FUNDAMENTALS OF MACHINE LEARNING

FINAL PROJECT

PROJECT GOAL:

The objective of the Titanic Train Dataset is to determine who survived the tragedy from test results. The outcome variable labels from the train set are used to build and assess the machine learning algorithm. For this I used the classification techniques in machine learning like Logistic Regression and Decision trees.

INTRODUCTION:

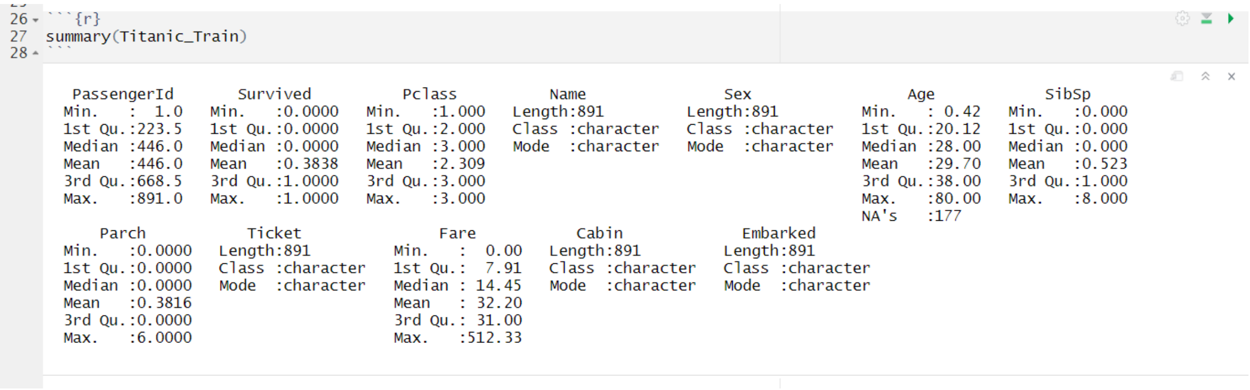
722 persons escaped the Titanic tragedy, which resulted in 1502 deaths. Therefore, the baseline survival rate is 32.46%. In the competition, we are provided two datasets to use for prediction: one with labels (survived or not) and the other with the same features (attributes) but no labels. According to the theory, passenger attributes contain data that can anticipate the outcome.

Since there are 418 rows in the test set, we are only required to predict the outcomes of 418 passengers, or more specifically, the survival of around 136 individuals. Taking note of characteristics where most people survived.

Data Information:

- **Survived**, integer, binary indicator
- **Pclass**, integer, an ordinal variable for the passenger class.
- **Name**, Factor with 891 levels (one level per passenger).
- **Sex**, Factor with two levels: “female”, “male”.
- **Age**, numerical, has 177 missing values coded as NA.
- **SibSp**, integer, an ordinal variable for the number of siblings or spouses.
- **Parch**, integer, an ordinal variable for the number of parents or children.
- **Ticket**, Factor with 681 levels.
- **Fare**, numerical, is in Pounds Sterling.
- **Cabin**, Factor with 147 levels, has 687 missing values.
- **Embarked**, Factor with

Summary of the Data:



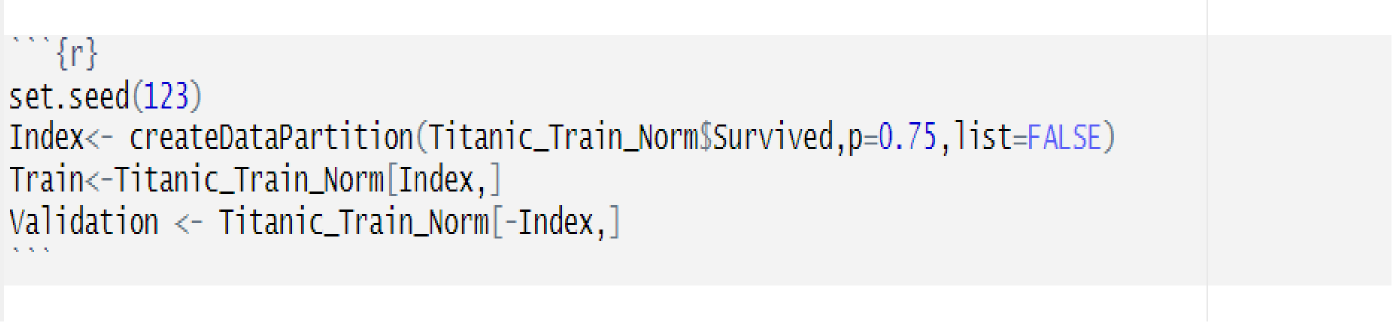
~~~
```{r}
summary(Titanic_Train)
```
~~~

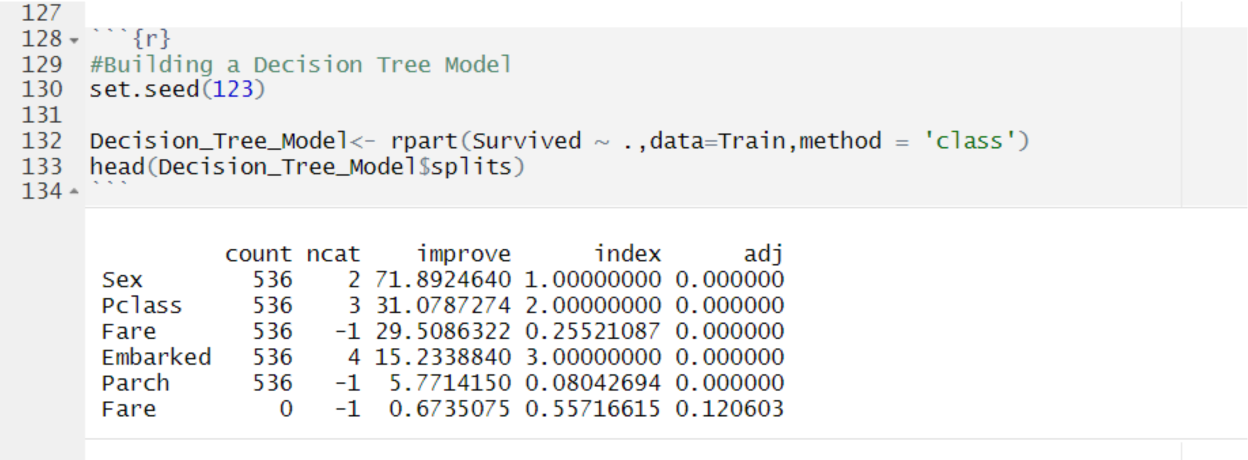
```
  PassengerId       Survived          Pclass          Name               Sex                 Age            SibSp      
 Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891         Length:891         Min.   : 0.42   Min.   :0.000  
 1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character   Class :character   1st Qu.:20.12   1st Qu.:0.000  
 Median :446.0   Median :0.0000   Median :3.000   Mode  :character   Mode  :character   Median :28.00   Median :0.000  
 Mean   :446.0   Mean   :0.3838   Mean   :2.309                                         Mean   :29.70   Mean   :0.523  
 3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000                                         3rd Qu.:38.00   3rd Qu.:1.000  
 Max.   :891.0   Max.   :1.0000   Max.   :3.000                                         Max.   :80.00   Max.   :8.000  
                                                                                        NA's   :177                    
     Parch           Ticket               Fare           Cabin              Embarked        
 Min.   :0.0000   Length:891         Min.   :  0.00   Length:891         Length:891        
 1st Qu.:0.0000   Class :character   1st Qu.:  7.91   Class :character   Class :character  
 Median :0.0000   Mode  :character   Median : 14.45   Mode  :character   Mode  :character  
 Mean   :0.3816                      Mean   : 32.20                                        
 3rd Qu.:0.0000                      3rd Qu.: 31.00                                        
 Max.   :6.0000                      Max.   :512.33                                        
```

The dataset contains 12 attributes in total and in those attributes Passenger ID is just an index but not the attribute of the passenger.

DATA PARTITION:

After there are no missing values in the data, by using the caret package we divide the data into training and test sets. The partition is set for 0.8% which is training set of 80% and the rest 20% is the testing set.

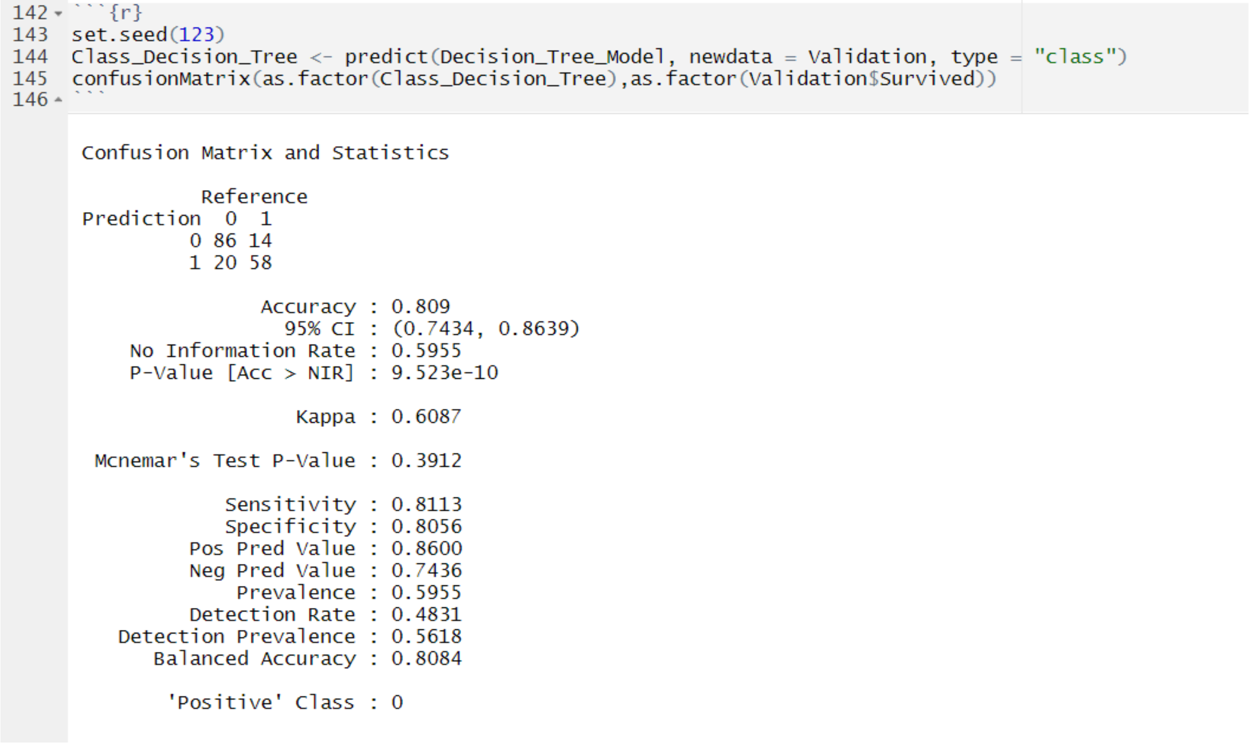


~~~
```{r}
set.seed(123)
Index<- createDataPartition(Titanic_Train_Norm$Survived,p=0.75,list=FALSE)
Train<-Titanic_Train_Norm[Index,]
Validation <- Titanic_Train_Norm[-Index,]
~~~
```

DETAILS OF MODELLING STRATEGY:

For the modelling strategy I used the classification techniques, In machine learning for the 80% of training data to determine the most accurate model for the best results.

BUILDING THE DECISION TREES MODEL:



```

```{r}
#Building a Decision Tree Model
set.seed(123)

Decision_Tree_Model<- rpart(Survived ~ .,data=Train,method = 'class')
head(Decision_Tree_Model$splits)
```

         count ncat   improve   index   adj
Sex   536   2 71.8924640 1.00000000 0.000000
Pclass   536   3 31.0787274 2.00000000 0.000000
Fare   536   -1 29.5086322 0.25521087 0.000000
Embarked   536   4 15.2338840 3.00000000 0.000000
Parch   536   -1 5.7714150 0.08042694 0.000000
Fare   0   -1 0.6735075 0.55716615 0.120603
```
```

THE CONFUSION MATRIX FOR DECISION TREES MODEL:



~~~
```{r}
set.seed(123)
Class_Decision_Tree <- predict(Decision_Tree_Model, newdata = Validation, type = "class")
confusionMatrix(as.factor(Class_Decision_Tree),as.factor(Validation$Survived))
```
~~~

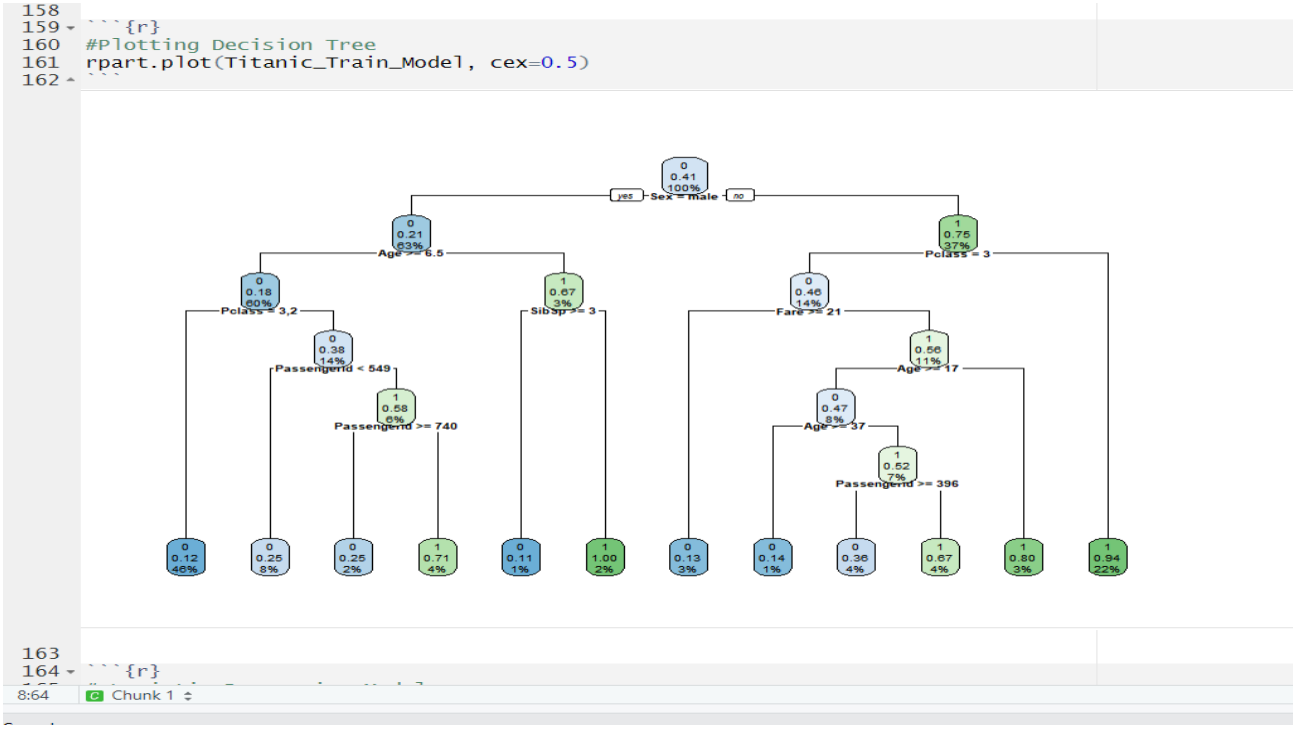
```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 86 14
         1 20 58
                                          
               Accuracy : 0.809           
                 95% CI : (0.7434, 0.8639)
    No Information Rate : 0.5955          
    P-Value [Acc > NIR] : 9.523e-10       
                                          
                  Kappa : 0.6087          
                                          
 Mcnemar's Test P-Value : 0.3912          
                                          
            Sensitivity : 0.8113          
            Specificity : 0.8056          
         Pos Pred Value : 0.8600          
         Neg Pred Value : 0.7436          
             Prevalence : 0.5955          
         Detection Rate : 0.4831          
   Detection Prevalence : 0.5618          
      Balanced Accuracy : 0.8084          
                                          
       'Positive' Class : 0               
```

From the above model can observe that the “ACCURACY” of the model is 80.09%,

And the “SENSITIVITY” is 81.13%,

And the “SPECIFICITY” is 80.56%.

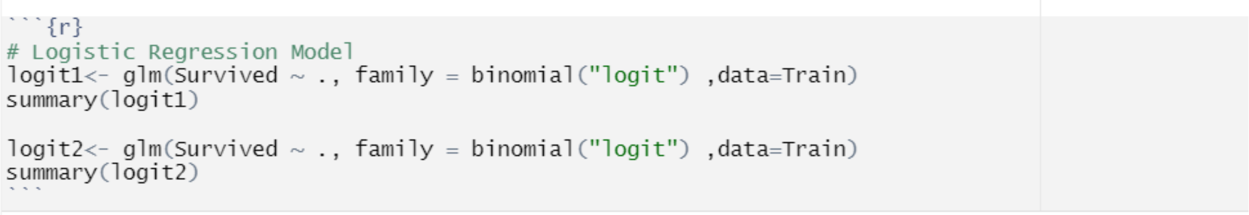
DECISION TREES RPLOT:



~~~
```{r}
#Plotting Decision Tree
rpart.plot(Titanic_Train_Model, cex=0.5)
```
~~~

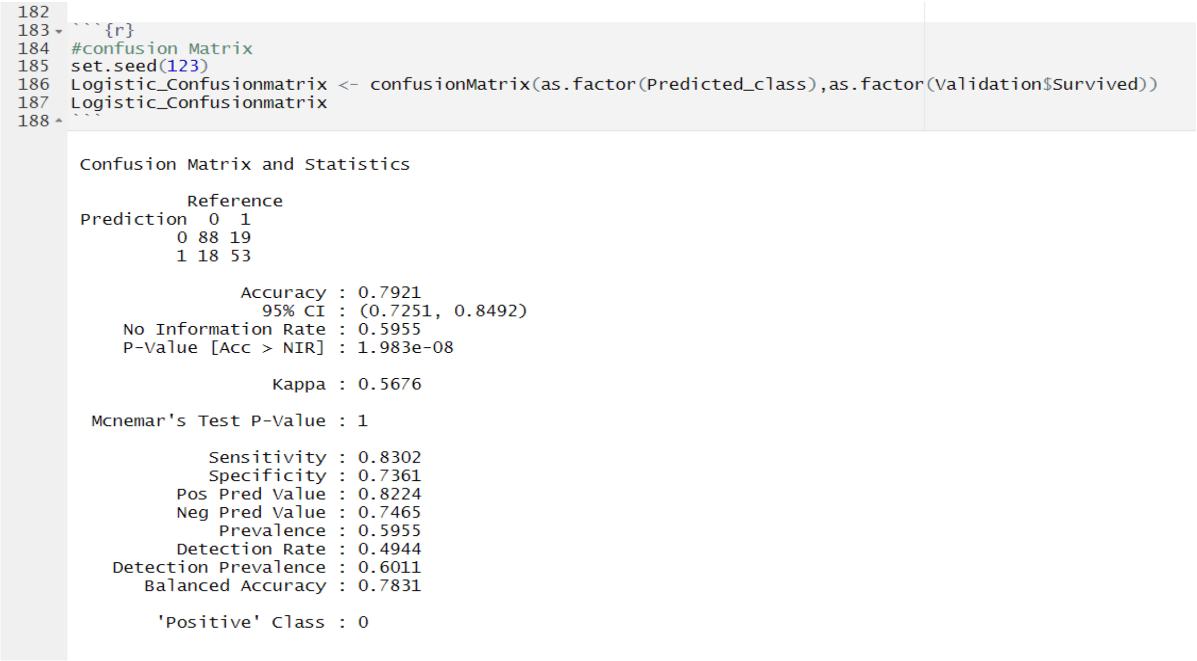
~~~
```{r}
~~~

BUILDING THE LOGISTIC REGRESSION MODEL:



~~~
```{r}
# Logistic Regression Model
logit1<- glm(Survived ~ ., family = binomial("logit") ,data=Train)
summary(logit1)

logit2<- glm(Survived ~ ., family = binomial("logit") ,data=Train)
summary(logit2)
```
~~~

BUILING THE CONFUSION MATRIX FOR LOGISTIC REGRESSION:



~~~r
```{r}
#confusion Matrix
set.seed(123)
Logistic_Confusionmatrix <- confusionMatrix(as.factor(Predicted_class),as.factor(Validation$Survived))
Logistic_Confusionmatrix
~~~
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 88 19
         1 18 53

               Accuracy : 0.7921
                 95% CI : (0.7251, 0.8492)
    No Information Rate : 0.5955
    P-Value [Acc > NIR] : 1.983e-08

                  Kappa : 0.5676

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.8302
            Specificity : 0.7361
         Pos Pred Value : 0.8224
         Neg Pred Value : 0.7465
             Prevalence : 0.5955
         Detection Rate : 0.4944
   Detection Prevalence : 0.6011
      Balanced Accuracy : 0.7831

       'Positive' Class : 0
```

From the above, we can observe that

The “ ACCURACY” is 79.21%

The “SENSITIVITY” IS 83.02%

The “Specificity” is 73.61%

Conclusion:

From the Above models we can conclude that the Decision trees model gives us the best Accuracy that is 80% whereas the Logistic regression model gives us 79% Accuracy which is less than the Decision trees.

So, we are using the Decision trees model for our Titanic Train dataset on the test data as it gives the better accuracy than other models.
```

**<u>REFERENCES</u>-** KAGGLE