata.org/
[6] https://scikit-learn.org/



Obrázek 5.1: Pandas Series a DataFrame

- **Strojové učení s učitelem** - výběr atributů, algoritmy SVN, nejbližších sousedů, neuronových sítí, metody pro posílení robustnosti modelů aj.
- **Strojové učení bez učitele** - shluková analýza, rozklad signálů, analýza odlehlých hodnot, odhady hustoty aj.
- **Výběr modelů a vyhodnocení** - křížová validace, ladění parametrů modelů, vyhodnocení kvality predikování atd.

5.6 Ostatní použité knihovny

Moduly zmíněné níže sice již nejsou jádrem podpory pro dolování v Pythonu (u Matplotlibu by se dalo o tom polemizovat), případně jejich rozsah použití není tak komplexní. Nicméně jsou zde uvedeny, protože byly použity pro případovou studii. Všechny podporují či používají moduly NumPy a Pandas, kromě Alpha vantage jsou všechny opět dostupné pod BSD licencí.

Matplotlib

Jak již název napovídá, modul Matplotlib[7] zprostředkovává nástroje pro nejrůznější vizualizace. Objektově orientované uživatelské rozhraní lze použít jak v Pythonu, tak v MATLABu. Možnosti balíku jsou obrovské, zahrnují např. veškeré typy grafů, teplotních map, práci a transformaci grafických objektů.

[7] https://matplotlib.org/

Py-earth

Py-earth[8] zprostředkovává implementaci modelu MARS (Multivariate Adaptive Regression Splines) neboli vícerozměrných adaptivních regresních splajnů. Model bude podrobněji představen níže - 7.2.3.

TA-Lib a Alpha Vantage

Oba moduly se zaměřují na technické analýzy údajů o finančním trhu. Budou použity k získání hodnot popisných atributů pro predikci.

TA-Lib[9] zahrnuje okolo 200 indikátorů používaných jak finančními analytiky, tak vývojáři software v této oblasti. Rozhraní je multiplatformní, opět volně dostupné.

Alpha Vantage[10] je již komerční rozhraní, oproti TA-Lib však nabízí jednoduše získat historická i aktuální data akciových trhů, kurzů cizích měn a kryptoměn. Bohužel ale základní verze rozhraní vyžaduje autentizaci, po které je vydán API klíč. Navíc počet dotazů na rozhraní je časově limitován.

[8] https://contrib.scikit-learn.org/py-earth/
[9] https://www.ta-lib.org/
[10] https://www.alphavantage.co/

Kapitola 6

Úvod do problematiky případové studie

Předmětem této kapitoly je úvodní seznámení s tím, co je cílem případové studie, vysvětlení její domény a pohled na finanční analýzy jako takové.

Případová studie byla inspirována výzkumem profesora Luise Torga (Univerzita Porto) v jeho titulu „Data Mining with R“ [24], kde tuto studii praktikoval v jazyce R.

Součástí je i návrh postupu řešení celé studie, zřetel bude kladen na hlavní úkoly, které mohou být specifické pro danou konkrétní úlohu.

Po tomto úvodu bude následovat v kapitole 7 teoretické vysvětlení stěžejních metod, principů a modelů použitých v studii, aby následně mohly být tyto principy použity a ukázány výsledky v kapitole 8.

6.1 Finanční analýza

Predikce vývoje finančních trhů je spolu s např. úvěrovou politikou bank, profilováním zákazníků, predikcí bankrotů či zjišťováním praní špinavých peněz jednou z mnoha úloh finanční analýzy. Data v této množině jsou obvykle podrobně zpracována a úspěšné techniky dolování dat v tomto odvětví zpravidla vedou k velkým ziskům. Proto je zde potenciál vysoký [12].

Zároveň jako protiklad hovoří hypotézy ekonomů, které se snaží dokázat, že chování finančních trhů je nevyzpytatelné, rychle se měnící a že neexistuje prostor, aby v dlouhodobém měřítku byly prognózy úspěšné a profitující [24].
Aplikace dolování dat na doménu finanční analýzy skrývá následující specifika [12]:

- predikování mnoharozměrných časových řad s vysokou mírou zašumění
- odlišné vyhodnocovací metriky (např. maximální výnos)
- nutnost vysvětlení modelů pro predikci osobám zodpovědným za významná investiční rozhodnutí
- schopnost pracovat s útlými vzory v datech v krátkém časovém úseku
- zohlednit dopad účastníků trhů na jeho vývoj

Nicméně touha po zisku analytiků je velká, tudíž není nouze o analýzy tohoto typu. Navíc se objevila nová velmi specifická doména, kde výše zmíněné prognózy nemusí platit. Jsou to kryptoměny.

Predikování vývoje indexu S&P 500

Předmětem zájmu případové studie je vývoj akciového trhu na americké burze. Podle rozdělení v kapitole 4 je to typ úlohy **prediktivní**, obsahem je **jak klasifikace, tak regrese**. Hlavní myšlenkou je nalézt určité vzory v historických datech indexu S&P 500 a na základě těchto dat se pokusit co nejpřesněji predikovat hodnoty budoucí.

S&P 500 (Standard & Poor's 500 v nezkrácené verzi) je index amerických akciových trhů, který zohledňuje tržní kapitál 500 největších společností s akciemi na burzách NYSE nebo NASDAQ. Tržní hodnota těchto 500 korporátních společností nesmí klesnout pod 6 miliard amerických dolarů a objem veřejných akcií musí přesáhnout hodnotu 250 000 za měsíc. Váhy akcií jednotlivých podniků v indexu se určují podle jejich kapitálu. Zhruba desetinu S&P 500 ovlivňují 3 největší společnosti - Microsoft, Apple a Amazon. Díky těmto dobře nastaveným podmínkám je index jedním z nejpoužívanějších amerických indexů [11].

6.2 Vstupní data

Vstupními daty případové studie jsou tedy hodnoty zmíněného indexu S&P 500. Každý záznam představuje údaje o indexu v rámci 1 pracovní dne. Jedná se tím pádem o analýzu časových řad.

Struktura vstupních dat má následující podobu:

Date	Open	High	Low	Close	Volume	AdjClose
1970-01-02	92.06	93.54	91.79	93.00	8050000.0	93.00
1970-01-05	93.00	94.25	92.53	93.46	11490000.0	93.46
1970-01-06	93.46	93.81	92.13	92.82	11460000.0	92.82
1970-01-07	92.82	93.38	91.93	92.63	10010000.0	92.63
1970-01-08	92.63	93.47	91.99	92.68	10670000.0	92.68

Popis jednotlivých atributů vstupních dat:

- **Date** - datum pracovního dne
- **Open** - hodnota indexu na začátku dne
- **High** - nejvyšší dosažená hodnota v rámci dne
- **Low** - nejnižší dosažená hodnota
- **Close** - hodnota indexu na konci dne
- **Volume** - objem transakcí
- **AdjClose** - hodnota indexu na konci dne upravená akcemi účastníků trhu v rámci aktuálního dne (pro tuto studii nepodstatná)

Hodnoty indexu S&P 500 jsou lehce dostupné skrze různé finanční webové portály, typicky se index zobrazuje pomocí svíčkových grafů[1] ukazující všechny atributy S&P 500.

[1] Candlestick chart - viz např. https://www.tradingview.com/chart/?symbol=OANDA%3ASPX500USD

6.3 Postup řešení úlohy případové studie

Tato sekce obsahuje návrh postupu řešení predikování indexu S&P 500. Budou zde zmíněny základní kroky implementované v praktické části jako jednotlivé body procesu získávání znalostí z databází (kapitola 3). Rovněž by měl tento nástin sloužit k pochopení souvislostí a společných vazeb jednotlivých kroků.

Řešení případová studie se bude skládat z kroků, jak to znázorňuje obrázek 6.1.

Výběr dat

Vstupními daty jsou historické hodnoty indexu S&P 500, jak již bylo zmíněno výše. Data z období mezi roky 1970 až 2009 jsou načteny ze souboru dostupného na webových stránkách[2]. Pro následné období od září 2009 do roku 2019 bylo využito rozhraní Alpha Vantage, které získává z webu aktuální údaje. Rovněž studie naznačila i postup načítání dat z databáze. Ale vzhledem k tomu, že údaje v databázi jsou totožné jak v souboru, není s nimi nadále pracováno.

Předzpracování dat

Načtené hodnoty neobsahují žádná nesprávná či chybějící data. Naopak transformace dat je pro tuto úlohu jedna ze stěžejních.

Transformace dat

Hned prvním bodem transformace je sjednocení načtených dat ze souboru a z webu. Cílovým atributem při regresním modelování je vytvořený indikátor spojitého typu, který určuje následný trend vývoje trhu. V případové studii se ale řeší i klasifikační úlohy. V tomto případě se převedou hodnoty indikátoru podle určených prahových hodnot na signály Sell, Hold a Buy, cílový atribut je tedy kategorický. Pro jeho predikci bude sloužit velké množství dalších vlastností vytvořených pomocí různých statistických metod (směrodatná odchylka, zpoždění časových řad, průměr posuvného okna pro poslední dny) a především technických identifikátorů, které jsou specifické pro oblast analýzy finančních trhů.

Jejich důležitost a korelaci k indikátoru určí **metoda náhodných stromů** (Random Forest) jako jedna z technik fáze dolování. Zde se ukazuje vzájemná iterace jednotlivých kroků v procesu. Pouze vlastnosti s nejlepšími výsledky jsou nadále využívány.

V posledním kroku jsou zahozeny i hodnoty indexu S&P 500. Do kroku dolování tedy vstupuje pouze časová řada posledních 50 let každodenních údajů o 8 nejlepších popisných atributech a cílovém indikátoru.

Dolování z dat

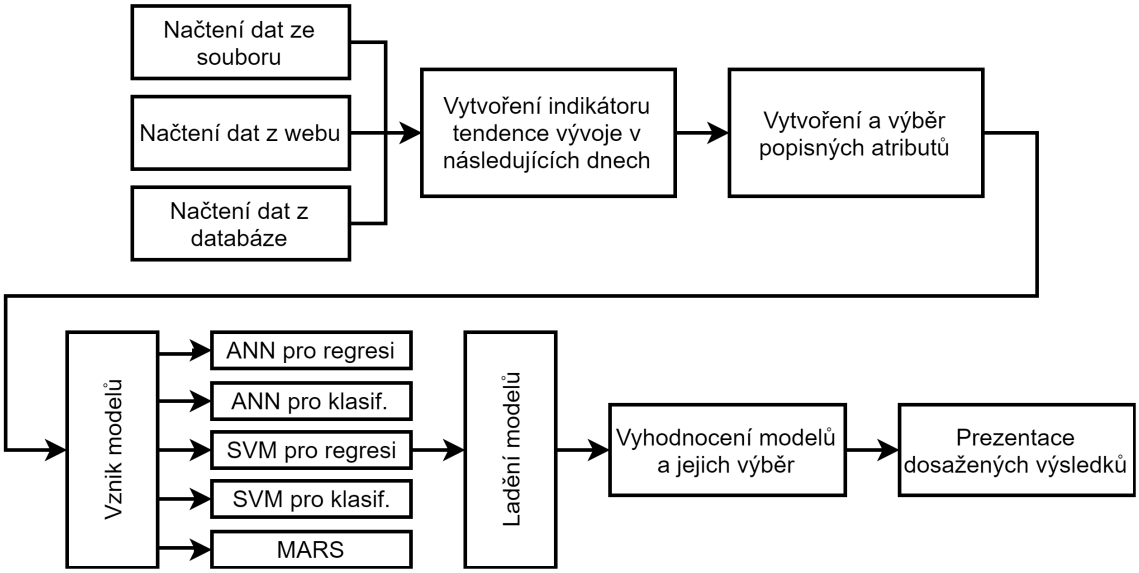
Data jsou nyní připravena pro hlavní krok procesu. Pro regresní predikování hodnot indikátoru jsou použity algoritmy neuronových sítí, metody podpůrných vektorů a mnohorozměrný adaptivní regresní spline (MARS). Zároveň jsou použity i klasifikační varianty neuronových sítí a metody podpůrných vektorů v případech, kdy byl indikátor transformován na signály Sell, Hold a Buy pro aktivity na trhu.

[2] Webové stránky k publikaci Data Mining with R - http://www.dcc.fc.up.pt/~ltorgo/DataMiningWithR/

Vyhodnocení a prezentace výsledků

Pro vyhodnocování modelů bude sloužit metoda Monte Carlo, která vyhodnotí velkým množstvím pokusů na různých časových obdobích jednotlivé přístupy a modely pro predikování.

Nejlepší modely budou popsány podrobněji pomocí dalších vyhodnocovacích metrik.



Obrázek 6.1: Znázornění konceptu řešení úlohy případové studie

Kompletní popis a forma implementace studie se nachází v kapitole 8. Následující kapitola obsahuje podrobnější teorii jednotlivých kroků, metod a modelů použitých v případové studii.

Kapitola 7

Predikce

V předchozí kapitole bylo nastíněno, z jakých částí se případová studie skládá. Jedná se tedy o prediktivní typ úlohy. Tato kapitola se zaměří na teoretické aspekty postupů použitých ve studii. Kromě specifických rysů analýzy časových řad budou vysvětleny principy všech použitých predikčních modelů a metoda pro jejich vyhodnocování (sekce 7.2, resp. 7.3).

7.1 Analýza časových řad (time series)

Specifikem popisované analýzy je fakt, že každý jednotlivý záznam obsahuje údaj o čase, kdy byly hodnoty naměřeny. Tato sekce popíše speciální vlastnosti časových řad.

Časová řada je tedy množina hodnot atributů v nějakém časovém intervalu. Formálně lze popsat vztah takto:

Nechť existuje atribut A, časová řada je množina n hodnot $\{< t_1, a_1 >, < t_2, a_2 >, ..., < t_n, a_n >\}$. Pro každou korespondující hodnotu A existuje n hodnot času. Na hodnoty atributu může být nahlíženo jako na vektor $< a_1, a_2, ..., a_n >$. Převzato z [4].

Typické analýzy časových řad obvykle zahrnují některé či všechny tyto body: podobnost dvou různých řad, identifikace vzorů v rámci řady a predikce budoucích hodnot.
Obvykle lze zpozorovat v tomto typu dat následující vzory:

- **Trendy** - systematické neopakující se změny hodnot atribůtu v čase, typicky se jedná o dlouhodobější růst či klesání
- **Sezónní vzory** - založené na specifickém časovém období, opakující se převážně každý rok
- **Cykly** - další periodicky se opakující vzory

Pro **analýzu trendů** se využívá několik přímočarých technik. Jednou z nich je vyhlazování (smoothing), které odstraní zašumění (nesystematické chování). Typicky je vyhlazování prováděno za pomocí **klouzavých průměrů** (moving averages). V praxi to pak vypadá, že v daném časovém bodě se nepoužije aktuální hodnota, nýbrž průměr či medián hodnot v určitém časovém intervalu kolem daného bodu(v rámci tzv.„klouzavého okna“). Častěji se využívá medián, který je oproti průměru méně náchylný na případné odlehlé hodnoty.

V případě detekce sezónních vzorů se využívá metody **zpoždění** (lag). Originální časová řada se porovnává s totožnou řadou, která je ale posunuta (zpožděna) o hodnotu k. Časové

řady mají poté podobu $X = < x_1, x_2, ..., x_{n-k} >$ a $Y = < x_{k+1}, x_{k+2}, ..., x_n >$. Pro různé hodnoty zpoždění se vypočítává **Pearsonův korelační koeficient**. Nechť X je časová řada a Y její zpožděná forma s průměry $\bar{X}$ a $\bar{Y}$ a obsahují stejný počet záznamů, Pearsonův korelační koeficient se vypočítá tímto způsobem:

$$\frac{\sum(x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum(x_i - \bar{X})^2 \sum(y_i - \bar{Y})^2}} \tag{7.1}$$

Výsledná hodnota koeficientu se pohybuje v intervalu $< -1, 1 >$, kde hodnota 1 značí totožné časové řady a -1 přesně „opačné“ [4].

7.2 Modely pro predikci

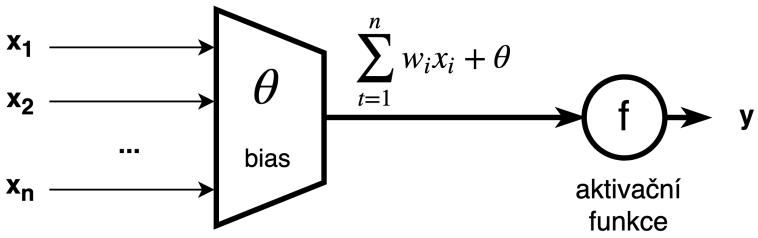
Jednou z hlavních otázek jakéhokoli dolování dat je volba správného modelu. V rámci případové studie budou vyzkoušeny 2 hojně rozšířené postupy - model neuronových sítí a metoda podpůrných vektorů. Oba tyto modely jsou využívány jak pro klasifikaci, tak pro regresi.

Posledním modelem použitým pro případovou studii je MARS - mnohorozměrný adaptivní regresní spline. Sice není tolik známý jako výše zmíněné modely a používá se pouze pro regresi, ale je vhodný i na mnohorozměrné úlohy, kterou predikování S&P 500 indexu je.

V následujících sekcích budou představeny základní principy a myšlenky těchto tří modelů.

7.2.1 Model neuronových sítí

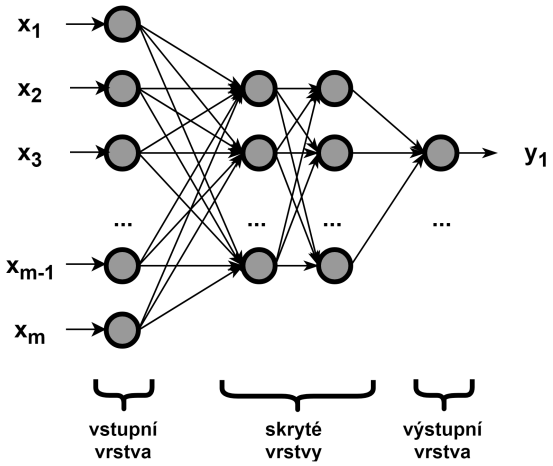
Model neuronových sítí vychází z principů lidské nervové soustavy. Základní jednotkou je umělý neuron znázorněný na obrázku 7.1, kde hodnoty vstupních dat $x_1, x_2, ..., x_n$ jsou upraveny váhami $w_1, w_2, ..., w_n$. Společně s přičtením vnitřní konstanty θ (nazývaná také jako **práh** (bias)) je jejich suma transformována aktivační funkcí a poslána na výstup.



Obrázek 7.1: Schéma umělého neuronu. Převzato z [27].

Algoritmus Backpropagation

Nejčastějším algoritmem používaným na neuronových sítích v rámci dolování dat je **Backpropagation**. **Vícevrstvá dopředná neuronová síť** se skládá ze vstupní a výstupní vrstvy. Mezi nimi se nachází jedna či více skrytých vrstev. Všechny neurony jsou navzájem plně propojené. Příklad takovéto sítě představuje obrázek 7.2. Výstupů z neuronové sítě může být na rozdíl od schématu více (např. vektor). Na základě toho se pak odvíjí i počet umělých neuronů ve výstupní vrstvě.



Obrázek 7.2: Schéma vícevrstvé dopředné neuronové sítě

Učení neuronové sítě probíhá v iteracích. Na konci každé iterace je výstupní vektor porovnán s přesnou hodnotou vektoru cílového atributu záznamu trénovací sady. Na základě této přesnosti jsou pak upravovány hodnoty vah a prahů jednotlivých neuronů, aby se minimalizovala střední kvadratická chyba (viz definice 7.2, [15]). Úprava hodnot probíhá opačným směrem než prvotní průchod sítí. Tato fáze se opakuje tak dlouho, dokud síť nebude adekvátně reagovat.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2, \tag{7.2}$$

kde pro výpočet střední kvadratické chyby (MSE, rozptyl) je n počet prvků v souboru, $\tilde{y}_i$ výstup neuronové sítě pro danou hodnotu a y_i její přesná hodnota.

Počáteční nastavení vah a prahů je pomocí malých náhodných čísel. V případě atributů spojitého charakteru jsou hodnoty normovány do intervalů <0,1> či <-1,1>. Ostatní členy algoritmu se definují následovně:

- Aktivační funkce neuronů: $y = \frac{1}{1+e^{-x}}$
- Chyba neuronu na výstupní vrstvě (pro jiné vrstvy se připočítávají i chyby neuronů následujících): $Err_j = O_j(1 - O_j)(T_j - O_j)$, kde T_j je přesná hodnota a O_j aktuální výstup neuronu j
- Vzorce pro úpravu vah: $\triangle w_{ij} = (l)Err_jO_i$; $w_{ij} = w_{ij} + \triangle w_{ij}$, kde w_{ij} je váha mezi neurony i a j
- Vzorce pro úpravu prahů: $\triangle\theta_j = (l)Err_j$; $\theta_j = \theta_j + \triangle\theta_j$
- Koeficient učení l vyskytující se v předchozích vzorcích: Jedná se o reálné číslo z intervalu <0,1>, typicky podle vztahu $l = 1/t$, kde t je index iterace. Díky němu jsou prahy a váhy ovlivněny především úvodními záznamy [27].

Klady a zápory použití neuronových sítí

Výhody neuronových sítí jsou především ve zvládání zašuměných dat a rozpoznání vzorů v neznámých datech. Výborně se hodí pro predikci spojitých dat [9] a zvládají redundantní data [21].

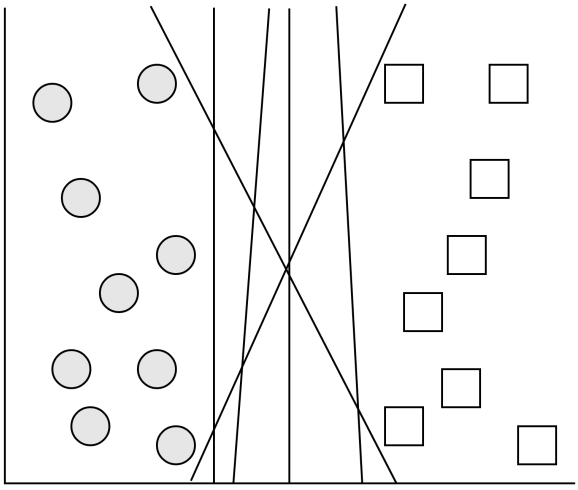
Na druhou stranu se mnohdy objevují problémy s návrhem neuronové sítě kvůli velkému množství veličin (počty uzlů v jednotlivých vrstvách, váhy, prahy, aktivační funkce), které analytik může upravovat. V konečném důsledku je obtížné vytvořit ideální neuronovou síť. V trénovací množině musí být odstraněny chybějící hodnoty a především učení neuronové sítě je **časově náročné** [21].

7.2.2 Support Vector Machines

Klasifikační a regresní technika strojového učení s učitelem, která v mnoha praktických aplikacích vykazovala slibné výsledky a zvládá dobře vysocedimenzionální data, je **metoda podpůrných vektorů** neboli **Support Vector Machines (SVM)** [21].

Lineárně oddělitelná data

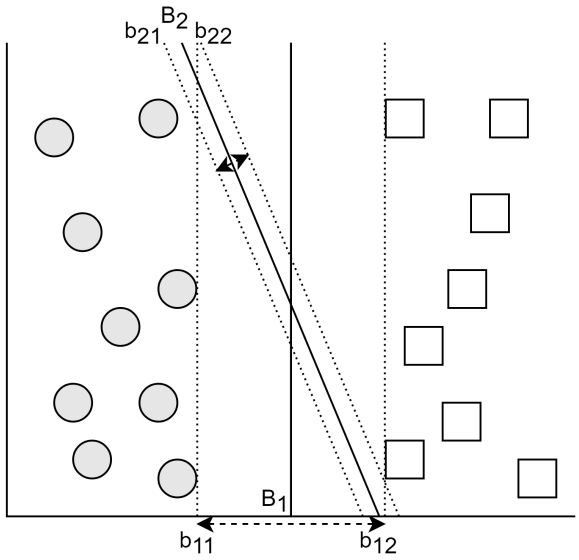
Základní myšlenka bude popisována na nejjednodušší možné variantě, lineárním klasifikátoru, který zařazuje objekty ve dvourozměrném prostoru do dvou tříd, a zároveň data jsou lineárně rozdělitelná. Cílem této metody je nalézt v obecném n-rozměrném prostoru **nadrovinu**, která objekty trénovací množiny rozděluje na poloprostory, a rovněž všechny objekty stejné třídy se nenachází v odlišném poloprostoru. Možných nadrovin pro tento případ je neomezený počet (znázorňuje obrázek 7.3). Ačkoliv splňují všechny nadroviny veškeré požadavky, nemusí být minimální klasifikační chyba různých nadrovin stejná i na dříve neznámých objektech.



Obrázek 7.3: Možné nadroviny v dvourozměrném prostoru (zde přímky), tj. trénovací objekty různých tříd se nachází v odlišných poloprostorech.

Intuitivně by se dalo očekávat, že nejvhodnější nadrovina bude taková, kde je její vzdálenost od nejbližších objektů obou tříd co největší. Je tomu tak. SVM metoda tedy vyhledává vždy nadrovinu s maximálním odstupem (maximum marginal hyperplane).

Tento odstup se dá neformálně definovat jako plocha ohraničená dvěma rovinami rovnoběžnými k nadrovině a zároveň nadrovina je ve stejné vzdálenosti k oběma hranicím. Šířka rozestupu hranic od nadroviny se rovná nejkratší vzdálenosti k nejbližšímu objektu. Na obrázku 7.4 se nachází 2 potenciální nadroviny B_i, kde $i \in \{1, 2\}$, s hranicemi odstupu b_{i1} a b_{i2}. Nadrovinou s maximálním odstupem (maximal margin) je B_1.



Obrázek 7.4: Nadroviny a jejich maximální odstupy (B_1 - optimální nadrovina). Převzato z [21].

Formální definice oddělující nadroviny pro množinu trénovacích dat $(\vec{x_i}, y_i)$, kde $\vec{x_i}$ je vektor objektu a y_i, $y \in \{1, -1\}$ údaj, do které třídy objekt patří, je následující (převzato z [9]):

$$\vec{w} \cdot \vec{x} + b = 0, \tag{7.3}$$

kde $\vec{w}$ je normála nadroviny a b skalár, často označován jako zkreslení.
Pro všechny objekty trénovací množiny tedy platí:

$$\begin{aligned} b_{11} &: \ \vec{w} \cdot \vec{x_i} + b \leq 1 \, pro \, y_i = -1, \\ b_{12} &: \ \vec{w} \cdot \vec{x_i} + b \geq 1 \, pro \, y_i = +1, \end{aligned} \tag{7.4}$$

což lze sloučit do nerovnice

$$y_i(\vec{w} \cdot \vec{x_i} + b) - 1 \geq 0, \ \forall i. \tag{7.5}$$

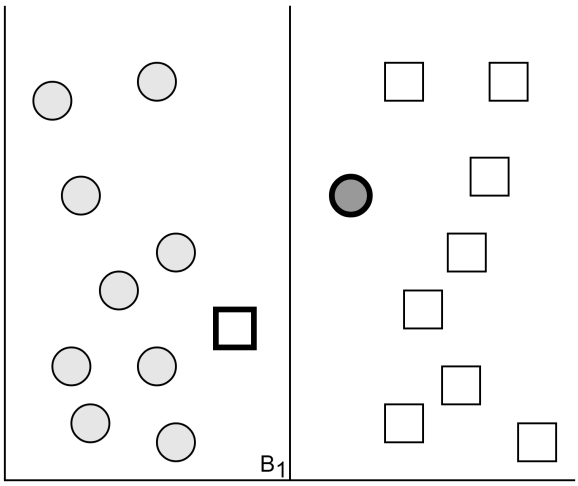
Veškeré trénovací objekty, které patří do nadrovin H_1 a H_2, se nazývají **podpůrné vektory**. Vzdálenost oddělující nadroviny k jakémukoliv bodu v H_1 je $\frac{1}{||\vec{w}||}$, kde $||\vec{w}||$ je Euklidova norma $\vec{w}$, tedy $\sqrt{\vec{w} \cdot \vec{w}}$. Vzhledem k tomu, že to samé platí i pro nadrovinu H_2, maximální odstup je $\frac{2}{||\vec{w}||}$ [9].

Pro nalezení nadroviny s maximálním odstupem, tedy vyřešení nerovnice 7.5 se typicky řeší pomocí přepsání do tvaru Lagrangeovy funkce, následně se využívají Karush-Kuhn-Tuckerovy podmínky. Jedná se o konvexní optimalizační problém kvadratického programování [21].

Složitost klasifikátoru je charakterizována zpravidla počtem podpůrných vektorů než dimenzí dat, nejkritičtější vektory pro naučení leží nejblíže hranicím maximálního odstupu. Metoda podpůrných vektorů je často odolná vůči přeučení [9].

Nelineárně oddělitelná data

Předchozí sekce popisovala hlavní myšlenky lineární metody podpůrných vektorů. Nyní bude nastíněno řešení úlohy nelineárního SVM, jako např. na obrázku 7.5. První krok spočívá v „přenesení“ dat z originálního do vícerozměrného prostoru pomocí nelineárního mapování. Druhý krok je už známé řešení lineárně oddělitelných dat v novém prostoru.



Obrázek 7.5: Případ nelineárně oddělitelných dat metody podpůrných vektorů. Nadrovina B_1 nemůže být použita, zároveň není možné vytvořit žádnou nadrovinu splňující kritéria.

S tímto přístupem ale vznikají další problémy. Jaké mapování by měl analytik použít? Rovněž ve výpočtech se častokrát objeví **skalární součin** dvou vektorů ve vysoce rozměrném prostoru, který je výpočetně velmi náročný. Tyto problémy řeší transformace skalárních součinů pomocí **jádrové funkce** (kernel trick), díky ní lze najít poměrně rychlé řešení. Obecně není dáno, která funkce povede k co nejpřesnějším výsledkům. Typicky však z praxe vyplývá, že rozdíly u následujících tří jsou minimální [9]. Nejčastější jádrové funkce jsou:

- **Polynomické jádro stupně** h**:** $K(\vec{x_i}, \vec{x_j}) = (\vec{x_i} \cdot \vec{x_j} + 1)^h$
- **Gaussova RBF (Radial basis function):** $K(\vec{x_i}, \vec{x_j}) = e^{-||\vec{x_i} - \vec{x_j}||^2/2\sigma^2}$
- **Sigmoida:** $K(\vec{x_i}, \vec{x_j}) = \tanh(\kappa \vec{x_i} \cdot \vec{x_j} - \delta)$

Metoda podpůrných vektorů má mnoho žádoucích vlastností, které ji dělají jednou z nejrozšířenější klasifikačních algoritmů. Je to díky řešení konvexního optimalizačního problému, pro který jsou dostupné efektivní algoritmy, či možné aplikaci na kategorická data [21].

V této sekci bylo nastíněno pouze řešení binární klasifikace. Existují však přístupy, které jsou schopny převést více třídní klasifikace na binární (např. one-against-one, ECOC)[9].

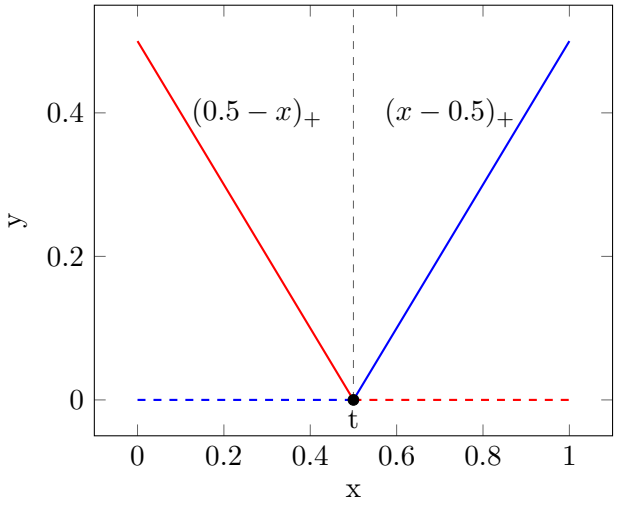
7.2.3 MARS

Mnohorozměrný adaptivní regresní spline (zkráceně MARS) je metoda regresní analýzy vhodná pro mnohorozměrné úlohy. Je považován za rozšíření lineárních modelů. MARS byl představen v roce 1991 Jeromem Haroldem Friedmanem. Lze ji chápat jako zobecnění postupné lineární regrese pro zlepšení výsledků [13].

Základem je množina lineárních **závěsných funkcí** (hinge functions) ve tvaru $(x-t)_+$ a $(t-x)_+$, kde „+“ značí pozitivní část (ekvivalentem jsou rovnice (7.6a) a (7.6b)). Platnost funkcí je pouze v určitém intervalu. Bod, ve kterém se 2 různé funkce protnou, se nazývá **uzel** (knot). Příklad takové funkce demonstruje obrázek 7.6.

$$(x-t)_+ = \begin{cases} x-t, & \text{když } x > t \\ 0, & \text{jinak} \end{cases} \tag{7.6a}$$

$$(t-x)_+ = \begin{cases} t-x, & \text{když } x < t \\ 0, & \text{jinak} \end{cases} \tag{7.6b}$$



Obrázek 7.6: Základní závěsné funkce $(x-0.5)_+$ a $(0.5-x)_+$ s vyznačeným uzlem t. Převzato z [10].

Hlavní myšlenka modelu MARS je vytvořit podobný pár funkcí pro každý vstup $\vec{X_j}$ s uzlem v každé hodnotě x_{ij} vstupu. Kolekce základních funkcí vypadá poté následovně:

$$C = \{(\vec{X_j} - t)_+, (t - \vec{X_j})_+\}, \tag{7.7}$$

kde $t \in \{x_{1j}, x_{2j}, ..., x_{Nj}\}$; $j = 1, 2, ..., p$.

Pokud by se veškeré vstupní hodnoty lišily, bylo by celkem $2Np$ základních funkcí. Sice každá funkce závisí z počátku na X_j, může být ovšem použita i pro jinou vstupní hodnotu.

Stavba modelu používá strategii **dopředné postupné lineární regrese** (forward stepwise linear regression), místo originálních hodnot jsou pak použity funkce z kolekce C a jejich součiny. Definice modelu poté je

$$f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(X), \tag{7.8}$$

kde každá funkce $h_m(X)$ je funkce z C nebo součin dvou či více takových funkcí, a β koeficienty [10].

Na konci vytváření procesu model obsahuje velké množství funkcí. Obvykle je model přeučen, proto se využívá zpětné optimalizace pro mazání funkcí, které neovlivní zásadně chybu modelu. Typicky se pro toto generalizování využívá metody **křížové validace**. Prahovou hodnotu, která specifikuje, jaké funkce budou vynechány, určuje parametr křížové validace „**penalty number**“, který se pro MARS pohybuje mezi hodnotami 2 a 3.

Model MARS se stal využívaný díky své flexibilitě. Dále je vhodný jak pro spojitá, tak i pro kategorická data. Oproti SVM je predikce rychlejší, tudíž je použitelný i pro větší datové sady [10].

7.3 Vyhodnocování modelů

V předchozí sekci bylo představeno několik modelů, které budou vyzkoušeny v rámci případové studie. Aby byl vybrán ten nejpřesnější, je nutné modely komplexně otestovat a vyhodnotit. Opět existuje několik přístupů, jak lze toho docílit.

Pro vyhodnocení je nutné zkoumat vyhodnocovací statistiky v závislosti na datech obsažených v trénovacích a testovacích množinách. Důvodem je povaha dat, jež se může v jednotlivých úsecích vstupní množiny dat měnit. Pokud např. trénovací množina obsahuje pouze úsek s extrémním a ojedinělým chováním, nebude model pracovat korektně při predikování hodnot „normálního režimu“. I tento případ ale může nastat, tudíž je nutné všechny možné scénáře zahrnout do vyhodnocování.

7.3.1 Metoda křížové validace

Jedním z přístupů řešení tohoto problému je metoda křížové validace. Křížová validace praktikuje **náhodné vzorkování záznamů** do trénovacích a testovacích množin. Vstupní data, u kterých známe hodnotu cílového atributu, jsou rozdělena do k množin s přibližně stejným počtem prvků. K trénovaní je použito $k-1$ množin, pro testování se použije zbylá množina. Pro každou iteraci je použit jiný testovací vzorek a model je vyhodnocen. Díky tomu je nasbíráno k statistik a výsledné metriky pro vyhodnocení jsou komplexnější [27].

Metoda křížové validace kvůli náhodnému vzorkování mění pořadí záznamů. To v případě časových řad není možné. Proces vyhodnocování musí zaručit, že model bude vždy využívat pro trénování modelu data historická, ne data budoucí [24].

Z tohoto důvodu bude v případě případové studie použit kvantitativní přístup metody Monte Carlo.

7.3.2 Monte Carlo

Metoda Monte Carlo se řadí mezi experimentální numerické simulační metody. Za pomocí **generování pseudonáhodných čísel** je experimentováno se stochastickým modelem za účelem určení střední hodnoty veličiny, která vzniká hodnocením náhodných experimentů. Po provedení velkého množství simulací se dají výsledky zpracovat a metoda začíná být přesná [18].

Nejznámější aplikací této metody je Buffonova úloha, kdy náhodné vrhy jehlou dopadaly na čtvercovou plochu s vepsanou čtvrtinou kružnice. Pravděpodobnost dopadu do kružnice se rovnala nejen obsahu útvaru, ale také hodnotě čísla π. S počtem hodů se zpřesňoval i jeho

desetinný rozvoj. Během 2. světové války sloužilo Monte Carlo k vývoji americké jaderné zbraně v projektu Manhattan.

Vzhledem k povaze metody se nedá očekávat, že metoda bude patřit k těm nejpřesnějším. Její přesnost znázorňuje vztah 7.9, tedy až 1 milion opakování experimentu zaručí chybu 10%.

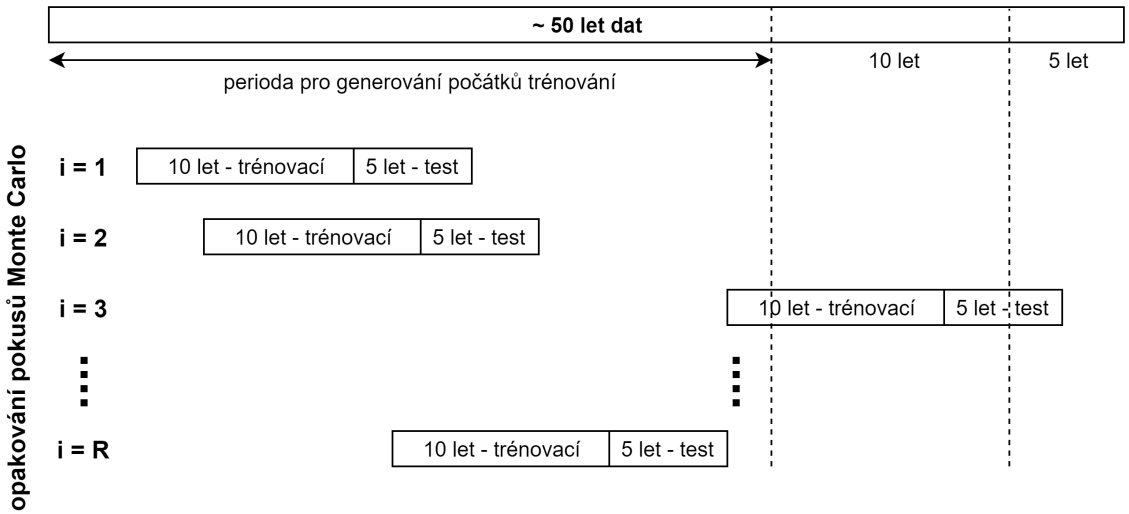
$$err = \frac{1}{\sqrt{N}}, \tag{7.9}$$

I přes svou jednoduchost nachází Monte Carlo uplatnění v mnoha oblastech. Především se používá k výpočtům složitých určitých integrálů a diferenciálních rovnic, ale také v hazardních hrách a ve financích. Zde se využívá pro modelování cash-flow, oceňování opcí, cenných papírů a úroků [14].

Uplatnění metody při řešení případové studie

Pro úlohu případové bude metoda Monte Carlo použita pro náhodné určení úseků, na kterých budou modely vyhodnocovány. Důvodem je nestálost indexu S&P 500. Trendy pro jednotlivá období se rapidně mění, objevují se i mimořádné události, kdy se index skokově odchýlil. Proto je nutné vyhodnotit jednotlivé modely ve více iteracích na různých úsecích dat, aby výsledná metrika popisující úspěšnost modelu byla co nejpřesnější.

K dispozici je cca. 50 let dat. Všechny modely používají pro natrénování množinu dat 10 let a jsou testovány na následujících 5 letech. Monte Carlo bude generovat náhodné hodnoty v úseku prvních 35 let, výsledky všech pokusů budou v závěru zesumarizovány, čímž se získá finální hodnocení modelu. Schématicky je tato metoda znázorněna na obrázku 7.7.



Obrázek 7.7: Experimentální proces Monte Carlo. Převzato z [24].

Kapitola 8

Řešení případové studie v jazyce Python

V této kapitole je demonstrováno využití jazyka Python a jeho prostředků pro dolování dat. Podrobně je popsána implementace jednotlivých kroků již zmíněných v kapitole 6 za pomocí využití modulů popsaných v kapitole 5. Výstupem této studie je skript v jazyce Python a jeho prezentace za pomocí Jupyter Notebooku.

Některá důležitá rozhodnutí prezentovaná v této kapitole jsou převzaty z [24].

8.1 Načtení vstupních dat

Vstupní data, tj. hodnoty indexu S&P 500 popsaného výše (kap. 6.2), jsou načteny pomocí tří odlišných přístupů. Modul Pandas nabízí metody pro načtení dat z různých zdrojů, výsledkem je objekt `pandas.DataFrame`.

DataFrame je dvourozměrná potencionálně heterogenní datová struktura připomínající tabulku s proměnlivou velikostí a popsanými osami. Jedná se o základní datový objekt modulu Pandas. Lze na něm jednoduše provádět aritmetické operace na obou osách. Jednotlivé sloupce představují objekty typu `pandas.Series` [17].

Důležitou vlastností DataFrame je index, což je identifikátor každého záznamu v objektu. V tomto případě je indexem datum aktuálního dne datového typu `pandas.Datetime`. Výhodou je jednoduchý intuitivní přstup k jednotlivým časovým obdobím.

Načtení dat ze souboru

Soubor obsahující hodnoty indexu je k dispozici na webových stránkách[1] autora knihy Data Mining with R [24], kterou byla případová studie inspirována.

Data jsou uložena v obecně rozšířeném formátu .CSV (Comma-separeted Values), Pandas nabízí metodu `pandas.read_csv` pro jeho zpracování.

Hodnoty obsažené v souboru datují časové období mezi lety 1970 a 2009.

[1] Webové stránky k publikaci Data Mining with R - http://www.dcc.fc.up.pt/~ltorgo/DataMiningWithR/

Načtení dat z internetu

Dosud byla pro načítání aktuálních dat z webu hojně využívána knihovna pandas-datareader[2], která v Pythonu zprostředkovávala údaje z Yahoo! Finance API. Bohužel v roce 2017 přestala společnost Yahoo! tuto službu podporovat.

Pro získání aktuálnějších dat bylo proto využito rozhraní Alpha Vantage[3], které nabízí historická i aktuální finanční data, údaje o vývoji cizích měn, kryptoměn a technických indikátorů. Nevýhodou tohoto rozhraní je nutná mailová autentizace, k tomu zároveň i časový limit přístupů do API. Použití pro získání denních hodnot je následující:

```
# Loading data from the internet
from alpha_vantage.timeseries import TimeSeries

ts = TimeSeries(key='YOUR_API_KEY',output_format='pandas', indexing_type='
    date')
web_data = ts.get_daily(symbol='SPX', outputsize='full')[0]
```

Získané údaje indexu S&P 500 pochází z období od roku 2000 do současnosti. Průnik těchto dat a hodnot ze souboru je použit pro další výzkum.

Načtení dat z databáze

Obdobně jako ze souboru načítá Pandas data i z databáze. Společnost MySQL[4] nabízí kromě komunitní edice databázového serveru i modul pro připojení k databázi v rámci Python skriptu a následné vykonání dotazu.

Vytvoření tabulky indexu v databázi bylo dosaženo za pomocí SQL skriptu uloženého na stránkách knihy [3]. Přístup načtení dat z databáze je v případové studii pouze pro ukázku, s těmito daty se nadále již nepracuje.

8.2 Cílový atribut

Cílem této studie je předpovědět budoucí hodnotu indexu S&P 500. Konkrétněji se jedná o predikování hodnoty, která popisuje jakým směrem a jakou intenzitou se bude index vyvíjet v dalších několika dnech. V praktické části i v následujících částech textu bude nazývána tato hodnota jako **indikátor**. Počet zohledněných dnů symbolizuje hodnota k , zde konkrétně $k = 10$.

Nechť i symbolizuje aktuální den a V_i je množina k procentuálních odchylek $\bar{P}_{i+j}$ (průměr hodnot High, Close a Low dne $i + j$) od C_i (hodnoty Close) pro následujících k dní:

$$V_i = \left\{ \frac{\bar{P}_{i+j} - C_i}{C_i} \right\}_{j=1}^{k} \tag{8.1}$$

Indikátor T_i je pak suma hodnot v z V_i, které překračují práh p v negativním či pozitivním smyslu (pro tento případ je $p = 2.5\%$):

$$T_i = \sum_{v} \{v \in V_i : v > p\% \vee v < -p\%\} \tag{8.2}$$

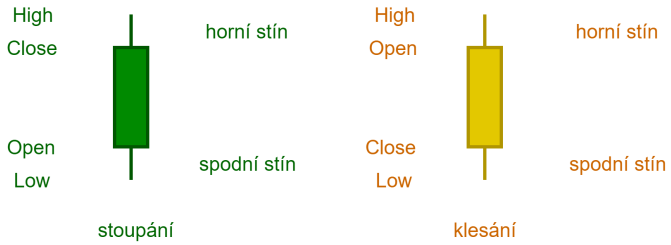
[2] pandas-datareader - https://pandas-datareader.readthedocs.io/en/latest/index.html
[3] Alpha Vantage - https://www.alphavantage.co/
[4] MySQL - https://dev.mysql.com/

Hodnota p zamezuje nepatrným změnám indexu ovlivnit drasticky hodnotu cílového atributu. Pokud tedy výsledek indikátoru je větší než 0, index bude stoupat a případná koupě akcií by byla profitující. V opačném případě bude probíhat pokles a prodej akcií je výhodný [24].

Indikátor T_i je ve skriptu implementován funkcí `indicator` a následně ukládán do následujících DataFrame jako sloupec `Indicator`.

Typické znázornění indexu S&P 500 představuje svíčkový graf skládající se z jednotlivých svíček (obrázek 8.1) pro každý pracovní den. Kompletní svíčkový graf i s cílovým indikátorem je ukázán na obrázku 8.2.

K implementaci svíčkového grafu byl použit modul `mpl_finance` balíčku Matplotlib. Rovněž pro tvorbu všech grafů případové studie je využito knihovny Matplotlib.



Obrázek 8.1: „Svíčka“ svíčkového grafu znázorňující všechny hodnoty indexu S&P 500 pro jednotlivý den. Tělo svíčky znázorňuje rozmezí hodnot Open a Close (pořadí otočeno u jednotlivých režimů), stíny poté maximální a minimální hodnoty v průběhu dne

8.3 Popisné atributy

Výše vytvořený cílový indikátor souhrnně popisuje chování trhu v následujících dnech na základě znalosti těchto „budoucích“ hodnot indexu. Nyní je ale nutné predikovat hodnoty tohoto indikátoru pouze z aktuálních a historických dat.

Hlavní předpoklad této studie zní, že **budoucí chování trhu je ovlivněno jeho vývojem v minulosti** [24]. Na základě výsledků studie bude v závěru zhodnoceno, zda se tento předpoklad zakládá na pravdě.

Popisných atributů, které by měly zachycovat chování indikátoru, bude hned několik, a zároveň budou cílit na různé vlastnosti. Posléze bude vyhodnoceno, které budou pro danou úlohu nejvhodnější.

Typickým přístupem při analýze časových řad je používání zpoždění a klouzavých průměrů (uvedeno v kapitole 7.1). Týdenní trendy se snaží vyhledat atributy aritmetické návratnosti hodnot Close ($\frac{C_{i-h}-C_i}{C_i}$) s hodnotami Close zpožděnými o 1 až 10 dnů, ve studii používané pod názvy `Delt.h.arithmetic`, kde h je zpoždění ve dnech. Klouzavé okno je využito u výpočtu průměrné hodnoty a směrodatné odchylky.

Všechny ostatní popisné atributy jsou specifické pro finanční analýzu a hojně využívané ekonomickými analytiky. Navzdory jejich diskutabilního využití pro predikování budou využity v této studii [24]. Důraz bude kladen převážně na hodnotu indexu na konci dne (atribut Close).

Obrázek 8.2: Svíčkový graf indexu S&P 500 a hodnoty indikátoru pro první čtvrtinu roku 2009

8.3.1 Technické identifikátory

Nabízí se použití termínu přejatého z anglického jazyka **technické indikátory** (technical indicators). Z praktických důvodů a kvůli možné záměně s cílovým atributem je v této práci použito pojmu **technický identifikátor**.

Jedná se o heuristické nebo matematické kalkulace založené na ceně nebo objemu transakcí ve snaze identifikovat příležitosti pro obchodování a analyzovat možné vzory v datech [22].
Technické identifikátory se dělí na 4 základní typy:

- **tendenční** (Trend) - předpovídají, jakým směrem se bude trh pohybovat
- **oscilační** (Momentum) - vychází z cyklického opakování pohybů a snaží změřit sílu trhu
- **objemové** (Volume) - analyzují tok peněz
- **volatilní** (Volatility) - určují, kdy je trh nestálý a může docházet ke kolísání hodnot

Pro tuto úlohu se z celkových 14 identifikátorů (vybraných na základě [24]) jako nejvhodnější ukázaly identifikátory AD line a MACD.

Linie akumulace/distribuce (AD line) naznačuje tok prostředků v rámci cenných papírů a patří do identifikátorů objemu.

Počítá se na základě vztahu

$$AD_i = AD_{i-1} + CLV \cdot Volume_i, \tag{8.3}$$

kde AD_{i-1} je hodnota identifikátoru z minulého dne, $Volume_i$ hodnota objemu transakcí z indexu a CLV je definováno jako

$$CLV = \left(\frac{(Close_i - Low_i) - (High_i - Close_i)}{(High_i - Low_i)} \right) \tag{8.4}$$

MACD je zkratka pro Moving Average Convergence Divergence. Tento oscilační identifikátor pracuje s exponenciálním klouzavým průměrem hodnoty Close (tzn. vážený klouzavý průměr, kde novějším hodnotám jsou přiřazeny největší váhy řídící se exponenciálou). MACD se poté získá z odečtení 26denního exponenciálního klouzavého průměru od 12denního. Studie využívá jeho signální křivku, což je 9denní exp. klouzavý průměr hodnoty MACD.

Implementace jednotlivých identifikátorů nabízí v Pythonu moduly ta[5] a TA-Lib, kterým jsou předávány požadující hodnoty indexu. V případě jednoduchých formulí identifikátorů není problém využít nástrojů pro matematické operace knihovny pandas a vytvořit vlastní definici identifikátoru.

8.3.2 Výběr popisných atributů

Pro výběr nejvhodnějších atributů k predikci indikátoru bude použita metoda náhodných lesů (Random Forest) strojového učení, která určí atributy s největší korelací vzhledem k indikátoru. Pouze tyto budou použity pro dolování.

Do výběru vstupují atributy popsané výše (v sekcích 8.3 a 8.3.1). Celkově se jedná o 26 atributů spojitého typu.

Přístup je implementován modulem Scikit-learn. Po inicializaci je model natrénován množinou dat obsahující všechny vytvořené atributy a cílový indikátor. Poté pomocí atributu `feature_importances_` klasifikátoru získáme vzájemné porovnání důležitosti jednotlivých atributů pro predikci.

8.4 Vytvoření modelů, předzpracování dat a kritéria vyhodnocování

Samotné dolování z popisných atributů je provedeno dvěma způsoby. Tím prvním je **regrese**, kde bude přímo predikován cílový indikátor. V druhém případě se jedná o **klasifikaci**. Hodnoty indikátoru jsou transformovány do podoby **signálů** tímto způsobem:

$$sig_{y_i} = \begin{cases} Buy, & \text{když } y_i > 0.1 \\ Hold, & \text{když } -0.1 \leq y_i \leq 0.1 \\ Sell, & \text{jinak} \end{cases} \tag{8.5}$$

Hodnoty signálů jsou pak vstupem do trénovací množiny jako cílový atribut. Pro modelování bude použito metod již zmíněných v kapitole 7. Metoda podpůrných vektorů stejně

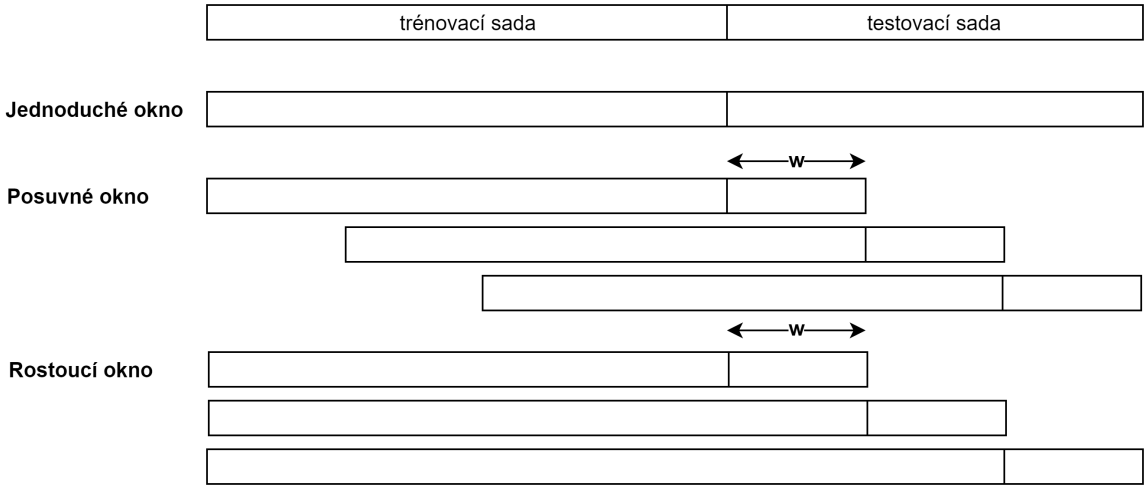
[5] modul ta - https://pypi.org/project/ta/

tak jako model neuronových sítí jsou použity jak pro regresi, tak pro klasifikaci. Mnohorozměrný adaptivní regresní spline (MARS) slouží pouze k predikování hodnot indikátorů. Pro vytváření a práci s modely bylo využito objektově orientovaného přístupu, který jazyk Python nabízí. Kromě modulárnosti a lepší strukturalizaci kódu vlastní bázová třída `PredictorBase` abstrahuje rozdílnou inicializaci a chování jednotlivých modelů, díky níž velice zjednodušuje jejich následné použití.

8.4.1 Trénovací metody

Mimo různých typů modelů se liší i způsoby, jakým jsou použita trénovací data při predikování. Vzhledem k chování časových řad v rozdílných obdobích, kdy se tendence indexu mohou rapidně změnit, se dají očekávat co nejlepší výsledky při zahrnutí nejaktuálnějších dat do trénovací množiny. Proto kromě jednoduchého přístupu **jednotného okna**, kdy je celá testovací množina predikována najednou, jsou implementovány i metody **posuvného** a **rostoucího okna**.

Pro testovací sadu je stanoven krok w reprezentující určitý počet dnů, po kterém jsou modely s učící metodou posuvného a rostoucího okna přeučeny. Nové trénovací sady zahrnují poté i hodnoty popisných atributů posledních dnů. U posuvného okna zůstává počet záznamů v trénovací sadě stejný, u rostoucího se zvětšuje. Přehledně jsou metody naznačeny na obrázku 8.3.



Obrázek 8.3: Techniky jednoduchého, posuvného a rostoucího trénovacího okna s krokem w. Převzato z [24].

Pro každou variantu modelu je tedy použito 5 variant trénovacích metod (délka kroku w 120 a 240 dní u posuvné a rostoucí metody). Pouze u modelu MARS je z důvodu výpočetní náročnosti využito jen jednoduchého okna.

8.4.2 Vyhodnocovací kritéria modelů

Jak jsou modely úspěšné, bude určeno na základě hlavního vyhodnocovacího kritéria - **přesnosti** (precision). Kritérium vychází z matice záměn používané pro klasifikaci. Výstupy regresních modelů jsou také transformovány na signály, tudíž lze porovnat i tyto modely.

Matice záměn obsahuje souhrn predikovaných a pravdivých hodnot v testovací sadě (tabulka 8.1). Např. hodnota $n_{s,s}$ určuje, kolik bylo signálů Sell správně predikováno, hodnota $n_{s,h}$ určuje, kolik signálů Sell bylo chybně určeno jako Hold, apod.

Tabulka 8.1: Matice záměn

		Predikované		
		Sell	Hold	Buy
Pravdivé	Sell	$n_{s,s}$	$n_{s,h}$	$n_{s,b}$
	Hold	$n_{h,s}$	$n_{h,h}$	$n_{h,b}$
	Buy	$n_{b,s}$	$n_{b,h}$	$n_{b,b}$

Přesnost poté určuje procentuální hodnotu správně predikovaných signálů daného typu oproti celkovému počtu predikcí této hodnoty. Tento atribut je zkoumán pouze u signálů Sell a Buy, které jsou stěžejní pro případné akce na trhu. Hodnota Hold je typicky zastoupena naprostou většinou a nesymbolizuje žádnou akci na trhu.

$$\text{Přesnost} = \frac{n_{s,s} + n_{b,b}}{N_{.,s} + N_{.,b}} \tag{8.6}$$

Pro lepší představu o výsledku predikování jednotlivých modelů jsou zmíněny i 2 další vlastnosti. Nejsou už ale kritérii pro vyhodnocení.

Úplnost (recall) označuje podíl všech odhalených hodnot dané třídy k celkovému počtu výskytů této třídy. Není tak podstatný oproti přesnosti, jelikož případné nevyužití příležitosti k investici by nevedlo k tak velkým finančním ztrátám jako u přesnosti.

$$\text{Úplnost} = \frac{n_{s,s} + n_{b,b}}{N_{s,.} + N_{b,.}} \tag{8.7}$$

U finální podoby výsledků modelů je znázorněna i jejich **správnost** (accuracy). Jedná se o podíl správně predikovaných hodnot všech tříd k celkovému počtu všech vzorků.

$$\text{Správnost} = \frac{n_{s,s} + n_{h,h} + n_{b,b}}{N} \tag{8.8}$$

8.4.3 Transformace dat pro jednotlivé modely

Pro modely neuronových sítí a metody podpůrných vektorů byly v případové studii použity tyto transformace:

1. normalizace
2. vyvážení minoritní třídy vzorků

Oba výše zmíněné typy modelů jsou citlivé na různá měřítka jednotlivých popisných atributů [24]. Proto je nutné hodnoty vstupních dat trénovacích množin pro tyto modely nejprve **normalizovat** (vztáhnout hodnoty proměnné k nějaké definované hodnotě). Používá se také výraz **standardizace**.

V tomto případě budou všechny atributy vztaženy k jejich směrodatné odchylce. Normalizovaná hodnota je definovaná podle vztahu:

$$y_i = \frac{x_i - \bar{x}}{\sigma_x}, \tag{8.9}$$

kde $\bar{x}$ je průměrná hodnota atributu X a σ_x jeho směrodatná odchylka. Výsledné hodnoty jsou poté vycentrované k nule a vztahy mezi jednotlivými vzorky atributu zůstávají zachovány. V případě regrese musí být poté predikované hodnoty indikátoru zpátky transformovány do původní podoby.

Vzhledem k **nevyváženému počtu vzorků** v jednotlivých třídách signálů je nutné použití vyvažovacích technik. V opačném případě by byl model zkreslen velkým množstvím Hold signálů.

U neuronových sítí byly v trénovacích množinách uměle duplikovány záznamy se signály Sell a Buy tak, aby finální trénovací množina obsahovala stejný počet záznamů jak s cílovým signálem Hold, tak i společně Buy a Sell.

U metody podpůrných vektorů byly naopak nastaveny váhy záznamů minoritních tříd na trojnásobek oproti majoritní třídě.

Různé typy normalizací a další druhy předzpracování dat nabízí modul Scikit-learn.

Naopak adaptivní spline jako jeden z lineárních modelů používá nenormalizovaná data, zároveň nebylo využito ani žádné z technik předzpracování.

8.5 Vyhodnocení a výběr modelů

Jak již bylo zmíněno, nejenom index S&P 500, ale i ostatní odvětví se ve finanční analýze potýkají se značně odlišnými obdobími vývoje trhu. Zároveň datová množina této studie obsahuje skoro 50 let záznamů indexu. Proto bylo nutné otestovat modely na velkém množství dat a s co největším počtem provedení experimentu Monte Carlo.

Bohužel trénování modelů a také samotná metoda Monte Carlo (její použití pro tuto studii již zmíněno v kapitole 7.3.2) je velice výpočetně náročná, tudíž počet provedení experimentu byl limitován.

Pro experiment Monte Carlo trénovací sada obsahovala 2540 záznamů (cca. 10 let) a testovací sada 1270 záznamů (cca. 5 let). Byly pseudonáhodně vybrány období 15 let pro vyhodnocení modelů.

Pro každý model byla metoda vykonána 10krát s výjimkou MARS, jenž byl vykonán pouze 5krát s ohledem na jeho výpočetní dobu, která je oproti ostatním modelům až desetinásobná. I přesto se celková výpočetní doba experimentu pohybovala okolo 5 hodin.

Výsledky prezentuje tabulka 8.2 a graf 8.4. Modely jsou nazvány `net` pro neuronové sítě, `svm` pro metodu podpůrných vektorů a `mars` pro mnohorozměrný adaptivní regresní spline. Písmena `C` a `R` na konci názvů značí, zda je model klasifikační nebo regresní. Následuje určení trénovací metody ve formátu `p` (posuvný), `r` (rostoucí), `j` (jednoduchý). V závorkách je uvedena velikost okna.

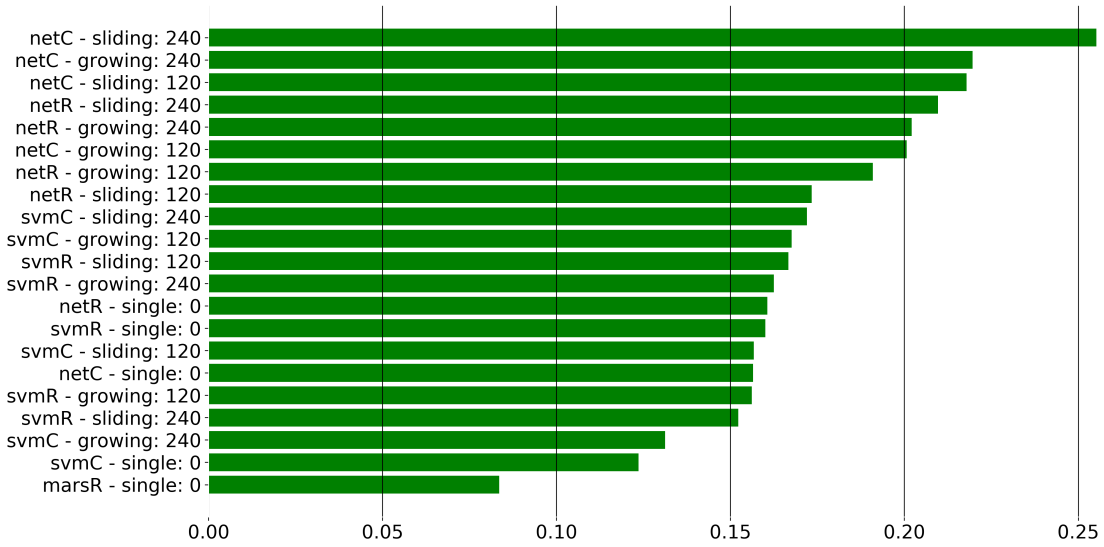
Je na první pohled patrné, že nejlepších výsledků dosáhl algoritmus neuronových sítí. Při porovnání všech variant trénovacích metod doslova „převálcoval“ ostatní metody jak mezi klasifikačními, tak i regresními modely.

Potvrdila se taky myšlenka, že použití posuvného a rostoucího okna u predikování povede k lepším výsledkům. Stojí za zvážení obětovat výpočetní čas a zmenšit velikost okna, aby model měl k dispozici ještě aktuálnější data.

Zatímco výsledky modelů metody podpůrných vektorů jsou srovnatelné s neuronovými sítěmi, adaptivní spline neobstál ani mezi modely s jednoduchou trénovací metodou. S přesností okolo pouhých 8% nelze považovat model za validní.

Tabulka 8.2: Výsledky experimentu Monte Carlo (seřazeno **podle atributu Průměr** (z hodnot přesnost a správnost)) - **5 nejlepších**

Model	Počet opak.	Přesnost	Úplnost	Správnost
netC-p(240)	10	0.2552	0.3000	0.6085
netC-r(240)	10	0.2196	0.2547	0.6097
netC-p(120)	10	0.2179	0.2615	0.5921
netR-p(240)	10	0.2097	0.2189	0.5898
netR-r(240)	10	0.2021	0.1842	0.6108



Obrázek 8.4: Zobrazení nejlepších modelů seřazených podle přesnosti

Kapitola 9

Zhodnocení jazyka Python pro úlohy dolování dat

V této kapitole zhodnotím jazyk Python a jeho využití při dolování dat. Představím výhody, nevýhody a zkušenosti, se kterými jsem se při vypracovávání praktické části setkal a kterým bylo nutné čelit.

9.1 Výhody

V první řadě bych chtěl vyzdvihnout přehlednost jazyka a jeho syntaxi. Proto je považován za jeden z ideálních programovacích jazyků pro začátečníky. Z mého hlediska je výhodou, že je koncipován pro programování, ne pro vědeckou činnost jako ostatní jazyky používané v dolování. Tudíž zaběhlé zvyklosti z jazyků C, Java apod. zde naleznou své uplatnění.

S tím souvisí možnost objektově orientovaného přístupu. Díky dobrému návrhu jednotlivých tříd bylo v závěru případové studie jednoduché použití i odlišných druhů modelů, které se lišily jak svou inicializací, tak i odlišnými způsoby zacházení s nimi.

Většinu prostředků potřebných pro dolování v Pythonu nabízí moduly Pandas a Scikit-learn. Díky tomu se práce a volání jednotlivých funkcí značně neliší a při nastudování využívání objektů `pandas.Series` a `pandas.DataFrame` lze tyto dovednosti jednoduše používat i nadále. V případě absence jakékoliv funkcionality lze využít další moduly, kterých je pro Python implementováno velké množství. A většinou pak člověk nalezne to, co hledá.

V neposlední řadě bych chtěl zmínit jako výhodu projekt Jupyter Notebook. Tato webová aplikace opravdu přehledně prezentuje jednotlivé bloky implementace a je k užitku nejen k závěrečné prezentaci výsledků, kterou jsem využil i já, ale i k prvotnímu studiu jazyka Python.

9.2 Nevýhody

Za hlavní nevýhodu jazyka Python považuji jeho rychlost vykonávání kódu z důvodu jeho interpretace. Výpočetní čas v případové studii byl kritickým faktorem a negativně ovlivnil výsledné vyhodnocování modelů. Pro potřeby úloh podobného typu je nutné využít výkonnějších strojů než je obyčejný stolní počítač.

Druhým záporným bodem je finanční analýza v jazyce Python. Vše potřebné pro práci s časovými řadami a vytváření ekonomických popisných atributů lze sice implementovat, např. ale jazyk R a jeho modul `ttr` nabízí rozsáhlejší možnosti.

Celkově ale hodnotím jazyk Python pro použití v oblasti dolování dat kladně.

Kapitola 10

Závěr

V rámci této bakalářské práce byly dopodrobna představeny základní myšlenky a principy oblasti dolování dat. Základem vyhledávání důležitých vzorů z velkého množství dat je proces získávání znalostí z databází. Vzhledem k obsáhlosti problematiky dolování byly teoreticky popsány jenom ty body, které byly následně použity pro řešení případové studie.

Zároveň byly nabyté znalosti prakticky využity při demonstraci dolovací úlohy z oblasti financí. Pro tyto účely jsem využil možnosti skriptovacího jazyka Python. Prostředky pro získávání dat nabízí především moduly Pandas a Scikit-learn. Jejich použití je snadné a v případě chybějící funkcionality v těchto modulech lze jednoduše využít funkce z velkého množství jiných knihoven, což je jednou z velkých výhod jazyka Python.

Předpovídání hodnot vývoje finančních trhů je rozšířenou doménou v oblasti dolování. Je to pochopitelné. Kromě potenciálního finančního zisku nabízí doména velké množství dat vhodných k analýze. V případě predikování indexu S&P 500 se vychází z předpokladu, že existuje vztah mezi historickými daty a aktuálními hodnotami. A to teorie nezanedbatelného počtu ekonomických analytiků vyvrací.

I přesto případová studie pomocí technik dolování tento vztah mezi historickými a aktuálními daty dokazuje. Modely neuronových sítí, metody podpůrných vektorů a lineární regresní metody MARS však nedosáhly takových výsledků, aby mohly být využity v praxi. Zejména případné aplikování predikovaných hodnot modelů v reálném investičním světě by vedlo k velkému úbytku finančních prostředků.

Ovšem stále existuje prostor, jak výslednou přesnost modelů vylepšit. Případová studie se musela vypořádat s velkou výpočetní náročností zejména při trénování modelů a provádění experimentu metodou Monte Carlo. To se promítlo i omezením některých přístupů, které by pravděpodobně přispěly k lepším výsledkům.

Jednou z možností v případě použití výkonnějších prostředků pro dolování by bylo zkrácení hodnoty učícího kroku u technik posuvného a rostoucího okna a větší počet provedení experimentu Monte Carlo. To však rapidně zvedá výpočetní čas. Také při inicializaci modelů by mohlo být využito technik pro nastavení ideálních hodnot parametrů pro danou konkrétní úlohu, např. pomocí modulu Scikit-learn. Poslední navrhované řešení pro rozšíření praktické části je podrobnější studium vhodnosti technických identifikátorů použitých jako popisné atributy indikátoru indexu S&P 500. To ovšem bylo nad rámec mých znalostí i celé případové studie.

Ve výsledku však bakalářská práce názorně popsala oblast dolování dat a prakticky předvedla tyto znalosti na konkrétní dolovací úloze. Praktická část je navržena komplexně a do budoucna lehce rozšířitelná.

Literatura

[1] *Top 8 programming languages every data scientist should master in 2019. [Online; navštíveno 16.04.2019].*
URL https://bigdata-madesimple.com/top-8-programming-languages-every-data-scientist-should-master-in-2019/

[2] *Danubianu, M.: Step by step Data Preprocessing for Data mining. a case study. 2014, [Dostupné online, navštíveno 12.04.2019].*
URL https://www.researchgate.net/publication/288825433_STEP_BY_STEP_DATA_PREPROCESSING_FOR_DATA_MINING_A_CASE_STUDY

[3] *Data Mining With R - learning with case studies, websites. [Online; navštíveno 08.05.2019].*
URL http://www.dcc.fc.up.pt/~ltorgo/DataMiningWithR/

[4] *Dunham, M. H.: Data Mining. Introductory and Advanced Topics. Prentice-Hall, 2003, ISBN 0-13-088892-3, 315 s.*

[5] *Data mining: Cesta od vyzobávání rozinek k těžbě zlata. [Online; navštíveno 11.04.2019].*
URL https://www.ekontech.cz/clanek/data-mining-cesta-od-vyzobavani-rozinek-tezbe-zlata

[6] *Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P.: The kdd process for extracting useful knowledge from volumes of data.* Journal of the ACM, *1996: s. 39(11):27–34.*

[7] *Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P.; aj.: Advances in Knowledge Discovery and Data Mining. AAAI Press, 1996, ISBN 978-0-26-256097-9.*

[8] *IBM Predicts Demand For Data Scientists Will Soar 28navštíveno 16.04.2019].*
URL https://www.forbes.com/sites/louiscolumbus/2017/05/13/ibm-predicts-demand-for-data-scientists-will-soar-28-by-2020

[9] *Han, J.; Kamber, M.; Pei, J.: Data Mining Concepts and Techniques, Third Edition. Elsevier Inc., 2012, ISBN 978-0-12-381479-1, 740 s.*

[10] *Hastie, T.; Tibshirani, R.; Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer-Verlag New York, 2009, ISBN 978-0-3878-4857-0.*

[11] *S&P 500 index – jeden z nejdůležitějších indexů v USA. [Online; navštíveno 17.04.2019].*
URL https://www.lynxbroker.cz/vzdelavani/sp-500-index/

[12] *Maimon, O.; Rokach, L.: Data Mining and Knowledge Discovery Handbook. Springer US, 2009, ISBN 978-0-387-09823-4.*

[13] *Multivariate adaptive regression splines. [Online; navštíveno 24.04.2019].*
URL https://machinelearningcatalogue.com/algorithm/alg_multivariate-adaptive-regression-splines.html

[14] *Jak využít simulace Monte Carlo ve financích. [Online; navštíveno 25.04.2019].*
URL https://www.mesec.cz/clanky/jak-vyuzit-simulace-monte-carlo-ve-financich/

[15] *Machine learning: an introduction to mean squared error and regression lines. [Online; navštíveno 01.05.2019].*
URL https://medium.freecodecamp.org/machine-learning-mean-squared-error-regression-line-c7dde9a26b93

[16] *About NumPy. [Online; navštíveno 16.04.2019].*
URL https://www.numpy.org/

[17] *Pandas API Reference. [Online; navštíveno 08.05.2019].*
URL
https://pandas.pydata.org/pandas-docs/stable/reference/index.html

[18] *Peringer, P.; Hrubý, M.: Modelování a simulace, interní studijní materiály k předmětu IMS. [Online; navštíveno 25.04.2019; dostupné pouze přes IS FIT VUT].*

[19] *ProPublica: Machine Bias. [Online; navštíveno 11.04.2019].*
URL https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[20] *What is Python? [Online; navštíveno 16.04.2019].*
URL https://pythoninstitute.org/what-is-python/

[21] *Tan, P.-N.; Steinbach, M.; Kumar, V.: Introduction to Data Mining (First Edition). Pearson Education, Inc., 2016, ISBN 0-321-42052-7.*

[22] *Technical Indicator. [Online; navštíveno 09.05.2019].*
URL https://www.investopedia.com/terms/t/technicalindicator.asp

[23] *TIOBE Index for April 2019. [Online; navštíveno 16.04.2019].*
URL https://www.tiobe.com/tiobe-index/

[24] *Torgo, L.: Data Mining with R: Learning with Case Studies. Chapman and Hall/CRC, 2011, ISBN 978-1-4398-1018-7.*

[25] *Data Warehousing - Metadata Concepts. [Online; navštíveno 12.04.2019].*
URL https://www.tutorialspoint.com/dwh/dwh_metadata_concepts.htm

[26] *Testování modelů a jejich výsledků. [Online; navštíveno 15.04.2019].*
URL https://cw.fel.cvut.cz/b181/_media/courses/a6m33dvz/03-testovanimodelu.pdf

[27] *Zendulka, J.; Bartík, V.; Lukáš, R.; aj.: Získávání znalostí z databází, Studijní opora. Jaroslav Zendulka a kol., 2010.*

Příloha A

Obsah přiloženého média

Součástí této práce je i přiložený datový nosič s následujícím obsahem:

- **src** - adresář, ve kterém jsou obsaženy veškeré zdrojové soubory a Jupyter Notebooky
- **data** - adresář obsahující zdrojová data použitá v případové studii
- **notebooks** - adresář obsahující vygenerované Jupyter Notebooky do formátu HTML
- **readme.txt** - soubor obsahující podstatné informace k zdrojovým souborům
- **thesis** - adresář obsahující zdrojové soubory textové zprávy bakalářské práce
- **xdudaj01.pdf** - textová zpráva v elektronické podobě