Metody i narzędzia analizy dużych zbiorów danych

**Projekt 2**

*Inteligentny system analizy danych*

*oparty na platformie Apache Spark*

Wykonali:

*Paweł Suchanicz*

*Anna Zybek*

Apache Spark stał się jednym z najczęściej używanych i obsługiwanych narzędzi open-source’owych do uczenia maszynowego i analizy danych.

Tematem projektu było zbudowanie przynajmniej dwóch modeli z wybranego przez siebie problemu zagadnień uczenia maszynowego. Projekt miał być oparty na platformie Apache Spark wykorzystujący strukturę Spark DataFrame.

W naszym projekcie zajęliśmy się Regresją.

1. **Opis problemu regresji**

Analiza regresji jest narzędziem pozwalającym badać zależność między zmiennymi. W klasycznej postaci wymaga się, aby zmienne objaśniane (zależne) były ilościowe. Zmienne, które służą do obliczenia wartości innej zmiennej są nazywane zmiennymi objaśniającymi (niezależnymi).

Jednym z modeli, którym posłużyliśmy się w tym projekcie jest model regresji liniowej. Ten model zakłada, że pomiędzy zmienną zależną, a zmiennymi niezależnymi istnieje jakaś zależność liniowa. Jeśli istnieje taka zależność mamy wówczas model regresji liniowej. W naszym modelu wykorzystaliśmy więcej niż jedną zmienną objaśniającą. Taki model jest nazywany modelem regresji wielorakiej.

Kolejnym modelem regresji był model drzew decyzyjnych. Drzewa decyzyjne są popularną rodziną metod klasyfikacji i regresji Przy ich pomocy w oparciu o pomiary zmiennych objaśniających dany obiekt jest przypisywany do klasy zmiennej zależnej. Drzewa decyzyjny mogą być wykorzystywane zarówno w klasyfikacji jak i regresji, te wykorzystywane w regresji nazywamy drzewami regresyjnymi. Służą one do przewidywania wartości zmiennej docelowej, charakteryzuje je zmienna docelowa typu ciągłego.

1. **Opis danych**

Wybrane przez nas dane opisują informacje statystyczne z aktów urodzenia w USA. Statystyki urodzeń w USA są zbierane przez National Center for Health Statistics.

Dane zawarte są w pliku CSV. Plik waży 508 MB. Wszystkie dane dotyczą urodzeń z roku 2018.

Większość zmiennych charakteryzującą się wartością tekstową została zakodowana numerycznie.

**Struktura pliku CSV (wypisana większość cech)**

|  |  |  |
| --- | --- | --- |
| Kolumna | Opis | Przykład |
| attend | Pracownik odbierający poród [id] | 1 |
| bfacil | Miejsce urodzenia | 1 |
| bmi | Wskaźnik masy ciała (BMI) | 33.3 |
| cig\_0 | Palenie papierosów przed ciążą | 0 |
| dbwt | Waga dziecka (zaraz po urodznieu) [g] | 3785 |
| dlmp\_mm | Ostatni miesiąc miesiączki | 3 |
| dlmp\_yy | Ostatni rok miesiączki | 2017 |
| dmar | Stan cywilny | 1 |
| dob\_mm | Miesiąc urodzenia | 1 |
| dob\_tt | Godzina urodzenia | 1958 |
| dob\_wk | Dzień tygodnia | 3 |
| dob\_yy | Rok urodzenia | 2018 |
| fagecomb | Wiek ojca | 27 |
| feduc | Edukacja ojca | 7 |
| fhispx | Latynoskie pochodzenie ojca | 0 |
| ill\_br | Czas od ostatniego porodu | 24 |
| ilop\_r | Odstęp czasu od ostatniej innej ciąży | 30 |
| imp\_sex | In vitro | N |
| ip\_gon | Rzeżączka | N |
| mager | Wiek matki | 26 |
| meduc | Edukacja matki | 6 |
| mhispx | Latynoskie pochodzenie matki | 0 |
| mm\_aicu | Intensywna terapia | N |
| no\_infec | Brak infekcji | 1 |
| m\_ht\_ln | Wysokość matki [cal] | 63 |
| m\_mmorb | Brak chorób u matki | 1 |
| no\_risks | Brak zgłoszonych czynników ryzyka | 1 |
| precare | Miesiąc rozpoczęcia opieki prenatalnej | 2 |
| previs | Liczba wizyt prenatalnych | 15 |
| priordead | Liczba poronień | 0 |
| priorlive | Liczba urodzonych dzieci | 2 |
| pwgt\_r | Waga kobiety przed ciążą | 126 |
| restatus | Status pobytu | 1 |
| rf\_cesar | Cesarskie cięcie | N |
| sex | Płeć dziecka | M |

Link do pobrania danych: <https://www.kaggle.com/des137/us-births-2018>

Rozmiar danych: 508 MB (3801534 rekordów).

Zbiór powstał na podstawie surowych danych dostępnych na rządowej stronie iinternetowej Centers for Disease Control and Prevencion z USA : <https://www.cdc.gov/nchs/data_access/vitalstatsonline.htm#Tools>

W kolejnych punktach przedstawione zostaną implementacje poszczególnych metod oraz ich wynik

1. **Środowisko**

Cały projekt – zbudowane modele zostały zaprojektowane na platformie DataBricks. Platforma ta automatycznie tworzy kontekst SparkContext (dostępny pod zmienną ***sc)***.

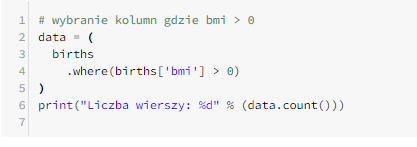
Utworzony został klaster dla wersji Databricks *6.2 ML (includes Apache Spark 2.4.4, Scala 2.11)*

Po przygotowaniu klastra, na którym później pracowaliśmy, należało załadować dane, poprzez załadowanie pliku *csv* na platformę Databricks. Została utworzona struktura Spark DataFrame, przypominająca SQL-ową tabelę. Ustawione zostały odpowiednie typy danych i nazwy kolumn.

1. **Analiza i preprocessing danych**

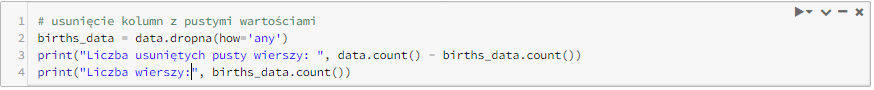


*Rys. 1 Wczytanie struktury DataFrame do zmiennej*

**

*Rys. 2 Wybranie kolumn o wartościach numerycznych*

Przed zbudowanie modeli regresji, potrzebne było odpowiednie przygotowanie danych. Ważnym usunięcie wierszy zawierających puste wartości ze zbioru danych.

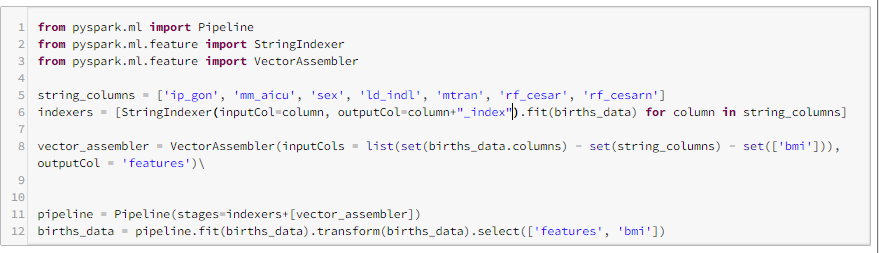
**

*Rys. 3 Usunięcie wartości pustych*

*Liczba usuniętych pusty wierszy: 666965*

*Liczba pozostałych wierszy: 3134569*

Następnie należało zamienić dane tekstowe na liczbowe oraz za pomocą obiektu VectorAssembler wybrać kolumny, które będą odgrywać rolę zmiennych objaśniających w modelu i kolumnę, która będzie badana. Do zbudowania modeli wykorzystane zostały wszystkie zmienne. Zmienną objaśnianą była zmienna **bmi**, a pozostałe zmienne to zmienne objaśniające.



*Rys. 4 Przygotowanie danych do modelu*

Następnie dane podzieliliśmy na dane treningowe i dane testowe.

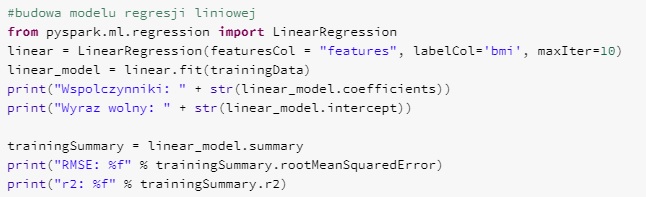


*Rys. 4 Podział na zbiór treningowy i testowy*

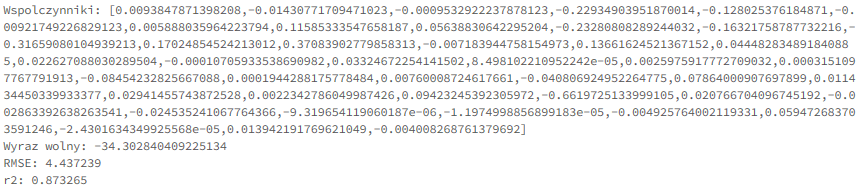
1. **Regresja Liniowa**

Pierwszą metodą, od której zaczęliśmy naszą analizę danych jest regresja liniowa – wieloraka, bo do zbudowania modelu użyliśmy więcej niż jednej zmiennej niezależnej.

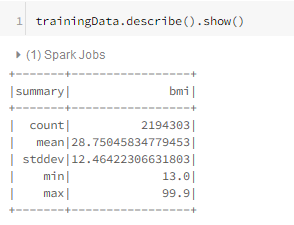
Nasz model regresji wielorakiej został zbudowany za pomocą funkcji *LinearRegression* z biblioteki *pyspark.ml.regression.* Do modelu wykorzystano 70% danych - danych treningowych.



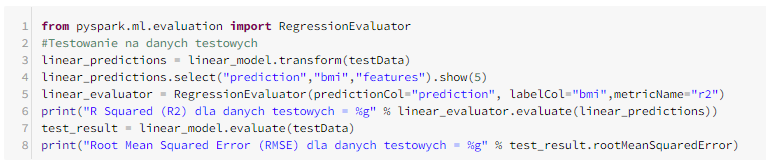
Aby sprawdzić jak nasz model został dopasowany do danych, sprawdziliśmy wartość błędu średniokwadratowego i R2.

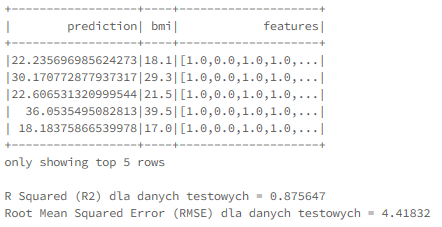


RMSE mierzy różnice między wartościami przewidywanymi przez model a wartościami rzeczywistymi. Dla naszego modelu błąd ten wynosi około. 4.62. Wynik ten nie jest zbyt duży, możemy sądzić, że nasz model jest dobrze dopasowany, zwłaszcza gdy zestawimy ten błąd na tle wartości rzeczywistych bmi takich jak średnia, wartość minimalna i maksymalna.



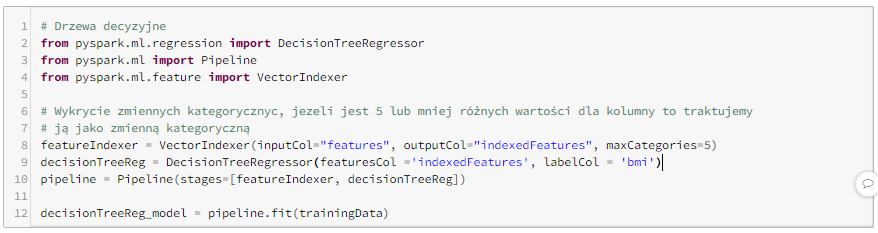
Przetestowaliśmy nasz model na danych testowych (30% wszystkich danych). Poniżej zamieszczamy wyniki predykcji dla 5 obserwacji i otrzymany błąd średniokwadratowy modelu dla danych testowych:



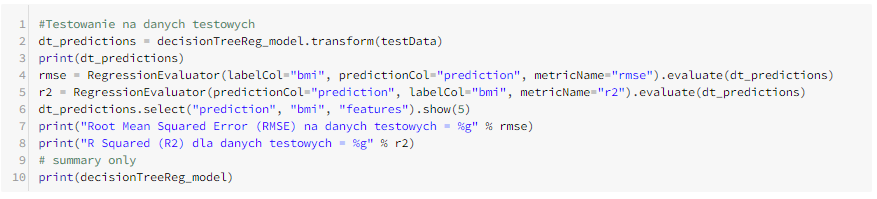


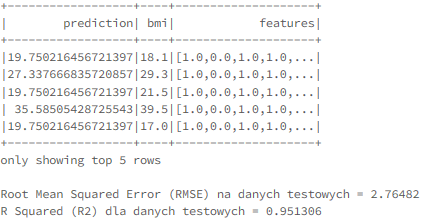
1. **Regresja za pomocą drzew decyzyjnych**

Kolejną metodą, była regresja z wykorzystaniem drzew decyzyjnych. Do zbudowania tego modelu regresji wykorzystana została klasa DecisionTreeRegressor z pakietu *pyspark.ml.regression.* Skorzystano z wcześniej już przygotowanych i podzielonych na zbiory treningowe i testowe danych - tych samych co przypadku regresji liniowej.



Przetestowaliśmy nasz model na danych testowych (30% wszystkich danych). Poniżej zamieszczamy wyniki predykcji dla 5 obserwacji i otrzymany błąd średniokwadratowy:





Jak widać powyżej współczynnik R squared jest większy niż przy regresji liniowej, RMSE jest mniejsze, więc drzewa decyzyjne były lepszym wyborem dla naszych danych niż regresja liniowa.