**CSE 587- ASSIGNMENT 3**

Prabhakara Reddy Gaddam Pranali Deogade Siddarth Gaikwad

**Part-1 : Movie Genre Prediction Basic Model:**

To create a basic model for movie genre prediction we have used a binary relevance method which treats each label as a separate single classification problem. The key assumption here though, is that there is no correlation among the various labels. Binary relevance is used along with logistic regression. The union of all classes that were predicted is then taken as the multi-label output

We have created two data frames using the spark framework which are data\_spark\_df to hold the training data and lables\_spark\_df to store the labels of the movie genre. A matrix is created with all the 20 genres present attributing it to the training data by iterating through a for loop. The two data frames are later merged and are sent into the regexTokenizer, with input as the plot and output as words. Stop words are removed using stopWordsRemover. HashingTF is used taking input as words and output columns as raw features, which has vector size, vector indices and values as it’s attributes. The regexTokenizer, stopWordsRemover, HashingTF will later be put in a pipeline, which is further used to fit the training data to create our model, named as model. The same process will be applied to the test data as well, naming it as model2.

Logistic regression is performed on each label in labelCols and prediction of each label is added to dfList. dfList is then converted to temp\_df which consists of movie\_id and all the predictions against 20 labels. Then it is converted to a csv file i.e. here “predictions\_part1.csv”, which contains movie id and predictions (size 20) as specified in sample.csv. The macro f1 score observed after uploading “predictions\_part1.csv” file in the Kaggle is **0.97380**.

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

There are two data frames created using the spark framework. data\_spark\_df, which holds the training data and lables\_spark\_df, which stores the labels of the movie genre. A matrix is created with all the 20 genres present attributing it to the training data by iterating through a for loop. The two data frames are later merged and are sent into the regexTokenizer, with input as the plot and output as words. HashingTF is used taking input as words and output columns as raw features, which has vector size, vector indices and values as it’s attributes. Furthermore, an idf has been initialized taking input as the raw features and output as features. The regexTokenizer, HashingTF, idf will later be put in a pipeline, which is further used to fit the training data to create our model, named as model. The same process will be applied to the test data as well, naming it as model2.

The Logistic Regression model has been used to fit the training data set and obtain the predictions of test data using the transform function. A temp\_df is created which consists of movie\_id and all the predictions against 20 labels. Then it is converted to a csv file, which contains movie id and predictions (size 20) as specified in sample.csv. The macro f1 score observed after uploading “predictions\_part2.csv” file in the Kaggle is **0.97923**.

**Part-3: Custom Feature Engineering:**

To further improve the performance of the model, we have used word2vec as our modern text-based feature methodology. The flow continues the same as that of part-2 with few additional changes in the code. After the two data frames being created using the smart framework: data\_spark\_df and lables\_spark\_df, and after the matrix being created with all the 20 genres present attributing it to the training data by iterating through a for loop, the data frames are later merged and is sent into the Tokenizer. Instead of using HashingTF, idf, it has been replaced by word2Vec, which accepts words as input column and output column as features. The Tokenizer and word2Vec will later be put in a pipeline, which is further used to fit the training data to create our model, named as model. The same process will be applied to the test data as well, naming it as model2.

A Logistic Regression model has been used to fit the training data set and obtain the predictions of test data using the transform function. In the final csv file, a list named dfList has been created to append the movie id and predictions alongside the genre labels.

The macro f1 score observed after improvising the model to word2Vec is **1.00000**.

**Link to Videos:**

<https://drive.google.com/drive/folders/1dHcSlEfF7FI-VmnEpBLOPFfyypTMLhsv?usp=sharing>

**Note: Try to run the submitted notebooks in colab which makes task easier**