Title: Opinion Mining

Group Number: 88

First Name	Last Name
Dhruva	Juloori
Lakshmi Anuhya	Koduri

Table of Contents

1. Introduction

Opinion mining is a process of determining whether a review is Positive, Negative or neutral. It is a form of sentimental analysis. The goal of our project is to extract the opinion of the review. We try to decide the polarity of views in customer reviews by applying different classification techniques for a given text and analyze which technique is better.

Opinion Mining attempts to determine which features of text are indicative of its context and build systems to take advantage of these features.

Motivation: Personally, we make a lot of purchases on e-commerce sites like amazon based on the user reviews by reading couple of them. Sometimes we end-up buying a good product and sometimes not. We wanted to provide an insight of all the reviews to user by interpreting the patterns and also help the user to buy a reliable product of his choice.

2. Data

Link: http://jmcauley.ucsd.edu/data/amazon/

Data is obtained from the above link. It contains numerous amount of reviews of amazon products in the form of JSON format. In order to use it we need to parse the JSON objects and insert the data in a new data frame.


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the
piano.  He is having a wonderful time playing these old hymns.
The music  is at times hard to read because we think the book
was published for singing from more than playing from.  Great
purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


- reviewerID - ID of the reviewer
- asin - ID of the product
- reviewerName - name of the reviewer
- helpful - helpfulness rating of the review
- reviewText - text of the review
- overall - rating of the product
- summary - summary of the review

- unixReviewTime - time of the review (unix time)
- reviewTime - time of the review (raw)

3. Problems to be solved

As the sales in e-commerce sites are increasing day by day, reviews play a vital role in each purchase. Each product may have numerous amount of reviews and it is impossible for the user to read all the reviews, sometimes ending up purchasing a product with bad reviews. It is important to provide a clear understanding of the all (day to day) the reviews to the users so that they can make a valid purchases and utilize the money efficiently, so there arises a need to classify the reviews.

To support our hypothesis, we use an existing dataset which contains numerous reviews of digital music and then try to analyze “review text” using different classifiers and provide a new label called “result” which can be one among ‘positive’ and ‘negative’. And also we come to know what features of text will influence the polarity of the review.

4. KDD

4.1. Data Processing

We only use the columns “reviewText” and “Overall” for the analysis, so we do following steps to clean the text and then use the text for analysis:

1) Removed HTML from the text

2) Non-Letters removal

3) Converting the words into a lowercase

4) Stemming and Stop words removal

After text cleaning we create a new column called “result” and set a filter for overall rating. If the rating is greater than 3 then we classify the review as “positive” and if it is less than or equal to 3 we consider it as “negative”. After setting a filter we store all the results in the “result” column and proceed for further analysis.

4.2. Data Mining Methods and Processes

After the data preprocessing, we do the following steps:

1) Load data, and run numerical summaries
2) Split the data into training and testing
3) Generate Features by computing TF-IDF scores.
4) Fit the model using Classifiers by injecting the features.
5) Compute accuracies.

TF-IDF (Term Frequency – Inverse Document Frequency):

Term Frequency (TF): The number of times a term occurs in a particular document.

Inverse Document Frequency (IDF): It is used to diminish the weight of the terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

TF-IDF is a product of term frequency and inverse document frequency. This technique is used for feature extraction.

When we apply this technique to the text we get a list of features with their respective scores as an output which we supply as an input to the classifiers.

```
In [12]: display_scores(sklearn_tfidf, sklearn_representation)
good      Score: 963.8136261722113
great     Score: 927.8512748440774
music     Score: 896.3785840256347
best      Score: 796.5405301417262
love      Score: 792.05824841352
really    Score: 693.390719011357
get       Score: 662.983887308178
time      Score: 657.3766689149833
first     Score: 656.150786567188
still     Score: 594.0752786914153
track     Score: 579.8738282382426
tracks    Score: 578.6805419044736
albums    Score: 574.6538712328617
well      Score: 566.5857961450569
would     Score: 564.6010239429548
sound     Score: 557.7108356534154
much      Score: 537.9435758940025
rock      Score: 527.2527540538091
band      Score: 525.9521533998541
```

The above figure illustrates the list of features generated by TF-IDF with their respective scores in the decreasing order.

Modelling Techniques:

NAIVE BAYES CLASSIFIER:

Naive Bayes is a classifier applied using Bayes Rules based on conditional probability. It assumes that each feature is independent and calculates the probability the feature appears in the given class. The probability of a class given the feature set is the product of the probability that the class will occur and the probability of each of the feature vectors. The process is repeated for all the classes and text is classified according to the maximum probability. More formally, c = argmax c ∈ C P (c | d) = argmax c∈ C P (d |c) P(c) P (d) = argmax c∈ C P (d | c) P(c) where c is the most likely class. First the feature vectors from the training set are used to train the Multinomial Naïve Bayes classifier in sklearn library and then test set is used to predict the polarity of the reviews with the trained Naïve Bayes model. The key advantage is, it needs less training data to estimate the parameters of classification. It is highly used for gender classification and spam filtering.

DECISION TREE CLASSIFIER:

Decision Tree Classifier is a simple and widely used classification technique. It applies a straight forward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receive an answer, a follow-up question is asked until a conclusion about the class label of the record is reached. The goal of a decision tree is to create a model that predicts the value of a target variable based on several input variables. This classifier uses the whole training set to build a tree considering all the features. It has an effective method of dealing with missing values and will not prevent splitting the data for building the trees. And also they are intuitive and easy to explain.

RANDOM FOREST CLASSIFIER:

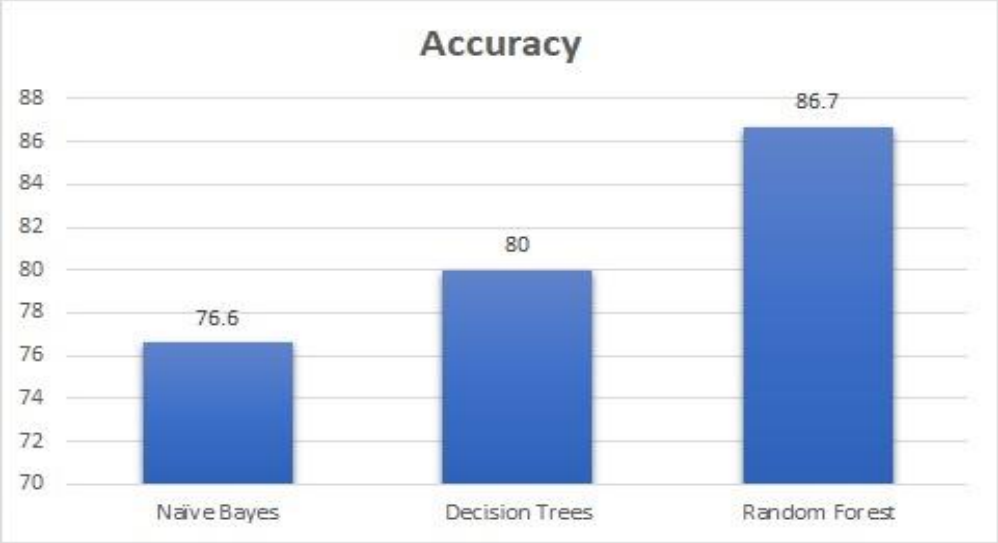
The Random Forest algorithm is one of the most widely used machine learning algorithm for classification. It is also be used for regression models (Continuous target variable) but it mainly performs well on classification model. This algorithm, is operated by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It gives an estimate of what variables are important in the classification. Also, it has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing. The advantage of using this algorithm is, it reduces the chances of overfitting. High Model Performance and Accuracy are observed by using this algorithm.

5. Evaluations and Results

5.1. Evaluation Methods

Accuracy:

Accuracy is used as a performance metric. This gives us the performance of the algorithms in the sense how many of the predictions made are correct. Accuracy is the basic measure to test the performance of the algorithm. The accuracy that should satisfy depends on the reliability that the customer would have on it and the costs matrix of false predictions.



5.2. Results and Findings

1) The accuracy for the random forest fit is 86.7%
2) The accuracy for the Naïve Bayes fit is 76.7%
3) The accuracy for the Decision tree fit is 80.03%
4) Potentially Naïve Bayes classifier may have an overfitting problem
5) From another perspective, our data is too small. The results by hold-out evaluation are probably not that reliable

6. Conclusions and Future Work

6.1. Conclusions

Baseline Performance:

Distinguishing positive from negative reviews is relatively easily for humans, especially when the whole review is read properly. But even a random guess has a probability of 50% to be correct. We can also see that there are certain words that people use to express sentiments. This word list might suffice to depend on them to classify text. A test performed on a human generated keyword explained in the

paper ‘Thumbs up? Sentiment Classification using Machine Learning Techniques’ gives accuracy of 58%. These provide us with baselines for experimental comparisons with our results.

6.2. Limitations

1) Here we try to determine the subjective value of a review, i.e. how positive or negative is the content of a review. Unfortunately, for this purpose these Classifiers fail to achieve the same accuracy. This is due to the subtleties of human language; sarcasm, irony, context interpretation, use of slang, cultural differences and the different ways in which opinion can be expressed (subjective vs comparative, explicit vs implicit).

2) Sometimes it gets difficult for the classifiers to interpret the result if a sentence contains equal number of positive and negative words.

3) If there is an imbalance of data then the model of the classifiers like SVM gets over trained.

4) It was difficult to load large datasets as the machine took very long time to load them.

6.3. Potential Improvements or Future Work

1) Accuracy of the models, can be improved by using machine leaning algorithms such as SVM(Support Vector Machine) and other classification algorithms.

2) Also, by increasing the size of the dataset we can get more accurate results.

3) As far as this dataset is considered the results by holdout evaluation may not be reliable so we should consider implanting the same techniques using cross validation and compare the accuracies of each classifier.