Sergio Robledo

IMDB MILESTONE REPORT

1. PROJECT OVERVIEW

The objective of this project is to provide IMDB with a rating system that will be able to summarize a person's written review into either a positive or negative review. The classification of the movie can then be turned into a rating from 0 to 100. So for example if 20 people rate a movie and 17 out of those 20 reviews are classified as positive, the movie will obtain a rating of 17/20 or 85%; further optimization can be applied to give more importance to high ranking critics and make those critics' review weigh more towards the final movie rating. However, the sole focus of this project will be to develop the machine learning model that will be classifying the review as either positive or negative. The data I will use to train the model comes from Stanford University.

A. Deliverables

i. Model Code

1. *This code will import the data, clean and wrangle obtained data, apply exploratory data visualizations, analyses and inferential statistics, and finally, a capable predictive model to classify new reviews.*

ii. PowerPoint Slide Deck

1. *Final PowerPoint will explain data exploration results, as well as summarize predictive model results and performance achieved.*

2. THE DATASET

The data set comes is provided by Stanford University; it contains the text of 50,000 reviews and each review is labeled as positive or negative reviewer sentiment. Out of all the reviews, 25,000 will be used to train the classifier and the other 25,000 will be used to test it. The first few lines in the jupyter notebook are used to import the data. Once the data has been imported, the following code cleans up both the training and testing review datasets. Before any preprocessing was done on the datasets, there were about 280,000 different words being used in the reviews, and reviews used an average of 230 words per review. Below will be a list of the preprocesses performed on the datasets and the reasoning behind those processes; the data cleaning and wrangling steps below are all needed to provide the Keras deep learning model with functioning and optimized data ready for training and testing.

A. Regular Expression Removal

i. Removed punctuation, capitalization and other characters commonly found in reviews that added zero sentiment information to the review. For example, all “it’s” got turned into “its”.

ii. Removing characters that do not add sentiment lets the predictive model learn faster and more accurately, since it has less unnecessary data to look through before it finds a useful pattern. Removing data that does not help in the training of the model is also known as “noise reduction”.

B. Stop Words Removed

i. Stop words, just like punctuation and capitalization, do not add sentiment information to the review; examples of stop words are the words “and”, “but”, “they”, “him” and etcetera.

ii. Removing these words also allows for noise reduction when the model trains. This means that the model won’t try to figure out how stop words play a role in determining the sentiment of a review, since stop words do not help an agent learn if the text in a review is positive and negative.

C. Uncommon & Other Word Removals

i. If a word is seen less than 100 times in all 25,000 reviews, it will be removed. This removes words that do not appear common enough for the model to be able to notice a link between the given word and whether a review is expressing a positive or negative sentiment.

ii. Along with removing the most uncommon words, we will remove the most common words observed that do not contain positive or negative sentiment. For example, the word “movie” is the number one most commonly word in training reviews. However, since the word “movie” is not expressing positive or negative sentiment feedback, it does not aid in the classification of a review as positive or negative. Also, given how frequent the word “movie” occurs, it will add a significant amount of noise to the training of the model. Other common words with zero sentimental information have also been removed.

iii. Both the removal of uncommon words and neutral common words will aid in a faster and more accurate pattern recognition by the model by reducing the noise created by those unnecessary words.

D. One-Hot Encoding & Sequence Padding

i. One-hot encoding accepts a review and the vocabulary size of the data. Each unique word found in all the training reviews is assigned an integer ranging from 1 to the length of the vocabulary. Each word in the reviews is then converted into the integer corresponding to that word, and the review is turned into a sequence of integers. Then in order to make all reviews have an equal length, we will pad the end of the review integer sequences with zeroes that don't represent anything.

ii. We need to convert the text data into integer sequences of equal lengths, because the Keras model we will use requires inputs to be integers and all integer vectors to be of the same dimensions.

At the end of the data cleaning and wrangling steps, the average length of reviews went from about 230 words per review to about 94 words per review. Also, the number of unique words went from about 280,000 words to 6,700 words. This amount of noise reduction will allow the model to train more efficiently and achieve better results than it would otherwise have achieved if the original data had been used.

3. EXPLORATORY DATA ANALYSIS

After conducting all preprocessing steps on the datasets, the data is ready to be explored. Exploratory data analysis lets us understand our data, gain insights on patterns the machine learning model will exploit when classifying and helps us build a machine learning model optimized for the data observed. Analyses range from simple information retrieval about the data along with visualizations to statistical tests needed to confirm the observations are statistically significant and not due to random chance. Below are going to be some observations and tests made during the exploratory data analysis stage.

A. Review Lengths

i. The average review length after preprocessing is about 94 words in length with more than 50% of all those reviews being under 71 words in length. Out of the 25,000 reviews, only 14 of those are longer than 500 words.

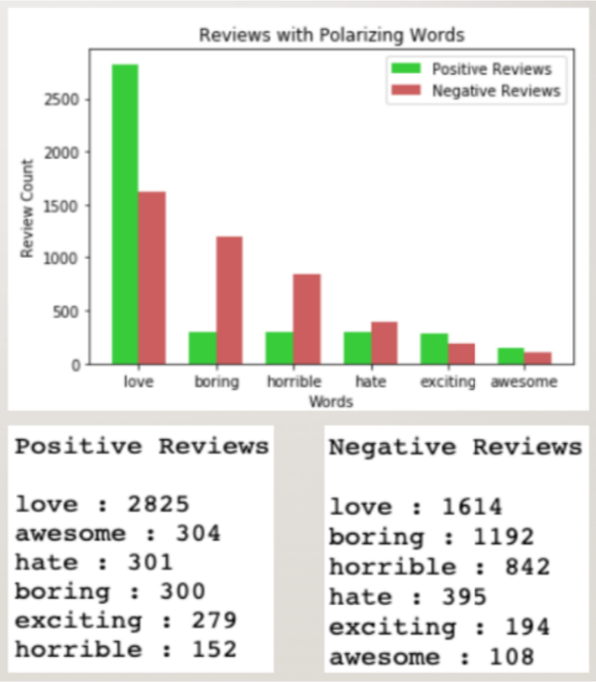
B. Words in Either Positive or Negative Reviews

i. Out of the 25,000 reviews used for training, 12,500 are positive and 12,500 are negative. After looking for the most common words found in positive reviews but not in negative reviews and vice versa, it can be observed that the words found in either type of review do not seem to communicate sentiment. For example, the most common word found in positive reviews only is "edie"; the word "edie" does not help us classify a review as either positive or negative, since it has no sentiment attached to it.

ii. This observation showcases the fact that the machine learning model will not be able to rely on stand-alone words to classify movie reviews. Instead, it will have to use the context of words in relation to one another. For example, the word "good" is very common among both positive and negative reviews. However, the context of the word "good" will matter; a positive review could be saying, "this movie is really good", while a negative review also containing the word "good" might say "this movie is really not good". Both sentences are very similar, but through the use of Keras' embedding layer, the classifier model will be able to take into account the context of the words, and it will notice the word "not" came right before the word "good".

C. Polarizing Words in Reviews

i. By counting the number of reviews containing polarizing words, or words rich in sentiment, it can be noted that there is a difference between the frequency a word is used in either positive or negative reviews. The words "love", "boring", "horrible", "hate", "exciting" and "awesome" were counted. In all of them a clear difference was observed; words relating to positive sentiment were used more frequently by positive reviews than in negative reviews and vice versa. For example, the word "love" is in 2,825 of the 12,500 positive reviews and only in 1,614 of the 12,500 negative reviews; this is a difference of 1,211 reviews. Below is a side by side bar chart depicting the number of reviews each word appeared in.



```
Positive Reviews

love : 2825
awesome : 304
hate : 301
boring : 300
exciting : 279
horrible : 152
```

```
Negative Reviews

love : 1614
boring : 1192
horrible : 842
hate : 395
exciting : 194
awesome : 108
```

Figure 1: Review Counts of Reviews Containing Polarizing Word

ii. The review counts show a clear difference in the types of words most frequently used by either type of review. Positive reviews use positive words more often than negative reviews and vice versa. This difference means that stand-alone words are not useless for the machine learning model, and both context and stand-alone words will be used by the model when classifying a review as positive or negative. In order to verify these observations are not due to chance, we will conduct permutation replicates of the data with the hypothesis that all reviews have an equal probability of having a given word. Permutation replicates are done by combining the positive and negative review datasets, shuffling that big dataset, creating two new positive and negative datasets from that shuffled dataset and finally, recording the difference between the number of reviews containing the word being tested. In order to get significant results, 100,000 permutation replicates will be created. From those permutation replicates we can then see what the probability is of a review difference equal to or greater than the one we observed is, since the permutation replicates will have simulated datasets where both positive and negative reviews have an equal probability of having in it the word being tested. If the number of differences being equal to or greater than what was observed is small enough, we can conclude that our observations would not be normal under the hypothesis and they are not due to chance. We will be testing the words "love" and "hate".

1. *"love"*

   a. The word "love" is seen in 2,825 out of 12,500 positive reviews and in 1,614 out of 12,500 negative reviews. After conducting 100,000 permutation replicates, zero out of the 100,000 simulations had a review difference of 1,211 or greater. More than 99% of all simulations had a difference of 155 or less. This lets us conclude with almost 100% certainty that the difference of 1,211 we observed is not due to chance, and the word "love" will be useful to the machine learning model when it trains to classify a review. If a review contains the word "love", the probability that it is a positive review becomes higher. Since more tests can always be done, we can never conclude with a 100% confidence any given result. Below is the distribution plot of the 100,000 review differences obtained for the word "love".

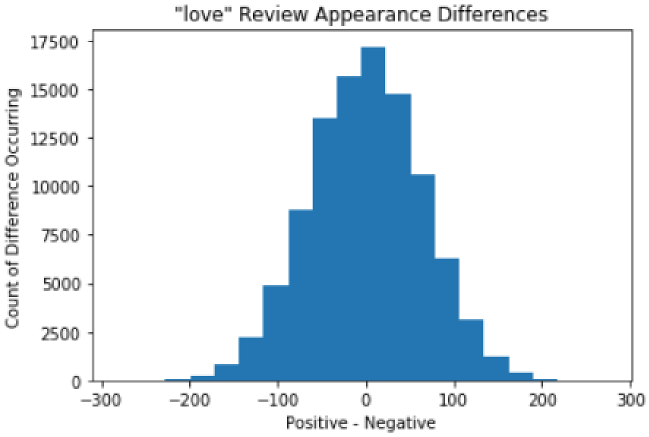


Figure 2: Review Difference Distribution of "love"

2. *"hate"*

    a. The word "hate" is in 301 out of 12,500 positive reviews and 395 out of 12,500 negative reviews. The difference between the reviews containing "hate" in it is much smaller than "love"; there are 94 more negative reviews than positive reviews containing "hate". After conducting 100,000 permutation replicates, 16 out of the 100,000 simulations had a review difference of 94 or greater. More than 99% of all simulations had a difference of 68 or less. This lets us conclude with more than 99.9% certainty that the difference of 94 reviews we observed is not due to chance, and the word "hate" will prove useful when training the machine learning model to classify a review. If a review contains the word "hate", it is more probable that the review is expressing a negative sentiment. Below is a distribution plot of the 100,000 permutation replicates done on "hate".

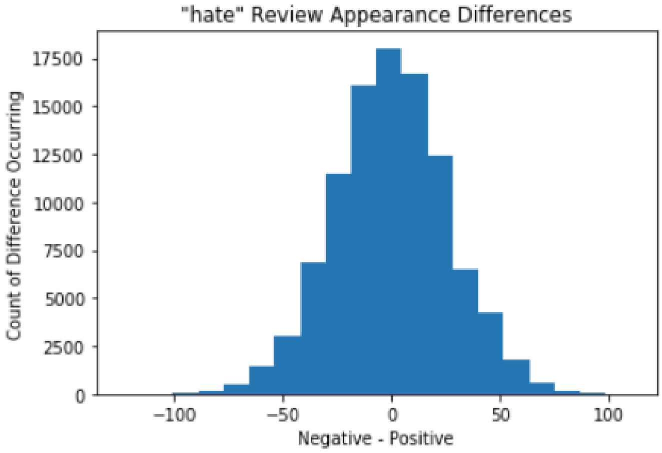


Figure 3: Review Difference Distribution of "hate"