Data Mining – podstawy analizy danych
Cześć pierwsza

Wprowadzenie

Usługa Data Mining w środowisku serwera SQL 2005 jest jednym z komponentów technologii business intelligence. Umożliwia ona budowę złożonych modeli analitycznych oraz powiązanie tych działań z zastosowaniami biznesowymi, pozwalające podejmować strategiczne decyzje marketingowe, odpowiadając na kluczowe pytania z zakresu prowadzenia tej działalności:

- Jakie produkty ludzie są skłonni kupować razem?
- Jaka jest spodziewana sprzedaż produktu w następnym miesiącu?
- Jakie jest ryzyko kredytowe związane z tym klientem?
- Jaki jest ogólny profil moich klientów?

Większość osób najprawdopodobniej korzystała z aplikacji Data Mining. Jeśli kupowaliśmy w sklepie internetowym płytę CD lub książkę, w wielu sytuacjach otrzymywaliśmy podpowiedź, że „…inni klienci wraz z tą książką kupowali …". Podobnie, w przypadku gdy klient korzysta z określonego ubezpieczenia (np. domu), proponuje mu się odpowiedni pakiet ubezpieczeń na samochód, porównując przy pomocy usług analitycznych, propozycje przedstawiane innym klientom i efekty ekonomiczne wynikające z takiego a nie innego pakietu ubezpieczeń.

SQL Server 2005 Data Mining jest częścią technologii business intelligence, która może być powiązana z pozostałymi komponentami serwera:

- SQL Server 2005 Integration Services. Umożliwia budowę złożonych mechanizmów przepływu danych
- SQL Server 2005 Analysis Services. Umożliwia budowę całościowego modelu analizy danych poprzez dodanie mechanizmów business intelligence do struktur data mining.
- SQL Server Reporting Services. Usługa umożliwiające realizowanie elastycznych i efektywnych raportów opartych o usługi serwera web.

Najważniejszymi cechami opisującymi usługę Data Mining są: duża skalowalność, łatwe zarządzanie, możliwość wdrożenia mechanizmów zabezpieczeń oraz szeroka dostępność.

SQL Server Data Mining oparty jest o architekturę klient-serwer. Pozwala to wykorzystywać usługi data mining zarówno w sieciach lokalnych jak i rozległych. Standardowy interfejs programowy (API) pozwalają budować aplikację na różnych platformach klienta. Architektura usług data mining od podstaw posiada architekturę równoległą, co pozwala wykorzystywać opisywane rozwiązania zarówno w niewielkich środowiskach jak i na platformie obsługującej tysiące użytkowników realizujących miliony zapytań w ciągu dnia. Zastosowanie zintegrowanego narzędzia administracyjnego pozwala budować skomplikowane modele analityczne bez konieczności używania dodatkowych środowisk programistycznych. Zastosowanie mechanizmów bezpieczeństwa w usługach data mining pozwala przypisywać poszczególnym użytkowników zakresy uprawnień do zdefiniowanego modelu analitycznego.

Proces analizy danych możemy podzielić na trzy etapy:

- tworzenie modelu analitycznego i „nauczenie" go na danych treningowych, gdzie odpowiedzi są znane a'priori. W naszym przykładzie, kolumna zawierająca informacje dotyczące zakupu roweru („yes" lub „no") jest wypełniona (patrz – „zdefiniowanie problemu"). Określony algorytm dokona analizy danych, wykrywając zależności pomiędzy parametrami wejściowymi a wartością przewidywaną.
- Drugi etap umożliwia predykcję danych, a więc określenie na podstawie parametrów wejściowych, identycznych jak w etapie pierwszym, jaką wartość powinna przyjąć kolumna „kupi rower". Oczywiście analizowane dane nie dają stuprocentowej pewności. Mamy tu do czynienia ze statystyką – czyli możemy mówić o prawdopodobieństwie wystąpienia takiego zdarzenia. W zależności od zastosowanego algorytmu oraz opisu problemu mogą się pojawić rozbieżności w wynikach. Rozwiązaniem tej sprzeczności jest trzeci etap procesu analizy.
- Potwierdzenie przewidywanych danych może nastąpić w momencie, kiedy dane zjawiska sprawdzą się w rzeczywistości. Etap ten pozwala porównać przewidywania przy pomocy różnych algorytmów z faktycznymi wartościami uzyskanymi w życiu i określić, który z algorytmów najlepiej odzwierciedla rozwiązanie naszego problemu.

Usługi Data Mining w SQL 2005 pozwalają zrealizować wszystkie trzy etapy analizy danych.

Zdefiniowanie problemu

Usługi Data Mining w SQL Server 2005 wykorzystują szereg nowych algorytmów, umożliwiających budowę złożonych mechanizmów analitycznych dla różnych zastosowań. Algorytmy możemy podzielić na kilka kategorii:

- Algorytmy klasyfikujące pozwalające przyporządkować dane w kategorii np. „Dobry" lub „Zły". Tego typy algorytmy umożliwiają ocenę ryzyka kredytowego, analizę sprzedaży czy też wybór produktu dla którego zostanie zrealizowana kampania marketingowa. Przedstawicielem tej klasy algorytmów są : Drzewo decyzyjne (decision tree), naiwny klasyfikator Bayes'a (naive Bayes) czy też sieć neuronowa (neural nets).
- Algorytmy grupujące, pozwalające pogrupować dane, posiadające podobne cechy np. według wieku, czy tez ceny produktu. Wykorzystanie tego typu algorytmów do kampanii mailowej, czy też analiza profilu klientów przynosi znakomite skutki. Przedstawicielami tego typu algorytmów w SQL Server 2005 są algorytmy: podobieństwa (clustering) i podobieństwa sekwencyjne (sequence clustering).
- Algorytmy kojarzące, pozwalają wyszukiwać zaawansowane korelacje pomiędzy danymi wejściowymi. Tego typu algorytmy pozwalają analizować zaawansowane powiązania w procesie sprzedaży produktów (koszyk sprzedaży) czy też powiązania pomiędzy poszczególnymi produktami. Przedstawicielami tej grupy algorytmów są: reguły związków (association rules).
- Algorytmy umożliwiające prognozowanie sprzedaży czy też ilości produktów na magazynie. Przedstawicielem tej grupy jest algorytm szeregi czasowe (time series).

Budowę modelu analitycznego musimy rozpocząć od zdefiniowania problemu, który zamierzamy rozwiązać. W naszym przykładzie spróbujemy przedstawić zależność decyzji o

zakupie roweru w odniesieniu od określonych cech konsumentów: liczby dzieci, liczby samochodów, przychodów rocznych, płci, odległości do miejsca pracy oraz wieku.
W pierwszym etapie przeanalizujemy dane treningowe przy pomocy drzewo decyzyjnego. Następnie rozbudujemy ten etap o analizę przy pomocy naiwnego klasyfikatora Bayes'a.
W drugim etapie dokonamy predykcji danych, chcąc określić prawdopodobieństwo zakupu rowerów przy pomocy obu algorytmów.
W trzecim etapie dokonamy porównania wyników rzeczywistych z przewidywaniami przy pomocy obu algorytmów.
Do rozwiązania naszego przykładu wykorzystamy dane zawarte w bazie danych AdventureWorksDW. Jednak w procesie budowy modelu analitycznego nie zawsze mamy gotowy zbiór danych do analizy. W naszym przykładzie musimy przygotować odpowiedni zbiór danych, wykorzystując tabelę Customers oraz powiązane z nią odpowiednie tabele.

Skrypt tworzący odpowiedni widok dla naszych potrzeb wygląda następująco:

```
/******  View [dbo].[vDMLabCustomerTrain]  ******/

CREATE VIEW [dbo].[vDMLabCustomerTrain]  AS
SELECT
        c.[CustomerKey],
        c.[FirstName],
        c.[LastName],
        CASE
                WHEN Month(GetDate()) < Month(c.[BirthDate])
                        THEN DateDiff(yy,c.[BirthDate],GetDate()) - 1
                WHEN Month(GetDate()) = Month(c.[BirthDate])
                AND Day(GetDate()) < Day(c.[BirthDate])
                        THEN DateDiff(yy,c.[BirthDate],GetDate()) - 1
                ELSE DateDiff(yy,c.[BirthDate],GetDate())
        END AS [Age],
        c.[MaritalStatus],
        c.[Gender],
        c.[YearlyIncome],
        c.[NumberChildrenAtHome],
        CASE c.[HouseOwnerFlag] WHEN 0 THEN 'No' ELSE 'Yes' END as HouseOwner,
        c.[NumberCarsOwned],
        g.EnglishCountryRegionName,
        c.[CommuteDistance] As Commute,
        CASE x.[Bikes]
                WHEN 0 THEN 'No'
                ELSE 'Yes'
        END AS [BikeBuyer]
    FROM
        [dbo].[DimCustomer] c INNER JOIN (
                SELECT
                        [CustomerKey]
                        ,[Region]
                        ,[Age]
```

```
                ,Sum(
                        CASE [EnglishProductCategoryName]
                                WHEN 'Bikes' THEN 1
                                ELSE 0
                        END) AS [Bikes]
        FROM
                [dbo].[vDMPrep]
        GROUP BY
                [CustomerKey]
                ,[Region]
                ,[Age]
        ) AS [x]
    ON c.[CustomerKey] = x.[CustomerKey]
join dimgeography g
on c.geographykey = g.geographykey
go
```

View - dbo.vD...CustomerTrain | MIA-SQL\SQLIN...wCreation.sql

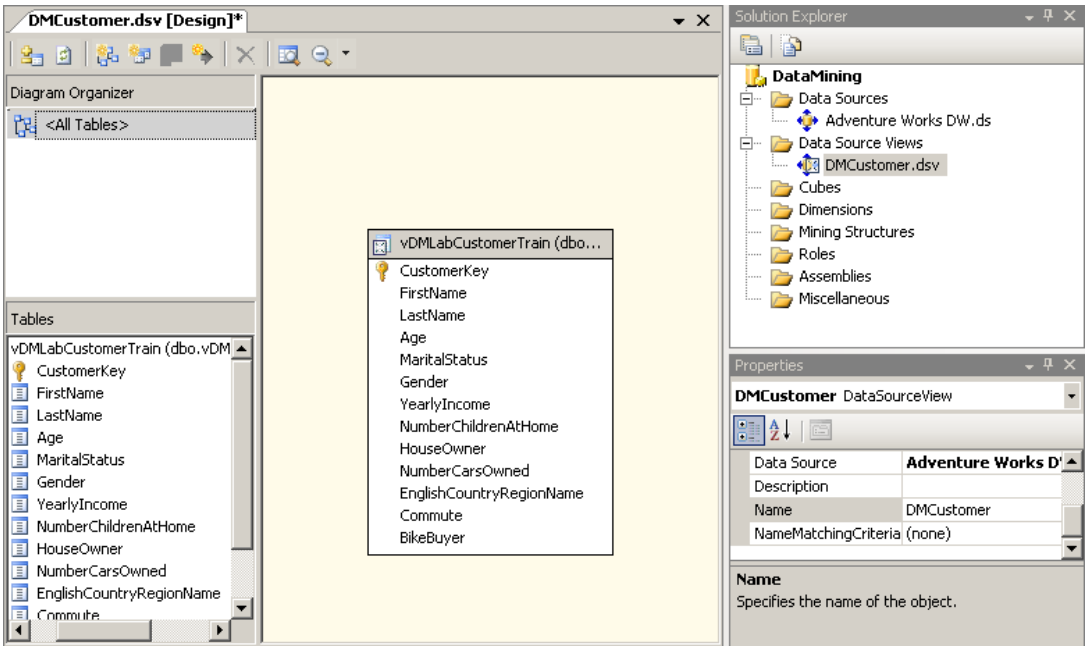
Custo...	FirstName	LastName	Age	Mari...	Ge...	YearlyIncome	Num...	H...	N...	EnglishCountry...	Commute	BikeBu...
11000	Jon	Yang	40	M	M	90000.0000	0	Yes	0	Australia	1-2 Miles	Yes
11001	Eugene	Huang	41	S	M	60000.0000	3	No	1	Australia	0-1 Miles	Yes
11002	Ruben	Torres	41	M	M	60000.0000	3	Yes	1	Australia	2-5 Miles	Yes
11003	Christy	Zhu	38	S	F	70000.0000	0	No	1	Australia	5-10 Miles	Yes
11004	Elizabeth	Johnson	38	S	F	80000.0000	5	Yes	4	Australia	1-2 Miles	Yes
11005	Julio	Ruiz	41	S	M	70000.0000	0	Yes	1	Australia	5-10 Miles	Yes
11006	Janet	Alvarez	41	S	F	70000.0000	0	Yes	1	Australia	5-10 Miles	Yes
11007	Marco	Mehta	42	M	M	60000.0000	3	Yes	2	Australia	0-1 Miles	Yes
11008	Rob	Verhoff	42	S	F	60000.0000	4	Yes	3	Australia	10+ Miles	Yes
11009	Shannon	Carlson	42	S	M	70000.0000	0	No	1	Australia	5-10 Miles	Yes
11010	Jacquelyn	Suarez	42	S	F	70000.0000	0	No	1	Australia	5-10 Miles	Yes
11011	Curtis	Lu	43	M	M	60000.0000	4	Yes	4	Australia	10+ Miles	Yes
11012	Lauren	Walker	39	M	F	100000.0000	0	Yes	2	United States	1-2 Miles	No
11013	Ian	Jenkins	38	M	M	100000.0000	0	Yes	3	United States	0-1 Miles	No
11014	Sydney	Bennett	38	S	F	100000.0000	0	No	3	United States	1-2 Miles	No
11015	Chloe	Young	27	S	F	30000.0000	0	No	1	United States	5-10 Miles	Yes
11016	Wyatt	Hill	27	M	M	30000.0000	0	Yes	1	United States	5-10 Miles	Yes
11017	Shannon	Wang	62	S	F	20000.0000	0	Yes	2	Australia	5-10 Miles	Yes
11018	Clarence	Rai	62	S	M	30000.0000	0	Yes	2	Australia	5-10 Miles	Yes

1 of 18484

Rys 1. Struktura danych w widoku ***vDMLabCustomerTrain***

Zwracam uwagę na odpowiednie przygotowanie danych, dotyczących np. posiadanych rowerów czy też posiadanych domów. W definiowanych strukturach danych musi wystąpić unikalny klucz (pole) pozwalający jednoznacznie określić daną krotkę.
Mając tak zdefiniowane parametry projektu możemy przejść do technicznych aspektów tworzenia modelu analitycznego.

Etap pierwszy – wykorzystanie danych treningowych do analizy
Budowę modelu rozpoczynamy od uruchomienia BI Development Studio i zdefiniowania nowego projektu – nazwijmy go „DataMinig”. W procesie generowania projektu, następuje zdefiniowanie odpowiednich folderów i struktur danych.
Możemy przystąpić do zdefiniowania źródła danych i widoku źródła danych. W naszym przypadku źródłem danych jest baza danych AdventureWorksDW a jako widok źródła danych definiujemy wcześniej utworzony widok w tabeli o nazwie: *dbo.vDMLabCustomerTrain*.



Rys 2. Definicja źródeł danych (Adventure Works DW) i widoku źródeł danych (DMCustomer)

Źródło danych (Data Source) przechowuje informacje niezbędne do połączenia się ze źródłem danych. Widoki źródeł danych (Data Source Views) zawierają informacje na temat istotnych podzbiorów tabel w źródłowej bazie danych. Te informacje nie są ograniczone do fizycznej struktury tabel w źródłowej bazie danych; można dodać informacje takie jak relacje, przyjazne nazwy tabel i kolumn, kolumny obliczane i nazwane zapytania.

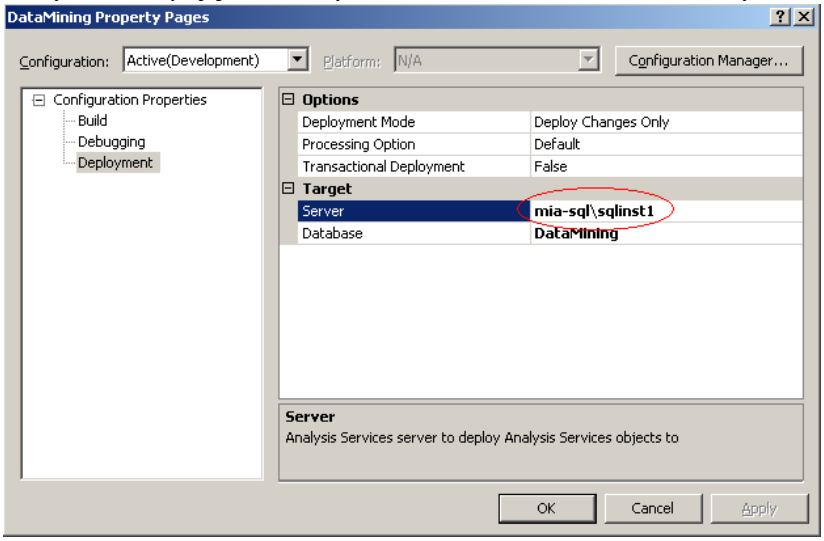
Widoki źródeł danych mogą być udostępniane między projektami analitycznymi (kostki analityczne, data mining oraz projektami SSIS). Widoki źródeł danych są bardzo przydatne, ale zwłaszcza gdy mamy do czynienia z następującymi sytuacjami:

- Źródłowa baza danych zawiera tysiące tabel, z których jedynie względnie niewielka liczba jest przydatna w naszym projekcie,
- Baza danych usług Analysis Services korzysta z danych z różnych źródeł np. bazy OLTP, OLAP, bazy Access, pliki XML znajdujące się na różnych serwerach,
- Twórca projektu analitycznego musi pracować w trybie offline, bez połączenia ze źródłową bazą danych. Zadania projektowania i programowania mają miejsce w oparciu o widok źródła danych, który jest odłączony od źródłowej bazy danych.

Przystąpmy do tworzenia właściwego modelu analitycznego. Kliknięcie prawym przyciskiem myszy na folderze „Mining Structures” i wybieramy opcję „New Mining Structure”. Efektem naszego działania będzie pojawienie się kreatora, za pomocą którego zbudujemy podstawy modelu analitycznego. W pierwszym kroku określamy źródło danych – w naszym przypadku będzie to baza OLTP. Następnie przechodzimy do wyboru algorytmu. Tak jak w założeniach projektu, wybieramy algorytm drzewa decyzyjnego.

Zwracam uwagę, że w tym kroku kreator łączy się już z serwerem analitycznym, pobierając listę dostępnych algorytmów. W przypadku, gdy zamierzamy się połączyć z określonym serwerem, bądź z określoną instancją, we właściwościach projektu należy zdefiniować

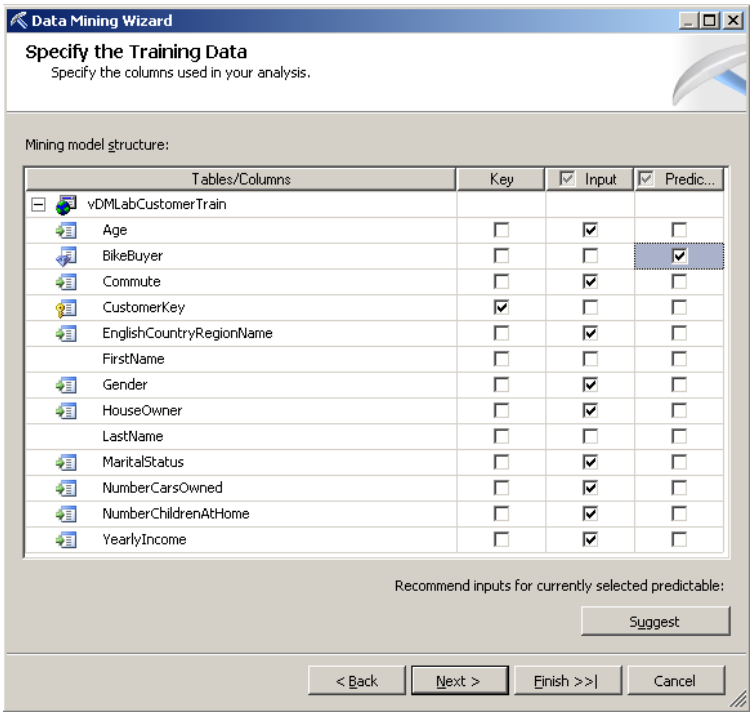
parametry połączenia – z menu „project” wybieramy opcję „DataMining properties”. Oczywiście opcję te należy definiować zanim uruchomimy kreatora.



Rys 3. Parametry projektu

Gdy mamy wybrany algorytm analizy, możemy przejść do wyboru źródła danych. Kreator w tym kroku wyświetla nam listę zdefiniowanych widoków źródeł danych. W naszym przypadku jest to oczywiście „DMCustomer”. Kolejny krok to wybór odpowiedniego zapytania, widoku bądź tabeli. W naszym projekcie dostępny jest widok *vDMLabCustomerTrain.* Oczywiście wskazany widok jest elementem wyboru – stąd zaznaczona opcja „case”.

Kolejny krok pozwala nam zdefiniować, które z wartości są parametrami wejściowymi a które elementy są przewidywane. Zaznaczając kolumnę „BikeBuyer” jako wartość przewidywaną możemy wykorzystać opcję „Suggest”, która pokaże od jakich wartości wejściowych zależy zaznaczona kolumna. Pole „Score” określa poziom, jaki ma wpływ dana kolumna na wartość przewidywaną.W naszym przypadku są to wszystkie kolumny z wyjątkiem FirstName, LastName (będące w istocie elementami unikalnymi dla każdej krotki) oraz wcześniej zaznaczone pola: CustomerKey (klucz unikalny) oraz „BikeBuyer” – wartość przewidywana.



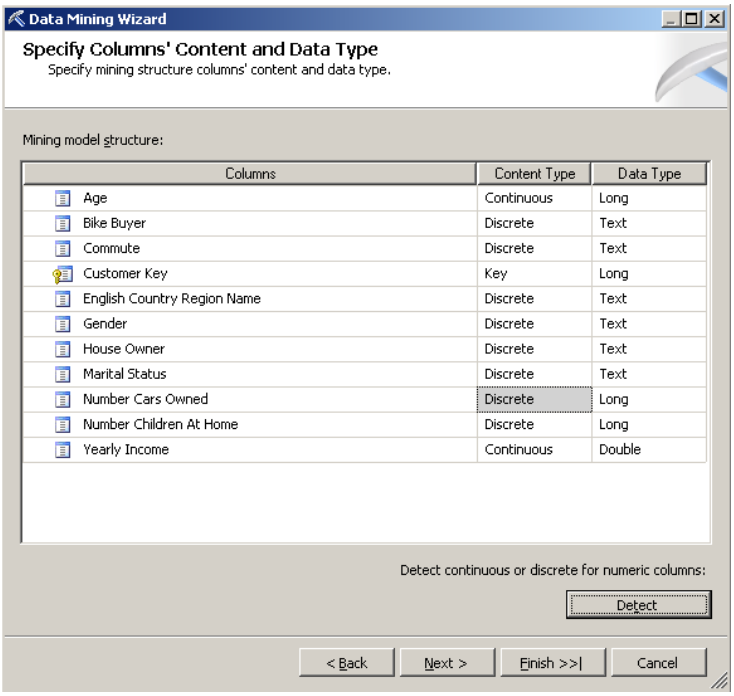
Rys 4. Definicja danych do analizy

Kolejny krok budowy modelu analitycznego pozwala nam określić typ danych występujących w poszczególnych polach oraz rodzaj danych. W zależności od wykorzystywanych algorytmów, dopuszczalne są różne typy danych:

- ciągłe (continuous)
- cykliczne (cyclical)
- nieciągłe (discrete)
- przedziałowe(discretized)
- uporządkowane (oredered)

Istnieje możliwość zmiany typów danych z wykorzystaniem widoków źródeł danych.

W tym kroku również możemy wykorzystać sugestię kreatora, w celu określenia właściwych rodzajów i typów danych.



Rys 5. Typy i rodzaj danych

Ostatnim krokiem w procesie budowy projektu jest zdefiniowanie nazwy naszego modelu analitycznego oraz samej struktury – nadajmy nazwę „DMCustomer_DT”. Zaznaczenie opcji „allow grill thru” pozwala analizować szczegółowe dane.

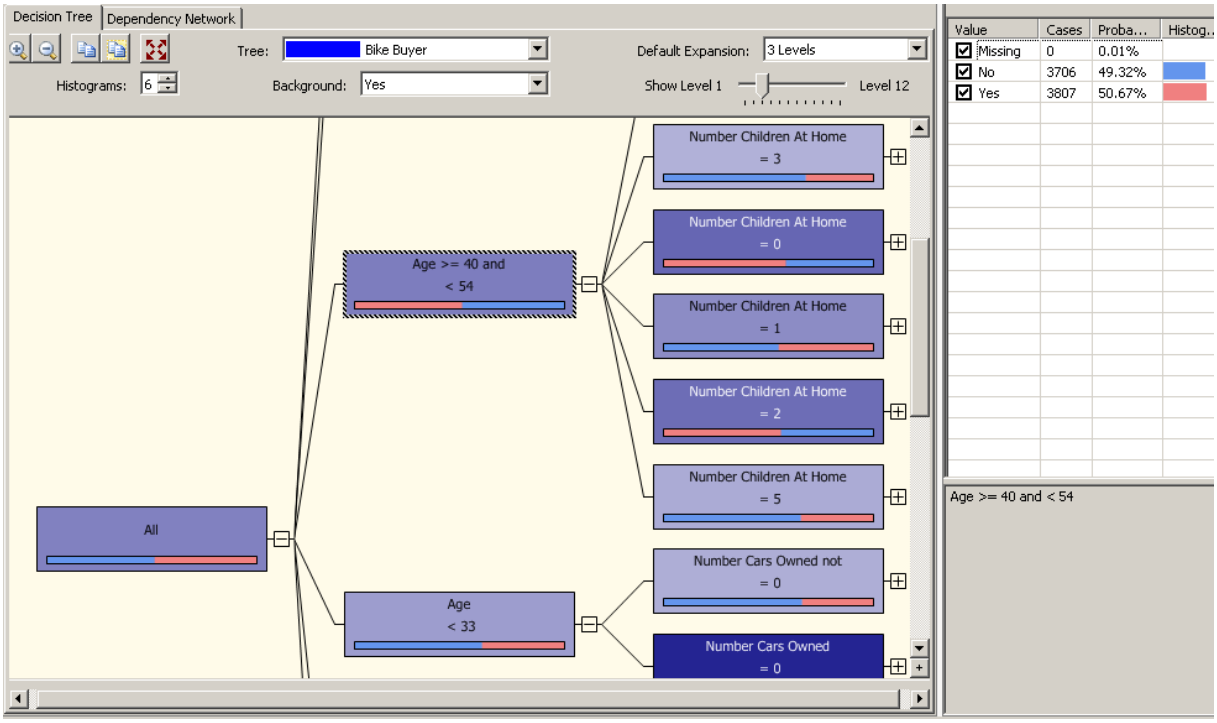
Po kliknięciu na przycisk „Finsz” następuje wygenerowanie struktur modelu, jednak na razie bez przetwarzania danych. Przełączając się pomiędzy odpowiednimi zakładkami możemy w dalszym ciągu dokonywać zmian w samej strukturze modelu.

Wykonajmy proces analizy danych na podstawie zbudowanego modelu. W tym celu z menu „Bulid” wybieramy opcje „build”. Jeżeli nie popełniliśmy błędu, następuje wygenerowanie informacji o powiązaniu poszczególnych danych. Do przeglądania uzyskanych wyników, BI Development Studio posiada wbudowaną przeglądarkę do mechanizmów data mining.

Przełączając się na zakładkę „Mining Model Viewer” możemy przeanalizować struktury danych. Dobierając odpowiednio parametry do przeglądania możemy:

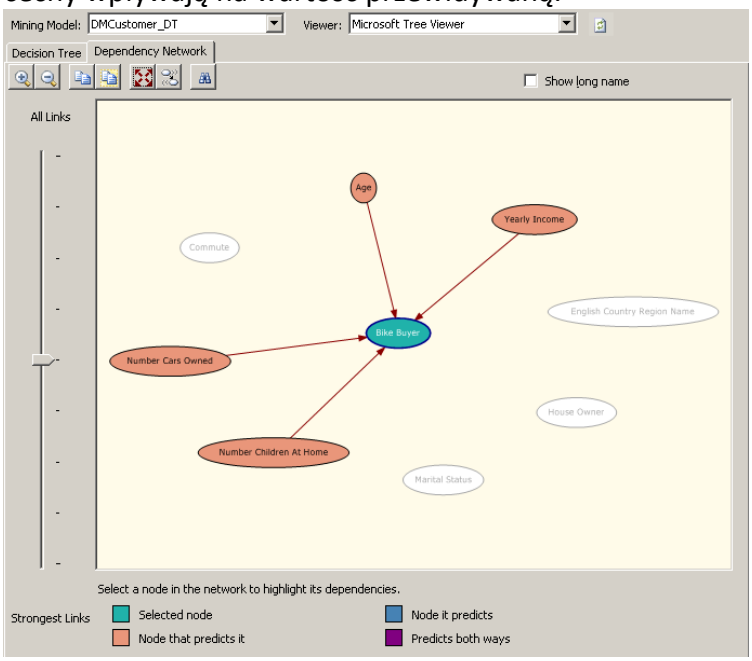
- Ustalić ilość poziomów przeglądanego drzewa,
- Określić jakie znaczenie ma zabarwienie poszczególnych obiektów
- Ustalić powiększenie

Sposób prezentowania danych jest uzależniony od wybranego algorytmu analitycznego. W przypadku drzewa decyzyjnego, możemy zobaczyć jak kształtują się zależności pomiędzy poszczególnymi cechami i wartością przewidywaną. Intensywność zabarwienia poszczególnych kostek może określać całą populację, wartość pozytywną cechy (yes) lub negatywną (no) związana z daną cechą. W interpretacji wyników pomaga też legenda pojawiająca się po prawej stronie, pokazująca w postaci histogramu wartość danej cechy. Ponadto możemy na podstawie „odległości” danej cechy w drzewie od korzenia określić jej wpływ na wartość przewidywaną. Jak widać z poniższego wykresu cecha „wiek” jest jedną z dominujących wartości.



Rys 6 Przeglądarka modelów analitycznych

Precyzyjnie możemy określić zależności pomiędzy cechami przełączając się na zakładkę „dependency network”. Przesuwając suwak pionowy możemy zobaczyć w jakiej kolejności cechy wpływają na wartość przewidywaną.



Rys 9. Zależność wartości przewidywanej od parametrów wejściowych

NetSystem Tomasz Skurniak
CNI, CNE, MCT, MCSE, MCDBA, MCTS, MCITP
Ul. J. Burszty 25, 61-422 Poznań
E: Tomasz@Skurniak.pl
W: www.protis.pl
T: +48 601761013
F: +48 618308249