CSE 142 Final Project Report

By Wei Yao Lu, Kevin Xu, Colin Xie

6/11/2020

Introduction

With the rise in user accessibility of the internet rising every year, user data management has become a much more relevant topic of research for developers. Most businesses and organizations are now searching for ways to analyze the behaviors and preferences of people who use their products and make changes according to the response. Whether an online shopper is satisfied with their purchase or employee performance reviews from the employer, are all data that is used for predicting trends, grouping the user base, or making any types of benefits for the future. The use of Sentiment Analysis has created a new approach for how systems review people's opinions online. Sentiment Analysis is the interpretation and classification of emotions within the text data using text analysis techniques. One of the applications of sentiment analysis is that it allows businesses to identify customer sentiment towards products, brands, or services in online conversations or feedback. Using NLP or Natural Language Processing, we can produce rules and algorithms that can perform sentiment analysis and identify user experience towards a product. Our goal is to improve the current accuracy of user satisfaction with this project and also introduce ourselves into the vast world of machine learning. So we can get started with a project that can help propel our careers forward.

Data Processing

The dataset that we are given consists of 3 .tsv and 1 .csv file. We can make assumptions that the reviews are exclusively in English, and that the data is labeled. To make the data parsable by the program, we have to tokenize each review into parsed sentences with only lower case words and remove non-letters. This is done given the starter code, from "KaggleWord2VecUtility.py". After cleaning these reviews, stop words are filtered and words chosen from vectorization are to be the parameters for our model. This is done by importing the nltk package from python.

Methodology

Once the data has been processed, we are then able to implement a random forest model to predict the sentiment of the movie reviews. This model uses the bagging method which combines learning methods to increase score accuracy. We used the scikit-learn random forest model as our main algorithm for generating the results of our experiment and running it through the data set. In this case, y will be the sentiment we are predicting with either a value of 0 or 1 (whether the review is negative or positive) and each x will be a parameter of the model which will be taken from each user review. We believe that a user's emotion towards a movie correlates to word choices and how long the review is. By analyzing a selection of words with positive or negative connotations and determining the word count as well, we can generate accurate readings of all the user inputs. The parameters we are using to build the model are occurrences of keywords that we will extract from the training data and the length of the review. More parameters may be added by the model to improve the accuracy of our prediction. As for our stop loss function, we will use the confusion matrix to test the accuracy of our prediction. This is because the data is binary, our results are only positive and negative reviews and we are not predicting the score of the review based on the content.

Experiment & Result

id	sentiment
12311_10	1
8348_2	0
5828_4	1
7186_2	0
12128_7	1
2913_8	1
4396_1	0
395_2	0
10616_1	0
9074_9	1

First we pick specific words from the train data given and categorize them if they are commonly found in positive or negative movie reviews. We also have to consider the length of a review because positive reviews are more likely to be written longer. The given chart above represents the first 10 predictions using a random forest model.

```
submission.csv                                              0.85216        0.85216
25 minutes ago by kevinlixu
Tweaks
```

Conclusion / Result Analysis

Overall, the Random Forest model was quite good. Having an accuracy score of ~85%, the decision model for our design was actually better than we expected. Although not perfect, we possibly could have lowered our accuracy by a more simple hypothesis. We could have tried something more complex which could have given us better results.

Challenges & Future Works

One of the challenges that we faced during this project was the lower accuracy of our initial model. To help this we redesigned our predicting algorithm to become a random forest type decision model. With the use of naive bayes, our accuracy was only 72~78%. I believe this was due to the fact that the hypothesis created from the naive bayes model was too simple and was too based and made many assumptions about the data.

I believe this project overall was quite a satisfying introduction to the world of machine learning. Overall, we had a good experience working together on this project. I believe that we will all apply this knowledge in the future.

Contributions

Kevin Xu - 40%

Wei Yao Lu - 30%

Colin Xie - 30%

Bibliography

https://en.wikipedia.org/wiki/Random_forest

https://www.dataquest.io/blog/naive-bayes-tutorial/

https://www.kaggle.com/c/word2vec-nlp-tutorial/overview

https://cs224d.stanford.edu/reports/SadeghianAmir.pdf

https://monkeylearn.com/sentiment-analysis/

https://developers.google.com/machine-learning/crash-course/classification/accuracy

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html