Datová analýza účtenek e-shopu

Semestrální projekt AK0ZP

Bc. Jiří Valášek

Fakulta Aplikované Informatiky
Univerzita Tomáše Bati ve Zlíně
2023

1 Introduction

V rámci semestrálního projektu jsem se rozhodl pro datovou analýzu účtenek z e-shopu. Pro potřeby toho projektu jsem využil data z kaggle dataset eCommerce purchase history from electronics store. Můj postup pro tento projekt byl:

1. Stáhnout data.
2. Udělat preprocessing:
   - Doplnit prázdná pole pro strojové zpracování.
   - Převést texty na čísla.
   - Odfiltrovat nákupy neregistrovaných.
   - Seřadit podle uživatele a nákupu.
   - Vytvořit Bag-of-Words (BoW) pro uživatele a nakoupené předměty / kategorie předmětů.
   - Vytvořit Term Frequency - Inverse Document Frequency (TF-IDF) tabulku na základě BoW.
3. Udělat vizualizaci dat.
4. Udělat clustering uživatelů na základě TF-IDF.
5. Vytvořit asociační pravidla pro jednotlivé clustery.

V případě zájmu jsou kódy dostupné v rámci projektu na GitHubu.

2 Vstupní data

Vstupní data použita v tomto projektu obsahovala 2 633 521 zakoupených produktů, jednotlivé záznamy obsahovali:

- event_time - čas zakoupení produktu,
- order_id - identifikační číslo objednávky,
- product_id - identifikační číslo produktu,
- category_id - identifikační číslo kategorie,
- category_code - kód kategorie občas chybí,
- brand - značka produktu občas chybí,
- price - cena produktu,
- user_id - identifikační číslo uživatele u neregistrovaných chybí.

event_time	order_id	product_id	category_id
2020-04-24 11:50:39 UTC	2294359932054536986	1515966223509089906	2268105426648170900
2020-04-24 11:50:39 UTC	2294359932054536986	1515966223509089906	2268105426648170900
2020-04-24 14:37:43 UTC	2294444024058086220	2273948319057183658	2268105430162997728
2020-04-24 14:37:43 UTC	2294444024058086220	2273948319057183658	2268105430162997728
2020-04-24 19:16:21 UTC	2294584263154074236	2273948316817424439	2268105471367840086

Tabulka 1: Vstupní data - sloupce 1-4

category_code	brand	price	user_id
electronics.tablet	samsung	162.01	1515915625441993984
electronics.tablet	samsung	162.01	1515915625441993984
electronics.audio.headphone	huawei	77.52	1515915625447879434
electronics.audio.headphone	huawei	77.52	1515915625447879434
	karcher	217.57	1515915625443148002

Tabulka 2: Vstupní data - sloupce 5-8

3 Preprocessing

V rámci předzpracování byly:

- Převedeny časy nákupu položek na UTC timestamp.
- Doplněny prázdné řetězce za chybějící category_code.
- Doplněny prázdné řetězce za chybějící brand.
- Byly odfiltrovány záznamy neregistrovaných uživatelů.
- Záznamy byly seřazeny podle uživatele a objednávky.
- Množství záznamů bylo omezeno na prvních 100 000 pro schůdnost výpočtů s 32 GB RAM.

Z použitých seřazených 100 000 záznamů byly vytvořeny BoW tabulky uživatelů a zakoupených produktů / kategorií zakoupených produktů. První BoW matice obsahovala 32837 řádků uživatelů a 13003 sloupců s počty zakoupených produktů. Druhá BoW matice obsahovala 32819 řádků uživatelů a 690 sloupců s počty zakoupených produktů z jednotlivých kategorií. Rozdílné počty řádků jsou z důvodu odmazání prázdných řádků pro následující výpočty TF-IDF matic.
TF-IDF bylo vypočítáno pomocí následujících vzorců: První byly normalizovány vektory uživatelů pomocí TF

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}},$$

kde $f_{t,d}$ je počet zakoupených produktů t uživatele d. Poté byly vypočteny váhy jednotlivých produktů na základě jejich vypovídající hodnotě přes IDF

$$idf(t,D) = \log 10 \left(\frac{N}{|d \in D : t \in d|} \right),$$

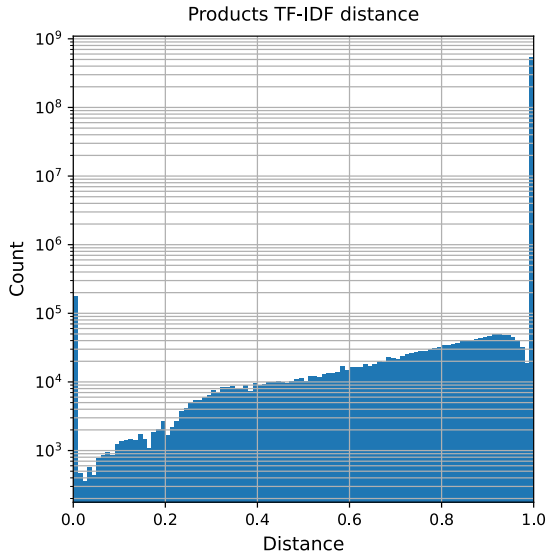
kde N počet uživatelů, $|d \in D : t \in d|$ je počet uživatelů, kteří zakoupili produkt t. Posledním krokem bylo spojení vypočtených TF a IDF do TF-IDF

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D).$$

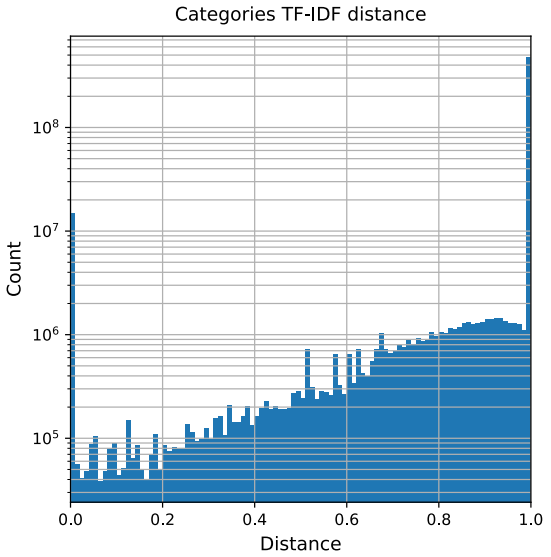
Pro TF-IDF kategorií je t produkt z dané kategorie.

4 Vizualizace

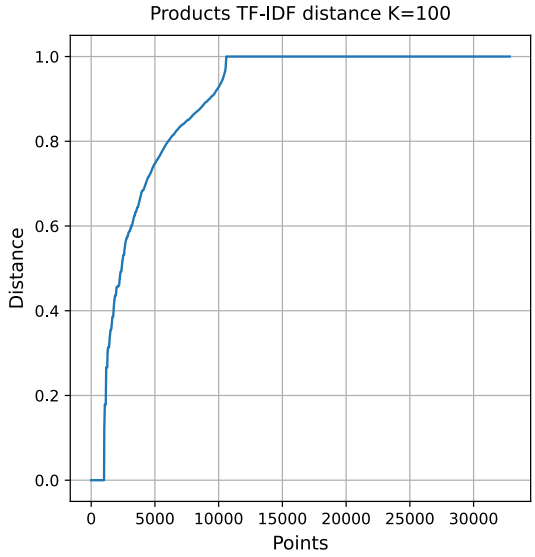
Pro lepší porozumění datasetu byly vytvořeny vizualizace - první byly udělány histogramy vzájemných kosinových vzdáleností mezi všemi vektory zákazníků v TF-IDF maticích.



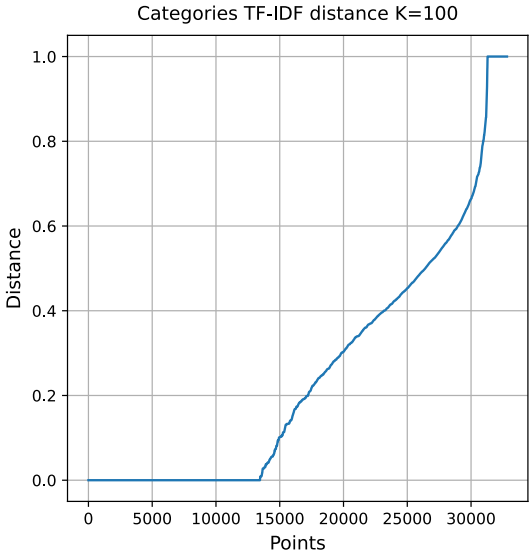
(a) Rozdělení vzdáleností uživatelů na základě nakoupených produktů



(b) Rozdělení vzdáleností uživatelů na základě kategorií nakoupených produktů



(c) Elbow-rule pro počty uživatelů v dané vzdálenosti nakoupených produktů

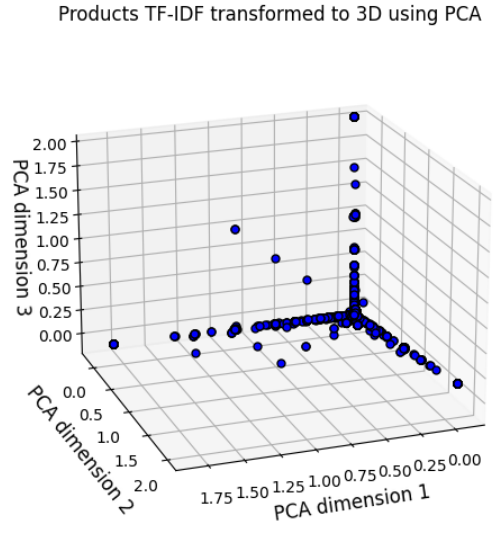


(d) Elbow-rule pro počty uživatelů v dané vzdálenosti kategorií nakoupených produktů

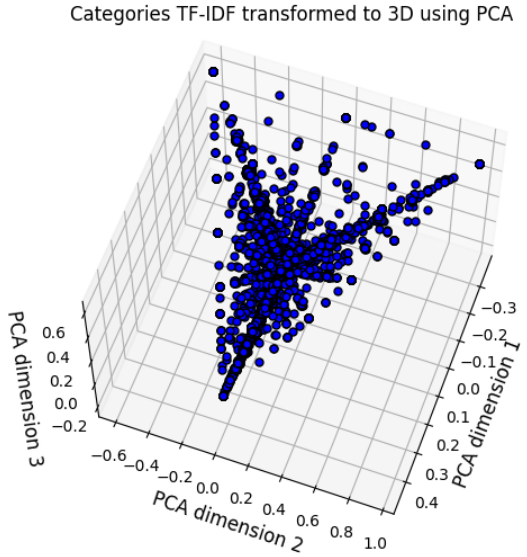
Obrázek 1: Vizualizace dat

Z rozdělení vzájemných vzdáleností vidíme, že máme poměrně polarizovaný dataset - velká část má kosinovou vzdálenost 0 nebo 1. To znamená že hodně uživatelů koupila stejné produkty nebo nekoupila žádný společný produkt. Na pravidle lokte pro $K = 100$ vidíme, že loket je velmi ostrý a odpovídal by hodnotám 0.9 pro produkty a 0.6 pro kategorie. Nicméně toto nastavení vede k vytvoření pouze jednoho velkého shluku a šumu, proto byly použity menší čísla pro *eps* experimentálně získána tak, aby tvořili i shluky o nastavené minimální velikosti.

V rámci vizualizací jsme také využili Principal Component Analysis (PCA) dimenzionální redukci na zobrazení datasetů o 13003 / 690 dimenzích do 3D



(a) Vizualizace datasetu produktů s redukovanou dimenzionalitou pomocí PCA



(b) Vizualizace datasetu kategorií s redukovanou dimenzionalitou pomocí PCA

Obrázek 2: Vizualizace datatasetů

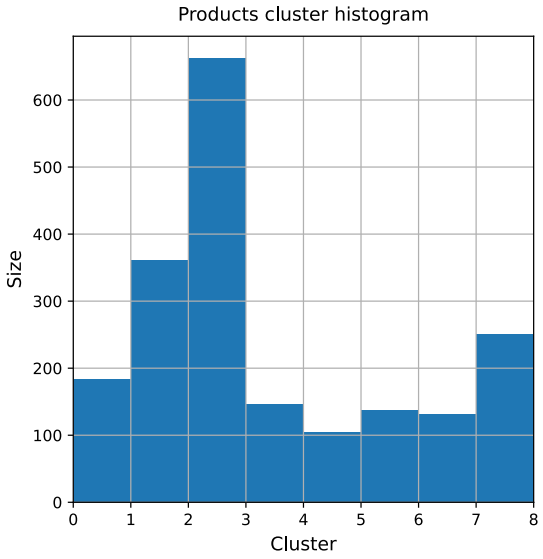
5 Clustering

Pro clustering byl použit algoritmus DBSCAN s nastaveními v tab. 3. Toto nastavení bylo zjištěno experimentálně pro vytvoření shluků uživatelů, které jsou dostatečně velké pro vytěžení netriviálních asociačních pravidel.

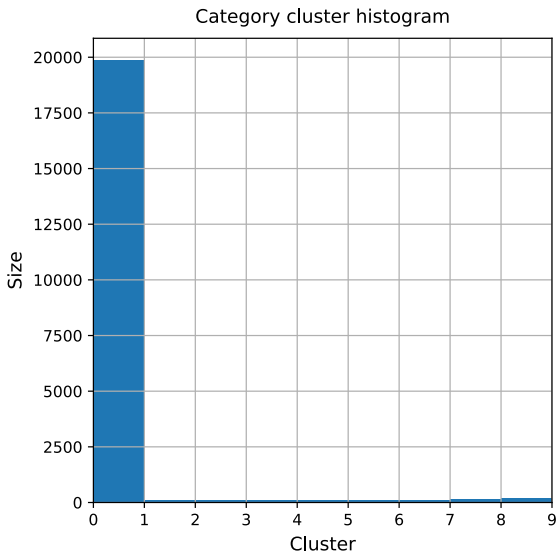
dataset	eps	minimální velikost
produkty	0.3	100
kategorie	0.25	100

Tabulka 3: Nastavení DBSCAN algoritmu

Na základě tohoto nastavení bylo vygenerováno 9 clusterů uživatelů z TF-IDF matice zakoupených produktů s nejmenší velikostí 105 členů. Z matice kategorií uživatelů bylo vygenerováno 10 clusterů, kde nejmenší měl 89 členů. Velikost menší než nastavených 100 je z důvodu, že použitá implementace DBSCAN anotuje hraniční datové body jako náležící pouze k prvnímu shluku u kterého zjištěny. Distribuce shluků je zobrazena na obr. 3. Pro produkty bylo označeno jako šum 30 859 bodů a pro kategorie šum obsahoval 11 964 datových bodů.



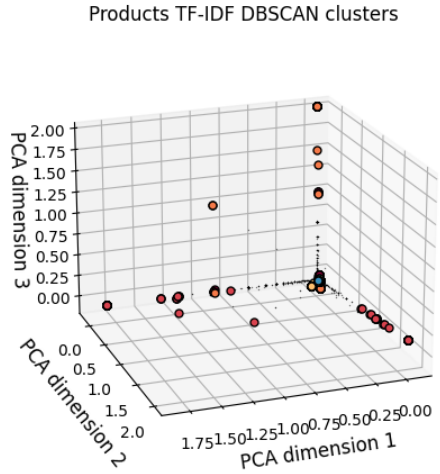
(a) Vizualizace distribuce clusterů z datasetu produktů



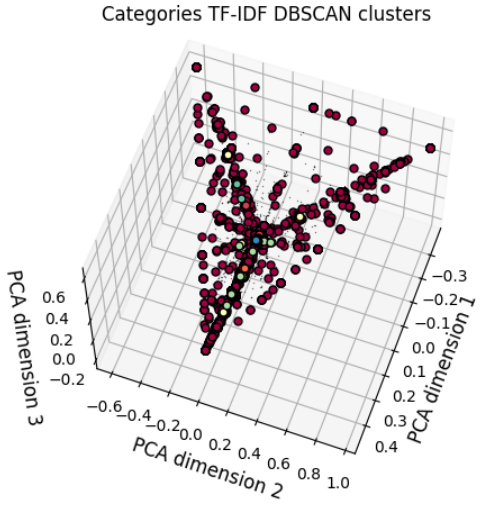
(b) Vizualizace distribuce clusterů z datasetu kategorií

Obrázek 3: Distribuce clusterů

Grafické zobrazení clusterů v prostoru s redukovanou dimenzionalitou pomocí PCA je k dispozici na obr. 4. Zajímavé na tomto zobrazení je, že pro kategorie největší shluk opisoval čtyřstěn a další shluky byly vepsané vrstvy. Toto zobrazení připomínalo techniku vepsaných krychlí pro vyobrazení vyšších dimenzí v 3D prostoru.



(a) Vizualizace clusterů z datasetu produktů s redukovanou dimenzionalitou pomocí PCA



(b) Vizualizace clusterů z datasetu kategorií s redukovanou dimenzionalitou pomocí PCA

Obrázek 4: Vizualizace clusterů

6 Asociační pravidla

Pro generování asociačních pravidel bylo použito následující nastavení apriori algoritmu:

dataset	minimální support	minimální confidence	minimální lift
produkty	0.01	0.5	1.001
kategorie	0.015	0.5	1.001

Tabulka 4: Nastavení DBSCAN algoritmu

Support (podpora) pravidla `IF A THEN B` je:

$$support(\texttt{IF A THEN B}) = support(A \cup B),$$
$$support(A \cup B) = support(X) = \frac{|\{t \in T : X \subseteq t\}|}{|T|},$$

kde A a B jsou množiny produktů / kategorií, t je množina nakoupených produktů / kategorií produktů daného uživatele a T je množina uživatelů. Support udává relativní četnost výskytu všech produktů / kategorií pravidla mezi produkty / kategoriemi uživatelů v clusteru.

Následně confidence $conf(\texttt{IF A THEN B})$ značící důvěru v platnost pravidla je:

$$conf(\texttt{IF A THEN B}) = \frac{support(A \cup B)}{support(A)}$$

Confidence je v rozsahu $\langle 0, 1 \rangle$, kde 1 značí, že antecedent `A` je přítomen vždy a pouze i se sukcedentem `B`.

Posledním použitým parametrem je $lift(\texttt{IF A THEN B})$

$$lift(\texttt{IF A THEN B}) = \frac{support(A \cup B)}{support(A) \cdot support(B)}.$$

V nastavení používáme 1.001 jako $lift(\texttt{IF A THEN B}) > 1$. Lift vypovídá o závislosti antecedenta A a sukcedenta B:

- $lift() < 1$: Položky sukcedenta B jsou zaměnitelné za položky antecedenta A.
- $lift() = 1$: Položky antecedenta A a sukcedenta B jsou vzájemně nezávislé.
- $lift() > 1$: Pozitivní korelace mezi B a A. Takovéto pravidlo je vhodné pro nabízení nových produktů.

Asociační pravidla splňující stanovené parametry nebyly nalezeny pro všechny clustery, zejména z důvodu restrikce na minimální support a lift. Všechny nalezené pravidla jsou k dizpozicí zde jako

⟨typ⟩_cluster_⟨pořadí_clusteru⟩_ruleset_⟨pořadí_pravidel⟩.json, kde

- "items": $A \cup B$
- "support": support(IF A THEN B)
- "rules"
  - "antecedent": A
  - "consequent": B
  - "lift": $lift(\texttt{IF A THEN B})$
  - "confidence": $conf(\texttt{IF A THEN B})$

K dispozici jsou i popisy jednotlivých shluků ⟨typ⟩_cluster_⟨pořadí_clusteru⟩_info.json, kde

- "users": Množina uživatelů v daném clusteru
- "user_categories" / "user_products": Množina množin produktů / kategorií uživatelů (bez duplicit)
- "settings":
  - "min_support": A
  - "min_lift": B
  - "min_confidence": $lift(\texttt{IF A THEN B})$

7 Implementace

V rámci implementace tohoto projektu byly využity knihovny Scikit-learn a Apyori, pro výpočty DBSCAN, PCA a Apriori algoritmu generujícího asociační pravidla.

8 Shrnutí

V rámci tohoto projektu bylo plánováno otestování paralelních výpočtů na Cuda jádrech pomocí knihovny CuPy, nicméně po zprovoznění bylo zjištěno, že implementace funkcionalit knihovny Numpy v CuPy jsou omezeny velikostí paměti grafické karty. Protože byla snaha využít co největší část datasetu, tak bylo od tohoto řešení upuštěno upřednostňujíce Numpy využívajících CPU a 32 GB RAM před CuPy s GPU a 8 GB paměti.

Použitý dataset se vzhledem ke omezeným zdrojům neprokázal jako nejlepší volba, protože při omezení na 100 000 záznamů (limit RAM pro práci s BOW / TF-IDF maticemi) obsahoval neúměrně velké množství unikátních produktů a kategorií. Tato skutečnost způsobila, že BoW / TF-IDF matice byly velmi řídké:

- Počet nenulových položek u BoW produktů byl jen 2.6 ± 3.6 z 13003.
- Počet nenulových položek u BoW kategorií byl jen 2.2 ± 2.5 z 690.

Clustery vytvořené z TF-IDF poté byly okolo centrálních produktů / kategorií např. iPhone / electronics.smartphone, a měly málo položek mimo tu / ty centrální.

Tento fakt komplikoval clustering a posléze i generování pravidel. Při clusteringu bylo složité nastavit *eps* tak, aby cluster měl vice než jednu "centrální"položku a zároveň se všichni uživatele nespojili do jednoho clusteru přes propojující uživatele (rodiče, firmy) kteří koupili více centrálních položek např. iPhone / electronics.smartphone a iphone / computers.notebook. Toto spojení je už v při současném nastavení DBSCAN silně patrné u clusteru 0 u kategorií na obr. 3b a lehce u clusteru 2 na obr. 3a.

U generování pravidel, byl hlavním problémem že shluky měly zpravidla jen desetinu unikátních seznamů produktů / kategorií oproti počtu uživatelů a dominovali zde seznamy s malými počty, což ve výsledku vyžádalo nastavení podpory pravidel na 0.01 / 0.015, aby byly vůbec nějaká pravidla nalezena.

9 Závěr

Během tohoto projektu jsem si ozkoušel práci s DBSCAN a generováním asociačních pravidel pomocí apriori algoritmu. Přestože jsem nepovažoval dataset o 500 Mb za "Big Data", přistupování k datům jako k proudu by byla asi lepší volba, která by umožnila zpracovat všechny dostupné data, ne pouhá 4 %. Alternativou k dolování proudu dat by bylo použití Sparse Matrices, nicméně v pythonu nebyla nalezena knihovna s dostatečnou podporou.