Ferdův návod k prvnímu projektu

Martin Ralbovský

<martin.ralbovsky@gmail.com>

2006-6-3

Obsah

Abstrakt

Tento dokument je určen začátečníkům ve Ferdovi, kteří neví jak se systémem pracovat. Obsahuje postup při vytvoření jednoduché 4ft úlohy nad zkušební databází fiktivní banky Barbora.

Úvod

Vítejte ve Ferdovi ;) Jestliže jste program Ferda spustili a nevíte si s ním rady, nebo jestli se potřebujete doučit některou ze základních dovedností Ferdy, čtěte pozorně tento návod a doufáme že po jeho přečtení vám bude vše jasnější.

Systém Ferda a LISp-Miner v kostce

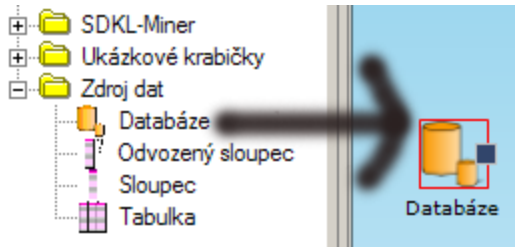
Ferda je systém pro dobývání znalostí z dat. Navazuje na starší systém LISp-Miner, který se už řadu let úspěšně vyvíjí na VŠE v Praze. Ferda i LISp-Miner dobývají znalosti pomocí asociačních pravidel a používají k tomu rozšíření metody GUHA. Podrobnější teoretické informace o procedurách najdete na webových stránkách

Náš příklad

V tomto návodu se pokusíme získat znalosti z údajů fiktivní banky Barbora o výhodnosti jejich půjček. Použijeme k tomu nejznámější z LISp procedur, proceduru 4ft. Tato procedura počítá čtyřpolní kontingenční tabulky jednotlivých hypotéz a porovnává je vzhledem ke kvantifikátorům. Více o proceduře a jejích parametrech se dozvíme v průbehu tohoto návodu.

1. Vytvoření první krabičky

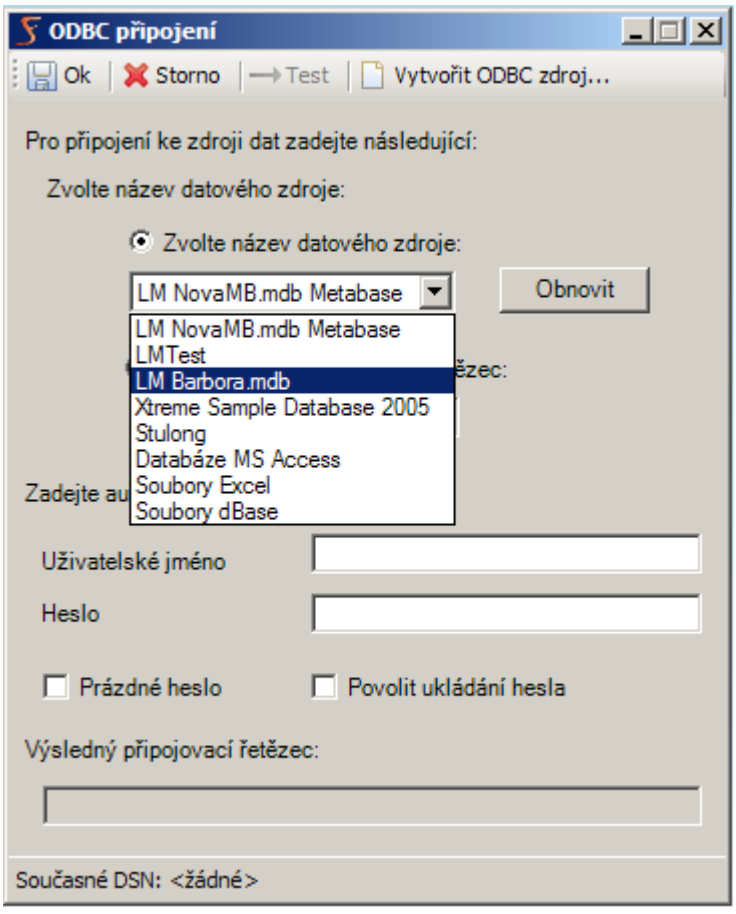
Ferda je systém, který používá pro tvorbu svých úloh tzv. vizuální programování, tj. úlohy se tvoří na ploše propojováním jednotlivých vizuálních prvků a nastavování vlastností těchto prvků. Ve Ferdovi se vizuální prvek nazývá krabička. Panel se všemi krabičkami, které máme k dispozici, je na levé straně dokovacího prostředí a obsahuje několik složek. Proces dobývání obvykle začíná identifikací datového zdroje. K tomu slouží krabička `Databáze` ve složce `Zdroj dat`. Kliknutím na krabičku a jejím přetažením na pracovní plochu vytvoříme svoji první krabičku.



2. Nastavení ODBC připojení a Moduly pro nastavení

Rozhraní ODBC poskytuje jednotný způsob připojení k datovému zdroji pro různé typy zdrojů. Pro náš příklad použijeme databázi fiktivní banky Barbora s informacemi o účtech. Můžete si ji stáhnout spolu se instalačním programem Ferdy na domovské stránce Ferdy.

K tomu, aby krabička `Databáze` správně fungovala, potřebujeme ji připojit k datovému zdroji. K tomu slouží vlastnost `ODBC připojovací řetězec`. Když klikneme levým tlačítkem na krabičku na ploše, zobrazí se nám všechny její vlastnosti v `Panelu Vlastostí`. v levé horní části dokovacího prostředí. Jesliže na vlastnost `ODBC připojovací řetězec` klikneme, můžeme buď vepsat připojovací řetězec rovnou do políčka, nebo kliknout na tlačítko se třema tečkama a použít pro nastavení vlastnosti modul. Modulů pro nastavení se využívá, jestliže je nastavování nějakým způsobem složité. Po kliknutí na tečky se zobrazí dialog pro nastavení ODBC připojení.



V dialogu se zobrazí všechny definované ODBC zdroje. Jestliže jste ještě Barboru nezavedli jako datový

zdroj, klikněte na `Vytvořit datový zdroj` na tlačítkové liště. V tomto dialogu můžete mimojiné testovat připojení k definovaným datovým zdrojům. Vyberte ze seznamu datový zdroj odpovídající Barboře. K tomu, abychom zjistili zda je datový zdroj platný, můžeme použít tlačítko `Test` na dialogu.

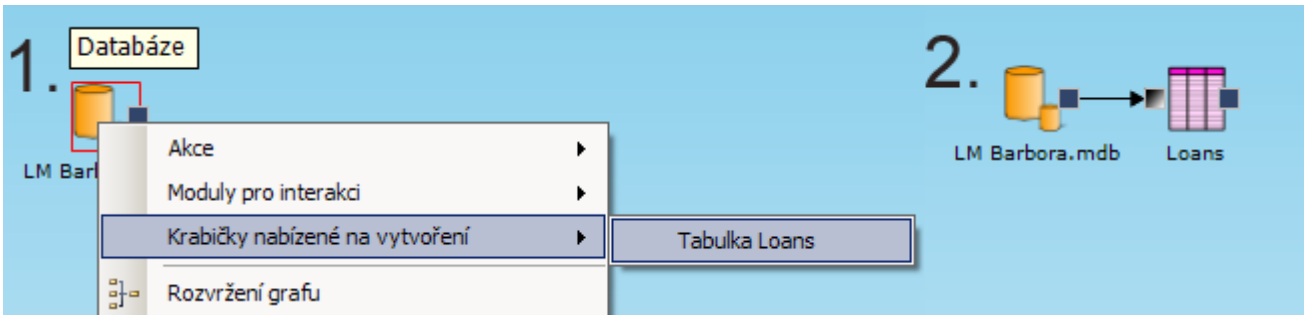
Poznámka

Ferda podporuje pouze vytvoření Uživatelského a Systémového zdroje dat v prvním kroku průvodce, který se spustí (nepodporuje Souborový zdroj dat).

Pro znalce LM (LISp-Mineru): Při práci se systémem LISp-Miner bylo třeba definovat metabázi (obvykle soubor databáze Microsoft Access (LM_Empty.mdb)) pro uložení projektu a vzájemnou výměnu dat mezi jednotlivými moduly systému LISp-Miner. Používání metabáze je v systému Ferda zcela transparentní tj. uživatel o ní neví. Uživatel pracuje pouze se soubory projektu (*.xfp). (Implementace: kdykoli krabičky Ferdy spouští moduly LISp-Mineru (jeho *Gen.exe) dávají mu jako parametr metabázi, kterou předtím přímo vygenerují. Jakmile *Gen.exe ukončí svou práci, je výsledek přečten z metabáze a ta je smazána.)

3. Vytvoření tabulky a Krabičky nabízené na vytvoření

Když už máme vytvořenou a správně zapojenou krabičku `Databáze`, můžeme pomocí ní tvořit další krabičky. Slouží k tomu postup nazvaný `Krabičky nabízené na vytvoření`. Například krabička `Databáze` reprezentuje datový zdroj, který může obsahovat několik tabulek. Konkrétně databáze Barbora obsahuje jednu tabulku s názvem `Loans`. Proto z této krabičky lze vytvořit krabičku `Tabulka`, které se nastaví jméno v databázi na `Loans`. Uděláme to tak, že klikneme nad krabičku Barbory pravým tlačítkem a z kontextového menu krabičky vybereme skupinu `Krabičky nabízené na vytvoření` a v podmenu vybereme `Tabulka Loans`. Vytvoří se nová krabička, která nyní reprezentuje tabulku.



Krabička `Loans` se dá vytvořit i bez použití Krabiček nabízených na vytvoření. Obdobný postup by byl udělat prázdnou krabičku `Tabulka` přetažením z nových krabiček, propojením výstupní zásuvky krabičky `Barbora` se vstupní zásuvkou nové krabičky a nastavením vlastnosti `Jméno` na `Loans`.

Aby Ferdovy procedury pro dobývání znalostí mohli fungovat, musíme označit primární klíč tabulky, nad kterou pracujeme. Označme proto hodnotu `loan_id` ve vlastnosti `Primární klíč` tabulky.

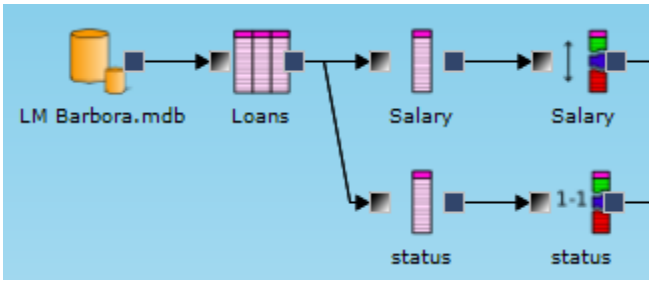
5. Vytvoření atributů

Princip krabiček nabízených na vytvoření lze aplikovat i na jiné krabičky než `Databáze`. Například `Tabulka` nabízí na vytvoření všechny sloupce, které obsahuje. Tabulka `Loans` obsahuje sloupce, které se týkají výhodnosti půjček u klientů s různými charakteristikami. Managera naší fiktivní banky by třeba mohlo zajímat, jakým způsobem na sobě závisí plat věřitele a výhodnost jeho úvěrů. Můžeme si následující úlohu zkusit vyřešit na Ferdovi.

Nejdříve vytvoříme krabičky `Sloupec` pro sloupce `Salary` udávající plat klienta a `status` pro výhodnost

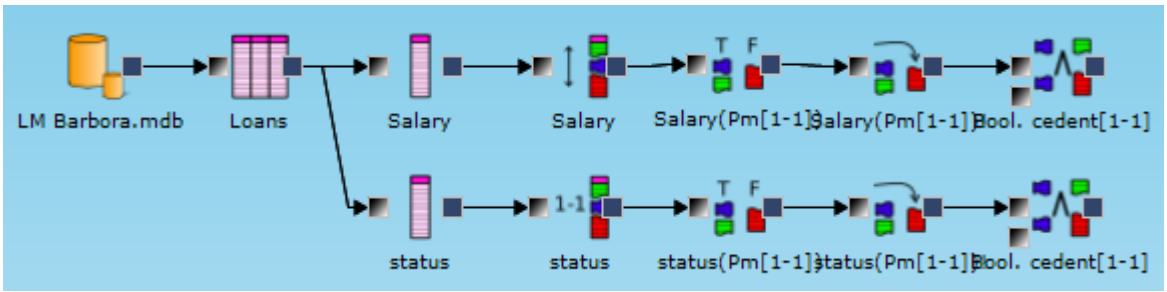
úroku. Poté z nich vytvoříme atributy. Atribut znamená vhodná kategorizace hodnot domény (sloupce). Jestliže si zobrazíme vlastnosti sloupce `Salary`, vidíme například že ve sloupci je 76 různých hodnot. Není zřejmě vhodné, abychom počítali se všemi hodnotami, mnohem lepší by bylo vytvořit jen pár intervalů, které by všechny možné hodnoty pokrývali. K tomuto slouží krabička nabízená na vytvoření `Equidistant intervals attribute`. Ta vezme všechny hodnoty sloupce a rozdělí je do x stejně dlouhých intervalů, kde x se dá zvolit v Panelu vlastností. Vytvořme tedy tuto krabičku a zadejme do vlastnosti `Délky` číslo 1000 - intervaly budou rozdělené po 1000 korunách.

Sloupec `status` má na druhou stranu jenom 4 možné hodnoty: A, B, C, D. V tomto případě již má smysl vytvořit pro každou hodnotu jednu kategorii. K tomu slouží krabička nabízená na vytvoření `Each value one category attribute`. Po vytvoření budeme mít atribut, který obsahuje 4 kategorie.



6. Tvorba úlohy až po zadání dílčích booleovských cedentů

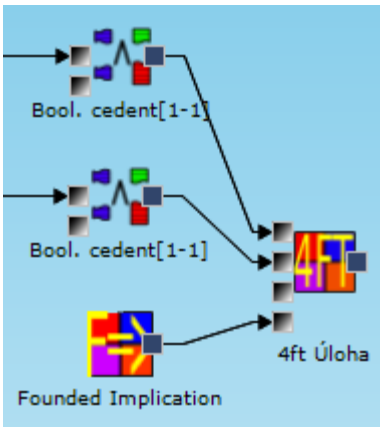
Pokračujme tedy v úloze a zadávejme pomocí krabiček všechny parametry které jsou potřebné pro zadání cedentu, který je už vstup pro 4ft proceduru. Budeme opět pokračovat pomocí krabiček nabízených na vytvoření. Nejdříve vytvoříme `Zadání atomu`. Tato krabička vyjadřuje, jakým způsobem budou kategorie vystupovat v pozdějším literálu. Pro náš příklad ponecháme všechny implicitní vlastnosti. Dále pokračujeme `Zadaním literálu`. Literál je tvořen z atomů a je základním prvkem při konstrukci cedentů. U této krabičky se dá nastavit buď jeho typ, nebo znaménko. Zde nastavíme vlastnost `Znaménko literálu` na pozitivní, protože je lepší zajímat se o pozitivní literály (například je pro nás cennější informace o zákaznících se statutem A než o těch kteří mají jiný status). Nakonec vytvoříme ze `zadání literálu` krabičku `Booleovský (dílčí) cedent`. Cedent je konjunkce literálů a vstupuje do 4ft procedury jako jeden z parametrů. Ve vlastnostech krabičky se nastavuje délka dílčího cedentu, opět můžeme nechat výchozí hodnoty.



7. Dokončení 4ft úlohy

Procedura 4ft je nejrozšířenější a nejvíce intuitivní ze LISp-Miner procedur. Vstupují do ní 3 cedenty nazvané `antecedent`, `sukcedent` a `podmínka`. Procedura zkoumá souvislost antecedentu a sukcedentu za dané podmínky. Pro výpočet úlohy také `4ft Kvantifikátor`. Je to vzoreček (funkce) definovaná nad čtyřpolní tabulkou, vůči kterému se ověřují hypotézy. Pro náš příklad použijeme nejčastější 4ft kvantifikátor, `fundovanou implikaci`. Tento kvantifikátor má jeden vstupní parametr `p` a testuje se, jestli alespoň `p` procent objektů splňující antecedent splňuje také sukcedent. Neboli jinými slovy jestli to že objekt splňuje antecedent, znamená že splňuje také sukcedent alespoň v `p` procentech případů.

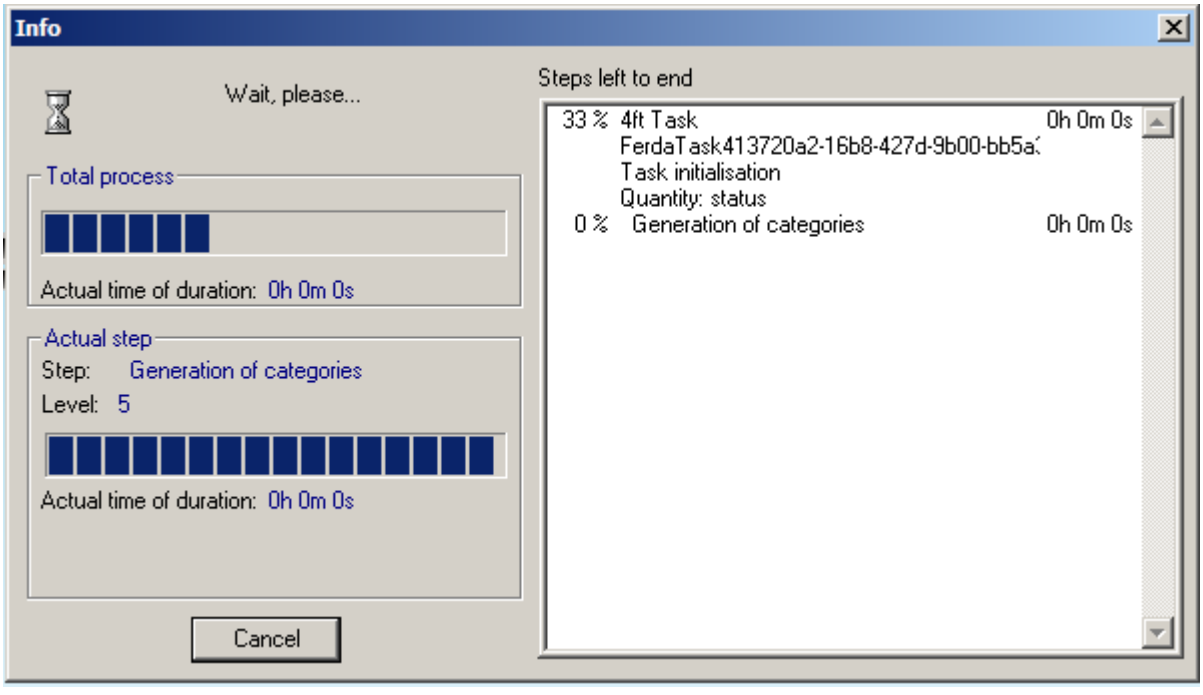
Nyní už máme připravené vše abychom mohli dokončit naši úlohu. Z panelu `Nové krabičky` ze složky `4ft-Miner` přetáhneme za dílčí cedenty krabičku `4ft Úloha`. Tato krabička má 4 vstupní zásuvky - `Zadání antecedentu`, `Zadání sukcedentu`, `Zadání podmínky` a `4ft Kvantifikátor`. Nyní do `Zadání antecedentu` zapojíme zadání cedentu vzniklého ze sloupce `Salary`, do `Zadání sukcedentu` zapojíme zadání cedentu vzniklého ze sloupce `status`. Podmínku v tomto případě nepoužijeme. Do poslední zásuvky pro kvantifikátor potřebujeme kvantifikátor `Founded implication`, který najdeme v nových krabičkách ve složce `Kvantifikátory` a `4ft-Miner`. Úloha je připravená ke spuštění.



8. Spuštění 4ft úlohy a Akce krabiček

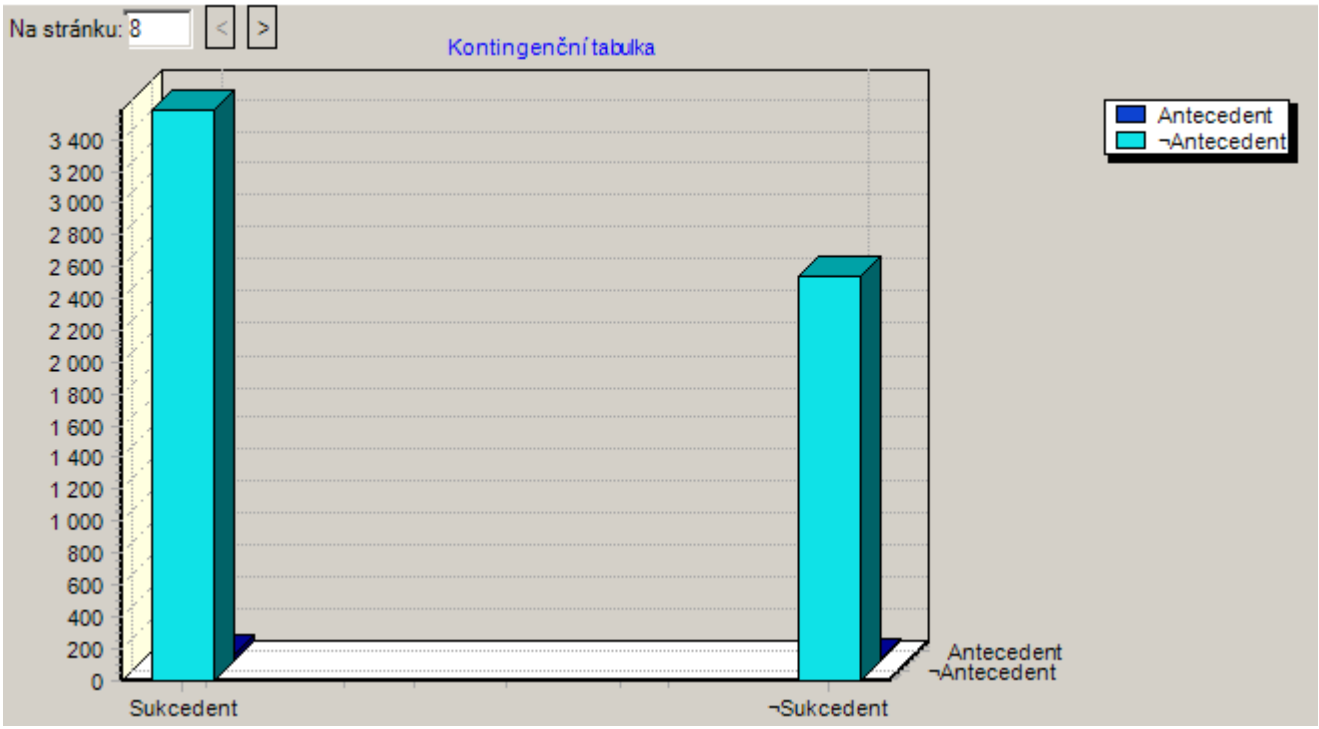
Spouštění akcí krabiček je další možné využití krabičky. Akce definované pro krabičku jsou přístupné z položky `Akce` kontextového menu nad krabičkou. Jak vidíme krabička `4ft Úloha` má pouze jednu akci nazvanou `Spusť`. Tato akce spustí generování hypotéz pro příslušnou úlohu. Můžete si všimnout že i jiné krabičky mají své akce.

Když tedy klikneme na spuštění generování hypotéz, spustí se dialog, ve kterém je znázorněný průběh generování. Poté znovu označíme na ploše krabičku 4ft úlohy. Pozorný uživatel si všimne, že se krabičce změnili vlastnosti. Vlastnost `Stav` ukazuje hodnotu `Completed`, to znamená že generování doběhlo. Bohužel se nám z 16 verifikací nepodařilo vybrat žádnou, která by vyhovovala podmínkám kvantifikátoru. Nastavíme tedy vlastnost `p` v kvantifikátoru z 0,9 na 0,8 a zkusíme spustit úlohu znovu. Výborně, nyní už jedna hypotéza prošla.



9. Zobrazení výsledků a Moduly pro interakci

Pomocí `Modulů pro interakci` krabičky zobrazují svůj obsah uživateli. Takže abychom zobrazili naše hypotézy, použijeme k tomu modul pro interakci `Modul pro prohlížení výsledků` krabičky `4ft Úloha`. V kontextovém menu nad označenou krabičkou na něho klikneme v podmenu `Moduly pro interakci`. Následně se nám zobrazí tabulka s hypotézami a pod ní prostor pro graf. Jestliže vybereme naši (jedinou) hypotézu, ukáže se nám graf reprezentující čtyřpolní tabulku pro danou hypotézu a v Panelu vlastností se zobrazí další informace o hypotéze.



10. Interpretace výsledků

Výsledná hypotéza říka, že dvojice antecedent - plat mezi 11110 a 12110 a sukcedent - status C splňuje kvantifikátor fundované implikace na 80%. To znamená, že jestliže bude mít některý klient tento plat, tak je velká pravděpodobnost, že jeho výsledný status bude C. Na první pohled se to může zdát jako dobrý výsledek. Když se ale podíváme na čtyřpolní tabulku, vidíme že podpora této hypotézy je velmi malá - antecedent splňuje jenom 45 záznamů z celkových asi 6000 které v tabulce máme (a to není ani 1 procento). Když se podíváme na modul pro interakci `Modul frekvence sloupců` u atributu s platem, zjistíme velmi nerovnoměrné rozložení hodnot. Nabízí se nám možnost například použít ekvifrekvenčního atributu, modifikace kvantifikátorů.... Doufáme, že nyní už víte, jak s Ferdou pracovat aby ste dosáhli co nejlepších výsledků. Hodně štěstí.