International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 3, May-June-2020, ISSN (Online): 2395-566X

# Analyzing Titanic Disaster using Machine Learning Algorithm

**Ramandeep Singh**
Computer Science Engineering
Dronacharya College of Engineering
Khentawas, Farrukh nagar, Gurugram, Haryana 123506, INDIA
Raman77768@gmail.com

***Abstract* – Titanic disaster occurred 108 years ago, killing 1496 passengers and crew members. This collision led to curiosity among number of communities of researchers and analysts regarding the factors which could have decided the survival of some passengers and demise of the others. This Titanic dataset consists of various features and we attempt to determine the correlation among various features such as passenger fare , cabin, age , sex, ticket class etc. leading to the survival of the passenger. We use machine learning algorithm namely**

**Random Forest to analyze correlation between these features and to predict survival of passengers.**

***Keywords*– Machine Learning Algorithm, titanic disaster, random forest, r language.**

## I. INTRODUCTION

As the field of machine learning is constantly moving forward it has allowed various individuals, analysts and researchers to uncover meaningful studies from large amount of data. It has unlocked various possibilities such as giving a new perspective to the past events and trying to predict the future events.

Titanic disaster is one of the most famous shipwrecks in the world history. Titanic was a British cruise liner that sank in the North Atlantic Ocean a few hours after colliding with an iceberg. While there are facts available to support the cause of the shipwreck, there are various speculations regarding the survival rate of passengers in the Titanic disaster. Over the years, data of survived as well as deceased passengers has been collected. The dataset is available to public on a website called Kaggle.com [1]. This dataset has been studied and analyzed using various machine learning algorithms such as Random Forest , SVM , KNN algorithm etc.

Various languages and tools are used to implement these algorithms including R, Java, Python and other statistical tools. Our approach is centered on R and Python as there are vary proficient for executing algorithms- Naïve Bayes, Logistic Regression, Decision Tree, and Random Forest. The prime objective of the research is to analyze Titanic disaster to determine a correlation between the survival of passengers and characteristics(i.e age, sex, fare etc.) of the passengers using random forest machine learning algorithms.

## II. DATASET

This dataset is available on the website Kaggle.com with the name of Titanic extended dataset. Each row in this dataset belongs to an unique individual and its details. For each passenger we are provided with whether they survived or not, their Ticket class, sex, age, number of siblings/ spouses aboard the titanic, number of parents / children aboard the titanic, ticket number , fare ,cabin , Port of Embarkation, WikiId, Name, Hometown, Boarded, Destination, Lifeboat, body, class. The data is in the form of CSV (Comma Separated Value) file. The whole dataset is divided into two parts ,first is train dataset which we will feed into our machine learning model/algorithm to tell our model about the data so that it can learn from it and second part is the test dataset which we will use to test our model's performance/ accuracy and other meaningful scores.

The structure of dataset including sample rows:

Table -I:Kaggle Dataset

| PassengerId | Survived | Pclass | Name |
|---|---|---|---|
| 1 | 0 | 3 | Braund, Mr. Owen Harris |
| 2 | 1 | 1 | Cumings, Mrs. John Bradley |

Table -II: Kaggle Dataset (Contd.)

| Sex | Age | Sibsp | parch | ticket |
|---|---|---|---|---|
| male | 22 | 1 | 0 | A/5 21171 |
| female | 35 | 1 | 0 | PC 17599 |

Table -III: Kaggle Dataset (Contd.)

| cabin | Lifeboat | Boarded | Destination | Fare |
|---|---|---|---|---|
| | | Southampton | Qu'Appelle Valley, Saskatchewan, Canada | 7.25 |
| C85 | 4 | Cherbourg | New York, New York, US | 71.2833 |

Before fitting any model or algorithm on the dataset we would do exploratory analysis on the dataset using r language, so that we can understood the data better and see if we can find any early signs of correlation. We started from basic generic x-y plots and boxplots on various features/ columns.

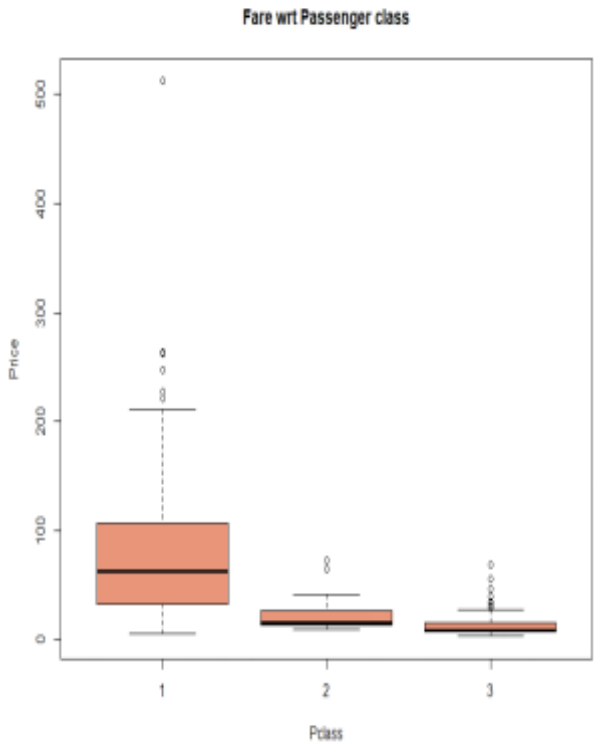


Fig .1.FARE ~ PCLASS.

FIG 1 shows boxplot of fare for all the three passenger classes and showing outliners in the fare column of the dataset .FIG 2 shows boxplot of age of survived people and people who did not make it.
As we can see in fig1 there are large number of outliners for 1st and 3rd passenger class whereas in fig2 there are some outliners in boxplot for survived people.
FIG 3 is a barplot of survival rate of males and females per each passenger class.

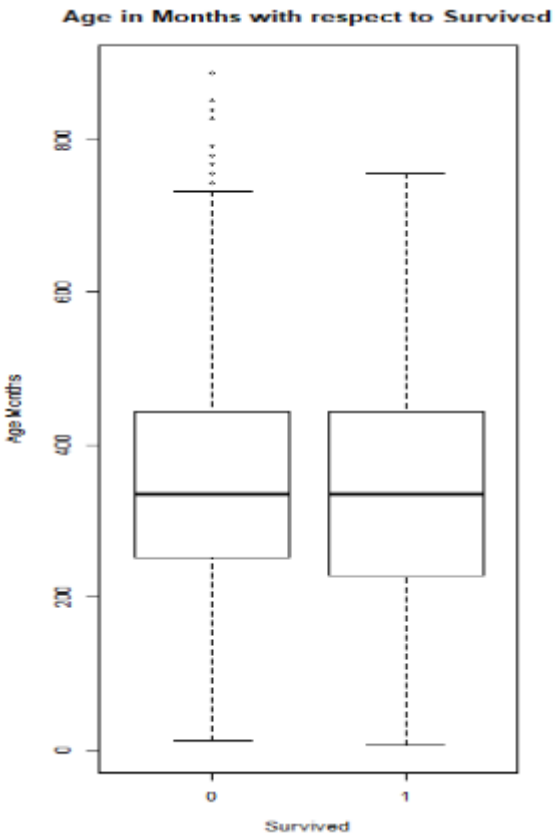


Fig. 2.Age.

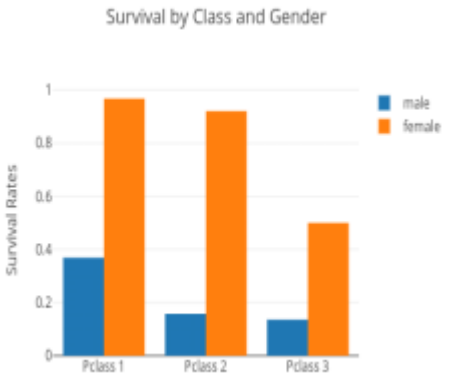


Fig.3. Survived.

After analyzing and exploring the data we performed some data cleaning operations such as handling NA values and assigning reasonable values to the outliers so that it cannot hinder our models learning process.

## III. METHODOLOGY

One of the most important steps were cleaning the data , handling all the null values , zeroes , NA's etc. On first exploration there were more than 170 plus NA's on

various rows and columns. Most of the outliners were in the fare column and age column.
Our second step was to create more meaningful columns such as extracting countries from boarding and destination locations. Changing age column values (in years ) to months , Creating new attributes such as lifeboat support etc.
Fig 4 consist of correlation graph between various features such as age, fare etc.

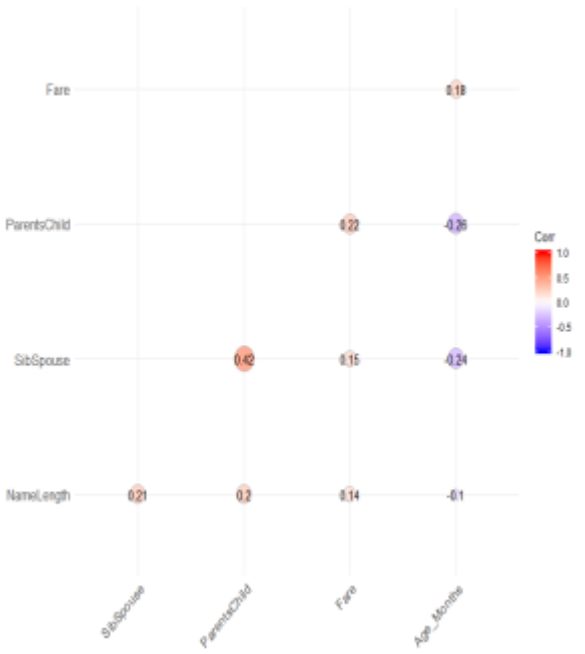


Fig.4.Correlation Plot.

Then in order to analyze important and unimportant features in the dataset we applied Boruta algorithm on the dataset. The Boruta algorithm is a wrapper built around the random forest classification algorithm. It tries to capture all the important, interesting features you might have in your dataset with respect to an outcome variable (in our case Survived Column).

```
> btrain$finalDecision
      PassengerId          Pclass            Name
         Rejected       Confirmed        Rejected
     ParentsChild    TicketNumber           Cabin
        Confirmed       Confirmed       Confirmed
     Destination DestinationCountry        Lifeboat
         Rejected        Rejected       Confirmed
Levels: Tentative Confirmed Rejected
```

```
      NameLength             Sex         SibSpouse
       Tentative       Confirmed       Confirmed
      Age_Months HometownCountry         Boarded
       Confirmed       Confirmed       Tentative
  LifeboatSupport        Fare_new
       Confirmed       Confirmed
```

Fig. 5. Boruta Final Decision.

Boruta algorithm assigns different levels to each feature in the dataset , each level telling whether the feature should be confirmed (i.e. important) or should be rejected or is tentative.
Then we trained our random forest model on the cleaned training dataset with different parameters such as number of trees. We trained our first model with more than 200 trees.

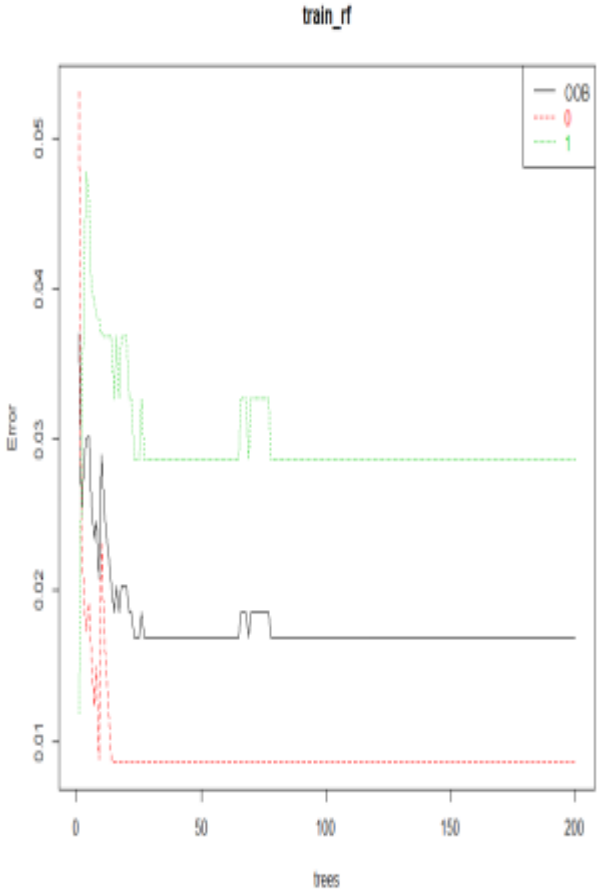


Fig. 6. Randomforest on Train Dataset.

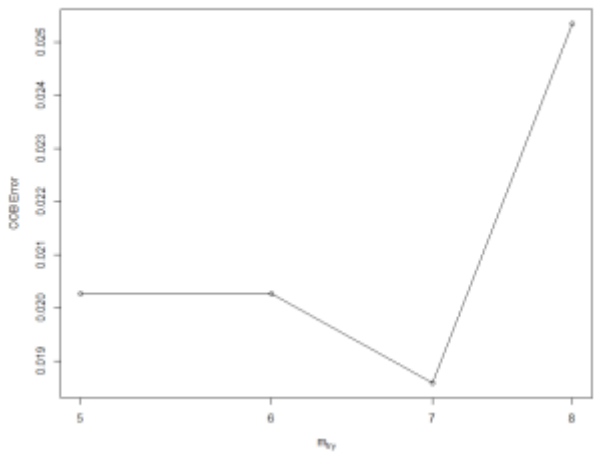


Fig.7.OBB Error Plot.

The out-of-bag (OOB) error is the average error for each calculated using predictions from the trees that do not contain in their respective bootstrap sample. This allows the RandomForestClassifier to be fit and validated whilst being trained. In the fig 7 out of bag error is lowest for 7 estimators after which the graph rises.

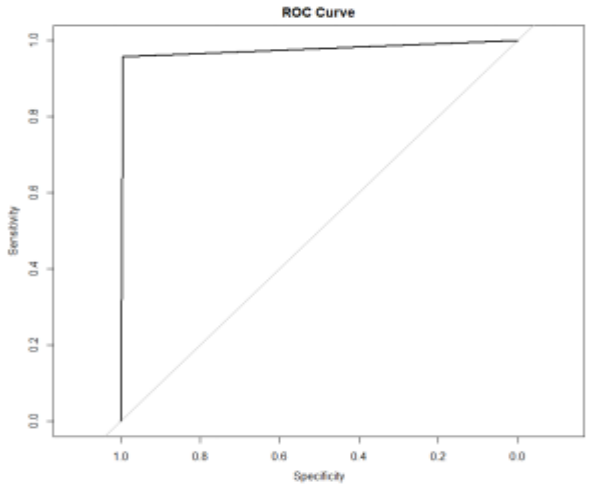


Fig.8. Roc Curve.

A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate against the false positive rate at various threshold settings and our curve seems to be doing very well.

## IV. CONCLUSION/FUTURE WORK

After applying Boruta algorithm and tuning the randomforest with various parameters we have used 7 estimators in the final prediction. This randomforst model got a accuracy of approximately 98% on the test dataset.

It got a F1 Score of 0.9860724 . And when applying confusion matrix we got a balance accuracy of 0.9835 .
Our tuned randomforest model suggested that Pclass, sex, age, children and SibSp were one of the most important features of the model and the survival of a passanger was highly dependent on these features. It would be interesting to play more with dataset and introducing more attributes which might lead to good results.
Various other machine learning techniques like SVM, K-NN classification can be used to solve the problem.

```
> confusionMatrix(as.factor(test1$Survived),test_pred2)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 177   1
         1   4  90

               Accuracy : 0.9816
                 95% CI : (0.9576, 0.994)
    No Information Rate : 0.6654
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9591

 Mcnemar's Test P-Value : 0.3711

            Sensitivity : 0.9779
            Specificity : 0.9890
         Pos Pred Value : 0.9944
         Neg Pred Value : 0.9574
             Prevalence : 0.6654
         Detection Rate : 0.6507
   Detection Prevalence : 0.6544
      Balanced Accuracy : 0.9835

       'Positive' Class : 0
```

Fig. 9. Confusion Matrix.

## REFERENCES

[1]. Kaggle, Titanic: Machine Learning form Disaster [Online]. Available: https://www.kaggle.com/pavlofesenko/titanic-extended
[2]. Eric Lam, Chongxuan Tang. Titanic – Machine LearningFromDisaster. Available FTP: cs229.stanford.edu Directory: proj2012 File: LamTang-TitanicMachineLearningFromDisaster.pdf
[3]. Wikipedia. ROC Curve [Online]. Available:https://en.wikipedia.org/wiki/Receiver_operating_characteristic
[4]. Miron B. Kursa, Witold R. Rudnicki - Feature Selection with the Boruta Package. Available: https://www.jstatsoft.org/article/v036i11
[5]. Trevor Stephens. (2014). Titanic: Getting Started With R - Part 3: Decision Trees [Online]. Available: http://trevorstephens.com/kaggletitanic-tutorial/r-part-3-decision-tree