ASSIGNMENT ON

WEB AND SOCIAL MEDIA ANALYTICS

Done & Submitted By

Akshay Subramanian

Table of Contents

OVERVIEW AND OBJECTIVE OF THE DATASET

The main dataset consists of 19 columns in which the column "**deal**" is considered as the dependent variable.

Dataset Name: Dataset.csv

The objective of the assignment is to:

1. Do the text mining for "**description**" column.
2. Run models like CART, Random Forest, and Logistic regression and determine the performance metrics.
3. Add a new variable called "**ratio**" and then again run the above three models.
4. See that if there is any impact in the model after adding the "**ratio**" variable.

WORKING ON THE DATASET

SETTING UP WORKING DIRECTORY

In order to access the dataset from the directory where it has been stored, the working directory must be set.

```
#setting the working directory

setwd("D:/BABI/BABI-17th Residency/WSMA/Assignment/Assignment_work")
getwd()
```

LOADING THE .csv FILE

After setting up the directory, the csv file has to be loaded for the initial analysis.

```
#loading the csv file

data <- read.csv("Dataset.csv")
View(data)
```

LOADING THE REQUIRED LIBRARY FILES

For the **text mining**, the following packages have to be installed and loaded.

1. tm
2. SnowballC
3. WordCloud

TEXT MINING

After installing and loading those packages, the text mining can be started.

Column in which the text mining has to be done: **description**.

For text mining, the following preliminary work has to be done:

1. The text in the "**description**" column has to be converted into **corpora** (singular "**corpus**": a large set of text).
2. Then, the corpora text has to be converted into lower case so that it will be easier for the text mining process.
3. The text has to be removed of the stop words like "**the**", "**a**", "**an**" and the **white space**.
4. Then, the words which are recurring in various forms have to be stemmed.

WORDCLOUD

From the above word cloud, we can see that the most frequently used words are:

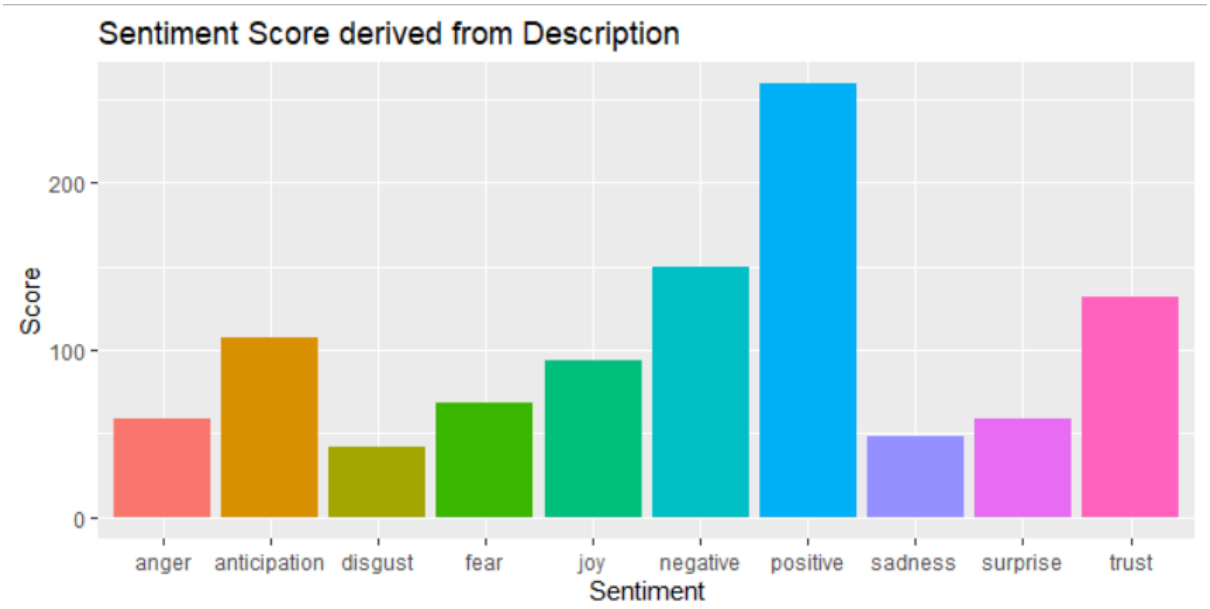
1. can
2. use
3. product
4. compani

SENTIMENT ANALYSIS

To know the dominant sentiments in the text mentioned in the dataset, we are doing sentiment analysis to the corpus we obtain by mining the "**description**" column. For that, the below packages are installed and loaded.

1. sentimentr
2. syuzhet

On running the sentiment analysis on the obtained corpus the following plot based on sentiment scores is obtained,



From the above graph, we can see that the top five sentiments of the customers are:

1. Positive
2. Negative
3. Trust
4. Anticipation
5. Fear

FORMATION OF DOCUMENT-TERM MATRIX (DTM)

After initial cleaning, we determine the frequencies of each and every word (stemmed ones) so that, we can eliminate the least occurring words. After removing the least occurring words, we are forming the **Document-Term Matrix (DTM)**.

```
frequenciesstc <- DocumentTermMatrix(corpusstc)
#getting the number of times of the occurrence of each word

sparsestc <- removeSparseTerms(frequenciesstc, 0.995) #removing the least occurring words
stcsparse <- as.data.frame(as.matrix(sparsestc)) #forming the DTM

View(stcsparse)

colnames(stcsparse) <- make.names(colnames(stcsparse))
```

After Forming the DTM, the work has to be done based on that and hence, the following columns from the parent dataset have to be copied to the new DTM and then make it as a new data.frame for modeling purposes.

1. deal
2. askedFor
3. exchangeForStake
4. valuation

After adding the columns, the data.frame is used for the modeling purposes.

```
#copying the existing columns in the dataset to the newly formed DTM

stcsparse$deal <- data$deal
stcsparse$askedFor <- data$askedFor
stcsparse$exchangeForStake <- data$exchangeForStake
stcsparse$valuation <- data$valuation

str(stcsparse$deal)
stcsparse$deal <- as.factor(stcsparse$deal) #converting the dependent variable into categorical variabl
```

We can see that, there are **244 FALSE** values and **251 TRUE** values in the "**deal**" column.

```
> summary(stcsparse$deal)
FALSE  TRUE
  244   251
>
```

EXPLAINING THE TREE SPLIT

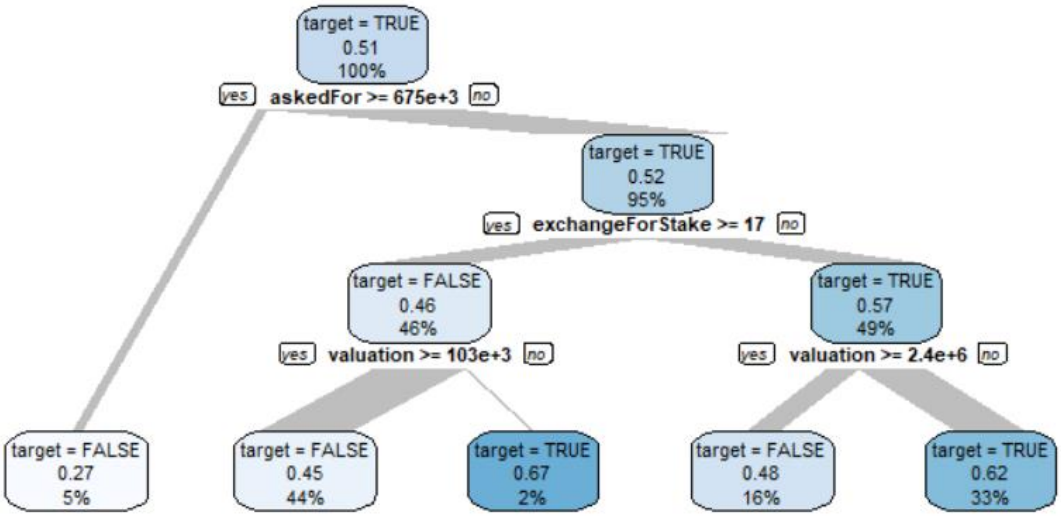
In the parent dataset, after running the **explain_tree** command with the target variable as "**deal**", we can see that the below tree has got generated. The other independent variables selected for the explanation of the tree split is:

1. askedFor
2. exchangeForStake
3. valuation

For that, the library files that are corresponding to the ML are installed and loaded. The packages are:

1. mlr
2. dplyr
3. explore

The TREE diagram



Findings

1. When the target (which is **deal** variable) = TRUE with the splitting criterion as the askedFor variable value is greater than or equal to 675000, we can see that 5% of the data have target as FALSE and the rest 95% of the data have target as TRUE.
2. When the exchangeForStake is greater than or equal to 17, 46% of the data have target as FALSE and 49% of them have the target as TRUE.
3. When the valuation is greater than or equal to 103000, 44% of them have target as FALSE and 2% of them have target as TRUE.
4. When the valuation is greater than or equal to 2400000, 16% of the records have target as FALSE and the remaining 33% of the records have target as TRUE.

MODELLING PART

SPLITTING OF DATA

Before we proceed with the modeling part, we can split the data in the ratio of **80:20**. After splitting, we will be proceeding with the below models.

1. CART model
2. Logistic Regression
3. Random Forest

For different models, the splitting of data is done separately. For the data partition, the packages "**caTools**", "**caret**" are installed and loaded.

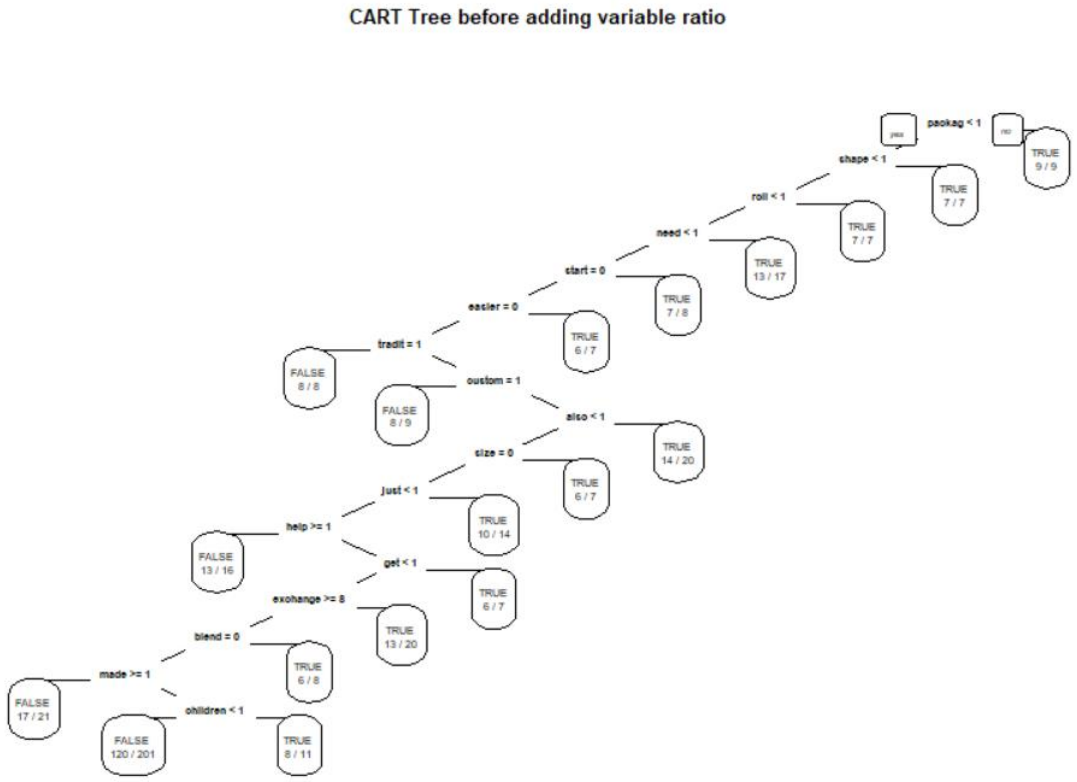
```
#splitting of data (80-20 ratio)

set.seed(123)

train.index <- createDataPartition(stcsparse$deal, p = .8, list = FALSE)
train <- stcsparse[train.index,]
test  <- stcsparse[-train.index,]
```

CART MODEL

For the CART model, the packages like "**rpart**" and "**rpart.plot**" have to be loaded. Then, the model is created with the training dataset "**train**". Also, the cart tree of the model has been saved in a pdf file for better viewing (*Refer Annexure 1 – Cart.pdf*).

```
library(rpart)
library(rpart.plot)

library(DataExplorer)
library(greybox)
library(rattle)

SharktankCart = rpart(deal ~ ., data=train, method="class")
rpart.plot(SharktankCart)

#printing the cart tree in a pdf document

pdf("Cart.pdf")
fancyRpartPlot(SharktankCart, palettes=c("Greys", "Oranges"),
               main = "CART model before adding variable ratio")
dev.off()

#CART Diagram
prp(SharktankCart, extra=2, main = "CART Tree before adding variable ratio")

summary(SharktankCart)
attributes(SharktankCart)

#confusion matrix

train$predict.class <- predict(SharktankCart, train, type="class")
train$predict.score <- predict(SharktankCart, train)

confmattrain = table(train[,c("deal","predict.class")])
confusionMatrix(confmattrain)

#predicting for the test dataset

test$predict.class <- predict(SharktankCart, test, type="class")
test$predict.score <- predict(SharktankCart, test)

confmattest = table(test[,c("deal","predict.class")])
confusionMatrix(confmattest)
```

CART tree



From the above diagram, we can see that the columns **package, shape** and **roll** are some of the key columns for the splitting of the CART tree. This CART diagram was obtained by running the command **prp(SharktankCart, extra=2)**.

Variable Importance

Variable importance					
packag	roll	shape	tradit	need	start
7	6	6	5	5	4
get	size	also	easier	just	children
4	4	4	4	4	3
custom	exchangeForStake	made	help	blend	your
3	3	3	3	3	2
convers	individu	sugar	readi	realli	whether
2	2	1	1	1	1
coconut	anyth	came	healthi	spice	young
1	1	1	1	1	1
coffe	can	compani	that	hill	plate
1	1	1	1	1	1
safe	smartphon	take	happi	partner	tone
1	1	1	1	1	1
guitar	clever	feet	glass	measur	pressur
1	1	1	1	1	1

From the above table, we can see that the columns **package**, **roll**, **shape**, **tradit**, **need** and **start** have a significant importance in splitting criteria for the CART tree. (*Refer Annexure 2 – Cart_excel.xlsx*)

CONFUSION MATRIX

After running the model, to evaluate the model, we first run the model in training dataset to check for the predict and actual values. To evaluate the performance of the model (based on accuracy), we run the same model on the test dataset. The accuracy obtained on the test dataset tells us how effective the model is.

After running the model in both the training and the testing datasets, the key metrics of the confusion matrix are tabulated below.

Metric	CART_train	CART_test
Accuracy	70.03%	54.08%
Sensitivity	65.10%	52.46%
Specificity	78.87%	56.76%

On looking into the above table, the accuracy obtained for the training dataset and the testing dataset is **70.03%** and **54.08%** respectively.

LOGISTIC REGRESSION

For the logistic regression, we are splitting the dataset in **80:20** ratios. We create the model with the 80% of the dataset and evaluate it in the 20% of the dataset. (*Refer Annexure 3 – LR_excel.xlsx*).

```
#Logistic Regression (80:20 split)

library(ROCR)
library(lmtest)
library(pscl)

set.seed(1234)
split <- createDataPartition(stcsparse$deal, p = .8, list = FALSE)
train_lt <- stcsparse[split,]
test_lt  <- stcsparse[-split,]

lg_model <- glm(deal ~ .,data = train_lt,family = binomial())

summary(lg_model)

pR2(lg_model)

#Confusion matrix of train data

pred<-predict(lg_model,train_lt,type='response')
mat_tab<-table(train_lt$deal, pred > 0.5)
mat_tab

lt_acc <- sum(diag(mat_tab))/sum(mat_tab)
lt_acc*100

#Prediction and Confusion matrix of test data

tdata <- predict(lg_model,test_lt, type="response")
t_confmat <- table(test_lt$deal, tdata > 0.5)
t_confmat
test_acc <- sum(diag(t_confmat))/sum(t_confmat)
test_acc*100
```

Here, the dependent variable is "**deal**" which is a function of the several other variables which are obtained by text mining of the "**Description**" column from the parent dataset, and then the other variables include "**askedFor**", "**exchangeForStake**" and "**valuation**".

From the datatype of the variables, we can see that the target variable (**deal**) is categorical and the other independent variables (the DTM variables and the other above mentioned variables) are numerical.

CONFUSION MATRIX

After creating the model, it is run in both the testing and training datasets in order to evaluate. The key metric (accuracy) obtained from the testing dataset tells us how effective the model is.

Metric	LR_train	LR_test
Accuracy	100%	51.02%

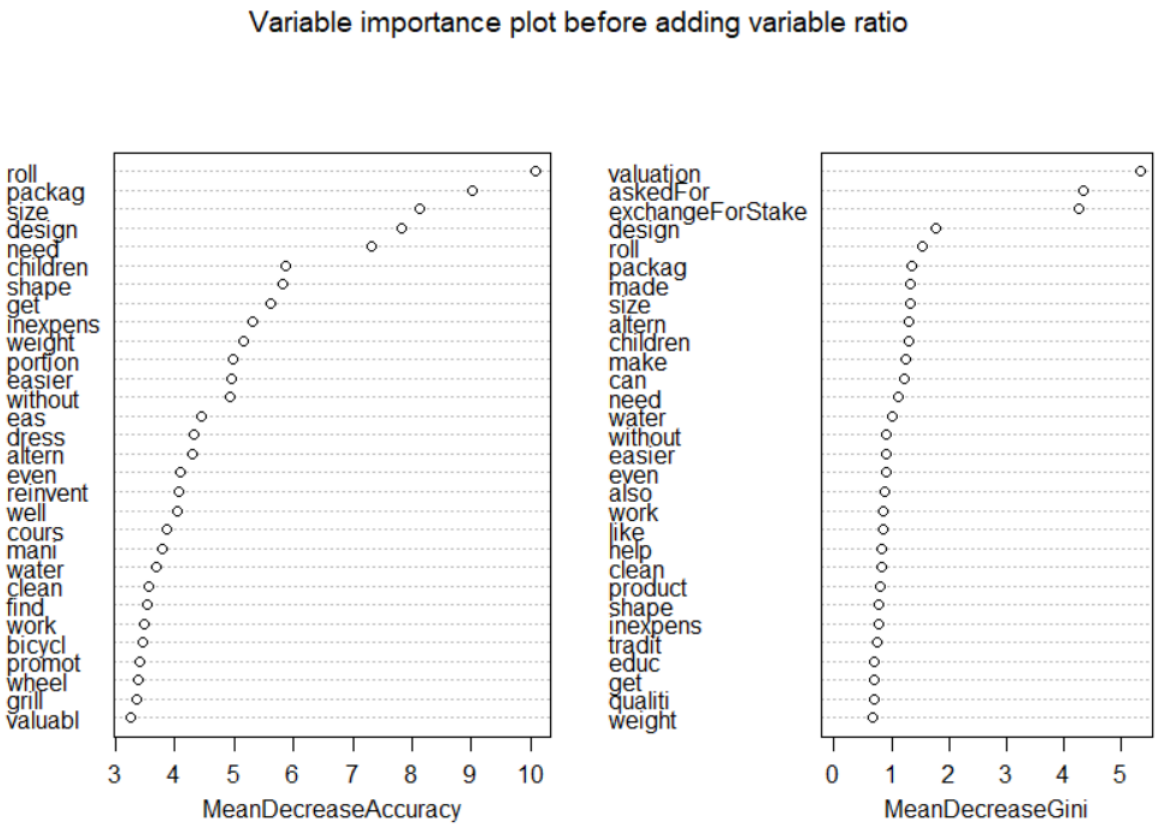
On looking into the above table, the accuracy obtained for the training dataset and the testing dataset is **100%** and **51.02%** respectively.

RANDOM FOREST

For running the random forest model, the package "**randomForest**" has to be installed and loaded. Also, for evaluating the model performance, the split of data is done in 80-20 ratio.

```
#Random Forest

#loading the required package for random forest

library(randomForest)

set.seed(1234)
split_rf <- createDataPartition(stcsparse$deal, p = .8, list = FALSE)
train_rf <- stcsparse[split_rf,]
test_rf  <- stcsparse[-split_rf,]

model_rf<-randomForest(deal~.,data=train_rf,importance=T)
model_rf

plot(model_rf, main = "Plot of random forest model before adding variable ratio")

varImpPlot(model_rf)

#confusion matrix

train_rf$predict.class <- predict(model_rf, train_rf, type="class")
train_rf$predict.score <- predict(model_rf, train_rf)

confmattrain_rf = table(train_rf[,c("deal","predict.class")])
confusionMatrix(confmattrain_rf)

#predicting for the test dataset

test_rf$predict.class <- predict(model_rf, test_rf, type="class")
test_rf$predict.score <- predict(model_rf, test_rf)

confmattest_rf = table(test_rf[,c("deal","predict.class")])
confusionMatrix(confmattest_rf)
```

After running the model, in order to analyze the importance of the importance of variables, the **variable importance plot** is made.



Based on the **MeanDecreaseGini** criteria, we can see that the top five important variables are:

1. valuation
2. askedFor
3. exchangeForStake
4. design
5. roll

CONFUSION MATRIX

After splitting the dataset in 80-20 manner, we build is built on the 80% of the dataset which is the trainig dataset and the model is evaluated on the 20% of the dataset which is testing dataset. The confusion matrix is created based on the predicted variable and the "**deal**" variable of the dataset and then the model is evaluated on the test dataset.

After running the model in both the training and the testing datasets, the key metrics of the confusion matrix are tabulated below.

Metric	RF_train	RF_test
Accuracy	99.24%	57.14%
Sensitivity	94.89%	55.36%
Specificity	100%	59.52%

On looking into the above table, the accuracy obtained for the training dataset and the testing dataset is **99.24%** and **57.14%** respectively.

ADDITION OF NEW VARIABLE

A new variable namely **ratio** is created which is obtained by the below formula:

Ratio = askedFor/valuation

```
#addition of new variable -- Ratio

dtmdata$ratio = dtmdata$askedFor/dtmdata$valuation

View(dtmdata)
```

After the addition of this variable, we again run the models to see if the performance of the models has increased or not. The models that are to be run again are:

1. CART model
2. Logistic Regression
3. Random Forest

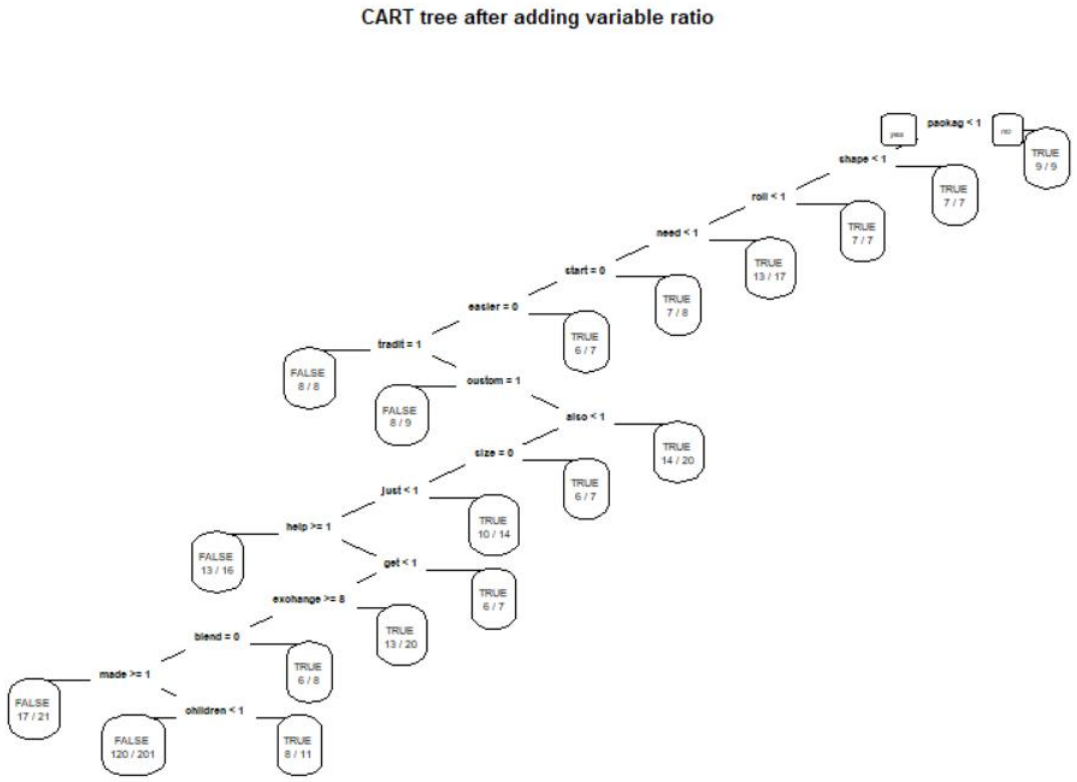
CART MODEL

Since the splitting for the model has been done before adding the "**ratio**" variable, we use the same splitting criteria so as to compare the accuracy of the models before and after the addition of "**ratio**" variable.

The model is run and a CART tree is generated. It is saved in pdf file for better viewing. (*Refer Annexure 4 – Cart_after.pdf*).

```
set.seed(1234)

train.index_aft <- createDataPartition(dtmdata$deal, p = .8, list = FALSE)
train_aft <- stcsparse[train.index_aft,]
test_aft  <- stcsparse[-train.index_aft,]

#CART model

SharktankCart_aft = rpart(deal ~ ., data=train_aft, method="class")
rpart.plot(SharktankCart_aft)

#printing the cart tree in a pdf document

pdf("Cart_after.pdf")
fancyRpartPlot(SharktankCart_aft, palettes=c("Greys", "Oranges"),
               main = "CART tree after adding variable ratio")
dev.off()

#CART Diagram
prp(SharktankCart_aft, extra=2, main = "CART tree after adding variable ratio")

summary(SharktankCart_aft)
attributes(SharktankCart_aft)

#confusion matrix

train_aft$predict.class <- predict(SharktankCart_aft, train_aft, type="class")
train_aft$predict.score <- predict(SharktankCart_aft, train_aft)

confmattrain_aft = table(train_aft[,c("deal","predict.class")])
confusionMatrix(confmattrain_aft)

#predicting for the test dataset

test_aft$predict.class <- predict(SharktankCart_aft, test_aft, type="class")
test_aft$predict.score <- predict(SharktankCart_aft, test_aft)

confmattest_aft = table(test_aft[,c("deal","predict.class")])
confusionMatrix(confmattest_aft)
```

CART Tree



From the above diagram, we can see that the columns **package, shape** and **roll** are some of the key columns for the splitting of the CART tree. This CART diagram was obtained by running the command **prp(SharktankCart_aft, extra=2)**.

Variable Importance

Variable importance					
packag	roll	shape	tradit	need	start
7	6	6	5	5	4
get	size	also	easier	just	children
4	4	4	4	4	3
custom	exchangeForStake	made	help	blend	your
3	3	3	3	3	2
convers	individu	sugar	readi	realli	whether
2	2	1	1	1	1
coconut	anyth	came	healthi	spice	young
1	1	1	1	1	1
coffe	can	compani	that	hill	plate
1	1	1	1	1	1
safe	smartphon	take	happi	partner	tone
1	1	1	1	1	1
guitar	clever	feet	glass	measur	pressur
1	1	1	1	1	1

From the above table, we can see that the columns **packag**, **roll**, **shape**, **tradit**, **need** and **start** have a significant importance in splitting criteria for the CART tree. (*Refer Annexure 2 – Cart_excel.xlsx*).

CONFUSION MATRIX

After creating the model in the training dataset, the model is evaluated in the test dataset, the values for the key metrics like **accuracy**, **sensitivity** and **specificity** is tabulated.

Metric	CART_train	CART_test
Accuracy	70.03%	54.08%
Sensitivity	65.10%	52.46%
Specificity	78.87%	56.76%

On looking into the above table, we can see that the accuracy of the training and testing datasets is **70.03%** and **54.08%** respectively.

LOGISTIC REGRESSION

For the logistic regression, we are splitting the dataset in **80:20** ratio. We create the model with the 70% of the dataset and evaluate it in the 30% of the dataset. (*Refer Annexure 3 – LR_excel.xlsx*)

```
#Logistic Regression (80:20 split)

set.seed(1234)
split_af <- createDataPartition(stcsparse$deal, p = .8, list = FALSE)
train_lt.af <- stcsparse[split_af,]
test_lt.af  <- stcsparse[-split_af,]

lg_model.aft <- glm(deal ~ .,data = train_lt.af,family = binomial())

summary(lg_model.aft)

pR2(lg_model.aft)

#Confusion matrix of train data

pred_aft<-predict(lg_model.aft,train_lt.af,type='response')
mat_tab.af<-table(train_lt.af$deal, pred_aft > 0.5)
mat_tab.af

lt_acc_aft <- sum(diag(mat_tab.af))/sum(mat_tab.af)
lt_acc_aft*100

#Prediction and Confusion matrix of test data

tdata_aft <- predict(lg_model.aft,test_lt.af, type="response")
t_confmat.aft <- table(test_lt.af$deal, tdata_aft > 0.5)
t_confmat.aft
test_acc.aft <- sum(diag(t_confmat.aft))/sum(t_confmat.aft)
test_acc.aft*100
```

Here, the dependent variable is "**deal**" which is a function of the several other variables which are obtained by text mining of the "**Description**" column from the parent dataset, and then the other variables include "**askedFor**", "**exchangeForStake**", "**valuation**" and "**ratio**".

From the datatype of the variables, we can see that the target variable (**deal**) is categorical and the other independent variables (the DTM variables and the other above mentioned variables) are numerical.

CONFUSION MATRIX

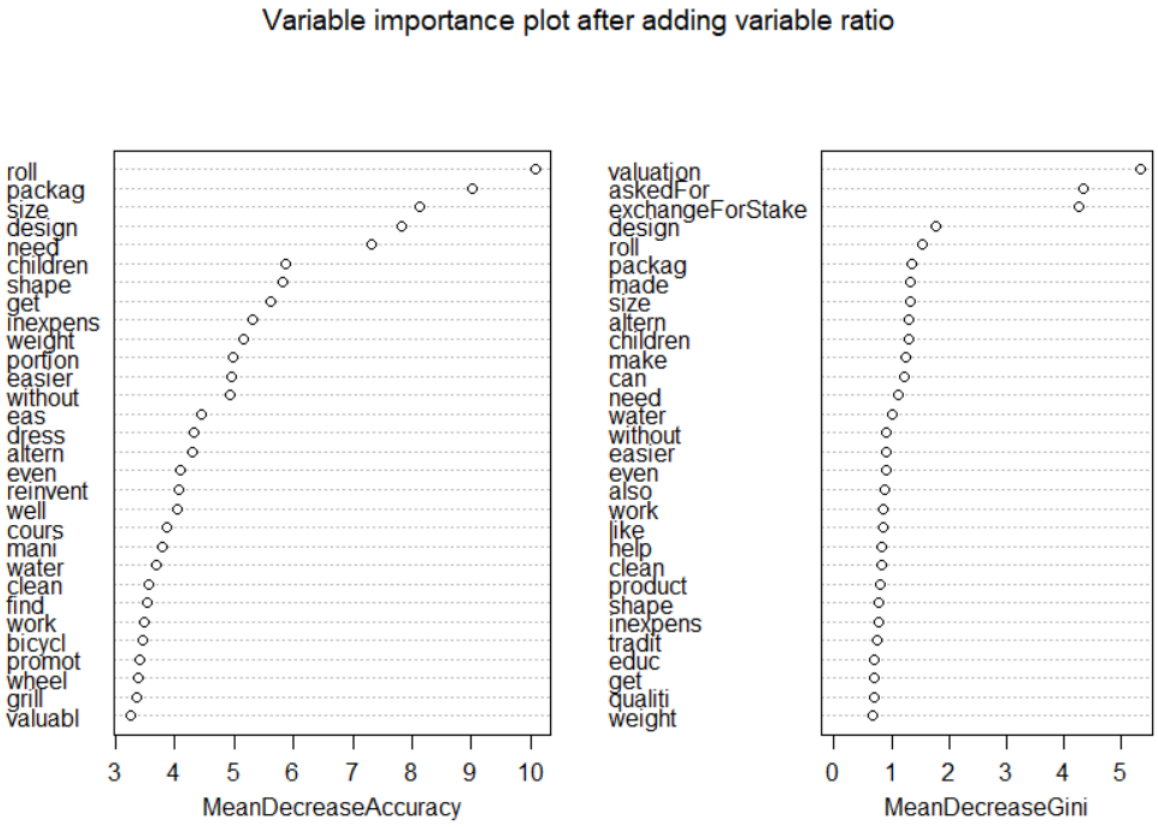
After creating the model, it is run in both the testing and training datasets in order to evaluate. The key metric (accuracy) obtained from the testing dataset tells us how effective the model is.

Metric	LR_train	LR_test
Accuracy	100%	51.02%

On looking into the above table, the accuracy obtained for the training dataset and the testing dataset is **100%** and **51.02%** respectively.

RANDOM FOREST

For running the random forest model, the package "**randomForest**" has to be installed and loaded. Also, for evaluating the model performance, the split of data is done in 80-20 ratio. After creating the model, the importance of each variable is created as a plot.



Based on the **MeanDecreaseGini** criteria, we can see that the top five important variables are:

1. valuation
2. askedFor
3. exchangeForStake
4. design
5. roll

CONFUSION MATRIX

After the creation of model, it is run on the testing dataset in order to evaluate the performance of the model. The confusion matrix is created based on the predicted variable and the "**deal**" variable of the dataset and then the model is evaluated on the test dataset. The values for the key metrics like **accuracy**, **sensitivity** and **specificity** is tabulated.

Metric	RF_train	RF_test
Accuracy	99.75%	60.20%
Sensitivity	99.49%	58.49%
Specificity	100%	62.22%

On looking into the above table, we can see that the accuracy of the training and testing datasets is **99.75%** and **60.20%** respectively.

SUMMARY OF THE DATASET

After running the

1. CART
2. Logistic Regression and
3. Random Forest

models, we have tabulated the accuracies of all the three models (only the testing dataset accuracy values).

The below table include both the accuracy values. i.e. before and after the addition of "**ratio**" variable. The accuracy obtained from the test datasets are being tabulated below.

	Accuracy	
Model Name	Before adding "**ratio**" variable	After adding "**ratio**" variable
CART	54.08%	54.08%
Logistic Regression	51.02%	51.02%
Random Forest	57.14%	60.20%

On looking into the accuracy values, we can say that:

1. Overall, **Random Forest model is better** when compared to CART and Logistic Regression.
2. The accuracy of the CART and Logistic Regression models remain unchanged even after introducing a new variable.
3. From the above point, we can infer that the variable "**ratio**" doesn't have any significant impact in the modeling part for the CART and Logistic regression.
4. But for the Random Forest model, the accuracy has increased significantly from **57.14%** to **60.20%**.
5. From this, we can assume that the variable "**ratio**" does have a significant impact in the modeling of Random Forest model.

ANNEXURES

Annexure 1

Cart.pdf

Annexure 2

CART_excel.xlsx

Annexure 3

LR_excel.xlsx

Annexure 4

Cart_after.pdf

R code

WSMA_Assignment.R