Haniel Edward Jacob

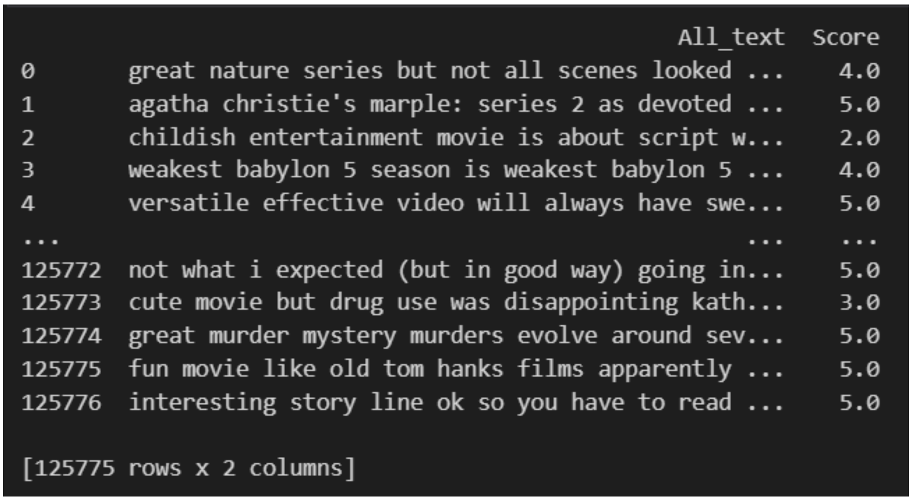
CS - 506 Midterm Report

Kaggle display name: T Haniel Edward Jacob

Finding key features

The process of feature selection is crucial to develop an accurate predictive model, and I approached it by focusing on the text columns 'Text' and 'Summary'. I found it intuitive that positive reviews would result in higher scores, so I chose to use only these two features in my analysis. To combine them effectively, I created a new column called 'All_text', recognizing that both columns contained valuable information for determining the positivity or negativity of the review text. I performed some pre-processing on this column to make it easier to predict the score. The steps I followed are:

1. Removed all the rows containing NaN values.
2. Split the text into tokens (words) and removed single quotes from each word.
3. Converted all letters to lowercase.
4. Removed stop words from the column.

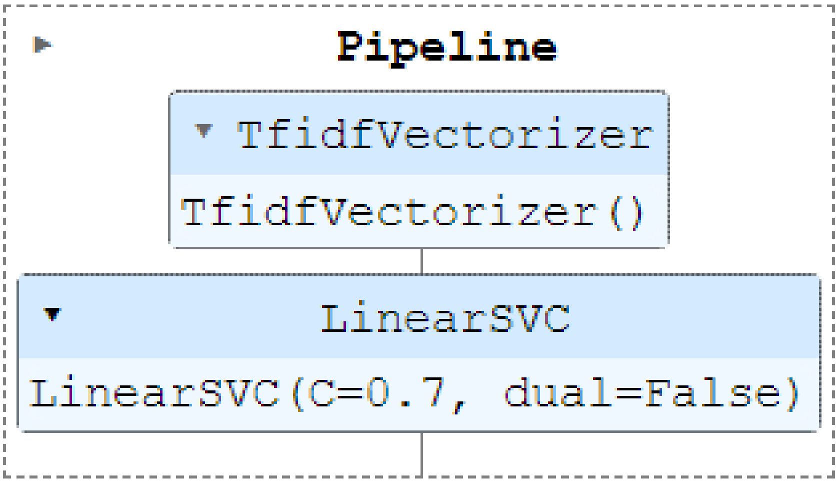


```
                                                All_text  Score
0       great nature series but not all scenes looked ...    4.0
1       agatha christie's marple: series 2 as devoted ...    5.0
2       childish entertainment movie is about script w...    2.0
3       weakest babylon 5 season is weakest babylon 5 ...    4.0
4       versatile effective video will always have swe...    5.0
...                                                   ...    ...
125772  not what i expected (but in good way) going in...    5.0
125773  cute movie but drug use was disappointing kath...    3.0
125774  great murder mystery murders evolve around sev...    5.0
125775  fun movie like old tom hanks films apparently ...    5.0
125776  interesting story line ok so you have to read ...    5.0

[125775 rows x 2 columns]
```

Selecting the model

In order to identify the most suitable model for the task of predicting scores based on review text, I employed a thorough and systematic approach. Utilizing a brute force method, I tested numerous models and analyzed their corresponding root-mean-square error (RMSE) scores. Through this process, I ultimately determined that the LinearSVC model was the optimal choice for accurately predicting scores based on review text. To effectively implement the LinearSVC model and streamline the analysis process, I integrated it into a pipeline. In addition, I utilized Tf-idf vectorization on the 'All_text' column to convert the text strings into sparse matrices, a strategy that effectively optimized the dataset for LinearSVC training requirements.



```
TfidfVectorizer()
LinearSVC(C=0.7, dual=False)
```

Tuning model parameters

Upon selecting the LinearSVC model, I devoted considerable effort to tuning its parameters to optimize its performance. After conducting an extensive analysis of various parameters and their corresponding values, I determined that the most effective parameters for my model were the C and dual parameters.

The C parameter is a crucial regularization parameter that plays a critical role in controlling the trade-off between training error and testing error. A lower value of C encourages a wider margin and simpler decision boundary, but this may come at the expense of lower accuracy on the training data. Conversely, a higher value of C leads to a narrower margin and a more complex decision boundary, which could result in overfitting to the training data. In addition, the dual parameter in LinearSVC is equally important as it determines whether to solve the primal or the dual optimization problem.

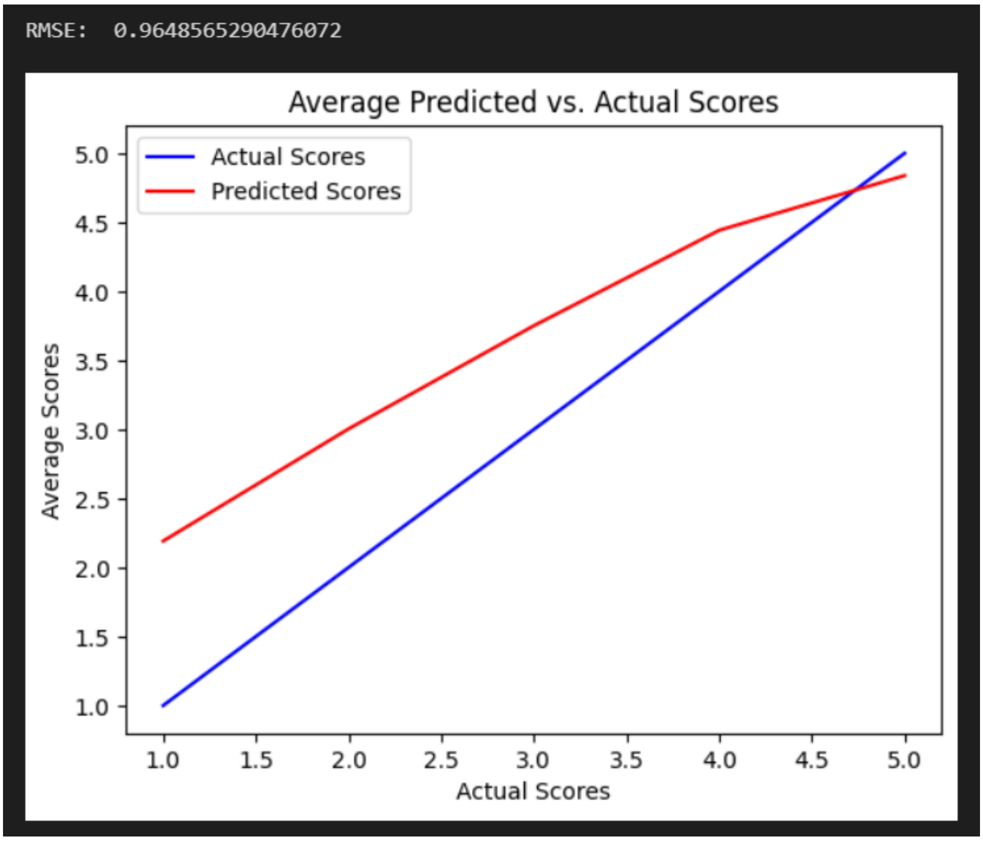
After careful consideration, I opted to set the value of C to 0.7, as this value was optimal in achieving a balance between simplicity and accuracy. Furthermore, I set the value of dual to False, a decision that I deemed appropriate given the specific characteristics of my dataset.

Validating the model

To validate the model I initially split the training data into training and testing sets with the testing set being 20% of the training set. Using this split of data, I tested various models with different parameters. I quickly realized that the LinearSVC model performed the best when compared to the other models. LinearSVC gave me the least root mean squared error compared to other models when I tried predicting the score for the testing set. This was an important step in my analysis as it ensured that my model did not overfit the data.

Challenges and key findings

One of the primary hurdles was comprehending the data and devising an appropriate approach to tackle it. With a vast amount of information to sift through and comprehend, it was challenging to distinguish between the pertinent and extraneous features. Initially, I opted to exclude the ‘text’ and ‘summary’ columns, applied one hot encoding on the ‘productId’ and ‘userId’ columns, and performed linear regression. However, the resultant root mean squared error score was notably high, highlighting the indispensability of the ‘text’ and ‘summary’ columns in the analysis.



```
RMSE:  0.9648565290476072
```

This graph was produced by plotting the actual scores against the average prediction scores. From the graph it is clear that the model performs well for higher scores and does not perform very well for lower scores. This makes sense as words such as “good” and “great” were very common in the data and these words are usually attributed to positive reviews or higher scores.