Name	**:**	**W.A.D. Chathuranga**
Reg. No.	**:**	**16APP2652**
Project	**:**	**Data Mining Project Report**

Classification Algorithms

Classification is the process of predicting the target/label or category of given data point. The task of approximating a mapping function (f) from discrete input variables (X) to classification predictive modelling (y). Classification is a type of supervised learning in which the input data is also delivered to the objectives.

There are lot of classification algorithms available. Such as Decision Tree, Random Forest, k-nearest neighbor, Gaussian Naïve Bayes, Logistic Regression, Stochastic Gradient Descent, Support Vector Machine, and Kernel Approximation. But it is impossible to say which one is superior to the other. It depends on the algorithm and nature of available data set.

Project Planning

# Introduction
# Import Required Libraries & Datasets
# Data Description
# Exploratory Data Analysis (EDA)
  # Summary statistics
  # Shape of Datasets
  # Data types
  # Missing data
  # Feature analysis
    # Categorical data analysis
    # Numerical data analysis
# Data preprocessing
  # Duplicated records handling
  # Data Transformation
  # Handle the Missing Values
# Feature engineering
# Feature encoding
# Splitting Dataset
# Feature scaling
# Create Model
# Hyperparameter Tuning
# Model Evaluation
  # Random Forest Classifier
  # Gaussian Naïve Bayes Classifier
  # Model comparison
# Final Submission
# Improve model accuracy

1. Introduction

The project is based on the Kaggle competition, **“Titanic - Machine Learning from Disaster**." The project goal is to build a machine learning model to learn the relationship between passenger features and their survival outcome on the Titanic and then make predictions of the survival of passengers' data that our model has not seen before. This problem is a binary classification problem in supervised learning.

2. Import Required Libraries & Datasets

First of all, I imported Libraries that are required and used in my code. Next, imported datasets and read them.

3. Data Description

PassengerId	It is a unique number automatically attributed to each passenger.
Survival	Survival (0 = No, 1 = Yes)
Pclass	Ticket class (1 = Upper, 2 = Middle, 3 = Lower)
Sex	Sex (Male, Female)
Age	Age in years
SibSp	Number of siblings (brother, sister, stepbrother, stepsister) / spouses (husband, wife) (mistresses and fiancés were ignored) aboard the Titanic
Parch	Number of parents (mother, father) /children aboard the Titanic
Ticket	Ticket number
Fare	Fare of the Ticket
Cabin	Cabin number
Embarked	Port of Embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

4. Exploratory Data Analysis (EDA)

Exploratory data analysis is the process of visualizing and analyzing data to extract insights. We need to summarize essential characteristics and trends in the dataset to understand the dataset better.

In this dataset, Rows represent passengers, and columns represent the features that describe the passengers, such as name, age, gender, etc.

I analyzed the dataset, including summary statistics, shapes of Datasets, Data types, Missing data, Feature analysis, Categorical data analysis, and numerical data analysis.

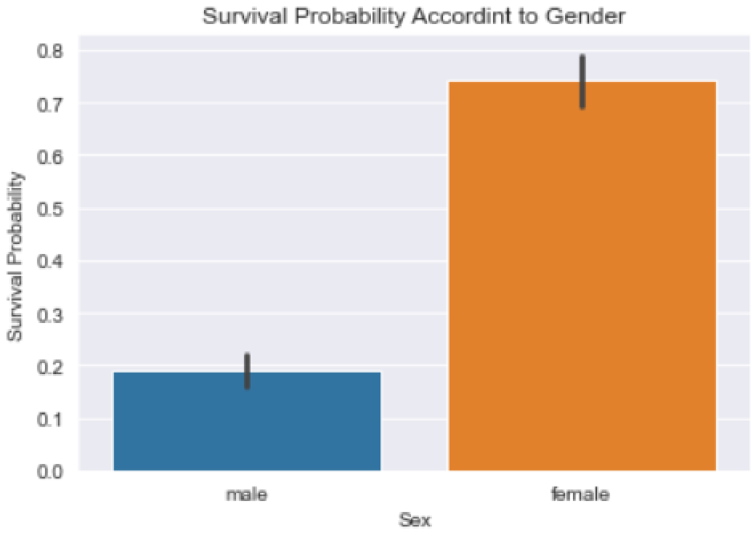
In this case, I split the whole dataset into categorical and numerical variables and Analyzed each feature separately to determine how it relates to survival. It is very important to select the appropriate visualization plots and distinguish between categorical and numerical variables. We can't calculate the average of a categorical variable like gender because an average can only be calculated for a numerical variable with a continuous distribution of values.

Categorical variables can be divided into two categories: Nominal (No particular order) and Ordinal (some ordered).

“Survived” (Binary), “Sex” (Binary), “Pclass” (Ordinal), and “Embarked” (Ordinal) are the categorical features of the dataset.
“SibSp”, “Parch”, “Age”, and “Fare” are the numerical features of the dataset.

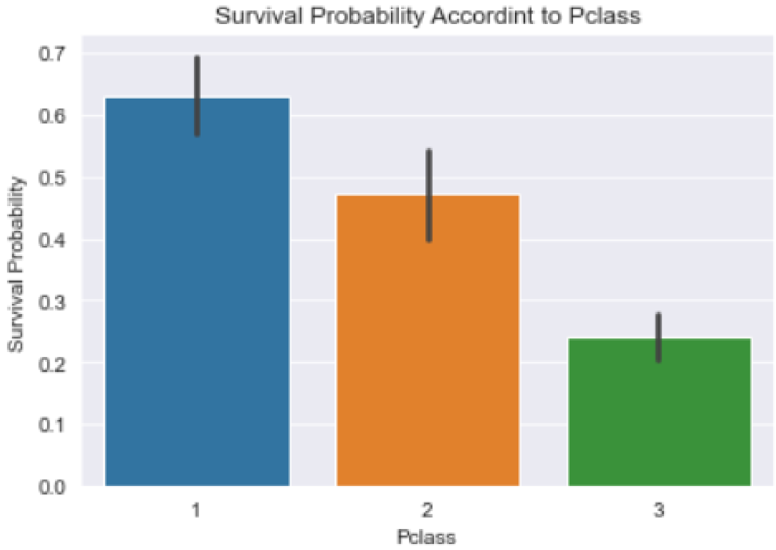
First of all, **I searched the outlier values of both the training and testing dataset and dropped them.** Then, I analyzed the features in this dataset individually and see how they correlate with survival probability using countplot, barplot, catplot, factorplot, FacetGrid, pointplot, heatmap, distplot, and kdeplot.

- **Sex vs Survival**



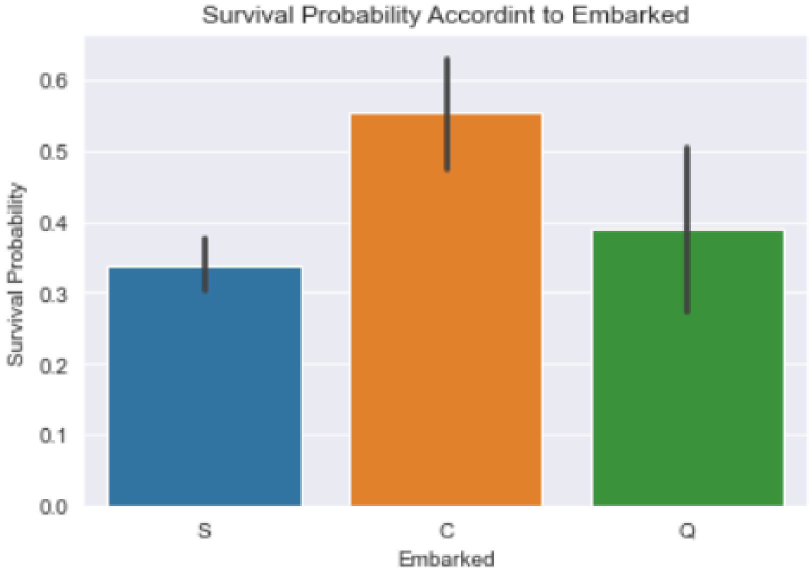
Female passengers are more likely to survive than male passengers.

- **Pclass vs Survival**



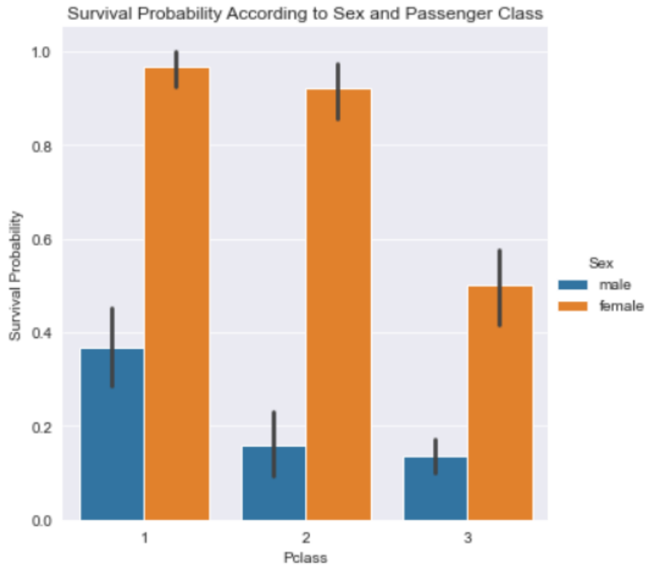
Passengers in the first class were more likely to survive than passengers in the second class, and passengers in the second class were more likely to survive than passengers in the third class. Because the first-class passengers may be rich, influential, and have high social status. During an evacuation, they have prioritized over the other passengers.

- **Embarked vs Survival**

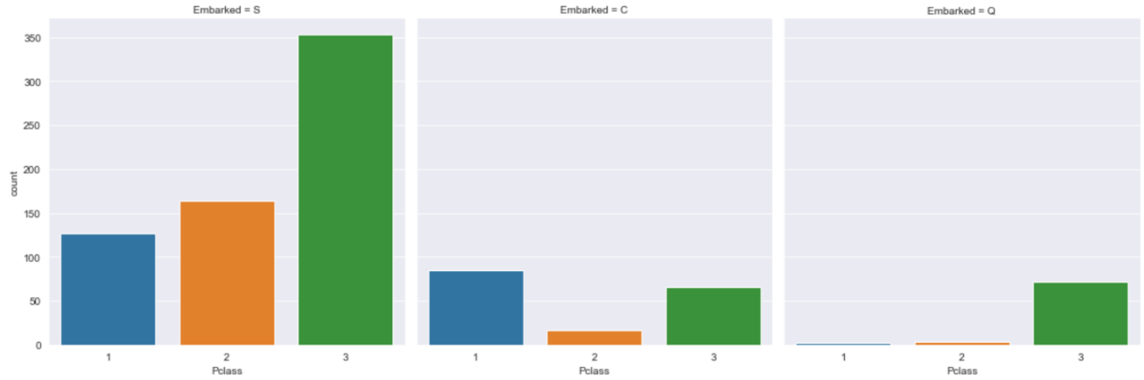


The graphs confirm, a passenger boarded from port S is more likely to survive than a passenger boarded from port Q and a passenger boarded from port Q is more likely to survive than a passenger boarded from port C.

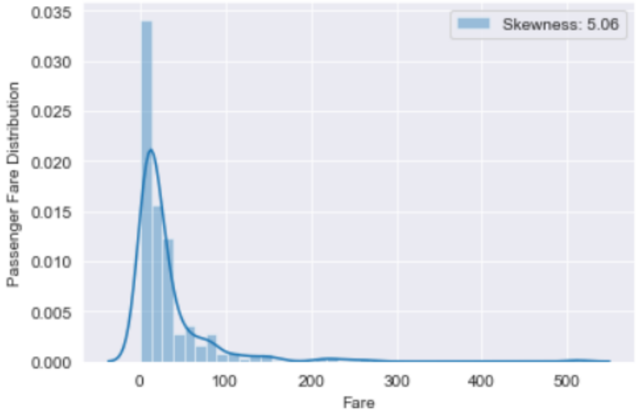
❖ Survived probability with Sex and Pclass



❖ Survived probability with Embarked and Pclass



❖ Fare



Fare seems to have a high positive skewness.

5. Data preprocessing

Data preprocessing is the essential process that uses for data mining. Because under the data preprocessing, we clean and prepare our dataset to train the model.

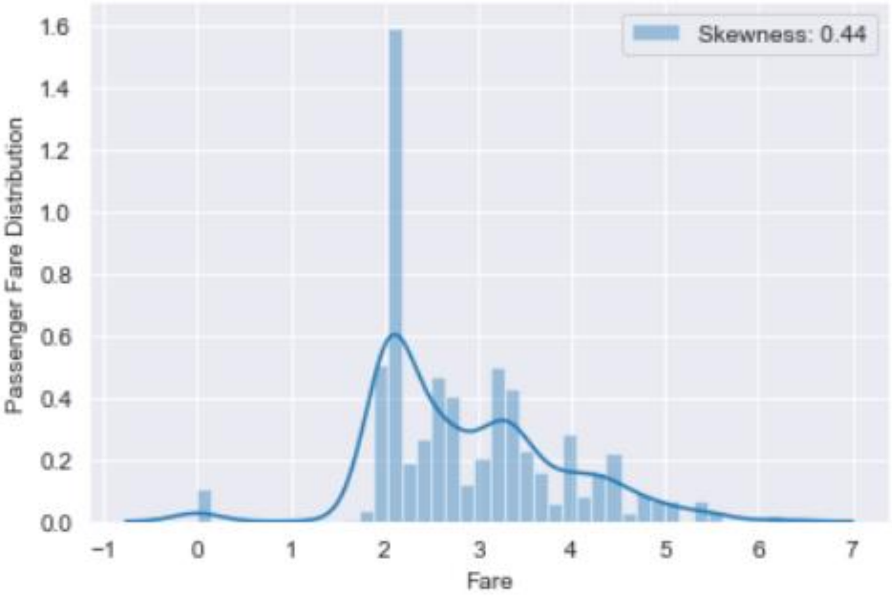
In my case, I checked and handle the duplicate data records, missing/null values, and data transformation. And also, I already handled the outlier values before starting the exploratory data analysis (EDA).

Duplicate data handle

There was no duplicate record founds from both train and test data sets.

Data transformation

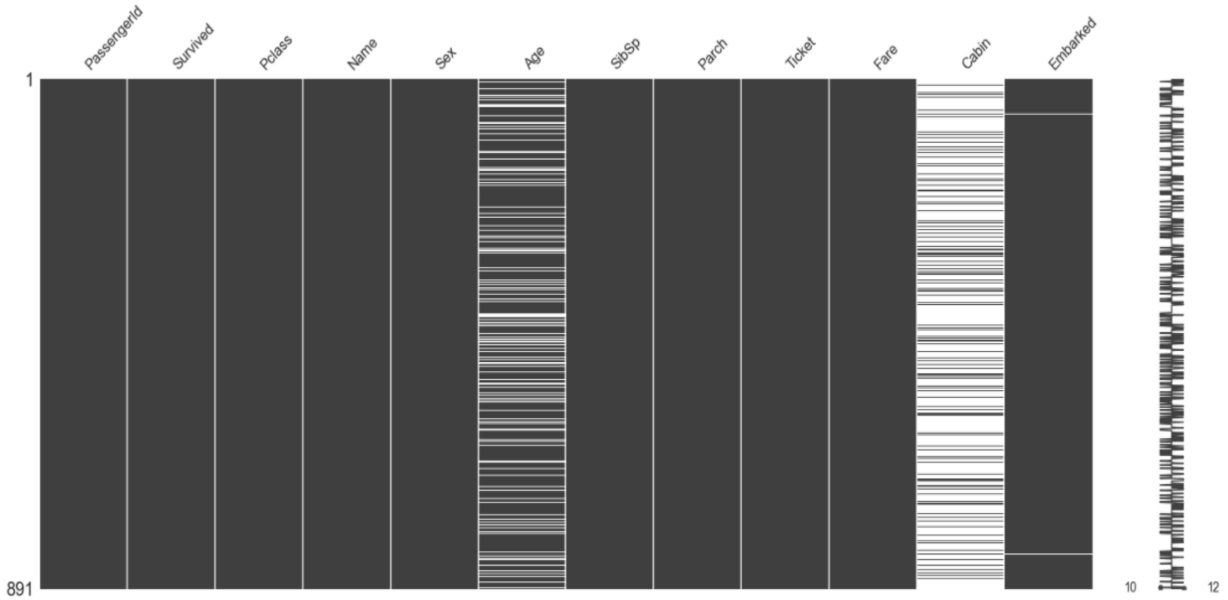
The passenger Fare column had a high skewness. Therefore, I applied log transformation to reduce the skewness from 5.06 to 0.44.



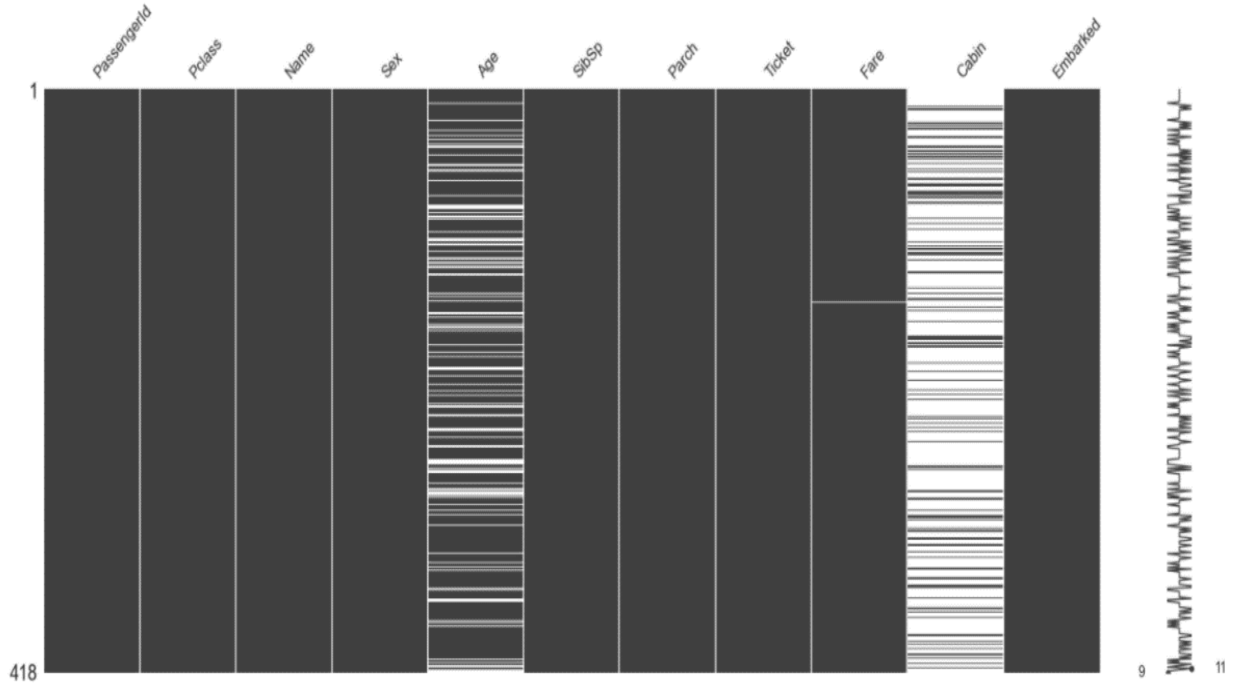
Missing values

There are two ways to handle the missing values. One way is to drop the missing values, and the other way is to fill the missing values.

Here, I found three available features with missing values of the **training dataset**, such as “Age”, “Cabin”, and “Embarked”.



And also found three features from the **testing dataset**, such as “Age”, “Cabin”, and “Fare”.



Fill missing values
“PassengerId” (not dependent), “Ticket” (not dependable), “Cabin” (there are many missing values) features are dropped from the training and testing datasets.

There were many missing values in the “Age” column of training and testing datasets.

Then, I filled the missing values of the “Age” column of the **training dataset** using its column average value of the **training dataset.**

Next, I filled the missing values of the “Age” column of the **testing dataset** using its column average value of the **testing dataset.**

There were two missing values in the “Embarked” column of the training dataset. I filled those values with the majority value of that column.

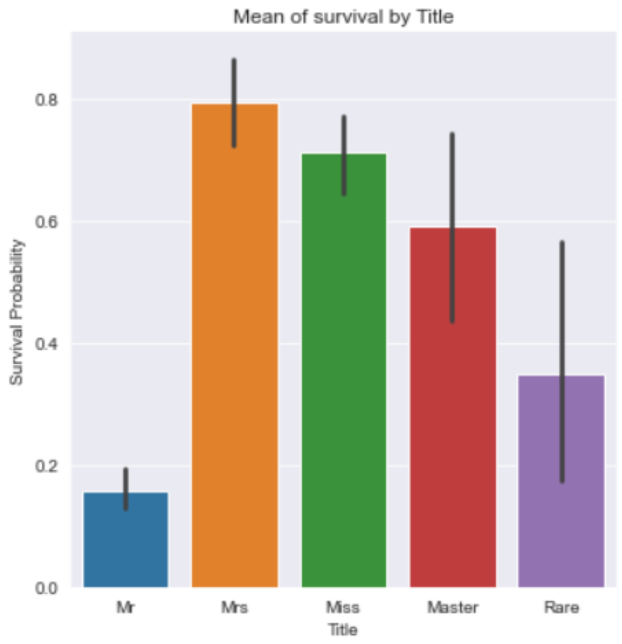
There was one missing value in the “Fare” column of the testing dataset, and I filled that value using its column average value.

6. Feature engineering

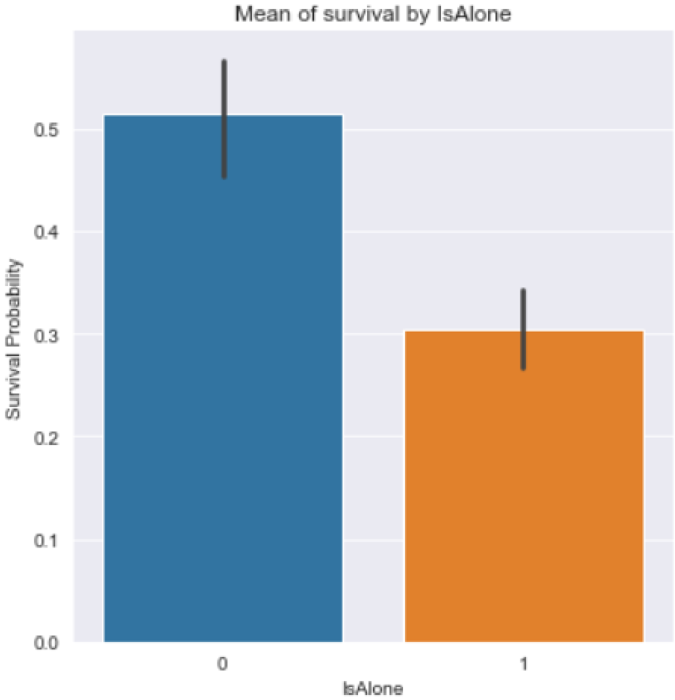
Feature engineering is the process of creating new features from existing features. It helps to better represent the underlying problem to the predictive models.

In my case, I created two new features that are “Title” and “IsAlone." The “Name” feature is used to create the “Title” feature, and “SibSp” and “Parch” features are used to create the “IsAlone” feature.

Title – Define the main five-category such as Mr, Mrs, Miss, Master, and Rare.



IsAlone – Whether the passenger is alone (1) or not alone (0)



7. **Feature encoding**

Most of machine learning algorithms can't handle string/categorical variables if we don't convert them into numerical values. And also, the performance of many algorithms is based on how categorical variables are encoded.

In here, I encoded two features that are "Sex", "Embarked" and "Title". I used the Label Encoder technique for "Sex" and "Embarked" features because they are ordinal. And I used the One Hot Encoder technique for the "Title" feature because it is nominal.

8. **Splitting Dataset**

Dataset is split into three parts which are X_train, y_train, and X_test. y_train is the survived column in our training set, and X_train is the rest of the other columns in the training set.

9. **Feature scaling**

Before creating a machine learning model, feature scaling is one of the most critical steps under preprocessing because machine-learning algorithms may assume larger values to be higher and smaller values to be lower, whatever of the unit of measurement. If feature scaling is not done, our predictions may go wrong.

There are some feature scaling techniques. Normalization (Min-Max) and Standardization are the most common feature scaling techniques among them.

In my case, I used the Normalization technique for feature scaling. This technique re-scales a feature or observation value with a distribution value between 0 and 1.

10. Create model

Scikit-learn is one of the most popular libraries for machine learning in Python, and that I also used it for this competition to create and train the model and make predictions.

There are three simple steps. Firstly, we need to instantiate our model, that is simply declaring a model and assigning it to a variable. Next, we need to fit the model to our training set, both the predictor variables and the response variable. Finally, we can use this model to make predictions on the test set.

I have chosen the **Random Forest Classifier** and **Gaussian Naive Bayes Classifier** to make predictions on the test set in this Titanic competition.

Training accuracy shows how well our model has learned from the training set. and cross-validation score shows how well our model is accurate.

11. Hyperparameter tuning

Hyperparameter tuning is the process of tuning the parameters of a model. When done the fitting the models to the training sets, then evaluated their accuracy at making predictions. Once the model is determined, I would also do hyperparameter tuning to avoid over-fitting, under-fitting, and further boost the model's performance. **GridSearchCV, RandomSearchCV,** and **Bayesian Optimization** are the most common hyperparameter tunning techniques.

Here I used GridSearchCV to tune the parameters of the Random Forest Classifier and Gaussian Naïve Bayes Classifier. Then, I found the best parameters to get better accuracy from Random Forest Classifier.

```
{criterion='gini', min_samples_leaf=1, min_samples_split=10,
 n_estimators=100, random_state=1}
```

and Gaussian Naive Bayes Classifier

```
{'var_smoothing': 0.33}
```

12. Model evaluation

Model evaluation is a step in the model-building process. It is the step that is decided whether the model performs better. Following metrics are used to evaluate the classification model.

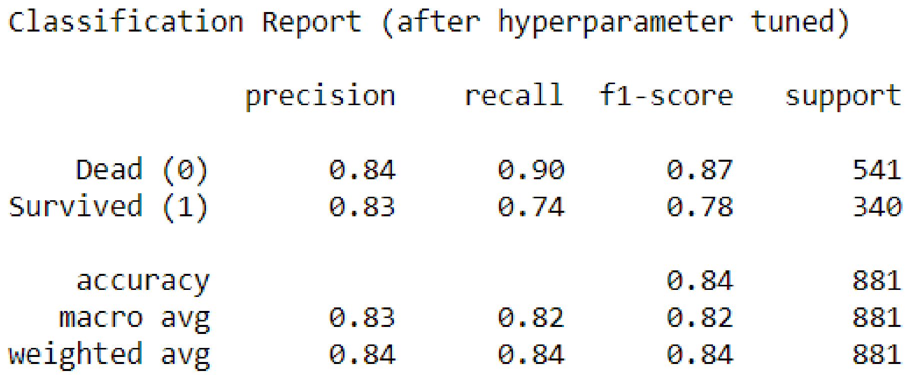
- Confusion matrix
- Accuracy
- Classification error
- Recall
- Precision

- F1-score
- Classification report
- Precision-Recall Curve
- ROC AUC Curve and Score

Once both models have been trained, the next step is to assess the performance of these models and select the one which has the highest prediction accuracy. Accuracy score is defined as the percentage of correct predictions for the test data. The F-score is computed with the harmonic mean of precision and recall.

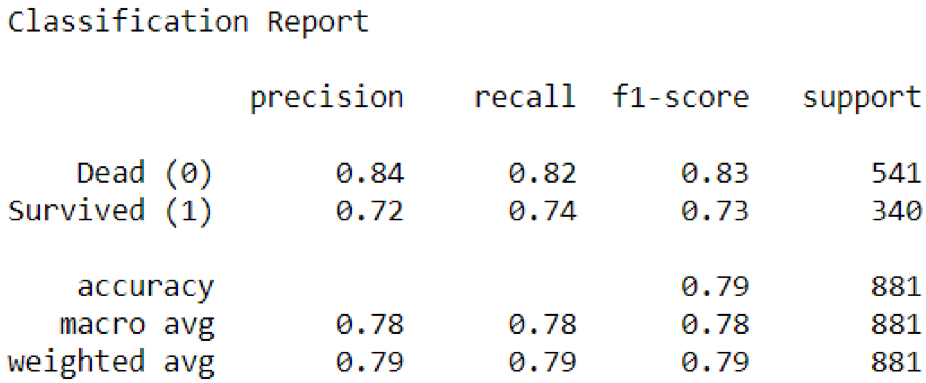
In my case, I used the Cross-validation score, Confusion matrix, Accuracy score, Precision score, Recall score, F1-score, Classification report, precision-recall curves, and ROC AUC curves to evaluate both classification models.

Classification Report of Random Forest Classifier



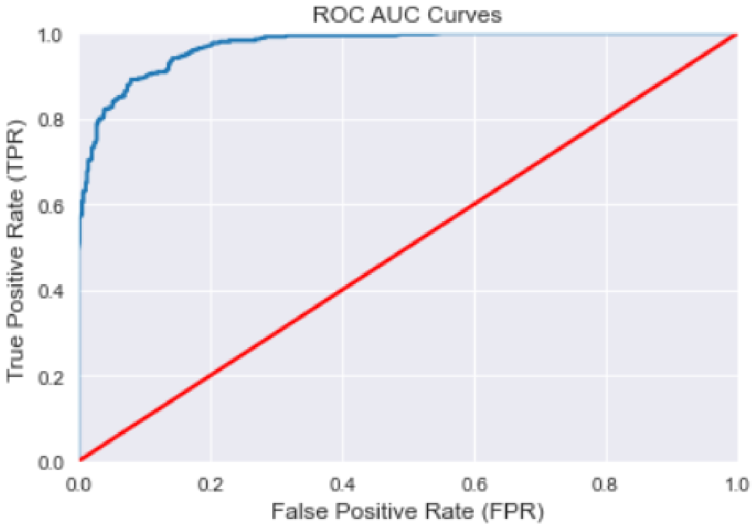
```
Classification Report (after hyperparameter tuned)

              precision    recall  f1-score   support

    Dead (0)       0.84      0.90      0.87       541
Survived (1)       0.83      0.74      0.78       340

    accuracy                           0.84       881
   macro avg       0.83      0.82      0.82       881
weighted avg       0.84      0.84      0.84       881
```

Classification Report of Gaussian Naïve Bayes Classifier

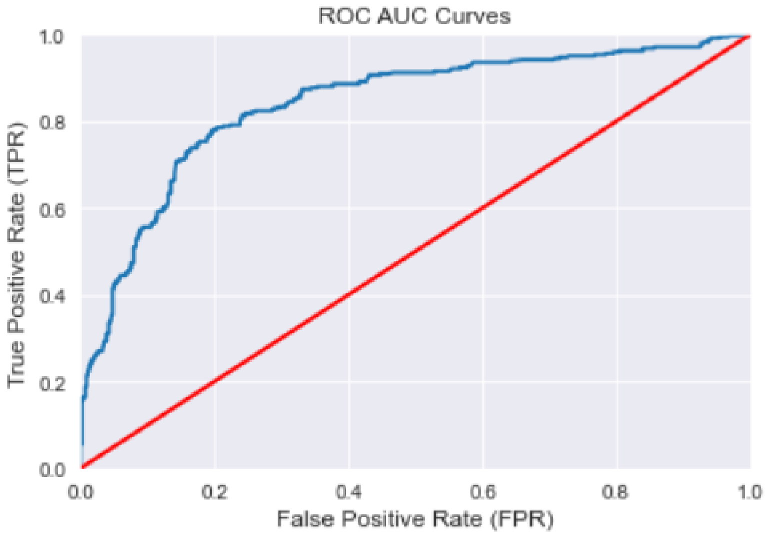


```
Classification Report

              precision    recall  f1-score   support

    Dead (0)       0.84      0.82      0.83       541
Survived (1)       0.72      0.74      0.73       340

    accuracy                           0.79       881
   macro avg       0.78      0.78      0.78       881
weighted avg       0.79      0.79      0.79       881
```

ROC AUC Curve of Random Forest Classifier



ROC AUC Curve of Gaussian Naïve Bayes Classifier



13. Final Submissions

After done every process, I predicted the result of passengers, whether they survive or not, According to their details of the testing set.
Then, I created my submission as a data frame including only "PassengerId" and "Survived" and export it as a CSV file. Next, I uploaded it to the Kaggle Titanic data mining competition. At the very bottom, I attached screenshots of the results of both classifiers that I developed and submitted to the Kaggle.

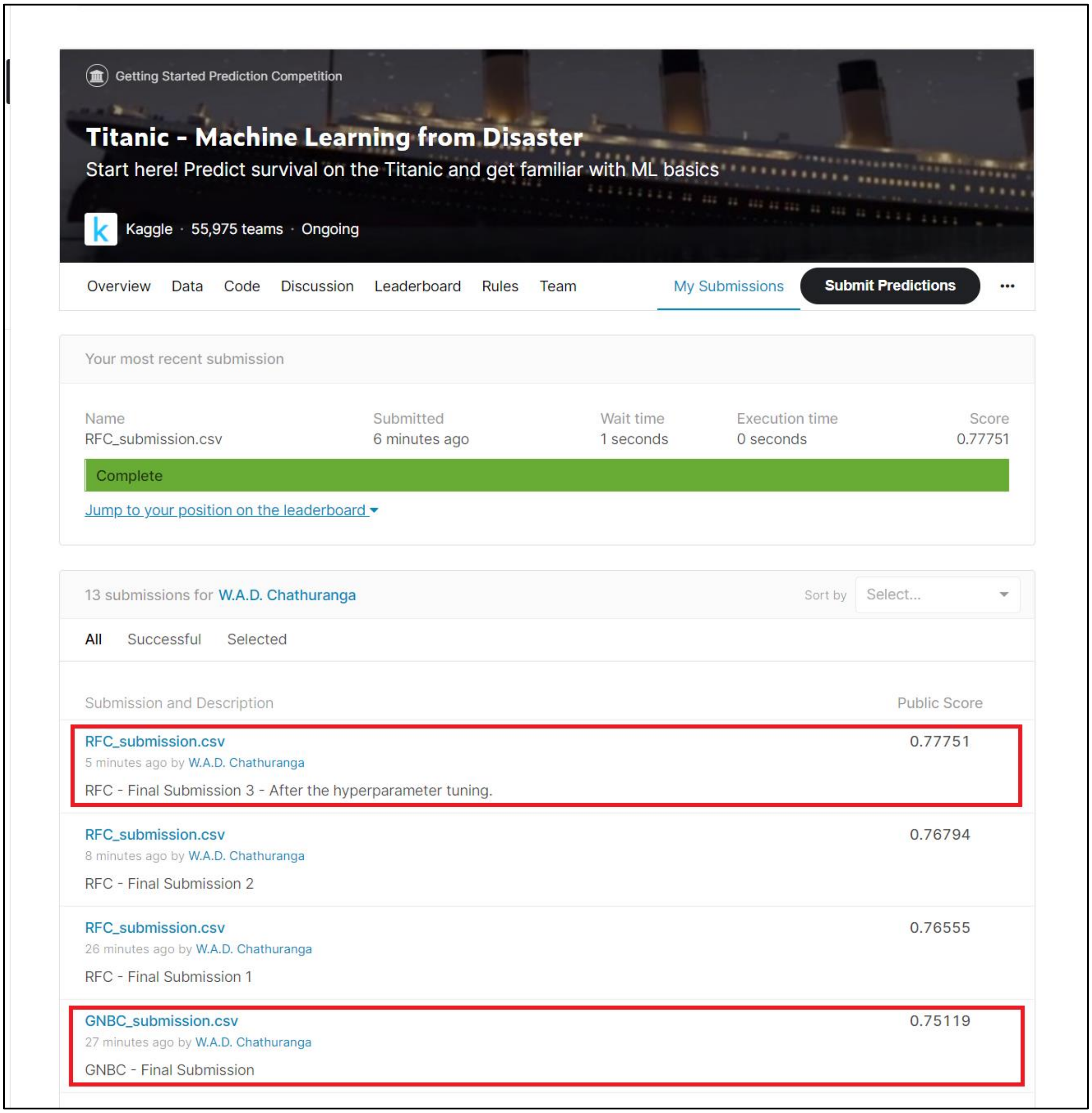
Finally, I achieved a **0.77751** predicted score from the **Random Forest Model** and **0.05119** from **Gaussian Naive Bayes Model.**

14. Improve the model accuracy

I got an accuracy score for both my classification models. Random Forest Classifier has a 0.77511 accuracy score, and the Gaussian Naive Bayes Classifier model has a 0.75119 accuracy score. Overall, both are good accuracy scores, but it should be more increase. If we need to increase our accuracy, we should go through the following points.

- The ticket and cabin columns should analyze further rather than dropping them.
- Come up with new features in feature engineering than what I have already done.
- Less important features should remove to reduce overfitting in the model.
- Ensemble modeling is a more advanced technique whereby you combine prediction results from multiple machine learning models.

Source Code: https://github.com/wadchathuranga/Data_Mining_Project-Kaggle-TITANIC_MachineLearningCompetition



Titanic - Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics

Kaggle · 55,975 teams · Ongoing

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
RFC_submission.csv	6 minutes ago	1 seconds	0 seconds	0.77751

Complete

Jump to your position on the leaderboard

13 submissions for W.A.D. Chathuranga

Submission and Description	Public Score
RFC_submission.csv 5 minutes ago by W.A.D. Chathuranga RFC - Final Submission 3 - After the hyperparameter tuning.	0.77751
RFC_submission.csv 8 minutes ago by W.A.D. Chathuranga RFC - Final Submission 2	0.76794
RFC_submission.csv 26 minutes ago by W.A.D. Chathuranga RFC - Final Submission 1	0.76555
GNBC_submission.csv 27 minutes ago by W.A.D. Chathuranga GNBC - Final Submission	0.75119