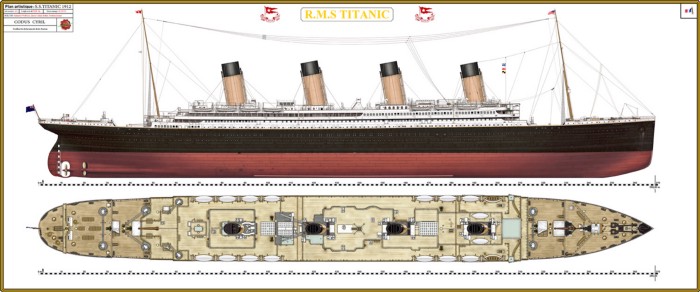
# TITANIC PROJECT



**The article contains the following sub-topics**:

1.      Problem Definition  
2.      Data Analysis  
3.      EDA Concluding Remarks  
4.      Pre-processing Pipeline  
5.      Building Machine Learning Models  
6.     Concluding Remarks

# Problem Definition

The Titanic Problem is based on the sinking of the ‘Unsinkable’ ship Titanic in early 1912. It gives you information about multiple people like their ages, sexes, sibling counts, embankment points, and whether or not they survived the disaster. Based on these features, you have to predict if an arbitrary passenger on Titanic would survive the sinking or not.

**Attribute Information**

* **Survived**: Outcome of survival (0 = No; 1 = Yes)
* **Pclass**: Socio-economic class (1 = Upper class; 2 = Middle class; 3 = Lower class)
* **Name**: Name of passenger
* **Sex**: Sex of the passenger
* **Age**: Age of the passenger (Some entries contain NaN)
* **SibSp**: Number of siblings and spouses of the passenger aboard
* **Parch**: Number of parents and children of the passenger aboard
* **Ticket**: Ticket number of the passenger
* **Fare**: Fare paid by the passenger
* **Cabin** Cabin number of the passenger (Some entries contain NaN)
* **Embarked**: Port of embarkation of the passenger (C = Cherbourg; Q = Queenstown; S = Southampton)

# Objectives

The objectives of this project are as follows

1. To experiment with different classification methods to see which gives the highest accuracy

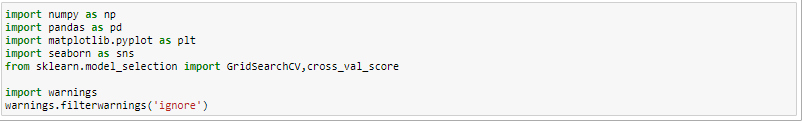
2. To determine which features are most affecting to predict if an arbitrary passenger on Titanic would survive the sinking or not.

3. This allows you to practice with hyper parameter tuning on e.g. decision tree algorithms looking at the ROC curve and the AUC value.

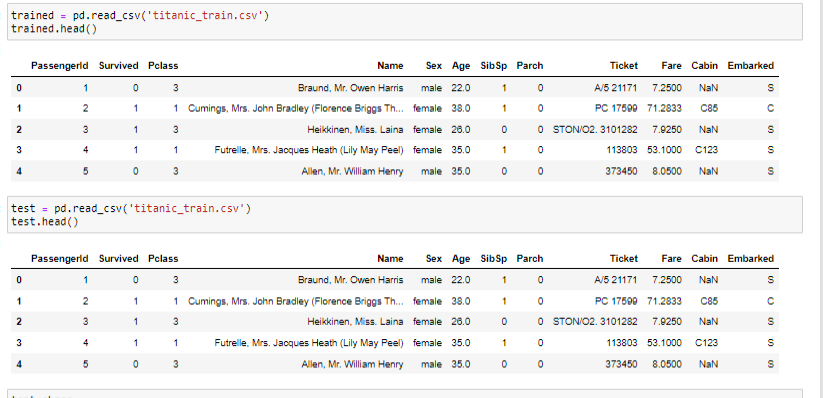
# Data Analysis

Exploratory data analysis is a very important step to get an idea of what the dataset is like and what kind of modifications we need to make. I started with importing the libraries and modules and reading the data I’ll be using into pandas dataframe.

**Importing Libraries**

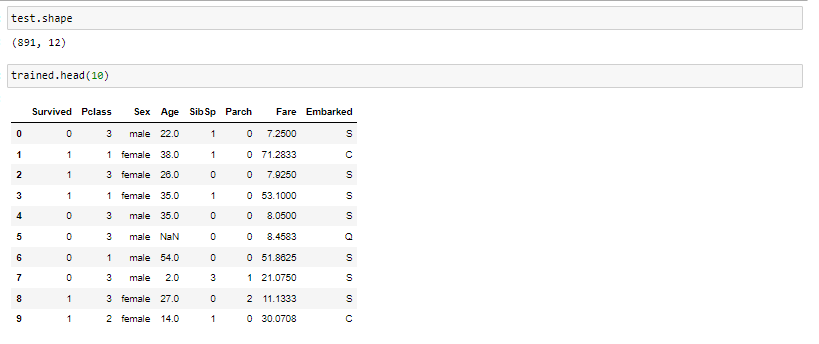


**Reading the Data**



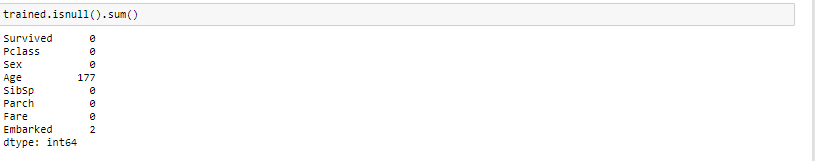
**Analyzing data**

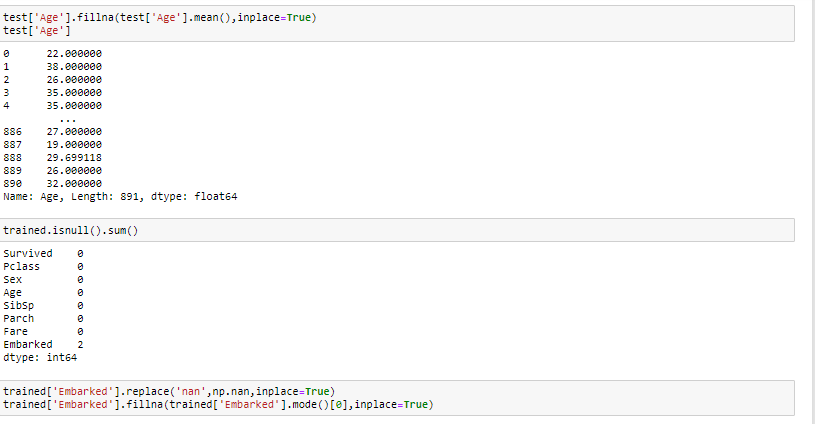
**As we have loaded dataset, let’s see how many records are present. Also we can check number of rows and columns.**



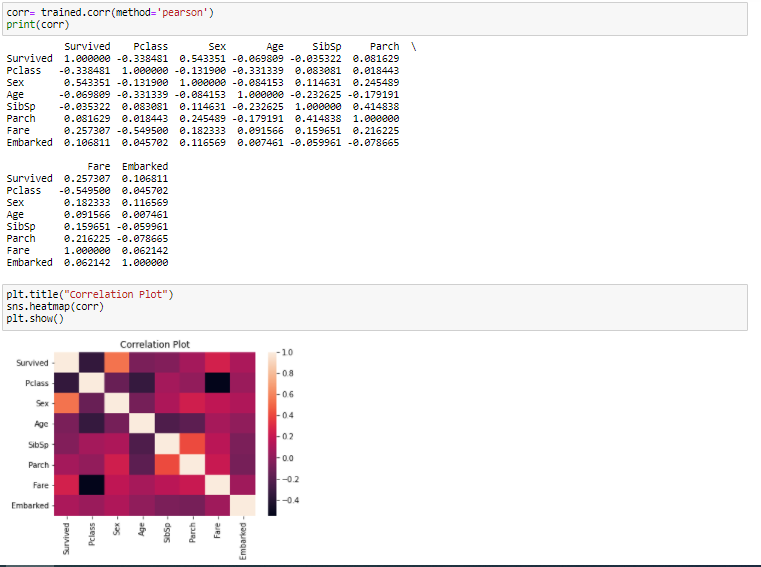
**Checking missing values**

There are missing values in ‘Age’ and ‘Embarked’ column. So we have handled those records. We can replace ‘Age’ column with mean value of that column and ‘Embarked’ column with mode value of that column.



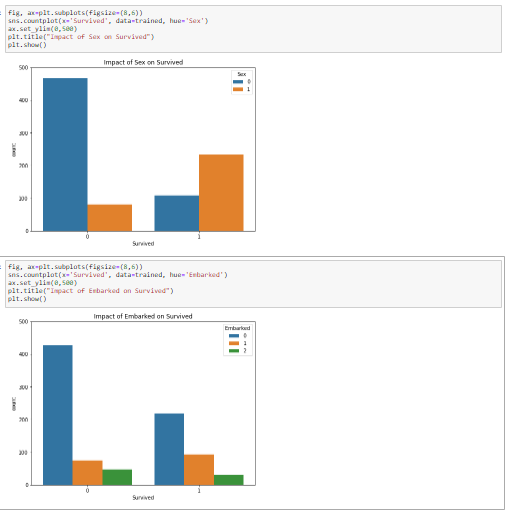


**Checking the correlation between columns**



All features are affecting ‘Survived’ column which is our target. Hence we have to train our model with all columns. It means we cannot drop any column while analyzing data and training our model.

# Data Visualization



I have used count plot here to visualize data. In 1st graph I have visualize ‘survived’ column with ‘sex’ column. In this I have observed that target column survived has 2 values (0 or 1) and it is affecting to sex column (F or M). In 2nd graph, I have compared ‘survived’ column with ‘Embarked’ column.

In 1st graph, I can see people who have not survived are male as compare to female. In case of people who have survived are more number in females and less in males. So we can say that survival ratio is more in female and less in male.

**Changing categorical values**

Changing sex column values as below:

0 for male and 1 for female

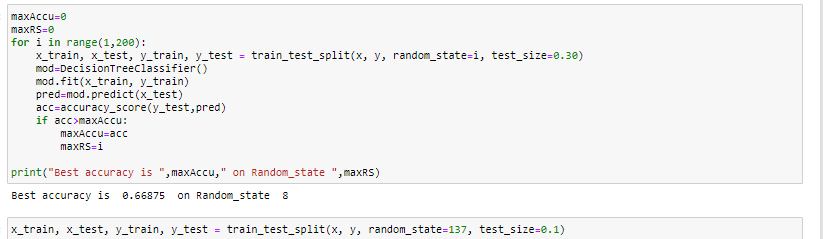
****

Changing Embarked column values as below:

C(Cherbourg)=1 and Q(Queenstown)=2 and S(Southampton)=0

****

**Train and test sets split**



**Validation and Model Selection**

In this part I trained several classification algorithms to find the best one for the dataset I used. Now we will train several Machine Learning models and compare their results. Note that because the dataset does not provide labels for their testing-set, we need to use the predictions on the training set to compare the algorithms with each other.

# Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).



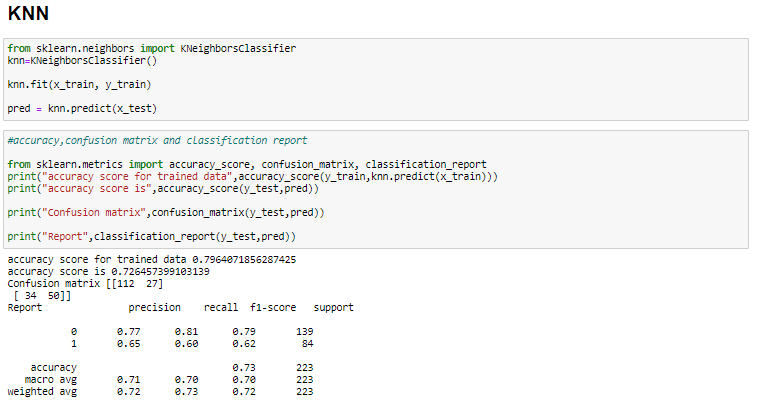
**SVC**

“Support Vector Machine” (SVM) is a supervised [machine learning algorithm](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2?utm_source=blog&utm_medium=understandingsupportvectormachinearticle) which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

****

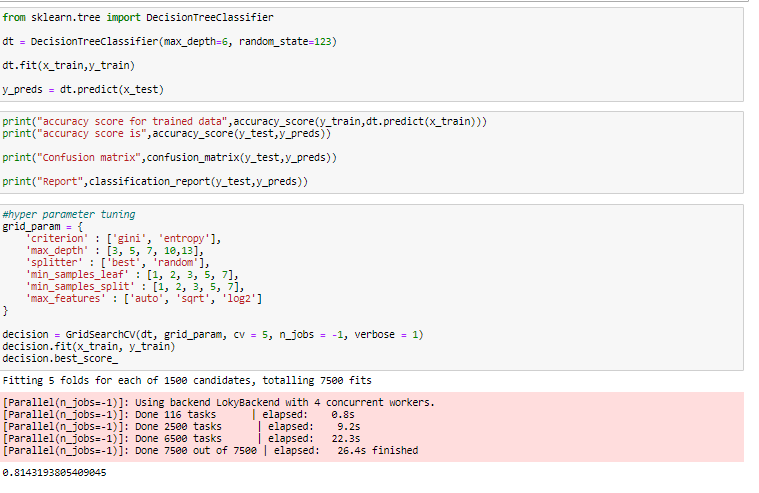
**KNN**

K nearest neighbors is a simple **algorithm** that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). **KNN** has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.

****

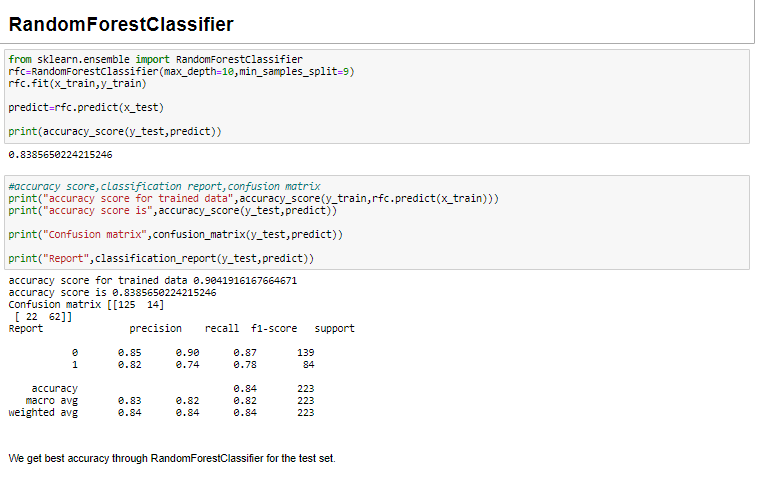
# Decision Tree Classifier

Decision tree a tree like structure whereby an internal node represents an attribute, a branch represents a decision rule, and the leaf nodes represent an outcome. This works by splitting the data into separate partitions according to an attribute selection measure, which in this case is the Gini index (although we can change this to information gain if we wanted). This essentially means that we each split aims to reduce Gini impurity which measures how impure a node is according to incorrectly classified results.



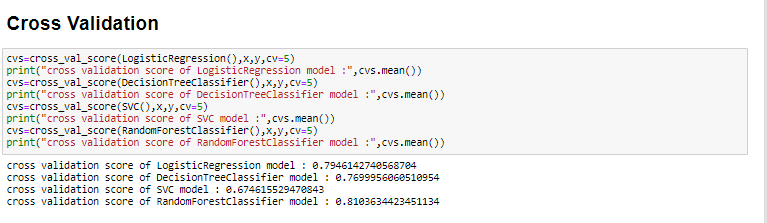
# Random Forest Classifier

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting.

****

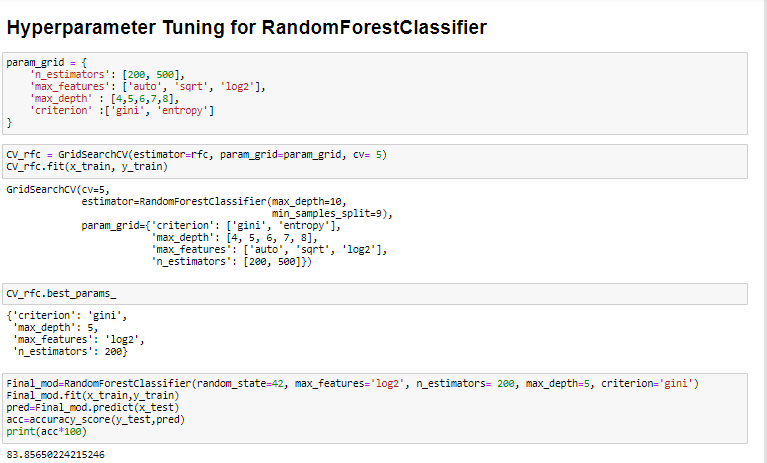
# Cross Validation Score

Cross-validation is a statistical method used to estimate the skill of machine learning models. That k-fold cross validation is a procedure used to estimate the skill of the model on new data. After comparing accuracy score and cross validation score, RFC is the best model which gives more accuracy.



**Hyper parameter Tuning**

In machine learning, hyper parameter optimization or tuning is the problem of choosing a set of optimal hyper parameters for a learning algorithm. A hyper parameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.



**Final Model**



# Conclusion

In this project, I used K-Nearest Neighbors, SVC, Logistic Regression with polynomial features, Decision Tree, and Random Forest. With the roc\_auc\_score in Scikit-learn I calculated the AUC score for each model. Also using cross\_val\_score method I found AUC score using cross validation method.

If we compare the cross validation scores and recall results of all models we can see that the best results were obtained with **RFC model.**