PROJECT 5

stat4squirrels

Xiaoqian Zhang

Xinyi Liu

Shan He

Rachneet Kaur

1. Introduction

The labeled training data set consists of 25,000 IMDB movie reviews, while the test data set contains 25,000 reviews without labels. In this project, we need to predict the label of the test data using sentiment analysis techniques.

2. Preprocessing

First, turn the reviews into a corpus. Then, remove the punctuation, html symbols, stopwords, numbers and extra blanks from the reviews, and turn all words to lower case. Next, transform the review data into a word matrix and adjust the sparsity of the matrix. During the transformation, we try different models, including Bag of Word, TF-IDF and LDA. Finally, turn the matrix and the label vector into a data frame. In this way we get the data frame for modeling.

2.1 Bag of Word

In this model, we only keep words with top 5000 word frequency. The Bag of Words model models each document by counting the number of times each word appears. In order to delete very frequent but useless words when we do classification like 'movie', we set max_df=0.5. We also consider include bigram, but it costs much more time and does not improve the AUC, so we only use unigram in our final models.

2.2 TF-IDF

This model is similar as the Bag of Word model. The only difference is that we consider not only word frequency but also inverse document frequency in this model. The values of parameters in this model are the same as Bag of Word.

2.3 LDA

LDA is a topic model which represents documents using the probability of certain number of topics. We try different number of topics and it shows that when num_topic equals to 5, the performance is best.

3. Model Selection

3.1 Logistic Regression

Use glm function and set the penalty as L2 to fit a logistic regression model to the data. Then, we can acquire the predicted label to default for test data. The AUC result performs very well which is 0.95164.

3.2 XGBoost

Split the data into train set and test set. Fit an XGBoost model to train set and do prediction on the test set. Then calculate AUC based on the predicted labels and true sentiment values. It turns out the AUC is 0.9289 for the test data.

3.3 Random Forest

Use random forest model and set the number of trees as 100. Train the model and get the AUC for the test data which is

4. Result

From the results below, we finally choose the TF-IDF+Logistic model since it has the best performance than other models.

Model	Score
TF-IDF+Logistic	0.95164
TF-IDF+XGBoost	0.93448
Bag of Word+Logistic	0.92563
TF-IDF+Random Forest	0.92124
Bag of Word+Random Forest	0.91806
LDA+Logistic	0.76406

5. Visualization

5.1 Sentiment Words Tagging

In this part, we construct positive, extreme positive, negative and extreme negative word lists, and then use them to match words in our reviews of test data and highlight them.

First, we run part I model to get the predicted value of sentiment in test data, which will be displayed in the html page. Then we use the same preprocessing method as Part I to clean our train reviews. After that, we group our data into two class according to the value of their sentiment.

For each class, we calculate their word frequency and keep the top 1000 words as a word list. For word list in the class which sentiment equals to 1, we name it as positive word list. For word list in the class which sentiment equals to 0, we name it as negative word list. Then, we compare these word lists and find the difference between them. If a word occurs in both positive and negative word list, we delete it. If a word appears in the positive word list but not in negative word list, we keep it in positive word list. Similarly, if a word appears in the negative word list but not in positive word list, we keep it in negative word list. In order to get more accurate word partition, we divide positive word list into two parts, positive words and extreme positive words according to their word frequency. We define words with frequency more than the average frequency as extreme positive words, and the remaining are positive words. Similarly, we get negative and extreme negative words.

Then, we use these word lists to match words in our reviews and use different color to highlight them in html pages. We use pink to highlight positive words, red to highlight extreme positive words, light blue to highlight negative words and blue to highlight extreme negative words.

- **12311_10** sentiment = 0.948615536126232

  Naturally in a film who's main themes are of mortality, nostalgia, and loss of innocence it is perhaps not surprising that it is rated more highly by older viewers than younger ones. However there is a craftsmanship and completeness to the film which anyone can enjoy. The pace is steady and constant, the characters full and engaging, the relationships and interactions natural showing that you do not need floods of tears to show emotion, screams to show fear, shouting to show dispute or violence to show anger. Naturally Joyce's short story lends the film a ready made structure as perfect as a polished diamond, but the small changes Huston makes such as the inclusion of the poem fit in neatly. It is truly a masterpiece of tact, subtlety and overwhelming beauty.

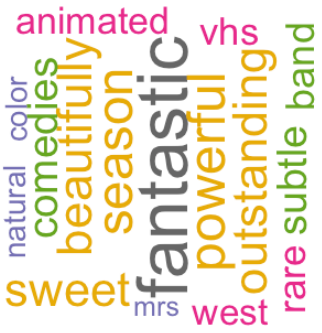
- **8348_2** sentiment = 0.0492080508955389

  This movie is a disaster within a disaster film. It is full of great action scenes, which are only meaningful if you throw away all sense of reality. Let's see, word to the wise, lava burns you; steam burns you. You can't stand next to lava. Diverting a minor lava flow is difficult, let alone a significant one. Scares me to think that some might actually believe what they saw in this movie. Even worse is the significant amount of talent that went into making this film. I mean the acting is actually very good. The effects are above average. Hard to believe somebody read the scripts for this and allowed all this talent to be wasted. I guess my suggestion would be that if this movie is about to start on TV ... look away! It is like a train wreck: it is so awful that once you know what is coming, you just have to watch. Look away and spend your time on more meaningful content.

5.2 Word Cloud

Using the prediction result of our model, we split the test data set by the sentiment label and plot the word cloud for each of the class:

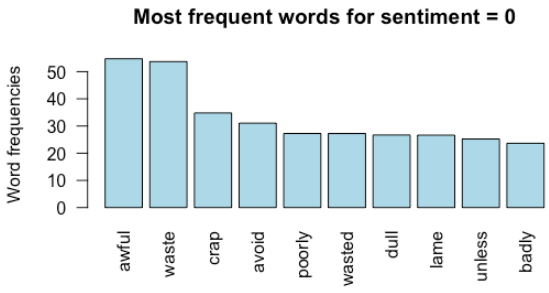
sentiment = 0

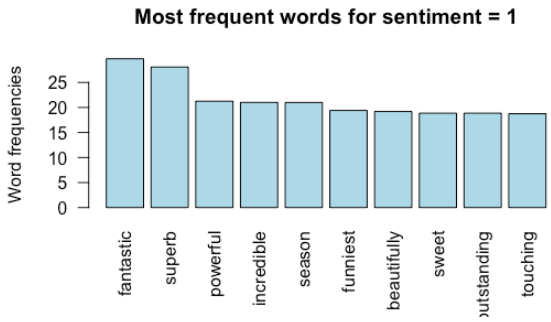


sentiment = 1

5.3 Word Frequency

We split the test data set by the sentiment label and plot the top 10 word based on the word frequency for each of the class:





6. Acknowledgment

In the preprocess and model part, we refer the code under the overview part of this project on Kaggle, which is written by Angela Chapman.

In the visualization part, we refer the code on Piazza, which is written by Professor Feng Liang.