Shark Tank Project

Agnes Munee

2/25/2022

Introduction: Executive Summary

The target variable or the variable to be predicted is deal This predicts whether the entrepreneur will get a deal or not. Since we are going to be predicting between 2 categories (TRUE or FALSE), we will use Two-Class Classification. The machine learning models that can be used for this class of project are:

1. Decision Trees
2. Random Forest
3. Logistic Regression
4. K Nearest Neighbors

Methods and Analysis

Install the packages to be used:

```
library(tidyverse)
library(caret)
```

data Cleaning

There are variables that need to be transfomed into factors; These are:

deal - This is the target variable

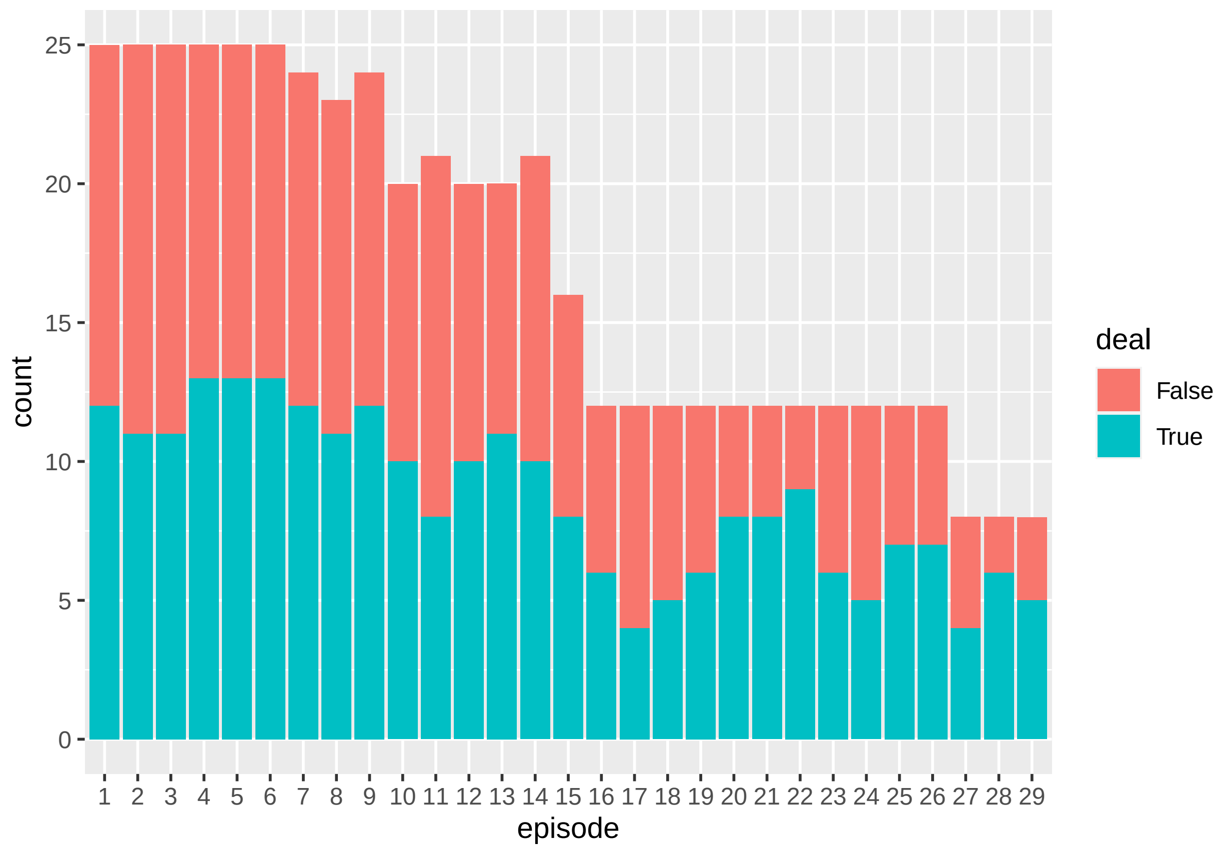
description - This variable describes the nature of the idea/business presented.
Each business idea has its own unique description.No much analyses can be done on this variable.

$epidode - This is the episode where each business idea was presented. This project has compiled project for 29 episodes.

```
project<-read.csv("Shark Tank.csv")
length(unique(project$episode))
```

```
## [1] 29
```

```
project%>%mutate(deal=factor(deal),episode=factor(episode))%>%group_by(deal)%>%ggplot(aes(episode,fill=
```




```
project%>%group_by(episode)%>%summarize(n=n(),n_false=round(sum(deal=="False")/n*100,0),n_true=round(sum
```

```
##    episode  n n_false n_true
## 1       27  8      50     50
## 2       28  8      25     75
## 3       29  8      38     62
## 4       16 12      50     50
## 5       17 12      67     33
## 6       18 12      58     42
## 7       19 12      50     50
## 8       20 12      33     67
## 9       21 12      33     67
## 10      22 12      25     75
## 11      23 12      50     50
## 12      24 12      58     42
## 13      25 12      42     58
## 14      26 12      42     58
## 15      15 16      50     50
## 16      10 20      50     50
## 17      12 20      50     50
## 18      13 20      45     55
## 19      11 21      62     38
## 20      14 21      52     48
## 21       8 23      52     48
## 22       7 24      50     50
```

```
## 23        9 24      50      50
## 24        1 25      52      48
## 25        2 25      56      44
## 26        3 25      56      44
## 27        4 25      48      52
## 28        5 25      48      52
## 29        6 25      48      52
```

```
n_episode<-project%>%group_by(episode)%>%summarize(n=n(),n_false=round(sum(deal=="False")/n*100,0),n_tru
cor(n_episode$episode,n_episode$n_false)
```

```
## [1] -0.4584958
```

```
cor(n_episode$episode,n_episode$n_true)
```

```
## [1] 0.4584958
```

From the above analysis, there seems to be a negative correlation between the number of episode and not getting a deal, i.e, the probability of getting a deal diminishes in later episodes. The first episodes in a series have a higher chance of giving an entrepreneur a deal.

category

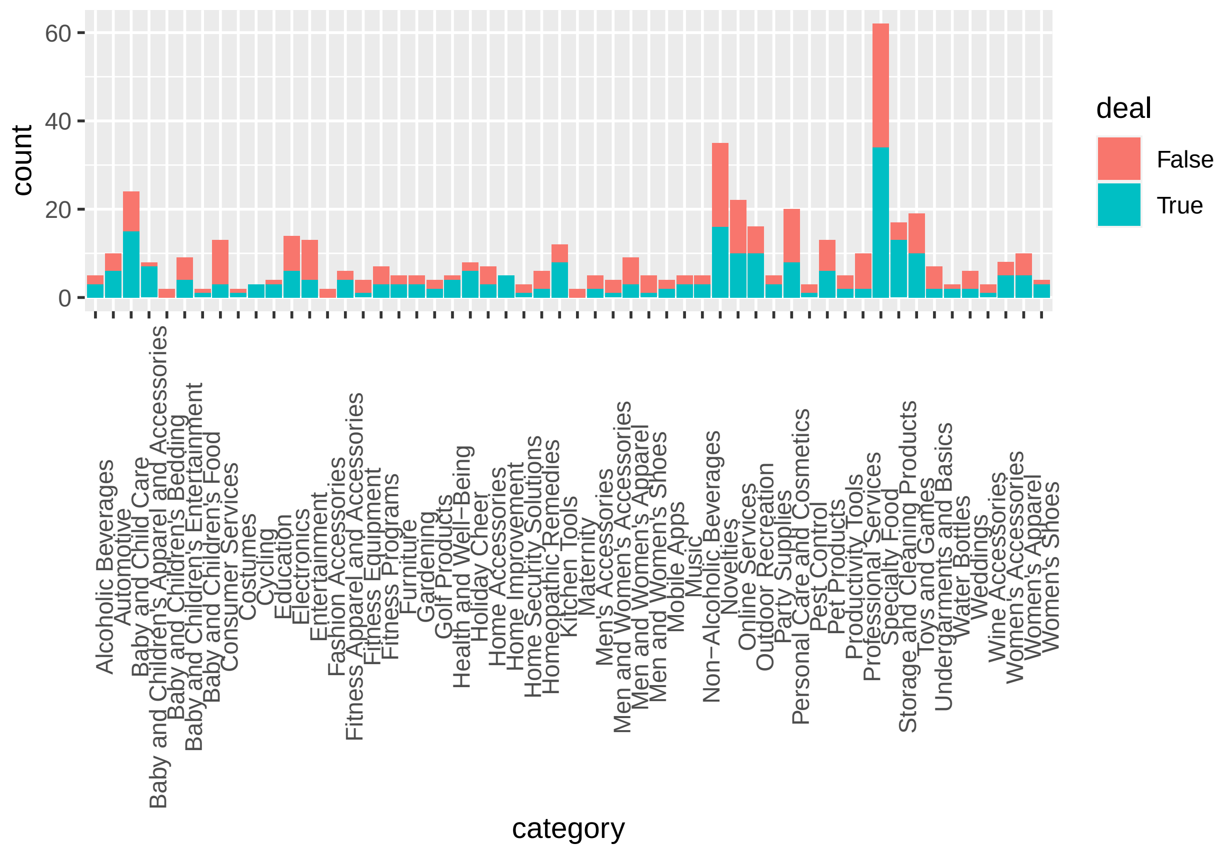
```
length(unique(project$category))
```

```
## [1] 54
```

There are 54 unique categories of each project presented to the sharks.

```
      n_category<- project%>%group_by(category)%>%summarize(n=n(),n_false=round(sum(deal=="False")/n*10

project%>%mutate(deal=factor(deal),category=factor(category))%>%group_by(deal)%>%ggplot(aes(category,fi
```



There is a strong relation between the category and whether an entrepreneur gets the deal or not.

askedFor This is the amount that the entreprener asks for from the sharks. This is a continuous variable.

exchangeForStake

This is the percentage of the business that the investor asked for in exchange for the

valuation This is the value of the business.

season
This is the season that the episodes

```
unique(project$season)
```

```
## [1] 1 2 3 4 5 6
```

This project contains project for the 6 seasons that the show aired.

episode.season This is the episode and the season that the show aired.

```
unique(project$episode.season)
```

```
##   [1] "1-1"  "1-2"  "1-3"  "1-4"  "1-5"  "1-6"  "1-7"  "1-8"  "1-9"  "1-10"
##  [11] "1-11" "1-12" "1-13" "1-14" "2-1"  "2-2"  "2-3"  "2-4"  "2-5"  "2-6"
##  [21] "2-7"  "2-8"  "2-9"  "3-1"  "3-2"  "3-3"  "3-4"  "3-5"  "3-6"  "3-7"
##  [31] "3-8"  "3-9"  "3-10" "3-11" "3-12" "3-13" "3-14" "3-15" "4-1"  "4-2"
##  [41] "4-3"  "4-4"  "4-5"  "4-6"  "4-7"  "4-8"  "4-9"  "4-10" "4-11" "4-12"
```

```
##  [51] "4-13" "4-14" "4-15" "4-16" "4-17" "4-18" "4-19" "4-20" "4-21" "4-22"
##  [61] "4-23" "4-24" "4-25" "4-26" "5-1"  "5-2"  "5-3"  "5-4"  "5-5"  "5-6"
##  [71] "5-7"  "5-8"  "5-9"  "5-10" "5-11" "5-12" "5-13" "5-14" "5-15" "5-16"
##  [81] "5-17" "5-18" "5-19" "5-20" "5-21" "5-22" "5-23" "5-24" "5-25" "5-26"
##  [91] "5-27" "5-28" "5-29" "6-1"  "6-2"  "6-3"  "6-4"  "6-5"  "6-6"  "6-7"
## [101] "6-8"  "6-9"  "6-10" "6-11" "6-12" "6-13" "6-14" "6-15" "6-16" "6-17"
## [111] "6-18" "6-19" "6-20" "6-21" "6-22" "6-23" "6-24" "6-25" "6-26" "6-27"
## [121] "6-28" "6-29"
```

There are 122 unique combinations of episode and season

Multiple.Entreprenuers This indicates whether more than one investor made an offer to the entrepreneur (i,e TRUE) , or only one investor made an offer, (i.e FALSE)

Shark1, shark2, shark3, shark4 and shark5 - are the investors that are present to the entrepreneur's business pitch

```
unique(project$shark1)
```

```
## [1] "Barbara Corcoran" "Lori Greiner"
```

```
unique(project$shark2)
```

```
## [1] "Robert Herjavec"  "Barbara Corcoran" "Kevin O'Leary"    "Steve Tisch"
```

```
unique(project$shark3)
```

```
## [1] "Kevin O'Leary"   "Robert Herjavec" "Daymond John"
```

```
unique(project$shark4)
```

```
## [1] "Daymond John"   "Jeff Foxworthy" "Kevin O'Leary"  "Mark Cuban"
```

```
unique(project$shark5)
```

```
## [1] "Kevin Harrington"  "Mark Cuban"        "Daymond John"
## [4] "John Paul DeJoria" "Nick Woodman"
```

We can check the number of offers given by each shark.

```
table(project$shark1,project$deal)
```

```
##
##                     False True
##   Barbara Corcoran   117  103
##   Lori Greiner       127  148
```

```
table(project$shark2,project$deal)
```

```
##
##                     False True
##   Barbara Corcoran    46   58
##   Kevin O'Leary        3    9
##   Robert Herjavec    193  182
##   Steve Tisch          2    2
```

```
table(project$shark3,project$deal)
```

```
##
##                    False True
##   Daymond John        3    9
##   Kevin O'Leary     195  184
##   Robert Herjavec    46   58
```

```
table(project$shark4,project$deal)
```

```
##
##                  False True
##   Daymond John     191  180
##   Jeff Foxworthy     4    4
##   Kevin O'Leary     46   58
##   Mark Cuban         3    9
```

```
table(project$shark5,project$deal)
```

```
##
##                      False True
##   Daymond John           4    4
##   John Paul DeJoria      1    3
##   Kevin Harrington      44   36
##   Mark Cuban           193  202
##   Nick Woodman           2    6
```

Next, we will transform project. Some variables will be transformed into factors -deal -episode -category -season -shark1 -shark2 shark3 -shark4 -shark5 -episode.season -multiple.Entreprenuers

```
project_trn<-project%>%mutate(deal=factor(deal),episode=factor(episode),category=factor(category),seaso
```

```
str(project_trn)
```

```
## 'data.frame':    495 obs. of  19 variables:
##  $ deal                  : Factor w/ 2 levels "False","True": 1 2 2 1 1 2 1 1 1 2 ...
##  $ description           : chr  "Bluetooth device implant for your ear." "Retail and wholesale pie fa
##  $ episode               : Factor w/ 29 levels "1","2","3","4",..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ category              : Factor w/ 54 levels "Alcoholic Beverages",..: 36 45 3 8 8 45 31 43 2 11 .
##  $ entrepreneurs         : chr  "Darrin Johnson" "Tod Wilson" "Tiffany Krumins" "Nick Friedman, Omar
##  $ location              : chr  "St. Paul, MN" "Somerset, NJ" "Atlanta, GA" "Tampa, FL" ...
##  $ website               : chr  "" "http://whybake.com/" "http://www.avatheelephant.com/" "http://co
##  $ askedFor              : int  1000000 460000 50000 250000 1200000 500000 200000 100000 500000 2500
```

```
##  $ exchangeForStake      : int  15 10 15 25 10 15 20 20 10 10 ...
##  $ valuation             : int  6666667 4600000 333333 1000000 12000000 3333333 1000000 500000 50000
##  $ season                : Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ shark1                : Factor w/ 2 levels "Barbara Corcoran",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ shark2                : Factor w/ 4 levels "Barbara Corcoran",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ shark3                : Factor w/ 3 levels "Daymond John",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ shark4                : Factor w/ 4 levels "Daymond John",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ shark5                : Factor w/ 5 levels "Daymond John",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ title                 : chr  "Ionic Ear" "Mr. Tod's Pie Factory" "Ava the Elephant" "College Foxe
##  $ episode.season        : Factor w/ 122 levels "1-1","1-10","1-11",..: 1 1 1 1 1 7 7 7 7 7 ...
##  $ Multiple.Entreprenuers: Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
```

Modeling Approach

split the project into train,test and validation sets.

```
set.seed(1)
index_test<-createDataPartition(project_trn$deal,times = 1,p=0.5,list = FALSE)
train<-project_trn[-index_test,]
test<-project_trn[index_test,]
dim(train)
```

```
## [1] 247  19
```

```
dim(test)
```

```
## [1] 248  19
```

We will start our modelling with our first Machine Learning Algorithm

1. Classification (Decision) Trees - Rpart

```
set.seed(1)
train_rpart<-train(deal ~episode+askedFor+exchangeForStake+valuation +season +shark1+shark2+shark3+shar
y_hat_rpart<-predict(train_rpart,test)
rpart_Accuracy<-confusionMatrix(y_hat_rpart,test$deal)$overall[["Accuracy"]]
rpart_Accuracy
```

```
## [1] 0.4637097
```

```
Results<-data.frame(Model="Decision Trees",Accuracy=rpart_Accuracy)
Results
```

```
##            Model  Accuracy
## 1 Decision Trees 0.4637097
```

This Accuracy is very low, (below 0.5) and thus we have to continue looking for a better model.

2.Random Forest

Random forest will address the shortcoings of the above decison tree, and thus giving a higher Accuracy.

```
library(caret)
set.seed(1)
train_rf <- randomForest(deal ~episode+askedFor+exchangeForStake+valuation +season +shark1+shark2+shark3
deal_hat<-predict(train_rf,test)
rf_Accuracy<-confusionMatrix(deal_hat,test$deal)$overall["Accuracy"]
```

```
Results<-rbind(Results,data.frame(Model="RandomForest",Accuracy=rf_Accuracy))
Results
```

```
##                     Model  Accuracy
## 1          Decision Trees 0.4637097
## Accuracy     RandomForest 0.4395161
```

The Accuracy is lower than that of the Decison Tree, and thus we will try cross validation.

Random Forest with crossvalidation

```
set.seed(1)
train_rf_cv <- train(deal ~episode+askedFor+exchangeForStake+valuation +season +shark1+shark2+shark3+sha
      method = "Rborist",
      tuneGrid = data.frame(predFixed = 2, minNode = c(3, 50)),
      data = train)
y_hat_rf_cv<-predict(train_rf_cv, test)
rf_cv_Accuracy<-confusionMatrix(y_hat_rf_cv,test$deal)$overall["Accuracy"]
```

```
Results<-rbind(Results,data.frame(Model="RandomForest with Cross Validation",Accuracy=rf_cv_Accuracy))
Results
```

```
##                                          Model  Accuracy
## 1                               Decision Trees 0.4637097
## Accuracy                          RandomForest 0.4395161
## Accuracy1 RandomForest with Cross Validation 0.4475806
```

The Random Forest after cross validation has improved our Accuracy.

3.Logistic Regression

```
set.seed(1)
train_glm<-train(deal ~episode+askedFor+exchangeForStake+valuation +season +shark1+shark2+shark3+shark4+
y_hat_glm<-predict(train_glm,test)
glm_Accuracy<-confusionMatrix(y_hat_glm,test$deal)$overall[["Accuracy"]]
```

```
Results<-rbind(Results,data.frame(Model="Logistic Regression",Accuracy=glm_Accuracy))
Results
```

```
##                                        Model  Accuracy
## 1                             Decision Trees 0.4637097
## Accuracy                        RandomForest 0.4395161
## Accuracy1 RandomForest with Cross Validation 0.4475806
## 11                       Logistic Regression 0.3951613
```

The glm model gives us a lower Accuracy.

4. K Nearest Neighbours

```
set.seed(1)
train_knn <- train(deal ~episode+askedFor+exchangeForStake+valuation +season +shark1+shark2+shark3+shar
                   data = train,
                   tuneGrid = data.frame(k = seq(9, 71, 2)))

control <- trainControl(method = "cv", number = 10, p = .9)

train_knn_cv <- train(deal ~episode+askedFor+exchangeForStake+valuation +season +shark1+shark2+shark3+sl
                      data= train,
                      tuneGrid = data.frame(k = seq(9, 71, 2)),
                      trControl = control)
knn_Accuracy<-confusionMatrix(predict(train_knn_cv, test, type = "raw"),
               test$deal)$overall["Accuracy"]
```

```
Results<-rbind(Results,data.frame(Model="KNN",Accuracy=knn_Accuracy))
Results
```

```
##                                        Model  Accuracy
## 1                             Decision Trees 0.4637097
## Accuracy                        RandomForest 0.4395161
## Accuracy1 RandomForest with Cross Validation 0.4475806
## 11                       Logistic Regression 0.3951613
## Accuracy2                                KNN 0.4959677
```

Conclusion

RandomForest with Cross Validation gives the highest Accuracy