
# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA PODNIKATELSKÁ
FACULTY OF BUSINESS AND MANAGEMENT

## ÚSTAV INFORMATIKY
INSTITUTE OF INFORMATICS


# VYUŽITÍ DOLOVÁNÍ SEKVENČNÍCH VZORŮ V GOOGLE ANALYTICS

USE OF SEQUENTIAL PATTERN MINING IN GOOGLE ANALYTICS

**BAKALÁŘSKÁ PRÁCE**
BACHELOR'S THESIS


**AUTOR PRÁCE** — **Gergő Viskievič**
AUTHOR

**VEDOUCÍ PRÁCE** — **Ing. Jan Luhan, Ph.D., MSc**
SUPERVISOR


**BRNO 2021**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA PODNIKATELSKÁ

# Zadání bakalářské práce

| | |
|---|---|
| Ústav: | Ústav informatiky |
| Student: | **Gergő Viskievič** |
| Studijní program: | Systémové inženýrství a informatika |
| Studijní obor: | Manažerská informatika |
| Vedoucí práce: | **Ing. Jan Luhan, Ph.D., MSc** |
| Akademický rok: | 2020/21 |

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

## Využití dolování sekvenčních vzorů v Google Analytics

**Charakteristika problematiky úkolu:**

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

**Cíle, kterých má být dosaženo:**

Navrhnout podporu analýzy chování uživatelů webových stránek prostřednictvím dolování sekvenčních vzorů v datech Google Analytics 4 se zaměřením na návrh a výběr vhodného algoritmu.

**Základní literární prameny:**

CLIFTON, B. Advanced web metrics with Google Analytics. 3rd ed. Hoboken: Wiley, 2012. 600 p. ISBN 978-1-118-23958-2.

DAVIS, J. J. Google Analytics Demystified: A Hands-On Approach. 3rd ed. CreateSpace Independent Publishing Platform, 2016. 772 p. ISBN 978-1532804311.

LACKO, L. Datové sklady, analýza OLAP a dolování dat. 1 vyd. Brno: Computer Press, 2003. 488 s. ISBN 80-7226-969-0.

LUTZ, M. Learning Python. 5th ed. Sebastopol: O'Reilly Media, 2013. 1648 p. ISBN 978-1-449-35-73-9.

MABROUKEH, N. R. and C. I. EZEIFE. A taxonomy of sequential pattern mining algorithms. ACM Computing Surveys. New York: Association for Computing Machinery, 2010. Vol. 43. Issue 1. Arcticle 3. 41 p. ISSN 0360-0300.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně dne 28.2.2021

L. S.

Mgr. Veronika Novotná, Ph.D.
ředitel

doc. Ing. Vojtěch Bartoš, Ph.D.
děkan

**Abstrakt**

Bakalárska práca sa zameriava na návrh a vývoj algoritmu pre dolovanie sekvenčných vzorov v Google Analytics 4 dátach. Predstavuje a analyzuje dostupné algoritmy pre dolovanie sekvenčných vzorov. Analyzuje dátový model a využitie služby Google Analytics 4. Na základe požiadaviek firemných procesov je navrhnutý algoritmus vhodný definovaným očakávaným vstupným dátam.

**Kľúčové slová**

Google Analytics 4, sekvenčné vzory, Python, dolovanie znalostí z databáz, algoritmus, sekvenčné pravidlá, dolovanie sekvenčných vzorov

**Abstract**

The bachelor thesis focuses on the design and development of an algorithm for sequential pattern mining in Google Analytics 4 data. Presents and analyzes available algorithms for sequential pattern mining. Analyzes the data model and the use of Google Analytics 4. Based on the requirements of business processes, the algorithm is proposed suitable for the expected input data.

**Key words**

Google Analytics 4, sequential patterns, Python, knowledge extraction, algorithm, sequential rules, sequential pattern mining

**Bibliografická citácia**

VISKIEVIČ, Gergő. *Využití dolování sekvenčních vzorů v Google Analytics*. Brno, 2021. Dostupné také z: https://www.vutbr.cz/studenti/zav-prace/detail/135346. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Jan Luhan.

**Čestné prehlásenie**

Prehlasujem, že predložená bakalárska práca je pôvodná a spracoval som ju samostatne. Prehlasujem, že citácia použitých prameňov je úplná, že som v práci neporušil autorské práva (v zmysle Zákona č. 121/2000 Sb., o práve autorskom a o právach súvisiacich s právom autorským).

V Brne dňa 8. mája 2021

……..

podpis autora

**Poďakovanie**

Rád by som sa poďakoval vedúcemu mojej práce Ing. Janovi Luhanovi, Ph.D., MSc za odborné vedenie a cenné rady, ktoré mi poskytol.

Ďalej by som sa rád poďakoval Ing. Pavlovi Šabatkovi za vecné rady a pripomienky, ktoré mal k tejto bakalárskej práci a za vypracovanie oponentúry tejto práce.

Tiež by som sa rád poďakoval rodu House of Řezáč za umožnenie realizovania tejto práce.

V neposlednom rade poďakovanie patrí mojím najbližším, za podporu počas celého obdobia štúdia.

# Obsah

# Úvod

Každá ľudská činnosť - nákup, výber peňazí z bankomatu, návšteva lekára, telefonát, používanie mobilu a webových stránok produkuje dáta, ktoré sú najrôznejšími spôsobmi zaznamenané. V dnešnej dobe platí, že vlastníctvo informácií, schopnosť ich získať, porozumieť a pracovať s nimi znamená konkurenčnú výhodu. O to viac toto tvrdenie platí pre firmy, ktorých oblasť podnikania sa sústreďuje do online prostredia. Klasické databázové systémy slúžia na ukladanie dát, ale nie na extrakciu informácií z nich.

Problematiku extrakcie informácií z rozsiahlych dát rieši oblasť dolovania znalostí z databáz. Tá má za úlohu nájsť skryté a užitočné informácie. Typickým využitím tejto oblasti je analýza nákupného košíka. Takáto analýza prináša znalosti o tom, aké najčastejšie produkty nakupujú zákazníci dohromady. Pokročilejším rozšírením tejto analýzy je využitie sekvenčných vzorov, ktorá zisťuje to, v akom poradí zákazníci tieto produkty nakúpili. Práve táto znalosť je veľmi užitočná pri správnom návrhu úprav a rozvoja webových stránok.

Väčšina firiem používa pre meranie výkonnosti vlastných webových stránok, elektronických obchodov alebo mobilných aplikácií najrozšírenejšiu službu pre meranie - Google Analytics. Google Analytics je silná služba so širokou škálou využití, ktorej databáza môže pri správnom nastavení merania ukrývať množstvo užitočných dát. Napriek tomu je najčastejšie využitie tejto služby ako počítadlo návštev jednotlivých stránok alebo hodnôt transakcií. Analytický tím firmy House of Řezáč s.r.o, s ktorým v spolupráci vznikla táto práca, dlhodobo využíva aj Google Analytics ako plnohodnotný a pokročilý nástroj vo svojich procesoch. Táto práca sa zaoberá návrhom správneho algoritmu pre dolovanie sekvenčných vzorov a využitím v online priestore pomocou služby Google Analytics 4 vo firemnom prostredí.

# Ciele práce, metódy a postupy spracovania

Hlavným cieľom tejto práce je návrh a vývoj vhodného algoritmu pre dolovanie sekvenčných vzorov v dátach Google Analytics 4 s čo najširším možným využitím v procesoch firmy House of Řezáč s.r.o. Pre dosiahnutie cieľa je práca rozdelená na tri hlavné časti, ktoré na seba logicky nadväzujú.

Prvá časť práce je venovaná teoretickým podkladom potrebným k dosiahnutiu ciele práce. V rámci jej je všeobecne predstavený proces získavania znalostí z databáz, jeho komplexné fázy a možné aplikácie. Vzhľadom na riešenú problematiku chovania užívateľov na webových stránkach je predstavená problematika sekvenčných vzorov, ich typy a rozdiely medzi nimi, a rozpoznanie možných sekvenčných vzorov v užívaní webových stránok. Pre vývoj algoritmu na dolovanie sekvenčných vzorov je potrebné sa zoznámiť s doteraz známymi všeobecnými spôsobmi a prístupmi k ich dolovaniu, spoznať ich výhody a nevýhody, a možnosti aplikácie na základe rôznych vstupných dát. Keďže sekvenčné vzory majú popisný charakter, t. j. opisujú časté opakujúce sa vzory v dátach, je predstavená problematika sekvenčných pravidiel, ktoré na základe sekvenčných vzorov sú schopní predikcie. Sekvenčnými pravidlami končí celok o sekvenčných vzoroch. Pre vývoj hocijakého algoritmu je nutné zvoliť si platformu resp. programovací jazyk, v ktorom bude algoritmus implementovaný. Pre dosiahnutie cieľa práce bol zvolený programovací jazyk Python, najbežnejší programovací jazyk pre prácu s dátami, ktorého vlastnosti sú opísané v poslednej časti teoretických východísk.

Druhá časť práce sa zaoberá službou Google Analytics 4. Vzhľadom na to, že práve Google Analytics sú najrozšírenejšia forma merania návštevnosti a chovania sa užívateľov na webových stránkach, a je to široko využívaný nástroj v rámci firmy House of Řezáč, sú predstavené všetky jej možnosti a spôsoby merania chovania užívateľov, podoba a vlastnosti dát , a možnosti prezentácie a extrakcie týchto dát

Posledná časť práce obsahuje vlastný návrh riešenia pre vytvorenie optimálneho algoritmu pre dolovanie sekvenčných vzorov. Hlavnou požiadavkou

algoritmu je, aby mal čo najväčšie využitie v procesoch firmy House of Řezáč a reflektoval aktuálne potreby a požiadavky firmy. Pre získanie týchto informácii bola zvolená metóda ideačného workshopu, ktorého účastníkmi boli kľúčový zamestnanci firmy schopní definovania týchto potrieb a požiadaviek. Priebehu a výstupom ideačného workshopu je venovaný prvý celok vlastného návrhu riešenia. Druhý celok obsahuje vlastný návrh algoritmu pre dolovanie sekvenčných vzorov, opretý o výstupy ideačného workshopu a stávajúce prístupy dolovania sekvenčných vzorov predstavených v teoretických východiskách. Sú prezentované spôsob interpretácie výsledkov a všetky možné implementácie a integrácie s inými nástrojmi nad rámec základného dolovania sekvenčných vzorov na lokálnom počítači.

# 1 Teoretické východiská práce

V tejto časti práce sú opísané všetky potrebné teoretické znalosti k ujasneniu problematiky a pojmov v tejto bakalárskej práci. Začína sa všeobecným prehľadom o oblasti zaoberajúcou sa získavaním a dolovaním znalostí ukrytých v uložených dátach. Tieto znalosti sa potom využívajú na podporu rozhodovania. Proces dolovania znalostí z databáz sa skladá z niekoľkých fáz, môže pracovať so širokou škálou vstupných databáz a je aplikovaný na rôzne typy dolovacích úloh, ktoré sú predstavené v nasledujúcich podkapitolách. Jeden z týchto dolovacích úloh je získavanie sekvenčných vzorov, na ktorý sa zameriava väčšinová časť teoretických východísk. Sekvenčné vzory sú postupnosti udalostí, ktoré sa nachádzajú v dátach často. Na začiatku celku o sekvenčných vzoroch sú predstavené sekvenčné vzory z bežného života v nákupnej oblasti a rôzne využitia v ostatných vedných odboroch. Ďalej sú opísané teoretická definícia sekvenčných vzorov a typy sekvenčných vzorov. Nakoľko sa táto práca zaoberá aplikovaním sekvenčných vzorov v dátach Google Analytics, ktorý je služba využívaná na meranie chovania užívateľov najmä na webových stránkach, sú predstavené typy sekvenčných vzorov a práca s nimi v tejto oblasti. Taktiež sú opísané najznámejšie stávajúce algoritmy pre dolovanie sekvenčných vzorov. Publikácie väčšiny predstavených algoritmov priniesli nové prístupy a spôsoby dolovania týchto vzorov a je potrebné ich spoznať spolu s ich výhodami, nevýhodami a obmedzeniami pred návrhom podobného algoritmu so špecifickým využitím. Podkapitola o stávajúcich algoritmoch pre dolovanie sekvenčných vzorov je ukončená sekvenčnými pravidlami a spôsobom ich získavania. Sekvenčné pravidlá sú konkrétne využitie sekvenčných vzorov, ktoré majú funkciu predikcie týchto vzorov. Pre vývoj algoritmu bol zvolený programovací jazyk Python, ktorého vlastnosti sú opísané v poslednej časti teoretických východísk.

## 1.1 Dolovanie znalostí z databáz

Dolovanie znalostí z databáz je proces získavania skrytých a komplexných informácií z veľkého množstva dát, ktoré môžu byť potenciálne užitočné. Komplexnými informáciami sa rozumejú tie informácie, ktoré nie sú možné

jednoduchým dotazovaním získať alebo vyčítať z dát. Získané znalosti môžu pomáhať pri manažérskom, obchodnom, finančnom alebo strategickom rozhodovaní a plánovaní podniku. Nasledujúce časti čerpajú zo zdrojov [1, 2, 3].

### 1.1.1 Proces dolovania znalostí z databáz

Proces dolovania znalostí z databáz je komplexný proces, ktorý sa skladá z nadväzujúcich častí. Behom týchto procesov sa vstupné dáta čistia, transformujú a pomocou dolovacích algoritmov sa z nich získavajú znalosti. Pre získanie presných a pochopiteľných informácií sa jednotlivé časti procesu môžu opakovať. Celý proces dolovania znalostí vyzerá nasledovne:

1. Selekcia - Príprava a výber relevantných dát. Dáta sú zvyčajne uložené v relačných databázach, preto selekcia zahrňuje výber správnych stĺpcov a dimenzií, ktoré opisujú sledovaný objekt
2. Príprava - Dochádza k odstránení chýb, nekonzistentných hodnôt a zjednoteniu dátových typov. Čím sú kvalitnejšia príprava vstupných dát, tým kvalitnejšie sú získané znalosti
3. Transformácia - Každý algoritmus vyžaduje iný formát dát na prácu. Z toho dôvodu je potrebné vstupné dáta transformovať do formátu, akým je algoritmus schopný pracovať.
4. Dolovanie - Táto časť je najnáročnejšia fáza procesu dolovania znalostí. Pomocou dolovacích algoritmov a úloh sa hľadajú súvislostí v dátach.
5. Interpretácia - V tejto časti dochádza k vyhodnoteniu užitočnosti dolovaných znalostí a k prezentácií výsledkov. Výsledky sa vizualizujú a prezentujú pomocou grafov a tabuliek zrozumiteľnou formou koncovým užívateľom.

### 1.1.2 Typy vstupných dát

Drvivá väčšina zbieraných dát je uložených v rozsiahlych úložiskách. Tieto úložiská môžu byť:

- Relačné databázy - Tieto databázy sú tvorené tabuľky, kde riadky tabuliek sú jednotlivé záznamy a stĺpce opisujú informácie o atribútoch týchto

záznamov. K dátam je možné sa dostať napr. pomocou dotazovacieho jazyka SQL.

- Transakčné databázy - Sú podobné relačným databázam, záznamy sa skladajú z jednoznačných identifikátorov záznamov a zoznamu položiek. Zoznam položiek môže obsahovať zakúpené produkty v obchode.
- Dátové sklady - Zvláštne druhy rozsiahlych relačných databáz, kde dáta sú integrované a uložené v štruktúrach, ktoré podporujú rýchle analýzy a dotazovanie. Práca s dátovými skladmi sa nazýva OLAP analýza.
- Databázy sekvencií - Úložisko obsahujúce postupnosti usporiadaných udalostí, ktoré môžu ale nemusia obsahovať časové záznamy o uskutočnení udalostí. Príkladom sú sekvencie nákupov produktov.
- Multimediálne databázy - Obsahujú textové súbory, obrázky, videá a audio. Týmito databázami sú napríklad databázy produktov v elektronických obchodoch

### 1.1.3 Typy dolovacích algoritmov

Základné typy dolovacích algoritmov, ktoré sa odvíjajú od cieľa dolovania znalostí, sa rozdeľujú na:

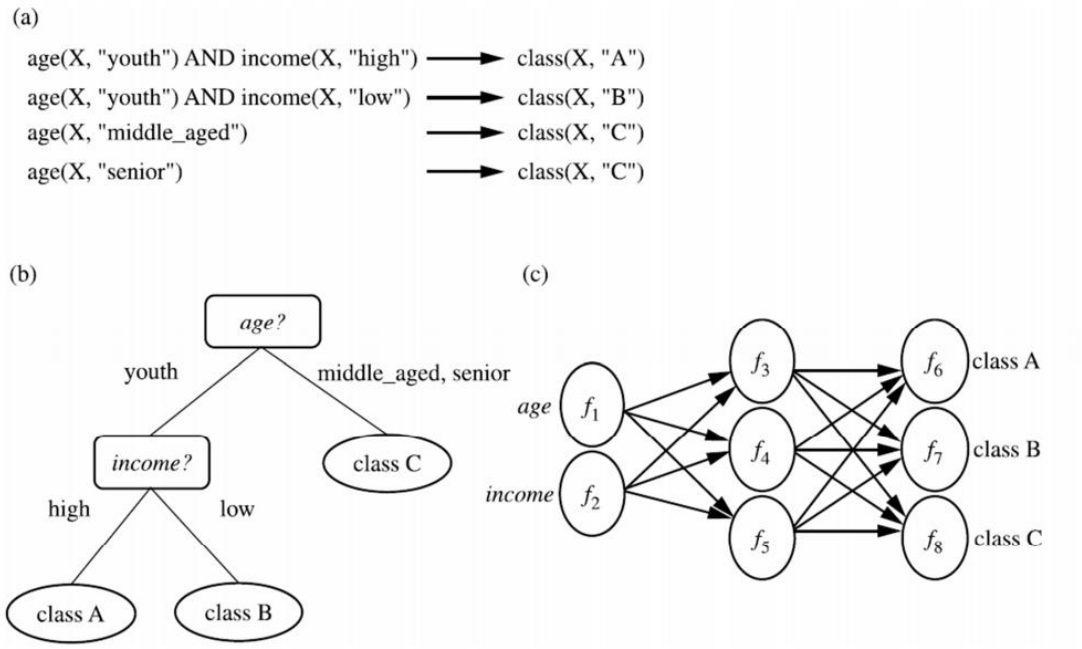
- Deskriptívne - Cieľom je získať všeobecné vlastnosti dát v databáze
- Prediktívne - Na základe súčasných dát sa snažia vytvoriť predpoveď budúcich hodnôt

#### 1.1.3.1 Klasifikácia a predikcia

Klasifikačné metódy majú široké využitie v oblastiach, kde sa zhromažďuje veľké množstvo dát. Sú definované ako klasifikácia objektov (zákazník, pacient) do tried, pričom triedou sa rozumie atribút, ktorý je prítomný u objektoch (plátca/neplátca, zdravý/chorý). Klasifikácia sa používa na predpoveď hodnôt diskrétneho charakteru. Predikcia sa používa na predpoveď hodnôt spojitého charakteru. Výsledný klasifikačný model môže mať rôzne podoby. Medzi najčastejšie patria:

- IF - THEN klasifikačné pravidlá (a) - predpoveď splácania schopnosti zákazníkov úverových bánk

- rozhodovacie stromy (b) - zistenie dôvodu nezmeraných transakcií
- neurónové siete - určenie platu zamestnanca na základe vlastností




**Obrázok č. 1:** Vizualizácia klasifikačných modelov (prevzaté z [21])

### 1.1.3.2 Zhlukovanie

Zhlukovanie alebo segmentácia má za úlohu nájsť objekty, ktoré sú vzájomne podobné, resp. majú podobné atribúty bez znalosti alebo definície týchto parametrov. Umožňuje zhlukovať objekty do skupín na základe vzájomných podobností, ktoré na prvý pohľad nie sú zrejmé.

### 1.1.3.3 Regresia

Regresné úlohy slúžia na vysvetlenie a predpoveď spojitých premenných za pomoci dostupných historických dát. Regresia sa od klasifikácie líši predovšetkým výsledkami. V regresií je výsledkom spojitá číselná hodnota, ktorá zvyčajne odpovedá na otázky typu “Čo sa stane, ak…”.

#### 1.1.3.4 Analýza anomálií

Databáza môže obsahovať záznamy, ktoré sa nepodobajú na chovanie alebo podobu väčšiny dát. Tieto záznamy sa nazývajú anomálie alebo odľahlé objekty. Väčšina dolovacích algoritmov tieto záznamy zahadzuje ako dátový šum. Niektoré úlohy, ako napríklad detekcia neobyčajného chovania alebo podvodov, priamo pracujú s týmito záznamami a považujú ich za zaujímavé. Väčšina algoritmov pre analýzu anomálií vzniká upravením ostatných typov algoritmov. Dobrým príkladom je zhluková analýza, ktorá je schopná tieto záznamy rozpoznať.

#### 1.1.3.5 Asociačné pravidlá

Snahou asociačných pravidiel je získať také vzťahy medzi položkami, že prítomnosť jednej alebo viacero položiek implikuje výskyt iných položiek. Príkladom asociačného pravidla je:

kúpi korčule a kúpi chrániče => kúpi helmu [podpora 10%, spoľahlivosť 50%]

Toto pravidlo označuje, že ak zákazník si kúpi korčule a chrániče, tak si kúpi aj helmu. Podpora ukazuje, v koľkých percentách nákupov si zákazníci kúpili korčule, chrániče a helmu. Spoľahlivosť hovorí, v koľkých percentách skúmaných prípadov bola splnená pravá časť implikácie za predpokladu, že bola splnená aj ľavá časť. 50% spoľahlivosť teda znamená, že v 50% nákupov, kde si zákazníci kúpili korčule a chrániče, si tiež kúpili helmu.

## 1.2 Dolovanie sekvenčných vzorov

Dolovanie sekvenčných vzorov je proces, ktorý hľadá frekventované vzory v databáze sekvencií. Sekvenčná databáza je databáza, ktorá ukladá množstvo záznamov a všetky záznamy sú sekvenciami usporiadaných udalostí, s konkrétnymi záznamami času, kedy sa uskutočnili alebo bez časových záznamov.

Príkladom sekvenčnej databázy sú transakcie obchodných zákazníkov alebo nákupné sekvencie v obchode s potravinami, ukazujúce zbierku produktov, ktoré

každý zákazník kúpil v obchode, každý týždeň po dobu jedného mesiaca. Tieto sekvencie nákupov zákazníkov môžu byť reprezentované následnou schémou:
[číslo transakcie / ID zákazníka, <usporiadaná sekvencia udalostí>], kde každá udalosť je sada obchodných produktov, napríklad chlieb, cukor, čaj, mlieko atď. Konkrétna sekvenčná databáza transakcií pre dvoch zákazníkov môže vyzerať nasledujúco:
[T1, <(chlieb, mlieko), (chlieb, mlieko, cukor), (mlieko), (čaj, cukor)>], [T2, (chlieb), (cukor, čaj)].

Na príklade vidno, že zákazník s číslom transakcie T1 uskutočnil nákup každý týždeň zo štyroch týždňov, druhý zákazník s číslom transakcie T2 tak urobil len počas dvoch týždňov. Zákazník si môže kúpiť jednu alebo viac položiek počas každej návštevy trhu. Záznamy v sekvenčnej databáze tým pádom môžu mať rôzne dĺžky a každá udalosť v sekvencií môže mať vo svojej sade jednu alebo viac položiek.

Algoritmy pre dolovanie sekvenčných vzorov sú zvyčajne aplikované na sekvenčné databázy pre nájdenie opakujúcich sa vzorov, známe ako frekventované sekvencie. Tieto frekventované sekvencie majú široké využitie, napríklad v biológií (sekvencia DNA), priechodoch webovými stránkami, marketingu (ktoré marketingové kampane viedli k uskutočnení konverzie), predikciách a plánovaniach podnikových procesov a mnoho ďalších. Táto práca sa zameriava na sekvenčné vzory v priechodoch webovými stránkami.

### 1.2.1 Definícia sekvenčných vzorov

Táto časť práce čerpá z [4]. Majme sekvenčnú databázu *D* so záznamami usporiadaných *udalostí (sekvencie)*, minimálnu podporu $\varepsilon$, množinu $I = \{i_1, i_2, i_3, \ldots, i_n\}$ všetkých unikátnych *položiek* a sekvenciu $S = <e_1, e_2, e_3, \ldots, e_m>$. *Sekvencia S* je sada usporiadaných *udalostí*, pre ktorú platí, že *udalosť* $e_2$ sa uskutočnila po *udalosti* $e_1$, *udalosť* $e_3$ sa uskutočnila po *udalosti* $e_2$, atď. Problém sekvenčných vzorov znie nasledovne: *Pomocou dolovania sekvenčných vzorov je potrebné nájsť všetky frekventované sekvencie, zložené z položiek I, v danej*

*sekvenčnej databáze D, ktorých frekvencia výskytu v množine sekvencií S je väčšia alebo rovná minimálnej podpore.*

Udalosť $e_m$ sa nazýva prvkom alebo elementom sekvencie S. Každá udalosť $e = (i_1, i_2, i_3, \ldots, i_o)$ predstavuje množinu jednej alebo viacerých položiek, pre ktorú platí, že $e \subseteq I$. Jedna položka môže existovať viackrát v rámci jednej sekvencie ale len raz v rámci jednej udalosti. Počet položiek v jednej sekvencií udáva dĺžku sekvencie $l$. Majme sekvencie $\alpha$= $<a_1, a_2, a_3, \ldots, a_n>$ a $\beta$= $<b_1, b_2, b_3, \ldots, b_m>$. Sekvencia $\alpha$ sa nazýva podsekvenciou $\beta$a naopak sekvencia $\beta$sa nazýva nadsekvenciou $\alpha$ označovanou $\alpha \subseteq \beta$ v prípade, ak existujú celé čísla $1 \leq j_1 \leq j_2 \leq \ldots \leq j_n \leq m$ také, že $a_1 \subseteq b_{j1}, a_2 \subseteq b_{j2}, \ldots, a_n \subseteq b_{jn}$. Napríklad $\beta$ je nadsekvenciou $\alpha$ v tomto prípade: $\alpha$ = <(**a**,**b**), **d**>, $\beta$=<(**a**,**b,**c), (**d**,e)> .

Majme určenú minimálnu podporu $\varepsilon$. Sekvencia $\alpha$ v databáze D so sekvenciami S je frekventovanou sekvenciou vtedy, ak $podpora_S(\alpha) \geq \varepsilon$. Pre frekventovanú sekvenciu sa tiež používa názov sekvenčný vzor. Minimálnu podporu je možné jednoducho interpretovať tak, že je to minimálny počet sekvencií, u ktorých má byť sekvenčný vzor prítomný.

Sekvenčný vzor sa nazýva uzavretým sekvenčným vzorom vtedy, keď v databáze existuje frekventovaná sekvencia $\alpha$ a neexistuje jej nadsekvencia $\beta$s rovnakou podporou. Napríklad majme sekvenčný vzor $\alpha$ = <a, b, d, e> s podporou 10 a sekvenčný vzor $\beta$= <a, b, d, e, h> s podporou 12. Aj keď sekvenčný vzor $\alpha$ je podsekvenciou sekvenčného vzoru $\beta$, obe sekvenčné vzory nemajú rovnakú podporu a tým pádom obe sekvenčné vzory sú uzavreté. V prípade, že by oba sekvenčné vzory mali rovnakú podporu, jednalo by sa o neuzavreté sekvenčné vzory.

Sekvenčným vzorom bez medzier sa nazýva taký sekvenčný vzor, ktorého prvky (udalosti) nastali v priamej nadväznosti jeden na druhý. Naopak, sekvenčným vzorom s medzerami sa nazýva taký sekvenčný vzor, ktorého prvky (udalosti) nenastali v priamej nadväznosti jeden na druhý, len sa uskutočnili za sebou. Majme sekvencie $\alpha$ = <a, b, d, e, d, e> a $\beta$= <a, b, d, e, h, a, d>. Sekvenčný vzor <a, b, d, e> je sekvenčný vzor je medzier, pretože v sekvenciách $\alpha$ a $\beta$ medzi prvkami

neexistujú iné prvky, to znamená, že nastali priamo za sebou. Sekvenčný vzor <a, e> je sekvenčný vzor s medzerami, pretože v sekvenciách $\alpha$ a $\beta$ medzi prvkami existujú iné prvky, to znamená, že nenastali v priamych nadväznostiach. Praktický príklad:
V tomto príklade databázu sekvencií bude tvoriť množina <s_id, s>, kde s_id je označenie sekvencie a s je sekvencia.

**Tabuľka č. 1:** Príklad sekvenčnej databáze (vlastné spracovanie)

| s_id | s |
|---|---|
| 1 | <a,(bc),(dfe),(ba),f> |
| 2 | <(de),(eba),(abc)> |
| 3 | <f,b,(def),a,b> |
| 4 | <e,f,(ab),c,b,c> |

Predpokladajme zadanú minimálnu podporu $\varepsilon$=2. Množina všetkých položiek, ktoré existujú v databáze je I = {a, b, c, d, e, f}. Sekvencia s označením 2 má tri udalosti - (de), (eba), (a,b,c), ktoré nastali v danom poradí. Táto sekvencia obsahuje 8 položiek. Sekvencia <e,(ba),(bc)> je podsekvenciou tejto sekvencie, pretože obsahuje rovnaké položky v rovnakom poradí, ako sekvencia v databáze. Sekvencie s označením 1, 2 a 4 obsahujú sekvenciu s <a,b,c>, splňuje zadanú minimálnu podporu a tým pádom je frekventovanou sekvenciou resp. sekvenčným vzorom. Sekvencia s je rovnako uzavretý, pretože v databáze neexistuje iná sekvencia, ktorá by bola jej nadsekvenciou a mala rovnakú podporu. Je taktiež sekvenciou bez medzier, pretože v databáze položky sekvencie s sa nachádzajú presne za sebou.

### 1.2.2 Sekvenčné vzory na webových stránkach

Aplikácia dolovania sekvenčných vzorov na webových stránkach je zameraná na hľadanie vzorov správania sa na webových stránkach. V prípade sekvenčných vzorov na webových stránkach jednotlivé sekvencie tvoria vždy jednoprvkové udalosti a nie množiny prvkov, pretože sa predpokladá, že v určitom momente je užívateľ schopný uskutočniť len jednu fyzickú udalosť, napríklad zobrazenie

stránky. Sekvenčnou databázou s týmito sekvenciami môže byť napríklad databáza, ktorá sa vytvára pri využívaní služby Google Analytics 4 pre meranie návštevnosti a využívania webu. Najjednoduchším príkladom sekvenčného vzoru, ktorý je všeobecne známy z pohľadu používania webových stránok, je zobrazenie detailu produktu, pridanie produktu do nákupného košíka, prechod objednávkovým procesom a nákup. Praktický príklad:

Majme množinu udalostí E = {a, b, c, d, e, f, g, h} na webovej stránke, ktorá predáva produkty. Jednotlivé udalostí znamenajú nasledovné:

- a: zobrazenie titulnej stránky
- b: zobrazenie zoznamu produktov
- c: zobrazenie detailu produktu A
- d: zobrazenie detailu produktu B
- e: zobrazenie detailu produktu C
- f: pridanie produktu do nákupného košíku
- g: nákup / zobrazenie ďakovacej stránky po nákupe

Ďalej majme nasledujúcu sekvenčnú databázu, kde session_id je označenie návštevy a s sú sekvencie.

**Tabuľka č. 2:** Príklad sekvenčnej databáze na webových stránkach (vlastné spracovanie)

| **sesssion_id** | **s** |
|---|---|
| 1 | <c, d, b, c, e, f, g> |
| 2 | <e, c, e, d, e, a, c, g> |
| 3 | <b, c, d, f, c, e, f, c, g> |
| 4 | <c, d, f, c, e, f, g> |

V tejto databáze vidíme sekvenciu <c, d, c, e, f, g> s podporou 3. Znamená, že 75% užívateľov webu navštívi detail produktu A, potom navštívi detail produktu B, potom znovu navštívi detail produktu A, následne navštívi detail produktu C a tento vzor vedie k pridaniu produktu do košíka a k nákupu. Na základe tohto sekvenčného vzoru s predpokladaným vysokým úspechom je možné spúšťať na stránke produktu

A remarketingové a propagačné reklamy pre ostatné produkty. Toto je jedno z mnoho užitočných využití sekvenčných vzorov v priechode webovými stránkami.

### 1.2.3 Typy sekvenčných vzorov na webových stránkach

V predošlej podkapitole bola predstavená základná problematika a využitie sekvenčných vzorov na webových stránkach. Typy sekvenčných vzorov budú predstavené z pohľadu jednej webovej stránky. Sekvenčné vzory z pohľadu webovej stránky:

**Sekvenčné vzory naprieč celého webu** - Tieto sekvenčné vzory predstavujú sekvencie zobrazenia konkrétnych podstránok webovej stránky. Môžu sa určovať na základe URL adresy stránky alebo na úrovni kategorizácií podstránok. Pri používaní samotných URL podstránok, nastáva problém s viacerými podobami URL adresy z dôvodu možnej existencie parametrov. Tento problém je bežný na webových stránkach, ktoré majú možnosť interného vyhľadávania, kedy sa pridáva vyhľadávaní reťazec do URL adresy alebo v prípade, ak vedú platené reklamy užívateľov na webovú stránky. V tomto prípade tiež je možná prítomnosť parametrov v URL adrese. Výsledky dolovania sekvenčných vzorov môžu byť skreslené a vôbec dolovanie je obťažné, pretože množina unikátnych položiek je niekoľko násobne väčšia ako reálny počet podstránok.

Kategorizácia podstránok je účinný nástroj, ktorý rieši problém existencie viacero verzií URL adresy a bežne používaný vo webovej analytike. Za pomoci dátovej vrstvy, ktorý je prítomný na každej podstránke webovej stránky, sa kategorizuje podstránka podľa typu stránky, napríklad:

- homepage - titulná stránka
- category - zoznam produktov
- searchresult - výsledky vyhľadávania
- product - detail produktu
- checkout - nákupný košík
- purchase - ďakovacia stránka po nákupe
- blog - články

Takáto kategorizácia znižuje veľkosť množiny položiek a zjednoduší proces dolovania sekvenčných vzorov. Tieto sekvenčné vzory môžu byť využité na typické chovanie užívateľov pred nákupom alebo inou konverziou, na cielené reklamy alebo odhalenie navigačných problémov na stránke.

**Sekvenčné vzory na jednej podstránke** - Tieto sekvenčné vzory predstavujú sekvencie udalostí, ktoré nastali na jednej podstránke. Z pohľadu podstránky s detailom produktu udalosti môžu byť:

- interakcia s fotogalériou
- interakcia s videami
- otvorenie sekcie s recenziami
- otvorenie sekcie s dotazmi o produkte
- pridanie produktu do košíku
- scrollovanie do konca stránky
- určitý čas strávený na stránke
- pridanie produktu do obľúbených
- zdieľanie produktu

Tieto sekvenčné vzory môžu byť využité na zistenie typického chovania užívateľov pred pridaním produktov do košíku, pred samotným nákupom alebo na odhalenie slabých a nefunkčných oblastí alebo funkcionalít podstránky.

**Sekvenčné vzory - kombinácia predošlých dvoch typov** - Tieto sekvencie zahrňujú v sebe všetky aktivity a udalosti, ktoré užívateľ previedol počas návštevy na webovej stránke. Výsledkom sú dlhé sekvencie s niekoľkými desiatkami prvkov a veľká množina unikátnych položiek. U týchto sekvencií nastáva problém správneho označenia udalostí, resp. konkrétnosť udalostí.

Druhé rozdelenie sekvenčných vzorov na webe je podľa návštevy užívateľov: **Sekvenčné vzory počas jednej návštevy** - Jedna návšteva je čas strávený od príchodu po odchod z webovej stránky. Tieto vzory môžu byť využité na odhalenie správania užívateľov pred uskutočnení konverzií.

**Sekvenčné vzory naprieč návštev** - Agregované sekvencie jednotlivých návštev, ktoré nastali v určitom období. Tieto vzory môžu byť využité sa pochopenie a hlbšiu

analýzu správania sa užívateľov v dlhšom čase a vzťah k webovej stránke alebo značke.

### 1.2.4 Typy algoritmov pre dolovanie sekvenčných vzorov

Algoritmy pre dolovanie sekvenčných vzorov je možné rozdeliť do nasledujúcich typov [1]:

1. Apriori algoritmy
2. Algoritmy založené na BFS (Breadth First Search)
3. Algoritmy založené na DFS (Depth First Search)
4. Algoritmy založené na uzavretých sekvenčných vzoroch
5. Inkrementálne algoritmy

#### 1.2.4.1 Apriori algoritmy

Prvé predstavenie Apriori algoritmu bolo v [5] s názvom AprioriAll na transakčnej databáze so sekvenciami nákupov zákazníkov. Transakčná databáza sa skladá z troch atribútov: ID zákazníka, času realizácie transakcie a nakúpeného produktu. Databáza je v tomto prípade reprezentovaná v horizontálnej podobe. To znamená, že sekvencie sa skladajú z usporiadaných zoznamov udalostí (prvkov). Proces dolovania sekvenčných vzorov má 5 krokov:

1. Triedenie: Roztriedenie transakčnej databáze podľa ID zákazníkov.
2. L - množina položiek: Získanie všetkých jedinečných položiek z triedenej databázy na základe minimálnej podpory.
3. Transformácia: Nahradenie sekvencií nákupov množinami položiek, ktoré obsahujú. Pre efektívne dolovanie sú všetky množiny položiek mapované do celočíselných radov. Následne sa pôvodná databáza transformuje na množinu sekvencií tvorených položkami z množiny.
4. Sekvencia: V tomto kroku sa generujú z transformovanej sekvenčnej databáze všetky frekventované sekvenčné vzory.
5. Posledný krok: Vylúčenie sekvenčných vzorov, ktoré majú príslušné nadsekvencie.

Napriek tomu, že Apriori algoritmy sú základom väčšiny ostatných algoritmov pre dolovanie sekvenčných vzorov, nie sú dostatočne efektívne. Autori práce na

základe algoritmu pomenovali vlastnosť, ktorá sa nazýva Apriori vlastnosť a znie nasledovne: *Všetky podsekvencie frekventovanej sekvencie musia byť tiež frekventované a naopak sekvencia nie je frekventovaná v prípade, že obsahuje nefrekventovanú podsekvenciu [5].* Táto vlastnosť hovorí, že získanie frekventovaných množín položiek je možné získaním frekventovaných jednoprvkových množín, následne pomocou frekventovaných jednoprvkových množín získať frekventované dvojprvkové množiny. Tento proces iteruje dovtedy, kým je možné získať najvyššie k-prvkové množiny. Algoritmy založené na tejto vlastnosti generujú veľké množstvo kandidátnych sekvencií a porovnávajú tieto kandidátne sekvencie s každou sekvenciou v databáze, čím vyžadujú viacnásobný prechod databázou. Z tohto dôvodu nie sú dostatočne efektívne pri práci s databázami s veľkými množstvami sekvencií.

#### 1.2.4.2 Algoritmy založené na BFS (Breadth First Search)

Tieto algoritmy sú založené na Apriori vlastnosti ale s tým rozdielom, že všetky k-prvkové kandidátne sekvencie sú vytvárané v k-iterácií prechodu vstupnou databázou. Bolo vyvinutých niekoľko algoritmov založené na BFS, najvýznamnejší z nich je predstavený v nasledujúcej časti.

##### 1.2.4.2.1 GSP algoritmus

Tento algoritmus vykonáva rovnakú prácu ako AprioriAll algoritmus, ale nevyžaduje nájdenie všetkých frekventovaných množín položiek ako prvý krok [6]. Algoritmus vykonáva veľký počet prechodov databázou nasledovne:

1. pri prvom prechode získa všetky frekventované jednoprvkové množiny
2. každým ďalším prechodom sa generujú a porovnávajú kandidátne sekvencie so sekvenciami v databáze, aby sa aktualizoval počet výskytov kandidátnych sekvencií v sekvenciách databázy.
3. Algoritmus končí  vtedy, ak v prechode neboli nájdené žiadne nové sekvenčné vzory alebo ak už nemôžu byť generované nové kandidátne sekvencie.

Pseudokód GSP algoritmu vyzerá nasledovne:

***Vstup:***

*Databáza sekvencií;*

*Minimálna podpora;*

***Výstup:***

*Množina frekventovaných sekvenčných vzorov;*

***Metóda:***

*Nájsť $F_1$ (množina jednoprvkových sekvenčných vzorov) po prechode databázou;*

*k = 1;*

*While $F_k$!= null do*

*Vygeneruj $C_{k+1}$(množinu k+1-prvkových kandidátnych sekvencií) z $F_k$;*

*If $C_{k+1}$!=null then*

*Prechod databázou, inkrementuj podporu kandidátov $C_{k+1}$, ktoré sú v databáze;*

*$F_{k+1}$= kandidáti v $C_{k+1}$, ktoré spĺňajú minimálnu podporu;*

*k = k + 1;*

*End;*

*End While;*

*Return $F_k$;*

Nevýhody tohto algoritmu sú:

1. generovanie veľkého počtu množín s kandidátnymi sekvenciami, ktoré vyžaduje viacnásobný prechod databázou
2. veľký počet vygenerovaných krátkych kandidátnych sekvencií pre získanie konečného sekvenčného vzoru - pre túto skutočnosť je tento algoritmus neefektívny pri dolovaní dlhých sekvencií

#### 1.2.4.3 Algoritmy založené na DFS (Depth First Search)

Tieto algoritmy sa snažia eliminovať proces viacnásobných prechodov vstupnou databázou a eliminovať generovanie veľkých počtov kandidátnych sekvencii. Na to využívajú nasledujúcu techniku [4]:

1. Vertikálna reprezentácia databázy - databáza je reprezentovaná množinou prvkov a každý prvok si uchováva informáciu o tom, v ktorých udalostiach (sekvenciách) existujú a v ktorých nie. Výpočet podpory kandidátnych sekvencií prebieha pomocou rýchlych operácií, napríklad logický prienik, bitový súčin atď.
2. Metódu vzrastu vzoru - Rieši problém neefektívneho generovanie veľkého počtu kandidátnych sekvencií a ich testovaním. Vstupná sekvenčná databáza je v procese dolovania postupne rozdelená na menšie časti a v týchto menších databázach sa hľadajú frekventované položky. Tieto položky sú následne spojené s prefixom, čím dochádza k postupnému vzrastu vzorov

##### 1.2.4.3.1 PrefixSpan algoritmus

PrefixSpan algoritmus predstavený v [7] je algoritmus, ktorý využíva projektované databázy a metódu vzrastu vzoru. Algoritmus PrefixSpan pomocou prefixov rozdeľuje vstupnú databázu do viacerých projektovaných databáz, vďaka čomu dochádza k zmenšeniu dolovaného prostredia a zrýchleniu procesu. Postupnými iteráciami v týchto rozdelených databázach hľadá frekventované položky, ktoré využíva na vytvorenie sekvenčných vzorov a projektovanie databáze. Pseudokód PrefixSpan algoritmu vyzerá nasledovne:

***Vstup:***

*Databáza sekvencií;*

*Minimálna podpora;*

***Výstup:***

*Množina frekventovaných sekvenčných vzorov;*

***Metóda:***

***function*** *PrefixSpan(α, i, S|α):*

*α - sekvenčný vzor (prefix);*

*i - dĺžka α;*

*S|α - α -projektovaná databáza, ak α != <>, inak je to Databáza sekvencií;*

*Priechod S|α raz, nájdi množinu všetkých frekventovaných prvkov b, kedy:*

1. *b je možné pripojiť k poslednému prvku poslednej udalosti α a vytvoriť tak sekvenčný vzor alebo*
2. *b je možné pripojiť za poslednú udalosť α a vytvoriť tak sekvenčný vzor;*

*Každý frekventovaný prvok b pripoj k prefixu α a vytvor tak sekvenčný vzorr α′, ulož ho;*

*Pre každý sekvenčný vzor α′ vytvor α′ - projektovanú databázu S|α′;*

*PrefixSpan(α′,i+1,S|α′);*

***End;***

*PrefixSpan(<>, Databáza sekvencií);*

Z uvedeného pseudokódu algoritmu vidno, že sa nevytvárajú veľké počty kandidátnych sekvencií. Naopak sa vytvára veľký počet projektovaných databáz, čo je hlavným výkonnostným problémom algoritmu. V porovnaní s GSP algoritmom je ale podstatne rýchlejší [4].

#### 1.2.4.3.2 SPAM algoritmus

Algoritmus predstavený v [7], využíva generovanie a testovanie kandidátnych sekvencií rovnako ako GSP algoritmus, vertikálnu reprezentáciu sekvenčnej databázy pomocou bitmapov, pomocou ktorej ukladá jednotlivé sekvencie, lexikografický sekvenčný strom a metódu hľadania sekvenčných vzorov do hĺbky.

Lexikografický sekvenčný strom predpokladá určité usporiadanie na množine prvkov. Ak prvok *a* sa v danom usporiadaní nachádza pred výskytom položky *b*,

ich vzťah je možné vyjadriť ako $a \leq b$. To isté platí pre sekvencie: ak je sekvencia $s_a$podsekvenciou $s_b$, tak v usporiadaní $s_a$sa nachádza pred $s_b$ $s_a \leq s_b$. Nech lexikografický sekvenčný strom má označenie T, koreň stromu označenie <>, uzol stromu *n*. Všetky poduzly uzlu *n* sú $n'$také, pre ktoré platí, že $n \leq n'$a $\forall m \in T: n' \leq m \Rightarrow n \leq m$ [5]. Behom prechodu stromom sú z každého uzlu generované ďalšie uzly pridaním jednoprvkovej udalosti na koniec rodičovskej sekvencie.

Bitmapová reprezentácia umožňuje výraznú kompresiu dát v procese dolovania sekvenčných vzorov a efektívnejšie počítanie podpory sekvenčných vzorov. Pre každú položku v sekvenciách databázy sa vytvára bitmapová reprezentácia. Každá bitmapa obsahuje bity, kde jeden bit reprezentuje jednu sekvenciu z databázy. Ak prvok *i* sa nachádza v sekvencií *j*, tak bit súvisiaci so sekvenciou *j* v bitmape pre prvok *i* nadobúda hodnotu 1, v opačnom prípade 0. Bitmapová reprezentácia databáze tiež umožňuje rýchle generovanie a testovanie kandidátnych sekvenčných vzorov pomocou operácie AND [7].
Pseudokód SPAM algoritmu vyzerá nasledovne:

***Vstup:***

*Databáza sekvencií;*

*Minimálna podpora;*

***Výstup:***

*Množina frekventovaných sekvenčných vzorov;*

***Metóda:***

***function*** *SPAM(n=($s_1$,...,$s_k$), $S_n$, $I_n$):*

$S_{temp} = \theta$;

$I_{temp} = \theta$;

***For each*** *(i ∈ $S_n$)* ***do***

***if*** *(<$s_1$,...,$s_k$, {i}> je frekventovaná)* ***do***

$S_{temp} = S_{temp} U \{i\}$;

***end;***

***end;***

***For each*** *(i ∈ $S_{temp}$)* ***do***

*SPAM(<$s_1$,...,$s_k$, {i}>,$S_{temp}$, všetky prvky v $S_{temp}$ > i);*

***end;***

***For each*** *(i ∈ $I_n$)* ***do***

***if*** *(<$s_1$,...,$s_k$, U {i}> je frekventovaná)* ***do***

$I_{temp} = I_{temp} U \{i\}$;

***end;***

***end;***

***For each*** *(i ∈ $I_{temp}$)* ***do***

*SPAM(<$s_1$,...,$s_k$, U {i}>,$S_{temp}$, všetky prvky v $I_{temp}$ > i);*

***end;***

***end;***

Na základe vlastností algoritmu a metód, ktoré využíva pre dolovanie sekvenčných vzorov, je SPAM algoritmus vhodný na databázy s dlhými sekvenciami a veľkým množstvom sekvencií. [4]

#### 1.2.4.4 Algoritmy založené na uzavretých sekvenčných vzoroch

Vyššie predstavené algoritmy pre dolovanie sekvenčných vzorov dolujú všetky možné sekvenčné vzory, ktoré prejdú testom minimálnej podpory. Avšak v prípade, kedy sa pracuje s dlhými sekvenciami (rádovo niekoľko desiatok udalostí a prvkov), tieto algoritmy vygenerujú pre jednu frekventovanú sekvenciu niekoľko ďalších frekventovaných podsekvencií, čo je časovo a priestorovo náročné. Algoritmy založené na uzavretých sekvenčných vzoroch dolujú a generujú menší počet sekvenčných vzorov a to vedie k lepšej efektívnosti týchto algoritmov. V nasledujúcej časti je predstavený uznávaný BIDE algoritmus [8].

##### 1.2.4.4.1 BIDE algoritmus

Bide algoritmus je aktuálne najefektívnejší algoritmus pre dolovanie uzavretých sekvenčných vzorov. Pristupuje k vstupnej databáze a dolovaniu sekvenčných

vzorov rovnakým spôsobom, ako algoritmus PrefixSpan. Rozdiel je ten, ako určuje, či sekvenčný vzor je uzavretý.

Pre zistenie, či generovaný sekvenčný vzor je uzavretý alebo nie, algoritmus BIDE používa *kontrolu uzavretosti obojstranného rozšírenia (BI-Directional Extension closure checking)* [8]. Táto kontrola je postavená nasledujúcom tvrdení: Majme sekvenciu s= <$e_1$,$e_2$, $e_3$,..., $e_n$>. Táto sekvencia nie je uzavretá v prípade, ak platí aspoň jedna z nasledujúcich podmienok:

1. V sekvenčnej databáze existuje sekvencia $s'$= <$e_1$,$e_2$, $e_3$,..., $e_n e'$>, pre ktorú platí, že podpora(s) = podpora($s'$). Udalosť $e'$ v tomto prípade sa nazýva *udalosťou dopredného rozšírenia* a $s'$sa nazýva *dopredne rozšírená sekvencia.*
2. V sekvenčnej databáze existuje sekvencia $s'$= <$e'$, $e_1$,$e_2$, $e_3$,..., $e_n$>, pre ktorú platí, že podpora(s) = podpora($s'$). Udalosť $e'$ v tomto prípade sa nazýva udalosťou spätného rozšírenia a $s'$sa nazýva spätne rozšírenou sekvenciou.

Pseudokód BIDE algoritmu je nasledovný:

***Vstup:***

*Sekvenčná databáza;*

*Minimálna podpora;*

***Výstup:***

*Množina uzavretých sekvenčných vzorov;*

***Metóda:***

***function*** *BIDE(α, S|α):*

*α - sekvenčný vzor (prefix);*

*S|α - α -projektovaná databáza, ak α != <>, inak je to Databáza sekvencií;*

*localFreqItems = Nájdi lokálne frekventované položky v S|α;*

*forwardCount = |{i* ***in*** *localFreqItems | podpora (i) == podpora(α)} |;*

*backwardCount = nájdi v S|α udalosti spätného rozšírenia;*

***if****(forwardCount == 0 && backwardCount ==0)* ***do***

*rozšír množinu uzavretých sekvenčných vzorov o α;*

```
        end;
        for each (i ∈ localFreqItems) do
            α′ = <α, {i} >;
            vytvor α′ - projektovanú databázu S|α′;
            if(projektovaná databáza môže obsahovať sekvenčný vzor) do
                BIDE(α′, S|α′);
            end;
        end;
    end;
    BIDE(<>, Sekvenčná databáza);
```

### 1.2.5 Sekvenčné pravidlá

Vydolované sekvenčné pravidlá majú informatívny charakter. Opisujú, v koľkých sekvenciách z celkového počtu všetkých sekvencií bol vzor prítomný. Avšak sekvenčné vzoru sa môžu využiť aj pre predpovedanie budúcnosti. K tomu slúžia tzv. sekvenčné pravidlá [5]. Sekvenčné pravidlo sa skladá z jednoduchej implikácie, kde na oboch stranách sa nachádza frekventovaný sekvenčný vzor. V prípade, ak existuje sekvenčný vzor $\alpha$ = <a, b, c> s podporou 15% a sekvenčný vzor $\beta$= <a, b, c, d> s podporou 10%, je možné z nich vytvoriť sekvenčné pravidlo:

<a, b, c> ⇒<a, b, c, d> (podpora = 10%, spoľahlivosť 66%)

Predpokladajme, že uvedené sekvencie sú kúpené produkty. Toto pravidlo vyjadruje, že v 10% všetkých transakcií zákazníci kúpili produkty *a, b, c* a následne kúpili produkt *d*. Spoľahlivosť predstavuje, s akou pravdepodobnosťou zákazník po kúpe produktov *a,b,c* kúpi aj produkt *d*. Pseudokód pre generovanie sekvenčných pravidiel vyzerá nasledovne:

***Vstup:***

*Množina sekvenčných vzorov;*

*Minimálna spoľahlivosť;*

***Výstup:***

*Množina sekvenčných pravidiel;*

***Metóda:***

***for each*** *(β ∈ množina sekvenčných vzorov)* ***do***

***for each*** *(podsekvencia α ∈ β)* ***do***

*spoľahlivosť = podpora(β) / podpora(α);*

***if****(spoľahlivosť > minimálna spoľahlivosť)* ***do***

*Rozšír množinu sekvenčných pravidiel o (α ⇒ β,* spoľahlivosť*);*

***end;***

***end;***

***end;***

## 1.3 Programovací jazyk Python

Pre vývoj algoritmu na dolovanie sekvenčných vzorov bol zvolený programovací jazyk Python, ktorý je často používaný na dátové analýzy. Python sa riadi medzi vysoko-úrovňové programovacie jazyky. Python je dynamický interpretovaný programovací jazyk. Používa čistú a pomerne jednoduchú syntax, vďaka ktorej je jednoduché sa ho naučiť a používať. Obsahuje veľké množstvo interných a externých knižníc, ktoré umožňujú vytvárať sofistikované programy a algoritmy.

### 1.3.1 Dátové typy

Python je beztypový jazyk, čo znamená, že do premenných je možné priradiť hociaký dátový typ a až po aktuálne priradenej hodnoty sa určí, ako sa má s danou premennou zaobchádzať. Dátové typy, ktoré využíva Python, sú uvedené v nasledujúcej tabuľke [9].

**Tabuľka č. 3:** Dátové typy jazyka Python (vlastné spracovanie) [23]

| Typ | Názov | Príklad deklarácie |
|---|---|---|
| Prázdny typ | NULL | x=NONE |
| Celé číslo | integer | x=3 |

| | | |
|---|---|---|
| Racionálne číslo | float | x=3.0 |
| Komplexné číslo | komplex | x=3.0 + 2j |
| Reťazec | string | x=”abc” |
| Zoznam | list | x=[1,2,3,”bc”] |
| N-tica | tuple | x=(3,6,2) |
| Set | set | x={“jablko”, “hruška”} |
| Slovník | dictionary | x={“štyri”: 4, “tri”: 3} |

**Zoznam** je usporiadaná postupnosť prvkov. Je to jeden z najpoužívanejších dátových typov v programovacom jazyku Python a je veľmi flexibilný.

**N-tica** je usporiadaná postupnosť prvkov, rovnako ako zoznam. Rozdiel je v tom, že N-tica je nemenná. Raz vytvorenú N-ticu nie je možné upraviť alebo meniť.

**Set** je neusporiadaná zbierka jedinečných hodnôt alebo objektov.

**Slovník** je neusporiadaná kolekcia kľúčovo-hodnotových párov. Spravidla sa používa pri veľkých množstvách dát. Slovníky sú optimalizované na načítanie dát. Pri získanie hodnoty je nutné poznať príslušný kľúč.

# 2 Analýza súčasného stavu

Táto kapitola sa venuje službe Google Analytics 4, ktorá je používaná na meranie návštevnosti a chovania sa užívateľov na webových stránkach a v mobilných aplikáciách. Táto služba ukladá veľké množstvo dát o interakciách užívateľov a navrhovaný algoritmus bude musieť byť schopný pracovať s týmito dátami. Z toho dôvodu sa táto analytická časť práce zaoberá aj podobou meraných dát a celkovým dátovým modelom služby. Táto služba je najviac používaným nástrojom webovej analytiky, ktorá je oblasť zaoberajúca sa používaním webových stránok a ich optimalizáciou. Je rovnako najpoužívanejší nástroj analytického oddelenia firmy House of Řezáč s.r.o , ktoré pomocou dát meraných touto službou spolupracuje s designovým oddelením na optimalizácií a rozvoji webových stránok. Ako každá služba a každý nástroj, aj táto má svoje výhody, nevýhody a obmedzenia, ktoré sú odhalené v nasledujúcich podkapitolách.

## 2.1 Google Analytics 4

Google Analytics 4 je bezplatná služba, predstavená spoločnosťou Google v októbri 2020 ako nástupca služby Google Analytics Universal Analytics. Viac o predchodcovi je možné zistiť tu [10]. Slúži na zber a analýzu dát o návštevnosti a interakciách užívateľov s webovými stránkami alebo mobilnými aplikáciami. Jej vlastnosti a aplikácia sú predstavené v nasledujúcich podkapitolách.

### 2.1.1 Google Analytics 4 ako nástroj webovej analytiky

Webová analytika je oblasť, ktorá sa zaoberá meraním, zberom a vyhodnotením internetových dát z používania webových stránok pre porozumenie správania sa užívateľov webov a optimalizáciu webových stránok. Zaoberá sa konkrétnymi vecami, ako napríklad celkový počet zobrazení podstránok webovej stránky, počet jedinečných návštev, z akých zdrojov a akým spôsobom sa užívatelia dostali na stránku, aktivitou užívateľov (interné vyhľadávanie, používanie funkcionalít na webe, čítanosť blogových článkov, najviac zakúpené produkty, najviac zobrazené

produkty, zobrazovanie reklám), atribúciou transakcií (ku ktorému zdroju návštevy pripísať transakciu), nefunkčnými oblasťami webových stránok (prečo užívatelia nečítajú články, prečo nenakupujú, prečo odchádzajú zo stránok) a inými. Webová analytika je tiež úzko spätá s designom webových stránok, kedy na základe nameraných dát a odporúčaniach je možné upravovať obsah, rozloženie prvkov stránky, objednávkový proces a mnoho ďalšie.

### 2.1.2 Nasadenie Google Analytics 4

Pre nasadenie služby Google Analytics 4 na webovú stránku alebo mobilnú aplikáciu je potrebné uskutočniť nasledujúce kroky:

1. vytvorenie účtu Analytics podľa [11]
2. vytvorenie služby Google Analytics 4 podľa [12]
3. pridanie dátového streamu podľa [12] - slúži na určenie, či služba bude nastavená na webovú stránku, Android aplikáciu alebo iOS aplikáciu
4. nasadenie základného meracieho kódu podľa [12]

#### 2.1.2.1 Základný merací kód

Základný merací kód je blok Javascriptového kódu, ktorý je pridaný na každú podstránku webovej stránky v HTML štruktúre webu za otváracou značku <head>. Pomocou základného kódu sa asynchrónne načíta Javascriptová knižnica s názvom gtag.js, ktorá má za úlohu posielanie dát Google Analytics serverom pomocou HTTP požiadaviek. Viacej informácií o HTTP požiadavkách v [13].

Merací kód je možné implementovať dvoma spôsobmi:

1. priame vloženie meracieho kódu gtag.js do HTML štruktúry webu [14]
2. vloženie pomocou správcu značiek Google Tag Manager [15]

Nasledujúci blok kódu je ukážka základného meracieho kódu gtag.js. Namiesto *ID* je potrebné vyplniť jedinečný merací kód, dostupný v službe Google Analytics 4.

```
<script async src="https://www.googletagmanager.com/gtag/js?id=ID">
</script>
```

```
<script>
    window.dataLayer = window.dataLayer || [];
    function gtag(){dataLayer.push(arguments);}
    gtag('js', new Date());
    gtag('config', ID');
</script>
```

### 2.1.3 Zber dát

Každá HTTP požiadavka na servery Google Analytics 4 je udalosť. Platí princíp,že každá interakcia s webovou stránkou, napríklad zobrazenie stránky, použitie tlačidla, opustenie stránky, sa zaznamenáva ako udalosť. S každou udalosťou, ktorá sa pošle do Google Analytics 4, sa tiež automaticky zaznamenávajú informácie o užívateľovi resp. o prehliadači a stránke, ktoré sú označované ako parametre udalostí. Medzi automaticky zaznamenané parametre udalostí patria:

**Tabuľka č. 4:** Zoznam automaticky meraných parametrov udalostí v GA4 (vlastná úprava) [16]

| **názov parametru** | **vysvetlenie** |
|---|---|
| language | nastavený jazyk prehliadača |
| page_location | URL stránky |
| page_referrer | URL predošlej zobrazenej stránky |
| page_title | názov stránky |
| screen_resolution | rozlíšenie obrazovky |
| event_timestamp | čas uskutočnenia udalosti |
| ga_session_id | jedinečné označenie relácie (návštevy) |
| session_engaged | označenie návštevy ako zaujatá návšteva |
| traffic_source | zdroj / pôvod návštevy |
| traffic_medium | všeobecná kategória zdroja |

Pomocou predošlých parametrov, ktoré sú odosielané v reťazci HTTP požiadavky na servery Google Analytics 4, sa ďalej dopočítavajú a extraktujú ďalšie parametre, ktoré sú automaticky spojené s udalosťou:

**Tabuľka č. 5:** Zoznam automaticky dopočítaných parametrov udalostí v GA4 (vlastná úprava) [16]

| **názov parametru** | **vysvetlenie** |
|---|---|
| ga_session_number | celkový počet relácií (návštev), ktoré uskutočnil užívateľ |
| engaged_session_event | poradové číslo udalosti v zaujatej návšteve |
| event_bundle_sequence_id | označenie udalosti v sekvencií udalostí |
| platform | zaradenie do typu dátového streamu (web, app) |
| web_hostname | doména stránky |

Udalosti sa v rámci Google Analytics 4 delia na nasledujúce skupiny:

1. automaticky zhromažďované udalosti
2. udalosti vylepšeného merania
3. odporúčané udalosti
4. vlastné udalosti

#### 2.1.3.1 Automaticky zhromažďované udalosti

Tieto udalosti sú zaznamenané automaticky pomocou implementovaného základného meracieho kódu. Udalosti a parametre udalostí majú preddefinované názvy a schému, a nie je možné ich meniť. Sú to nasledujúce udalosti:

**Tabuľka č. 6:** Zoznam automaticky zhromažďovaných udalostí v GA4 (vlastná úprava) [16]

| **názov udalosti** | **vysvetlenie** |
|---|---|
| first_visit | prvá návšteva webovej stránky užívateľom |

| | |
|---|---|
| page_view | načítanie novej stránky alebo zmena stavu histórie prehliadača |
| session_start | začiatok návštevy webovej stránky |
| user_engagement | zaujatosť užívateľa stránkou (čas strávený aktívnym používaním stránky) |

Kompletný zoznam automaticky zhromaždovaných udalostí v Android a iOS aplikáciách sa nachádza v [16].

#### 2.1.3.2 Udalosti vylepšeného merania

Udalosti vylepšeného merania sú špeciálna sada udalostí, ktoré sa merajú automaticky a bez nutnosti zmeny meracieho kódu. Je možné ich meranie zapnúť a vypnúť prostredníctvom užívateľského rozhrania služby Google Analytics 4 . Udalosti a parametre udalostí používajú preddefinované názvy a schému. Patria k nim nasledujúce udalosti:

**Tabuľka č. 7:** Zoznam udalostí vylepšeného merania (vlastná úprava) [16]

| **názov udalosti** | **parametre udalosti** | **vysvetlenie** |
|---|---|---|
| click | link_classes<br>link_domain<br>link_id<br>link_url<br>outbound | kliknutie na odkaz, ktorý vedie na webovú stránku s inou doménou |
| file_download | file_extension<br>link_id<br>link_domain<br>file_name<br>link_text<br>link_classes<br>link_url | kliknutie na odkaz vedúci k stiahnutiu súboru |
| scroll | | prvý moment, kedy sa užívateľ dostane na koniec stránky (90% výšky |

| | | |
|---|---|---|
| | | stránky) |
| view_search_results | search_term | použitie interného vyhľadávania na stránke (prítomnosť vyhľadávacieho parametru v URL stránky) |
| video_start | video_current_time<br>video_duration<br>video_percent<br>video_provider<br>video_title<br>video_url<br>visible | interakcia s videami na stránke |
| video_progress | | |
| video_complete | | |

Kompletný zoznam udalostí vylepšeného merania v Android a iOS aplikáciách, typy podporovaných stiahnuteľných súborov a podporované parametre interného vyhľadávania sa nachádzajú v [16].

#### 2.1.3.3 Odporúčané udalosti

Odporúčané udalosti majú preddefinované názvy udalostí a parametrov a sú odporúčané pre konkrétne obchodné odvetvia. Používanie odporúčaných udalostí s predpísanými parametrami poskytuje v reportoch služby GA4 maximálne množstvo podrobností a využívanie najnovších funkcií a integrácií [17]. V dobe, kedy táto práca vzniká, žiadne funkcie, integrácie ani výhody používania odporúčaných udalostí neexistujú. Tieto udalosti nie sú automaticky zhromažďované a pre ich používanie je nutné pridanie rozširujúceho kódu na webové stránky alebo aplikácie. Odporúčané udalosti sa na základe odvetvia webovej stránky delia na:

1. udalosti pre elektronické obchody
2. udalosti pre pracovné ponuky, vzdelávanie, miestne predajné akcie, nehnuteľnosti
3. udalosti pre cestovanie (hotely, letenky)
4. udalosti pre hry

Nasledujúca tabuľka obsahuje príklady odporúčaných udalostí a parametrov.. Pomenovanie väčšiny udalostí je rovnaké v odvetviach. Kompletný zoznam udalostí pre webové stránky a aplikácie sa nachádza na [18].

**Tabuľka č. 8:** Zoznam odporúčaných udalostí (vlastná úprava) [16]

| **názov udalosti** | **parametre udalosti** | **vysvetlenie** |
|---|---|---|
| purchase | transaction_id<br>items<br>value<br>tax<br>shipping<br>currency<br>coupon<br>affiliation | uskutočnenie nákupu |
| view_item | currency<br>items<br>value | zobrazenie detailu produktu |
| add_to_cart | currency<br>items<br>value | pridanie produktu do nákupného košíka |
| generate_lead | value<br>currency | kontaktovanie spoločnosti užívateľom |
| begin_checkout | coupon<br>currency<br>items<br>value | priechod objednávkovým procesom |
| select_item | items<br>item_list_name<br>item_list_id | výber produktu zo zoznamu produktov |
| view_cart | currency<br>items<br>value | zobrazenie obsahu košíka |
| view_promotion | items<br>promotion_id<br>promotion_name<br>creative_name<br>creative_slot<br>location_id | zobrazená reklama užívateľovi |

| | | |
|---|---|---|
| select_promotion | items<br>promotion_id<br>promotion_name<br>creative_name<br>creative_slot<br>location_id | kliknutie na zobrazenú reklamu užívateĺom |

Príklad využitia rozširujúceho kódu s odporúčanými názvami udalostí a parametrov pri dokončení a uskutočnení objednávky pomocou gtag.js:

```
gtag('event', 'purchase', {
    currency: 'CZK',
    items: [{
        item_id: '12345',
        item_name: 'Modré tričko L',
        item_category: 'tričká',
        item_variant: 'modré',
        price: 249,
        currency: 'CZK',
        quantity: 1
        }, {
        item_id: '12346',
        item_name: 'Čierne rifle 32',
        item_category: 'nohavice',
        item_variant: 'čierne',
        price: 499,
        currency: 'CZK',
        quantity: 1
    }],
    transaction_id: 'T_12345',
    shipping: 50,
    value: 590,
    tax: 157
})
```

#### 2.1.3.4 Vlastné udalosti

Do tejto skupiny patria všetky udalosti, ktoré nie sú zahrnuté v predošlých skupinách. Je potrebné implementovanie a pridanie rozširujúceho kódu. Google Analytics 4 odporúča pred používaním vlastných udalostí uistiť sa, či pre účel nepostačí úplné využitie automatických udalostí, udalostí vylepšeného merania a odporúčaných udalostí, a navyše vyhlasuje, že tieto udalosti nebudú kompatibilné s budúcimi funkciami služby.
Príklad využitia rozširujúceho kódu s vlastnými názvami udalostí a parametrov pri vložení recenzie produktu užívateľom:

```
gtag('event', 'nová_recenzia', {
    hodnotenie: 4/5,
    produkt_id: '12345',
    text_hodnotania: 'TEXT',
    názov_produktu: 'Ponožky XL',
    užívateľ_prihlásený: 'ne'
})
```

#### 2.1.3.5 Konverzia

Konverzia alebo konverzná udalosť sú aktivity, ktoré sú najdôležitejšie pre podnik. Sú to aktivity, ktoré prispievajú k úspechu webovej stránky, aplikácie a firmy. Medzi konverzie patria napríklad:

1. nákup (na webe elektronického obchodu)
2. dokončenie hernej úrovne (v mobilnej hernej aplikácii)
3. odoslanie formulára s kontaktnými údajmi (na marketingovom webe alebo webe generovania potenciálnych zákazníkov)

Google Analytics 4 automaticky označuje nasledujúce udalosti ako konverzie [17]:

1. first_open (aplikácia)
2. in_app_purchase (aplikácia)
3. app_store_subscription_convert (aplikácia)
4. app_store_subscription_renew (aplikácia)

5. purchase (web)

Okrem automaticky zhromažďovaných konverzných udalostí je možné v užívateľskom rozhraní GA4 označiť 30 udalostí ako konverzné udalosti. Tie sa určujú na základe názvy udalosti a parametrov udalosti.

#### 2.1.3.6 Zdroj/médium relácie

Pri každej relácii a udalostiach Google Analytics 4 zaznamenáva parametre pre zdroj a médium. Zdroj a médium sú užitočné dimenzie pre zistenie, skade a akým spôsobom sa užívatelia dostali na webovú stránku. Pomocou nich je možné segmentovať užívateľov na jednotlivé zdroje a médiá a porovnať ich chovania.

**Zdroj** je pôvod návštevnosti, napríklad vyhľadávač (google.com) alebo doména (príklad.cz).

**Médium** je všeobecná kategória zdroja, napríklad neplatené vyhľadávanie (organic), vyhľadávanie platené za preklik (cpc), odkazujúci zdroj na webovej stránke (referral).

| | Session source/medium + | ↓Users | Sessions | Engaged sessions | Average engagement time per session | Engaged sessions per user |
|---|---|---|---|---|---|---|
| | Totals | 1,603<br>100% of total | 2,453<br>100% of total | 1,776<br>100% of total | 1m 00s<br>Avg 0% | 1.108<br>Avg 0% |
| 1 | google / organic | 525 | 844 | 641 | 1m 20s | 1.221 |
| 2 | (direct) / (none) | 434 | 670 | 435 | 0m 55s | 1.002 |
| 3 | m.facebook.com / referral | 264 | 287 | 228 | 0m 25s | 0.864 |
| 4 | l.facebook.com / referral | 125 | 155 | 136 | 1m 21s | 1.088 |
| 5 | linkedin.com / referral | 51 | 77 | 58 | 0m 21s | 1.137 |
| 6 | ecomail / email | 47 | 123 | 70 | 0m 40s | 1.489 |
| 7 | t.co / referral | 33 | 44 | 32 | 0m 25s | 0.97 |
| 8 | facebook.com / referral | 19 | 19 | 18 | 0m 18s | 0.947 |

**Obrázok č. 2:** Náhľad prehľadu o zdrojoch a médiách návštev v GA4

### 2.1.4 Dimenzie a metriky

Všetky udalosti v Google Analytics 4 majú parametre udalostí, ktoré špecifikujú udalosť. Tieto parametre sa delia na dimenzie a metriky podľa toho, či sú vyplnené nečíselnými alebo číselnými hodnotami.

**Dimenzie** sú nečíselné atribúty (parametre) udalostí, ktoré slúžia na popis, segmentáciu, usporiadanie a triedenie údajov. Typickými dimenziami v Google Analytics 4 sú:

1. zdroj návštevy
2. štát
3. jazyk
4. typ prehliadača

**Metriky** sú vyjadrené číselnými hodnotami, predstavujú kvantitatívne merania údajov a ukazujú, akú výkonnosť má webová stránka v súvislosti s konkrétnou dimenziou. Typickými metrikami v Google Analytics 4 sú:

1. počet užívateľov
2. počet nových užívateľov
3. počet udalostí
4. počet relácií
5. počet transakcií
6. celková hodnota transakcií
7. konverzný pomer transakcií

Príklad dimenzie a metrík:

**Tabuľka č. 9:** Príklad dimenzie a metrík (vlastné spracovanie)

| **Dimenzia** | **Metrika** | **Metrika** | **Metrika** |
|---|---|---|---|
| **Mesto** | **Počet relácií** | **Počet udalostí page_view** | **Počet udalostí page_view na reláciu** |
| Košice | 5000 | 16500 | 3,3 |
| Praha | 4000 | 10000 | 2,5 |
| Berlín | 3000 | 12800 | 4,26 |

Dimenzie a metriky môžu prináležať udalostiam a užívateľovi. Dimenzie a metriky, ktoré patria udalostiam, sa nazývajú parametre udalostí. Dimenzie a metriky, ktoré prináležia užívateľovi, sa nazývajú **vlastnosti užívateľov.** Vlastnosti užívateľov sú parametre posielané s udalosťami, ktoré sa viažu na užívateľa a je ich možné využiť na popis segmentov užívateľov webovej stránky alebo aplikácií. Takýmito užívateľskými vlastnosťami môžu byť nasledujúce dimenzie a metriky:

1. preferovaný jazyk
2. geografické údaje
3. celková hodnota všetkých objednávok užívateľa
4. zaujatý užívateľ
5. nový užívateľ
6. vracajúci sa užívateľ
7. pohlavie
8. prihlásený / registrovaný užívateľ

Google Analytics 4 registruje užívateľské dimenzie a metriky spätne, to znamená, že ak v databáze sa nachádzajú záznamy od užívateľa s rovnakým označením, pridá nastavené dimenzie a metriky [17]. Príklad použitia rozširujúceho kódu pre nastavenie užívateľských dimenzií a metrík pri prihlásení užívateľa na webovej stránke:

```
gtag('set', 'user_properties', {
    prihlásený: 'ano',
    počet_objednávok: 3,
    hodnota_ojednávok: 10000
});
```

Všetky parametre udalostí (dimenzie a metriky) a užívateľov (vlastnosti užívateľov), ktoré nie sú automaticky merané službou Google Analytics 4, ale sú posielané do služby, sa nazývajú **vlastné dimenzie a metriky.** Vlastné dimenzie a metriky je potrebné v užívateľskom rozhraní Google Analytics 4 registrovať, aby sa nachádzali v jednotlivých reportoch v rozhraní. Vlastné dimenzie a metriky sa môžu implementovať jedine pomocou rozširujúcich kódov.

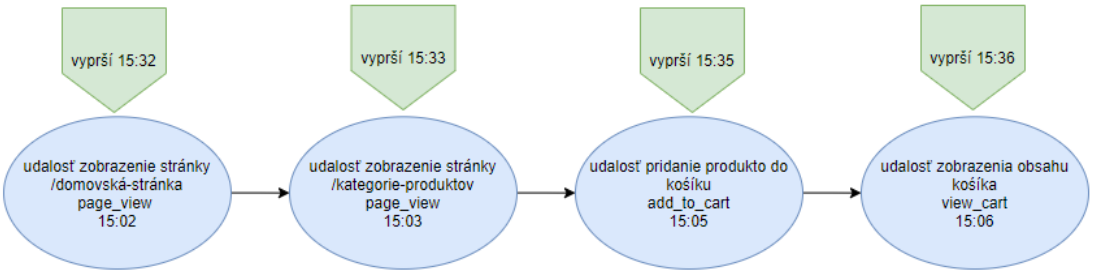
#### 2.1.4.1 Limity dimenzií a metrík

Pri posielaní udalostí, používaní a registrácií parametrov udalostí a užívateľov existujú určité obmedzenia [17]. Sú to:

1. jedna udalosť môže mať najviac 25 vlastných parametrov
2. vlastné dimenzie a metriky v rozsahu udalostí: v GA4 môže byť registrovaných a používaných najviac 50 jedinečne pomenovaných vlastných dimenzií a 50 jedinečne pomenovaných vlastných metrík
3. vlastné dimenzie a metriky v rozsahu užívateľa: v GA4 môže byť registrovaných a používaných najviac 25 jedinečne pomenovaných užívateľských vlastností

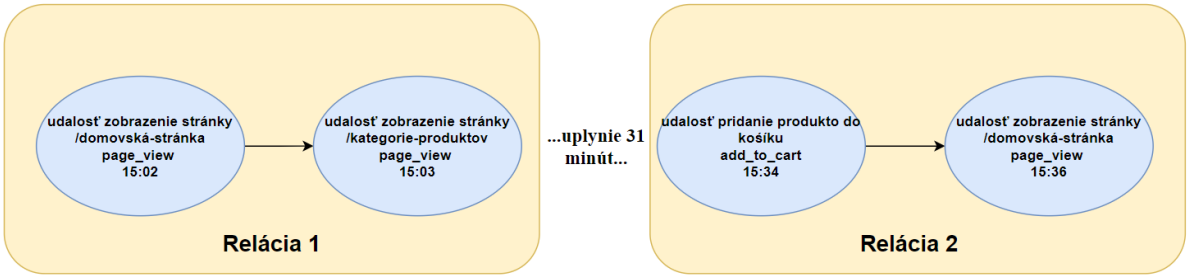
### 2.1.5 Relácie

Relácia resp. návšteva je skupina užívateľských interakcií (udalostí) na webovej stránke alebo v mobilnej aplikácii, ktoré sa uskutočnili v určitom časovom intervale. Meranie relácie sa spustí prvou odoslanou udalosťou zobrazenia stránky *(page_view)*, u ktorej je prítomná udalosť začiatku relácie *(session_start)*. Trvanie relácie je dĺžka časového obdobia medzi prvou a poslednou udalosťou v relácii. Relácia sa skončí po 30 minútach nečinnosti užívateľa [19]. Určenie relácie sa predvedie na nasledujúcom príklade.




**Obrázok č. 3:** Výpočet platnosti relácie (vlastné spracovanie)

Predpokladajme, že užívateľ interaguje s webovou stránkou. Po prvej udalosti zobrazenia stránky sa uplynutie platnosti relácie nastaví na čas 15:32. Keď užívateľ druhýkrát navštívi podstránku webu, čas uplynutia platnosti relácie sa nastaví na 15:33. Ako užívateľ prechádza webom a spúšťa udalosti, každá udalosť posúva čas vypršania platnosti relácie o ďalších 30 minút.




**Obrázok č. 4:** Výpočet relácie (vlastné spracovanie)

Predpokladajme, že užívateľ počas toho, ako prechádzal webovou stránkou, odišiel na obed, kde strávil 31 minút, potom sa vrátil a pokračoval v užívaní webovej stránky. Keď sa užívateľ vrátil z obeda a pokračoval v prehliadaní webových stránok, služba Analytics nastavila nový 30-minútový čas uplynutia platnosti, t. j. začala sa nová relácia. Spôsob určenia a výpočtu relácie v mobilných aplikáciách sa nachádza na [19].

Udalosti , ktoré sú zaznamenané počas trvania relácie, sú s danou reláciou spojené dvoma spôsobmi pomocou parametrov [19]:

1. *ga_session_id:* jedinečný identifikátor pridružený ku každej udalosti, ktorá sa vyskytne počas relácie
2. *ga_session_number:* parameter pridružený ku každej udalosti, ktorá sa vyskytne počas relácie. Identifikuje poradové miesto relácie vo vzťahu k používateľovi, napr. prvú alebo piatu reláciu používateľa. Je to užitočné, ak je potrebné zistiť, kedy sa vyskytujú určité typy udalostí. Je možné napríklad zistiť, že udalosti *purchase* sa v 85 % prípadov vyskytujú počas 5. až 7. relácie

#### 2.1.5.1 Dimenzie a metriky spojené s reláciami

Google Analytics 4 automaticky vypočítava dimenzie a metriky, ktoré sú spojené s reláciami a sú prístupné v reportoch v užívateľskom rozhraní [19]. Sú to:

1. *Sessions (relácie):* počet relácií, ktoré sa uskutočnili na webovej stránke alebo v mobilnej aplikácií
2. *Engaged sessions (zaujaté relácie):* počet relácií, ktoré trvali aspoň desať sekúnd, zaznamenali konverznú udalosť alebo zaznamenali dve zobrazenia stránok (page_view)
3. *Engaged session per user:* počet zaujatých relácií / počet užívateľov
4. *Average engagement time per session:* priemerný čas zaujatosti na reláciu
5. *Engagement rate:* zaujaté relácie / relácie
6. *Event count:* počet udalostí za reláciu
7. *Conversions:* počet konverzných udalostí za reláciu
8. *Conversion rate:* počet relácií s konverznými udalosťami / počet relácií

### 2.1.6 Prehľad a export dát

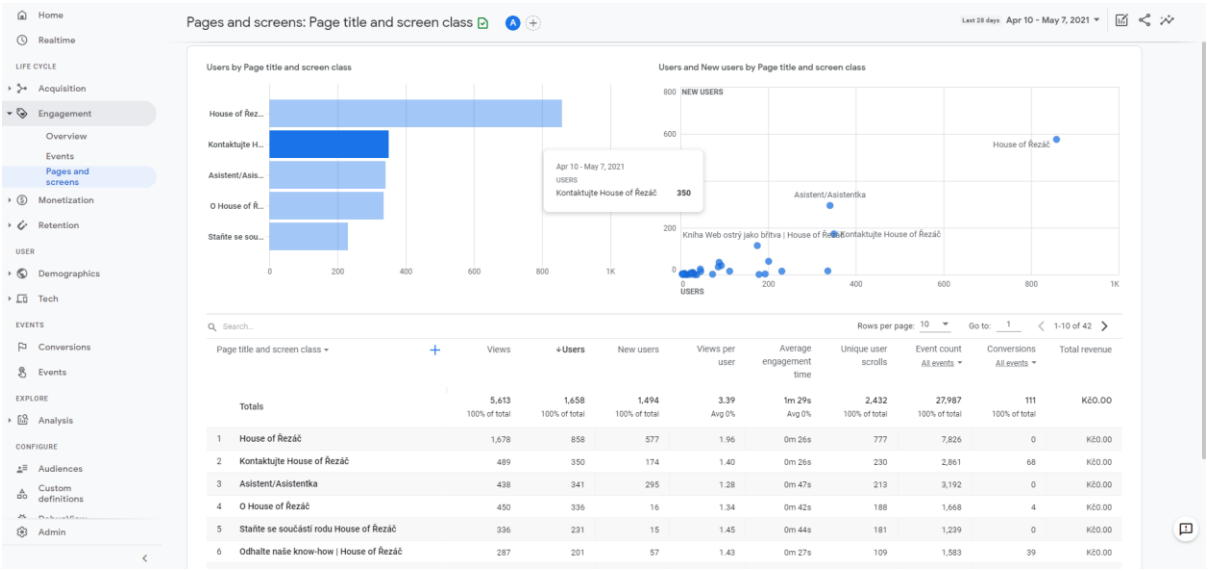
V nasledujúcich častiach sú predstavené všetky aktuálne dostupné spôsoby analýzy a vizualizácie dát meraných pomocou Google Analytics 4.

#### 2.1.6.1 Užívateľské rozhranie Google Analytics 4

Užívateľské rozhranie ponúka niekoľko jednotiek prehľadov dát, ktoré ukazujú súhrnné informácie a vizualizácie o užívateľoch a udalostiach z nasledujúcich pohľadov:

1. akvizícia - zdroje a médiá, prostredníctvom ktorých prišli užívatelia na web
2. zapojenie - zapojenie užívateľov podľa udalostí a stránok
3. speňažovanie - objem kupujúcich a výnosy podľa produktov
4. udržanie - udržanie podľa nových a vracajúcich sa užívateľov

5. demografické údaje - počet užívateľov podľa dimenzie demografickej skupiny
6. technológia - technológie používané na interakcie s webovou stránkou

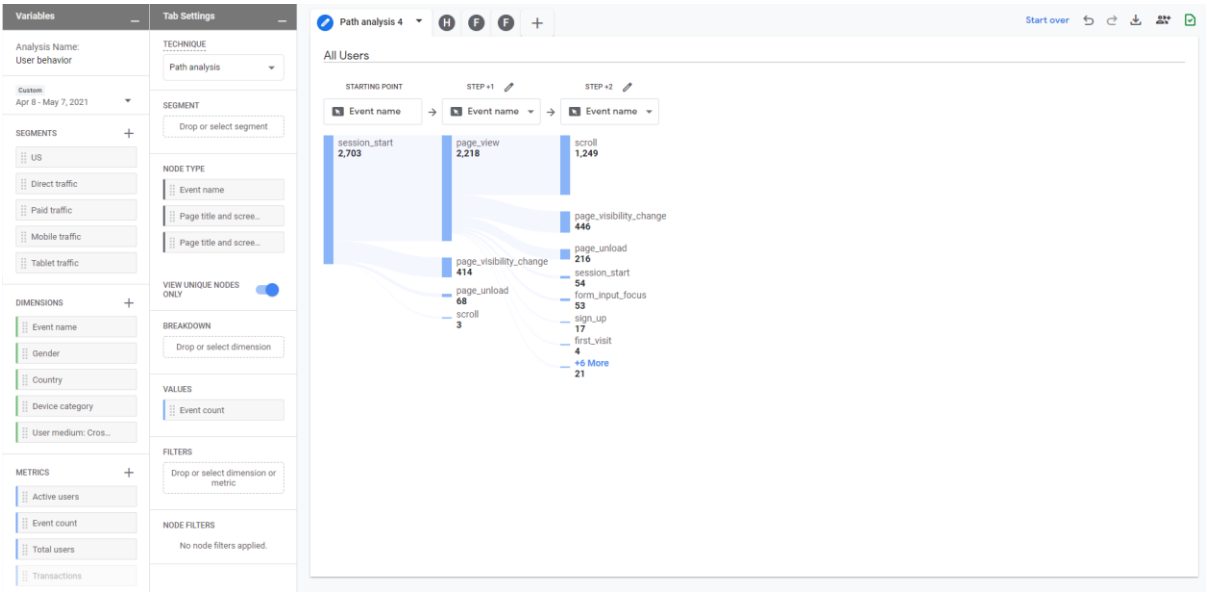



**Obrázok č. 5:** Náhľad užívateľského rozhrania Google Analytics 4 (vlastné spracovanie na základe dát GA4 webovej stránky www.houseofrezac.com)

Okrem základných prehľadov dát GA4 ponúka možnosť Analýzy (Analysis). Analýza predstavuje súbor pokročilých techník, ktoré svojou funkčnosťou presahujú rámec štandardných prehľadov a umožňuje získať podrobnejšie štatistiky správania užívateľov. Tieto techniky sú:

1. analýza lievika - Umožňuje zobraziť kroky, ktoré vykonali užívatelia v snahe uskutočniť nejakú udalosť a zistiť, ako sa im darí v jednotlivých krokoch.
2. analýza cesty - Ponúka preskúmanie cesty užívateľov v stromovom grafe, napr. najčastejšie zobrazené stránke po otvorení domovskej stránky, zistenie účinky udalosti na následné akcie užívateľov
3. tvorba a prekrytie segmentov - Vytváranie podskupín používateľov, relácií a udalostí, a porovnávanie týchto segmentov.
4. prieskum - Umožňuje vizualizovať údaje v tabuľke alebo grafe, usporiadať a zoradiť riadky a stĺpce podľa potreby, porovnať viacero metrík vedľa seba, vytvárať vnorené riadky a zoskupenie údajov, spresniť analýzu pomocou segmentov a filtrov

5. kohortová analýza - Kohorta je skupina užívateľov so spoločnou charakteristikou, napr. užívatelia s rovnakým dátumom príchodu na web patria do rovnakej kohorty. Kohortová analýza umožňuje skúmať správanie týchto skupín.
6. prieskumník užívateľa - Umožňuje analyzovať segmenty užívateľov a hĺbkovo skúmať aktivity jednotlivých používateľov




**Obrázok č. 6:** Náhľad analýzy cesty v GA4 (vlastné spracovanie na základe dát GA4 webovej stránky www.houseofrezac.com)

### 2.1.6.2 BigQuery

BigQuery je služba od spoločnosti Google, cloudové úložisko dát slúžiaci na ukladanie a dotazovanie rozsiahlych dátových skladov. Jeho hlavná výhoda je, že beží na Google serveroch a infraštruktúre, ktoré umožňujú mimoriadne rýchle spúšťať SQL dotazy na dátových skladoch. Táto služba je platená, kde sú spoplatnené ukladanie dát a dotazovanie na ne [20].

Google Analytics 4 poskytuje bezplatný export nameraných dát do BigQuery, kde je možné na ne dotazovať pomocou SQL alebo importovať externé dáta a kombinovať ich s GA4 dátami. Export je možné uskutočniť dvoma spôsobmi:

1. export raz denne
2. export streamovaného obsahu

Pomocou exportu streamovaného obsahu sa v BigQuery sprístupnia dáta behom niekoľkých minút z aktuálneho dňa. Počas exportu sa vytvárajú dva typy tabuliek:

1. tabuľky s názvom *events_YYYYMMDD* - YYYYMMDD je schéma dátumu dňa, obsahuje kompletný export udalostí z daného dňa,
2. tabuľku s názvom *events_intraday_YYYYMMDD* - je interná testovacia tabuľka, ktorá obsahuje záznamy o reláciách a udalostiach v príslušný deň. Vytvára sa počas exportu streamovaného obsahu a snaží sa o čo najväčšiu presnosť, nemusí však obsahovať všetky dáta. Dôvodom nepresnosti môže byť spracovávanie neskorších udalostí alebo chyby v nahrávaní. Dáta sú exportované priebežne počas celého dňa. Tabuľka je zmazaná akonáhle je koniec dňa a je hotová príslušná tabuľky pre denný export.

Výhody používania BigQuery sú:

1. dotazovanie pomocou SQL príkazov a používanie regulárnych výrazov
2. limity na počet parametrov udalostí neexistujú - v dátovom sklade sú všetky parametre uložené
3. import dát pre kombinovanie dát vo formátoch JSON a CSV
4. export dát možný vo formátoch JSON a CSV
5. stabilná služba pre ukladanie a zálohovanie dát
6. doplnkové funkcie pre dotazovanie - napr. funkcie pre rozdelenie URL na doménu a cestu
7. integrácia s ostatnými službami pomocou konektorov - napr. prepojenie v Google Docs alebo MS Excelom
8. uchovávanie dát teoreticky do nekonečna

Nevýhody používania BigQuery sú:

1. spoplatnená služba - ukladanie a zálohovanie dát je prakticky lacnejšie ako dotazovanie
2. absencia vizualizácie údajov pomocou grafov
3. GA4 export obsahuje vnorené záznamy - o niečo náročnejšie SQL dotazy

| Row | event_date | event_timestamp | event_name | event_params.key | event_params.value.string_value | event_params.value.int_value |
|---|---|---|---|---|---|---|
| 1 | 20210502 | 1619952994435190 | first_visit | ga_session_id | *null* | 1619952992 |
| | | | | page_referrer | https://www.houseofrezac.com/ | *null* |
| | | | | page_location | https://www.houseofrezac.com/o-nas | *null* |
| | | | | ga_session_number | *null* | 1 |
| | | | | page_title | O House of Řezáč | *null* |
| | | | | ignore_referrer | true | *null* |
| 2 | 20210502 | 1619952994435190 | session_start | ga_session_number | *null* | 1 |
| | | | | ga_session_id | *null* | 1619952992 |
| | | | | page_title | O House of Řezáč | *null* |
| | | | | page_referrer | https://www.houseofrezac.com/ | *null* |
| | | | | page_location | https://www.houseofrezac.com/o-nas | *null* |
| | | | | ignore_referrer | true | *null* |
| 3 | 20210502 | 1619952994435190 | page_view | ignore_referrer | true | *null* |
| | | | | session_engaged | 0 | *null* |
| | | | | debug_mode | *null* | 1 |
| | | | | entrances | *null* | 1 |
| | | | | ga_session_id | *null* | 1619952992 |
| | | | | page_title | O House of Řezáč | *null* |
| | | | | page_referrer | https://www.houseofrezac.com/ | *null* |
| | | | | page_location | https://www.houseofrezac.com/o-nas | *null* |
| | | | | ga_session_number | *null* | 1 |
| 4 | 20210502 | 1619958136211746 | page_unload | ga_session_id | *null* | 1619956586 |

**Obrázok č. 7:** Náhľad dát v tabuľke exportu v BigQuery (vlastné spracovanie na základe dát GA4 webovej stránky www.houseofrezac.com)

```sql
SELECT
  DISTINCT params.value.string_value,
  COUNT(params.value.string_value) AS pageviews
FROM
  `houseofrezac-ga4.analytics_220933723.events_*`,
  UNNEST(event_params) AS params
WHERE
  event_name = "page_view"
  AND params.key = "page_location"
  AND event_date BETWEEN FORMAT_DATE('%Y%m%d',DATE_SUB(CURRENT_DATE(), INTERVAL 30 DAY))
  AND FORMAT_DATE('%Y%m%d',DATE_SUB(CURRENT_DATE(), INTERVAL 1 DAY))
GROUP BY
  params.value.string_value
ORDER BY
  pageviews DESC
LIMIT
  10
```

Query results SAVE RESULTS EXPLORE DATA

Query complete (1.9 sec elapsed, 23.7 MB processed)

Job information | Results | JSON | Execution details

| Row | string_value | pageviews |
|---|---|---|
| 1 | https://www.houseofrezac.com/ | 1621 |
| 2 | https://www.houseofrezac.com/o-nas | 452 |
| 3 | https://www.houseofrezac.com/weby | 371 |
| 4 | https://www.houseofrezac.com/kontakt | 371 |
| 5 | https://www.houseofrezac.com/kariera | 326 |
| 6 | https://www.houseofrezac.com/kniha | 247 |
| 7 | https://www.houseofrezac.com/reference | 234 |
| 8 | https://www.houseofrezac.com/skoleni | 231 |
| 9 | https://www.houseofrezac.com/asistentka | 174 |
| 10 | https://www.houseofrezac.com/tvorba-webu | 169 |

**Obrázok č. 8:** Príklad využitia SQL dotazu pre zobrazenie prvých 10 najviac zobrazených podstránok za posledný mesiac v BigQuery (vlastné spracovanie na základe dát GA4 webovej stránky www.houseofrezac.com)

### 2.1.6.3 Google Analytics 4 Data API

Data API je sada nástrojov a protokolov určených k extrakcií a exportu surových dát z Google Analytics 4 do vlastných skriptov alebo programov, umožňujúce väčšiu mieru automatizácie a vyššiu efektivitu pri vytváraní analýz. Pomocou nej je možné sa dostať k jednotlivým parametrom udalostí, ktoré sa poslali pri jednotlivých HTTP požiadavkách. Najpravdepodobnejšie práve pomocou Data API sa budú získavať vstupné dáta pre algoritmus na dolovanie sekvenčných vzorov. Tento nástroj je v súčasnosti v beta verzií a jeho technická dokumentácia je málo opísaná [21.

# 3 Vlastný návrh riešenia

Táto časť práce obsahuje vlastný návrh riešenia a prístupu k návrhu algoritmu pre dolovanie sekvenčných vzorov v dátach Google Analytics 4 so zameraním na potreby interných procesov firmy House of Řezáč. House of Řezáč je designovo konzultačná agentúra a zameriava sa na webové stránky. Navrhovaný algoritmus bude používaný hlavne analytickým oddelením firmy a z výstupov algoritmu bude čerpať a využívať ich designové oddelenie. Pre zistenie potrieb a požiadaviek firmy na navrhovaný algoritmus a oblasti, na ktoré bude algoritmus využívaný bol zvolený ideačný workshop, ktorého priebeh a výstupy sú opísane v prvej časti vlastného návrhu riešenia. Na základe výstupov ideačného workshopu a znalostí o stávajúcich algoritmoch pre dolovanie sekvenčných vzorov bude v druhej časti práce navrhnutý podobný algoritmus spĺňajúci požiadavky firmy.

## 3.1 Sekvenčné vzory v designových procesoch

Táto časť práce sa zaoberá získaním a analýzou možných využití dolovania sekvenčných vzorov v designových procesoch firmy House of Řezáč a definovaním požiadaviek na navrhovaný algoritmus.

Pre získanie možných využití sekvenčných vzorov dolovaných v Google Analytics 4 dátach v designových procesoch je potrebné pochopiť, ako pracujú designéri s užívateľskými scenármi, predpokladanými prechodmi webom a pravidelnými očakávateľnými správaniami sa užívateľov. Pre tento účel bol zvolený ideačný workshop. Ideačný workshop bol navrhnutý tak, aby zoznámil zamestnancov firmy so sekvenčnými vzormi a v procese pochopenia práce designérov so vzormi a v procese ideácie o využitiach sekvenčných vzorov v designerských postupoch nepriamo sa nadefinovali technické požiadavky na algoritmus, tvorené hlavne potrebnými typmi sekvenčných vzorov na prácu.

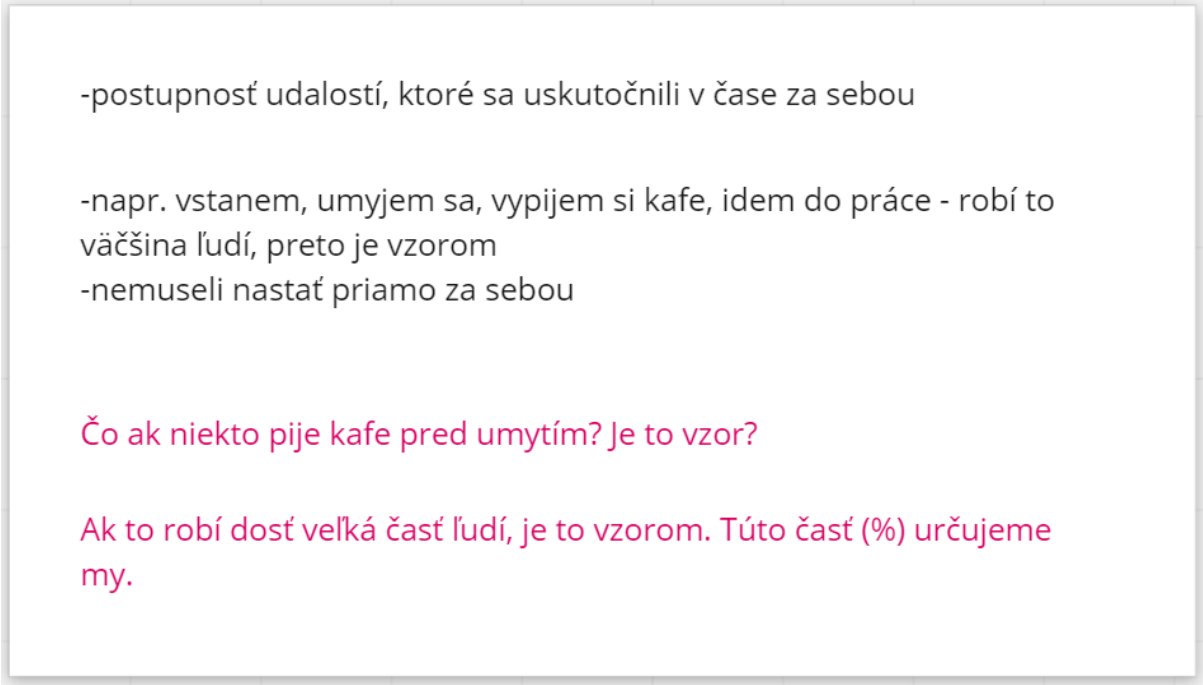
### 3.1.1 Ideačný workshop

Ideačný workshop slúži na kreatívne vymyslenie nápadov, nájdenie príležitostí a ich prioritizáciu. Účastníci workshopu boli traja designéri a jeden webový analytik firmy House of Řezáč. Workshop trval 2 hodiny a ciele boli stanovené nasledovne:

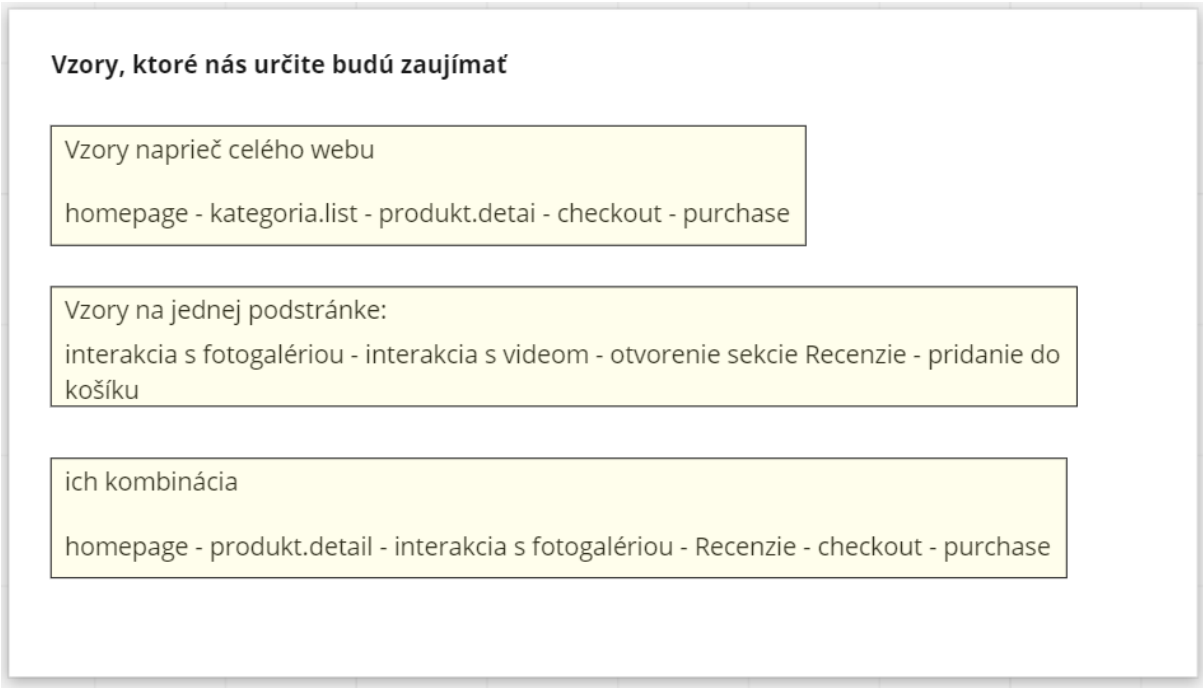
- vysvetliť designérom, čo sú sekvenčné vzory
- pochopiť prácu designérov so vzormi
- nájsť všetky možné využitia sekvenčných vzorov v designe
- čo najpresnejšia definícia podoby dát, s ktorými pracujú / by pracovali
- technické požiadavky na algoritmus pre dolovanie sekvenčných vzorov

Ideačný workshop bol vedený kontaktnou formou, ale pracovalo sa v rámci webovej aplikácie Miro [22]. Všetky nasledujúce výstrižky obrazovky sú z pracovného priestoru aplikácie, v ktorej sa pracovalo.

Prvú časť workshopu predstavovala prezentácia o tom, čo sú sekvenčné vzory a ako sa delia, rozdiel medzi uzatvorenými a neuzatvorenými sekvenčnými vzormi, rozdiel medzi sekvenčnými vzormi s medzerami a sekvenčnými vzormi bez medzier, jednoduché príklady sekvenčných vzorov zo života a predstavenie možných sekvenčných vzorov na webových stránkach, s ktorými sa bude pracovať.




**Obrázok č. 9:** Ukážka prezentácie o sekvenčných vzoroch na ideačnom workshope (vlastné spracovanie)

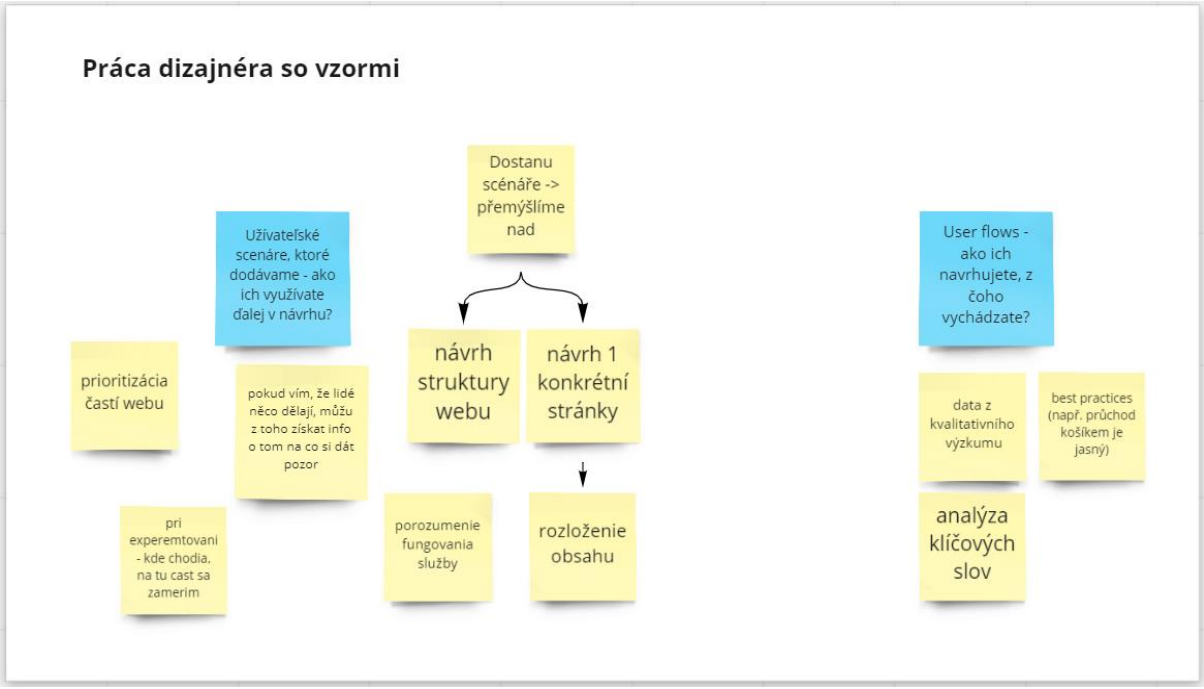



**Obrázok č. 10:** Ukážka prezentácie o sekvenčných vzoroch na webe (vlastné spracovanie)

V druhej časti workshopu designéri ideovali o tom, ako pracujú v súčasnej dobe so vzormi, užívateľskými scenármi, ktoré im dodávajú weboví analytici pomocou Google Analytics Universal Analytics, a ako navrhujú priechody

užívateľov webovou stránkou. Z tejto časti vyplynuli nasledujúce skutočnosti, ako pracujú so vzormi:

1. Pomocou užívateľských scenárov (najviac frekventované priechody webom) navrhujú štruktúru webu alebo štruktúru konkrétnej podstránky - rozloženie obsahu
2. V procese rozvoji webov užívateľské scenáre pomáhajú s porozumením fungovania služby a na prioritizáciu nefunkčných častí webu
3. Využívajú tzv. best practices, napríklad priechod nákupným košíkom
4. Data získavajú z analýzy kľúčových slov a kvalitatívnych výskumov




**Obrázok č. 11:** Ukážka ideácie o stávajúcej práci designéra so vzormi (vlastné spracovanie)

Tretiu časť workshopu tvoril brainstorming na otázku: Ako by sme mohli využiť sekvenčné vzory v dizajne? Podmienka odpovedí bola, aby boli s využitiami čo najviac konkrétni. Výsledky sú nasledovné:

1. Získavanie sekvencií navštívených stránok pred konverznými udalosťami - nákup v elektronických obchodoch alebo kontaktovanie na obsahových weboch - porovnať sekvenciu zobrazených stránok tých ľudí, ktorí

uskutočnili konverznú udalosť a tých, ktorí neuskutočnili a odstrániť možné problémy v navigácií

2. Sekvencie udalostí a interakcií na detaile produktov v elektronických obchodoch, ktoré viedli k nákupu alebo ktoré neviedli k nákupu - pomocou nich by sa dalo zmeniť obsah stránky, pre podporu konverzného pomeru užívateľov
3. Sekvencie marketingových aktivít a kontaktov so značkou - pomocou dát z CRM systémov
4. Získanie možných vzorov v odpade užívateľov v nákupnom procese v košíku - zlepšenie priechodu košíku
5. Zistenie existencie alternatívnych priechodov procesom nákupy alebo vyplňovania formulárov
6. Vzory užívateľov pred kontaktovaním zákazníckej podpory alebo použitím chatbota - zlepšenie navigácie a zviditeľnenie informačných podstránok
7. Analýza obsahu nákupných košíkov a transakcií, a nájdenie existencie určitých produktov, ktoré začínajú nejaké sekvencie - podpora remarketingu
8. Presné získanie užívateľských scenárov naprieč webom
9. Sekvenčné vzory ako podklad ku kvalitatívnemu výskumu - riešenie dôvodu existencie vzorov
10. Získanie nečakaných vzorov a ich pochopenie
11. Vzory v užívaní informačných systémov - odhalenie chýb, nefunkčnosti alebo neintuitívnosti systému
12. Vzory študentov na webovej stránke alebo v informačných systémoch pred podaním prihlášky na štúdium
13. Vzory študentov na webovej stránke alebo v informačných systémoch pred zápisom do štúdia
14. Vzory študentov na webovej stránke alebo v informačných systémoch pred ukončením štúdia


**Obrázok č. 12:** Ukážka z brainstormingu o využitiach sekvenčných vzorov v designe (vlastné spracovanie)

V štvrtej časti workshopu mali designéri za úlohu definovať 2 veci:

1. nakoľko užitočné by pre nich boli výstupy z brainstormingu o využitiach sekvenčných vzorov v ich procesoch a priorizovať ich pomocou škály 1 až 5, kde 1 predstavovala najmenej potrebné vzory a 5 predstavovala najviac užitočné a potrebné vzory
2. ktoré typy sekvenčných vzorov potrebujú v konkrétnych výstupoch (vzory v rámci jednej návštevy, vzory naprieč návštev - všetky návštevy užívateľa)

Nasledujúca tabuľka ukazuje výsledky štvrtej časti workshopu:

**Tabuľka č. 10:** Prioritizácia výstupov workshopu a potrebné sekvencie (vlastné spracovanie)

| **Sekvencie** | **Prioritizácia** | **Typy potrebných sekvenčných vzorov** |
|---|---|---|
| Sekvencie na detaile produktu | 4 | návšteva |
| Sekvencie zobrazených stránok | 5 | návšteva, užívateľ |

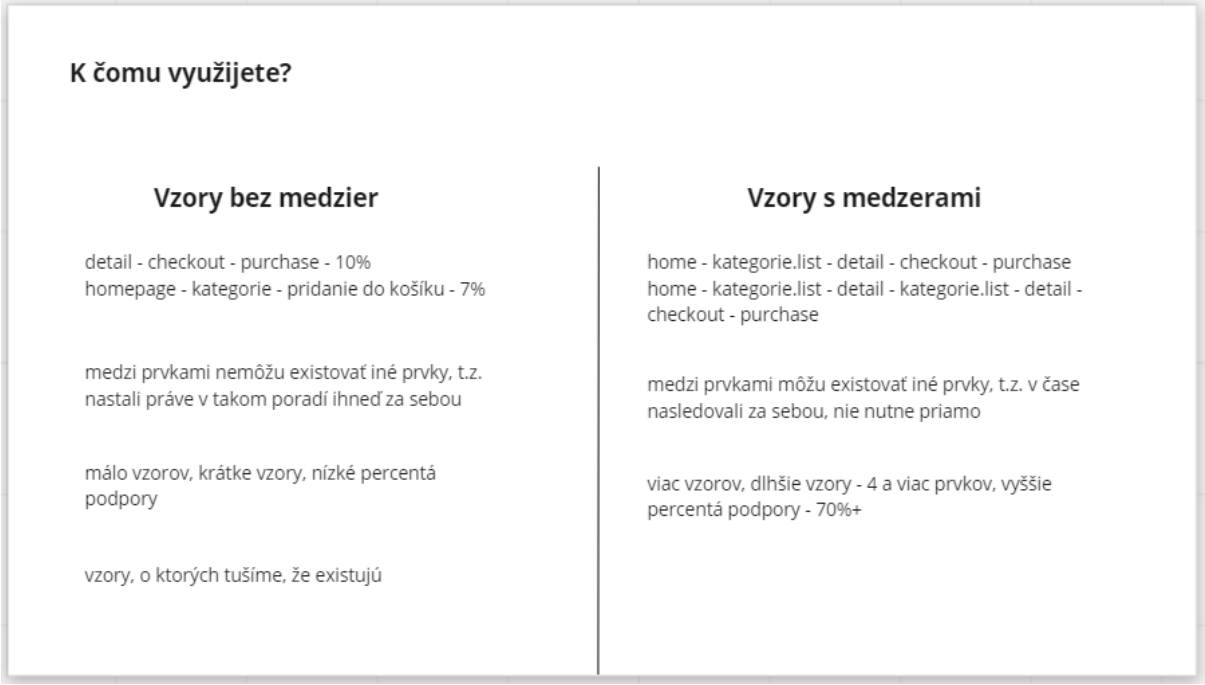
| | | |
|---|---|---|
| pred konverziou | | |
| Sekvencie kontaktov so značkou | 5 | užívateľ |
| Sekvencie v odpade objednávkovým procesov | 4 | návšteva |
| Sekvencie pred použitím chatbota / zákazníckej podpory | 3 | návšteva |
| Sekvencie ako podklad ku kvalitatívnemu výskumu | 5 | návšteva, užívateľ |
| Sekvencie obsahu nákupov | 3 | užívateľ |
| Sekvencie v používaní IS | 1 | užívateľ |

V poslednej časti workshopu boli predstavené sekvenčné vzory s medzerami a bez medzier. Úlohou designérov bolo určiť, aké typy sekvenčných vzorov (s medzerami a bez medzier) potrebujú pri jednotlivých výstupoch. Nasledujúca tabuľka ukazuje, ako sa rozhodli.

**Tabuľka č. 11:** Potrebné sekvenčné vzory u jednotlivých výstupoch workshopu (vlastné spracovanie)

| **Sekvencie** | **Typy potrebných vzorov** |
|---|---|
| Sekvencie na detaile produktu | vzory s medzerami |
| Sekvencie zobrazených stránok pred konverziou | vzory s medzerami |
| Sekvencie kontaktov so značkou | vzory s medzerami, vzory bez medzier |
| Sekvencie v odpade objednávkovým procesov | vzory s medzerami |
| Sekvencie pred použitím chatbota / | vzory s medzerami |

| | |
|---|---|
| zákazníckej podpory | |
| Sekvencie ako podklad ku kvalitatívnemu výskumu | vzory s medzerami, vzory bez medzier |
| Sekvencie obsahu nákupov | vzory s medzerami, vzory bez medzier |
| Sekvencie v používaní IS | vzory s medzerami |




**Obrázok č. 13:** Ukážka prezentácie o sekvenčných vzoroch s medzerami a bez medzier (vlastné spracovanie)

### 3.1.2 Požiadavky na algoritmus

Jedným z cieľom ideačného workshopu o využití sekvenčných vzorov v designe bolo definovanie požiadaviek na algoritmus pre dolovanie sekvenčných vzorov. Do úvahy budú brané tie výstupy, ktoré designéri označili prioritami vyššie ako 3.

Časť 4 workshopu, kde designéri definovali potrebné sekvenčné vzory u výstupoch (vzory v jednej návšteve alebo vzory naprieč návštevy), mala za úlohu definovať predpokladanú dĺžku vstupných sekvencií do algoritmu. Keďže v

relevantných výstupoch sa nachádzajú požiadavky, ktoré zahŕňajú sekvenciu zobrazených stránok naprieč návštevy, algoritmus má zvládať dlhé sekvencie ako vstupy.

Posledná časť workshopu mala definovať, či sú pre designérov potrebné vzory bez medzier alebo s medzerami. U relevantných výstupoch je zrejmé, že oba typy vzorov sú rovnako dôležité. Z toho vyplýva, že algoritmus má byť schopný generovať oba typy vzorov.

V kontexte sekvencií na webových stránkach a využitia znalostí zo sekvenčných vzorov je možné povedať, že pri prezentácií výstupov budú využívané najvyššie možné nadsekvencie s najväčšími podporami. Z toho dôvodu má byť algoritmu schopný pracovať s uzavretými sekvenčnými vzormi.

## 3.2 Navrhovaný algoritmus

Pri návrhu algoritmu bolo nutné dbať na používaní dostatočne rozsiahlej databázy s dlhými sekvenciami a veľkým množstvom jedinečných prvkov pre testovanie časového výkonu algoritmu a využívanej pamäte. Pre tieto účely boli používané 2 vstupné databázy. Prvá databáza obsahovala 3000 sekvencií s priemerným počtom udalostí 7 a 8 jedinečných prvkov. Táto databáza obsahovala kategorizované typy zobrazených podstránok webu. Druhá databáza obsahovala 85000 sekvencií s priemerných počtom udalostí 4 a až 50 jedinečných prvkov. Táto databáza obsahovala presné názvy URL webových stránok.

***Vstup:***

*sekvenčná databáza S*

*minimálna podpora min_sup*

*rozmedzie medzier M, N*

***Výstup:***

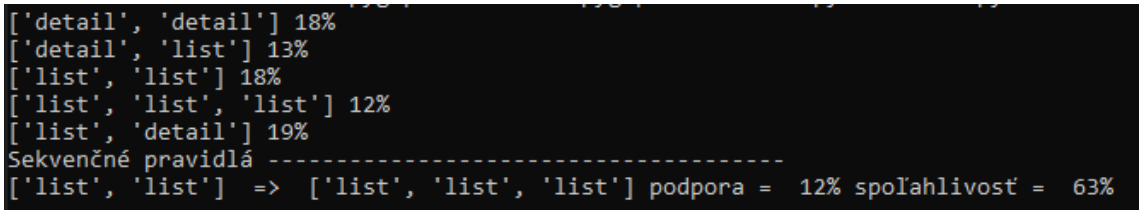
*množina sekvenčných vzorov*

***Metóda:***

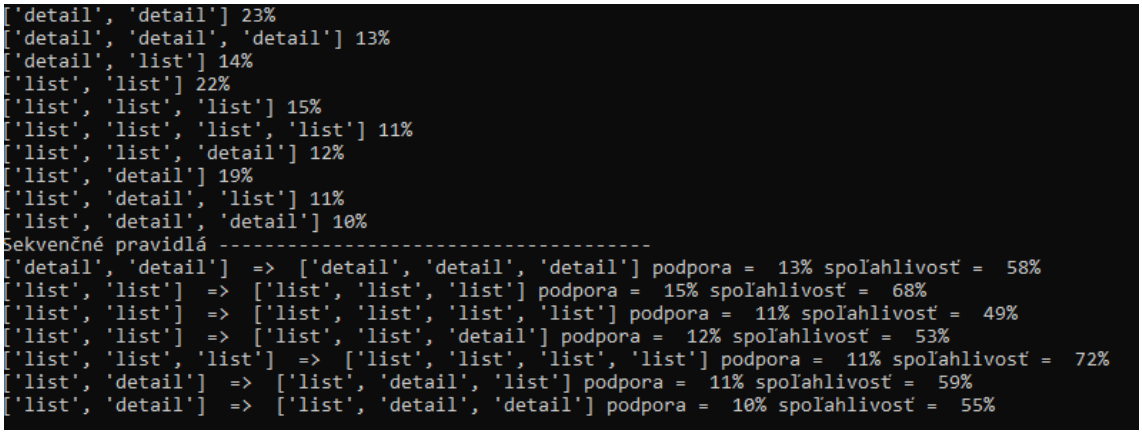
*Nájdi všetky frekventované jednoprvkové vzory L1*

```
for each item in L1:
    rastVzoru(item);
return;
Generuj z množiny sekvenčných vzorov sekvenčné pravidlá
function rastVzoru(prefix):
    prefix = predponový sekvenčný vzor
    Výstupom sú sekvenčné vzory s predponou prefix
    if(prefix obsahuje spätné rozšírenie):
        return
    if(prefix neobsahuje spätné rozšírenie ani dopredné rozšírenie):
        return prefix
    if(prefix obsahuje dopredné rozšírenie):
        Prehľadaj všetky priestory obsahujúce prefix a nájdi množinu
lokálne frekventovaných položiek L
        for each item in L:
            vytvor nový vzor Pnový = P + item
            rastVzoru(Pnový)
```

Algoritmus najskôr prehľadá vstupnú databázu a nájde množinu frekventovaných jednoprvkových vzorov, zavolá funkciu rastVzoru pre dolovanie uzavretých sekvenčných vzorov s medzerami veľkosti definovanými parametrami M a N pomocou vzoru P ako ich predpona. Funkcia rastVzoru skontroluje možné spätné rozšírenia vzoru P do vzdialeností M až N, potom skontroluje možné dopredné rozšírenia vzoru P do vzdialeností M až N a skontroluje uzavretosť rozšírení. Ak sú uzavreté, vráti sekvenčný vzor P. Ďalej prehľadá všetky dopredné priestory obsahujúce vzor P, nájde množinu lokálne frekventovaných položiek L, rozšíri vzor P o položky množiny lokálne frekventovaných položiek a znovu zavolá funkciu rastVzoru. Algoritmu končí v momente, kedy v dopredných priestoroch nenájde žiadne lokálne frekventované položky. Vypíše množinu vygenerovaných sekvenčných vzorov a nakoniec z konečnej množiny sekvenčných vzorov generuje a vypíše sekvenčné pravidlá. Forma prezentácie výsledkov v konzole je ukázaná pri výstupoch algoritmu so vstupnout databázou 1 na nasledujúcoch obrázkoch:




**Obrázok č. 14**: Ukážka výpisu sekvenčných vzorov pomocou navrhovaného algoritmu pri dolovaní uzavretých sekvenčných vzorov (vlastné spracovanie)




**Obrázok č. 15:** Ukážka výpisu sekvenčných vzorov pomocou navrhovaného algoritmu pri dolovaní neuzavretých sekvenčných vzorov (vlastné spracovanie)

Navrhovaný algoritmus vyzerá nasledovne:

```
class Algoritmus:
    def __init__(self, S, min_sup, m, n, min_pattern_length):
        "'S - vstupná sekvenčná databáza [[],[]]
        min_sup - minimálna podpora v % - ak 10% tak 0,1
        m,n - veľkosť medzier
        Ak bez medzier => m,n = 0,0 "'
        self.S = S
        self.min_sup = min_sup
        self.m = m
        self.n = n # IF -1, nájdi najväčšiu možnú
        self.min_pattern_length = min_pattern_length
        self.S_length = len(S)
        self.min_sup_count = round(self.S_length * self.min_sup)
        self.pocet_neuzavretych = 0
        self.pocet_uzavretych = 0
        self.pocet_prerezani = 0
        self.vysledky = [
        if self.n < 0:
            self.n = max(len(sekvencia) for sekvencia in S) - 2
            print(self.n)
```

```
    def run(self):
        l1_vzory = self.gen_l1_vzory()
        for vzor, sup, lok_S in l1_vzory:
            self.span(vzor, sup, lok_S)
        self.sekvencne_pravidla(self.vysledky)

    def sekvencne_pravidla(self, vzory):
        print('Sekvenčné pravidlá ----------------------------------------')
        sep = "."
        for i in range(len(vzory)):
            vzor = vzory[i]
            #string = sep.join(vzor[0])
            string = sep.join(str(x) for x in vzor[0])
            for n in range(len(vzory)):
                #test = sep.join(vzory[n][0])
                test = sep.join(str(x) for x in vzory[n][0])
                if vzor[1] > vzory[n][1] and len(vzor[0]) < len(vzory[n][0]) and test.find(string) == 0:
                    spolahlivost = "{:.0%}".format(vzory[n][1] / vzor[1])
                    print(vzor[0], " => ", vzory[n][0], "podpora = ", "{:.0%}".format(vzory[n][1] / self.S_length),
"spoľahlivosť = ", spolahlivost)

    def output(self, vzor, sup):
        if len(vzor) >= self.min_pattern_length:
            percentualna_podpora = "{:.0%}".format(sup / self.S_length)
            print(vzor, percentualna_podpora)
            self.vysledky.append([vzor,sup])

    def gen_l1_vzory(self): #Generuje množinu frekventovaných jednoprvkových vzorov
        S_dict = dict()
        for id_seq in range(self.S_length):
            seq = self.S[id_seq]
            for pozicia in range(len(seq)):
                if seq[pozicia] in S_dict:
                    S_dict[seq[pozicia]].append((id_seq, pozicia, pozicia))
                else:
                    S_dict[seq[pozicia]] = [(id_seq, pozicia, pozicia)]
        vzory = []
        for polozka, db in S_dict.items():
            podpora = len(set([i[0] for i in db]))
            if podpora >= self.min_sup_count:
                vzory.append(([polozka], podpora, db))
        return vzory

    def span(self, vzor, podpora, db):
        (spatne, prerezat) = self.spatne_rozsirenie(vzor,podpora,db)
        if prerezat:
            self.pocet_prerezani += 1
            return
        dopredne = self.dopredne_rozsirenie(vzor, podpora, db)
        if not(spatne or dopredne):
            self.pocet_uzavretych += 1
            self.output(vzor,podpora)
        else:
            self.pocet_neuzavretych += 1
        S_dict = dict()
        for (id_seq, zaciatok, koniec) in db:
            seq = self.S[id_seq]
            novy_zaciatok = koniec + 1 + self.m
```

```
            novy_koniec = koniec + 2 + self.n
            if novy_zaciatok >=len(seq):
                continue
            if novy_koniec > len(seq):
                novy_koniec = len(seq)
            for pozicia in range(novy_zaciatok, novy_koniec):
                if seq[pozicia] in S_dict:
                    S_dict[seq[pozicia]].append((id_seq,zaciatok,pozicia))
                else:
                    S_dict[seq[pozicia]] = [(id_seq,zaciatok,pozicia)]
        for polozka, nova_db in S_dict.items():
            podpora = len(set([i[0] for i in nova_db]))
            if podpora >= self.min_sup_count:
                #pridaj novy vzor
                novy_vzor = vzor[:]
                novy_vzor.append(polozka)
                self.span(novy_vzor,podpora,nova_db)

    def dopredne_rozsirenie(self, vzor, podpora, db):
        id_seqs = {}
        dopredne = False
        for(id_seq, zaciatok, koniec) in db:
            seq = self.S[id_seq]
            novy_zaciatok = koniec + 1 + self.m
            novy_koniec = koniec + 2 + self.n
            if novy_koniec >= len(seq):
                continue
            if novy_zaciatok > len(seq):
                novy_koniec = len(seq)
            for pozicia in range(novy_zaciatok,novy_koniec):
                if seq[pozicia] in id_seqs:
                    id_seqs[seq[pozicia]].append(id_seq)
                else:
                    id_seqs[seq[pozicia]] = [id_seq]
        for polozka, zoznam in id_seqs.items():
            podpora_vzoru = len(set(zoznam))
            if podpora_vzoru == podpora:
                dopredne = True
                break
        return dopredne

    def spatne_rozsirenie(self, vzor, podpora,db):
        id_seqs = {}
        spatne = False
        prerezat = False
        for(id_seq, zaciatok, koniec) in db:
            seq = self.S[id_seq]
            novy_zaciatok = zaciatok - self.n - 1
            novy_koniec = zaciatok - self.m
            if novy_koniec < 0:
                continue
            if novy_zaciatok < 0:
                novy_zaciatok = 0
            for pozicia in range(novy_zaciatok, novy_koniec):
                if seq[pozicia] in id_seqs:
                    id_seqs[seq[pozicia]].append(id_seq)
                else:
                    id_seqs[seq[pozicia]] = [id_seq]
```

```
for polozka, zoznam in id_seqs.items():
    podpora_vzoru = len(set(zoznam))
    podpora_zoznamu = len(zoznam)
    if podpora_zoznamu == len(db):
        prerezat = True
    if podpora_vzoru == podpora:
        spatne = True
    if spatne and prerezat:
        break
return (spatne, prerezat)
```

## 3.3 Ďalší postup

Algoritmus v aktuálnej podobe pracuje s databázou tvorenou vnoreným zoznamom (dátový typ list). Táto skutočnosť núti užívateľa algoritmu ručne pripraviť vstupnú databázu do takejto podoby. Ďalším postupom môže byť pridanie možnosti práce s dátami zo súboru typu CSV. Táto zmena ale stále nezmení nutnosť transformácie dát do podoby sekvenčnej databázy.

Algoritmus pracuje s parametrami pre minimálne a maximálne povolené medzery medzi udalosťami vzorov. Ak sa algoritmus zavolá s hodnotami pre minimálne a maximálne povolené medzery 1 a 3, algoritmus doluje a vypíše sekvenčné vzory bez informácie, ktorú podmienku spĺňa resp. aké veľké sú medzery medzi udalosťami daného sekvenčného vzoru.

Algoritmus prezentuje výsledné sekvenčné vzory textovou formou. Ďalšia možnosť prezentácie výsledkov je pomocou vizualizácie. Sekvenčné vzory je možné vizualizovať napríklad pomocou acyklických uzlových grafov, kde každý uzol predstavuje jednu udalosť vzoru.

Najpravdepodobnejšie využitia algoritmu budú:

1. Lokálne využívanie algoritmu pre dodanie reportov o užívateľskom chovaní na webových stránkach klientov firmy
2. Lokálne využívanie algoritmu na dodanie reportov o užívateľskom chovaní designérom
3. Automatizácia algoritmu na cloudových službách typu Google Colab [23] pre distribúciu výstupov klientom, prepojenie so

zdrojmi dát (Google Analytics 4, CRM systém) a transformácia dát do podoby vstupnej databáze algoritmu

# Záver

Cieľom tejto práce bolo navrhnúť a vyvinúť algoritmus pre dolovanie sekvenčných vzorov vhodný na Google Analytics 4 dáta a spĺňujúci požiadavky firemných procesov House of Řezáč.

Po zoznámení sa s problematikou sekvenčných vzorov a dostupných algoritmov pre ich dolovanie som sa zameral na analýzu dát služby Google Analytics 4. Následne som opísal výstupy ideačného workshopu vo firme, ktorá mala slúžiť ako podklad na technické požiadavky algoritmu a vytvoriť predstavy o využití v procesoch firmy House of Řezáč. Na základe týchto znalostí som implementoval techniku BIDE pre dolovanie uzavretých sekvenčných vzorov doplnenú o dolovanie sekvenčných vzorov bez medzier a s medzerami. Vydolované sekvenčné vzory sú prezentované v textovom formáte čitateľnom pre užívateľa, ktorý je zoznámený so sekvenčnými vzormi.

Nad rámec cieľa som implementoval generovanie sekvenčných pravidiel. Na základe nich je možné predpovedať budúce hodnoty. Pre testovanie algoritmu boli využité dva typy vstupných databáz, oba s odlišnými typmi dát. Overil som tak správnosť implementácie a časovú náročnosť algoritmu.

V poslednej časti práci som predstavil ďalšie postupy vylepšenia algoritmu a možnosti integrácie. Algoritmus bude slúžiť ako silný nástroj pre dodanie poznatkov o chovaní užívateľov k designerským procesom a klientom firmy.

# Zoznam použitej literatúry


[1] LACKO, L. Datové sklady, analýza OLAP a dolování dat. 1 vyd. Brno: Computer Press, 2003. 488 s. ISBN 80-7226-969-0.

[2] HAN, J. a M. KAMBER. Data Mining: Concepts and Techniques. Second Edition. Mor-gan Kaufmann Publishers, 2006, 770 s. ISBN 1-55860-901-6.

[3] ZENDULKA, J., BARTÍK, V., LUKÁŠ, R. a I. RUDOLFOVÁ. Získávání znalostí z databází. Studijní opora, 2006, Fakulta informačních technologií VUT v Brně.

[4] Nizar R. Mabroukeh and C. I. Ezeife. 2010. A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv.* 43, 1, Článok 3 (November 2010), 41 strán. Dostupné z:https://doi.org/10.1145/1824795.1824798

[5] R.Agrawal and R.Srikant. Mining sequential patterns. Proceedings of the Eleventh International Conference on Data Engineering, 1995.

[6] AGRAWAL, R. a R. SRIKANT. Mining Sequential Patterns: Generalizations and Per-formance Improvements. In Proceedings of the 5th International Conference on Exten-ding Database Technology: Advances in Database Technology. London, UK: Springer-Verlag, 1996. s. 3-17. ISBN 3-540-61057-X.

[7] J.Ayres, J.Flannick, J.Gehrke and T. Yiu. Sequential Pattern Mining Using a Bitmap Representation, Proceedings of Conference on Knowledge Discovery and Data Mining, pp. 429–435, 2002.

[8] J.Wang, J.Han.BIDE: Efficient mining of frequent closed sequences. In Proc. of 2004 Int. Conf. on Data Eng. Apr. 2004, Boston, MA. 79–90.

[9] LUTZ, M. Learning Python. 5th ed. Sebastopol: O'Reilly Media, 2013. 1648 p. ISBN 978-1-449-35573-9.

[10] Nastavení služby Analytics pro web (Universal Analytics) - Nápověda Analytics. *Google Help* [online]. Copyright © 2021 Google [cit. 01.05.2021]. Dostupné z: https://support.google.com/analytics/answer/10269537?hl=cs

[11] Admin page - Analytics Help. *Google Help* [online]. Copyright ©2021 Google [cit. 01.05.2021]. Dostupné z: https://support.google.com/analytics/answer/6132368

[12] [GA4] Nastavení Analytics pro web nebo aplikaci - Nápověda Analytics. *Google Help* [online]. Copyright © 2021 Google [cit. 01.05.2021]. Dostupné z:

https://support.google.com/analytics/answer/9304153?hl=cs&ref_topic=9303319#zippy=%2Cp%C5%99id%C3%A1n%C3%AD-glob%C3%A1ln%C3%AD-zna%C4%8Dky-webu-p%C5%99%C3%ADmo-do-webov%C3%BDch-str%C3%A1nek

[13] HTTP a WWW servery. *Faculty of Informatics Masaryk University* [online]. Dostupné z: https://www.fi.muni.cz/~kas/pv090/referaty/2015-jaro/http.html

[14] Add the Google Analytics tag for Google Analytics 4 properties to your site with gtag.js. *Google Developers* [online]. Dostupné z: https://developers.google.com/analytics/devguides/collection/ga4

[15] Tag Manager and gtag.js - Tag Manager Help. *Google Help* [online]. Copyright ©2021 Google [cit. 01.05.2021]. Dostupné z: https://support.google.com/tagmanager/answer/7582054

[16] [GA4] Automatically collected events - Analytics Help. *Google Help* [online]. Copyright ©2021 Google [cit. 01.05.2021]. Dostupné z: https://support.google.com/analytics/answer/9234069

[17] Universal Analytics versus Google Analytics 4 data - Analytics Help. *Google Help* [online]. Copyright ©2021 Google [cit. 01.05.2021]. Dostupné z: https://support.google.com/analytics/answer/9964640?utm_source=advocacy&utm_medium=social&utm_campaign=ga4&fbclid=IwAR0za1nPOn1safMJopfNPM8B-H6vJerjcm2gOzoeVqprIm14dfCA1gXNwYM

[18] [GA4] Recommended events - Analytics Help. *Google Help* [online]. Copyright ©2021 Google [cit. 01.05.2021]. Dostupné z: https://support.google.com/analytics/answer/9267735

[19] [GA4] Session calculation - Analytics Help. *Google Help* [online]. Copyright ©2021 Google [cit. 02.05.2021]. Dostupné z: https://support.google.com/analytics/answer/9191807?hl=en

[20] Pricing | BigQuery | Google Cloud. *Cloud Computing Services | Google Cloud* [online]. Dostupné z: https://cloud.google.com/bigquery/pricing

[21] Analytics Data API Overview | Google Analytics Data API. *Google Developers* [online]. Dostupné z: https://developers.google.com/analytics/devguides/reporting/data/v1

[22] An Online Visual Collaboration Platform for Teamwork | Miro. *An Online Visual Collaboration Platform for Teamwork | Miro* [online]. Copyright © 2021 [cit. 02.05.2021]. Dostupné z: https://miro.com/

[23] Google Colaboratory. *Google Colaboratory* [online]. Dostupné z: https://colab.research.google.com/notebooks/intro.ipynb

# Zoznam použitých obrázkov

# Zoznam použitých tabuliek

# Zoznam použitých skratiek a symbolov

| | |
|---|---|
| GA4 | Google Analytics 4 |
| SQL | Structured Query Language – dotazovací jazyk používaný pri práci s databázami |
| OLAP | Online Analytical Processing |
| URL | Uniform Resource Locator – reťazec znakov určujúci umiestnenie zdrojov informácií na internete |
| JSON | Javascript Object Notation – spôsob zápisu dát určený pre prenos dát |
| CSV | Comma Separated Values – jednoduchý súborový formát určený na výmenu tabuľkových dát |
| CRM | Customer Relationship Management – zákaznícky orientovaný management |