R CODE FOR WSMA

```
library(tm)
library(SnowballC)
Create corpus
corpus = Corpus(VectorSource(Dataset$description))
# Convert to lower-case
corpus = tm_map(corpus, tolower)
corpus = tm_map(corpus, removePunctuation)
# Word cloud before removing stopwords
library(wordcloud)
wordcloud(corpus,colors=rainbow(7),max.words=100)
corpus = tm_map(corpus, removeWords, c("the", "and",
stopwords("english")))
corpus = tm_map(corpus, stripWhitespace)
corpus = tm_map(corpus, stemDocument)
wordcloud(corpus,colors=rainbow(7),max.words=100)
#Document term matrix
frequencies = DocumentTermMatrix(corpus)
frequencies
sparse = removeSparseTerms(frequencies, 0.995)
# Convert to a data frame
Dataset =  as.data.frame(as.matrix(Dataset))
colnames(Dataset) = make.names(colnames(Dataset))
# Add dependent variable
Dataset$deal = Dataset$deal
#Get no of deals
table(Dataset$deal)
# Build CART model
library(rpart)
library(rpart.plot)
Dataset = rpart(deal ~ ., data=Dataset, method="class")
#CART Diagram
prp(Dataset, extra=2)
# Evaluate the performance of the CART model
predictCART = predict(Dataset, data=Dataset, type="class")
CART_initial <- table(Dataset$deal, predictCART)
# Baseline accuracy
```

```
BaseAccuracyCart = sum(diag(CART_initial))/sum(CART_initial)
#Random Forest Model
library(randomForest)
set.seed(123)
DatasetRF = randomForest(deal~ ., data = Dataset)
# Make predictions:
predictRF = predict(DatasetRF, data=Dataset)
# Evaluate the performance of the Random Forest
randomForestInitial <- table(Dataset$deal, predictRF>= 0.5)
# Baseline accuracy
BaseAccuracyRF = sum(diag(randomForestInitial))/sum(randomForestInitial)
#variable importance as measured by a Random Forest
varImpPlot(DatabaseRF,main='Variable Importance Plot: Shark Tank',type=2)
# Logistic Regression model
set.seed(123)
Datasetlogistic = glm(deal~., data = Dataset)
# Make predictions:
predictLogistic = predict(Dataset, data=Dataset)
# Evaluate the performance of the Random Forest
LogisticInitial <- table(Dataset$deal, predictLogistic> 0.5)
# Baseline accuracy
BaseAccuracyLogistic = sum(diag(LogisticInitial))/sum(LogisticInitial)
# Add ratio variable into Dataset
Dataset$ratio = Dataset$askedFor/Dataset$valuation
########CART Model###########
Datasetratio = rpart(deal ~ ., data=Dataset, method="class")
#CART Diagram
prp(Datasetratio, extra=2)
# Evaluate the performance of the CART model
predictCARTratio = predict(DatasetRatio, data=Dataset, type="class")
CART_ratio <- table(Dataset$deal, predictCARTratio)
# Baseline accuracy
BaseAccuracyRatio = sum(diag(CART_ratio))/sum(CART_ratio)
#########Random Forrest############
#Random Forrest Model
DatasetRFRatio = randomForest(deal ~ ., data=Dataset)
#Make predictions:
predictRFRatio = prediDatasetRFRatio, data=Dataset)
# Evaluate the performance of the Random Forest
RandomForestRatio <- table(Dataset$deal, predictRFRatio>= 0.5)
```

```
# Baseline accuracy
BaseAccuracyRFRatio = sum(diag(RandomForestRatio))/sum(RandomForestRatio)
#variable importance as measured by a Random Forest
varImpPlotDatasetRFRatio,main='Variable Importance Plot: Shark Tank with Ratio',type=2)
#########Logistic Regression##########
DatasetRatio = glm(deal~., data = Dataset)
# Make predictions:
predictLogisticRatio = predict(DatasetRatio, data=Dataset)
# Evaluate the performance of the Random Forest
LogisticRatio <- table(Dataset$deal, predictLogisticRatio>= 0.5)
# Baseline accuracy
BaseAccuracyLogisticRatio = sum(diag(LogisticRatio))/sum(LogisticRatio)
```