CS506 Midterm Report

Jiahe Zhang, U82392079

March 2023

1 Analysis / Exploration

In this section, I tried plotting some plots to understand the dataset's appearance. In the short term, we are trying to analyze the dataset with a total of 1397533 data entries in the training set and 9 features (8 features excluding Score).
As shown in the graph below, the higher the score is, the more reviews there are. I also plotted the most used words used in the Summary and Text columns, as shown in the second figure below.

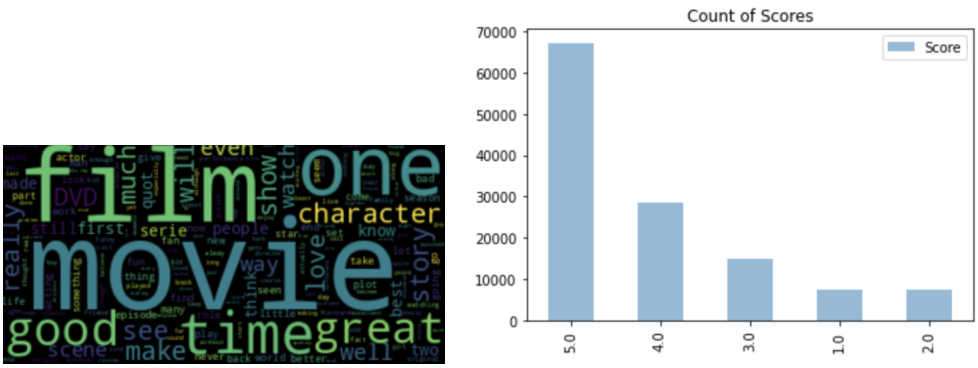


Figure 1: Analysis figures

2 Feature Extraction

In the data features, columns "HelpfulnessNumerator", "HelpfulnessDenominator", "Summary", and "Text" contain the most information, and the most important columns are the Summary column and the Text column.

1. Helpfulness Helpfulness is a newly added feature calculated by dividing HelpfulnessNumerator by HelpfulnessDenominator. By observation, there are some data with abnormal fractions of these two values, which we don't know the cause of. So I removed them in data cleaning.

2. Summary, Text
   As I mentioned in the summary of this section, the columns "Summary" and "Text" are the most important ones, and they are both pure string types. We can combine their texts and then tokenize the words as features. Then we can remove words that are rare

to appear or to often to appear in all rows, doing so will prevent overfitting the data to our models.

3. TF-IDF and CountVector
   TF-IDF and CountVectors are both derived from the combination of summary and text columns. These two features provide very useful information about the individual response to all of the sentences as a collection. they also help us to transfer string features into usable numerical features.

4. Other columns
   I have also considered and tried to use other columns to train and test my models, however, it didn't go so well since all other columns are not numerical and it would be hard to find a way of mapping these strings into meaningful numbers for further usages. I think they might mess up the whole algorithm if I transform them then use them as selected features

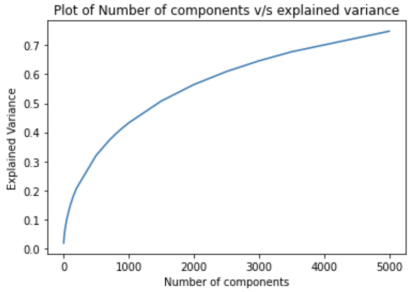
3 Workflow, Decisions, and Techniques Tried

1. Dataframe cleaning
   Dataframe has to be done in order for all other parts to function correctly. This is because data frame cleaning not only filters out unusable information and reduces error occurring rate, but it can also greatly reduce the size of the final matrix for training and testing, therefore reducing lots of time for processing. Procedure: first removed all rows with nan values in them, then I removed stopwords and punctuation using nltk module.

2. word count & tfidf transform
   Next, I calculated the word count vector using count vectorizer and transformed it with tfidf vectorizer using the word collection from the training set. The result will give us a huge matrix using each individual word as a column (feature) for training.

3. try applying svd
   I also tried to apply svd to select the most useful words in the tfidf matrix, as shown in the figure below, I tried to use different number of components for the truncatedSVD method, but there occurs some problems. 1. it cost too long to run the tests, 2. the best coverage was to cover about 80 percent of all words with 10000 words instead of a total of 30000+ words. (Still shown in the figure below) Since the coverage isn't good enough, I have to drop this part and move on.

4. Create ml model and fit model, then validating and output prediction. I always split the training dataset to a training set and a testing set, where I use the training set to train the model and the testing set to validate my prediction. More details will reflect the model testing section.



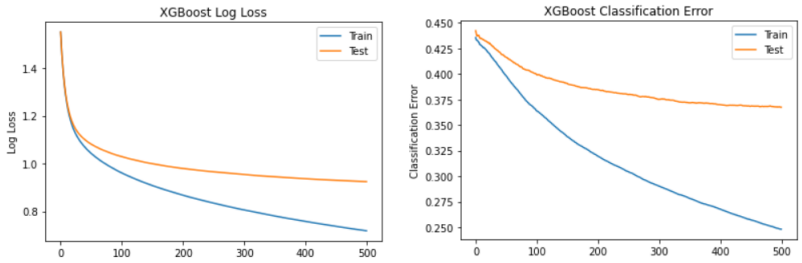
4 Model Tuning/Testing

There are mainly 3 models that I have tried, all my models over the cleaned data frames that I mentioned in the previous parts. The best model overall is the logistic regression model.

1. KNN (located in the very end of the file)
   As KNN is given in the starter code, it became the first model that I tried and tested. The best score I got is around 1.4-1.9 as I adjust the parameters of the model. The best N value is in the range of [11- 15], yet the score is too low to be submitted, thus I tried a few times and then give up with this method.

2. Logistic Regression
   For logistic regression, I tried multiple feature selections. In the beginning, I tried to take all numerical features as consideration, but the result is that the accuracy will be around 50 - 55%, and the RMSE within the range of 1.0 - 1.3, which is not ideal. Then I tried to use only tfidf transformed matrix to train the model and the RMSE reduced to 0.82 to 0.84, and accuracy of around 65% which is much better. I also tried using only countVector for logistic regression, and it is not ideal as tfidf vectors. Then, I tried the cost parameter based on the output performance and found that the model works best when the cost is 80.

3. grid search
   In this method, I used XGBClassifier() on grid search. I learned this method from a paper I have read. This classifier has lots of parameters to be tuned, I tried about 10 different parameter sets for the model and the one that works the best is $params = \{colsample_bytree = 0.8, gamma = 0, learning_rate = 0.1, max_depth = 6, min_child_weight = 1, reg_alpha = 1, subsample = 0.7, objective =' multi : softmax', n_estimators = 500\}$. However, this seems to have the problem of overfitting, since the model works well on the training set but not as good in the testing set.



5 Challenges/Effort

1. I tried different ways to clean the dataset, I tried using the stop words from sklearn but it was not effective, so I move with nltk. I also tried to use truncated svd on the new features I selected, but it does not work so well.

2. Most models are expensive to train. normally a single cell would cost me 20-30 mins to execute if successful. If unsuccessful, I will have to debug them and tune them then rerun the cell, which even enlarges the opportunity cost of searching for new features.

3. I read several papers on grid search, but the result does not meet my expectations.