*DATA INTENSIVE COMPUTING*

*CSE-587*

*Assignment 3*

*Predictive Analytics with Spark*

*Report*

*Team Members:*

*Jianping Zhang*

*Shiyu Lu*

*Anjali Bachani*

Part 1: Basic Model

Our data has train.csv file that contains 4 columns which are movie\_id, movie\_name, plot, and genre. We have one test set which has 3 columns which are movie\_id, movie\_name and plot. Our goal is to predict the genre from the plot of the movie.

Firstly, we use pandas and spark to process the training data, mapping data and test data and store them. There are 20 genres in the mapping, so the feature is 20-D. Also, the basically idea is to split the word in the plot and according the words to predict the genre of the movie. We used to do this directly in VM jupyter, and try to create a term-document matrix, but it seems to be out of memory. Finally we use google colab to help us to deal with the huge data. And then we need to use spark library. To do this, we set spark and java-8 environment. We used RegexTokenizer as it allows more advanced tokenization which is based on regular expression matching. After this, we added a “remover” to remove the words that should be excluded from the input and used “Count Vectorization” feature extractor. It helps to convert a collection of text documents to vectors of token counts. It can be used to extract vocabulary and generate Count Vectorization Model. When the data is learned, we fit a pipeline. And finally, we make predictions. We clean the data and make a list to store genre of the movies and to catch the exceptions, we add a try and catch block where we clean the genre list as they contain”[“ symbols which we need to replace and use ”,” instead and calculate the row length. Then we get feature matrix and dense matrix for our data. And using these matrices we make a dataframe to store the value of each genre for each movie in that dataset. Then before we apply the logistic regression, we process our data and features. Then we use logistic regression function from the “pyspark.ml.classification” module and store the resultant predictions and append them together. In the new table we have an extra column containing tokens and features. Then we format the predictions in the particular format and save these predictions by creating a csv file and storing the predictions of each movie in it.

After doing this whole process we got the F1 score of 0.97988.

Part 2: Using TF-IDF

TF-IDF stands for term frequency Inverse document frequency. It is a feature vectorization method which is globally used to mine text so that it can show how much a term is important in a data. We used Hashing TF to generate the term frequency vectors. It takes set of terms and convert it into a feature vector of a fixed length. By this way, we can map a raw feature by using hash function.

In this part, we changed our Count Vectorization and used HashingTF as it is very useful for discrete probabilistic models that model binary, rather than integer, and as our model has binary values for different genre, hashingTF is the right approach to use for this particular model.

After making these changes to the first part, our F1 score is 0.98461 for our part 2.

Part 3: Using Word2vec

Word2Vec is an Estimator which will use a sequence of words that represent a document and train a model based on it which is called “Word2VecModel”. If we look at the working of this model, we will find that it uses each word of the document to a unique fixed-size vector. Basically, it will just transfer the document into a vector form. Then moving forward these vectors are used as a feature to make predictions.

So, in part 3 we have substituted and used Word2Vec feature extractor to map our terms more accurately.

Conclusion

From this assignment, we get familiar with the Spark and gain the experience to deal with the huge data like google colab and pandas. We used to use Spark directly process the data, but we get wrong data in plot columns. So finally we change to use pandas and spark both. On the other hand, we have googled a lot of passage and reference to Spark library to help us with this assignment. It is very difficult at first part, but things go better when we are familiar with how to use Spark library. And from these experience, we get to know how convenience Spark is. Without this library, we may need many times code to complete these.