

Web and Social Media Analytics  
Group Assignment – Group5

**Submitted by:**

Naveen Kumar Sambangi

[Parthasarathy Budithi](https://olympus.greatlearning.in/groups/13039/users/7847)

[Sunitha CK](https://olympus.greatlearning.in/groups/13039/users/7858)

[Yogeshwar Singh Rawat](https://olympus.greatlearning.in/groups/13039/users/7865)

**Problem Statement:**

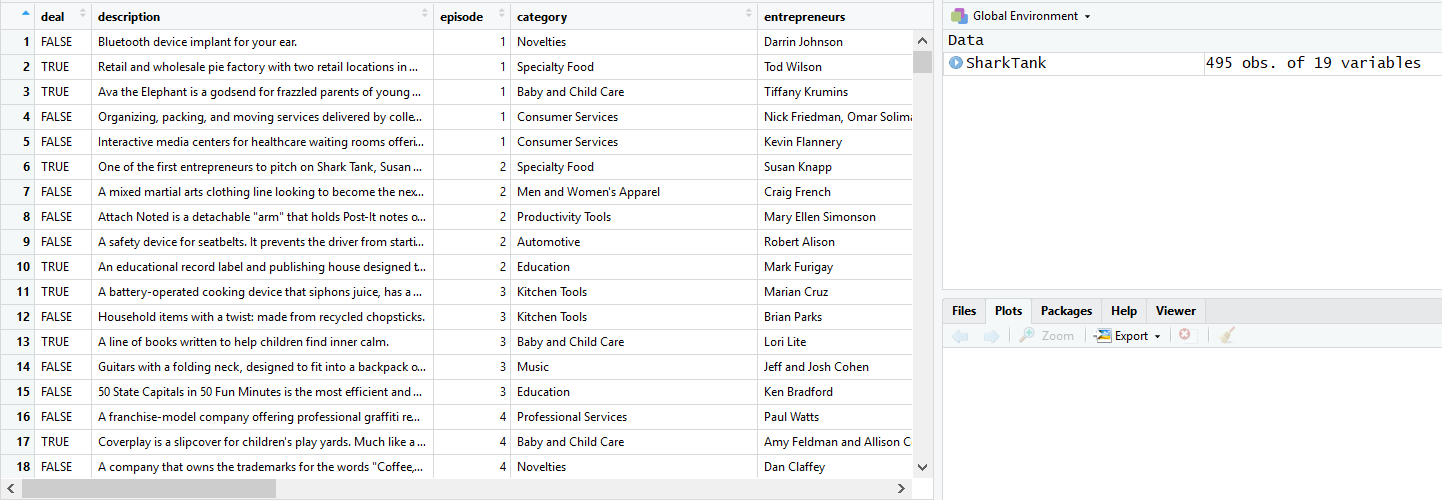
1. A dataset of Shark Tank episodes is made available. It contains 495 entrepreneurs making their pitch to the VC sharks.
2. You will ONLY use “Description” column for the initial text mining exercise.
3. Step 1:
   1. Extract the text into text corpus and perform following operations:
      1. Create DTM
      2. Use “Deal” as a Dependent Variable
      3. Use CART model and arrive at your CART diagram
      4. Build Logistic Regression Model and find out your accuracy of the model
      5. Build random Forest model and arrive at your varImpPlot
4. Step 2:
   1. Now, add a variable to your analysis called as “ratio”. This variable is “askedfor/valuation”. (This variable is to be added as a column to your data frame in Step 1.)
   2. Rebuild “New” models- CART, random Forest and Logistic Regression.

**Introduction:** The Shark Tank (Shark+Tank+Companies) dataset contains 495 entrepreneurs making their pitch to the VC sharks. We will use the description column from the given dataset for text mining and apply the predictive modelling algorithms to identify which investors make their deal (True) or not (False).

**Steps to resolve the problem:**

**1)Loading the Dataset:**  Firstly, we load the dataset into R and identify the number of variables/columns and records/rows.

|  |
| --- |
| setwd("E:/r direct/Web & Social Analytics/WSM Assignment") |
| SharkTank = read.csv("Shark+Tank+Companies.csv", stringsAsFactors=FALSE)  View(SharkTank)  table(SharkTank$deal) |



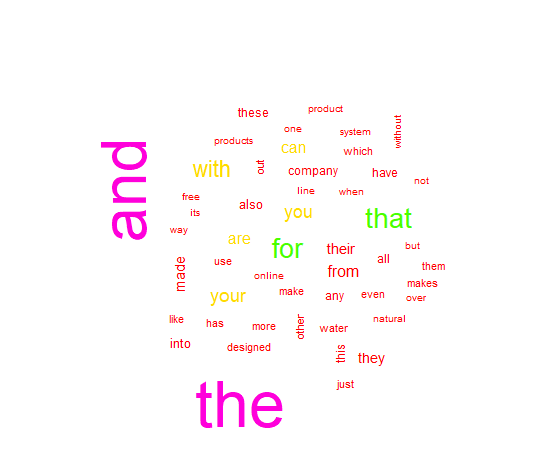
The dataset contains the deal variable of 244 false and 251 true values which indicates its balanced dataset.

**2)Importing the required packages and Corpus Creation:** Loading the required packages for text mining as mentioned below.

|  |
| --- |
| library(tm) |
| library(SnowballC) |
| library(wordcloud) |

Creating corpus with required variable (Description) to use the tm package.

|  |
| --- |
| corpusSharkTank = Corpus(VectorSource(SharkTank$description)) |
|  |

Word cloud before performing the data cleaning and stemming process, with the below image we have identified that the word **and** & **the** are unwanted words hence we can remove these words. 

**3)Data Cleaning:** To perform the data cleaning we are using the below mentioned steps.

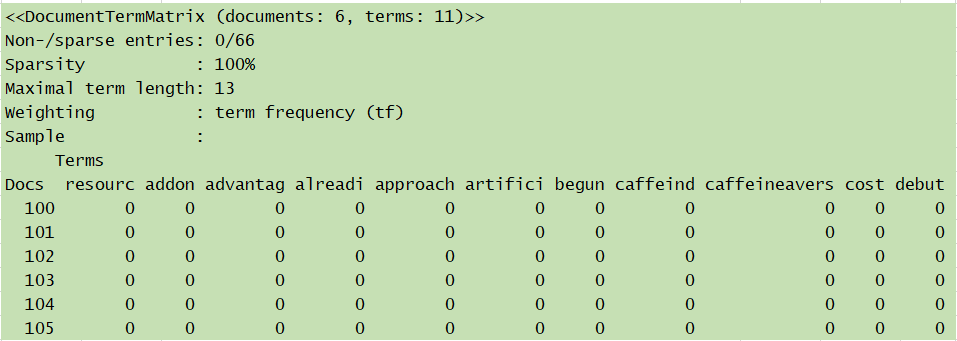
* Converting into lower-case
* Removing Punctuations.
* Removing stop words.
* Removing extra spaces.
* Stem the document.

After performed the data cleaning the world cloud looks as mentioned below.



**4)DTM (Document Term Matrix) Creation:**

We need to create the document term matrix for analyzing frequencies of unique words in the entire document. The DTM contains 495 rows and 3501 words/columns and we have visualized the DTM to check the sample words from 100-105 rows and 500 to 515 columns.



Checking some sample high frequency and low frequency words.

**Low Frequency Words:**

[1] "devic" "new" "children" "easi" "make" "one" "play"

[8] "provid" "servic" "offer" "design" "flavor" "food" "includ"

[15] "line" "product" "store" "cloth" "look" "mix" "get"

[22] "work" "made" "help" "fit" "fun" "keep" "kid"

[29] "compani" "also" "can" "like" "protect" "time" "use"

[36] "sell" "accessori" "onlin" "user" "bar" "just" "natur"

[43] "safe" "custom" "bottl" "featur" "allow" "need" "way"

[50] "busi" "even" "take" "creat" "without" "come" "system"

[57] "home" "water" "peopl"

**High Frequency Words:**

"bluetooth" "ear" "implant" "factori"

[5] "jersey" "locat" "pie" "retail"

[9] "two" "wholesal" "administ" "ava"

[13] "dispens" "eleph" "everywher" "experi"

[17] "frazzl" "godsend" "littl" "medicin"

[21] "parent" "play" "posit" "reinforc"

[25] "talk" "turn" "young" "colleg"

[29] "deliv" "move" "organ" "pack"

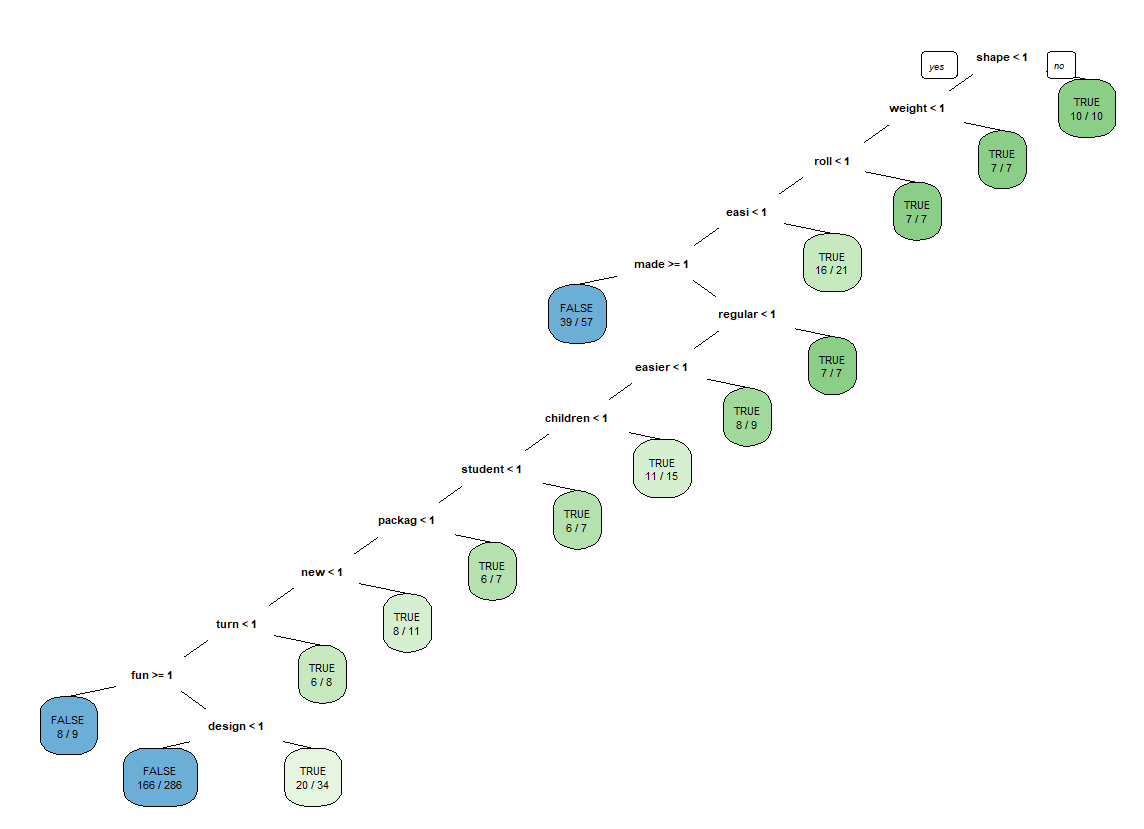
[33] "women" "access" "center" "educ"

[37] "healthcar" "inform" "interact" "media"

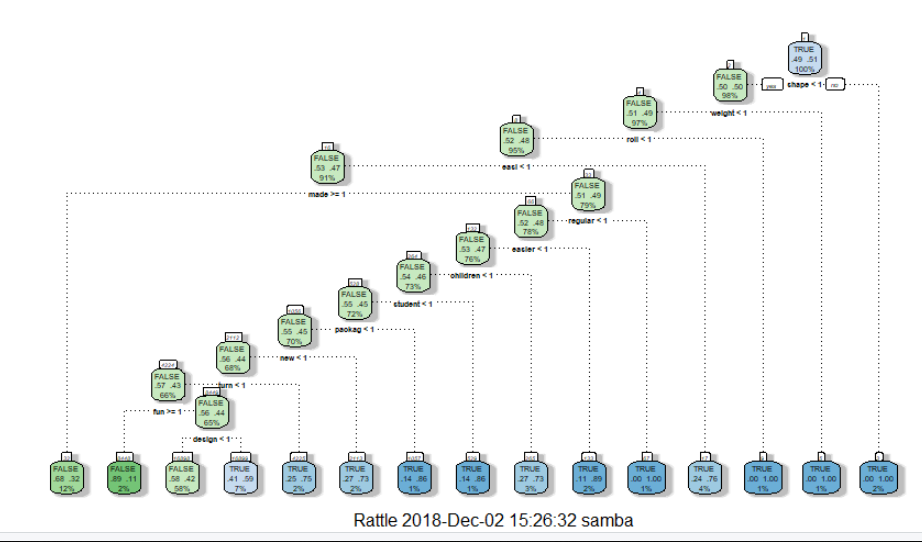
**Checking the Sparsity:**

Removing the words which are not present in at least 5 statements/documents, post this activity the number of column’s has been reduced from 3501 to 900.

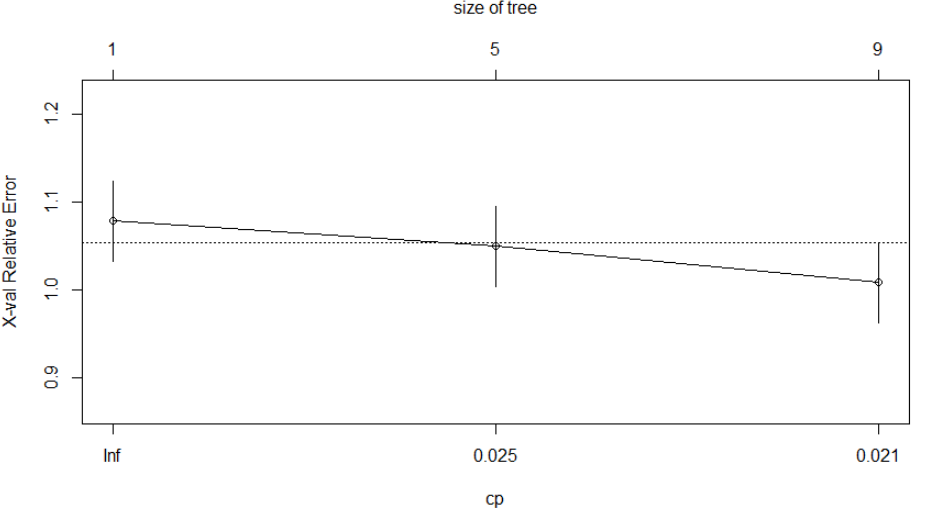
**5)CART Model:** Executed cart model on given dataset using rpart command and below is the final tree.



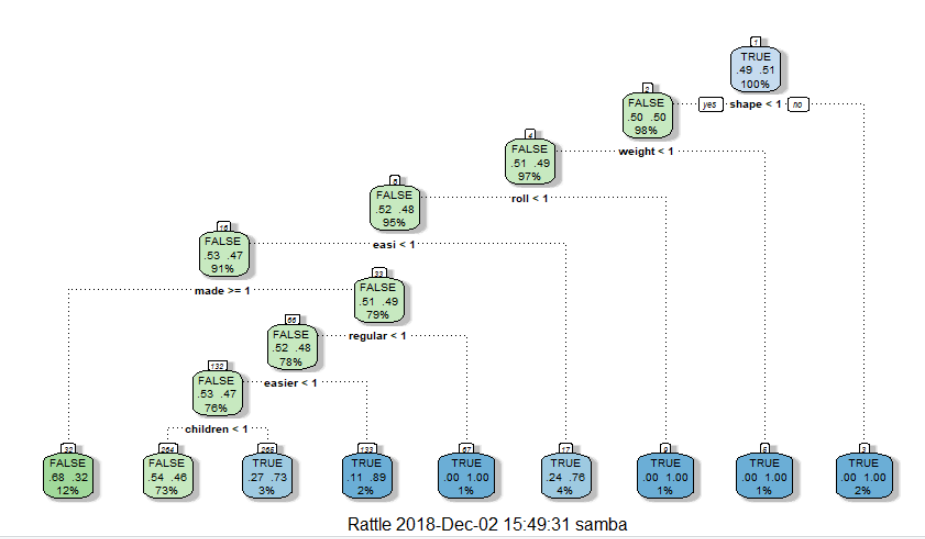
**Fancy R Plot:**



Below is the plotcp of the model.



**Pruned Tree:** Based on printcp value of model, pruned the tree by selecting complexity parameter having smallest cross validation error. Below is the resulting tree:

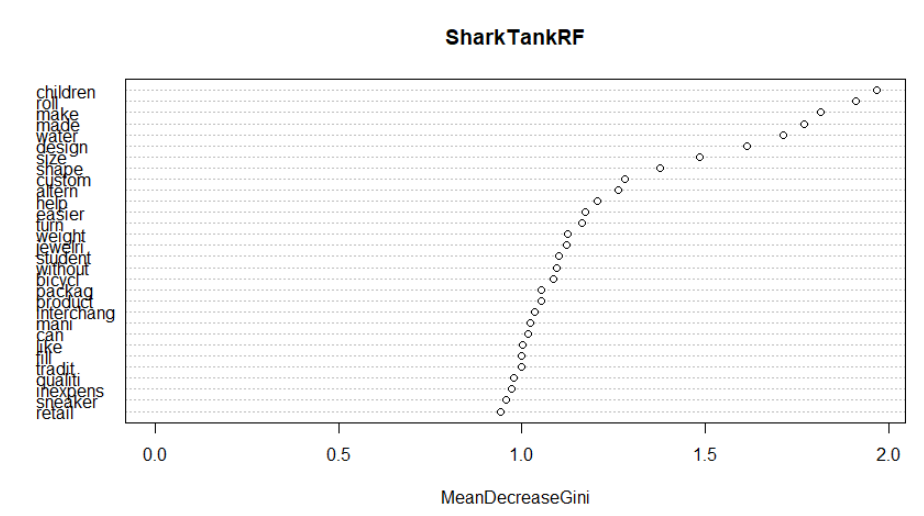


**Node 1**: In given dataset, the description contains the word shape at least once then the it turned to be a deal most of the times (10/10) hence shape is the highest priority word.

**Node 2**: The Description doesn’t contain the word shape at least once and it contains the word weight at least once then it turned to be deal in (7/7) times. The second priority word is weight.

**Node 3**: The Description doesn’t contain the word shape and weight at least once and it contains the word roll at least once then it turned to be deal in (7/7) times. The third priority word is roll.

**6)Random Forest Model:** Executed random forest model on given dataset using randomforest command and below are the important words.



From the random forest algorithm, we identified that the description contains the word children have the highest priority and then followed by roll and make and so on.

**7)Logistic Regression Model:** Executed logistic regression model on given dataset using glm command and below is the confusion matrix table and got the accuracy as 50.5% with cutoff value as 0.5.

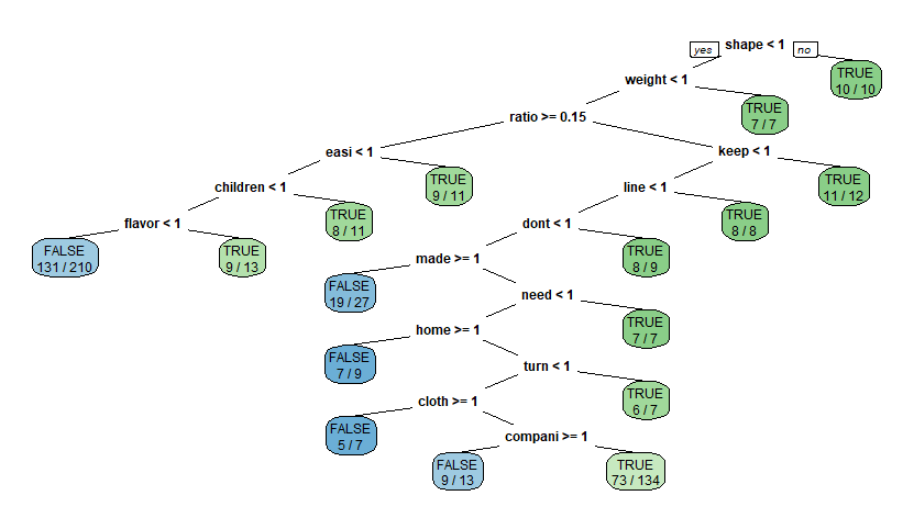
|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Reference | |
| Prediction |  | FALSE | TRUE |
| FALSE | **135** | **109** |
| TRUE | **136** | **115** |

**8)Inclusion of Ratio variable and comparison of various models:**

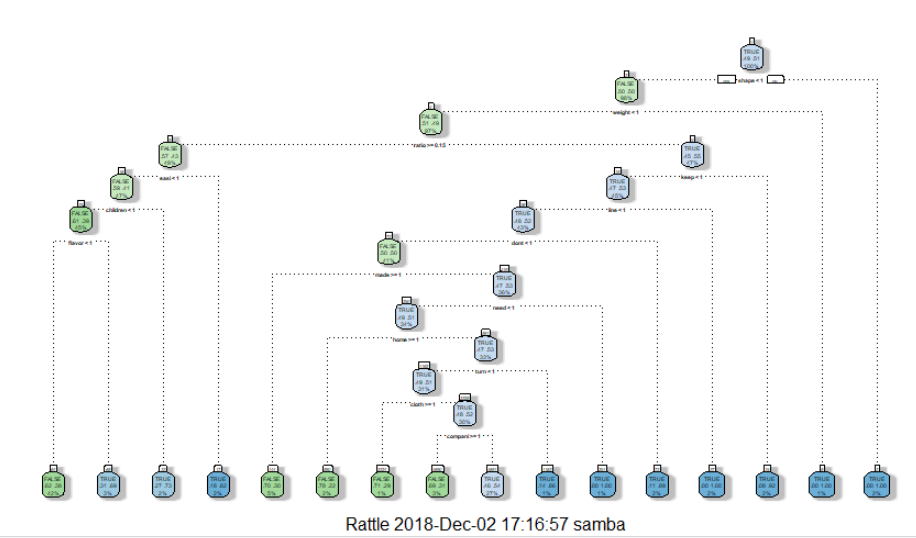
Now added the ratio variable to the data frame and performed the various models.

Ratio= askedfor/valuation

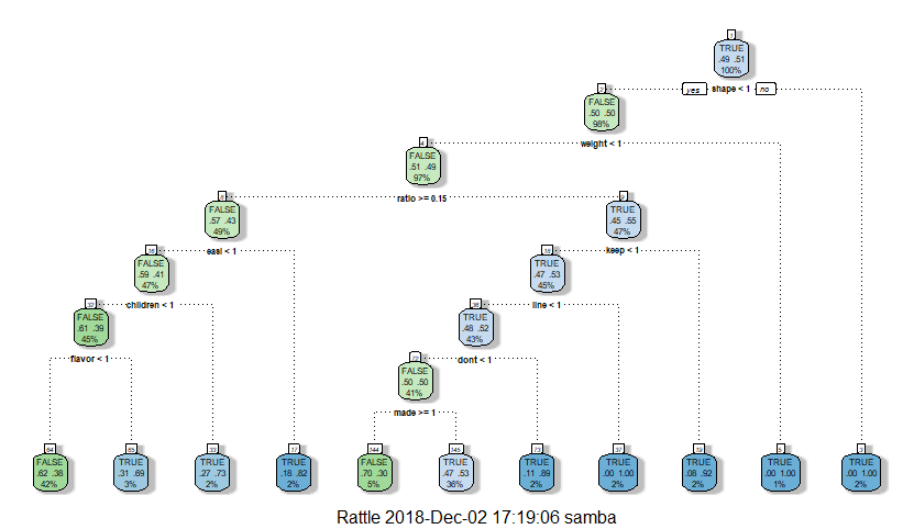
**8.a) CART Model:** Executed cart model on given dataset using rpart command and below is the final tree after adding the ratio variable.



**Fancy R plot:**



**Pruned Tree:** Based on printcp value of model, pruned the tree by selecting complexity parameter having smallest cross validation error. Below is the resulting tree:

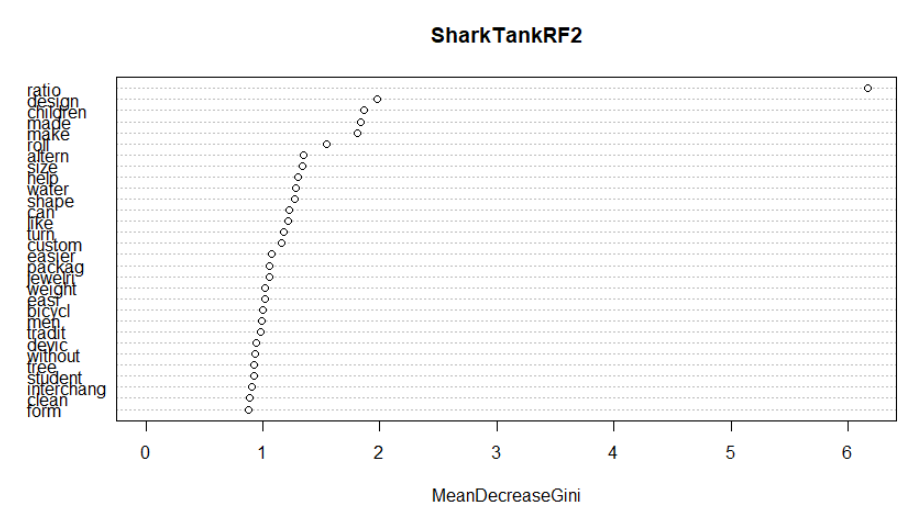


**Node 1**: After adding the ration variable to the given dataset, the description contains the word shape at least once then the it turned to be a deal most of the times (10/10), hence shape is the highest priority word.

**Node 2**: The Description doesn’t contain the word shape at least once and it contains the word weight at least once then it turned to be deal in (7/7) times. The second priority word is weight.

**Node 3**: The Description doesn’t contain the word shape and weight at least once and it contains the word ratio value less than or equal to 0.15 then it looking for the word keep at least once and turned to be deal in (11/12) times. Hence the third priority word is ratio.

**8.b) Random Forest Model:** After adding the ratio variable executed random forest model on given dataset using randomforest command and below are the important words.



From the random forest algorithm, we identified that the description contains the word ratio have the highest priority and then followed by design and children and so on. By adding the ratio variable to the model priority of the word has been changed.

**8.c) Logistic Regression Model:** After adding the ratio executed logistic regression model on given dataset using glm command and below is the confusion matrix table and got the accuracy as 54.7% with cutoff value as 0.5. The accuracy of the model has been increased from 50.5% to 54.7%.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Reference | |
| Prediction |  | FALSE | TRUE |
| FALSE | **125** | **119** |
| TRUE | **105** | **146** |

**9)Model’s Comparison:** Below are accuracy percentages of model before and after adding the ratio variable, hence it has significant role while building the model.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Performance Metric** | **CART** | | **Random Forest** | | **Logistic Regression** | |
| Before | After | Before | After | Before | **After** |
| Accuracy Classification score | 0.65 | 0.66 | 0.553 | 0.557 | 50.7 | 54.7 |

**10)Conclusion:** By using description and description+ratio variables of CART Model we were able to predict around 65.65% and 66.06% accurate results respectively. By using Random forest, we were able to predict 55.35% and 55.75% accurate results using only description and description+ratio respectively.

Logistic regression, it gave us 50.7% and 54.7% accuracy with description and description+ratio variables whereas this may further required validation with significant variables and remove unnecessary variables to get desire output.

**Appendix**

**setwd("E:/r direct/Web & Social Analytics/WSM Assignment")**

**SharkTank = read.csv("Shark+Tank+Companies.csv", stringsAsFactors=FALSE)**

**####Identifying the count of True and Flase Values**

**table(SharkTank$deal)**

**## FALSE TRUE**

**## 244 251**

**library(tm)**

**library(SnowballC)**

**library(wordcloud)**

**corpusSharkTank = Corpus(VectorSource(SharkTank$description))**

**wordcloud(corpusSharkTank,colors=rainbow(7),max.words=50)**

**## Converting into lower-case**

**corpusSharkTank = tm\_map(corpusSharkTank, tolower)**

**##Remiving Punctuations**

**corpusSharkTank = tm\_map(corpusSharkTank, removePunctuation)**

**## Removing stopwords**

**corpusSharkTank = tm\_map(corpusSharkTank, removeWords, c("the","and",stopwords("english")))**

**## Removing Extra Spaces**

**corpusSharkTank = tm\_map(corpusSharkTank,stripWhitespace)**

**# Stem document**

**corpusSharkTank = tm\_map(corpusSharkTank, stemDocument)**

**## Worl cloud after data cleaning process**

**wordcloud(corpusSharkTank,colors=rainbow(7),max.words=50)**

**frequenciesSharkTank = DocumentTermMatrix(corpusSharkTank)**

**### The DTM contains 495 rows and 3501 columns**

**###From 100 to 105 ideas picking the 505 to 515 words for visulization**

**inspect(frequenciesSharkTank[100:105,505:515])**

**## Checking the Sparsity**

**findFreqTerms(frequenciesSharkTank,highfreq = 20)**

**###Removing the words which are not present in atleast 5 descriptions/statements**

**sparseSharkTank = removeSparseTerms(frequenciesSharkTank, 0.995)**

**SharkTankSparse = as.data.frame(as.matrix(sparseSharkTank))**

**colnames(SharkTankSparse) = make.names(colnames(SharkTankSparse))**

**# Adding dependent variable**

**SharkTankSparse$DV = SharkTank$deal**

**SharkTankSparse$DV=as.factor(SharkTankSparse$DV)**

**# Building a CART model**

**library(rpart)**

**library(rpart.plot)**

**SharkTankCART = rpart(DV ~ ., data=SharkTankSparse, method="class")**

**prp(SharkTankCART,extra=2,box.palette = "auto")**

**library(rattle)**

**library(RColorBrewer)**

**fancyRpartPlot(SharkTankCART)**

**printcp(SharkTankCART)**

**ptree<- prune(SharkTankCART,cp= SharkTankCART$cptable[which.min**

**(SharkTankCART$cptable[,"xerror"]),"CP"])**

**fancyRpartPlot(ptree)**

**prp(SharkTankCART,extra=2,box.palette = "auto")**

**printcp(SharkTankCART)**

**plotcp(ptree)**

**### Accuracy**

**predictCARTRatio = predict(SharkTankCART, data=SharkTankSparse, type="class")**

**CART\_ratio <- table(SharkTank$deal, predictCARTRatio)**

**BaseAccuracyRatio = sum(diag(CART\_ratio))/sum(CART\_ratio)**

**##Building Random Forest Model**

**library(randomForest)**

**SharkTankRF=randomForest(DV~.,data=SharkTankSparse)**

**varImpPlot(SharkTankRF)**

**predictCARTRatio1 = predict(SharkTankRF, data=SharkTankSparse, type="class")**

**CART\_ratio1 <- table(SharkTank$deal, predictCARTRatio1)**

**BaseAccuracyRatio1 = sum(diag(CART\_ratio1))/sum(CART\_ratio1)**

**##BUilding Logistic Regression Model**

**SharkTankLogit=glm(DV~.,data=SharkTankSparse,family="binomial")**

**SharkTankPred=predict(SharkTankLogit,data=SharkTankSparse,type="response")**

**table(SharkTankSparse$DV,SharkTankPred>0.5)**

**(135+115)/495**

**predictCARTRatio2 = predict(SharkTankLogit, data=SharkTankSparse, type="response")**

**CART\_ratio2 <- table(SharkTank$deal, predictCARTRatio2)**

**BaseAccuracyRatio2 = sum(diag(CART\_ratio2))/sum(CART\_ratio2)**

**SharkTankSparse$ratio=(SharkTank$askedFor/SharkTank$valuation)**

**##CART Model After adding ratio Variable**

**SharkTankCART2 = rpart(DV ~ ., data=SharkTankSparse, method="class")**

**prp(SharkTankCART2,extra=2,box.palette = "auto")**

**library(rattle)**

**library(RColorBrewer)**

**fancyRpartPlot(SharkTankCART2)**

**printcp(SharkTankCART2)**

**ptree<- prune(SharkTankCART2,cp= SharkTankCART2$cptable[which.min**

**(SharkTankCART2$cptable[,"xerror"]),"CP"])**

**fancyRpartPlot(ptree)**

**plotcp(ptree)**

**##Randomforest Model After adding ratio Variable**

**SharkTankRF2=randomForest(DV~.,data=SharkTankSparse)**

**varImpPlot(SharkTankRF2)**

**##BUilding Logistic Regression Model after adding ratio variable**

**SharkTankLogit2=glm(DV~ratio,data=SharkTankSparse,family="binomial")**

**SharkTankLogit2Pred=predict(SharkTankLogit2,data=SharkTankSparse,type="response")**

**table(SharkTankSparse$DV,SharkTankLogit2Pred>0.5)**

**(125+146)/495**