Zeszyty Naukowe Politechniki Częstochowskiej
Zarządzanie Nr 26 (2017) s. 132–145
dostępne na: http://www.zim.pcz.pl/znwz

EKSPLORACJA DANYCH TRANSAKCYJNYCH SKLEPU INTERNETOWEGO

Maciej Pondel[1], Jerzy Korczak[2]

Uniwersytet Ekonomiczny we Wrocławiu
Wydział Zarządzania, Informatyki i Finansów
[1]Unity SA, Wrocław; [2]ICT4EDU, Wrocław

Streszczenie: Przy stale rozwijającym się rynku handlu elektronicznego właściciele sklepów internetowych oraz podmioty prowadzące działalność handlową w modelu Omnichannel muszą szukać przewag konkurencyjnych nie tylko w najniższej cenie oferowanych usług i produktów, ale również w innych obszarach. Dogłębne poznanie potrzeb i przyzwyczajeń klientów oraz ich preferencji zakupowych pomaga w dostarczeniu oferty dopasowanej do potrzeb klienta. Dzięki temu można zwiększyć lojalność klientów, co docelowo wpłynie pozytywnie na rentowność prowadzonych działań handlowych. Eksploracja danych jest procesem, przy pomocy którego można lepiej poznać swoich klientów. Artykuł ten skupia się na problemach, jakie można rozwiązać przy użyciu mechanizmów eksploracji danych. Prezentuje metodykę przyjętą w projekcie RTOM[1] oraz pokazuje przykłady procesów przeprowadzonych w narzędziu Orange i w systemie Hadoop przy użyciu silnika Spark i biblioteki MLlib.

Słowa kluczowe: eksploracja danych, Hadoop, MLlib, Orange, reguły asocjacyjne, Spark

DOI: 10.17512/znpcz.2017.2.12

Wprowadzenie

Eksploracja danych jest procesem automatycznego wykrywania nietrywialnych, nieznanych, potencjalnie użytecznych zależności, reguł, wzorców, schematów, podobieństw lub trendów w dużych zbiorach danych (Morzy 2013; Witten i in. 2017). Najogólniej mówiąc, zadaniem eksploracji jest analiza danych i procesów w celu lepszego ich poznania, zrozumienia i wykorzystania w toku podejmowania decyzji. Eksploracja danych jest dziedziną multidyscyplinarną, integrującą szereg obszarów badawczych – takich jak systemy informacyjne, bazy danych i hurtownie – statystykę, sztuczną inteligencję, obliczenia równoległe, badania operacyjne, wizualizację i grafikę komputerową. Współczesne systemy eksploracji wykorzystują szeroko technologie informacyjno-komunikacyjne, technologie Web, metody wyszukiwania informacji, techniki geolokalizacji, przetwarzania sygnałów i bioinformatyki.

Problematyka analizy i eksploracji dużych baz marketingowych jest przedmiotem wielu badań i projektów aplikacyjnych (Linoff, Berry 2011; Han, Kamber, Pei 2012;

[1] Projekt Real-Time Omnichannel Marketing (RTOM) jest realizowany przez firmę Unity SA w ramach poddziałania RPO WD 2014-2020. Numer umowy RPDS.01.02.02-02-0079/15-00.

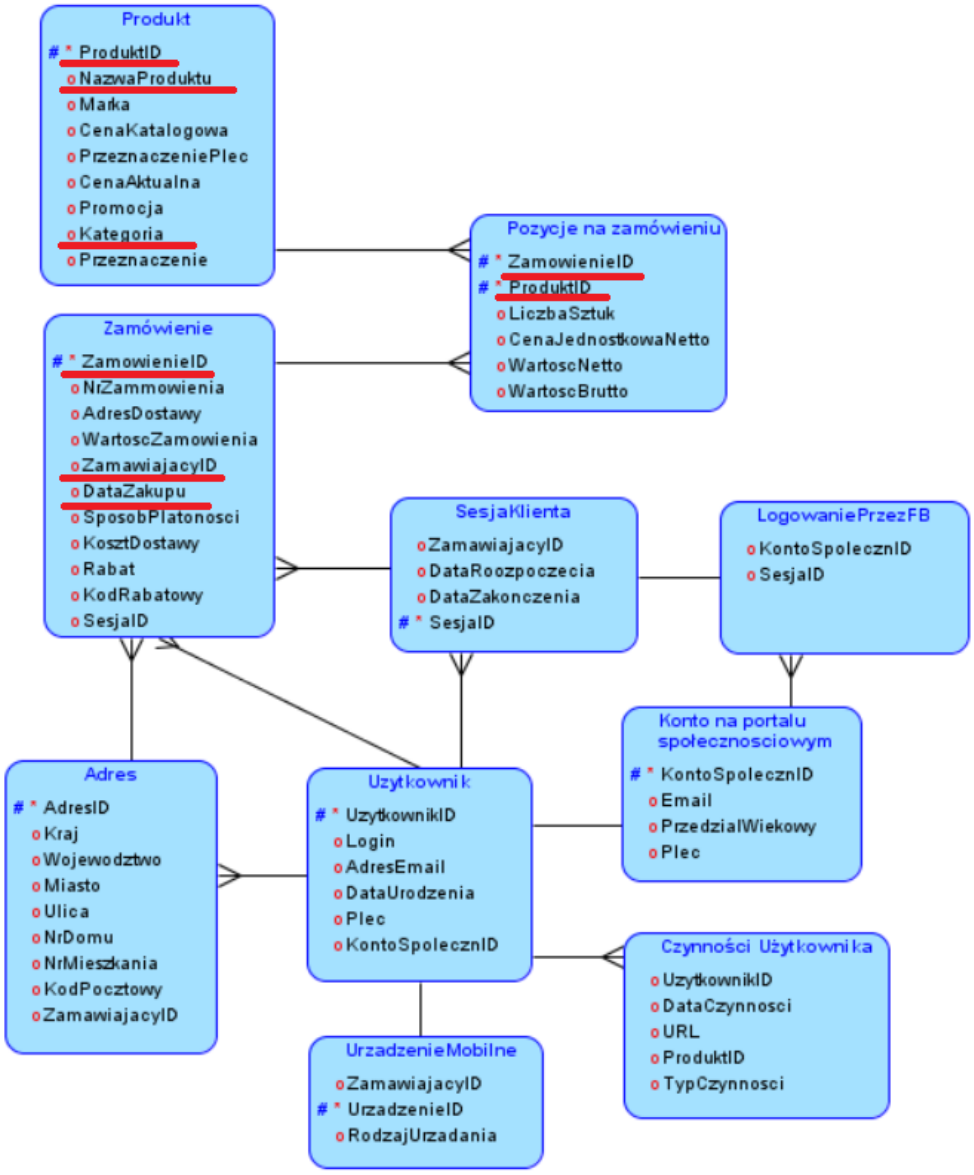
Korczak, Pondel 2017; Morzy 2013; Pawełoszek, Korczak 2017; Weichbroth 2009; Pondel 2015). Celem artykułu jest z jednej strony pokazanie metodyki eksploracji danych transakcyjnych sklepu internetowego, a z drugiej zaprezentowanie wykorzystania algorytmów analizy koszyka zakupów i wyszukiwania wzorców zachowania klientów w czasie. Problemy te zostaną przedstawione na rzeczywistych danych transakcyjnych, wykorzystanych w projekcie inteligentnej platformy analizy danych wielokanałowej sprzedaży (ang. projekt *Real-Time Omnichannel Marketing* – RTOM). W projekcie dane są gromadzone głównie w czasie rzeczywistym i przetwarzane w ogromnych ilościach, przy dużej heterogeniczności ich źródeł, formatów, wolumenu i intensywności napływu. Użytkownik platformy (menedżer, analityk marketingu) oczekuje nietrywialnej, nowej i użytecznej wiedzy, którą będzie mógł wykorzystać w procesie podejmowania decyzji. Wiedza pozyskana z zebranych danych powinna być użyta w sposób automatyczny w procesach komunikacji z klientem tak, aby zoptymalizować wybrany parametr biznesowy procesu, np. prawdopodobieństwo zakupu, satysfakcję klienta, ryzyko odejścia klienta, marżę na produkcie i wiele innych.

W artykule przedstawiono strukturę danych, do której trafiają informacje pochodzące z systemu e-commerce. Na potrzeby analityczne struktura została poddana niewielkim zmianom denormalizacyjnym w stosunku do struktury bazy danych sklepu internetowego. Omówiony został sposób pobierania danych ze struktury analitycznej na potrzeby zasilenia algorytmów eksploracji danych. W kolejnej części została przedstawiona metodyka, według której zdecydowaliśmy się przeprowadzać eksplorację. Zostały omówione także przykłady generowania reguł asocjacyjnych w oparciu o wybrane algorytmy.

Źródła danych transakcyjnych

Celem eksploracji jest wykrycie najczęściej kupowanych grup produktów przez klientów sklepu internetowego oraz określenie reguł asocjacyjnych opisujących relacje między często kupowanymi razem produktami. Zakładamy, że znalezione wzorce zakupów będą wykorzystane do opracowania strategii sprzedaży, akcji promocyjnych, organizacji stron internetowych czy doskonalenia katalogu oferowanych produktów.

Głównym źródłem informacji jest baza danych zwierająca transakcje zakupów klientów sklepu internetowego. Model konceptualny bazy ilustruje *Rysunek 1*. Biorąc pod uwagę zakreślony obszar tematyczny artykułu, w badaniach wykorzystamy tylko dane podkreślone w schemacie.



Rysunek 1. Schemat logiczny bazy danych

Źródło: Opracowanie własne w programie Oracle SQL Developer Data Modeler

W sklepie internetowym dane transakcyjne są przechowywane w relacyjnej bazie danych PostgreSQL. Platforma RTOM kopiuje dane transakcyjne do swojej własnej struktury opartej na technologii Apache Hadoop. Dane składowane są w systemie plików HDFS, a dostęp do nich zapewniają mechanizmy hurtowni danych Hive oraz Impala.

Z bazy tej dwoma kwerendami wybraliśmy informacje niezbędne do dalszych badań:

```
select NazwaProduktu, ZamawiajacyID, ZamowienieID, datediff(now(),
   from_unixtime(cast((place_date / 1000) AS BIGINT),"yyyy-MM-dd"))
   as DataZam
   from Produkt p join [Pozycje na zamowieniu] poz
   on p.ProduktID poz.ProduktID join Zamowienie z
   on p.ZamowienieID = z.ZamowienieID
   where DataZakupu > '2016-01-01'
   order by ZamowienieID, ZamawiajacyID
```

Wynikiem działania zapytania była tabela, zawierająca nazwę produktu, identyfikator zamówienia oraz identyfikator produktu, zaprezentowana jako *Tabela 1*. Badanie przeprowadzono w oparciu o bazę danych pochodzącą z rzeczywistego sklepu. W celu zapewnienia poufności danych nazwy produktów w niniejszym artykule zostały zamienione na przykładowe produkty A, B, C, D.

Tabela 1. Przykład danych pobranych z bazy

NazwaProduktu	ZamawiajacyID	ZamowienieID	DataZam
Produkt A	24243	3951353	-3
Produkt A	24243	3954763	-20
Produkt B	24362	3935375	-10
Produkt C	24362	3935375	-10
Produkt A	37436	3929848	-19
Produkt B	37436	3939923	-30
Produkt C	37436	3939923	-30
Produkt D	404907	3930528	-38
Produkt B	404907	3930528	-38

Źródło: Opracowanie własne

Następnie, przy pomocy kodu Python, wygenerowaliśmy 2 pliki. W pierwszym przypadku w jednej linii znajduje się cała zawartość jednego zamówienia. Format tego pliku to {*Ti, P1, P2,...,Pm*}, gdzie *Ti* oznacza identyfikator transakcji, zaś *Pj* – nazwę zakupionego produktu. Drugi plik natomiast zawiera wszystkie dodatkowe informacje o kliencie *Kj* i datę transakcji *Dk*. Przed zaprojektowaniem modelu eksploracji dane zostały wstępnie przetworzone i zagregowane. Wstępne przetwarzanie dotyczyło przetransformowania daty transakcji na liczbę dni dzielących zamówienie od aktualnej daty oraz dyskretyzacji wartości atrybutów ciągłych (Han, Kamber, Pei 2012). W ramach tych prac dokonano też agregacji produktów według klasyfikacji przyjętej w sklepie internetowym. Fragment tego pliku przedstawiono w *Tabeli 2*.

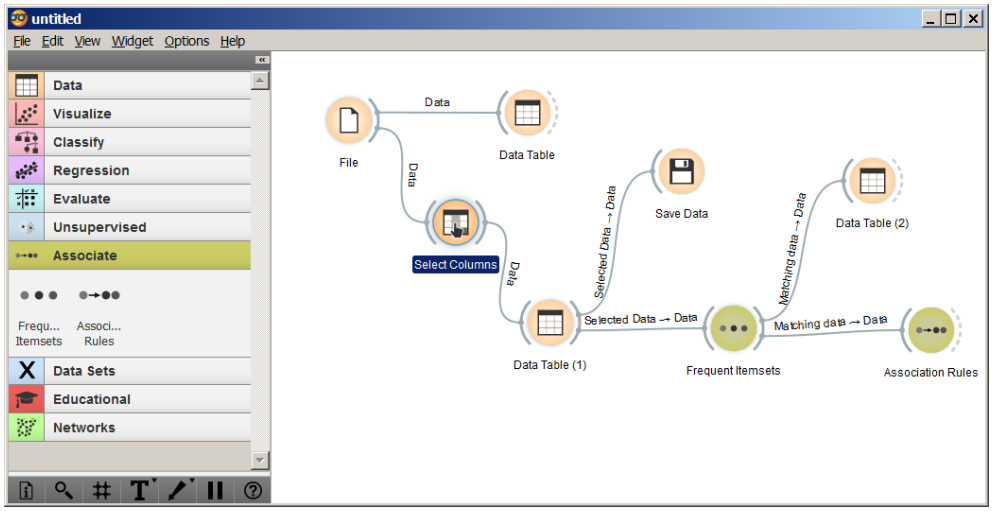
Tabela 2. Fragment pliku danych wejściowych z agregacją produktów

ZamowienieID	DataZam	Produkt1	Produkt2	Produkt3
3951353	-3	Produkt A		
3954763	-20	Produkt A		
3935375	-10	Produkt B	Produkt C	
3929848	-19	Produkt A		
3939923	-30	Produkt B	Produkt C	
3930528	-38	Produkt D	Produkt B	

Źródło: Opracowanie własne

Pilotowa wersja procesu eksploracji

Zgodnie z przyjętą metodyką eksploracji (Korczak, Pondel 2017; Piatetsky 2014), po zdefiniowaniu problemu, na podstawie uzyskanych danych, zaprojektowano prototypowy proces eksploracji danych, korzystając z pakietu Orange (*Rysunek 2*). Dane dostosowano do formatu wymaganego przez Orange, ograniczając liczbę atrybutów (kolumn) i liczbę instancji (wierszy).



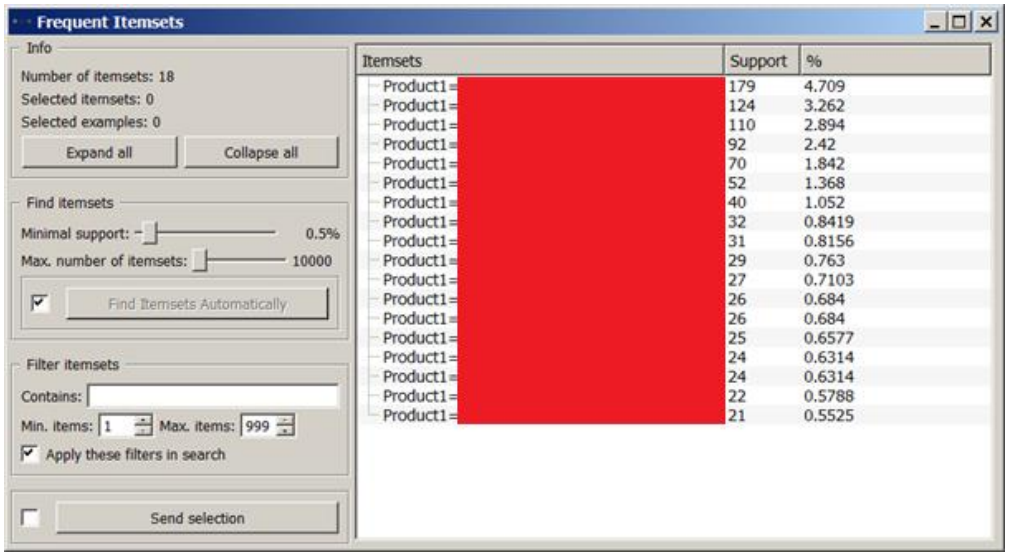
Rysunek 2. Schemat procesu eksploracji danych

Źródło: Opracowanie własne w programie Orange

Opracowany projekt prototypu procesu eksploracji miał na celu przeprowadzenie wstępnej weryfikacji i walidacji modelu oraz podejścia analitycznego przez menedżerów marketingu. W pakiecie Orange jest dostępny tylko jeden model wyszukiwania reguł asocjacyjnych zbudowany na algorytmie FP-Growth (Haoyuan 2008). W algorytmie proces wykrywania zbiorów częstych jest realizowany w dwóch krokach:

1. Kompresja bazy danych D do FP-drzewa: baza danych transakcji D jest kompresowana i przekształcana do postaci FP-drzewa.
2. Eksploracja FP-drzewa: FP-drzewo jest przeszukiwane w celu znalezienia zbiorów częstych.

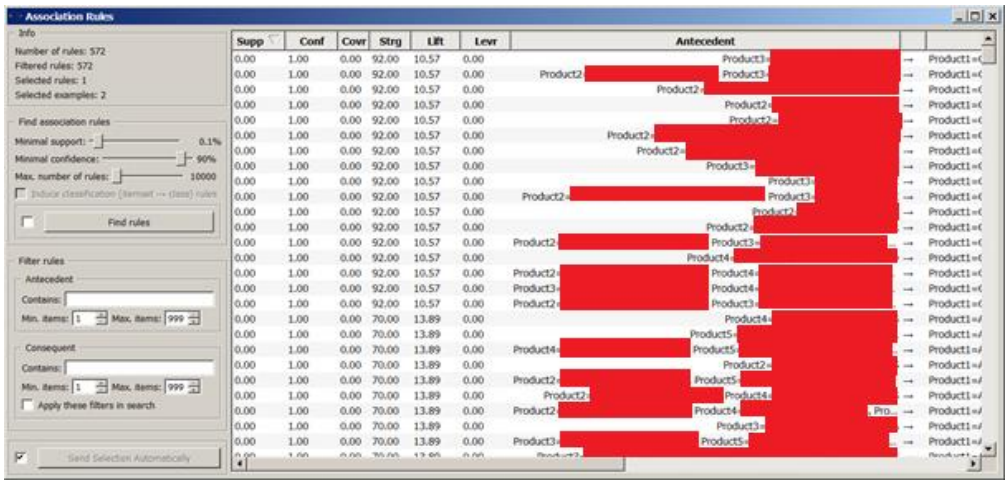
Progowe częstości zostały określone przez analityka parametrem: *Minimal support*. Na ogół wartość tego parametru wyznacza się eksperymentalnie w zależności od celów marketingowych. W naszym przykładzie, z uwagi na dużą liczbę transakcji zakupu, przyjęliśmy wartość progową równą 0,5%. Oznacza to zadanie wyszukania produktów, które występują w co najmniej 0,5% transakcji. Wykaz tych produktów (przy czym lista konkretnych produktów została utajniona) wraz z ich częstościami wystąpień pokazano na *Rysunku 3*.



Rysunek 3. Wykaz produktów częstych

Źródło: Opracowanie własne w programie Orange

Po określeniu produktów często kupowanych w drugim etapie wygenerowano listę reguł asocjacyjnych, spełniających podane w parametrach warunki minimalnego wsparcia (*minimal support*) oraz minimalnego wskaźnika ufności (ang. *minimal confidence*). Listę wygenerowanych reguł asocjacyjnych pokazano na *Rysunku 4*.



Rysunek 4. Wykaz reguł asocjacyjnych

Źródło: Opracowanie własne w programie Orange

Orange podaje informację o wartościach miar użyteczności dla każdej z wyszukanych reguł. Na *Rysunku 4* pokazano wartości: *Support*, *Confidence*, *Coverage*, *Strength*, *Lift and Leverage*.

Przykładowo reguła A → B ma wskaźnik wsparcia większy niż 0,1%, wskaźnik ufności 100%, wskaźnik siły 178,00, wskaźnik dźwigni 5,46.

Zaznaczmy, że oceny użyteczności dokonuje się dla reguły X → Y na podstawie następujących danych:

- n oznacza liczbę wszystkich transakcji w bazie danych,
- n_X oznacza liczbę sekwencji zawierających produkt X,
- n_Y oznacza liczbę sekwencji zawierających Y,
- n_{XY} oznacza liczbę sekwencji zawierających X i Y.

W pakiecie Orange mamy do wyboru następujące miary:

- wskaźnik wsparcia (ang. *support*) obliczany ze wzoru n_{XY}/n,
- wskaźnik ufności (ang. *confidence*) obliczany ze wzoru n_{XY}/n_X,
- wskaźnik pokrycia (ang. *coverage*) obliczany ze wzoru n_X/n,
- wskaźnik siły (ang. *strenght*) obliczany ze wzoru n_Y/n_X,
- wskaźnik dźwigni (ang. *lift*) obliczany ze wzoru $n \times n_{XY}/n_X \times n_Y$,
- wskaźnik wpływu (ang. *leverage*) obliczany ze wzoru $n \times n_{XY} - n_X/n_Y$.

W platformie Orange zgodnie z przyjętą metodyką wygenerowano jedynie wstępne wyniki. Ograniczenia wydajnościowe tej platformy nie pozwalały na pracę na pełnym zbiorze danych. Wstępne wyniki poddano ewaluacji przez analityków i menedżerów marketingu, którzy zdecydowali, że przyjęte podejście jest właściwe i nawet wstępne wyniki prowadzą do interesujących konkluzji biznesowych.

Po akceptacji tego etapu prac dokonano eksploracji na pełnym zbiorze danych przy użyciu biblioteki MLlib utworzonej przy użyciu silnika Spark, który wykorzystując paradygmat MapReduce, dzieli zadanie na fragmenty, które są wykony-

wane przez niezależne węzły obliczeniowe platformy. Dzięki takiemu podejściu zadanie wykonywane jest dużo wydajniej niż w przypadku klasycznego podejścia. W naszym przypadku użyto platformy Cloudera zainstalowanej na środowisku 8 maszyn wirtualnych o łącznej liczbie rdzeni procesorów wynoszącej 32 oraz łącznej pamięci RAM 48 GB. Dzięki rozproszonemu przetwarzaniu platforma umożliwia eksplorację nawet dużych zbiorów danych, z którymi wydajnościowo nie radzi sobie oprogramowanie Orange.

W bibliotece MLlib rozpoczęliśmy od budowy reguł asocjacyjnych przy użyciu algorytmu FPGrowth. Użyliśmy do tego celu biblioteki `org.apache.spark.mllib.fpm.FPGrowth`, która niestety nie zwraca wszystkich wskaźników ujętych w platformie Orange, stąd wartości większości wskaźników musimy obliczyć samodzielnie. Poniżej znajduje się kod w języku Scala, generujący model FPGrowth przy zadanej minimalnej wartości wskaźnika wsparcia oraz przy podanej liczbie partycji. Liczba partycji odpowiada właśnie liczbie węzłów obliczeniowych zaangażowanych w przetwarzanie. W aplikacjach biznesowych najczęściej stosowane są dwa wskaźniki: wsparcia i ufności. Odwołując się do naszego przykładu, wsparcie reguły określa liczbę transakcji klientów, którzy kupują zgodnie z daną regułą. Natomiast reguły o niewielkim wsparciu opisują zachowanie niewielkiej grupy klientów. Z drugiej strony – reguły o wysokim wsparciu są zazwyczaj mało interesujące dla menedżerów, ponieważ ze względu na swoją powszechność są im dobrze znane. Druga istotna miara, wskaźnik ufności reguły, określa, na ile wykryta reguła asocjacyjna jest „pewna". Reguły o niskiej ufności są mało wiarygodne, natomiast reguły charakteryzujące się wysoką ufnością są „prawie pewne".

Poniższy fragment kodu prezentuje na ekranie również wyniki w postaci listy często pojawiających się zbiorów (w naszym przypadku produktów). Każdy element listy opatrzony jest wartością wskaźnika suport oraz liczbą oznaczającą częstość wystąpienia (n_{XY}).

```
// inicjacja obiektu klasy FPGrowth z ustawieniem wartości parame-
   trów
val fpg = (new
   FPGrowth().setMinSupport(minSupport).setNumPartitions(10))
val model = fpg.run(transactions) //uruchomienie przetwarzania na
   liście transkacji
val fiArray = model.freqItemsets.collect(); //zwrócenie wyników mo-
   delu do tablicy
println("FP-Growth frequences: ")
      // wydrukowanie wyników - czesto występujących zbiorów
      model.freqItemsets.collect().foreach { itemset => {
val sup = round(itemset.freq.toDouble / userTransactions)
println(itemset.items.mkString("   [", ",", "]") + ", support=" +
   sup + ", freq=" + itemset.freq)
         }
      }
```

Wybrane wyniki wygenerowane przy pomocy przedstawionego kodu stanowiące częste zbiory przedstawiono na *Rysunku 5*.

```
[Produkt A], support=0.5077, freq=13198
[Produkt B], support=0.4489, freq=11669
[Produkt C], support=0.2816, freq=7321
[Produkt A, Produkt B], support=0.1613, freq=4193
[Produkt C, Produkt B], support=0.0871, freq=2264
[Produkt C, Produkt A], support=0.1491, freq=3877
```

Rysunek 5. Lista częstych zbiorów wygenerowana przez algorytm FPGrowth

Źródło: Opracowanie własne

Mając utworzony model, możemy wygenerować listę reguł asocjacyjnych. W przypadku reguł, biblioteka MLlib zwraca nam informację o częstości wystąpienia oraz o wskaźniku ufności (*confidence*). Biblioteka standardowo nie zwraca wartości wskaźnika wsparcia (*support*), stąd samodzielnie musimy obliczyć wskaźnik wsparcia. Poniższy kod języka Scala, będący kontynuacją wykorzystania modelu zbudowanego przy pomocy klasy FPGrowth, prezentuje wygenerowane reguły, oblicza wskaźnik wsparcia i drukuje reguły wraz z oboma wskaźnikami.

```
val ar = model.generateAssociationRules(minConfidence).collect()
// generowanie reguł

println("\n")
println("Association rules: ")
ar.foreach { rule =>

   val associationRulesItems =
   rule.antecedent.toSet.union(rule.consequent.toSet)

    val f = model.freqItemsets.filter(fi => {
        val fiItems = fi.items.toSet
        if(fiItems.size == associationRulesItems.size) {
            val intersected = fiItems.intersect(associationRulesItems)
         intersected.equals(associationRulesItems) && intersect-
         ed.equals(fiItems)
        }
        else {
            false
        }
    }).first.freq
    val support: Double = f.toDouble / userTransactions
       //obliczenie wskaźnika wsparcia
    println( //wydruk rezultatów
            rule.antecedent.mkString("   [", ",", "]")
            + " => " + rule.consequent.mkString("[", ",", "]")
            + ", support=" + round(support)
            + ", confidence=" + round(rule.confidence)
    )
```

Wybrane reguły asocjacyjne wraz z wartościami wymienionych miar zostały zaprezentowane na *Rysunku 6*. W dyskusji ze specjalistami marketingu potwierdzono zasadność oraz możliwość realizacji i wykorzystania modelu eksploracji w podejmowaniu decyzji marketingowych.

```
Association rules:
   [Produkt C] => [Produkt B], support=0.0871, confidence=0.3092
   [Produkt C] => [Produkt A], support=0.1491, confidence=0.5296
   [Produkt A] => [Produkt C], support=0.1491, confidence=0.2938
   [Produkt A] => [Produkt B], support=0.1613, confidence=0.3177
   [Produkt A, Produkt B] => [Produkt C], support=0.0131, confidence=0.5056
```

Rysunek 6. Reguły asocjacyjne wygenerowane w bibliotece MLlib

Źródło: Opracowanie własne

W niektórych przypadkach analityk może być tylko zainteresowany regułami, które zawierają określony produkt lub grupę produktów.

Analiza wygenerowanych reguł przez specjalistę od marketingu oraz handlu daje szereg korzyści biznesowych (por.: Chorianopoulos 2016):

1. Prowadzi do lepszego zrozumienia motywacji klientów do zakupu poszczególnych produktów.
2. Pozwala na kreowanie kampanii marketingowych, działań sprzedażowych czy polityki cenowej w stosunku do wybranych grup produktowych bądź segmentów klientów, które w założeniu będą skuteczniejsze niż tradycyjne metody oparte jedynie na intuicji handlowca.

Dodatkowo w oparciu o wygenerowany model możemy zbudować system automatycznych rekomendacji zakupowych prezentujący klientowi produkt, którym potencjalne mógłby być on zainteresowany (por.: Schutt, O'Neil 2013).

W dalszej części artykułu przedstawiono tylko wyniki dwóch procesów eksploracji, które pozwoliły na wygenerowanie bardziej interesujących dla menedżerów reguł aniżeli proste reguły analizy koszyka produktów.

Wykrywanie wielopoziomowych reguł asocjacyjnych oraz sekwencyjnych

Z punktu widzenia stopnia abstrakcji przetwarzanych danych wyróżniamy dwa rodzaje reguł asocjacyjnych: jednopoziomowe reguły asocjacyjne (ang. *single-level association rules*) oraz wielopoziomowe lub uogólnione reguły asocjacyjne (ang. *multilevel* lub *generalized association rules*) (Han, Fu 1999; Setia, Jyoti 2013). Regułę asocjacyjną nazywamy jednopoziomową regułą asocjacyjną, jeżeli dane występujące w regule reprezentują ten sam poziom abstrakcji.

Problem wykrywania reguł wielopoziomowych występuje w przypadku tzw. „rzadkich baz danych", których przykładem jest baza transakcji w RTOM-ie. Są to bazy, w których jest wiele transakcji i tysiące produktów, przy czym średni koszyk zawiera od kilku do kilkunastu produktów.

Zauważmy, że produkty występujące w transakcjach reprezentują różne poziomy abstrakcji: przykładowo produkt z kategorii „Telewizory" reprezentuje wyższy poziom abstrakcji aniżeli produkt opisany poprzez np. „Telewizor LG 40 cali". Produkt „Telewizory" jest, inaczej mówiąc, generalizacją różnych produktów tej samej kategorii. W bazie danych istnieje wiele hierarchii poziomów abstrakcji.

Reguły, które opisują asocjacje występujące pomiędzy danymi reprezentującymi różne poziomy abstrakcji, nazywa się wielopoziomowymi regułami asocjacyjnymi.

Wielopoziomowe reguły asocjacyjne posiadają często większą wartość poznawczą dla analityków i decydentów aniżeli jednopoziomowe reguły asocjacyjne. Operują one na ogólniejszych hierarchiach pojęciowych, które są czytelniejsze i łatwiejsze w interpretacji oraz reprezentują uogólnioną wiedzę. Należy nadmienić, że wielopoziomowych reguł asocjacyjnych nie można wyprowadzić ze zbioru jednopoziomowych reguł asocjacyjnych.

W przypadku reguł sekwencyjnych wykorzystaliśmy algorytm PrefixSpan znajdujący się w bibliotece `org.apache.spark.mllib.fpm.PrefixSpan`. Generuje on często pojawiające się sekwencje wraz z ich częstością. Niestety w przypadku tego algorytmu system nie zwraca reguł oraz nie oblicza wskaźników wsparcia oraz ufności (dlatego ich implementacji musieliśmy dokonać samodzielnie). Model tworzony przy pomocy algorytmu PrefixSpan zasilony jest tablicą trójwymiarową, gdzie pierwszy wymiar stanowią poszczególni klienci, w ramach drugiego wymiaru podawane są wszystkie transakcje klientów. Trzeci wymiar to produkty znajdujące się w poszczególnych transakcjach. Przykład pliku zasilającego znajduje się poniżej:

```
Array(Array('Produkt A', 'Produkt C'), Array('Produkt C')),
 Array(Array('Produkt A'), Array('Produkt C', 'Produkt B'), Array('Produkt
A', 'Produkt B')),
```

Po wprowadzeniu niemal 55 tys. transakcji klientów sklepu wygenerowano listę często kupowanych produktów (*Rysunek 7*).

```
PrefixSpan total 54844
support=0.09 freq=5166 [['Produkt A']]
support=0.06 freq=3404 [['Produkt B']]
support=0.04 freq=2012 [['Produkt C']]
support=0.03 freq=1880 [['Produkt B'], [' Produkt A']]
support=0.02 freq=1370 [['Produkt A '], ['Produkt A']]
support=0.03 freq=1683 [['Produkt C'], ['Produkt A']]
support=0.02 freq=1370 [['Produkt B'], ['Produkt B']]
```

Rysunek 7. Zbiory częste wygenerowane przez algorytm PrefixSpan z biblioteki MLlib

Źródło: Opracowanie własne

Zbiory częste wieloelementowe stanowiły podstawę do utworzenia sekwencyjnych reguł asocjacyjnych, natomiast odpowiadające im zbiory jednoelementowe posłużyły nam do obliczenia wskaźnika ufności tej reguły (częstość zbioru wieloelementowego dzielona przez częstość zbioru jednoelementowego). Na przykład wiedząc, że:

```
support=0.03 freq=1880 [['Produkt B'], [' Produkt A']]
support=0.06 freq=3404 [['Produkt B']]
```

możemy wygenerować regułę sekwencyjną:

```
['Produkt B'] => ['Produkt A'] support=0.03, confidence = 0.55
```

Budowa reguł asocjacyjnych może pomóc nam w następujących zadaniach:

1) poszukiwanie koszyka zakupów klientów sklepu internetowego,

2) poszukiwanie wzorców sekwencji zakupów klientów sklepu internetowego.

Oba te zadania sprowadzają się do zbudowania modelu, który pozwoli lepiej zrozumieć zachowania klienta oraz zaproponować efektywne rekomendacje zakupowe (por.: Pondel, Pondel 2011; Pondel 2011).

Przedstawione powyżej przykłady generowania reguł asocjacyjnych dotyczą całego dostępnego zbioru transakcji, ograniczonego jedynie czasowo. Przyjęliśmy, że najstarsze transakcje nie są dla nas interesujące, ponieważ w okresie dłuższym niż 3 lata mogą się radykalnie zmienić gusta klientów, mody, a co za tym idzie – wzorce zakupowe. Najwyższy wskaźnik ufności w takim przypadku wynosił niewiele powyżej 50% (co i tak świadczy o dużej wartości biznesowej danej reguły).

W kolejnych krokach postanowiliśmy posegmentować zbiór klientów ze względu na najbardziej istotne z perspektywy marketingu charakterystyki. Już przy pierwszym podziale zbioru na kobiety i mężczyzn udało nam się uzyskać dla wybranych reguł wskaźniki ufności na poziomie przekraczającym 70%. Co więcej, udało się znaleźć reguły, które w zbiorze mężczyzn charakteryzowały się niewielkim wskaźnikiem ufności (10%, natomiast w przypadku kobiet ten wskaźnik wynosił powyżej 70%, co dobitnie pokazało, że znaleźliśmy reguły mówiące o jasnych różnicach w preferencjach zakupowych poszczególnych segmentów klientów. W badaniach zajęliśmy się także analizą reguł asocjacyjnych zbudowanych w oparciu o segmenty klientów wraz z porównaniem wskaźników pomiędzy segmentami. Do segmentacji użyliśmy nie tylko naszej intuicji, ale również mechanizmów automatycznej segmentacji (ang. *Clustering*) dostępnych w bibliotece MLlib opartych na algorytmach K-means, Gaussian mixture czy Power Iteration Clustering (PIC). Jednakże z uwagi na ograniczone ramy artykułu wyniki tych prac nie zostały zaprezentowane w tej publikacji.

Podsumowanie

W artykule został przedstawiony problem eksploracji danych pochodzących z bazy transakcji sklepu internetowego w oparciu o zaproponowaną metodykę eksploracji danych wykorzystaną w projekcie RTOM. Problem biznesowy oraz aspekty implementacyjne zostały zaprezentowane w oparciu o algorytmy reguł asocjacyjnych FPGrowth oraz PrefixSpan dostępne w bibliotece MLlib silnika Spark. Zaprezentowane przykłady pochodzą z rzeczywistej platformy handlu elektronicznego, jednak na potrzeby artykułu dane zostały zanonimizowane, tak aby nie zdradzać szczegółów biznesowych przedsięwzięcia internetowego. Sam proces eksploracji przyniósł wartościowe wyniki dla menedżerów marketingu o wzorcach zachowań klientów, które do tej pory nie były odkryte. Dzięki rozbudowaniu procesu eksploracji i poddaniu tym samym działaniom różnych segmentów klientów wskazano specyfikę postępowania klientów należących do różnych segmentów, co przyczyniło się do skuteczniejszego przygotowania tak zwanych kampanii celowanych (ang. *targeted campaigns*).

Literatura

1. Chorianopoulos A. (2016), *Effective CRM Using Predictive Analytics*, John Wiley & Sons, Hoboken.
2. Han J., Fu Y. (1999), *Mining Multi Level Association Rules in Large Databases*, „IEEE Knowledge and Data Engineering", Vol. 11, s. 798-805.
3. Han J., Kamber M., Pei J. (2012), *Data Mining: Concepts and Techniques*, Elsevier.
4. Han J., Pei J., Yin Y. (2000), *Mining Frequent Patterns without Candidate Generation*, „ACM SIGMOD Record", Vol. 29, Issue 2, s. 1-12.
5. Haoyuan L., Wang Y., Zhang D., Zhang M., Chang E.Y. (2008), *PFP: Parallel FP-Growth for Query Recommendation*, „RecSys '08 Proceedings of the 2008 ACM Conference on Recommender Systems", October 23-25, s. 107-114.
6. Korczak J., Pondel M. (2017), *Metodyczne podejście do analizy i eksploracji danych marketingowych*, Kongres Informatyki Ekonomicznej, Poznań. (W druku).
7. Linoff G.S., Berry M.J.A. (2011), *Data Mining Techniques: for Marketing, Sales, and Customer Relationship*, Wiley Publishing, Indianapolis.
8. Morzy T. (2013), *Eksploracja danych. Metody i algorytmy*, Wydawnictwo Naukowe PWN, Warszawa.
9. Pawełoszek I., Korczak J. (2017), *From Data Exploration to Semantic Model of Customer*, IntelliSys, London. (W druku).
10. Pei J., Han J., Mortazavi-Asl B., Wang J., Pinto H., Chen Q., Dayal U., Hsu M. (2004), *Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach*, „IEEE Transactions on Knowledge and Data Engineering", Vol. 16(10), s. 1-17.
11. Piatetsky G. (2014), *CRISP-DM, Still The Top Methodology for Analytics, Data Mining, or Data Science Projects*, KDD News, http://www.kdnuggets.com/2014/10/crisp-dm-topmethodology-analytics-data-mining-data-scienceprojects.html (dostęp: 15.01.2017).
12. Pondel M. (2011), *Data Mining with Microsoft SQL Server 2008*, „Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu", nr 232, s. 98-107.
13. Pondel M. (2015), *A Concept of Enterprise Big Data and BI Workflow Driven Platform*, Federated Conference on Computer Science and Information Systems (FedCSIS), September 13-15, Łódź.
14. Pondel J., Pondel M. (2011), *Eksploracja danych w systemach e-commerce*, „Prace Naukowe / Uniwersytet Ekonomiczny w Katowicach": *Systemy wspomagania organizacji SWO 2011*, s. 212-223.
15. Schutt R., O'Neil C. (2013), *Doing Data Science: Straight Talk from the Frontline*, O'Reilly Media.
16. Setia S., Jyoti D. (2013), *Multi-Level Association Rule Mining: A Review*, „International Journal of Computer Trends and Technology (IJCTT)", Vol. 6, No. 3, s. 166-170.
17. Weichbrodth P. (2009), *Odkrywanie reguł asocjacyjnych z transakcyjnych baz danych*, „Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu. Informatyka Ekonomiczna", t. 14, nr 82, s. 301-309.
18. Witten I., Frank E., Hall M., Pal C. (2017), *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann.

EXPLORATION OF THE ONLINE STORE TRANSACTIONAL DATA

Abstract: With the ever-growing e-Commerce market, online shop owners and Omnichannel merchants need to find a competitive advantages not only at the lowest prices of services and products but also in other areas. In-depth knowledge of the needs and habits of customers and their purchase preferences will help to deliver a tailor-made offers. This enables to increase customer loyalty, which ultimately will have a positive impact on the profitability of our trading activities. Data exploration is a process by which we can better understand our customers. This paper focuses on the problems we can solve using Data Mining mechanisms. It shows the methodology adopted in the RTOM project and it shows examples of processes in Orange software and Hadoop platform using the Spark engine and the MLlib library.

Keywords: association rules, data exploration, Hadoop, MLlib, Orange platform, Spark