sentiment analyis & DEEP LEarning on Amazon product reviews

**Kiran Talreja**

**Bashaer Alkhathlan**

**Chandni Sharma**

**Abhijit Shinde**

# 

# Table of Contents

Contents

[Table of Contents 1](#_Toc469440122)

[Table of Figures 2](#_Toc469440123)

[Problem 3](#_Toc469440124)

[Analysis 3](#_Toc469440125)

[Scraping Reviews Process 3](#_Toc469440126)

[Methods 4](#_Toc469440127)

[Sentiment Analysis 4](#_Toc469440128)

[Plots and Tables 5](#_Toc469440129)

[Sentiment Analysis in Tableau 7](#_Toc469440130)

[Doc2Vec/Text2Vec 10](#_Toc469440131)

[Tuning Text2Vec & LDA 10](#_Toc469440131)

[DBN 10](#_Toc469440132)

[SVM 10](#_Toc469440133)

Performance [10](#_Toc469440134)

[References 10](#_Toc469440135)

# **Table of Figures**

[Figure 1 Reviews.scores for 1000+ Reviews 5](#_Toc469440101)

[Figure 2 Histogram for Reviews and Their Scores (Reviews.scores) for 1000+ Review 5](#_Toc469440102)

[Figure 3 Calculating Emotion and Polarity for Each Review 6](#_Toc469440103)

[Figure 4 Ggplot of Reviews Classified by Emotion 6](#_Toc469440104)

[Figure 5 Ggplot of Reviews Classified by Polarity 7](#_Toc469440105)

# 

# **Problem**

* Applying sentiment analysis to scraped reviews from the web using R
* Label positive/negative words from each review of scraped corpus
* Vectorize each review in corpus using Doc2Vec and apply both DBN/RBM and SVM to compare performances

# **Analysis**

* Amazon Reviews **dataset** consists of the columns:
* Text,
* Score,
* Emotion,
* and Polarity (Positive, Negative, and Neutral) for a 1000+ reviews.
* **Text column:** contains all scraped reviews of an Amazon product
* **Score column:** contains the scores of the positive/negative words in each review. Scores are calculated by using the function score.sentiment that takes sentences from product reviews and compare it to positive/negative wordlist dictionaries
* **Emotion column:** classify\_emotion function has been used from the Sentiment package to classify the emotion (e.g. anger, disgust, fear, joy, sadness, surprise, or unknown) of a set of reviews using a naive Bayes classifier
* **Polarity column:** classify\_polarity function has been used from the Sentiment package to classify the polarity (e.g. positive, neutral, and negative) of a set of reviews using the algorithm “bayes”

# **Scraping Reviews Process**

* We have created a scraper method in which using SelectorGadget tool in the browser, we figured out what selectors were used for which part of the web page. And as we are using reviews of the products so the comments of the users and their ratings were to be considered. For that, we took the url of that page and the product code. Then, considering the pages we want to read from, used the ‘read\_html()’ function to read all the comments of the users.
* For that, using html\_nodes() method we selected what css selector had what type of data like the author, title of the comment, etc and Converting that into html\_text.

# **Methods:**

## **Sentimental Analysis**

* Score.sentiment function has been created for this part, it basically takes each sentence, positive words, and negative words from each product review and then process them to calculate and return Scores data frame. This function first cleans sentences using gsub() function and then applies tolower() function to force sentences to appear in a lower case format. Then, the function splits sentences into words using the function str\_split() and then it unlists words to be compared and matched with dictionaries of positive/negative words. Then finally, and to calculate the scores of all reviews, the technique: score = sum(pos.matches) - sum(neg.matches) is used. True-false will be treated as 1/0 by the function sum() in this case.
* To plot the scores of reviews, we took all the product reviews and prepared them for plotting by removing any word punctuation and empty spaces between words. A function to handle any errors was also applied, it can handle errors such as dealing with missing values.
* To calculate emotion within reviews, classify\_emotion function was used from the Sentiment package to classify the emotion (e.g. anger, disgust, fear, joy, sadness, surprise, or unknown) of a set of reviews using a naive Bayes classifier with prior=1.0
* To calculate polarity within reviews, classify\_polarity function has been used from the Sentiment package to classify the polarity (e.g. positive, neutral, and negative) of a set of reviews using the algorithm “bayes”
* The results of both classifications were placed inside a data frame to be further plotted. For plotting, ggplot() function was used to give a clear representation of the sentiment analysis process that was applied on 1000+ Amazon product reviews

### 

### **Plots and Tables**



Figure 1 Few Reviews.scores for 1000+ Reviews

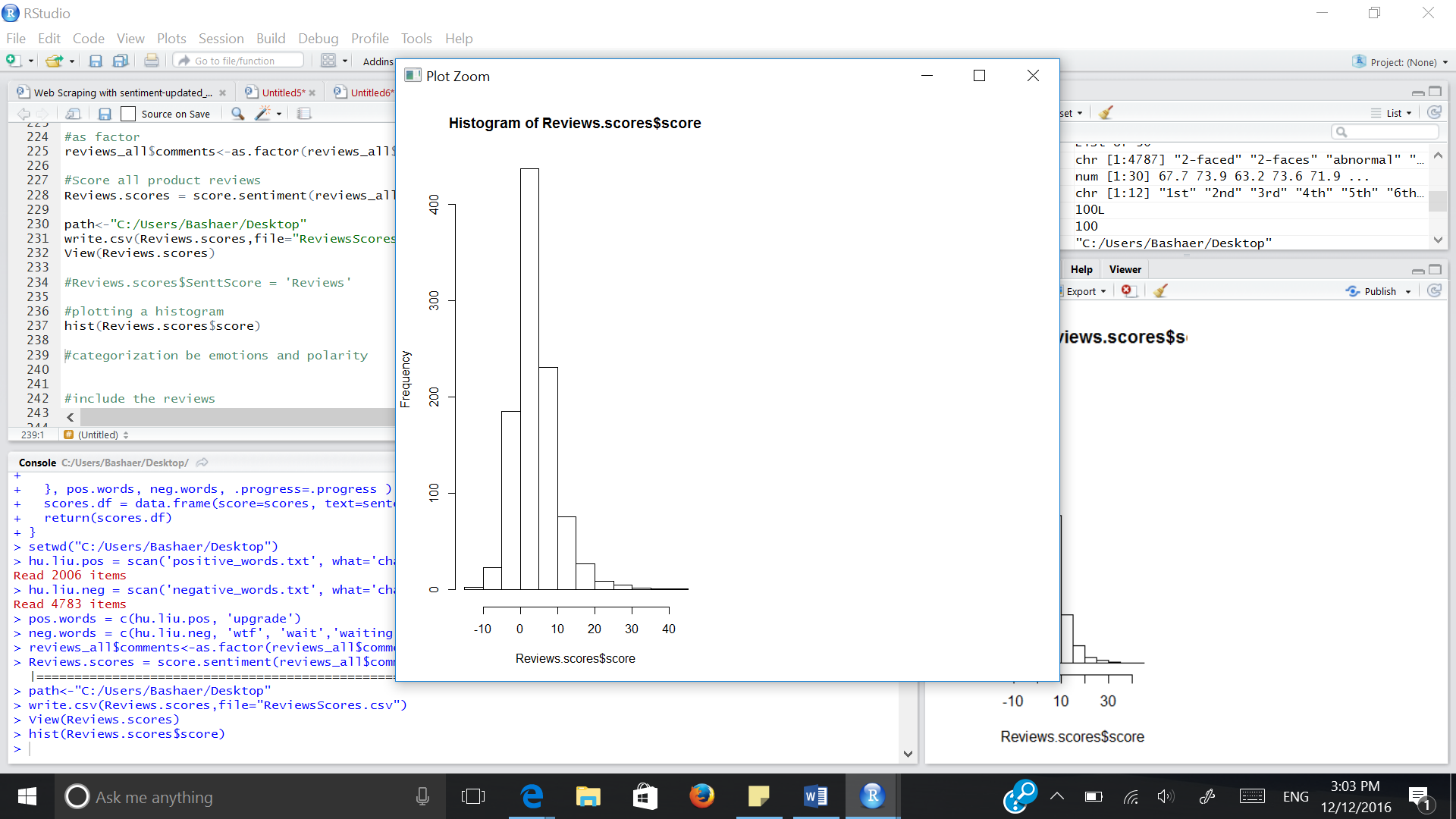


Figure 2 Histogram for Reviews and Their Scores (Reviews.scores) for 1000+ Review

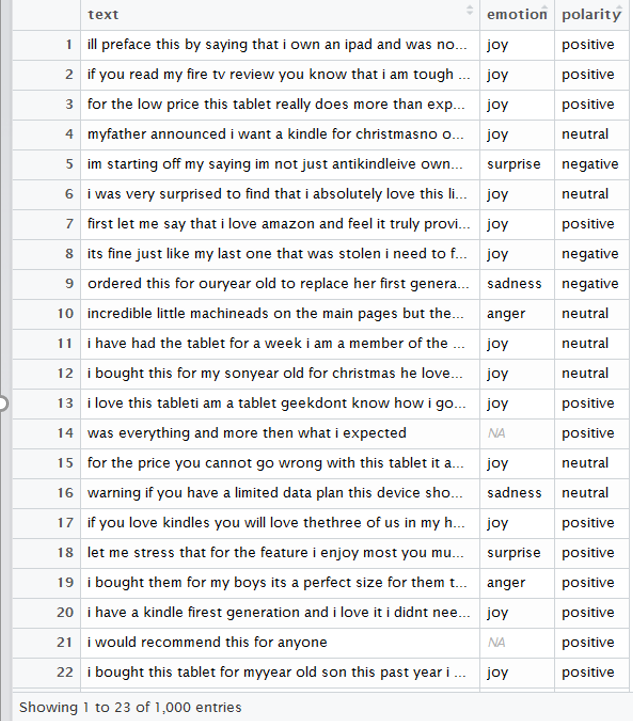


Figure 3 Calculating Emotion and Polarity for Each Review

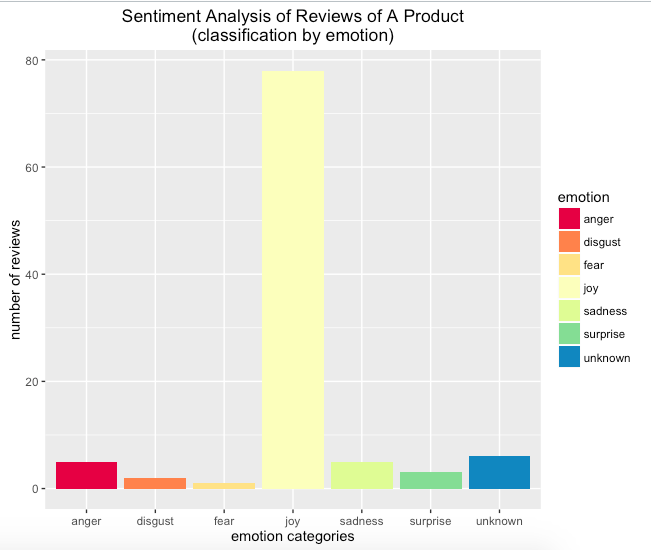
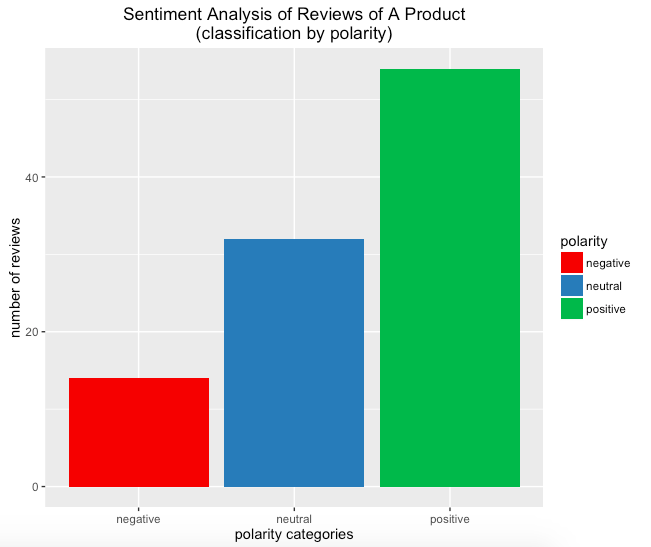


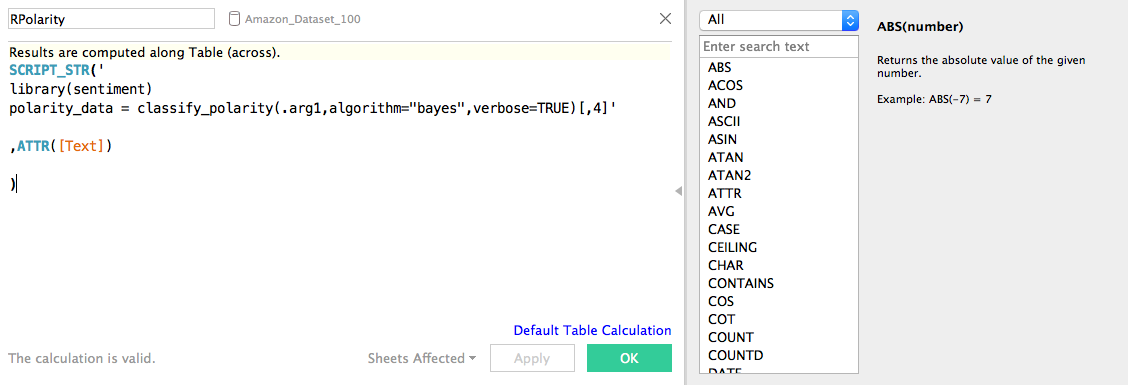
Figure 4 Ggplot of Reviews Classified by Emotion

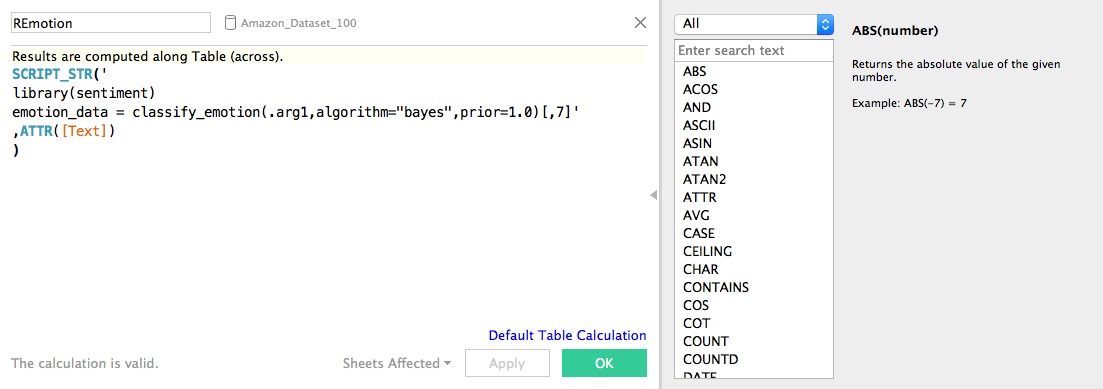


### Figure 5 Ggplot of Reviews Classified by Polarity

### **Sentimental Analysis in Tableau**

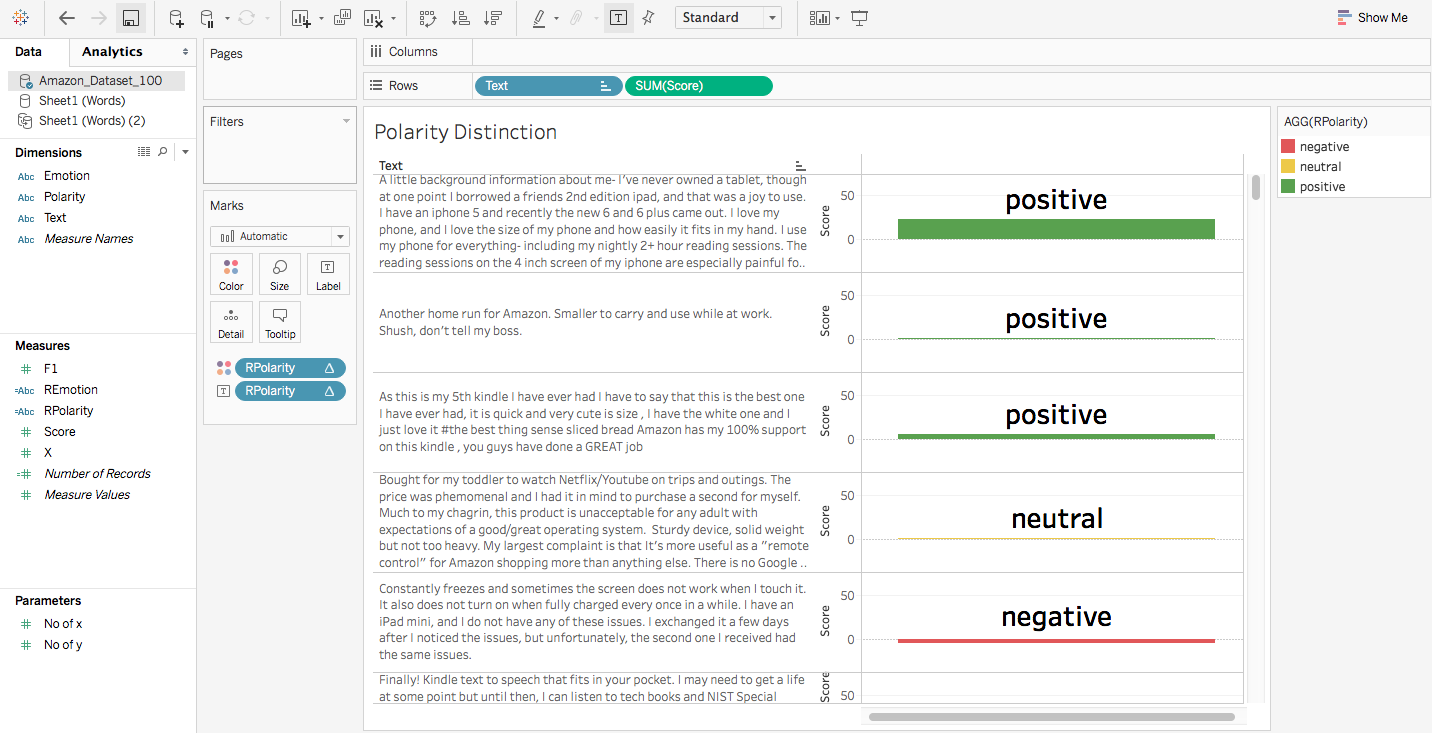
### In Tableau, we uploaded the csv file containing all the reviews of the customers. Also, we did **Integration of R with Tableau** by using scripts to do sentimental analysis as follows:





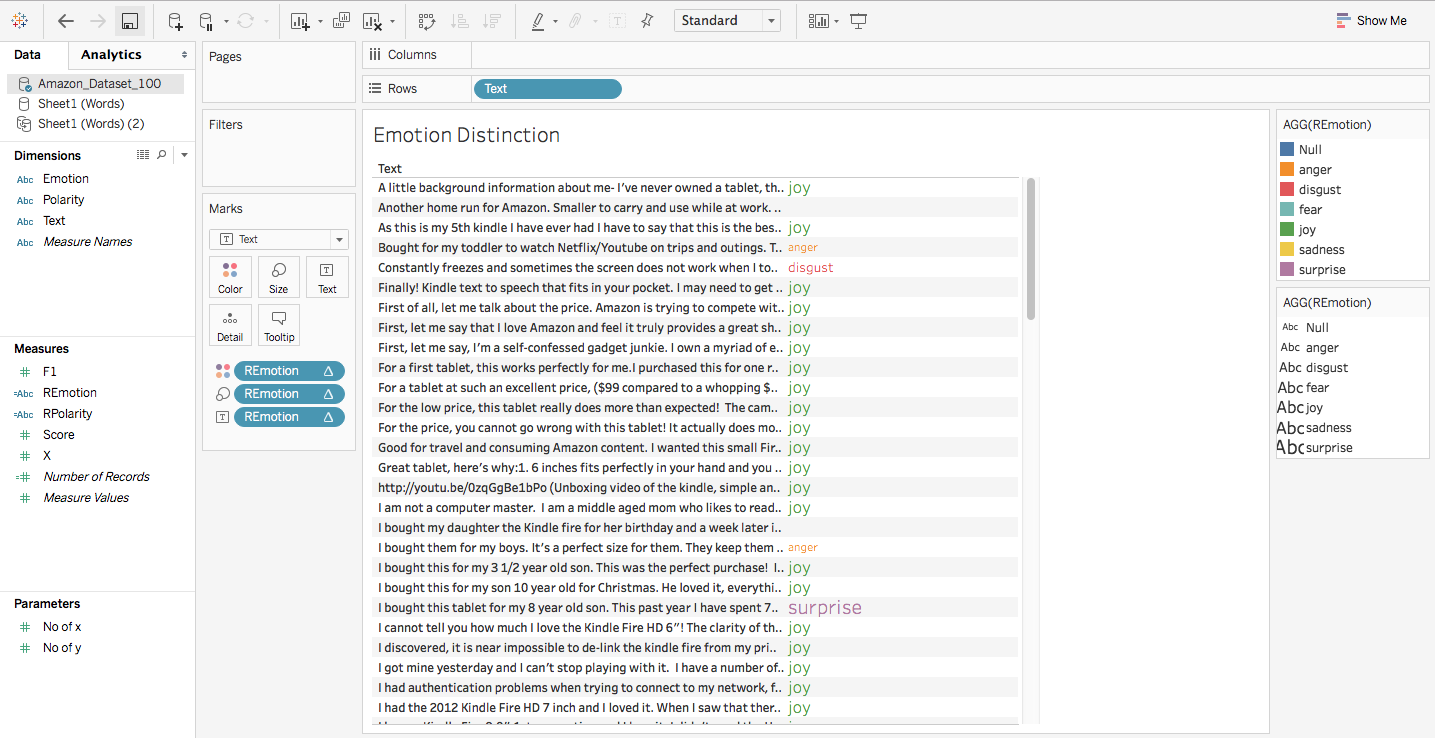
First we created the connection between R and Tableau by using the library ‘**RServe**’ in R and then connecting tableau to that server. In REmotion and RPolarity calculated fields, I used the SCRIPT\_STR function to classify emotion or polarity of the ‘sentiment’ library along with using the attribute from the csv file, ‘Text’. This produced the same results as in R scripting.

**Polarity Distinction:**



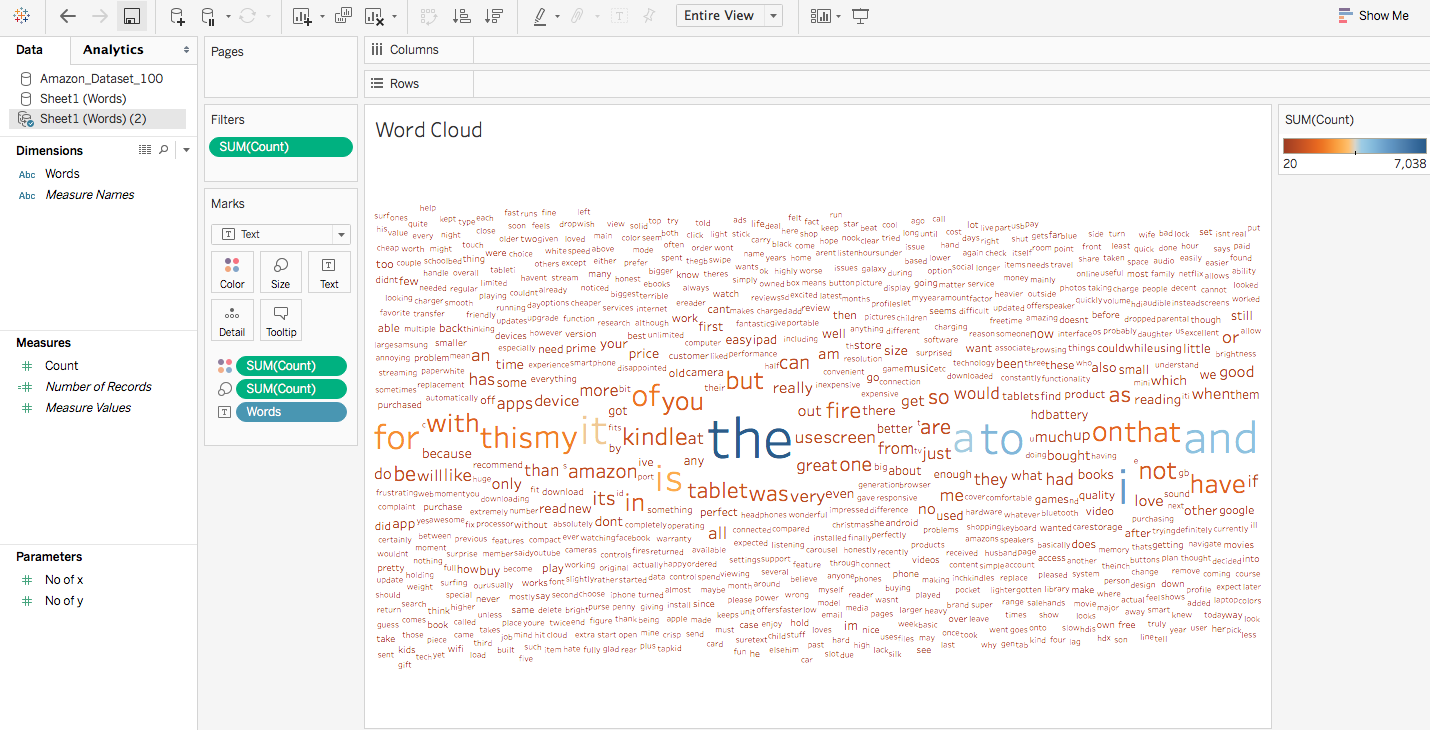
in this, we have used **RPolarity** calculated field to show the label of Polarity along with what color it can be associated with. We have also used score to show the relative scoring of those reviews.

**Emotions Distinction:**



In this, we have used **REmotion** calculated field to show the label of Emotion along with color. Also, the size is relative, meaning if that emotion is more, then that emotion word is shown bigger in size.

**Word Cloud:**



This is a Word Cloud with the count of all the words used in all the reviews. We have set the filter to show only those words which has more than 20 count. For this, we used another excel file containing only words and their counts. This file was generated using a **split\_text\_tool**.

## **Doc2Vec**

We have built Text2Vec for our **Documents to Vector conversion**. The main package used is **text2vec** in R and especially provides tools for Text Analysis & NLP.

**Vocabulary based Vectorization:**

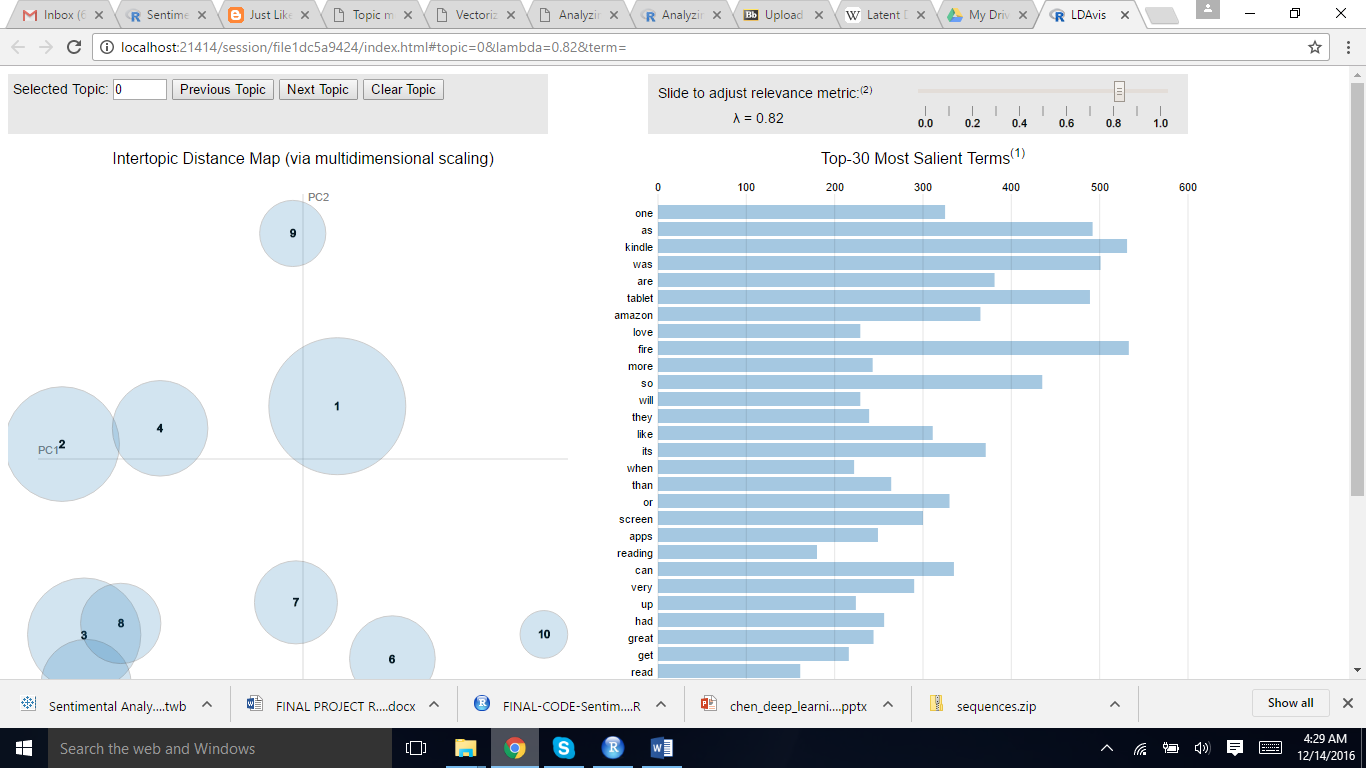
* Fast text vectorization using Vocabulary.
* We insert documents into Corpus which are nothing but C++ classes.
* Then we created DTM (Document Term Matrix) from the input documents.
* In DTM, we vectorize the text by creating mappings.
* Copy-on-modify semantics involves reading the documents in RAM but text2vec is far better.

**Tuning Text2Vec**

* We further Pruned the Vocabulary by removing the STOP WORDS (useless words) for better performance.
* We then apply **TF-IDF (Term Frequency – Inverse Document Frequency) transformation**, which increased the weight for document-specific terms and decrease weight for widely used terms.
* We observed that the columns reduced after performing Pruning since we removed many of the unnecessary words and it was helpful for our further modelling.

**LDA (Latent Dirichlet Allocation)**

* LDA is a statistical model which helps in telling the observations of the words like Salient terms (most noticeable or important words)
* The plot runs on the Web browser locally connected from the Rserver and shows the Top 30 Most Salient Terms in the dataset with an adjustable relevance metric (Lambda)



## 

## **DBN/DNN**

## We used the Darch package of R to perform Deep Belief Network (DBN/DNN).

Following are the steps:

1.       Divided Input and output variables

2.       Using **darch** function, we created the DBN/DNN model

3.       The parameters for above function has layers with 1026 inputs and 1 outputs and 4 layers

4.       Following are the steps which are performed in Darch

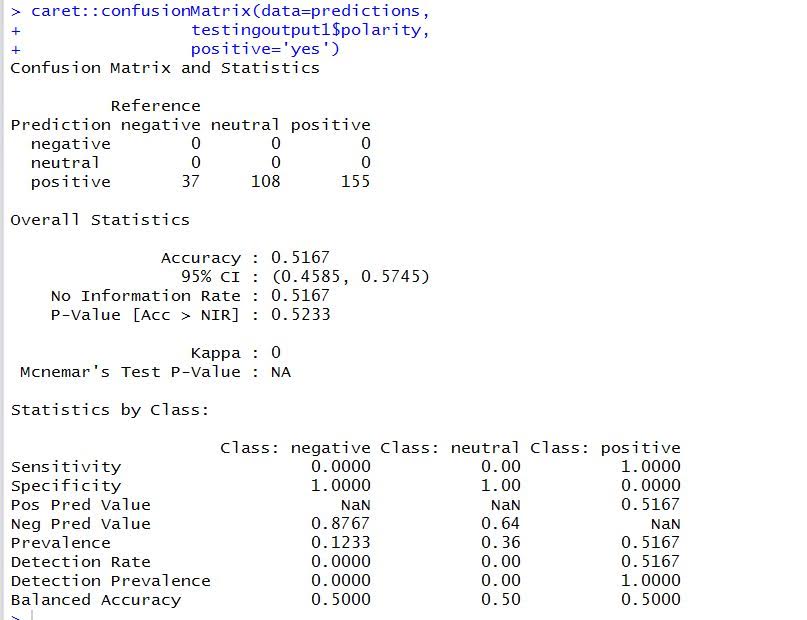
·       Create and configure a Darch instance.

·       Pre-train the network.

·       Fine-tune the network.

·       Back-propagate further data through the network to create predictions and Retuning.

## **DBN/DNN Performance Results: (Confusion Matrix)**

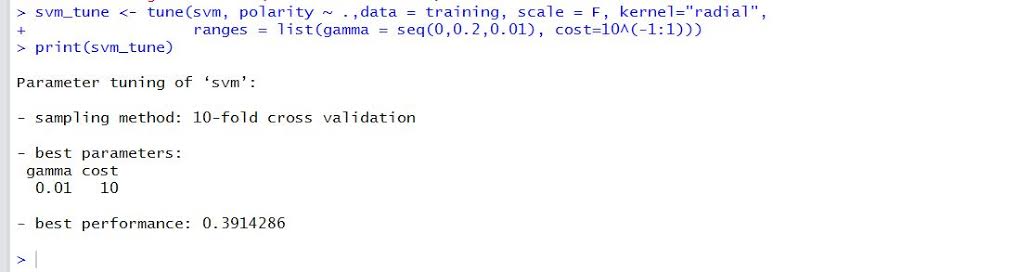


## 

## **SVM**

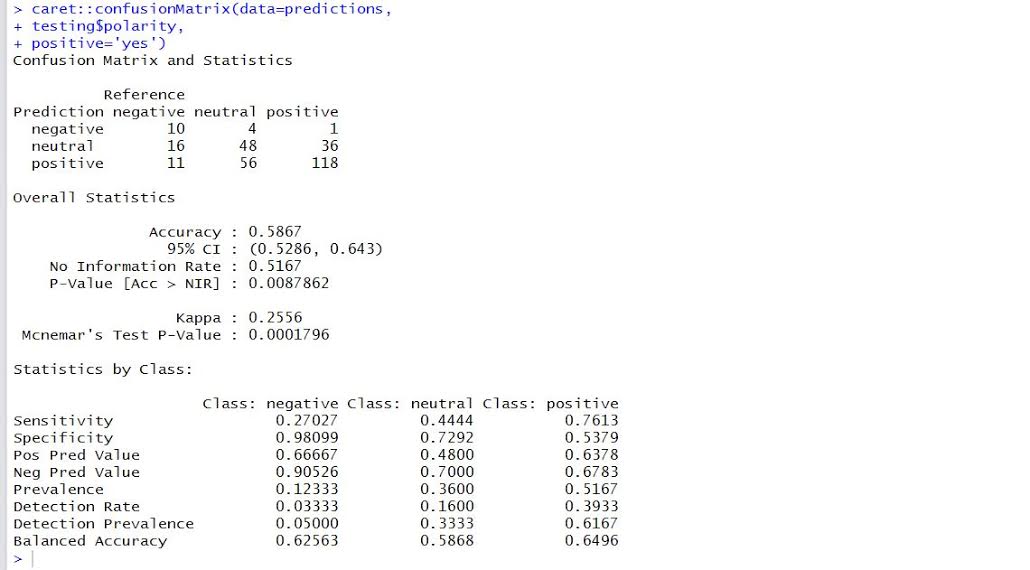
* SVM is a classiﬁcation technique that seeks to ﬁnd a hyperplane that partitions the data by their class labels
* It avoids over-ﬁtting the data by maximizing the margin of the separating hyperplane.
* SVMs are useful in Text & Hypertext Classification.
* SVM is known to significantly reduce the need for labels in the training instances.

Further, we calculated the Best Parameters for tuning the SVM (below results):



And then we tuned the SVM to get better accuracy.

## **SVM Performance Results: (Confusion Matrix)**



# **Performance Comparison between Models (Evaluations)**

We have evaluated the accuracy of the models based on the Confusion Matrix.

Interestingly, the tuned SVM model performed better with the accuracy of 59% as compared to the Deep Belief Network’s accuracy of 52 %.

|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| DNN | 52% |
| SVM | 50% |
| Tuned SVM | 59% |

# **References**

<https://cran.r-project.org>

<https://en.wikipedia.org/wiki>

<https://iralab.unt.edu/sentiment-analysis>

<https://boraberan.wordpress.com/2013/12/24/sentiment-analysis-in-tableau-with-r/>

<http://www.theinformationlab.co.uk/2016/03/02/text-analysis-101-sentiment-analysis-in-tableau-r/>

<http://alexloth.com/2016/01/31/how-to-implement-sentiment-analysis-in-tableau-using-r/>

<http://www.clearlyandsimply.com/clearly_and_simply/2015/03/word-clouds-with-tableau.html>