Assignment 3: Predictive Analysis using Spark

Team Members:

Swathi Ravindran(50314300)

Balagopal Madhukumar(50320449)

Introduction:

In this assignment, we will predict the genre of movies based on their plot using multi class logistic regression. We have to implement the movie genre prediction model on Apache Spark.

We are given a training dataset containing columns corresponding to movie_id, movie_name, plot and genre. We have to create a model that can predict the movie genre associated with the movie from a test dataset containing the movie_id, movie_name and plot columns.

Part 1: Basic Model:

- We start by creating a spark dataframe from the training, test and mapping csvs given to us.
- To do so, we use the pandas library to read the data from the csvs and then we preprocess the data.
- First, we apply the regex tokenizer on train and test data, which takes text from the plot column and breaks it down into individual word tokens.
- Then we remove stop words from the train and test data using the StopWordsRemover.
- Next, we use CountVectorizer to create the term document matrix and apply all the transformations to the train data and the test data.
- After that we wrote a function that would replace the genre column with 1 if the corresponding genre is tagged to a movie.
- We then update the dataframe with the mapping values of 1 or 0 for the presence or absence of a movie.
- Next we create 20 logistic regression models, that is one model each for each genre and training them to predict which plot belongs to what all genres.
- In the final output, we displayed the movie_id and its corresponding prediction.
- Part 1- F1 Score: 1

Part 2: Using TF-IDF to improve the model:

- Here too, We start by creating a spark dataframe from the training, test and mapping csvs given to us.
- We use the pandas library to read the data from the csvs and then we preprocess the data.
- First, we apply the regex tokenizer on train and test data, which takes text from the plot column and breaks it down into individual word tokens.
- Then we remove stop words from the train and test data using the StopWordsRemover.
- Next, we use CountVectorizer to create the term document matrix and apply all the transformations to the train data and the test data.
- The output of the count vectorizer we give to the input of the IDF.
- We then update the dataframe with the mapping values of 1 or 0 for the presence or absence of a movie.
- Next we create 20 logistic regression models, that is one model each for each genre and training them to predict which plot belongs to what all genres.
- In the final output, we displayed the movie_id and its corresponding prediction.
- Part 2 F1 score: 1

Part 3: Word to Vector:

- Here too, We start by creating a spark dataframe from the training, test and mapping csvs given to us.
- We use the pandas library to read the data from the csvs and then we preprocess the data.
- First, we apply the regex tokenizer on train and test data, which takes text from the plot column and breaks it down into individual word tokens.
- Then we remove stop words from the train and test data using the StopWordsRemover.
- We will then transform our train and test data using Word2Vec estimator. It takes sequences of words representing models and trains a Word2Vec model. The model then maps each word to a unique fixed size vector.
- We then update the dataframe with the mapping values of 1 or 0 for the presence or absence of a movie.

- Next we create 20 logistic regression models, that is one model each for each genre and training them to predict which plot belongs to what all genres.
- In the final output, we displayed the movie_id and its corresponding prediction.
- Part 3 F1 score: 1

Reference:

- https://spark.apache.org/docs/2.1.0/ml-features.html
- Lecture videos