Information Retrieval Final Project Report

Raman Dutt
1610110277

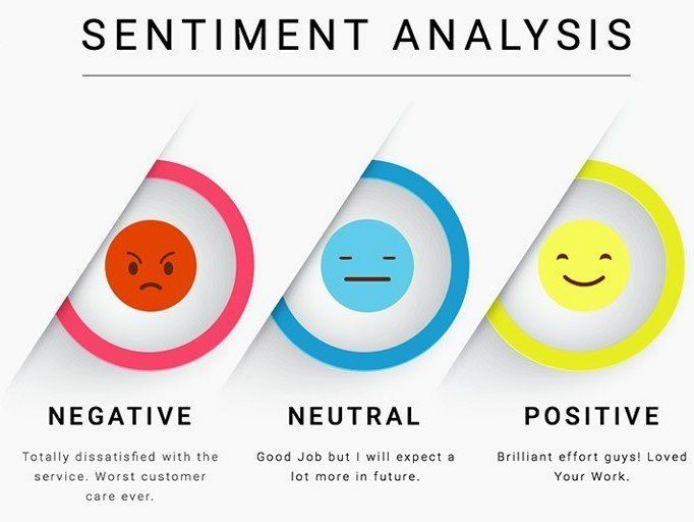
Hari Sai Raghuram
1610110145

Under the supervision and guidance of

Dr. Sonia Khetarpaul

sonia.khetarpaul@snu.edu.in
Assistant Professor, Department of Computer Science and Engineering

Sentiment Prediction from User Reviews



ABSTRACT

Sentiment analysis refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.

The end result of this project would be the accuracy or the error rate of our algorithm for the machine learning model which the companies might require to judge the usefulness of our system should they wish to use this on their websites and portals.

INTRODUCTION

Sentiment Prediction from User Reviews

The rise in E - commerce, has brought a significant rise in the importance of customer reviews. There are hundreds of review sites online and massive amounts of reviews for every product. Customers have changed their way of shopping and according to a recent survey, 70 percent of customers say that they use rating filters to filter out low rated items in their searches.

The ability to successfully decide whether a review will be helpful to other customers and thus give the product more exposure is vital to companies that support these reviews, companies like Google, Amazon and Facebook.

There are two main methods to approach this problem. The first one is based on review text content analysis and uses the principles of natural language process (the NLP method). This method lacks the insights that can be drawn from the relationship between customers and items. The second one is based on recommender systems, specifically on collaborative filtering, and focuses on the reviewer's point of view. Use of the user's similarity matrix and applying

neighbors analysis are all part of this method. This method ignores any information from the review text content analysis.

Data Overview

The data was from Multi Domain Sentiment Dataset. The Multi-Domain Sentiment Dataset contains product reviews taken from Amazon.com from 4 product types (domains):

- DVD
- Books
- Electronics
- Kitchen

Each domain has several thousand reviews, but the exact number varies by domain. Reviews contain star ratings (1 to 5 stars) that can be converted into binary labels if needed.

There are 4 directories corresponding to each of the four domains. Each directory contains 3 files called positive.review, negative.review and unlabeled.review. While the positive and negative files contain positive and negative reviews, these aren't necessarily the splits we used in the experiments.

Each file contains a pseudo XML scheme for encoding the reviews.

Data Pre-Processing and Cleaning

As dataset provided was present in pseudo XML format, it did not contain proper XML encoding. In order to process this data and feed it to the Machine Learning model, it had to be converted into another suitable format. This required quite some amount of data pre-processing.

The following steps were involved in Data Preprocessing -

1. The the pseudo XML format was converted into a proper XML format and the file was then encoded accordingly.

2. Xml.etree was used to parse xml file and extract the required data.

3. The positive and negative data have been mixed together randomly to generate a highly variable dataset which can be used for effective model training.

4. The extracted data was stored in a CSV file for using it as a pandas dataframe and various other reasons.

5. Even the unlabelled data was extracted and stored as the test set, on which our predictions will be done later on.

6. The extracted data was checked for frequently used words.

7. Stop words such as ‘of’, ‘the’ etc were removed from the dataset.

8. All the words in the dataset were converted to lowercase to maintain uniformity.

9. Some of the unwanted texts and symbols were also removed as they would cause interference during predictive analysis.

Training and Testing Models

The entire workflow was created in the form of a pipeline in order to keep things systematic and scalable. The pipeline that we created consisted of the following events -

- Countvectorizer
- tf-idf Transformer
- Machine Learning Model

Countvectorizer converts a collection of text documents to a matrix of token counts. To avoid ambiguity, we remove the stop words which cause unnecessary interference.

if-idf Transformer transforms a token count matrix to a normalized tf or tf-idf representation. We use this to provide less importance to more commonly occurring words and provide more importance to rare words which can help us classify the review.

In order to test the performance, we trained different models with different underlying algorithms. The following models were trained for predictive analysis -

- Naive Bayes Classifier[1]
- Support Vector Classifier[2]
- Random Forest Classifier[3]
- Logistic Regression
- Decision Trees Classifier[4]
- AdaBoost[5]
- K-Nearest Neighbours Classification[6]

All the algorithms were implemented using the Scikit-Learn[7] machine learning framework in python.

The models were trained to both predict the sentiment of the review and also the rating (0-5). For the rating also we have used classification technique as we want the ratings from 0 - 5.

Results - Sentiment Prediction

Algorithm Used	Accuracy
Naive Bayes	**0.826**
Support Vector Machine	**0.839**
Logistic Regression	**0.8245**
Random Forest Classifier	**0.826**

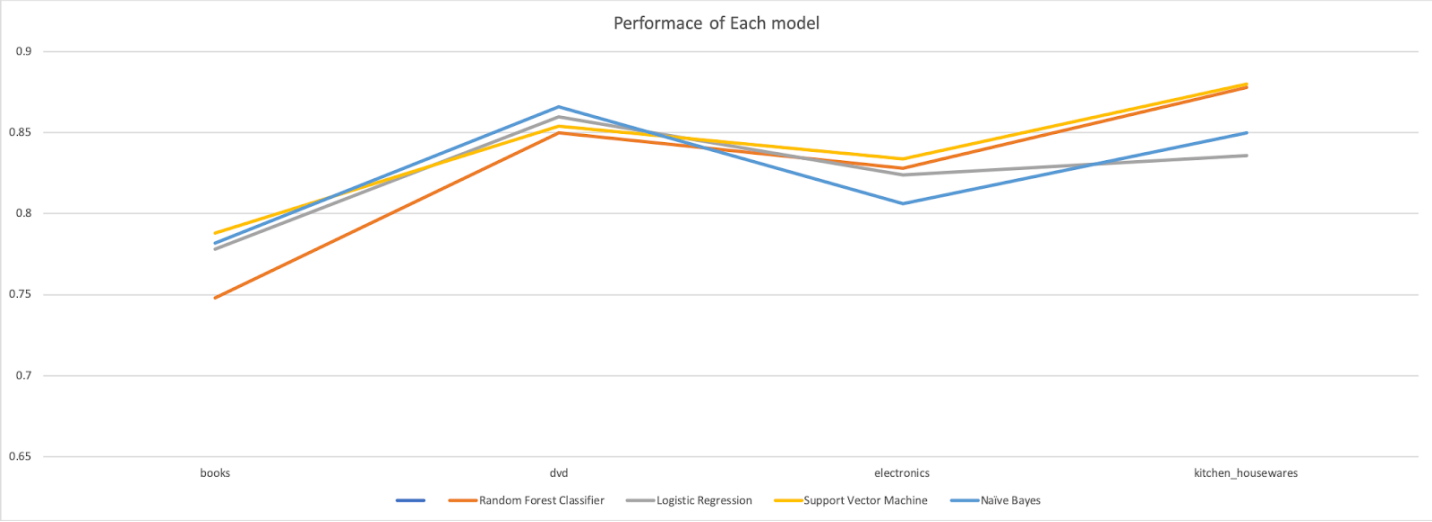


Fig 1. Performance of each model for Sentiment Prediction

Results - Rating Prediction

Algorithm Used	Accuracy
Naive Bayes	**0.5435**
Support Vector Machine	**0.6005**
Random Forest Classifier	**0.588**
Logistic Regression	**0.589**
Decision Trees	**0.4245**
AdaBoost	**0.5005**
KNN	**0.4135**

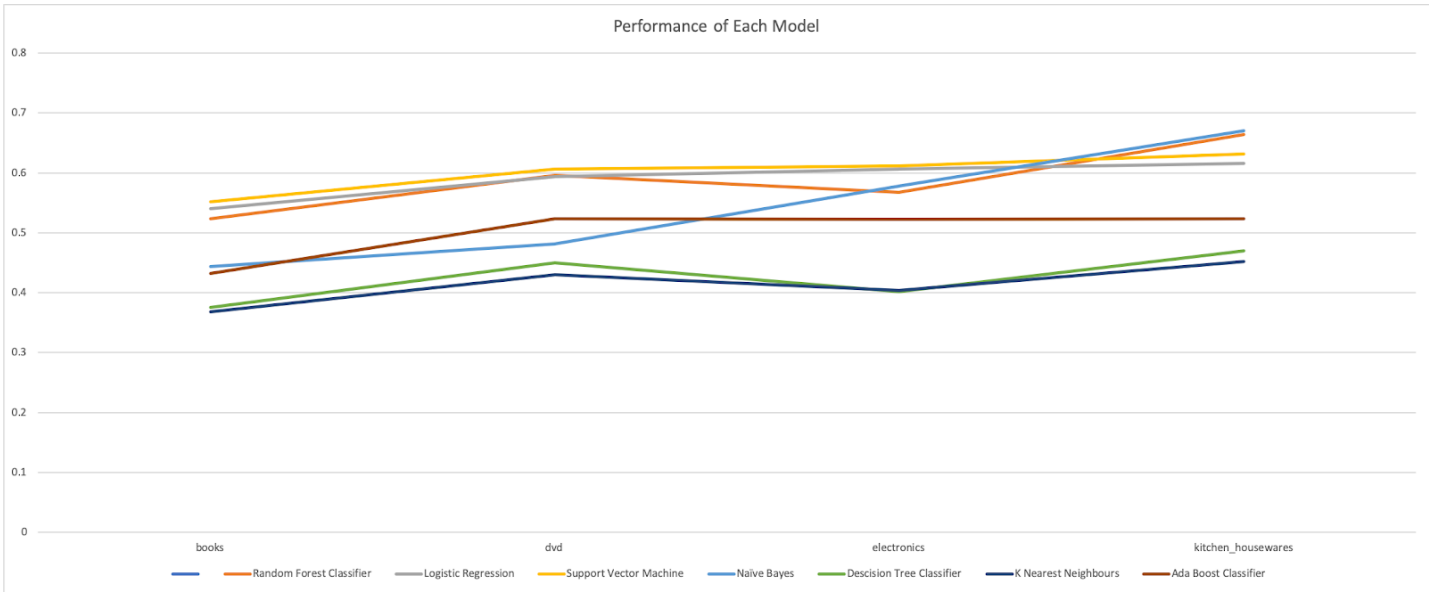


Fig 2. Performance of each model for Ratings Prediction

CONCLUSION

In this project, we have attempted to predict sentiment of the users from the user reviews. As this is a text classification problem, hence it required several steps in importing, preprocessing and cleaning the data. A pipeline of events was created for seamless and systematic work. These events included CountVectorizer, tf-idf transformer before finally feeding the data into the machine learning models. The data was divided into training and test sets for proper training and validation of the models.

From the results, we can clearly see that the Support Vector Machine gave the best results for both the tasks of Sentiment Prediction and Ratings Prediction. For Sentiment Prediction, the second-best performance was given by Naive Bayes Classifier and Random Forests (both with an accuracy of 82.6%. In case of Rating Prediction, Logistic Regression came second after SVMs with an accuracy of 58.9%.

REFERENCES

1. McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization* (Vol. 752, No. 1, pp. 41-48).
2. Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, *9*(3), 293-300.
3. Rodriguez, J. J., Kuncheva, L. I., & Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, *28*(10), 1619-1630.
4. Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, *21*(3), 660-674.
5. Rätsch, G., Onoda, T., & Müller, K. R. (2001). Soft margins for AdaBoost. *Machine learning*, *42*(3), 287-320.
6. Dudani, S. A. (1976). The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4), 325-327.
7. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... &

Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, *12*(Oct), 2825-2830.

8. Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 440-447).