Adriana Maszorek, Adam Pelikant
Wydział Informatyki
Wyższej Szkoły Informatyki w Łodzi

ZASTOSOWANIE ROZSZERZENIA DMX DO ZGŁĘBIANIA DANYCH NA PLATFORMIE MICROSOFT SQL SERVER

Streszczenie – Praca zawiera opis technologii zgłębiania danych oraz języka DMX (Data Mining Extension) przeznaczonego do analizy eksploracyjnej w środowisku MS SQL Server, a także wizualnej alternatywy w postaci narzędzia Microsoft Visual Studio. Dodatkowo omówione zostały algorytmy zgłębiania danych zaimplementowane przez Microsoft, przy szczególnym uwzględnieniu algorytmu Microsoft Sequence Clustering. Głównym celem pracy oprócz charakterystyki metod zgłębiania danych jest zbudowanie aplikacji windowsowej umożliwiającej tworzenie, trenowanie oraz ocenę dokładności modeli eksploracyjnych przy pomocy wspomnianego rozszerzenia DMX. Aplikacja napisana została w technologii .NET w języku C# i do trenowania modeli wykorzystuje dwa algorytmy: Microsoft Clustering oraz Microsoft Sequence Clusterin, zachowując przy tym maksimum uniwersalności konfiguracji obu tych procesów. Warto dodać również, że do komunikacji pomiędzy aplikacją a serwerem analitycznym Microsoft Analysis Services wykorzystana została biblioteka ADOMD.NET.

1. Wstęp

Tematem pracy jest rozszerzenie DMX, dlatego na wstępie warto wyjaśnić i rozwinąć ten skrót. Rozszerzenie DMX (Data Mining Extension) jest to język zapytań służący do analizy danych poprzez zgłębianie na platformie Microsoft SQL Server Analysis Services. Oferuje on bardzo bogatą funkcjonalność głównie w zakresie analizy biznesowej, ale nie tylko. Chcąc dobrze zrozumieć omawiane zagadnienie należy teraz przybliżyć wymienione już pojęcie zgłębiania danych (Data Mining).

Zgłębianie danych lub inaczej eksploracja, wyodrębnianie, drążenie, czy ekstrakcja danych jest jednym z etapów w procesie odkrywania wiedzy w bazach danych (Knowledge Discovery in Databases – KDD). Polega on na wyszukiwaniu wzorców, prawidłowości i zależności w dużych repozytoriach danych oraz na prognozowaniu głownie zjawisk

biznesowych, ale również medycznych czy meteorologicznych. Przykładów zastosowań można wymieniać naprawdę dużo. Najczęściej jednak zgłębianie danych wykorzystywane jest do oceny szans pozyskania lub ryzyka odejścia klienta, analizy sprzedaży i co za tym idzie planowania dostaw, analizy koszyka produktów, analizy ryzyka kredytowego, wykrywania oszustw poprzez badanie anomalii, ogólnej analizy rynku oraz prowadzenia ukierunkowanych kampanii reklamowych, analizy witryn internetowych czy wreszcie prognozowania zjawisk na podstawie danych historycznych.

Zagadnienie związane z eksploracją danych jest interesujące z kilku powodów. Głównymi czynnikami są ogromne możliwości znajdujące zastosowanie w wielu dziedzinach życia oraz praktyczne zastosowanie w biznesie, ze szczególnym uwzględnieniem poznania wiarygodności omawianego narzędzia analitycznego oraz programowalnych możliwości manipulowania obiektami analitycznymi Microsoft SQL Analysis Services. Dodatkowo jest to temat bardzo aktualny, wychodzący na przeciw bieżącym potrzebom rynku.

Na przestrzeni ostatnich lat, w związku z rozwojem i upowszechnieniem systemów bazodanowych oraz spadkiem cen sprzętu komputerowego i bogatą ofertą systemów informatycznych dla różnej wielkości firm, dokumenty papierowe zostały wyparte przez dokumenty w postaci elektronicznej, co z kolei zaowocowało zgromadzeniem przez firmy ogromnych ilości danych. Niestety dotychczasowe narzędzia takie jak deklaratywny język SQL nie dawały dużych możliwości pozyskiwania przydatnej wiedzy analitycznej z tak ogromnych pokładów informacji. Powstała sytuacja, w której firmy tonęły w danych jednocześnie nie potrafiąc ich efektywnie wykorzystać, dlatego zgodnie z maksymą, w myśl której „potrzeba (bądź nuda) jest matką wynalazków", nastąpił rozwój technologii zwanej zgłębianiem danych.

2 Zgłębianie danych z Microsoft SQL Server 2005

2.1 Analiza poprzez zgłębianie

Dzieląc technikę eksploracji pod względem etapów funkcjonalnych można wyodrębnić osiem kroków, począwszy od nieprzetworzonych danych, a skończywszy na wiedzy biznesowej i integracji z aplikacjami klienckimi.

2.2 Nieprzetworzone dane

Proces eksploracji danych zaczyna się od surowego zbioru informacji, którego źródłem najczęściej są relacyjne systemy

przetwarzające dane. Systemy te gromadzą dane będące wynikiem codziennej działalności przedsiębiorstw czy instytucji. Dodatkowo może zdarzyć się również, że na wejściu otrzymamy już całkowicie lub częściowo przetworzone dane w postaci hurtowni (Data Warhouse - DW). Tego typu obiekty zazwyczaj gromadzą informacje pochodzące z kilku źródeł takich jak: zasoby archiwalne, systemy OLTP (Online Transaction Precessing) czy dane zewnętrzne w postaci plików.

2.3 Oczyszczanie i przekształcanie danych

Kolejnym krokiem w omawianym procesie jest tak zwany krok ETL (Extract, Transform, Load), realizujący wyodrębnianie, przekształcanie i ładowanie danych. Etap ten polega na wyodrębnieniu potrzebnych danych (często z różnych źródeł), usunięciu niepełnych, niepoprawnych i niemających znaczenia danych oraz załadowaniu ich do jednej hurtowni.

Wcześniej jednak warto takie dane poddać oczyszczaniu, zwłaszcza gdy są one wprowadzane przez użytkowników z klawiatury, a systemy nie sprawdzają ich poprawności lub też robią to w niewystarczającym stopniu. Najczęściej spotykanym przykładem odpowiadającym tej sytuacji są dane teleadresowe uzupełniane podczas rejestracji. Jeżeli pola typu miasto i województwo są wprowadzane ręcznie, nie poprzez wybór opcji z rozwijanych list, wówczas na pewno jedna i ta sama wartość będzie zapisana w różnych formatach jak „Łódź", „Lodz" i „Lódz". Znajdzie to swoje negatywne odzwierciedlenie podczas analizy danych. System nie rozpozna błędnie wprowadzonych informacji, co zaowocuje gorszymi wynikami.

Inne przykłady czyszczenia i transformacji to między innymi zmiana w przypadku algorytmów lepiej współpracujących ze zmiennymi liczbowymi danych typu bool na integer, agregacja szczegółowych danych w bardzie ogólne zbiory, uzupełnianie brakujących informacji według określonych zasad lub też usuwanie nietypowych wartości znacznie odbiegających od zestawu danych, które również mają zły wpływ na jakość obliczeń.

2.4 Wybór metody eksploracji danych

Wybór algorytmu, który zostanie wykorzystany jest najważniejszym etapem w procesie zgłębiania danych. Zanim podejmiemy decyzję, pierwszą czynnością powinno być zapoznanie się z zagadnieniem, którego ma dotyczyć tworzony model oraz określenie celów, jakie chcemy osiągnąć poprzez przeprowadzaną analizę. Takie jasne określenie oczekiwań pozwoli na trafną klasyfikację problemu i dobór najodpowiedniejszego algorytmu rozwiązującego zagadnienie. W

praktyce zazwyczaj wybiera się kilka pasujących metod i porównuje ich wyniki przy użyciu odpowiednich narzędzi – w SQL Server Analysis Services jest to Lift Chart oraz Classification Matix. W tym miejscu należy jednak dodać, iż narzędzie to nie obsługuje algorytmów Microsoft Association Rules oraz Microsoft Sequence Clustering.

3 Eksploracja danych

Eksploracja danych jest etapem, w którym odkrywane są wzorce, prawidłowości i zależności występujące w uprzednio przygotowanym zestawie danych.

3.1 Analiza i ocena wyników

Kolejnym krokiem jest oszacowanie, ocena wyników oraz poprawności odkrytych wzorców. W tym celu niezbędne są dane testowe, które nie brały udziału w trenowaniu modeli, co zapewni dokładniejszą ocenę uzyskanych rezultatów. Podział danych na treningowe i testowe zazwyczaj wykonywany jest w proporcjach 4:1.

Jak już wspomnino wcześniej do sprawdzenia poprawności można wykorzystać wbudowane narzędzia Lift Chart oraz Classification Matrix. Pierwsze z nich służy do tworzenia wykresów przyrostu, które na osi odciętych mają zaznaczony procent całej populacji, natomiast oś rzędnych odpowiada procentowi populacji konkretnego stanu. Wykresy te mogą przedstawić zarówno dokładność dla konkretnego atrybutu jak i ogólną ocenę modelu. Drugie z dostępnych narzędzi umożliwia porównywanie przewidywanych wartości z możliwymi stanami.

3.2 Predykcja

Należy zauważyć predykcja jest najciekawszym etapem procesu zgłębiania danych. Jej funkcjonalność oparta jest na przewidywaniu różnego typu zjawisk, głownie biznesowych i wykorzystywana jest podobnie jak następna faza czyli raportowanie do wspomagania zarządzania i planowania w przedsiębiorstwach. Prognozowanie to wykonywane jest na podstawie danych historycznych, które są gromadzona podczas działalności systemów.

3.3 Raportowanie

Proces raportowania jako wartość biznesowa jest uwieńczeniem cyklu odkrywania wiedzy z baz danych, który ma na celu pozyskanie przydatnych informacji, czyli odkrycie wzorców i przedstawienie ich w czytelnej formie, która może zostać wykorzystana na przykład do celów

marketingowych. Niestety mimo swoich wielu zalet rozszerzenie DMX nie umożliwia podstawowych własności charakterystycznych dla języków raportujących takich jak funkcje agregacji czy grupowania wyników. Dlatego tutaj z pomocą przychodzi nam usługa raportująca Microsoft Reporting Services dająca możliwości wyświetlania danych pochodzących z modeli analitycznych w postaci czytelnych raportów.

3.4 Integracja z aplikacją

Ostatnim etapem w cyklu pozyskiwania wiedzy jest zintegrowanie systemu Business Intelligence z aplikacją kliencką, dzięki czemu końcowy użytkownik bez wiedzy technicznej będzie korzystał z możliwości jakie oferuje to narzędzie. Integracja będzie umożliwiała na przykład segmentację klientów w systemach CRM, lepsze planowanie zasobów w systemach ERP, prowadzenie ukierunkowanych kampanii reklamowych, szacowanie ryzyka klienta ubiegającego się o kredyt i wiele innych.

Przedstawiony cykl odkrywania wiedzy zazwyczaj jest wykonywany wielokrotnie, a w aplikacjach z którymi jest zintegrowany powtarza się w określonych odstępach czasu. Wynika to głównie ze stale przyrastających danych, które jako aktualne informacje powinny być włączone do procesu trenowania modelu i raportowania jeżeli chcemy by odkryte wzorce i reguły odpowiadały możliwie najlepiej aktualnemu stanowi przedsiębiorstwa.

3.5 Podstawowe pojęcia oraz obiekty zgłębiania danych

Zanim zostaną omówione algorytmy oraz język DMX przybliżenia wymaga kilka podstawowych pojęć związanych z eksploracją danych, które są niezbędne do zrozumienia omawianego zagadnienia. Obiekty z jakimi spotykamy się podczas zgłębiania danych oraz jakie zostały wykorzystane przedstawianym w projekcie to: ***źródło danych***, ***widok źródła danych***, ***struktura*** oraz ***model***. Przydatna jest również znajomość takich pojęć jak ***trenowanie modelu*** i dane treningowe oraz ***testowanie modelu*** i dane testowe.

Z uwagi na to, że zgłębianie jest analizą danych zebranych w hurtowni, która jest niczym innym jak bazą danych możemy zdefiniować połączenie do niej. Obiekt przechowujący to połączenie nazywany jest ***źródłem danych*** (Data Source). Na podstawie tego połączenia tworzony jest ***widok źródła danych*** (Data Source View), który zawiera zestaw wybranych tabel i widoków oraz relacje pomiędzy nimi. Dodatkowo w widoku tym możemy zdefiniować własne pola obliczane, wykorzystując w tym celu język SQL. Należy dodać w tym miejscu, iż obiektów tych nie można utworzyć przy pomocy rozszerzenia DMX.

Tworzone są one bezpośrednio przez zastosowanie narzędzi Microsoft Visual Studio.

Kolejnym wymienionym obiektem jest ***struktura*** (Structure), którą można rozumieć jako pojemnik, przechowujący ***modele*** (Model). Struktura określa ilość i rodzaje kolumn jakie mogą występować w zawartych w niej modelach. Rodzaj kolumn definiowany jest przez dwie etykiety. Pierwsza z nich określa typ (tekstowy, boolowski, liczbowy, zmiennoprzecinkowy), druga zaś rodzaj (ciągły, dyskretny, klucz). Model z kolei jest ostatnim i najważniejszym obiektem zgłębiania danych. Zawiera on zestaw kolumn identyfikacyjnych, wejściowych i predykcyjnych. Podobnie jak w przypadku struktury definiowane są one przez dwie właściwości. Pierwszą jest typ (tekstowy, boolowski, liczbowy, zmiennoprzecinkowy), natomiast drugą typ danych (dyskretny, ciągły, klucz i kolumna predykcyjna). Warto zauważyć, że dopiero na tym poziomie określa się kolumny predykcyjne, czyli różne modele należące do tej samej struktury mogą przewidywać wartości dla różnych kolumn. Ostatnią z opcji jaka jest określana przy definicji modelu to rodzaj algorytmu wybranego do trenowania.

Pojęcie ***trenowania modelu*** lub inaczej procesowanie jest etapem obliczeniowym algorytmu określonego podczas budowania modelu eksploracyjnego. W tej fazie odkrywane są wzorce i zależności w danych treningowych, które zapamiętywane są w modelu. Proces ten przebiega z pewną dokładnością, która sprawdzana jest podczas ***testowania modelu***. Jak nietrudno się domyślić w tym celu wykorzystywany jest zbiór danych testowych, a samo sprawdzenie polega na wykonaniu zadania realizowanego przez dany model i porównaniu otrzymanych wyników z rzeczywistymi wartościami zawartymi w danych testujących. Należy jednak pamiętać, że zbiór danych testowych nie powinien zawierać się w zbiorze danych treningowych, ponieważ mogłoby to negatywnie wpłynąć na oszacowanie dokładności modelu – nieuzasadniony wzrost dokładności.

Opisane w tym podrozdziale obiekty i pojęcia są niezbędne dla lepszego zrozumienia procesu zgłębiania danych, korzystania z rozszerzenia DMX oraz działania aplikacji zaprezentowanej w dalszej części pracy.

4 Algorytmy zgłębiania danych

SQL Server Analysis Services jako rozbudowana platforma analityczna oferuje, w postaci zaimplementowanych algorytmów, solidne wsparcie matematyczne, umożliwiając tym samym rozwiązywanie różnego typu zagadnień. Wbudowane algorytmy, które możemy wykorzystać do analizy i predykcji danych opisane zostały poniżej.

4.1 Typy rozwiązywanych zadań

Zadania rozwiązywane przez algorytmy eksploracji danych możemy podzielić na siedem grup. Według tej klasyfikacji wyróżniamy następujące elementy:

- Asocjacja

Asocjacja jest jednym z najczęściej wykorzystywanych zadań w procesach zgłębiania danych. Polega ona na odnajdywaniu ukrytych prawidłowości w zbiorach danych oraz na przewidywaniu obecności danego atrybutu na podstawie obecności innego. Umożliwia ona znalezienie wartości często występujących obok siebie np. analiza transakcji sprzedażowych wyłoni produkty, które klienci kupują razem. Możliwe jest również wyłonienie reguł asocjacyjnych, które towarzyszą danemu zjawisku np. określenie wszystkich reguł występujących równocześnie ze wzrostem sprzedaży zadanego produktu. Taka analiza koszyka jest najpopularniejszym celem asocjacji, jednak można ją wykorzystywać również w ekonomii (przewidywanie wpływów), marketingu (projektowanie ofert reklamowych) czy medycynie (asocjacja genetyczna).

- Grupowanie

Grupowanie lub segmentacja jest to proces wykrywania naturalnych podziałów w zbiorach danych na grupy/kategorie. Podział ten dokonywany jest ze względu na podobieństwo cech obiektów reprezentujących daną grupę. Za przykład może nam posłużyć segmentacja klientów, która wyłoni nam ich profile. Profile te mogą być utworzone na podstawie zadanych kryteriów tj. płeć, wiek czy zainteresowania, co z kolei umożliwi nam stworzenie ukierunkowanych na grupy kampanii reklamowych lub ofert promocyjnych.

- Klasyfikacja

Kolejnym bardzo często wykorzystywanym algorytmem w analizie danych jest klasyfikacja. Jej zadaniem jest przydzielenie rozpatrywanych przypadków do zadanych kategorii, które określa się poprzez wskazanie kolumn predykcyjnych. Podział ten odbywa się na podstawie zadanych atrybutów wejściowych, a dokładniej na ich możliwych wartościach. W tym wypadku przykładem może być analiza odejścia klienta na podstawie historii transakcji, ocena ryzyka kredytowego lub też dowolny problem biznesowy którego rozwiązanie polega na podjęciu decyzji zależnej od wielu czynników.

- Regresja

Zadania regresji są bardzo podobne do zadań rozwiązywanych w procesie klasyfikacji, z tą różnicą iż wspierają one predykcję wartości

ciągłych tj. dochód, prędkość czy wiek. W ogólnym rozumieniu metody regresji są po prostu parametrycznymi metodami aproksymacji funkcji, różnią się one jedynie doborem metod aktualizujących wartości tych parametrów, w ten sposób aby sparametryzowana formuła matematyczna jak najdokładniej odzwierciedlała dane, do których została dopasowana.

- Prognozowanie

Prognozowanie, inaczej predykcja jest kolejnym ważnym zadaniem eksploracji danych. Zadanie to polega na przewidywaniu różnych wartości na podstawie danych historycznych oraz określeniu ich prawdopodobieństwa. Bazuje na wykrytych prawidłowościach i trendach, które dla człowieka mogą być niezauważalne, zwłaszcza gdy mamy do przeanalizowania - w najlepszym wypadku - tysiące rekordów. Przewidywać możemy różne wartości, a głównym ograniczeniem jest tutaj postawienie pytania biznesowego. Przykładem może być sprzedaż w następnym miesiącu, stany magazynowe, wzrost liczby klientów w przeciągu następnych kilku miesięcy lub pogoda.

- Analiza sekwencyjna

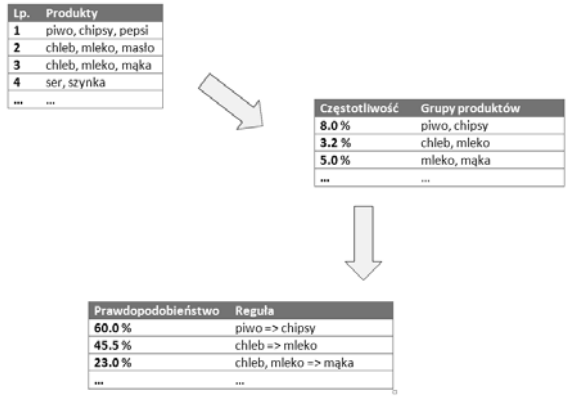
Analiza sekwencyjna, jak sama nazwa wskazuje, zajmuje się analizą dyskretnych sekwencji. Podobnie jak zadania asocjacji, polega ona na odnajdywaniu ukrytych prawidłowości w zbiorach danych, z tą różnicą, że każdy element zależy od poprzedniego, czyli ważna jest kolejność. W asocjacji interesują nas produkty występujące razem, natomiast w analizie sekwencyjnej najważniejsza byłaby kolejność w której trafiałyby one do koszyka – kolejność ta decydowałaby o wyborze następnych produktów. Analizę tą można wykorzystać między innymi do badania sekwencji kupowanych produktów jak również do badania kolejności kliknięć na stronie czy w medycynie do badania łańcuchów DNA.

- Analiza odchyleń

Ostatnia grupa zadań rozwiązywanych przez algorytmy eksploracji danych w SQL Server Analysis Services to analiza odchyleń. Zadania te polegają na odnajdywaniu w zbiorach danych przypadków unikatowych – odbiegających od innych. Główne zastosowanie to miedzy innymi wykrywanie oszustw w firmach ubezpieczeniowych lub bankach, gdzie rozpatrywane są miliony podań, a człowiek czytający nawet uważnie olbrzymie ilości dokumentów nie jest w stanie zauważyć występujących anomalii.

4.2 Microsoft Association Rules

Microsoft Association Rules jest algorytmem asocjacyjnym, przeprowadzającym tak zwaną analizę koszykową, co zostało omówione w poprzednim rozdziale. Składa się on z dwóch podstawowych kroków [8]. Pierwszy polega na obliczeniu częstotliwości występowania produktów, zarówno pojedynczych jak i zestawów. W następnym kroku tworzone są reguły asocjacyjne. Tworzenie tych reguł oparte jest na wyliczeniach z pierwszej fazy i jest najbardziej czasochłonnym etapem. Ideę działania tego procesu przedstawia rysunek 1.



Rys. 1. Microsoft Association Rules - Przebieg Procesu

Częstotliwość F_z dla zestawu produktów: piwo (W), chipsy (C) jest wyrażona jako:

$$F_z(\{W,C\}) = Ilośl\ transakcji(W,C)\ ,$$

natomiast prawdopodobieństwo P tej grupy produktów można opisać następującą zależnością:

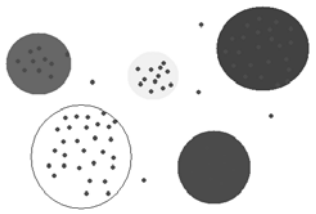
$$P(W \Rightarrow C) = P(C|W)\,F_z(W,C)/F_z(A)\ ,$$

gdzie P(C|W) - jest prawdopodobieństwem warunkowym zajścia zdarzenia C jeśli zajdzie zdarzenie W.

Dodatkowo należy zauważyć, że po lewej stronie reguły asocjacyjnej (znak „⇒”) znajdują się zarówno atrybuty wejściowe jak i wyjściowe modelu, co zależy od rodzajów kolumn predykcyjnych, natomiast po prawej stronie znajdują się tylko atrybuty predykcyjne. Rodzaje i typy kolumn w obiektach data mining będą omówione szczegółowo w dalszej części artykułu.

4.3 Microsoft Clustering

Microsoft Clustering jest algorytmem realizującym klasyfikację oraz segmentację danych. Jego główne zadanie polega na odkrywaniu związków, pomiędzy danymi, które można pominąć w trakcie zwykłej obserwacji. Rysunek 2 przedstawia zbiór danych podzielony na 5 segmentów, gdzie każdy z nich zawiera podzbiór danych ściśle ze sobą związanych według rozpatrywanych kryteriów:



Rys. 2. Microsoft Clustering

Zadanie to można realizować w dwojaki sposób, w zależności od wybranej metody grupowania. Do wybory mamy:

1.1. Algorytm k-means (k-średnich)

Ideą algorytmu k-means jest klasyfikacja przeprowadzana ze względu na odległości występujące pomiędzy poszczególnymi przypadkami. Odległości te są zwykle odległościami euklidesowymi, co oznacza, że algorytm ten może być wykorzystywany tylko w grupowaniu atrybutów ciągłych.

Klasyfikacja k-means polega na losowym wyborze k-punktów ze zbioru, którego elementy mają być grupowane. Wylosowane punkty, za pomocą wektora wartości atrybutów opisujących klasyfikowane obiekty, wyznaczają klastry, czyli kategorie dzielące rozpatrywany zbiór danych [2]. Klastry te definiowane są przez centra lub wartości centralna, a opisywane są przez wektory kategorii, które również określane są jako centra lub wektory centralne kategorii (1):

$$u^d = \left\langle u_1^d, u_2^d, u_3^d, \ldots, u_n^d \right\rangle \in A_1 \times A_2 \times A_3 \times \ldots \times A_n \quad (1)$$

gdzie d – kategoria(grupa), u_1^d – wartość atrybutu u_1 dla kategorii d.

W kolejnej fazie działania algorytmu, dla każdego przykładu $x \in X$ można obliczyć odległość, według metryki euklidesowej, pomiędzy wektorem wartości jego atrybutów (2)

$$x = \left\langle a_1(x), a_2(x), a_3(x), \ldots, a_n(x) \right\rangle \quad (2)$$

a wektorem u^d reprezentującym kategorię d. Odległość ta jest opisana za pomocą wzoru (3):

$$\delta(x,d) = \sqrt{\sum_{i=1}^{n} (a_i(x) - u_i^d)^2} \tag{3}$$

Przypisanie rozpatrywanego przykładu do klastra odbywa się poprzez znalezienie kategorii do której pasuje on najlepiej, czyli poprzez znalezienie takiej kategorii, której centrum leży najbliżej według metryki euklidesowej.

Po rozpatrzeniu wszystkich obiektów będących przedmiotem klasyfikacji cały powyższy proces jest powtarzany. Ponownie wyznacza się wektory centralne, jednak tym razem są one wyliczane jako mediany lub średnie arytmetyczne grup z poprzedniej iteracji. Kryterium zakończenia tego algorytmu jest niezmienność centrów, które charakteryzują się minimalną wariancją wewnątrz klastrów, a maksymalną na zewnątrz [2]. Co można zapisać w następujący sposób (4):

$$h(x) = \arg\ \min \delta(x,d) \tag{4}$$

,

gdzie $h \equiv \left\langle u^{d_1}, u^{d_2}, u^{d_3}, \ldots, u^{d_m} \right\rangle$ jest zbiorem wektorów wartości atrybutów u^{d_i} dla każdej kategorii d.

Algorytm k-means jest obarczony kilkoma wadami wynikającymi między innymi z braku jednoznacznej metody wyboru początkowych centrów. W zależności od ich wyboru otrzymane wyniki będą się różniły od siebie, dlatego wykorzystując ten algorytm zaleca się utworzenie jednocześnie kilku oddzielnych modeli i porównanie ich stopnia dokładności. Kolejna wada to możliwość zakończenia w lokalnym minimum, a także mała odporność na szumy. Dodatkowo ważnym elementem, wpływającym na jakość podziału danych na grupy jest decyzja dotycząca ilości tworzonych segmentów oraz wyboru zmiennych klasyfikujących, czyli atrybutów według których algorytm buduje grupy i przydziela do nich poszczególne przypadki. Natomiast na korzyść algorytmu przemawia łatwość implementacji oraz nieduża złożoność obliczeniowa.

1.2. Algorytm EM (Expectation Maximization)

Probabilistyczny algorytm EM (Expectation/Estimation Maximization) jest zmodyfikowanym algorytmem k-means. Jego inność polega na tym, iż w swych obliczeniach wykorzystuje on informacje o rozkładzie prawdopodobieństwa, bierze pod uwagę rozkład Gaussa dla każdej kategorii, a także średnie i standardowe odchylenie. Model ten zakłada, że przykłady trenujące (wektory liczbowych wartości atrybutów) zostały

wygenerowane za pomocą pewnej ustalonej liczby prostych rozkładów prawdopodobieństwa o znanej postaci, lecz nieznanych parametrach. Grupowanie polega wówczas na wyznaczeniu najbardziej prawdopodobnych rozkładów opisujących dane testowe i przypisaniu do nich poszczególnych przykładów. Postępowanie to można zawrzeć w dwóch krokach:

1. Estymacji - oszacowanie prawdopodobieństwa przynależności danych testowych $x \in T$ do poszczególnych grup, które podobnie jak w algorytmie k-means inicjalizowane są losowo, następnie wyliczane są ponownie w każdej iteracji, a ich wartość zależy od wartości poprzedniej.
2. Maksymalizacji prawdopodobieństwa – wyliczenie nowych wartości wektorów – parametrów rozkładów prawdopodobieństwa na podstawie przypisanych do nich przykładów.

Powyższe kroki powtarzane są tak długo, aż przestanie być widoczna różnica pomiędzy nowymi wartościami wektorów, a wartościami z poprzedniej iteracji.

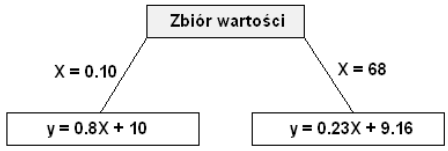
4.4 Microsoft Decision Trees

Microsoft Decision Trees jest najczęściej wykorzystywanym algorytmem klasyfikującym i regresyjnym. Zgodnie z nazwą głównie rozwiązuje on tak zwane problemy decyzyjne z wieloma rozgałęziającymi się wariantami. Mimo to jest on łatwy do wizualizacji i stosunkowo czytelny dla człowieka.

Definicja drzewa decyzyjnego mówi, iż jest to acykliczny i spójny graf, którego wierzchołkami są atrybuty opisujące obiekt, zaś wyprowadzone z niego gałęzie to wartości, które te atrybuty mogą przyjmować. Najniższy poziom, czyli liście zbudowanego drzewa tworzą klasy lub inaczej decyzje końcowe, które klasyfikują obiekty. Poniższy przykład przedstawia drzewo ułatwiające podjęcie decyzji o udzieleniu kredytu - rysunek 3. Przypisuje on ubiegających się klientów do dwóch grup: „wiarygodny” i „niewiarygodny” na podstawie zarobków oraz bieżących zadłużeń.

Rys. 3. Microsoft Decision Trees, predykcja kolumn dyskretnych

Poprzedni rysunek stanowi przykład drzewa gdzie przewidywane wartości są typu dyskretnego. Microsoft Decision Trees pozwala także na predykcję wartości ciągłych, wówczas każdy węzeł zawiera formułę regresyjną, czyli formułę matematyczną tworzącą obszary decyzyjne – rysunek 4.



Rys. 4. Microsoft Decision Trees, predykcja kolumn ciągłych

Istnieją różne metody konstruowania drzew decyzyjnych. Microsoft Decision Trees jest połączeniem dwóch algorytmów ID3, który w późniejszym czasie został ulepszony pod kątem odporności na szumy i brakujące dane – C4.5 oraz CART (Classification and Regression Tree).

Algorytmy budujące drzewa decyzyjne są algorytmami iteracyjnymi. Ogólna zasada ich działania polega na wyborze, w każdym kroku, atrybutu - węzła rozgałęziającego drzewo. Wybór ten nie jest przypadkowy, gdyż w każdej iteracji wybierany jest atrybut mający największy wpływ na predykcyjną wartość. Dodatkowo ma to ogromny wpływ na rozmiar i kształt budowanego drzewa, dlatego najlepszym kandydatem jest atrybut, który powoduje skrócenie ścieżki (w stosunku do innych możliwych atrybutów) prowadzącej od korzenia do liścia, nie powodując przy tym zbytniej rozległości drzewa. Microsoft Decision Trees oferuje nam kilka metod wyboru, a są to: wykorzystujące entropię Shanona, metoda Bayes'a z predykcją K2 lub z predykcją Dirchleta. W końcowym efekcie powstałe ścieżki prowadzące od korzenia do liścia reprezentują reguły dla przewidywanego zjawiska.

Teraz omówię wspomniane już zjawisko entropii, natomiast jeśli chodzi o metody Bayes'a, zostały one omówione w rozdziale traktującym o algorytmie Microsoft Naive Bayes.

Entropia jako kryterium wyboru atrybutu rozgałęziającego drzewo umożliwia ocenę średniej ilości informacji dostarczanej przez ten właśnie atrybut. Na tej podstawie w każdym kroku, dla nowo tworzonego węzła wybierany jest najbardziej znaczący atrybut, który optymalizuje rozmiar drzewa minimalizując przy tym negatywny wpływ na dokładność modelu.

Entropia informacji wyrażana jest wzorem [2] (5)

$$I(P) = -\sum_{d \in C} p^d \log(p^d) \tag{5}$$

gdzie p^d to prawdopodobieństwo zakwalifikowania obiektu do kategorii d, d – pojedyncza kategoria, C – zbiór kategorii.

Tak określona wartość informacji jest duża wtedy, kiedy liczba przykładów poszczególnych kategorii w zbiorze P jest zbliżona, natomiast maleje wraz ze wzrostem zróżnicowania rozkładu. Dodatkowo jest ona równa zero, gdy prawdopodobieństwa przynależności obiektu do każdej z grup są równe 1 lub równe 0.

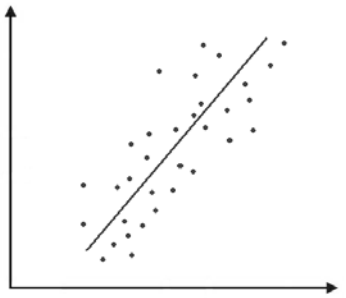
Przed wykorzystaniem omawianego algorytmu, chcąc uzyskać jak najlepsze wyniki w akceptowalnym czasie, należy zwrócić uwagę na kilka ważnych szczegółów. Jeżeli któryś z atrybutów wejściowych jest atrybutem dyskretnym przyjmującym dużo wartości (racjonalna liczba nie przekracza 100), wówczas powinno się zredukować nadmierną ilość stanów. Microsoft Decision Trees daje możliwość wykorzystania wbudowanego, dynamicznego grupowania, które ogranicza stany do 100, gdzie 99 z nich to te najbardziej popularne, natomiast 1 jest przeznaczony dla stanów pozostałych. Kolejnym punktem do rozważenia jest liczba atrybutów wejściowych, którą należy ograniczyć do tych naprawdę istotnych z punktu widzenia rozważanego problemu, co pozwoli zyskać na czasie przetwarzania i odciążyć jednostkę obliczeniową. Wreszcie trzeba zająć się również atrybutami wejściowymi typu ciągłego. Na szczęście, również w tym przypadku z pomocą przychodzi nam wbudowany mechanizm. Microsoft Decision Trees dzieli wartości tego atrybutu na tak zwane kosze (domyślnie 99), pomiędzy którymi zachodzą porządkowe relacje mniejszości i większości, mając na celu uzyskanie optymalnej liczby koszy o jak najwyraźniejszym rozdzieleniu danych wejściowych.

4.5 Microsoft Linear Regresssion

Microsoft Linear Regression jest odmianą algorytmu Microsoft Decision Trees ukierunkowaną na badanie liniowej zależności pomiędzy atrybutami ciągłymi. Może być on wykorzystany nie tylko do regresji liniowej, ale także do klasyfikacji i asocjacji.

Jak już to zostało zaznaczone wcześniej, w ogólnie regresja jest to dopasowanie parametrów funkcji aproksymującej dane, w ten sposób aby odzwierciedlała ona możliwie jak najdokładniej zbiór danych, do którego została dopasowana. Jej zadaniem jest wyznaczenie liniowej zależności pomiędzy zmiennymi. Do tego zadania możliwe jest zastosowanie algorytmu Microsoft Linear Regression. Wyznacza on relacje liniowe występujące pomiędzy atrybutami wejściowymi, a wiedzę tę używa w procesie predykcji.

Przykład regresji liniowej, przeprowadzonej testowego dla zbioru punktów przedstawia rysunek 5:



Rys. 5. Rysunek 5 - Microsoft Linear Regression

Równanie regresji liniowej określone jest wzorem (6):

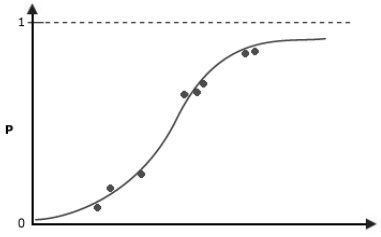
$$f(x) = ax + b \tag{6}$$

gdzie a i b są parametrami, przybliżającymi równanie do zbioru danych, z reguły wyznaczane metodą najmniejszych kwadratów. Wartość funkcji f(x) jest zmienną wyjściową, natomiast x jest zmienną wejściową.

4.6 Microsoft Logistic Regression

Pojęcie regresji logistycznej nie odbiega znacznie od pojęcia regresji liniowej. Idea jest taka sama, czyli polega na zbadaniu zależności pomiędzy atrybutami, a pewną zmienną zależną, która w przypadku regresji logistycznej znajduje się na skali dychotomicznej, co oznacza, że przyjmuje tylko dwie wartości (na przykład tak i nie).

Algorytm Microsoft Logistic Regression jest zmodyfikowanym algorytmem Microsoft Neural Network, który w połączeniu z regresją logistyczną rozwiązuje problem jaki pojawia się przy regresji liniowej, mianowicie pozwala na badanie wpływu atrybutów na zmienne przyjmujące wartości z zakresu <0, 1>. Funkcjonalność ta została osiągnięta poprzez zastąpienie (w stosunku do regresji liniowej) prostej, nieliniową funkcją, której przykładowy przebieg przedstawia rysunek 6.



Rys. 6. Microsoft Logistic Regression

Na osi odciętych znajdują się wartości kolumn wejściowych, natomiast na osi rzędnych zaznaczone jest prawdopodobieństwo przyjęcia danego stanu przez zmienną.

4.7 Microsoft Naive Bayes

Algorytm Microsoft Naive Bayes jest algorytmem klasyfikującym opartym na twierdzeniu o prawdopodobieństwie warunkowym wg teorii Bayes'a. Oblicza on warunkowe prawdopodobieństwo pomiędzy parametrami wejściowymi, a przewidywaną wartością, dodatkowo zakładając „naiwnie", iż parametry wejściowe są wzajemnie niezależne. Dlatego korzystając z tego algorytmu dobrze jest zadbać o to, aby pomiędzy atrybutami wejściowymi była jak najmniejsza zależność. Twierdzenie Bayes'a opisane jest wzorem:

$$P(H|X) = \frac{P(H) \cdot P(X|H)}{P(X)} \tag{7}$$

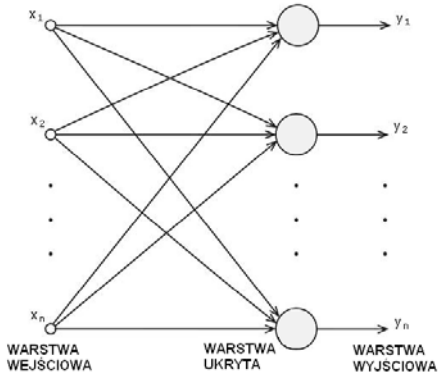
przy założeniach, że X jest daną właściwością rozpatrywanego obiektu, natomiast H jest hipotezą zakładającą, że obiekt ten o właściwości X należy do danej klasy C. Powyższe twierdzenie pozwala wyznaczyć prawdopodobieństwo warunkowe P(H|X), że dla obiektów o właściwości X prawdziwa jest hipoteza o przynależności do grupy C. Pozostałe wielkości użyte we wzorze P(X) oraz P(X|H) oznaczają odpowiednio bezwarunkowe prawdopodobieństwa (a priori), że dla każdego obiektu prawdziwa jest hipoteza H (należy on do klasy C), dalej że dowolnie wybrany obiekt posiada właściwość X oraz warunkowe prawdopodobieństwo P(X|H) mówiące, że jeżeli obiekt należy do klasy C, to posiada on właściwość X. Opisując sytuację bardziej obrazowo przytoczę przykład ukierunkowany na konkretne zdarzenia. Właściwością X będzie fakt, że „klient interesuje się nowymi technologiami", natomiast hipoteza H zakłada, że „klient będzie zainteresowany kupnem telefonu najnowszej generacji". Dla tego konkretnego przypadku, twierdzenie Bayes'a pozwoli określić prawdopodobieństwo, że klient będzie zainteresowany kupnem nowego telefonu, jeśli jego zainteresowania obejmują nowinki technologiczne. Wynik ten otrzymamy wykorzystując dane o liczbie klientów kupujących nowe modele telefonów (pozwoli nam to obliczyć P(H)), o liczbie klientów interesujących się nowymi technologiami (P(X)) oraz liczbie klientów którzy kupili nowe telefony wśród klientów, których hobby to technologia (P(X|H)).
Microsoft Naive Bayes może być stosowany zarówno w przypadku klasyfikacji dwu- jak i wielo-wartościowej, udzielając odpowiedzi na pytania:

- Czy klient weźmie kredyt w banku?
- Czy klient jest wiarygodny? – ocena ryzyka kredytowego
- Czy klient zostanie naszym stałym kontrahentem, czy może przejdzie do konkurencji?
- Czy klient będzie zainteresowany danym produktem?
- Klasyfikacja grupy do której należy klient na przykład (rozwijającej się, stabilnej, nieznaczącej, upadającej).

Dodatkowo przy użyciu wbudowanego narzędzia eksplorującego modele w SQL Analysis Services 2005 w łatwy sposób można zaobserwować różnice pomiędzy klientem, który korzysta z propozycji kredytowej, a tym co rezygnuje, czy pomiędzy klientem zainteresowanym kupnem danego produktu, a klientem obojętnym na przedstawianą ofertę.

4.8 Microsoft Neural Network

Microsoft Neural Network jest algorytmem opartym na wielowarstwowych sieciach neuronowych zwanych perceptronami. Wielowarstwowość w przypadku tego rozwiązania jest ograniczona do trzech warstw, warstwy wejściowej, opcjonalnej warstwy ukrytej (pośredniczącej pomiędzy warstwami wejścia i wyjścia) oraz warstwy wyjściowej. Sieć ta może rozwiązywać zarówno zadania klasyfikacji jak i regresji, wyznaczać nieliniowe zależności pomiędzy atrybutami wejściowymi i predykcyjnymi, jednak zadania te są rozwiązywane w dłuższym czasie niż z wykorzystaniem algorytmów Microsoft Decision Trees lub Microsoft Naive Bayes. Dodatkowo Microsoft Neural Network wspiera dyskretny oraz ciągły typ kolumn wyjściowych zapewniając sprzężenie wyprzedzające.

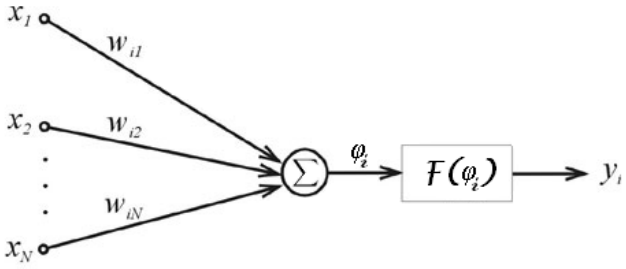


Rys. 7. Microsoft Neural Network - Sień Neuronowa

System ten jest uproszczonym modelem biologicznego układu nerwowego. Składa się on z połączonych między sobą w sieć neuronów, które odpowiadają poszczególnym atrybutom. Połączenia te są realizowane za pomocą krawędzi obarczonych pewnymi wagami, których kierunek oznacza porządek przepływu danych. Głównym atutem sieci neuronowych jest zdolność uczenia się. Proces ten przebiega na zasadzie dostosowywania parametrów charakteryzujących poszczególne neurony, mając na uwadze wzrost efektywności modelu.

Idea działania tego algorytmu jest stosunkowo prosta. Atrybuty wejściowe są normalizowane i mapowane do postaci neuronów w warstwie wejściowej. W kolejnej fazie, czyli w warstwie ukrytej dokonywany jest proces obliczeniowy i uaktywniane są neurony warstwy wyjściowej, generujące ostateczne rozwiązania, które są mapowane powtórnie, tym razem w celu zwrócenia oryginalnych wartości. Ostatni etap to obliczenie błędu dla każdego wyjścia oraz przypisanie nowych wag do modelu. Najbardziej czasochłonną fazą jest oczywiście faza obliczeniowa, która polega na najlepszym doborze wag modelu, które w pierwszej iteracji, podobnie jak ma to miejsce w większości omawianych algorytmów, są przypisywane losowo.

Omówmy teraz najbardziej elementarną strukturę tego algorytmu, czyli neuronu, który jest podstawową jednostką budulcową sieci. Każdy neuron jest jednostką obliczeniową o następującej budowie:



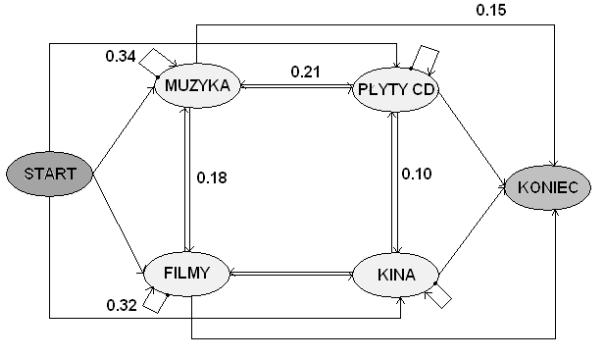
Rys. 8. Microsoft Neural Network - Budowa Neuronu

Wprowadzone sygnały wejściowe x_i mnożone są przez odpowiadające im wagi w_{ji}, które decydują o ich znaczeniu dla rozwiązywanego problemu. Otrzymane iloczyny są sumowane i stanowią argument φ_i dla funkcji aktywacji *F*, która jest ważnym elementem w budowie neuronów. Od tej funkcji zależy wynik podany na wyjściu y_i. Może ona być w postaci liniowej, progowej, sigmoidalnej czy tangensa hiperbolicznego.

4.9 Microsoft Sequence Clustering

Microsoft Sequence Clustering jak sama nazwa wskazuje jest algorytmem łączącym w sobie technikę zwykłej segmentacji oraz sekwencji Markowa. Jest on przeznaczony do analizy danych, które można przedstawić w postaci sekwencyjnej, za przykład może posłużyć sprzedaż produktów, nawigacja po portalach, gdzie sekwencją są wybrane linki lub dane medyczne takie jak łańcuchy DNA.

Segmentacja została opisana dokładnie w rozdziale dotyczącym algorytmu Microsoft Clustering, dlatego teraz przybliżone zostanie pojęcie łańcuchów Markowa. Struktury te w najprostszej postaci są sekwencjami losowych wartości, w których każdy kolejny element zależy od poprzedniego, ale nie jest zależny od procesu z którego poprzedni element wynika. Sytuację tą ilustruje rysunek 9, który przedstawia przykładową sekwencję ścieżki dla strony Web.



Rys. 9. Microsoft Sequence Clustering - Łańcuch Markowa

Liczby zmiennoprzecinkowe zaznaczone przy krawędziach grafu oznaczają prawdopodobieństwo przejścia z jednego stanu w inny. Na przykład zapis $P(X_i = ″PŁYTY\ CD″ | X_{i-1} = ″MUZYKA″) = 0.21$ czytamy: prawdopodobieństwo, że internauta oglądający zawartość strony „MUZYKA” przejdzie do podstrony „PŁYTY CD” jest równe 0.21.

W bardziej złożonej postaci łańcuchów Markowa do predykcji prawdopodobieństwa dalszego elementu wykorzystuje się więcej niż jeden stan poprzedni i znaczącą rolę odgrywa tu kolejność tych stanów, co w ogólnej postaci możemy opisać zgodnie z (8):

$$P(X) = P(X_n | X_{n-1}, X_{n-2}, X_{n-3}, \ldots, X_0) \cdot P(X_{n-1} | X_{n-2}, X_{n-3}, \ldots, X_0) \cdot P(X_{n-2} | X_{n-3}, \ldots, X_0) \cdot \ldots \cdot P(X_0) \cdot \quad (8)$$

gdzie n + 1 jest długością sekwencji, a model ten nazywamy modelem Markowa rzędu n+1. Rząd modelu mówi nam ile poprzednich stanów jest wykorzystywanych do wyznaczania prawdopodobieństwa następnego.

Wyznaczone prawdopodobieństwa możliwych stanów sekwencji przechowywane są w tak zwanej macierzy transakcji, której rozmiar rośnie wraz ze wzrostem długości sekwencji, co powoduje również wzrost czasu przetwarzania modelu.

Jak już zaznaczono poprzednio Microsoft Sequence Clustering jest kombinacją dwóch technik: segmentacji oraz łańcuchów Markowa. Hybryda ta jest oparta na omówionym algorytmie probabilistycznym EM, dodatkowo jednak dla każdego klastra wyznaczany jest zestaw ścieżek oraz macierz stanów i ich prawdopodobieństw.

4.10 Microsoft Time Series

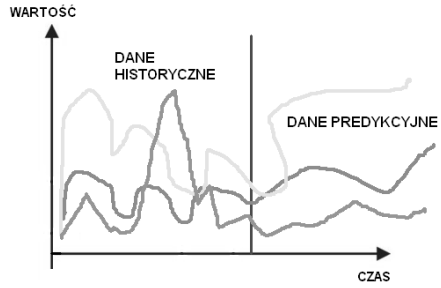
Algorytm Microsoft Time Series jest algorytmem regresyjnym umożliwiającym analizę szeregów czasowych oraz predykcję kolumn typu ciągłego. Skoro szereg czasowy, to znajduje zastosowanie we wszystkich problemach, w których obserwacja jest dokonywana w czasie. Wykorzystywany jest na przykład do przewidywania popytu i sprzedaży, co w konsekwencji pozwala na planowanie dostaw, tak aby nie zabrakło produktów o najlepszej sprzedaży, a równocześnie, aby produkty nie budzące zainteresowania nie zajmowały powierzchni magazynowych. Można go również zastosować do przewidywania kursów walut i notowań giełdowych.

Microsoft Time Series bazuje na analizie trendów występujących w wejściowym zestawie danych (szeregu czasowym), gdzie tak zwana zmienna objaśniająca (czynnik mający wpływ na przebieg danego zjawiska) została zastąpiona zmienną czasową. Należy zauważyć jednak, że algorytm ten jest nowatorskim spojrzeniem na analizę szeregów czasowych, łączącym w sobie techniki autoregresji oraz drzew decyzyjnych, co powoduje, że często jest on nazywany również autoregresyjnym drzewem ART (AutoRegression Tree).

Technika drzew decyzyjnych została już omówiona w poprzedniej części przy okazji algorytmu Microsoft Decision Trees, dlatego teraz omówione zostanie pojęcie autoregresji w szeregach czasowych. Autoregresja definiowana jest jako statystyczna metoda przewidywania kolejnych wartości szeregu czasowego (zmiennych objaśnianych) na podstawie wartości z poprzednich chwil czasowych, czyli wartości historycznych, co można opisać zależnością 9

$$X_t = f(X_{t-1}, X_{t-2}, X_{t-3}, \ldots, X_{t-n}) + \varepsilon_t \tag{9}$$

gdzie X_t – aktualnie rozpatrywana wartość szeregu w chwili t, n – liczba wyrazów szeregu, ε – błąd modelu.



Rys. 10. Microsoft Time Series - Szereg czasowy

Zaletą omawianego algorytmu jest również umiejętność wyznaczania, na podstawie wejściowego zbioru danych, zależności pomiędzy poszczególnymi atrybutami. Ową zależność można przedstawić w postaci funkcji określonej wzorem (10):

$$X_t = a_1 X_{t-1} + a_2 X_{t-2} + a3_1 X_{t-3} + \ldots + a_n X_{t-n}) + \varepsilon_t \qquad (10)$$

gdzie a_i – współczynniki autoregresji.

Wspomniane już autoregresyjne drzewo ART jest modelem zgłębiającym dane, w którym granice wyznaczane są przez drzewo decyzyjne, a liście tego drzewa zawierają autoregresyjny liniowy model funkcji. Jednak możliwości tego algorytmu są dużo większe, a mianowicie pozwalają na modelowanie nieliniowych zależności w szeregu czasowym, a także mogą analizować szeregi czasowe charakteryzujące się sezonowością. Zjawisko to jest bardzo powszechne dla danych pochodzenia biznesowego, za przykład może posłużyć wzrost sprzedaży w grudniu oraz jej spadek w styczniu, gdy portfele klientów są puste po bożonarodzeniowej gorączce zakupów. Sezonowość ta wykrywana jest automatycznie podczas tworzenia model. W tym celu wykorzystywana jest szybka transformacja Fouriera analizująca częstotliwości.

5 Rozszerzenie DMX

DMX (Data Mining Extension) jest to rozszerzenie analizy danych języka SQL oferujące bardzo bogatą funkcjonalność w zakresie analizy biznesowej i nie tylko. Umożliwia ono tworzenie i odpytywanie struktur oraz modeli eksploracji danych w Microsoft SQL Server Analysis Services (SSAS). Język ten podobnie jak klasycznego SQL-a można podzielić ze względu na rodzaj wykonywanych operacji. Podział ten wyłania nam dwie grupy, a mianowicie rozróżniamy zapytania DDL (Data Definition Language) definiujące obiekty oraz DML (Data

Manipulate Language) manipulujące danymi. Dodatkowo dostępne są również operatory i predefiniowane funkcje ułatwiające analizę budowanych struktur. Warto nadmienić także, iż rozszerzenie to jest zintegrowane z interfejsem OLE DB dla Data Mining, które opiera się na dobrze znanych zasadach dostępu do danych.

5.1 Tworzenie obiektów analitycznych z wykorzystaniem DMX

Modele analityczne można tworzyć według dwóch różnych sposobów. Pierwszy polega na utworzeniu struktury i dodaniu do niej modeli, drugi natomiast umożliwia utworzenie obu tych obiektów w jednym kroku, przy czym tak powstała struktura posiada wówczas identyczną budowę jak model, a nazwa jej jest tworzona według zasady: [nazwa modelu]_Structure. Kolejny etap obejmuje trenowanie utworzonego obiektu analitycznego i dopiero po tej fazie model może być odpytywany z wzorców i zależności, które zostały znalezione [3] [5].

Wszystkie wymienione powyżej etapy mogą być zrealizowane za pomocą rozszerzenia DMX, co jest tematem przewodnim tego rozdziału. Sposób realizacji każdego z nich jest przedstawiony w postaci ogólnej popartej konkretnym przykładem.

5.2 Tworzenie struktury eksploracyjnej

Kod tworzący strukturę analityczną jest bardzo podobny pod względem składniowym do kodu SQL tworzącego tabele. W nagłówku podajemy nazwę tworzonego obiektu, natomiast w nawiasach podawane są kolumny wraz z ich typem oraz rodzajem.

```
CREATE MINING STRUCTURE DMX_Structure
(
  [Gender] TEXT DISCRETE,
  [IncomeGroup] TEXT DISCRETE,
  [NumberCarsOwned] LONG DISCRETE,
  [OrderNumber] TEXT KEY,
  [Region] TEXT DISCRETE,
  [vTrainingAssocSeqLineItems] TABLE
  (
    [Category] TEXT DISCRETE,
    [LineNumber] LONG KEY SEQUENCE,
    [Model] TEXT DISCRETE
  )
)
```

Oczywiście skoro możemy tworzyć struktury, możemy je również usuwać przy pomocy polecenia DROP.

```
DROP MINING STRUCTURE DMX_Structure
```

5.3 Tworzenie modelu eksploracyjnego

Jak już zaznaczono wcześniej istnieją dwie drogi tworzenia modelu eksploracyjnego. Pierwsza z nich polega na modyfikacji istniejącej struktury poprzez dodanie do niej modelu, co przedstawia poniższy przykład. W definicji budowanego modelu podajemy dowolne kolumny zawarte w strukturze, określamy, które z nich będą predykcyjne, jaki algorytm ma być zastosowany w fazie treningowej wraz z jego parametrami, a także czy ma być dostępna opcja *drillthrough*.

```
ALTER MINING STRUCTURE DMX_Structure
ADD MINING MODEL DMX_Model
(
  [NumberCarsOwned],
  [OrderNumber],
  [Region] PREDICT,
  [vTrainingAssocSeqLineItems]
  (
    [Category]PREDICT,
    [LineNumber]
  )
)
USING   Microsoft_Sequence_Clustering(CLUSTER_COUNT   =
10)
WITH DRILLTHROUGH
```

Druga droga to utworzenie modelu, dla którego struktura utworzy się automatycznie i zawierać będzie wszystkie pola zdefiniowane w modelu.

```
CREATE MINING MODEL DMX_Model
(
  [NumberCarsOwned] LONG DISCRETE,
  [OrderNumber] TEXT KEY,
  [Region] TEXT DISCRETE PREDICT,
  [vTrainingAssocSeqLineItems] TABLE
  (
    [Category] TEXT DISCRETE PREDICT,
    [LineNumber] LONG KEY SEQUENCE
  )
)
USING   Microsoft_Sequence_Clustering(CLUSTER_COUNT   =
10)
WITH DRILLTHROUGH
```

Poza tym, że możemy tworzyć nowe modele, możemy również je kopiować, a dokładnie kopiować ich strukturę określając przy tym nazwę nowego modelu oraz algorytm jaki ma być do niego wykorzystany.

Model, który zamierzamy skopiować podawany jest po słowie kluczowym FROM.

```
SELECT INTO DMX_New_Model
USING Microsoft_Sequence_Clustering WITH DRILLTHROUGH
FROM DMX_Model
```

Utworzone modele, podobnie jak struktury można usuwać przy pomocy polecenia DROP.

```
DROP MINING MOEL <DMX_Model>
```

5.4 Trenowanie modelu eksploracyjnego

Kolejny etap, czyli trenowanie modelu realizowane jest poprzez instrukcję INSERT INTO. W jej kodzie musimy podać listę kolumn wejściowych i wyjściowych modelu, źródło danych oraz zapytanie SQL odnoszące się do tego źródła. Połączenie ze źródłem danych może być otwierane na dwa różne sposoby. Przy pomocy słowa kluczowego OPENQUERY wskazywane jest nazwane źródło, które można utworzyć w serwisie analitycznym wykorzystując Microsoft Visual Studio. Natomiast OPENROWSET służy do bezpośredniego wskazywania źródła danych poprzez podanie rodzaju dostawcy oraz adresu. Poniżej został przedstawiony jeden z nich.

```
INSERT INTO MINING MODEL [DMX_Model]
(
   [NumberCarsOwned],
   [OrderNumber],
   [Region],
   [vTrainingAssocSeqLineItems]
   (
        [Category],
        [LineNumber],
        SKIP
   )
)
SHAPE{ OPENQUERY([Adventure Works DW],
        'SELECT  [NumberCarsOwned],
              [OrderNumber],
              [Region]
        FROM
   AdventureWorksDW.dbo.vTrainingAssocSeqOrders
        ORDER BY    [OrderNumber]')}
APPEND({ OPENQUERY([Adventure Works DW],
        'SELECT  [Category],
              [LineNumber],
```

```
                [OrderNumber]
            FROM
        AdventureWorksDW.dbo.vTrainingAssocSeqLineItems
            ORDER BY    [OrderNumber]')}
    RELATE [OrderNumber] TO [OrderNumber]) AS [LineItems]
```

W tym miejscu warto zwrócić uwagę na element ORDER BY w zapytaniach do tabel podrzędnych i nadrzędnych. Element ten jest obowiązkowy, w przypadku jego braku serwer analityczny zwróci nam komunikat o nieprawidłowym uporządkowaniu danych.

Wytrenowany model może zostać "wyczyszczony", odkryte wzorce i rekordy w nim zawarte można usunąć korzystając z instrukcji DELETE.

```
DELETE FROM MINING STRUCTURE DMX_Structure
DELETE FROM MINING MODEL DMX_Model.CONTENT
```

5.5 Odpytywanie modelu eksploracyjnego

Wytrenowany model możemy odpytywać wykorzystując do tego celu instrukcję SELECT. Między innymi możemy sprawdzić jego zawartość, rekordy z bazy treningowej, które zawiera, możliwe stany atrybutów lub też przeprowadzić predykcję pewnych wartości.

```
SELECT ([Region])
FROM   DMX_Model
SELECT ([vTrainingAssocSeqLineItems])
FROM   DMX_Model
```

Najprostsze zapytanie, pozbawione jakichkolwiek modyfikatorów zwraca najbardziej prawdopodobną wartość dla kolumny podanej po wyrażeniu SELECT, w tym przypadku dla kolumny Region. Należy wiedzieć jednak, że użycie tak skonstruowanej instrukcji jest możliwe tylko dla kolumn predykcyjnych.

Poniżej zostały przedstawione przykładowe zapytania z różnymi modyfikatorami.

- DISTINCT

```
SELECT DISTINCT vTrainingAssocSeqLineItems.Category FROM
DMX_Model
```

Modyfikator DISTINCT zwraca możliwe stany atrybutu podanego w zapytaniu. Korzystając z niego można wyświetlić stany tylko dla jednej kolumny.

- CONTENT

```
SELECT * FROM [DMX_Model].CONTENT
```

Modyfikator CONTENT zwraca informacje o zawartości modelu. Wywołując takie zapytanie można sprawdzić nazwę modelu, nazwę schematu, zbiór węzłów i rodziców, dzieci każdego węzła, rekordy które zawiera czy jego typ oraz prawdopodobieństwo.

- DIMENSION_CONTENT

```
SELECT * FROM [DMX_Model].DIMENSION_CONTENT
```

Modyfikator DIMENSION_CONTENT podobnie jak CONTENT zwraca informacje o modelu, które można użyć jako wymiar (dimension) w kostkach OLAP będących obok zgłębiania danych potężnym narzędziem analitycznym.

- CASES

```
SELECT * FROM [DMX_Model].CASES
```

Modyfikator CASES zwraca zbiór wszystkich rekordów trenujących modelu.

- SAMPLE_CASES

```
SELECT * FROM [DMX_Model].SAMPLE_CASES
WHERE IsInNode('000000001')
```

Modyfikator SAMPLE_CASES zwraca zbiór rekordów zawartych w węźle podanym w warunku WHERE. W przypadku tego wariantu warunek ten jest elementem obowiązkowym.

- PREDICTION_JOIN

Ostatni z wymienionych modyfikatorów służy do przeprowadzania predykcji na podstawie danych pobranych podzapytaniem z zewnętrznego źródła. Źródło to wskazywane jest za pomocą tych samych instrukcji co w przypadku trenowania modelu: OPENQUERY oraz OPENROWSET.

```
SELECT
  t.[Region],
  t.[IncomeGroup],
```

```
PredictSequence([DMX_Model].[vTrainingAssocSeqLineItems]
)
From
  [DMX_Model]
PREDICTION JOIN SHAPE { OPENQUERY([Adventure Works DW],
    'SELECT
      [Region],
      [IncomeGroup],
      [NumberCarsOwned],
      [OrderNumber]
    FROM
      [dbo].[vTestAssocSeqOrders]
    ORDER BY
      [OrderNumber]')}
  APPEND({OPENQUERY([Adventure Works DW],
    'SELECT
      [LineNumber],
      [Category],
      [OrderNumber]
    FROM
      [dbo].[vTestAssocSeqLineItems]
    ORDER BY
      [OrderNumber]')}
    RELATE [OrderNumber] TO [OrderNumber]) AS
[vTestAssocSeqLineItems] AS t
ON
  [DMX_Model].[Region] = t.[Region] AND
  [DMX_Model].[NumberCarsOwned] = t.[NumberCarsOwned]
AND
  [DMX_Model].[vTrainingAssocSeqLineItems].[LineNumber]
= t.[vTestAssocSeqLineItems].[LineNumber] AND
  [DMX_Model].[vTrainingAssocSeqLineItems].[Category] =
t.[vTestAssocSeqLineItems].[Category]
```

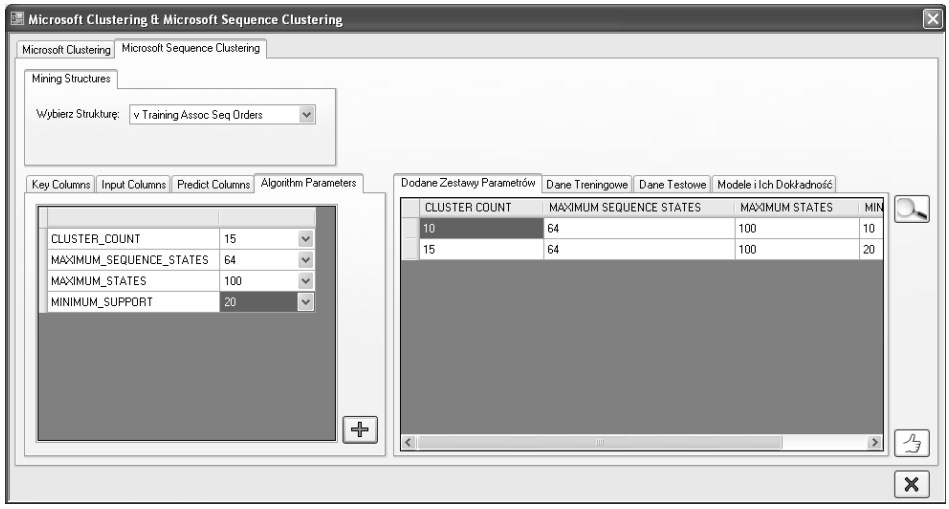
W powyższym kodzie został również zaprezentowany sposób użycia przykładowej funkcji *PredictSequence()* (prognozuj sekwencję), akceptującej jeden lub dwa parametry. Obowiązkowo należy wskazać kolumnę modelu typu tabela, która zawiera sekwencje oraz opcjonalnie można podać długość przewidywanej sekwencji (domyślnie długość wynosi 1). Dodatkowo funkcja ta zwraca wyniki w postaci hierarchicznej, rezultat ten możemy „spłaszczyć” przy pomocy słowa kluczowego FLATTENED.

W wymienionych przeze mnie przykładach można dostrzec dużą analogię w stosunku do języka SQL. Jest to duży plus tego rozszerzenia, zwłaszcza dla osób znających ów język i pozwala w łatwy i szybki sposób zrozumieć podstawy DMX.

6 Aplikacja windowsowa testująca dokładność wybranych modeli analitycznych

Opracowana, autorska aplikacja służy do tworzenia i testowania dokładności wybranych modeli analitycznych, a dokładniej modeli trenowanych za pomocą algorytmów Microsoft Clustering lub Microsoft Sequence Clustering. Minimalne wymagania, jakie muszą zostać spełnione, aby można było ją uruchomić to zainstalowany Microsoft SQL Server 2005 wraz z Analysis Services, środowisko uruchomieniowe .NET Framework 2.0 lub wyższe oraz biblioteka dostępu do Microsoft Analysis Services 2005 - ADOMD.NET.

Zadaniem aplikacji jest budowanie modeli analitycznych na podstawie wcześniej utworzonych struktur (na przykład z poziomu Visual Studio lub Menagement Studio). Budowanie modelu polega na wyznaczeniu kolumn wejściowych i predykcyjnych, a następnie określeniu różnych zestawów parametrów dla algorytmu trenującego – rysunek 11. Po wykonanej fazie treningowej program automatycznie sprawdza dokładność każdego z modeli, dzięki czemu umożliwia szybką identyfikację najlepszych parametrów przetwarzania dla rozwiązywanego zagadnienia.



Rys. 11. Aplikacja Testująca Modele Analityczne

Aplikacja składa się z dwóch głównych zakładek. Pierwsza przeznaczona jest dla modeli trenowanych przy pomocy algorytmu Microsoft Clustering, druga zaś przy pomocy algorytmu Microsoft Sequence Clustering. W obu przypadkach schemat tworzenia modeli jest taki sam. Różnica polega na wspomnianym już algorytmie

trenującym, a także na dopuszczalnej strukturze modeli. W przypadku klasteringu sekwencyjnego możliwe jest tworzenie modeli z pojedynczym zagnieżdżeniem tabel, natomiast w przypadku zwykłego grupowania takiej możliwości nie ma. W obu przypadkach jeżeli struktura bazy danych jest bardziej skomplikowana, rozwiązaniem jest tworzenie w bazie danych widoków, co zostało wykorzystane w tym przypadku.

Główne okna aplikacji podzielone są na trzy sekcje. W pierwszej, z listy struktur wcześniej utworzonych na serwerze analitycznym, wybierana jest struktura, na podstawie której tworzone są modele eksploracyjne.

W drugiej znajdują się trzy lub cztery w przypadku klasteringu sekwencyjnego zakładki: **Input Columns**, gdzie wybierane są kolumny wejściowe, **Predict Columns**, gdzie wybierane są kolumny predykcyjne oraz **Algorithm Parameters** pozwalająca tworzyć modele o tej samej budowie, ale o różnych parametrach przetwarzania. Czwarta zakładka charakterystyczna dla algorytmu Microsoft Sequence Clustering to niemodyfikowalna zakładka **Key Columns** zawierająca informacje o kluczach określonych w strukturze analitycznej.

Wreszcie ostatnia sekcja służy do mapowania kolumn modelu i bazy treningowej oraz testującej. Składa się ona z czterech zakładek. **Dodane Zestawy Parametrów**, która zawiera wszystkie dodane zestawy parametrów algorytmu Microsoft Clustering. **Dane Treningowe** i **Dane testowe** jak już wspomniałam służą do mapowania kolumn, natomiast zakładka **Modele i Ich Dokładność** wyświetla dokładność każdego z wytrenowanych modeli. Dokładność ta przedstawiana jest w dwóch różnych wariantach w zależności o typu kolumny predykcyjnej. Jeżeli przewidywane wartości są typu dyskretnego wówczas określany jest procent dokładności modelu wyliczany na podstawie stosunku ilości poprawnie przewidzianych przypadków do całkowitej liczby testowanych przykładów. Wariant drugi przewidziany jest dla kolumn typu ciągłego takich jak liczba całkowita, liczba zmiennoprzecinkowa lub data. W tym wypadku program podaje średni błąd modelu, który obliczany jest według wzoru 11:

$$Błąd = \frac{suma\ błędów\ w\ każdym\ przypadku}{średnia\ wartość\ rozpatrywanego\ atrybutu} \tag{11}$$

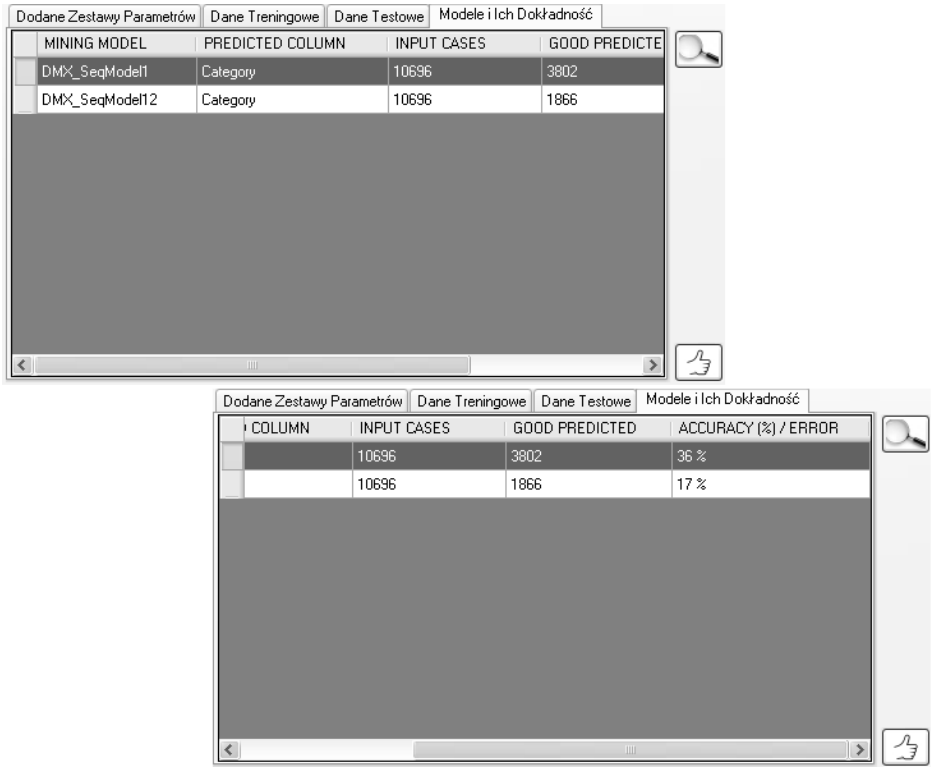
dla kolumn liczbowych, natomiast dla daty jest to wyrażenie (12):

$$Błąd = \frac{suma\ błędów\ w\ każdym\ przypadku}{ilość\ rozpatrywanych\ przypadków} \tag{12}$$

Po wybraniu struktury, pól wejściowych i predykcyjnych oraz po zmapowaniu danych treningowych i danych testowych użytkownik może

przejść do tworzenia modeli. Modele te są tworzone i dodawane do wybranej struktury, następnie są one trenowane oraz szacowana jest ich dokładność. Sprawdzanie dokładności oparte jest o widok testujący vTestAssocSeqLineItems, który jest różnej długości sekwencją kupowanych produktów. Na jej podstawie algorytm wyznacza następną sekwencję o długości zadanej w parametrze funkcji *PredictSequenc()* – w naszym przypadku długość ta wynosi jeden. Dokładność sprawdzana jest względem ostatniego elementu sekwencji testującej, a zasady obliczania dokładności lub błędu odpowiadają regułom opisanym powyżej. Najdłuższym etapem jest oczywiście trenowanie modeli, które ściśle zależy od ilości danych treningowych.

Jeżeli wszystkie fazy zakończą się pomyślnie, użytkownik zostanie o tym poinformowany, a wyniki można przeglądać na zakładce Modele i Ich Dokładność – rysunek 12.



Rys. 12. Aplikacja Testująca Modele Analityczne - Rezultat Przetwarzania Modeli

Zakładka Modele i Ich Dokładność zwiera informacje o nazwie modeli, kolumnach predykcyjnych, ilości przypadków testujących oraz o ilości przypadków trafnie przewidzianych i dokładności lub też błędzie

modelu. Kolumna predykcyjna Category modelu sekwencyjnego zawiera wartości dyskretne, dlatego szacowana dokładność podana jest w procentach. Warto nadmienić również, że ani Microsoft SQL Server Analysis Services, ani Microsoft Visual Studio nie wspierają sprawdzania dokładności modeli trenowanych algorytmami Microsoft Association Rules oraz Microsoft Sequence Clustering.

7 Biblioteka ADOMD.NET dlaklieenta Microsoft Analysis Services

Nie jest intencją autorów opisywanie całego kodu, ponieważ nie jest to główny temat poruszany w artykule. Przedstawiony zostanie tylko fragment aplikacji, który dotyczy programowego dostępu do Microsoft Analysis Services i jest ściśle związany z poruszanym zagadnieniem [1], [7].

Biblioteka ADOMD.NET jest następcą biblioteki ADOMD, dostępnej w Microsoft Analysis Services 2000 i została ona zaprojektowana z myślą o aplikacjach klienckich na platformie .NET korzystających z usług Microsoft Analysis Services 2005 i 2008. Dodatkowo została ona zaprojektowana według wzorca dostępu do danych z biblioteki ADO.NET, implementuje standardowe interfejsy z przestrzeni nazw System.Data wzbogacając je właśnie o dedykowane funkcje analityczne.

ADOMD.NET do komunikacji z serwerem używa protokołu XML for Analysis (XMLA), który umożliwia łatwe tworzenie tak zwanych inteligentnych aplikacji zawierających funkcje analityczne. Specyfikacja ta określa funkcjonalność udostępnianą przez Microsoft Analysis Services i pozwala na wysyłanie różnego typu instrukcji: Multidimensional Expressions (MDX), Data Mining Extensions (DMX), Analysis Services Scripting Language (ASSL) lub w ograniczonym stopniu zapytania SQL. Instrukcje te można podzielić na dwa typy: Discover i Execute. Pierwszy z nich pozwala na uzyskanie informacji i metadanych charakteryzujących serwer oraz utworzone obiekty. Między innymi umożliwia pobranie listy dostępnych struktur i modeli analitycznych wraz z opisem typów i rodzajów ich pól. Natomiast żądania typu Execute umożliwiają wykonywanie na serwerze poleceń rozszerzenia DMX, takich jak utworzenie nowej struktury lub modelu oraz wytrenowanie i odpytywanie obiektów analitycznych.

Podsumowanie

Celem pracy było wykorzystanie rozszerzenia DMX do zgłębiania danych. Zgodnie z tym zamiarem stworzona została aplikacja realizująca procesy tworzenia modeli analitycznych, a także ich trenowania i testowania dokładności. Dodatkowo została omówiona

alternatywna wizualna metoda wykonywania tych czynności z wykorzystaniem dedykowanego narzędzia Microsoft Visual Studio .NET.

Tworząc program przez cały czas na uwadze autorów pozostawało zachowanie możliwie jak największej uniwersalności jego konfiguracji i działania, przy czym nie by był celem kopiowanie narzędzi dostępnych w Microsoft Visual Studio. Zdaniem autorów cel ten został osiągnięty, a aplikacja cechuje się ogólnym charakterem i dużą elastycznością. W ramach wcześniej utworzonych struktur użytkownik ma możliwość dowolnej manipulacji kolumnami oraz parametrami algorytmu trenującego. Sukcesem zakończyła się również implementacja modułu sprawdzającego dokładność zarówno dla algorytmu Microsoft Clustering, jak i Microsoft Sequence Clustering. W przypadku algorytmu sekwencyjnego jest to bardzo wartościowa funkcja, ponieważ zabrakło jej w narzędziach udostępnionych przez Microsoft.

Literatura

[1] A. J. Brust, S. Forte, *Programowanie Microsoft SQL Server 2005* / Microsoft Press, 2006

[2] P. Cichosz, *Systemu uczące Się* / WNT, 2007

[3] D. Hand, H. Mannila, P.Smyth, *Metody i modele eksploracji danych* / WNT, 2005

[4] R. Jacobson, S. Misner, H. Consulting, *SQL Server 2005 Analysis Services*, Microsoft Press, 2006

[5] D. T. Larose, *Metody i modele eksploracji danych* / PWN, 2008

[6] D. T. Larose, *Odkrywanie wiedzy z danych* / PWN, 2006

[7] Z. Tang, J. MacLennan, *Data Mining with SQL Server 2005* / Wiley, 2005

[8] SQL Server 2005 Books Online – Data Mining Algorithms, http://msdn.microsoft.com/en-us/library/ms175595(SQL.90).aspx

IMPLEMENTATION OF DMX EXTENSION FOR DATA MINING ON MS SQL SERVER PLATFORM

Summary **-** The paper describes methods and algorithms of data mining at the MS Analysis Services platform. At beginning the main idea of such process was presented. All steps from the relational schema through integration, schema construction till reporting was discuss. In the next step all implemented in presented tool models was presented with special attention to its mathematical principles. The main care was pointed to Microsoft Sequence Clustering, as a main subject of this work. The DMX SQL extension was presented as the most important tool to build, process and test mining structures and models. This language was used in the client application, which was created by C# .NET. It gives opportunity to create data mining clustering models for any tables from chosen relational schema. The main original part of this work is the tool to testify Microsoft Sequence Clustering model, which is not presented in Analysis Services toolkit. The ADOMD library was used to the communication between server engine and client application, and was shortly described.