INTERMEDIATE ANALYTICS

ALY6015, FALL 2019

FINAL PROJECT REPORT

SUBMITTED BY: ANUPREETA MISHRA, SHIVANI ADSAR, KEYUR SHAH

NUID: 001050752 ,001399374, 001089242

CRN: 71933

SUBMITTED TO: PROF. LI, TENGLONG

DATE: 12/10/2019

Introduction

We use the training set of the Kaggle competition "Titanic: Machine Learning from Disaster" (https://www.kaggle.com/c/titanic/data). Here we will try to make prediction on survivability of passenger using Box-Plot, Lasso & Regression, Random Forest, K-means and Decision tree. This project aims at using the practical and theoretical knowledge to use R and its functions on the "Titanic" dataset.

Analysis

Questions, Methods and their Interpretations

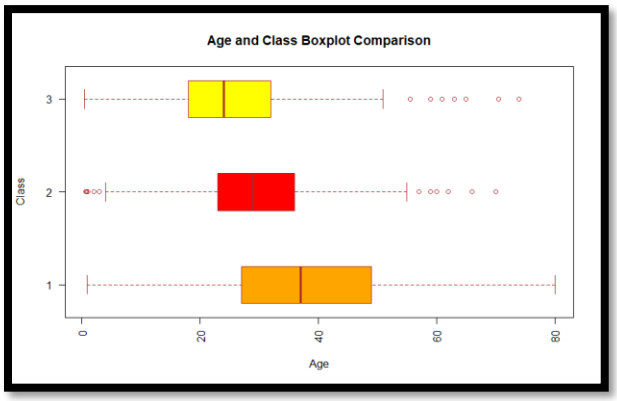
a) **Question posed**: If the person's gender, age or status affected the survivability of the person on the ship

- Comparison between the Age and Class variables using Box Plot
  Using the function, boxplot() as follows:

```
boxplot(TitanicTrainingData$Age ~ TitanicTrainingData$Pclass,
        main = "Age and Class Boxplot Comparison",
        at = c(1,3,5),
        names = c("1", "2", "3"),
        las = 2,
        col = c("orange","red","yellow"),
        border = "brown",
        horizontal = TRUE,
        notch = FALSE
        , xlab = "Age", ylab = "Class"
)
```

Fig.1: R code for Boxplot

- **Output and Interpretations of the boxplot() function,**



Observations: As seen, the median value of the passengers in Class 1 is higher than that of passengers in Class 2 and Class 3. It can be interpreted that the passengers in Class 3 had the youngest group of people. As per the plot, the Class 2 and Class 3 has the outliers.

Fig.2: Boxplot for Age and Class

- Next, we had performed on regularization methods like Ridge and LASSO Regression Methods for predicting the best fitting models of "Titanic" data set and also learning the parameters of residual standard errors to minimize the prediction error.
  •First , we load training dataset of "Titanic" and load library (glmnet) and convert it in Model matrix and transform the other variable 'y' into vector.
  Code:

  TitanicTrainingData <- read.csv("C:/Users/shahk/Downloads/train.csv")

```
View(TitanicTrainingData)
x <- model.matrix(TitanicTrainingData$Survived ~ TitanicTrainingData$PassengerId
+TitanicTrainingData$Pclass +TitanicTrainingData$Name, TitanicTrainingData)[,-1]
y <- TitanicTrainingData$Survived
```

- Next, we used lambda as tuning parameters and estimating the risk of underfitting and overfitting in ridge regression model. Lambda.min.ration and nlambda used for fitting.

  Code:

```
par(mfrow = c(1, 2))
fit_ridge_lambda = glmnet(x, y, alpha = 0,nlambda=100, lambda.min.ratio=0.0001)
plot(fit_ridge_lambda)
set.seed(1)
fit_cv <- cv.glmnet(x, y, alpha=0, nlambda=100, lambda.min.ratio=0.0001)
plot(fit_cv)
best.fit_lambda_ <- fit_cv$lambda.min
best.fit_lambda_
```

  Output: The optimal model accounted for 16% of variance in training data

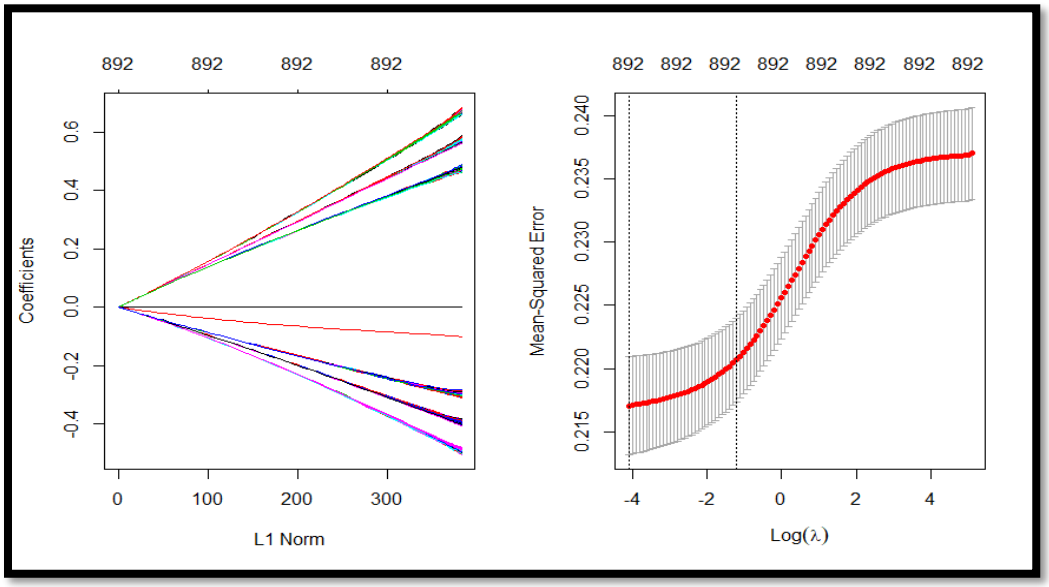


Fig.3: LASSO Regression

```
> best.fit_lambda_
[1] 0.01646099
```

- Next, we will implement the ridge in training and test dataset. So, we will split the data.

  Code:

```
row_number_hitt <- sample(x=1:nrow(TitanicTrainingData),
size=0.9*nrow(TitanicTrainingData))
train_hit = TitanicTrainingData[row_number_hitt,]
test_hit = TitanicTrainingData[-row_number_hitt,]
```

- Now, we will use ridge regression model with best fitting parameter and lambda estimates from training data.
  Code:
  x_test_hit <- model.matrix(test_hit$Survived ~test_hit$PassengerId + test_hit$Pclass +test_hit$Name, test_hit)[,-1]
  y_test_hit <- test_hit$Pclass
  y_predicted_hit <- predict(fit_ridge_lambda, s = best.fit_lambda_, newx = x_test_hit,type = "response")
  y_test_hit
  head(y_predicted_hit,4)
  Output:

```
> y_test_hit
 [1] 3 1 1 3 3 2 3 3 3 3 3 1 3 3 1 2 3 2 1 3 3 3 2 3 3 1 1 3 3 3 1 1 3 1 3 1 3 3 2 1 3 3 3 3 3 2 3 3
[49] 2 3 3 1 3 1 3 1 3 3 3 3 1 2 1 3 3 1 1 1 2 3 3 3 1 3 3 3 3 1 2 2 3 3 3 3 3 1 1 2 2 3
> head(y_predicted_hit,4)
           1
6  0.01042693
28 0.01698143
32 0.98435459
48 0.97768458
```

Fig.4: R code for prediction output

- Next, we will predict the accuracy error for our model.
  Code:
  sst_hit <- sum((mean(y_test_hit) - y_test_hit)^2)
  sse_hit <- sum((y_predicted_hit - y_test_hit)^2)
  rsq_hit<- 1 - sse_hit / sst_hit
  Output:

```
> rsq_hit
[1] -5.758793
> sse_hit
[1] 457.2699
> sst_hit
[1] 67.65556
```

- Next ,we will implement the lasso by changing the alpha parameter as ‘1’ in glmnet().
  Code:
  library(glmnet)
  par(mfrow = c(1, 2))
  fit_LASSO_lambda = glmnet(x, y, alpha = 1,nlambda=100, lambda.min.ratio=0.0001)
  plot(fit_LASSO_lambda,xvar = "lambda")
  set.seed(1)
  fit_lasso_cv <- cv.glmnet(x, y, alpha=1, nlambda=100, lambda.min.ratio=0.0001)
  plot(fit_lasso_cv)
  best.lasso <- fit_lasso_cv$lambda.min
  best.lasso

Result: The optimal model accounted for 2% of variance in training data

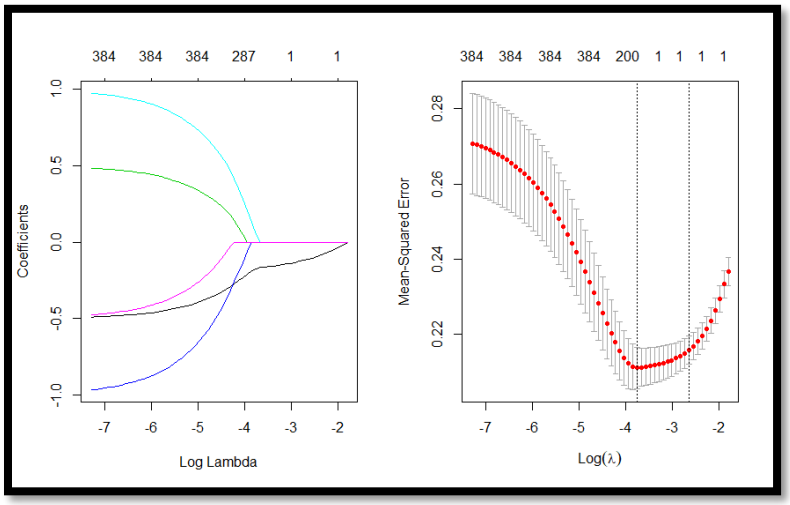


Fig.5: Big lasso

> best.lasso
[1] 0.02333302

- Now, we will use lasso regression model with best fitting parameter and lambda estimates from training data.

  Code:

  x_test_hit_lasso <- model.matrix(test_hit$Survived ~ test_hit$PassengerId +test_hit$Pclass +test_hit$Name, test_hit)[,-1]
  y_test_hit_lasso <- test_hit$Survived
  y_predicted_hit_lasso <- predict(fit_LASSO_lambda, s = best.lasso, newx = x_test_hit_lasso)
  head(y_predicted_hit_lasso,4)
  head(y_test_hit_lasso,4)

  Result:

```
> head(y_predicted_hit_lasso,4)
           1
6  0.2562237
28 0.6069716
32 0.6069716
48 0.3039084
> head(y_test_hit_lasso,4)
[1] 0 0 1 1
>
```

- Next we implement the accuracy and residual error parameters for models

  Code

  sst_hit_lasso <- sum((mean(y_test_hit_lasso) - y_test_hit_lasso)^2)
  sse_hit_lasso <- sum((y_predicted_hit_lasso - y_test_hit_lasso)^2)
  rsq_hit_lasso<- 1 - sse_hit_lasso / sst_hit_lasso

  Result: R squared value is greater than 1 , The model is simple .

```
> sst_hit_lasso
[1] 21.15556
> sse_hit_lasso
[1] 17.1675
> rsq_hit_lasso
[1] 0.1885112
>
```

b) **Question posed**: If the passengers of the upper class have better survival rate than its population

- Performing hypothesis for Z-test analysis
- Using the hypothesis as:
  Ho: There is no difference in the chances of survival of upper and lower class
  Ha: There are better chances of survival for upper class people

```
> #function for z test
> z.test2 = function(a, b, n){
+   sample_mean = mean(a)
+   pop_mean = mean(b)
+   c = nrow(n)
+   var_b = var(b)
+   zeta = (sample_mean - pop_mean) / (sqrt(var_b/c))
+   return(zeta)
+ }
> #call function
> z.test2(new_data$Survived, TitanicTrainingData$Survived, new_data)
[1] 7.423828
```

- **Output and Interpretations of the z.test() function**:
  It can be seen that the Z-value is 7.4238. Using this interpretation, we would reject the null hypothesis. Hence, there are better chances of survival for the upper class passengers.

K-MEANS CLUSTERING

- Clustering is a technique used in data analysis which involves finding homogenous subgroups within the data on the basis of Euclidean distance between the clusters.
- We have imported the “Titanic” data into a variable.
- We have then created the subset of the data and plotted the data.

```
> dat=titanic[,c(3,8)]
> plot(dat1,main="favourable responses",pch=20,cex=2)
```

Fig.6: R Code for plotting

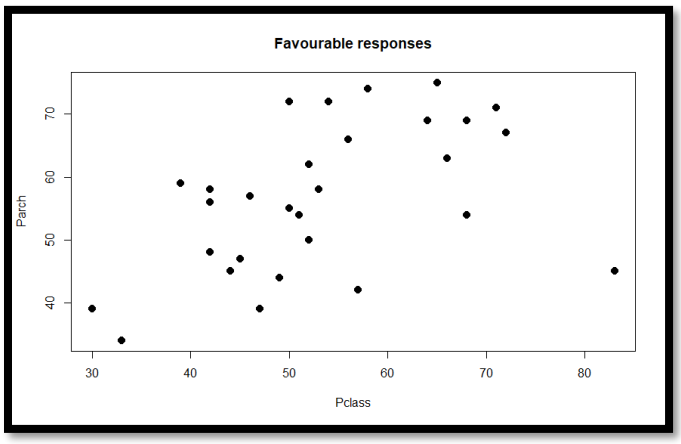
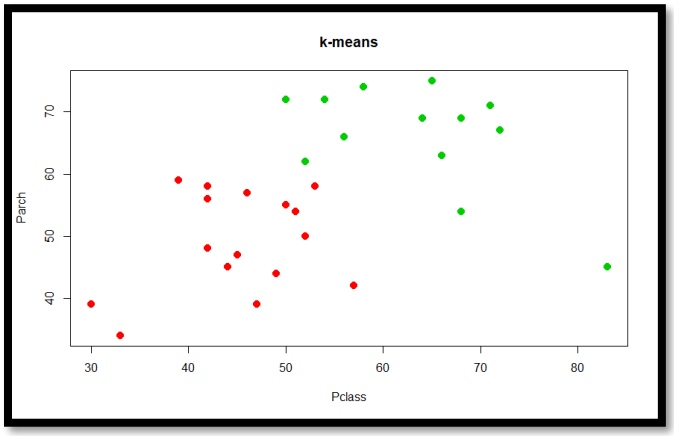


Fig.7: Favourable responses Parch vs. Pclass

- We have performed k-means using 2 clusters on the dataset

```
> plot(dat1,col=(km1$cluster +1), main="k-means",pch=20,cex=2,xlab="Pclass",ylab="Parch")
> km1$cluster
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
 1  1  2  1  2  1  1  1  2  1  1  1  1  2  2  2  2  2  1  2  1  2  1  1  1  2  2  1  2  1
```

Fig.8: R Code for plotting k-means and its output



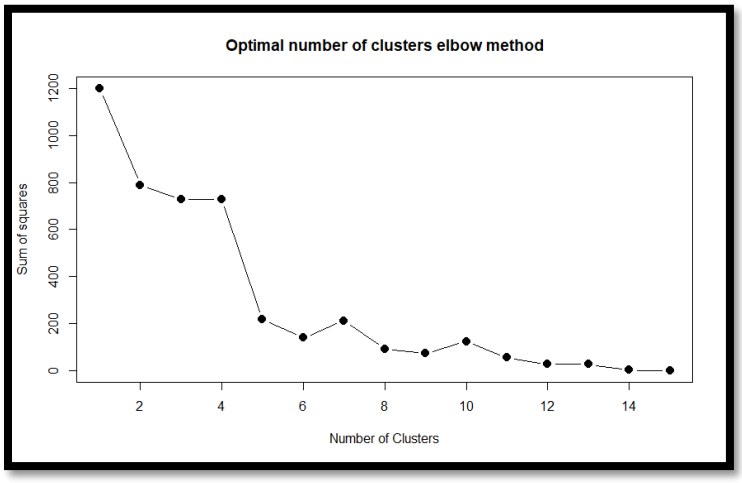
Observations: We have plotted the clusters using k-means clustering algorithm.

Fig.9: K-means of Parch vs. Pclass

- Deciding the optimal number of clusters in the data

```
mydata<-dat
wss<-(nrow(mydata)-1)*sum(apply(mydata,2,var))
for(i in 2:15)wss[i]<-sum(kmeans(na.omit(mydata),centers=i)$withinss)
plot(1:15,wss,type="b",xlab="Number of Clusters",
     ylab="Sum of squares",
     main="Optimal number of clusters elbow method",
     pch=20,cex=2)
```

Fig.10: R Code using Elbow Method



Observations: We have used the Elbow method to decide the optimal number of clusters. This method gives a precise value of "K" clusters on the basis of the sum of squares an centroids. As seen, the optimal number of clusters for the data is 6.

Fig.11: Plot representing Optimal number of clusters using elbow method

- Performing k-means on the dataset based on the optimal number of clusters

```
> km3 = kmeans(dat, 6, nstart=100)
> km3
K-means clustering with 6 clusters of sizes 98, 381, 134, 15, 100, 163

Cluster means:
    Pclass    Parch
1 3.000000 1.438776
2 3.000000 0.000000
3 2.000000 0.000000
4 2.733333 4.133333
5 1.480000 1.370000
6 1.000000 0.000000
```

Fig.12: R Code showing k-means and clustering

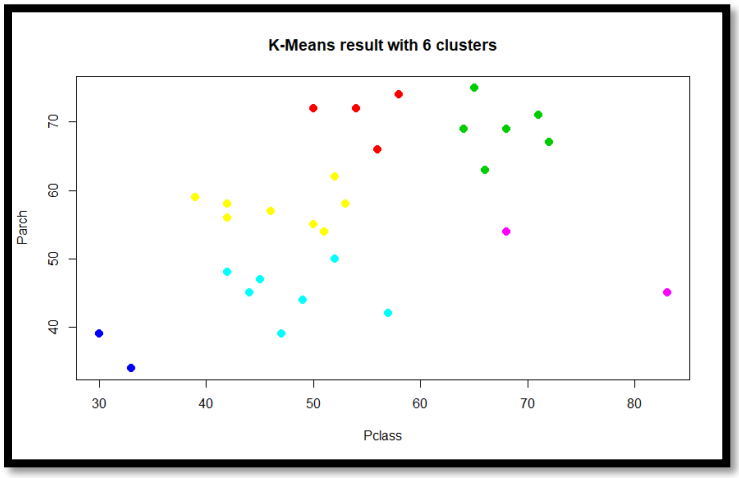


Fig.13: K-Means clustering

Observations: As seen, the clusters have been formed between Parch and Pclass.

DECISION TREES USING TITANIC DATASET

- A decision tree is a chart which is used in statistics to represent the predictions based on the decisions.
- Using the "Titanic" dataset, we are predicting the chances of survival of males and females.
- We are splitting the dataset into Train and Test data

```
allset= split.data(Titanic, p = 0.7)
titanic_trainingdataset = allset$trainingdataset
titanic_testingdataset = allset$testingdataset
install.packages('party')
require('party')
train.ctree = ctree(Survived ~ Pclass + Sex + Age + SibSp + Fare + Parch + Embarked, data=titanic_trainingdataset)
train.ctree
plot(train.ctree, main="Applying cTree on Titanic Dataset", tp_args = list(fill = c("green","red")))
```

Fig.14: R Code showing the Test and Train splitting

- Training and create decision tree using the ctree() function

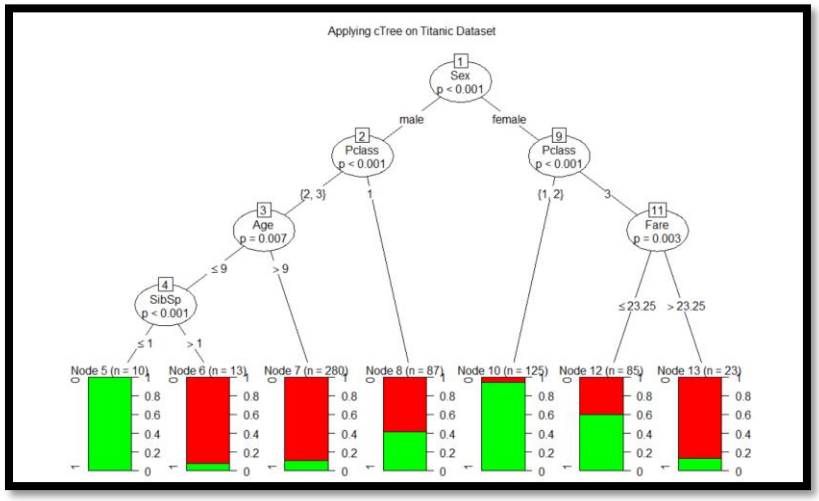
```
> require('party')
> train.ctree = ctree(Survived ~ Pclass + Sex + Age + SibSp + Fare + Parch + Embarked, data=titanic_trainingdataset)
> train.ctree

        Conditional inference tree with 6 terminal nodes

Response:  Survived
Inputs:  Pclass, Sex, Age, SibSp, Fare, Parch, Embarked
Number of observations:  623

1) Sex == {female}; criterion = 1, statistic = 180.207
  2) Pclass == {3}; criterion = 1, statistic = 59.081
    3)*  weights = 92
  2) Pclass == {1, 2}
    4)*  weights = 111
1) Sex == {male}
  5) Pclass == {1}; criterion = 1, statistic = 21.732
    6) Age <= 38; criterion = 0.994, statistic = 11.188
      7)*  weights = 58
    6) Age > 38
      8)*  weights = 38
  5) Pclass == {2, 3}
    9) Age <= 3; criterion = 1, statistic = 15.978
```

Fig. 15: R Code showing the ctree() function



Observations: The decision tree shows that the node 9 depicts a clear picture about the survival rates of the passengers. It can be seen that the female passengers in class 1 and class 2 survived the most.

Fig.16: Decision Tree showing the predictions

c) **Question posed**: If there is a significant relationship between Sex, Pclass and the Survival

- We will use the Chi-square test to perform the test.
- Chi-Square is used to test the difference between the expected and observed frequencies.
  H0: Both the categorical variables are independent
  Ha: Both the variables are dependent

- **Output and Interpretations of the z.test() function**:

Using the function, chisq.test(TitanicDS$Survived, TitanicDS$Sex)

```
        Pearson's Chi-squared test with Yates' continuity correction

data:  TitanicDS$Survived and TitanicDS$Sex
X-squared = 260.72, df = 1, p-value < 2.2e-16
```

Probability lesser than 0.05, shows that the relationship between the variables is at 95% confidence. It can be concluded that, the P-value is less than 0.05, Sex and Pclass are very significant variables for performing this test. Also, since our, P-value is lesser than 0.05 ($p<0.05$), we can reject our null hypothesis. This indicates that both the variables are dependent.

Decision Tree Prediction :

Using Rattle library and using the function rpart,we get the decision tree below. Okay, now the decisions that have been found to go a lot deeper than what we saw last time when we looked for them manually. Decisions have been found for the SipSp variable, as well as the port of embarkation one that we didn't even look at. And on the male side, the kids younger than 6 years old have a better chance of survival, even if there weren't too many aboard. That resonates with the famous naval law we mentioned earlier.Now,we make a prediction of passengers' suvival in test dataframe from this tree.We point the function to the model's fit object, which contains all of the decisions we see above, and tell it to work its magic on the test dataframe. The rpart package automatically caps the depth that the tree grows by using a metric called complexity which stops the resulting model from overfitting the data.

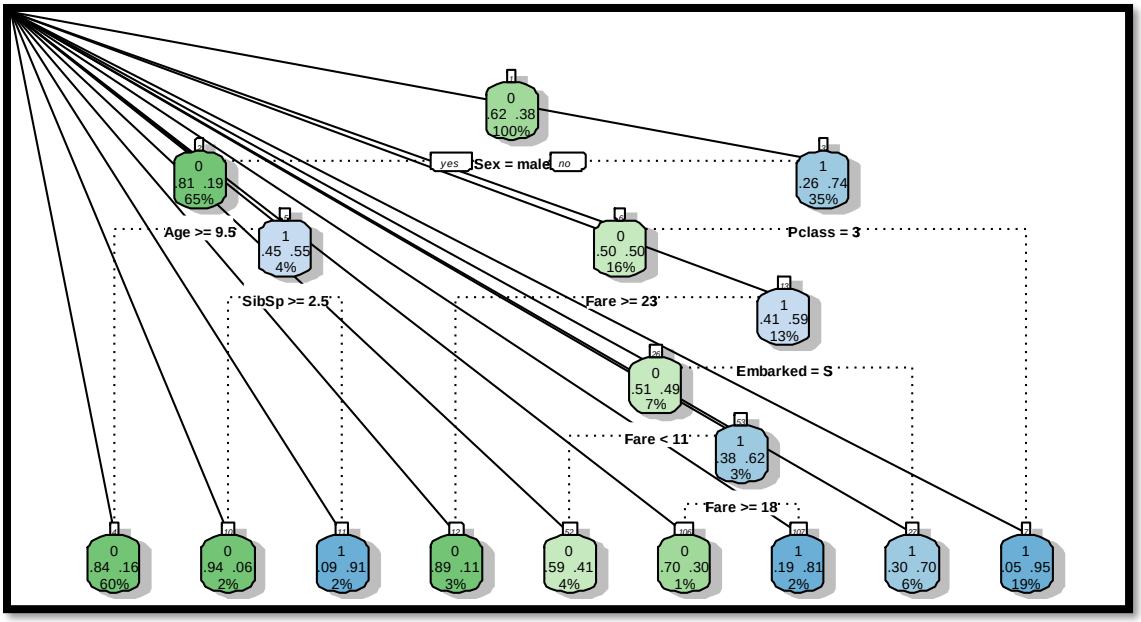


Fig.17: Decision Tree

Next, we use the predict function to see the accuracy of predictions.

```
> prop.table(table(train$Survived))

        0         1
0.6161616 0.3838384
> #0         1
> #0.6161616 0.3838384
> prop.table(table(Prediction))
Prediction
        0         1
0.6746411 0.3253589
```

We see that 32% passengers ,from the test dataset,should be surviving which is pretty close to the 38% survival rate in the train dataset. Thus, we have predicted the survival chances with

A: Combining both test and train dataset :

Random forest uses Bagging method to create multiple random sample with replacement from dataset. Hence the best course of action is to combine data as it would give us the most random sample possible in this way and there will be no data lost.

B: Feature Generation :

Until now we have used “Pclass”, a proxy for socio-economic status (SES) in our analysis, we also take care of the NA values in this section. Now, we break down Passenger name into additional meaningful variables ,which feed predictions , passenger title is contained within the passenger name variable and we can use surname to represent families. Then we find the family size, to see if it affected the survivability of the passenger, and then plot it . From the graph at the right, we can see that people with families of size between 2 to 4 had higher chances to survive. Hence, we divide our data into 3 categories based on family size, i.e. single, small, large.

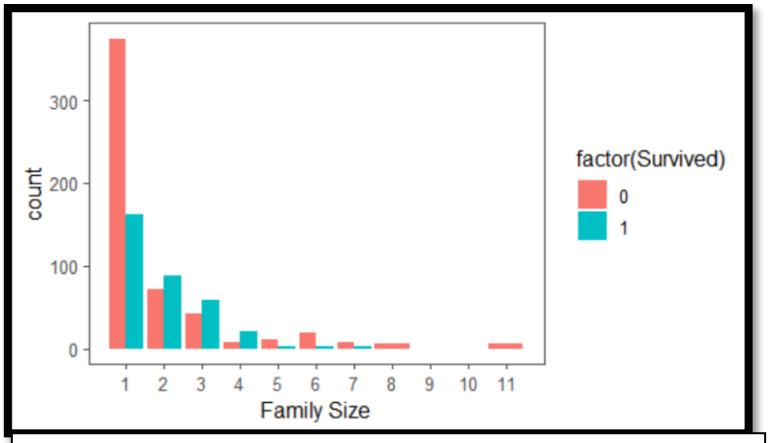


Fig 19: Count of people that survived VS Family Size

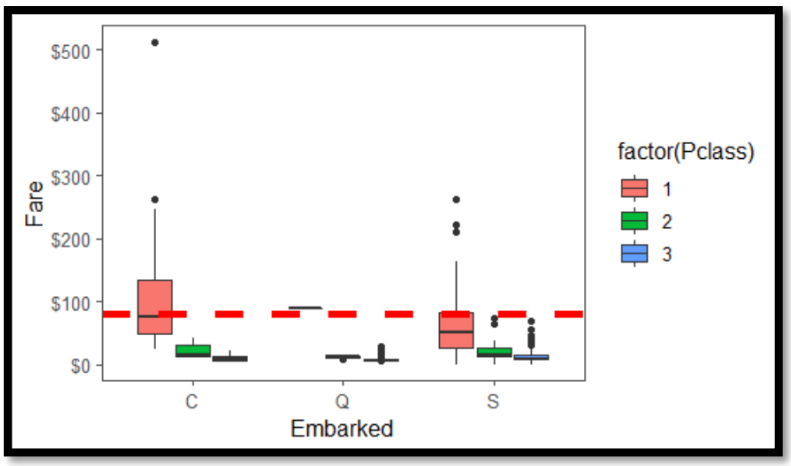


Fig 20: Fare Vs Embarked Passangers Box Plot

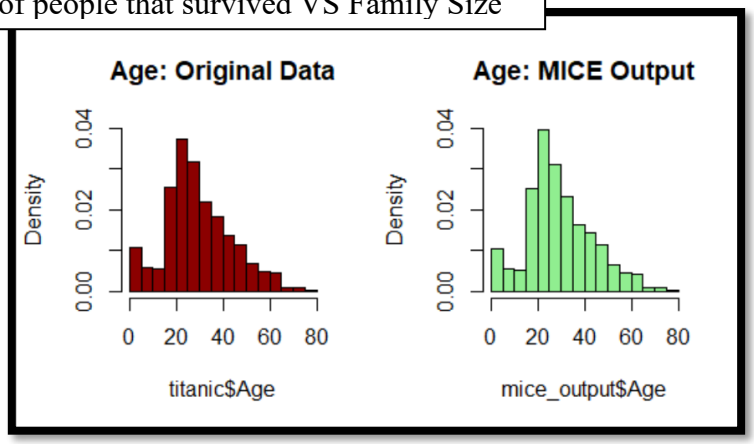


Fig 21: Validating corrections of NA values

Next, we use ggplot2 to visualize embarkment, passenger class, & median fare. From the graph below,note that median fare for a first class passenger departing from Charbourg ('C') coincides nicely with the $80 paid by our embarkment-deficient passengers. Based on this, we replace NA values with 'C' for Embarked. We use the MICE model to replace NA values in age , since the graphs look almost the same. Then we check if sex was a significant predictor.

C: Prediction using Random Forest:

We divide the data into train and test using:

```
> train <- titanic[1:891,]
> test <- titanic[892:1309,]
```

Then we create a model using randomForest

```
>
> titanic_model <- randomForest(Survived ~ Pclass + Sex + Age + SibSp + Par
ch +
+                                 Fare + Embarked + title +
+                                 fsizeD + Child + Mother,
+                               data = train)
```

Then we check the model error using plot(titanic_model, ylim=c(0,0.36)). The black line shows the overall error rate which falls below 20%. The red and green lines show the error rate for 'died' and 'survived' respectively. We can see that right now we're much more successful predicting death than we are survival.

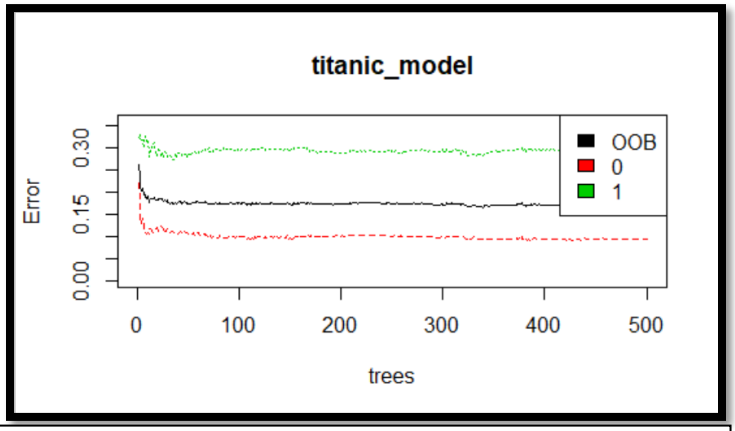


Fig 22: Checking for Error

D: Variable importance and Final Prediction:

We try to find the importance of a passengers attributes using the importance formula. And plotting its graph. As we can see that the title of the person was the most important factor to survivability.

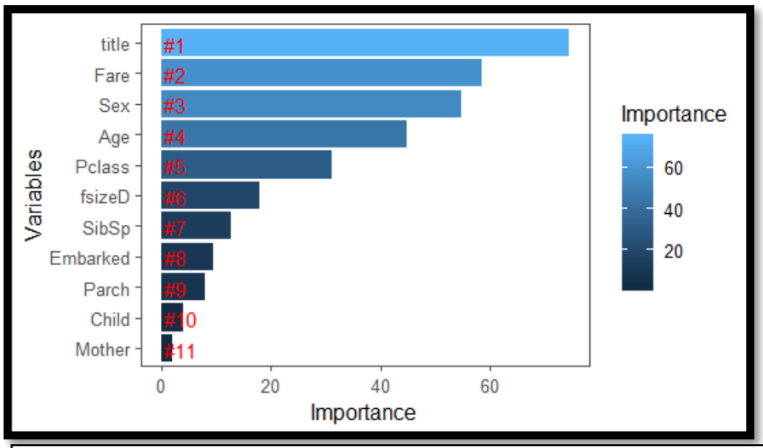


Fig 23: Ranking of the columns by importance to survival

```
> prop.table(table(train$Survived))

        0         1
0.6161616 0.3838384
> prop.table(table(prediction))
prediction
        0         1
0.6339713 0.3660287
```

We see that 36% passengers ,from the test dataset,should be surviving which is pretty close to the 38% survival rate in the train dataset.

Thus, we have predicted the survival chances with fair enough accuracy using Random Forest.

Conclusion and Interpretations

We can see that; the Class 3 passengers have the youngest people based on the Box-plot analysis. Moreover, based on the Z-test hypothesis, there are better chances of survival for the upper class passengers. It can be noted from the chi-squared test, that there exists a dependency on the parameters ie. Sex and Pclass.People with higher class mothers,families with less than 4 people have higher survival rate.

References

- Dalal, S. (n.d.). Predicting passenger survival using classification. Retrieved from https://www.polyglotdeveloper.com/r-projects/2016-09-01-Predicting-passenger-survival-using-classification/#reading-the-titanic-dataset-from-a-csv-file.
- Dabbura, I. (2019, September 3). K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks. Retrieved from https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a.
- https://www.kaggle.com/pradeeptripathi/prediction-of-titanic-survival-using-r
- https://rpubs.com/Swastik96/188313