Predicting chances of surviving the disaster using Machine Learning Algorithms

Samama Imtiaz Butt
Department of Computer Science,
***National University of Computer and Emerging Sciences* Lahore, Pakistan**
l181882@lhr.nu.edu.pk

***Abstract*— Titanic disaster occurred 100 years ago on April 15, 1912, killing about 1500 passengers and crew members. The critical episodes still urge the scientists and examiners to comprehend what could have prompted the survival of certain travellers and the destruction of the others. With the use of machine learning methods and a dataset consisting of 891 rows in the train set and 418 rows in the test set, we attempt to determine the correlation between factors such as age, sex, passenger class, fare etc. to the chance of survival of the passengers. These factors might have affected the survival rates of the travellers.**

Various machine learning algorithms such as Logistic Regression, Naïve Bayes, Decision Tree, Random Forest has been used on this dataset to predict the survival of passengers. Specifically, we will compare these algorithms by implementing it on the dataset in this report.

I. INTRODUCTION

In recent years, 50% of the entire world's data has been produced. Since a lot companies has collected this data they began to think how they could use this data to get additional benefits. These databases were extremely large and was beyond human capacity to analyse. A wise man suggested to do predictive analysis on such databases. Predictive analytics is a procedure that incorporates the use of computational methods to determine important and useful patterns in large data. The algorithms that we use in predictive analysis are of two types; Supervised Learning algorithms or Unsupervised Learning algorithms. Supervised learning models aim to predict a target variable, represented by a single column in the dataset, by using the other variables or columns in the dataset. Supervised learning is also known as predictive modelling. The most common predictive modelling algorithms are classification when dealing with a categorical target variable or regression in the context of a continuous target variable. Unsupervised learning does not have a target variable, but rather builds a model using clusters of the data. Unsupervised learning models are referred to as descriptive modelling. The next paragraph provides a description of a supervised learning problem involving passengers on the RMS Titanic.

The field of machine learning has allowed analysts to uncover insights from historical data and past 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 publically available on a website called Kaggle.com. The fateful incidents still compel the researchers and analysts to understand what could have led to the survival of some passengers and demise of the others.

Titanic dataset has been studied and analyzed using various machine learning algorithms like Random Forest, SVM etc. Various languages and tools are used to implement these algorithms including Weka, Python, R, Java etc. Our approach is centered on Python for executing these algorithms- Naïve Bayes, Logistic Regression, Decision Tree, and Random Forest. The prime objective is to compare various machine learning algorithms using this dataset.

II. DATASET

If anyone is interested to start machine learning predictions with introductory level problem RMS Titanic is the best option. The dataset we use was provided by the Kaggle website. Kaggle is one of the largest data science community and it organizes open predictive modelling competitions. The data is consists of 891 rows in the train set which is a passenger sample with their associated labels. For each passenger, we were also provided with the name of the passenger, sex, age, his or her passenger class, number of siblings or spouse on board, number of parents or children aboard, cabin, ticket number, fare of the ticket and embarkation. The data is in the form of a CSV (Comma Separated Value) file. For the test data, we were given a sample of 418 passengers in the same CSV format.

A. Data Exploration/Analysis

The training set is consist of 891 examples and 11 features excluding target feature i.e. Survived. 2 of the features are floats, 5 are integers and 5 are objects. As shown below:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Sex            891 non-null object
Age            714 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          204 non-null object
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
```

A short description of these features is given as following:

Feature	*Short Description*
survival	Survival
passengerId	Unique Id of a passenger
pclass	Ticket class
sex	Sex
Age	Age in years
Sibsp	Number of siblings/spouses aboard the Titanic
Parch	Number of parents/children aboard the Titanic
Ticket	Ticket Number
Fare	Passenger fare
Cabin	Cabin Number
Embarked	Port of Embarkation

We can conclude from the statistics given blow that 38% of the passengers from the training set survived the Titanic. Count shows that there are some missing values in 'Age' feature. We can also see that passenger ages are from 0.4 to 80.

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Now let's look at the first 9 rows of the training set.

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S

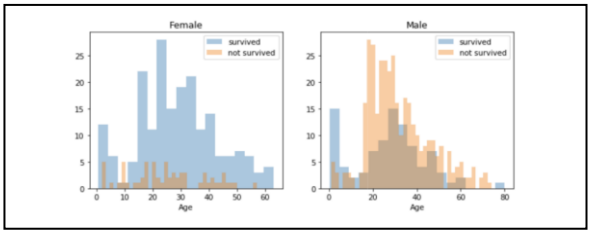
From this table, we can note that we will have to convert few features into numeric values so that they can be processed by the machine learning algorithms. In addition, we can see that features have different wide ranges and they should be normalized. Also, we will be dealing with the features having missing values.

Following table shows the details of the missing features:

	Total	%
Cabin	687	77.1
Age	177	19.9
Embarked	2	0.2
Fare	0	0.0
Ticket	0	0.0

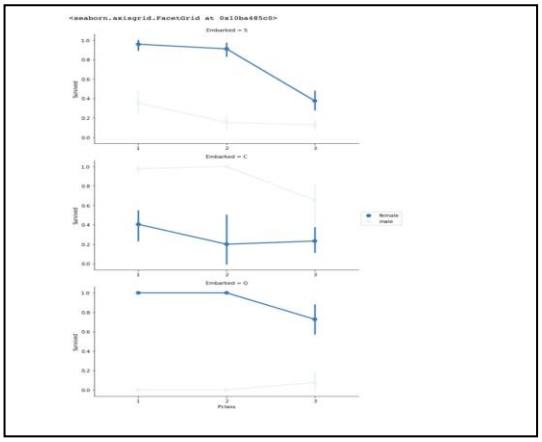
We can observe from the above that 'Embarked' feature has only 2 missing values which can be easily manipulated but it will be very difficult to deal with the 'Age' feature since it has 177 missing values in total. We might need to drop the 'Cabin' feature because 77% of its data is missing.
Now we have to figure out that which feature will be useful to predict the survival of the passenger accurately. We can say that 'Name', 'PassengerId' and 'Ticket; will not contribute much in prediction.

1) Age and Sex:

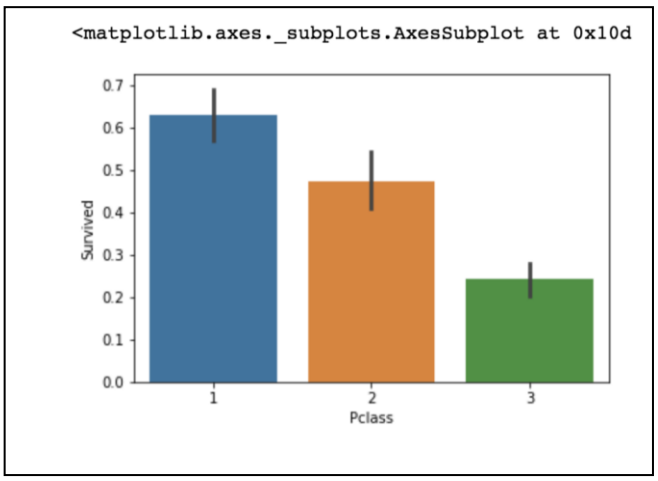


The graph displays that men have higher probability rate of survival having ages between 18 and 30. For women the situation is slightly different they have higher survival chances between 14 and 40. The survival rate is bit low between the ages of 5 and 18 in men but it does not apply on women. Another thing to note that babies have higher probability of survival.

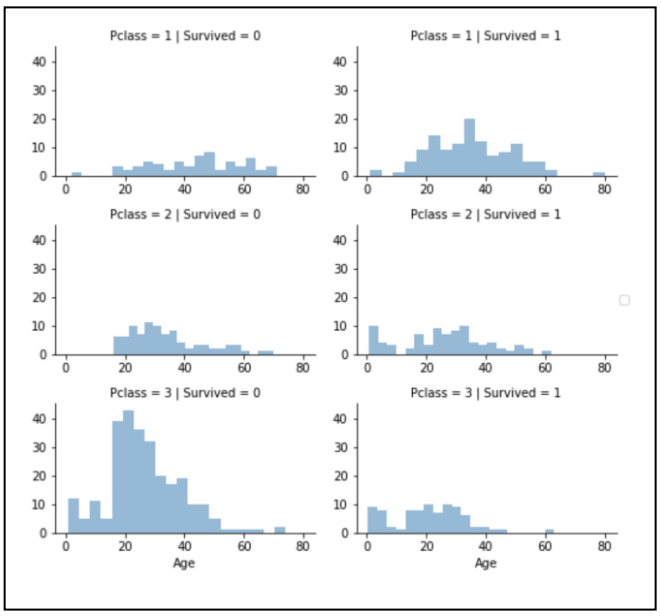
2) Embarked, Pclass and Sex:



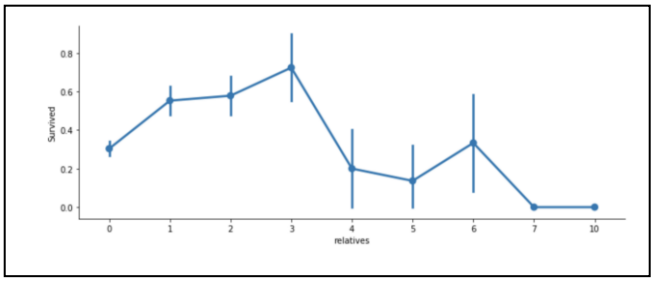
Embarked is correlated with survival depending on the gender. Women on port Q and S have higher chances of survival and less at port C. On the other hand, men have higher survival probability at port C and less at port Q and S.
Pclass can also help us in predicting survival. Let's generate a separate graph for it.



Here we can see clearly that probability of survival at Pclass 1 is high. We will plot one more graph.



3) Sibsp and Parch:
Create a feature 'Relative' after combining Sibsp and Parch and it shows the stats given below:



We can observer that a passenger has higher chances of survival if he/she has relatives between 1 and 3. Survival chances become lower if a passenger has less than 1 or greater than 3 relatives except for the cases with six relatives.

B. Data Normalization

.Firstly, I will drop 'PassengerId' from the training set because it is not helpful in predicting chances of survival.

1) Missing Data

We have to deal with three features i.e. Cabin, Age and Embarked. I was planning to drop Cabin feature directly then I found that Cabin contains deck number which we can extract to create a new feature and convert it into numeric values. We will place missing values with 0.
Now let's tackle the missing values of feature Age. I will create an array containing random values based on the mean age in regards to the standard deviation and is_null. Since Embarked feature has only two missing values we will fill these with most common one.

2) Converting Features

- Converting Fare feature from float to int64 using "astype ()" method pandas provide.
- Extracting titles from Name feature and creating a new feature.
- Converting Sex feature into numeric value.
- As Ticket feature contains 681 unique tickets, it will be tricky to convert them into useful categories so we will drop it from the dataset.
- Converting Embarked attribute into numeric value.

3) Creating Categories

Now we will create categories within the following features.

a) Age
We will create a new 'AgeGroup' by categorizing every age into a group. Here we have to make sure that our most of the population does not fall into one group.

b) Fare
For Fare attribute we need to do the same as with Age. But if we group it by few big categories then 80% of the data fall into first category. Fortunately, we have "qcut" method to deal with this issue.

4) Creating new features

I will add two more features to the dataset that I compute out of other features i.e. **Age times Class and Fare per Person**. After adding these features dataset will look like below:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	relatives	not_alone	Deck	Title	Age_Class	Fare_Per_Pers
0	0	3	0	2	1	0	0	0	1	0	8	1	6	0
1	1	1	1	5	1	0	3	1	1	0	3	3	5	1
2	1	3	1	3	0	0	0	0	0	1	8	2	9	0
3	1	1	1	5	1	0	3	0	1	0	3	3	5	1
4	0	3	0	5	0	0	1	0	0	1	8	1	15	1
5	0	3	0	4	0	0	1	2	0	1	8	1	12	1
6	0	1	0	6	0	0	3	0	0	1	5	1	6	3
7	0	3	0	0	3	1	2	0	4	0	8	4	0	0
8	1	3	1	3	0	2	1	0	2	0	8	3	9	0
9	1	2	1	1	1	0	2	1	1	0	8	3	2	1
10	1	3	1	0	1	1	2	0	2	0	7	2	0	0

III. Related Work

Many researchers have worked on the Titanic problem in order to compare various different machine learning

techniques in terms of the efficiency of the algorithm to predict the survival of the passengers. Studies have tried to trade-off between different features of the available dataset to provide the best prediction results. Lam and Tang et al. used the Titanic problem to compare and contrast between three algorithms- Naïve Bayes, Decision tree analysis and SVM. They concluded that sex was the most dominant feature in accurately predicting the survival. They also suggested that choosing important features for obtaining better results is important. There were no significant differences in accuracy between the three methods they used [2]. Shawn Cicoria and John Sherlock et al. performed Decision tree classification and Cluster analysis to suggest that sex is the most important feature as compared to other features in determining the likelihood of the survival of passengers [3]. Kunal Vyas and Lin et al. suggested that dimensionality reduction and playing more with the dataset could improve the accuracy of the algorithms. The most important conclusion provided by them was that more features utilized in the models do not necessarily make results better [4]. Although many researchers have worked hard to determine the actual cause of the survival of some passengers and demise of others, we attempt to get better results and accuracy by utilizing various different combination of features and different machine learning methods.

IV. STOCHASTIC GRADIENT DESCENT (SGD)

The word '*stochastic*' means a system or a process that is linked with a random probability. SGD is an optimizing machine learning technique. In SGD, few random samples are selected instead of whole training data for each iteration. Batch is the total number of samples you use to calculate the gradient in single iteration. In typically gradient descents, batch is taken as the whole training data. Although it is good to use the whole data to get the gradient since it gives you the minima in a less noisy or random manner but it is computationally very expensive when dataset it huge.

This problem is solved in SGD where we take single sample per each iteration as a batch. The sample is randomly shuffled and selected for each iteration.
One more thing to be noted, SGD is much noisier than the typical gradient descent because of its randomness in its descent. Even though it requires a higher number of iterations to reach the minima than the indigenous gradient descent, it is computationally less expensive.

V. DECISION TREES

The decision trees are non-parametric supervised machine learning algorithm used for classification and regression. The main objective is to build a model to predict that predicts the value of the target variable by learning some decision rules using the training data.

1) Advantages

- Decision trees are simple to understand and visualize.
- It requires little data preprocessing.
- The cost of decision tree is the number of features used to train the data.
- Can handle both numerical and categorical data.
- Able to handle multiple output problems.

2) Disadvantages

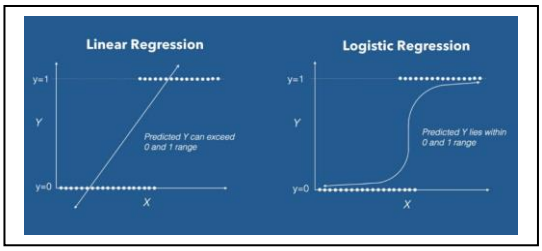
- Decision trees are unstable because different variables in data might create a completely different decision tree.
- If some classes dominates it creates a biased decision tree.
- It does not generalize well and overfitt the data.
- XOR, parity and multiplexer type problems are hard to learn in decision trees.

VI. .RANDOM FOREST

Random Forest is a supervised learning model. Like its name it creates a forest of decision trees and somehow make it random. The forest is an ensemble of Decision Trees most of the trained with the bagging method. The idea of bagging method is that combination of learning models increases the overall results.
Random Forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. It prevents overfitting by creating random subsets of the features and building smaller trees using these subsets.

VII. LOGISTIC REGRESSION

Logistic Regression is a machine algorithm used to solve



classification problems.
We can call Logistic Regression as Linear Regression but there are few differences between them. Logistic Regression uses a sigmoid function as a cost function. The main goal of this sigmoid function it to limit the output between 0 and 1.

$$f(x) = \frac{1}{1 + e^{-(x)}}$$

Sigmoid function can be represented as:

$$\sigma(Z) = \sigma(\beta_0 + \beta_1 X)$$

Linear Regression formula of the hypothesis is:

$$h\Theta(x) = \beta_0 + \beta_1 X$$

For Logistic Regression, it is modified a bit:
We have expected that our hypothesis will give us the values between 0 & 1.

$$Z = \beta o + \beta 1X$$

$$h\Theta(x) = sigmoid(Z)$$

$$i.e.\ h\Theta(x) = 1/(1 + e^{-(\beta o + \beta 1X)})$$

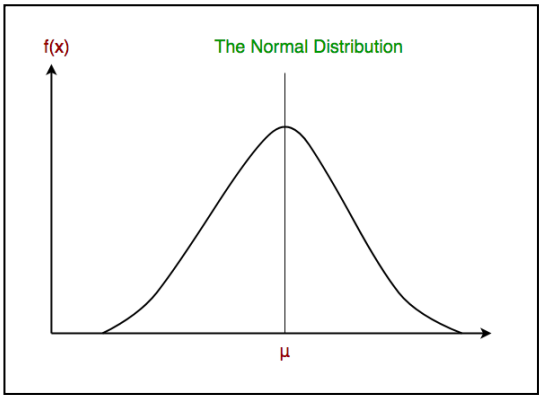
Next step is to apply the gradient descent to find global minimum of the cost value.

VIII. K NEAREST NEIGHBOR

KNN is a machine learning algorithm used to solve both classification and regression problems. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. KNN captures the idea of similarity by finding the distance between two points in a graph. There are many ways to compute the distance between two points but Euclidean distance is most popular among them.

IX. GAUSIAN NAÏVE BAYES

In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a **Gaussian distribution**. A Gaussian distribution is also called Normal distribution. When plotted, it gives a bell shaped curve which is symmetric about the mean of the feature values as shown below:

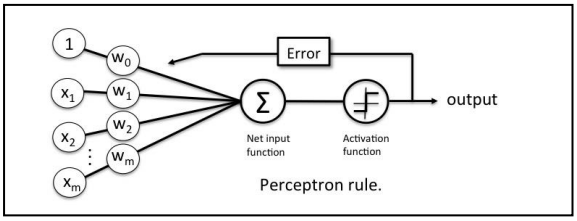


X. PERCEPTRON

In 1957, Frank Rosenblatt started investigating the Artificial Neural Network at the Cornell Aeronautical Laboratory which he later called Perceptron. We can relate Perceptron to the neuron information processing where a neuron accepts signals from dendrites and passes electoral signals down to the cell body.

Similarly, Perceptron takes input from training data that we weight and combined it in a linear equation. Combined Linear Equation generates an output or prediction using transfer function. It can be described as a classification algorithm for problems with two classes i.e. 0 and 1 where a linear equation can be used to separate two classes. A perceptron can have multiple inputs but outputs only a binary label.

In perceptron learning, we create a weight vector by initializing the weights randomly. Then we update these weights by iteratively applying perceptron rule to each training data input. The process is being repeated until the correct classification of each training example.



Perceptron is further divided into two types;

A. *Single Layer Perceptron*

It only learns linearly separable patterns.

B. *Multilayer or Feedforward Perceptron*

It has greater processing capacity and can learn non-linear patterns.

XI. LINEAR SUPPORT VECTOR MACHINE

SVM is a supervised machine learning algorithm which can be used for both classification and regression problems but mostly used for classification. In SVM, we plot each data point into n dimensional space (where n is the number of features) with value of each feature being the value of particular coordinate. Then classification is done by forming a hyper-plane that differentiate two classes.

EVALUATION

Score	Model
92.82	Random Forest
92.82	Decision Tree
87.32	KNN
81.14	Logistic Regression
80.81	Support Vector Machines
80.70	Perceptron
77.10	Naive Bayes
76.99	Stochastic Gradient Decent

Results shows that Random Forest outperforms all other algorithms. Let's check its check its performance on cross validation.

A. K-Fold Cross Validation

It splits the data into random K subsets called folds. Let's suppose our training data into 4 folds i.e. K=4. Our Random Forest algorithm will be trained and test 4 times using different fold each time, while it would be trained on remaining 3 folds.
We will perform K-Fold cross validation on Random Forest algorithm using 10 folds. The results of K-Fold cross validation will be an array containing 10 different scores and we then compute mean and standard deviation of those results as shown below:

```
Scores: [ 0.76666667  0.82222222  0.7752809   0.82022472  0.85393258  0.86516854
  0.83146067  0.76404494  0.85393258  0.85227273]
Mean: 0.820520655998
Standard Deviation: 0.0367333665466
```

The states that our model has the average accuracy of 82% with the standard deviation of 4% which is quite impressive.

B. Confusion Matrix

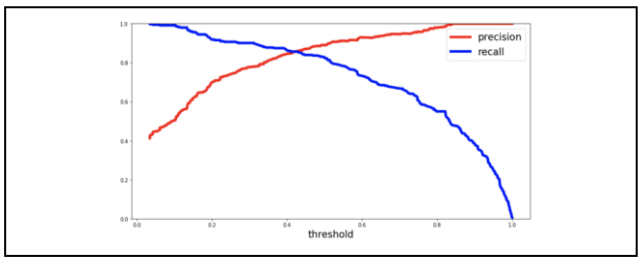
According to the confusion matrix, 488 passengers is correctly classified as **Not Survived** and 61 were wrongly classified. On the other hand 95 passengers were misclassified as **Survived** and 247 were properly classified.

C. Precision Call

A confusion matrix gives you information about how well your model works but we can get more detailed information by using precision call.
Our model predicts 81% of the time, a passenger's survival correctly (precision). The recall tells us that it predicted the survival of 73 % of the people who actually survived.

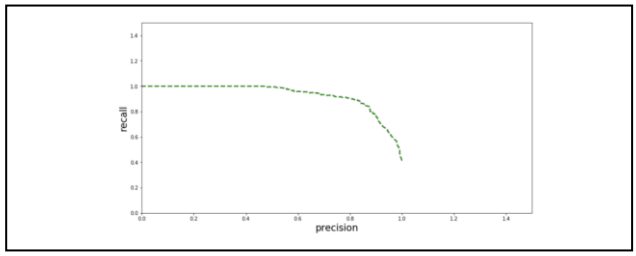
D. F-Score

F-Score combines the precision and recall score together. It can be computed by taking the harmonic mean of precision and recall. It assigns a lot of weight to the low values due to which a model only gets a high F-Score if it's both precision and recall values high.
There we have it, a 77% F-Score. The score is not that high



because we have a recall of 73%. Unfortunately, F-Score favors the classifiers having that have similar precision and recall. The problem is that sometimes increase in recall results in decrease in precision and it is called precision/recall tradeoff.

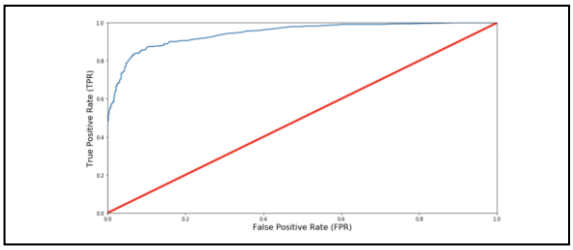
Above you can see clearly that recall and precision are intersecting at some point this can be your threshold to get maximum precision/recall tradeoff.
Another way is to plot precision and recall against each other like shown below:



E. ROC AUC

ROC AUC curve plots the true positive rate against the false positive rate instead of plotting precision versus recall.

The red line represents a purely random classifier and our classifier should be away from this line. Seems like Random Forest has done a good job.
We also have a tradeoff here like the more classifier produces false positive results the higher the true positive rate is.



ROC AUC score can be computer from the ROC AUC curve. It is simply computed by measuring the area under the curve which is called AUC.
A classifier which 100% accurate, would have a ROC AUC score of 1 and a random classifier would have a score of 0.5. Our model ROC AUC score is **0.945067587** which is good.

REFERENCES

[1] Kaggle, Titanic: Machine Learning form Disaster [Online]. Available: http://www.kaggle.com/

[2] Eric Lam, Chongxuan Tang. Titanic – Machine LearningFromDisaster.AvailableFTP: cs229.stanford.edu Directory: proj2012 File: LamTang-TitanicMachineLearningFromDisaster.pdf.

[3] Cicoria, S., Sherlock, J., Muniswamaiah, M. and Clarke, L, "Classification of Titanic Passenger Data and Chances of Surviving the Disaster," pp. 4-6, May 2014.

[4] Vyas, K., Zheng, Z. and Li, L, "Titanic-Machine Learning From Disaster," pp. 6, 2015.

[5] Mikhael Elinder. (2012). Gender, social norms, and survival in maritime disasters [Online]. Available: http://www.pnas.org/content/109/33/13220.full.