Marcin Mazurek, Łukasz Walkiewicz

Wydział Cybernetyki
Wojskowa Akademia Techniczna w Warszawie

Wykorzystanie języka PMML w systemach wspomagania decyzji medycznych

1. Wstęp

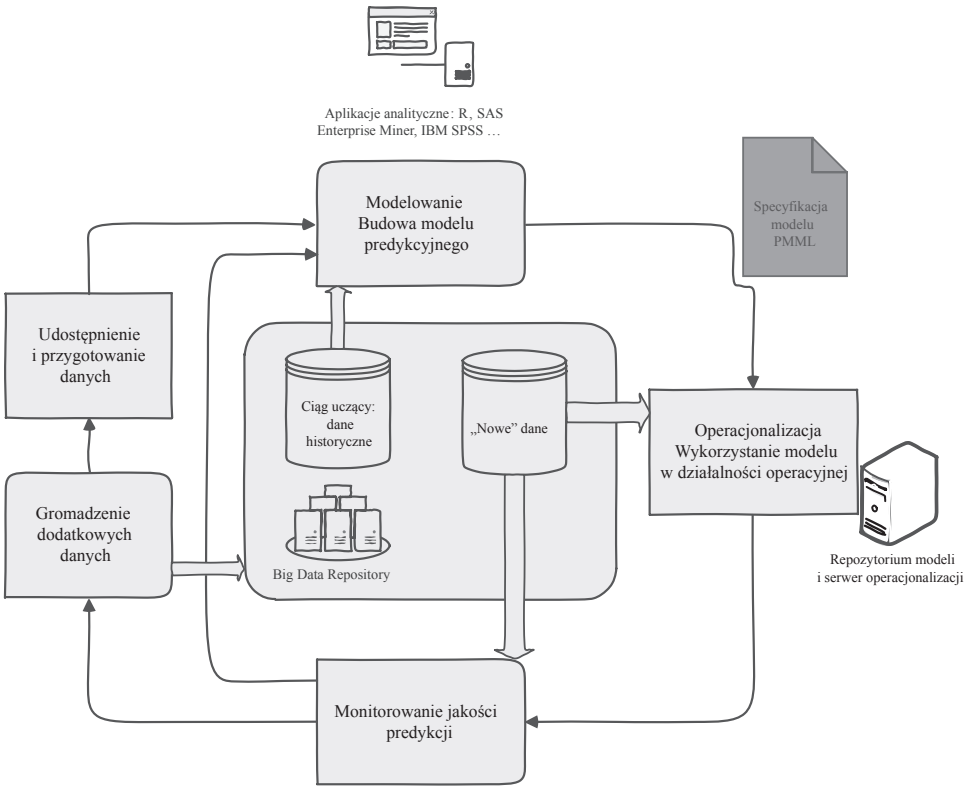
Modele predykcyjne budowane na podstawie zgromadzonych danych opisujących przypadki medyczne mogą być istotnym wsparciem diagnostyki medycznej. Historyczne dane mogą stanowić zasób, który posłuży do wyodrębnienia wiedzy dotyczącej reguł diagnozowania czy najskuteczniejszych sposobów terapii.

Repozytoria *Big Data* stwarzają możliwość gromadzenia i przetwarzania dużych wolumenów danych. Większa liczba rekordów może prowadzić do wypracowania modeli predykcyjnych, które będą cechowały się większą dokładnością, jednak wiąże się to jednocześnie z większą złożonością architektury systemu.

Cykl życia modelu predykcyjnego budowanego na podstawie danych zgromadzonych w repozytoriach *Big Data* składa się z następujących elementów (rysunek 1):

- przygotowanie i udostępnienie danych;
- modelowanie – budowa modeli predykcyjnych, szacowanie ich parametrów, ocena dokładności predykcji;
- operacjonalizacja – wdrożenie satysfakcjonującego modelu w środowisku aplikacyjnym, umożliwiającym jego wykorzystanie w codziennej praktyce personelu medycznego; możemy wyróżnić dwa scenariusze wykorzystania modelu:
  - wsadowe – model jest wykorzystywany do wyznaczenia nieznanej wartości zmiennej celu dla dużej liczby rekordów zgromadzonych w repozytorium, np. wytypowanie w populacji osób podatnych na ryzyko zachorowania,
  - interaktywe – gdy procedura kalkulacji jest przeprowadzana dla pojedynczego rekordu; przykładem może być postawienie diagnozy podczas wizyty pacjenta u lekarza specjalisty;
- monitorowanie jakości predykcji – w przypadku rekordów, dla których model oszacował nieznaną wartość zmiennej, konfrontujemy ją z rzeczywistymi wartościami dostępnymi po upływie pewnego czasu; działanie to ma charakter szacowania

błędów prognozy *a posteriori* i może prowadzić do wycofania modelu w sytuacji, gdy dokładność jego predykcji jest niezadowalająca; może to mieć miejsce, gdy zmienia się natura choroby bądź objawów; wówczas należy przystąpić ponownie do procesu konstrukcji modelu, opcjonalnie rozszerzając zakres informacyjny danych;
- gromadzenie danych – jeżeli prognozy modelu wykazują dużą rozbieżność, przyczyną może być brak w ciągu uczącym istotnych atrybutów przypadku medycznego; wówczas należy rozszerzyć model danych o nowe atrybuty i sprawdzić, czy zwiększają one jakość predykcji.



Rysunek 1. Cykl życia modelu predykcyjnego

Źródło: opracowanie własne.

W przedstawionym procesie zbudowane modele predykcyjne są wdrażane na serwerze, którego zadaniem jest wyznaczenie prognozy bądź zaklasyfikowanie nowych obserwacji. Przenoszenie modeli odbywa się z wykorzystaniem standardu specyfikacji modeli eksploracji danych (*Predictive Modelling Markup Language* – PMML), który jest otwartym, niezależnym od platformy standardem opisu modeli predykcyjnych.

Język ten służy do zapisu zbudowanego modelu tak, aby mógł on zostać uruchomiony na danych niezależnie od miejsca i próby danych, na których powstał.

Wykorzystanie tego standardu umożliwia odseparowanie technologiczne środowiska budowy modelu oraz środowiska jego operacyjnego wykorzystania. Dzięki temu naukowcy, inżynierowie danych i analitycy opracowujący modele predykcyjne mogą posługiwać się najwygodniejszymi dla siebie narzędziami. W praktyce jedyny element łączący środowisko budowy modelu oraz jego operacjonalizacji to nazwy zmiennych wykorzystywanych w modelu.

W niniejszym artykule zostało opisane środowisko analiz predykcyjnych dużych wolumenów danych, które umożliwia klasyfikację przypadków medycznych zgromadzonych w repozytorium Hadoop. Wykorzystano w nim język PMML do opisu uproszczonego modelu predykcyjnego. Poniżej przedstawiono najważniejsze elementy opisu specyfikacji PMML, następnie omówiono komponenty środowiska *Big Data* wykorzystane w procesie budowy systemu. W dalszej kolejności opisano konstrukcję środowiska oraz sformułowano wnioski.

2. Język PMML

Język PMML jest niezależnym standardem zapisu modeli eksploracji danych, rozwijanym przez Data Mining Group[1]. Pozwala na przenoszenie definicji procedur eksploracji danych pomiędzy środowiskami pochodzącymi od różnych dostawców, a w konsekwencji umożliwia całkowite odseparowanie środowiska budowy modelu oraz jego operacjonalizacji. Zapewnia aplikacjom możliwość definiowania modeli w sposób niezależny od producenta oprogramowania tak, aby wyeliminować kwestie licencyjne i niekompatybilność formatów modeli różnych aplikacji. Pozwala na stworzenie modelu przy wykorzystaniu oprogramowania jednego producenta, a następnie na dokonanie analizy, oceny lub wizualizacji stworzonego modelu za pomocą oprogramowania oferowanego przez innego producenta. Najważniejszymi komponentami opisu modelu predykcyjnego są (w nawiasach podano źródłowe nazwy elementów ze schematu):

- nagłówek (Header) – podstawowy opis modelu, informacje o autorze;
- słownik danych (DataDictionary) – lista zmiennych wykorzystywanych w modelu, zawierająca typ danych zmiennej, zakres dopuszczalnych wartości, sposób zakodowania brakujących wartości;

[1] http://www.dmg.org.

- transformacje danych (TransformationDictionary) – lista sparametryzowanych operacji przetwarzania zbioru wejściowego do postaci wymaganej przez model; wśród zdefiniowanych przez PMML transformacji są:
  - normalizacja,
  - dyskretyzacja zmiennych,
  - odwzorowanie wartości,
  - transformacja tekstów do macierzy częstości występowania słów,
  - agregacja;
- model – definiuje model eksploracji danych; aktualna wersja standardu PMML (4.2) umożliwia zapis następujących modeli:
  - modelu asocjacji (AssociationModel) oraz analizy sekwencji (SequenceModel),
  - modelu grupowania (ClusteringModel),
  - modeli regresji (GeneralRegressionModel),
  - naiwnego klasyfikatora Bayesa (NaiveBayesModel),
  - modelu najbliższych sąsiadów kNN (NearestNeighborModel),
  - sieci neuronowych (NeuralNeworks),
  - metody wektorów nośnych (SupportVectorMachineModel),
  - szeregów czasowych (TimeSeriesModel),
  - drzew decyzyjnych (TreeModel).

Każdy model ma właściwe dla siebie elementu opisu, stanowiące część standardu. Przykład pliku PMML dla modelu regresji został przedstawiony na rysunku 2.

```
<PMML xmlns="http://www.dmg.org/PMML-4_2" version="4.2">
<Header copyright="DMG.org"/>
<DataDictionary numberOfFields="3">
<DataField name="x1" optype="continuous" dataType="double"/>
<DataField name="x2" optype="continuous" dataType="double"/>
<DataField name="y" optype="continuous" dataType="double"/>
</DataDictionary>
<RegressionModel functionName="regression" modelName="Sample for stepwise
polynomial regression" algorithmName="stepwisePolynomialRegression"
normalizationMethod="softmax" targetFieldName="y">
<MiningSchema>
<MiningField name="x1"/>
<MiningField name="x2"/>
<MiningField name="y" usageType="target"/>
</MiningSchema>
<RegressionTable targetCategory="no" intercept="125.566018">
<NumericPredictor name="x1" coefficient="-28.6617384"/>
<NumericPredictor name="x2" coefficient="-20.42027426"/>
</RegressionTable>
<RegressionTable targetCategory="yes" intercept="0"/>
</RegressionModel>
</PMML>
```

Rysunek 2. Przykład specyfikacji modelu regresji liniowej w języku PMML

Źródło: http://www.dmg.org.

Przedstawiony przykład zawiera opis równania regresji logistycznej z dwoma parametrami odpowiadającym zmiennym wejściowych – *x1* i x^2 – oraz wyrazem wolnym.

Narzędzia eksploracji danych mogą być elementami zarówno budującymi PMML, jak i konsumującymi model zapisany w PMML. Należy zauważyć, że zakres modeli eksploracji danych możliwych do wyspecyfikowania z zastosowaniem języka PMML często wykracza poza możliwości jego interpretacji przez narzędzia dostawców. Wybrane narzędzia obsługują podzbiór modeli możliwych do zdefiniowania w PMML[2].

3. Budowa modeli predykcyjnych w środowisku *Big Data*

Wiodącą technologią gromadzenia i przetwarzania dużych wolumenów danych jest Apache Hadoop[3]. Poniżej zostanie przedstawiony przegląd narzędzi, które umożliwiają implementację procedur eksploracji danych zgromadzonych w środowisku Apache Hadoop oraz wykorzystanie zbudowanych modeli predykcyjnych do klasyfikacji bądź estymacji nowych obserwacji w operacyjnej działalności (operacjonalizacja modelu)[4].

3.1. R

Pakiet obliczeń statystycznych R[5] udostępnia bogatą bibliotekę algorytmów uczenia maszynowego, stanowiąc w ten sposób alternatywę dla komercyjnych rozwiązań. Podstawowym interfejsem pracy analityka danych może być zintegrowane środowisko programistyczne RStudio[6]. Głównym ograniczeniem platformy jest możliwość przetwarzania danych o rozmiarze mieszczącym się w pamięci operacyjnej serwera bądź stacji roboczej. Ograniczenie to można wyeliminować, przenosząc przetwarzanie danych na serwery dostępne w chmurze bądź wykorzystując komercyjne rozszerzenia pakietu REvolution Analytics[7].

---

2 Specyfikacja obsługiwanych modeli: http://www.dmg.org/products.html.

3 http://hadoop.apache.org.

4 A. Bifet, W. Fan, *Mining Big Data: Current Status and Forecast to the Future*, „SIGKDD Explorations" 2013, vol. 14, issue 2, s. 1–5.

5 http://www.r-project.org.

6 http://www.rstudio.com.

7 http://www.revolutionanalytics.com.

3.2. Apache Hive

Apache Hive jest oprogramowaniem hurtowni danych ułatwiającym tworzenie zapytań oraz zarządzanie dużymi zbiorami danych znajdującymi się w rozproszonym systemie Hadoop Distributed File System, ukrywającym przed programistą szczegóły związane z rozproszeniem obliczeń na węzłach[8]. Apache Hive dostarcza deklaratywny język definiowania zapytań Query Language (QL), składniowo podobny do SQL. Apache Hive nie pozwala na wykonanie z jego pomocą operacji aktualizacji lub usuwania danych. Podczas wykonania zapytania zachodzi translacja kodu QL na zadania MapReduce. Z tego powodu Hive jest używany przede wszystkim do wsadowego przetwarzania dużych zbiorów danych.

3.3. Apache Mahout

Mahout jest to mająca otwarty kod źródłowy, napisana w języku Java biblioteka służąca do uczenia maszynowego i eksploracji danych. Została stworzona przez fundację Apache w celu udostępnienia darmowej implementacji wielu algorytmów uczenia maszynowego oraz eksploracji danych. Duża część spośród wspieranych algorytmów ma implementację pozwalającą na uruchomienie na pojedynczym komputerze, jak również na platformie Apache Hadoop. Biblioteka jest rozwijana głównie w obszarach odpowiedzialnych za rekomendację, klasteryzację oraz klasyfikację. W chwili obecnej algorytmy, które są wraz z nią dostarczane i mogą zostać użyte w środowisku *Big Data*, to[9]:

- algorytmy silników rekomendacyjnych (Item-Based Collaborative Filtering i inne),
- algorytmy wykorzystywane w klasyfikacji:
  - naiwny klasyfikator Bayesa,
  - lasy losowe,
  - regresja logistyczna;
- algorytmy wykorzystywane w klasteryzacji oparte na algorytmie grupowania k-średnich.

Na powyższej liście brakuje wielu algorytmów uczenia maszynowego. Biblioteka ta jest jednak ciągle rozwijana i braki w algorytmach mogą zostać w niedalekiej przyszłości uzupełnione (regresja logistyczna została dodana w 2014 r.).

Na podkreślenie zasługuje fakt, że Mahout umożliwia budowę modeli bezpośrednio w środowisku *Big Data*, bez konieczności przenoszenia danych do narzędzia analitycznego. Umożliwia to wykorzystanie w roli ciągu uczącego wolumenów danych, których

[8] E. Capriolo, D. Wampler, J. Rutherglen, *Programming Hive*, O'Reilly Media Inc., Cambridge 2012.
[9] http://mahout.apache.org.

przetworzenie przekracza możliwości pojedynczego serwera. W takim scenariuszu rozproszone przetwarzanie na wielu węzłach, pomimo nakładu związanego z podziałem zadań oraz scaleniem wyników, znacząco przyspiesza proces eksploracji danych w porównaniu z analogiczną operacją na pojedynczym komputerze.

3.4. Cascading Pattern

Cascading Pattern jest rozszerzeniem platformy Cascading[10], które udostępnia funkcjonalność translacji języka PMML na zadania MapReduce platformy Apache Hadoop. Wspierane przez bibliotekę metody eksploracji obejmują:

- klasteryzację metodą hierarchiczną;
- klasteryzację metodą k-średnich;
- regresję liniową;
- regresję logistyczną,
- lasy losowe.

Przedstawione powyżej technologie pokazują, że trendy rozwojowe technologii *Big Data* są wyznaczane przez oprogramowanie *Open Source*. Otwartość platform sprzyja szybkiemu rozwojowi funkcjonalności narzędzi, towarzyszy temu jednak mnogość standardów i interfejsów. Architektura analitycznych systemów *Big Data* musi być weryfikowana praktycznie w postaci prototypowych systemów potwierdzających przyjęte założenia. Przykładem takiej implementacji jest opisany poniżej system analityczny wspomagający diagnostykę medyczną.

4. Opis implementacji

Celem implementacji było stworzenie środowiska eksploracji dużego wolumenu danych oraz ewaluacja funkcjonalności dostępnych narzędzi analitycznych w kontekście repozytorium *Big Data*. Wykorzystano dane publikowane przez National Cancer Institute[11] w ramach programu SEER (*Surveillance, Epidemiology, and End Results*)[12]. Są to rutynowo zbierane informację o:

- demografii pacjentów;
- pierwotnym nowotworze;

[10] http://www.cascading.org.
[11] http://www.cancer.gov/aboutnci.
[12] http://seer.cancer.gov/data.

- morfologii guza;
- stopniu rozwoju guza w momencie diagnozy;
- pierwszym kursie leczenia;
- obserwowanym stanie zdrowia.

Stworzony prototyp środowiska analitycznego wspiera diagnostykę medyczną przez:

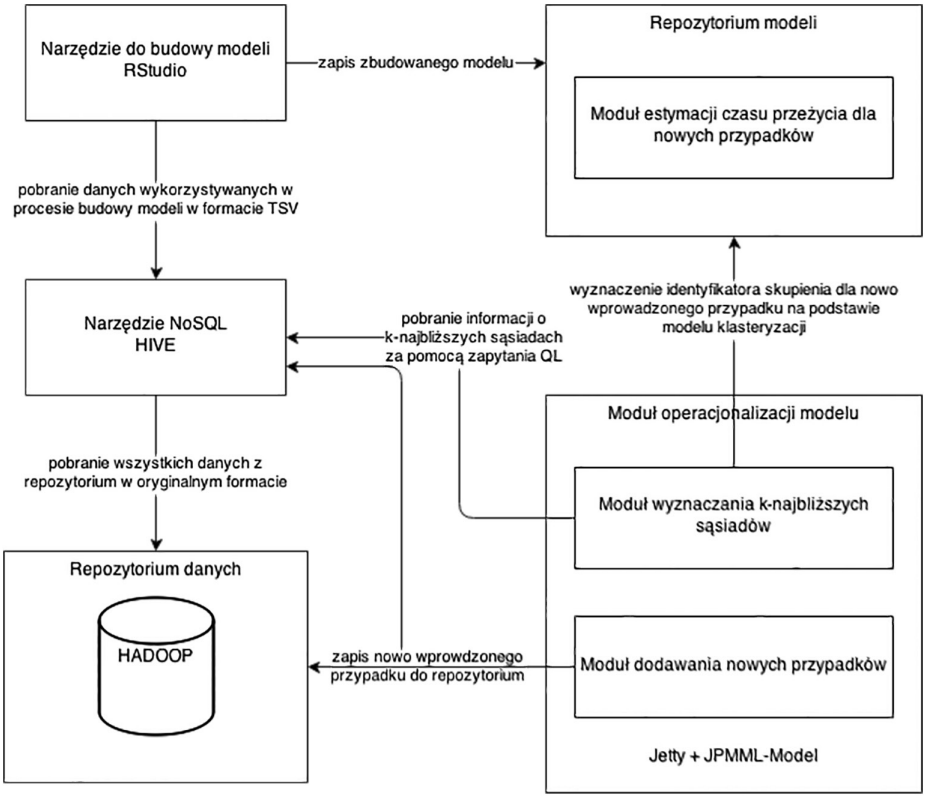
- pogrupowanie pacjentów i przedstawienie typowych scenariuszy choroby;
- zbudowanie modelu prognozującego pozostałą długość życia;
- wskazanie podobnych przypadków dla nowych obserwacji, przez wskazanie k-najbliższych przypadków medycznych.

Należy zaznaczyć, że zbudowane modele analityczne nie były w żaden sposób weryfikowane pod kątem jakości wyników. Przyjęte zagadnienia predykcyjne nie muszą również odzwierciedlać rzeczywistych problemów diagnostyki klinicznej w tym obszarze. Zbudowane modele predykcyjne posłużyły jako scenariusze testowe dla potwierdzenia poprawności konstruowanej architektury systemu.

4.1. Komponenty środowiska

W skład środowiska wchodzą następujące komponenty, przedstawione schematycznie na rysunku 3:

- repozytorium danych Hadoop i narzędzia NoSQL to elementy architektury odpowiedzialne za przechowywanie, integrację oraz selekcję danych wykorzystywanych w procesie eksploracji danych;
- narzędzie do budowy modeli R wraz ze zintegrowanym środowiskiem deweloperskim RStudio;
- repozytorium modeli eksploracji danych, w którym są przechowywane zbudowane modele wraz z ocenami trafności predykcji;
- moduł operacjonalizacji modelu, uruchamiający procedurę obliczeniową opisaną przez model predykcyjny dla nowych rekordów.



Rysunek 3. Architektura zbudowanego środowiska eksploracji *Big Data*

Źródło: opracowanie własne

Poniżej znajduje się rozszerzony opis poszczególnych komponentów.

4.2. Środowisko budowy modelu

Kluczowym elementem środowiska konstrukcji modeli predykcyjnych jest sposób odwołania się do danych przechowywanych na serwerze w rozproszonym systemie plików HDFS. Na uwagę zasługują dwa interfejsy dostępu do danych (rysunek 4):

- zapytanie Hive QL, którego wyniki były ładowane bezpośrednio do środowiska RStudio;
- zapytanie QL, którego wyniki były zapisywane w rozproszonym systemie plików, a następnie odczytywane przez środowisko RStudio dzięki wykorzystaniu biblioteki *rhdfs*[13].

[13] https://github.com/RevolutionAnalytics/RHadoop/wiki/rhdfs.

Kod w języku R odpowiedzialny za wczytanie danych z pliku, budowę oraz zapis modelu do pliku PMML został przedstawiony rysunku 4. Umieszczono na nim obydwa sposoby odwołania się do danych (w praktyce uruchamiany jest jeden).

```
library(pmml)
#library(rhdfs)

#odczyt danych #1
incience.all <- rhive.query("SELECT
                                race,
                                sex,
                                age_dx,
                                type,
                                srv_time_mon
                            FROM incidence")
#odczyt danych #2
incidence.all <- hdfs.read.text.file(path='/rinput/rinput.tsv')


#budowa modelu
incidence.fit <-  kmeans(incidence.all, 32)

#zapis modelu
models_repository <- '/home/hadoop/seer/models'
saveXML(pmml(incidence.fit), file=paste(models_repository, "clustering.model.xml", sep="/"))
```

Rysunek 4. Przykład kodu w języku R budującego skupienia

Źródło: opracowanie własne.

W wyniku wykonania powyższego kodu powstaje plik PMML zawierający model klasteryzacji metodą k-średnich, który może być następnie zastosowany do segmentacji innych danych. Specyfikacja modelu jest przenoszona do repozytorium modeli.

4.3. Operacjonalizacja modelu z wykorzystaniem PMML

Zbudowany w środowisku R model został następnie wykorzystany w celu nadania identyfikatorów klastrów istniejącym w repozytorium danych Hadoop rekordom. Do wyznaczenia przynależności nowo wprowadzonego przypadku do klastra została wykorzystana biblioteka JPMML Model[14].

Klasyfikacja nowych obserwacji odbywa się przez przydzielenie obserwacji etykiety klastra oraz wyznaczenie wartości zmiennych celu na podstawie metody k-najbliższych sąsiadów, ograniczonych do obserwacji pochodzących z tego samego klastra. Podejście to znacznie przyspiesza proces porównywania nowo wprowadzonego przypadku do tych istniejących już w bazie.

Środowisko umożliwia dwa tryby uruchamiania procedury predykcyjnej – wsadowo oraz interaktywnie. W tym drugim scenariuszu dla wprowadzania atrybutów przypadku

[14] https://github.com/jpmml.

medycznego została zbudowana aplikacja WWW umożliwiająca uwzględnienie nowego przypadku (rysunek 5).

Rysunek 5. Ekran aplikacji do wprowadzania nowego przypadku

Źródło: opracowanie własne.

Ocena nowych przypadków medycznych została udostępniona również w postaci usługi *Web Service*. Przykładowa struktura zapytania została przedstawiona poniżej:

```
http://91.230.204.41:7070/json?type=8&race=1&sex=
    1&age _ dx=37&models%5B%5D=clustering.
```

Wykorzystywany do oceny przypadku model predykcyjny jest parametrem zapytania, co oznacza, że lekarz uruchamiający moduł wspomagania decyzji może odwołać się do dowolnego modelu przechowywanego w repozytorium modeli w postaci PMML. Dysponując dodatkowymi metadanymi tych modeli, takimi jak trafność prognoz czy historia uruchomień, lekarz diagnosta może potraktować wyniki działania tych modeli jako dodatkowe przesłanki podjęcia eksperckiej decyzji.

5. Podsumowanie i uwagi końcowe

Możliwość budowy modeli eksploracji danych przy wykorzystaniu dużych repozytoriów danych stanowi ogromną szansę poprawy jakości tych modeli oraz zwiększenia

ich wykorzystania w praktyce lekarskiej. Na podstawie dostępnych komponentów *Open Source* został zbudowany prototyp, pozwalający na realizację wszystkich czynności składających się na cykl życia modelu predykcyjnego. Powstałe środowisko stanowi swoisty dowód wykonalności systemu, w którym analitycy używający różnych narzędzi analitycznych mogą współdzielić wyniki swojej pracy i udostępniać je lekarzom. Jednocześnie zaobserwowano ograniczenia architektury, związane głównie z dostępem do danych. Opisane powyżej narzędzia budowy modeli predykcyjnych, mimo odwołania się do danych przechowywanych w HDFS, działają „lokalnie" – przenoszą dane do pamięci operacyjnej węzła analitycznego. Ogranicza to w istotny sposób wolumen danych, jaki może zostać użyty w procesie uczenia pojedynczego modelu. Pożądanym sposobem działania oprogramowania w tym zakresie są biblioteki ukrywające lokalizację danych i „tłumaczące" procedury statystyczne i optymalizacyjne na polecenia wykonywane bezpośrednio w HDFS. W tym kierunku podążają twórcy biblioteki Apache Mahout. Biblioteka ta ma jednak znaczącą wadę – zaawansowane wykorzystanie biblioteki wymaga znajomości języka Java. Migracja użytkowników z takich platform jak RStudio lub SAS wymaga więc od nich nabycia umiejętności programowania w wyżej wymienionym języku. Nie udostępnia ona również kompatybilności z żadną popularną platformą służącą do eksploracji danych, co sprawia, że nie ma możliwości wykorzystania modelu zbudowanego w innym środowisku.

Jako znacznie dojrzalsze należy ocenić technologie operacjonalizacji modeli wykorzystujące specyfikację PMML. Istnieją tu rozwiązania, które można zastosować lokalnie (większość komercyjnych środowisk eksploracji danych może pracować w trybie konsumenta modelu PMML), w rozwiązaniach *Big Data* (opisywane rozwiązanie Cascading Pattern) czy też w chmurze. Przykładem tego ostatniego jest silnik predykcyjny Zementis ADAPA[15]. W przypadku tego komponentu architektury problemem może się okazać zakres interpretowanych modeli eksploracji danych, gdyż rozwój standardu wyprzedza technologię.

Zbudowany prototyp systemu powinien zostać rozszerzony o bazę terminów medycznych bądź szerzej – moduł implementujący ontologię medyczną. Umożliwiłaby ona standaryzację pojęć występujących w części słownika danych standardu PMML, doprowadzając w ten sposób do prawdziwej interoperacyjności modeli predykcyjnych. W kontekście użyteczności systemu informatycznego dla procesu diagnostyki medycznej niezbędna jest również ocena wyników predykcji i dopasowania modelu do danych.

[15] http://zementis.com/products/adapa.

Bibliografia

Bifet A., Fan W., *Mining Big Data: Current Status and Forecast to the Future*, „SIGKDD Explorations", 2013, vol. 14, issue 2, s. 1–5.

Capriolo E., Wampler D., Rutherglen J., *Programming Hive*, O'Reilly Media Inc., Cambridge 2012.

Guazzelli A., Stathatos K., Zeller M., *Efficient Deployment of Predictive Analytics through Open Standards and Cloud Computing*, „ACM SIGKDD Explorations Newsletter" 2009, vol. 11, issue 1, June, s. 32–38.

Lantz B., *Machine Learning with R*, PACKT Publishing, Birmingham 2013.

Mazurek M., *Architektura systemu wspomagania decyzji medycznych wykorzystująca technologię przetwarzania danych Big Data*, „Roczniki" Kolegium Analiz Ekonomicznych, z. 35, Oficyna Wydawnicza SGH, Warszawa 2014.

Savage N., *Better Medicine Through Machine Learning*, „Communications of the ACM" 2012, vol. 55, no. 1, s. 17–19.

White T., *Hadoop: The Definite Guide*, O'Reilly Media Inc., Cambridge 2012.

Źródła sieciowe

https://github.com/jpmml.

https://github.com/RevolutionAnalytics/RHadoop/wiki/rhdfs.

http://hadoop.apache.org.

http://mahout.apache.org.

http://nexr.github.io/RHive.

http://seer.cancer.gov/data.

http://zementis.com/products/adapa.

http://www.cancer.gov/aboutnci.

http://www.cascading.org/projects/pattern.

http://www.dmg.org.

http://www.dmg.org/v4-2-1/GeneralStructure.html.

http://www.revolutionanalytics.com.

http://www.r-project.org.

http://www.rstudio.com.

* * *

The usage of PMML in health care predictive analytics

Summary

The Big Data technology makes it possible to process huge volumes of data which can be utilized to build better predictive models in health care. There are some tools and libraries that support data scientist in Big Data analytics, but they are poorly standardized. As a consequence, any concept of architecture should be proved by means of prototyping. The paper presents the implementation of the Big Data analytical environment, where operationalization of the predictive models is achieved by utilizing the PMML standard. Key elements of the PMML specification are presented along with the open-source components upon which the system is built.

Keywords: Big Data, predictive analytics, PMML