České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra softwarového inženýrství

Obor: Aplikace informatiky v přírodních vědách

Analýza příčin vzniku shrinku produktů společnosti na základě logistických dat

Root Cause Analysis of Shrinkage Based on Logistics Data

DIPLOMOVÁ PRÁCE

Vypracoval: Bc. Anna Gruberová
Vedoucí práce: Ing. Martin Plajner, Ph.D.
Rok: 2023

České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra softwarového inženýrství Akademický rok 2022/2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student:	**Bc. Anna Gruberová**
Studijní program:	**Aplikace informatiky v přírodních vědách**
Název práce:	**Analýza příčin vzniku shrinku produktů společnosti na základě logistických dat**
Název práce anglicky:	**Root Cause Analysis of Shrinkage Based on Logistics Data**

Pokyny pro vypracování:

1. Seznamte se s procesy v dodavatelském řetězci. Pozornost věnujte štíhlé logistice a plýtvání v logistických procesech.
2. Proveďte přípravu dat – definujte shrink, rozpoznejte a označte jej v obdržených datech.
3. Na základě předchozího kroku najděte a analyzujte příčiny, které vedou ke vzniku shrinku produktů dané společnosti. Zjištěné výsledky vhodně vizualizujte.
4. Navrhněte automatizaci tohoto postupu, kterou by bylo možné aplikovat i na logistická data dalších společností.

Doporučená literatura:

[1] JIRSÁK, P., MERVART M., VINŠ M. *Logistika pro ekonomy - vstupní logistika.* 1. vydání. Praha: Wolters Kluwer ČR, 2012. ISBN 978-80-7357-958-6.

[2] SUTHERLAND J., BENNET B. *The Seven Deadly Wastes of Logistics: Applying Toyota Production System Principles to Create Logistics Value.* Bethelem, PA: Lehigh University, 2007.

[3] HASTIE, T., TIBSHIRANI R., FRIEDMAN J. H. *The elements of statistical learning: data mining, inference, and prediction.* 2nd ed. New York: Springer, c2009. Springer series in statistics. ISBN 978-0-387-84857-0.

Jméno a pracoviště vedoucího práce:

Ing. Martin Plajner, Ph.D.
Oddělení matematické teorie rozhodování, Ústav teorie informace a automatizace AV ČR, v.v.i.

. . . . . .
vedoucí práce

Datum zadání diplomové práce: 12. 10. 2022

Termín odevzdání diplomové práce: 3. 5. 2023

Doba platnosti zadání je dva roky od data zadání.

. . . . . .
garant oboru

. . . . . .
vedoucí katedry

. . . . . .
děkan

V Praze dne 12. 10. 2022

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracovala samostatně a použila jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne ....

Bc. Anna Gruberová

Poděkování

Chtěla bych poděkovat za vedení mé bakalářské práce, za cenné rady a připomínky k tvorbě této práce a za čas strávený touto pomocí.

Bc. Anna Gruberová

Název práce:

Analýza příčin vzniku shrinku produktů společnosti na základě logistických dat

Autor:	Bc. Anna Gruberová
Studijní program:	Aplikace přírodních věd
Obor:	Aplikace informatiky v přírodních vědách
Druh práce:	Diplomová práce
Vedoucí práce:	Ing. Martin Plajner, Ph.D. Oddělení matematické teorie rozhodování, Ústav teorie informace a automatizace AV ČR, v.v.i.
Konzultant:	–

Abstrakt:

Klíčová slova: Datová analýza, Logistika

Title:

Root Cause Analysis of Shrinkage Based on Logistics Data

Author: Bc. Anna Gruberová

Abstract:

Key words: Data Analysis, Logistics

Obsah

Úvod

Tato diplomová práce se zabývá

První kapitola se věnuje

V následující kapitole

Ve třetí kapitole se nachází

Poslední kapitola

- Data - jak jsou data uložená v DB u zákazníka - provázané - SQL příkazy - výběr proměnných - target hodnoty - cost, množství - produktová Hierarchie - Vymazaní outlierů - outlier metody - businessově

- Co vysvetluje target - Miner - PCA

- Korelační analýza mezi produkty v rámci kategorie - korelace - Rozčlenění produktů

- Vizualizace dat - Jak funguje PBI - Seznam metri

Kapitola 1

Logistika

1.1 Definice Logistiky

Logistika zahrnuje všechny operace, které se týkají doručení zboží nebo služeb od výrobce k zákazníkovi, s výjimkou samotné výroby zboží nebo provádění služby. Výrobou je naopak rozuměno vše, co mění podobu materiálu. Během výroby se však logistika uplatňuje, například jako přesun materiálu nebo polotovarů mezi jednotlivými výrobními zařízeními. Operace lze rozdělit do tří hlavních toků: materiálový, informační a finanční tok. Materiálový obsahuje všechny pohyby týkající se fyzického materiálu, tedy jeho získávání, přesuny a skladování, a to jak mezi zákazníky, dodavateli či výrobními areály a sklady, tak i vnitřní pohyby mezi produkčními linkami nebo skladovými pozicemi. Informační tok popisuje procesy vznikající během materiálového toku, dále se do něj řadí analýzy již proběhlých toků a plánování a předpovědi budoucích toků. Poslední kategorie, finanční tok mapuje náklady způsobené předešlými dvěma zmíněnými toky.[1]

Pojem logistika je úzce propojen s pojmem Supply Chain Management (SCM)[1]. Zatímco logistika se zabývá toky zboží, služeb či lidí, Supply Chain Management zahrnuje operace logistiky, navíc ale sleduje vztahy mezi procesory, které koordinuje a optimalizuje za účelem naplnění určitých cílů. Tímto cílem bývá často snížení nákladů v rámci částí procesu nebo zvýšení konkurenceschopnosti podniku [4]. Supply Chain Management se tedy prolíná s pojmem logistika a bývají často zaměňovány. Důvodem může být i to, že se jedná o nový pojem, který byl poprvé použitý v roce 1982.[2]

1.2 Štíhlá logistika

Štíhlost neboli „lean“ je koncept neustálého vylepšování procesu vytváření produktu nebo služby pomocí odstranění jakéhokoli plýtvání. Plýtváním rozumíme jakoukoli

[1]Do češtiny lze Supply Chain Management přeložit jako řízení či správa dodavatelského řetězce. V českém prostředí se používá jak anglická tak česká podoba.

činnost, která v očích zákazníka nezvyšuje hodnotu produktu a tedy není ochotný za tuto činnost zaplatit ve formě vyšší prodejní ceny. Z této definice plýtvání je patrné, že pohled zákazníka hraje důležitou roli při vytváření hodnoty produktu ve štíhlých systémech.[19, **?**]

Svůj původ nachází štíhlá logistika na začátku 20. století, kdy henry Ford zavedl pohyblivou montážní linku při výrobě automobilu Ford modelu T. Tato linka měla za následek několikanásobné snížení výrobního času a odstartovala sériovou výrobu aut. Díky čemuž se snížila prodejní cena, a automobily tak byly dostupné nejen nejbohatší vrstvě společnosti. Po druhé světové válce navázala automobilová společnost Toyota Motor Company na Fordovu efektivní montážní linku a vytvořila systém nazvaný Toyota Production System (TPS), který je přímým předchůdcem štíhlé logistiky.[18]

1.2.1 Toyota Production System

Toyota Production System je založen na pěti základních principech. Nejdůležitějším krokem je odstranit plýtvání. Je třeba se soustředit na jednotlivé procesy a na vazby mezi nimi. Pomocí metody genchi genbutsu[2] se nasbírají data a informace o procesech přímo na místě, kde procesy probíhají, aby případné problémy a zdroje plýtvání mohly být přesně určeny. Po této analýze se aplikuje přístup řešení problémů zvaný kaizen[3], jehož cílem je kontinuální zlepšování procesů. Posledním z principů je dodržování vzájemného respektu mezi všemi oddělení společnosti, jak vedoucími pracovníky, tak zaměstnanci u výrobních linek. [18]

V TPS je plýtvání rozděleno do tří kategorií - Muda (plýtvání), Mura (nevyváženost) a Muri (přetěžování) [21]. V následující části jsou podrobněji popsány jednotlivé typy.

Muda

Japonské označení Muda v překladu znamená plýtvání, neužitečnost či marnost. Muda zahrnuje všechny činnosti, které nepřispívají ke zvyšování hodnoty produktu. Mudu lze rozdělit na dva podtypy – 1. typ zahrnuje aktivity, které jsou nezbytné pro koncového zákazníka, např. testování, zda je produkt nebo služba bezpečná. Druhý typ obsahuje ty procesy, které již zákazník nepotřebuje, či dokonce nechce, neboť mohou mít vliv na rychlost výroby produktu (výkonu služby) nebo přímo na jeho kvalitu.

Taiichi Ohno, manažer ve společnosti Toyota, identifikoval sedm typů plýtvání, někdy nazývané *seven deadly wastes*. Klasifikace a popis včetně příkladů je uveden níže [18]:

1. **Nadprodukce** – Pokud je vyrobeno více produktů, než je možné expedovat k zákazníkovi, nebo více materiálu, než kolik je požadováno k další výrobě či okamžité spotřebě.

[2]Genchi v překladu znamená skutečná lokace a genbutsu skutečná věc.
[3]Kaizen je japonský překlad slova zlepšení.

2. **Zpoždění/čekání** – Jakákoli prodleva mezi dvěma na sebe navazujícími procesy, např. čekání jedné montážní linky na meziprodukty z jiné linky vlivem rozdílných výrobních časů nebo vlivem nedostatečné výrobní kapacita jednoho ze strojů, dále sem patří také čekání zaměstnanců z důvodu kontroly odvedené práce, pomalého načítání počítačového programu nebo čekání na konkrétní instrukce k výkonu práce [19].

3. **Transport** – Zbytečný přesun produktů, materiálů nebo informací. Tento transport navíc může vést k poškození produktu. Příkladem tohoto typu plýtvání může být situace, kdy materiál, který je nejvíce potřebný pro výrobu produktů je umístěn v největší vzdálenosti, nebo pokud přístup k jedné položce ve skladu je blokovaný jinými položkami.

4. **Pohyb** – Zbytečný pohyb lidí, vzniklý špatným rozmístěním objektů v prostoru, např. nepřiměřeně dlouhotrvající chůze, natahování se pro předměty, vyhýbání se lidem či předmětům.

5. **Skladování** – Pokud je naskladněno více surovin, rozpracovaných výrobků a hotových produktů, než kolik je požadováno, např. předčasná dovážka položek do skladu, chyba v dodávce, naskladnění položek do zásoby tzv. pro jistotu nebo z důvodu množstevní slevy.

6. **Nadbytečné zpracování** – Při výrobě dochází k použití více energie nebo prostředků než nutné, nebo je vytvořen koncový produkt, který má vyšší hodnotu, než jaký je dohodnutý a požadovaný standard.

7. **Defekty** – Produkty či meziprodukty, které je nutné přepracovat nebo odstranit z výroby z důvodu vady.

Tyto podoby plýtvání aplikované v TPS byly inspirací pro identifikaci sedmi typů plýtvání v logistice [18, 5]:

1. **Nadprodukce** – V případě logistiky je nadprodukce chápána jako doručení produktů dříve nebo ve větším množství něž bylo požadováno.

2. **Zpoždění/čekání** – Jakákoli prodleva mezi dvěma na sebe navazujícími procesy, např. čekání na převoz meziproduktů mezi dvěma výrobními linkami, příjezd kamionu mimo časové okno, doba mezi příjezdem kamionu a jeho naložením nebo čas mezi přijetím objednávky a zahájením její realizace.

3. **Transport** – Zbytečný přesun produktů, materiálů nebo informací, např. materiál, který je nejvíce potřebný pro výrobu produktů je umístěn v největší vzdálenosti, nebo pokud přístup k jedné položce ve skladu je blokovaný jinými položkami.

4. **Pohyb** – Zbytečný pohyb lidí, např. vzniklý špatnou organizací předmětů ve skladu, kdy položky, ke kterým se nejčastěji přistupuje, jsou v méně přístupných pozicích skladu, nebo dokonce sklad není strukturovaný vůbec, nebo nutnost změnit trasu při převozu položek ve skladu kvůli nedostatečně širokým uličkám.

5. **Skladování** – Pokud je naskladněno více surovin, rozpracovaných výrobků a hotových produktů, než kolik je požadováno, např. předčasná dovážka položek do skladu, chyba v dodávce, naskladnění položek do zásoby tzv. pro jistotu.

6. **Prostor** – Neoptimální využití dostupného místa, např. nedostatečná výška regálů ve skladech, nevyužitá kapacita regálů, neoptimální naložení kamionu, přetížení dostupných kapacit.

7. **Defekty** – Činnosti, které způsobí nutnost opakovat určitý proces, znehodnocení produktu nebo zvýší náklady, např. špatné zavezení produktu, špatné nebo chybějící označení produktu, chyby v evidenci.

V devadesátých letech, kdy se metody TPS začaly aplikovat ve společnostech, byl mezi sedm typů plýtvání Muda začleněn osmý typ - Dovednosti. V tomto případě dochází k neefektivitě kvůli nevyužití lidského potenciálu a talentů jednotlivých zaměstnanců. K tomu může docházet například striktním rozdělením na manažery a zaměstnance, kde role zaměstnanců je poslouchat nařízení shora a vykonávat práci tak, jak byla navržena vedoucími pracovníky. Avšak právě zaměstnanci pracující přímo v terénu lépe identifikují případné problémy a snadněji naleznou řešení díky svým zkušenostem.[19]

Mura

Mura lze přeložit jako nestejnoměrnost, nevyrovnanost a nepravidelnost. Jedná se o plýtvání vznikající špatnou provázaností jednotlivých procesů a to jak interních, tak externích. Následkem nevyváženosti je pak vznik plýtvání Muda. [21, 5]

Plýtvání v podobě Mura se rozlišuje jak v procesech informačního, tak hmotného toku. V případě informačního toku je nejvýznamnějším zdrojem plýtvání situace, kdy je chybně predikována poptávka mezi jednotlivými články logistického řetězce. Ignorování vztahů mezi procesy může vést k chybovosti i v řádu desítek procent. Informace, jejichž opomíjeni způsobuje chybovost předpovídání poptávky, mohou být např. v jaké fázi životního cyklu se výrobek nachází, plánování promoakcí nebo výrobní a logistická omezeni dodavatelů. Další zdroj Mura v informačním toku je nedostatečná znalost stavu zásob mezi dodavatelem a odběratelem. Následkem čehož dochází k méně častým zavážkám avšak s větším objemem, což vede k vyšším pojistným zásobám ve skladech. Většinu zmíněných situací lze eliminovat aplikováním konceptu „Just-in-Time“ do jednotlivých procesů. Plýtvání také vzniká při administrativě, pokud nejsou vhodně standardizované dokumenty používané v logistickém řetězci. Příkladem může být špatná evidence pohybů ve skladu či tvorba objednávek. Nesjednocenost v administrativních procesech vede ke zpomalení navazujících činností nebo dokonce k chybám, které způsobí nemožnost dokončení celého procesu. Pak je nutné vybrané procesy provést znovu a napravit chyby.[5]

Plýtvání v hmotném toku je přímým důsledkem chyb vznikajících v informačním toku. Lze identifikovat i takové zdroje plýtvání, které nesouvisejí přímo s informačním tokem, a to například dodržování různých standardů přepravních prostředků na

straně dodavatele a odběratele. To má pak za následek nadbytečné překládání materiálu do podoby, kterou druhá strana používá a se kterou je schopna následně efektivněji manipulovat.[5]

Muri

Pojem Muri označuje přetěžování. Muri často vzniká při snaze zvýšit produktivitu a odstranit tak předešlé typy plýtvání, v konečném důsledku může ale vést k výrazně větší chybovosti i celkovému selhání. Přetíženi mohou být zaměstnanci, ale i stroje. V obou případech vytížení na více než 100 % se může projevit na snížení kvality výstupu. Lidé mohou být méně pozorní a může docházet k nehodám, které mohou v menší či větší míře negativně ovlivnit i větší část logistického řetězce. Stroje mohou produkovat zmetkové výrobky, nebo může dojít k jejich poškození až zničení.[5, 21]

Příklad plýtvání Muda, Mura a Mudi

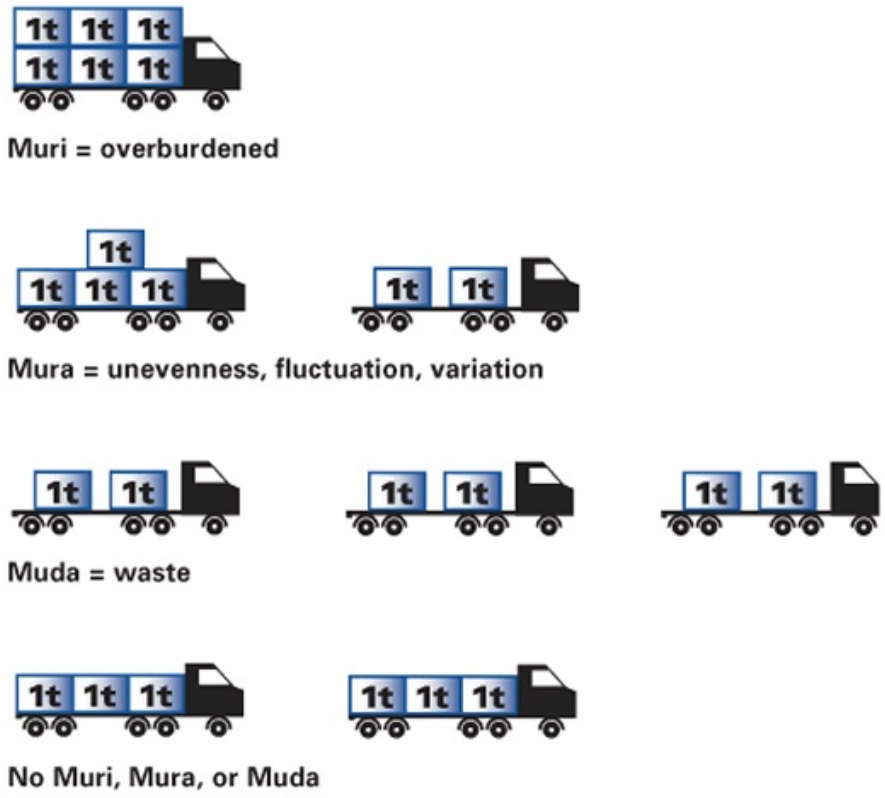
Všechny tři zmíněné typy plýtvání Muda, Mura a Muri jsou navzájem propojené. Tuto skutečnost je třeba brát v potaz při řešení zefektivňování procesů a eliminaci plýtvání. Pro představu je uvedena následující situace. Společnost potřebuje zákazníkovi přivézt šest tun materiálu, uloženého ve stejných jednotunových kontejnerech. Možné způsoby řešení této úlohy jsou znázorněné na obr. 1.1.[21]

Nejjednodušší možností je naložit na jeden kamion veškerý požadovaný materiál. V takovém případě společnost ušetří na počtu vozidel a eliminuje tak plýtvání přepravními prostředky, ušetří čas při nakládce a vykládce, protože není nutné obsluhovat více vozidel, zároveň . Na druhou stranu ale hrozí přetížení kamionu. Následkem přetížení se může zvýšit riziko nehody vozidla, firma může být pokutována nebo vozidlu nemusí být umožněn vjezd na určitá místa.

Opačným extrémem je použít tři kamiony, každý se dvěma tunami materiálu. Potom ale není efektivně využita dostupná kapacita a je patrné, že dochází k mnoha druhům plýtvání typu Muda.

Třetí možností je využití dvou kamionů, kdy první je naložen čtyřmi a druhý dvěma tunami. Toto rozložení nepodléhá žádným pravidlům a patrně proces nakládky není dostatečně spjatý s ostatními procesy nebo neprobíhá správný přenos informací o požadavcích mezi jednotlivými procesy. Nakládka a vykládka prvního velmi naloženého kamionu vyžaduje více času než druhého kamionu. Z toho plyne, že buď není možné v dostupném čase stihnout obsloužit první kamion a dochází k přetížení, anebo v případě druhého kamionu je zbude velké množství času a zaměstnanci zbytečně čekají. Z této volby plyne, že plýtvání typu Mura může způsobit Mudu i Muru.[21]

Optimální řešení je naložit dva kamiony po třech tunách, což je jejich ideální kapacita. V takovém případě společnost minimalizuje za daných podmínek všechny tři typy plýtvání. V reálném světě jsou situace mnohonásobně komplexnější a ne vždy existuje jednoznačné optimální řešení, které je navíc snadno dosažitelné. Důležité ale



Obrázek 1.1: Příklady plýtvání Muda, Mura a Muri při transportu šesti tun materiálu.[21]

je soustředit se na všechny tři typy současně, protože optimalizace pouze jednoho kritéria může způsobit jiný druh plýtvání nebo kolaps části systému.

V roce 2011 bylo realizováno dotazníkové šetření Vysokou školou ekonomickou v Praze, které mapovalo, kolik procent logistických expertů se zabývá odstraněním zmiňovaných tří typů plýtvání. Plýtvání Muda se snaží odstranit z logistických procesů 72 % respondentů, Murou se zabývá 39 % a plýtvání Muri řeší 30 % dotazovaných.[5]

1.2.2 Plýtvání v logistických procesech

Tato sekce se zabývá třinácti vybranými logistickými procesy z hlediska plýtvání, jak jsou uvedeny v knize *Logistika pro ekonomy – Vstupní logistika*. Analýza vychází z již zmíněného dotazníkového šetření z roku 2011.

Kapitola 2

Použité metody

2.1 Redukce dimenzionality

2.1.1 Analýza hlavních komponent

Analýza hlavních komponent (anglicky *Principal component analysis*, dále jako PCA) je statistická metoda využívaná pro pro extrakci proměnných, redukci vícedimenzionálních dat nebo vizualizaci dat. Lze ji aplikovat pouze na kvantitativní data s numerickými, spojitými hodnotami, neboť metoda využívá lineární algebraické techniky, jako je například kovarianční matice, pro jejíž výpočet se předpokládají spojité hodnoty.

Jednotlivá pozorování obsažená v datech bývají popsána několika různými příznaky. Tyto příznaky jsou často vzájemně korelované a obsahují šum. Metoda PCA dovede extrahovat pouze důležité informace z proměnných a snížit šum. K tomu je třeba vypočítat nové ortogonální proměnné, nazývané hlavní komponenty, které se získají jako lineární kombinace původních proměnných [7]. Hlavní komponenty reprezentují směry největšího rozptylu původních dat a jsou řazeny podle své významnosti. Jinými slovy, první hlavní komponenta zachycuje co nejvíce variability v datech, druhá hlavní komponenta zachycuje co nejvíce variability, která nebyla zachycena první hlavní komponentou, pro zbylé komponenty analogicky. [9]

Princip

Předpokládáme množinu dat $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$, kde N j počet pozorování a každý vektor $\boldsymbol{x}_i$ přísluší jednomu pozorování popsanému M proměnnými. $\mathbf{X}$ je potom matice rozměru $N \times M$ vstupních dat. Dále je definovaný výběrový průměr $\bar{\boldsymbol{x}}$ jako

$$\bar{\boldsymbol{x}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i, \tag{2.1}$$

a výběrová kovarianční matice $\mathbf{C}$

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}). \tag{2.2}$$

První hlavní komponentu, která popisuje největší rozptyl dat označíme y_{1i} a vypočteme následovně jako lineární kombinaci původních proměnných

$$y_{1i} = \boldsymbol{a}_1^\top(\boldsymbol{x}_i - \bar{\boldsymbol{x}}), \quad \text{pro } i = 1, \ldots, N, \tag{2.3}$$

kde $\boldsymbol{a}_1 = (a_{11}, \ldots, a_{M1})^\top$ je vektor vah.

Optimální vektor $\boldsymbol{a}_1$ je takový vektor, který maximalizuje výběrový rozptyl nové proměnné y_{1i} za podmínky $\boldsymbol{a}_1^\top\boldsymbol{a}_1 = 1$. Pakliže je výběrový rozptyl y_{1i} definován jako

$$D(y_{11}, \ldots, y_{1N}) = \boldsymbol{a}_1^\top\mathbf{C}\boldsymbol{a}_1 \tag{2.4}$$

můžeme maximalizační úlohu vyřešit pomocí metody Lagrangeových multiplikátorů. Lagrageova funkce s parametrem λ_1 má následující tvar

$$\mathcal{L}(\boldsymbol{a}_1, \lambda_1) = \boldsymbol{a}_1^\top\mathbf{C}\boldsymbol{a}_1 - \lambda_1(\boldsymbol{a}_1^\top\boldsymbol{a}_1 - 1). \tag{2.5}$$

Derivaci funkce položíme rovnou nule

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{a}_1} &= 2\mathbf{C}\boldsymbol{a} - 2\lambda_1\boldsymbol{a}_1 \stackrel{!}{=} 0\\ (\mathbf{C} - \lambda_1\mathbf{I})\boldsymbol{a}_1 &\stackrel{!}{=} 0,\end{aligned}$$

kde $\mathbf{I}$ je jednotková matice.

Řešíme soustavu lineárních rovnic pro neznámý parametr $\boldsymbol{a}_1$, která má řešení právě tehdy, když je matice $\mathbf{C} - \lambda_1\mathbf{I}$ singulární, tedy platí, že její determinant je roven nule. λ_1 je pak největší vlastní číslo matice $\mathbf{C}$ a $\boldsymbol{a}_1$ vlastní vektor příslušný tomuto vlastnímu číslu. Toto tvrzení se matematicky zapíše následovně

$$\mathbf{C}\boldsymbol{a}_1 = \lambda_1\boldsymbol{a}_1. \tag{2.6}$$

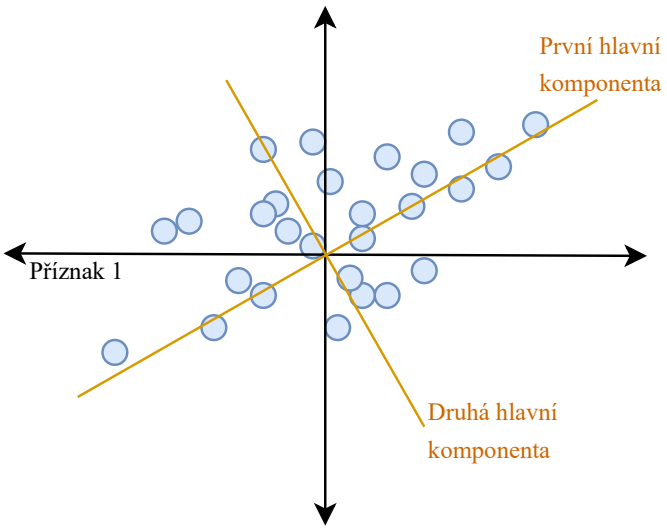
Po vynásobení vektorem $\boldsymbol{a}_1$ zleva získáme řešení pro maximální rozptyl proměnné y_{1i}

$$D(y_{11}, \ldots, y_{1N}) = \boldsymbol{a}_1^\top\mathbf{C}\boldsymbol{a}_1 = \lambda_1. \tag{2.7}$$

Druhá hlavní komponenta

$$y_{2i} = \boldsymbol{a}_2^\top(\boldsymbol{x}_i - \bar{\boldsymbol{x}}), \quad \text{pro } i = 1, \ldots, N, \tag{2.8}$$

se vypočte obdobným způsobem s přidanou podmínkou ortogonality vzhledem k první hlavní komponentě – druhá hlavní komponenta nesmí být korelovaná s předchozí, první hlavní komponentou. Potom popisuje druhý největší možný rozptyl v datech.



Obrázek 2.1: Znázornění dvou hlavních komponent na pro dvě proměnné. Zdroj: vlastní.

Znázornění dvou hlavních komponent ve dvoudimenzionálním prostoru je vyobrazeno na obrázku 2.1. Vektor $\boldsymbol{a}_2$ se opět získá jako jednotkový vlastní vektor kovarianční matice $\mathbf{C}$ příslušící druhému největšímu vlastnímu číslu λ_2.[7, 9]

Získání předpisů pro další hlavní komponenty je analogické. Obecně lze zapsat metodu PCA a převod původních proměnných následujícím maticovým zápisem

$$\mathbf{Y} = \mathbf{X}\mathbf{A}, \tag{2.9}$$

kde $\mathbf{X}$ je matice vstupních dat, $\mathbf{A}$ je matice vlastních vektorů kovarianční matice $\mathbf{C}$. Pro matici $\mathbf{A}$ zároveň platí $\mathbf{C} = \mathbf{A}\mathbf{\Lambda}\mathbf{A}^\top$, kde $\mathbf{\Lambda}$ je diagonální matice vlastních čísel $\mathbf{C}$.[8]

2.1.2 Korespondenční analýza

Vícenásobná korespondenční analýza (anglicky *Multiple correspondence analysis*, dále jako MCA) je metoda, která umožňuje popsat vztahy mezi daty, které jsou popsané kategorickými proměnnými, vytvořením kontingenční tabulky. V případě, že se popisuje vzájemná relace pouze dvou proměnných, se použije základní korespondenční analýza[1]. MCA je alternativou k PCA, pokud jsou analyzovanými daty kategorická data. [10]

Značení

Nechť $\mathbf{N}$ je matice dat s rozměry $I \times J$, kde I odpovídá počtu pozorování a J je počet kategorií. Matice $\mathbf{N}$ je převedena na korespondenční matici $\mathbf{P}$ vydělením matice $\mathbf{N}$ jejím celkovým součtem $n = \sum_{i=1}^{I}\sum_{j=1}^{J} n_{ij} = \mathbf{1}_I^\top \mathbf{N} \mathbf{1}_J$. To zaručuje, že součet prvků

[1] anglicky *correspondence analysis* (CA)

matice $\mathbf{P}$ je roven jedné. Tyto kroky lze shrnout následujícím matematickým zápisem

$$\mathbf{P} = \frac{1}{n}\mathbf{N}, \qquad \mathbf{P} = \{p_{ij}\}, \qquad \sum_{i=1}^{I}\sum_{j=1}^{J} p_{ij} = 1. \tag{2.10}$$

Součet $i$tého řádku, resp. součet $j$tého sloupce je značen následovně

$$r_i = \sum_{j=1}^{J} \qquad \text{pro } i = 1, \ldots, I,$$
$$c_j = \sum_{i=1}^{I} \qquad \text{pro } j = 1, \ldots, J.$$

Vektor $\boldsymbol{r} = \mathbf{P}\mathbf{1}_J$ obsahuje všechny řádkové součty matice $\mathbf{P}$, analogicky vektor $\boldsymbol{c} = \mathbf{P}^\top \mathbf{1}_I$ obsahuje všechny sloupcové součty téže matice.

Pro další výpočty zavedeme značení pro diagonální matice, které mají na diagonále řádkový, resp. sloupcový součet

$$\mathbf{D}_r = \mathrm{diag}(\boldsymbol{r}), \quad \text{resp.} \quad \mathbf{D}_c = \mathrm{diag}(\boldsymbol{c}). \tag{2.11}$$

Výpočetní algoritmus základní korespondeční analýzy [15, 11]

Označme $\mathbf{S} = \{s_{ij}\}$ následující matici

$$\mathbf{S} := \mathbf{D}_r^{-\frac{1}{2}}(\mathbf{P} - \boldsymbol{r}\boldsymbol{c}^\top)\mathbf{D}_c^{-\frac{1}{2}}. \tag{2.12}$$

Po té proveďme singulární rozklad této matice

$$\mathbf{S} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top, \tag{2.13}$$

kde $\boldsymbol{\Lambda} = \boldsymbol{\Delta}^2$ je matice vlastních čísel λ_k pro $k = 1, \ldots, K$, kde $K = min\{I-1, J-1\}$. Potom rozměry matice $\mathbf{U}$, resp. $\mathbf{V}$ jsou $I \times k$, resp. $J \times k$. Dále platí $\mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}$.

Korespondenční analýza měří míru váženého rozptylu, tzv. inercii pomocí vlastních čísel λ_k matice $\mathbf{S}$, λ_k se pak nazývají hlavní inercie. Celková inercie je rovna

$$I = \sum_{k=1}^{K} \lambda_k = \sum_{i=1}^{I}\sum_{j=1}^{J} s_{ij}^2. \tag{2.14}$$

Hlavní komponenta řádků $\mathbf{F}$ je rovna

$$\mathbf{F} = \mathbf{D}_r^{-\frac{1}{2}}\mathbf{U}\boldsymbol{\Delta}. \tag{2.15}$$

Hlavní komponenta sloupců $\mathbf{G}$ je rovna

$$\mathbf{G} = \mathbf{D}_c^{-\frac{1}{2}}\mathbf{V}\boldsymbol{\Delta} \tag{2.16}$$

Výpočetní algoritmus MCA

Předpokládejme, že původní matice kategorických dat má tvar $N \times Q$, tj. N pozorování a Q proměnných. Matici dat převedeme na indikátorovou matici. Indikátorová matice $\mathbf{Z}$ je vytvořena tak, že kategorická data jsou rozepsána do pomocných proměnných. Pokud $q$tá proměnná je má J_q typů kategorií, tak příslušná indikátorová matice bude mít $J = \sum_{q=1}^{Q} J_q$ sloupců a N. Tzn. počet proměnných byl tímto rozepsáním rozšířen z počtu původních Q proměnných na J proměnných. První způsob MCA aplikuje základní algoritmus korespondenční analýzy na matici $\mathbf{Z}$, takto se získají souřadnice pro N pozorování a J kategorií.

2.1.3 Korelační analýza

Korelační koeficient

Další způsoby měření závislosti

Pro určení kolik informace o jedné proměnné nese druhá proměnná, je popsáno pomocí vzájemné informace [12]. Informací lze rozumět obsah jakéhokoli oznámení nebo údaje, který se přenáší v daném čase a prostoru. Podle Shannona, zakladatele teorie informace, je informace míra množství neurčitosti nebo nejistoty o nějakém náhodném jevu, která se odstraní realizací daného jevu [13]. Informací tak může být stanovení výsledku náhodného jevu, tedy se jedná o hodnotu náhodné veličiny [12]. Pro definování vzájemné informace je třeba definovat ještě vlastní informace a pojem entropie.

Dále jsou sepsány předpoklady pro výpočet množství informace. Pokud má náhodný jev X n realizací, pak je množství informace funkcí n. Pakliže je $n = 1$, množství informace se rovná nule, neboť se jedná o jev jistý. Pokud jevy X A Y probíhají nezávisle, ale ve stejný čas, tj. $p_{XY}(x, y) = p_X(x) \cdot p_Y(y)$, potom množství informace obou jevů se tovná součtu jejich množství. Pokud jev X má n realizací a jev Y m realizací, kde $m > n$, potom se očekává, že množství informace jevu Y je větší než množství in formace jevu X. [13] Pokud je pravděpodobnost každé realizace stejná, tj. $p_X(x) = 1/n$, pak Hartleyho míra informace je definována jako funkce $I : \mathbf{N} \leftarrow \mathbf{R}$ ve tvaru $I(n) = \log n$. Pro vlastní míru informace obsažené ve výsledku x pak platí: [13, 14]

$$I(x) = -\log p(x). \tag{2.17}$$

Množství informace celého jevu je popsáno entropií náhodné veličiny. Entropie $H(X)$ náhodné veličiny X s hodnotami $x_1, x_2, \ldots$ s pravděpodobnostní funkcí $p(x)$ je rovna: [13, 3]

$$H(X) = -\sum_{x} p(x) \log p(x). \tag{2.18}$$

Nechť je dán vektor (X, Y), kde X, resp. Y je náhodná veličina nabývající hodnot $x_1, x_2, \ldots$, resp. $y_1, y_2, \ldots$. Náhodný vektor nabývá hodnot $(x_1, y_1), (x_2, y_2), \ldots$.

Sdružená entropie vektoru (X, Y) má tvar: [12, 14]

$$H(X) = -\sum_{x} p(x, y) \log p(x, y). \tag{2.19}$$

Pokles entropie se měří pomocí vzájemné informace, tj. platí věta [14]:

$$I(X; Y) = -H(X, Y) + H(X) + H(Y). \tag{2.20}$$

Vzájemná informace měří ztrátu informace v důsledku závislosti X a Y. Jinými slovy, kolik informace o jedné proměnné X nese druhá proměnná Y. Matematicky je vzájemná informace definována následovně: [12, 14, 3]

$$I(X; Y) = \sum_{(x,y)} + \log \frac{p_{X|Y}(x|y)}{p_X(x)} \tag{2.21}$$

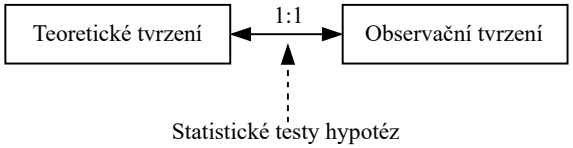
Theilovo U a Cramerovo V rozptylový inflační faktor (VIF)

2.1.4 Metoda GUHA

Metoda GUHA je původní česká metoda používaná pro nexplorační analýzu dat. První článek o této metodě vyšel v roce 1966. V současné době je jedním znejrozáshlejších implementací metody systém LISp-Miner. Jedná se o software vyvíjený na Fakultě informatiky a statistiky Vysoké školy ekonomické v Praze, kde se zároveň používá pro výuku a výzkum dobývaní znalostí z databází [38]. Zároveň je také implementována knihovna *CleverMiner* v jazyce Python, která disponuje částí funkcionalit softwaru LISp-Miner.

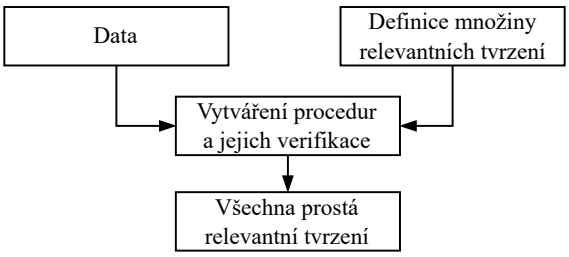
Základní princip metody

Cílem metody GUHA je získat z pozorovaných dat všechny vztahy, které jsou jsou pravdivé pro množinu objektů, ze které pochází zkoumaná data. Využívají se k tomu statistické testy hypotéz, které dovolují na základě platnosti určitého tvrzení o vzorku dat přijmout tvrzení o celé množině objektů. Pravdivé tvrzení o celé množině dat se nazývá *teoretické tvrzení*. Tvrzení o vzorku dat se nazývá *observační tvrzení*. Vztah $1:1$ mezi těmito tvrzeními zprostředkovávají statistické testy, znázorněno na obr. 2.2.[38]



Obrázek 2.2: Vztah mezi tvrzeními vzorku dat a celých dat v metodě GUHA. Zdroj: vlastní.

Základní postup GUHA procedury je na obrázku 2.3. Vstupem pro procedury jsou vstupní data a parametry, které definují množinu relevantních tvrzení. Na základě definice jsou vytvořena všechna relevantní observační tvrzení, která jsou verifikována podle dat. Výstupem jsou pak všechna všechna prosté observační tvrzení vycházející ze vstupů. Prosté relevantní tvrzení je takové tvrzení, které je pravdivé ve vstupních datech a zároveň neplyne již z uvedeného jiného tvrzení ve výstupu.[38]



Obrázek 2.3: Základní postup procdury GUHA. Zdroj: vlastní.

Důležité pojmy

Pro podrobnější popis procedur je nejprve třeba definovat několik pojmů, se kterými se v procedurách pracuje. Metoda pracuje s nádledujícími pojmy [38]:

- **Matice dat a atributy** – Řádky matice jsou jednotlivá pozorování. Atributem se rozumí sledovaná vlastnost, jedná se o sloupec matice.
- **Základní booleovský atribut** – Jedná se o výraz $\mathbf{A}(\boldsymbol{\alpha})$, kde $\mathbf{A}$ je atribut a $\boldsymbol{\alpha}$ je vlastní podmnožina $\mathbf{A}$. $\boldsymbol{\alpha}$ může obsahovat více prvků než jeden.
- **Booleovský atribut** – Každý základní boolovský atribut je booleovský atribut. Boolovské atributy jsou i negace, konjunkce a disjunkce základních boolovských atributů.

  Pro každý řádek i matice $\mathbf{M}$ nabývá boolovský atribut $\mathbf{A}$ hodnotu 0, nebo 1.

  $$\mathbf{A}[i] = 1 \Rightarrow \text{boolovský atribut } \mathbf{A} \text{ je pravdivý pro řádek } i.$$

  $$\mathbf{A}[i] = 0 \Rightarrow \text{boolovský atribut } \mathbf{A} \text{ je nepravdivý pro řádek } i.$$

- **Literál** – Základní boolovský atribut nebo jeho negace.
- **Dílčí cedent** – Konjunkce nebo disjunkce literálů.
- **Cedent** – Jedná se o konjunkci dílčích cedentů. Příkladem cedentu je booleovský atribut, který vznikl konkujnkcí a disjunkcí dalších atributů.

Další pojmy se týkají vztahů, se kterými procedury pracují [38]:

- **Asociační pravidlo** – Výraz X $\rightarrow$ Y, kde X a Y jsou konjunkce dvojic atribut a jeho hodnota. Dále v textu je používaná pro tento pojem zkratka AP.
- **Konfidence AP** – Podíl počtu řádků, které splňují A a zároveň S a počtu řádků, které splňují pouze S.
- **Podpora AP** – Podíl počtu řádků, které splňují A a zároveň S a počtu řádků vstupní matice dat.

Častou úlohou pro dobývání AP je nalezení všech AP, u kterých je hodnota konfidence a podpory AP větší nebo rovna danému prahu. V rámci GUHA se AP zkoumají jako vztah dvou obecných boolovských atributů, které jsou odvozené ze sloupců vstupní matice. GUHA asociační pravidlo (GUHA AP) je výraz

$$\varphi \approx \psi, \tag{2.22}$$

kde φ, ψ jsou boolovské atributy, které nemají obsažený žádný společný boolovský atribut. φ se nazývá *antecedent* a ϕ *sukcedent*[2]. Symbol $\approx$ odpovídá *4ft-kvantifikátoru*, viz dále v této sekci. Existují také podmíněná GUHA AP, která mají tvar $\varphi \approx \psi|\chi$, kde χ je boolovský atribut.[38]

Pravdivost GUHA AP v matici dat $\mathbf{M}$ se určuje pomocí tzv. *4ft-tabulky*. Nechť je dána matice vstupních dat $\mathbf{M}$, antecedent φ, sukcedent ψ. Pak *4ft-tabulka* $4\text{ft}(\varphi, \psi, \mathbf{M})$ je definována jako čtveřice čísel (a, b, c, d), pro které platí:

[2]Antecedent, jako cedent, který předchází a sukcedent, jako cedent, který následuje.

- a je počet řádků matice M, které splňují oba boolovské atributy φ, ψ.
- b je počet řádků matice M, které splňují φ, ale nesplňují ψ.
- c je počet řádků matice M, které nesplňují φ, ale splňují ψ.
- d je počet řádků matice M, které nesplňují ani jeden atribut φ, ψ.[38]

Reprezentace této tabulky je zobrazena v tab. 2.1.

Tabulka 2.1: *4ft-tabulka* matice $\mathbf{M}$ s asociačním pravidlem $\varphi \approx \psi$.

$\mathbf{M}$	ψ	$\neg\,\psi$
φ	a	b
$\neg\,\varphi$	c	d

4ft-kvanitifikátor, symbol $\approx$, definuje podmínku, která se týká hodnot (a, b, c, d) v *4ft-tabulce*. Kvantifikátor je formálně definovaný pomocí funkce $F_{\approx}$, která každé čtveřici nezáporných čísel přiřazuje hodnotu 1, resp. 0 pokud je, resp. není podmínka splněna. Zapisujeme $F_{\approx}(a, b, c, d)$ nebo zkráceně $\approx (a, b, c, d)$.[38]

$$\begin{aligned}&\text{GUHA AP } \varphi \approx \psi \text{ je pravdivé v matici dat } \mathbf{M}\\ &\qquad \Leftrightarrow \approx (a, b, c, d) = 1, \text{ formálně zapsáno jako } \mathrm{Val}(\varphi \approx \psi) = 1.\\ &\text{GUHA AP } \varphi \approx \psi \text{ je nepravdivé v matici dat } \mathbf{M}\\ &\qquad \Leftrightarrow \approx (a, b, c, d) = 1, \text{ formálně zapsáno jako } \mathrm{Val}(\varphi \approx \psi) = 1.\end{aligned} \tag{2.23}$$

Pro podmíněné AP $\varphi \approx \psi|\chi$ platí obdobné vztahy. Předpokládáme však, že boolovský atribut χ nemá ani jeden společný .atribut s atributy φ a ψ. Platí tvrzení [38]:

$$\begin{aligned}&\text{Nechť } \mathbf{M} \text{ je matice vstupních dat, } \varphi, \psi, \chi \text{ boolovské atributy, } \approx \text{ kvantifikátor.}\\ &\text{Podmíněné AP } \varphi \approx \psi|\chi \text{ je pravdivé v } \mathbf{M} \Leftrightarrow \varphi \approx \psi \text{ je pravdivé v matici } \mathbf{M}|\chi.\end{aligned} \tag{2.24}$$

Procedury

V dokumentaci [38] je popsáno sedm procedur – *4ft-Miner, SD4ft-Miner, CF-Miner, SDCF-Miner, KL-Miner, SDKL-Miner, Ac4ft-Miner* [38]. V knihovně v jazyce Python jsou implementované pouze metody *4ft-Miner, SD4ft-Miner, CF-Miner* [39]. V této práci jsem použila metodu pouze první metodu, proto další je další teoretický popis věnován pouze metodě *4ft-Miner*.

Tato procedura pracuje s AP $\varphi \approx \psi$, nebo s podmíněnými AP $\varphi \approx \psi|\chi$. V knihovně *Cleverminer* lze v hlavní funkci `cleverminer` předat vstupní DataFrame s daty, který reprezentuje vstupní matici dat, další parametr je jedna ze tří implementovaných procedur, dále seznam podmínek pro vyhodnocení tvrzení, vyplnutí

optimalizace, limit pro výsledná tvrzení a seznam cedentů. Cedenty jsou rozdělny na antecedenty (parametr `ante`, tj. boolovský atribut φ), sukcedenty (parametr `succ`, tj. atribut ψ) a podmínky (parametr `cond`, tj. boolovský atribut χ). Každý z boolovských atributů libovolného typu cedentu může mít tyto atributy:

- `name` – Název příznaku matice, tj. název sloupce v DataFramu.
- `type` – Jakým pravidlem se řídí výběr více kategorií v příznaku. Jedna z hodnot `subset, lcut, rcut, seq, one`.
- `minlen` – Minimální počet kategorií v daném příznaku.
- `maxlen` – Maximální počet kategorií v daném příznaku.[39]

Příznaky musí být kategorické a musí být možné je seřadit. Druhá vlastnost je třeba pro vybírání více kategorií v jednom cedentu určitými způsoby selekce. Pro textové řetězce reprezentující kategorie jsou názvy kategorií řazeny podle abecedy.[39]

Pro názornost jsou dále uvedeny příklady pro jendotlivé druhy atributu `type`. Nechť je dán příznak **A** s kategoriemi 1, 2, 3, 4, 5 a parametry jsou definovány následovně: `minlen=1`, `minlen=3`. Pokud je typ `one`, bere se jedna z kategorií daného příznaku, tuto kategorii je třeba specifikovat. Pro typ `subset` jsou vybrány všechny následující možnosti:

- Délka je rovna 1 – $\mathbf{A}(\mathbf{1}), \mathbf{A}(\mathbf{2}), \mathbf{A}(\mathbf{3}), \mathbf{A}(\mathbf{4}), \mathbf{A}(\mathbf{5})$.
- Délka je rovna 2 – $\mathbf{A}(\mathbf{1},\mathbf{2}), \mathbf{A}(\mathbf{1},\mathbf{3}), \mathbf{A}(\mathbf{1},\mathbf{4}), \mathbf{A}(\mathbf{1},\mathbf{5}), \mathbf{A}(\mathbf{2},\mathbf{3}), \mathbf{A}(\mathbf{2},\mathbf{4}), \mathbf{A}(\mathbf{2},\mathbf{5}), \mathbf{A}(\mathbf{3},\mathbf{4}), \mathbf{A}(\mathbf{3},\mathbf{5}), \mathbf{A}(\mathbf{4},\mathbf{5})$.
- Délka je rovna 3 – $\mathbf{A}(\mathbf{1},\mathbf{2},\mathbf{3}), \mathbf{A}(\mathbf{1},\mathbf{2},\mathbf{4}), \mathbf{A}(\mathbf{1},\mathbf{2},\mathbf{5}), \mathbf{A}(\mathbf{2},\mathbf{3},\mathbf{4}), \mathbf{A}(\mathbf{2},\mathbf{3},\mathbf{5}), \mathbf{A}(\mathbf{3},\mathbf{4},\mathbf{5})$.[38]

Pro typ sekvence, `seq` by se pak vybraly následující možnosti:

- Délka je rovna 1 – $\mathbf{A}(\mathbf{1}), \mathbf{A}(\mathbf{2}), \mathbf{A}(\mathbf{3}), \mathbf{A}(\mathbf{4}), \mathbf{A}(\mathbf{5})$.
- Délka je rovna 2 – $\mathbf{A}(\mathbf{1},\mathbf{2}), \mathbf{A}(\mathbf{2},\mathbf{3}), \mathbf{A}(\mathbf{3},\mathbf{4}), \mathbf{A}(\mathbf{4},\mathbf{5})$.
- Délka je rovna 3 – $\mathbf{A}(\mathbf{1},\mathbf{2},\mathbf{3}), \mathbf{A}(\mathbf{2},\mathbf{3},\mathbf{4}), \mathbf{A}(\mathbf{3},\mathbf{4},\mathbf{5})$.[38]

Pro typ `lcut` se vybírají možnosti:

- Délka je rovna 1 – $\mathbf{A}(\mathbf{1})$.
- Délka je rovna 2 – $\mathbf{A}(\mathbf{1},\mathbf{2})$.
- Délka je rovna 3 – $\mathbf{A}(\mathbf{1},\mathbf{2},\mathbf{3})$.[38]

Anlogicky pro typ `rcut`.

Literály v rámci cedentů lze také kombinovat obdobnými způsoby. Opět lze přiřadit minimální a maximální délku, typ pro kombinování leterálů je výběr konkunkce,

nebo disjunkce. Tyto monžosti lze zpecifikovat pro antecedenty, sukcedenty i podmínky. Zadání podmínek není nezbytné v atributech funkce `cleverminer`.

Další parametry, které lze předat této funkci jsou:

- `Base` – Minimální počet řádků, které splňují antecedenty i sukcedenty (číslo a v tabulce 2.1).
- `RelBase` – Hodnota Base vydělená celkovým počtem řádků dat (případně počtem řádků v matici s aplikovanou podmínkou).
- `conf` – Konfidence, pravděpodobnost $/mathrm{P}(\psi|\varphi)$. Jinými slovy procentuální zastoupení řádků, které vyhovují ψ (sukcendentům) z těch řádků, které vyhovují i φ (antecendtům).
- `aad` (nadprůměrná závislost) – Jak moc φ zvyšuje pravděpodobnost ψ. Kolikrát se zvýší pravděpodobnost splnění sukcedentů, když se vezmou pouze záznamy, které vyhovují antecentům, oproti všem záznamům minus 1.
- `bad` (podprůměrná závislost) – Jak moc φ snižuje pravděpodobnost ψ.

Příklad volání funkce `cleverminer` je sepsaný v ukázce kódu č. 2.1.

```
cleverminer(df = data,
            proc = "4ftMiner",
            quantifiers = {"conf":0.6, "Base":1000},
            ante = {
                    "attributes":
                    [
                        {
                            "name":"weekday",
                            "type":"subset",
                            "minlen":1, "maxlen":3
                        },
                        {
                            "name":"quarter",
                            "type":"lcut",
                            "minlen":1, "maxlen":4
                        }
                    ],
                    "minlen":1, "maxlen":3, "type":"con"
                    },
            succ = {
                    "attributes":
                    [
                        {
                            "name":"L3",
                            "type":"subset",
                            "minlen":1, "maxlen":3
                        }
                    ],
                    "minlen":1, "maxlen":1, "type":"con"
                    },
            cond = {
                    "attributes":
                    [
                        {
                            "name":"promo",
                            "type":"one",
                            "value":"promo"
                        }
```

```
                    ],
                    "minlen":1, "maxlen":1, "type":"con"
                    }
            )
```

Kód 2.1: Příklad volání funkce `cleverminer`.

Kapitola 3

Shrink

Cílem této práce je analyzovat shrinky produktů, které byly zaznamenány v datech dané společnosti, a zjistit příčiny jejich vzniku. V následující části je vysvětlen pojem shrink a popsány kategorie, které vybraná společnost rozeznává ve svých datech.

3.1 Definice

Defince pojmu shrink v oblasti retailu není jednotná. Problematikou, jak přesně označit ztrátu produktů, se zabývá Beck ve svém článku [37]. V této práci je slovem shrink označována ztráta zisku z neuskutečněného prodeje hotového produktu. Tento produkt je vyroben, či naskladněn, ale z nějakého důvodu nemohl být prodán zákazníkovi. Tímto důvodem může být například poničení produktu, jeho ztráta nebo prošlá doba spotřeby. Za shrink produktu lze označovat i stav, kdy cena produktu je neplánovaně snížena v důsledku zmíněných důvodů. Shrinkem je potom rozdíl plánované prodejní ceny a ceny, za kterou byl produkt skutečně prodán [22]. Tuto definici jsem zvolila vzhledem k povaze dat anlyzované společnosti.

Často se v literatuře lze setkat s pojmem shrink, resp. anglickým *shrinkage* v retailových společnostech především v souvislosti se ztrátou zboží z důvodu krádeže – interní neboli zamšstnanecká, externí neboli zákaznická, dále z důvodu administrativního pochybení, nebo podovodem na straně dodavatele. [36, 37]

3.2 Typy shrinků

Shrinky je možné kategorizovat podle Becka do čtyř kategorií podle čtyř oblasti, kde se shrink odehrává. Kategorie se dále dělí na ztráty, které jsou známé a neznámé. Neznámé ztráty z povahy věci nejde evidovat, takže se dále již nedělí. Známou ztrátu lze dále rozdělit na úmyslné a neumýslné. Rozdělení je podrobně znázorněno na obrázku 3.1.

Vybraná společnost rozlišuje ve svých datech tři kategorie shrinku – shirnky způsobené inventurou, škodami a cenové snížení. Dále se text věnuje popisu jednotlivých typů

Obrázek 3.1: Topologie shrinků. Zdroj: [37]

v rámci těchto kategorií v analyzované společnosti. Každý typ má přiřazeno jednoznačné identifikační číslo, podle kterého je zaznamenáván v databázi. Z důvodů anonymizace dat v práci nejsou uvedené přesné hodnoty těchto ID, namísto toho jsou uvedeny pouze názvy, které definují shrinky.

Shrinky způsobené inventurou Tato kategorie sdružuje všechny shrinky týkající se změn ve stavech zásob. Tyto změny se projeví při inventuře. V tabulce 3.1 se nachází přehled všech evidovaných typů. Některé typy mají obdobný význam a jsou duplicitní. K tomu mohlo dajít patrně tím, že některé subjekty používají dřívější značení pro inventuru, než jiné subjekty, které mohli přejít na nový, podorbnější způsob záznamu.

Shrinky způsobené škodami

Do kategorie shrinků způsobených škodami jsou řazeny zbylé důvody k odstranění produktu z prodeje z důvodu degradace produktu. V následující tabulce 3.2 jsou vypsané všechny typy, které moho být evidovány.

Snížení ceny

Tento typ shrinku vzniká v důsledku snížení ceny na prodejně. Tento shrink není přímo evidovaný v datech, ale lze jej vypočítat ze záznamů prodejů. Jedná se o situaci, kdy přímo na prodejně je nějaký produkt zlevněný v důsledku blížící se expirace nebo z důvodu poničení obalu. Nejedná se tak o klasickou promoakci, ale o

zlevnění, které není evidováno systémem, protože se netýká všech produktů daného typu, ale pouze jednoho či několika konkrétních produktů na vybrané prodejně.

Postup pro zjištění velikosti shrinku pro jeden konkrétní produkt je následovný. Pro každou účtenku je třeba porovnat cenu každého prodaného produktu s ceníkovou cenou, případně promoční slevou. Pokud si tyto ceny nejsou rovné, pak rozdíl těchto cen je shrink daného produktu.

Vzhledem k tomu, že denně se na každé prodejně zaevidují stovky účtenek, bylo by toto postupné procházení velmi časově náročné. Zároveň tento shrink postihuje jen velmi malou část celkového prodaného objemu. Tento shrink jsem ve svých analýzách již dále nezkoumala, protože nebyl shledán prioritním. Určení příčin vzniku takového shrinku se může lišit v závislosti na konkrétních prodejnách, a to jak na zaměstnaních, které vytváří snížení cen, tak na spotřebitelích, kteří na konkrétních prodejnách nakupují.

Tabulka 3.1: Přehled jednotlivých typů shrinků z kategorie inventory.

Název	Popis
Inventura - příjem	Kladné připsání zboží během inventury.
Inventura - odpis	Záporné odepsání zboží během inventury.
Inventura - velká	Velká inventura skladu.
Inventury - oprava	Dodatečné opravy, které bylo třeba provést po dokončení velké inventury.
Inventura - částečná	Odpis, nebo naskladnění zboží při inventuře položek.
Neuznané reklamace centrálním skladem	Odpis zboží, které bylo fyzicky dodané z centrálního skladu na prodejnu, ale prodejna jej vrátila, ale vratka nebyla uznána.
Inventura	Starší verze ID používaného pro inventuru.
Neexistující zboží	Odpis prokazatelně ukradeného zboží nebo i ztraceného zboží.

Tabulka 3.2: Přehled jednotlivých typů shrinků z kategorie damages.

Název	Popis
Poškození	Odpis zboží, které bylo poškozené. Např. nedopečené, spálené, špatně vyrobené nebo poškozené zaměstnancem nebo zákazníkem (kdy nelze uplatnit reklamaci na zákazníka.)
Prošlé a zkažené zboží	Odpis zboží, kterému prošla doba spotřeby (v případě výrobků, kde je datum uvedené), zkažené či shnilé zboží (ovoce, zelenina) nebo ztvrdlé pečivo.
Zákaznické reklamace	Odpis zboží, které zákazník reklamoval a reklamace byla uznána, ale zároveň nelze toto zboží reklamovat u dodavatele.
Reklamace centrálního skladu	Odpis zboží, které fyzicky nedorazilo z distribučního centra a nebylo možné ho reklamovat z důvodu nesplnění limitu pro vytvoření reklamace na distribučním centru. Také obsahuje odpisy neprodaných položek po ukončení výprodeje.
Kompostéry	Odpis zboží, které je prošlé nebo poškozené a které prodejna zlikviduje v kompostéru.
Potravinová banka	Odpis potravinářského zboží, které bylo darováno potravinovým bankám. Jedná se o produkty, které nebylo možné zařadit znovu do oběhu.
Zvířecí útulky	Odpis potravinářského zboží, které bylo darováno do útulků zvířat. Jedná se o produkty, které nebylo možné zařadit znovu do oběhu.
Poškození vnějšími vlivy	Odpis zboží, které bylo poškozeno nebo zničeno vlivem třetí strany (výbuch, vytopení, poškození majetku) nebo přírodními živly. Zboží se tedy na prodejně nenachází a nemůže proto být zlikvidováno.
Zničení	Jinak zničené zboží

Kapitola 4

Zpracování dat

Tato kapitola se zabývá popisem práce s konkrétní datovou sadou, kterou jsem obdržela. Z důvodu ochrany dat se v textu nevyskytují přesná pojmenování, ani není možné zobrazit přesnou strukturu uložení dat.

4.1 Popis obdržených dat

Všechna data poskytnutá společností jsou uložena v databázi, ke které byl zhotoven omezený přístup pro účely získání dat pro analýzy shrinku produktů společnosti. Zároveň s možností přístupu jsem obdržela i tabulku, která stručně komentuje všechny tabulky v databázi a sloupce v jednotlivých tabulkách. Celkem se v databázi nachází přes čtyři sta tabulek, z nichž bylo potřeba vybrat pouze ty, které obsahují relevantní data pro úlohu shrinků.

Z důvodu ochrany dat nelze uvádět přesné názvy tabulek, nicméně pro lepší orientaci v textu, každé použité tabulce přiřadím název, který odpovídá obsaženým datům v tabulce.

Číselníky

Základní číselník s údaji o produktech, se nachází v tabulce `produkt` se 27 sloupci. Pro analýzu vzniku shrinků jsem z této tabulky vybrala jako možné významné údaje následující sloupce:

- **ID produktu**
- **ID prodejní varianty** – Určuje o jaký typ balení daného produktu se jedná
- **Expirace** – Expirace produktu ve dnech (hodnoty 0, 999 a NULL označují neomezenou expiraci)
- **ID kategorie** – Kategorie produktu v číselné struktuře (pro lepší interpretaci, o jakou kategorii zboží se jedná, je vhodnější použít strukturu podle úrovní, kterou lze získat napojením na tabulku `produkt_kategorie`.)

- **Aktivní** – Zda je tento produkt stále aktivní v portfoliu, nebo se jedná o produkt, který se již neprodává

Tabulka `produkt_kategorie` obsahuje převod z číselné struktury do struktury pomocí produktové hierarchie. V obdržených datech má produktová hierarchie šest úrovní. Hierarchie produktů tvoří tedy strom se šesti úrovněmi. Nejvyšší úroveň, tj. úroveň číslo 1 má šest kategorií.

V tabulce 4.1 jsou uvedeny počty podkategorií pro každou z kategorií z nejvyšší úrovně. Také je uvedeno procentuální zastoupení kategorií v nejvyšší úrovni v rámci produktového portfolia vybrané společnosti. Zastoupení je odvozeno podle počtu produktů v kategorii.

Tabulka 4.1: Počet podkategorií na jednotlivých úrovních a zastoupení nejvyšší kategorie v rámci produktového portfolia.

	Název kategorie	Počty kategorií					Zastoupení kategorie
Úroveň:	1	2	3	4	5	6	
	Nepotravinářské	1	7	27	76	179	76,12%
	Suché	3	13	33	147	494	7,28%
	Kosmetika a drogerie	1	4	21	59	193	7,07%
	Čerstvé	5	11	27	111	469	4,27%
	Velmi čerstvé	6	10	31	92	271	4,04%
	Ostatní	4	4	4	5	5	1,06%
	Tabák	1	1	1	3	8	0,17%

Poslední, šestá úroveň hierarchie je přímo napojená na hodnotu číselné struktury, která je uvedena v číselníku produktů (v tabulce `produkt`). Pro získání všech úrovní kategorizace po úrovních k danému produktu je třeba vyhledat v tabulce `produkt` číselné ID kategorie daného produktu a napojit jej na poslední úroveň v tabulce produktové hierarchie (`produkt_kategorie`). V této tabulce je pak uvedena rodičovská kategorie z úrovně 5. Poté je potřeba opět vyhledat v tabulce `produkt_kategorie` tuto hodnotu a zjistit její nadřazenou kategorii. Takto se postupuje dokud není dosaženo nejvyšší úrovně. Tyto operace jsem provedla SQL příkazem přímo nad databází. Použila jsem vnitřní spojení na každou úroveň hierarchii na sloupce kategorie a rodičovská kategorie.

Další tabulka, se kterou jsem pracovala obsahuje informace o velikosti a hmotnosti produktů. Tato tabulka je důležitá z toho důvodu, že některé položky jsou vážené. Pokud se udává jejich množství udává se v gramech, zatímco neválené položky jsou uvedeny v kusech. Aby bylo možné porovnávat oba číselné údaje, ke každému váženému produktu existuje přepočet na počet kusů (ozn. SKU). K tomu jsou využity údaje o počtu kusů na jednu vychystávací jednotku (dále označeno jako SKU_{VJ}) a hmotnost jedné vychystávací jednotky daného produktu (ozn. m_{VJ}).

Vychystávací jednotka je jednotka množství používaná pro vychystávání produktů – jeho balení a transport. Postup pro přepočet hmotnosti produktu na počet kusů (SKU_{v}) je následovný:

$$SKU = \frac{m}{m_{\mathrm{VJ}}} \cdot SKU_{\mathrm{VJ}},$$

kde m je hmotnost produktu. Ze vzorce vyplývá, že může vejít neceločíselný počet kusů. Vzhledem k tomu, že tento přepočet se použije k porovnávání velikosti objemů, nikoli k objednávání zboží, tak tato skutečnost není problém.

Číselník prodejen je obsažen v tabulce `prodejny`. Vybrala jsem z tabulky následující sloupce.

- **ID prodejny** – Označení prodejny nebo skladu
- **Název** – Název prodejny, který obsahuje název města, kde se prodejna nachází.
- **ID kategorie prodejny** – Do jaké kategorie prodejna nebo sklad patří - zda se jedná o malou nebo velkou prodejnu nebo o sklad.

S číselníkem prodejen souvisí číselník pro jejich zařazení do skupin `prodejny_skupiny`. Skupiny se mohou v čase měnit. Pro analýzu jsou relevantní tyto sloupce:

- **ID prodejny** – Označení prodejny nebo skladu
- **ID skupiny prodejen** – Prodejny jsou sdruženy do skupin. Ty se například používají pro hromadné objednávání, nebo pro plánování promoakcí.

Promoakce se nachází v tabulce `promoakce`. Z této tabulky jsou pro následnou analýzu potřebné údaje o ID produktu, počátečním a koncovém datu promoakce a ID skupiny prodejen, na kterých promoakce platí. Promoakce nejsou přiřazené na konkrétní prodejny, ale na skupiny prodejen. Pro další analýzy shrinků je třeba zjistit, zda byl konkrétní zaznamenaný shrink v době záznamu v promoakci, nebo ne. Z tohoto důvodu bylo potřeba tabulky spojit pomocí příkazu `JOIN` s číselníkem `prodejny_skupiny` podle ID prodejny.

Tabulky transakcí

V tabulce `transakce` se nachází údaje o všech provedených transakcích, a to jak skladové transakce, tak prodeje a další pohyby na prodejnách. V případě prodejů prodejen jsou údaje agregované podle prodejny, konkrétního produktu a dne transakce, tzn. v této tabulce nelze rozlišit konkrétní prodeje na jednotlivých pokladnách, ale pouze souhrn za jeden den. Tabulka obsahuje údaje za posledních dvanáct měsíců.

Tabulka transakcí obsahuje 21 sloupců, jako možné podstatné sloupce pro analýzu jsem vybrala následující sloupce:

- **ID transakce** – Jedinečné pro každou transakci.
- **ID produktu** – Produkt kterého se transakce týká. Každá transakce obsahuje údaje pouze o jediném produktu.

- **ID prodejny** – Transakce je takto přiřazená prodejně, případně skladu.
- **Datum transakce** – Jedná se o obchodní datum, pokud samotná transakce proběhne až po půlnoci uvedeného dne, tak se posílá s datem z předchozího dne, neboť obchodně patří do toho dne.
- **ID promoce** – Příznak zda a v jaké promoční akci se produkt nacházel v čase uvedeném v datu transakce. V rámci zpracování dat vyplynulo, že tento příznak není zcela věrohodný
- **ID shrinku** – Obsahuje označení jednotlivých typů shrinků viz sekce 3.2. Celkem je identifikováno sedmnáct typů shrinků. V databázi tento sloupec označuje i jiná ID než ta, která se týkají shrinků, z toho plyne, že bylo třeba vyfiltrovat pouze ta data, která obsahují sedmnáct identifikačních čísel označující shrinky.
- **Objem** – Množství produktu uvedené v transakci. U kusových produktů se jedná o celočíselný údaj u vážených to je desetinné číslo.
- Hodnota transakce v nákladové ceně (desetinné číslo).
- Hodnota transakce v prodejní ceně včetně DPH – v případě prodejů se jedná o skutečnou cenu, u zbylých transakcích je uvedena odpovídající cena podle ceníku.

Velikost tabulky transakcí za jeden kalendářní rok je přibližně 3.5 GB.

Tabulku, která obsahuje údaje o jednotlivých prodejích na prodejnách společnosti, jsem pro účely této práce nazvala `transakce_prodeje`. Celkem obsahuje třináct sloupců. Tato tabulka je vhodná pro analýzu shrinků typu snížení ceny, analýzou tohoto typu se tato práce nezabývá. Pro ostatní typy, není tato tabulka relevantní. Stejně tak není třeba zkoumat ceník jednotlivých produktů, protože v souhrnné tabulce transakcí je již uvedená hodnota transakce v prodejní ceně.

Další datové zdroje

Dále jsem pracovala s daty z databáze Českého statistického úřadu [35]. Na webové stránce úřadu je dostupný odkaz ke stažení souboru ve formátu xlsx. Soubor obsahuje údaje o 237 českých městech za posledních několik desítek let. Některá města obsahují záznamy až sto let nazpět, jiné nemají tak dávno zaznamneanou historii. Dataset obsahuje údaje o lokalite, o počtu obyvatel, o sňatcích, rozvodech, stěhování obyvatel a další. V rámci přípravy dat bylo potřeba napojit prodejny k údajím o okresu, kraji a počtu obyvatel, kteří žijí v okolí prodejny. Soubor s demografickými údaji bylo třeba převést do tabulkové struktury, kde každý řádek patří jednomu městu, protože původní struktura byla nastavená, co list v souboru, to jedno město. Navíc stejné informace nejsou vždy umístěné stejně na každém listu.

Kapitola 5

Příprava vzorku dat pro další analýzy

Vzhledem k rozsáhlosti dat Následující část text bude věnována rozboru dat pro shrinky typu damages v období jednoho kalendářního měsíce.

5.1 Předzpracování dat

Vzhledem k vysokému počtu dat pro jeden kalendářní rok, v roce 2022 bylo v databázi evidováno přes 32 milionů záznamů o týkající se shrinků , jsem se rozhodla provést analýzu na měsíčním výběru dat z tohoto období. Jako zkoumaný měsíc jsem vybrala měsíc říjen, neboť v porovnání s letními měsíci a Vánocemi se v říjnu nevyskytují významné sezónní výkyvy.

Zkoumaná březnová data obsahují přes dva miliony řádků a patnáct sloupců. Každý řádek odpovídá jednomu záznamu v databázi shrinku daného produktu. Sledované údaje ve sloupcích jsou:

- ID prodejny, kategorická proměnná,
- ID produktu, kategorická proměnná,
- datum transakce, kategorická proměnná,
- ID shrinku, kategorická proměnná,
- 1, kategorická proměnná,
- 2, kategorická proměnná,
- 4, kategorická proměnná,
- 5, kategorická proměnná,
- 6, kategorická proměnná,
- expirace, kategorická proměnná,
- množství, spojitá proměnná,

- ztracená nákladová cena, spojitá proměnná,
- den v týdnu, kategorická proměnná,
- číslo den, kategorická proměnná,
- období v měsíci (rozdělení měsíce na pět částí), kategorická proměnná.

Původní sloupec datum jsem rozdělila na tři jiné proměnné, a to den v týdnu, číslo dne a období v měsíci a sloupec datum jsem vynechala.

5.1.1 Výběr dat

Nejprve jsem graficky analyzovala zastoupení shrinků v závislosti na vybraných proměnných pomocí nástroje Power BI, viz obr. 5.1. V návaznosti na zjištěné zastoupení shrinků v datech jsem se rozhodla vybrat pouze ty typy shrinků, které tvoří více jak jedno procento z celkových nákladů (tj. náklady činily alespoň jeden milion korun). Vynechala jsem tedy shrinky s označením 5 až 9 a naopak shrinky 0 až 4 byly ponechány. Obdobně jsem přistupovala k záznamům i z hlediska kategorie produktu úrovně 1, jelikož z grafu je patrné, že majoritní zastoupení mají pouze dvě kategorie, a to kategorie superfresh a fresh produktů. Všechny záznamy se zbylými kategoriemi jsem z datasetu odstranila. Těmito kroky jsem zredukovala původní počet řádků datasetu na necelých jeden a půl milionu řádků.

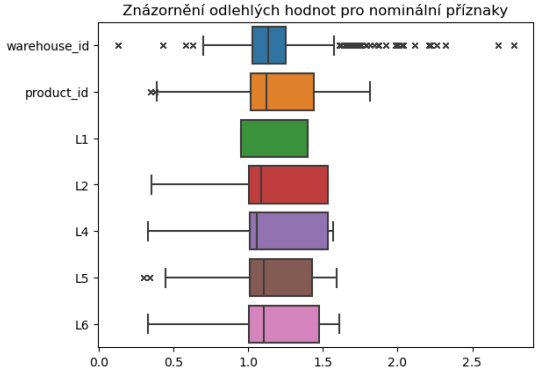
Obrázek 5.1: Zastoupení shrinků typu damage a zastoupení kategorie 1 v datech z března roku 2023.

Jako cílové sloupce (*target* sloupce) jsem určila sloupec s typem shrinku, množstvím produktu a nákladovou cenou. Zbylých jedenáct sloupců slouží jako vysvětlující proměnné, dále budou označovány jako příznaky pro cílový sloupec. Všechny vybrané příznaky jsou kategorické proměnné, které lze dále rozdělit na nominální a ordinální. Nominální proměnné jsou ID prodejny, ID produktu, kategorie 1, 2, 4, 5, 6. Ordinální proměnné jsou expirace, den v týdnu, číslo dne a období měsíce. Ordinální příznaky jsem přeznačila tak, aby každá obsahovala pouze hodnoty od nuly do n_p, kde n_p je počet kategorií v p-tém příznaku.

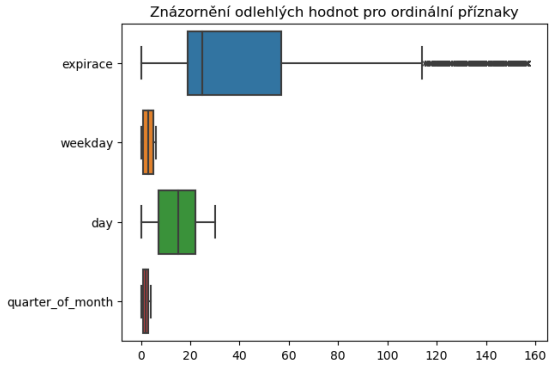
Pro další výpočty bylo vhodné přesunout se z nominálních kategorických hodnot na číselné hodnoty. Pro tyto účely jsem zvolila metodu *target encoding*. Neboť toto kódování na numerické hodnoty zachovává velikost datového souboru, to je klíčové vzhledem k tomu, že nominální proměnné ve zkoumaných datech obsahují

velký počet kategorií. Např. počet unikátních produktů v datech je 19 026, což odpovídá stejnému počtu kategorií pro tuto proměnnou. Pokud bych použila one-hot kódování[1] mohlo by dojít k zásadnímu zvýšení počtu sloupců v datech, v tomto případě až o desítky tisíc. *Target kódování* je podobné převodu, který jsem použila pro ordinální proměnné. Avšak na rozdíl od něj, hodnota, která je kategorii přiřazena, souvisí se zastoupením této skupiny v cílovém sloupci a nesouvisí s uspořádáním hodnot uvnitř příznaku. Nevýhodou je, že takto upravená data mohou být náchylná na overfitting, proto je potřeba při predikování použít křížovou validaci.[24]

Dále jsem se zabývala identifikací odlehlých hodnot. Nejprve jsem vizualizovala hodnoty pomocí grafu, obrázky 5.2 a 5.3. Z grafu je patrné, že problémová je proměnná `warehouse_id`, která označuje ID prodejny. Prodejny, které tvoří outliery mohou být malé prodejny, které kvůli menšímu počtu celkových produktů neevidují větší počet shrinků.



Obrázek 5.2: Znázornění odlehlých hodnot pro nominální příznaky.



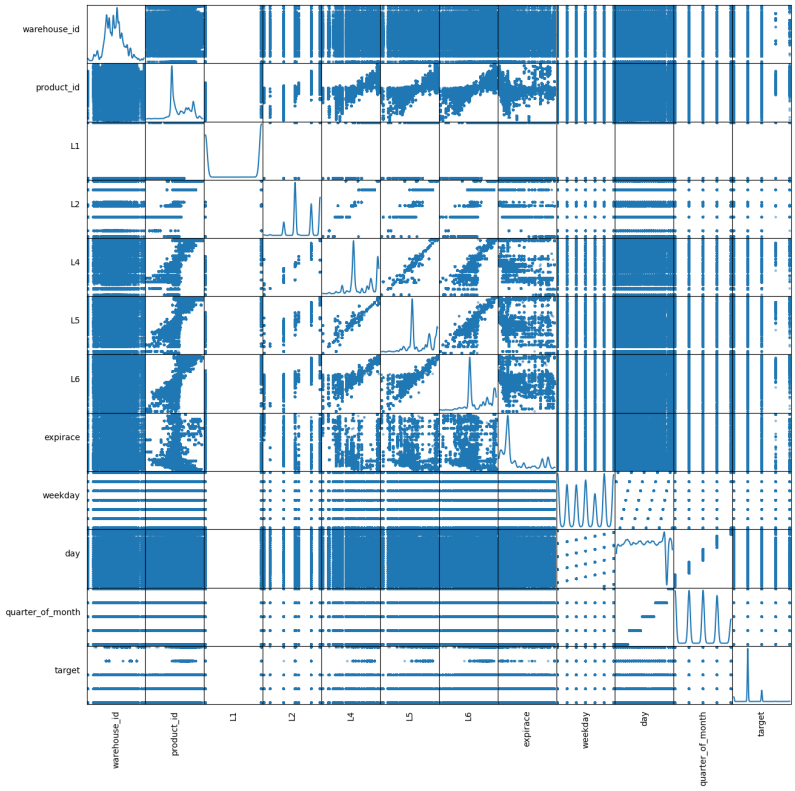
Obrázek 5.3: Znázornění odlehlých hodnot pro nominální příznaky.

Pomocí Tukeyho testu jsem identifikovala přes 150 000 outlierů pro příznak ID prodejny (`warehouse_id`), čímž se dataset zredukoval na 1 218 453 řádků. S tímto krokem klesl i počet ostatních outlierů.

V dalším kroku jsem se zaměřila na míru korelace mezi proměnnými. Vizualizovala jsem data pomocí scatter matice pro všechny proměnné, matice je možné vidět na obr. č. 5.4. Z této matice můžeme na první pohled vidět, že příznaky odpovídající produktové hierarchii a ID produktu vykazují závislost, což plyne z definice uspořádání této hierarchické kategorizace. V následujících krocích je cílem vybrat tu kategorii, která nejlépe popisuje data ve vztahu k shrinkům.

Jako první metodu jsem zvolila χ^2 test. Vzhledem k vysokému počtu dat je matice příliš řídká, a proto nejsou výsledné hodnoty vypovídající a test je tedy pro tuto úlohu nespolehlivý. Jiným měřítkem pro korelaci mezi proměnnými je Pearsonův korelační koeficient. Výslednou matici popisující korelační vztahy mezi příznaky jsem vizualizovala teplotní mapou, která je zobrazena na obrázku 5.5. Z výsledků

[1] One-hot kódování převádí kategorické hodnoty na numerické takovým způsobem že pro každou kategorii vytvoří samostatný sloupec s binárními hodnotami, kde 1 odpovídá dané kategorii a 0 zbylým kategoriím.



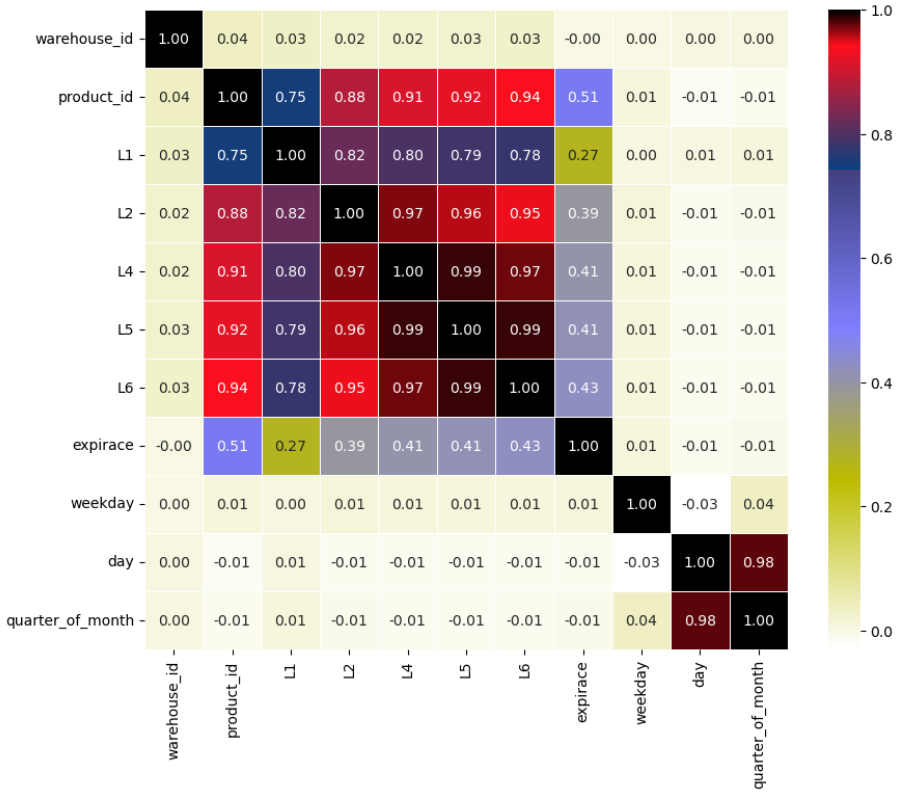
Obrázek 5.4: Scatter matice příznaků.

je opět patrné, že mezi jednotlivými kategoriemi produktů a produkty je silná korelace. Toto zjištění je zcela logické, neboť se jedná o stromovou strukturu kategorií. Zároveň existuje korelace mezi produktovými kategoriemi a expirací produktu. p-hodnota odpovídající jednotlivým koeficientům byla vždy nulová, kromě pro koeficient týkající se dvojice proměnných expirace a ID prodejny a expirace a pořadí dne v týdnu. Je tedy možné považovat výsledky (kromě těchto dvou výjimek) za statisticky významné.

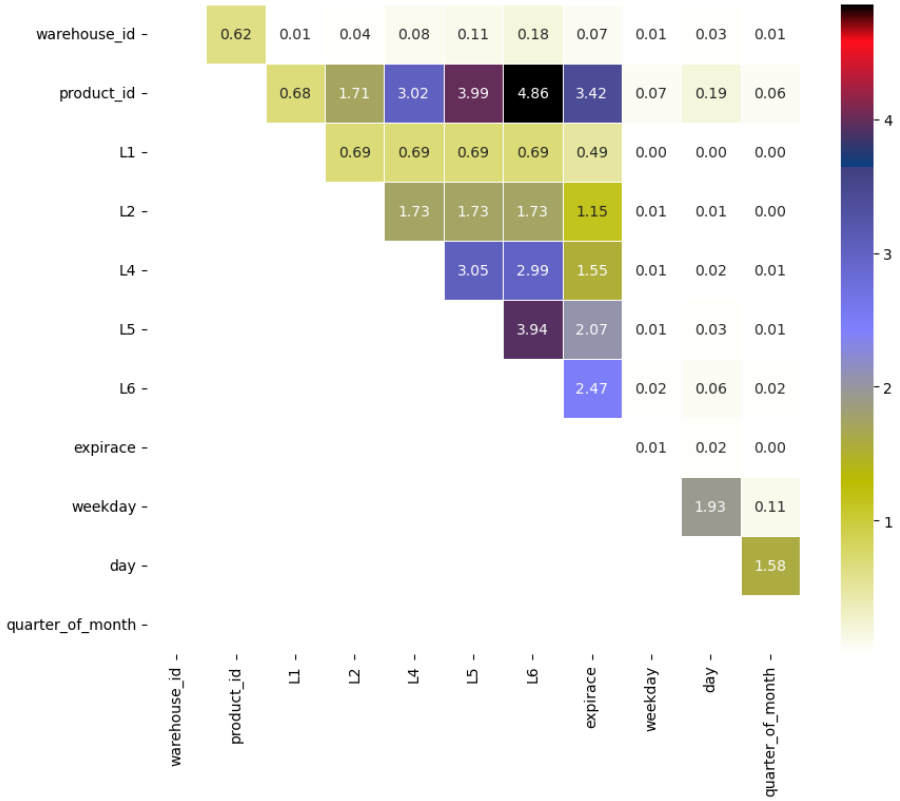
Dále jsem použila výpočet koeficientů vzájemné informace[2], který říká, jaká je podobnost mezi dvěma proměnnými [25]. Matice vypočítaných koeficientů je na obr. 5.6, jelikož se jedná o symetrickou vlastnost, jsou vynechány hodnoty pod vedlejší diagonálou. Z výsledků je opět vidět, že ID produktu sdílí informaci s úrovněmi kategorizace tím více, čím je kategorizace jemnější.

Dále jsem pro znázornění vztahu mezi proměnnými použila koeficient Cramerovo V. Koeficient jsem postupně počítala pro každou dvojici příznaků. Koeficient nabývá hodnot mezi 0 a 1. Číslopř blízké nule indikuje, že mezi proměnnými není asociace, číslo blízké jedničce vysokou závislost [27]. Na obr. 5.7 lze vidět, že pro kategorie 1 až 6 je hodnota koeficientu po zaokrouhlení rovna jedné. Vysoká závislost je pak i mezi příznakem expirace a ID produktu a kategorií 1. Dále logicky mezi číslem dne

[2] *mutual information*

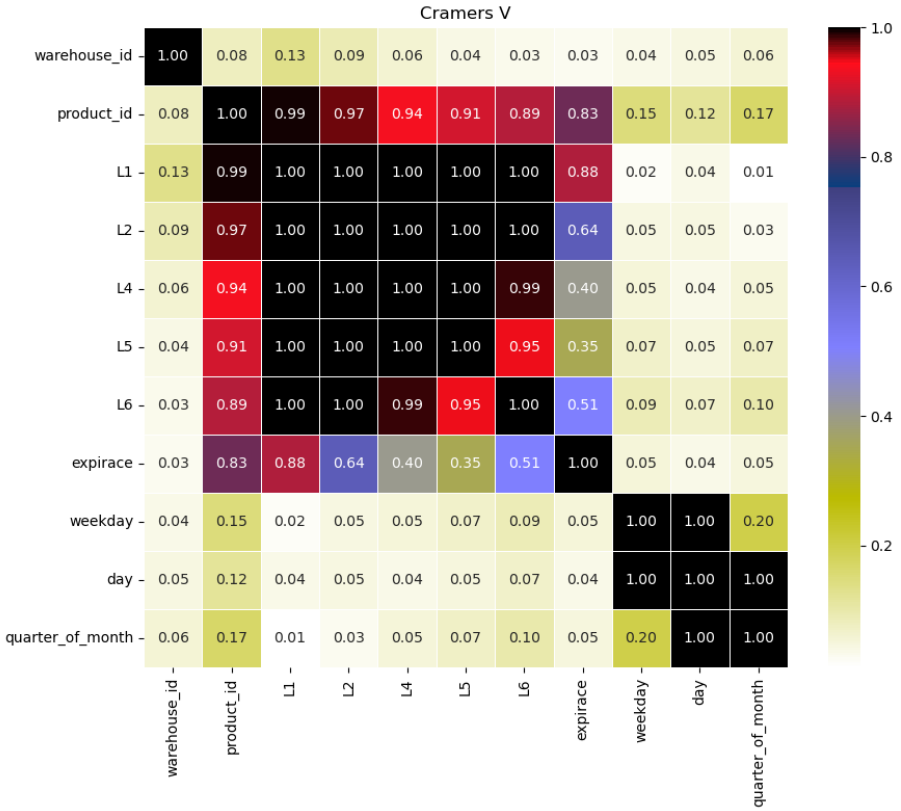


Obrázek 5.5: Matice korelačních koeficientů mezi příznaky.



Obrázek 5.6: Matice koeficientů vzájemné informace mezi příznaky.

a dnem v týdnu a obdobím v měsíci.

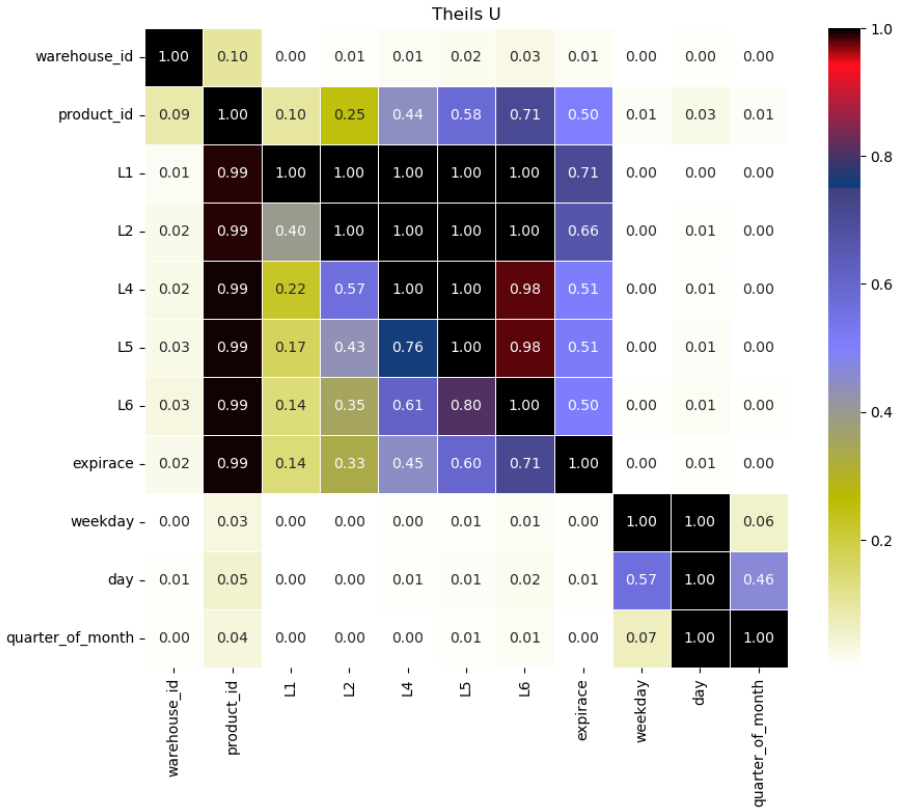


Obrázek 5.7: Matice koeficientů Cramerovo V mezi příznaky.

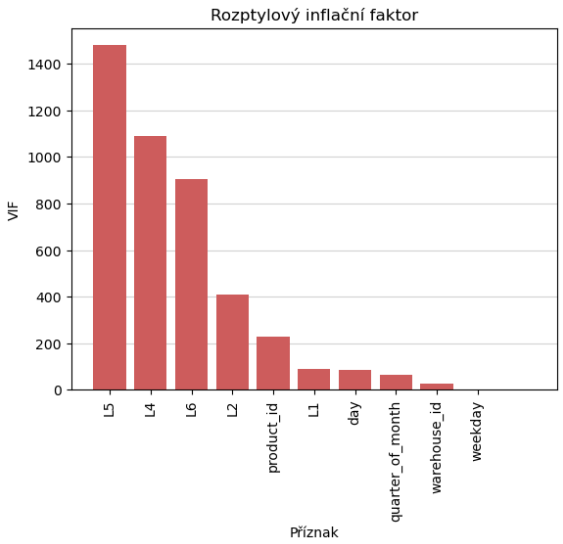
Další statistikou spočtenou na datech je Theilovo U (neboli koeficient nejistoty), který opět nabývá hodnot mezi 0 a 1 a měří vztah mezi dvěma proměnnými. Na rozdíl od předchozích statistik tento koeficient není symetrický a z výsledků lze vyvodit, ze které proměnné ze dvou zkoumaných můžeme vyvodit informaci o druhé proměnné [28]. Z výsledků zobrazených v matici na obr. 5.8 plyne, že z ID produktu lze vyvodit část informace o kategoriích a expiraci. Zatímco úrovně 1 a 2 o ID produktu mnoho informace nenesou. Jak bylo ukázáno i v předchozích statistikách a jak vyplývá z logiky pro získání dne v týdnu a období měsíce, číslo dne nese informaci o těchto dvou příznacích.

Z vypočítaných statistik na datasetu je patrné, že některé příznaky jsou významně závislé, a proto je třeba je z dat odstranit. Kandidáti na vynechání jsou kategorie 2, 4, 6 a číslo dne. V dalších testech budou také vybráni kandidáti a v závěru vyhodnotím, které příznaky byly podle aplikovaných metod vybrány jako vhodné k vynechání a které nikoli.

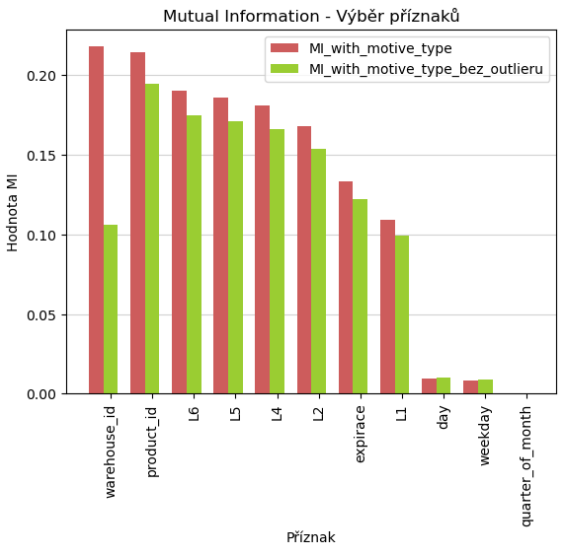
V dalším testu jsem otestovala multikolinearitu dat pomocí rozptylového inflačního faktoru (VIF). Jako hraniční faktor jsem zvolila hodnotu 40 VIF. Vysvětlující proměnné jsem odebírala z datasetu postupně a odebírání jsem ukončila až, když hodnota VIF nebyla nižší než hraniční. Tímto došlo k redukci příznaků z jedenácti na pět, a to na kategorii 1, číslo dne, období měsíce, ID prodejny a den v týdnu. Hodnoty koeficientu VIF na datech jsou na obr. 5.9.



Obrázek 5.8: Matice koeficientů Theilovo U mezi příznaky.



Obrázek 5.9: Rozptylový inflační faktor.

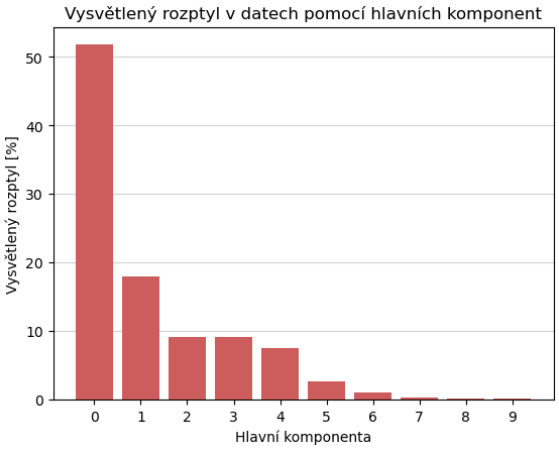


Obrázek 5.10: Koeficienty vzájemné informace mezi příznaky a cílovým sloupcem typ shrinku.

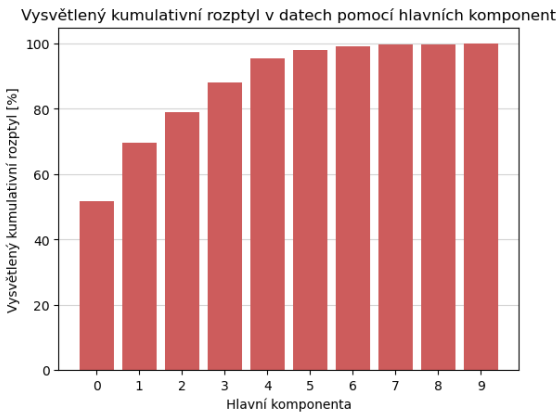
Jako další metodu po výběr příznaků jsem vypočítala hodnotu koeficientů vzájemné informace mezi všemi příznaky s cílovým sloupcem - ID shrinku. Na obrázku 5.10 lze vidět, jak jednotlivé proměnné souvisí s cílovým sloupcem. Pro výpočet tohoto koeficientu jsem použila jak data bez outlierů, tak tenotkrát data před jejich odstraněním. Zde můžeme vidět, že významnost příznaku ID prodejny klesla o téměř polovinu. Nejvíce informace je sdíleno s ID produktu, kategorií 6, dále 5, 4, 2 a expirace. Příznaky související s časovými údaji podle tohoto kritéria nenesou mnoho společné informace.

Jako hlavní metodu pro výběr proměnných jsem se rozhodla použít metodu PCA, tuto metodu je možné použít protože kategorické proměnné jsem převedla na číselné hodnoty v předchozích krocích. Alternativou by bylo použití metody MCA, která se používá pro kategorické datasety, viz dále. Ve své práci jsem využila implementaci PCA v knihovně *Prince* v jazyce Python. Předtím než jsem metodu aplikovala jsem otestovala předpoklad homoskedasticity, tedy shodnost rozptylů v datech, pomocí Bartlettova testu implementovaného v knihovně *factor_analyzer*. Nulová hypotéza o shodnosti rozptylů nebyla vyvrácena (p-hodnota vyšla nulová). Metodu PCA je proto možné použít.

Na obrázcích 5.11 a 5.12 je znázorněno prvních deset komponent a rozptyl který v datech vysvětlují. Na základě hodnot jsem vybrala prvních pět komponent. Již pátá komponenta (označená č. 4) spolu s předchozími vysvětluje více jak 95 % variability dat. V dalším kroku jsem vypočítala příspěvky příznaků k těmto pěti komponentám a vybrala jsem ty příznaky, které přispívají nejvíce k prvním pěti komponentám. Jejich příspěvek je znázorněný na obr. 5.13. Na základě výsledků analýzy hlavních komponent byly vybrány jako vhodné příznaky pro další práci s daty tyto příznaky - ++ ID prodejny, den v týdnu, expirace, den a období v měsíci.

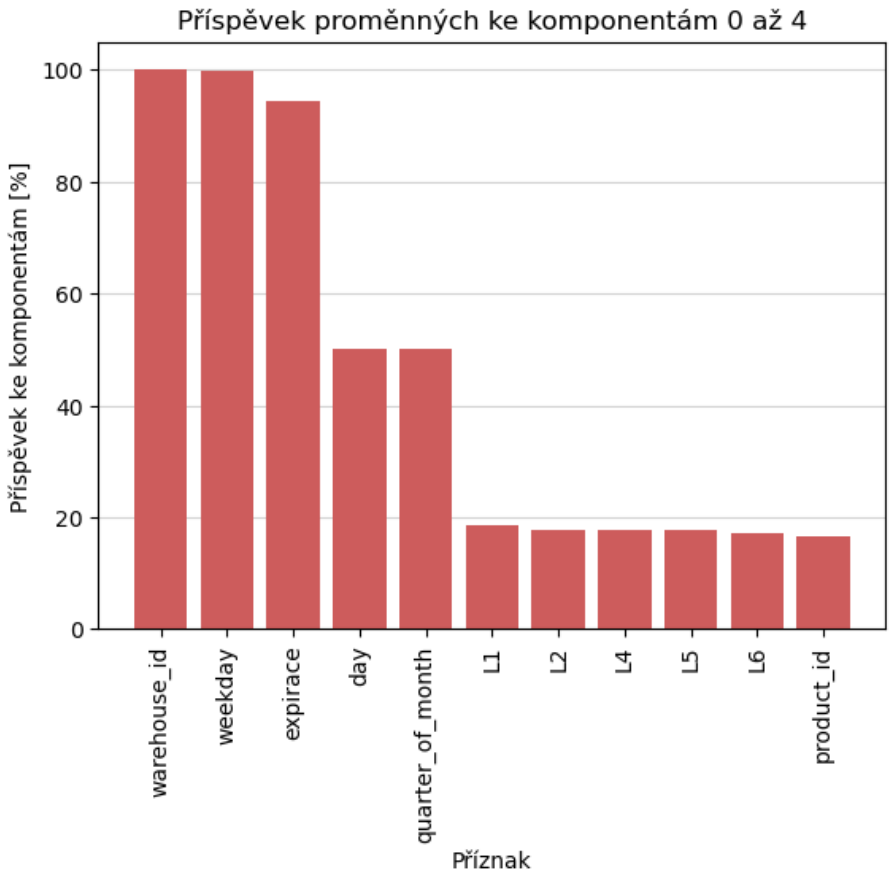


Obrázek 5.11: PCA - vysvětlený rozptyl hlavních komponent.



Obrázek 5.12: PCA - kumulativní vysvětlený rozptyl hlavních komponent.

Jak již bylo zmíněno pro redukci dimenzionality u kategorických dat lze použít metodu MCA, opět jsem využila implementaci z knihovny *Prince*. V této implementaci jsou nominální kategorické hodnoty kódovány tak, že narůstá počet sloupců, a proto bylo nutné, vzhledem k nárokům na paměť k uložení matice, omezit množství dat. Vybrala jsem náhodný 20% vzorek dat, na které jsem MCA aplikovala. Vypočítala



Obrázek 5.13: Příspěvek proměnných ke komponentám 0 až 4.

jsem prvních pět komponent, které dohromady popisují 79 % variability dat. Jelikož byla každá kategorie chápána jako samostatná proměnná příspěvky jednotlivých příznaků ke komponentám byly rozmístěny mezi všechny kategorie, nikoli k jednotlivým příznakům. Po agregaci podle původních příznaků největší příspěvek mělo ID produktu, kategorie 6, 5, 4, zatímco nejmenší ID prodejny, 1, den v týdnu a období měsíce. Tyto výsledky je třeba brát se zvážením neboť výpočty probíhali na řádově menším vzorku než u předchozích metod.

Shrnutí pro výběr dat

Na základě předchozích metod byly původní příznaky datasetu zredukovány na menší počet. Vzhledem k tomu, že různé metody vybraly různé příznaky, bylo stanoveno více možných výběrů.

Korelované jsou hodnoty ID produktu, 6, 5, 4 a expirace. Dále také z označení dne lze určit období měsíce. Ze zmíněných korelovaných příznaků není proto vhodné začlenit více než jeden příznak. Pokud je tato myšlenka aplikována na výsledky metod PCA a MCA a výsledků zjištěných pomocí hraniční hodnoty VIF.

1. Následující sloupce byly získány podle hodnoty rozptylového inflačního fak-

toru. Touto metodou byl navržen i sloupec s číslem dne, ten však z důvodů korelace nebyl zahrnutý

1.1. 1, období měsíce, ID prodejny a den v týdnu.

K této variantě existují i dvě alternativy, ve kterých je obměněna úroveň kategorizace produktu:

1.2. 5, období měsíce, ID prodejny a den v týdnu

1.3. 4, období měsíce, ID prodejny a den v týdnu

2. Metodou PCA bylo zjištěno, které příznaky nejvíce přispívají ke komponentám, které popisují téměř 96 % rozptylu v původních datech - jedná se o příznaky ID prodejny, den v týdnu, expirace, období v měsíci a číslo dne. Naopak metoda MCA vybrala kategorie 4 až 6 jako důležité. Sloučením a přihlédnutím ke korelačním koeficientům byly vybráno pět příznaků

2.1. ID prodejny, den v týdnu, expirace, období v měsíci, 5.

Tato varianta příznaků byla ještě rozšířena o příznaky, které se týkají produktů. Přidané příznaky jsou spolu korelované, přesto

2.2. ID prodejny, den v týdnu, expirace, období v měsíci, 5, 2

2.3. ID prodejny, den v týdnu, expirace, období v měsíci, 5, 2, ID produktu

2.4. ID prodejny, den v týdnu, období v měsíci, 2, ID produktu

2,5, obdobi v mesici, ID prodejny, den v týdnu, ID produktu, target

Všech sedm možných výběrů bylo otestováno metodou gradient boosting. Pro další výpočty byla použita pouze varianta, která vykazovala nejlepší přesnost. Tabulka 5.1 uvádí získané přesnosti.

Tabulka 5.1: Tabulka dosažených přesností dosažených metodou gradient boosting pro varianty výběru příznaků.

Varianta	**Přesnost** [%]	
	Trénovací data	Testovací data
1.1.	79,27	79,15
1.2.	82,88	82,77
1.3.	82,80	82,67
2.1.	83,21	83,07
2.2.	83,38	83,30
2.3.	83,67	83,54
2.4.	83,44	83,33
2.5	83,60	83,45

5.1.2 Klasifikace dat

Tato část se věnuje předpovědi typu shrinku z dostupných dat. V předchozích sekcích bylo popsáno předzpracování dat a výběr vhodných příznaků pro úlohu klasifikace. Byly navrženy dvě skupiny příznaků, na kterých budou provedeny výpočty. K obě variantám se bude přistupovat stejným postupem a následně budou porovnány dosažené výsledky.

Cílový sloupec, který je předpovídán, je pouze jeden. Jedná se o ID shrinku. To obsahuje pět různých kategorií (označené číslicemi od 0 do 4). Proto úlohu můžeme označit jako klasifikační úlohu pro více tříd (neboli *multiclass classification*). Jazyk Python nabízí v knihovně *scikit-learn* řadu metod, které podporují klasifikace do více tříd.[26]

Vybrala jsem následující metody pro klasifikaci ID shrinku:

- logistická regrese OVR,
- multinomická logistická regrese,
- random forest klasifikátor,
- gradient boosting klasifikátor.

Logistickou regresi jsem použila jako základní metodu pro klasifikaci v případě, že vstupní dataset se skládá z kategorických proměnných [30]. Balíček *scikit-learn* umožňuje klasifikaci do více tříd spočítat dvěma způsoby, které se liší v přístupu provedení klasifikace. První přístup využívá schématu OVR (*One-vs-Rest* neboli jeden proti všem). Při použití OVR se každá třída trénuje samostatně. Pro každou třídu je úloha převedena na binární klasifikaci, kdy zkoumaná třída je označena jako pozitivní a všechny zbylé jako negativní. Pokud máme N tříd, pak je vyhodnoceno N binárních logistických regresí. Naproti tomu multinomická log. regrese nevyhodnocuje třídy odděleně, ale používá funkci softmax. Ta predikuje zda, daný bod náleží do jedné z tříd.[31]

Další zvolenou metodu je klasifikátor implementující random forest algoritmus. Tento algoritmus jsem zvolila vzhledem ke skutečnosti, že je úspěšně využíván pro problémy z reálného světa a dovede pracovat s velkým objemem dat, které tyto úlohy obvykle zahrnují [32]. Zároveň volba parametrů pro tuto metodu je intuitivní. Poslední zvolenou metoudou je klasifkátor, který využívá gradient boosting. Tento klasifikátor také vytváří rozhodovací stromy jako random forest. Narozdíl od zmíněného klasifikátoru, jsou ale stromy vytvářeny postupně v závislosti na naposledy vytvořeném stromu. Stromy jsou k sobě agregovány během procesu trénování. Zatímco random forest vytváří stromy nezávisle a agreguje je až na konci procesu.[33]

Výsledky

Nejprve jsem pracovala s vybranými příznaky - 1, období měsíce, ID prodejny a den v týdnu. Data jsem rozdělila na tři skupiny - data pro trénování, validaci a testování v poměru 8:1:1.

Naimplementovala jsem metodu `perform_classification` umožňuje spustit vybraný model s požadovanými parametry z knihovny *scikit-learn*, provede k-fold crossvalidaci, nafituje model na trénovací data a poté ověří přesnost na trénovacích datech. V případě, že jsou předány i parametry pro ladění, na validačních je model doladěn a pak na nejlepších parametrech opět otestován. Celá metoda se nachází v příloze této práce.

```
01 def perform_classification(
02     model,
03     parameters,
04     train_x, train_y, valid_x, valid_y, test_x, test_y,
05     tuning_parameters,
06     k-fold=5)
```

V tabulce 5.2 jsou uvedeny přesnosti pro čtyři vybrané klasifikační metody. Uvedena je jak přesnost na trénovacích datech, tak na testovacích datech. Ve všech metodách byla použita metoda křížové validace, kdy data byla rozdělena do pěti skupin. Výsledná přesnost je pak průměrem dílčích přesností.

Tabulka 5.2: Tabulka dosažených přesností pro čtyři vybrané klasifikační metody na datech se shrinky typu damages s vybranými příznaky podle varianty 1.

Metoda	Přesnost [%]	
	Trénovací data	Testovací data
Logistická regrese OVR	77,04	76,95
Multinomická logistická regrese	77,33	77,27
Random forest	82,54	82,12
Gradient boosting	83,80	83,78

Nejlepších výsledků dosahuje klasifikátor gradient boosting. Přesnost na testovacích datech je téměř 84 %. V obou metodách, které využívají rozhodovací stromy, jsem implementovala ladění parametrů. Na následujících obrázcích **??** je znázorněna závislost mezi jednotlivými hodnotami parametrů a dosažené přesnosti.

V případě random forest klasifikátoru se jedná o parametry, které určují počet stromů, hloubku stromu, minimální počet vzorků, který má obsahovat list a minimální počet vzorků, kdy se může rozdělit uzel stromu. Pro klasifikátor gradient boosting byly také laděny parametry pro počet stromů, hloubku a dále míru učení.

Metoda random forest

Kapitola 6

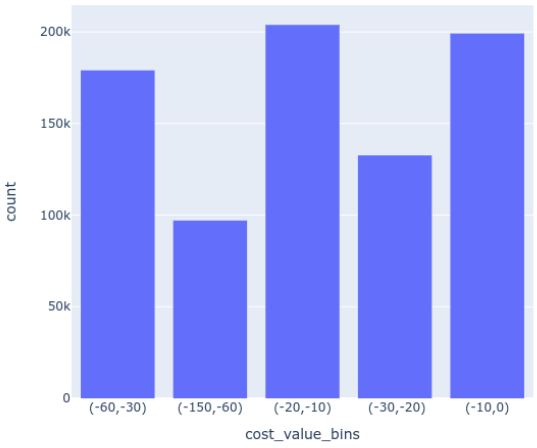
Analýza pomocí metody 4ftMiner

Pomocí metody 4ftMiner, která je jednou z metod procedury GUHA jsem provedla analýzu shrinků produktu. Tato metoda je implemnetovaná v knihovně *Cleverminer* pro jazyk Python. Pracovala jsem pouze se vzorem dat jednoho měsíce a s kategorií produktů *Velmi čerstvé* a *Čerstvé*. Princip metod, které se používají v knihovně, a důležité pojmy týkající se GUHA procedur jsou popsány v sekci 2.1.4.

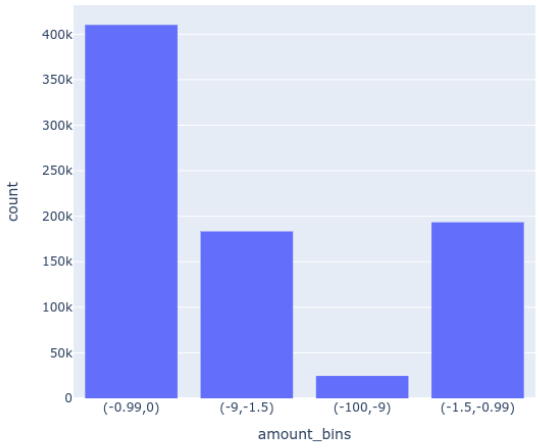
Zkoumaný dataset se záznamy shrinků produktů jsem rozšířila o další sledované sloupce, které dávají do srovnání hodnotu shrinku a objem tržeb. Vytvořila jsem takto sloupce: podíl shrinku na celkových tržbách prodejny, podíl shrinku na tržbách shrinkovaného produktu na prodejně, podíl shrinku a tržeb v kategorii úrovně 1.

Na základě zastoupení jednotlivých typů shrinků, kde prošlé a zkažené zboží zaujímá více než 64 % shrinků, se následující analýzy zaměřují pouze na tento typ.

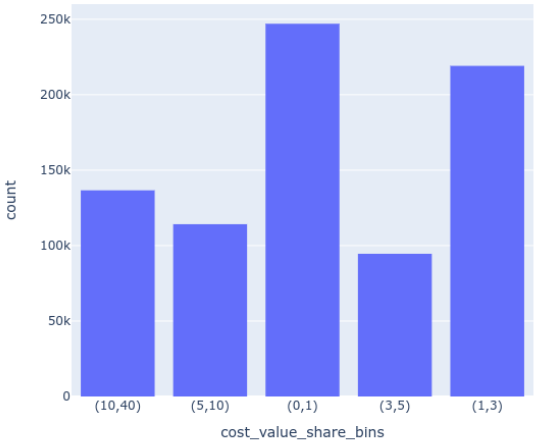
Metoda pracuje pouze s diskrétními hodnotami, proto bylo nutné kategorizovat sloupce s hodnotou shrinku, s množstvím shrinkovaných produktů a s jednotlivými podíly. Na obrázcích 6.1 až 6.5 jsou zobrazené četnosti záznamů v kategoriích.



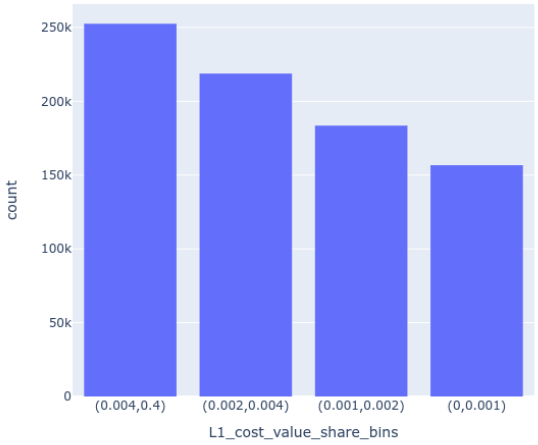
Obrázek 6.1: Histogram pro hodnoty velikosti shrinku v peněžních jednotkách.



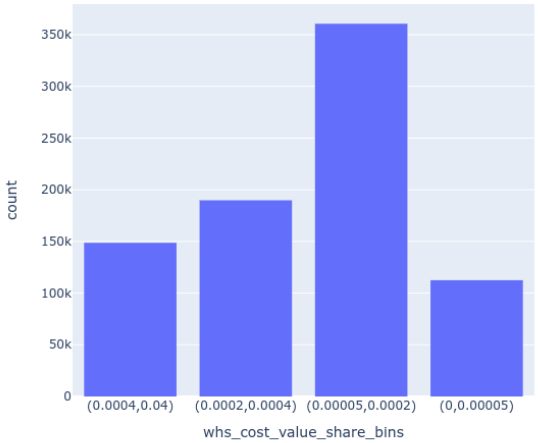
Obrázek 6.2: Histogram pro hodnoty objemu shrinku v kusech.



Obrázek 6.3: Histogram podílu shrinku na tržbách shrinkovaného produktu.



Obrázek 6.4: Histogram podílu shrinku a tržeb v kategorii úrovně 1.



Obrázek 6.5: Histogram podílu shrinku na celkových tržbách prodejny.

6.1 Hypotézy

Data jsem analyzovala metodou *4ftMiner*, které jsem nastavovala parametry podle zkoumané hypotézy.

Hypotéza č. 1: Množství prošlého zboží je závislé na typu výprodeje a dni v týdnu

V datech je zboží bez promoce zastoupeno 58,2 %, zboží v postpromočním výprodeji 23,2, zboží v promoci 18,6 procentem.

Hypotéza č. 2: Na některých lokalitách vyhazují často stejné produkty

Nejčastější shrinkovanou kategorií jsou masné produkty (úroveň hierarchie 3). S větší než 60% pravděpodobností tato kategorie byla zaznamenána u prodejen v okresech: Cheb, Jindřichův Hradec, Kladno, Ústí nad Labem, Nymburk, Písek, Strakonice a Praha-východ. Pokud se vynechá v analýzách tato kategorie, pak se jedná o kategorii pečiva. Ta se s pracděpodobnostyí alespoň 60 % týkala okresu Pardubice a Plzeň-město. Po vynechání je to kategorie zelenina, ovšem už jen s nejvyšší pravděpodobností pod 35 %.

Hypotéza č. 3: Na některých v některých lokalitách mají často zaznamenaný shrink

Hypotéza č. 4: Některé produkty se vyhazují častěji než jiné, ale v malém množství.

Kategorie MEAT PRODUCTS (PROC. MEAT SERVICE)-(masné produkty, šunka, salámy, klobásy) byla zaznamenána téměř 300 tisíckrát, a v 90 procentech se jednalo o množství odpovídající do jednoho balení. Pokud se vyhazují čerstvé ryby (fresh fish), tak v 94 % záznamů je to množství do jednoho kusů. Tapas se vyhazují v 89 % po jednom kusu (obvykle se jedná o sendviče a bagety) Kategorie vejce se vyhazuje v 82 % po jednom kusu balení Kategorie pečiva se vyhazuje v 56 % v počtu kusů do 10 ks v až 94 tis. záznamech. Kategorie CORE FRUIT (hrušky a jablka) se vyhazuje 74 % případech záznamů (14 000 záznamů) v množství do jednoho kusu - váhový přepočet.

Hypotéza č. 5: Některé vyhazované kategorie produktů jsou výrazně nákladnější.

Pokud se vyhazují čerstvé ryby (fresh fish), tak v téměř 80 % případech záznamů jsou ztracené náklady vyšší - 60-150 balení. Pokud se vyhazuje kategorie Red meat, tak z téměř 60 % ve větším množství - 60-150 balení. Kategorie CHILLED PRODUCT SERVICE, která obsahuje čerstvé chlebíčky, saláty a pochutiny na pultovém prodeji, se v 50 % vyhazují v množství

Hypotéza č. 6: Shrink některých produktů je v porovnání s tržbami těchto produktů na stejné prodejně velký.

Nejedná se o porovnání s celkovými tržbami prodejny, ale pouze o týdenní tržbu těch produktů, které měli zaznamenaný v daném týdnu shrink. Tapas se mají share shrinku v 84 % zaznamenaných případech mezi 10-40 %. Cukrářské výrobky mají share shrinku v 74 % zaznamenaných případech mezi 10-40 %. Banány mají share shrinku v 80 % zaznamenaných případech do 1 %. Více než 30 tis. záznamů je u Citrusů, Jablek a hrušek a v okolo 65 % je share shrinku do 1 %.

Pokud mezi produkty, kterým byl zaznamenán dražší (30-60 Kč) shrink, jsou jogurty, tak jejich share shrinku na tržbách je mezi 10-40 %, totéž se týká Tapas. Pokud mezi produkty, kterým byl zaznamenán levný (do 10 Kč) shrink, je ovoce, tak jejich share shrinku na tržbách je do 1 %, totéž se týká in kořenové zeleniny.

Hypotéza č. 7: Kategorie má vliv na zastoupení shrinku na celkových tržbách prodejny v dané kategorii úrovně 1.

S pracděpodobností vyšší než 50 % se toto tvrzení potvrdilo pouze u kategorie Bylinky z úrovně 4. Kdy (0.002,0.005).

Hypotéza č. 8: Den v týdnu nebo čtvrtina měsíce mají vliv na záznamy.

Ve středu, čtvrtek a pátek v poslední čtvrtině měsíce je dvakrát více záznamů než v jiných dnech. Ostatní části měsíce jsou konzistentní.

Kapitola 7

Korelační analýza

Tato kapitola se věnuje popisu korelační analýza pro zjištění důvodu shrinků produktů. V kapitole je popsán postup a jeho implementace v jazyce Python. Analýza se zabývá korelací mezi hodnotou shrinkem a tržbami. Jedno pozorování je na agregované na produkt, prodejnu a den záznamu.

7.1 Postup

V rámci analýzy se porovnávají pouze záznamy produktů, které se vyskytují ve stejné kategorii. Základní hypotéza je, že shrink produktu mohou ovlivnit promoakce jiných produktů v kategorii.

Hodnotu shrinku jsem porovnávala s následujícími ukazateli.

- Tržby daného produktu.
- Tržby daného produktu, které byly v daný den v promoakci - ukázalo se, že takové, až na výjimky nejsou.
- Součet tržeb všech ostatních produktů v kategorii.
- Součet tržeb všech ostatních produktů v kategorii, které byly v daný den v promoakci.
- Součet tržeb všech ostatních produktů v kategorii, které byly v daný den v promoakci nebo byly v rozmezí jednoho týdne po promoakci.

Ke každému ukazateli, jsem ještě vytvořila analogický ukazatel, který uvažoval zpoždění shrinku. V takovém ukazateli, se nebrala hodnota prodeje ze stejného dne, jako byl den záznamu shrinku, ale hodnota z předchozího dne. Důvodem pro vytvoření takových ukazatelů byla hypotéza, že shrink se může projevit až další den po uskutečněných tržbách. Důvodem může být to, že

Na základě korelační analýzy je možné roztřídit produkty v kategorii do pěti skupin:

Kategorie P - Produkty, které si samy způsobují shrink.

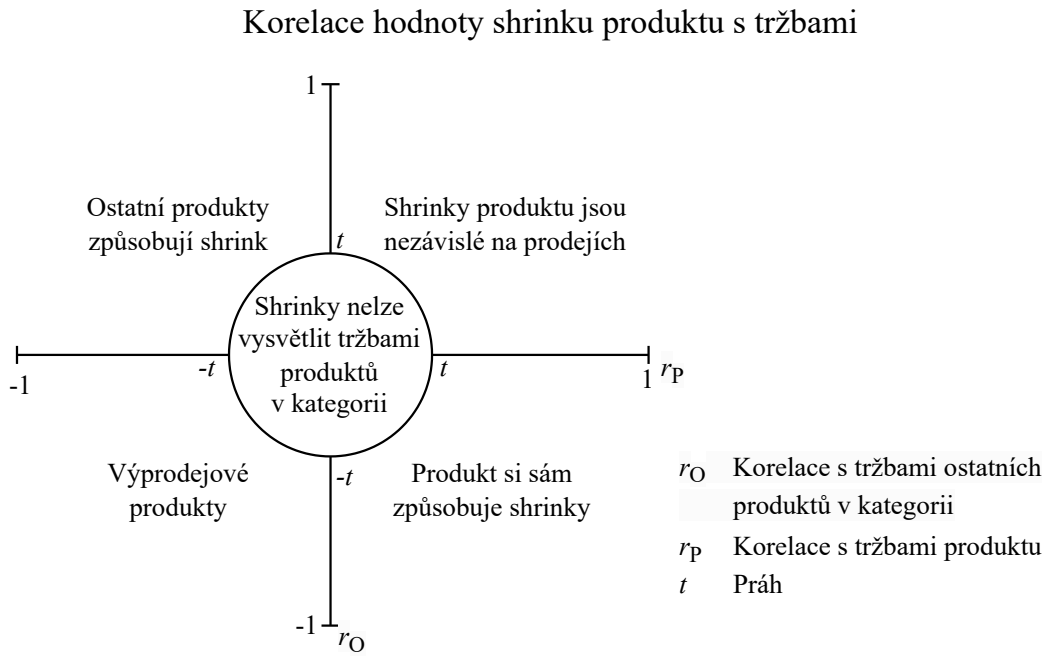
Kategorie O - Produkty, jejichž shrink je způsoben tím, že ostatní produkty v kategorii jsou v promoakci.

Kategorie X - Produkty, jejichž shrink se nepodařilo vysvětlit pomocí korelační analýzy.

Kategorie V - Produkty, které jsou úspěšné ve výprodejích. Produkt se hodně prodává a zároveň má malé shrinky.

Kategorie N - Produkty, jejichž shrink je nezávislý na svých tržbách i tržbách ostatních produktů v kategorii. Shrink zůstává v podobném poměru, tj. pokud jsou tržby vyšší je i shrink vyšší, pokud jsou tržby nižší, je nižší i shrink.

Na obrázku 7.1 je znázorněno rozdělení produktů vzhledem ke korelačnímu koeficientu.



Obrázek 7.1: Kategorizace produktů podle korelace hodnoty shrinku produktu s tržbami.

Hypotéza pro zařazení do kategorie P je následující:

Pokud je korelace zaznamenaného shrinku s tržbami téhož produktu kladná, produkt si způsobuje shrinky sám. Abych mohla tuto hypotézu potvrdit, nebo vyvrátit, je třeba statisticky otestovat významnost korelačního koeficientu. Formulovala jsem nulovou hypotézu H_0 a alternativní hypotézu H_A pro koeficient r_P, který měří korelaci mezi hodnotou shrinku a tržbami produktu.

$$H_0: \quad r_P = 0 \qquad \text{Výběry nejsou korelované.}$$
$$H_A: \quad r_P \neq 0 \qquad \text{Výběry jsou korelované.}$$

Hypotéza pro zařazení do kategorie O je následující:

Pokud jsou kladně korelované hodnoty zaznamenaného shrinku a tržby ostatních produktů a zároveň korelace shrinků produktu s vlastními tržbami je záporná, potom lze vyslovit hypotézu, že shrinky na produktu jsou způsobené ostatními produkty v promoakci. Pro toto tvrzení je opět nutné statisticky otestovat koeficienty korelace. Pro koeficient r_{P} je statistický test stejný jako v předchozím případě. Pro koeficient r_{O} měřící, jak jsou korelované shrinky a tržby ostatních produktů, je třeba otestovat následující hypotézy.

$$\begin{aligned} H_0: &\quad r_{\mathrm{O}} = 0 \qquad \text{Výběry nejsou korelované.} \\ H_{\mathrm{A}}: &\quad r_{\mathrm{O}} \neq 0 \qquad \text{Výběry jsou korelované.} \end{aligned}$$

Pokud na zvolené hladině významnosti zamítneme nulovou hypotézu pro zkoumané korelační koeficienty, můžeme tvrdit že s danou pravděpodobností je koeficient statisticky významný. Na základě hodnoty korelace lze pak produkt zařadit do příslušné kategorie. Produkty, u kterých nelze zamítnout, není možné zařadit do tří uvedených kategorií.

Pro výpočet korelačního koeficientu je ještě třeba ověřit předpoklady. Pro Pearsonův korelační koeficient se jedná o předpoklad normality dat, shodnost rozptylů a nezávislost dat. Pro Spearmanův korelační koeficient není třeba splňovat tyto předpoklady.

7.2 Implementace

V této části je uveden přesný postup pro získání kategorizace produktů. Kód je napsaný v jazyce Python. Součástí kódu je výběr kategorií, které jsou zkoumány, propojení dat shrinků, prodejů a promoakcí, výpočet korelace a ověření předpokladů, statistické testování a rozřazení produktů.

7.2.1 Vstupy a výstupy

Pro korelační analýzu zaznamenaných shrinků s tržbami dalších produktů je třeba zajistit data, které se týkají zaznamenaných prodejů, produktů a prodejen. V následující části jsou popsány tabulková data, která jsou nezbytná pro správné spuštění analýzy. Dále jsou definované i vstupy, které musí definovat uživatel pro specifikování názvů konkrétních sloupců v souborech a parametry pro analýzu.

Celkem je požadováno pět vstupních tabulek - záznamy shrinků, záznamy prodejů, záznamy o promoakcích, číselník produktů s rozdělením produktové hierarchie. Tabulka se zaznamenanými shrinky musí obsahovat sloupec s datem záznamu, ID produktu, ID prodejny, hodnotu zaznamenaného shrinku. Tabulka s prodeji potřebuje stejné sloupce jako tabulka se shrinky s výjimkou že hodnota prodejů je celková prodaná částka, která byla zaznamenaná na dané prodejně v jeden den u daného produktu. Tabulka s údaji o promoakcích by měla obsahovat ID produktu, kterého se promoakce týká, začáteční a koncové datum promoakce a ID prodejny, pro kterou promoakce platí. Všechny záznamové tabulky musí pokrývat stejné časové období.

Období může být libovolně dlouhé. Tabulka produktové hierarchie obsahuje ID produktu, jeho název a libovolně hluboký strom hierarchií. Každá úroveň stromu má vlastní sloupec. Všechny úrovně jsou vyplněné pro každý produkt, tato podmínka je nutná jen pro kategorie, které bude chtít uživatel využít při analýze. Tabulka s hierarchie produktů slouží k tomu, aby mohla být napojena na ostatní tabulky a data se pak mohla vyfiltrovat pouze na záznamy týkající se vybrané kategorie.

Před spuštěním hlavní výpočetní části musí uživatel vypsat konkrétní pojmenování sloupců v tabulce do proměnných. Sloupce, které v různých tabulkách označují tytéž hodnoty, musí mít stejný název. V následujícím kódu 7.1 je ukázka zadání. V komentářích je slovní popis o jaký sloupec se jedná. Sloupec by však měl být jasný přímo z názvu proměnné.

```
product_col      = "product_id"              # Product ID column
product_name_col = "name"                    # Product name
    column
whs_id_col       = "warehouse_id"            # Store ID column
date_col         = "date_of_transaction"     # Date of
    transactions column - for sales and shrinks tables
value_col_shrink = "cost_value"              # Column with value
    of shrinks (shrink table)
value_col_sales  = "cost_value"              # Column with value
    of total sales (sales table)
promo_col_from   = "promotion_date_from"     # Starting date of
    promotion (promotion table)
promo_col_to     = "promotion_date_to"       # Starting date of
    promotion (promotion table)
categories       = ["L3","L4","L5","L6", "name"]  # Categories
    that we want to map to product ID (product hierarchy)
```

Kód 7.1: Definice konkrétních názvů sloupců.

Uživatel dále zadefinuje formát data, který se používá v datumových sloupcích, aby se tyto sloupce mohly převést z textového řetězce na typ `datetime`. V proměnné `category_column` je třeba vybrat jednu kategorii (název sloupce). Na této úrovni se poté budou procházet jednotlivé kategorie, v rámci každé z nich se pak budou porovnávat a třídit produkty. V dalších proměnných může uživatel změnit umístění tj. název složky, kam se ukládají výsledky kategorizace a grafy. Složky s těmito názvy se vytvoří jako podsložky aktuální cesty.

7.2.2 Spuštění analýzy

Analýzu lze spustit pomocí předpřipraveného Jupyter Notebooku v jazyce Python. V první buňce notebooku se načítají potřebné balíčky a modul s definovanými funkcemi pro analýzu.

V dalším buňce jsou definovány vstupní parametry do funkcí - názvy sloupců a úrovně produktové hierarchie. V následující buňce se načítají potřebné datasety. Přehled potřebných vstupů je v sekci 7.2.1. V závislosti na konkrétních datech je třeba specifikovat, jak se mají tabulková data načíst - jedná se např. o parametry pro oddělovač hodnot v řádku, nebo značení desetinné čárky v datech. Pokud nahrané datasety pro prodeje, shrinky a promoakce mají pouze sloupec ID produktu s nenapojenou produktovou hierarchií, je třeba ji připojit.

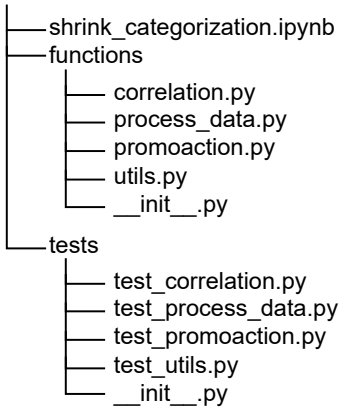
V další buňce se spouští samotná analýza. Nejprve se spustí funkce, která vrátí seznam kategorií, které jsou nejrizikovější. Je třeba definovat na které úrovni hierarchie se budou kategorie prohledávat a také, kolik kategorií budeme chtít prozkoumat. Nalezené kategorie se dále prochází v cyklu.

Z datasetů se vyfiltrují pouze záznamy dané kategorie. Pokud jsou v prodejích záznamy, kde je prodej kladný, tak se tyto záznamy vynechají. V dalším kroku se k údajům o prodejích navážou promoakce. Poté je spuštěna korelační analýza, lze definovat, jaká metoda se má použít s jakou alternativní hypotézou. Případně zda uživatel chce zkoumat shrinky oproti zpožděným prodejům a zda se má analýza zabývat pouze promočními prodeji, nebo i popromočními.

Vypočítané korelační koeficienty se kategorizují a výsledky se uloží do souboru. Zároveň se pro zkoumanou kategorii uloží i graf závislosti shrinků na promočních prodejích.

7.2.3 Popis funkcí a struktura kódu

Kód pro korelační analýzu je umístěn ve složce `shrink_categorization`, struktura složky je vidět na obrázku 7.2.



Obrázek 7.2: Struktura souborů pro kód zpracovávající korelační analýzu.

Funkce jsou rozčleněny do modulů podle toho, na jaký výpočet jsou zaměřené. Každá funkce má je zdokumentovaná pomocí docstring obsaženého ve své definici. Dokumentace funkce se skládá ze stručného popisu, co funkce dělá, jaké má vstupní parametry a jaký je jejich význam a co funkce vrací. Funkce jsou otestované pomocí unit testů.

Pro práci s tabulkovými daty, které jsou hlavním vstupem, jsem použila balíček *pandas* jazyka Python.

V závorce je nástin toho, co budu popisovat (TBD)

- Výběr kategorie (Aby se nemuselo zadávat, přesné názvy kategorií systém vybere prvních n nejsilnějších kategorií z pohledu shrinků.)

- Propojení dat
- Korelace (Pearsonův, Spearmanův korelační koeficient) (ověření předpokladů (Kolmogorov-Smirnov test pro IID), testování statistické významnosti)
- Kategorizace (Nastavení prahu pro velikost korelačního koeficientu)
- Pomocné funkce

Funkce pro přiřazení kategorií k produktům

Jak je uvedeno na začátku sekce 7.2.1, uživatel musí specifikovat názvy sloupců kategorií, které bude v analýze používat. Seznam těchto kategorií je pak parametrem pro funkci `assign_levels`. Další parametry jsou DataFrame, kam se mají kategorie napojit a DataFrame odkud se kategorie napojují. Tyto DataFramy musí mít společný sloupec, podle kterého se napojení provede. Defaultně se jedná o sloupec s ID produktu. Defaulně se provádí *left join*, aby nedošlo ke ztrátě dat, kdyby nějaký produkt neměl v DataFramu kategorií zastoupení. Funkci je také možné předat další argumenty, které se dají volat ve funkci `merge` knihovny *pandas*. V analýze shrinků jeden řádek dat odpovídá transakci jednoho produktu, proto byl zvoleno ID produktu jako propojovací sloupec.

Funkce pro vytipování rizikových kategorií

Funkce `define_risk_categories` vybere prvních n kategorií v dané produktové hierarchii, kde suma hodnot v dané kategorii, je nejvyšší, resp. nejnižší. Funkce vrací seznam těchto kategorií. Prvním vstupní parametrem je DataFrame, který obsahuje minimálně tři sloupce. Tyto sloupce je třeba definovat jako další parametry funkce. Jedná se o sloupec `value_column`, ve kterém jsou hodnoty, které ohodnocují řádky DataFramu a kategorie. Další sloupec je jedna z úrovní produktové hierarchie, ve sloupci se nachází názvy, nebo jiné označení, kategorií. Posledním povinným parametrem je počet kategorií, které má funkce vrátit. Pokud je zadán tento počet tak, že je větší než je počet unikátních kategorií, vrátí se všechny kategorie seřazené od nejrizikovější. Dale je funkci možné předat keyword argumenty, které se předají funkci `sort_values` z knihovny *pandas*. Jedná se např. o parametr pro vzestupné, nebo sestupné řazení. Defaultní řazení je vzestupné, což znamená, že se vezmou kategorie s nejnižší hodnotou. V této analýze sledujeme vyhozené množství, resp. peníze. Tento ukazatel je záporný, tedy vzestupné řazení vybere ty kategorie, jejichž ztráta byla nejvyšší. Vrácený seznam kategorií je tedy seřazen od nejrizikovější kategorie.

Funkce pro výběr pouze dané kategorie ze všech záznamů

Ve funkci `select_category` jsou vstupem DataFrame, název kategorie a úroveň, ve které se daná kategorie nachází. Funkce vrací DataFrame pouze s těmi řádky, kde je

obsažena jmenovaná kategorie. V případě, že tato kategorie v datech není, je vrácen prázdný DataFrame.

Tato funkce je volána ve funkci `process_dataframes`. Vstupy jsou totožné, avšak namísto jednoho DataFramu je možné jich zadat více jako samostatné parametry. Tělo funkce je v kódu 7.2. Funkce vrací seznam všech vstupních DataFramů, a to pouze řádky, které obsahují zadanou kategorii.

```
01 def process_dataframes(category: str, category_column: str, *
   dataframes) -> list[pd.DataFrame]:
02     result_dataframes = []
03 
04     print("Sizes of dataframes: ")
05 
06     for df in dataframes:
07         result_df = select_category(df, category,
   category_column)
08         print("Shape of original df: " , df.shape, "Shape of new
    df: " , result_df.shape)
09         result_dataframes.append(result_df)
10 
11     return result_dataframes
```

Kód 7.2: Funkce pro výběr pouze dané katageorie z více DataFramů.

Funkce pro přiřazení promoakcí

V rámci korelační analýzy bylo potřeba přiřadit k jednotlivým zaznamenaným transakcím, zda byl produkt v den záznamu v promoakci nebo nikoli. V ideálním případě by tento příznak mohl být již uvedený u každého záznamu. Pokud tomu tak, ale není, je nutné data o promocích provázat na základě data, produktu a prodejny podle číselníku promoakcí. Data vybrané společnosti, na jejíchž datech analýza probíhá, nemají promoakce přímo napojené na již proběhlé, zaznamenané transakce. Data o promoakcích jsou uložená v číselníku promoakcí. Ten obsahuje ID produktu, prodejny, začátek a konec promoakce a prioritu promoakce. V jeden den může být více promoakcí, v takovém případě platí ta s nejvyšší prioritou.

Základní funkce pro přiřazování promoakcí k záznamům s transakcemi se nazývá `map_all_promotions`. Tato funkce propojí DataFrame s promoakcemi s druhým DataFramem s transakcemi. Může se jednat jak o záznamy shrinků, tak i o prodeje. Důležité je, že tento DataFrame má sloupec s datem, protože díky datu pak lze identifikovat správnou promoakci. Nalezení správné promoakce je implementováno až ve funkci `promo_`, viz dále v této sekci. Funkce `map_all_promotions` nejprve provede vnitřní spojení (neboli *inner join*) obou vstupních DataFramů podle definovaných sloupců v parametrech funkce. Tyto sloupce jsou vzhledem k datům společnosti - sloupce s ID produktu a ID prodejny. Tím je docíleno toho, že z promoakcí získáme pouze ty záznamy pouze těch produktů, které se prodali, a které zároveň byly v promoakci. Duplicitní záznamy se vynechají. U každého takového záznamu spočítá, kolikrát byl k němu byla přiřazeno promoakce (tj. kolik bylo promoakcí ve sledovaném období pro daný produkt a prodejnu) a ke každému záznamu toto číslo přiřadí. Spolu s číslem se přiřadí i identifikátor promoakce pro dvojici produkt-prodejna (viz tabulka C), tím je určena skupina k sobě patřících záznamů. Takto

označené záznamy se připojí k původnímu DataFramu s transakcemi. Záznamy, kde není žádná promoakce je počet promoakcí roven nule, zbylé hodnoty nejsou definované. Dále se identifikátor upraví tak, že dokáže rozlišit unikátní promoakci na trojici produkt-prodejna-datum (viz tabulka D). V tabulce 7.1 jsou umělá ukázková data, na kterých je znázorněno spojování dat.

Z ukázky a z popsaného postupu plyne, že výsledný DataFrame může mít více řádků než ten původní, ke kterému se přidávali promoakce. V dalším kroku je tedy potřeba určit, která z přiřazených promoakcí probíhala ve stejný čas jako je čas transakce. K tomu jsem vytvořila funkci `label_date_with_promo`. V této funkci je každý řádek promoakce označen jednou ze tří možností: `no promo, promo, after promo`. Tedy zda je datum transakce během promoakce, nebo nikoli, nebo zda je v rozmezí týden po evidované promoakci. Vzniklý příznak byl pojmenován jako typ promoakce. Ve funkci se pracuje pouze se záznamy u nichž byla nalezena alespoň jedna možná promoakce, tj. transakce, kde dvojice produkt-prodejna existuje i v promoakcích. Zbylé řádky tato funkce neoznačuje. V tabulce 7.2 jsou podle těchto pravidel označené jednotlivé řádky[1].

V dalším kroku je třeba vybrat pouze jednu přiřazenou promoakci o to se stará funkce `find_duplicated_records`. Tato funkce vrací seznam indexů řádků DataFramu, které se mohou zahodit. Algoritmus je znázorněný na obr. **??** *TBD: obrázek UML*. Postupně se prochází každý řádek DataFramu. v pomocné proměnné se zaznamenává aktuální identifikátor určující jednoznačnou trojici produkt-prodejna-datum. Nejdříve se do pomocného seznamu nahrají všechny indexy řádků, které mají aktuální identifikátor. Potom se iteruje přes všechny tyto vybrané řádky. Pokud je typ promoakce iterovaného řádku typu `promo`, běh se zastaví a tento řádek se vybere ze skupiny záznamů, uloží se a pokračuje se na další skupinu. Pokud typ promoakce nebyl `promo`, ale `after promo`, tak se vybere tato promoakce, následné kroky jsou analogické předchozímu případu. Pokud nenastala ani jedna z možností zbývá situace, kdy typ promoakce je `no promo`. Až jsou takto prohledané všechny záznamy, na základě seznamu vybraných řádkových indexů se vytvoří seznam indexů ke smazání jako rozdíl všech indexů v DataFramu a indexů s vybranými promoakcemi.

Funkce `match_promo_to_sales` sdružuje dříve popsané funkce, které zpracovávají promoakce. Vstupními parametry funkce jsou DataFramy transakcí a promoakcí a názvy sloupců. Názvy sloupců mají předdefinovanou hodnotu, kterou lze změnit. v dalším volitelném parametru je možné specifikovat formát datumu. Všechny sloupce obsahující datumy se převedou na typ `datetime`. Poté se zavolá funkce `map_all_promotions`, která spojí transakce s promoakcemi. Může vzniknout DataFrame, který má více řádků než původní. Výsledný DataFrame se předá funkci `label_date_with_promo`, kde se označí u napojených promoakcích typ promoakce. Dále se pomocí funkce `find_duplicated_records` vyberou všechny řádky, které obsahují redundantní záznamy. Tyto řádky se odstraní z DataFramu s namapovanými promoakcemi. Ke všem řádkům, ke kterým neexistuje promoakce v číselníku promoakcí, je přiřazen příznak `no promo`. Na závěr funkce zobrazí souhrn o velikostech dílčích DataFramů, aby měl uživatel informaci o počtech duplicitních záznamů. Během

[1] Zbylé sloupce jsou vynechané, protože pro ukázku příznaku nejsou podstatné.

Tabulka 7.1: Umělá data pro znázornění přiřazování promoakcí k transakcím.

Tab. A: Tabulka promoakcí

ID produktu	Prodejna	Začátek promoakce	Konec promoakce
0001	01	2023-03-01	2023-03-05
0001	01	2023-03-15	2023-03-25
0002	02	2023-03-15	2023-03-25
0003	10	2023-03-15	2023-03-25
0004	02	2023-03-15	2023-03-25

Tab. B: Tabulka transakcí

Produkt	Prodejna	Datum transakce
0001	01	2023-03-02
0001	01	2023-03-09
0002	02	2023-03-15
0003	02	2023-03-15
0004	10	2023-03-15
0004	11	2023-03-30

Tab. C: Tabulka souhlasných dvojic promoakce-produkt

Produkt	Prodejna	Začátek promoakce	Konec promoakce	Identifikátor dvojice	Počet promoakcí
0001	01	2023-03-01	2023-03-05	1	2
0001	01	2023-03-15	2023-03-25	1	2
0002	02	2023-03-15	2023-03-25	2	1

Tab. D: Tabulka souhlasných trojic promoakce-produkt-datum

Produkt	Prodejna	Datum transakce	Začátek promoakce	Konec promoakce	Identifikátor trojice	Počet promoakcí
0001	01	2023-03-02	2023-03-01	2023-03-05	1A	2
0001	01	2023-03-02	2023-03-15	2023-03-25	1A	2
0001	01	2023-03-09	2023-03-01	2023-03-05	1B	2
0001	01	2023-03-09	2023-03-15	2023-03-25	1B	2
0002	02	2023-03-15	2023-03-15	2023-03-25	2A	1
0003	10	2023-03-15	NaN	NaN	NaN	0
0004	10	2023-03-15	NaN	NaN	NaN	0
0004	11	2023-03-30	NaN	NaN	NaN	0

Tabulka 7.2: Tabulka transakcí a promoakcí s přidaným příznakem typ promoakce.

Produkt	Prodejna	Datum transakce	Začátek promoakce	Konec promoakce	Typ promoakce
0001	01	2023-03-02	2023-03-01	2023-03-05	promo
0001	01	2023-03-02	2023-03-15	2023-03-25	no promo
0001	01	2023-03-09	2023-03-01	2023-03-05	after promo
0001	01	2023-03-09	2023-03-15	2023-03-25	no promo
0002	02	2023-03-15	2023-03-15	2023-03-25	promo
0003	10	2023-03-15	NaN	NaN	no promo
0004	10	2023-03-15	NaN	NaN	no promo
0004	11	2023-03-30	NaN	NaN	no promo

výpočtu jsou procesy iterování sledovány pomocí knihovny *tqdm*.

Funkce pro korelační analýzu

Funkce `aggregate_sum` je pomocná funkce použitá v kategorizaci produktů. Funkce zagreguje vstupní DataFrame podle uvedených sloupců a sečte hodnoty ve všech numerických sloupcích. Ve výsledném DataFramu resetuje označení řádků a vrátí ho.

Hodnoty korelačních koeficientů se počítají ve funkci `correlation`. Funkci je předán DataFrame a sloupce, kterých se korelace týká. Tato analýza je zaměřena na korelaci hodnoty shrinku s dalšími ukzateli, proto je jedním vstupem název sloupce se shrinky a dalším vstupem je seznam sloupců ostatních ukazatelů. Obecně se nemusí jednat o sloupec shrinků, základní myšlenkou ale je, že korelace je počítána pro každý sloupec ze seznam sloupců s právě tímto jedním shrink sloupcem. Jedním z volitelných parametrů funkce je určení metody pro získání korelačního koeficientu. Implementovány jsou dvě metody Pearsonův korelační koeficient a Spearmanův korelační koeficient. Pro výpočet jsou využité metody z knihovny *scipy*. Těmto metodám lze předat argument, zda se má uvažovat jednostranná nebo oboustranná alternativní hypotéza. Defaultní metodou je Pearsonův korelační koeficient a oboustranná alternativní hypotéza [34].

Před spuštěním výpočtů korelací jsou sloupce testované pro předpoklady IID. Pro testování, zda dva zkoumané sloupce patří do stejného rozdělení byl použitý Kolmogo-

rov-Smirnovův test implementovaný v knihovně *scipy*. Pro nezávislost Ljung-Boxova metoda implementovaná v knihovně *statsmodels*.

Pro každý vypočtený koeficient je spočtena i p-hodnota, díky které lze hodnotu koeficientu označit za statisticky významnou, nebo ne. Pro určení významnosti byla implementována pomocná funkce `significance`. Ta vrací `True`, resp. `False` pro

statisticky významné, resp. nevýznamné výsledky, tedy pokud je p-hodnota menší, resp. větší než α. Předpokládaná hladina významnosti α je 5 %. Výši hladiny lze změnit v parametru funkce pro výpočet korelace, odtud se předá funkci pro určení významnosti. Vypočtené koeficienty a boolovský příznak o jejich významnosti se ukládají do dvou seznamů, které funkce vrací. Oba seznamy mají takový počet hodnot, jaká je délka vstupního seznamu sloupců.

V parametru `days` funkce `correlation` lze specifikovat, zda se má korelace spočítat pouze mezi sloupcem shrinků se všemi sloupci ze seznamu sloupců anebo navíc se všemi sloupci ze seznami, kde jsou ale hodnoty v tomto sloupci posunuté o parametr `days`. Pokud například `days=1`, pak k hodnotě shrinku zaznamenané v jistý den nebude náležet hodnota prodejů v témže dni, ale hodnota ze dne předchozího. Tato volba byla přidána na základě hypotézy, že shrink se může projevit se zpožděním. Pokud jsou data takto posunutá, je třeba nahradit data na začátku sledovaného období.

Funkce `product_sales_correlation` je zastřešující funkcí pro korelační analýzu na datových vstupech. Vstupními daty jsou DataFrame se záznamy shrinků a se záznamy prodejů včetně informace o promoakcích. K tomu je třeba definovat názvy sloupců potřebných pro analýzu. Jedná se o sloupec s hodnotami shrinků, hodnotou prodejů, ID produktů, ID prodejen a daty transakcí. Názvy sloupců mají defaultní hodnotu, kterou je samozřejmě možné změnit podle zkoumaných dat. Dále má funkce volitelný parametr `after_promo`, jehož defaultní hodnota je `False`, který zohledňuje, zda se pro analýzu s promočními prodeji použijí jen prodeje uskutečněné přímo během promoakce nebo i prodeje, které nastaly týden po promoakci. Další parametry jsou volitelné parametry, které se předávají funkcím, které jsou volány v rámci zastřešující funkce (metoda, alternativní hypotéza, hladina významnosti, počet dní posunu).

Funkce vrací tři proměnné. První je DataFrame, který obsahuje seznam produktů a ke každému z nich napočítané korelační koeficienty hodnoty shrinku s ukazateli a statistickou významnost tohoto koeficientu. Déle je vrácen seznam produktů, které neměly žádný promoční prodej ve sledovaném období a případně i produktů, které neměly žádný prodej.

Funkce nejprve vytiskne hlášku, která metoda pro výpočet korelace se použije. Poté se incializují názvy sloupců pro ukládání korelací a příznaku o statistické významnosti. Počet sloupců se liší v závislosti na tom, zda se v analýze zkoumá i varianta se zpožděním shrinku oproti prodejům. Sloupce jsou seřazené tak, aby sloupce týkající se korelace s jedním ukazatelem byly vedle sebe v následujícím pořadí: korelační koeficient, statistická významnost, korelační koeficient se zpožděním, statistická významnost pro koeficient se zpožděním. Takto budou hodnoty uložené ve výsledném DataFramu. Pro všechny ukazatele se čtveřice (v případě zpoždění) nebo dvojice (bez zpoždění), opakuje. Dále se inicializuje prázný DataFrame pro ukládání výsledků s názvem sloupce pro ID produktu spolu s nově vytvořenými názvy.

Dále je třeba ze vstupního DataFramu prodejů vybrat pouze záznamy produktů, které se prodaly během promoakce. Pokud je parametr `after_promo` je `True`, pak se kromě záznamů produktů v promoakci vyberou i ty, kde produkty byly prodány

v rámci týdne po promoakci. Dále se inicializují prázdné seznamy pro uchování produktů, které nemají žádné prodeje, resp. promoční prodeje.

Následně probíhá iterace přes všechny unikátní produkty, pro které byl zaznamenaný shrink. Počet zkoumaných produktů se vytiskne. Na začátku každé iterace je třeba z DataFramů shrinků vybrat pouze záznamy s daným produktem. DataFrame se potom agreguje podle sloupců datum transakce a ID prodejny. Stejný postup se aplikuje pro DataFrame s prodeji. Navíc se obdobný postup aplikuje i na DataFramy s promočními záznamy a se všemi prodeji s tím rozdílem, že se vyhledají záznamy všech produktů kromě iterovaného produktu. Výdledné DataFramy se potom sloučí do jednoho podle sloupců ID prodejny a datumu. Jelikož může nastat situace, že ne všechny hodnoty jsou definované na každém řádku, nahradí se nedefinované hodnoty nulou.

Na složený DataFrame se použije funkce `correlation`, které se předají příslušné parametry. Výsledky se pak vloží jako nový řádek do DataFramu pro ukládání výsledků. Pokud nebylo možné spočíst korelace, z důvodu, že rozptyl hodnot byl nulový - nastane pokud produkt nemá žádné prodeje - nahradíme nedefinovanou korelaci nulou, která indikuje, že mezi veličinami není závislost.

Výsledný DataFrame s korelací je vstupem do funkce `categorization`. Dalšími vstupy je název sloupce, který obsahuje korelačními koeficienty shrinků produktu s jeho vlastními tržbami a sloupce s koeficienty shrinků produktu s prodeji ostatních produktů. Ve funkci se vytvoří nový DataFrame pro uložení výsledků kategorizace. Jeho indexem jsou ID produktů. Samotná kategorizace se získá spuštěním funkce `categorize_products`, která vrací seznam kategorií pro každý řádek vstupního DataFramu. Funkce `categorize_products` roztřídí produkty do pěti kategorií: `itself, other, sellout, independent, none`. V textu se o těchto kategoriích mluví jako o kategoriích P, O, V, N, X. Postup roztřídění produktů do těchto kategorií je popsaný v sekci **??**.

Poté, co má každý produkt přiřazenou kategorii se ve funkci `categorization` označí každý produkt s kategorií, zda je výsledek statisticky reprezentativní, nebo ne. Rozhodující hodnota je získána pomocí funkce `unsignificant_rows`. Která vrací logickou hodnotu výroku:

$$\text{Významnost}(r_i) \vee ((\text{Koeficient(korelace produktu se sebou)} \leq 0)$$

$$\wedge \text{Významnost(korelace produktu s ostatními))}$$

Funkce pak vrátí DataFrame s takto označenými a kategorizovanými produkty. `categorization`, `categorize_products`, `unsignificant_rows`,

Pomocné funkce

Funkce `create_folder` vytvoří složku se zadaným jménem v aktuální cestě, pouze pokud již taková složka neexistuje. Další pomocná funkce je `format_date`, která využívá funkci z knihovny *pandas* `to_datetime`. Pro základní vizualizaci jsem vytvořila funkci, která pomocí knihovny *matplotlib* vytváří bodový graf dvou proměnných. Graf je buď uložen nebo zobrazený při spuštění funkce. Funkci lze předat DataFrame

a názvy dvou sloupců, které reprezentují vstupy pro osy x a y grafu. Další vstupy jsou názvy os a grafu, případně název souboru, pokud uživatel graf uložit.

7.2.4 Testování

Pro testování funkcí jsem použila knihovnu *pytest* jazyka Python. Testy lze spustit příkazem `python -m pytest tests` v kořenovém adresáři projektu.

7.3 Výsledky

Zvolila jsem 5% hladinu významnosti pro testování statistické významnosti koeficientů korelace r_P a r_O.

Nejprve jsem se zaměřila na kategorie z úrovně 4, a to prvních deset kategorií s nejvyšší hodnotou shrinků (tj. s nejvyšší zaznamenanou ztrátou)

V tabulce č. 7.3 jsou porovnání výsledků kategorizace pro kategorii Masné výrobky a Slané pečivo. Měřila jsem postupně korelaci velikosti shrinku s různými ukazateli pro celkové tržby ostatních produktů. Pro určení míry korelace jsem zvolila Spearmanův korelační koeficient, jelikož data nesplňují předpoklady, které jsou nutné pro použití Pearsonova korelačního koeficientu - data nejsou nezávislá a stejně rozdělená. Z uvedených počtů produktů u jednotlivých kategoriích pro různé ukazatele, je patrné, že výsledky se příliš neliší. Pokud bychom se ale zaměřovali na celkové prodeje, nikoli promoční, tak získáváme větší množství produktů, u nichž nebylo možné vysvětlit shrink pomocí korelace.

Masné výrobky - pultový prodej

Shrink byl zaznamenaný u 111 produktů v této kategorii úrovně 4. 96 produktů bylo klasifikováno jako kategorie P, dva jako kategorie O, další dva jako X. U zbylých produktů nebyl koeficient korelace statisticky významný a proto nejde u těchto produktů vyslovit hypotézu pro jejich zařazení.

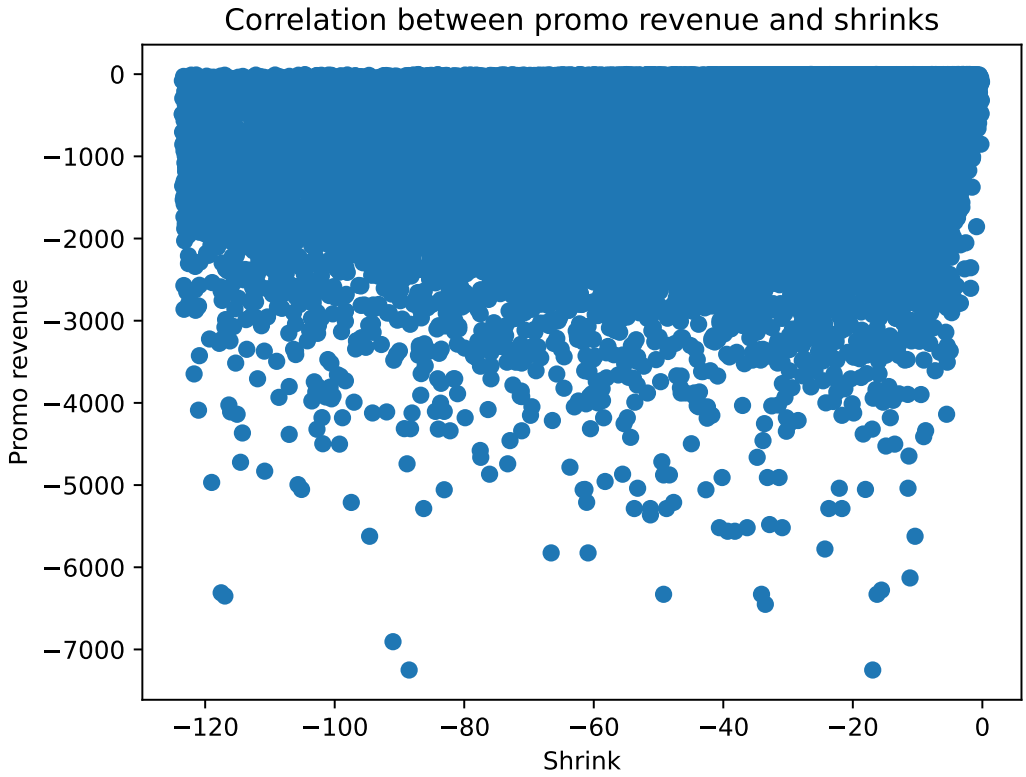
Produkty, které patří do kategorie O: Velikonoční klobása a Velikonoční česnekový šál - jedná se zcela jistě o sezónní výrobky. *TBD: vyjmenovat i další produkty, keré si způsobují samy, ale je to většina produktů a salámy, šunky, z kategorie...*

Dále jsem zkoumala podkategorie Masných výrobků. Porovnávala jsem prodeje v rámci kategorií na šesté úrovni produktové hierarchie. V podkategroii Salámy s krátkou dobou spotřeby se kategorizace potvrdila. Pro kategorii, do níž patří sezónní výrobky - Netučné masné výrobky, nově z této podkategorie byl jako kategorie O označen i produkt Kladenská pečeně.

Slané pečivo

Produkty, které patří do kategorie O: pletýnka adélka, pletýnka malena, pletýnka sypaná mákem, pletýnka sypaná solí a km., bramborové pečivo s cibulí, veka cupeko kb, rohlík grahamový aspec, rohlík pivní ora, rohlík staročeský, rohlík obilný mam, rohlík na strouh.karlova, rohl.n str.penam, houska mašek, houska raženka, bageta

MEAT SERVICE.pdf



Obrázek 7.3: Závislost mezi tržbami produktu a tržbami ostatních produktů v kategorii během promoakce (Masné výrobky - pultový prodej).

Tabulka 7.3: Počet produktů v kategoriích v závislosti na různých ukazatelích. Kategorizace proběhla na základě Spearmanova korelačního koeficientu

	Korelace s tržbami ostatních produktů					
	Všechny prodeje		Prodeje v promoakci		Prodeje v a po promoakci	
Kategorie	Stejný den	Další den	Stejný den	Další den	Stejný den	Další den
Masné výrobky - pultový prodej						
P	96	96	96	96	96	96
O	1	1	2	2	2	2
X	3	3	2	2	2	2
Nevýzn.	11	11	11	11	11	11
Slané pečivo						
P	131	131	131	131	131	131
O	9	13	24	24	24	24
X	16	12	1	1	1	1
Nevýzn.	90	90	90	90	90	90

rust.poh., bageta s grahamem, bageta chlebová, banketka cereální, dalamánek, kostka cereál malena, ciabatta mini natural, twistr se sýrem a špenatem, anglický rohlík,

Sladké pečivo

čokorolka, závin mák , skořicový vrut, donut bílý se sušnkami, závitek cereální nugátový, rohlík lístový s ořech. náp, kobliha vanilková šiška, kobliha vanilková v koš, kobliha s jablky a skořicí, kobliha s lísko.náp.a pol, koblih kapsa s jablky pécé, koláč wellartův, koláč tlač. vícezrnný, koláč šátek makový, koláč s ovocnou nápl, koláček švestkový , koláček meruňkový , koláček borůvkový, koláč rohový tři náplně , koláč s makovou náplní, koláč s tvarohovo nápl, máslový koláč tvaroh, šátek kyn. tvarohová nápl, šáteček s náplní višňovou , šáteček s tvaroh.náplní, šátek makový, loupák o., loupák v., loupák m., závin tvaroh , závin kvásk. makový, bavorská hvězdice, makovka pletená mašek, croissant máslový

Plodová zelenina

Shrink byl zaznamenaný u 28 produktů v této kategorii. 20 produktů bylo klasifikováno jako kategorie P, u zbylých produktů nebyl koeficient korelace statisticky významný a proto nejde u těchto produktů vyslovit hypotézu pro jejich zařazení.

Kapitola 8

Vizualizace dat

Závěr

Cílem práce bylo

Literatura

[1] BAUDIN, Michel. *Lean Logistics: The Nuts and Bolts of Delivering Materials and Goods*. New York: Productivity Press, 2005. ISBN 978-1563272967.

[2] CHRISTOPHER, Martin. *Logistics & Supply Chain Management*. 5th ed. Harlow: Pearson Education Limited, 2016. ISBN 9781292083797.

[3] HASTIE, T., TIBSHIRANI R., FRIEDMAN J. H. *The elements of statistical learning: data mining, inference, and prediction.* 2nd ed. New York: Springer, 2009. Springer series in statistics. ISBN 978-0-387-84857-0.

[4] What is the difference between Logistics and Supply Chain Management. In: *IIM Udaipur Chronicles* [online]. 11. 10. 2019. [cit. 2022-11-07] Dostupné z: `https://www.iimu.ac.in/blog/what-is-the-difference-between-logistics-and-supply-chain-management/`

[5] JIRSÁK, Petr, MERVART, Michal, VINŠ, Marek. *Logistika pro ekonomy – vstupní logistika.* 1. vydání. Praha: Wolters Kluwer ČR, 2012.

[6] JONES, Daniel T., HINES Peter a RICH Nick. Lean logistics. *International Journal of Physical Distribution & Logistics Management*. 1997, **27**(3/4), 153-173. ISSN 0960-0035. Dostupné z: doi:10.1108/09600039710170557

[7] KURITA, Takio. *Principal component analysis (PCA). Computer Vision: A Reference Guide*. 2019, 1-4. [cit. 2022-11-07] Dostupné z: `https://link.springer.com/content/pdf/10.1007/978-3-030-03243-2_649-1.pdf`

[8] Analýza hlavních komponent. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2023-03-04]. Dostupné z: `https://cs.wikipedia.org/wiki/Anal\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{y\global\mathchardef\accent@spacefactor\spacefactor}\accent19y\egroup\spacefactor\accent@spacefactorza_hlavn\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{\OT1\i\global\mathchardef\accent@spacefactor\spacefactor}\accent19\OT1\i\egroup\spacefactor\accent@spacefactorch_komponent`

[9] JAADI, Zakaria. *A Step-by-Step Explanation of Principal Component Analysis (PCA)* [online]. [cit. 2023-03-04]. Dostupné z: `https://builtin.com/data-science/step-step-explanation-principal-component-analysis`

[10] DI FRANCO, Giovanni. *Multiple correspondence analysis: one only or several techniques?*. Quality & Quantity, 2016, 50.3: 1299-1315. [cit. 2023-03-05]. Dostupné z: doi:10.1007/s11135-015-0206-0

[11] ABDI, Hervé, VALENTIN, Dominique. *Multiple correspondence analysis*. In: *Encyclopedia of measurement and statistics*. 2007, 2.4: 651-657. [cit. 2023-03-05]. Dostupné z: `https://personal.utdallas.edu/~Herve/Abdi-MCA2007-pretty.pdf`

[12] NAVARA, Mirko. *Teorie informace.* [online). 3. 1. 2017 [cit. 2023-12-15]. Dostupné z: `https://cmp.felk.cvut.cz/~navara/psi/TI_ebook.pdf`

[13] PŘICHYSTAL, Jan. *Úvod do teorie informace.* [online). 3. 1. 2007 [cit. 2023-12-15]. Dostupné z: `https://akela.mendelu.cz/~jprich/predn/teoinf.pdf`

[14] KROUPA, Tomáš. *Úvod do teorie informace: Matematické základy komprese a digitální komunikace.* [online). [cit. 2023-12-15]. Dostupné z: `https://math.fel.cvut.cz/en/people/gollova/tik/TI_prednasky.pdf`

[15] GREENACRE, Michael. *Correspondence analysis in practice*. chapman and hall/crc, 2017. [cit. 2023-03-05].

[16] Correspondence analysis. In *Wikiwand* [online]. [cit. 2023-03-06]. Dostupné z: `https://www.wikiwand.com/en/Correspondence_analysis`

[17] WRONKA, Anna. LEAN LOGISTICS. *Journal of Positive Management*. 2017, **7**(2), 55-63. ISSN 2392-1412. Dostupné z: doi:10.12775/JPM.2016.012

[18] SUTHERLAND Joel, BENNETT Bob. *The Seven Deadly Wastes of Logistics: Applying Toyota Production System Principles to Create Logistics Value*. Bethelem, PA: Lehigh University, 2007. Dostupné z: `https://www.researchgate.net/publication/265356600`

[19] SKHMOT, Nawras. *The Lean Way Blog: The 8 Wastes of Lean. The Lean Way* [online]. 5. 8. 2017 [cit. 2022-11-17]. Dostupné z: `https://theleanway.net/The-8-Wastes-of-Lean`

[20] SKHMOT, Nawras. *The Lean Way Blog: What is Lean?. The Lean Way* [online]. 5. 8. 2017 [cit. 2022-11-17]. Dostupné z: `https://theleanway.net/what-is-lean`

[21] SKHMOT, Nawras. *The Lean Way Blog: What is Muda, Mura, and Muri?. The Lean Way* [online]. 5. 8. 2017 [cit. 2022-11-17]. Dostupné z: `https://theleanway.net/muda-mura-muri`

[22] Learning the Lingo: 3 definitions related to unsold food inventory. In: *Blog - Spoiler Alert* [online]. 17. 06. 2019. [cit. 2022-02-07] Dostupné z: `https://blog.spoileralert.com/3-definitions-unsold-food-inventory`

[23] HALFORD, M. Prince., *Prince* [online]. [cit. 2023-03-08]. Dostupné z: `https://github.com/MaxHalford/prince`

[24] BAIJAYANTA, Roy., *All about Categorical Variable Encoding* [online]. [cit. 2023-03-13]. Dostupné z: https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02

[25] scikit-learn Machine Learning in Python [online]. 2023 [cit. 2023-03-21]. Dostupné z: https://scikit-learn.org/stable/

[26] Multiclass and multioutput algorithms. scikit [online]. [cit. 2023-04-10]. Dostupné z: https://scikit-learn.org/stable/modules/multiclass.html

[27] BOBBITT, Zach. *How to Interpret Cramer's V* In *Statology* [online]. 2021 [cit. 2023-03-21]. Dostupné z: https://www.statology.org/interpret-cramers-v/

[28] ZYCHLINSKI, Shaked. *The Search for Categorical Correlation* [online]. 2018 [cit. 2023-03-14]. Dostupné z: https://towardsdatascience.com/the-search-for-categorical-correlation-a1cf7f1888c9

[29] HOLČÍK, Jiří, KOMENDA, Martin (eds.) a kol. *Matematická biologie: e-learningová učebnice* [online]. 1. vydání. Brno: Masarykova univerzita, 2015. [cit. 2023-03-14]. ISBN 978-80-210-8095-9.

[30] FROST, Jim. *Choosing the correct type of regression analysis.* [online]. [cit. 2023-04-10]. Dostupné z: https://statisticsbyjim.com/regression/choosing-regression-analysis/

[31] Multiclass logistic regression. In *Refactored* [online]. [cit. 2023-04-10]. Dostupné z: https://refactored.ai/

[32] BIAU, Gérard; SCORNET, Erwan. *A random forest guided tour*. In: *Test* [online]. 2016, 25: 197-227. [cit. 2023-04-10].

[33] LOK, Leon. *Decision trees, random forests and gradient boosting: What's the difference?* [online]. 5. 1. 2022. [cit. 2023-04-10]. Dostupné z: https://leonlok.co.uk/blog/decision-trees-random-forests-gradient-boosting-whats-the-difference/

[34] Scipy.stats.pearsonr. In *scipy.stats.pearsonr - SciPy v1.11.4 Manual* [online]. [cit. 2023-12-10]. Dostupné z: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html

[35] BILÍK, Jan. *Databáze demografických údajů za vybraná města ČR* [online]. 30. 05. 2023 [cit. 2023-07-12]. Dostupné z: https://www.czso.cz/csu/czso/databaze-demografickych-udaju-za-vybrana-mesta-cr

[36] HUBER, Nicholas, MICHAEL, Katina, *Minimizing Product Shrinkage across the Supply Chain using Radio Frequency Identification: a Case Study on a Major Australian Retailer,*. In: *International Conference on the Management of Mobile Business (ICMB 2007)*. Toronto, ON, Canada, 2007, 45-45. [cit. 2023-12-16]. Dostupné z: doi: 10.1109/ICMB.2007.43.

[37] BECK, Adrian. *Moving beyond shrinkage: developing a definition and typology of total retail loss.* Secur J 31, 93–110 (2018). https://doi.org/10.1057/s41284-017-0090-5

[38] RAUCH, Jan, ŠIMŮNEK, Milan *Metoda GUHA a Systém LISp-Miner* [online]. [cit. 2023-12-15]. Dostupné z: `lispminer.vse.cz`

[39] MASA, Petr. *CleverMiner – Beyond apriori.* [online]. [cit. 2023-12-15]. Dostupné z: `www.cleverminer.org`

Příloha A

Obsah přiloženého CD

BP_Gruberova.pdf – soubor s elektronickou verzí této bakalářské práce.

PreprocessData – složka se zdrojovými kódy naimplementovaného balíku PreprocessData.jl.