Sentiment Analysis on

Amazon Alexa

Project REPORT



Data Mining | 28 April 2019

Table of Contents

[Background 2](#_Toc7352865)

[Data Description 3](#_Toc7352866)

[Data Source 3](#_Toc7352867)

[Description of Variables 3](#_Toc7352868)

[Preliminary EDA 4](#_Toc7352869)

[Customer Rating 4](#_Toc7352870)

[Word Count and word cloud 5](#_Toc7352871)

[Word Cloud: Positive/negative word cloud segregation 6](#_Toc7352872)

[Sentiment Ratio 6](#_Toc7352873)

[Sentiment analysis using different lexicons 7](#_Toc7352874)

[Measuring performance of our sentiment analysis 9](#_Toc7352875)

[Bigrams 11](#_Toc7352876)

[Model 13](#_Toc7352877)

[Decision tree 14](#_Toc7352878)

[Random Forest 16](#_Toc7352879)

[Conclusi0n 17](#_Toc7352880)

[References 17](#_Toc7352881)

# Background

Focusing on a sort of data that has become extremely common as the internet has become one big channel of communication i.e. text data.

Many legacy applications still produce or record text. Medical records, consumer complaint logs, product inquiries, and repair records are still mostly intended as communication between people, not computers, so they’re still “coded” as text. Exploiting this vast amount of data requires converting it to a meaningful form.

In business, understanding customer feedback also requires understanding text. There are star ratings as well, but we want to “listen to the customers”, we will have to read on the comments- in reviews, feedback forms, opinion pieces and email messages.

Text data can also be called an “unstructured” data as it doesn’t have the cost of structure that we normally expect for usual data- records with data in tabular format. Text has a structure, but it is linguistic structure- intended for consumption by humans and not computers. Words can have varying lengths and people may write ungrammatically, they misspell words, run words together, abbreviate and punctuate wrongly. Thus, since texts are difficult, we preprocess it differently to make it more understandable. Thus, we wanted to understand customer reviews and see how people react to a product. This is our motivation for this project.

# Data Description

## Data Source

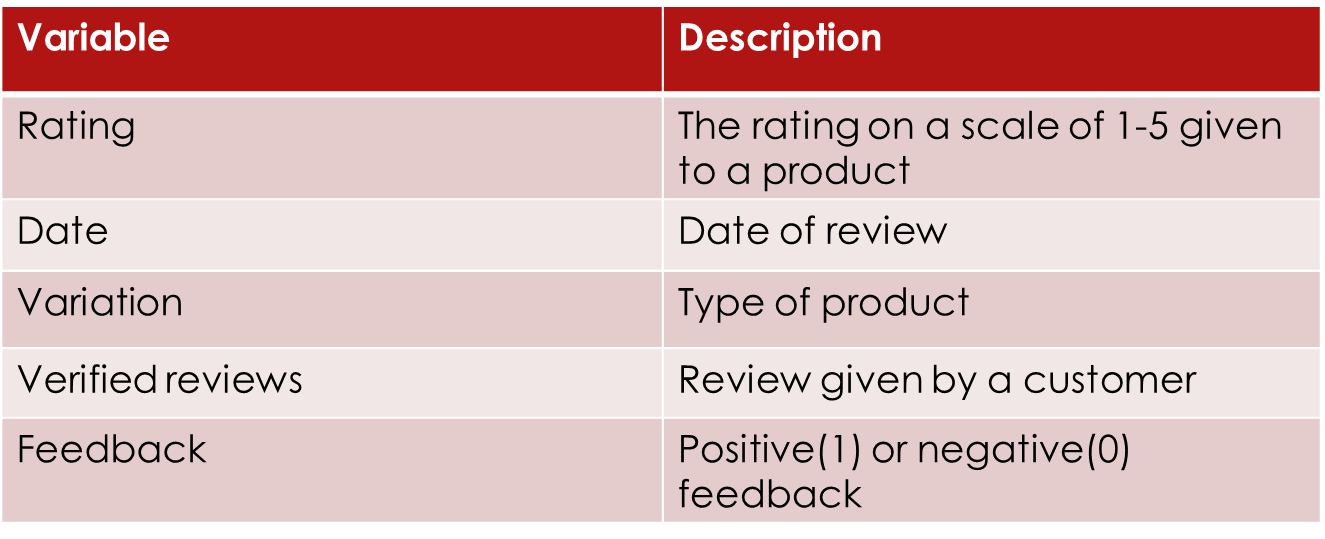
The Amazon Alexa reviews dataset is obtained from the Kaggle website ([link](https://www.kaggle.com/sid321axn/amazon-alexa-reviews)).

## Description of Variables

We are looking at the customer reviews for Amazon Alexa products. Dataset was obtained from Kaggle. It consists of a nearly 3000 Amazon customer reviews (input text), star ratings, date of review, variant and feedback of various amazon Alexa products like Alexa Echo, Echo dots, Alexa Firesticks etc.

We are looking at the customer reviews, which we will mine and look at individual words to understand customer sentiments to better understand about our product. The whole intention of any product-maker is to make customers happy. If we can understand the reviews, it will help us understand our customer requirements better thereby providing them better service.

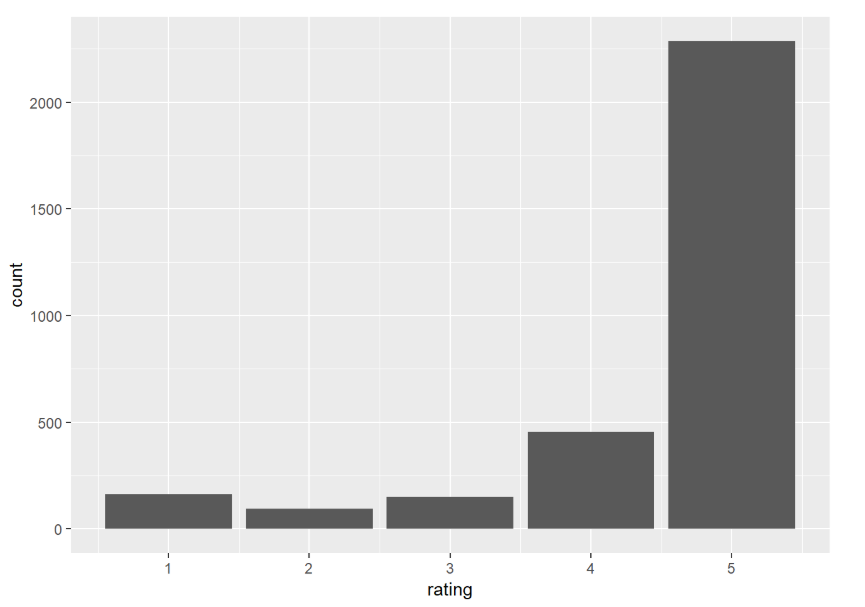
[Table – 1: Variable Description]



# Preliminary EDA

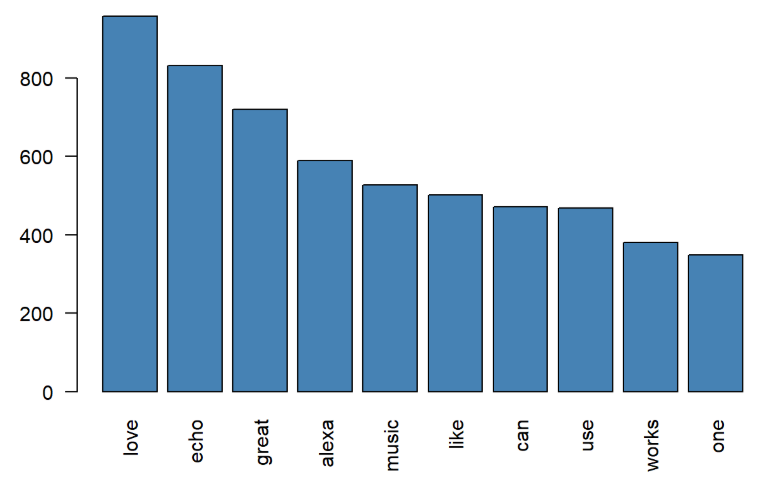
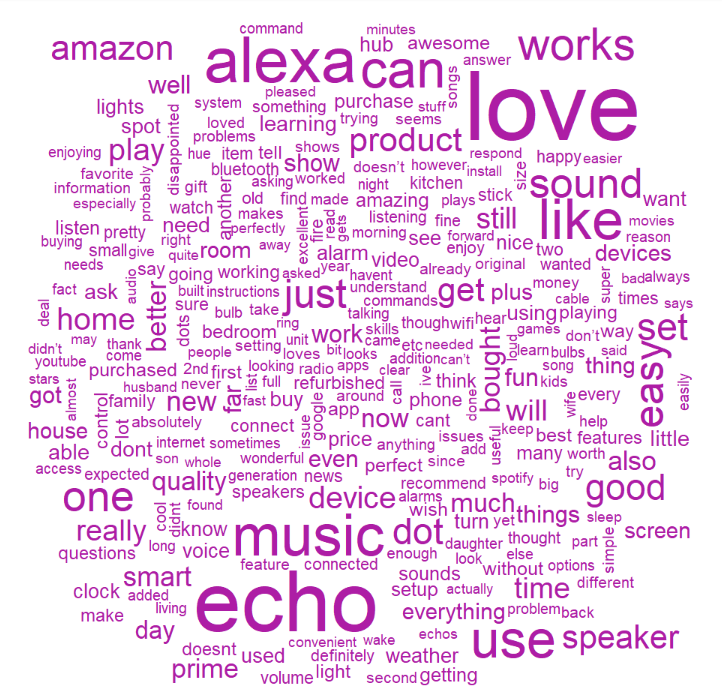
## Customer Rating

[Plot – 1: Count of Ratings]



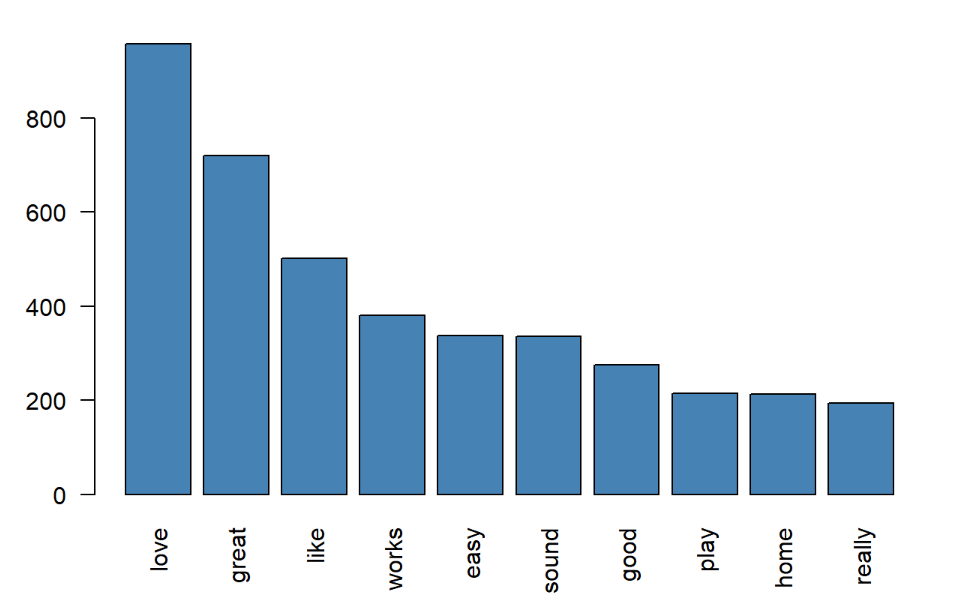
According to this graph, it seems more than 70% of users liked the products and have rated them as 5. However, this might be biased. People tend to feel more than what they show by their rating.

## Word Count and word cloud

[Plot – 2: Count of Words in Reviews] [Plot – 3: Word Cloud for Reviews]

Words like echo, music, Alexa have appeared a greater number of times, which is obvious. So, in the next step, we remove the most obvious words and look at the word cloud.

[Plot – 4: Count of relevant words in Reviews] [Plot – 5: Word Cloud for relevant words]

## Word Cloud: Positive/negative word cloud segregation

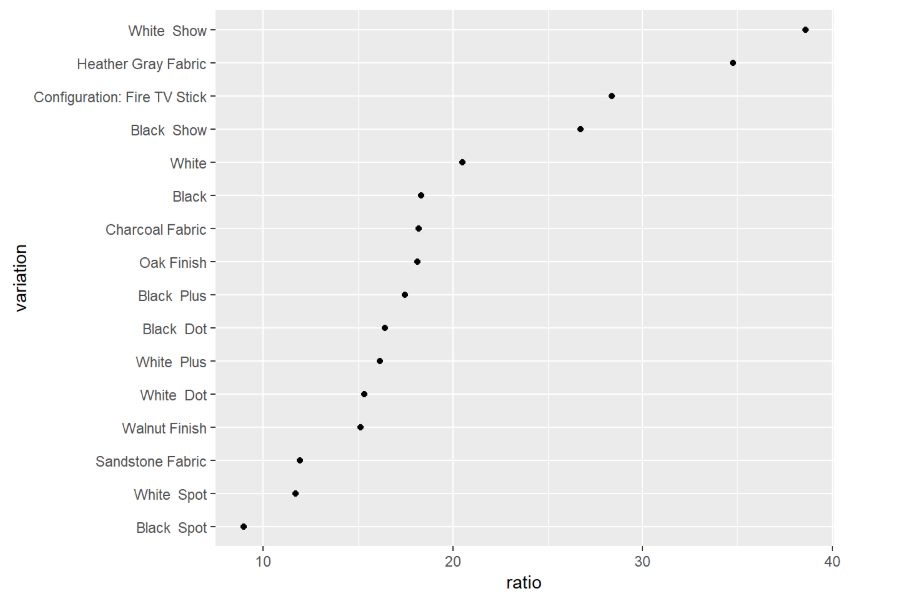
 

[Plot – 6: Positive/Negative Word Cloud] [Plot – 7: Negative Word Cloud]

We observe that positive words (in green) like **love** and **great** are dominating. However, there are also negatives ones (in red) like- **disappointing**, **frustrating**, **disabled**.

## Sentiment Ratio

Sentiment Ratio = Number of positive words for the product / Number of negative words for that product.

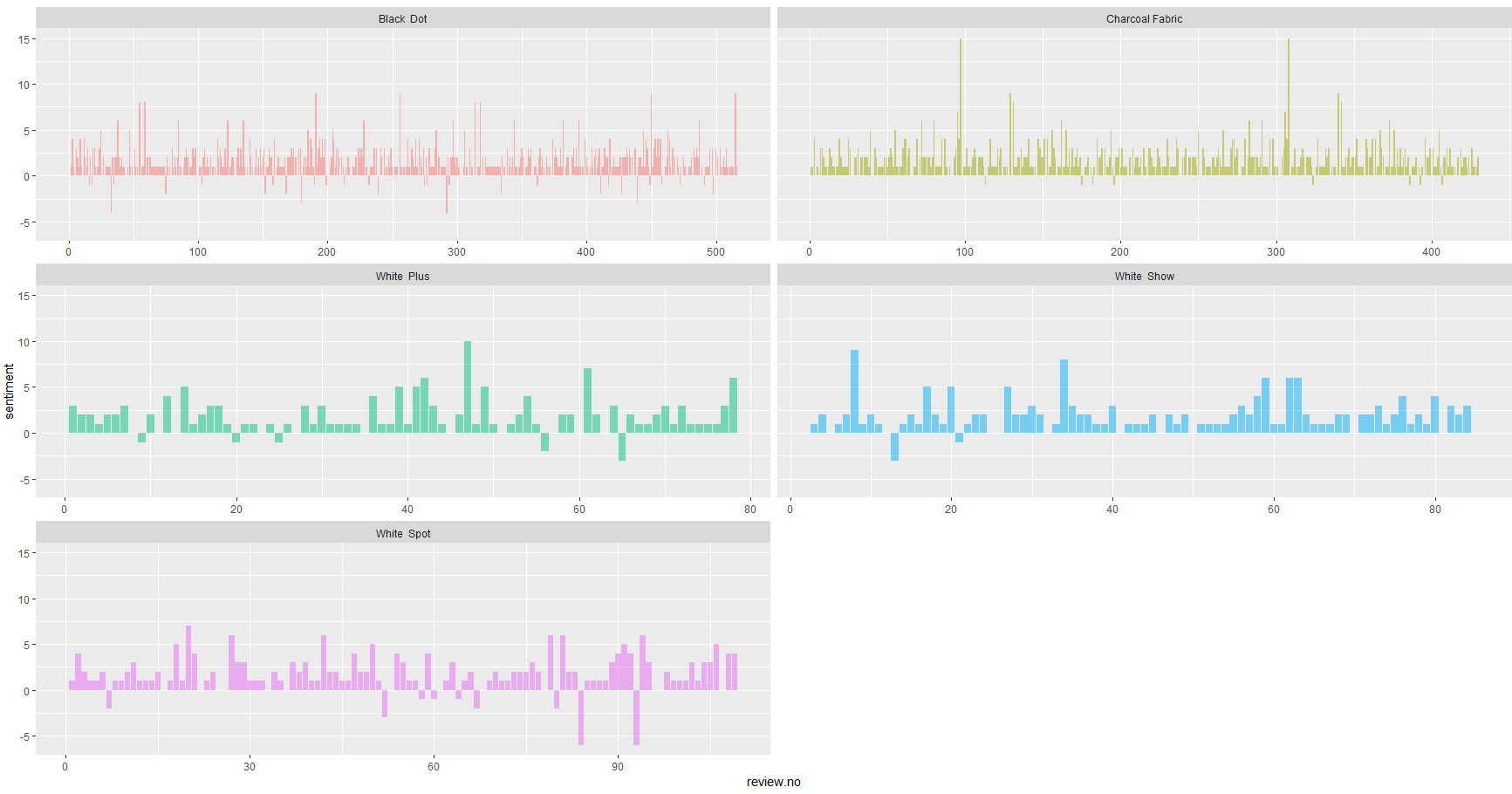


[Plot – 8: Sentiment Ratio Plot for each Product]

Looking at the positive/negative ratio across the different Alexa products, we find that **White Show** is appreciated the most whereas **Black Spot** isn’t taken that well.

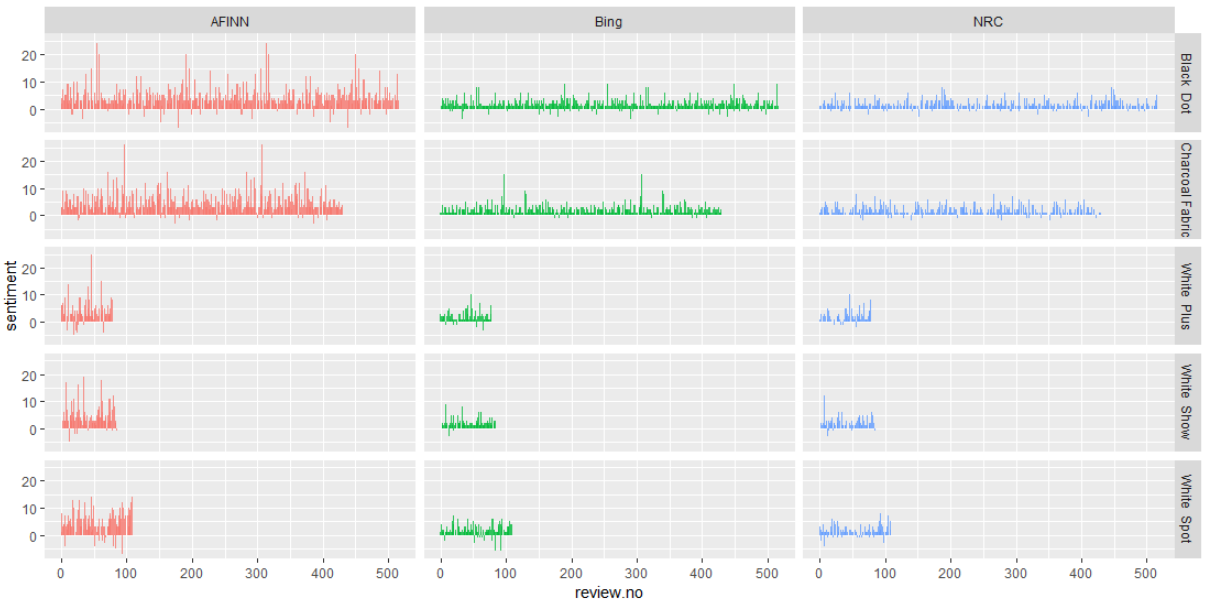
# Sentiment analysis using different lexicons

We performed sentiment analysis on the Alexa dataset using different lexicons, namely, AFINN, Bing and NRC. The results were as below:



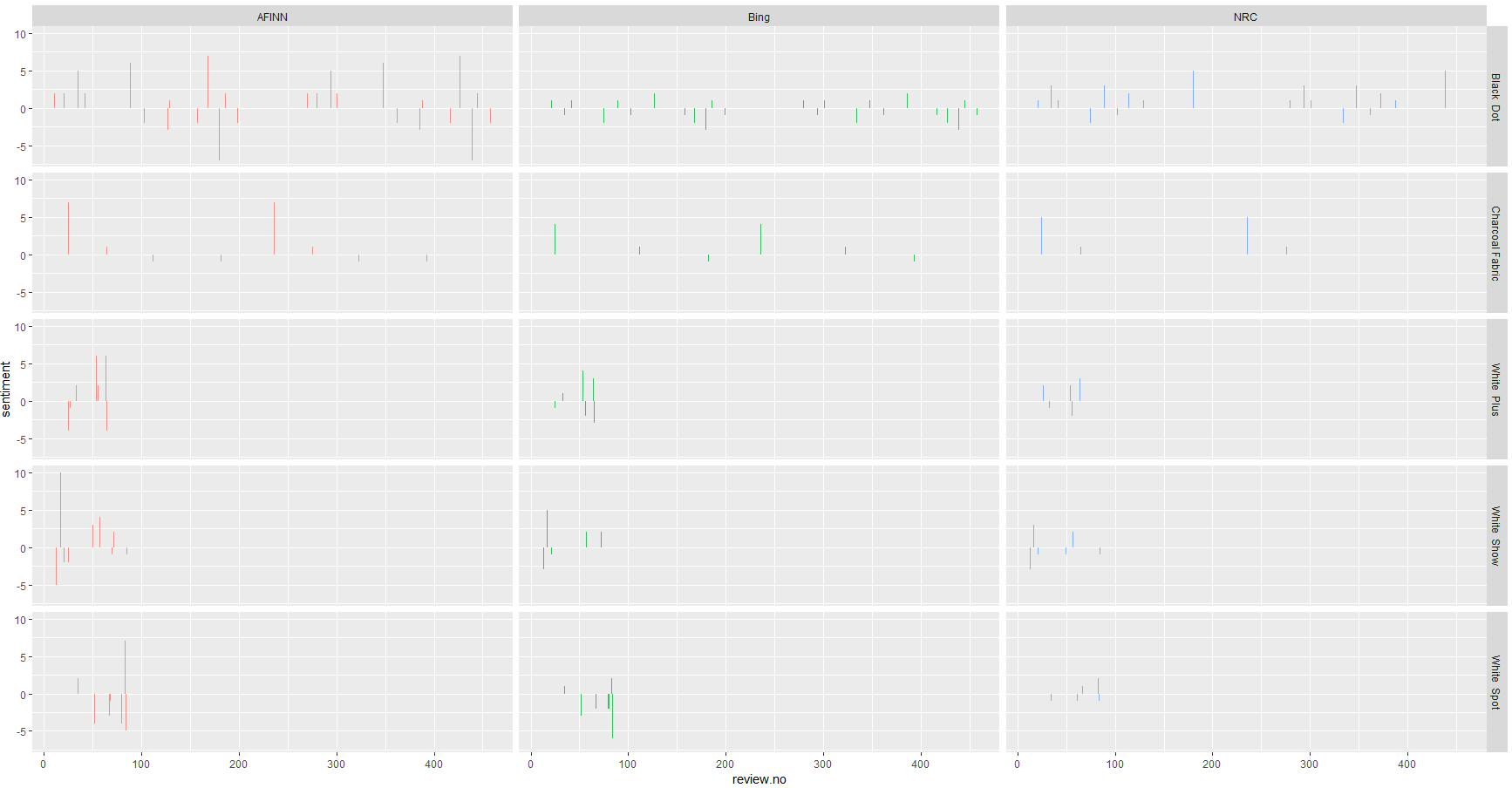
[Plot – 9: bing lexicon]

The above plot shows the Sentiment Score for all reviews each product using bing lexicon. The score above zero shows a positive review and below that indicates a negative review.



[Plot – 10: Sentiment score of Product reviews using lexicons]

The plot 10 shows the Sentiment Score for all reviews of 5 products using 3 different lexicons. We observe that the results look similar from all lexicons.



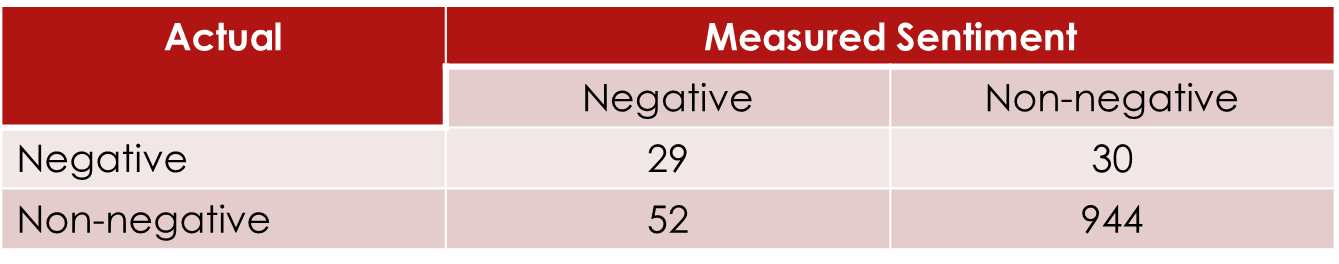
[Plot – 11: Sentiment score for negative reviews]

Next, we analyzed only the negative reviews for all products, as shown in plot 11. We see that the negative reviews are sparse. Also, there is some clear misclassification observed from the plot, since the negative reviews have been classified as a positive sentiment in our analysis.

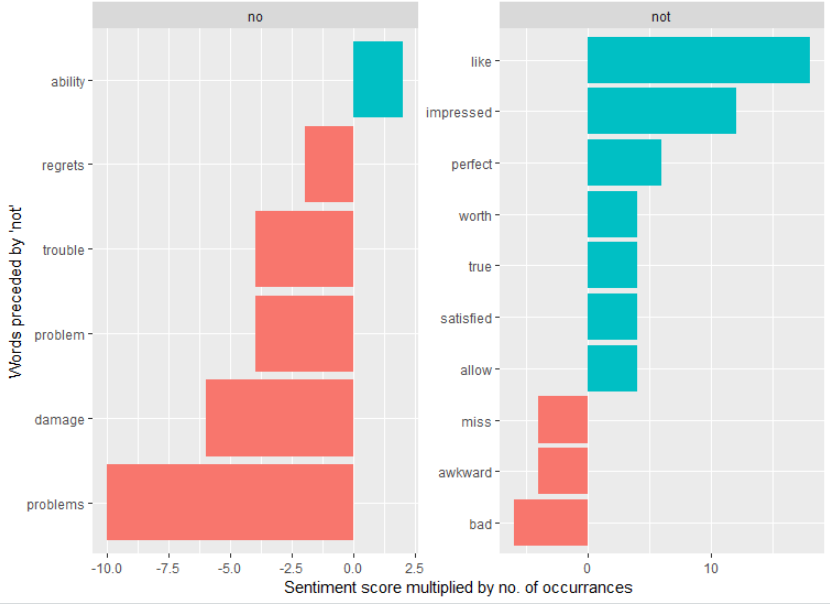
# Measuring performance of our sentiment analysis

The customer review range was 1-5. If actual customer review <=2, then that review has been considered as negative. Otherwise non –negative. Similarly, if sentiment score < 0 then negative review, else non-negative. The sentiment analysis performance was as below:

[Table – 2: Sentiment Analysis confusion matrix]



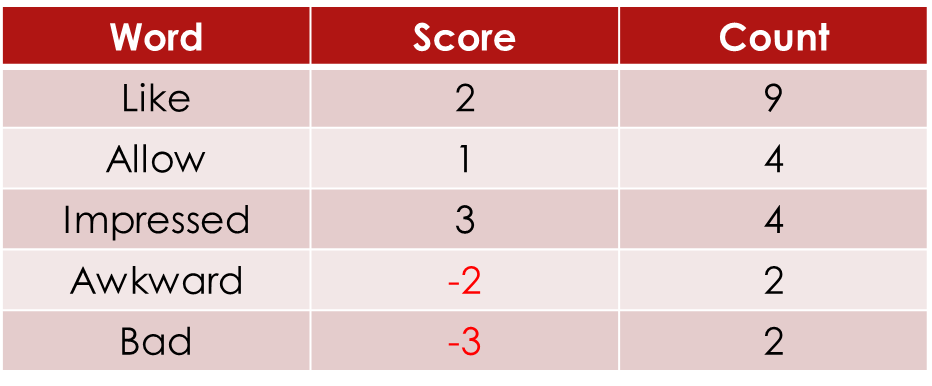
The misclassification rate 7.7%, which is very less. But we only have a few negative reviews and hence the case.



[Plot – 12: Negated words]

We can see the total impact the negated cases had on mis specifying sentiment. For example, we see that the topmost word preceded by “not” is “like”. The sentiment score for “like” is +2; however, “like” was preceded by “not” 9 times which means the sentiment could easily have been overstated by 9 × 2 = 18 points.

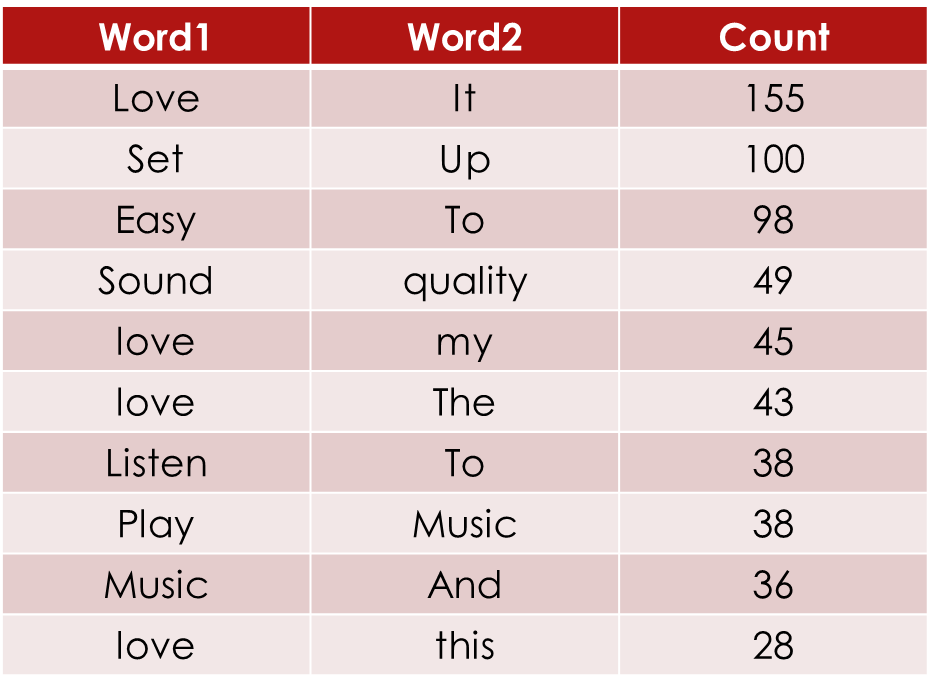
[Table – 3: Words preceded by “not”]



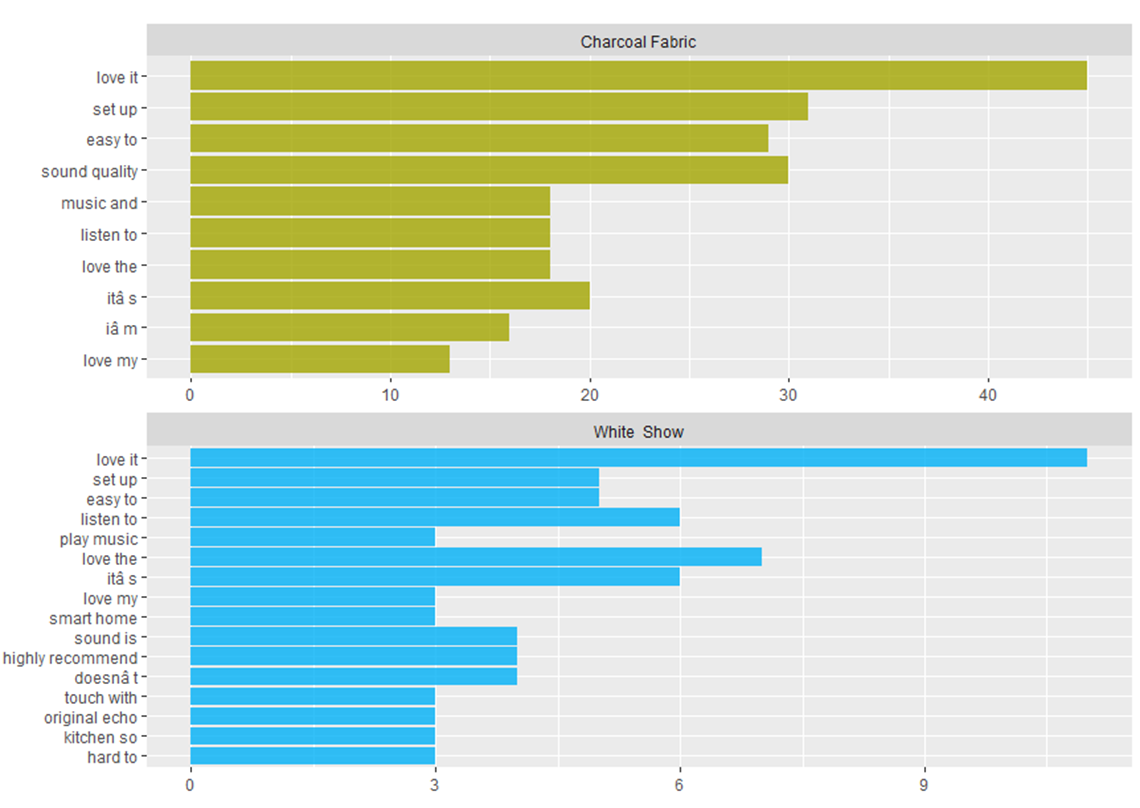
# Bigrams

We then had a look at the Bigrams apart from the negated words, that is, those words which appeared most together.

[Table – 4:Bigrams with their frequencies]



As we see, the words ‘love it’ and ‘set up’ have been used the most in reviews by customers, followed by ‘easy to’ and ‘sound quality’.

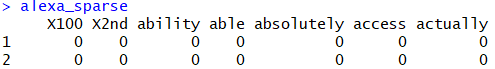
[Plots – 13, 14: Bigrams for different variations]

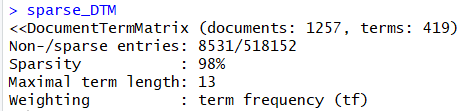
Plots 13 and 14 show the top bigrams for different variations of Alexa. The most repeated combination of words for each variation is ‘love it’, followed by ‘set up’.

# Model

* Reduced the given data to a document term matrix form with all the words in the column and reviews in rows. If a review had a word, there would be a matching 1 otherwise 0
* Sparsity = 98%, which means that we kept only those terms (words in columns) that appear in >=2% or more of the reviews
* Used customer reviews as predictor – converted the reviews to negative or non-negative using the rating (if rating <=2, then “negative” else “non-negative”)

The below shows the sparse data with a column for each word used in the reviews

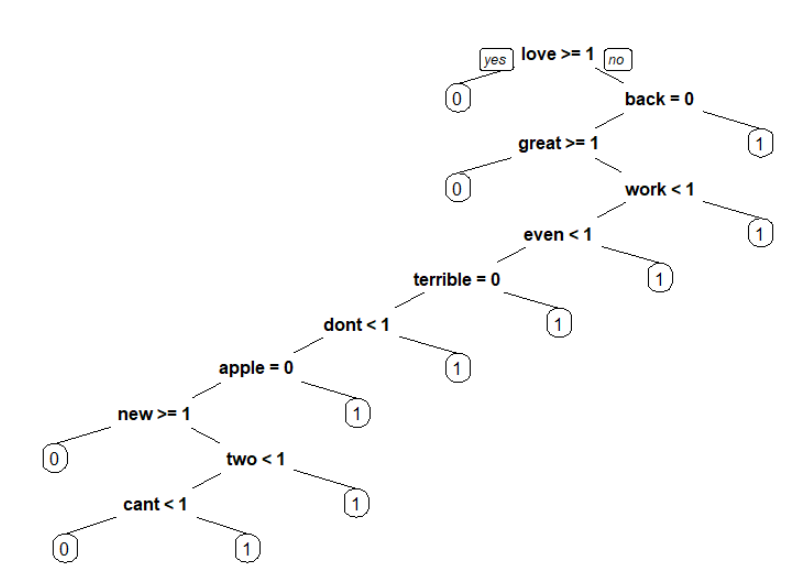




## Decision tree

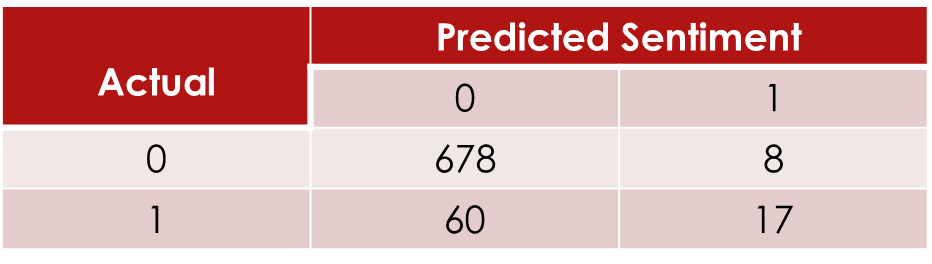
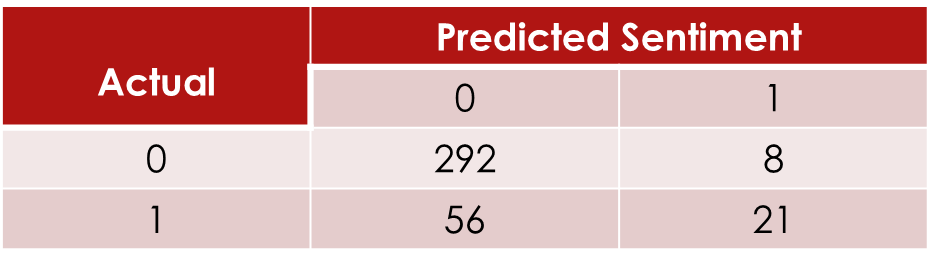
We initially fit a decision tree to the data and the results we as below (plot 15). The interpretation of the tree is as follows:

* If “love” is in the review, then we go left – meaning non-negative. Else we go right.
* If “love” is not in the review and if the word back is there in the review, then we go right, meaning, negative review. The interpretation is similar for the entire tree.

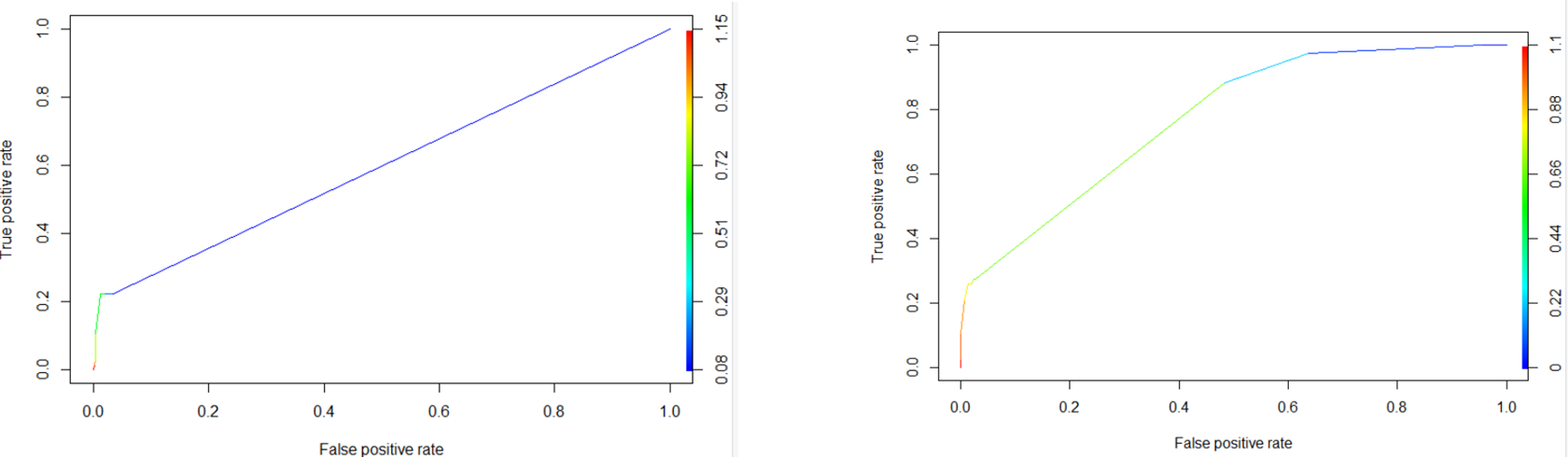


[Plots – 15: Decision tree]

Table 5: Full data Table 6: After Undersampling

The confusion matrix for the test data is shown above. Using the full data, we got a misclassification rate of 8.9%. The MR is low only because we have very few ones in the data. Model performance not that good. Then we performed undersampling to improve prediction accuracy of negative reviews, as shown in table 6, and got a misclassification rate of 16.9%.



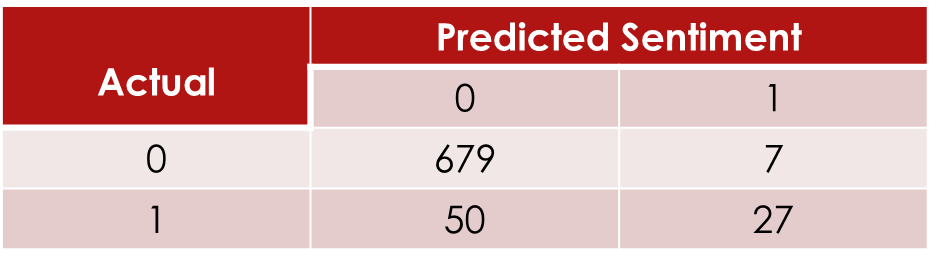
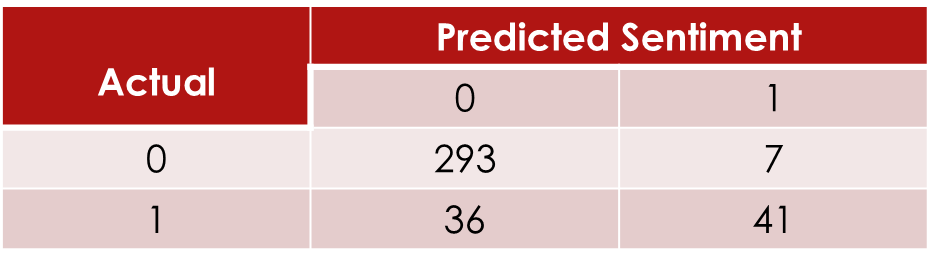
[Plot – 16: ROC for full data] [Plot – 17: ROC for undersampled data]

The AUC plots for full data and undersampled data is shown above. Full data had an AUC of 0.59, whereas the model performed well undersampled data, having an AUC of 0.77.

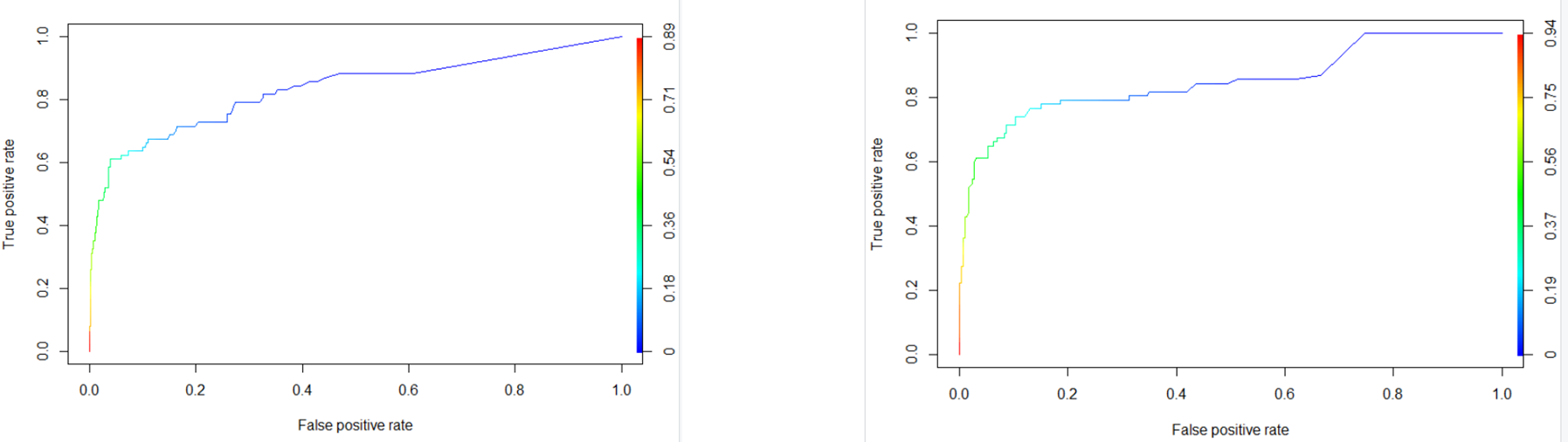
## Random Forest

Then we used random forest model on the full data as well as the undersampled data, and the results are as below.

Table 7: Full data Table 8: After Undersampling

The full data had a misclassification of 7.47% and the undersampled data had a misclassification of 11.1%.



[Plot – 18: ROC for full data] [Plot – 19: ROC for undersampled data]

The AUC plots for full data and undersampled data is shown above. Full data had an AUC of 0.83, whereas the model performed well undersampled data, having an AUC of 0.85. Thus, in both random forest and decision trees, we observe that the model performed better with undersampled data performed than with the full data.

# Conclusi0n

* Sentiment Analysis using lexicons AFINN, Bing and NRC performed better compared with statistical modeling techniques to predict the sentiment
* Random Forest performs better than Decision Trees – both for full data and for undersampled data

# References

* Dataset: Kaggle ([Link](https://www.kaggle.com/sid321axn/amazon-alexa-reviews))
* University of Cincinnati – R, Github ([Link1](https://uc-r.github.io/tidy_text), [Link2](https://uc-r.github.io/sentiment_analysis))
* RPubs – Anil Kumar ([Link](https://rstudio-pubs-static.s3.amazonaws.com/126346_fd9e1e5dfbe241b3be65e8b2053c3c50.html))
* Random Forest ML ([Link](https://www.youtube.com/watch?v=3kYujfDgmNk&t=15s))