VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

ZÍSKÁVÁNÍ ZNALOSTÍ PRO MODELOVÁNÍ NÁSLEDNÝCH AKCÍ

DATA MINING FOR SUGGESTING FURTHER ACTIONS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. MARTIN VESELOVSKÝ**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. VLADIMÍR BARTÍK, Ph.D.**
SUPERVISOR

BRNO 2017

Zadání diplomové práce/20133/2016/xvesel60

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2016/2017

Zadání diplomové práce

Řešitel: **Veselovský Martin, Bc.**
Obor: Inteligentní systémy
Téma: **Získávání znalostí pro modelování následných akcí**
Data Mining for Suggesting Further Actions
Kategorie: Data mining

Pokyny:

1. Seznamte se s problematikou dolování z dat a s dostupnými prostředky na podporu dolování z dat v jazyce Python.
2. Analyzujte data z reklamních kampaní, z prostředí aukčních, reklamních systémů, s cílem určení veškerých atributů majících vliv na výkonnost reklam.
3. Realizujte datový sklad obsahující předzpracovaná data určená k pokročilým analýzám a získávaní nových znalostí.
4. Navrhněte systém na predikci vývoje úspěšnosti reklamy v budoucnu na základě historických hodnot a realizujte dolování znalostí. Proveďte experimentální vyhodnocení.
5. Rozšiřte systém o modelováni akcí, které se na reklamě mohou provést v závislosti na jejím aktuálním stavu s cílem dosažení maximálního budoucího úspěchu.
6. Zhodnoťte dosažené výsledky a další možné pokračování tohoto projektu.

Literatura:

- Ponniah, P.: Data Warehousing Fundamentals. John Wiley and Sons, 2001.
- Laberge, R.: Datové sklady - Agilní metody a business intelligence, Computer Press, Brno, 2012.
- Han, J., Kamber, M.: Data Mining - Concepts and Techniques, 2nd Edition. Morgan Kaufmann Publishers, 2006.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Bartík Vladimír, Ing., Ph.D.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2016
Datum odevzdání: 24. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Získavanie znalostí z databáz je komplexný problém zahrňujúci integráciu, prípravu dát, dolovanie znalostí metódami strojového učenia a vizualizáciu výsledkov. Práca pojednáva o celom procese získavania znalostí, špeciálne o problematike budovania dátových skladov, kde prináša návrh a implementáciu dátového skladu pre spoločnosť ROI Hunter, a.s.
V oblasti dolovania z dát sa práca zameriava na klasifikáciu a predikciu reklamných dát dostupných z pripraveného dátového skladu a to predovšetkým klasifikáciou rozhodovacím stromom. Pri predikcii vývoja nových reklám sa kladie dôraz na zdôvodnenie predikcie ako aj na návrh pre úpravu nastavení reklamy tak, aby predikcia skončila pozitívne, a teda aby s istou pravdepodobnosťou reklama v skutočnosti získala lepšie výsledky.

Abstract

Knowledge discovery from databases is a complex issue involving integration, data preparation, data mining using machine learning methods and visualization of results. The thesis deals with the whole process of knowledge discovery, especially with the issue of data warehousing, where it offers the design and implementation of a specific data warehouse for the company ROI Hunter, a.s.
In the field of data mining, the work focuses on the classification and forecasting of the advertising data available from the prepared data warehouse and, in particular, on the decision tree classification. When predicting the development of new ads, emphasis is put on the rationale for the prediction as well as the proposal to adjust the ad settings so that the prediction ends positively and, with a certain likelihood, the ads actually get better results.

Kľúčové slová

Dolovanie z dát, Získavanie znalostí, Dátový sklad, Predspracovanie dát, Klasifikácia, Predikcia, Rozhodovací strom, Inzercia, Reklama.

Keywords

Data mining, Knowledge discovery, Data warehouse, Data preprocessing, Classification, Prediction, Decision tree, Advertising, Advertisement.

Citácia

VESELOVSKÝ, Martin. *Získávání znalostí pro modelování následných akcí.* Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Bartík Vladimír.

Získávání znalostí pro modelování následných akcí

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Ing. Vladimíra Bartíka a že som uviedol všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Martin Veselovský
22. mája 2017

Poďakovanie

Chcem sa poďakovať pánovi Ing. Vladimírovi Bartíkovi za odborné vedenie diplomovej práce, praktické poznámky a užitočné odporúčania.

Chcem sa poďakovať pánovi Ing. Karlovi Tlusťákovi, riaditeľovi firmy ROI Hunter, a.s., za dôveru a príležitosť vypracovať diplomovú prácu nad reálnymi dátami z medzinárodného marketingového prostredia.

Chcem sa tiež poďakovať pánovi Ing. Pavlovi Šafaříkovi za odborné konzultácie v oblasti marketingu a pánovi Ing. Michalovi Belicovi za praktické poznámky z oblasti informatiky.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Kapitola 1

Úvod

Cieľom mnohých spoločností je rozvoj a expanzia na nové trhy, nové krajiny. S týmito krokmi zvyčajne narastá počet zamestnancov, klientov, veľkosť spracovávaných a generovaných dát, náročnosť organizácie a plánovania. Práve preto sa v niekoľkých posledných rokoch až desaťročiach upriamuje pozornosť na techniky podpory rozhodovania. Vzniká nový pojem *Business intelligence* (skr. BI) zahrňujúci podnikovú analýzu, odhaľovanie slabých stránok, plánovanie ďalšieho rozvoja a získavanie informácií dôležitých pre vedenie firmy.

Do procesu podpory rozhodovania a plánovania vstupujú informačné technológie ponúkajúce pokročilé možnosti analýzy veľkorozmerných dát. Jedná sa o relatívne mladú disciplínu získavania znalostí z databáz (angl. KDD - Knowledge Discovery in Databases), ktorej súčasťou je okrem analýzy dát aj ich predspracovanie, čistenie, transformácia, odvodzovanie nových súvislostí a v neposlednom rade vizualizácia znalostí.

Táto práca sa venuje príprave a implementácii podnikového skladu poskytujúceho kvalitné dáta na rôznych úrovniach agregácie a abstrakcie. Tento dátový sklad je následne využitý pre účely dolovania znalostí metódami strojového učenia.

Dolovanie z dát sa venuje analýze a získavaniu deskriptívnych, ale aj prediktívnych znalostí, špeciálne sa potom práca venuje klasifikácii a predikcii dát z reklamného prostredia. Pri predikcii vývoja nových dát sa pritom zohľadňuje zdôvodnenie predikcie. V prípadoch, kedy je predikcia negatívna, pristupujeme k hľadaniu takých akcií pre úpravu nastavení reklám, aby ďalší vývoj týchto reklám maximalizoval svoj potenciál.

Úvodné kapitoly tejto práce obsahujú základný teoretický výklad z oblasti získavania znalostí a dolovania z dát. Kapitola 2 popisuje celkový proces získavania znalostí a zameriava sa na metódy predspracovania dát, dolovanie z dát metódami strojového učenia a prehľad klasifikačných algoritmov spolu so spôsobmi vyhodnotenia ich úspešnosti a vzájomného porovnania. Kapitola 3 sa následne špeciálne venuje problematike budovania dátových skladov, opisuje architektonické možnosti stavby skladu a postupy implementácie.

V 4. kapitole sú zhrnuté informácie o dátach určených pre klasifikáciu, pričom sa jedná o dáta z oblasti internetovej inzercie a marketingu. Kapitola popisuje možnosti pri nastavovaní nových reklám a štatistické údaje, ktoré následne bežiaca reklama generuje.

V kapitole 5 je navrhnutá architektúra dátového skladu, jeho integrácia s operačnými podnikovými databázami a jeho konkrétna implementácia. V tejto kapitole je ďalej opísaný návrh a konkrétna implementácia procesu klasifikácie reklamných dát z pripraveného dátového skladu za účelom predikcie ďalšieho vývoja daných reklám. Pri predikcii sa navyše kladie dôraz na vysvetlenie výsledku predikcie ako aj odporučenie pre iné nastavenie niektorých atribútov, ktoré by dopomohlo k lepším výsledkom zisku z inzercie.

V kapitole 6 je porovnaná úspešnosť klasifikácie rozhodovacím stromom, naivnou bayesovskou klasifikáciou a metódou SVM.

V záverečnej kapitole 7 je zhrnutý obsah práce, jej vlastný prínos, výsledky testov a návrhy na pokračovanie a možnosti ďalšieho rozvoja tohto výskumu.

Táto diplomová práca nadväzuje na semestrálny projekt s rovnakou témou, ktorý bol autorom vypracovaný a obhájený v zimnom semestri. Semestrálny projekt sa venoval predovšetkým teoretickým východiskám pre budovanie dátového skladu a návrhu dátového skladu, pričom tieto poznatky boli prevzaté aj do diplomovej práce. Náplňou diplomovej práce bola implementácia dátového skladu, spracovanie teórie z oblasti dolovania z dát, návrh procesu klasifikácie a predikcie reklamných dát a implementácia tohto procesu spolu s implementáciou systému pre zdôvodnenie predikcie a systému pre odporučenie úpravy konkrétnych atribútov s cieľom dosiahnutia lepších budúcich výsledkov zmenou negatívnej klasifikačnej triedy na pozitívnu.

Kapitola 2

Získavanie znalostí

Získavanie znalostí z databáz je možné definovať ako netriviálne získavanie implicitných, dosiaľ neznámych a potenciálne užitočných informácií z dát [4]. Jedná sa o netriviálny proces, pre ktorý je potrebné použiť sofistikované prístupy, ktoré sú nad rámec jednoduchých SQL (Standard Query Language; štruktúrovaný vyhľadávací jazyk) dotazov na databázu. Hľadáme skryté informácie, ktoré nie sú kvôli komplexite a veľkosti zdrojových dát na prvý pohľad vidieť. Výslednou znalosťou by mala byť užitočná informácia efektne popisujúca zdrojové dáta a významná pri rozhodovaní a volení ďalších akcií.

Podstatou získavania znalostí je prepojenie niekoľkých vedných disciplín ako strojové učenie, databázové systémy, štatistika, umelá inteligencia, vizualizácia dát a vysoko náročné výpočty. Získavanie znalostí je proces, ktorý je možné chápať ako niekoľko krokov, ktoré na seba nadväzujú a v ktorých sa prelínajú jednotlivé disciplíny. [3]

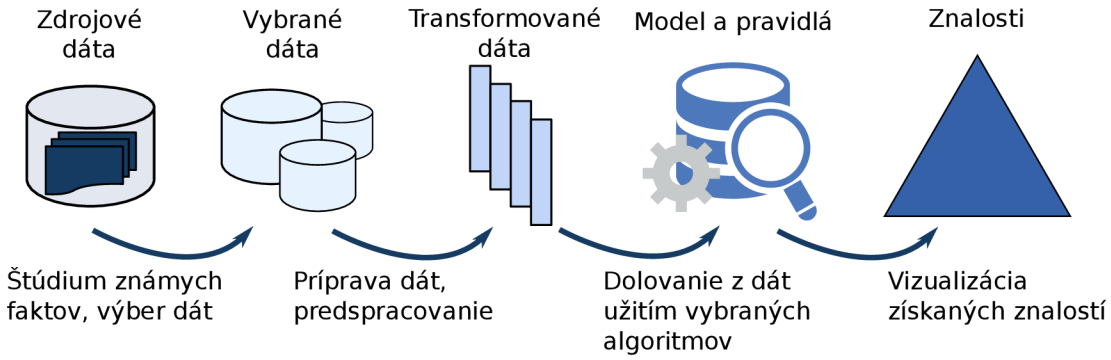
2.1 Proces získavania znalostí

Uviedli sme charakteristiky znalostí, ktoré sa snažíme získať. Hľadané znalosti môžeme ďalej rozdeliť aj podľa typu. Dolovanie z dát je kľúčovým krokom procesu získavania znalostí, kedy používame známe algoritmy z oblasti umelej inteligencie pre vytvorenie modelu nad zdrojovými dátami s cieľom odpovedať na otázku v zadaní úlohy, resp. na vyslovenú hypotézu. Úlohy dolovania môžu byť rozdelené na deskriptívne a prediktívne. Cieľom deskriptívneho dolovania je charakterizovať dáta a získať lepšie poznatky o vzťahoch, ktoré v dátach existujú. Cieľom prediktívneho dolovania je vytvoriť mechanizmus, ktorý dokáže podľa aktuálnych dát robiť predikcie o budúcich dátach.

Celkový proces získavania znalostí zahrňuje štúdium aplikačnej domény, prípravu a skúmanie dát, dolovanie aj vyhodnotenie a interpretáciu získaných vzorov. Na celkový proces získavania znalostí môžeme hľadieť ako na nasledujúce kroky [3]:

- Prvým krokom je skúmanie a pochopenie aplikačnej domény, relevantných, dopredu známych znalostí a identifikácia cieľov dolovania znalostí z pohľadu užívateľa.
- Druhým krokom je výber a príprava vzorky dát (angl. dataset), na ktorej bude vykonávaná analýza.
- Tretím krokom je čistenie dát a predspracovanie. Základnými operáciami sú odstránenie šumu, určenie stratégie na vysporiadanie sa s chýbajúcimi atribútmi a iné.

- Štvrtým krokom je redukcia dát a hľadanie užitočných vzorov, ktoré reprezentujú dáta v závislosti na cieli danej úlohy. Pomocou redukcie je možné odstrániť nerelevantné atribúty a zmenšiť tak množinu dát, s ktorou sa bude ďalej pracovať.
- Piatym krokom je výber vhodného dolovacieho algoritmu podľa definovanej úlohy a jej typu. Jedná sa napríklad o sumarizáciu, klasifikáciu, regresiu, zhlukovanie a podobne. Vybraný algoritmus je použitý k vytvoreniu modelu s vhodne zvolenými parametrami. Dolovacím algoritmom a technikám je venovaná kapitola 2.3.
- Šiestym krokom je interpretácia a vizualizácia vydolovaných vzorov a modelov. Pri prediktívnych úlohách je tiež zdôraznená úspešnosť predikcie pre nové dáta s cieľom vyzdvihnúť dôveryhodnosť modelu.
- Posledným krokom je priame použitie novej znalosti alebo natrénovaného predikčného modelu, predanie novej znalosti do iného systému za účelom ovplyvnenia ďalšej akcie alebo dokumentácia a oznámenie získaných vedomostí ďalším osobám.



Obr. 2.1: Proces získavania znalostí.

2.2 Predspracovanie dát

Proces získavania znalostí si vyžaduje dáta v kvalitnom stave, pretože nie je neobvyklé, že v klasických podnikových databázach sa vyskytujú nekonzistencie, šum a chýbajúce položky. K tomuto stavu typicky dochádza kvôli obrovským rozmerom databáz, ktoré môžu mať potenciálne viac zdrojov, napr. užívateľský vstup vyplnením formuláru. Kvalitné dáta vedú k lepším výsledkom a menšej chybovosti pri získavaní znalostí, preto je typické, že pred dolovaním znalostí sú dáta najskôr predspracované.

Hlavným dôvodom predspracovania sú však veľmi často špecifické nároky dolovacieho algoritmu na reprezentáciu vstupných dát. Niektoré dolovacie algoritmy nedokážu pracovať s istým typom dát vôbec, niektoré to naopak dokážu, ale k efektívnejšiemu chodu algoritmu by bolo vhodné dáta upraviť do prijateľnejšej podoby.

V rámci predspracovania býva často uplatnená transformácia dát, ako napríklad normalizácia. Normalizácia numerických hodnôt do určeného intervalu často zvyšuje presnosť algoritmov založených na určovaní vzdialenosti objektov v priestore. Iné časté transformácie sú prevody medzi typmi, typicky numerické atribúty na nominálne a naopak.

2.2.1 Čistenie dát

Predspracovanie typicky začína procesom čistenia dát, ktorý zahrňuje vyplnenie chýbajúcich hodnôt v databáze, detekciu odľahlých hodnôt, prípadné vyhladenie odľahlých hodnôt, napríklad spriemerovaním atribútov, a tiež opravu nekonzistencií v dátach ako napríklad rozdielne merné jednotky.

2.2.2 Ošetrenie chýbajúcich hodnôt

Niektoré dolovacie algoritmy sú schopné vyrovnať sa s chýbajúcimi hodnotami, ale nie je to pravidlo a často je lepšie uvažovať o ošetrení chýbajúcich hodnôt ešte pred dolovaním. Existuje niekoľko možností ako sa zachovať pri chýbajúcich dátach pri výbere vstupu do vybraného procesu dolovania.

- Ignorujeme databázovú položku, ktorej chýba atribút potrebný pre analýzu. Táto možnosť je najjednoduchšia, ale pokiaľ je vo vstupnej databáze takýchto položiek veľa, tak množstvo dát na vstupe do dolovacieho procesu je výrazne zredukované.
- Manuálne doplnenie chýbajúcich atribútov, ktoré je však časovo náročné a v prípade veľkých databáz môže byť dokonca neuskutočniteľné.
- Doplnenie chýbajúcej hodnoty globálnou konštantou, napr. "neznáma hodnota". Tento postup však nie je príliš vhodný, lebo dolovacie algoritmy môžu nadobudnúť mylný pocit, že takto doplnené položky majú spoločnú črtu, čo v skutočnosti nemusí byť pravda.
- Doplnenie chýbajúcej hodnoty priemernou hodnotou v rámci databázy pre daný atribút.
- Doplnenie chýbajúcej hodnoty priemernou hodnotou v rámci danej triedy, do ktorej patrí daný atribút. Pokiaľ sa v databáze vyskytujú známe triedy dát, tak vypočítame priemernú hodnotu atribútu pre každú triedu zvlášť.
- Doplnenie nájdením najpravdepodobnejšej hodnoty na základe ostatných atribútov danej položky. Pre hľadanie vhodnej hodnoty je tak možnosť využiť niektorý z dolovacích algoritmov z kapitoly 2.3 už vo fáze predspracovania.

2.2.3 Vyhladenie šumu v dátach

Šum je náhodná chyba alebo odchýlka hodnoty atribútu, ktorá môže vzniknúť chybou človeka, ale aj chybou v programe. Hodnotu, ktorá sa od iných významne líši, potom nazývame odľahlou hodnotou. Techniky pre vyhladenie šumu nahrádzajú aktuálne hodnoty atribútu inými hodnotami podľa danej techniky.

- Vyhladenie hodnôt pomocou plnenia zoradenej postupnosti numerických atribútov do tzv. košov (angl. binning) rovnakej veľkosti. V každom koši sú potom všetky hodnoty vyhladené, tj. nahradené, hodnotou priemeru daného koša alebo hraničnými hodnotami daného koša.
- Vyhladenie hodnôt zostrojením regresnej krivky a nahradením aktuálnych hodnôt hodnotami zostrojenej funkcie.

- Odhalenie odľahlých hodnôt metódou zhlukovania. Hodnoty, ktoré nebudú patriť do žiadneho zhluku, sú potom typicky označené ako odľahlé. Určenie veľkosti zhlukov a vzdialenosti, ktorú považujeme za priveľkú, však môže byť problematické. Často je potrebné nastavovať parametre zhlukovacej metódy empiricky.

2.2.4 Transformácia dát

Normalizácia

Normalizácia je transformácia, ktorá upravuje hodnoty vybraného numerického atribútu tak, aby spadali do určeného intervalu hodnôt.

Min-max normalizácia lineárne transformuje pôvodné hodnoty vzťahom

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

kde min_A a max_A sú minimálna a maximálna hodnota atribútu A. Min-max normalizácia mapuje hodnotu atribútu A do intervalu $< min_A, max_A >$.

Z-score normalizácia transformuje pôvodné hodnoty atribútu A na základe priemeru $\bar{A}$ a štandardnej odchýlky σ_A hodnôt atribútu A. Normalizovaná hodnota v' hodnoty v je určená vzťahom

$$v' = \frac{v - \bar{A}}{\sigma_A}$$

Táto metóda normalizácie je užitočná, keď pôvodná minimálna a maximálna hodnota sú neznáme alebo keď atribút obsahuje odľahlé hodnoty, ktoré dominujú v min-max normalizácií.

Normalizácia dekadickou zmenou mierky transformuje hodnotu posúvaním desatinnej čiarky hodnoty atribútu A. Počet posunutých desatinných miest závisí na maximálnej hodnote atribútu. Normalizovaná hodnota v' je vypočítaná z hodnoty v vzťahom

$$v' = \frac{v}{10^j}$$

kde j je najmenšie celé číslo také, že $max(|v^{'}|) < 1$.

Diskretizácia numerických dát

Pri získavaní znalostí často dochádza k potrebe diskretizovať numerické atribúty. Prvým dôvodom je, že niektoré dolovacie algoritmy pracujú len s nominálnym typom atribútov. Druhým dôvodom je, že nás často zaujímajú len intervaly numerických hodnôt, pre ktoré platí nejaký vzťah.

Štandardnou technikou diskretizácie je plnenie do košov (angl. binning), podobne ako v prípade vyhladenia zašumených dát. Všetkých n utriedených hodnôt numerického atribútu rozdelíme do k košov, pričom používame dva spôsoby plnenia do košov.

- uvažujeme pevnú šírku každého koša, tj. pevne určený interval, kde v každom koši môže byť 0 až n hodnôt tak, aby počet prvkov vo všetkých košoch bol rovný n
- prvky rozdelíme do košov podľa frekvencie tak, že koše nemusia reprezentovať rovnako široký interval, ale na rozdiel od prvého spôsobu bude v každom koši n/k hodnôt

Pôvodný numerický atribút je ďalej reprezentovaný ako nominálny atribút, ktorý určuje interval, kam patrí pôvodná numerická hodnota.

Kódovanie kategorických atribútov

Atribúty, ktoré môžu nadobudnúť len dopredu definovaný počet hodnôt (kategórií), označujeme ako kategorické. Tieto kategórie bývajú často reprezentované ako množina slov. Jednoduchou transformáciou je však možné každej kategórii priradiť číslo a v ďalších krokoch pracovať len s touto výpočtovo jednoduchšou reprezentáciou, ktorú niektoré dolovacie algoritmy dokonca explicitne požadujú.

Ak sú kategórie reprezentované číselne, tak je potrebné najskôr overiť možnosti algoritmu vzhľadom k práci s kategorickými atribútmi, inak môže dôjsť ku skutočnosti, kedy algoritmus na číselne vyjadrených kategóriách hľadá usporiadanie. Možným riešením je v tomto prípade tzv. **1 z *n* kódovanie**, kedy pre n kategórií vytvoríme n rozmerné pole čísel, kde všetky čísla okrem jedného budú nulové. Index jediného nenulového čísla v poli potom reprezentuje danú kategóriu.

2.3 Dolovanie z dát

Dolovanie z dát môžeme rozdeliť na niekoľko kategórií podľa typu dolovania a použitých dolovacích algoritmov. Táto kapitola poskytuje stručný prehľad typov algoritmov, pričom sa zameriava hlavne na klasifikačné algoritmy.

2.3.1 Charakterizácia a diskriminácia

V prípade, že dáta sú asociované s určitou triedou alebo konceptom, tak hľadáme popis danej triedy. Využívame sumarizáciu obecných vlastností analyzovanej triedy a dolujeme charakteristické vlastnosti. Z hľadiska diskriminácie nás zaujíma špecifikácia vlastností, ktoré oddeľujú jednu triedu od druhej.

2.3.2 Zhluková analýza

Zhlukovanie patrí medzi metódy umelej inteligencie pre strojové učenie bez učiteľa (angl. unsupervised) s cieľom rozdeliť dáta do k skupín podľa hodnôt atribútov jednotlivých položiek. Pracujeme s metódami založenými na rôznych princípoch, akými sú metódy založené na rozdeľovaní (k-means), hierarchické metódy (Chameleon), metódy založené na hustote (DBSCAN), metódy založené na mriežke (WaveCluster) alebo metódy založené na modeloch (SOM).

2.3.3 Klasifikácia a predikcia

V prípade, že dáta sú asociované s určitou triedou, podobne ako pri charakterizácii, zaujíma nás existencia takých charakteristických vlastností, aby bolo možné odvodiť príslušnú triedu zo zvyšných atribútov danej položky. V prípade vytvorenia takéhoto modelu je potom možné predikovať príslušnú triedu pre položky, ktoré neboli súčasťou trénovania modelu. Klasifikačné metódy sa kategorizujú podľa kritérií presnosti, rýchlosti či škálovateľnosti.

Dôležitou vlastnosťou je tiež interpretovateľnosť, aj keď niekedy na úkor presnosti. Vhodným reprezentantom dobre interpretovateľného klasifikačného algoritmu je algoritmus rozhodovacieho stromu (Decision tree), ktorý vytvára model na základe informačnej entrópie jednotlivých atribútov. Výsledkom je zaradenie konkrétnej položky do triedy podľa jej atribútov, pričom z modelu je možné vyčítať, na základe akých rozhodnutí bolo toto zaradenie vykonané.

Matica zámien		**Predikcia**	
		negatívna	pozitívna
Skutočnosť (True)	negatívna	Skutočne negatívna (TN)	Falošne pozitívna (FP)
	pozitívna	Falošne negatívna (FN)	Skutočne pozitívna (TP)

Tabuľka 2.1: Matica zámien (angl. confusion matrix).

Pri voľbe klasifikačného algoritmu sa riadime spomenutými parametrami - presnosť, rýchlosť, škálovateľnosť, interpretovateľnosť. Preto pred výberom konkrétneho algoritmu je vhodné študovať všetky možnosti a hľadať taký algoritmus, ktorý najviac vyhovuje našim potrebám.

Hodnotenie úspešnosti klasifikátorov

Pre vzájomné porovnanie úspešnosti klasifikátorov sa používajú konkrétne metriky. Úspešnosť je možné vyhodnotiť takým spôsobom, že pre trénovanie modelu vyberieme len časť trénovacích vzoriek a druhú časť použijeme pre testovanie modelu. Keďže príslušnosť testovacích vzoriek ku klasifikačnej triede je dopredu známa, tak je možné porovnať výsledok predikcie so skutočným zaradením vzorky do triedy. Na základe tohto testovania je definovaných niekoľko metrík [11].

Matica zámien (angl. confusion matrix), tabuľka 2.1, sleduje počty správne a nesprávne klasifikovaných vzoriek a poskytuje tak prehľad úspešnosti klasifikátora pri predikcii klasifikačnej triedy pre testovacie vzorky, tj. vzorky, ktoré neboli použité pri trénovaní klasifikátora.

Presnosť (angl. precision) je percento správne klasifikovaných pozitívnych vzoriek.

$$precision = \frac{TP}{TP + FP}$$

Úplnosť (angl. recall) je percento pozitívnych vzoriek správne klasifikovaných do svojej skutočnej triedy.

$$recall = \frac{TP}{TP + FN}$$

Správnosť (angl. accuracy) je percento vzoriek správne klasifikovaných do svojej skutočnej triedy.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

F-metrika (angl. F-measure, F1-score) je vážený priemer presnosti a úplnosti.

$$accuracy = 2 * \frac{precision * recall}{precision + recall}$$

ROC graf (angl. Receiver operating characteristics) je technika pre vizualizáciu, organizáciu a výber klasifikátorov na základe ich úspešnosti. [2]. Os y v ROC grafe vyjadruje *senzitivitu*, čo je pomer medzi počtom správne klasifikovaných pozitívnych vzoriek a všetkých pozitívnych vzoriek. Senzitivita je metrikou úplnosti $sensitivity = \frac{TP}{TP+FN}$. Os x v ROC

grafe vyjadruje *špecifickosť*, čo je schopnosť klasifikátora vylúčiť falošne negatívne výsledky, $specifity = \frac{TN}{TN+FP}$.

ROC graf vyjadruje pomer senzitivity a špecifickosti, teda pomer medzi množstvom falošne pozitívnych a falošne negatívnych pozorovaní. Výstupom diskrétneho klasifikátora je bod v ROC priestore, ale výstupom niektorých klasifikátorov je, alebo môže byť, aj pravdepodobnosť príslušnosti do klasifikačnej triedy. Ak je takýmto klasifikátorom špecifikovaný *prah* pravdepodobnosti, od ktorého je vzorka považovaná za pozitívnu, tak vzniká diskrétny klasifikátor. Rôzne nastavenie prahu naviac generuje iný bod v ROC priestore, a preto je možné so zmenou prahu v priestore vykresliť krivku. ROC analýza je preto okrem vizualizácie vzťahu falošne pozitívnych a falošne negatívnych pozorovaní vhodná najmä k určeniu optimálneho prahového bodu.

2.3.4 Klasifikácia rozhodovacím stromom

Spôsob reprezentovania znalostí v podobe rozhodovacích stromov je dobre známy z rôznych oblastí, napríklad pri kategorizácií rastlín a živočíchov v biológii. Pri tvorbe rozhodovacieho stromu sa uplatňuje metóda *rozdeľ a panuj*, kde trénovacie dáta sú postupne rozdeľované na menšie a menšie podmnožiny tak, aby v týchto podmnožinách prevládali položky jednej triedy. Rozhodovací strom je graf stromovej štruktúry, kde každý vnútorný uzol reprezentuje test hodnoty istého atribútu a koncové uzly reprezentujú triedu, do ktorej je daný objekt klasifikovaný. Takýto strom je okrem iného možné ľahko previesť na jednoduché pravidlá.

Pre klasifikáciu pomocou rozhodovacieho stromu je kľúčové vytvoriť vhodný strom. Preto najdôležitejšou otázkou je, ako vybrať atribút pre vetvenie stromu, pričom zrejmým cieľom je vybrať taký atribút, ktorý čo najviac odlíši položky patriace do rôznych tried. Vodítkom pre voľbu sú charakteristiky atribútu prevzaté z teórie informácie a pravdepodobnosti, a to *entrópia*, *informačný zisk*, *Gini index* a ďalšie [6].

Entrópia a informačný zisk - ID3

Prvým algoritmom pre selekciu vhodného atribútu pre vetvenie rozhodovacieho stromu je algoritmus ID3, ktorý je založený na informačnom zisku atribútov. Pre vetvenie stromu je vybraný atribút s najvyšším informačným ziskom, pričom informačný zisk je vypočítaný pre každý atribút ako rozdiel entrópie aktuálnej množiny trénovacích vzoriek a entrópie množiny trénovacích vzoriek po rozdelení daným atribútom. Entrópia je pojem používaný v prírodných vedách pre vyjadrenie miery neusporiadanosti. Matematicky je celková entrópia množiny vzoriek D vyjadrená ako

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

kde p_i je pravdepodobnosť, že vzorka náleží do triedy C_i a m značí počet klasifikačných tried. Logaritmus so základom dva je použitý, pretože informácia je zakódovaná v bitoch.

Entrópia množiny vzoriek D po rozdelení atribútom A, ktorý nadobúda v možných hodnôt, je vyjadrená ako

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

kde D_j vyjadruje množinu vzoriek, v ktorých hodnota atribútu A je a_j.

Pomerný informačný zisk - C4.5

Informačný zisk preferuje selekciu atribútov, ktoré majú veľa hodnôt. Atribút, ktorý je unikátny v každej trénovacej vzorke, by bol dokonca zvolený ako bezchybný kandidát pre rozdelenie množiny vzoriek, avšak rozdelenie množiny na základe takéhoto atribútu neprináša žiadny úžitok. Algoritmus C4.5 vychádza z algoritmu ID3, ale informačný zisk normalizuje.

Gini index

Ako kritérium pre voľbu atribútu je možné použiť aj Gini index, ktorý je rovnako ako entrópia minimálny, ak všetky vzorky patria do jednej z tried a maximálny, ak sú rovnomerne rozložené. Výhodou tohto indexu je, že nie je potrebné počítať logaritmus. Na druhú stranu nevýhodou je, že pomocou Gini indexu je možné vytvárať len binárne stromy, tj. stromy, v ktorých vedú z každého uzlu maximálne dve vetvy. Gini index pre množinu vzoriek D a m klasifikačných tried je vyjadrený ako

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

kde p_i je pravdepodobnosť, že vzorka náleží do triedy C_i.

Gini index množiny vzoriek D po rozdelení atribútom A, ktorý nadobúda v možných hodnôt, je vyjadrený ako

$$Gini_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Gini(D_j)$$

Pre vetvenie stromu je potom vybraný atribút A, u ktorého je rozdiel $Gini(D)$ a $Gini_A(D)$ najväčší.

Orezanie stromu

Po vytvorení rozhodovacieho stromu môžu niektoré vetvy predstavovať anomálie v trénovacích dátach kvôli šumu alebo odľahlým hodnotám. Algoritmy pre orezanie stromu riešia tento problém, označovaný aj ako *pretrénovanie modelu*, tzn. model je málo obecný. Orezané stromy sú zväčša menšie, menej zložité a preto aj rýchlejšie pri tej istej alebo lepšej schopnosti klasifikovať testovacie vzorky.

Existujú dva prístupy pre orezanie stromu. Orezanie sa vykonáva priamo pri vytváraní stromu alebo až po skončení tvorby stromu. Častejším prístupom je orezávanie stromu po skončení tvorby. Algoritmus orezania používaný po konštrukcii stromu pomocou Gini indexu je *cena zložitosti*, kde sa počíta chybovosť klasifikácie pre konkrétne podstromy. Následne je vytvorených niekoľko variant orezaného stromu a ako výsledok je vybraný podstrom s najmenšou cenou zložitosti, teda strom, ktorý produkuje najmenej chýb pri klasifikácii [6].

2.3.5 Jednoduchá bayesovská klasifikácia

Bayesovská klasifikácia je založená na podmienenej pravdepodobnosti a Bayesovom teoréme [6]. Tento teorém vyjadruje *posteriórnu pravdepodobnosť* $P(c_j|X)$ príslušnosti príkladu X do kategórie c_j na základe popisu atribútov $x_0, ..., x_i$ príkladu X. Teorém je vyjadrený nasledovným vzťahom.

$$P(c_j|X) = \frac{P(X|c_j)P(c_j)}{P(X)}$$

Podobne, $P(X|c_j)$ je posteriórna pravdepodobnosť toho, že príklad X obsahuje konkrétne hodnoty atribútov $x_0, ..., x_i$ na základe kategórie c_j. Túto pravdepodobnosť určíme podľa vzorca o podmienenej pravdepodobnosti [9].

$$P(X|c_j) = \frac{P(X \cap c_j)}{P(c_j)}$$

Pre každý atribút $x_0, ..., x_i$ teda vyjadríme túto pravdepodobnosť ako podiel súčasného výskytu konkrétnej hodnoty atribútu s kategóriou c_j a pravdepodobnosti celkového výskytu kategórie c_j. Ak sú hodnoty atribútu spojité, tak je možné pre výpočet pravdepodobnosti využiť Gaussovo rozloženie, strednú hodnotu a štandardnú odchýlku [1].

Pre klasifikáciu príkladu X do konkrétnej kategórie sú porovnané posteriórne pravdepodobnosti všetkých kategórií a príklad X je klasifikovaný do kategórie c_j, ktorá má túto pravdepodobnosť najvyššiu.

Tento algoritmus sa nazýva jednoduchým, resp. naivným, pretože predpokladá, že jednotlivé atribúty sú navzájom nezávislé.

2.3.6 Klasifikácia metódou SVM

Aloritmus support vector machines (SVM) je založený na hľadaní lineárnej hranice medzi vzorkami z rôznych tried. Základným princípom je prevod pôvodného vstupného priestoru do iného, viacdimenzionálneho. Takže, ak triedy nie sú lineárne separovateľné v danom priestore, tak sú prevedené do priestoru s vyššou dimenziou, kde lineárne separovateľné už budú.

Algoritmus hľadá takú lineárnu hranicu medzi triedami, ktorá zabezpečí najmenšiu chybovosť klasifikácie nových vzoriek. Optimálna hranica je taká, ktorá poskytuje najširšie pásmo medzi vzorkami z rôznych tried. SVM je často veľmi presný, ale je viac náchylný k pretrénovaniu modelu [6].

2.3.7 Klasifikácia neurónovými sieťami

Klasifikácia pomocou neurónových sietí je založená na sieti vzájomne prepojených neurónov, ktoré sú reprezentované aktivačnou funkciou a číselnými váhami pre každý vstup neurónu. Algoritmov založených na neurónových sieťach je mnoho ale najznámejším je algoritmus *backpropagation* pracujúci so sieťou neurónov, ktorú je možné vyjadriť acyklickým grafom. Táto sieť sa skladá z niekoľkých vrstiev neurónov, kde prvá vrstva sa označuje ako vstupná, posledná ako výstupná a vrstvy medzi nimi ako tzv. *skryté* vrstvy. Pri klasifikácii vzoriek s k atribútmi obsahuje vstupná vrstva k neurónov a pri výstupnej vrstve záleží od počtu klasifikačných tried.

Informácia v neurónovej sieti sa šíri zo vstupnej vrstvy na výstupnú, pričom šírenie je ovplyvnené aktiváciou jednotlivých neurónov. Trénovanie napokon prebieha tak, že ak na výstupnej vrstve došlo k chybnej klasifikácii, tak je informácia o chybe šírená spätne od výstupnej vrstvy až po vstupnú a v každom neuróne sú na základe chyby upravené váhy.

Trénovanie neurónovej siete je väčšinou časovo náročnejšie, ale následná predikcia nových dát má potenciál vysokej presnosti. Hlavnou nevýhodou neurónových sietí je nízka interpretovateľnosť znalostí obsiahnutých v sieti [6].

2.3.8 Klasifikácia metódou k najbližších susedov

Pri klasifikácií metódou k najbližších susedov (angl. k-nearest-neighbor, k-NN) je každá vzorka opísaná n atribútmi a reprezentuje jeden bod v n-dimenzionálnom priestore. Pri klasifikácii novej vzorky sa v priestore obsahujúcom trénovacie vzorky hľadá k najbližších bodov (vzoriek) a nová vzorka je následne klasifikovaná do triedy, ktorá je najviac zastúpená u vybraných susedov. Vzdialenosť dvoch vzoriek je určená vybranou metrikou, napr. Euklidovskou [6].

Kapitola 3

Dátový sklad

Dátový sklad (angl. data warehouse, DW; ďalej často len ako sklad) je databáza obsahujúca kolekciu dát určenú pre podporu rozhodovacích procesov na základe vykonávania analýz a dolovania nových znalostí z tejto kolekcie dát. Dátový sklad má nasledovné vlastnosti [7]:

Je subjektovo orientovaný. Dáta v dátovom sklade sa vzťahujú k podnikovo špecifickým hlavným subjektom ako napr. zákazník, produkt, predaje, objednávky. Každé oddelenie spoločnosti sa typicky zameriava na iný subjekt, ale rozdiel oproti operačným databázam je v tom, že v dátovom sklade sú len dáta, ktoré je možné použiť pre rozhodovanie.

Je integrovaný a konzistentný. Integrácia dát je kľúčovou vlastnosťou dátového skladu, pretože dáta sú do skladu pridávané z viacerých rozličných zdrojov. Tieto zdroje môžu obsahovať množiny dát, ktoré sa vzťahujú k rovnakému subjektu. Preto sa ešte pred uložením do skladu hľadajú vzťahy medzi množinami. Zároveň má často každý zdroj dát iné konvencie pre názvy položiek, štruktúru dát, merné jednotky, formáty dát (napr. formát dátumu) a iné fyzické charakteristiky dát, pretože pri návrhu týchto úložísk sa dopredu nepredpokladalo zjednotenie dát s inými databázami. Tieto konflikty je potrebné pred uložením do dátového skladu vyriešiť konverziou, transformáciou a sumarizáciou. Výsledkom je jedna fyzická reprezentácia všetkých užitočných podnikových dát pre podporu rozhodovania.

Obsahuje nemenné dáta. Dáta do dátového skladu sú pravidelne nahrávané a pristupované, ale nedochádza k aktualizácii skôr uložených dát. Nahrávanie nových dát prebieha len v určenom intervale a v čase, keď so skladom nepracuje žiadny užívateľ, to znamená, že dátový sklad je v podstate možné chápať ako databázu určenú len na čítanie. Dáta by zo skladu nemali byť nikdy zmazané, a preto je pri nových zmenách vytvorený snímok nových dát, ktorý je uložený a nezmení predošlé dáta. Týmto spôsobom v sklade zostáva história dát.

Obsahuje dáta závislé na čase. Každá jednotka dát má priradený časový údaj, ktorý jednoznačne určuje čas, kedy boli dané hodnoty aktuálne. Sklad je sofistikovanou sériou pravidelne vytvorených snímkov údajov operačných databáz. Čas je v dátovom sklade kľúčovou veličinou.

3.1 Výhody dátového skladu

Dátový sklad je označenie pre podnikovú databázu určenú primárne pre analytické účely a pre podporu rozhodovania na základe uložených faktov. Dôležitou vlastnosť skladu je, že je to databáza oddelená od operačných databáz podniku, pričom poskytuje integrovaný pohľad na dáta v operačných databázach všetkých organizačných jednotiek spoločnosti za účelom analýzy [10]. Oddelená databáza poskytuje nasledujúce výhody:

- predchádzanie zvýšenej záťaže operačných databáz analytickými dotazmi, ktoré môžu byť často veľmi náročné na výpočet
- integrácia viacerých databáz a jednotné rozhranie pre analytické dotazy, ktoré by boli veľmi náročné, niekedy dokonca neuskutočniteľné, pokiaľ pracujeme z viacerými zdrojmi dát
- predspracované dáta, ktoré sú konzistentné a kvalitné

Ďalšie rozdiely dátového skladu oproti operačným databázam sú spojené s typom úkonu. Operačné úkony vykonávajú transakcie, ktoré obecne zapisujú / čítajú malé množstvo položiek z / do niekoľkých tabuliek spojených jednoduchými vzťahmi. Tento typ úkonov sa označuje ako *On-Line Transaction Processing (OLTP)*. Úkony (požiadavky / otázky) na dátový sklad sa označujú ako *On-Line Analytical Processing (OLAP)*. OLAP úkony slúžia k multidimenzionálnej analýze, ktorá potrebuje prejsť obrovským množstvom záznamov v databáze a vytvoriť množiny agregovaných dát.

3.2 Architektúry dátového skladu

Nasledujúce architektonické vlastnosti sú kľúčové pre systém dátového skladu [8]:

- **Separácia.** Ako bolo spomenuté už v predošlej kapitole, oddelenie dátového skladu od operačných databáz prináša niekoľko výhod.
- **Škálovateľnosť.** Hardware a software dátového skladu by mali byť ľahko rozšíriteľné v súvislosti so zvyšovaním veľkosti dát a nárastu užívateľských požiadaviek. Dátový sklad by mal byť v takom stave, aby bolo možné rozložiť záťaž na viac serverov.
- **Rozšíriteľnosť.** Architektúra skladu by mala byť navrhnutá vzhľadom k možnosti pridávania nových dátových zdrojov a funkcií bez nutnosti zmeny návrhu danej architektúry.
- **Bezpečnosť.** Dátový sklad obsahuje strategické dáta podniku, a preto by mal byť prístup k skladu zabezpečený a kontrolovaný.
- **Spravovanie.** Správa dátového skladu by nemala byť príliš zložitá.

Jednovrstvová architektúra

V tejto architektúre je dátový sklad realizovaný ako služba pracujúca priamo nad dátami v operačných databázach. Takýto sklad je označovaný ako *virtuálny* sklad, pretože poskytuje multidimenzionálne pohľady na operačné dáta, ale nepracuje s vlastnou databázou. Analytické dotazy sú najskôr interpretované skladom a následne postúpené operačnej databáze.

Táto architektúra vytvára minimálne požiadavky na uloženie dát pre analýzu (pretože sa využívajú dáta v operačnej databáze), ale nespĺňa požiadavku na separáciu dátového skladu. Nevýhodou je teda, že analytické dotazy môžu ovplyvňovať výkon operačnej databázy.

Dvojvrstvová architektúra

Oddelenie databáz hrá kľúčovú úlohu pri definovaní typickej architektúry dátového skladu. Hoci sa táto architektúra nazýva dvojvrstvová pre zvýraznenie oddelenia od operačných databáz, tak v skutočnosti sa skladá zo štyroch vrstiev [5]:

1. **Zdrojová vrstva.** Heterogénne zdroje dát ako sú relačné databázy, dokumentové databázy, textové súbory, multimediálne súbory a podobne.

2. **Vrstva ETL** ETL je skratka pre *Extraction, Transformation, and Loading tools*, čo je proces zahrňujúci extrakciu dát, aplikovanie požadovaných transformácií a následné nahranie upravených dát do ďalšieho kroku, ktorým je v tomto prípade uloženie dát do dátového skladu.

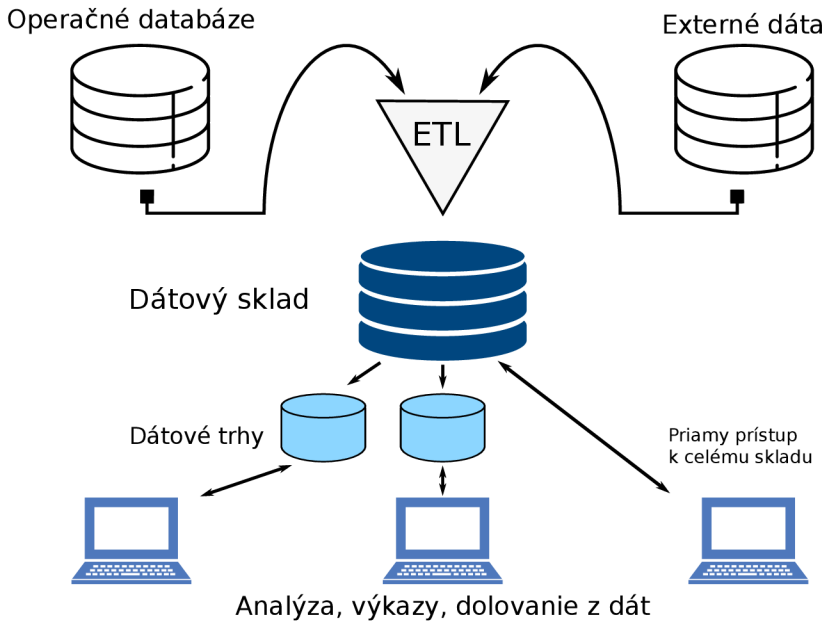
3. **Vrstva dátového skladu.** Informácie sú uložené do jedného, logicky centralizovaného repozitára - dátového skladu. Okrem samotných dát sú v sklade uložené aj tzv. *meta dáta*, ktoré obsahujú informácie o zdrojoch dát, použitých transformáciách, užívateľoch, organizačných jednotkách a podobne.

   Dátový sklad môže byť používaný priamo alebo nepriamo vytvorením tzv. *dátových trhov*, ktoré sú dizajnované pre jednotlivé organizačné jednotky podniku. Dátový trh (ang. data mart) je podmnožinou alebo agregáciou dát uložených v primárnej databáze skladu. Zahrňuje časť informácií, ktoré sú relevantné pre špecifickú podnikovú oblasť, oddelenie alebo skupinu užívateľov. Výhodou takýchto trhov je, že vybraní užívatelia majú obmedzený prístup k dátam celého podniku, resp. pracujú len s informáciami, ktoré sú pre nich užitočné. Oddelený dátový trh takisto poskytuje možnosti lepšej škálovateľnosti a rýchlosti spracovania užívateľských dotazov.

4. **Analýza.** V tejto vrstve sú dáta efektívne a flexibilne pripravené pre vytváranie výkazov, simuláciu hypotetických podnikových scenárov a dynamickú analýzu. Technologicky sa jedná o vrstvu poskytujúcu prívetivé grafické užívateľské rozhranie.

Trojvrstvová architektúra

Treťou vrstvou oproti dvojvrstvovej architektúre je vrstva *urovnaných dát*, ktorá sa nachádza medzi prvou a druhou vrstvou dvojvrstvovej architektúry a obsahuje materializované operačné dáta po integrácii zdrojových dát a čistení dát. Táto definícia v podstate spĺňa požiadavky na dátový sklad, ale rozdiel je v tom, že túto vrstvu môžu používať aj operačné procesy podniku, a teda býva použitá aj na iné účely ako na analýzu. Dátový sklad je potom vytvorený z tejto vrstvy. Ostatné vrstvy sú rovnaké ako v dvojvrstvovej architektúre.



Obr. 3.1: Architektúra systému dátového skladu. Dáta, integrované z rôznych zdrojov, sú extrahované, transformované a načítané (ETL) do dátového skladu. Užívateľ pracuje s celým dátovým skladom alebo pre neho relevantnou podmnožinou dostupnou v dátovom trhu.

3.3 Multidimenzionálny model

Dátové sklady a OLAP nástroje sú založené na multidimenzionálnom modele dát. Tento model nazerá na dáta vo forme n-dimenzionálnej kocky a umožňuje interaktívne dolovanie na rozličných úrovniach abstrakcie. Model je typicky organizovaný podľa hlavnej témy, ktorá je reprezentovaná tabuľkou faktov. **Fakty** sú numerické metriky ako napr. počet predajov, výška rozpočtu, výška zisku a pod. Tabuľka faktov môže byť analyzovaná z rôznych pohľadov, ktoré nazývame **dimenzie**. Každá dimenzia môže byť modelovaná ako tabuľka dimenzií, ktorá je spojená s tabuľkou faktov. Dimenziou môže byť napr. čas, ktorý môžeme uvažovať na rôznych stupňoch abstrakcie: hodina, deň, mesiac, rok a podobne.

V prípade tabuľky faktov a n dimenzií, modelovaných n tabuľkami, vzniká n-dimenzionálna dátová kocka. Hlavnou výhodou kocky je možnosť rôznych náhľadov na dáta podľa toho, ktoré dimenzie nás práve zaujímajú.

Príklad: V tabuľke faktov je numerický atribút zisk. Existuje jedna dimenzia čas. V prípade analýzy celkového zisku bude výsledkom suma zisku. V prípade analýzy zisku za posledný rok bude výsledkom suma zisku, avšak obmedzená v dimenzii čas, a teda výsledkom je suma zisku za zvolené obdobie.

Schéma databázových tabuliek

V prípade relačnej databázy často vyjadrujeme multidimenzionálny model pomocou ER diagramu. Existujú tri základné schémy, a to schéma hviezdy, schéma snehovej vločky a schéma súhvezdia [6].

Schéma hviezdy je základnou schémou, kde tabuľka faktov leží v strede a okolo nej sú naviazané tabuľky dimenzií. Centrálna tabuľka, tabuľka faktov, obsahuje veľké množstvo dát bez redundancie.

Schéma snehovej vločky zavádza normalizáciu tabuliek dimenzií. Pre niektoré tabuľky dimenzií sú vytvorené nové tabuľky s cieľom znížiť redundanciu.

Schéma súhvezdia obsahuje viac než jednu tabuľku faktov, takže modeluje viac než jednu dátovú kocku. Fakty pritom medzi sebou môžu niektoré dimenzie zdieľať.

OLAP operácie

Pri práci s dátovou kockou využívame nasledovné OLAP operácie:

- **Roll-up** operácia vytvára agregáciu na dátovej kocke a to znížením počtu dimenzií alebo zvolením vyššej abstrakcie v rámci hierarchie niektorej dimenzie. Ak sa zaujímame o celkovú sumu niektorého atribútu tabuľky faktov, tj. nezaujíma nás žiadna dimenzia, tak sa jedná o maximálnu agregáciu. Ak vezmeme do úvahy niektorú dimenziu, napr. čas, tak v jej hierarchii môžeme postúpiť vyššie, napr. z mesiaca na rok.
- **Drill-down** operácia je opakom predchádzajúcej a zaujímame sa o detailnejšie hodnoty. Pridávame teda nové dimenzie, čím obmedzujeme výsledok alebo meníme mieru abstrakcie v aktuálne použitých dimenziách.
- **Slice & Dice** je operácia selekcie nad jednou, resp. viacerými dimenziami podľa zvoleného kritéria.
- **Pivot** je operácia za účelom zmeny vizualizácie výsledku otočením pohľadu na dimenzie, tj. otočenie ôs v 2-D kocke.

3.4 Postupy implementácie dátového skladu

Dátový sklad je často realizovaný pomocou relačnej databázy, ale server schopný pracovať s OLAP operáciami, môže byť založený aj na inej databáze. Uvedieme tri spôsoby:

- Relačný OLAP (ROLAP) server je realizovaný relačnou databázou a pre poskytovanie OLAP operácií mapuje tieto operácie na štandardné SQL dotazy pomocou klauzúl WHERE a GROUP BY.
- Multidimenzionálny OLAP (MOLAP) server podporuje náhľady na dimenzie pomocou databázových štruktúr založených na poliach. Dokáže tak uložiť štruktúru celej predpočítanej dátovej kocky. Uloženie dát je však neefektívne, ak je dátová kocka riedka.
- Hybridný OLAP (HOLAP) kombinuje predchádzajúce prístupy.

Kapitola 4

Dáta z reklamných kampaní

Cieľom tejto práce je vybudovanie reálneho dátového skladu a následne analýza podnikových dát spoločnosti ROI Hunter, a.s. [1]. Spoločnosť ROI Hunter, a.s. sa zameriava na inzerciu v oblasti internetového obchodu, tzv. e-commerce, s cieľom optimalizácie reklamných kampaní.

Dáta, ktorým sa budeme v práci venovať, sa týkajú nastavení a štatistík reklamných kampaní prevádzkovaných spoločnosťou ROI Hunter na platforme Facebook [2]. Aplikácia ROI Hunter pracuje s viacerými zdrojmi štatistických dát, kde dôležitými zdrojmi sú metriky Facebook Marketing [3] a Google Analytics [4].

4.1 Reklamné dáta a ich charakteristiky

Jednotlivé reklamy (angl. advertisement, skr. ad) sú spojované do skupín (angl. adset), ktoré sú charakteristické hlavne nastavením cielenia na špecifické skupiny užívateľov a tiež nastavením rozpočtu, ktorý môže byť využitý pre inzerciu. Zoznamy takýchto skupín sú ďalej spojované do reklamných kampaní (angl. campaign), ktoré udávajú marketingový cieľ inzerenta. Túto hierarchiu budeme chápať ako vrstvy reklamy, kde každá vrstva má svoje špecifické vlastnosti a nižšie vrstvy môžu tieto vlastnosti v niektorých prípadoch prepisovať.

4.2 Nastavenia reklám a reklamných kampaní

Nastavenia reklám pozostávajú z niekoľkých kategorických atribútov, numerických atribútov a iných špecifických atribútov ako napríklad názov reklamy, internetové odkazy na obrázky alebo videá, rôzne identifikátory, dátum a čas vytvorenia reklamy a podobne. Pre dolovanie znalostí sa obmedzíme len na kategorické a numerické atribúty, ktoré majú potenciálny vplyv na úspešnosť reklamy. Výpis tých najzaujímavejších atribútov, zvolených pre dolovanie znalostí metódami strojového učenia, poskytuje tabuľka č. 4.1

[1] https://www.roihunter.com/
[2] https://www.facebook.com/
[3] https://developers.facebook.com/docs/marketing-api/insights/fields/v2.8
[4] https://developers.google.com/analytics/devguides/reporting/core/dimsmets

Atribút	Popis
Reklama	
Typ	Typ reklamy z hľadiska umiestnenia na Facebooku, prípadne na Instagrame.
Typ kreatívy	Typ reklamnej kreatívy. Jednoduchá reklama, reklama na stránku, sponzorovaná reklama a pod.
Typ príspevku	Typ príspevku na Facebooku. Odkaz, udalosť, fotka, video, a pod.
Typ akčného tlačítka	Akčné tlačítko príspevku na Facebooku ako kúpiť produkt, dozvedieť sa viac, inštalovať aplikáciu, registrovať sa, kontaktovať predajcu a pod.
Skupina reklám	
Optimalizačný cieľ	Optimalizácia zobrazovania reklamy za špecifickým účelom ako je počet zhliadnutí, počet kliknutí, počet návštev obchodu, rozširovanie povedomia o značke, inštalácia konverzií, nákupy v internetovom obchode a pod.
Typ rozpočtu	Spôsob čerpania stanoveného rozpočtu. Rozpočet je stanovený na deň alebo na celé obdobie inzercie.
Výška rozpočtu	Výška rozpočtu určeného na deň alebo celé obdobie.
Automatické nastavovanie ponuky (bid)	Automatické nastavovanie určovania ponuky v súťaži s ostatnými reklamami o zobrazenie reklamy.
Výška ponuky (bid)	Horná hranica ponuky reklamy v súťaži o zobrazenie.
Vek cieľovej skupiny	Horné a spodné ohraničenie veku cieľovej skupiny.
Vzdelanie cieľovej skupiny	Dokončená úroveň vzdelania. Špecifikácia rozsahu rokov dokončenia vysokej školy.
Pohlavie	Cielenie reklamy podľa pohlavia.
Vzťahy cieľovej skupiny	Aktuálny stav vzťahu (slobodný, vo vzťahu,..). Tiež špecifikácia pohlavia, o ktoré sa človek zaujíma.
Kampaň	
Účel kampane	Hlavný účel celej reklamnej kampane. Zvýšenie počtu konverzií, zhliadnutí príspevkov, preklikov do internetového obchodu, rozširovanie povedomia o značke a pod.
Horná hranica útraty	Finančný limit reklamnej kampane.

Tabuľka 4.1: Vybrané kategorické a numerické atribúty reklám použité pre dolovanie znalostí. Kategorické atribúty sú zvýraznené modrastým pozadím.

Skratka	Metrika	Popis
Clicks	Počet kliknutí (kliky)	Celkový počet kliknutí na reklamu.
Impressions	Zobrazenia	Celkový počet zobrazení reklamy užívateľom.
Spend	Investícia do reklamy	Výška útraty za reklamu.
Frequency	Frekvencia zobrazení	Priemerný počet zhliadnutí reklamy jedným užívateľom.
Total actions	Počet akcií s reklamou	Celkový počet akcií, ktoré vykonal užívateľ v súvislosti s reklamou.
Total actions value	Hodnota akcií	Celková hodnota akcií, ktoré vykonal užívateľ v súvislosti s reklamou.
Reach	Dosah	Počet ľudí, ktorým bola reklama ponúknutá.
Social	Cielenie podľa Facebook priateľstva	Počet klikov, zobrazení a ľudí, ktorí prišli do kontaktu s reklamou na základe ohodnotenia reklamy ich priateľmi.
Engagement	Počet interakcií	Celkový počet všetkých dostupných interakcií, ktoré boli s reklamou vykonané.
Link clicks	Kliknutia na odkaz	Celkový počet kliknutí na odkaz v reklame vedúci do internetového obchodu.
Revenue	Zisk	Čistý zisk z predajov na základe reklamy.
Sessions	Počet návštev	Počet návštev internetového obchodu na základe reklamy.
Transactions	Počet transakcií	Počet vykonaných transakcií v internetovom obchode na základe reklamy.

Tabuľka 4.2: Vybrané základné štatistické metriky reklám použité pre dolovanie znalostí.

4.3 Štatistické údaje reklamy

Štatistické údaje sú vždy asociované k reklame, ktorá ich vygenerovala. Štatistiky skupiny reklám, resp. reklamnej kampane potom predstavujú agregáciu týchto štatistík pre danú vrstvu. Existuje mnoho štatistických metrík užívaných v marketingu. Pre dolovanie znalostí sa zameriame na základné metriky vyjadrené v tabuľke 4.2, ale pre úplnosť uvádzame aj často užívané odvodené štatistiky v tabuľke 4.3.

Pri štatistických metrikách sa často odvolávame na ***akcie***, resp. ***transakcie***. Akcie sú rôzneho typu, pričom sa delia do troch skupín podľa prostredia, kde k ním došlo. Pri každej akcii sledujeme počet a unikátny počet jej vykonaní. Unikátny počet je taký, kde nerátame viacnásobne vykonané rovnaké akcie tým istým užívateľom. Pri niektorých akciách naviac sledujeme hodnotu danej akcie.

Skratka	Vzťah	Conversion
ROI	zisk / útrata	Return on Investment. Návratnosť investície.
COS	útrata / zisk	Cost of sales. Náklady na predaj.
CPA	útrata / akcie	Cost per action. Cena za akciu.
CPT	útrata / transakcie	Cost per transaction. Cena za transakciu.
CPC	útrata / kliky	Cost per click. Cena za klik.
CTR	kliky / zobrazenia	Cost through rate. Pomer počtu klikov k počtu zobrazení.
Conversion rate	akcie / kliky	Conversion rate. Pomer vykonaných akcií ku klikom.
Per click value	zisk / kliky	Per click value. Zisková hodnota kliku.

Tabuľka 4.3: Odvodené štatistické metriky často používané v internetovej inzercii a marketingu.

Akcie na Facebooku

Akcie vykonané na Facebooku sú založené na type príspevku a interakcii s ním.

Like	Páči sa mi to
Unlike	Zrušenie "Páči sa mi to"
Comment	Pridanie komentára k príspevku
Link click	Kliky na odkaz v reklame
Photo view	Zhliadnutie reklamného obrázku
Video view	Zhliadnutie reklamného videa
App install	Inštalácia mobilnej aplikácie

Akcie v internetovom obchode

Akcie vykonané v internetovom obchode sa označujú aj ako konverzné akcie.

Registration	Registrácia účtu v internetovom obchode
Key page view	Návšteva kľúčovej internetovej stránky obchodu
Add to wishlist	Nastavenie záujmu o momentálne nedostupný tovar
Add to cart	Vloženie tovaru do košíku
Checkout	Pristúpenie k platbe
Purchase	Zaplatenie za tovar

Akcie v mobilnej aplikácii

Akcie v mobilnej aplikácii sú podobné ako pri akciách v internetovom obchode. Množina mobilných akcií je však rozšírená o niekoľko ďalších možností podľa typu aplikácie.

Activate	Aktivácia aplikácie v mobilnom telefóne
Tutorial	Dokončenie tutoriálu v aplikácii
Achievement	Odomknutie špecifického obsahu
Credits	Útrata kreditov v aplikácii

4.3.1 Atribučné modely

Život a štatistiky reklamy je možné sledovať viacerými spôsobmi. Atribučné modely predstavujú jeden z možných spôsobov sledovania života reklamy, ktorý sa týka prideľovania užívateľských akcií (štatistických hodnôt) reklamám aj niekoľko dní spätne.

Facebook podporuje atribučné modely, ktoré sa týkajú započítania užívateľskej akcie 1, 7 alebo 28 dní po zhliadnutí reklamy alebo po kliku na reklamu. Ak teda užívateľ klikol na reklamu pred tromi dňami, ale až dnes vykonal nákup v obchode, ktorý túto reklamu inzeroval, tak danej reklame bude pripočítaný nákup v atribučnom modele 7 a 28 dní, ale už nie v atribučnom modele 1 deň, pretože už prešli tri dni.

Pomocou atribučných modelov je možné sledovať vplyv reklamy na rýchlosť, s akou užívateľ vykoná akciu, napríklad nákup.

Kapitola 5

Návrh a implementácia

Táto kapitola sa venuje návrhu dátového skladu a dolovaniu znalostí prostredníctvom klasifikácie. Po návrhu pristupujeme priamo k implementácii, pričom ako hlavný programovací jazyk pre implementáciu bol zvolený jazyk Python[1]. Dátový sklad je vybudovaný nad relačnou databázou PostgreSQL[2].

5.1 Dátový sklad

Návrh dátového skladu sa pridŕža dvojvrstvovej architektúry z kapitoly 3.2.

5.1.1 Aktualizácia dát v dátovom sklade

Analýzy môžu byť v dlhodobom aj krátkodobom časovom horizonte. V prípade dátového skladu sa uchováva história niekoľko rokov dozadu, a preto poskytuje vhodné prostredie pre dlhodobé analýzy. Pri procese aktualizácie dát musíme brať na vedomie dve okolnosti. V akom časovom horizonte potrebujeme analyzovať a ako veľmi môžeme zaťažovať operačné databázy. Dátový sklad navrhujeme s procesom nahrávania nových dát do dátového skladu každých 24 hodín, a to v noci. Každý deň je teda možné pracovať s dátami od predošlého dňa smerom do minulosti, čo pokrýva dlhodobejšie analýzy.

Špecifickou vlastnosťou reklamných dát je, že štatistiky môžu byť generované aj spätne niekoľko dní dozadu. V tomto prípade je to až 28 dní dozadu, preto každý deň je pri nahrávaní nových dát nutné vykonať aj aktualizáciu štatistík daný počet dní spätne. Proces aktualizácie dát v tomto prípade vedome odporuje požiadavke na nemennosť dát uložených v sklade podľa definície 3.

5.1.2 Integrácia databáz

Jednou z veľkých predností dátového skladu je spájanie dát z rôznych zdrojov. Dáta reklamných kampaní, o ktorých táto práca pojednáva, sú logicky aj fyzicky oddelené. Všetky nastavenia a špecifikácie reklám sú uložené v relačnej databáze a vygenerované štatistiky sú uložené v dokumentovej databáze MongoDB[3].

Už samotným fyzickým oddelením súvisiacich dát do rôznych typov databáz vzniká veľká potreba pre možnosť kladenia otázok nad obidvomi databázami zároveň. Dátový sklad

[1] https://www.python.org/
[2] https://www.postgresql.org/
[3] https://www.mongodb.com/

bude preto primárne integrovať tieto dve databázy do novej, relačnej databázy pre analytické účely s potenciálom pridávania nových dátových zdrojov v budúcnosti.

Všetky štatistiky pre reklamy sú numerického charakteru a obsahujú identifikátor reklamy, reklamnej skupiny, reklamnej kampane a klienta. Dôležitou informáciou je takisto dátum, ku ktorému sú tieto štatistiky platné. Špecifikácie reklám sa skladajú z atribútov na rôznych úrovniach reklamy (reklama, skupina, kampaň) a sú nielen numerické, ale aj nominálne a textové. Integráciu je možné riešiť jednoducho, a to mapovaním identifikátorov reklamy a štatistík.

5.1.3 Integrácia tabuliek

Špecifikácie reklám sa vždy vzťahujú k niektorej úrovni reklamy, ale v operačnej databáze sú uložené vo viacerých tabuľkách. Pre analýzu však tieto tabuľky neprinášajú žiadne výhody, a preto dátový sklad budujeme s cieľom maximalizovať jednoduchosť štruktúry databázy a zjednodušiť tak vytváranie SQL dotazov.

Integráciu tabuliek relačnej databáze riešime operáciou JOIN, ktorej výsledkom je spojenie dvoch a viacerých tabuliek. K integrácii pristupujeme podľa vzťahu tabuliek:

- **Vzťah 1:1** Tabuľky môžeme bezproblémovo zlúčiť do jednej tabuľky. Dôležité je definovať konvenciu pomenovania jednotlivých stĺpcov tak, aby bolo kedykoľvek znovu možné rekonštruovať pôvodnú schému.
- **Vzťah 1:n** Tabuľka A môže mať priradených viac záznamov z tabuľky B.
  - Pokiaľ číslo n označujúce počet priradených záznamov z tabuľky B nezvykne byť veľké, tak môže byť výhodou použiť taký spôsob zlúčenia, kedy v tabuľke A vytvoríme nové stĺpce typu pole, v ktorom budú uložené záznamy tabuľky B ako zoznam hodnôt. Pokiaľ sa však zaujímame o vzťahy medzi hodnotami tabuľky B, tak tento spôsob nie je vhodný, pretože môžeme stratiť informáciu o tom, ktoré hodnoty tabuľky B boli pôvodne uložené ako jeden záznam.
  - Vytvoríme novú tabuľku C, ktorá bude obsahovať všetky záznamy z tabuliek A aj B, pričom záznamy tabuľky A budú uložené redundantne. Získavame jednu výslednú tabuľku za cenu straty normalizovaných tabuliek a zvýšenia nákladov na priestor.
- **Vzťah m:n** Tabuľka A môže mať priradených viac záznamov z tabuľky B a naopak. Využiť môžeme podobné spôsoby ako pri vzťahu 1:n. Za zváženie v tomto prípade stojí aj agregácia súvisiacich dát v oboch tabuľkách a vytvorenie agregovaného nového záznamu v tabuľke C, pokiaľ je agregácia možná a takýto záznam by bol užitočný.

5.1.4 Architektúra dátového skladu

Dátový sklad modelujeme schémou snehovej vločky. Tabuľka faktov je, v prípade reklamných dát, tabuľka reklamných štatistík, preto práve tabuľka štatistík bude stredom schémy snehovej vločky. Jednotlivé fakty zodpovedajú štatistikám opísaným v kapitole 4.

Dimenzie tabuľky faktov sú nastavenia jednotlivých vrstiev reklamy: reklama, skupina reklám, kampaň. Tieto vrstvy definujú hierarchiu nastavení, ale každá vrstva má špecifické nastavenia a vlastnosti, preto každú vrstvu chápeme ako samostatnú dimenziu, napriek vzťahom medzi nimi. Dané dimenzie modelujeme tabuľkami dimenzií.

Ďalšou dimenziou je klient, modelovaný tabuľkou. Tabuľka klienta je sčasti normalizovaná a náleží k nej tabuľka užívateľov, v ktorej sú definované role užívateľ, manažér a dozorca.

Špecifickou dimenziou je čas, ktorý však ponecháme ako súčasť tabuľky faktov. Nevzniká teda nová tabuľka dimenzie. Cieľom je v tomto prípade zjednodušiť dotazovanie na tabuľku faktov, nakoľko filtrovanie štatistík podľa času je veľmi častou operáciou.

5.1.5 Operácie nad dátovým skladom

Nad dátovým skladom realizovaným relačnou databázou môžeme vykonávať štandardné OLAP operácie ich transformáciou na príslušné SQL dotazy. K analytickým dotazom sme sa rozhodli využiť užívateľsky prívetivý open-source nástroj **Metabase**[4]. Tento nástroj ponúka webové rozhranie pre prácu s dátovými kockami, pričom je schopný vizualizácie 2-D podkociek. Užívateľské dotazy, ktoré je ľahko možné vytvoriť jednoduchým klikaním myšou, sú transformované na SQL dotazy.

Nástroj Metabase je napojený priamo na databázu a vytvorené SQL dotazy priamo sprostredkuje danej databáze. Výsledok je potom možné vizualizovať tabuľkou alebo grafmi, v prípade geografických údajov dokonca bodmi na mape. Výhodou je tiež možnosť vytvorenia tzv. *dashboardov*, násteniek, na ktoré je možné pripnúť výsledky z viacerých uskutočnených dotazov, pričom výsledky sa pravidelne aktualizujú v špecifikovaných časových intervaloch. Je tak možné vytvoriť celkový firemný prehľad na jednej webovej stránke. Takýto prehľad, ktorý je navyše možné parametrizovať, je tiež možné vložiť ako panel do inej stránky, napríklad priamo do hlavnej aplikácie spoločnosti.

Materializované pohľady

Pretože pracujeme s obrovskými dátami, a často nás zaujímajú vyššie úrovne agregácie, potrebujeme často realizovať SQL dotazy, ktoré obsahujú požiadavku k agregácii. V relačných databázach je možné takéto časté agregácie riešiť tzv. *materializovanými pohľadmi*. Takýto pohľad je fyzicky uložený a štrukturovaný výsledok daného SQL dotazu.

Materializovaný pohľad sa užívateľovi javí ako ďalšia tabuľka, nad ktorou môže vykonávať analýzy, avšak dáta sú oproti pôvodným tabuľkám predpripravené, tj. agregované na potrebnej úrovni agregácie. Pri analýzach, ktoré potrebujú pracovať len s danou úrovňou agregácie, je potom výhodnejšie pristupovať k týmto pohľadom než k pôvodným tabuľkám. Efektívnejšie je to predovšetkým v časovej náročnosti SQL dotazu. Tento prístup pochopiteľne zvyšuje nároky na priestor, pretože údaje sú uložené duplicitne k hlavným, pôvodným tabuľkám. Materializované pohľady preto vytvárame len v prípade, kde vidíme časté použitie danej agregácie.

5.2 Klasifikácia a predikcia

Pripravený dátový sklad v ďalšom texte využijeme ako vstupný bod pre dolovanie znalostí metódami strojového učenia. Na základe štúdia spôsobov dolovania znalostí v kapitole 2.3 sme sa zamerali na klasifikáciu a predikciu. Najväčší dôraz pritom kladieme na klasifikáciu prostredníctvom rozhodovacieho stromu, a to hlavne kvôli jeho dobrej interpretovateľnosti. Pre porovnanie úspešnosti iných klasifikátorov nad dostupnými dátami sme však využili aj bayesovskú klasifikáciu a klasifikátor SVM.

[4] http://www.metabase.com/

5.2.1 Návrh procesu klasifikácie

Napriek tomu, že dáta dostupné v pripravenom dátovom sklade už boli niektorými metódami upravované, tak pre účely klasifikácie potrebujeme niektoré špecifické metódy predspracovania. Proces klasifikácie a predikcie rozdelíme na niekoľko logických krokov tak, aby sme v každom kroku mohli uložiť medzivýsledok využiteľný v nasledujúcom kroku.

Stanovenie klasifikačných tried

Klasifikačné triedy, teda cieľové skupiny predikcie, stanovíme ako dve možnosti na základe metriky návratnosti investícií, teda metriky ROI špecifikovanej v 4.3. Cieľové skupiny budú označené ako **dobrá** alebo **zlá** návratnosť investícií, pričom koeficient ROI menší ako 1 znamená, že reklama je v strate a naopak koeficient ROI vyšší ako 1 znamená, že reklama generovala zisk. Zameriame sa teda na binárnu klasifikáciu, pretože klasifikujeme vzorky len do dvoch tried.

Selekcia vzoriek

Klasifikátor netrénujeme nad celým dátovým skladom ale snažíme sa zamerať na novšie dáta len za posledných niekoľko mesiacov. Čím viac trénovacích vzoriek dodáme do klasifikátoru, tým sú šance na dosiahnutie zobecneného modelu lepšie. Aby bol však klasifikačný model k niečomu užitočný, tak by mal na vstupe dostať ideálne vyrovnaný počet vzoriek zo všetkých cieľových skupín. Preto je pre správne trénovanie modelu selekcia vhodných vzoriek dôležitá.

V tomto kroku je možné vykonať aj selekciu klientov podľa ich primárneho cieľu inzercie. Pre niektorých klientov nie je metrika ROI hlavným ukazateľom úspešnosti, a preto budúca predikcia ich reklám s cieľom predikovať ROI nie je užitočná. V tejto práci sa obmedzíme len na klientov, ktorí sa primárne zaujímajú o návratnosť svojich investícií.

Selekcia a úprava atribútov

Ďalším krokom je selekcia vhodných atribútov pre klasifikáciu a ich predspracovanie do podoby, ktorú očakáva klasifikátor. Atribúty vstupných vzoriek zvolíme podľa sekcie 4, pričom kľúčové atribúty sú atribúty nastavenia reklamy. Zo štatistických hodnôt vyberáme len základné metriky, ktoré nie sú vo vzťahu s inými štatistickými metrikami.

Pri štatistikách vždy špecifikujeme dátum a vyberáme buď štatistiky z konkrétneho dňa alebo súčet štatistických údajov za zvolené obdobie. S dátumom štatistík pracujeme vždy relatívne k dátumu vzniku reklamy. Môžeme preto pracovať napr. so štatistikami z prvých štyroch dní života reklamy s určovaním reklám do klasifikačných tried podľa štatistík generovaných v piaty deň od vzniku reklamy. To znamená, že na takomto modele môžeme predikovať reklamy, ktoré boli vytvorené pred štyrmi dňami a zaujíma nás predikcia na nasledujúci deň. Podľa zvoleného počtu dní štatistík môžeme tvoriť rôzne klasifikačné modely zamerané na inú dĺžku života reklamy.

Trénovanie klasifikátora

Tretím krokom je trénovanie vybraného klasifikátora nad pripravenými dátami a uloženie natrénovaného modelu do binárneho súboru na disk. Pretože vytvorenie modelu vždy súvisí s nastavovaním špecifických parametrov zvoleného klasifikátora, tak je vhodné zabezpečiť trénovanie klasifikátora s použitím rôznych parametrov.

Pre overenie správnosti a obecnosti klasifikátora by sa presnosť nemala overovať na vzorkách, ktoré boli zároveň použité pre trénovanie. Vhodným testom presnosti klasifikátoru je rozdeliť pripravené vzorky na tzv. trénovacie a testovacie. Model potom trénujeme použitím vzoriek určených pre trénovanie a test úspešnosti a presnosť modelu overujeme na vzorkách určených pre testovanie, ktoré model dosiaľ nevidel. Tento spôsob je možné ďalej vylepšiť **krížovou validáciou**, kedy rozdelíme pripravené vzorky dát na n skupín a klasifikátor so zvolenými parametrami vytvoríme celkom n krát, ale pre trénovanie použijeme zakaždým iné skupiny a taktiež použijeme iné skupiny pre testovanie daného modelu.

Krok trénovania klasifikátora zahrňuje okrem samotnej tvorby modelu aj validáciu modelu a export vlastností modelu, jeho štruktúru a presnosť.

Predikcia

Posledným krokom je predikcia nových hodnôt natrénovaným klasifikačným modelom. V tomto kroku predpokladáme dostupný a aktuálny klasifikačný model, ktorý môžeme využiť pre predikciu. Po tom, čo užívateľ špecifikuje identifikátor reklamy, resp. skupiny reklám, tak systém pre predikciu pripraví vstupné hodnoty reklamy rovnako ako boli pripravené hodnoty pre trénovanie v kroku 5.2.1 s výnimkou prípravy klasifikačnej triedy, pretože práve túto triedu nepoznáme a chceme ju predikovať.

Pretože predikcia je interpretovaná klientovi, tak je veľmi žiadané podporiť dôveryhodnosť danej predikcie jej zdôvodnením. Okrem predikcie sa tiež zameriame na automatické odvodenie odporučenia pre klienta. Takéto odporučenie je akcia pre zmenu vlastností reklamy s dopadom na zmenu predikovanej hodnoty do pozitívnej triedy. Ak teda predikujeme vývoj návratnosti investície reklamy a predikovaná trieda bude **zlá** návratnosť investícií, tak ponúkneme nápad pre zmenu nastavení reklamy tak, aby bola predikovaná trieda **dobrá** návratnosť investícií.

5.2.2 Implementácia procesu klasifikácie

Proces klasifikácie reklamných dát je implementovaný v programovacom jazyku Python. Kľúčovými knižnicami projektu sú pritom knižnica `SQLAlchemy`[5] pre prácu s databázou a knižnica `scikit-learn`[6], ktorá poskytuje rozhranie pre prácu s klasifikačnými modelmi napísanými v jazyku C a užitočné funkcie pre manipuláciu s týmito modelmi.

Proces je rozdelený do niekoľkých krokov, podobne ako je to špecifikované v návrhu. Časovo veľmi náročnou úlohou je príprava dát, ktoré je možné predať na vstup klasifikátora. Tento krok si vyžaduje niekoľko náročných SQL dotazov a transformáciu dát, preto predovšetkým výsledok tohto kroku vždy uložíme na disk do súboru, aby bolo jednoduché a rýchle trénovať viac klasifikátorov. Súbor s pripravenými dátami je na disk uložený binárne. V aplikácii so súborom pripravených dát operujeme prostredníctvom knižnice `pandas`[7], ktorá poskytuje výborné rozhranie pre prácu s virtuálnou tabuľkou dát, a to konkrétne operácie pre selekciu riadkov a stĺpcov, filtrovanie záznamov a rôzne agregačné funkcie.

Kategorické atribúty

Klasifikačné modely v knižnici scikit-learn natívne nepodporujú kategorické atribúty, a preto kategorické nastavenia reklám musia byť najskôr upravené do vhodnej podoby. Pre trans-

[5] https://www.sqlalchemy.org/
[6] http://scikit-learn.org
[7] http://pandas.pydata.org/

parentnú prácu s atribútmi je implementovaný objekt **CategoricalFeature**, ktorý kategorický atribút transformuje do podoby **1 z *n*** podľa 2.2.4, takže tento atribút je ďalej prezentovaný ako pole binárnych čísel, kde len jedno číslo je jednička vyjadrujúca hodnotu kategórie jej indexom v poli a ostatné čísla sú nulové. Objekt CategoricalFeature je následne zodpovedný aj za spätnú transformáciu hodnôt.

Numerické atribúty

Pre prácu s numerickými atribútmi sú implementované objekty **NumericalFeature** a **NumericalFeatureRelative**, ktoré pracujú s numerickými atribútmi a sú tiež schopné normalizácie týchto atribútov. Objekt NumericalFeatureRelative naviac pracuje s dátumom, takže je možné vybrať štatistiky reklamy z konkrétneho obdobia relatívne k dátumu vzniku reklamy. Štatistiky vo vybranom období sú agregované.

Trénovanie modelu a úspešnosť klasifikácie

Po vybraní konkrétneho klasifikátora a pripravení vstupných hodnôt pristupujeme k samotnému trénovaniu. Každý klasifikátor požaduje nastavenie špecifických parametrov, ktoré ovplyvňujú úspešnosť klasifikácie, ale väčšinou nie sú dopredu známe. Pre tento prípad je výhodné využiť funkciu `GridSearchCV` z knižnice `scikit-learn`, do ktorej je možné predať zoznam rôznych parametrov klasifikátora, a táto funkcia následne vytvorí klasifikačné modely nad všetkými kombináciami parametrov a vyberie ten najlepší na základe vyhodnotenia úspešnosti pomocou krížovej validácie (angl. cross validation - CV).

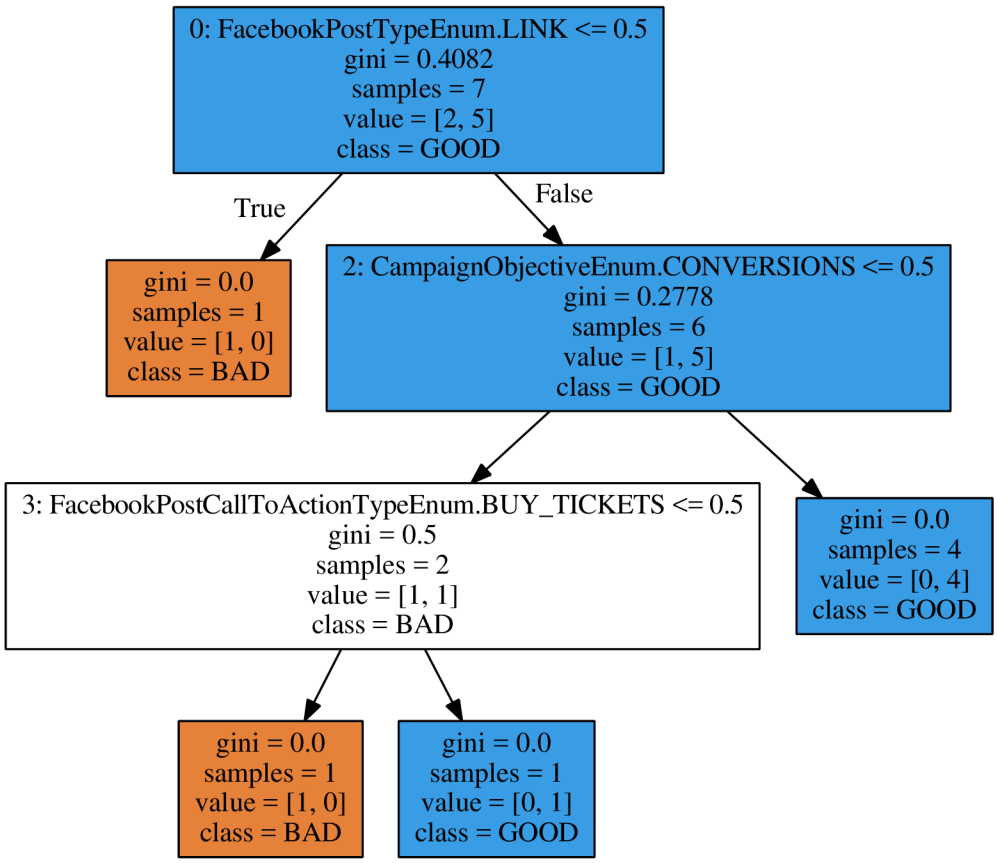
Úspešnosť klasifikátora vyjadríme metrikami zo sekcie 2.3.3. Konkrétne teda metrikami presnosti, úplnosti, správnosti, f-metriky a grafom ROC. Pre klasifikáciu použijeme tri rôzne klasifikátory, a to rozhodovací strom, naivný bayesovský klasifikátor a klasifikátor SVM s cieľom porovnať navzájom ich úspešnosť. Metriky väčšinou vzťahujeme k pozitívnej klasifikačnej triede. V našom prípade, pri triedach dobrá a zlá návratnosť investícií, nás však zaujímajú rovnako obe triedy. Pokiaľ je návratnosť investícií dobrá, tak je to pozitívne, ak je ale zlá, tak dostávame priestor pre zlepšenie, viď. sekcia 5.2.4.

Predikcia

Klasifikačný model je vždy natrénovaný pre nastavenia reklamy a rôzny počet štatistických dní. K predikcii pristupujeme tak, že zvolíme odpovedajúci klasifikačný model podľa aktuálnej dĺžky života danej reklamy. Ak reklama existuje n dní, tak je ideálne použiť klasifikačný model, ktorý bol natrénovaný s použitím $n+1$ dní štatistík od vzniku trénovacích reklám. Pre vyššiu variabilitu predikcie preto uvažujeme o väčšom počte klasifikačných modelov podľa počtu štatistík na vstupe. Pre predikciu vývoja práve vytvorenej reklamy preto môžeme použiť len nastavenia reklamy a žiadne štatistiky, pretože reklama ešte žiadne nevygenerovala.

5.2.3 Zdôvodnenie rozhodnutia o predikcii

Pre vyššiu vierohodnosť predikcie je ideálne užívateľovi poskytnúť nejaké informácie o tom, prečo predikcia dopadla práve takto a nechať následne aj na jeho uvážení, či sa rozhodne podniknúť nejaké kroky na základe výsledku predikcie. Predikciu vykonávame pomocou natrénovaného klasifikátora, pričom z dôvodu možnosti zdôvodnenia predikcie sa zameriame



Obr. 5.1: Príklad grafického zobrazenia vnútornej štruktúry rozhodovacieho stromu. Všetky uzly v strome testujú kategorické atribúty, ktoré sú reprezentované v kódovaní **1 z n**, tj. prítomnosť kategórie označuje jednička, neprítomnosť nula. Uzol s podmienkou `FacebookPostTypeEnum.LINK <= 0.5` sa vetví vpravo, pokiaľ má vstupný atribút `FacebookPostTypeEnum` hodnotu `LINK`, inak sa vetví vľavo. *Pozn. Hodnoty sú vymyslené.*

na rozhodovací strom. V tom prípade zdôvodnenie predikcie priamo súvisí s cestou od koreňového uzlu k listovému uzlu v strome. Každý uzol stromu obsahuje testovaciu podmienku a práve na základe týchto podmienok je možné predikovať klasifikačnú triedu novej reklamy a zároveň aj slovne vyjadriť rozhodnutie klasifikátora.

Príklad Zdôvodnenie klasifikácie rozhodovacím stromom so štruktúrou vyjadrenou na obr. 5.1 je výpis podmienok na ceste rozhodovacím stromom k predikovanej klasifikačnej triede.

```
0 GOOD  AD_ID: 1
Node 0: value 1.0 >= 0.5 (FacebookPostTypeEnum.LINK).
Node 2: value 1.0 >= 0.5 (CampaignObjectiveEnum.CONVERSIONS).
Node 6: leaf.
```

*Predikovaná návratnosť investícií pre reklamu s číslom 1 je **dobrá**, pretože typ príspevku na Facebooku je LINK a cieľom kampane sú konverzie. Správnosť predikcie podporujú 4 reklamy zo 4 reklám s rovnakým nastavením (100% podpora).*

```
1 BAD  AD_ID: 2
Node 0: value 1.0 >= 0.5 (FacebookPostTypeEnum.LINK).
Node 2: value 0.0 < 0.5 (CampaignObjectiveEnum.CONVERSIONS).
Node 3: value 0.0 < 0.5 (FacebookPostCallToActionTypeEnum.BUY_TICKETS).
Node 4: leaf.
```

*Predikovaná návratnosť investícií pre reklamu s číslom 2 je **zlá**, pretože typ príspevku na Facebooku je LINK, cieľom kampane nie sú konverzie a akčným tlačítkom príspevku na Facebooku nie je BUY_TICKETS. Správnosť predikcie podporuje 1 reklama z 1 reklamy s rovnakým nastavením (100% podpora).*

5.2.4 Odporučenie pre úpravu dát s cieľom zmeniť výsledok predikcie

V prípade rozhodovacieho stromu sú klasifikačné triedy reprezentované listami uzlu. To znamená, že v strome existujú cesty vedúce k jednej aj k druhej klasifikačnej triede (v prípade binárnej klasifikácie). Ak je pre novú reklamu predikovaná trieda negatívna, tak má zmysel uvažovať o takej zmene parametrov reklamy, ktoré by viedli k zmene predikcie a tým pádom k pravdepodobne lepšiemu vývoju reklamy.

Alternatívne cesty rozhodovacím stromom

Uvažujme o predikcii, ktorá skončila v liste stromu predstavujúcom negatívnu triedu. Hľadajme alternatívnu cestu s minimálnou zmenou v parametroch reklamy, ktorá končí v liste stromu predstavujúcom pozitívnu triedu. Možným riešením je rekurzívne hľadať najbližšiu vhodnú cestu, ale pretože topológia stromu je známa dopredu, tak poznáme všetky listy reprezentujúce pozitívnu triedu.

Ak poznáme všetky pozitívne cesty, tak môžeme uvažovať o dvoch spôsoboch hľadania najbližšej pozitívnej cesty k aktuálnej negatívnej ceste stromov, tj. cesta, ktorá vedie k listu predstavujúcemu negatívnu triedu.

1. Hľadáme najdlhší spoločný prefix uzlov pozitívnych ciest s negatívnou cestou. Vybraná pozitívna cesta potom zdieľa niekoľko uzlov s negatívnou cestou, ale v istom uzle pokračuje alternatívnou vetvou. Počet uzlov na tejto alternatívnej vetve predstavuje počet zmien, ktoré je potrebné vykonať.

2. Prechádzame uzly každej pozitívnej cesty a testujeme podmienku uzlu voči reklame na vstupe. Následne vyberieme takú pozitívnu cestu, kde došlo k najmenšiemu počtu nesplnených podmienok. Tieto nesplnené podmienky potom reprezentujú zmeny, ktoré je potrebné vykonať pri nastaveniach reklamy tak, aby predikcia danej reklamy nasledovala vybranú cestu.

Nevýhoda pri prechádzaní stromovou štruktúrou a hľadaní alternatívnej cesty spočíva v možných uzlových podmienkach. Ak totiž uzol testuje štatistický atribút, tak užívateľ tento atribút nie je schopný zmeniť. Odporučenie má preto zmysel len vtedy, ak môže k zmene predikovanej triedy dôjsť na základe zmeny niektorého nastavenia reklamy.

Kapitola 6

Testovanie, porovnanie a výsledky

Testovanie a vyhodnotenie klasifikácie je realizované na vybranej množine dát obsahujúcich nastavenia a štatistiky reklám. Klasifikujeme do dvoch klasifikačných tried GOOD a BAD, ktoré sú založené na návratnosti investície reklamy, tj. ROI. ROI reklamy, podľa ktorého stanovujeme klasifikačnú triedu pri trénovaní klasifikátora, sledujeme vo vybraný deň. Tento spôsob nám poskytuje možnosť predikovať novým reklamám metriku ROI pre daný deň v budúcnosti. Pretože úspešnosť reklamy vyniká skôr v dlhodobejšom horizonte než jeden deň, tak dochádza k faktu, že aj dobrá reklama môže mať vo vybraný deň slabý výkon. Pri určovaní ROI po jednotlivých dňoch preto vzniká nepomer, kedy väčšia časť reklám vykazuje neúspech.

Druhou možnosťou klasifikácie je použiť pre stanovenie klasifikačnej triedy agregované ROI za nejaký interval, napr. nasledujúci týždeň namiesto jedného dňa. Pri predikcii vývoja nových reklám potom uvažujeme nie o tom, aký výkon reklama prinesie v nasledujúcom dni, ale o tom, aký výkon reklama prinesie v nasledujúcom období. Klasifikáciu môžeme teda realizovať tak, že vyberieme rôzny počet štatistických dní, 0 až n, pre trénovanie modelu a rôzny počet štatistických dní pre určovanie ROI, teda 1 deň alebo interval. Štatistické dni sa vzťahujú k dátumu vzniku danej reklamy.

Dataset použitý pre nasledujúce testovanie obsahuje 12 793 reklám, 20 unikátnych atribútov nastavení reklám a 27 unikátnych štatistických metrík, ktoré sú pripravené ku každému zo 14-ich dní od vzniku reklamy. Reálny počet atribútov nastavení reklám je však kvôli kódovaniu kategorických atribútov metódou **1 z *n*** (kapitola 2.2.4) 135. Ak trénujeme model s použitím atribútov nastavení reklám a siedmych dní štatistických atribútov, tak je celkový počet atribútov na vstupe klasifikátora rovný 324 (135 + 27 * 7). Táto množina reklám bola vytvorená ako reprezentujúca vzorka dát pre klientov, u ktorých je hlavným cieľom reklamy dosahovanie návratnosti investícií, tj. ROI.

6.1 Testovanie úspešnosti klasifikátorov

Úspešnosť klasifikátorov hodnotíme predovšetkým metrikami z kapitoly 2.3.3 založených na matici zámien. Napriek tomu, že sa práca venuje predovšetkým klasifikácií algoritmom rozhodovacieho stromu, tak pre porovnanie uvádzame úspešnosť klasifikácie danej množiny dát aj inými typmi klasifikátorov.

Úspešnosť overíme pre dva prípady. Na vstup klasifikátoru dodáme dáta s nastaveniami reklám a siedmymi dňami štatistických údajov a klasifikujeme na základe úspešnosti reklám

v nasledujúcich siedmych dňoch; pracovné označnenie **prípad 7+7**. Druhý prípad uvažuje na vstupe klasifikátora len nastavenia reklám a klasifikuje na základe úspešnosti reklám v nasledujúcich 14-ich dňoch; pracovné označenie **prípad 0+14**. V maticiach zámien sa prvá hodnota viaže vždy k prvému prípadu, druhá hodnota k druhému prípadu. Reklám zaradených do klasifikačnej triedy GOOD je v prvom prípade 12% a v druhom prípade 7% z celkového počtu reklám v pripravenej množine.

Rozhodovací strom

Optimálne parametre rozhodovacieho stromu boli empiricky nájdené s využitím postupu opísanom v kapitole 5.2.2. Ako kritérium pre voľbu atribútu určeného k vetveniu stromu sme použili Gini index. Výška stromu je obmedzená do maximálnej úrovne 8.

Matica zámien		**Predikcia**	
		BAD (ROI <1)	GOOD (ROI ≥ 1)
Skutočnosť (True)	BAD	11811, 11032	229, 443
	GOOD	288, 1059	465, 259

Tabuľka 6.1: Matica zámien modelu rozhodovacieho stromu vytvorená po krížovej validácii s tromi preloženiami trénovacích a testovacích vzoriek. Hodnoty oddelené čiarkou sa viažu k prvému, resp. k druhému testovaciemu prípadu.

Metriky úspešnosti vychádzajúce z matice zámien pri klasifikácii rozhodovacím stromom vyzerajú nasledovne.

Metrika		prípad 7+7	prípad 0+14
Správnosť (Accuracy)		0.95959	0.88259
Presnosť (Precision)	pre triedu GOOD	0.66955	0.36635
	pre triedu BAD	0.97620	0.91269
Úplnosť (Recall)	pre triedu GOOD	0.61753	0.19653
	pre triedu BAD	0.98098	0.96139
F-metrika (F-measure)	pre triedu GOOD	0.64198	0.23934
	pre triedu BAD	0.97858	0.93620

Tabuľka 6.2: Metriky úspešnosti klasifikácie pomocou rozhodovacieho stromu.

Výsledky naznačujú, že štatistické údaje na vstupe do klasifikátora zohrávajú výraznú rolu pri správnosti klasifikácie a predikcie.

Bayesovská klasifikácia

Tabuľka 6.3 matice zámien zachytáva úspešnosť klasifikácie naivnou bayesovskou klasifikáciou. Metriky úspešnosti vychádzajúce z matice zámien pri Bayesovskej klasifikácií sú v tabuľke 6.4. Z výsledkov vidíme, že bayesovská klasifikácia má tendenciu lepšie odhaliť vlastnosti reklám patriacich do kategórie GOOD, avšak celková správnosť modelu je oproti rozhodovaciemu stromu nižšia.

Matica zámien		Predikcia	
		BAD (ROI <1)	GOOD (ROI ≥ 1)
Skutočnosť (True)	BAD	11385, 8492	655, 2983
	GOOD	413, 35	340, 1283

Tabuľka 6.3: Matica zámien modelu Bayesovského klasifikátora vytvorená po krížovej validácii s tromi preloženiami trénovacích a testovacích vzoriek. Hodnoty oddelené čiarkou sa viažu k prvému, resp. k druhému testovaciemu prípadu.

Metrika		prípad 7+7	prípad 0+14
Správnosť (Accuracy)		0.91651	0.76408
Presnosť (Precision)	pre triedu GOOD	0.38575	0.34197
	pre triedu BAD	0.96510	0.99572
Úplnosť (Recall)	pre triedu GOOD	0.45153	0.97344
	pre triedu BAD	0.94559	0.74004
F-metrika (F-measure)	pre triedu GOOD	0.39445	0.49276
	pre triedu BAD	0.95502	0.84378

Tabuľka 6.4: Metriky úspešnosti Bayesovskej klasifikácie.

SVM klasifikácia

Klasifikácia metódou SVM je časovo omnoho náročnejšia než klasifikácia rozhodovacím stromom alebo bayesovskou klasifikáciou. Pre čiastočné zníženie tejto náročnosti bol zvolený limit počtu iterácií algoritmu na 1000 iterácií.

Matica zámien		Predikcia	
		BAD (ROI <1)	GOOD (ROI ≥ 1)
Skutočnosť (True)	BAD	12035, 10841	5, 634
	GOOD	742, 1089	11, 229

Tabuľka 6.5: Matica zámien modelu SVM klasifikátora vytvorená po krížovej validácii s tromi preloženiami trénovacích a testovacích vzoriek. Hodnoty oddelené čiarkou sa viažu k prvému, resp. k druhému testovaciemu prípadu.

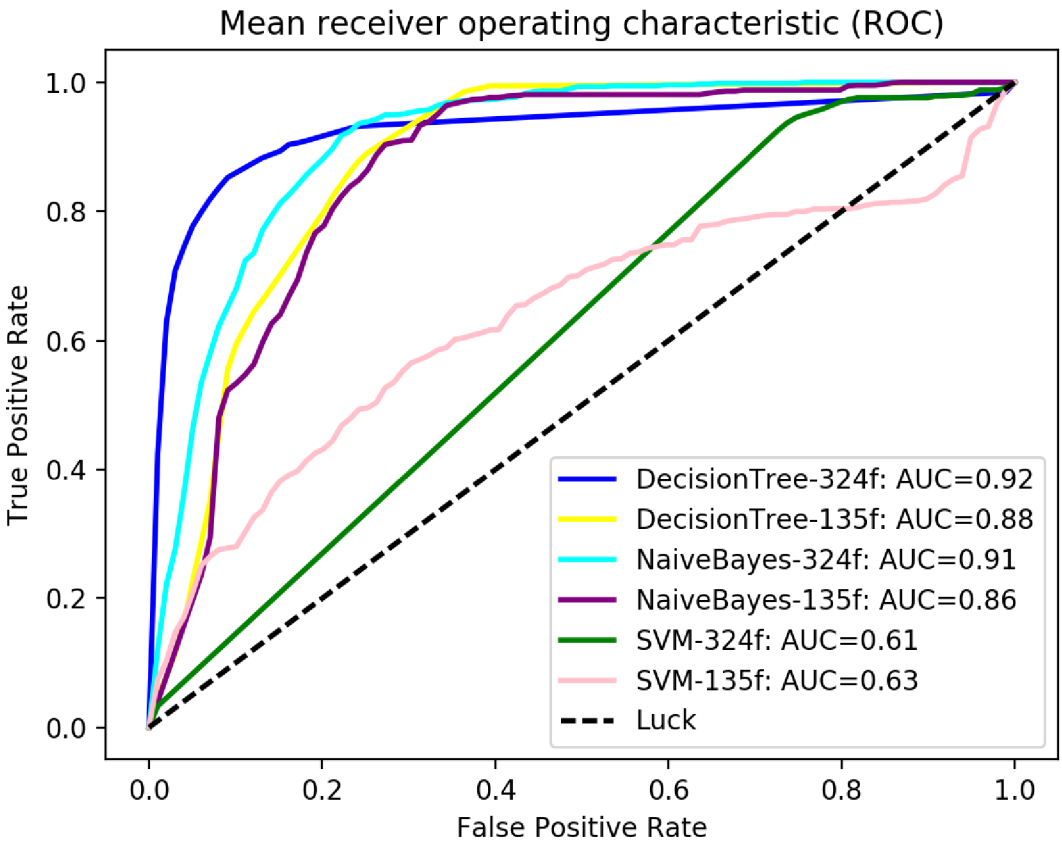
Metrika		prípad 7+7	prípad 0+14
Správnosť (Accuracy)		0.94161	0.86531
Presnosť (Precision)	pre triedu GOOD	0.63636	0.35616
	pre triedu BAD	0.94193	0.90884
Úplnosť (Recall)	pre triedu GOOD	0.01461	0.17379
	pre triedu BAD	0.99958	0.94475
F-metrika (F-measure)	pre triedu GOOD	0.02809	0.20611
	pre triedu BAD	0.96990	0.92576

Tabuľka 6.6: Metriky úspešnosti klasifikácie pomocou SVM klasifikátora.

Metriky úspešnosti vychádzajúce z matice zámien pri klasifikácií klasifikátorom SVM sú v tabuľke 6.6. Výsledky naznačujú dobrú presnosť SVM klasifikátora, pričom pri odstránení limitu na počet iterácií by presnosť pravdepodobne ešte stúpla.

6.1.1 Porovnanie pomocou ROC grafov

ROC graf na obrázku 6.1 zachytáva vzťah medzi falošne pozitívnymi a falošne negatívnymi pozorovaniami klasifikátorov. Výsledky klasifikátorov zodpovedajú prvému, resp. druhému testovaciemu prípadu, teda s použitím 135 atribútov alebo 324 atribútov pri trénovaní. Vyhodnotenie klasifikátorov je realizované na základe krížovej validácie s tromi preloženiami a graf zobrazuje priemernú hodnotu úspešnosti po tejto validácií. Skratka AUC vyjadruje plochu pod krivkou (angl. area under curve) a čím je číslo väčšie, tým je správnosť modelu vyššia. Rozhodovací strom v tomto prípade vychádza ako najlepší klasifikačný model pre danú vzorku dát, ako v prípade 7+7, tak aj v prípade 0+14.



Obr. 6.1: Graf ROC zachytáva priemernú hodnotu falošne pozitívnych a falošne negatívnych pozorovaní klasifikátora vo vzťahu ku klasifikačnej triede GOOD.

6.2 Významné atribúty

Rozhodovací strom pracuje s matematickým vyjadrením informácie, ktorú nesú dané atribúty a pomocou tejto informácie vyberá atribúty pre vetvenie. Preto môžeme predpokladať, že niektoré atribúty sú dôležitejšie a je preto významné zvoliť pri nich správnu hodnotu. Z modelu rozhodovacieho stromu sme schopný vybrať najdôležitejšie atribúty a získať tak nové deskriptívne znalosti.

Zistili sme, že dôležitým atribútom s klasifikačnou silou 57%, je maximálny vek ľudí, na ktorých cielime, `Adset.targeting_age_max`. Po následnom preskúmaní tohto atribútu vzhľadom ku klasifikačným triedam sme zistili, že až 78% reklám, ktoré sú zaradené do triedy GOOD, má maximálny vek nastavený na 65 rokov.

Dôležitými atribútmi sú tiež výška rozpočtu s klasifikačnou silou 12% a výška ponuky do súťaže (bid) s klasifikačnou silou 11%. Pri klasifikácii s použitím 7 dní štatistík má počet sedení (sessions) zo 7. dňa klasifikačnú silu 78%.

Keďže klasifikujeme na základe návratnosti investícií, tak celkom očakávane je najlepšou hodnotou nastavenia optimalizačného cieľa skupiny reklám OFFSITE_CONVERSIONS, teda optimalizácia konverzií na stránkach klienta. Podobne pri špecifikovaní účelu kampane je vhodné zvoliť hodnotu PRODUCT_CATALOG_SALES, teda optimalizácia predaja produktov z katalógu klienta s klasifikačnou silou 40%.

6.3 Porovnanie metód pre odporučenie úprav atribútov

V kapitole 5.2.4 sme navrhli dve metódy pre odporučenie vykonania úpravy atribútov novej vstupnej vzorky tak, aby bola predikovaná klasifikačná trieda pozitívna, a teda aby sa zvýšila reálna pravdepodobnosť, že reklama bude v budúcnosti naozaj úspešná. Vhodnou metrikou pre porovnanie týchto metód je počet úprav, ktoré po užívateľovi požadujú, aby vykonal. Ideálny stav je preto požadovať čo najmenej zmien, ktoré postačia na zmenu predikovanej hodnoty.

Prvú metódu (1) nazveme *najdlhší spoločný prefix*, druhú metódu (2) nazveme *test všetkých uzlov*. Metódy porovnáme s využitím klasifikátora natrénovaného nad testovacou množinou dát a priemerný počet požiadaviek na úpravu atribútov vypočítame z 5000 náhodne zvolených reklám, u ktorých predikcia skončila v triede BAD. Z tabuľky 6.7 môžeme vidieť, že pri druhej metóde je priemerný počet požiadaviek na zmenu výrazne nižší, a preto túto metódu hodnotíme ako lepšiu.

Metóda	Priemerný počet požiadaviek na úpravu
1. metóda - najdlhší spoločný prefix	3.6
2. metóda - test všetkých uzlov	1.8

Tabuľka 6.7: Porovnanie priemerného počtu požiadaviek na úpravu hodnôt atribútov metódami pre tvorbu odporučení na základe ciest v rozhodovacom strome vedúcich k listom, ktoré klasifikujú do pozitívnej klasifikačnej triedy.

Kapitola 7

Záver

Získavanie znalostí z databáz je komplexný problém zahrňujúci integráciu, prípravu dát, dolovanie znalostí metódami strojového učenia a vizualizáciu výsledkov. Špecifickým krokom pri analýzach rozsiahlych podnikových dát je vytvorenie dátového skladu, ktorý poskytuje kvalitné dáta pre analýzu. Práca pojednáva o problematike budovania dátových skladov a v kapitole 5.1 prináša návrh a implementáciu konkrétneho dátového skladu pre spoločnosť ROI Hunter, a.s.

V oblasti prípravy dát pre dolovanie sme uviedli niekoľko štandardných postupov, medzi ktoré patria postupy pre vysporiadanie sa s chýbajúcimi hodnotami a odľahlými hodnotami, postupy pre odstránenie šumu v dátach, postupy normalizácie a diskretizácie numerických atribútov.

Práca sa ďalej venovala analýze a klasifikácii reklamných dát, opísaných v kapitole 4, dostupných z vytvoreného dátového skladu. Pri klasifikácii sme sa zamerali predovšetkým na klasifikáciu rozhodovacím stromom a vybudovali systém pre dynamickú analýzu nových dát, predikciu ich správania a odporučenie ideálnej ďalšej akcie inzerentovi tak, aby sme maximalizovali konkrétne marketingové ciele v rámci príslušnej reklamnej kampane. Predikciu ďalšieho vývoja úspešnosti reklamy zdôvodňujeme cestou v stromovej štruktúre rozhodovacieho stromu a je vyjadrená ako výpis podmienok. Odporučenie ďalšej akcie vyjadrujeme ako návrh na zmenu niekoľkých atribútov reklamy tak, aby sa predikovaná hodnota tejto reklamy zmenila z negatívnej do pozitívnej klasifikačnej triedy. Návrh a implementácia procesu klasifikácie, zdôvodnenie predikcie a tvorby odporúčaní je opísaný v kapitole 5.2.

V kapitole 6 sme s použitím reprezentatívnej vzorky reklám predviedli vzájomné porovnanie klasifikačných algoritmov rozhodovacieho stromu, naivnej bayesovskej klasifikácie a klasifikácie metódou SVM. Rozhodovací strom pritom vychádza ako najlepší klasifikátor na tejto množine dát, čo je veľmi pozitívne, pretože práca sa venuje hlavne algoritmu rozhodovacieho stromu, pretože klasifikačný proces implementovaný v tejto práci tvorí zdôvodnenie predikcie a odporučenie ďalšej akcie práve na základe vnútornej stromovej štruktúry rozhodovacieho stromu.

Diplomová práca priniesla implementáciu dátového skladu, systém pre klasifikáciu a predikciu vývoja reklám podľa návratnosti investícií a vytvorila tak základ pre ďalší rozvoj dolovania z dát spoločnosti ROI Hunter, a.s. V tejto práci sme sa obmedzili len na klasifikáciu a predikciu vývoja reklám pre klientov, u ktorých je cieľom návratnosť investícií (ROI). Niektorí klienti však majú iné hlavné ciele inzercie, tzv. Key performance indicator (KPI), a preto by bolo vhodné vytvoriť klasifikačné modely pre každú skupinu klientov združenú podľa ich hlavného cieľa.

Reklama väčšinou náleží k nejakému konkrétnemu produktu, pričom produkty vždy náležia do nejakej kategórie. Je pravdepodobné, že reklamy produktov rôznych kategórií dosahujú úspešnosť rôznymi marketingovými stratégiami. Špecifický klasifikačný model podľa kategórie produktu by mohol priniesť vyššiu presnosť predikcie.

Pri dátach s časovou známkou, teda konkrétne pri štatistikách, ktoré sú generované každý deň, môžeme objaviť sezónnosť alebo periodicitu v správaní reklám, resp. zákazníkov. Je napríklad známe, že ľudia pred vianočnými sviatkami nakupujú viac, než počas všedných dní. Sledovať sezónne zmeny a prispôsobiť im klasifikačný model je takisto potenciálne užitočné.

Ďalším námetom na pokračovanie práce je sledovanie klientových reakcií na dodané odporúčania podľa predikcie z natrénovaného klasifikačného modelu. Ak sa klient rozhodne dané odporučenie využiť, tak sa potvrdzuje správnosť modelu. Naopak, ak klient nevyužíva poskytnuté odporúčania, tak to môže indikovať chybný model. Preto je vhodné sledovať spätnú väzbu od klienta a uvažovať nad jej využitím pre vylepšenie aktuálneho modelu.

Literatúra

[1] Duda, R. O.; Hart, P. E.; Stork, D. G.: *Pattern Classification (2nd Ed)*. Wiley, 2001.

[2] Fawcett, T.: An Introduction to ROC Analysis. *Pattern Recogn. Lett.*, ročník 27, č. 8, Červen 2006: s. 861–874, ISSN 0167-8655, doi:10.1016/j.patrec.2005.10.010.
URL http://dx.doi.org/10.1016/j.patrec.2005.10.010

[3] Fayyad, U.; Piatetsky-shapiro, G.; Smyth, P.: From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, ročník 17, 1996: s. 37–54.

[4] Fayyad, U. M.; Smyth, P.: *Advances in knowledge discovery and data mining*. California: MIT Press, vyd. 1. vydání, 1996, ISBN 0262560976.

[5] Golfarelli, M.; Rizzi, S.: *Data warehouse design*. New York: McGraw-Hill, 2009, ISBN 9780071610391.

[6] Han, J.: *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005, ISBN 1558609016.

[7] Inmon, W.: *Building the Data Warehouse*. John Wiley & Sons, 2005, ISBN 0471081302.
URL http://fit.hcmute.edu.vn/Resources/Docs/SubDomain/fit/ThayTuan/DataWH/Bulding%20the%20Data%20Warehouse%204%20Edition.pdf

[8] Kelly, S.: *Data Warehousing in Action*. Wiley, 1997, ISBN 9780471966401.

[9] Korb, K. B.; Nicholson, A. E.: *Bayesian artificial intelligence*, ročník 1. CRC press, 2004.

[10] Lechtenbörger, J.: *Data Warehouse Schema Design*. Dissertationen zu Datenbanken und Informationssystemen, Aka, 2001, ISBN 9781586032142.
URL https://books.google.sk/books?id=BBOM25evBHkC

[11] Makhoul, J.; Kubala, F.; Schwartz, R.; aj.: Performance Measures For Information Extraction. In *In Proceedings of DARPA Broadcast News Workshop*, 1999, s. 249–252.
URL http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=D523AC9D0B4CE5EB1294D8271F8A14FA?doi=10.1.1.27.4637&rep=rep1&type=pdf

Prílohy

Príloha A

Obsah CD

Priložené CD a rovnako aj elektronicky odovzdaný komprimovaný adresár obsahujú kompletnú elektronickú verziu diplomovej práce. Tá zahrňuje implementovaný program realizácie dátového skladu a procesu klasifikácie a predikcie a ďalej zdrojové súbory technickej správy. Konkrétne:

- `README.txt` textový súbor obsahujúci popis obsahu CD
- `doc.pdf` technická správa o diplomovej práci
- `doc/` adresár so zdrojovými súbormi technickej správy (systém LaTeX)
- `app/` adresár so zdrojovými súbormi aplikácie vytvorenej v rámci diplomovej práce

Príloha B

Inštalácia aplikácie

Aplikácia, teda programové vyhotovenie diplomovej práce, nesie názov `Hildy` čo je akronym anglickej vety *Hidden layers discovery*, tzn. objavovanie skrytých vrstiev. Program je vyhotovený v jazyku Python a používaný v *nix prostredí (Ubuntu/Debian). Pre spustenie aplikácie je preto vhodné použiť práve operačný systém založený na linuxe. Potrebné systémové balíčky sú `python3`, `python3-pip`, `python3-psycopg2`, `python3-tk`, `python3-numpy`, `python3-scipy`, `python3-pydot`, `cron` a `libpq-dev`.

B.0.1 Docker

Ďalšou možnosťou je vytvoriť aplikačný kontajner nástrojom Docker[1], v ktorom budú všetky systémové závislosti už pripravené. Tvorba kontajneru z adresára so zdrojovými súbormi aplikácie a spustenie tohto kontajnera sa vykoná nasledovne:

```
$ docker build --rm -t hildy-image .
$ docker run --name hildy -it -p 3000:3000 hildy-image
```

Po spustení príkazu `run` bude kontajner vyžadovať užívateľské heslo do databázy - heslo užívateľa `postgres` je `postgres`. Na upozornenie `You are going to drop & create all tables in localhost:5432/hildy database. Press 'c' to continue` reagujte stlačením klávesy `c`. Tým budú inicializované potrebné tabuľky v databáze, ktoré budú následne aj naplnené priloženou vzorkou dát, viď. B.0.3.

Pri spustení kontajneru je na pozadí spustený aj nástroj Metabase, takže na porte 3000 bude prístupné webové rozhranie nástroja Metabase, v ktorom je možné pracovať s databázou. Tento nástroj však pri prvom spustení treba inicializovať, tj. nastaviť užívateľský účet a pripojenie na databázu vytvorenú v kontajneri (adresa `127.0.0.1:5342`, databáza `hildy`, užívateľ `postgres`, heslo `postgres`).

Vytvorený kontajner je spustený v interaktívnom režime, takže priamo po spustení je k dispozícii terminál, v ktorom môžeme pracovať s aplikáciou. Ešte predtým ale treba aktivovať virtuálne prostredie pre Python príkazom `source .venv/bin/activate`, viď. nasledujúca kapitola.

Použitím Docker kontajnera prakticky odpadajú nasledujúce kroky vytvorenia prostredia pre Python, inštalácia Python balíčkov a príprava aplikačnej databázy, nakoľko tieto kroky sú zahrnuté vo vytvorenom kontajneri, ktorý je vhodný predovšetkým na vývoj a demonštráciu (preto v sebe obsahuje aj databázu).

[1] https://www.docker.com/

B.0.2 Python prostredie

Aplikácia Hildy využíva niekoľko voľne dostupných knižníc. Dobrým pravidlom je vytvorenie virtuálneho prostredia pre Python pre každý samostatný projekt (Pri použití Docker kontajneru nie je potrebné virtuálne prostredie vytvárať nakoľko je už pripravené, je však potrebné ho aktivovať príkazom `source .venv/bin/activate`). Vytvorenie takéhoto prostredia s následnou inštaláciou potrebných knižníc sa vykonáva nasledovne:

```
$ virtualenv -p /usr/bin/python3 .venv
$ source .venv/bin/activate
$ pip install -U pip wheel
$ pip install --use-wheel -r requirements.txt
```

B.0.3 Databáza a aplikačné nastavenia

Aplikácia Hildy pracuje s databázou PostgreSQL, ktorú je preto potrebné nainštalovať a inicializovať. Iniciálizácia v tomto prípade znamená vytvorenie databázy a jej schémy, pričom pre vytvorenie databázovej schémy je možné využiť príkaz `python -m hildy create-schema` viď. C. Adresu vytvorenej databáze je potrebné nastaviť v lokálnych nastaveniach aplikácie `hildy/config/settings_local.py`. Tento súbor je na začiatku potrebné vytvoriť a každé nastavenie, ktoré v tomto súbore špecifikujeme prepisuje východzie nastavenia. Tie sa nachádzajú v súbore `hildy/config/settings_default.py`.

Priložená vzorka dát Súčasťou CD je aj vzorka anonymizovaných dát o reklamách rôznych klientov, ktoré boli použité pre testovacie účely. Tieto dáta sú v textovom formáte vytvorené príkazom `pg_dump`. Tieto dáta je možné importovať do novej databáze nasledujúcim príkazom:

```
$ cat anonymized-dump/*.dump | psql -U postgres -h 127.0.0.1 -p 5432 hildy
```

B.0.4 Metabase

Pre analytické dotazy do pripravenej databázy je možné využiť systém Metabase. Pre inštaláciu tohto systému postupujte podľa oficiálneho návodu na http://www.metabase.com/start/. V prípade použitia Docker kontajneru je nástroj Metabase už nainštalovaný, ale je potrebné ho nakonfigurovať, tzn. vytvoriť užívateľský účet a nastaviť pripojenie na aplikačnú databázu.

Príloha C

Manuál aplikácie

Pre prvý kontakt s aplikáciou je vhodné požiadať o nápovedu. Jej skrátená verzia vyzerá nasledovne:

```
$ python -m hildy --help
Hildy - Hidden layers discovery - Data analysis tool.

Usage:
    hildy sync-clients
    hildy sync [-f=<from_date>] [-t=<to_date>]

    hildy eval-client-layer <client-id> (--ads | --adsets | --campaigns)
    hildy eval-kpi-forever
    hildy update-client-kpi <client-id> <kpi-goal>

    hildy create-schema [--drop-existing]
    hildy create-views
    hildy refresh-views
    hildy drop-views

    hildy mining prepare
    hildy mining train [--with-cross-validation]
    hildy mining describe
    hildy mining predict [<ad_ids> ...]
    hildy mining testing

Options:
    -f, --from-date=<from_date>                    Start date.
    -t, --to-date=<to_date>                        End date.
    -v, --verbose                                  Print additional info.
```

C.0.1 Príkaz sync

Aplikácia obsahuje hneď niekoľko príkazov, ktoré sú rozdelené do sekcií. Príkazy s prvým slovom **sync** slúžia pre synchronizáciu dát z dvoch operačných databáz podniku do dátového skladu. Pri korektnom pripojení na všetky databázy je možné týmto príkazom synchronizo-

vať údaje o klientoch, užívateľoch, reklamách, skupinách reklám, reklamných kampaniach a štatistikách.

C.0.2 Príkaz eval

Príkazy s prvým slovom **eval** slúžia pre vyhodnotenie výkonnosti vzhľadom ku KPI (angl. key performance indicator), teda vzhľadom k hlavnému cieľu klienta. Tento príkaz vyhodnocuje, či reklamy dosahujú stanovený cieľ. Táto práca o tomto príkaze nepojednáva.

C.0.3 Príkaz create

Tieto príkazy sa týkajú dátového skladu a teda databáze, ktorú používame ako dátový sklad. Sú to `create-schema`, `create-views`, `refresh-views` a `drop-views`. Prvý príkaz je užitočný pri tvorbe novej databázy. Ostatné sa týkajú vytvorenia alebo aktualizácie materializovaných pohľadov v databáze, pomocou ktorých zrýchľujeme užívateľské otázky.

C.0.4 Príkaz mining

Príkazy s prvým slovom **mining** náležia procesu dolovania z dát. Tieto príkazy pracujú len s dátovým skladom, ktorý môžeme naplniť dátami buď synchronizáciou z operačných databáz príkazom **sync** (odsek C.0.1) alebo importovaním priloženej vzorky dát na CD (odsek B.0.3).

- Príkaz `prepare` pripravuje dáta z dátového skladu do formy vhodnej pre klasifikátor a výsledok uloží ako binárny súbor. Pri pripravovaní dát môže byť použitý ľubovoľný počet dní pre štatistiky k reklamám.
- Pripravený súbor na svojom vstupe následne požaduje príkaz `train`, ktorý vyvolá trénovanie klasifikátoru s danými dátami. V rámci súboru sa pri trénovaní môžu využiť rôzne počty štatistických dní, takže trénovanie môže prebiehať napríklad s použitím len siedmich štatistických dní, aj keď pripravený súbor obsahuje až štrnásť štatistických dní.
- Príkaz `predict` predpovedá budúci vývoj nových reklám, ktoré špecifikujeme identifikátormi. Ak na vstupe nezadáme žiadne identifikátory, tak je pre demonštráciu vybraných niekoľko náhodných reklám. Predikcia je zároveň zdôvodnená na základe použitej cesty v rozhodovacom strome. Ak je predikovaná trieda negatívna, tak je výsledok predikcie doplnený o odporúčania pre zmenu atribútov reklamy.
- Príkaz `describe` slúži pre výpis parametrov klasifikačného modelu, zobrazenie vnútornej štruktúry klasifikačného modelu (orientovaného grafu v prípade rozhodovacieho stromu), najdôležitejších atribútov, ktoré boli použité v rámci trénovanie k vzájomnému rozlíšeniu vstupných vzoriek a vyhodnotenia úspešnosti po trénovaní použitím krížovej validácie.
- Príkaz `testing` je spojenie príkazu `train` a `describe`, pričom trénuje a vyhodnocuje rôzne klasifikačné algoritmy (rozhodovací strom, naivný bayesovskú klasifikáciu a metódu SVM) v rozličných prípadoch (prípad 7+7 a prípad 0+14). Tento príkaz preto slúži k porovnaniu úspešnosti daných klasifikátorov v rôznych situáciách.