Fakulta informačních technologií VUT v Brně

Získávání znalostí z databáze transakcí studentského klubu

Řešení

Ondřej Ondryáš, Filip Čižmár

10. prosince 2023

Obsah

1 Datová sada

Úlohy budou řešeny nad daty ze skutečného provozu studentského klubu U Kachničky na FIT VUT. Všichni zákazníci klubu se musí registrovat v informačním systému, při každém nákupu se prokazují členským průkazem a objednávky jsou spárovány s jejich profily, data tedy přesně zachycují jejich chování.

Databáze obsahuje data o transakcích v období cca červen 2021 až září 2023. Sledovanými typy transakcí jsou nákup, vydání zdarma, poskytnutí dobrovolného členského příspěvku, inventura zásob a naskladnění zásob. Dále jsou v ní reprezentovány související informace o uživatelích, nabízených produktech a jejich štítcích. Některé entity typu produkt reprezentují neprodejné skladové položky, ze kterých se skládají jiné entity typu produkt. V databázi jsou proto oddělena data zachycující pohyby prodejných produktů

a pohyby všech skladových položek. Schéma databáze je zobrazeno na diagramu v příloze A.

Data jsou uložena v relačním systému PostgreSQL, pro účely získávání znalostí tedy bude patrně nutné je extrahovat a transformovat do vhodné podoby, to by však mělo být otázkou několika SQL dotazů.

1.1 Anonymizace a úprava dat

Použitá databáze vznikla kopií produkční databáze dne 6. 10. 2023. Z kopie byly následně odstraněny některé relace a atributy, které v zamýšlených úlohách nemají význam. Odstraněny byly především informace o cenách a finančních tocích. Osobní údaje uživatelů byly nahrazeny charakteristikami, které neumožňují identifikaci konkrétní osoby.

2 Úlohy získávání znalostí

Cílem práce je vytvořit modely pro odhad spotřeby zásob a pro odhad počtu návštěvníků v klubu, a to za účelem optimalizace objednávek zásob a plánování směn obsluhy. Dalším cílem je rozdělit zákazníky do předem neznámých skupin podle chování při nakupování, a to za účelem identifikace typů zákazníků a vylepšování nabídky. Poslední úlohou je výpočet frekventovaných a sekvenčních vzorů za účelem vylepšení organizace uložení zásob v klubu.

2.1 Plánování směn obsluhy

Klub obsluhují dobrovolníci bez nároku na odměnu, typicky je otevřeno dvakrát týdně po dobu 6 hodin, obsluhu je tedy možné (a nutné) plánovat vcelku dynamicky, po hodinách. Nutné personální zdroje jsou přitom závislé na počtu návštěvníků – na začátku semestru může být potřeba i dvakrát více obsluhujících než na jeho konci.

Počet návštěvníků je závislý na postupu semestru, ale značnou měrou i na termínu otevření (obvykle je otevřeno v pondělky a ve středy, kdy bývá návštěvnost větší). Pro plánování směn je nutné odhadovat návštěvnost po jednotlivých hodinách. Zde je tedy opět žádoucí vytvořit regresní model, který na základě data (čísla týdne semestru a dne v týdnu) a hodiny určí přibližný počet návštěvníků v klubu. Z dat bude nutné odfiltrovat speciální akce, zajímavé jsou zde pouze obvyklé otvírací hodiny.

Zajímavý by také mohl být model, který na základě zadaných atributů návštěvníka odhadne, jak dlouho se zdrží. Tato informace pro nás však není stěžejní.

2.2 Optimalizace objednávek zásob

Do klubu se pravidelně nakupují v zásadě tři kategorie zásob: výčepní nápoje (sudy); drobné občerstvení a suroviny pro přípravu toastů. Spotřeba všech tří se razantně mění v závislosti na postupu semestru: přirozeně zákazníků ubývá se zvyšováním studijní zátěže; tyto změny však nejsou lineární a spotřeba jednotlivých typů zásob se vyvíjí různým způsobem. Jednotlivé typy zásob navíc mají poněkud jiný charakter nákupu: sudy se objednávají cca jednou týdne, drobné občerstvení jednou za měsíc a suroviny bezprostředně před otevřením. V současnosti se zásoby nakupují čistě na základě intuice, což není optimální.

Je tedy žádoucí vytvořit regresní model, který na základě data (resp. kombinace čísla týdne semestru a dne v týdnu) a typu zásob určí vhodný objem k nakoupení.

2.3 Segmentizace zákazníků

Nabídka v klubu je dosud také sestavována pouze intuitivně (zejm. v kategorii drobného občerstvení), přitom je zjevné, že je návštěvníky možné rozdělit do jistých skupin podle charakteru jejich nákupů – skladby objednávek (množství a typu produktů) a jejich frekvence. Zjištění takovýchto skupin a jejich zastoupení mezi návštěvníky nám pomůže určit, ve kterých typech produktů má smysl nabídku rozšiřovat, a které jsou naopak spíše okrajové.

2.4 Frekventované a sekvenční vzory

Informace o konkrétních produktech (nebo úzkých množinách produktů), které návštěvníci nakupují společně, bychom využili k lepšímu uspořádání zásob v prostoru baru a ve skladech – společně zakupované položky by měly být blíže u sebe.

3 Postup a výsledky

3.1 Plánování směn obsluhy

Zadáním úlohy je odhadovat počet návštěvníků v klubu po hodinách. Takový odhad je s dostupnými daty možné provést na základě počtu unikátních nakupujících zákazníků za hodinu. Pracovat je zde možné přímo s tabulkou operací, ze které jsou vybrány pouze záznamy typu nákup a vydání zdarma:

```
select o.id        transaction_id,
       o.user_id   user_id,
       o.time      datetime
from operation o
where o.type in (0, 1)
```

Zdrojový kód 1: SQL dotaz výběru operací

Transformace

Pro zavedení regresního modelu je nutné sadu transformovat do tvaru vysvětlující proměnné $\rightarrow$ závisle proměnná. Vysvětlujícími proměnnými zde jsou dle zadání číslo týdne semestru, den v týdnu a hodina dne. Závisle proměnný je pak průměrný počet zákazníků.

V této úloze je cílem modelovat pouze běžný chod klubu, neboť speciální akce mohou údaje o návštěvnosti podstatně deformovat. V datové sadě tedy musí být vyfiltrovány pouze pondělky a středy v časech od 15 hodin (včetně) do 22 hodin (mimo).

Transformace a filtrace byla provedena v prostředí RapidMiner subprocesem naznačeným na obrázku v příloze F. Z atributu `datetime` bylo extrahováno číslo dne v týdnu, podle kterého byly záznamy filtrovány; podobně pro hodiny ve dni. Čísla dne v týdnu byla pro použití v regresním modelu namapována na číselné hodnoty -1 (pondělí) a 1 (středa). Číslo týdne v semestru bylo vypočítáno jako vzdálenost od nejbližšího předchozího data počátku semestru pomocí operátoru *Execute Script* s vlastním kódem uvedeným v příloze B. Atribut `datetime` byl upraven odstraněním časové složky, poté byly záznamy seskupeny podle klíče (datum, den, hodina, číslo týdne), přičemž byly agregovány počty unikátních `user_id`. Druhá agregace proběhla podle klíče (den, hodina, číslo týdne), hodnoty počtů byly zprůměrovány.

Výsledná datová sada pro trénování regresního modelu má atributy:

- `datetime_day` $\in \{-1, 1\}$ – pondělí / středa

- `datetime_hour` $\in \{15, 16, \ldots, 21\}$ – hodina ve dne
- `sem_week` $\in \{1, 2, \ldots, 13\}$ – číslo týdne v semestru
- `avg_unique_customers` $\in (3{,}33;\ 55{,}75)$ – průměrný počet unikátních zákazníků (závisle proměnná)

Modelování

Jde o prediktivní úlohu, která má svým charakterem nejblíže k regresi: modelovaná proměnná je spojitá, nicméně vysvětlující proměnné jsou diskrétní. RapidMiner nabízí celou řadu regresních modelů, které modelují funkce, a umožňuje v regresních úlohách použít i méně typické prediktivní modely založené na rozhodovacích stromech nebo SVM. Při hledání vhodného modelu jsme provedli experimenty s některými operátory z kategorie Modeling/Predictive/Functions, dále jsme na datové sadě vyzkoušeli některé předpřipravené procesy dostupné ve funkci *Auto Model*.

Nejprve byly použity operátory Linear Regression, Polynomial Regression, Local Polynomial Regression (LPR), Gaussian Process (GP) a Relevance Vector Machine (RVM), a to ve výchozím nastavení. Každý z nich byl natrénován na 70 % dat, zbylých 30 % bylo použito pro testování. Sledovanými parametry byla odmocnina střední kvadratické chyby (RMSE), průměrná absolutní chyba, průměrná relativní chyba a korelace skutečné hodnoty s predikovanou. Na takto natrénovaném modelu bylo dále vyhodnocováno, pro jakou kombinaci hodnot vrátí maximální odhad. Použitý proces je naznačen na obrázku v příloze G, výsledky měření jsou v tabulce 1. Naznačují, že klasické regresní metody poskytují horší predikce.

	RMSE	Abs. chyba	Rel. chyba [%]	Kor.	D	H	SW	Pred
Linear	10,2	8,2±6	69±129	0,557	1	21	1	42,89
Polynomial	11,4	8,8±7,3	60±101	0,329	1	21	1	81,14
LPR	8,5	6,2±5,9	36±51	0,771	1	20	4	55,75
GP	6,7	5,1±4,4	25±23	0,805	1	19	4	54
RVM	16,5	13,4±9,6	53±21	0,737	1	19	4	46

Tabulka 1: Výsledky modelů ve výchozím nastavení, testovaných na 30 % dat. Sloupce D, H, SW označují vstupní hodnoty dne/hodiny/čísla týdne, pro které model vrací největší predikci. Ta je uvedena ve sloupci Pred.

Pro další experimenty byly vybrány modely LPR, GP a RVM. V každém z nich bylo identifikováno několik vhodných parametrů, nad kterými byl proveden *grid search* – trénování modelu na všech permutacích nakonfigurovaných hodnot parametrů za účelem nalezení jejich nejlepší kombinace. Metrikou správnosti modelu zde byla zvolena průměrná absolutní chyba, nicméně i při minimalizaci RMSE byly nalezené parametry obdobné. Naměřené hodnoty jsou v tabulce 2. Optimalizační proces je naznačen v příloze H.

	RMSE	Abs. chyba	Rel. chyba [%]	Kor.	D	H	SW	Pred
LPR*	5,6	4,3±3,6	20±20	0,859	0	21	5	613,65
					1*	19	5	50,16
GP	4,9	3,7±3,2	18±19	0,888	1	19	4	55,67
RVM	4,9	3,9±3	19±19	0,886	1	20	4	55,75

Tabulka 2: Výsledky modelů s optimalizovanými parametry. V případě LPR byl při hledání maximální predikce nejprve den určen na neplatnou hodnotu 0, platné hodnoty byly získány po zafixování dne na hodnotu 1.

Oproti původním výsledkům je zde patrné značné zlepšení ve všech sledovaných parametrech, jejichž hodnoty se napříč modely podobají. Zajímavé je chování modelu LPR v neplatné hodnotě dne 0, kde model vrací o řád vyšší hodnoty, na rozdíl od ostatních modelů.

RapidMiner nabízí předpřipravené procesy pro regresi s využitím modelů typu Generalized Linear Model (GLM), Decision Tree (DT), Random Forest (RF), Gradient Boosted Trees (GBT) a Support Vector Machine (SVM). Vzhledem ke značně omezeným doménám atributů může být použití stromových algoritmů vhodné a výpočetně nenáročné. Tabulka 3 ukazuje sledované parametry při použití těchto procesů s povolením automatické optimalizace. Protože procesy využívají jiný způsob validace modelu, provedli jsme také měření výše vytvořených modelů vložených do předpřipravených procesů. V tabulce jsou proto uvedeny také výsledky metod LPR, GP a RVM.

	RMSE	Abs. chyba	Kor.	D	H	SW	Pred
GLM	9,5	7,4±0,9	0,578	0,952	20,995	1,059	44,33
DT	5,1	4,3±0,5	0,890	0,758	20,165	5,084	55,63
RF	4,8	3,8±0,7	0,930	0,045	19,604	4,412	46,14
GBT	5,5	4,7±0,5	0,932	0,758	18,293	3,594	40,51
SVM	5,1	4,0±0,9	0,865	0,985	18,890	5,179	48,47
LPR*	6,6	4,9±1,1	0,816	0,035	15,323	4,303	984,23
				1*	18,839	4,824	50,37
GP	8,5	5,6±1,9	0,777	0,999	19,158	3,695	59,92
RVM	4,8	4,0±0,6	0,880	0,994	19,633	3,691	56,45

Tabulka 3: Výsledky předpřipravených modelů a výše diskutovaných modelů použitých v obdobném procesu.

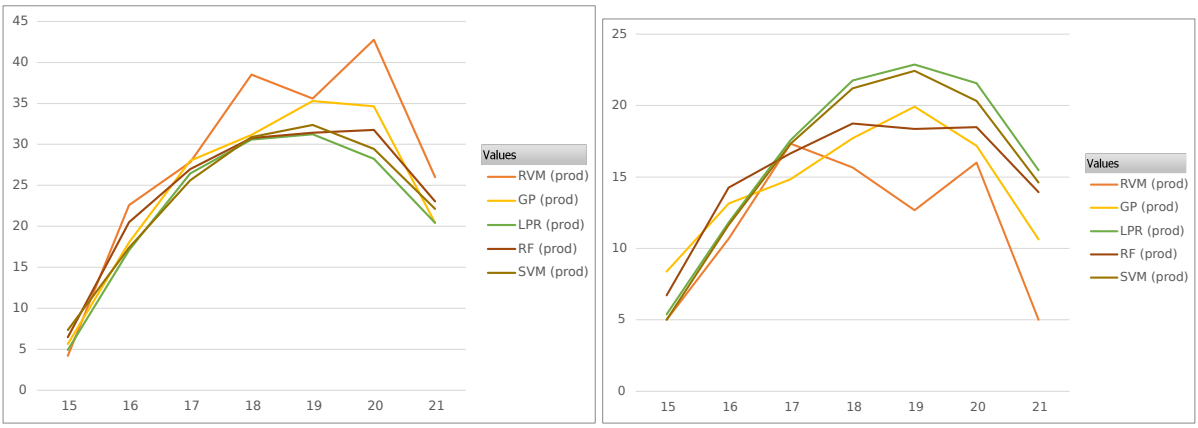
Nejlepší výsledky s ohledem na RMSE i průměrnou absolutní chybu poskytují modely RF a SVM, případně DT. V tomto případě simulátor modelu interpretoval proměnné D, H a SW jako reálné a při hledání největší predikované hodnoty je zde patrné, že některé modely mají tendence poskytovat maxima v nesmyslných hodnotách. Blízko k validní hodnotě 1 určily svou největší hodnotu modely GLM a SVM, hůře zde vychází modely GT a GBT. Model RF sice dosáhl maxima v D = 0,045, ukázalo se však, že při nastavení na 1 se predikce nemění, zřejmě zde tedy začíná interval příslušného stromu. U modelu LPR se projevila stejná vlastnost jako v předchozím měření.

Interpretace výsledků

Pro praktičtější vyhodnocení jednotlivých modelů byla vygenerována datová sada všech možných kombinací vstupních parametrů, která poté byla vyhodnocena uloženými modely RF, SVM, LPR, GP a RVM (viz schéma procesu v příloze I). Obrázek 1 ukazuje průběhy predikovaných hodnot napříč jedním dnem pro čtyři různá nastavení parametrů D a SW.

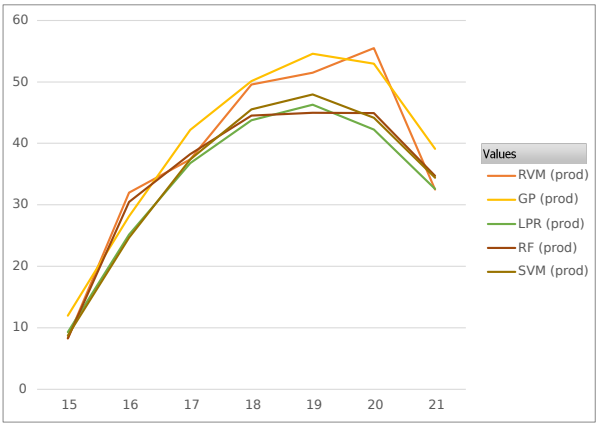
Podle zkušeností je 3. týden typicky jedním z „nejsilnějších“, v 11. týdnu bývá naopak návštěvnost nízká. V pondělí je typicky návštěvnost výrazně nižší než ve středy. Grafy ukazují, že všechny modely tuto skutečnost spolehlivě zachytily. Za povšimnutí stojí podobnosti chování jednotlivých modelů. Modely LPR a SVM ve všech případech mají velice podobné chování, tvar křivky naznačuje inklinaci k reprezentaci normálního nebo obdobného rozdělení. Modely GP a RVM mají větší tendenci zabíhat k extrémům, tvar křivky je nepravidelný. Tyto vlastnosti mohou signalizovat overfitting, v kontextu zkušeností z klubu však není možné tvrdit, že by ukazované chování nemohlo odpovídat realitě.

Pro účely plánování směn je vhodné zobrazit také chování modelu napříč semestrem. Obrázek 2 zachycuje predikce pro středu, 18 hodin (obecně jeden z nejfrekventovanějších

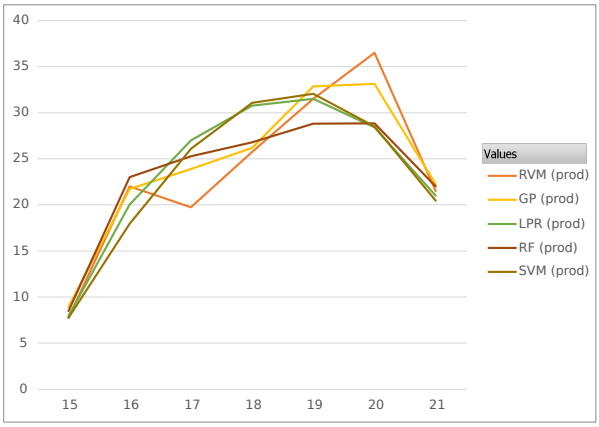


(a) Pondělí, 3. týden

(b) Pondělí, 11. týden



(c) Středa, 3. týden



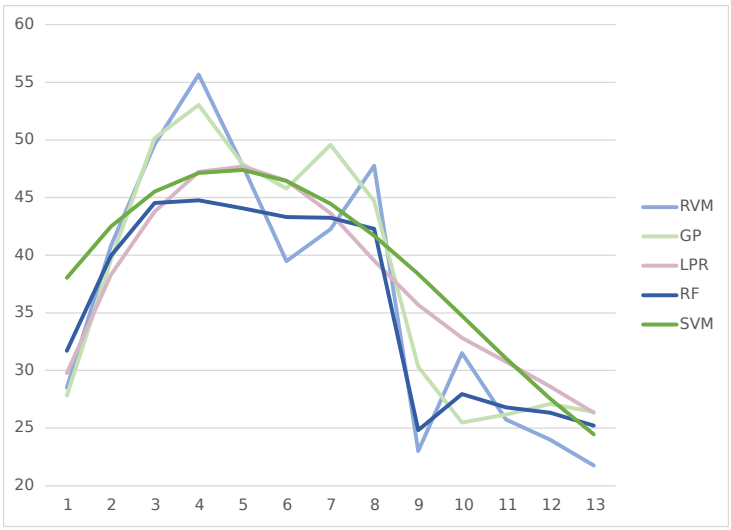
(d) Středa, 11. týden

Obrázek 1: Predikce jednotlivých modelů natrénovaných na celé datové sadě. Osa *x* označuje hodinu, osa *y* predikovaný počet zákazníků.

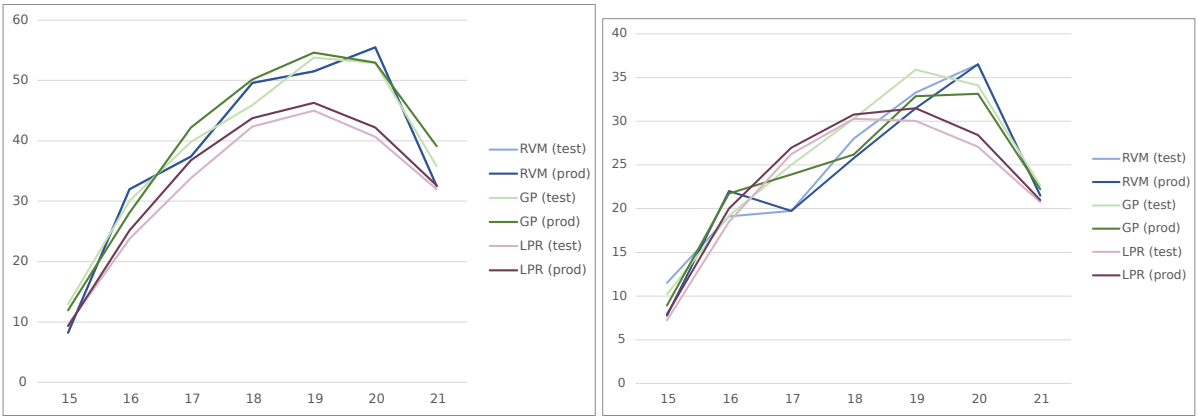
časů otvírací doby) v jednotlivých týdnech semestru. I pro tuto vstupní proměnnou platí obdobné závěry jako v předchozím případě. Modely LPR a SVM mají výrazně vyhlazený průběh podobný normálnímu rozdělení. Modely RVM a GP naopak mnohem přesněji odpovídají intuici z provozu klubu: ve 3.–5. týdnu je návštěvnost nejvyšší, v 5.–7. klesá z důvodu půlsemestrálních zkoušek a prvních projektů, následuje několik volnějších týdnů a rychlý propad především z důvodu práce na projektech.

Zejména v případě modelů RVM a GP se dá polemizovat, zda nejsou zasaženy overfittingem, zároveň však jejich průběhy lépe odpovídají zkušenostem z provozu. Za prakticky použitelný bychom označili zejména model GP, který nepodává tak extrémní výsledky jako RVM, zároveň však dobře reflektuje křivku provozu napříč semestrem. U modelů samotných LPR a SVM pro účely plánování směn hrozí naopak příliš silná generalizace. Model RF se svým chováním vymyká – zejm. v kladných špičkách patrně podhodnocuje.

Na závěr je vhodné podotknout, že pro přesnější regresi patrně není vstupní datová sada dostatečně velká. Ačkoliv celá databáze obsahuje před 70 tisíc transakcí, prodejních je z toho pouze 37 tisíc a výsledkem série filtrací a agregací popsaných výše je pouze 182 záznamů – což je vzhledem k frekvenci otvírání klubu očekávatelné. Pro získání přesnějších modelů by bylo možné sadu rozšířit o historická data, není však jasné, zda jsou tato ještě někde archivovaná.



Obrázek 2: Predikce jednotlivých modelů pro středu 18 hodin napříč jednotlivými týdny semestru. Osa *x* označuje týden semestru, osa *y* predikovaný počet zákazníků.



(a) Středa, 3. týden

(b) Středa, 11. týden

Obrázek 3: Predikce jednotlivých modelů trénovaných na testovací sadě (70 %) a na celé sadě. Osa *x* označuje hodinu, osa *y* predikovaný počet zákazníků.

3.2 Optimalizace objednávek zásob

Při tomto úkolu je cílem vytvořit model, který je schopen predikovat spotřebu daného sortimentu podle dne v týdnu a týdne semestru. Jmenovitě je sledováno zboží patřící do kategorií výčepních nápojů, drobného občerstvení (rozděleného na slané a sladké) a surovin na toasty.

Úloha tedy pracuje hlavně s tabulkou pohybu zásob, ale také s tabulkami operací a kategorií. SQL dotaz pro získání pohybu zásob slaného drobného občerstvení může vypadat například následovně:

```sql
select sf.id,
       sf.operation_id,
       sf.article_id,
       sf.amount,
       o.time,
       extract(week from o.time) as week_number,
from stockflow sf
join operation o on sf.operation_id = o.id
where sf.article_id in (30, 42, 47, 48, 49, 76, 92)
    and o.type IN (0,1)
```

Zdrojový kód 2: SQL dotaz výběru pohybu zásob slaného drobného občersvení

V tomto případě jsou jednotlivé prodejní položky vyhledány podle konkrétního *id*, jelikož neexistuje kategorie slaného občerstvení, ale v jiných případech (např. výčepní nápoje) lze tabulku pohybu zásob ještě spojit s tabulkou kategorií a tabulkou mapování kategorií na produkty a následně vybírat produkty podle kategorie.

Atribut *week_number* je do selekce přidán pro zjednodušení práce v nástroji RapidMiner.

Následná transformace dat do podoby potřebné pro dolování probíhá v samotném procesu, který je v přílohách C, D a E.

Proces začíná vytvořením nového atributu typu *time* z *datetime* atributu a následnou filtrací záznamů, tak aby zůstaly jen záznamy mezi 15 hodin (včetně) a 22 hodin (mimo). Zároveň vzniká nový atribut s informací o dni v týdnu ([Mon, Tue, Wed...]). V příloze D jsou zase filtrovány záznamy, které nepatří do řádného trvání školního semestru. Data jsou agregována pro daný den v týdnu a týden v roce. Atribut týdne v roce je dále posunut do rozmezí [1, 13]. V příloze E jsou datasety semestrů spojeny do jednoho datasetu a atributy dne v týdnu přeměněny na numerické hodnoty v rozmezí [1, 7]. Následně jsou odfiltrovány speciální akce, které nepatří do normálního chodu klubu (pondělky a středy) a dny v týdnu namapovány na hodnoty -1 pro pondělí a 1 pro středu. Výsledná datová sada vypadá následovně:

- `datetime_day` $\in \{-1, 1\}$ – pondělí / středa
- `sem_week` $\in \{1, 2, \ldots, 13\}$ – číslo týdne v semestru
- `amount_sum` $\in (1;\ 14)$ – suma spotřeby produktu (závislá proměnná), v tomto případě jsou hodnoty minima a maxima dány pro slané občerstvení

Pro modelování je použito rozdělení na trénovací/testovací sadu v hodnote 70%/30%. Modelovaný úkol je podstatou velmi podobný úloze plánování změn obsluhy s tím rozdílem, že cílem je predikovat spotřebu na celý den, nikoli v dané hodině. Detaily přístupu k modelování byly blíže popsány v sekci plánování obsluhy.

Pro účely tohoto úkolu byly provedeny experimenty s lineární, polynomiální, lokální polynomiální regresí a Gradient Boosted Tree.

Z tabulky 4 lze vidět, že přesnost výsledných modelů je méně než žádoucí. Toto může být zapříčiněno poměrně malým počtem vstupních dat pro modely (4 semestry x 13 týdnů x 2 dny + 3 týdny na začátku 5. semestru x 2 dny). Z tohoto důvodu byly použity předpřipravené procesy nástroje RapidMiner pro predikci. Jelikož celkový počet transformovaných dat byl 96 a RapidMiner pro funkci Auto Model vyžaduje četnost alespoň 100, z dat byla vypuštěna filtrace speciálních akcí. Výsledky předpřipravených procesů jsou následující:

	RMSE	Abs. chyba	Rel. chyba [%]	Kor.
Linear	2,728	2,111±1,728	145,25±146,15	0,460
Polynomial	3,755	2,852±2,442	97,55±126,32	0,528
LPR	5,166	3,560±3,743	104,00±124,71	0,589
GBT	4,102	3,327±2,399	113,71±117,40	0,466

Tabulka 4: Výsledky modelů testovaných na 30% dat.

	RMSE	Abs. chyba	Rel. chyba [%]	Kor.
GLM	4,708	3,981±0,450	57,8±6,9	0,033
Deep Learning	4,572	3,755±0,589	56,5±6,3	0,071
Decision Tree	4,548	3,226±1,157	45,4±7,4	0,561
Random Forest	4,733	3,357±1,044	45,4±6,4	0,505
GBT	4,743	3,463±1,057	49,5±6,1	0,5
SVM	5,041	3,595±1,149	46,2±5,0	0,408

Tabulka 5: Výsledky modelů vytvořených funkcí Auto Model v nástroji RapidMiner.

Přestože absolutní a relativní chyby těchto procesů již vypadají lépe, většina těchto modelů predikuje hodnoty v úzké blízkosti 5, čímž se snižuje absolutní chyba, ale ignorují se odlehlé hodnoty, které se v datech vyskytují. Takový model je v praxi nepoužitelný.

Posledním pokusem bylo zanedbání rozdělení *datetime* atributu na semestr a den a použití samotné *date* hodnoty. Výsledky modelů lze vidět na tabulce 6.

	RMSE	Abs. chyba	Rel. chyba [%]	Kor.
GLM	4,463	3,488±1,154	46,7±4,2	0,57
Deep Learning	4,508	3,404±1,043	45,3±4,3	0,559
Decision Tree	4,362	3,317±1,629	45,6±8,6	0,349
Random Forest	4,717	3,656±1,126	45,8±1,7	0,579
GBT	4,807	3,561±1,111	43,9±2,7	0,503
SVM	4,041	3,171±1,01	52,3±9,2	0,517

Tabulka 6: Výsledky modelů při zanedbání rozdělení atributu *datetime* na položky týdne semestru a dne.

Pro všechny modely v tabulkách byly použity záznamy o slaném drobném občerstvení. Modely pro ostatní kategorie produktů byly vytvořeny obdobně. Jelikož četnost dat je i v těchto případech podobná a proces tvorby modelu je stejný, výsledky modelů jsou obdobně nedostačující. I přesto, že takový model by mohl být pro spolek velmi užitečný, pro použití v praxi by bylo nutné úkol dále optimalizovat, případně změnit přístup k úkolu od základu.

3.3 Segmentizace zákazníků

Pro segmentizaci zákazníků byla spojena data z tabulky operací s údaji o pohlaví zákazníka, příslušnosti ke studentské unii a ke škole. Byly použity pouze operace typu nákupu zákazníkem. Konkrétní produkt byl nahrazen jeho kategorií, přesněji se jedná o tyto kategorie: *Cider*, *Jídlo*, *Káva*, *Nealko*, *Palačinky*, *Pivo*, *Toasty* a *Víno*.

Jelikož všechny hodnoty jsou nominální, pro převod na numerické hodnoty byl použit *dummy coding*, čímž vznikl nový atribut pro každou nominální hodnotu. Jednotlivé nákupy byly následně agregovány pro každého uživatele tak, aby výsledkem pro danou hodnotu

atributu byla celková suma nákupů dané položky. Toto platí pouze pro jednotlivé kategorie produktů vyskutujících se v datech.

Data byla následně normalizována *Z-normalizací* a použita v algoritmu *K-Means*. Počet clusterů je 6. Blokové schéma úlohy z nástroje RapidMiner lze vidět v příloze J a graf centroidů v příloze K.

Zajímavým je např. *Cluster 0*, který negativně koreluje zákazníky ženského pohlaví s patřičností k *FIT*, ale pozitivně s patřičností k *VUT*. Zároveň *Cluster 3*, který ukazuje, že studenti, kteří nepatří pod *FIT* nebo *VUT*, mírně preferují toasty a pivo oproti ostatním produktům.

3.4 Frekventované a sekvenční vzory

V této úloze bylo nutné pracovat s datovou sadou, která obsahuje spojení transakcí a nakoupených produktů v transakci. SQL dotaz pro vytvoření této sady je uveden v příloze L. Dotaz vytvoří i několik pomocných atributů pro snadnější zařazení produktů do významných skupin. Každý záznam obsahuje atributy:

- `transaction_id` – ID transakce
- `article_id` – ID produktu
- `amount` – počet kusů produktu v transakci
- `article_name` – název produktu
- `user_id` – ID uživatele
- `datetime` – datum a čas transakce
- `is_tap_item` – příznak – produkt je nápoj na výčepu
- `is_returnable_bottle` – příznak – produkt je vratná lahev
- `is_prepared_food_item` – příznak – produkt je připravované občerstvení (toasty, palačinky apod.)

Pro účely hledání frekventovaných množin uvedený SQL dotaz ponechává pouze transakce, ve kterých byl zakoupen více než jeden produkt, neboť jednoproduktové transakce datové sadě značně dominují, což vede k nutnosti příliš snižovat minimální podporu. Pro účely hledání sekvenčních vzorů je příslušná klauzule where (od ř. 23) odstraněna, zde má smysl uvažovat všechny transakce.

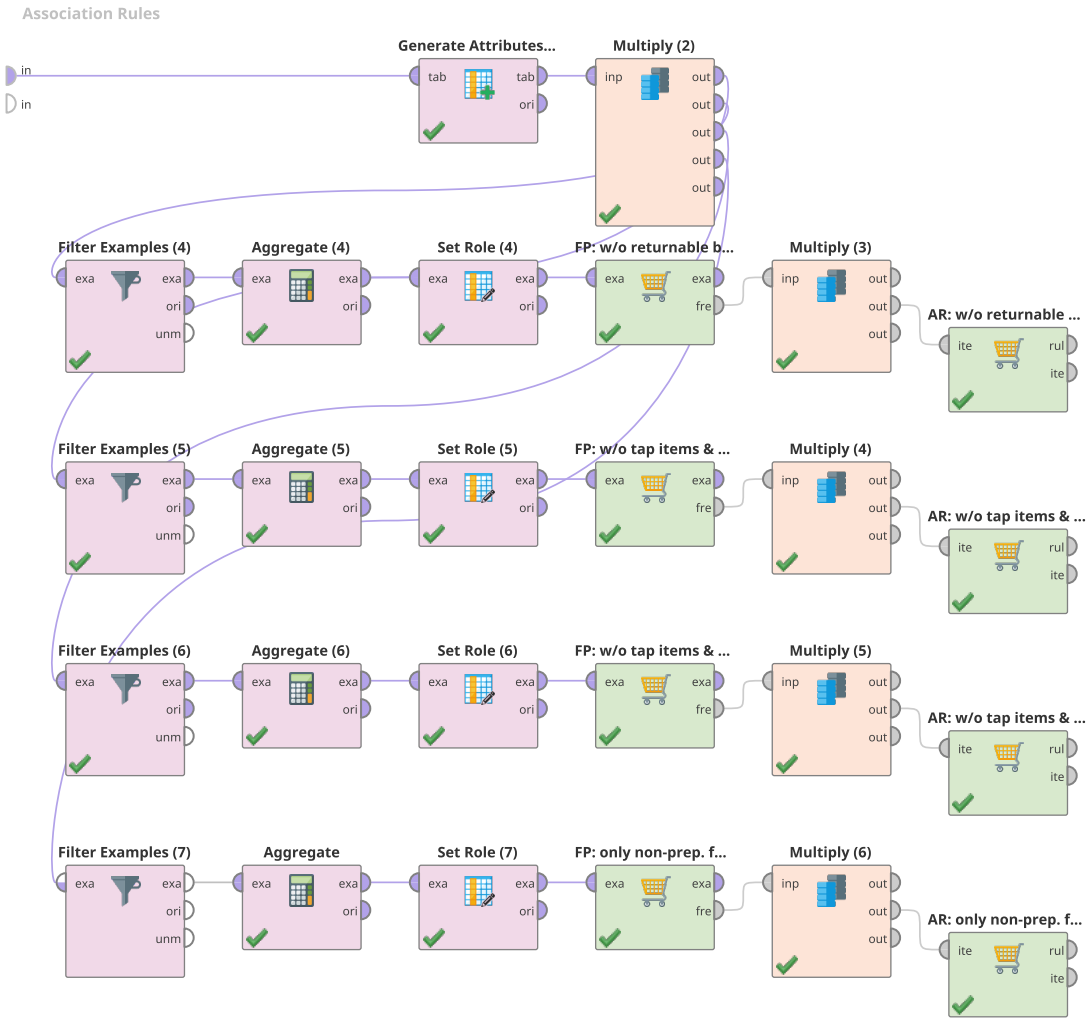
Frekventované množiny

Nástroj RapidMiner implementuje operátory pro tvorbu FP-stromů (a také počítání asociačních pravidel). Je však nutné provést úpravu dat, která vhodně reprezentuje vztah 1 transakce – *n* produktů. Použito bylo kódování s oddělovači, kdy jsou pomocí agregace všechny názvy produktů příslušící jedné transakce zkonkatenovány.

Počáteční experimenty ukázaly na nepříznivou charakteristiku datové sady, ve které jsou dominantně zastoupeny prodeje několika konkrétních produktů (dva druhy piva, toasty, Kofola). Přítomny jsou také virtuální prodeje produktů typu „vratná lahev“, které slouží pro vydání hotovosti při navrácení zálohované lahve. Pro získání zajímavých znalostí byly tedy frekventované množiny počítány pro čtyři různé varianty dat, jak je naznačeno v procesu na obrázku 4:

1. bez vratných lahví

2. bez čepovaných nápojů a vratných lahví
3. bez čepovaných nápojů, připravovaného jídla a vratných lahví
4. bez připravovaného jídla a vraných lahví



Obrázek 4: Proces generování frekventovaných množin nad různými částmi datové sady.

I přes provedené úpravy dat však výsledky nepřinášejí příliš mnoho zajímavých informací. Náhled na frekventované 1-množiny ukazují, že přes 41 % objednávek obsahuje pivo (Svijany Máz 0,5 l) a téměř 40 % objednávek obsahuje toast. S odstupem následuje další druh piva (20 %), rozlévaná Kofola (15 %), točená Kofola (12 %), brambůrky (4,5 %), cider Kingswood (4 %), tyčinka 3BIT (3,8 %) a tonic (3,7 %).

Všechny frekventované 2-množiny s podporou větší než 0,1 % pak obsahují kombinace piva Svijany a nějakého dalšího produktu prakticky ve výše uvedeném pořadí. Frekventovaným množinám varianty 2. očekávaně dominují položky, kde je jedním produktem toast, druhým pak prakticky jakýkoliv jiný produkt.

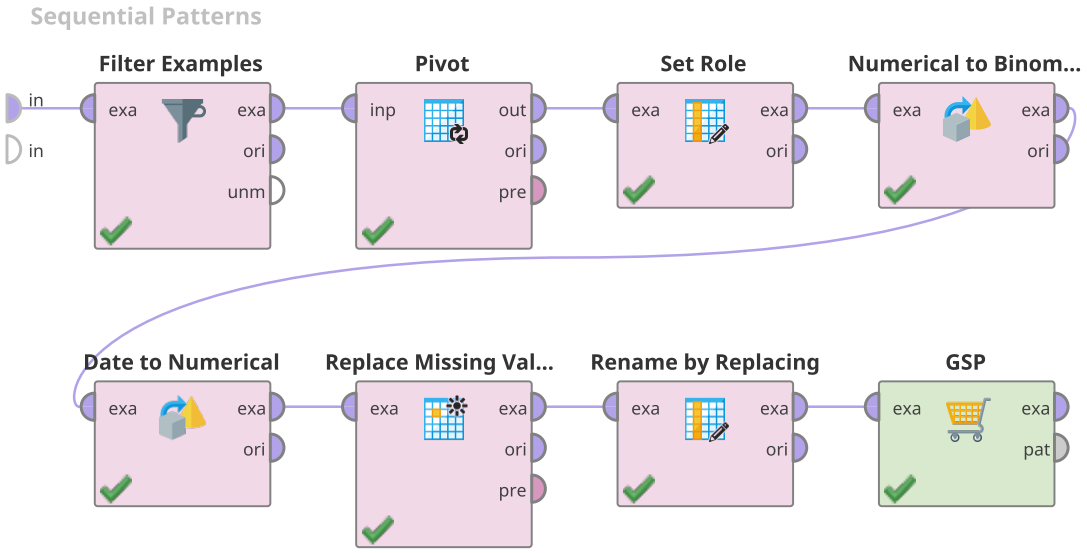
Nahlédneme-li na frekventované množiny varianty 3., největší podporu, a to pouhá 1,3 %, má rozlévaná Kofola (0,5 l) a brambůrky, obdobně Kofola a tyčinka 3BIT. Překvapivé je zastoupení dvojice Kofola 0,5 l a Kofola 0,33 l s podporou 0,8 %.

Celkově se dá říct, že tato úloha žádné užitečné znalosti nepřinesla. Výskyty méně prodávaných produktů jsou v databázi příliš nízké, rozhodování podle frekvenčních množin s takto

nízkou podporou hraničí s náhodou. Pro zajímavost jsme si nechali spočítat také asociační pravidla, která očekávaně také nepřinesla zajímavé informace. Poněkud triviálním závěrem těchto experimentů tedy je, že zákazníci nakupují pivo a toasty.

Sekvenční vzory

Pro použití datové sady s operátorem Generalized Sequential Patterns (GSP) bylo nutné převést ji do podoby, ve které každý řádek odpovídá jedné transakci s určeným datem a všechny možné produkty tvoří binomické atributy (sloupce) vyjadřující, zda daný produkt v transakci byl, nebo ne. Dále byly z vyhledávání rovnou vyloučeny všechny položky typu čepovaný nápoj a vratná lahev. Datové údaje musí být vyjádřeny číselným atributem, datum a čas byly proto převedeny na počet minut od unixové epochy. Tento proces je naznačen na obrázku 5.



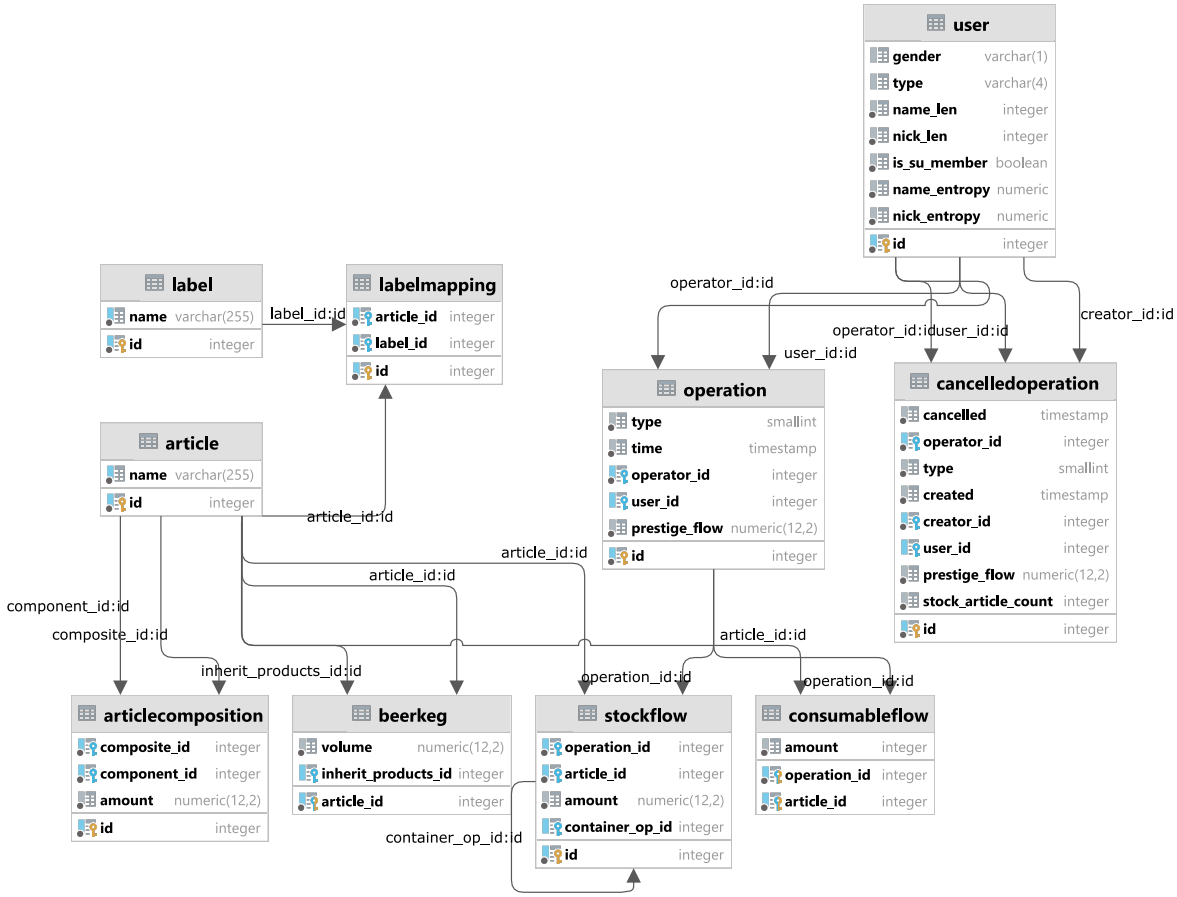
Obrázek 5: Proces generování sekvenčních vzorů.

GSP vyžaduje nastavení parametru velikosti okna (transakce v těsné časové souslednosti jsou považovány za jednu transakci), minimálního rozestupu a maximálního rozestupu. Tyto byly nastaveny na 15, 15 a 360 minut.

Výsledky vykazovaly rovněž značně nízké hodnoty podpory. S tímto nastavením nejsilnější vzor obsahoval pouze jednu transakci se dvěma položkami (Kofola, toast), a to s podporou 17,3 %. Druhý vzor byl Kofola 0,5l̃ a za ní další Kofola 0,5 l, podpora 11,6 %. Následují vzory jako toast → toast, Kofola → toast. Tyto vzory naznačují, že zákazníci často nezůstanou u jedné sklenice nebo jednoho toastu, nicméně tato informace není příliš užitečná. Překvapivý je výskyt sekvence párek v rohlíku → toast, zde už ale s velmi nízkou podporou 2,2 %.

Přílohy

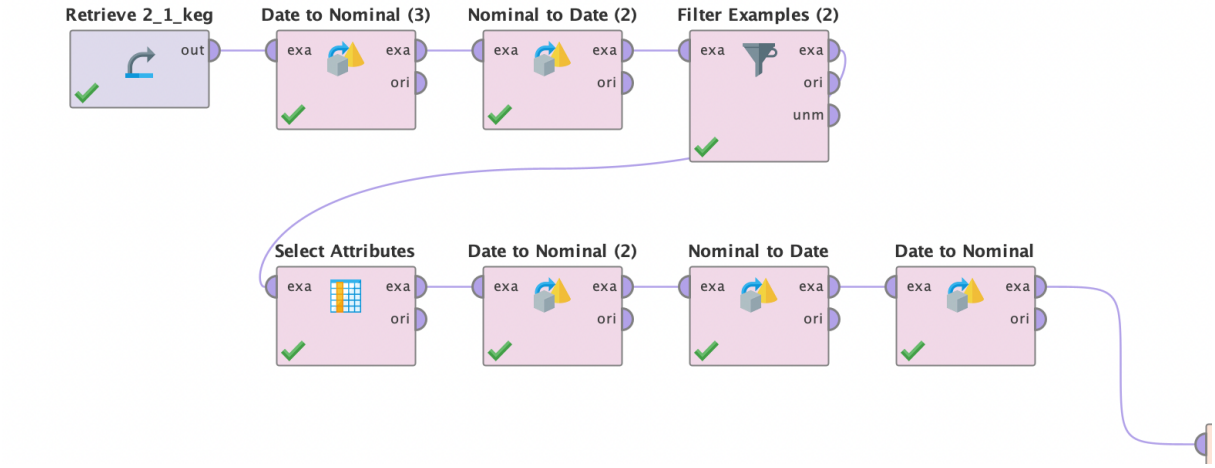
A ER diagram vstupní databáze



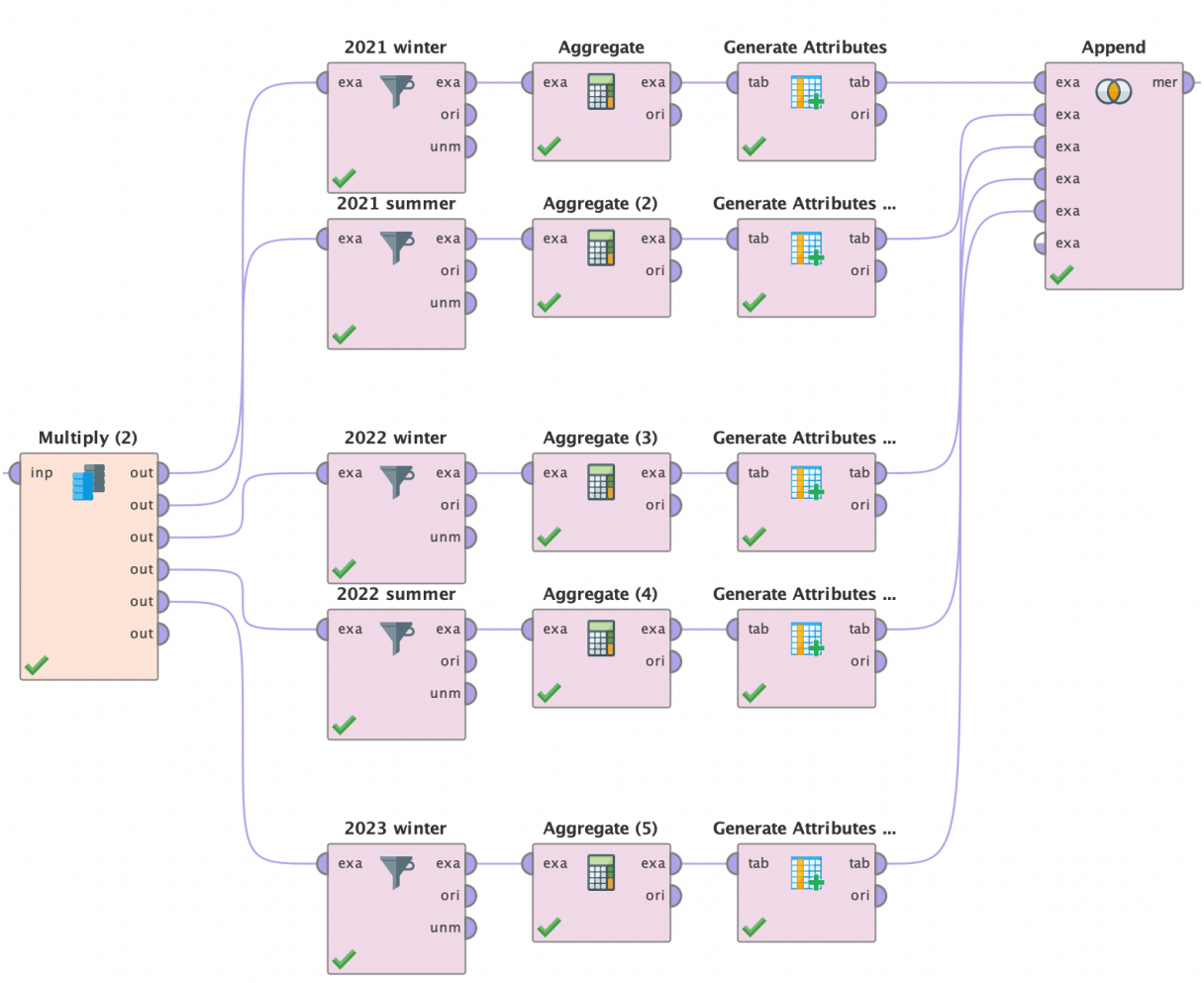
B Výpočet čísla týdne v semestru

```
import com.rapidminer.tools.Ontology;
import java.text.SimpleDateFormat;
import java.util.Date;

ExampleSet inputData = input[0];

// The semester start reference dates
def referenceDates = ["2021-02-08", "2021-09-20", "2022-02-07", "2022-09-19",
  "2023-02-06", "2023-09-18"] as List;
def sdf = new SimpleDateFormat("yyyy-MM-dd");

// Convert strings to dates
referenceDates = referenceDates.collect{ sdf.parse(it) };

// Create new attribute for sem_week
def semWeekAttribute = AttributeFactory.createAttribute("sem_week",
    Ontology.INTEGER);
inputData.getExampleTable().addAttribute(semWeekAttribute);
inputData.getAttributes().addRegular(semWeekAttribute);

// Iterate over each example
def datetimeAttribute = inputData.getAttributes().get("datetime");
for (example in inputData) {
    def currentDate = example.getDateValue(datetimeAttribute);
    def nearestDate = referenceDates.min { currentDate.time - it.time > 0
        ? currentDate.time - it.time
        : Long.MAX_VALUE };
    def weeksElapsed = (currentDate.time - nearestDate.time)
        / (1000 * 60 * 60 * 24 * 7) + 1;
    example.setValue(semWeekAttribute, weeksElapsed as int);
}

return inputData;
```

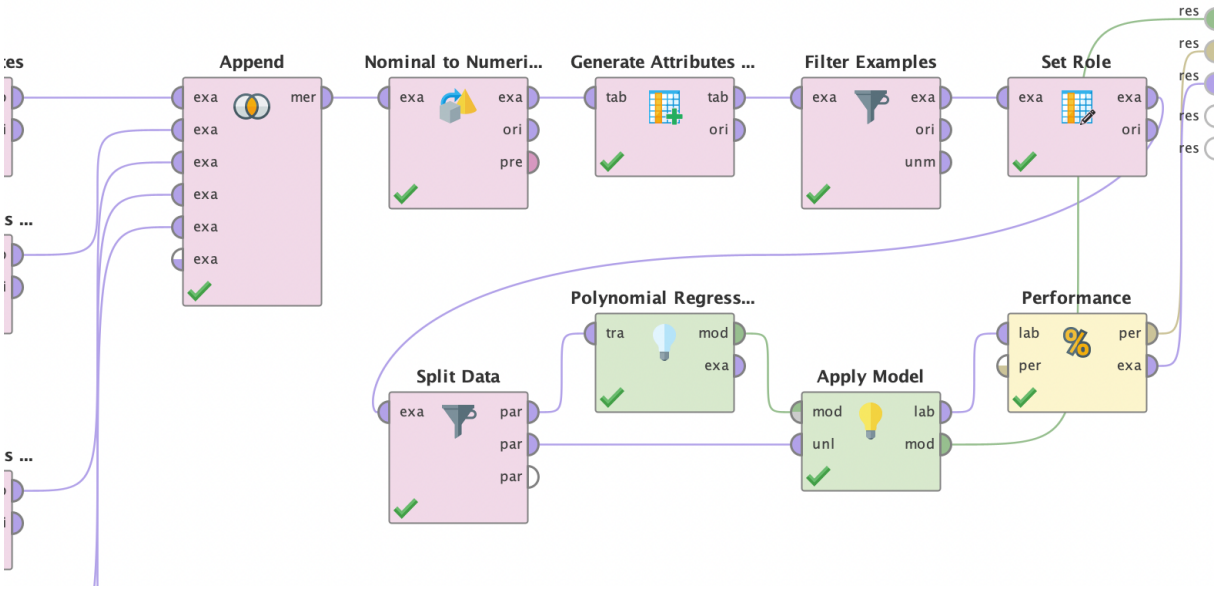
C Optimalizace objednávek zásob – proces – 1. část



D Optimalizace objednávek zásob – proces – 2. část

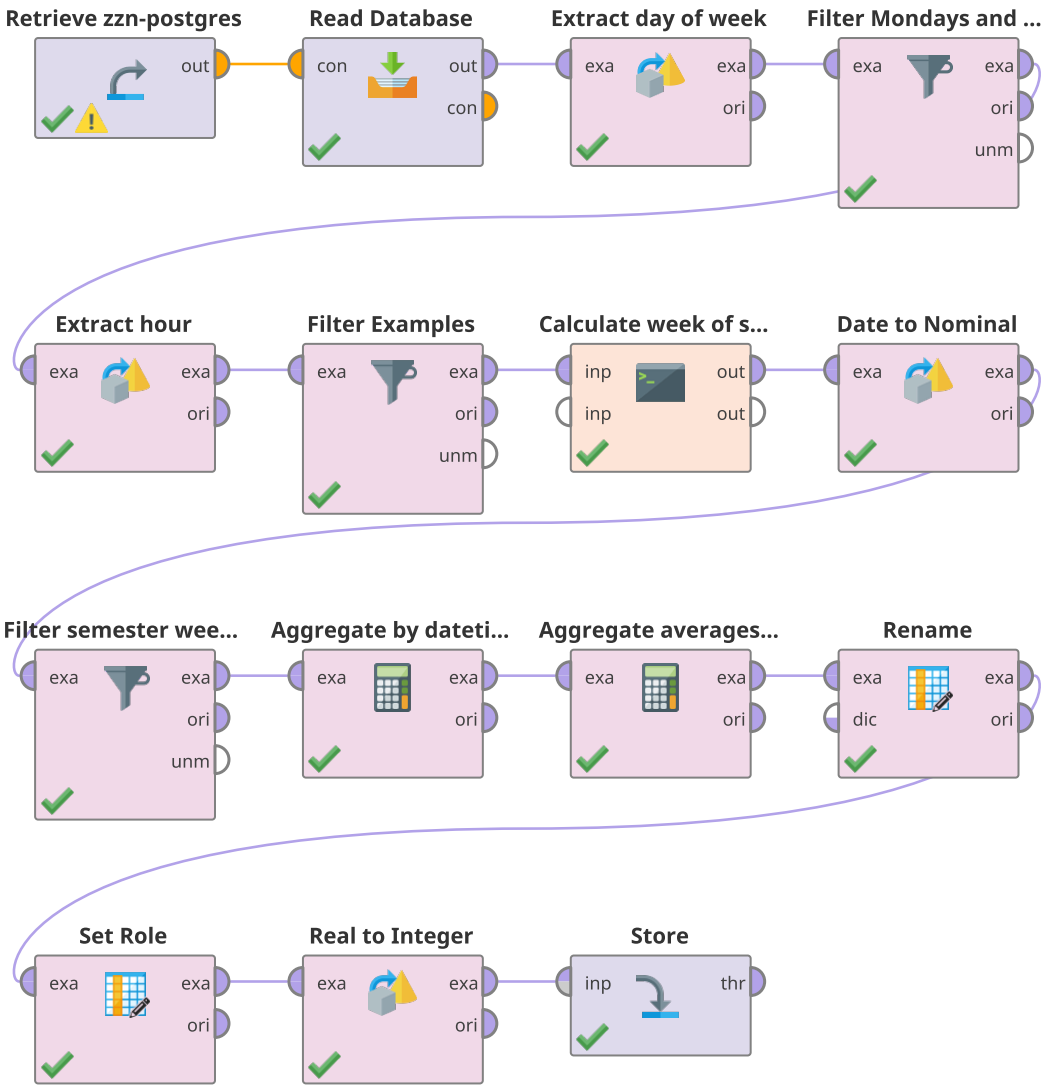


E Optimalizace objednávek zásob – proces – 3. část

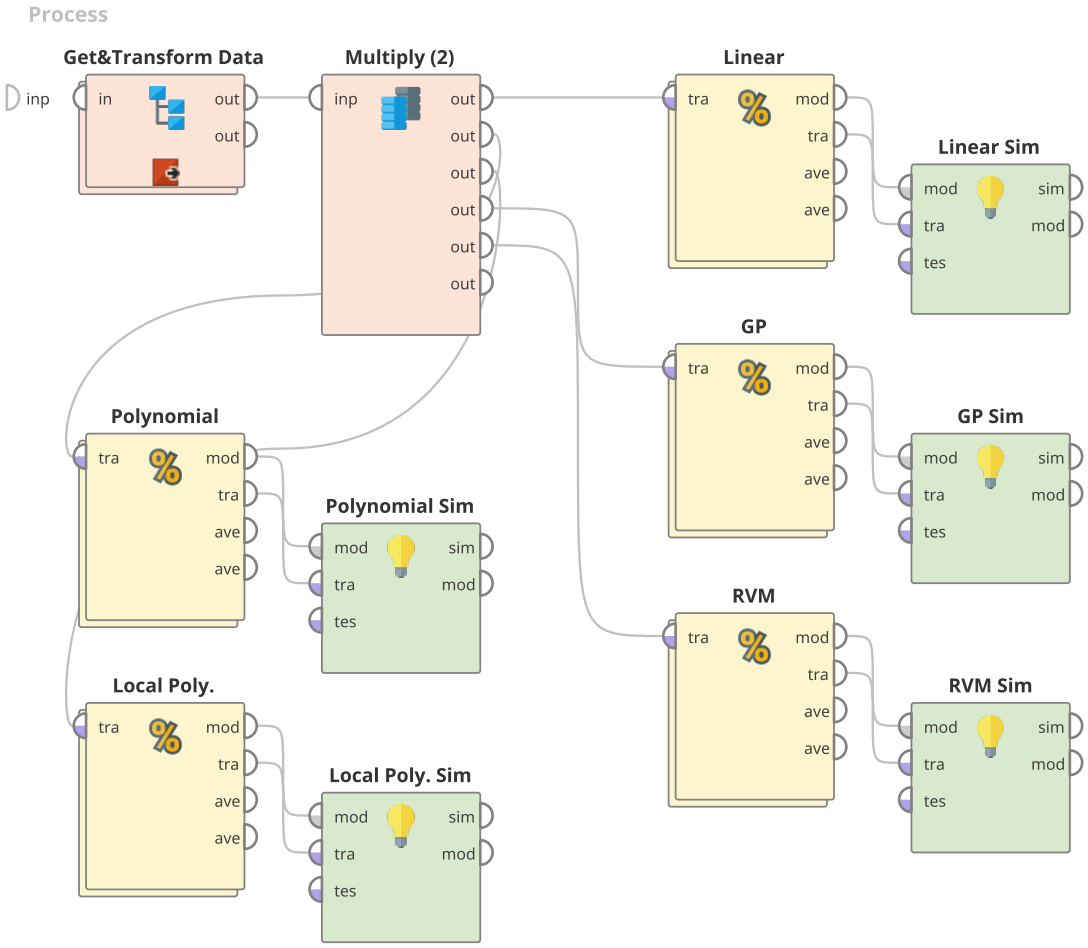


F Plánování směn – transformace dat

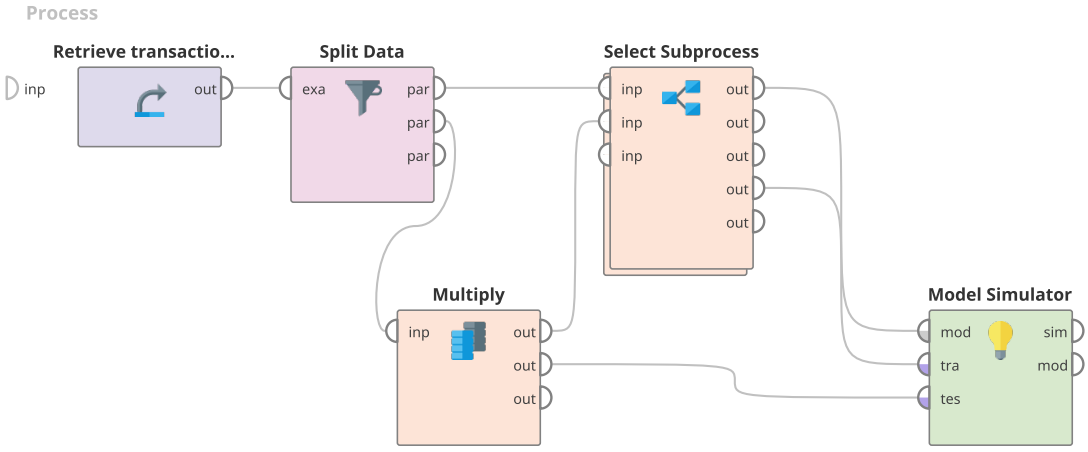
Get&Transform Data



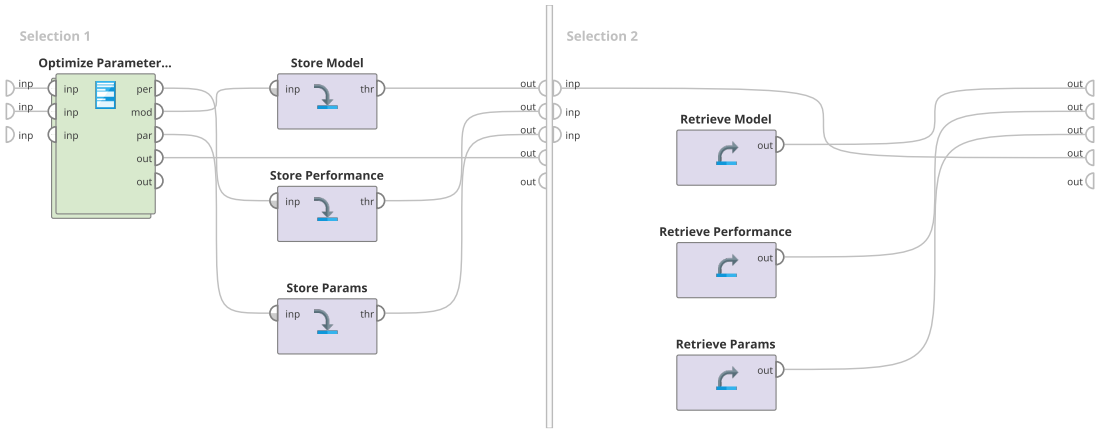
G Plánování směn – testování modelů



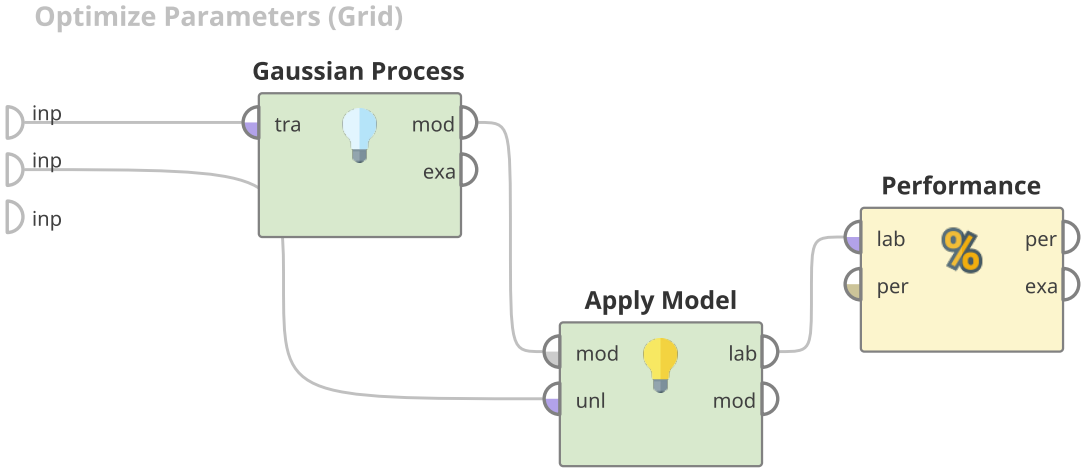
H Plánování směn – optimalizace



Proces s možností výběru už vytvořeného modelu pro simulaci

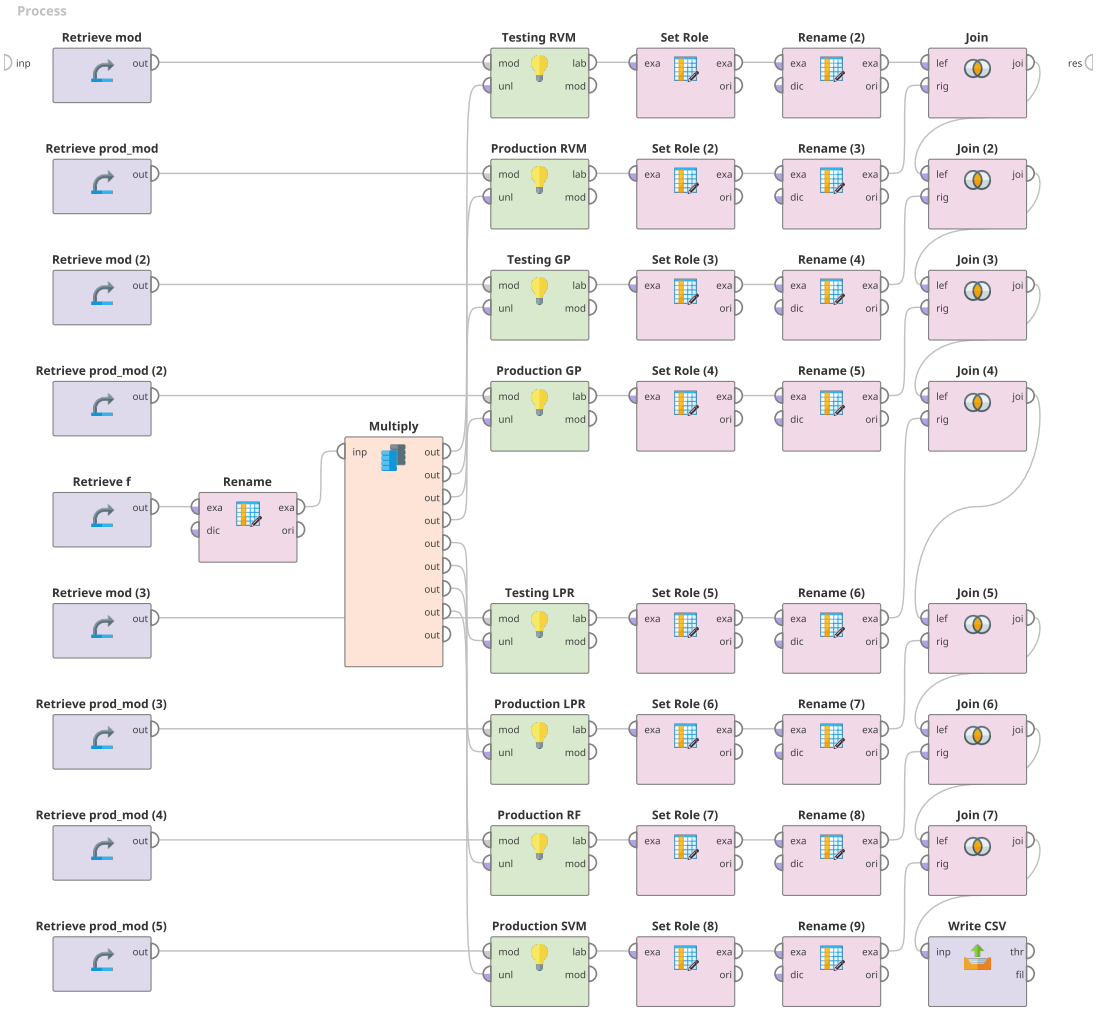


Rozhodovací blok

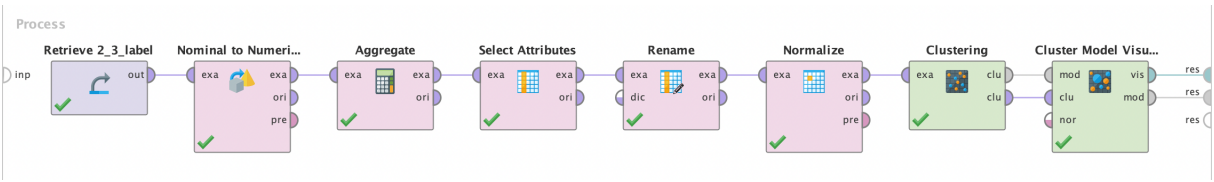


Optimalizace pomocí grid search

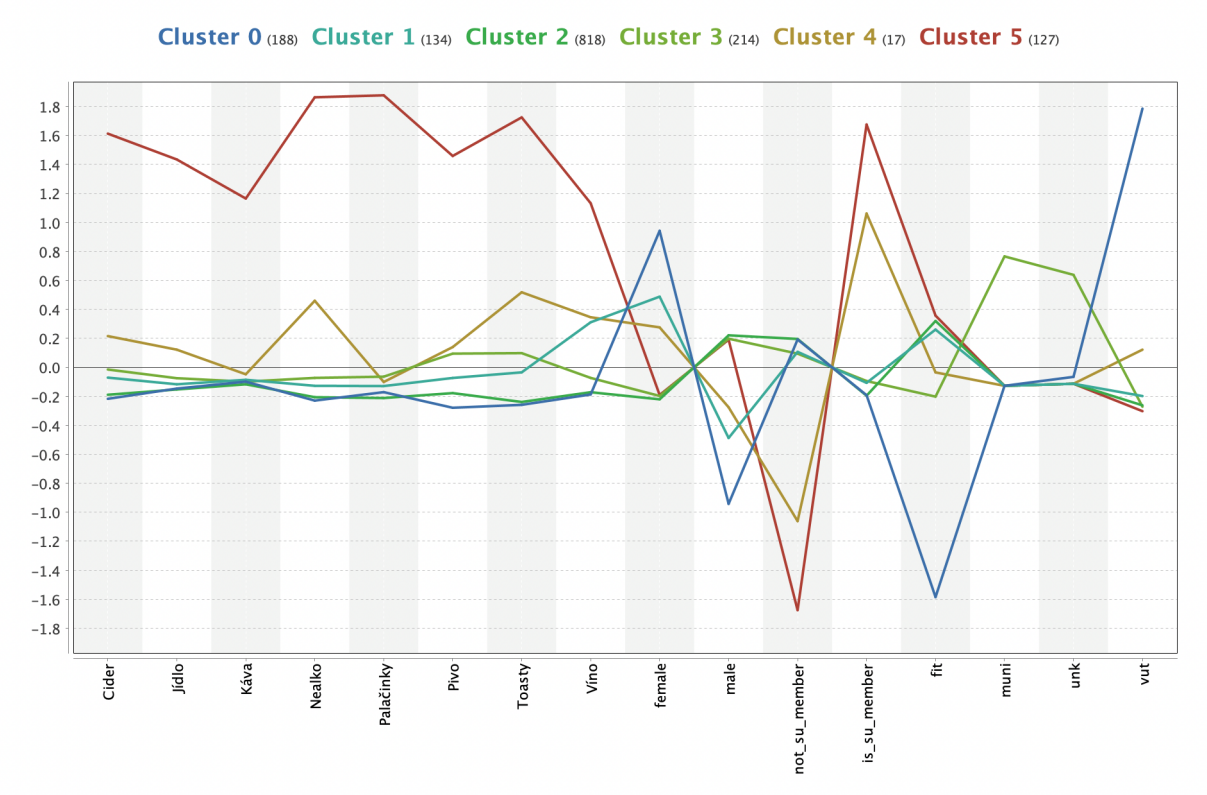
I Plánování směn – generování predikcí



J Segmentizace zákazníků – schéma



K Segmentizace zákazníků – graf centroidů



L Frekventované a sekvenční vzory – SQL dotaz

```
select cf.operation_id            transaction_id,
   a.id                           article_id,
   amount                         amount,
   a.name                         article_name,
   o.user_id                      user_id,
   o.time                         datetime,
   (select count(*) > 0
    from articlecomposition aco
    where aco.composite_id = a.id
      and aco.component_id in
          (select article_id from beerkeg)) is_tap_item,
   (select count(*) > 0
    from labelmapping lm
    where lm.article_id = a.id
      and lm.label_id = 20)              is_returnable_bottle,
   a.id in (175, 212, 29, 64, 220, 28,
            59, 118, 117, 60, 231)       is_prepared_food_item,

from consumableflow cf
join article a on cf.article_id = a.id
join operation o on cf.operation_id = o.id

where cf.operation_id in
    (select o.id
     from operation o
     join consumableflow cf on cf.operation_id = o.id
     left join article a on cf.article_id = a.id
     group by o.id
     having count(a.id) > 1)
```