Филип Самарџиски 211097

Домашна задача 3

Анализа на податоци и машинско учење со Apache Spark и Kafka

Во оваа домашна задача обработуваме податоци од множеството *Diabetes Health Indicators Dataset* и извршуваме различни машински модели за. Дополнително, користиме Apache Kafka за обработка на податоци во реално време.

Најпрво, податоците ги вчитуваме со помош на Apache Spark, меѓутоа бидејќи сакаме да ги трансформираме податоците, го користиме модулот pandas преку кој исто така ги вчитуваме податоците. Податочното множество *diabetes_binary_health_indicators_BRFSS2015.csv* го делиме на две подмножества:

- **offline.csv** (80%) – се користи за тренирање на моделите во офлајн фазата.
- **online.csv** (20%) – се користи за реално-временска обработка во онлајн фазата.

При поделбата се задржува соодносот на класите користејќи **train_test_split** од sklearn. Овие поделени множества понатаму ги делиме на уште две множества, едно без колоната за класата што се предвидува и едно со само таа колона. Потоа се извршуваат трансформации на податоците, односно нормализација со RobustScaler со цел да нема преголеми отстапки од еден до друг податок, за моделот да не дава поголемо значење на едни во однос на други податоци без вистинска причина.

Следната фаза е тренирањето на три различни модели за класификација со различни хиперпараметри: XGBoost Classifier, Random Forest Classifier, Logistic Regression. Правиме избор на најдобриот модел според **F1-score**, користејќи **GridSearchCV** и **cross-validation (K-fold)**. Најдобриот модел се зачувува со помош на **joblib** што понатаму ни овозможува повторно да го користиме моделот со цел да предвидуваме класа од новите податоци кои ќе доаѓаат во реално време. Исто така се зачувува и скалерот кој сме го тренирале на offline(train) податоците.

Понатаму во produce_messages.py правиме Producer, чијашто цел е да чита податоци од **online.csv** податочното множество и да ги испраќа ред по ред податоците во Kafka topic-от **health_data**, без класата којашто се предвидува.

Во produce_predictions.py се читаат испратените податоци од Kafka topic-от **health_data**. Вчитувањето се прави во stream(тек на податоци) со помош на Apache Spark, преку SparkSession. Се дефинира шемата на податоците според колоните во податочното множество. Потоа во формат на **Unbounded Data Frame** од модулот **pandas**, се извршуваат истите трансформации како во офлајн фазата, односно тие се скалираат со зачуваниот и истриниран скалер, а потоа се користи и зачуваниот и истриниран модел од машинско учење за предвидување на класата. Записите се збогатуваат со предвидената класа и вака обработените податоци се испраќаат во нов Kafka topic **health_data_predicted**. Начинот на

запишување на податоците е append, којшто ги запишува само новите податоци кои доаѓаат со текот на времето.

На крај во consume_messages.py имаме Consumer, кој ги чита и прикажува новите податоци заедно со предвидената класа од topic-от **health_data_predicted.**

Со целиот овој процес, симулираме вистинско сценарио каде што најпрво треба да избереме најдобар модел за машинско учење од неколку опции и со различни хиперпараметри, кој потоа би го користеле за предвидувања над податоци кои пристигнуваат во реално време.