Project Report

Qazi Zaahirah (B00704591)

12/16/2015

Index:

1. Dataset
2. Problem Statement: Sentiment Analysis
3. Data Pre-processing
4. Feature Selection
5. Data analysis:
    a. SVM
    b. Naïve Bayes
    c. Decision Trees
6. Evaluate the Model

Dataset:

I chose to work on the IMDB dataset. This dataset consists of the reviews for the movie. The dataset has 2000 example reviews with the reviews written in plain English. At the end of the review there is a rating of the movie in the form of stars denoted by *. However for this project I only consider the words that describe review of the movie. For the domain of this project, I take the reviews as the bag of words to find the polarity of the review.

Problem Definition:

For the project I chose to get the sentiment of the review on the basis of the words that describe the review. The positive and negative inclination of the words depends on the presence and frequency of the words in the respective class. The analysis shows how the word frequency shows that the review belongs to a particular class.

It is important to note that the words that are considered for this dataset might not be of the same polarity for any other dataset. This is due to the fact that the jargon for different domain is not same. Common approaches use bag of words for determining sentiment polarity. In this project I chose to use a novice approach called Delta TFIDF[1].

The approach has been shown to produce better results for the accuracy.

1. $C_{t,d}$ is the number of times term t occurs in document d
2. P_t is the number of documents in the positively labeled training set with term t
3. $|P|$ is the number of documents in the positively labeled training set.
4. N_t is the number of documents in the negatively labeled training set with term t
5. $|N|$ is the number of documents in the negatively labeled training set.
6. $V_{t,d}$ is the feature value for term t in document d.

Since our training sets are balanced:

$$\begin{aligned} V_{t,d} &= C_{t,d} * \log_2\left(\frac{|P|}{P_t}\right) - C_{t,d} * \log_2\left(\frac{|N|}{N_t}\right) \\ &= C_{t,d} * \log_2\left(\frac{|P|}{P_t}\frac{N_t}{|N|}\right) \\ &= C_{t,d} * \log_2\left(\frac{N_t}{P_t}\right) \end{aligned}$$

The proposed method boosts the importance of words that are unevenly distributed among the positive and negative class and discounts evenly distributed words. Use this frequency I created a matrix for evaluation.

Data Pre-processing:

The IMDB data is raw file which are review passages that need to be cleaned and processed before the words can be analysed.

1. First and foremost remove all the stop words, punctuations and numbers. Since I am not using any star rating I remove the special characters as well.
2. Create the corpus of all the words both for positive and negative classes. This corpus has only unique words and no redundant words. The size of this corpus is about 30,000 words for 2000 documents
3. A corpus was created for positive and negative classes separately. This corpus is used to remove the common words (intersection) in the main corpus. Researches show that the words that are present in both the classes hardly contribute to the classification of the data. Also a lot of these words are neutral and do not have polarity, hence their presence unnecessarily increases the feature space. words like movie, plot, actor,

girl, main lead, director get illuminated which reduces the corpus size to 14600(approx.)

4. Using the log function I give the words weight according to their importance in the class.
5. For all the documents find the final frequency.
6. The feature space is further reduced by applying the threshold on the frequency matrix. This eliminates the words which occur occasionally in the documents and make the matric sparse. To increase the density of the matrix, these words are removed using a threshold.
7. This decreases the feature space to 362.
8. The pre-processing part was coded in java.

Data Analysis:

This part was coded using R using the inbuilt functions. Since this is a classification problem I used supervised learning methods for learning the data. I used three learning algorithms to classify the data.

1. SVM
2. Naïve Bayes
3. Decision Trees

SVM:

Support vector machine is used for the classification of the binary data. Since my data is also binary classification. Also the research which I have taken the inspiration from uses SVM to evaluate the delta tfidf method of getting the improved feature space. The function used for SVM is radial. I have not used cross validation to evaluate the model.

Results: The results are shown as below: The data is divided into the ratio of 80:20 where 80% of the data is used for the training and the remaining 20% s used for the testing. The confusion matrix is plotted for both training and testing data to get the training and test accuracy respectively

Confusion Matrix and Statistics for the training data (80%)

```
          Reference
Prediction   0   1
         0 800   0
         1 260 540

               Accuracy : 0.8375
```

```
                  95% CI : (0.8185, 0.8553)
    No Information Rate : 0.6625
    P-Value [Acc > NIR] : < 2.2e-16

                   Kappa : 0.675
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7547
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.6750
             Prevalence : 0.6625
         Detection Rate : 0.5000
   Detection Prevalence : 0.5000
      Balanced Accuracy : 0.8774

       'Positive' Class : 0
```

Confusion matrix for the test data, the accuracy for this data is approximately 83%.

Confusion Matrix and Statistics for the test data (20 %)

```
          Reference
Prediction   0   1
         0 200   0
         1  87 113

               Accuracy : 0.7825
                 95% CI : (0.7388, 0.822)
    No Information Rate : 0.7175
    P-Value [Acc > NIR] : 0.001883

                  Kappa : 0.565
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.6969
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.5650
             Prevalence : 0.7175
         Detection Rate : 0.5000
   Detection Prevalence : 0.5000
      Balanced Accuracy : 0.8484

       'Positive' Class : 0
```

The accuracy for the test data is also reasonable 78%.I tried to change the ratio of the training and test set to check what effect does it have on the accuracy. I divided the set into the ration of 60:40

Confusion Matrix and Statistics for training (60%)

```
          Reference
Prediction   0   1
         0 600   0
         1 200 400

               Accuracy : 0.8333
                 95% CI : (0.811, 0.854)
    No Information Rate : 0.6667
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6667
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7500
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.6667
             Prevalence : 0.6667
         Detection Rate : 0.5000
   Detection Prevalence : 0.5000
      Balanced Accuracy : 0.8750

       'Positive' Class : 0
```

Confusion Matrix and Statistics

```
          Reference
Prediction   0   1
         0 400   0
         1 161 239

               Accuracy : 0.7988
                 95% CI : (0.7693, 0.826)
    No Information Rate : 0.7012
    P-Value [Acc > NIR] : 2.646e-10

                  Kappa : 0.5975
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7130
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.5975
```

```
            Prevalence : 0.7013
        Detection Rate : 0.5000
  Detection Prevalence : 0.5000
     Balanced Accuracy : 0.8565

      'Positive' Class : 0
```

The accuracy of the both training and test data does not have much effect if the ratio of the train and test data is changed.

Naïve Bayes:

Naïve Byes is widely used for the text mining. I am using the Naïve Bayes to see what effect it has on the polarization of the words in terms of the sentiment. I used the Laplacian smoothening and the kernel is being used. The results show that the accuracy is less that of the SVM. A confusion matrix is being plotted.

```
Confusion Matrix and Statistics for the Naïve Bayes

          Reference
Prediction   0   1
         0   0   8
         1 400 392

               Accuracy : 0.49
                 95% CI : (0.4548, 0.5252)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.7261

                  Kappa : -0.02
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.0000
            Specificity : 0.9800
         Pos Pred Value : 0.0000
         Neg Pred Value : 0.4949
             Prevalence : 0.5000
         Detection Rate : 0.0000
   Detection Prevalence : 0.0100
      Balanced Accuracy : 0.4900

       'Positive' Class : 0
```

Decision Tree:

The decision tree is being constructed and then pruned to see what effect it has on the decision tree.

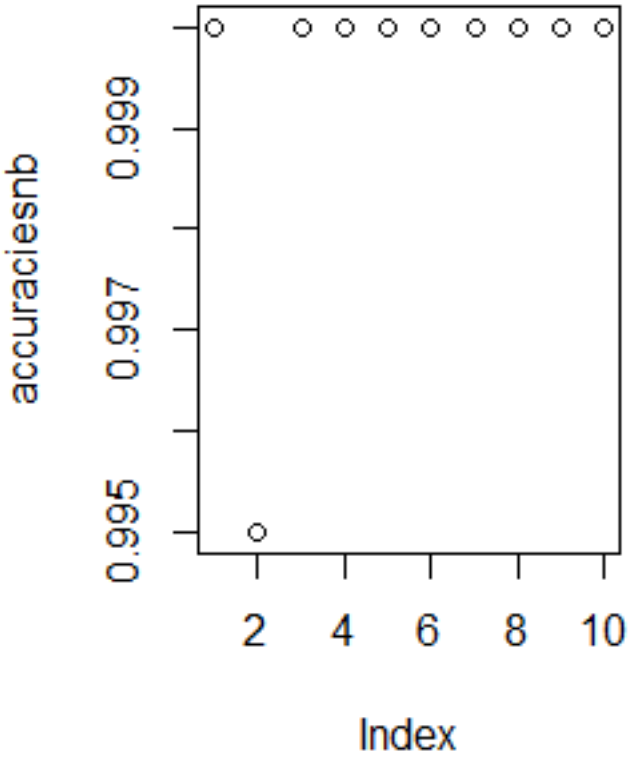
```
Confusion Matrix and Statistics for decision tree bef
ore Prunning

          Reference
Prediction   0   1
         0 350 330
         1   0  20

               Accuracy : 0.5286
                 95% CI : (0.4908, 0.5661)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.0702

                  Kappa : 0.0571
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.00000
            Specificity : 0.05714
         Pos Pred Value : 0.51471
         Neg Pred Value : 1.00000
             Prevalence : 0.50000
         Detection Rate : 0.50000
   Detection Prevalence : 0.97143
      Balanced Accuracy : 0.52857

       'Positive' Class : 0


Confusion Matrix and Statistics after pruning

          Reference
Prediction   0   1
         0 350 350
         1   0   0

               Accuracy : 0.5
                 95% CI : (0.4623, 0.5377)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.5151

                  Kappa : 0
 Mcnemar's Test P-Value : <2e-16
```

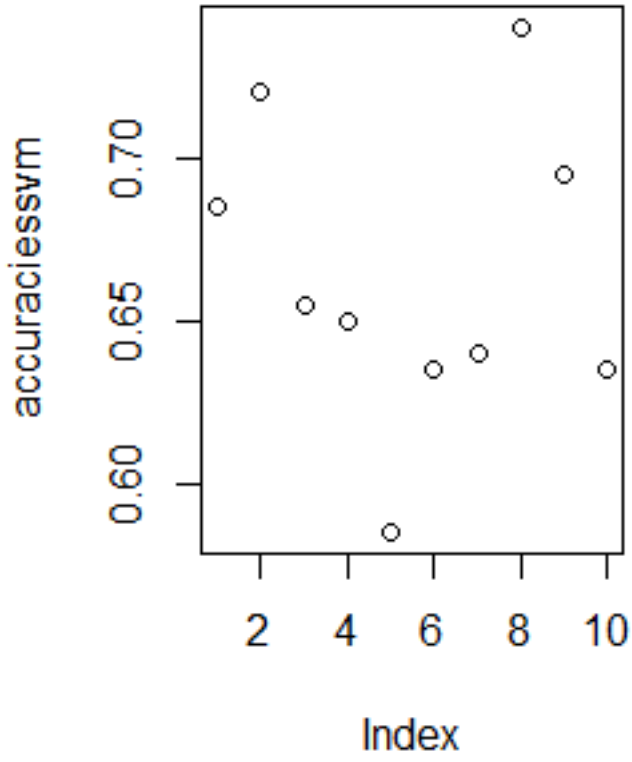
```
            Sensitivity : 1.0
            Specificity : 0.0
         Pos Pred Value : 0.5
         Neg Pred Value : NaN
             Prevalence : 0.5
         Detection Rate : 0.5
   Detection Prevalence : 1.0
      Balanced Accuracy : 0.5

       'Positive' Class : 0
```

The accuracy of the decision tree after pruning decreases which infers that pruning does not benefit the decision tree.

Evaluating the model:

I used cross validation and T test across SVM and Naïve Bayes to check which model is better for classification of the data.
I use a for loop to get the accuracies for SVM and NB


accuraciesnb
0.995
0.997
0.999
2
4
6
8
10
Index



The mean of the accuracies for the Naïve Bayes is 99% and the mean of the acc uracies for the SVM is 66%. The standard deviation of the Naïve Bayes classifie r is 0.001581139
and for the SVM is 0.04605552. From the standard deviation it seems like Naïv e Byes is better as a classification algorithm.

T test:

The null hypothesis is that both Naïve Byes and SVM models are equal. If the p value is much smaller than the significance level (0.05) the null hypothesis is rejected. For the null hypothesis to be significantly true the p value should be way higher than the significance threshold.

```
        Welch Two Sample t-test

data:  accuraciessvm and accuraciesnb
t = -23.023, df = 9.0212, p-value = 2.528e-09
alternative hypothesis: true difference in means is n
ot equal to 0
95 percent confidence interval:
 -0.3684537 -0.3025463
```

```
sample estimates:
mean of x mean of y
   0.6640    0.9995
```

The p value of the T test is `2.528e-09`which is way lower than the significant threshold (0.05), hence the Null hypothesis (The two models are similar) is rejected

References:

[1] Martineau, J., & Finin, T. (2009, May). Delta TFIDF: An Improved Feature Space for Sentiment Analysis. In *ICWSM*.