’

Capstone Project (E-Commerce)

Name: Abheer Bandodker

G2-PGPBABIO-May

Batch A

1. Introduction

Problem Statement:

The dataset is comprised of 23486 rows and 10 feature variables related to customer review of the products offered by the company. These variables are clothing ID, age, title, review Text, rating, recommended IND, positive feedback count, division name, department name and class name. Study all variables provided in the dataset to understand the customer behaviour associated with products within the categories (Division, Department & Class).

Need to study the problem (Objective):

I. Understanding the positive/negative sentiments associated with various women's clothing products present in the dataset
II. Building a predictive model to predict whether a customer will recommend the product/company services to other potential customers
III. Study the correlation among different variables to understand the conclusions derived from predictive model
IV. Provide recommendations that can help retailer maximize their profits, sales and customer satisfaction
V. Analyze the variables that are most significant/critical to enhance the business performance and efficiency

Understanding business/social opportunity:

The model will provide the retailer the opportunity to understand the products with high demand among customers. This will provide the retailer with actionable insights related to inventory management by efficiently investing in only those products that are increasingly popular among its customers.

2. Data Report:

Data collection methodology in terms of time, frequency and methodology:

The data was collected through the e-commerce platform customer relationship management system. The dataset comprised of unstructured and structured data/variables.

Visual inspection of the attributes:

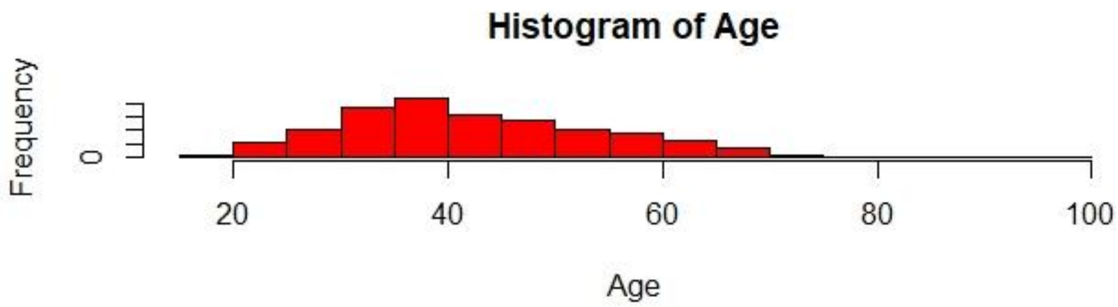
1) **Clothing ID: (Type Integer)** Integer Categorical variable that discusses to the explicit piece being reviewed.
2) **Age: (Type Integer)** Positive Integer variable of the reviewers age.
3) **Title: (Type Factor)** String variable for the title of the review.
4) **Review Text: (Type Factor)** String variable for the review body.
5) **Rating: (Type Integer)** Positive Ordinal Integer variable for the product score granted by the customer from 1 Worst, to 5 Best.
6) **Recommended IND: (Type Factor)** Binary variable stating where the customer recommends the product where 1 is recommended, 0 is not recommended.
7) **Positive Feedback Count: (Type Integer)** Positive Integer documenting the number of other customers who found this review positive.
8) **Division Name: (Type Factor)** Categorical name of the product high level division.
9) **Department Name: (Type Factor)** Categorical name of the product department name.
10) **Class Name: (Type Factor)** Categorical name of the product class name

Note: Actions have been taken to transform the data as mentioned below

1) The title and review text variables with factor type have been to transformed into character type
2) The names of categories such as Division Name, Department Name and Class Name have been changed to Division, Department and Class respectively
3) Similarly, name of Recommended IND, Review.Text & Positive Feedback Count have been changed to Recommended, Review_Text and Positive_Feedback_Count

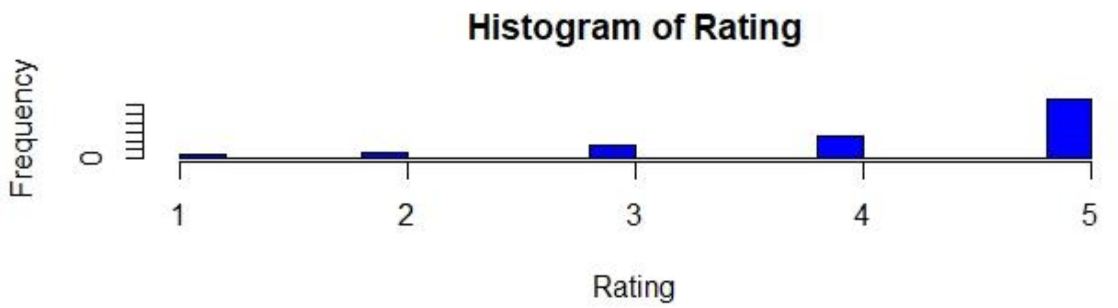
3. Exploratory Data Analysis:

Univariate Analysis:



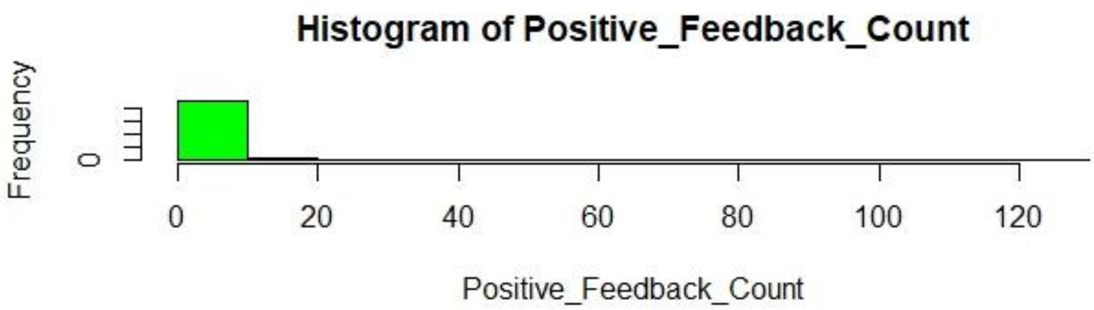
Insights:

1) The above data shows the distribution of age parameter
2) As depicted in the graph the distribution follows a normal distribution. Most of the customers are between the 35 to 45
3) This also provides the company with insights regarding the strategy it should perceive in terms inventory management and promotional offers and development of product combos



Insights:

1) The above data shows the distribution of Rating parameter
2) As depicted in the above graph majority of the customers have allotted 4 and 5 rating to the company. This provides a clear outlook as to how the company is being perceived by its customers.

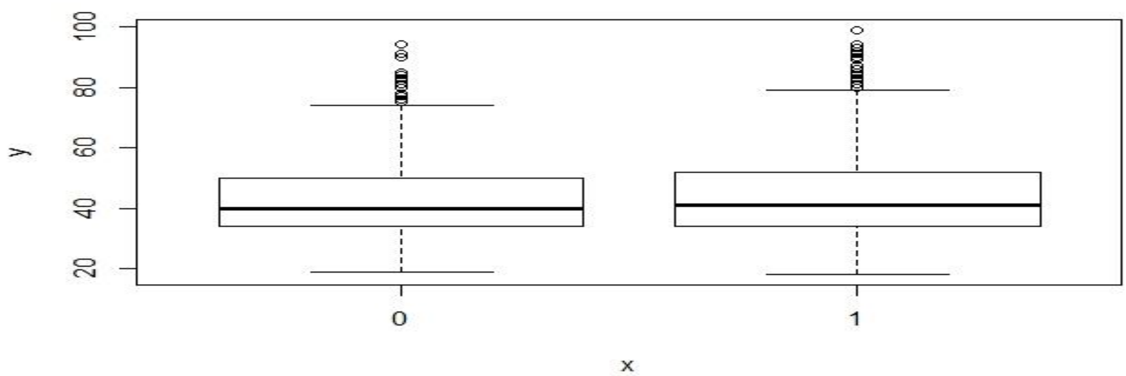


Insights:

1) The above data shows the distribution of Positive_feedback_Count parameter
2) As depicted in the above graph majority of the reviews have been allotted a positive count between 0 to 10. This parameter has shown a descriptive viewpoint as to whether the reviews provided the customers have a positive or a negative impact on the purchases made by other customers.

Bivariate Analysis:

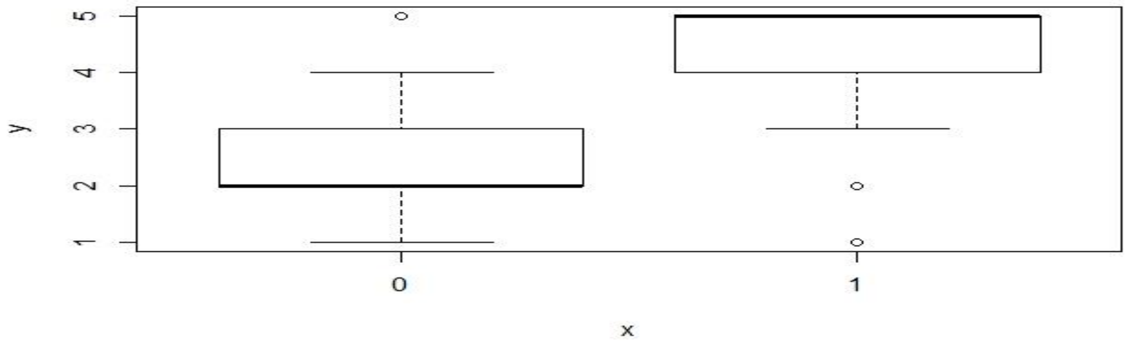
Age vs Recommended



Insights:

1) The above boxplot shows the distribution of age parameter among the customers who have the recommended the products and not recommended the products
2) Clearly, as the distribution is similar to among both the categories, indicating that age is not a significant factor in deciding whether a customer will recommend the product or not

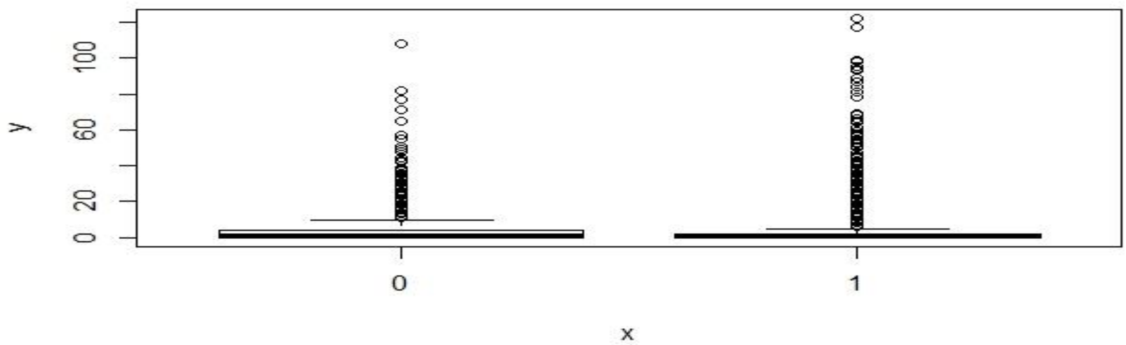
Rating vs Recommended



Insights:

1) The above boxplot shows the distribution of Rating parameter among the customers who have the recommended the products and not recommended the products
2) Moreover, as the distribution is very distinct among both the categories, indicating that rating is a critical factor in deciding whether a customer will recommend the product or not

Rating vs Recommended



Insights:

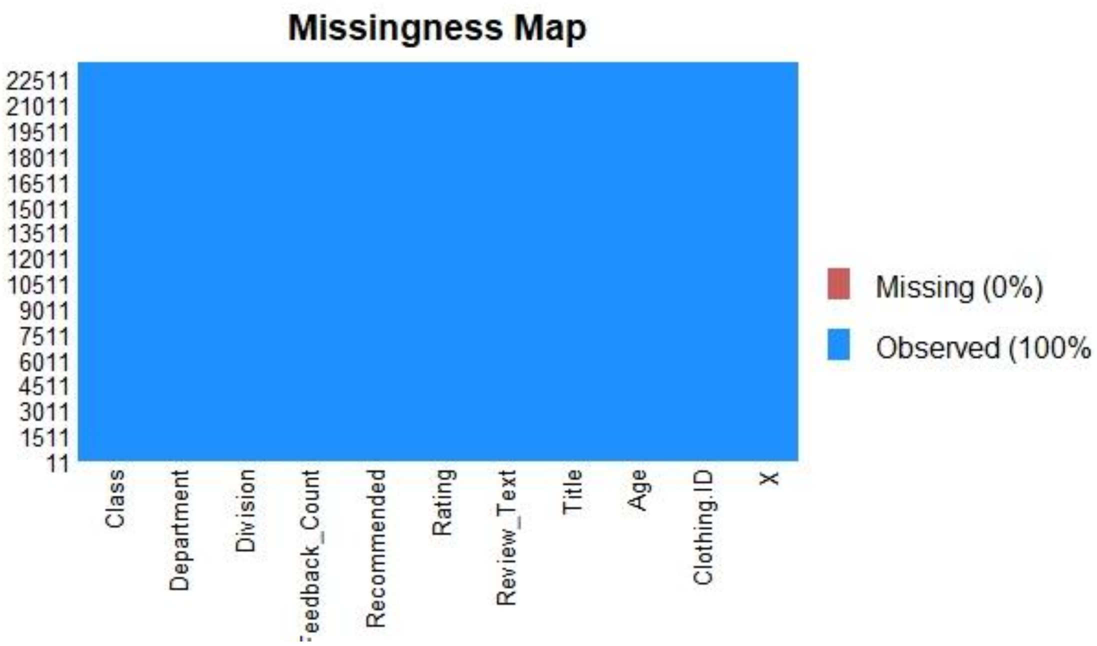
1) The above boxplot shows the distribution of Positive_Feedback_Count parameter among the customers who have the recommended the products and not recommended the products
2) Clearly, as the distribution is similar to among both the categories, indicating that Positive_Feedback_Count is not a significant factor in deciding whether a customer will recommend the product or not

Removal of unwanted variables:

The x column that comprised of serial numbers and clothing ID consisting of unique IDs of various products have been removed.

Note: The note for it is given in the appendix

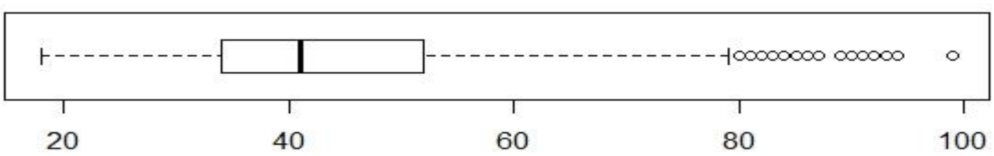
Handling missing values:



As we can in the above graph there are no missing values in the dataset.

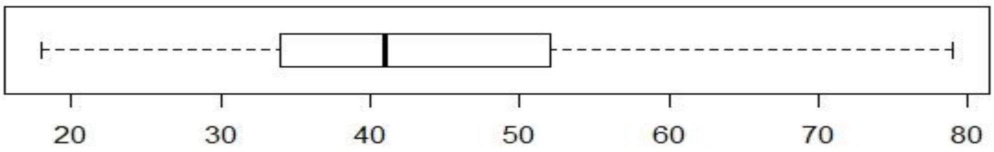
Removing outliers:

Age variable:

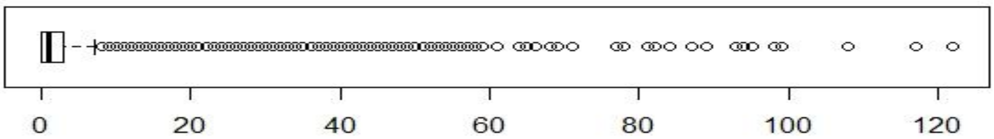


Insights:

1) The above boxplot shows the outliers present in the age variable
2) The graph shown below depicts the age variable transformed by removing its outliers



Positive_Feedback_Count variable:



Insights:

3) The above boxplot shows the outliers present in the Positive_Feedback_Count variable
4) The graph shown below depicts the Positive_Feedback_Count variable transformed by removing its outliers

Variable transformation:

1) The recommended variable was transformed from integer type to factor type
2) The review_text and title were transformed from factor to character types
3) We have also combined the Title and Review_Text variables into one as it simplify the sentimental analysis process

4. Exploratory Data Analysis Insights:

Note:

1) Checking whether the dataset is balanced or not
- Recommended : 19314
- Not Recommended : 4172

- Percent of Recommended

19314/(19314 + 4172)

= 0.8223 = 82.23%

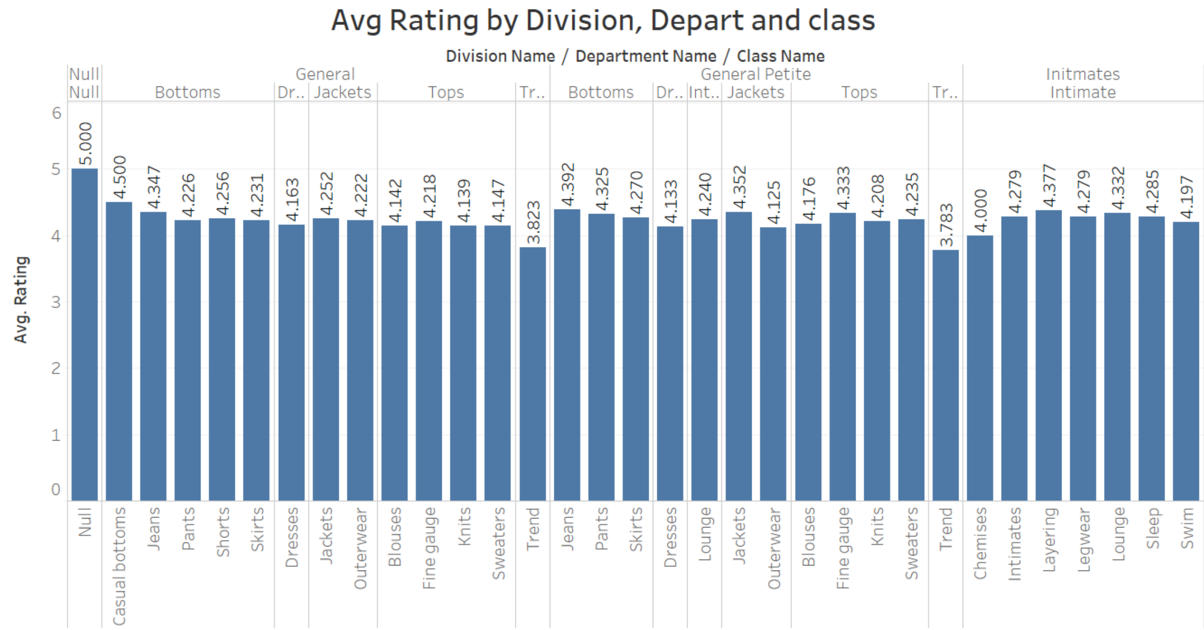
- Percent of Not-Recommended

4172/(19314 + 4172)

= 0.1776 = 17.76%

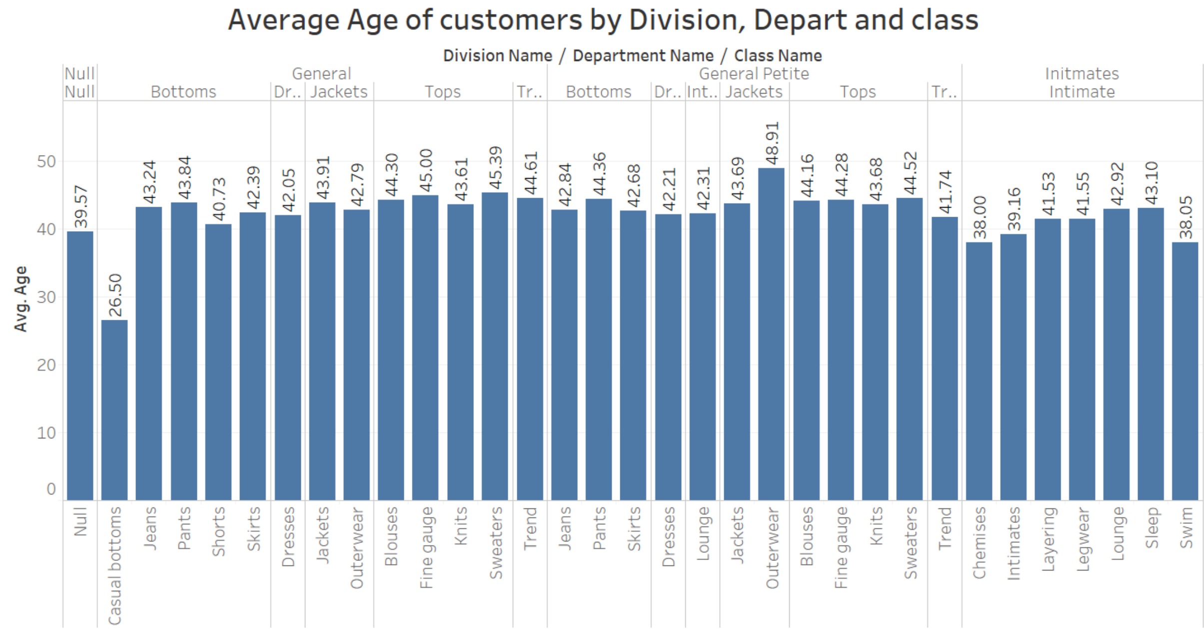
- The data is imbalanced
- We can use smote to balance the dataset depending on the accuracy level

2) Most of the insights have been mentioned below the charts. The below given insights are miscellaneous ones done with tableau.



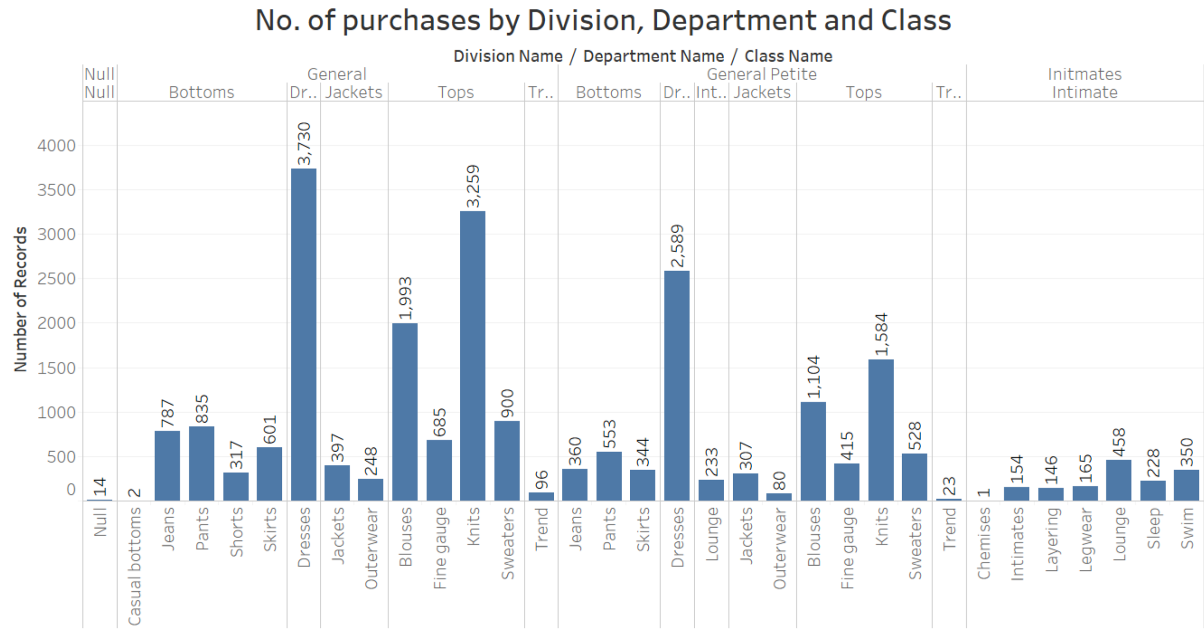
Average of Rating for each Class Name broken down by Division Name and Department Name.

Insights: As we see the ratings in all the division, department and classes are the similar. Therefore, we can conclude that no category is performing extremely bad or extremely good as compared to its counterparts.



Average of Age for each Class Name broken down by Division Name and Department Name.

Insights: As we see the average age in all the division, department and classes are the similar. Therefore, we can conclude that age is not a defining factor for the success of failure of any products across categories.



Sum of Number of Records for each Class Name broken down by Division Name and Department Name.

Insights:

1) As we see the number of purchases across categories different by very large margins. The dresses in the General division are the most popular among customers followed by Knites, Dresses in General Petite Division.
2) The retailer must focus on increasing its product line in these high performing categories

Model Building and Tuning

K-Fold Tuning for Logistic Regression

Output with all the variables:

```
                Estimate Std. Error z value Pr(>|z|)    
(Intercept)    -9.781598   0.299009 -32.713  < 2e-16 ***
Age             0.007329   0.003044   2.408  0.01606 *  
Rating          3.200878   0.061450  52.089  < 2e-16 ***
Feedback_Count -0.045292   0.013806  -3.281  0.00104 ** 
```

Output with only the significant variables:

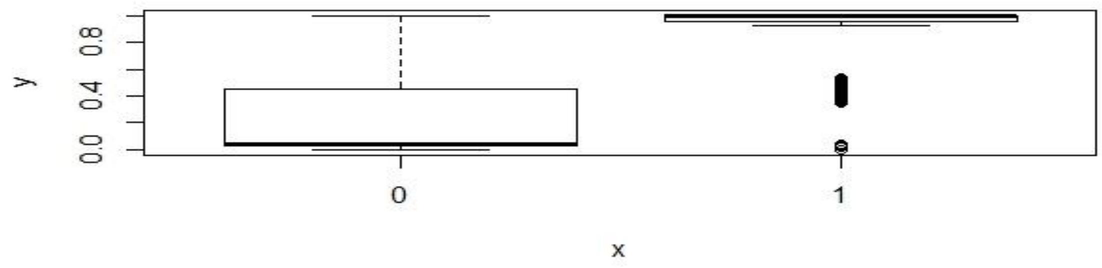
```
Coefficients:
                Estimate Std. Error z value Pr(>|z|)    
(Intercept)   -10.014665   0.240126 -41.706  < 2e-16 ***
Age             0.007284   0.003021   2.411 0.015899 *  
Rating          3.199102   0.061309  52.180  < 2e-16 ***
Feedback_Count -0.046621   0.013691  -3.405 0.000661 ***
```

Model Interpretation and Building:

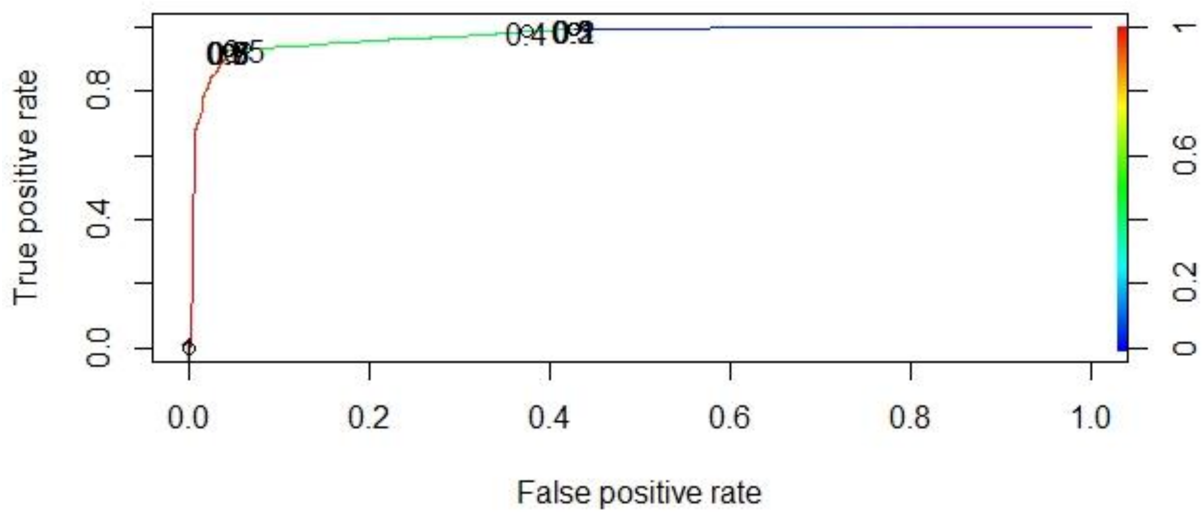
- **Interpretation:**
- If the Age increases by 1 unit the customer recommendation level will increase by 0.007284 units
- If the Rating increases by 1 unit the customer recommendation level will increase by 3.1991 units
- If the Feedback_Count increases by 1 unit the customer recommendation level will decrease by 0.0466 units
- Among the variables the age variable is most significant as it has the lowest p value

Model Tuning:

- The 5 fold cross validation was used to increase the accuracy of the model
- After conducting the 5 fold cross validation the model accuracy was found to be 93.609%
- The AIC score was 5383.9
- After conducting the cross validation, the Age, Rating and Feedback_Count was determined as the significant variables the AIC was reduced to 5358. The determines that the model tuning was successful as it was able the reduce the relative error in the model.
- The above boxplot shows there is a significant difference int the outputs depicting that the model is highly capable to differentiating whether the customer



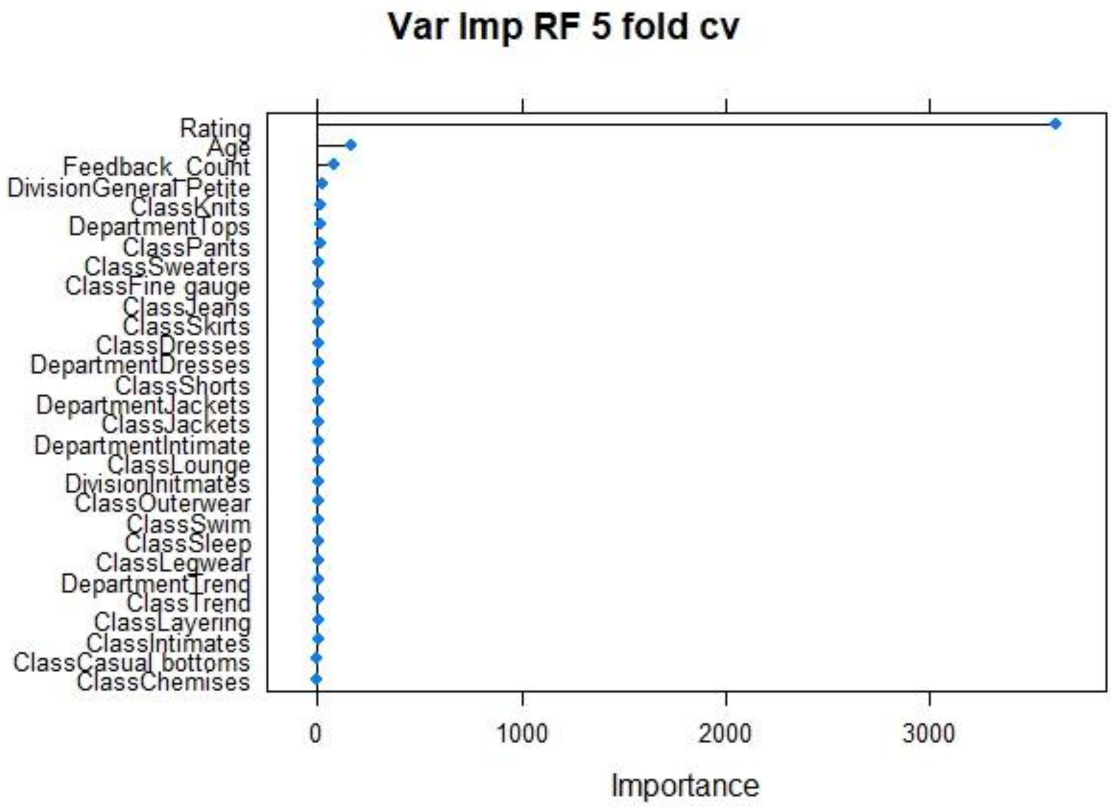
- The ROC curve was plot to determine the correct threshold needed to maximize the sensitive and specificity
- The earlier threshold was 0.5. The ROC curve shows that the accuracy increase by a small margin by using the threshold as 0.6. Therefore, the threshold of 0.6 was used.



- The confusion matrix has an accuracy of 93.16%
- The model has a sensitivity of 92.61%
- The model has a specificity of 95.68%
- The model has AIC score of 5358

Logistic regression		
	Not Recommended	Recommended
Not Recommended	798	36
Recommended	285	3575

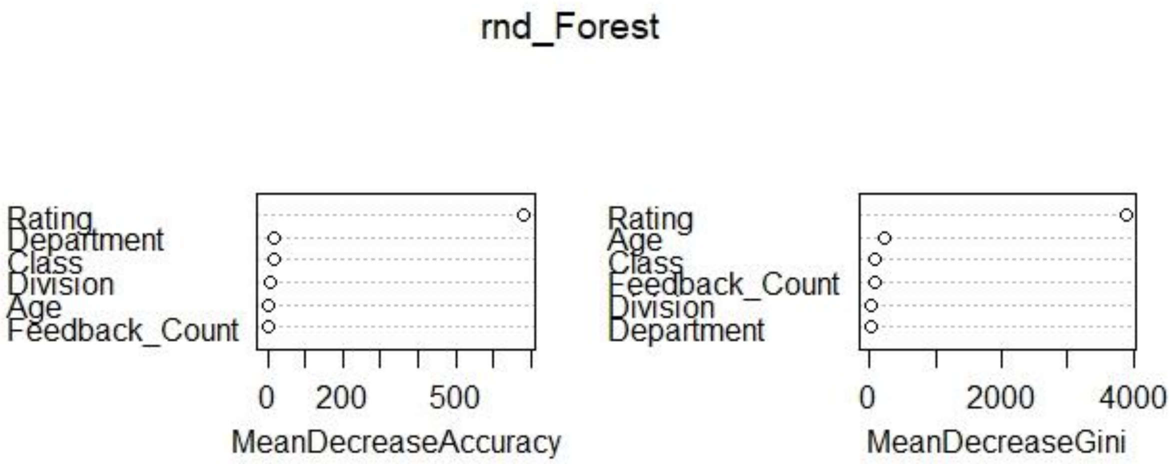
K-Fold Tuning for Random Forest



Model Building

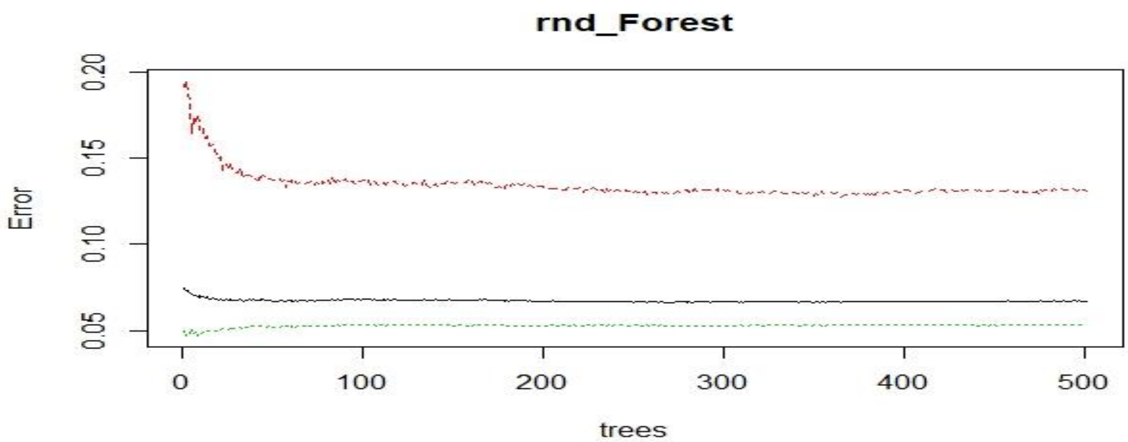
- As depicted in the above graph the model shows that the Rating variable is highly significant as compared to its other variables. The Rating variable is capable to determining whether the customer will recommend the services to others solely by itself.
- The model shows which variables are highly significant compared to its counterparts. As shown, The Rating parameter is highly significant as it reduces maximum Gini impurity and its removal will have sharp decline in accuracy.

Random Forest (Importance)		
	Decrease in Accuracy	**Mean Decrease in Gini**
Age	2.42	242.72
Rating	683.77	3895.22
Feedback_Count	-0.33	90.57
Division	7.92	33.24
Department	15.26	25.89
Class	14.91	92.44

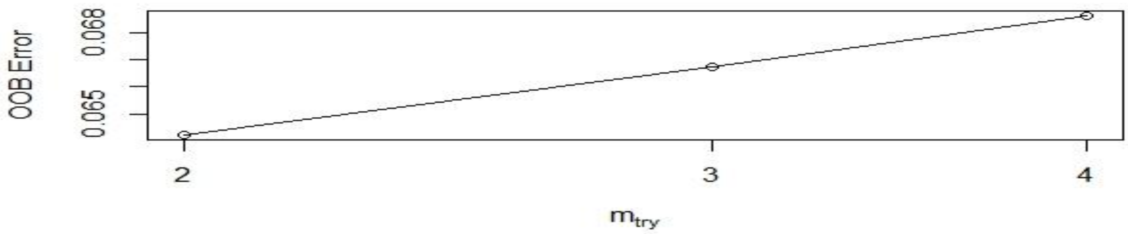


Model Tuning

- The model tuning began by using 5 fold cross validation for random forest to determine the best parameters
- After the cross-validation accuracy was 93.64%. The p value was < 2.2e-16 meaning the model was highly significant. Sensitivity was 91.25% and specificity was 94.15%.
- After that the random forest function was implemented and OBB (out of bag error rate) was found to be 6.69%. The class error for not recommended was 13% and recommended was 5.3%.
- The below graph was plotted to determine the optimal number of trees beyond which error rate does not decrease. Through the below graph the ntreetry was determined to be 51 as beyond it there is no significant decrease in error rate.



- The mtry was determined to be 2 as it reduce the OBB error rate to 6.42%. As shown in the below graph.

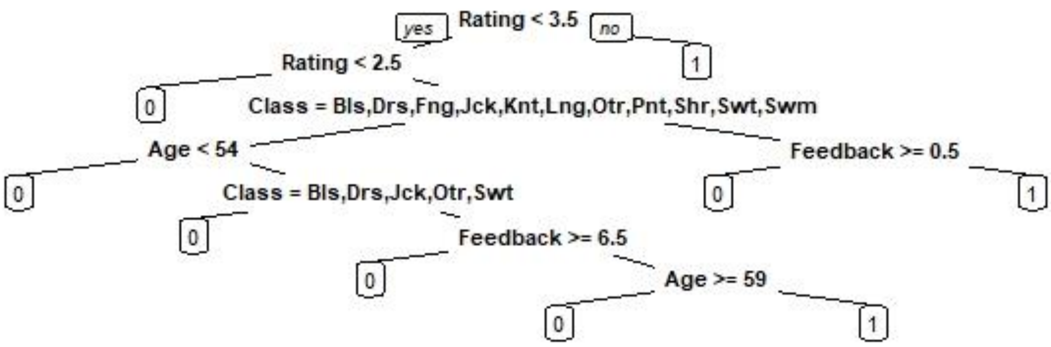


- The data was divided in 10 quantiles where the probabilities of them being recommended was determined. It was determined that the top 6 classes have high probability to being recommended.
- The confusion matrix was developed to assess its performance parameters
- The accuracy is 92.79, sensitivity was 94.04 and specificity was 87.05

Random Forest		
	Not Recommended	Recommended
Not Recommended	726	108
Recommended	230	3630

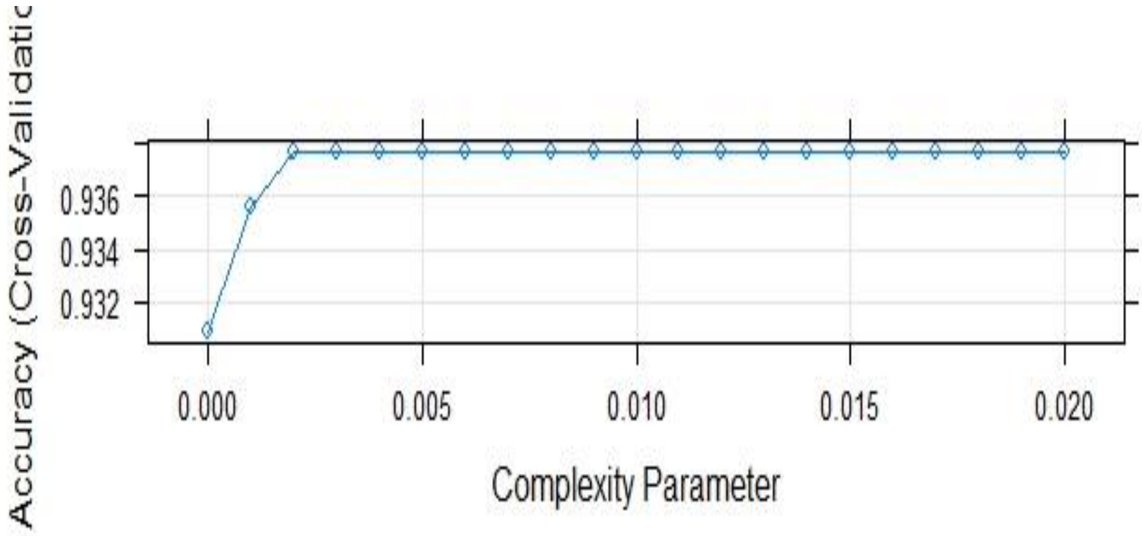
K-Fold Tuning for CART/Decision Tree

Model Building:

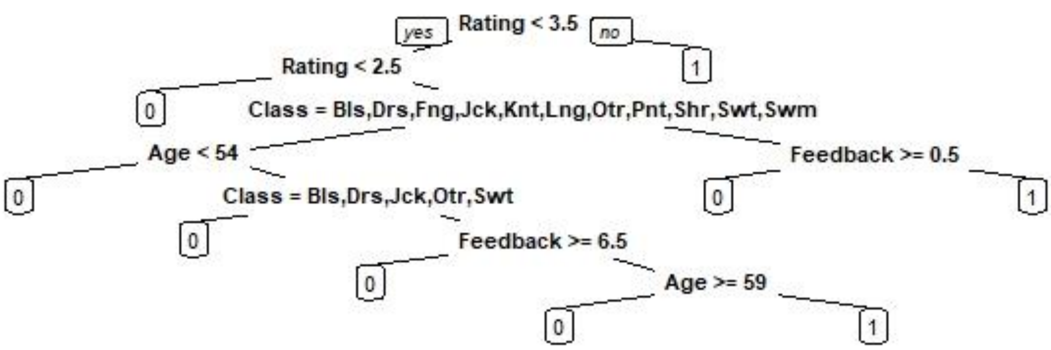


- The model was built using the CART model. According to the model Rating is the root node as it reduced the Gini impurity or entropy the most. The Class, Age and Feedback_Count are the other important variables followed by the Rating.

Model Tuning:



- The model tuning accomplished using 5 fold cross validation. A grid of various complexity parameters is developed and it is tested against the accuracy. The complexity parameter beyond which the model accuracy does not improve that the complexity parameter is used in the model building.
- The complexity parameter of 0.02 was used in the model building to avoid overfitting of data.



- Confusion Matrix was developed by testing the model on test dataset to understand the performance parameters
- The accuracy was 92.99%, sensitivity was 93.17% and specificity was 92.95%

CART / Decision Tree		
	Not Recommended	Recommended
Not Recommended	777	57
Recommended	272	3588

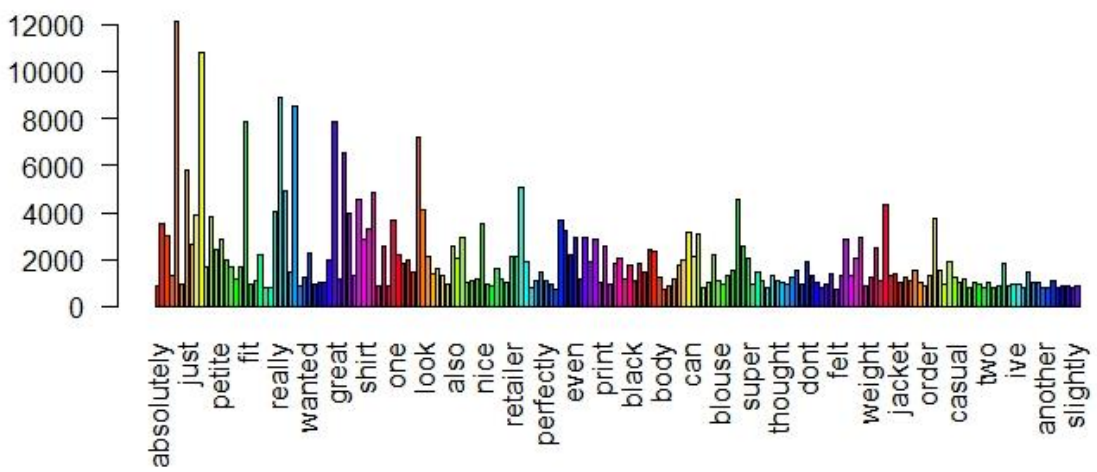
Model Selection

Model	Accuracy	Sensitivity	Specificity
Logistic Regression	93.16	92.61	95.68
Random Forest	92.79	94.94	87.05
CART	92.99	93.17	92.95

- The Logistic regression model performs the best in-terms of accuracy, sensitivity and specificity. The performs consistently across all parameters.
- The random forest on the other hand has the highest sensitivity however the model specificity suffers drastically compared to other models
- The CART model also is consistent performer as it performs well across all the parameters.
- Random forest being an ensemble method becomes a very convincing choice as the model takes care of overfitting by itself. However, as the Logistic regression performs well across all parameters it will the model recommended to the customers irrespective of any bias.

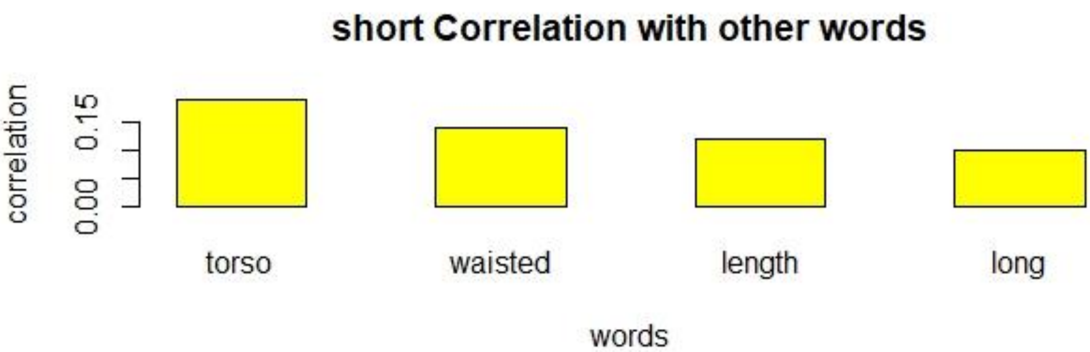
Text Analytics Model

Insights from text analytics



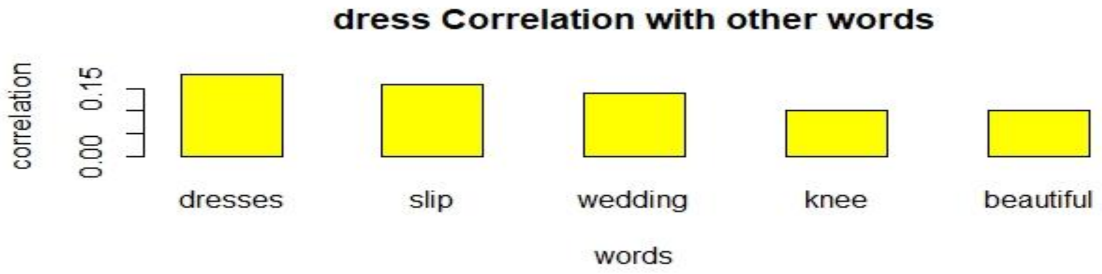
- The above mentioned are the terms or words that are used most frequently used that are present in at least 97% of the documents. This criteria was used to eliminate the most sparse terms from matrix.
- The sparsity was tested for 97%, 96%, 92% and 95% and finally 97% was chosen has it gave 188 words that were considered optimum for the procedure.
- Below is the word-cloud for the text analytics

- We used correlation analysis among the popular words to check whether to derive certain insights.
- The correlation for the short was with words as below:
    - Torso
    - Waisted
- This meant the products sizes need to be readjusted for the following parts as it is causing negative impact on the customers

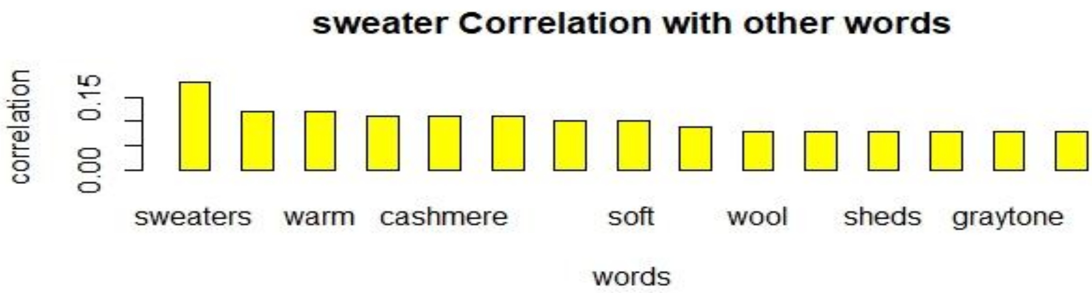


- The correlation for the dress was with words as below:
    - wedding
    - Beautiful

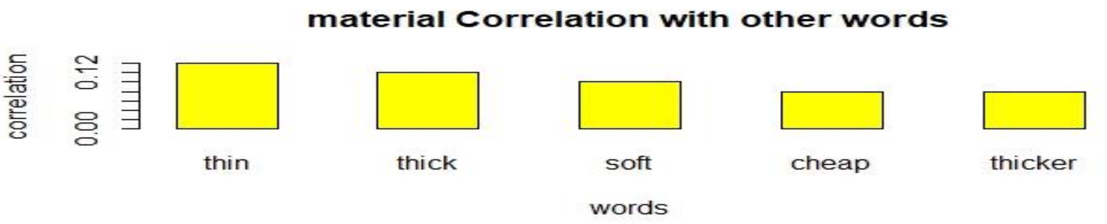
- This meant the dresses for the wedding occasion are very popular among the customers. The company should increase its product line horizontally and increase its production.



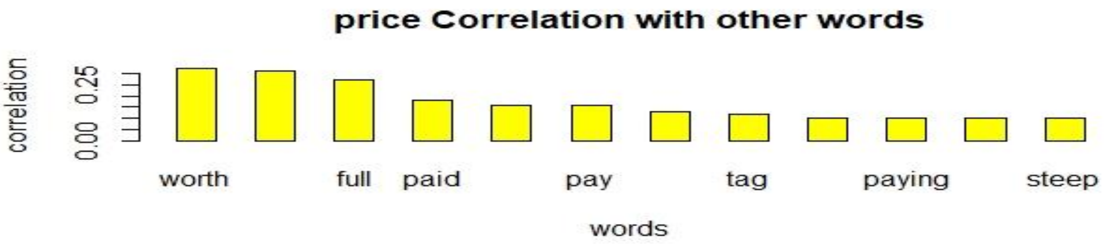
- The correlation for the sweater was with words as below:
    - sheds
    - itchy
- This meant the some of the sweaters in the product line are itchy and they shed wool after usage. This is a major deterrent and can affect customer perception. The issue should closely be monitored and should be resolved as soon as possible.



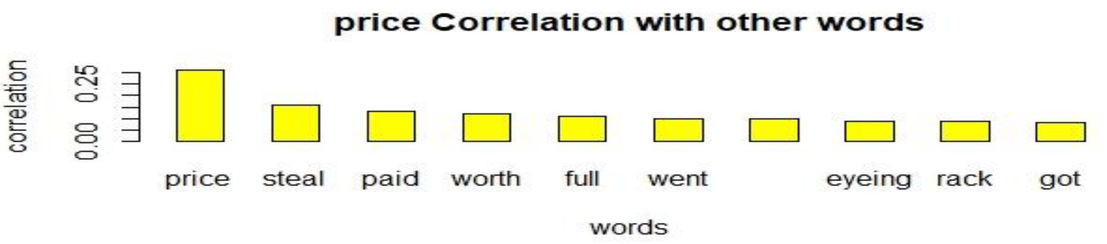
- The correlation for the material was with words as below:
    - thin
    - thick
    - cheap
    - thicker
- This meant the some of the products are overly thin or thick or made of cheap quality. The retailer should invest in revaluating their material procurement strategy to enhance their material to best accommodate customer needs.



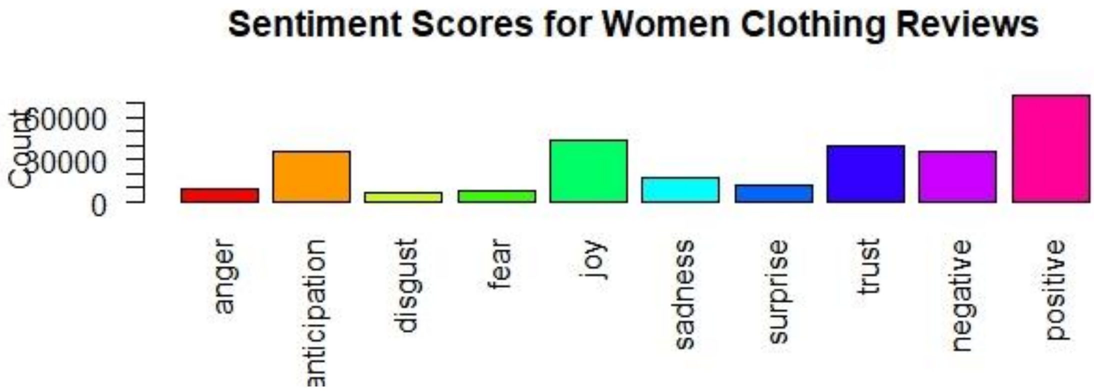
- The correlation for the price was with words as below:
  - worth
  - steal
  - sale
  - steep
- This meant the most of the products have been priced appropriately however some products are steep. The retailer need to revaluate their pricing strategy to satisfy these customers.



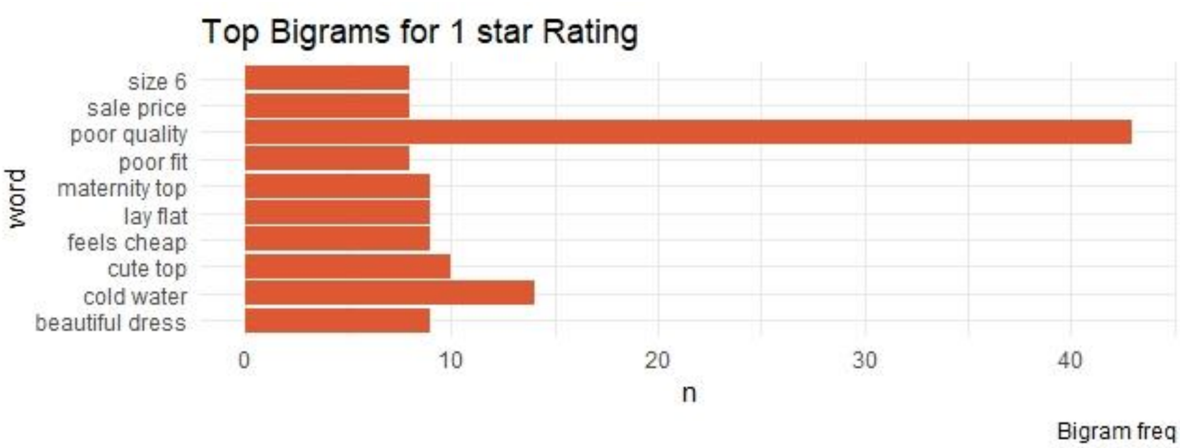
- The correlation for the sale was with words as below:
  - price
  - steal
  - worth
  - waited
  - eyeing
- This meant the sale events are much awaited by the customers. The company should use the sale events to increase the inventory turnover ratio. Moreover, it will help the company to quickly finish up with slow moving inventory thereby increasing the amount of money brought in by the company.



The overall sentimental analysis of the text analytics was



- The overall analysis is positive as we can see the percentage positive sentiment is more than the negative sentiment.
- The most commonly used sentiment are joyfulness followed by trust, and anticipation. The company should invest in enhancing the trust shown towards the retailer by their customers. The anticipation shows us that we should also increase the frequency of sale based events.



- As we can see that the most frequently found issue in the 1 star rated reviews is that they clothes made out of poor quality. The company should undertake a new raw material procurement strategy to improve the above seen bigram graph.

Recommendations

1. Through sentimental analysis it was discovered that customers prefer sales events. The company should hold various sales events throughout the year for every festival or occasion to boost sales. This will also help the company to quicken its cash flow cycle and increase profitability.
2. The Rating variable has determined as a significant variable in anticipating whether a customer will recommend the services to others. Therefore, they should use rating provided by the customers promote sales of the products on the ecommerce platform.
3. Some of the customers that rated the products with 1 star rating find the price is to high. Therefore, they should decrease the price of certain products to increase the sales through economies of scale.
4. The company should focus on improving the product quality of 1 star rated products as the most prominent issue for the 1 star rated products is that the material quality is cheap. Hence, undertake a new material procurement strategy to enhance its product quality across 1 star rated products.

Appendix

Code for the project

```
> rm(list=ls())
> setwd("C:/Users/abheer/Desktop/Data science/cp")
>
> df <- read.csv("Womens Clothing E-Commerce Reviews.csv", header=T)
>
> names(df)
 [1] "X0"                      "Clothing.ID"             "Age"
 [4] "Title"                   "Review.Text"             "Rating"
 [7] "Recommended.IND"         "Positive.Feedback.Count" "Division.Name"
[10] "Department.Name"         "Class.Name"
> attach(df)
The following objects are masked from df (pos = 23):

    Age, Class.Name, Clothing.ID, Department.Name, Division.Name,
    Positive.Feedback.Count, Rating, Recommended.IND, Review.Text, Title,
X0

> str(df)
'data.frame':	23472 obs. of  11 variables:
 $ X0                     : int  0 1 2 3 4 5 6 7 8 9 ...
 $ Clothing.ID            : int  767 1080 1077 1049 847 1080 858 858 1077
1077 ...
 $ Age                    : int  33 34 60 50 47 49 39 39 24 34 ...
 $ Title                  : Factor w/ 13985 levels "","\"beach business\""
,..: 1 1 11443 8051 4361 8763 1972 10664 4295 11757 ...
 $ Review.Text            : Factor w/ 22622 levels "","- this really is lo
vely. the overall design from the arms, front, and back makes this poncho
unique. it's not t"| __truncated__,..: 247 13172 5543 8021 20312 7983 3329
8845 7374 2670 ...
 $ Rating                 : int  4 5 3 5 5 2 5 4 5 5 ...
 $ Recommended.IND        : int  1 1 0 1 1 0 1 1 1 1 ...
 $ Positive.Feedback.Count: int  0 4 0 0 6 4 1 4 0 0 ...
 $ Division.Name          : Factor w/ 3 levels "General","General Petite",
..: 3 1 1 2 1 1 2 2 1 1 ...
 $ Department.Name        : Factor w/ 6 levels "Bottoms","Dresses",..: 3 2
2 1 5 2 5 5 2 2 ...
 $ Class.Name             : Factor w/ 20 levels "Blouses","Casual bottoms"
,..: 6 4 4 14 1 4 9 9 4 4 ...


>

> ################### Removing varaibles ###################
>
> df1 <- df[,-c(1,2,4,5)]
>
> df1$Recommended.IND <- as.factor(df1$Recommended.IND)
>
> #################### Importing Libraries ####################
>
> # Exploratory Data Analysis (EDA)
>
> library(tidyverse)
> library(ggplot2)
> library(caret)
> library(caretEnsemble)
> library(psych)
> library(Amelia)
> library(GGally)
> library(rpart)
```

```
> library(ggplot2)
>
> ###################### Missing Value #####################
>
> # MIssing value teartment
>
> # Treating missing values
> library(mice)
> library(VIM)
>
> # Displaying a graph to detect any missing data in the dataset
> missmap(df1)
>
> #################### Outlier treatment ####################
>
> boxplot(df1$Age, horizontal = T)
> bench1 = 52 + 1.5 * IQR(df1$Age)
> bench1
[1] 79
> df1$Age[df1$Age > bench1] <- bench1
> boxplot(df1$Age, horizontal = T)
>
> boxplot(df1$Positive.Feedback.Count, horizontal = T)
> bench2 = 3 + 1.5 * IQR(df1$Positive.Feedback.Count)
> bench2
[1] 7.5
> df1$Positive.Feedback.Count[df1$Positive.Feedback.Count > bench2] <- ben
ch2
> boxplot(df1$Positive.Feedback.Count, horizontal = T)
>
> #################### Multicollienrity #####################
>
> # Creating a seperate dataset to check multicollinearity
> library(faraway)
>
> df_MC <- df1
>
> # Changing variable types for the test
> df_MC$Recommended.IND <- as.integer(df_MC$Recommended.IND)
> df_MC$Division.Name <- as.integer(df_MC$Division.Name)
> df_MC$Department.Name <- as.integer(df_MC$Department.Name)
> df_MC$Class.Name <- as.integer(df_MC$Class.Name)
>
> mymodel = lm(Recommended.IND ~ ., data = df_MC)
>
> vif(mymodel)
                    Age                  Rating Positive.Feedback.Count
               1.010176                1.009158                1.013209
          Division.Name         Department.Name              Class.Name
               1.029923                1.013903                1.033806
>
> # No evidence of multicollinearity was found
> # The VIF value for all the variables was less than 4
>
> #################### Renaming variables #################
>
> # Change the name of the following varaibles
> library(reshape)
>
> df1 <- rename(df1, c( Recommended.IND = 'Recommended'))
> df1 <- rename(df1, c( Division.Name = 'Division'))
> df1 <- rename(df1, c( Department.Name = 'Department'))
> df1 <- rename(df1, c( Class.Name = 'Class'))
> df1 <- rename(df1, c( Positive.Feedback.Count = 'Feedback_Count'))
> df1 <- rename(df1, c( Review.Text = 'Review_Text'))
>
> str(df1)
'data.frame':   23472 obs. of  7 variables:
 $ Age            : num  33 34 60 50 47 49 39 39 24 34 ...
```

```
 $ Rating        : int  4 5 3 5 5 2 5 4 5 5 ...
 $ Recommended   : Factor w/ 2 levels "0","1": 2 2 1 2 2 1 2 2 2 2 ...
 $ Feedback_Count: num  0 4 0 0 6 4 1 4 0 0 ...
 $ Division      : Factor w/ 3 levels "General","General Petite",..: 3 1 1
2 1 1 2 2 1 1 ...
 $ Department    : Factor w/ 6 levels "Bottoms","Dresses",..: 3 2 2 1 5 2
5 5 2 2 ...
 $ Class         : Factor w/ 20 levels "Blouses","Casual bottoms",..: 6 4
4 14 1 4 9 9 4 4 ...
> attach(df1)

The following objects are masked from df (pos = 25):

    Age, Rating


> ################ Univariate Analysis ###################
>
> # Develop histogram of Age
> hist(Age, col = "Red")
>
> # Develop histogram of Rating
> hist(Rating, col = "Blue")
>
> # Develop histogram of Rating
> hist(FeedbackCount, col = "Green")
>
> ################# Bivariate Analysis ###################
>
> # Understanding the correlation between independent and
> # dependent variable (Recommended)
>
> plot(Recommended, Age)
> plot(Recommended, Rating)
> plot(Recommended, Feedback_Count)

>

> #################### Train Test Split ###################
>
> set.seed(42)
> ind <- createDataPartition(df1$Recommended, p = 8/10, list = FALSE)
> traindf <- df1[ind,]
> testdf <- df1[-ind,]
>
> str(traindf)
'data.frame':   18778 obs. of  7 variables:
 $ Age           : num  33 34 60 50 47 49 39 24 34 53 ...
 $ Rating        : int  4 5 3 5 5 2 4 5 5 3 ...
 $ Recommended   : Factor w/ 2 levels "0","1": 2 2 1 2 2 1 2 2 2 1 ...
 $ Feedback_Count: num  0 4 0 0 6 4 4 0 0 7.5 ...
 $ Division      : Factor w/ 3 levels "General","General Petite",..: 3 1 1
2 1 1 2 1 1 1 ...
 $ Department    : Factor w/ 6 levels "Bottoms","Dresses",..: 3 2 2 1 5 2
5 2 2 2 ...
 $ Class         : Factor w/ 20 levels "Blouses","Casual bottoms",..: 6 4
4 14 1 4 9 4 4 4 ...
> summary(traindf)
      Age           Rating        Recommended Feedback_Count
 Min.   :18.00   Min.   :1.000   0: 3338     Min.   :0.00  
 1st Qu.:34.00   1st Qu.:4.000   1:15440     1st Qu.:0.00  
 Median :41.00   Median :5.000               Median :1.00  
 Mean   :43.19   Mean   :4.199               Mean   :1.77  
 3rd Qu.:52.00   3rd Qu.:5.000               3rd Qu.:3.00  
 Max.   :79.00   Max.   :5.000               Max.   :7.50  

           Division         Department         Class     
 General       :11120   Bottoms :3039   Dresses :5052  
 General Petite: 6471   Dresses :5052   Knits   :3891  
```

```
 Initmates      : 1187    Intimate:1372    Blouses :2482
                          Jackets : 831    Sweaters:1128
                          Tops    :8386    Pants   :1099
                          Trend   :  98    Jeans   : 929
                                           (Other) :4197
```

```
> ####################################################################
> ############ K-Fold Tuning for Logistic Regression ###############
> ####################################################################
>
> logit_control <- trainControl(method = "cv",
+                               number = 5,
+                               search = "random",
+                               savePredictions = T)
>
>
> logit_fitcv <- train(Recommended~.,
+                      data = na.exclude(traindf),
+                      method = "glm",
+                      family = "binomial",
+                      trControl = logit_control)
>
>
> summary(logit_fitcv)

Call:
NULL

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6132   0.0562   0.0640   0.2637   3.6864

Coefficients: (5 not defined because of singularities)
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)               -9.781598   0.299009 -32.713  < 2e-16 ***
Age                        0.007329   0.003044   2.408  0.01606 *
Rating                     3.200878   0.061450  52.089  < 2e-16 ***
Feedback_Count            -0.045292   0.013806  -3.281  0.00104 **
`DivisionGeneral Petite`   0.023403   0.077317   0.303  0.76212
DivisionInitmates          0.110254   0.456069   0.242  0.80897
DepartmentDresses         -0.295948   0.193077  -1.533  0.12533
DepartmentIntimate        -0.511642   0.562858  -0.909  0.36335
DepartmentJackets         -0.363806   0.368219  -0.988  0.32315
DepartmentTops            -0.268953   0.204786  -1.313  0.18907
DepartmentTrend            0.082691   0.498334   0.166  0.86821
`ClassCasual bottoms`      9.353588 324.743756   0.029  0.97702
ClassChemises              9.667024 324.743823   0.030  0.97625
`ClassFine gauge`         -0.006205   0.195785  -0.032  0.97472
ClassIntimates             0.898413   0.605748   1.483  0.13804
ClassJackets               0.274641   0.396369   0.693  0.48838
ClassJeans                 0.197008   0.255896   0.770  0.44137
ClassKnits                 0.002594   0.121437   0.021  0.98296
ClassLayering              0.620111   0.624480   0.993  0.32071
ClassLegwear               0.294312   0.566429   0.520  0.60335
ClassLounge                0.256336   0.387015   0.662  0.50775
ClassPants                -0.343371   0.233899  -1.468  0.14210
ClassShorts               -0.221059   0.346787  -0.637  0.52383
ClassSleep                 0.331756   0.501847   0.661  0.50857
ClassSweaters             -0.220432   0.170698  -1.291  0.19658
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 17575.5  on 18777  degrees of freedom
Residual deviance:  5333.9  on 18753  degrees of freedom
AIC: 5383.9
```

```
Number of Fisher Scoring iterations: 11

>
> logit_fitcv
Generalized Linear Model

18778 samples
    6 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 15022, 15023, 15022, 15022, 15023
Resampling results:

  Accuracy   Kappa
  0.9360953  0.7948059

> caret::confusionMatrix(table((logit_fitcv$pred)$pred,
+                              (logit_fitcv$pred)$obs))
Confusion Matrix and Statistics


        0     1
  0  3016   878
  1   322 14562

               Accuracy : 0.9361
                 95% CI : (0.9325, 0.9396)
    No Information Rate : 0.8222
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7948

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

            Sensitivity : 0.9035
            Specificity : 0.9431
         Pos Pred Value : 0.7745
         Neg Pred Value : 0.9784
             Prevalence : 0.1778
         Detection Rate : 0.1606
   Detection Prevalence : 0.2074
      Balanced Accuracy : 0.9233

       'Positive' Class : 0


>
> # Developing a new model with only the varaibles > 95%
> # significance levels
>
> logit_fitcv_sig <- train(Recommended ~ Age + Rating +
+                              Feedback_Count,
+                          data = na.exclude(traindf),
+                          method = "glm",
+                          family = "binomial",
+                          trControl = logit_control)
>
> summary(logit_fitcv_sig)

Call:
NULL

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5973   0.0585   0.0632   0.2821   3.7136
```

```
Coefficients:
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)    -10.014665   0.240126 -41.706  < 2e-16 ***
Age              0.007284   0.003021   2.411 0.015899 *  
Rating           3.199102   0.061309  52.180  < 2e-16 ***
Feedback_Count  -0.046621   0.013691  -3.405 0.000661 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 17575.5  on 18777  degrees of freedom
Residual deviance:  5350.8  on 18774  degrees of freedom
AIC: 5358.8

Number of Fisher Scoring iterations: 7

> logit_fitcv_sig
Generalized Linear Model 

18778 samples
    3 predictor
    2 classes: '0', '1' 

No pre-processing
Resampling: Cross-Validated (5 fold) 
Summary of sample sizes: 15022, 15023, 15023, 15022, 15022 
Resampling results:

  Accuracy   Kappa    
  0.9374269  0.8025428

> caret::confusionMatrix(table((logit_fitcv_sig$pred)$pred,
+                              (logit_fitcv_sig$pred)$obs))
Confusion Matrix and Statistics

   
        0     1
  0  3109   946
  1   229 14494
                                          
               Accuracy : 0.9374          
                 95% CI : (0.9339, 0.9408)
    No Information Rate : 0.8222          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.8026          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.9314          
            Specificity : 0.9387          
         Pos Pred Value : 0.7667          
         Neg Pred Value : 0.9844          
             Prevalence : 0.1778          
         Detection Rate : 0.1656          
   Detection Prevalence : 0.2159          
      Balanced Accuracy : 0.9351          
                                          
       'Positive' Class : 0               

> Logistic_model <- glm(Recommended ~ Age + Rating + 
+                           data = traindf,
+                           family=binomial)
> summary(Logistic_model)

Call:
glm(formula = Recommended ~ Age + Rating + Feedback_Count, family = binomi
al,
```

```
    data = traindf)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5973   0.0585   0.0632   0.2821   3.7136

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -10.014665   0.240126 -41.706  < 2e-16 ***
Age              0.007284   0.003021   2.411 0.015899 *
Rating           3.199102   0.061309  52.180  < 2e-16 ***
Feedback_Count  -0.046621   0.013691  -3.405 0.000661 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 17575.5  on 18777  degrees of freedom
Residual deviance:  5350.8  on 18774  degrees of freedom
AIC: 5358.8

Number of Fisher Scoring iterations: 7

> plot(as.factor(Logistic_model$y), Logistic_model$fitted.values)

> # As shown in the model the boxplot has a very high distinctive and
> # predictive power as the boxplots differ in a larger manner
>
> res <- predict(Logistic_model, testdf, type = “response”))

> #################### Confusion Matrix #######################
>
> table(ActualValue = testdf$Recommended,
+                     PredictedValue = res > 0.5)
           PredictedValue
ActualValue FALSE TRUE
          0   784   50
          1   262 3598
> ################### Optimizing Threshold #####################
>
> library(ROCR)
Loading required package: gplots

Attaching package: ‘gplots’

The following object is masked from ‘package:stats’:

    lowess

>
> ROCR_Pred <- prediction(res, testdf$Recommended)
> ROCR_Pref <- performance(ROCR_Pred, "tpr", "fpr")
>
> plot(ROCR_Pref, colorize = T, print.cutoffs.at = seq(0.1, by = 0.1))

>
> ############ Re-configuring the Threshold ######################
>
> table(ActualValue = testdf$Recommended,
+       PredictedValue = res > 0.6)
           PredictedValue
ActualValue FALSE TRUE
          0   798   36
          1   285 3575
>


> ####################################################################
```

```
> ############### K-Fold Tuning for Random Forest #################
> ################################################################
>
>
> # Hyperparametr tuning Random Forest
>
> fitcontrol <- trainControl(method = "cv",
+                            number = 5,
+                            search = "random",
+                            savePredictions = T)
>
> fitcontrol_repeated <- trainControl(method = "repeatedcv",
+                                     number = 5,
+                                     search = "random",
+                                     repeats = 3,
+                                     savePredictions = T)
>
> # We can use fitcontrol or fitcontrol_repeated for repeated cv
>
> rf_fit_cv <- train(Recommended~.,
+                    data = na.exclude(traindf),
+                    method = "rf",
+                    trControl = fitcontrol,
+                    tuneLength = 10,
+                    ntree = 100)
>
> rf_fit_cv$bestTune
  mtry
2    5
>
> rf_fit_cv
Random Forest

18778 samples
    8 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 15023, 15023, 15022, 15022, 15022
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.9537758  0.8468878
   5    0.9543083  0.8486497
  12    0.9489829  0.8287276
  19    0.9455214  0.8156889
  24    0.9456813  0.8154559
  26    0.9452552  0.8143738
  27    0.9455216  0.8148210
  29    0.9455747  0.8152391
  30    0.9448824  0.8130936

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 5.
>
> # Ploting variance importance plot
> # It gives the most import varaible
> plot(varImp(rf_fit_cv, scale = F), main = "Var Imp RF 5 fold cv")
Error in .Call.graphics(C_palette2, .Call(C_palette2, NULL)) :
  invalid graphics state
>
> ####### Developing a confusion matrix with best parameters ######
>
> Optimal_rf_ = subset(rf_fit_cv$pred, rf_fit_cv$pred$mtry ==
+                      rf_fit_cv$bestTune$mtry)
>
> caret::confusionMatrix(table(Optimal_rf_$pred,Optimal_rf_$obs))
Confusion Matrix and Statistics
```

```
         0     1
  0   3049   569
  1    289 14871

               Accuracy : 0.9543
                 95% CI : (0.9512, 0.9573)
    No Information Rate : 0.8222
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8487

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

            Sensitivity : 0.9134
            Specificity : 0.9631
         Pos Pred Value : 0.8427
         Neg Pred Value : 0.9809
             Prevalence : 0.1778
         Detection Rate : 0.1624
   Detection Prevalence : 0.1927
      Balanced Accuracy : 0.9383

       'Positive' Class : 0

>
> #################################################################
>
> ###### Initial EDA ########
>
> nrow(traindf)
[1] 18778
> sum(traindf$Recommended == "1")/nrow(traindf)
[1] 0.8222388
>
> ######  #######
>
> library(randomForest)
> set.seed(100)
>
> rnd_Forest <- randomForest(Recommended ~.,
+                            data = traindf,
+                            ntree = 501,
+                            mtry = 3,
+                            nodesize = 10,
+                            importance = TRUE)
>
> print(rnd_Forest)

Call:
 randomForest(formula = Recommended ~ ., data = traindf, ntree = 501,
mtry = 3, nodesize = 10, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 501
No. of variables tried at each split: 3

        OOB estimate of  error rate: 4.65%
Confusion matrix:
     0     1 class.error
0 2980   358  0.10724985
1  515 14925  0.03335492
>
> varImpPlot(rnd_Forest)
Error in plot.new() : figure margins too large
>
>
> # prediction on through confusion matrix
```

```
> prediction_Random <- predict(rnd_Forest, testdf[,-c(3,8,9)])
Error in eval(predvars, data, env) : object 'predict_class' not found
> table(observed = testdf[,3], predicted = prediction_Random)
        predicted
observed    0    1
       0  726  108
       1  230 3630
>
>
> # Printing the error rate drecrease along vs no. of trees
> print(rnd_Forest$err.rate)
            OOB         0          1
  [1,] 0.05557962 0.1546645 0.03435583
  [2,] 0.05498586 0.1523279 0.03438303
  [3,] 0.05425638 0.1432049 0.03529106
  [4,] 0.05224448 0.1346292 0.03463137
  [5,] 0.05127292 0.1313771 0.03393012
  [6,] 0.05217887 0.1356527 0.03404521
  [7,] 0.04977225 0.1285893 0.03270712
  [8,] 0.05129186 0.1309158 0.03407959
  [9,] 0.05088687 0.1276984 0.03426955
 [10,] 0.05134685 0.1273828 0.03491565
 [11,] 0.05146073 0.1289251 0.03473897
 [12,] 0.05015506 0.1255268 0.03387516
 [13,] 0.05039774 0.1245113 0.03440218
 [14,] 0.05002133 0.1254878 0.03372025
 [15,] 0.04978943 0.1233123 0.03390380
 [16,] 0.04977617 0.1233123 0.03389281
 [17,] 0.04944060 0.1229017 0.03356227
 [18,] 0.04937943 0.1198681 0.03414097
 [19,] 0.04953659 0.1198681 0.03433310
 [20,] 0.04947542 0.1204314 0.03413434
 [21,] 0.04947542 0.1195327 0.03432865
 [22,] 0.04931303 0.1180348 0.03445596
 [23,] 0.04856747 0.1159377 0.03400259
 [24,] 0.04835446 0.1168364 0.03354922
 [25,] 0.04840771 0.1165368 0.03367876
 [26,] 0.04814144 0.1138406 0.03393782
 [27,] 0.04808819 0.1147394 0.03367876
 [28,] 0.04755565 0.1129419 0.03341969
 [29,] 0.04744914 0.1120431 0.03348446
 [30,] 0.04723613 0.1126423 0.03309585
 [31,] 0.04734263 0.1126423 0.03322539
 [32,] 0.04712962 0.1117436 0.03316062
 [33,] 0.04681010 0.1105452 0.03303109
 [34,] 0.04739589 0.1111444 0.03361399
 [35,] 0.04744914 0.1117436 0.03354922
 [36,] 0.04734263 0.1114440 0.03348446
 [37,] 0.04728938 0.1123427 0.03322539
 [38,] 0.04776867 0.1138406 0.03348446
 [39,] 0.04734263 0.1135410 0.03303109
 [40,] 0.04787517 0.1138406 0.03361399
 [41,] 0.04750240 0.1141402 0.03309585
 [42,] 0.04744914 0.1141402 0.03303109
 [43,] 0.04755565 0.1129419 0.03341969
 [44,] 0.04760890 0.1129419 0.03348446
 [45,] 0.04734263 0.1132415 0.03309585
 [46,] 0.04744914 0.1132415 0.03322539
 [47,] 0.04755565 0.1123427 0.03354922
 [48,] 0.04712962 0.1117436 0.03316062
 [49,] 0.04734263 0.1117436 0.03341969
 [50,] 0.04702311 0.1114440 0.03309585
 [51,] 0.04702311 0.1117436 0.03303109
 [52,] 0.04744914 0.1126423 0.03335492
 [53,] 0.04707637 0.1123427 0.03296632
 [54,] 0.04734263 0.1132415 0.03309585
 [55,] 0.04707637 0.1123427 0.03296632
 [56,] 0.04691660 0.1120431 0.03283679
 [57,] 0.04686335 0.1111444 0.03296632
```

```
 [58,] 0.04723613 0.1117436 0.03329016
 [59,] 0.04734263 0.1117436 0.03341969
 [60,] 0.04744914 0.1120431 0.03348446
 [61,] 0.04696986 0.1099461 0.03335492
 [62,] 0.04681010 0.1081486 0.03354922
 [63,] 0.04707637 0.1108448 0.03329016
 [64,] 0.04702311 0.1093469 0.03354922
 [65,] 0.04728938 0.1117436 0.03335492
 [66,] 0.04744914 0.1120431 0.03348446
 [67,] 0.04744914 0.1120431 0.03348446
 [68,] 0.04739589 0.1108448 0.03367876
 [69,] 0.04734263 0.1114440 0.03348446
 [70,] 0.04707637 0.1096465 0.03354922
 [71,] 0.04728938 0.1105452 0.03361399
 [72,] 0.04707637 0.1096465 0.03354922
 [73,] 0.04744914 0.1114440 0.03361399
 [74,] 0.04712962 0.1102457 0.03348446
 [75,] 0.04734263 0.1114440 0.03348446
 [76,] 0.04718287 0.1111444 0.03335492
 [77,] 0.04718287 0.1111444 0.03335492
 [78,] 0.04696986 0.1102457 0.03329016
 [79,] 0.04707637 0.1111444 0.03322539
 [80,] 0.04712962 0.1108448 0.03335492
 [81,] 0.04712962 0.1105452 0.03341969
 [82,] 0.04702311 0.1099461 0.03341969
 [83,] 0.04686335 0.1096465 0.03329016
 [84,] 0.04696986 0.1099461 0.03335492
 [85,] 0.04681010 0.1096465 0.03322539
 [86,] 0.04686335 0.1093469 0.03335492
 [87,] 0.04707637 0.1108448 0.03329016
 [88,] 0.04675684 0.1099461 0.03309585
 [89,] 0.04702311 0.1096465 0.03348446
 [90,] 0.04649057 0.1090473 0.03296632
 [91,] 0.04702311 0.1099461 0.03341969
 [92,] 0.04659708 0.1099461 0.03290155
 [93,] 0.04691660 0.1093469 0.03341969
 [94,] 0.04686335 0.1102457 0.03316062
 [95,] 0.04665034 0.1087478 0.03322539
 [96,] 0.04659708 0.1090473 0.03309585
 [97,] 0.04665034 0.1084482 0.03329016
 [98,] 0.04665034 0.1081486 0.03335492
 [99,] 0.04617105 0.1069503 0.03303109
[100,] 0.04633081 0.1078490 0.03303109
[101,] 0.04627756 0.1075494 0.03303109
[102,] 0.04611780 0.1069503 0.03296632
[103,] 0.04622431 0.1069503 0.03309585
[104,] 0.04622431 0.1069503 0.03309585
[105,] 0.04633081 0.1072499 0.03316062
[106,] 0.04638407 0.1072499 0.03322539
[107,] 0.04606454 0.1057519 0.03316062
[108,] 0.04638407 0.1078490 0.03309585
[109,] 0.04622431 0.1063511 0.03322539
[110,] 0.04622431 0.1069503 0.03309585
[111,] 0.04643732 0.1081486 0.03309585
[112,] 0.04665034 0.1090473 0.03316062
[113,] 0.04622431 0.1069503 0.03309585
[114,] 0.04633081 0.1072499 0.03316062
[115,] 0.04617105 0.1072499 0.03296632
[116,] 0.04627756 0.1078490 0.03296632
[117,] 0.04638407 0.1081486 0.03303109
[118,] 0.04643732 0.1078490 0.03316062
[119,] 0.04627756 0.1075494 0.03303109
[120,] 0.04617105 0.1072499 0.03296632
[121,] 0.04627756 0.1078490 0.03296632
[122,] 0.04611780 0.1078490 0.03277202
[123,] 0.04622431 0.1081486 0.03283679
[124,] 0.04633081 0.1084482 0.03290155
[125,] 0.04654383 0.1090473 0.03303109
[126,] 0.04633081 0.1087478 0.03283679
```

```
[127,] 0.04633081 0.1081486 0.03296632
[128,] 0.04622431 0.1078490 0.03290155
[129,] 0.04633081 0.1087478 0.03283679
[130,] 0.04633081 0.1081486 0.03296632
[131,] 0.04638407 0.1078490 0.03309585
[132,] 0.04611780 0.1084482 0.03264249
[133,] 0.04617105 0.1078490 0.03283679
[134,] 0.04617105 0.1078490 0.03283679
[135,] 0.04622431 0.1078490 0.03290155
[136,] 0.04633081 0.1078490 0.03303109
[137,] 0.04654383 0.1087478 0.03309585
[138,] 0.04627756 0.1078490 0.03296632
[139,] 0.04665034 0.1081486 0.03335492
[140,] 0.04643732 0.1081486 0.03309585
[141,] 0.04649057 0.1087478 0.03303109
[142,] 0.04654383 0.1087478 0.03309585
[143,] 0.04643732 0.1078490 0.03316062
[144,] 0.04638407 0.1081486 0.03303109
[145,] 0.04643732 0.1081486 0.03309585
[146,] 0.04643732 0.1084482 0.03303109
[147,] 0.04654383 0.1084482 0.03316062
[148,] 0.04649057 0.1081486 0.03316062
[149,] 0.04627756 0.1075494 0.03303109
[150,] 0.04643732 0.1081486 0.03309585
[151,] 0.04654383 0.1084482 0.03316062
[152,] 0.04659708 0.1090473 0.03309585
[153,] 0.04659708 0.1081486 0.03329016
[154,] 0.04659708 0.1084482 0.03322539
[155,] 0.04654383 0.1078490 0.03329016
[156,] 0.04643732 0.1078490 0.03316062
[157,] 0.04654383 0.1084482 0.03316062
[158,] 0.04681010 0.1093469 0.03329016
[159,] 0.04686335 0.1096465 0.03329016
[160,] 0.04675684 0.1090473 0.03329016
[161,] 0.04691660 0.1090473 0.03348446
[162,] 0.04691660 0.1096465 0.03335492
[163,] 0.04691660 0.1096465 0.03335492
[164,] 0.04675684 0.1087478 0.03335492
[165,] 0.04670359 0.1093469 0.03316062
[166,] 0.04675684 0.1087478 0.03335492
[167,] 0.04681010 0.1084482 0.03348446
[168,] 0.04675684 0.1090473 0.03329016
[169,] 0.04691660 0.1096465 0.03335492
[170,] 0.04670359 0.1084482 0.03335492
[171,] 0.04670359 0.1087478 0.03329016
[172,] 0.04643732 0.1078490 0.03316062
[173,] 0.04654383 0.1090473 0.03303109
[174,] 0.04659708 0.1084482 0.03322539
[175,] 0.04659708 0.1081486 0.03329016
[176,] 0.04654383 0.1081486 0.03322539
[177,] 0.04659708 0.1078490 0.03335492
[178,] 0.04638407 0.1072499 0.03322539
[179,] 0.04649057 0.1069503 0.03341969
[180,] 0.04649057 0.1072499 0.03335492
[181,] 0.04670359 0.1081486 0.03341969
[182,] 0.04659708 0.1072499 0.03348446
[183,] 0.04686335 0.1081486 0.03361399
[184,] 0.04691660 0.1081486 0.03367876
[185,] 0.04665034 0.1078490 0.03341969
[186,] 0.04654383 0.1069503 0.03348446
[187,] 0.04659708 0.1069503 0.03354922
[188,] 0.04681010 0.1081486 0.03354922
[189,] 0.04681010 0.1084482 0.03348446
[190,] 0.04665034 0.1075494 0.03348446
[191,] 0.04686335 0.1087478 0.03348446
[192,] 0.04707637 0.1099461 0.03348446
[193,] 0.04681010 0.1087478 0.03341969
[194,] 0.04686335 0.1087478 0.03348446
[195,] 0.04696986 0.1090473 0.03354922
```

```
[196,] 0.04707637 0.1087478 0.03374352
[197,] 0.04707637 0.1090473 0.03367876
[198,] 0.04681010 0.1090473 0.03335492
[199,] 0.04691660 0.1084482 0.03361399
[200,] 0.04665034 0.1078490 0.03341969
[201,] 0.04675684 0.1078490 0.03354922
[202,] 0.04681010 0.1081486 0.03354922
[203,] 0.04665034 0.1081486 0.03335492
[204,] 0.04665034 0.1078490 0.03341969
[205,] 0.04659708 0.1084482 0.03322539
[206,] 0.04670359 0.1084482 0.03335492
[207,] 0.04649057 0.1069503 0.03341969
[208,] 0.04659708 0.1072499 0.03348446
[209,] 0.04659708 0.1072499 0.03348446
[210,] 0.04659708 0.1066507 0.03361399
[211,] 0.04649057 0.1066507 0.03348446
[212,] 0.04665034 0.1069503 0.03361399
[213,] 0.04681010 0.1075494 0.03367876
[214,] 0.04638407 0.1063511 0.03341969
[215,] 0.04675684 0.1072499 0.03367876
[216,] 0.04654383 0.1075494 0.03335492
[217,] 0.04654383 0.1063511 0.03361399
[218,] 0.04665034 0.1069503 0.03361399
[219,] 0.04686335 0.1078490 0.03367876
[220,] 0.04675684 0.1075494 0.03361399
[221,] 0.04675684 0.1072499 0.03367876
[222,] 0.04654383 0.1069503 0.03348446
[223,] 0.04649057 0.1069503 0.03341969
[224,] 0.04638407 0.1063511 0.03341969
[225,] 0.04649057 0.1066507 0.03348446
[226,] 0.04622431 0.1054524 0.03341969
[227,] 0.04633081 0.1057519 0.03348446
[228,] 0.04617105 0.1054524 0.03335492
[229,] 0.04633081 0.1054524 0.03354922
[230,] 0.04654383 0.1066507 0.03354922
[231,] 0.04633081 0.1060515 0.03341969
[232,] 0.04659708 0.1063511 0.03367876
[233,] 0.04659708 0.1057519 0.03380829
[234,] 0.04643732 0.1057519 0.03361399
[235,] 0.04638407 0.1057519 0.03354922
[236,] 0.04643732 0.1063511 0.03348446
[237,] 0.04670359 0.1066507 0.03374352
[238,] 0.04659708 0.1060515 0.03374352
[239,] 0.04659708 0.1063511 0.03367876
[240,] 0.04643732 0.1066507 0.03341969
[241,] 0.04643732 0.1063511 0.03348446
[242,] 0.04649057 0.1066507 0.03348446
[243,] 0.04649057 0.1066507 0.03348446
[244,] 0.04659708 0.1066507 0.03361399
[245,] 0.04659708 0.1066507 0.03361399
[246,] 0.04659708 0.1069503 0.03354922
[247,] 0.04670359 0.1078490 0.03348446
[248,] 0.04659708 0.1078490 0.03335492
[249,] 0.04649057 0.1063511 0.03354922
[250,] 0.04654383 0.1072499 0.03341969
[251,] 0.04654383 0.1069503 0.03348446
[252,] 0.04649057 0.1066507 0.03348446
[253,] 0.04649057 0.1063511 0.03354922
[254,] 0.04638407 0.1057519 0.03354922
[255,] 0.04654383 0.1066507 0.03354922
[256,] 0.04638407 0.1060515 0.03348446
[257,] 0.04643732 0.1066507 0.03341969
[258,] 0.04643732 0.1063511 0.03348446
[259,] 0.04633081 0.1057519 0.03348446
[260,] 0.04622431 0.1063511 0.03322539
[261,] 0.04627756 0.1063511 0.03329016
[262,] 0.04622431 0.1063511 0.03322539
[263,] 0.04633081 0.1075494 0.03309585
[264,] 0.04627756 0.1075494 0.03303109
```

```
[265,] 0.04611780 0.1069503 0.03296632
[266,] 0.04606454 0.1066507 0.03296632
[267,] 0.04606454 0.1063511 0.03303109
[268,] 0.04627756 0.1069503 0.03316062
[269,] 0.04627756 0.1069503 0.03316062
[270,] 0.04633081 0.1072499 0.03316062
[271,] 0.04633081 0.1069503 0.03322539
[272,] 0.04638407 0.1072499 0.03322539
[273,] 0.04627756 0.1066507 0.03322539
[274,] 0.04633081 0.1066507 0.03329016
[275,] 0.04633081 0.1066507 0.03329016
[276,] 0.04633081 0.1066507 0.03329016
[277,] 0.04643732 0.1075494 0.03322539
[278,] 0.04638407 0.1069503 0.03329016
[279,] 0.04627756 0.1072499 0.03309585
[280,] 0.04654383 0.1075494 0.03335492
[281,] 0.04659708 0.1075494 0.03341969
[282,] 0.04649057 0.1069503 0.03341969
[283,] 0.04643732 0.1075494 0.03322539
[284,] 0.04659708 0.1084482 0.03322539
[285,] 0.04659708 0.1078490 0.03335492
[286,] 0.04643732 0.1075494 0.03322539
[287,] 0.04643732 0.1072499 0.03329016
[288,] 0.04670359 0.1075494 0.03354922
[289,] 0.04633081 0.1069503 0.03322539
[290,] 0.04638407 0.1066507 0.03335492
[291,] 0.04649057 0.1075494 0.03329016
[292,] 0.04654383 0.1075494 0.03335492
[293,] 0.04638407 0.1066507 0.03335492
[294,] 0.04643732 0.1069503 0.03335492
[295,] 0.04654383 0.1075494 0.03335492
[296,] 0.04649057 0.1069503 0.03341969
[297,] 0.04659708 0.1078490 0.03335492
[298,] 0.04643732 0.1069503 0.03335492
[299,] 0.04659708 0.1075494 0.03341969
[300,] 0.04643732 0.1072499 0.03329016
[301,] 0.04638407 0.1072499 0.03322539
[302,] 0.04638407 0.1066507 0.03335492
[303,] 0.04659708 0.1069503 0.03354922
[304,] 0.04627756 0.1063511 0.03329016
[305,] 0.04638407 0.1069503 0.03329016
[306,] 0.04665034 0.1075494 0.03348446
[307,] 0.04649057 0.1075494 0.03329016
[308,] 0.04643732 0.1069503 0.03335492
[309,] 0.04633081 0.1069503 0.03322539
[310,] 0.04627756 0.1069503 0.03316062
[311,] 0.04633081 0.1069503 0.03322539
[312,] 0.04654383 0.1072499 0.03341969
[313,] 0.04649057 0.1078490 0.03322539
[314,] 0.04654383 0.1078490 0.03329016
[315,] 0.04643732 0.1069503 0.03335492
[316,] 0.04659708 0.1075494 0.03341969
[317,] 0.04649057 0.1078490 0.03322539
[318,] 0.04649057 0.1075494 0.03329016
[319,] 0.04665034 0.1078490 0.03341969
[320,] 0.04638407 0.1072499 0.03322539
[321,] 0.04659708 0.1075494 0.03341969
[322,] 0.04643732 0.1072499 0.03329016
[323,] 0.04659708 0.1078490 0.03335492
[324,] 0.04654383 0.1078490 0.03329016
[325,] 0.04675684 0.1078490 0.03354922
[326,] 0.04654383 0.1072499 0.03341969
[327,] 0.04670359 0.1084482 0.03335492
[328,] 0.04649057 0.1069503 0.03341969
[329,] 0.04659708 0.1072499 0.03348446
[330,] 0.04654383 0.1075494 0.03335492
[331,] 0.04665034 0.1072499 0.03354922
[332,] 0.04659708 0.1072499 0.03348446
[333,] 0.04675684 0.1078490 0.03354922
```

```
 [ reached getOption("max.print") -- omitted 168 rows ]
>
> # Ploting the error rate drecrease along vs no. of trees

> plot(rnd_Forest)
>
> # Determining the important parameters

> importance(rnd_Forest)
                       0          1 MeanDecreaseAccuracy
Age            -4.468233  -8.236691            -9.720530
Rating          7.967131  18.258839            16.928392
Feedback_Count  1.171001  -4.879819            -2.851775
Division       -2.731518  -4.838858            -5.949374
Department     15.857477 -14.521295             9.315871
Class          18.606466 -15.360932             6.854518
predict_class  17.025415  10.282388            18.659581
prob_of_1     120.977855  57.112334            98.798242
               MeanDecreaseGini
Age                   113.94325
Rating                996.28816
Feedback_Count         50.14553
Division               18.84537
Department             26.48130
Class                  79.76658
predict_class        1157.35154
prob_of_1            2309.39443
>
> # Tuning random forest
>
> set.seed(100)
> tRnd_Forest <- tuneRF(x = traindf[,-c(3)],
+                       y = traindf$Recommended,
+                       mtryStart = 3,
+                       stepFactor = 1.5,
+                       ntreeTry = 51,
+                       improve = 0.0001,
+                       nodesize = 10,
+                       trace = TRUE,
+                       plot = TRUE,
+                       doBest = TRUE,
+                       importance = TRUE)
mtry = 3  OOB error = 4.7%
Searching left ...
mtry = 2        OOB error = 4.53%
0.03737259 1e-04
Searching right ...
mtry = 4        OOB error = 4.95%
-0.09294118 1e-04
>
> # Incorporating a predicted class and their probabilities column

> traindf$predict_class <- predict(tRnd_Forest, traindf, type = "class")
> traindf$prob_of_1 <- predict(tRnd_Forest, traindf, type = "prob")[,"1"]
> head(traindf)
  Age Rating Recommended Feedback_Count       Division Department
1  33      4           1              0      Initmates   Intimate
2  34      5           1              4        General    Dresses
3  60      3           0              0        General    Dresses
4  50      5           1              0 General Petite    Bottoms
5  47      5           1              6        General       Tops
6  49      2           0              4        General    Dresses
      Class predict_class prob_of_1
1 Intimates             1     1.000
2   Dresses             1     0.998
3   Dresses             0     0.042
4     Pants             1     1.000
5   Blouses             1     0.996
6   Dresses             0     0.020
```

```
>
> nrow(traindf)
[1] 18778
>
> # Developing a table to determine error rate
> tbl <- table(traindf$Recommended, traindf$predict_class)
> print(( tbl[1,2] + tbl[2,1] ) / 18778 )
[1] 0.05304079
>
> # Dividing the data into 10 quantiles based on probabilities
> # based of them recommmending the services to others

> qs <- quantile(traindf$prob_of_1, probs = seq(0, 1, length = 11))
> print(qs)
    0%    10%    20%    30%    40%    50%    60%    70%    80%    90% 
0.0020 0.0614 0.4080 0.9960 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 
  100% 
1.0000 
>
> # Determining accuracy of the data
> threshold <- 0.99
> mean(traindf$Recommended[traindf$prob_of_1 > threshold] == "1")
[1] 0.9964199
>
> # Fitting the tuned random forest to test dataset to get probabilities
> testdf$predict_class <- predict(tRnd_Forest, testdf, type = "class")
> testdf$prob_of_1 <- predict(tRnd_Forest, testdf, type = "prob")[,"1"]
> head(testdf)
   Age Rating Recommended Feedback_Count       Division Department
7   39      5           1              1 General Petite       Tops
17  34      3           1              2        General    Bottoms
25  55      5           1              0        General       Tops
26  31      3           0              0      Initmates   Intimate
27  33      2           0              0        General       Tops
33  21      5           1              0 General Petite    Bottoms
       Class predict_class prob_of_1
7      Knits             1     1.000
17     Pants             1     0.508
25   Blouses             1     1.000
26    Lounge             0     0.200
27  Sweaters             0     0.018
33     Pants             1     0.990
>
> nrow(testdf)
[1] 4694
>
> # Developing a table to determine error rate in the test dataset
> tbl <- table(testdf$Recommended, testdf$predict_class)
> print(( tbl[1,2] + tbl[2,1] ) / 4694 )
[1] 0.07030251
>
> # Determining accuarcy for test data
> mean(testdf$Recommended[testdf$prob_of_1 > threshold] == "1")
[1] 0.9906076
>
> # Optimize the nodesize to eliminate overfitting
```

```
> ##################################################################
> ####################### Decision tree / CART  ####################
> ##################################################################
> library(e1071)
> library(rattle)
Rattle: A free graphical interface for data science with R.
Version 5.3.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
Type 'rattle()' to shake, rattle, and roll your data.
> library(rpart.plot)
> library(RColorBrewer)
>
```

```
> Cart_control <- trainControl(method = "cv",
+                              number = 5,
+                              search = "random",
+                              savePredictions = T)
>
> cart_grid <- expand.grid(.cp = (0:20)*0.001)
>
> Cart_fit_cv <- train(Recommended~.,
+                      data = na.exclude(traindf),
+                      method = "rpart",
+                      trControl = Cart_control,
+                      tuneGrid = cart_grid)
>
> Cart_fit_cv
CART

18778 samples
    6 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 15023, 15022, 15022, 15023, 15022
Resampling results across tuning parameters:

  cp     Accuracy   Kappa
  0.000  0.9309298  0.7674098
  0.001  0.9356162  0.7958171
  0.002  0.9376397  0.8062499
  0.003  0.9376397  0.8062499
  0.004  0.9376397  0.8062499
  0.005  0.9376397  0.8062499
  0.006  0.9376397  0.8062499
  0.007  0.9376397  0.8062499
  0.008  0.9376397  0.8062499
  0.009  0.9376397  0.8062499
  0.010  0.9376397  0.8062499
  0.011  0.9376397  0.8062499
  0.012  0.9376397  0.8062499
  0.013  0.9376397  0.8062499
  0.014  0.9376397  0.8062499
  0.015  0.9376397  0.8062499
  0.016  0.9376397  0.8062499
  0.017  0.9376397  0.8062499
  0.018  0.9376397  0.8062499
  0.019  0.9376397  0.8062499
  0.020  0.9376397  0.8062499

Accuracy was used to select the optimal model using the
 largest value.
The final value used for the model was cp = 0.02.
> plot(Cart_fit_cv)
> # Developing a tree using complexity parameter with lowest error
>
> tree_rp <- rpart(Recommended ~ .,
+                  data = na.exclude(traindf),
+                  method = "class",
+                  control = rpart.control(cp = 0.001))
>
> tree_rp <- rpart(Recommended ~ .,
+                  data = na.exclude(traindf),
+                  method = "class",
+                  cp = 0.001)
>
> fancyRpartPlot(tree_rp, caption = NULL)

> # Confusion matrix

> confusionMatrix(tree_predictions, testdf$Recommended)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  777  272
         1   57 3588

               Accuracy : 0.9299
                 95% CI : (0.9222, 0.9371)
    No Information Rate : 0.8223
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7822

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

            Sensitivity : 0.9317
            Specificity : 0.9295
         Pos Pred Value : 0.7407
         Neg Pred Value : 0.9844
             Prevalence : 0.1777
         Detection Rate : 0.1655
   Detection Prevalence : 0.2235
      Balanced Accuracy : 0.9306

       'Positive' Class : 0

> # Visualizing a decision tree
> prp(tree_rp)
```

```
> ##############################################################
> ############### Sentimental Analysis #########################
> ##############################################################
>
> library(tidyr)
>
> # Concatinating the Title and Reveiws columns into single column
> s_df1 <- unite(df, "Title&Reviews", Title, Review.Text,
+              sep = " ", remove = T)
> names(s_df1)
 "Class.Name"
>
> # Building a corpus
>
> # Importing text mining library
> library(tm)
>
> Reviews_corpus <- iconv(s_df1$`Title&Reviews`, to = "UTF-8")
> R_corpus <- Corpus(VectorSource(Reviews_corpus))
```

```
> knitr::opts_chunk$set(echo = TRUE)
> library(knitr)
> opts_chunk$set(message = FALSE, warning = FALSE, cache = TRUE)
> options(width = 100, dplyr.width = 100)
> library(ggplot2)
> theme_set(theme_light())
>
> library(tidytext)
> library(dplyr)
>
> reviews_bigrams <- R_corpus %>%
+   unnest_tokens(bigram, text, token = "ngrams", n = 2)
Error in UseMethod("unnest_tokens_") :
  no applicable method for 'unnest_tokens_' applied to an object of clas
s "c('SimpleCorpus', 'Corpus')"
> reviews_bigrams
```

```
Error: object 'reviews_bigrams' not found
>
> # install.packages("NLP")
> # install.packages("data.table")
> # install.packages("rJava")
> # install.packages("RWeka")
> # install.packages("SnowballC")
> library(NLP)
> library(data.table)
> library(rJava)
Error: package or namespace load failed for ‘rJava’:
 .onLoad failed in loadNamespace() for 'rJava', details:
  call: fun(libname, pkgname)
  error: JAVA_HOME cannot be determined from the Registry
> library(RWeka)
Error: package or namespace load failed for ‘RWeka’:
 .onLoad failed in loadNamespace() for 'rJava', details:
  call: fun(libname, pkgname)
  error: JAVA_HOME cannot be determined from the Registry
> library(SnowballC)
> library(ggplot2)
> library(tm)
> library(RColorBrewer)
> library(wordcloud)
>
> library(tidyr)
>
> # Clean text
> r_corpus <- tm_map(R_corpus, tolower)
> inspect(r_corpus[1:5])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 5

[1]  absolutely wonderful - silky and sexy and comfortable
[2]  love this dress!  it's sooo pretty.  i happened to find it in a sto
re, and i'm glad i did bc i never would have ordered it online bc it's p
etite.  i bought a petite and am 5'8".  i love the length on me- hits ju
st a little below the knee.  would definitely be a true midi on someone 
who is truly petite.
[3] some major design flaws i had such high hopes for this dress and rea
lly wanted it to work for me. i initially ordered the petite small (my u
sual size) but i found this to be outrageously small. so small in fact t
hat i could not zip it up! i reordered it in petite medium, which was ju
st ok. overall, the top half was comfortable and fit nicely, but the bot
tom half had a very tight under layer and several somewhat cheap (net) o
ver layers. imo, a major design flaw was the net over layer sewn directl
y into the zipper - it c
[4] my favorite buy! i love, love, love this jumpsuit. it's fun, flirty,
 and fabulous! every time i wear it, i get nothing but great compliments!
[5] flattering shirt this shirt is very flattering to all due to the adj
ustable front tie. it is the perfect length to wear with leggings and it
 is sleeveless so it pairs well with any cardigan. love this shirt!!!
> r_corpus <- tm_map(r_corpus, removePunctuation)
Warning message:
In tm_map.SimpleCorpus(r_corpus, removePunctuation) :
  transformation drops documents
> inspect(r_corpus[1:5])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 5

[1]  absolutely wonderful  silky and sexy and comfortable
[2]  love this dress  its sooo pretty  i happened to find it in a store
and im glad i did bc i never would have ordered it online bc its petite
i bought a petite and am 58  i love the length on me hits just a little
below the knee  would definitely be a true midi on someone who is truly
petite
```

```
[3] some major design flaws i had such high hopes for this dress and rea
lly wanted it to work for me i initially ordered the petite small my usu
al size but i found this to be outrageously small so small in fact that
i could not zip it up i reordered it in petite medium which was just ok
overall the top half was comfortable and fit nicely but the bottom half
had a very tight under layer and several somewhat cheap net over layers
imo a major design flaw was the net over layer sewn directly into the zi
pper  it c
[4] my favorite buy i love love love this jumpsuit its fun flirty and fa
bulous every time i wear it i get nothing but great compliments
[5] flattering shirt this shirt is very flattering to all due to the adj
ustable front tie it is the perfect length to wear with leggings and it
is sleeveless so it pairs well with any cardigan love this shirt
> r_corpus <- tm_map(r_corpus, removeNumbers)
Warning message:
In tm_map.SimpleCorpus(r_corpus, removeNumbers) :
  transformation drops documents
> inspect(r_corpus[1:5])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 5

[1]  absolutely wonderful  silky and sexy and comfortable
[2]  love this dress  its sooo pretty  i happened to find it in a store
and im glad i did bc i never would have ordered it online bc its petite
i bought a petite and am   i love the length on me hits just a little be
low the knee  would definitely be a true midi on someone who is truly pe
tite
[3] some major design flaws i had such high hopes for this dress and rea
lly wanted it to work for me i initially ordered the petite small my usu
al size but i found this to be outrageously small so small in fact that
i could not zip it up i reordered it in petite medium which was just ok
overall the top half was comfortable and fit nicely but the bottom half
had a very tight under layer and several somewhat cheap net over layers
imo a major design flaw was the net over layer sewn directly into the zi
pper  it c
[4] my favorite buy i love love love this jumpsuit its fun flirty and fa
bulous every time i wear it i get nothing but great compliments
[5] flattering shirt this shirt is very flattering to all due to the adj
ustable front tie it is the perfect length to wear with leggings and it
is sleeveless so it pairs well with any cardigan love this shirt
> r_corpus <- tm_map(r_corpus, removeWords, stopwords('english'))
Warning message:
In tm_map.SimpleCorpus(r_corpus, removeWords, stopwords("english")) :
  transformation drops documents
> inspect(r_corpus[1:5])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 5

[1]  absolutely wonderful  silky  sexy  comfortable
[2]  love  dress   sooo pretty   happened  find    store  im glad   bc
never   ordered  online bc  petite   bought  petite     love  length
hits just  little   knee   definitely   true midi  someone   truly petit
e
[3]  major design flaws    high hopes   dress  really wanted   work    i
nitially ordered  petite small  usual size   found    outrageously small
small  fact     zip    reordered   petite medium   just ok overall  top
half  comfortable  fit nicely   bottom half    tight  layer  several som
ewhat cheap net  layers imo  major design flaw   net  layer sewn directl
y   zipper   c
[4]  favorite buy  love love love  jumpsuit  fun flirty  fabulous every
time  wear   get nothing  great compliments
[5] flattering shirt  shirt   flattering   due   adjustable front tie
perfect length  wear  leggings    sleeveless   pairs well   cardigan lov
e  shirt
>
> # Removing url
> cleanset <- tm_map(r_corpus, content_transformer(remove_url))
```

```
Warning message:
In tm_map.SimpleCorpus(r_corpus, content_transformer(remove_url)) :
  transformation drops documents
> inspect(cleanset[1:5])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 5

[1]  absolutely wonderful  silky  sexy  comfortable
[2]  love  dress   sooo pretty   happened  find    store  im glad   bc
never   ordered  online bc  petite   bought  petite      love  length
hits just  little   knee   definitely   true midi  someone   truly petit
e
[3]  major design flaws    high hopes   dress  really wanted   work    i
nitially ordered  petite small  usual size   found    outrageously small
small  fact      zip    reordered   petite medium   just ok overall  top
half  comfortable  fit nicely   bottom half    tight  layer  several som
ewhat cheap net  layers imo  major design flaw   net  layer sewn directl
y   zipper   c
[4]  favorite buy  love love love  jumpsuit  fun flirty  fabulous every
time  wear   get nothing  great compliments
[5] flattering shirt  shirt   flattering   due   adjustable front tie
perfect length  wear  leggings    sleeveless   pairs well   cardigan lov
e  shirt
>
> # Removing username
> remove_username <- function(x) gsub('@', '', x)
> cleanset <- tm_map(r_corpus, content_transformer(remove_username))
Warning message:
In tm_map.SimpleCorpus(r_corpus, content_transformer(remove_username)) :
  transformation drops documents
> inspect(cleanset[1:5])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 5

[1]  absolutely wonderful  silky  sexy  comfortable
[2]  love  dress   sooo pretty   happened  find    store  im glad   bc
never   ordered  online bc  petite   bought  petite      love  length
hits just  little   knee   definitely   true midi  someone   truly petit
e
[3]  major design flaws    high hopes   dress  really wanted   work    i
nitially ordered  petite small  usual size   found    outrageously small
small  fact      zip    reordered   petite medium   just ok overall  top
half  comfortable  fit nicely   bottom half    tight  layer  several som
ewhat cheap net  layers imo  major design flaw   net  layer sewn directl
y   zipper   c
[4]  favorite buy  love love love  jumpsuit  fun flirty  fabulous every
time  wear   get nothing  great compliments
[5] flattering shirt  shirt   flattering   due   adjustable front tie
perfect length  wear  leggings    sleeveless   pairs well   cardigan lov
e  shirt
>
> cleanset <- tm_map(cleanset, stripWhitespace)
Warning message:
In tm_map.SimpleCorpus(cleanset, stripWhitespace) :
  transformation drops documents
> inspect(cleanset[1:5])
+                                                      max = 3))
> tdmreview <- TermDocumentMatrix(cleanset, control =
+                                     list(tokenize = tokreview))
> TermFreqReview <- rowSums(as.matrix(tdmreview))
> TermFreqVectorReview <- as.list(TermFreqReview)

>
```

```
> library(tidyr)
```

E-Commerce Dataset

```
> names(s_df1)
 [1] "X0"                  "Clothing.ID"             "Age"
 [4] "Title&Reviews"       "Rating"                  "Recommended.IND"
 [7] "Positive.Feedback.Count" "Division.Name"       "Department.Name"
[10] "Class.Name"
>
> # Building a corpus
>
> # Importing text mining library
> library(tm)
>
> Reviews_corpus <- iconv(s_df1$`Title&Reviews`, to = "UTF-8")
> R_corpus <- Corpus(VectorSource(Reviews_corpus))
> library(tidytext)
> library(itunesr)
Error in library(itunesr) : there is no package called ‘itunesr’
> library(tidyverse)
-- Attaching packages --------------------------------------- tidyverse 1.
2.1 --
v tibble  2.1.3     v purrr   0.3.2
v readr   1.3.1     v stringr 1.4.0
v tibble  2.1.3     v forcats 0.4.0
-- Conflicts ------------------------------------------ tidyverse_conflict
s() --
x ggplot2::annotate()   masks NLP::annotate()
x data.table::between() masks dplyr::between()
x dplyr::filter()       masks stats::filter()
x data.table::first()   masks dplyr::first()
x dplyr::lag()          masks stats::lag()
x data.table::last()    masks dplyr::last()
x purrr::transpose()    masks data.table::transpose()
> ecom_reviews_5 <- data.frame(
+   txt = s_df1$`Title&Reviews`[s_df1$Rating == 5],
+                              stringsAsFactors = FALSE)
> ecom_reviews_5 %>%
+   unnest_tokens(output = word, input = txt) %>%
+   anti_join(stop_words) %>%
+   count(word, sort = TRUE)
Joining, by = "word"
# A tibble: 10,750 x 2
   word           n
   <chr>      <int>
 1 love        7660
 2 dress       6800
 3 size        4978
 4 top         4429
 5 fit         4250
 6 wear        4085
 7 perfect     3676
 8 color       2731
 9 flattering  2709
10 beautiful   2614
# ... with 10,740 more rows
> ecom_reviews_5 %>%
+   unnest_tokens(word, txt, token = "ngrams", n = 2) %>%
+   separate(word, c("word1", "word2"), sep = " ") %>%
+   filter(!word1 %in% stop_words$word) %>%
+        caption = " Bigram Freq")
> ecom_reviews_5 %>%
+   unnest_tokens(word, txt, token = "ngrams", n = 2) %>%
+   separate(word, c("word1", "word2"), sep = " ") %>%
+   filter(!word1 %in% stop_words$word) %>%
+   filter(!word2 %in% stop_words$word) %>%
+   unite(word,word1, word2, sep = " ") %>%
+   count(word, sort = TRUE) %>%
+   slice(1:10) %>%
+   ggplot() + geom_bar(aes(word, n), stat = "identity", fill = "Green") +
+   theme_minimal() +
+   coord_flip() +
```

```
+   labs(title = "Top Bigrams for 5 star Rating",
+        caption = " Bigram Freq")
> ecom_reviews_1 <- data.frame(
+   txt = s_df1$`Title&Reviews`[s_df1$Rating == 1],
+   stringsAsFactors = FALSE)
> ecom_reviews_1 %>%
# A tibble: 3,081 x 2
   word             n
   <chr>        <int>
 1 dress          400
 2 top            282
 3 fabric         273
 4 fit            265
 5 size           192
 6 quality        164
 7 shirt          162
 8 material       161
 9 wear           157
10 disappointed   139
# ... with 3,071 more rows
> ecom_reviews_1 %>%
+   separate(word, c("word1", "word2"), sep = " ") %>%
+   filter(!word1 %in% stop_words$word) %>%
+   filter(!word2 %in% stop_words$word) %>%
+   unite(word,word1, word2, sep = " ") %>%
+   count(word, sort = TRUE) %>%
+   slice(1:10) %>%
+   ggplot() + geom_bar(aes(word, n), stat = "identity", fill = "#de5833")
+
+   theme_minimal() +
+   coord_flip() +
+   labs(title = "Top Bigrams for 1 star Rating",
+        caption = "Bigram freq")
> ecom_reviews_Not_Recommended <- data.frame(
+   txt = s_df1$`Title&Reviews`[s_df1$Recommended.IND == 0],
+   stringsAsFactors = FALSE)
> ecom_reviews_Not_Recommended %>%
+   anti_join(stop_words) %>%
+   count(word, sort = TRUE)
Joining, by = "word"
# A tibble: 6,702 x 2
   word         n
   <chr>    <int>
 1 dress     2100
 2 top       1681
 3 fit       1534
 4 fabric    1368
 5 size      1229
 6 love      1069
 7 material   781
 8 color      779
 9 wear       773
10 cute       741
# ... with 6,692 more rows
> ecom_reviews_Not_Recommended %>%
+   unnest_tokens(word, txt, token = "ngrams", n = 2) %>%
+   separate(word, c("word1", "word2"), sep = " ") %>%
+   filter(!word1 %in% stop_words$word) %>%
+   filter(!word2 %in% stop_words$word) %>%
+   unite(word,word1, word2, sep = " ") %>%
+   count(word, sort = TRUE) %>%
+   slice(1:10) %>%
+   ggplot() + geom_bar(aes(word, n), stat = "identity", fill = "#de5833")
+
+   theme_minimal() +
+   coord_flip() +
+   labs(title = "Top Bigrams for where customers will not recommend",
+        caption = "Bigram freq")
```

```
> # Term Document Matrix
> tdm <- TermDocumentMatrix(cleanset)
Error in TermDocumentMatrix(cleanset) : object 'cleanset' not found
> tdm
> tdm <- as.matrix(tdm)
Error in as.matrix(tdm) : object 'tdm' not found
> tdm[1:10,1:10]
>
> # Checking the dimension of the tdm matrix
> dim(tdm)
>
>
> # Creating the Term Document Matrix to remove sparse terms
> tdm <- TermDocumentMatrix(cleanset)
Error in TermDocumentMatrix(cleanset) : object 'cleanset' not found
>
>
> ####################################################################
>
> # Remove sparse terms that occur in less 96% of the documents
> # This is an effective way to remove outliers
> sparse_96 <- removeSparseTerms(tdm, 0.96)
> sparse_96
> dim(sparse_96)
>
> # After removing sparse terms we get 183 terms that
> sparse1_96 <- as.matrix(sparse_96)
> sparse1_96[1:10,1:10]
>
> # Barplot
> w_96 <- rowSums(sparse1_96)
> w_96 <- subset(w_96 , w_96 >= 500)
> barplot(w_96, las = 2, col = rainbow(50))
> word_freq_96 <- data.frame(term = names(w_96), freq = w_96)
> word_freq_96
>
> ####################################################################
>
> # Remove sparse terms that occur in less 95% of the documents
> # This is an effective way to remove outliers
> sparse_95 <- removeSparseTerms(tdm, 0.95)
> sparse_95
> dim(sparse_95)
>
> # After removing sparse terms we get 183 terms that
> sparse1_95 <- as.matrix(sparse_95)
> sparse1_95[1:10,1:10]
>
> # Barplot
> w_95 <- rowSums(sparse1_95)
> w_95 <- subset(w_95, w_95 >= 500)
> barplot(w_95, las = 2, col = rainbow(50))
> word_freq_95 <- data.frame(term = names(w_95), freq = w_95)
:
  object 'w_95' not found
> word_freq_95
>
> ####################################################################
>
> # Remove sparse terms that occur in less 92% of the documents
> # This is an effective way to remove outliers
> sparse_92 <- removeSparseTerms(tdm, 0.92)
> sparse_92
> dim(sparse_92)
>
> # After removing sparse terms we get 183 terms that
> sparse1_92 <- as.matrix(sparse_92)
> sparse1_92[1:10,1:10]
>
```

```
> # Barplot
> w_92 <- rowSums(sparse1_92)
> w_92 <- subset(w_92, w_92 >= 500)
> barplot(w_92, las = 2, col = rainbow(50))
  object 'w_92' not found
> word_freq_92 <- data.frame(term = names(w_92), freq = w_92)
  object 'w_92' not found
> word_freq_92
>
> ##################################################################
> dim(sparse_97)
>
> # After removing sparse terms we get 183 terms that
> sparse1_97 <- as.matrix(sparse_97)
> sparse1_97[1:10,1:10]
>
> # Barplot
> w_97 <- rowSums(sparse1_97)
> w_97 <- subset(w_97, w_97 >= 500)
> barplot(w_97, las = 2, col = rainbow(50))
  object 'w_97' not found
> word_freq_97 <- data.frame(term = names(w_97), freq = w_97)
  object 'w_97' not found
> word_freq_97
>
> ##################################################################
>
> library(wordcloud)
> x <- sort(rowSums(sparse1_97), decreasing = T)
> set.seed(123)
> wordcloud(words = names(x),
+           freq = x,
+           max.words = 150,
+           random.order = F,
+           colors = brewer.pal(8, 'Dark2'),
+           scale = c(3, 0.3),
+           rot.per = 0.2)
> ##################################################################
>
> library(tm)
> list1 <- findAssocs(tdm, "short", 0.1)
> corr_df1 <- t(data.frame(t(sapply(list1, c))))
> corr_df1
>
> barplot(t(as.matrix(corr_df1)), beside = TRUE, xlab = "words",
+         ylab = "correlation", col = "blue",
+         main = "Fabric Correlation with other words",
+         border = "black")
>
> ##################################################################
>
> list2 <- findAssocs(tdm, "retailer", 0.09)
> corr_df2 <- t(data.frame(t(sapply(list2, c))))
> corr_df2
>
> barplot(t(as.matrix(corr_df2)), beside = TRUE, xlab = "words",
+         ylab = "correlation", col = "red",
+         main = "Fabric Correlation with other words",
+         border = "black")
>
> ##################################################################
>
> list3 <- findAssocs(tdm, "dress", 0.1)
> corr_df3 <- t(data.frame(t(sapply(list3, c))))
> corr_df3
>
> barplot(t(as.matrix(corr_df3)), beside = TRUE, xlab = "words",
+         ylab = "correlation", col = "yellow",
+         main = "Fabric Correlation with other words",
```

```
+         border = "black")
>
> ##################################################################
>
> list4 <- findAssocs(tdm, "love", 0.075)
> corr_df4 <- t(data.frame(t(sapply(list4, c))))
> corr_df4
>
> barplot(t(as.matrix(corr_df4)), beside = TRUE, xlab = "words",
+         ylab = "correlation", col = "yellow",
+         main = "Fabric Correlation with other words",
+         border = "black")
>
> ##################################################################
>
> list5 <- findAssocs(tdm, "sweater", 0.075)
> corr_df5
>
> barplot(t(as.matrix(corr_df5)), beside = TRUE, xlab = "words",
+         ylab = "correlation", col = "yellow",
+         main = "Fabric Correlation with other words",
+         border = "black")
>
>
> ##################################################################
>
> list6 <- findAssocs(tdm, "material", 0.075)
> corr_df6 <- t(data.frame(t(sapply(list6, c))))
> corr_df6
>
> barplot(t(as.matrix(corr_df6)), beside = TRUE, xlab = "words",
+         ylab = "correlation", col = "yellow",
+         main = "Fabric Correlation with other words",
+         border = "black")
>
> ##################################################################
>
> list7 <- findAssocs(tdm, "shirt", 0.06)
> corr_df7 <- t(data.frame(t(sapply(list7, c))))
> corr_df7
>
> barplot(t(as.matrix(corr_df7)), beside = TRUE, xlab = "words",
+         ylab = "correlation", col = "yellow",
+         main = "Fabric Correlation with other words",
+         border = "black")
>
> ##################################################################
>
> list8 <- findAssocs(tdm, "fabric", 0.08)

> corr_df8 <- t(data.frame(t(sapply(list8, c))))

> corr_df8
>
> barplot(t(as.matrix(corr_df8)), beside = TRUE, xlab = "words",
+         ylab = "correlation", col = "yellow",
+         main = "Fabric Correlation with other words",
+         border = "black")
>
> ##################################################################
>
> list9 <- findAssocs(tdm, "price", 0.08)
> corr_df9 <- t(data.frame(t(sapply(list9, c))))
> corr_df9
>
> barplot(t(as.matrix(corr_df9)), beside = TRUE, xlab = "words",
+         ylab = "correlation", col = "yellow",
+         main = "Fabric Correlation with other words",
+         border = "black")
```

```
>
> ################################################################
>
> list10 <- findAssocs(tdm, "sale", 0.08)
> corr_df10 <- t(data.frame(t(sapply(list10, c))))
> corr_df10
>
> barplot(t(as.matrix(corr_df10)), beside = TRUE, xlab = "words",
+         ylab = "correlation", col = "yellow",
+         main = "Fabric Correlation with other words",
+         border = "black")

>
> ################################################################
>
> list11 <- findAssocs(tdm, "fit", 0.08)

> corr_df11 <- t(data.frame(t(sapply(list11, c))))
> corr_df11
>
> barplot(t(as.matrix(corr_df11)), beside = TRUE, xlab = "words",
+         ylab = "correlation", col = "yellow",
+         main = "Fabric Correlation with other words",
+         border = "black")
>
> ################################################################
>
>
>
> # Sentiment Analysis
> library(syuzhet)
> library(lubridate)

Attaching package: ‘lubridate’

The following object is masked from ‘package:igraph’:

    %--%

The following objects are masked from ‘package:data.table’:

    hour, isoweek, mday, minute, month, quarter, second, wday,
    week, yday, year

The following object is masked from ‘package:base’:

    date

> library(scales)

Attaching package: ‘scales’

The following object is masked from ‘package:syuzhet’:

    rescale

> library(dplyr)
> library(ggplot2)
>
> # Reading file
> Reviews_corpus <- iconv(df1$`Title&Reviews`, to = "UTF-8")

  object 'df1' not found
>
> # Obtaining sentiment scores
> s <- get_nrc_sentiment(Reviews_corpus)

> head(s)
```

```
> Reviews_corpus[6]

>
> # Barplot
> barplot(colSums(s),
+         las = 2,
+         col = rainbow(10),
+         ylab = 'Count',
+         main = 'Sentiment Scores for Women Clothing Reviews')
>
```

Feed back from mentors for notes 1 and 2

Notes 1

Good Effort !!! Problem Understanding and Data Report was presented and well explained. EDA, Univariate Analysis only analysis for 3 variables was done, should have explored the data more in detail. All the best for next phase of the Capstone Project.

Notes 2

What you have presented and submitted in text Mining and Sentiment Analysis. You should used classification algorithms like CART/ Random Forest on the dataset. Also Clustering & LDA could have been tried. Please re-work on the model building before presentation and final report.