Predictive Analytics with Spark

Shanmukha Haripriya Kota	Ravikiran Sunnam	Uneeth Akula
Department of Computer Science	Department of Computer Science	Department of Computer Science
University at Buffalo, NY	University at Buffalo, NY	University at Buffalo, NY
skota@buffalo.edu	rsunnam@buffalo.edu	uneethak@buffalo.edu
UBIT: skota	UBIT: rsunnam	UBIT: uneethak

Abstract

The objective of this assignment is to perform predictive Analytics with Apache Spark. The goals of the assignment are to use Spark Libraries to implement an end to end Predictive Analytics Pipeline. Dataset used is Movie Genre Prediction.

Kaggle Team Name:

I am Groot

Setup and Initialization:

- JAVA Version in the VM has been changed to 1.8

1 Term-document matrix

Objective: To create a machine learning model in spark to use the information provided in the train set to predict the genres associated with a movie.

Data Processing:

- The input data file in CSV format is read using the Pandas library initially as a part of preprocessing.
- Then the pandas DataFrame is converted to Spark DataFrame using SQLContext.
- Tokenizer API in pyspark.ml is used to tokenize the plot in the input data.
- RegexTokenizer API from pyspark.ml is used to remove all punctuations and then tokenize the plot column in the input.
- StopWordsRemover API from pyspark.ml is used to remove stop words after tokenizing the plot.
- CountVectorizer API from pyspark.ml is used to create a feature vector from the processed plot column in the data frame.
- For the labels, i.e., genre, the genre column in the DataFrame is split to create a separate column for every genre.
- If the movie belongs to that genre, it is given as 1 else given as 0.

- Thus modified feature vector is used as an input to 20 different machine leaning models. Each model performs a binary classification, which says if that particular movie belongs to that particular genre or not.

Machine Learning:

- Machine Learning model used in this assignment is LogisticRegression from pyspark.ml.
- Logistic Regression is selected because for performing a binary classification, it gave a better performance.
- Hyperparameters used for performance tuning are: maximum number of iterations in Logistic Regression, Vocabulary size in CountVectorizer.

Output:

- The predictions from each individual model are then concatenated to create an output predictions DataFrame.
- The output DataFrame is then joined with initial test DataFrame to create the output file with movie_id and predictions as output.
- The modified output DataFrame is then converted to CSV using Pandas.

Result:

- Score received in Kaggle: 0.94197

Fig 1: Kaggle Score for Task 1

2 TF-IDF to improve the model

Objective: To improve the performance of the previous model using TF-IDF score.

Data Processing:

- The input data file in CSV format is read using the Pandas library initially as a part of preprocessing.
- Then the pandas DataFrame is converted to Spark DataFrame using SQLContext.
- Tokenizer API in pyspark.ml is used to tokenize the plot in the input data.
- RegexTokenizer API from pyspark.ml is used to remove all punctuations and then tokenize the plot column in the input.
- StopWordsRemover API from pyspark.ml is used to remove stop words after tokenizing the plot.
- CountVectorizer API from pyspark.ml is used to create a feature vector from the processed plot column in the data frame.
- This modified DataFrame is then used as an input to normalize the scores using Inverse Document Frequency (IDF)
- IDF API from pyspark.ml is used to create a feature vector that is based on TF-IDF score. This is given as an input to ML model.

- For the labels, i.e., genre, the genre column in the DataFrame is split to create a separate column for every genre.
- If the movie belongs to that genre, it is given as 1 else given as 0.
- Thus modified feature vector is used as an input to 20 different machine leaning models. Each model performs a binary classification, which says if that particular movie belongs to that particular genre or not.

ML:

- Same model that has been used for task1 has been used.
- Machine Learning model used in this assignment is LogisticRegression from pyspark.ml.
- Logistic Regression is selected because for performing a binary classification, it gave a better performance.
- Hyperparameters used for performance tuning are: maximum number of iterations in Logistic Regression, Vocabulary size in CountVectorizer.

Output:

- The predictions from each individual model are then concatenated to create an output predictions DataFrame.
- The output DataFrame is then joined with initial test DataFrame to create the output file with movie_id and predictions as output.
- The modified output DataFrame is then converted to CSV using Pandas.

Result:

- Score received in Kaggle: 0.97139

Fig 2: Kaggle Score for Task 2

3 Custom Feature Engineering to improve the model.

Objective: To improve performance of the model further by implementing any one of the modern text-based feature engineering methods.

Data Processing:

- The input data file in CSV format is read using the Pandas library initially as a part of preprocessing.
- Then the pandas DataFrame is converted to Spark DataFrame using SQLContext.
- Tokenizer API in pyspark.ml is used to tokenize the plot in the input data.
- RegexTokenizer API from pyspark.ml is used to remove all punctuations and then tokenize the plot column in the input.
- StopWordsRemover API from pyspark.ml is used to remove stop words after tokenizing the plot.
- Thus modified DataFrame is then used as an input to Word2Vec API from pyspark.ml
- Word2Vec takes bag of words as an input and transforms it as a feature vector.
- For the labels, i.e., genre, the genre column in the DataFrame is split to create a separate column for every genre.

- If the movie belongs to that genre, it is given as 1 else given as 0.
- Thus modified feature vector is used as an input to 20 different machine leaning models. Each model performs a binary classification, which says if that particular movie belongs to that particular genre or not.

ML:

- Same model that has been used for task1 has been used.
- Machine Learning model used in this assignment is LogisticRegression from pyspark.ml.
- Logistic Regression is selected because for performing a binary classification, it gave a better performance.
- Hyperparameters used for performance tuning are: maximum number of iterations in Logistic Regression, Vocabulary size in CountVectorizer.

Output:

- The predictions from each individual model are then concatenated to create an output predictions DataFrame.
- The output DataFrame is then joined with initial test DataFrame to create the output file with movie_id and predictions as output.
- The modified output DataFrame is then converted to CSV using Pandas.

Result:

- Score received in Kaggle: 0.98358

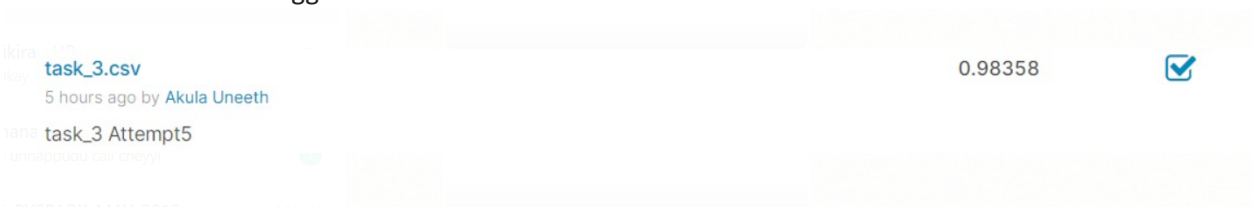

Fig 3: Kaggle Score for Task 3

Final Kaggle Scores Submitted:

- Score received in Kaggle for task 3: 0.98358
- Score received in Kaggle for task 2: 0.97139