TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

VZORY CHOVÁNÍ
UKRYTÉ V PROVOZNÍCH DATECH

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Markéta Malá**
Vedoucí práce: RNDr. Klára Císařová, Ph.D.

Liberec 2017

Technická univerzita v Liberci
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: **2016/2017**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Markéta Malá**
Osobní číslo:	**M15000178**
Studijní program:	**N2612 Elektrotechnika a informatika**
Studijní obor:	**Informační technologie**
Název tématu:	**Vzory chování ukryté v provozních datech**
Zadávající katedra:	**Ústav mechatroniky a technické informatiky**

Zásady pro vypracování:

1. Prostudujte problém asociačních a rozhodovacích pravidel v úlohách data miningu.
2. Zabývejte se příspěvkem českých vědců k této problematice. Zpracujte jako téma pro kurz Data mining.
3. Algoritmy, které umí vyhledat vzory chování ukryté ve velkých datech, analyzujte, naprogramujte, případně optimalizujte.
4. Celé téma zpracujte jako interaktivní výkladovou aplikaci pro e-learningový portál tak, aby student mohl jednotlivé algoritmy testovat.

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **40–50 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

[1] **BERKA, Petr. Dobývání znalostí z databází. Praha: Academia, 2003, s.18. ISBN 80-200-1062-9**

[2] **SKALSKÁ, Hana. Data mining a klasifikační modely. Vyd. 1. Hradec Králové: GAUDEAMUS, 2010, 154 s. ISBN 978-80-7435-088-7**

[3] **MAYER-SCHONBERGER, Viktor; CUKIER, Kenneth. Big Data. Vyd. 1. Praha: Computer Press, 2014, 256 s. ISBN 978-80-251-4119-9**

[4] **PUDIL, Petr. Metodology and Advances in Feature Selection for Statistical Pattern Recognition. Disertace DrSc., UTIA AV ČR, Praha 2001**

[5] **HÁJEK P; HAVRÁNEK T. CHYTIL M. METODA GUHA. Automatická tvorba hypotéz. Academia, Praha 1983**

Vedoucí diplomové práce: **RNDr. Klára Císařová, Ph.D.**

Ústav mechatroniky a technické informatiky

Datum zadání diplomové práce: **10. října 2016**

Termín odevzdání diplomové práce: **15. května 2017**

Technická univerzita v Liberci - 14 -

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2016

Prohlášení

Byla jsem seznámena s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracovala samostatně s použitím uvedené literatury a na základě konzultací s vedoucím práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 15.5.2014

Podpis:

Poděkování

Ráda bych na tomto místě poděkovala mé vedoucí RNDr. Kláře Císařové, Ph.D. za její cenné rady, které mi pomohly při vytvoření mé diplomové práce, za náměty a nápady, jak práci dále obohatit, za trpělivost a ochotu, s jakou se mi věnovala, i za kritiku a občasné připomínky. Spolupráce s ní pro mě byla velkým přínosem. Dále bych chtěla poděkovat své rodině a přátelům za motivaci a podporu, kterou mi poskytovali během studia.

Abstrakt

Tato diplomová práce se zabývá asociačními metodami v data miningu, asociačními pravidly jakožto vzory chování ukrytými v datech a vizualizací některých problémů pro vybranou data miningovou úlohu.

Cílem práce bylo seznámit se s užitím asociačních metod při řešení data miningových úloh a na základě získaných znalostí vytvořit interaktivní výukovou aplikaci, která bude pracovat s rozsáhlými datovými soubory, těžit z nich informace a odhalovat skryté asociace, a kterou bude možné dále využívat jako podpůrný prostředek při výuce data miningu.

Výsledná aplikace zpracovává transakční data obsažená v textovém souboru, analyzuje je a třídí, dle požadovaných kritérií vyhledává vztahy v podobě asociačních pravidel s využitím některých známých algoritmů zefektivňujících tento proces a výsledky zprostředkovává uživateli vizuálně prostřednictvím přehledů a tabulek.

Aplikace funguje jako výukový nástroj a je částečně výkladová, kromě slovních popisů vybraných pojmů z problematiky asociačních metod obsahuje rovněž grafické vizualizace těchto problémů usnadňujících jejich pochopení a měla by být užitečnou pomůckou studentům data miningu při studiu.

Pro vývoj aplikace byl zvolen programovací jazyk C# a byla vytvořena ve vývojovém prostředí Visual Studio 2015.

Klíčová slova: data mining, asociace, asociační metody, asociační pravidla, vizualizace

Abstract

This thesis deals with association methods in data mining, with association rules as behaviour patterns hidden in data and with data visualization of some problems for the selected data mining task.

The aim of this work was to become familiar with using the association methods for solving data mining tasks and on the basis to this knowledge create an interactive educational application, which will work with large data sets, obtain certain information of them and uncover hidden associations, and which will be also used as a support tool for teaching data mining.

This application processes some transactional data contained in a text file, analyses and categorizes them, according to the required criteria it retrieves some relationships in the form of association rules using some well-known algorithms accelerating this process and it conveys these results to a user visually through a variety of reports and tables.

Final application works as a teaching tool and is partly educational. In addition to verbal descriptions of selected terms of the issue of association methods it also includes some graphical visualizations of these problems, which makes them easy to understand and it should by a useful tool for students studying data mining.

For a development of this application was chosen programming language C# and it was created in the integrated development environment Visual Studio 2015.

Key words: data mining, associations, association methods, association rules, visualization

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Data mining znamená volně řečeno získávání, vytěžování nebo také dolování znalostí z dat. Jedná se o soubor matematických metod sloužících k hledání netriviálních, zajímavých a potencionálně užitečných informací v datech a k objevování vztahů, které jsou v nich skryté a nejsou na první pohled zjevné. Vznikl jako nástroj umožňující zpracování objemných datových souborů, k jejichž vyhodnocování se klasické dříve používané statistické metody ukázaly jako nedostatečné.

Data mining se začal vyvíjet od 60. let 20. století a zpočátku se jednalo pouze o ojedinělé, zejména akademické záležitosti. To se částečně změnilo s rozvojem statistických metod, databázových aplikací a umělé inteligence a také s růstem rychlosti a velikosti paměti počítačů, které umožnily první systematické využití data miningové metodologie v praxi. Společnost byla ovšem vůči výsledkům data miningu stále spíše skeptická a o jejich důvěryhodnosti přetrvávaly pochyby. Obrat nastal až v 90. letech, poptávka po data miningových nástrojích pak postupně vzrůstala a v současné době jsou již součástí podpory fungování mnoha organizací, v případě špičkových organizací pak součástí nedílnou.

Data mining řeší problémy z mnoha různých oborů od marketingu a bankovnictví, kde je používán pravděpodobně nejvíce, přes oblast bezpečnosti až po některá specializovaná odvětví vědy, například medicíny nebo astrofyziky. Úlohy se rozdělují do několika typů, přičemž pro řešení každého typu úloh je využíváno různých data miningových metod. Mezi nejznámější patří rozhodovací stromy, metody založené na shlukové analýze a neuronových sítích a asociační metody, které jsou hlavním tématem této práce.

Cílem celé práce je podrobně se seznámit s problematikou asociačních metod a se způsoby, jak ve velkých datových souborech vyhledávat vzory chování, například chování nakupujících a jejich preference ve výběru zboží, chování klientů bank při používání účtů, preference studentů při výběru volitelných předmětů a mnoho jiných, a dále s algoritmy, které proces hledání urychlují. Mým praktickým úkolem je pak tyto algoritmy naprogramovat a aplikovat na vybraná data a celý výsledek implementovat ve formě výukové aplikace usnadňující studentům data miningu pochopení dané problematiky.

2 Vzory chování

Vzory chování se ukrývají v datech získávaných a ukládaných během provozu organizací či společností, tedy tzv. provozních datech, v mnoha oborech a lze je využít při rozhodování řady problémů.

Preference zákazníků ve výběru zboží

Přizpůsobení nabídky poptávce zákazníků umožňuje obchodním společnostem zvýšit jejich finanční zisk. Ke zjišťování, o jaké produkty mají lidé zájem, se využívá analýzy nákupního košíku, jejíž podstatou je nalezení asociací mezi jednotlivými druhy zboží, přesněji nalezení takových kombinací zboží, které se často objevují v provedených transakcích a zakoupení jednoho z druhů tak tedy pravděpodobně podmiňuje zakoupení jiného. Na základě zjištěných informací je pak možné nabízet vybrané kombinace produktů za zvýhodněné ceny, vhodně rozmístit zboží v regálech obchodů či na webových stránkách e-shopů nebo sestavovat pro zákazníky zajímavé reklamní nabídky.

K přesnějšímu cílení reklamních nabídek slouží rovněž segmentace zákazníků, která umožňuje jejich rozdělení do homogenních skupin dle segmentačních kritérií, například věku, pohlaví, sociálních či demografických charakteristik anebo pozorovaných vzorců v nákupním chování. Zaměření se pouze na určité skupiny lidí vede ke snížení výdejních nákladů spojených například s propagací reklamy.

Prevence odchodu zákazníků ke konkurenci

Finanční a telekomunikační společnosti zkoumají chování klientů s cílem odhalit jejich případný záměr odejít ke konkurenci. Zde je využíváno dat obsahujících informace o chování lidí, kteří již ke konkurenci migrovali, a na základě nich jsou hledáni stávající klienti s podobnými zvyklostmi. Takoví jsou posléze oslovováni a společnosti jim mohou nabídnout určité výhody či slevy ve snaze v odchodu jim zabránit.

Vyhledávání skupin zákazníků s podobným chováním se v těchto typech úloh provádí pomocí algoritmů odvozených z principů shlukové analýzy, ale také použitím rozhodovacích stromů, neuronových sítí či logistické regrese.

Odhalování podvodů

Další oblastí, kde se data miningové metody a hledání skrytých vzorů chování uplatňuje, je bankovnictví a pojišťovnictví. Banky či společnosti nabízející půjčky, stejně tak jako pojišťovny, si obvykle důkladně prověřují své budoucí klienty ještě před uzavřením smluv, aby předešly finančním podvodům a případnému pozdějšímu nesplácení půjček nebo odhalili jinak rizikové osoby.

Banky dále sledují také pohyby na účtech svých klientů a uskutečněné transakce s cílem detekovat možné podvodné jednání, například praní špinavých peněz. V tomto případě není předem definováno, jak by se měl podezřelý subjekt chovat, je však předpokládáno, že většina klientů se chová korektně, zatímco v chování podvodníků se vykytují určité anomálie. Takový přístup umožňuje soustředit se primárně na nejvíce podezřelé a snáze tak odhalit skutečné podvody.

S podvodným jednáním se kromě již zmíněných společností potýkají také organizace vypisující granty nebo přidělující dotace. Zkoumat podrobně každou žádost a prověřovat všechny žadatele není technicky ani finančně možné a podobně jako v případě prevence praní špinavých peněz se i zde uplatňují data mingové metody detekce anomálních případů.

V těchto typech data miningových úloh se modelování obvykle staví na hledání asociačních pravidel popisujících chování "objektů", jimiž mohou být například klienti banky, žadatelé o dotace nebo oznamovatelé pojistných událostí. Poté řešení pokračuje hledáním anomálií v chování těchto objektů.

Chování uživatelů internetu

Internet se stal nedílnou součástí života člověka. Spojuje uživatele sdílející podobné názory a stejné ideologie, umožňuje vzdělávat se, komunikovat a seznamovat se, nakupovat, odreagovat se sledováním filmu či poslechem hudby on-line a mnoho dalších aktivit. Chování uživatelů internetu je sledováno a analyzováno, je ukládána historie vyhledávání a navštívených webových stránek a je tak možné odhalit řadu problémů od rizika rozšíření epidemie v určité zeměpisné oblasti po možné vazby osob na radikální organizace či hrozbu teroristického útoku.

V těchto typech úloh je hledání vzorů chování spojené s nasazením algoritmů shlukové analýzy, klasifikačních algoritmů, asociačních metod a také algoritmů text minigu.

Využití asociací ve vědě

Odhalování asociací se kromě komerční sféry využívá rovněž ve vědě, konkrétně v genetické epidemiologii [1], což je věda, zkoumající genetické faktory a faktory prostředí podílející se na rozvoji a šíření chorob. Z DNA člověka se dá pomocí data miningových nástrojů vyčíst, jak vysoké je riziko, že dotyčný bude trpět určitou nemocí na základě nalezení asociace mezi touto chorobou a změnou genové exprese související s jejím vznikem.

Data mining se uplatňuje také při vývoji nových léků, umožňuje predikovat míru působení a dopad léčebných látek na organizmus. Účinky jednotlivých látek se často výrazně liší od společných účinků kombinace těchto látek a díky objevování vztahů mezi nimi lze předejít problémům spojeným s kontraindikací, snížit finanční náklady na výzkum a zkrátit dobu potřebnou pro vývoj léku.

Z uvedeného výčtu typů úloh i používaných data miningových postupů je vidět, že behaviorální analýza, tedy analýza chování zkoumaných objektů, je velmi široké téma překračující rozsah diplomové práce. K podrobnějšímu zkoumání pro tuto diplomovou práci byly proto vybrány asociační metody a asociační algoritmy.

3 Asociační metody

Asociační metody jsou jedním z nástrojů užívaných k objevování skrytých vztahů v podobě asociací mezi zkoumanými objekty. Vznik asociačních metod je oficiálně datován do první poloviny 90. let 20. století, v odborné literatuře jsou jako jejich autoři uváděni Rakesh Agrawal, Tomasz Imielinski a Arun Swami. Ti roku 1993 vydali článek *Mining association rules between sets of items in large databases*, v němž se zabývali jednou z nejklasičtějších data miningových úloh, analýzou nákupního košíku. Asociacemi a jejich hledáním se však zabývala také skupina Čechů (P. Hájek, T. Havránek a M. Chytil) již v 70. letech.

Asociační metody se v současné době využívají k řešení různých typů data miningových úloh, nejčastěji z oblasti marketingu a komerční sféry, ovšem své uplatnění nacházejí také ve vědeckém výzkumu, v sociologii i dalších odvětvích.

3.1 Asociační pravidla

Jádrem asociačních metod je získávání asociačních pravidel z dat. Asociační pravidla představují predikční model zapsaný formou implikace [2], kde každá je doplněna vhodnými statistikami, které zvyšují její informační hodnotu a umožňují nalezené implikace třídit a posuzovat jejich kvalitu. Pravidla jsou chápána jako vzory chování zkoumaných objektů, speciální variantou jsou pak sekvenční pravidla, kdy se v předpokladech implikací objevují vzory chování časově uspořádané.

Syntaxe asociačních pravidel je následující: *IF antecedent - THEN consequent,* neboli v češtině: *JESTLIŽE předpoklad - PAK závěr.*

Slovní formulace asociačních pravidel může být: *„Jestliže je splněna podmínka (předpoklad), pak platí závěr“.*

Ke značení implikace se a užívá symbol „→“ a zapisujeme $\boldsymbol{A \rightarrow B}$, kde *A* je předpoklad asociačního pravidla a *B* je jeho závěr.

Příklady asociačních pravidel:

***IF** pohlaví = žena **THEN** tv_žánr = romantika*

***IF** pohlaví = muž **THEN** tv_žánr = sport*

***IF** příjem = nízký **AND** konto = nízké **THEN** úvěr = ne*

***IF** kolečkové brusle **THEN** helma **AND** chrániče*

***IF** mléko **AND** máslo **THEN** pečivo*

Jak je vidět na uvedených příkladech, předpoklad i závěr asociačního pravidla mohou být tvořeny jednou položkou, stejně tak ovšem i dvěma či více položkami.

3.1.1 Numerické charakteristiky asociačních pravidel

Aby bylo možné s nalezenými asociačními pravidly dále pracovat, třídit je, posuzovat jejich kvalitu a informační hodnotu a vyvozovat z nich závěry, je nutné je kvantitativně hodnotit.

U každé implikace se v první řadě zjišťuje, kolik případů splňuje její levá strana, tedy předpoklad (anglicky *antecedent*) a kolik její pravá strana, tedy závěr (anglicky *consequent* nebo *succedent*), dále kolik případů splňují předpoklad i závěr současně, kolik případů splňuje předpoklad, ovšem nikoli závěr, kolik případů naopak nesplňuje předpoklad, zatímco závěr ano, a kolik případů nesplňuje ani předpoklad a ani závěr.

Veškeré výše zmíněné hodnoty se zapisují do kontingenční (čtyřpolní) tabulky. Ta pro asociační pravidlo ve tvaru

$$Ant\ A \rightarrow Cons\ B$$

kde *Ant A* a *Cons B* jsou kombinace kategorií vypadá následovně:

Tabulka 1: Kontingenční tabulka pro n prvků

	Cons B	*¬Cons B*	∑
Ant A	a	b	$a + b$
¬Ant A	c	d	$c + d$
∑	$a + c$	$b + d$	n

n... celkový počet objektů

$$n = a + b + c + d$$

a... počet případů, kdy platí současně předpoklad i závěr

$$a = n(Ant\ A \wedge\ Cons\ B)$$

b... počet případů, kdy platí předpoklad a neplatí závěr

$$b = n(Ant\ A \wedge \neg Cons\ B)$$

c... počet případů, kdy neplatí předpoklad a platí závěr

$$c = n(\neg Ant\ A \wedge Cons\ B)$$

d... počet případů, kdy současně neplatí ani předpoklad, ani závěr

$$c = n(\neg Ant\ A \wedge \neg Cons\ B)$$

O těchto hodnotách se mnohdy místo jako o počtu objektů pokrytých danou kombinací mluví jako o *četnosti* či *frekvenci* kombinace a na jejich základě lze zjišťovat různé *numerické charakteristiky asociačních pravidel* [3].

Základní numerické charakteristiky

- **Podpora implikace** (implication support) je počet objektů splňující zároveň předpoklad i závěr.

  Absolutní: $Supp(Ant \wedge Cons) = a$

  Relativní: $P(Ant \wedge Cons) = \frac{a}{a+b+c+d}$

- **Podpora předpokladu** (antecedent support) je počet objektů splňující předpoklad bez ohledu na to, zda splňují či nesplňují závěr:

  Absolutní: $Supp(Ant) = a + b$

  Relativní: $P(Ant) = \frac{a+b}{a+b+c+d}$

- **Podpora závěru** (consequent support) je počet objektů splňující závěr bez ohledu na to, zda splňují předpoklad.

  Absolutní: $Supp(Cons) = a + c$

  Relativní: $P(Cons) = \frac{a+c}{a+b+c+d}$

  Relativní hodnoty četnosti kombinace ve všech třech výše zmíněných případech jsou chápány jako odhad *pravděpodobnosti* výskytu této kombinace v datech.

- **Spolehlivost** (confidence) rovněž nazývaná jako **platnost** je podmíněná pravděpodobnost platnosti závěru v případě platnosti předpokladu [3]. Je to počet objektů splňující závěr z takových, které splňují předpoklad.

  $$Conf(Ant \rightarrow Cons) = P(Cons|Ant) = \frac{a}{a+b}$$

- **Pokrytí** (coverage) neboli také **úplnost** je podmíněná pravděpodobnost platnosti předpokladu v případě platnosti závěru [3].

  $$P(Ant|Cons) = \frac{a}{a+c}$$

- **Navýšení** (lift) je relativní zvýšení pravděpodobnosti platnosti závěru za platnosti předpokladu [3]. Udává, kolikrát se zvýší pravděpodobnost platnosti závěru, bude-li využito asociační pravidlo.

  $$Lift(Ant \rightarrow Cons) = \frac{P(Cons|Ant)}{P(Cons)} = \frac{\frac{a}{a+b}}{\frac{a+c}{a+b+c+d}} = \frac{a(a+b+c+d)}{(a+b)(a+c)}$$

- **Uplatnění** (deployability) je rozdíl mezi podporou předpokladu a podporou celé implikace. Jsou to případy, kdy platí předpoklad, ale neplatí závěr.

  $$Dep(Ant \rightarrow Cons) = P(Ant) - P(Ant \wedge Cons) = \frac{a+b}{a+b+c+d} - \frac{a}{a+b+c+d}$$

- **Kvalita** (quality) je vážený součet spolehlivosti a pokrytí [3].

  $$Quality = w_1 \frac{a}{a+b} + w_2 \frac{a}{a+c}$$

  Konstanty w_1 a w_2 se obvykle volí tak, aby $w1 + w_2 = 1$.

Rozšířené numerické charakteristiky

- Kauzální podpora (causal support)

$$P(Ant \wedge Cons) + P(\neg Ant \wedge \neg Cons) = \frac{a+d}{a+b+c+d}$$

- Kauzální spolehlivost (causal confidence)

$$\frac{1}{2}P(Cons|Ant) + \frac{1}{2}P(\neg Ant|\neg Cons) = \frac{1}{2}\frac{a}{a+b} + \frac{1}{2}\frac{d}{b+d}$$

- Deskriptivní potvrzení (descriptive confirmation)

$$P(Ant \wedge Cons) - P(Ant \wedge \neg Cons) = \frac{a-d}{a+b+c+d}$$

- Kauzální potvrzení (causal confirmation)

$$P(Ant \wedge Cons) + P(\neg Ant \wedge \neg Cons) - 2P(Ant \wedge \neg Cons) = \frac{a+d-2b}{a+b+c+d}$$

- Ujištění (conviction)

$$\frac{P(Ant)P(\neg Cons)}{P(Ant \wedge \neg Cons)} = \frac{(a+b)(b+d)}{d(a+b+c+d)}$$

- Zajímavost (interestingness)

$$\frac{P(Ant \wedge Cons)}{P(Ant)P(Cons)} = \frac{a(a+b+c+d)}{(a+b)(a+c)}$$

- Závislost (dependency)

$$P(Cons|Ant) - P(Cons) = \frac{a}{a+b} - \frac{a+c}{a+b+c+d}$$

Dělení asociačních pravidel

Asociační pravidla lze na základě jejich platnosti a pokrytí dělit do následujících tří skupin*:*

- **Konzistentní pravidla** jsou implikace, u kterých je levá strana postačující podmínkou pro splnění pravé strany. Jejich platnost je rovna 1.
- **Úplná pravidla** jsou implikace, u kterých je levá strana nutnou, nikoli však postačující podmínkou pro splnění pravé strany. Jejich pokrytí je rovno 1.
- **Deterministická pravidla** jsou implikace, u kterých je levá strana nutnou a postačující podmínkou pro splnění pravé strany. Jejich platnost i pokrytí jsou rovny 1.

3.1.2 Požadavky na asociační pravidla

Fakt, že v data miningu jsou zpravidla zpracovávány rozsáhlé datové soubory, má za následek nalezení velkého počtu asociačních pravidel a nabízí se tedy otázka, jak toto množství redukovat – jaké vlastnosti by implikace měly či neměly mít. Platí, že asociační pravidla by měla být snadno pochopitelná a vysvětlitelná, použitelná a netriviální [4]. Co to znamená?

- **Snadno pochopitelná** asociační pravidla
  - platí, že jakmile je v datech objeven nějaký vztah, lze jej snadno ověřit
- **Vysvětlitelná** asociační pravidla
  - nalezené vztahy se dají vysvětlit, jsou smysluplné
- **Použitelná** asociační pravidla
  - podávají užitečné, dále využitelné informace, vedou k dalším intervencím
- **Netriviální** asociační pravidla
  - cílem asociačních metod je odkrývání zajímavých vztahů ukrytých v datech, nikoli nepodstatných informací či faktů již předem známých, zjevně patrných a od počátku jasných

3.2 Vyhledávání asociačních pravidel v datech

Struktura datových souborů

Asociační pravidla jsou vyhledávána v datových soborech, které rozděluje data mining na dva typy: transakční nebo tabulární data. Toto rozdělení je dáno strukturou příslušného souboru, která je v případě transakčních dat celkově jednodušší, nežli je tomu u dat tabulárních, ta jsou ovšem vhodnější z hlediska zpracování data miningovými nástroji. Pro vysvětlení rozdílu mezi zmíněnými typy budeme uvažovat jednoduchý soubor obsahující informace o transakcích zákazníků fiktivního obchodu:

Tabulka 2: Soubor transakcí

ID zákazníka	*Nákup*
1	džem
2	mléko
3	džem, chléb
4	džem, chléb, mléko

Transakční data

Transakční data vznikají při průchodu zákazníka pokladnou a v takové podobě se také ukládají. Pro každou položku transakce existuje oddělený záznam obsahující dvě informace: ID zákazníka a název položky. Pokud zákazník v rámci jednoho nákupu zakoupí více než jeden druh zboží, vytvoří se příslušný počet záznamů obsahující stejné ID zákazníka a lišící se v názvech položek.

V následující tabulce je uvedeno, jak by vypadal výše zmíněný ukázkový datový soubor převedený do formátu transakčních dat:

Tabulka 3: Transakční data

ID zákazníka	*Nákup*
1	džem
2	mléko
3	džem
3	chléb
4	džem
4	chléb
4	mléko

Tabulární data

V případě tabulárních dat nejsou vytvářeny oddělené záznamy pro každou položku transakce, ale vzniká vždy jediný záznam, který kromě ID zákazníka však obsahuje všechny položky nabídky bez ohledu na to, zda dané zboží zákazník skutečně zakoupil či nikoli. Pro každý produkt je pak pomocí pravdivostních hodnot (true/false, T/F, 1/0) specifikována jeho přítomnost či absence v příslušné transakci.

Příklad tabulárních dat vytvořených na základě ukázkového datového souboru je uveden v následující tabulce:

Tabulka 4: Tabulární data

ID zákazníka	*Džem*	*Chléb*	*Mléko*
1	T	F	F
2	F	F	T
3	T	T	F
4	T	T	T

Oba formáty datových souborů, transakční i tabulární, jsou mezi sebou vzájemně snadno převoditelné. Tabulární data vznikají z transakčních převodem *kategoriální proměnné* (v našem případě název zboží) na tzv. *indikátorové proměnné* (specifikace přítomnosti/absence položky v transakci, tedy hodnoty T/F).

Silná asociační pravidla

Jak již bylo řečeno v kapitole 3.1.2, z rozsáhlých datových souborů lze získat velké množství asociačních pravidel. Aby bylo hledání smysluplné a jeho výsledky užitečné, je nutné tuto množinu redukovat a vybrat pouze takové implikace, které jsou netriviální a dále použitelné, mají dostatečnou informační hodnotu a zároveň odhalují nová fakta, nikoli již známá.

Před započetím analýzy dat je nutné určit požadavky, které musí vybraná asociační pravidla splňovat. Implikace, které vyhovují zvoleným kritériím, se nazývají *silná asociační pravidla* [3] a jsou to takové, které mají vysokou hodnotu podpory a spolehlivosti – dosahují předem stanovené hodnoty anebo ji převyšují.

Množina silných asociačních pravidel (angl. *strong association rules*) je definována takto:

$$SAR = \{AR \mid conf(AR) \geq minconf \wedge sup(AR) \geq minsup\}$$

AR.....asociační pravidlo tvaru $A \rightarrow B$, kde A je předpoklad a B je závěr implikace
A.....konjunkce predikátů tvaru $a_1 \wedge a_2 \ldots \wedge a_m$
B.....konjunkce predikátů tvaru $b_1 \wedge b_2 \ldots \wedge b_n$
conf(AR) .....spolehlivost pravidla
sup(AR).....podpora pravidla
minconf.....požadovaná minimální spolehlivost
minsup .....požadovaná minimální podpora

Silným asociačním pravidlem tedy nazýváme každé takové pravidlo, jehož podpora je vyšší nebo rovna minimální podpoře (její hodnota je předem určená ještě před začátkem procesu vyhledávání) a jehož spolehlivost je vyšší nebo rovna minimální předem určené spolehlivosti.

Průběh hledání asociačních pravidel

Proces vyhledávání implikací v datech probíhá ve dvou etapách: nejprve jsou vygenerovány tzv. *frekventované množiny* kandidátů a z nich následně získána vlastní asociační pravidla.

A) Generování frekventovaných množin

- hledání kandidátů – položek a množin, u kterých bude následně zjištěna hodnota jejich podpory v datech

- nalezení tzv. frekventovaných množin – takových, jejichž podpora dosahuje zadané minimální podpory anebo ji převyšuje

B) Získání asociačních pravidel

- využití frekventovaných množin nalezených v předchozím kroku
- rozložení každé frekventované množiny na podmnožiny a vygenerování kandidátních asociačních pravidel tvaru $A \rightarrow B$
- pro každou implikaci $A \rightarrow B$, kde levá strana A, tedy předpoklad pravidla, je množina $A = \{a_1, a_2, \ldots, a_m\}$ a pravá strana B, tedy závěr pravidla, je množina $B = \{b_1, b_2, \ldots, b_n\}$, musí platit:
  - $A \cup B$ je frekventovaná množina
  - pokud $A \cup B$ je frekventovaná množina, potom také každá podmnožina C taková, že $C \subset (A \cup B)$ a zároveň $C \neq \emptyset$, je rovněž frekventovaná
- je-li $L = A \cup B$ frekventovaná množina a $|L| = k$, pak existuje právě 2^k-2 kandidátních asociačních pravidel, ignorují se implikace $L \rightarrow \emptyset$ *a* $\emptyset \rightarrow L$, tedy případy, kdy předpoklad, resp. závěr implikace tvoří prázdná množina
- výpočet spolehlivosti pro kandidátní asociační pravidla, odstranění takových, jejichž spolehlivost nedosahuje požadované předem určené minimální spolehlivosti
- nalezení silných asociačních pravidel

Nalezená asociační pravidla se dále testují aplikací na konkrétní data – ověřuje se jejich platnost a zjišťuje se, zda splňují požadavky na další hodnoty numerických charakteristik (viz kapitola 3.1.1).

3.2.1 Generování frekventovaných množin

První etapou procesu získávání asociačních pravidel je vždy vygenerování všech kombinací položek a následné zjišťování, s jakou četností se vyskytují v analyzovaném datovém souboru. Vygenerování kombinací (konjunkcí) hodnot atributů může být realizováno několika způsoby, podle [3] existují tři metody, kterými to lze provést:

A) Generování kombinací do šířky

- jedná se o generování kombinací podle délek – generuje se od nejkratších po nejdelší
- nejprve se vygenerují všechny kombinace délky jedna, tedy jednoprvkové množiny, kde každá je tvořena právě jednou z veškerých existujících kategorií atributů

- z jednoprvkových množin jsou vytvářeny všechny přípustné dvouprvkové množiny, z nich poté tříprvkové vznikající jako sjednocení dvouprvkových, čtyřprvkové vznikající jako sjednocení tříprvkových atd., dokud není dosaženo maximální možné délky kombinace, která je rovna počtu kategorií atributů

B) Generování kombinací do hloubky

- nalezne se první kombinace délky jedna tvořená některou z kategorií atributů, která se postupně prodlužuje přidáváním dalších hodnot (nové kategorie atributů), dokud je to možné
- v případě, že již nelze kombinaci prodloužit, změní se kategorie posledního atributu, není-li možné ani to (všechny existující kategorie atributu již byly vyčerpány), kombinace se zkrátí o jeden člen a této se změní kategorie posledního atributu
- generování je ukončeno, jsou-li vyčerpány všechny kategorie všech atributů

C) Generování kombinací podle četnosti – heuristická metoda

- v případě této metody nejsou kombinace generovány v libovolném náhodném pořadí, nýbrž v pořadí závisejícím na jejich výskytu v datech od nejfrekventovanějších po nejméně frekventované
- kombinace, které se v analyzovaném souboru vyskytovaly s nejvyšší četností, se objevují na začátku výsledného seznamu, kombinace s nulovou četností potom naopak na jeho konci

Počet vygenerovaných kombinací je exponenciálně závislý na počtu atributů a jejich kategorií. Uvažujeme-li počet atributů rovný n a počet kategorií atributu A_i rovný K_{Ai}, pak:

- počet kombinací délky 1 je roven $\sum_{i=1}^{n} K_{A_i}$
- počet kombinací délky 2 je roven $\sum_{i,j=1;i\neq j}^{n} K_{A_i}K_{A_j}$
- počet kombinací délky 3 je roven $\sum_{i,j,k=1;i\neq j\neq k}^{n} K_{A_i}K_{A_j}K_{A_k}$
- celkový počet existujících kombinací pak je roven $\prod_{\mathrm{i=1}}^{\mathrm{n}}\left(1+K_{A_i}\right)-1$

Z kombinací vygenerovaných některým z výše zmíněných postupů jsou následně vybrány pouze frekventované položky, resp. frekventované množiny, tedy takové, jejichž četnost výskytu v datech je dostatečně vysoká a splňuje zadané kritérium – dosahuje minimální požadované podpory nebo ji převyšuje, naopak kombinace, které tomuto kritériu nevyhovují, jsou vyřazeny a následně se s nimi již nepracuje.

3.2.2 Získání asociačních pravidel

Asociační pravidla (implikace tvaru $A \rightarrow B$) jsou získávána z frekventovaných množin vygenerovaných tak, jak je popsáno v kapitole 3.2.1, přičemž spolehlivost těchto implikací musí splňovat kritérium požadované minimální spolehlivosti.

Postup při získávání asociačních pravidel

A) Rozložení frekventovaných množin na podmnožiny

- každá frekventovaná množina je rozložena na všechny existující podmnožiny
- z frekventované množiny délky *n* lze získat právě $\binom{n}{1}$ jednoprvkových podmnožin, $\binom{n}{2}$ dvouprvkových podmnožin a až $\binom{n}{n-1}$ (n - 1)prvkových podmnožin, tedy dohromady získat $\sum_{k=1}^{n-1}\binom{n}{k}$ podmnožin

B) Nalezení kandidátů a výpočet jejich spolehlivosti

- z podmnožin získaných v předchozím kroku se vygeneruje množina všech kandidátních asociačních pravidel následovně:
  - frekventovaná množina L je rozložena na podmnožiny L_1 a L_2, že platí $L_1 \cup L_2 = L$
  - z takto získaných podmnožin L_1 a L_2 lze vytvořit dvě implikace: $L_1 \dashrightarrow L_2$ a $L_2 \dashrightarrow L_1$
- pro každé získané asociační pravidlo je vypočítána jeho spolehlivost jako podíl podpory předpokladu a podpory závěru tohoto pravidla

C) Zisk silných asociačních pravidel

- spolehlivost nalezených kandidátních implikací je porovnána s požadovanou minimální spolehlivosti a jsou vyřazeny všechny ty, které této hodnoty nedosahují, výsledná redukovaná množina kandidátů je pak označována jako množina silných asociačních pravidel

Vyhledávání asociačních pravidel v rozsáhlých datových souborech či souborech obsahujících velké množství atributů nabývajících mnoha různých hodnot je časově náročný proces, k jeho zefektivnění byla proto navržena řado algoritmů. V následujících kapitolách této práce budou zmíněny dva: algoritmus APRIORI a jeho optimalizovaná verze – algoritmus APRIORI TID.

4 Algoritmus APRIORI

Pravděpodobně nejznámějším algoritmem užívaný pro zefektivnění vyhledávání asociačních pravidel v datech je algoritmus APRIORI. Zveřejněn byl roku 1994. V souvislosti s analýzou nákupního košíku ho navrhl Rakesh Agrawal.

Jedná se o algoritmus, který při generování frekventovaných množin (jejich podpora je vyšší nebo rovna požadované minimální podpoře) využívá jejich apriori vlastnosti, tedy že pro každou frekventovanou množinu platí: je-li $X = \{x_1, x_2, \ldots x_m\}$ frekventovaná množina, pak musí být také všechny množiny Y takové, že $Y \subset X$ frekventované.

Algoritmus pracuje následovně:

- Nejprve jsou nalezeny všechny jednoprvkové frekventované množiny standardním způsobem: vznikají jako množiny tvořené jednotlivými kategoriemi atributů, následně je zjištěna četnost jejich výskytu v datovém souboru a dojde k vyřazení všech kandidátů, jejichž podpora není dostatečně vysoká
- Kandidátní množiny délky 2 jsou vytvářeny jako sjednocení jednoprvkových frekventovaných množin, u takto vygenerovaných kandidátů je opět vypočtena jejich podpora a porovnána s požadovanou minimální podporou. Jsou nalezeny všechny frekventované množiny délky 2.
- Při generování kandidátních množin délky k se využije frekventovaných množin délky $k-1$, jejichž sjednocováním vznikají množiny délky k. U nově vzniklých kandidátů se ověří, zda se jedná o frekventované množiny, z takových je pak možné získat kandidáty délky $k+1$.

Po nalezení všech frekventovaných množin jsou vytvářena vlastní asociační pravidla na základě předem určeného kritéria minimální spolehlivosti, tento proces probíhá tak, jak bylo popsáno v kapitole 3.2.2 a algoritmus APRIORI jej nijak neurychluje.

Průběh algoritmu APRIORI

A) Generování frekventovaných množin

Algoritmus apriori

- do množiny L_1 přiřaď všechny kategorie atributů, které dosahují požadované minimální podpory nebo ji převyšují
- polož $k = 2$
- dokud $L_{k-1} \neq \emptyset$
  - pomocí funkce ***apriori-gen*** vygeneruj na základě L_{k-1} množinu kandidátů K_k
  - do množiny L_k zařaď všechny takové kombinace z K_k, které vyhovují požadavku minimální podpory
  - inkrementuj k

Funkce apriori-gen (L_{k-1})

- pro všechny dvojce kombinací l_1, l_2 z L_{k-1}
  - pokud se l_1 a l_2 shodují v $k-2$ kategoriích, přidej do K_k sjednocení $l_1 \cup l_2$
- pro každou kombinaci c z K_k
  - pokud některá z jejich podkombinací délky $k-1$ není obsažena v L_{k-1} odstraň c z K_k

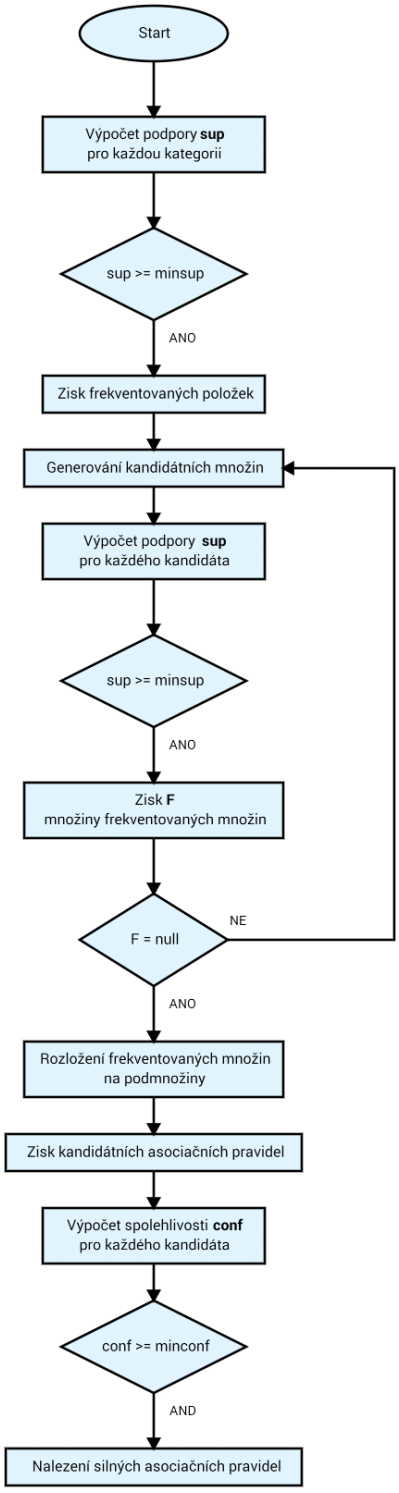
B) Vytváření asociačních pravidel

- každou kombinaci c z F (množina nalezených frekventovaných množin) rozděl na dvojice podkombinací *ant* a *cons* tak, že $cons = c - ant$
- vytvoř implikaci $ant \rightarrow cons$ a vypočítej její spolehlivost
- porovnej zjištěnou spolehlivost s požadovanou minimální spolehlivostí, musí platit vztah

$$conf(ant \rightarrow cons) \geq minconf$$

- přesouvej kategorie z *ant* do *cons* a vytvářej nová asociační pravidla

Blokové schéma algoritmu APRIORI



Obrázek 1: Blokové schéma algoritmu APRIORI

Algoritmus APRIORI je relativně jednoduchým algoritmem, jeho princip je dobře pochopitelný a implementace snadná. S jeho využitím dochází k urychlení generování frekventovaných množin a tím také získávání asociačních pravidel. Míra zefektivnění procesu ovšem není konstantní, závisí na struktuře zpracovávaných dat. Algoritmus nepracuje s žádnou pomocnou strukturou a výpočet podpory kandidátních množin stejně tak jako spolehlivosti implikací provádí na základě celého datového souboru, jehož opakované procházení tedy vyžaduje. V případě analýzy rozsáhlých datových souborů je to velká nevýhoda a efektivita algoritmu značně klesá.

Ke zdokonalení algoritmu APRIORI vznikla řada dalších optimalizací, které odstraňují některé jeho nedostatky. Příkladem je algoritmus APRIORI TID popsaný v následující kapitole.

5 Algoritmus APRIORI TID

APRIORI TID je jednou z optimalizací původního algoritmu APRIORI (viz kapitola 4). I tento algoritmus pracuje s datovými soubory, z nichž umožňuje čerpat zajímavé informace a skryté vztahy v podobě asociačních pravidel.

Obdobně jako APRIORI využívá i APRIORI TID při generování frekventovaných množin jejich apriori vlastnosti (platí, že všechny podmnožiny frekventované množiny jsou rovněž frekventované), tyto množiny ovšem nejsou sestavovány na základě původních dat, ale využívá se pomocné datové struktury (značeno C_k pro $k \in N$), která je vytvořena při startu algoritmu a za jeho běhu dále průběžně modifikována tak, aby vždy obsahovala pouze relevantní záznamy a docházelo k maximální možné redukci původního souboru.

Algoritmus pracuje následovně:

- Před zahájením vlastního algoritmu je na základě původního datového souboru vytvořena pomocná struktura C_k pro $k = 1$, v níž každý záznam původního souboru je interpretován jako množina obsahující jednoprvkové množiny tvořené položkami nacházejícími se v příslušném záznamu. Jako příklad uvažujme soubor transakcí uvedený v kapitole 3.2, pro který by pomocná datová struktura C_1 vypadala následovně:

Tabulka 5: APRIORI TID - Pomocná datová struktura C_1

ID zákazníka	*Nákup*	C_1
1	džem	{{džem}}
2	mléko	{{mléko}}
3	džem, chléb	{{džem}, {chléb}}
4	džem, chléb, mléko	{{džem}, {chléb}, {mléko}}

- Na základě pomocné struktury C_1 jsou nalezeny všechny frekventované množiny délky 1, které vznikají jako množiny tvořené jedním prvkem – některou z kategorií atributů. U těchto kandidátů je následně zjištěna četnost jejich výskytu v datech a dojde k vyřazení všech kandidátů, jejichž podpora není dostatečně vysoká.
- Postupně vznikají nové kandidátní množin délky k pro $k > 1$ jako sjednocení vhodných již objevených frekventovaných množin délky $k - 1$, které zároveň splňují apriori vlastnost (všechny podmnožiny kandidátní množiny jsou frekventované). Zároveň je postupně upravována a v ideálním případě také redukována pomocná struktura C_k, k čemuž dochází s každou inkrementací k ve chvíli, kdy jsou vytvořeny nové kandidátní množiny, a to následovně:

- Vznikají sjednocení délky k množin obsažených v jednotlivých transakcích ve struktuře C_k.
- Dojde k odstranění všech takových množin, které se nevyskytují v nově vzniklé množině kandidátních množin.
- Pokud některá transakce již neobsahuje žádnou množinu, je ze struktury C_k odstraněna.
- Pro výše zmíněná ukázková data (pokud by byly za frekventované množiny považovány všechny existující) by modifikace pomocné struktury C_k probíhala takto:

Tabulka 6: APRIORI TID - Pomocná datová struktura C_2

C_2
{{džem, chléb}}
{{džem, chléb}, {džem, mléko}, {chléb, mléko}}

Tabulka 7: APRIORI TID - Pomocná datová struktura C_3

C_3
{{džem, chléb, mléko}}

  - Po vygenerování kandidátů délky k je vypočtena jejich podpora na základě datové struktury C_k, zjištěná hodnota je porovnána s požadovanou minimální podporou a jsou nalezeny všechny frekventované množiny délky k.
  - Proces generování frekventovaných množin je ukončen v okamžiku dosažení maximální možné velikosti k, která je rovna počtu atributů.

Po nalezení všech frekventovaných množin jsou vytvářena vlastní asociační pravidla na základě předem stanoveného kritéria minimální spolehlivosti, stejně jako v případě algoritmu APRIORI tento proces neurychluje ani APRIORI TID a probíhá tedy tak, jak bylo popsáno v kapitole 3.2.2.

Průběh Igoritmu APRIORI TID

A) Generování frekventovaných množin

- vytvoř pomocnou datovou strukturu C_1 na základě původního datového souboru, kde každý záznam (transakce) je interpretován jako množina obsahující jednoprvkové množiny tvořené položkami, které se v něm nacházejí
- do množiny L_1 přiřaď všechny jednoprvkové množiny tvořené kategoriemi atributů, které dosahují požadované minimální podpory nebo ji převyšují, pro výpočet podpory využij pomocnou strukturu C_1
- polož $k = 2$
- dokud $L_{k-1} \neq \emptyset$
  - pomocí funkce ***apriori-gen*** vygeneruj na základě L_{k-1} množinu kandidátů K_k
  - modifikuj pomocnou strukturu C_k, kde každá množina reprezentující transakci bude obsahovat všechny množiny tvořené *k*-ticemi kategorií vznikající jako sjednocení některých dvou množin délky *k-1* příslušné transakce, zároveň obsažené také v K_k
  - do množiny L_k zařaď všechny takové kombinace z K_k, které vyhovují požadavku minimální podpory zjištěné na základě C_k
  - inkrementuj *k*

Funkce apriori-gen (L_{k-1})

- pro všechny dvojce kombinací l_1, l_2 z L_{k-1}
  - pokud se l_1 a l_2 shodují v *k-2* kategoriích, přidej do K_k sjednocení $l_1 \cup l_2$
- pro každou kombinaci *c* z K_k
  - pokud některá z podkombinací délky *k-1* není obsažena v L_{k-1}, odstraň *c* z K_k

B) Vytváření asociačních pravidel

- každou kombinaci c z *F* (množina nalezených frekventovaných množin) rozděl na dvojice podkombinací *ant* a *cons* tak, že *cons* = *c* – *ant*
- vytvoř implikaci *ant* → *cons* a vypočítej její spolehlivost
- porovnej zjištěnou spolehlivost s požadovanou minimální spolehlivostí, musí platit vztah

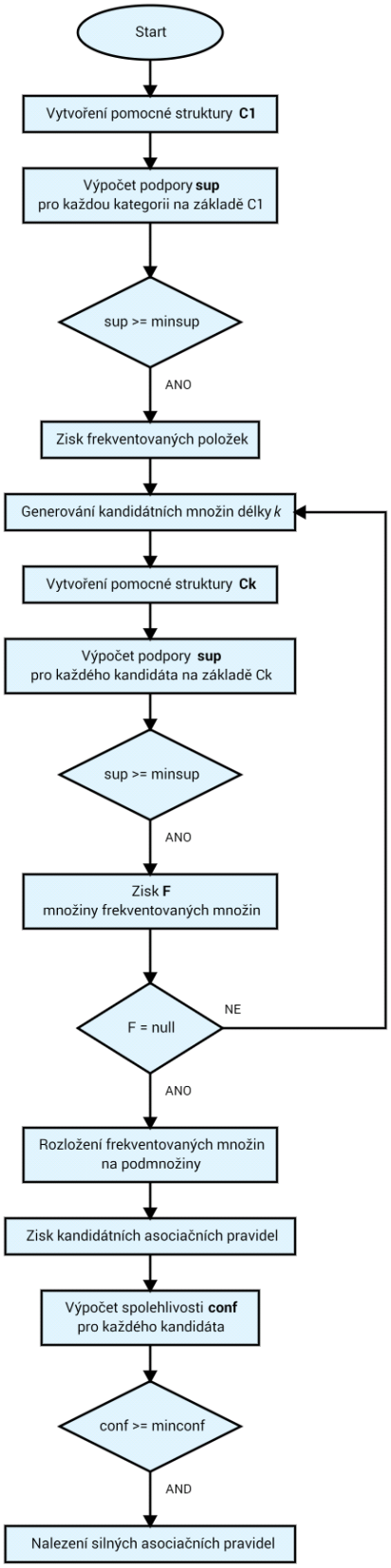
$$conf(ant \rightarrow cons) \geq minconf$$

- přesouvej kategorie z *ant* do *cons* a vytvářej nová asociační pravidla

Algoritmus APRIORI TID částečně řeší nedokonalost algoritmu APRIORI spočívající v nutnosti mnohonásobného opakovaného procházení původního, mnohdy velmi rozsáhlého, datového souboru během generování frekventovaných množin tím, že pracuje s pomocnou strukturou C_k pro $k \in N$, kterou postupně modifikuje a redukuje, tento proces ovšem vyžaduje určitý čas, který především pro nízká k není zcela zanedbatelný. Rychlost algoritmu tedy stejně jako v případě APRIORI není konstantní a závisí na struktuře zpracovávaných dat. Porovnání rychlosti algoritmů APRIORI a APRIORI TID je blíže diskutováno v praktické části práce

Další optimalizací algoritmu APRIORI je algoritmus APRIORI Hybrid kombinující vlastnosti APRIORI a APRIORI TID [5]. Tento algoritmus pracuje nejprve jako APRIORI, tedy generuje frekventované množiny na základě původního datového souboru, a teprve po určité době přejde do režimu APRIORI TID, kdy začne pracovat s pomocnou strukturou. Při vyhledávání asociačních pravidel tak dosahuje ještě lepších výsledků a vyšší efektivity.

Blokové schéma algoritmu APRIORI TID



Obrázek 2: Blokové schéma algoritmu APRIORI TID

6 Metoda GUHA

GUHA je zkratka pro *General Unary Hypothesis Automaton*, v překladu automat na obecné unární hypotézy. Jedná se o původní českou metodu pro systematické vytváření hypotéz na základě empirických dat pomocí vhodných procedur realizovaných na počítači umožňující automatické vytváření implikačních, asociačních a korelačních hypotéz. Tato metoda vznikla v 70. letech 20. století a je možné ji považovat za předchůdce asociačních metod, kterými se později zabýval R. Agrawal. Jejími autory jsou Petr Hájek, Tomáš Havránek a Metoděj Chytil.

Metoda pracuje s dvouhodnotovými daty, která vznikají z kategoriálních dichotomizací, tedy převodem kategoriální proměnné na indikátorovou – binární, v případě číselných dat je pak nutné nejprve je vhodným způsobem převést na kategoriální a následně na data dvouhodnotová. Transformaci číselných dat na kategorie nazýváme kategorizací a lze zvolit více postupů podle typu úlohy a definovaných cílů.

Základním principem metody GUHA je podle [6] *nabízet vše zajímavé*, tedy generovat hypotézy, které jsou zajímavé vzhledem k analyzovaným datům i k řešenému problému. Výsledkem jsou obecné multidimenzionální pravidla.

6.1 Důležité pojmy a princip metody

Důležité pojmy

- **Predikát** – symbolické jméno veličiny, elementární formule, je chápán jako vlastnost nebo vztah
- **Formule** – predikáty složené pomocí logických spojek negace (¬), konjunkce (∧) nebo disjunkce (∨)
- **Kvantifikátor** – symbol určující druh a kvantitativní intenzitu souvislosti, tedy jak silný daný vztah je
- **Formální sentence** – pravidlo, zapsané ve tvaru $\boldsymbol{f_1\ q\ f_2}$**.** kde f_1 a f_2 jsou formule a q je kvantifikátor, jehož pravdivost v datech se testuje
- **Pravdivá sentence** (hypotéza) – sentence, jejíž pravdivost je potvrzena vyhodnocením kvantifikátoru v datech (hodnota kvantifikátoru je rovna 1)
- **Antecedent** (předpoklad) – predikát vyskytující se uvnitř formule na levé straně sentence, tedy před kvantifikátorem
- **Sukcedent** (závěr) – predikát vyskytující se uvnitř formule na pravé straně sentence, tedy za kvantifikátorem

Princip metody

Metoda GUHA systematicky generuje všechny hypotézy ze zadaných množin antecedentů a sukcedentů a následně testuje, zda jsou podporovány zpracovávanými daty.

Pro analýzu dat jsou podstatné charakteristiky veličin v rámci celých dat, nikoli hodnoty jednotlivých objektů. Podle těchto charakteristik = frekvencí a podle zvoleného kvantifikátoru se pak usuzuje, zda je daná sentence v datech pravdivá anebo ne.

Nechť M je tabulka vzniklá pozorováním n dvouhodnotových veličin $X_1, X_2, \dots X_n$. Pro každou n-tici možných hodnot veličin, tj. $e = (e_1, e_2, \dots e_n) \in \{0, 1\}^n$ je frekvence $fr(e, M)$ definována jako počet objektů z M, pro které jsou hodnoty veličin rovné $e = (e_1\ e_2, \dots e_n)$. [6] Řádky tabulky v citovaném textu označované jako M jsou jednotlivá pozorování, kterých je m.

Pro $n = 2$, (uvažujeme dvě dvouhodnotové veličiny X_1 a X_2) jsou definovány čtyři frekvence a, b, c, d následovně:

- a – počet objektů, pro které $X_1 = X_2 = 1$
- b – počet objektů, pro které $X_1 = \neg X_2 = 1$
- c – počet objektů, pro které $\neg X_1 = X_2 = 1$
- d – počet objektů, pro které $\neg X_1 = \neg X_2 = 1$

Schematicky lze frekvence a, b, c, d zapsat do formy čtyřpolní tabulky:

Tabulka 8: GUHA – čtyřpolní frekvenční tabulka

	X_2	$\neg X_2$	$\sum$
X_1	a	b	$a + b$
$\neg X_1$	c	d	$c + d$
$\sum$	$a + c$	$b + d$	m

Kvantifikátory

Kvantifikátory jsou symboly určující kvantitativní intenzitu souvislosti [6]. Představují druh zjištěného vztahu mezi antecedentem a sukcedentem v datech a stanovují, jak silný tento vztah je. Mohou být podle typu prezentovaného pravidla trojího typu: implikační, asociační nebo korelační.

A) Implikační kvantifikátory

- značení: $A \Rightarrow B$; slovní formulace: A (asi, většinou) je příčinou B; platí-li A, pak platí B; B vyplývá z A
- ***Fundovaná implikace*** (pro p ∈ (0; 1⟩, $s > 0$)

  $\Rightarrow_{p,s} (a, b, c, d) = 1$ je-li $a \geq s$ a $a \geq p^*(a + b)$

  Jde o to, aby podíl frekvence případů, kdy současně platí antecedent i succedent, a frekvence případů, kdy platí antecedent, byl větší nebo roven p. Parametr s pak vylučuje ze statistického hlediska nezajímavé případy.

- ***Dolní kritická implikace*** (pro $p \in (0; 1\rangle$, $s > 0$, $\alpha \in \langle 0; 0{,}5\rangle$)

  $\Rightarrow^{!}_{p,s,\alpha}(a,b,c,d) = 1$ je-li $\sum_{i=a}^{r}\binom{r}{i}p^i(1-p)^{r-i} \leq \alpha$

  Jde o test na hladině významnosti α, který spočívá v testování nulové hypotézy, že podmíněná pravděpodobnost platnosti sukcedentu v případě platnosti antecedentu je menší nebo rovna *p*, proti alternativní hypotéze, že je tato hodnota větší než *p* [6]. Vyhodnocení 1 indikuje přijetí alternativní hypotézy a znamená, že přítomnost antecedentu zvyšuje pravděpodobnost nastání sukcedentu.

  Dolní kritická implikace zajišťuje, že pravděpodobnost vygenerování pravidla $A \Rightarrow B$ při jeho současné neplatnosti v datech je rovna hodnotě α.

- ***Horní kritická implikace*** (pro $p \in (0; 1\rangle$, $s > 0$, $\alpha \in \langle 0; 0{,}5\rangle$)

  $\Rightarrow^{?}_{p,s,\alpha}(a,b,c,d) = 1$ je-li $\sum_{i=0}^{a}\binom{r}{i}p^i(1-p)^{r-i} > \alpha$

  Jde o test na hladině významnosti α, který spočívá v testování nulové hypotézy, že podmíněná pravděpodobnost platnosti sukcedentu v případě platnosti antecedentu je větší nebo rovna *p*, proti alternativní hypotéze, že je tato hodnota menší než *p* [6]. Vyhodnocení 1 indikuje nezamítnutí nulové hypotézy, nelze tedy vyloučit, že přítomnost antecedentu zvyšuje pravděpodobnost nastání sukcedentu.

  Horní kritická implikace zajišťuje, že pravděpodobnost nevygenerování pravidla $A \Rightarrow B$ při jeho současné platnosti v datech je rovna hodnotě α.

B) Asociační kvantifikátory

- značení: $A \sim B$; slovní formulace: A (asi, většinou) souvisí s B
- ***Prosté vychýlení*** (pro $\delta \geq 0$)

  $\sim_{\delta}(a,b,c,d) = 1$ je-li $ad > e^{\delta}bc$

- ***Fisherův kvantifikátor*** (pro $\alpha \in \langle 0; 0{,}5\rangle$)

  $\sim^{1}_{\alpha}(a,b,c,d) = 1$ je-li $ad > bc$ a $\sum_{i=a}^{\min(r,k)}\frac{\binom{k}{i}\binom{m-k}{r-i}}{\binom{m}{r}} \leq \alpha$

  Jde o test na hladině významnosti α založený na testování hypotézy, že mezi veličinami neexistuje závislost proti alternativní hypotéze, že mezi nimi závislost existuje. Vyhodnocení 1 indikuje přijetí alternativní hypotézy.

- ***χ^2-kvantifikátor*** (pro $\alpha \in \langle 0; 0{,}5\rangle$)

  $\sim^{2}_{\alpha}(a,b,c,d) = 1$ je-li $ad > bc$ a $\frac{(ad-bc)^2}{rkls}m \geq \chi^2_{\alpha}$

  χ^2_{α} je (1 - 2 α)-kvantil χ^2-rozložení s jedním stupněm volnosti.

  Stejně jako v případě Fisherova kvantifikátoru jde o test na hladině významnosti α založený na testování hypotézy, že mezi veličinami neexistuje závislost proti alternativní hypotéze, že závislost existuje, zde se ovšem jedná o test asymptotický [6]. Vyhodnocení 1 indikuje přijetí alternativní hypotézy.

C) Korelační kvantifikátory

- značení: $A\ corr\ B\ /\ F$; slovní formulace: za podmínky *F* spolu hodnoty *A* a *B* (asi, většinou) korelují
- ***Spearmanův kvantifikátor*** (pro $\alpha \in \langle 0;\ 0{,}5\rangle$)

  $s\text{-}ccorr_\alpha(\langle t_1, t_2\rangle) = 1$ je-li $\sum_{i=1}^{m} R(i)Q(i) \geq k_\alpha$

  Zde t_1 a t_2 jsou reálněhodnotové veličiny, *R(i)* počet objektů, pro které je hodnota t_1 menší než t_1 pro i-tý objekt, *Q(i)* počet objektů, pro které je hodnota t_2 menší než t_2 pro i-tý objekt a k_α je vhodně zvolená konstanta.

  Jde o test nulové hypotézy nezávislosti veličin t_1 a t_2 proti alternativní hypotéze o jejich kladné závislosti [6]. Vyhodnocení 1 indikuje přijetí alternativní hypotézy.
- ***Kendallův kvantifikátor*** (pro $\alpha \in \langle 0;\ 0{,}5\rangle$)

  $k\text{-}corr_\alpha(\langle t_1, t_2\rangle) = 1$

  je-li $\sum_{i \neq j}(sign(R(i) - R(j)).sign(Q(i) - Q(j)) \geq k_\alpha$

  Význam t_1, t_2, *R(i)*, *Q(i)* a k_α je stejný jako v případě Spearmanova kvantifikátoru, rovněž tak i statistická interpretace.
- ***Pořadově ekvivalenční kvantifikátor***

  $e\text{-}ccorr_\alpha(\langle t_1, t_2\rangle) = 1$ je-li *R(i)* = *Q(i)* pro i = 1,… m

  Význam t_1, t_2, *R(i)*, *Q(i)* a k_α je opět stejný jako v případech výše.

Dedukční pravidla

Při hledání asociací využívá metoda GUHA dedukčních pravidel a redukuje množinu sentencí, které je nutné testovat v datech (eliminuje sentence, které je možné odvodit z jiných hypotéz). Dochází tak k určitému zefektivnění celého procesu.

A) Pravidlo záměny ekvivalentních formulí

Nechť *S(F)* je sentence pravdivá v datech *M* obsahující formuli *F*. Pak formuli *F* je možné nahradit formulí *F'* ekvivalentní formuli *F*. Vzniklá sentence *S'(F')* je v datech *M* opět pravdivá.

B) Pravidlo úprav elementární implikace

Nechť implikace $A \Rightarrow^* B$ je pravdivá v datech *M*. Pak je možné:

- převést členy z antecedentu do sukcedentu za současné změny znaku negace u příslušného členu
- přidat nové členy do sukcedentu

C) Pravidlo symetrie

Nechť sentence $A \sim^* B$ je pravdivá v datech M a $\sim^*$ je symetrický kvantifikátor. Pak sentence $B \sim^* A$ vzniklá záměnou antecedentu a sukcedentu je rovněž pravdivá v datech M.

D) Pravidlo konzervativního zlepšování

Nechť S je sentence pravdivá v datech M obsahující konjunkci K. Platí, že do K je možné přidat nové členy nebo jejich negace, které K konzervativně zlepšují[1], a výsledná sentence S' je opět pravdivá v datech M.

E) Pravidlo ostrého zlepšování pro konjunktivní asociace

Nechť $A \sim^* B$ je sentence pravdivá v datech M, kde A a B jsou konjunkce. Pak lze do konjunkce antecedentu A i sukcedentu B přidat nové členy nebo jejich negace, které ji ostře zlepšují[2], a výsledná sentence je v datech M opět pravdivá.

Neúplná informace

Metoda GUHA je schopná pracovat vedle úplné i s neúplnou informací, kdy tedy v případě dvouhodnotových dat M mohou členy nabývat třech hodnot: 0, 1 nebo X, kde X je neznámá hodnota. Pro data M je nutné provést tzv. dvouhodnotové doplnění, což je každá taková tabulka M', která vznikne nahrazením všech X hodnotami 0 nebo 1 [6]. V úvahu je nutné brát všechna možná dvouhodnotová doplnění M' původních dat M.

Pro vyhodnocení formulí složených z predikátů obsahujících neúplnou informaci platí následující pravidla:

$$\neg p = \begin{cases} 1 & pro\ p = 0 \\ 0 & pro\ p = 1 \\ X & pro\ p = X \end{cases}$$

$$p_1 \wedge \ldots \wedge p_n = \begin{cases} 1 & pro\ p_1 = \ldots = p_n = 1 \\ 0 & pokud\ některé\ p_i = 0 \\ X & pro\ zbylé\ případy \end{cases}$$

$$p_1 \vee \ldots \vee p_n = \begin{cases} 1 & pokud\ některé\ p_i = 1 \\ 0 & pro\ p_1 = \ldots = p_n = 0 \\ X & pro\ zbylé\ případy \end{cases}$$

[1] Nechť K je elementární konjunkce pravdivá v datech M. Pak predikát p konzervativně zlepšuje K, jestliže se p nevyskytuje v K a platí, že formule $K \wedge p$ je ekvivalentní formuli K v datech M.

[2] Nechť $A \sim^* B$ je konjunktivní asociace pravdivá v datech M. Predikát p_1 ostře zlepšuje antecedent A, jestliže se p nevyskytuje v A a platí, že formule $A \wedge B$ je ekvivalentní formuli $A \wedge B \wedge p_1$ v datech M. Obdobně predikát p_2 ostře zlepšuje sukcedent B, jestliže se p nevyskytuje v B a platí, že formule $A \wedge B$ je ekvivalentní formuli $A \wedge B \wedge p_2$ v datech M. Dále platí, že jestliže predikát p konzervativně zlepšuje antecedent (succedent) konjunktivní asociace, pak jej zlepšuje také ostře.

Funkce kvantifikátoru q nad daty M je definována takto:

$$q(M)\begin{cases} 1 & pokud\ q(M') = 1\ pro\ všchna\ doplnění\ M'dat\ M \\ 0 & pokud\ q(M') = 0\ pro\ všchna\ doplnění\ M'dat\ M \\ X & pro\ zbylé\ případy \end{cases}$$

Pro elementární konjunkci, elementární disjunkci a kvantifikátory platí tzv. princip zabezpečení, tedy že každá formule *F* nabývá pro libovolný objekt dvouhodnotových dat *M* obsahujících neúplnou informaci hodnoty *X* právě tehdy, existuje-li dvouhodnotové doplnění *M'* dat *M*, kde *F* = 1 a jiné dvouhodnotové doplnění *M''* dat *M*, kde *F* = 0 [6].

V případě dat s neúplnou informací se pracuje s devítipolní frekvenční tabulkou:

Tabulka 9: GUHA – devítipolní frekvenční tabulka

	1	*X*	*0*	∑
1	*a*	*i*	*b*	*r*
X	*o*	*m*	*p*	
0	*c*	*j*	*d*	*s*
∑	*k*		*l*	

Přechod k některému dvouhodnotovému doplnění *M'* původních dat *M* se projeví následovně:

- *i* bude dílem přičteno k *a* a dílem k *b*
- *j* bude dílem přičteno k *c* a dílem k *d*
- *o* bude dílem přičteno k *a* a dílem k *c*
- *p* bude dílem přičteno k *b* a dílem k *d*
- *m* bude dílem přičteno k *a, b, c* a *d*

6.2 Procedury metody

Metoda GUHA se skládá z velkého množství procedur. První část z nich se využívá k předzpracování původních dat, odstranění chyb, adaptaci tvaru a formy dat a k vytváření nového datového souboru na magnetickém médiu (původní podoba metody byla navržena pro práci s daty zadanými v písemné formě). K tomuto účelu slouží následující pětice procedur:

- **Procedura DICHOT** převádí původní vstupní data na dvouhodnotové veličiny na základě zvolených parametrů a vytváří soubor informací o veličinách CINFQ, který bude vstupem procedury G1.
- **Procedura G1** slouží pro čtení a kontrolu děrnoštítkového souboru CINFQ a pro vytvoření nového souboru INFQ informací o veličinách uloženém na magnetickém médiu určeném pro další zpracování.

- **Procedura G2** využívá výstupy procedur DICHOT a G2, jejichž sloučením vytváří nový soubor obsahující datovou matici a informace o datech.
- **Procedura G3** převádí soubor získaný pomocí procedury G2 na tvar potřebný pro zpracování dalšími procedurami (ASSOC, IMPL, CORREL).
- **Procedura FREQ** slouží pro vytvoření souboru FTAB obsahujícím frekvenční tabulku pro vybrané veličiny.

Nejdůležitější a pravděpodobně také nejzajímavější skupinou procedur metody jsou tzv. vlastní GUHA procedury, tedy procedury sloužící ke generování hypotéz a ověřování jejich pravdivosti v datech. Jedná se o čtyři procedury: ASSOC, IMPL, COLLAPS a CORREL.

- **Procedura ASSOC** slouží k vyhledávání asociací, tedy sentencí s asociačními kvantifikátory, kde antecedent a succedent jsou formule ve tvaru elementárních konjunkcí, v dvouhodnotových datech. Výstupem procedury jsou všechny sentence pravdivé v datech, generované od nejjednodušších po nejsložitější, vyhovující předem zadaným kritériím. Antecedent i succedent musí obsahovat alespoň jeden důležitý predikát (jejich množina je zadána před zahájením procesu) a současně libovolné množství dalších predikátů v povolených tvarech.
- **Procedura IMPL** slouží k vyhledávání sentencí s implikačními kvantifikátory, kde antecedent je formule ve tvaru elementárních konjunkce a succedent je formule ve tvaru elementární disjunkce, v dvouhodnotových datech. Výstupem procedury jsou obdobně jako v případě procedury ASSOC jsou všechny sentence pravdivé v datech splňující předem určené požadavky, kde antecedent i succedent obsahují alespoň jeden důležitý predikát a libovolné množství dalších predikátů v povolených tvarech.
- **Procedura COLLAPS** slouží k vyhledávání zdrojů závislosti, tedy dvojic predikátů, které spolu co nejvíce souvisejí a jejichž numerické charakteristiky vyhovují zadaným podmínkám, v nominálních datech. Výstupem procedury je binární strom nejlepších hypotéz, který vyjadřuje strukturu závislosti v datech.
- **Procedura CORREL** slouží k vyhledávání elementárních konjunkcí, pro které je podmíněná korelace dvou vybraných reálných veličin v datech vysoká. Výstupní sentence jsou tvaru

  $$(p_1\ corr\ p_1)/f$$

  kde p_1 a p_2 jsou zvolené reálněhodnotové veličiny, *corr* zvolený korelační kvantifikátor a *f* podmínka v podobě elementární konjunkce predikátů požadovaného tvaru obsahující alespoň jeden důležitý predikát pravdivá v datech.

K vyhodnocení výstupu některé z výše zmíněných procedur metody GUHA slouží procedura REPORT, která provádí přehledný tisk záznamu uloženého na magnetickém médiu, a to buď v původním uspořádání tak, jak byly hypotézy generovány, anebo v jiném pořadí s využitím dalších procedur (např. MERGE nebo SORT). K interpretaci získaných výsledků pak slouží procedura INTERP.

Každá relevantní pravdivá sentence vyjadřující v datech ověřený vztah mezi veličinami odpovídá nějaké teoretické hypotéze o vztahu těchto veličin v univerzu (analyzovaná data jsou výběrem z tohoto celku). O platnosti této teoretické hypotézy lze rozhodnout na základě vyhodnocení datového vzorku, ovšem její pravdivost nikdy zcela zaručena není.

Metoda GUHA byla implementována ve formě různých programů na řadu softwarových platforem, například systém LISp-Miner [7]. Výstupem metody jsou hypotézy svou podobou blízké asociačním pravidlům, jímž je věnovaná většina této diplomové práce. Hlavní rozdíl mezi klasickými asociačními pravidly a vztahy získanými s využitím metody GUHA spočívá v tom, že při vyhledávání klasických asociačních pravidel jsou pro posuzování kvality a zajímavosti používány pouze dvě míry intenzity asociačního pravidla, tj. podporu a spolehlivost, zatímco metoda GUHA nabízí mnohem více možností hodnocení testovaných hypotéz.

Metoda GUHA je velice komplexní a obsahuje mnoho podnětů a přístupů. Evidentně předběhla vývoj ve světě, jednalo se o první z metod vytěžování dat vůbec a obdobnou myšlenkou asociací se zabývala skupina kolem R. Agravala až o desetiletí později, je proto škoda, že v konečné míře zůstala vcelku neznámá. Zpracování metody GUHA pro předmět *Data mining* by mělo přiblížit studentům tento český příspěvek do problematiky velkých dat, na který můžeme být jako národ hrdi.

7 Praktická část práce

V této práci jsem se zabývala vzory chování popsatelnými pomocí asociačních pravidel. Cílem práce bylo seznámit se s problematikou asociačních metod, s asociačními pravidly a se způsoby, jak tato pravidla získávat z datových souborů a také jak je dále využívat. Mým úkolem bylo naprogramovat vybrané z algoritmů, které proces vyhledávání asociačních pravidel realizují a urychlují, v programovacím jazyce C#.

Pro praktickou část práce jsem si zvolila algoritmus APRIORI, pravděpodobně nejznámější prostředek pro urychlení vyhledávání asociačních pravidel vůbec, a jednu z jeho optimalizovaných verzí – algoritmus APRIORI TID. Oba algoritmy jsou relativně jednoduché, snadno pochopitelné, dobře implementovatelné a jejich požadavky na výpočetní prostředky nejsou příliš vysoké.

Zaměřila jsem se na dva typy úloh: analýzu preferencí studentů ve výběru volitelných předmětů provedenou na základě reálných dat získaných ze STAGU Technické univerzity v Liberci a analýzu nákupních preferencí zákazníků, tzv. analýzu nákupního košíku, provedenou na základě uměle vytvořených dat.

Program, který jsem v rámci své práce vytvořila, slouží jako výukový nástroj usnadňující pochopení problematiky asociačních metod studentům předmětu *Data mining*. Demonstruje průběh výše zmíněných algoritmů a umožňuje jejich aplikaci na konkrétní data, přičemž uživatel má možnost použít buď reálná data (musí být v požadovaném formátu) anebo vytvářet vlastní datové soubory obsahující fiktivní transakční data (viz kapitola 3.2). Nalezená asociační pravidla lze pak v rámci programu dále podrobněji zkoumat a hodnotit prostřednictvím jejich číselných charakteristik, jejichž výpočty jsou rovněž vysvětleny a demonstrovány.

Mezi další funkce programu patří porovnávání rychlostí, kterých dosahují algoritmy APRIORI a APRIORI TID při vyhledávání asociačních pravidel v datových souborech a ukázka způsobu, jak lze nalezená pravidla využívat v praxi při sestavování reklamních nabídek obchodů.

Program nabízí řadu grafických vizualizací, které znázorňují a ještě více usnadňují pochopení následujících problémů:

- způsob získávání asociačních pravidel z dat, a to buď samotný průběh tohoto procesu bez použití algoritmu, který by jej urychlil, anebo s pomocí algoritmu APRIORI
- postupy výpočtů některých numerických charakteristik asociačních pravidel, konkrétně výpočtu podpory (support), spolehlivosti (confidence), navýšení (lift) a uplatnění (deployability)

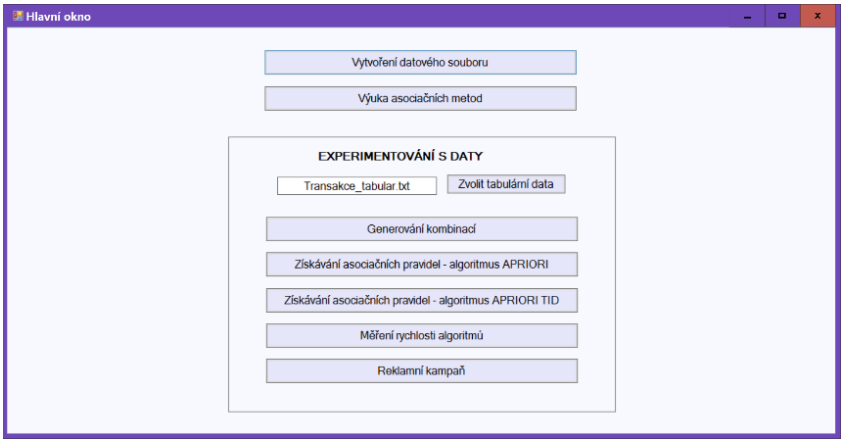
Program je dle mého názoru přehledný a snadno ovladatelný. Poskytuje studentům možnost zabývat se problematikou asociačních metod nejen teoreticky v rámci výuky *Data miningu*, ale také prakticky. Umožňuje hlouběji zkoumat a porozumět způsobu fungování algoritmů pro vyhledávání asociačních metod, nikoli pouhou aplikaci vybraného algoritmu na data bez znalosti jeho hlubší podstaty a principu.

Druhou částí mé práce bylo seznámit se s metodou GUHA, jíž se zabývala v 70. letech 20. století skupina českých vědců P. Hájek, M. Chytil a I. Havel. Jedná se o původní českou metodu pro systematické vytváření hypotéz na základě empirických dat umožňující automatické vytváření implikačních, asociačních a korelačních pravidel. Mým úkolem zde bylo zpracovat toto téma pro e-learningový kurz *Data miningu*.

7.1 Struktura a funkce programu

7.1.1 Úvodní obrazovka

Po spuštění aplikace se uživateli zobrazí úvodní obrazovka, odkud má přístup ke všem hlavním funkcím programu. Struktura tohoto okna je následující:



Obrázek 3: Úvodní obrazovka

První tlačítko odkazuje do části aplikace umožňující vytváření datových souborů obsahujících fiktivní transakce. Tato funkce je užitečná v případě, že uživatel nemá k dispozici reálná data, která by mohl analyzovat a aplikovat na ně některý z naprogramovaných algoritmů.

Druhé tlačítko odkazuje do výukové části aplikace, kde jsou nejdůležitější pojmy související s problematikou asociačních metod a pravidel vysvětleny prostřednictvím grafických vizualizací na malých datových maticích tak, aby byly způsoby jejich řešení dobře viditelné a snadno pochopitelné. Konkrétně je zde demonstrován průběh generování frekventovaných množin, zisku asociačních pravidel a výpočtu jejich číselných charakteristik a dále také princip algoritmu APRIORI.

Zbylá tlačítka odkazují do částí programu, kde je zpracováván datový soubor obsahující reálná, případně uživatelem vytvořená, transakční data. Procesy, které je možné s těmito soubory provádět, jsou následující:

- Generování kombinací produktů obsažených ve vybraném souboru
- Získávání asociačních pravidel pomocí algoritmu APRIORI
- Získávání asociačních pravidel pomocí algoritmu APRIORI TID
- Měření a porovnávání rychlostí, kterých dosahují algoritmy APRIORI a APRIORI TID při hledání asociačních pravidel
- Využití algoritmu APRIORI k nalezení zajímavých asociačních pravidel a vytváření reklamních nabídek na základě stanovených požadavků

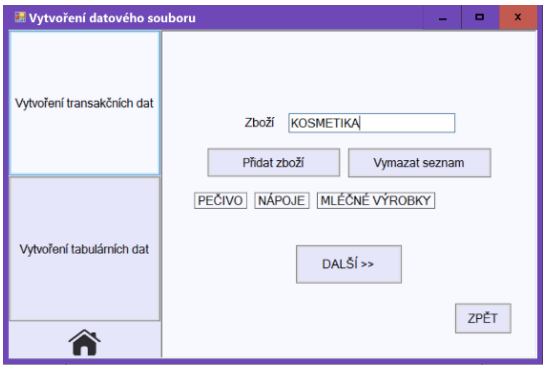
Všechny výše zmíněné funkce budou podrobněji popsány v následujících kapitolách.

7.1.2 Vytváření datových souborů

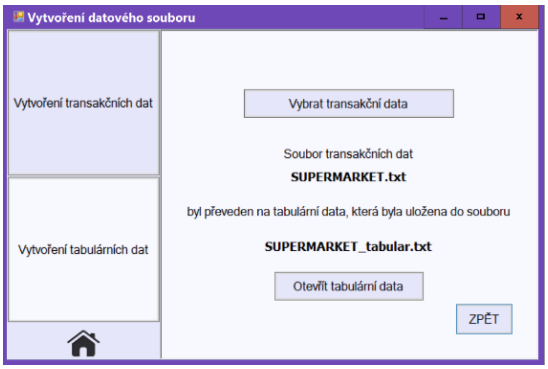
Program slouží jako podpůrný nástroj pro pochopení problematiky asociačních metod, asociačních pravidel a principu fungování algoritmů sloužících k jejich rychlejšímu hledání. Aby bylo možné studovat tyto problémy experimentálním způsobem, nikoli pouze teoreticky, je zapotřebí mít k dispozici datové soubory, ideálně data reálná, jejichž zisk může ovšem být značně problematický.

Z tohoto důvodu byla do aplikace implementována funkce vytváření vlastních souborů fiktivních transakčních dat. Jednotlivé druhy zboží (v libovolném množství), které bude výsledný soubor obsahovat, volí sám uživatel, stejně jako počet zákazníků, pro něž se transakce náhodně vygenerují. Tímto způsobem vytvořený datový soubor bude ve formátu transakčním (rozdíl mezi transakční a tabulární podobou dat z hlediska data miningu je vysvětlen v kapitole 3.2 a samozřejmě i v samotné aplikaci), takže je následně nutné převést jej v programu na formát tabulární, který jsou vhodnější pro další zpracovávání.

Výstupy z této části aplikace vypadají následovně:



Obrázek 4: Zisk transakčních dat



Obrázek 5: Zisk tabulárních dat

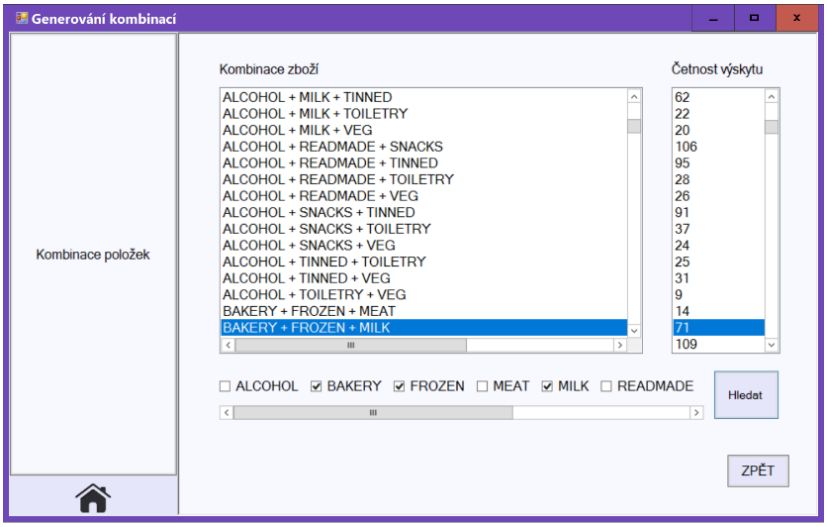
Bylo by jistě možné (a z pohledu uživatele pravděpodobně i pohodlnější) generovat soubory přímo ve formátu tabulárním, mým cílem však bylo co nejvíce se přiblížit realitě a vytvářet soubory primárně ve stejném formátu, v jakém je ukládají skutečné obchodní řetězce.

7.1.3 Generování kombinací

V této části aplikace jsou popsány tři způsoby, jimiž lze generovat kombinace položek obsažených v datovém souboru: generování kombinací do šířky, generování kombinací do hloubky a generování kombinací podle četností (viz kapitola 3.2.1). V programu je pak implementován algoritmus pro první z výše zmíněných způsobů, tedy generování do šířky, kdy je na počátku získán seznam všech samostatných kategorií atributů (v případě analýzy nákupního košíku všech druhů nabízeného zboží), který je dále postupné rozšiřování o dvouprvkové, tříprvkové a víceprvkové množiny.

Program analyzuje uživatelem zvolený datový soubor, nalezne všechny kategorie atributů, které obsahuje, následně vygeneruje veškeré existující kombinace kategorií atributů a spočítá četnost, s jakou se v datech vyskytují. Nalezené kombinace jsou řazeny od nejkratších po nejdelší, ve výsledném seznamu je navíc možné rychle vyhledávat označením položek, které by měla obsahovat hledaná množina a kliknutím na tlačítko *Hledat*.

Výstup z této části aplikace vypadá následovně:



Obrázek 6: Generování kombinací produktů

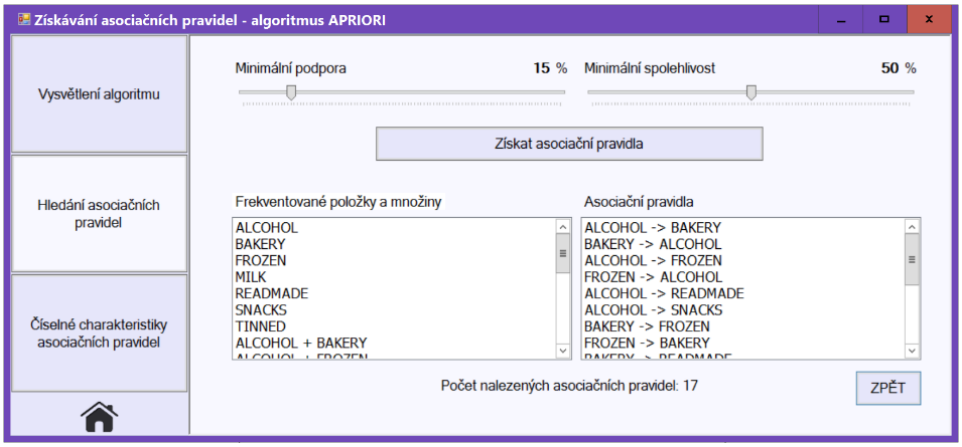
Generování kombinací samo o sobě není nikterak zajímavou úlohou nehledě na fakt, že jejich nalezením nedochází k získání žádných zajímavých informací ani odkrývání nových vztahů, což je podstatou data miningu. Nicméně tento proces je nezbytnou součástí získávání asociačních pravidel, proto nebylo možné tento bod v práci opomenout.

7.1.4 Získávání asociačních pravidel pomocí algoritmu APRIORI

Hlavním cílem mé práce bylo naprogramování vybraných algoritmu pro získávání asociačních pravidel z dat. Prvním algoritmem, který jsem si pro tento účel zvolila, byl algoritmus APRIORI patřící pravděpodobně mezi nejznámější algoritmy urychlujícími vyhledávání implikací vůbec.

Tato část programu se skládá ze tří podčástí. První z nich slouží především jako podpůrný nástroj vhodný pro výukové účely, algoritmus APRIORI je zde podrobně popsán, každý jeho krok je vysvětlen, doplněn vzorovým příkladem usnadňujícím jeho pochopení a posléze demonstrován na vybraném datovém souboru. Nejprve jsou vygenerovány frekventované množiny produktů odpovídající kritériu uživatelem určené minimální požadované podpory s využitím jejich apriori vlastnosti (viz kapitola 4), z těchto množin jsou poté získávána vlastní asociační pravidla splňující podmínku minimální požadované spolehlivosti.

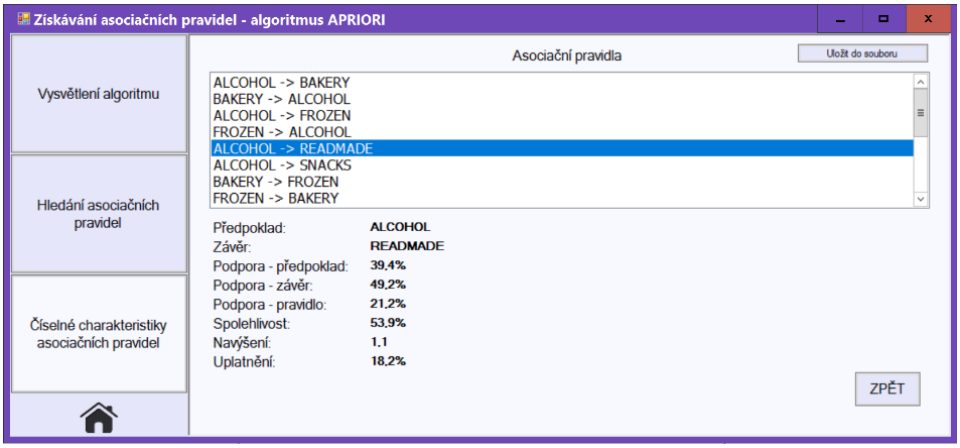
Druhá podčást slouží pro přímé získání asociačních pravidel dosahujících požadovaných hodnot podpory a spolehlivosti z datového souboru pomocí algoritmu APRIORI. Uživatel nastaví vlastnosti, jímž musí vyhovovat nalezené implikace, klikne na tlačítko *Získat asociační pravidla* a program vrátí dva seznamy: vygenerované frekventované množiny a z nich sestavená asociační pravidla včetně informace o jejich celkovém počtu. Výstup pak vypadá tak, jak je zobrazeno níže.



Obrázek 7: Získávání asociačních pravidel pomocí algoritmu APRIORI

Podrobnější informace o nalezených asociačních pravidlech jsou k dispozici ve třetí podčásti programu, která se věnuje výpočtu numerických charakteristik implikací. Uživatel vybere ze seznamu asociační pravidlo, o něž se zajímá, a program vrátí následující údaje: podpora předpokladu pravidla, podpora závěru pravidla, podpora celého pravidla, spolehlivost, navýšení a uplatnění. Získané výsledky má uživatel rovněž možnost uložit do textového soboru.

Výstup z této části aplikace vypadá následovně:



Obrázek 8: Numerické charakteristiky asociačních pravidel

7.1.5 Získávání asociačních pravidel pomocí algoritmu APRIORI TID

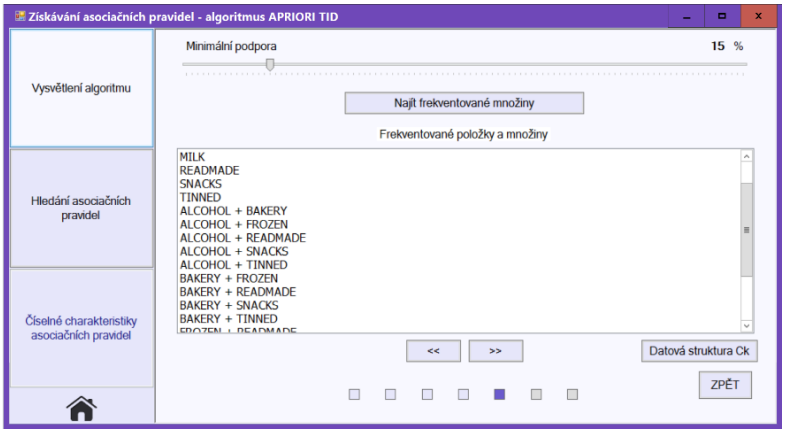
Algoritmus APRIORI TID je druhým algoritmem zefektivňujícím získávání asociačních pravidel z dat, kterým jsem se v rámci své práce zabývala a který jsem naprogramovala.

Jedná se o optimalizaci původního algoritmu APRIORI odstraňující jeho nedostatek spočívající v nutností mnohonásobného procházení původního datového souboru během generování frekventovaných množin.

Tato část programu se skládá obdobně jako část věnovaná algoritmu APRIORI ze tří podčástí. První z nich slouží opět jako podpůrný nástroj vhodný pro výukové účely. Je zde podrobně popsán princip fungování algoritmus APRIORI TID, každý jeho krok je vysvětlen, pro názornost doplněn vzorovými příklady usnadňujícími jeho pochopení. a posléze demonstrován na vybraném datovém souboru.

Nejprve je na základě původních dat vytvořena pomocná struktura C_1 (viz kapitola 5), s níž algoritmus dále pracuje během generování frekventovaných množin produktů odpovídajícím kritériu předem stanovené minimální požadované podpory s využitím jejich apriori vlastnosti. Pomocná struktura je průběžně modifikována a redukována, což má uživatel v rámci aplikace rovněž možnost sledovat a lépe tak porozumět způsobu, jakým k této redukci dochází.

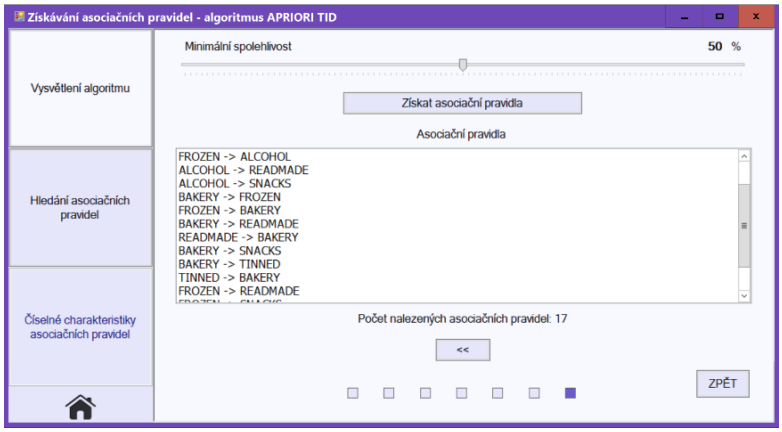
Podoba některých výstupů z této podčásti je následující:



Obrázek 9: Algoritmus APRIORI TID – generování frekventovaných množin

Obrázek 10: Algoritmus APRIORI TID – pomocná datová struktura

Z vygenerovaných frekventovaných množin jsou následně získávána vlastní asociační pravidla splňující podmínku minimální požadované spolehlivosti stejným způsobem jako v případě algoritmu APRIORI: každá frekventovaná množina je rozložena na podmnožiny, z kterých jsou sestavována kandidátní pravidla, na základě původních dat je zjišťována jejich spolehlivost a ta je porovnávána se stanovenou minimální hodnotou. Nalezené implikace vrací program uživateli v podobě seznamu, jak je vidět na obrázku níže:



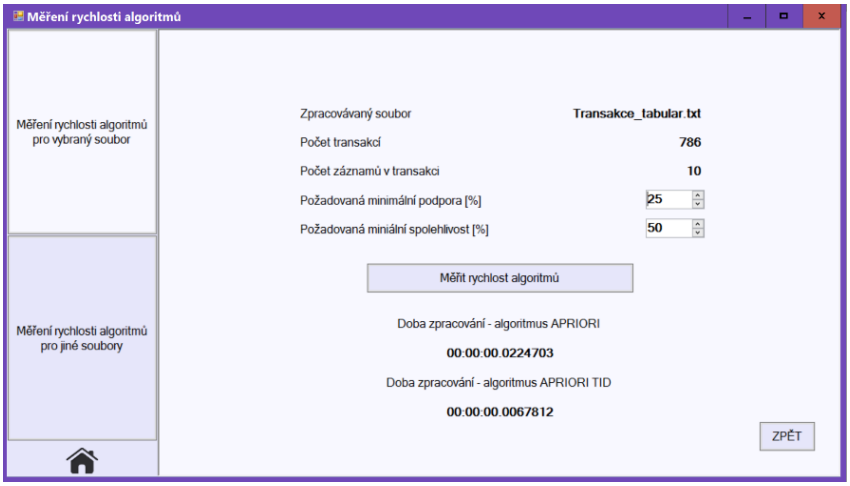
Obrázek 11: Algoritmus APRIORI TID – asociační pravidla

Zbývající dvě podčásti části aplikace věnující se algoritmu APRIORI TID jsou stejné jako v případě algoritmu APRIORI. Druhá podčást tedy slouží pro přímé získání asociačních pravidel dosahujících požadovaných hodnot podpory a spolehlivosti z datového souboru pomocí algoritmu APRIORI TID a třetí umožňuje zjišťovat o těchto pravidlech podrobnější informace v podobě jejich numerických charakteristik.

7.1.6 Porovnávání rychlostí algoritmů APRIORI a APRIORI TID

Algoritmy APRIORI a APRIORI TID slouží pro zefektivnění a urychlení procesu získávání frekventovaných množin a tím i asociačních pravidel z dat. Rychlosti, jakých dosahují, závisí na velikosti a struktuře analyzovaných datových souborů. Jejich hodnoty lze přesně měřit a porovnávat právě v této části aplikace.

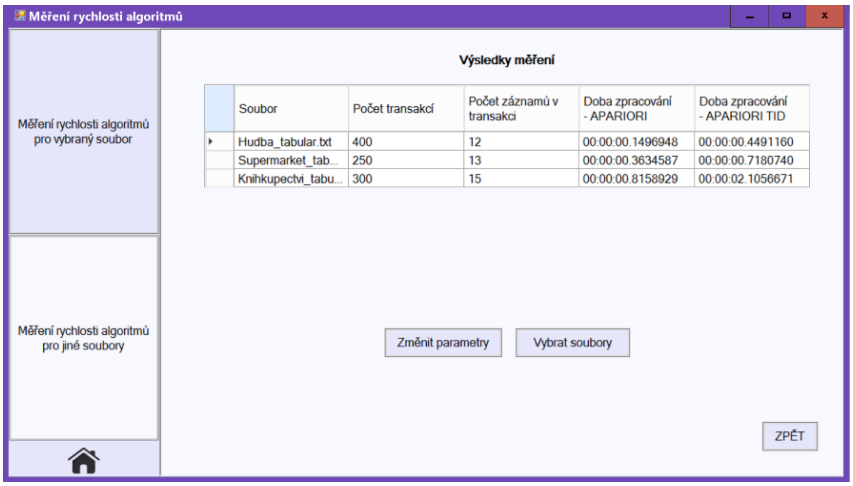
První funkcí je měření rychlosti vyhledávání asociačních pravidel v datovém souboru vybraném globálně pro celý program v jeho úvodním okně. Uživatel stanoví kritéria, jimž musí vyhovovat hledané implikace (požadovanou minimální podporu a spolehlivost) a klikne na tlačítko *Měřit rychlost algoritmů.* Asociační pravidla jsou pak vyhledávána na pozadí v odděleném vlákně a nikde se nezobrazují, což by celý proces zpomalilo a zkreslilo skutečné výsledky, ukáží se pouze finální hodnoty naměřených rychlostí. Výstup z této části aplikace vypadá následovně:



Obrázek 12: Měření rychlostí algoritmů pro vybraná transakční data

Druhou funkcí je měření rychlosti vyhledávání asociačních pravidel v jednom či více datových souborech vybraných zvlášť přímo pro tuto část programu umožňující porovnání dosažených výsledků pro různá data při stejných počátečních parametrech. I zde uživatel nejprve stanoví kritéria, jimž musí vyhovovat hledané implikace – požadovanou minimální podporu a spolehlivost. Poté vybere datové soubory, které chce analyzovat a klikne na tlačítko *Měřit rychlost algoritmů.*

Vyhledávání asociačních pravidel probíhá opět na pozadí v odděleném vlákně, uživateli se zobrazuje pouze průběžný stav procesu, nikoli nalezené implikace, ve finále jsou pak hodnoty naměřených rychlostí pro všechny vybrané soubory zapsány do tabulky, jak je vidět na následujícím obrázku.



Obrázek 13: Měření rychlostí algoritmů pro více datových souborů

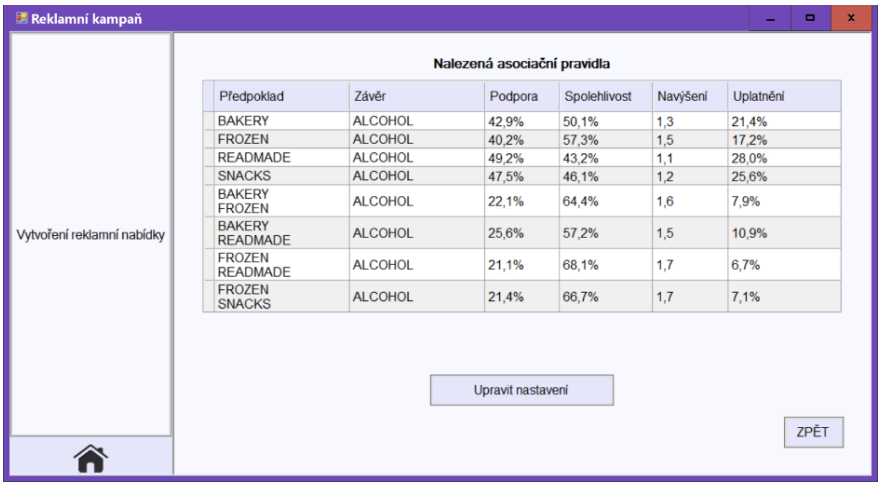
Jak je uvedeno v kapitole 5 algoritmus APRIORI TID vznikl jako optimalizace původního APRIORI, tedy lze předpokládat, že by měl dosahovat lepších výsledků a vyšších rychlostí nežli jeho předchůdce. Experimentálním testováním v této části aplikace se ovšem ukázalo, že tomu tak zdaleka není vždy. V případě malých datových souborů, souborů neobsahujících mnoho atributů anebo naopak takových, kdy jsou kombinace tvořeny velkým množstvím položek, je doba potřebná k vytvoření či modifikaci pomocné struktury C_k, s níž pracuje APRIORI TID, vyšší než doba potřebná k projití původních dat, a proto zde tento algoritmus dosahuje horších výsledků.

7.1.7 Vytváření reklamních nabídek

Analýza nákupního košíku, jíž se ve své práci mimo jiné zabývám, je prostředek, který umožňuje majitelům obchodních společností a prodejcům odhalovat skryté vzory chování svých zákazníků a využívat tyto informace ke zlepšení obchodní strategie a zvýšení finančního zisku. Díky objeveným asociacím je možné mnohem sofistikovanějším způsobem sestavovat reklamní nabídky, oslovovat primárně ty zákazníky, u nichž lze předpokládat zájem o nabízené zboží anebo určovat, které produkty je vhodné nabízet například za zvýhodněnou cenu či v kombinaci s jinými druhy. Právě takové fiktivní reklamní nabídky lze vytvářet i v této aplikaci.

Program nejprve z hlavičky vybraného souboru transakčních dat načte všechny kategorie zboží, které nabízel daný obchod. Následně dokáže sestavit reklamní nabídku, respektive doporučit vhodné položky, které by se v ní měly objevit, na základě uživatelem zvolených požadavků následovně:

- uživatel vybere produkty, jejichž prodej by chtěl coby obchodník podpořit
- jsou určeny minimální hodnoty podpory a spolehlivosti asociačních pravidel, která budou podkladem pro sestavení reklamy
- aplikace nalezne s využitím algoritmu APRIORI všechna asociační pravidla vyhovující stanoveným kritériím a zobrazí následující tabulku



Předpoklad	Závěr	Podpora	Spolehlivost	Navýšení	Uplatnění
BAKERY	ALCOHOL	42,9%	50,1%	1,3	21,4%
FROZEN	ALCOHOL	40,2%	57,3%	1,5	17,2%
READMADE	ALCOHOL	49,2%	43,2%	1,1	28,0%
SNACKS	ALCOHOL	47,5%	46,1%	1,2	25,6%
BAKERY FROZEN	ALCOHOL	22,1%	64,4%	1,6	7,9%
BAKERY READMADE	ALCOHOL	25,6%	57,2%	1,5	10,9%
FROZEN READMADE	ALCOHOL	21,1%	68,1%	1,7	6,7%
FROZEN SNACKS	ALCOHOL	21,4%	66,7%	1,7	7,1%

Obrázek 14: Vytváření reklamních nabídek

Cílové produkty, tedy ty, na které byla kampaň zaměřena, jsou zobrazeny ve druhém sloupci tabulky a jedná se o závěry vygenerovaných implikací. V prvním sloupci jsou pak vypsány druhy zboží případně jejich kombinace, s nimiž je vhodné zvolené produkty nabízet (předpoklady vygenerovaných implikací), a ze zbylých sloupců lze zjistit, jak spolehlivá nalezená pravidla jsou nebo do jaké míry se díky nim zvýší prodej.

7.1.8 Vizualizační funkce

Aplikace vytvořená v rámci této diplomové práce by měla v budoucnu sloužit jako podpůrný prostředek pro výuku asociačních metod v rámci předmětu *Data mining* a umožňovat studentům mimo jiné samostudium této problematiky. Proto jsem se rozhodla implementovat do programu také několik grafických vizualizací vybraných pojmů souvisejících s asociačními metodami a pravidly usnadňujících jejich pochopení. Jedná se o následující problémy:

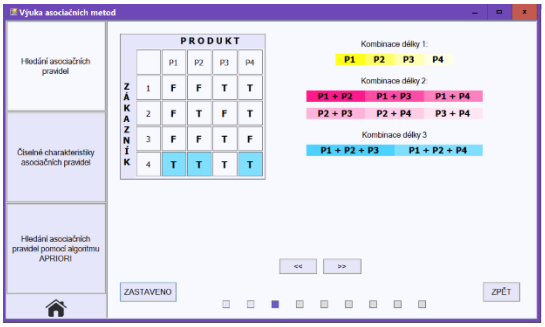
- generování frekventovaných množin a získávání asociačních pravidel z dat
- základní numerické charakteristiky asociačních pravidel a jejich výpočet
- princip algoritmu APRIORI a jeho postup při hledání asociačních pravidel

Všechny výše zmíněné body jsou demonstrovány na ukázkových maticích malých velikostí představujících fiktivní transakční data a jsou alespoň dle mého názoru dostatečně slovně popsány a vysvětleny.

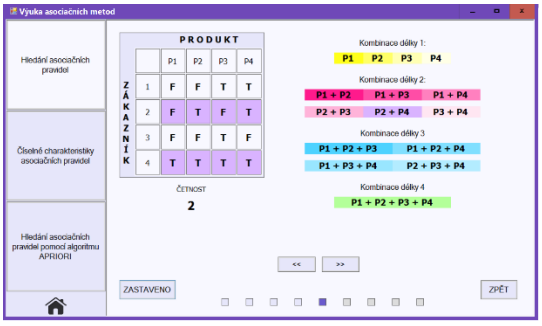
Generování frekventovaných množin a získávání asociačních pravidel z dat

Jedná se o grafickou vizualizaci postupu při generování frekventovaných množin, jejich následném rozkladu na podmnožiny a vytváření asociačních pravidel klasickým způsobem bez použití algoritmu, který by tento proces urychlil.

V datové matici, která je v tomto případě tvořena čtyřmi druhy zboží obsaženými v transakcích čtyř zákazníků obchodu, jsou nejprve identifikovány kategorie nabízených produktů, z nichž jsou vytvořeny jednoprvkové množiny. Jejich sjednocením poté vznikají kombinace délky 2, tedy dvouprvkové množiny, z těchto opět jejich sjednocováním vznikají množiny tříprvkové a konečně sjednocením vhodných dvou tříprvkových množin (shodují se právě ve dvou prvcích) vznikne poslední kandidát – množina obsahující všechny čtyři kategorie produktů. Tímto způsobem jsou vygenerovány veškeré existující kandidátní množiny, v případě zvolených ukázkových dat je jich dvanáct.



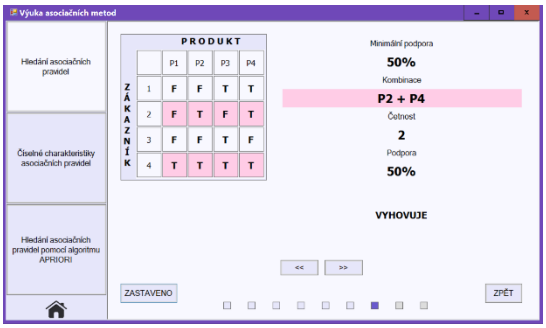
Obrázek 15: Grafická vizualizace – generování kandidátních množin



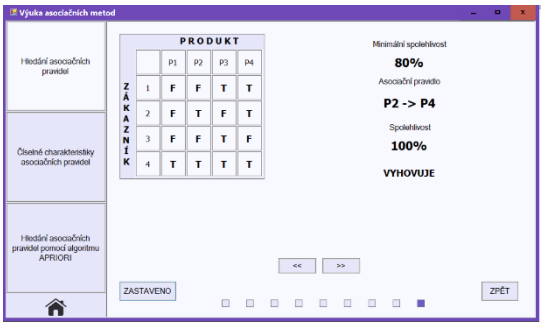
Obrázek 16: Grafická vizualizace – četnost výskytu kandidátů

Pro nalezené kandidátní kombinace zboží tvořené alespoň dvěma prvky je nutné v dalším kroku vypočítat četnost jejich výskytu v datech. Zjištěné hodnoty jsou posléze vyjádřeny procentuálně a porovnány s požadovanou minimální podporou (termín *podpora* je uživateli samozřejmě vysvětlen), jež je pro tento příklad pevně stanovená na 50%. Kandidáti vyhovující zadanému kritériu jsou označeni jako frekventované množiny.

Následuje vysvětlení principu, jakým jsou získávána asociační pravidla. Aplikace pracuje postupně s každou nalezenou frekventovanou množinou, rozloží ji na podmnožiny a sestaví kandidátní implikace, u kterých musí vypočítat jejich spolehlivost (termín *spolehlivost* je uživateli opět vysvětlen). Zjištěná hodnota je porovnána s požadovanou minimální spolehlivostí, pro tento příklad zvolenou 80%, a kandidáti, kteří jí dosahují, jsou označeni jako tzv. silná asociační pravidla.



Obrázek 17: Grafická vizualizace – frekventované množiny



Obrázek 18: Grafická vizualizace – asociační pravidla

Tímto je vizualizace ukončena. Každý její krok je podrobně slovně popsán a lze jej dle potřeby pozastavit či zopakovat. Mezi jednotlivými kroky je rovněž možné libovolně přepínat – přeskakovat je anebo se k nim vracet.

Základní numerické charakteristiky asociačních pravidel a jejich výpočet

Zde je popsáno a graficky vizualizováno, jak lze kvantitativně hodnotit asociační pravidla a posuzovat tak jejich kvalitu a informační hodnotu na základě některých číselných charakteristik, konkrétně následujících čtyř:

- podpora (support)
- spolehlivost (confidence)
- navýšení (lift)
- uplatnění (deployability)

K demonstraci postupů při zjišťování jejich hodnot je využito datové matice obsahující fiktivní transakční data, tentokrát se jedná o sedm druhů zboží obsažených v nákupních košících sedmi zákazníků, větší počet záznamů jsem zvolila pro lepší názornost výpočtů.

Grafické vizualizace číselného hodnocení vybraného asociačního pravidla vypadají následovně:

- **Grafická vizualizace podpory**

  V programu je znázorněn výpočet podpory pro celé asociační pravidlo (tedy kombinaci všech kategorií atributů, jimiž je tvořeno) a dále také podpory jeho předpokladu a závěru.

  Matice dat je řádek po řádku postupně procházena a v případě, že obsahuje kombinaci položek, jejíž podpora je zjišťována, je příslušná transakce zvýrazněna a inkrementována hodnota četnosti výskytu dané kombinace. Výsledek je následně vyjádřen procentuálně.

- **Grafická vizualizace spolehlivosti**

  Spolehlivost asociačního pravidla je vyjádřena jako podíl podpory celé implikace a podpory jejího předpokladu. Tyto hodnoty jsou tedy nejprve zjištěny, přičemž jsou podobně jako v předchozím případě vždy zvýrazněny jim odpovídající transakce v matici dat.

  Následně je graficky vizualizován význam spolehlivosti: jedná se procentuální zastoupení implikací, u nichž platí závěr, mezi všemi takovými, u nichž platí předpoklad.

- **Grafická vizualizace navýšení**

  Navýšení asociačního pravidla je vyjádřeno jako podíl jeho spolehlivosti a podpory závěru. Obě hodnoty jsou tedy zjištěny, tento proces probíhá stejně, jak je popsáno výše, poté je vypočten jejich podíl.

  Z praktického hlediska v případě analýzy nákupního košíku znamená navýšení, kolikrát vzroste pravděpodobnost prodeje určitého druhu zboží v případě, že jej obchodník bude nabízet s položkami obsaženými v předpokladu asociačního pravidla, nikoli vybranými zcela náhodně. Grafická vizualizace tohoto významu tedy probíhá tak, že je množství reprezentující podporu závěru implikace, které je znázorněno jako obdélník určité velikosti, vynásobeno navýšením (opět znázorněno jako obdélník). Velikost nově vzniklého obrazce se pak rovná velikosti obdélníku znázorňujícího spolehlivost implikace.

- **Grafická vizualizace uplatnění**

  Uplatnění asociačního pravidla je vyjádřeno jako rozdíl podpory předpokladu a podpory celé implikace. I zde jsou nejprve obě hodnoty zjištěny a posléze je vypočten jejich rozdíl.

  Grafická vizualizace vypadá tak, že v tabulce transakcí jsou zvýrazněny všechny takové, ve kterých je zastoupen předpoklad implikace, a z nich jsou následně vybrány pouze ty, kde není splněn závěr implikace.

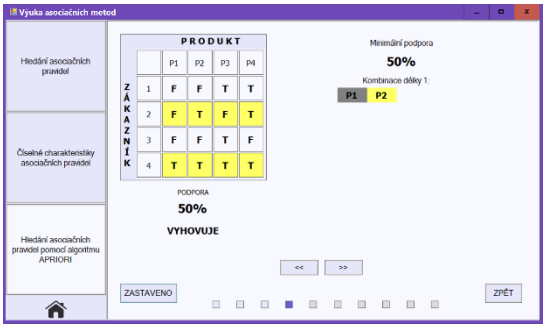
Všechny výše zmíněné grafické vizualizace jsou opět doplněny podrobným slovním popisem, názvy jednotlivých pojmů jsou v popisech uvedeny česky a rovněž anglicky, neboť v data miningu jsou prakticky výhradně užívány právě tyto termíny, nikoli jejich česká synonyma.

Princip algoritmu APRIORI a jeho postup při hledání asociačních pravidel

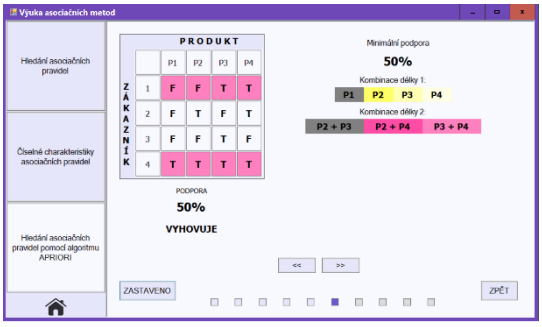
Poslední grafická vizualizace popisuje způsob, jakým jsou generovány frekventované množiny a z nich následně získávána asociační pravidla s použitím algoritmu APRIORI. Datová matice představující fiktivní transakční data, na kterých je celý proces demonstrován, je totožná jako v případě první vizualizace a tvoří ji čtyři druhy zboží obsažené v nákupních košících čtyř zákazníků.

V transakcích jsou i zde nejprve identifikovány kategorie nabízených produktů, z nichž jsou vytvořeny jednoprvkové kandidátní množiny. U nalezených kandidátů je zjištěna četnost jejich výskytu v datech, tato hodnota je vyjádřena procentuálně a porovnána s požadovanou minimální podporou, pro tento případ pevně stanovenou na 50%.

Vyřazením kandidátů s nedostatečnou podporou vzniká množina frekventovaných položek – kombinací produktů délky 1, na základě které generuje algoritmus APRIORI v dalším kroku kandidátní množiny délky 2. Ty vznikají jako sjednocení dvojic frekventovaných množin délky 1 a zároveň musí platit, že všechny jejich podmnožiny jsou rovněž frekventované. Tato vlastnost je v případě dvouprvkových kombinací již splněna, neboť jediné dvě podmnožiny, které obsahuje (neuvažujeme-li prázdnou množinu) jsou právě ty, jejichž sjednocením vznikla, a proto ji není nutné dále ověřovat. U nalezených kandidátů je posléze opět zjištěna četnost jejich výskytu v datech, její hodnota je vyjádřena procentuálně a porovnána s požadovanou minimální podporou. Algoritmus takto vygeneruje dvouprvkové frekventované množiny.



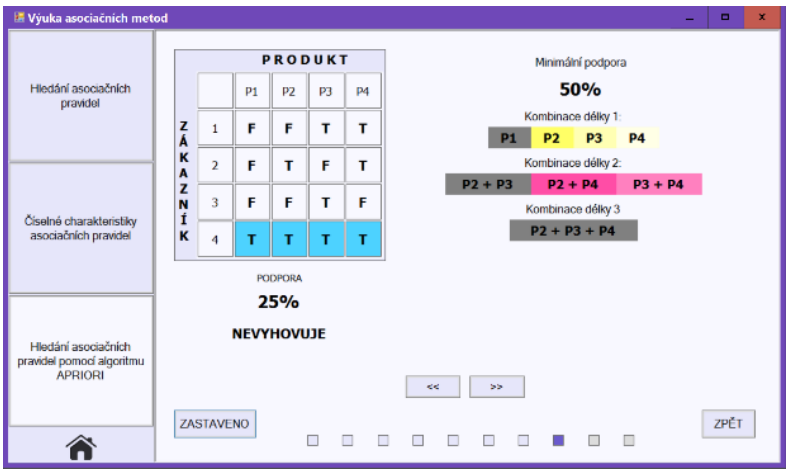
Obrázek 19: Grafická vizualizace APRIORI – frekventované položky



Obrázek 20: Grafická vizualizace APRIORI – frekventované množiny délky 2

Následuje vysvětlení principu generování frekventovaných množin délky k pro $k = 3$. Každá kandidátní kombinace délky k vzniká sjednocením vhodných dvou frekventovaných množin délky $k\text{-}1$ shodujících se právě v $k\text{-}2$ prvcích a musí splňovat apriori vlastnost – všechny její podmnožiny jsou frekventované. U nalezených kandidátů v tomto jednoduchém příkladu se jedná pouze o jednu tříprvkovou množinu, jejíž četnost výskytu je následně nutné zjistit z původních dat, vyjádřit procentuálně a vypočtenou hodnotu porovnat s požadovanou

minimální podporou. V tomto bodě je generování frekventovaných množin u konce, neboť již není možné vytvořit žádné nové kandidáty délky vyšší než 3.



Obrázek 21: Grafická vizualizace APRIORI – frekventované množiny délky 3

Dále je vysvětlen postup při získávání asociačních pravidel. Každá frekventovaná množina je rozložena na podmnožiny, z nichž jsou sestaveny kandidátní implikace. U těchto kandidátu je nutné vypočítat jejich spolehlivost a tu porovnat s požadovanou minimální spolehlivostí, pro tento případ zvolenou 80%. Jako silná asociační pravidla jsou označena všechna taková pravidla, která této hodnoty dosahují. Ukázky zde již neuvádím, neboť se jejich podoba nijak neliší od první vizualizace.

Vizualizace principu algoritmu APRIORI probíhá nad stejnými daty a hodnotami požadované minimální podpory a spolehlivosti, které byly použity i v případě vizualizace demonstrující prosté generování asociačních pravidel bez použití algoritmu, a vede tedy ke stejným výsledkům. Uživatel má tak možnost posoudit míru, do jaké algoritmus APRIORI získávání implikací zefektivní.

Stejně jako tomu bylo v předchozích dvou příkladech i v případě poslední grafické vizualizace je každý její krok doprovázen podrobným slovním popisem a lze jej dle potřeby pozastavit či zopakovat. Mezi jednotlivými kroky je rovněž možné libovolně přepínat – přeskakovat je anebo se k nim vracet.

7.2 Zpracování reálných dat

Mým původním záměrem bylo, aby program dokázal kromě uměle vytvořených datových souborů zpracovat a analyzovat také libovolný soubor obsahující reálná transakční data získaná od skutečného obchodního řetězce, nicméně taková data se mi nakonec získat nepodařilo. Jako alternativu jsem se tedy rozhodla využít informace o zápisu studentů na volitelné a povinně volitelné předměty zajišťované fakultou mechatroniky Technické univerzity v Liberci získané ze STAGU.

Měla jsem k dispozici anonymizovaná data z let 2011 až 2015 obsahující údaje o studentech a všech předmětech, na které byli zapsaní v příslušném akademickém roce zajišťovaných libovolnou fakultou Technické univerzity. Dále jsem měla k dispozici data obsahující výčet všech povinných předmětů zajišťovaných fakultou mechatroniky v příslušných letech. Na základě těchto informací jsem získala nové datové soubory obsahující záznamy o studentech a jimi vybraných volitelných předmětech zajišťovaných fakultou mechatroniky (včetně projektů, bakalářských a diplomových prací) v letech 2011 až 2015. Tato data jsem následně zpracovala v aplikaci a nalezla skryté vztahy v podobě asociačních pravidel, jejichž číselné charakteristiky vyhovují požadovaným kritériím.

Podpora nalezených implikací byla vždy velmi nízká vzhledem k velkému množství nabízených volitelných předmětů (v průměru 110), kdy nebylo rozlišováno, pro jaký obor je předmět vypisován či v jakém ročníku jej student absolvoval. Naopak spolehlivost těchto implikací se v průměru pohybovala mezi 80% a 90%.

Výsledky analýzy pro zvolené hodnoty minimální podpory 7% a minimální spolehlivosti 60% uvádím v příloze 1.

Pro dosažení vyšší podpory asociačních pravidel jsem zpracovala obdobná data pouze pro studenty 3. ročníků čtyř bakalářských oborů Technické univerzity, a to Informační technologie, Informatika a logistika, Elektronické informační a řídicí systémy a Nanomateriály. Redukcí počtu studijních oborů, a tedy i nabízených volitelných předmětů (v průměru 20) jsem docílila toho, že výsledná datová matice, v níž byly vyhledávány implikace, nebyla tak řídká a možná podpora pro generovaná pravidla znatelně vzrostla.

I zde bylo nejprve nutné upravit datové soubory získané ze STAGU do podoby, aby bylo možné zpracovat je v aplikaci. Odstranila jsem nepotřebné či neúplné informace a ponechala pouze záznamy obsahující identifikační číslo studenta a jím vybrané volitelné předměty. V takto připravených datech jsem následně hledala skryté vztahy v podobě asociačních pravidel, jejichž číselné charakteristiky dosahují požadovaných hodnot podpory a spolehlivosti.

V tomto případě se podpora nalezených implikací pohybovala mezi 40% až 50% a jejich spolehlivost činila v průměru 80%.

Výsledky analýzy pro zvolené hodnoty minimální podpory 40% a minimální spolehlivosti 60% uvádím v příloze 2.

Pro stejné hodnoty minimální podpory a spolehlivosti (tedy 40% a 60%) jsem dále provedla analýzu dat obsahujících záznamy o studentech 3. ročníků oboru Informační technologie a jimi vybraných volitelných předmětech zajišťovaných fakultou mechatroniky. Počet nabízených volitelných předmětů byl v tomto případě v průměru 17 a získané výsledky uvádím v příloze 3.

Z výsledků analýz vyplývá, že mezi studenty převládá zájem o předměty, které se věnují moderním trendům nebo pokročilým metodám v programování, operačnímu systému Unix a bezpečnosti. Naopak zájem o předměty zabývajícími se zpracováním signálů, zpracováním obrazu nebo elektrotechnikou je minimální.

Ze zjištěných informací by bylo možné vycházet při sestavování nabídek volitelných a povinně volitelných předmětů v budoucnu, umožnit studentům hlouběji se zabývat problematikou, která je skutečně zajímá.

Závěr

Ve své diplomové práci jsem se zabývala rozhodovacími a asociačními pravidly a jejich možným praktickým významem a využitím v oblastech marketingu, bankovnictví, pojišťovnictví, vědy a mnoha dalších. Dále jsem se zabývala způsoby a prostředky, kterými lze v datech odhalovat skryté vztahy a s algoritmy, díky nimž je možné proces jejich hledání realizovat, případně zefektivnit.

Věnovala jsem se dvěma typům úloh: analýze preferencí studentů ve výběru volitelných a povinně volitelných předmětů a analýze nákupních preferencí zákazníků. Při hledání asociací jsem ovšem nevyužívala žádného jiného komerčního data miningového nástroje (IBM SPSS Modeler, Knime, Weka aj.), ale vytvořila jsem vlastní aplikaci, která tento problém dokáže řešit, sloužící nejen jako nástroj pro posouzení dat, vytěžování skrytých informací a vyhledávání vztahů, ale zároveň i jako prostředek, který bude moci být využíván ve výuce data miningu, usnadní studentům pochopení asociačních metod a umožní jim jejich samostudium skrze grafické vizualizace vybraných problémů a možnost experimentování s daty. V tomto vidím hlavní přínos celé práce.

V současnosti jsou rámci interaktivní výukové aplikace implementovány dva algoritmy pro vyhledávání asociačních pravidel, algoritmus APRIORI a algoritmus APRIORI TID, v budoucnu by bylo možné zařadit další algoritmy, například algoritmus CARMA, případně jiné optimalizace původního algoritmu APRIORI. Další možnost rozšíření vidím v přidání nových vizualizačních funkcí, například grafické vizualizace algoritmu APRIORI TID.

V diplomové práci jsem se kromě algoritmů zmiňovaných v souvislosti s asociačními metodami ve světě zabývala také metodou GUHA sloužící k systematickému vytváření hypotéz na základě dat, která je původní českou metodou, je možné ji považovat za předchůdce asociačních metod a byla jednou z prvních metod těžení dat vůbec. Celé téma jsem zpracovala jako výkladový text pro e-learningový kurz *Data miningu.* Jeho zařazení do předmětu by mělo přispět k posílení vědomí studentů o schopnostech českých vědců. Jak jsem studiem metody zjistila, týmu profesora Hájka se podařilo přispět originálními postupy k zpracování "velkých" dat a být prvními, kdo přišel s myšlenkou asociací. Práce na tématu by mohla dále pokračovat například studentským projektem, který by doplnil výkladový text ukázkovými příklady a vizualizací důležitých problémů pro lepší porozumění základním principům metody GUHA.

Díky své diplomové práci jsem si rozšířila znalosti z oblasti data miningu a behaviorální analýzy a rovněž své dovednosti v programování v jazyce C#, který se v současnosti těší mezi programátory velké oblibě a je velmi rozšířený. Práce pro mne byla v mnoha ohledech velkým přínosem.

Použité zdroje

[1] „Statistické metody v genetické epidemiologii,“ [Online]. Available: https://is.muni.cz/th/184569/prif_b/PongracovaM.pdf. [Přístup získán Prosinec 2016].

[2] „Data miningové modely, asociační pravidla a analýza sekvencí,“ [Online]. Available: http://acrea.cz/centrum-vyuky-acrea/prehled-kurzu/dataminingove-modely-asociacni-pravidla-a-analyza-sekvenci.html. [Přístup získán Leden 2017].

[3] P. BERKA, Dobývání znalostí z databází, Praha: Academia, 2003.

[4] „Dobývání znalostí,“ [Online]. Available: http://ksvi.mff.cuni.cz/~mraz/datamining/lecture/Dobyvani_Znalosti_Prednaska_Asociacni_pravidla.pdf. [Přístup získán Leden 2017].

[5] „Fast Algorithms for Mining Association Rules,“ [Online]. Available: http://rakesh.agrawal-family.com/papers/vldb94apriori.pdf. [Přístup získán Únor 2017].

[6] P. HÁJEK, T. HAVRÁNEK a M. CHYTIL, Metoda GHUA - Automatická tvorba hypotéz, Praha: Academia, 1983.

[7] „Užití asociačních pravidel při analýze dat získaných z průzkumu v oblasti podnikové informatiky,“ [Online]. Available: http://www.cssi.cz/cssi/system/files/all/SI_2012_01_05_Chudan.pdf. [Přístup získán Květen 2017].

[8] H. SKALSKÁ, Datamining a klasifikační modely, Praha: GAUDEAMUS, 2010.

[9] V. MAYER-SCHONBERGER a K. CUKIER, Big Data, Praha: Computer Press, 2014.

[10] „Association Analysis: Basic concepts and algorithms,“ [Online]. Available: https://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf. [Přístup získán Leden 2017].

[11] „Practical issues of mining association rules,“ [Online]. Available: http://www-users.cs.umn.edu/~ptan/dmclass/lecture/assoc3. [Přístup získán Únor 2017].

[12] „Získávání znalostí z databází - asociační pravidla,“ [Online]. Available: http://www.fit.vutbr.cz/study/courses/ZZD/public/seminar0304/GUHA-text.pdf. [Přístup získán Prosinec 2016].

[13] „Data mining,“ [Online]. Available: http://www.msps.cz/data-mining. [Přístup získán Prosinec 2016].

Přílohy

- Příloha 1: Výsledky analýzy závislostí mezi volitelnými předměty u studentů fakulty mechatroniky Technické univerzity v Liberci
- Příloha 2: Výsledky analýzy závislostí mezi volitelnými předměty u studentů 3. ročníků fakulty mechatroniky Technické univerzity v Liberci
- Příloha 3: Výsledky analýzy závislostí mezi volitelnými předměty u studentů 3. ročníků oboru Informační technologie Technické univerzity v Liberci
- Příloha 4: Obsah přiloženého CD

PŘÍLOHA 1

Akademický rok 2011/2012					
PŘEDPOKLAD	**ZÁVĚR**	**PODPORA**	**SPOLEHLIVOST**	**NAVÝŠENÍ**	**UPLATNĚNÍ**
MTI/VES - Vestavné systémy	NTI/PBE - Počítačová bezpečnost	7,80%	97,20%	7,81	0,20%
NTI/ADA - Algoritmy a datové struktury + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	7,60%	91,90%	7,38	0,70%
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	8,70%	90,70%	7,29	0,90%
MTI/PMZ - Programování mobilních zařízení + NTI/ADA - Algoritmy a datové struktury	NTI/PBE - Počítačová bezpečnost	8,00%	90,00%	7,23	0,90%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/ADA - Algoritmy a datové struktury	8,00%	90,00%	7,79	0,90%
NTI/ADA - Algoritmy a datové struktury	NTI/PBE - Počítačová bezpečnost	10,20%	88,50%	7,11	1,30%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	NTI/ADA - Algoritmy a datové struktury	7,60%	87,20%	7,54	1,10%
MTI/PMZ - Programování mobilních zařízení	NTI/ADA - Algoritmy a datové struktury	8,90%	87,00%	7,53	1,30%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	8,90%	87,00%	6,99	1,30%
NTI/UI - Unix a Internet	NTI/ADA - Algoritmy a datové struktury	8,20%	86,00%	7,45	1,30%
NTI/PBE - Počítačová bezpečnost	NTI/ADA - Algoritmy a datové struktury	10,20%	82,10%	7,11	2,20%
NTI/UI - Unix a Internet	NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	7,60%	79,10%	7,74	2,00%
MTI/PMZ - Programování mobilních zařízení	NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	8,00%	78,30%	7,66	2,20%
NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	8,00%	78,30%	7,66	2,20%
NTI/ADA - Algoritmy a datové struktury	MTI/PMZ - Programování mobilních zařízení	8,90%	76,90%	7,53	2,70%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	7,10%	74,40%	7,28	2,40%
NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	7,60%	73,90%	7,74	2,70%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	8,90%	71,40%	6,99	3,60%
NTI/ADA - Algoritmy a datové struktury	NTI/UI - Unix a Internet	8,20%	71,20%	7,45	3,30%
MTI/PMZ - Programování mobilních zařízení	NTI/UI - Unix a Internet	7,10%	69,60%	7,28	3,10%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	8,70%	69,60%	7,29	3,80%
NTI/ADA - Algoritmy a datové struktury	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	8,00%	69,20%	7,79	3,60%
NTI/ADA - Algoritmy a datové struktury	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	7,60%	65,40%	7,54	4,00%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení + NTI/ADA - Algoritmy a datové struktury	8,00%	64,30%	7,23	4,40%
NTI/PBE - Počítačová bezpečnost	MTI/VES - Vestavné systémy	7,80%	62,50%	7,81	4,70%
NTI/PBE - Počítačová bezpečnost	NTI/ADA - Algoritmy a datové struktury + NTI/UI - Unix a Internet	7,60%	60,70%	7,38	4,90%

Akademický rok 2012/2013					
PŘEDPOKLAD	ZÁVĚR	PODPORA	SPOLEHLIVOST	NAVÝŠENÍ	UPLATNĚNÍ
NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	7,20%	92,90%	7,78	0,60%
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	8,80%	92,90%	7,79	0,70%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	8,70%	87,60%	7,35	1,20%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	7,40%	77,60%	7,83	2,10%
MTI/PMZ - Programování mobilních zařízení	NTI/UI - Unix a Internet	7,40%	74,20%	7,83	2,60%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	8,80%	73,80%	7,79	3,10%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	8,70%	72,90%	7,35	3,20%
NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python	7,20%	60,70%	7,78	4,70%

Akademický rok 2013/2014					
PŘEDPOKLAD	ZÁVĚR	PODPORA	SPOLEHLIVOST	NAVÝŠENÍ	UPLATNĚNÍ
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	8,70%	93,50%	7,95	0,60%
NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	7,50%	90,80%	7,73	0,80%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	8,60%	87,70%	7,46	1,20%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	7,00%	75,60%	7,72	2,30%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	8,70%	73,70%	7,95	3,10%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	8,60%	73,10%	7,46	3,20%
MTI/PMZ - Programování mobilních zařízení	NTI/UI - Unix a Internet	7,00%	71,50%	7,72	2,80%
NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python	7,50%	63,50%	7,73	4,30%

Akademický rok 2014/2015					
PŘEDPOKLAD	**ZÁVĚR**	**PODPORA**	**SPOLEHLIVOST**	**NAVÝŠENÍ**	**UPLATNĚNÍ**
MTI/PMZ - Programování mobilních zařízení + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	7,10%	96,80%	8,12	0,20%
MTI/PMZ - Programování mobilních zařízení + NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	7,00%	94,40%	7,92	0,40%
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	9,00%	93,30%	7,83	0,60%
NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	7,90%	90,50%	7,6	0,80%
NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení	7,00%	88,80%	8,75	0,90%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	8,80%	87,20%	7,32	1,30%
NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení	7,40%	85,10%	8,39	1,30%
NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	7,00%	80,40%	9,09	1,70%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	7,10%	80,00%	8,27	1,80%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python	7,00%	79,30%	9,09	1,80%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	7,10%	78,40%	7,73	1,90%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	9,00%	75,70%	7,83	2,90%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	7,30%	75,60%	7,45	2,40%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	8,80%	74,30%	7,32	3,10%
MTI/PMZ - Programování mobilních zařízení	NTI/PJP - Programovací jazyk Python	7,40%	73,30%	8,39	2,70%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	7,10%	73,20%	8,27	2,60%
MTI/PMZ - Programování mobilních zařízení	NTI/UI - Unix a Internet	7,30%	72,10%	7,45	2,80%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	7,10%	69,80%	7,73	3,10%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	7,00%	69,20%	8,75	3,10%
NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python	7,90%	66,30%	7,6	4,00%

Akademický rok 2015/2016					
PŘEDPOKLAD	**ZÁVĚR**	**PODPORA**	**SPOLEHLIVOST**	**NAVÝŠENÍ**	**UPLATNĚNÍ**
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	8,60%	93,60%	8,39	0,60%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	8,40%	84,60%	7,59	1,50%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	7,20%	77,50%	7,82	2,10%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	8,60%	77,40%	8,39	2,50%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	8,40%	75,20%	7,59	2,80%
MTI/PMZ - Programování mobilních zařízení	NTI/UI - Unix a Internet	7,20%	72,10%	7,82	2,80%

PŘÍLOHA 2

Akademický rok 2011/2012					
PŘEDPOKLAD	**ZÁVĚR**	**PODPORA**	**SPOLEHLIVOST**	**NAVÝŠENÍ**	**UPLATNĚNÍ**
MTI/VES - Vestavné systémy	NTI/PBE - Počítačová bezpečnost	40,50%	97,00%	1,5	1,30%
NTI/ADA - Algoritmy a datové struktury + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	40,50%	94,10%	1,46	2,50%
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	45,60%	92,30%	1,43	3,80%
NTI/ADA - Algoritmy a datové struktury	NTI/PBE - Počítačová bezpečnost	55,70%	91,70%	1,42	5,10%
MTI/PMZ - Programování mobilních zařízení + NTI/ADA - Algoritmy a datové struktury	NTI/PBE - Počítačová bezpečnost	41,80%	91,70%	1,42	3,80%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/ADA - Algoritmy a datové struktury	41,80%	89,20%	1,47	5,10%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	NTI/ADA - Algoritmy a datové struktury	40,50%	88,90%	1,46	5,10%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	46,80%	88,10%	1,36	6,30%
NTI/UI - Unix a Internet	NTI/ADA - Algoritmy a datové struktury	43,00%	87,20%	1,43	6,30%
NTI/PBE - Počítačová bezpečnost	NTI/ADA - Algoritmy a datové struktury	55,70%	86,30%	1,42	8,90%
MTI/PMZ - Programování mobilních zařízení	NTI/ADA - Algoritmy a datové struktury	45,60%	85,70%	1,41	7,60%
NTI/UI - Unix a Internet	NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	40,50%	82,10%	1,47	8,90%
MTI/PMZ - Programování mobilních zařízení	NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	41,80%	78,60%	1,41	11,40%
NTI/ADA - Algoritmy a datové struktury	MTI/PMZ - Programování mobilních zařízení	45,60%	75,00%	1,41	15,20%
NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	41,80%	75,00%	1,41	13,90%
NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	40,50%	72,70%	1,47	15,20%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	46,80%	72,50%	1,36	17,70%
NTI/ADA - Algoritmy a datové struktury	NTI/UI - Unix a Internet	43,00%	70,80%	1,43	17,70%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	45,60%	70,60%	1,43	19,00%

NTI/ADA - Algoritmy a datové struktury	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	41,80%	68,80%	1,47	19,00%
NTI/ADA - Algoritmy a datové struktury	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	40,50%	66,70%	1,46	20,30%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení + NTI/ADA - Algoritmy a datové struktury	41,80%	64,70%	1,42	22,80%
NTI/PBE - Počítačová bezpečnost	MTI/VES - Vestavné systémy	40,50%	62,70%	1,5	24,10%
NTI/PBE - Počítačová bezpečnost	NTI/ADA - Algoritmy a datové struktury + NTI/UI - Unix a Internet	40,50%	62,70%	1,46	24,10%

Akademický rok 2012/2013					
PŘEDPOKLAD	ZÁVĚR	PODPORA	SPOLEHLIVOST	NAVÝŠENÍ	UPLATNĚNÍ
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	46,80%	94,80%	1,51	2,60%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	45,50%	89,90%	1,43	5,10%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	46,80%	74,50%	1,51	16,00%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	45,50%	72,40%	1,43	17,30%

Akademický rok 2013/2014					
PŘEDPOKLAD	**ZÁVĚR**	**PODPORA**	**SPOLEHLIVOST**	**NAVÝŠENÍ**	**UPLATNĚNÍ**
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	46,30%	94,60%	1,54	2,60%
NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	41,40%	91,30%	1,49	4,00%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	45,80%	89,70%	1,46	5,30%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	46,30%	75,50%	1,54	15,00%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	45,80%	74,80%	1,46	15,40%
NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python	41,40%	67,60%	1,49	19,80%

Akademický rok 2014/2015					
PŘEDPOKLAD	ZÁVĚR	PODPORA	SPOLEHLIVOST	NAVÝŠENÍ	UPLATNĚNÍ
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	45,60%	93,80%	1,58	3,00%
NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	42,30%	90,60%	1,53	4,40%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	46,00%	88,40%	1,49	6,00%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	46,00%	77,40%	1,49	13,40%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	45,60%	76,80%	1,58	13,80%
NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python	42,30%	71,20%	1,53	17,10%

Akademický rok 2015/2016					
PŘEDPOKLAD	**ZÁVĚR**	**PODPORA**	**SPOLEHLIVOST**	**NAVÝŠENÍ**	**UPLATNĚNÍ**
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	45,90%	93,90%	1,61	3,00%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	45,90%	86,60%	1,49	7,10%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	45,90%	78,70%	1,49	12,40%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	45,90%	78,70%	1,61	12,40%

PŘÍLOHA 3

Akademický rok 2011/2012					
PŘEDPOKLAD	**ZÁVĚR**	**PODPORA**	**SPOLEHLIVOST**	**NAVÝŠENÍ**	**UPLATNĚNÍ**
MTI/VES - Vestavné systémy + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	42,90%	100,00%	1,24	0,00%
MTI/VES - Vestavné systémy	NTI/PBE - Počítačová bezpečnost	50,80%	97,00%	1,2	1,60%
MTI/VES - Vestavné systémy + NTI/ADA - Algoritmy a datové struktury	NTI/PBE - Počítačová bezpečnost	44,40%	96,60%	1,19	1,60%
NTI/ADA - Algoritmy a datové struktury + NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	41,30%	96,30%	1,19	1,60%
NTI/ADA - Algoritmy a datové struktury + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	50,80%	94,10%	1,16	3,20%
NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	47,60%	93,80%	1,16	3,20%
NTI/ADA - Algoritmy a datové struktury	NTI/PBE - Počítačová bezpečnost	69,80%	93,60%	1,16	4,80%
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	57,10%	92,30%	1,14	4,80%
MTI/PMZ - Programování mobilních zařízení + NTI/ADA - Algoritmy a datové struktury	NTI/PBE - Počítačová bezpečnost	52,40%	91,70%	1,13	4,80%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/ADA - Algoritmy a datové struktury	52,40%	89,20%	1,2	6,30%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	NTI/ADA - Algoritmy a datové struktury	50,80%	88,90%	1,19	6,30%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	58,70%	88,10%	1,09	7,90%
MTI/VES - Vestavné systémy	NTI/ADA - Algoritmy a datové struktury	46,00%	87,90%	1,18	6,30%
MTI/VES - Vestavné systémy + NTI/PBE - Počítačová bezpečnost	NTI/ADA - Algoritmy a datové struktury	44,40%	87,50%	1,17	6,30%
NTI/UI - Unix a Internet	NTI/ADA - Algoritmy a datové struktury	54,00%	87,20%	1,17	7,90%
NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	NTI/ADA - Algoritmy a datové struktury	41,30%	86,70%	1,16	6,30%
NTI/PBE - Počítačová bezpečnost	NTI/ADA - Algoritmy a datové struktury	69,80%	86,30%	1,16	11,10%
MTI/PMZ - Programování mobilních zařízení	NTI/ADA - Algoritmy a datové struktury	57,10%	85,70%	1,15	9,50%
MTI/VES - Vestavné systémy	NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	44,40%	84,80%	1,21	7,90%

NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení	42,90%	84,40%	1,27	7,90%
NTI/PJP - Programovací jazyk Python	NTI/ADA - Algoritmy a datové struktury	42,90%	84,40%	1,13	7,90%
MTI/VES - Vestavné systémy + NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	42,90%	84,40%	1,36	7,90%
NTI/UI - Unix a Internet	NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	50,80%	82,10%	1,17	11,10%
MTI/VES - Vestavné systémy	NTI/UI - Unix a Internet	42,90%	81,80%	1,32	9,50%
MTI/VES - Vestavné systémy	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	42,90%	81,80%	1,43	9,50%
NTI/PJP - Programovací jazyk Python	NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	41,30%	81,30%	1,16	9,50%
MTI/PMZ - Programování mobilních zařízení	NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	52,40%	78,60%	1,13	14,30%
NTI/ADA - Algoritmy a datové struktury	MTI/PMZ - Programování mobilních zařízení	57,10%	76,60%	1,15	17,50%
NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	52,40%	75,00%	1,13	17,50%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	MTI/VES - Vestavné systémy	42,90%	75,00%	1,43	14,30%
NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	50,80%	72,70%	1,17	19,00%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	58,70%	72,50%	1,09	22,20%
NTI/ADA - Algoritmy a datové struktury	NTI/UI - Unix a Internet	54,00%	72,30%	1,17	20,60%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	44,40%	71,80%	1,08	17,50%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	57,10%	70,60%	1,14	23,80%
NTI/ADA - Algoritmy a datové struktury	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	52,40%	70,20%	1,2	22,20%
NTI/UI - Unix a Internet	MTI/VES - Vestavné systémy	42,90%	69,20%	1,32	19,00%
NTI/UI - Unix a Internet	MTI/VES - Vestavné systémy + NTI/PBE - Počítačová bezpečnost	42,90%	69,20%	1,36	19,00%
NTI/ADA - Algoritmy a datové struktury	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	50,80%	68,10%	1,19	23,80%

MTI/PMZ - Programování mobilních zařízení	NTI/UI - Unix a Internet	44,40%	66,70%	1,08	22,20%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení + NTI/ADA - Algoritmy a datové struktury	52,40%	64,70%	1,13	28,60%
MTI/PMZ - Programování mobilních zařízení	NTI/PJP - Programovací jazyk Python	42,90%	64,30%	1,27	23,80%
NTI/ADA - Algoritmy a datové struktury + NTI/PBE - Počítačová bezpečnost	MTI/VES - Vestavné systémy	44,40%	63,60%	1,21	25,40%
NTI/PBE - Počítačová bezpečnost	MTI/VES - Vestavné systémy	50,80%	62,70%	1,2	30,20%
NTI/PBE - Počítačová bezpečnost	NTI/ADA - Algoritmy a datové struktury + NTI/UI - Unix a Internet	50,80%	62,70%	1,16	30,20%
NTI/ADA - Algoritmy a datové struktury	MTI/VES - Vestavné systémy	46,00%	61,70%	1,18	28,60%

Akademický rok 2012/2013					
PŘEDPOKLAD	ZÁVĚR	PODPORA	SPOLEHLIVOST	NAVÝŠENÍ	UPLATNĚNÍ
NTI/PJP - Programovací jazyk Python + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	40,30%	96,00%	1,17	1,70%
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	61,30%	94,80%	1,15	3,40%
MTI/PMZ - Programování mobilních zařízení + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	46,20%	94,80%	1,15	2,50%
MTI/PMZ - Programování mobilních zařízení + NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	43,70%	94,50%	1,15	2,50%
NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	51,30%	93,80%	1,14	3,40%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	59,70%	89,90%	1,09	6,70%
NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení	43,70%	85,20%	1,28	7,60%
NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení	46,20%	84,60%	1,27	8,40%
NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	43,70%	80,00%	1,34	10,90%
NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	NTI/UI - Unix a Internet	40,30%	78,70%	1,22	10,90%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	46,20%	77,50%	1,2	13,40%
NTI/PJP - Programovací jazyk Python	NTI/UI - Unix a Internet	42,00%	76,90%	1,19	12,60%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	48,70%	75,30%	1,13	16,00%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	46,20%	75,30%	1,13	15,10%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	61,30%	74,50%	1,15	21,00%
NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	40,30%	73,80%	1,2	14,30%
MTI/PMZ - Programování mobilních zařízení	NTI/UI - Unix a Internet	48,70%	73,40%	1,13	17,60%

MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python	43,70%	73,20%	1,34	16,00%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	59,70%	72,40%	1,09	22,70%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	46,20%	71,40%	1,2	18,50%
MTI/PMZ - Programování mobilních zařízení	NTI/PJP - Programovací jazyk Python	46,20%	69,60%	1,27	20,20%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	46,20%	69,60%	1,13	20,20%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	43,70%	65,80%	1,28	22,70%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	NTI/PJP - Programovací jazyk Python	40,30%	65,80%	1,2	21,00%
NTI/UI - Unix a Internet	NTI/PJP - Programovací jazyk Python	42,00%	64,90%	1,19	22,70%
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	40,30%	62,30%	1,22	24,40%
NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python	51,30%	62,20%	1,14	31,10%

Akademický rok 2013/2014					
PŘEDPOKLAD	ZÁVĚR	PODPORA	SPOLEHLIVOST	NAVÝŠENÍ	UPLATNĚNÍ
MTI/PMZ - Programování mobilních zařízení + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	46,50%	96,40%	1,19	1,70%
NTI/PJP - Programovací jazyk Python + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	42,40%	96,10%	1,19	1,70%
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	61,00%	94,60%	1,17	3,50%
MTI/PMZ - Programování mobilních zařízení + NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	46,50%	94,10%	1,16	2,90%
ITE/MT - Multimediální technologie	NTI/PBE - Počítačová bezpečnost	41,30%	93,40%	1,16	2,90%
NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	54,70%	91,30%	1,13	5,20%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	60,50%	89,70%	1,11	7,00%
NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení	46,50%	85,10%	1,26	8,10%
NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení	49,40%	82,50%	1,22	10,50%
NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	46,50%	77,70%	1,28	13,40%
NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	NTI/UI - Unix a Internet	42,40%	77,70%	1,2	12,20%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python	46,50%	76,90%	1,28	14,00%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	46,50%	76,90%	1,19	14,00%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	46,50%	76,20%	1,13	14,50%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	61,00%	75,50%	1,17	19,80%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	60,50%	74,80%	1,11	20,30%

NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	48,30%	74,80%	1,11	16,30%
NTI/PJP - Programovací jazyk Python	NTI/UI - Unix a Internet	44,20%	73,80%	1,14	15,70%
MTI/PMZ - Programování mobilních zařízení	NTI/PJP - Programovací jazyk Python	49,40%	73,30%	1,22	18,00%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	46,50%	72,10%	1,19	18,00%
MTI/PMZ - Programování mobilních zařízení	NTI/UI - Unix a Internet	48,30%	71,60%	1,11	19,20%
NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	42,40%	70,90%	1,16	17,40%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	NTI/PJP - Programovací jazyk Python	42,40%	69,50%	1,16	18,60%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	46,50%	69,00%	1,26	20,90%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	46,50%	69,00%	1,13	20,90%
NTI/UI - Unix a Internet	NTI/PJP - Programovací jazyk Python	44,20%	68,50%	1,14	20,30%
NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python	54,70%	67,60%	1,13	26,20%
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	42,40%	65,80%	1,2	22,10%

Akademický rok 2014/2015					
PŘEDPOKLAD	ZÁVĚR	PODPORA	SPOLEHLIVOST	NAVÝŠENÍ	UPLATNĚNÍ
MTI/PMZ - Programování mobilních zařízení + NTI/PJP - Programovací jazyk Python + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	40,10%	97,80%	1,23	0,90%
MTI/PMZ - Programování mobilních zařízení + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	48,60%	97,30%	1,23	1,40%
NTI/PJP - Programovací jazyk Python + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	45,50%	96,20%	1,21	1,80%
MTI/PMZ - Programování mobilních zařízení + NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	49,50%	94,00%	1,19	3,20%
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	61,30%	93,80%	1,18	4,10%
ITE/MT - Multimediální technologie	NTI/PBE - Počítačová bezpečnost	43,20%	92,30%	1,16	3,60%
NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost	56,80%	90,60%	1,14	5,90%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	61,70%	88,40%	1,11	8,10%
NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python + NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	40,10%	88,10%	1,26	5,40%
NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení	49,50%	87,30%	1,25	7,20%
NTI/PJP - Programovací jazyk Python + NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	41,00%	86,70%	1,24	6,30%
ITE/MT - Multimediální technologie	NTI/PJP - Programovací jazyk Python	40,10%	85,60%	1,37	6,80%
NTI/PJP - Programovací jazyk Python + NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	40,10%	84,80%	1,37	7,20%
NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení	52,70%	84,20%	1,21	9,90%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	NTI/PJP - Programovací jazyk Python	40,10%	82,40%	1,32	8,60%
MTI/PMZ - Programování mobilních zařízení + NTI/UI - Unix a Internet	NTI/PJP - Programovací jazyk Python	41,00%	82,00%	1,31	9,00%

MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	NTI/UI - Unix a Internet	40,10%	80,90%	1,24	9,50%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python	49,50%	80,30%	1,28	12,20%
NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	NTI/UI - Unix a Internet	45,50%	80,20%	1,23	11,30%
MTI/PMZ - Programování mobilních zařízení + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	40,10%	80,20%	1,41	9,90%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	48,60%	79,40%	1,14	12,60%
NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	49,50%	79,10%	1,28	13,10%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	48,60%	78,80%	1,21	13,10%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	61,70%	77,80%	1,11	17,60%
MTI/PMZ - Programování mobilních zařízení + NTI/PJP - Programovací jazyk Python	NTI/UI - Unix a Internet	41,00%	77,80%	1,19	11,70%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	61,30%	77,30%	1,18	18,00%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	50,00%	76,60%	1,1	15,30%
MTI/PMZ - Programování mobilních zařízení + NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	40,10%	76,10%	1,24	12,60%
MTI/PMZ - Programování mobilních zařízení	NTI/PJP - Programovací jazyk Python	52,70%	75,50%	1,21	17,10%
NTI/PJP - Programovací jazyk Python	NTI/UI - Unix a Internet	47,30%	75,50%	1,16	15,30%

NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	48,60%	74,50%	1,21	16,70%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	NTI/PJP - Programovací jazyk Python	45,50%	74,30%	1,19	15,80%
NTI/PJP - Programovací jazyk Python	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	45,50%	72,70%	1,19	17,10%
NTI/UI - Unix a Internet	NTI/PJP - Programovací jazyk Python	47,30%	72,40%	1,16	18,00%
MTI/PMZ - Programování mobilních zařízení	NTI/UI - Unix a Internet	50,00%	71,60%	1,1	19,80%
NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python	56,80%	71,60%	1,14	22,50%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	49,50%	71,00%	1,25	20,30%
NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení + NTI/UI - Unix a Internet	40,10%	70,60%	1,41	16,70%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	48,60%	69,70%	1,14	21,20%
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	45,50%	69,70%	1,23	19,80%
NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení + NTI/UI - Unix a Internet	41,00%	65,50%	1,31	21,60%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení + NTI/PJP - Programovací jazyk Python	40,10%	65,40%	1,24	21,20%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/PJP - Programovací jazyk Python + NTI/UI - Unix a Internet	40,10%	65,00%	1,37	21,60%
NTI/PJP - Programovací jazyk Python	ITE/MT - Multimediální technologie	40,10%	64,00%	1,37	22,50%
NTI/PJP - Programovací jazyk Python	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	40,10%	64,00%	1,32	22,50%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení + NTI/PJP - Programovací jazyk Python	41,00%	62,80%	1,19	24,30%

NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení + NTI/PJP - Programovací jazyk Python	49,50%	62,50%	1,19	29,70%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení + NTI/UI - Unix a Internet	48,60%	61,40%	1,23	30,60%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost + NTI/PJP - Programovací jazyk Python	40,10%	61,40%	1,24	25,20%

Akademický rok 2015/2016					
PŘEDPOKLAD	**ZÁVĚR**	**PODPORA**	**SPOLEHLIVOST**	**NAVÝŠENÍ**	**UPLATNĚNÍ**
MTI/PMZ - Programování mobilních zařízení + NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	50,40%	96,90%	1,24	1,60%
NTI/UI - Unix a Internet	NTI/PBE - Počítačová bezpečnost	62,00%	93,90%	1,2	4,00%
ITE/MT - Multimediální technologie	NTI/PBE - Počítačová bezpečnost	44,80%	89,60%	1,14	5,20%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost	62,00%	86,60%	1,1	9,60%
ITE/MT - Multimediální technologie	MTI/PMZ - Programování mobilních zařízení	42,80%	85,60%	1,2	7,20%
MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	50,40%	81,30%	1,23	11,60%
NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	50,40%	81,30%	1,14	11,60%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení	62,00%	79,10%	1,1	16,40%
NTI/PBE - Počítačová bezpečnost	NTI/UI - Unix a Internet	62,00%	79,10%	1,2	16,40%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení	52,00%	78,80%	1,1	14,00%
NTI/UI - Unix a Internet	MTI/PMZ - Programování mobilních zařízení + NTI/PBE - Počítačová bezpečnost	50,40%	76,40%	1,23	15,60%
MTI/PMZ - Programování mobilních zařízení	NTI/UI - Unix a Internet	52,00%	72,60%	1,1	19,60%
MTI/PMZ - Programování mobilních zařízení	NTI/PBE - Počítačová bezpečnost + NTI/UI - Unix a Internet	50,40%	70,40%	1,14	21,20%
NTI/PBE - Počítačová bezpečnost	MTI/PMZ - Programování mobilních zařízení + NTI/UI - Unix a Internet	50,40%	64,30%	1,24	28,00%

PŘÍLOHA 4

Obsah přiloženého CD

Text diplomové práce

- Diplomova_prace_2017_Marketa_Mala.docx
- Diplomova_prace_2017_Marketa_Mala.pdf
- Kopie zadání diplomové práce: zadani_prace.pdf

Aplikace

- Soubory potřebné ke spuštění aplikace v adresáři *Soubory aplikace*
- Spustitelná aplikace: *App.exe* v adresáři *Aplikace/App/bin/Debug*
- Datové soubory použitelné k testování aplikace v adresáři *Transakcni data*