**Titanic Survival Prediction**

Problem Definition:

The RMS Titanic was a British passenger liner that sank in the North Atlantic Ocean in the early morning hours of 15 April 1912, after it collided with an iceberg during its maiden voyage from Southampton to New York City. There were an estimated 2,224 passengers and crew aboard the ship, and more than 1,500 died, making it one of the deadliest commercial peacetime maritime disasters in modern history. It gives you information about multiple people like their ages, sexes, sibling counts, embarkment 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.

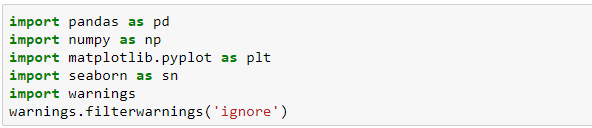
|  |  |  |
| --- | --- | --- |
| Variable | Definition | Key |
| Survival | Survival | 0 = No, 1 = Yes |
| pclass | Ticket class | 1 = First, 2 = Second, 3 = Third |
| Sex | Gender of Passenger |  |
| Age | Age in years |  |
| sibsp | Number of siblings/ spouses aboard the Titanic |  |
| parch | Number of parents/children aboard the Titanic |  |
| ticket | Ticket number |  |
| fare | Passenger fare |  |
| Cabin | Cabin Number |  |
| Embarked | Port of Embarkation | C = Cherbourg, Q = Queenstown, S = Southampton |

**Understanding the Data**

* Each row represents the data of 1 passenger.
* Column represents the features of the dataset.

**Data Analysis**

**Step 1: Importing the basic libraries**

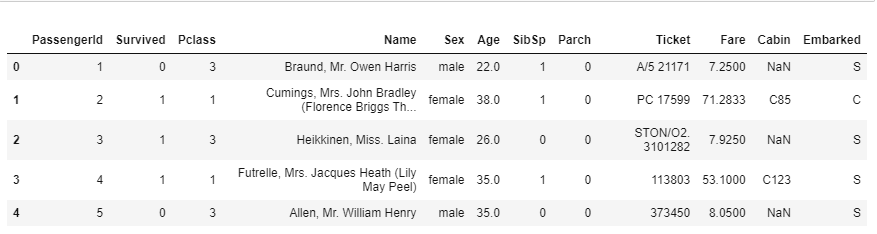


**Step 2: Download and Load the Dataset**



To load the dataset into memory, we used the read \_ csv function from the Pandas library. The data will be loaded as a Pandas data frame.

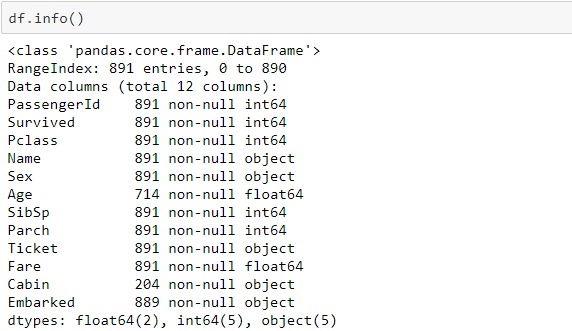
**Check the dataset using head () function**



The above dataset represent the snapshot of the dataset that we will be working on. Using df.shape I found out that the total number of rows are 891 and Columns are 12.

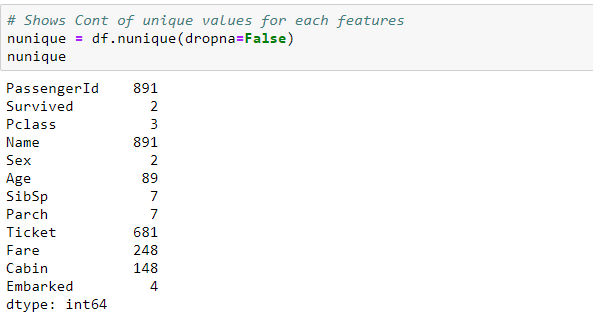
**Step 3: Data exploration**

In this section we will try to draw insights from the Data, and get familiar with it, so we can create more efficient models.



The first thing that caught my eye when running the .info function is that not every passenger has a value for cabin and age. **Dataset has 891 examples and 11 features + the target variable (survived).** 2 of the features are floats, 5 are integers and 5 are objects.

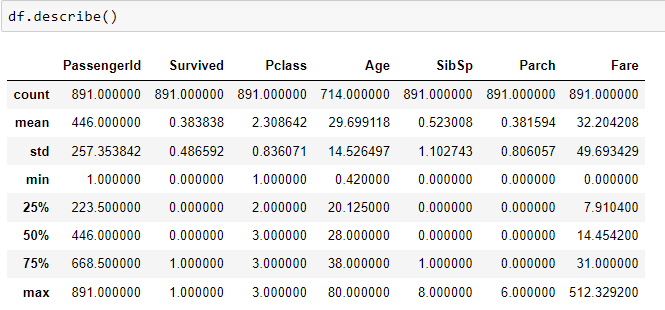
As you can see, we have 891 entries in total but some of the columns have less than 891 entries so that means we have missing values in these columns namely Age, Cabin & Embarked. So, we have to pre-process our data first before training our ml model.



**Inferences:** The above nunique function is used to find out the unique values present in each column.

1. We can clearly see that Passenger Id and Name contain 891 different entries
2. Survived and Sex Column has only two possible outcomes.

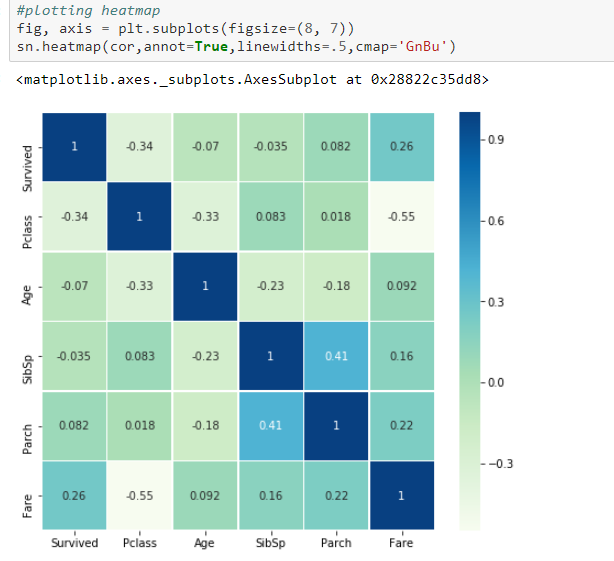
**Using Describe Method**



As 1 in survived column indicates person survived and 0 means person died so by looking at the mean in survived column, we can say that only 38% approx. survived the sinking. We can also see that the passenger ages range from 0.4-80 value. The average age of survivors is 28, so young people tend to survive more.

**Heatmap:**

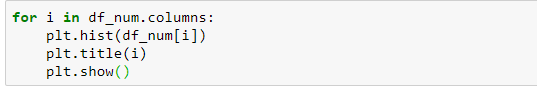
Now we will analyse our data to see which variables are actually important to predict the value of the target variable. Hence, we are going to plot a heat map to see the correlation between the parameters and the target variable (Survived).

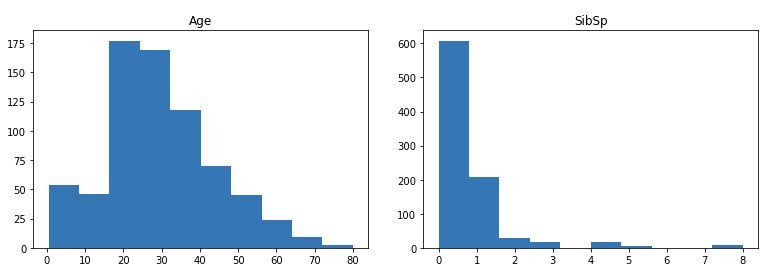


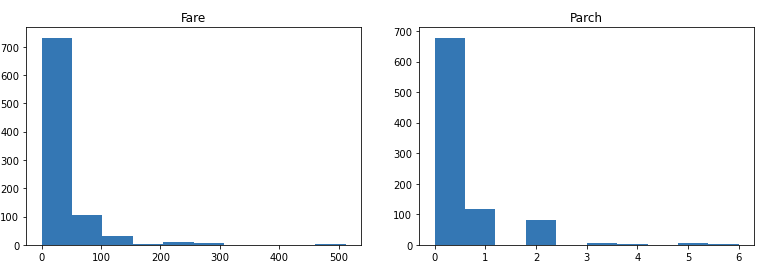
Now see the decimal values in the above colour 2D matrix. These values are the correlation values. Just compare the survived column with the rest of the columns. The lighter the colour is the more correlated the value is. Let’s compare the Survived with Sibsp you’re getting the value -0.035. It means that SibSp is not correlated to Survived.  Then Parch has a 0.082 value which shows very little correlation. Then Age, again no correlation. In the end, we have Fare whose value of correlation with the Survived variable is 0.26 which shows that the more the fare is, the more are the chances of survival.

**Conclusion:** But it does not mean that the other features are useless. We’ll explore more about them later.

**Plotting of numerical data**

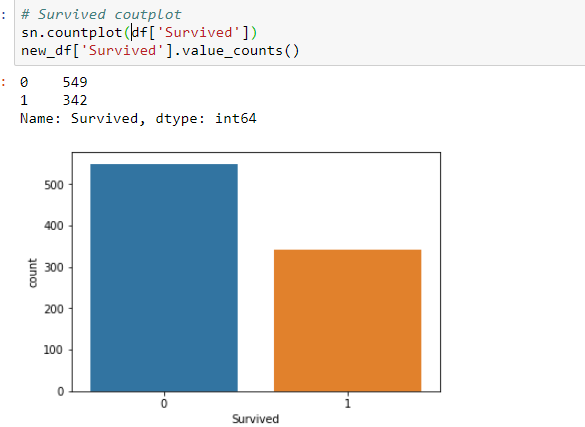


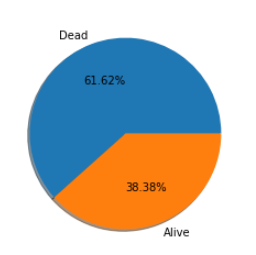




As we can see that most of the distribution are scattered except age which is showing normal distribution which may be considered for normalizing later.

Moving on, now we will understand all the features one by one. We’ll visualize the impact of each feature on the target variable.



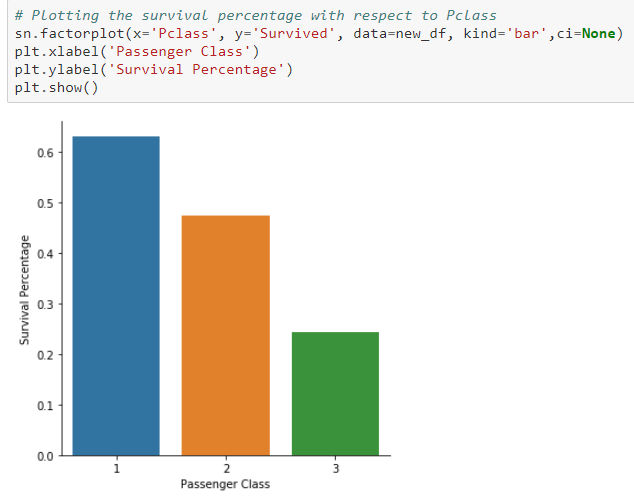


There are almost 891 out of which 549 died.

 Most of the people died in the shipwreck, only around 342 people survived.

If we consider approximately then ration of death to alive is 6 to 4 .

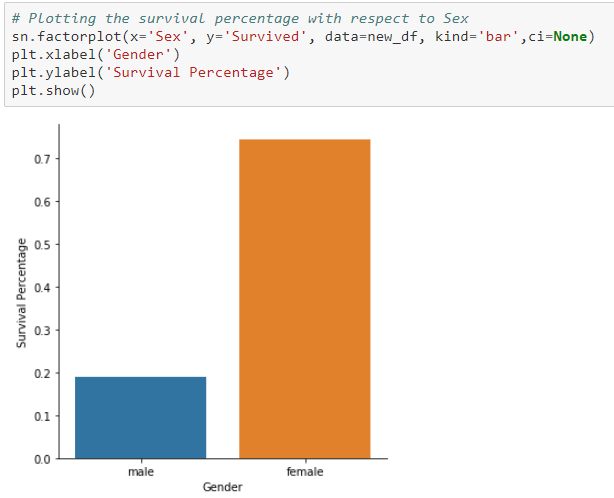
**Survivorship by Passenger Class**



The plot above shows distribution of survivorship based on passenger class. First class had the highest survivorship density, followed by second class, then third class. This could possibly be due to the much higher density of males in third class than females.

Here we see clearly, that Pclass is contributing to a person’s chance of survival, especially if this person is in class 1.

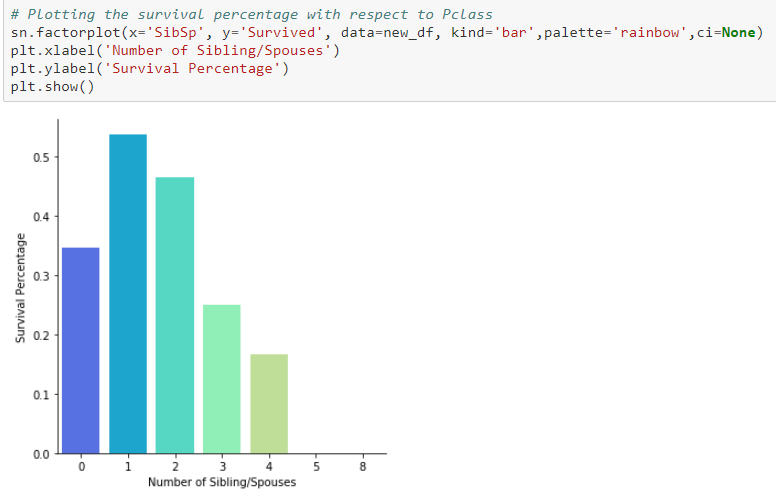
**Survivorship by Sex**



You can see from the above graph it’s quite obvious to say that man has less chances of survival over females.

Most of the women survived, and the majority of the male died in the shipwreck.

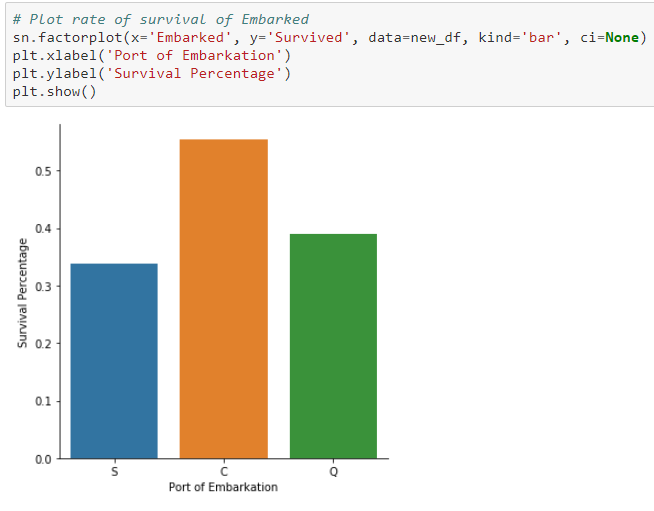
**Survivorship by Number of Siblings**



The above plot shows the relationship between number of siblings/spouses each person has and survival percentage for them.

It is observed that the passenger having 0,1 or 2 siblings have higher chances of survival. - Having sibling greater than 4 have very less probability of survival.

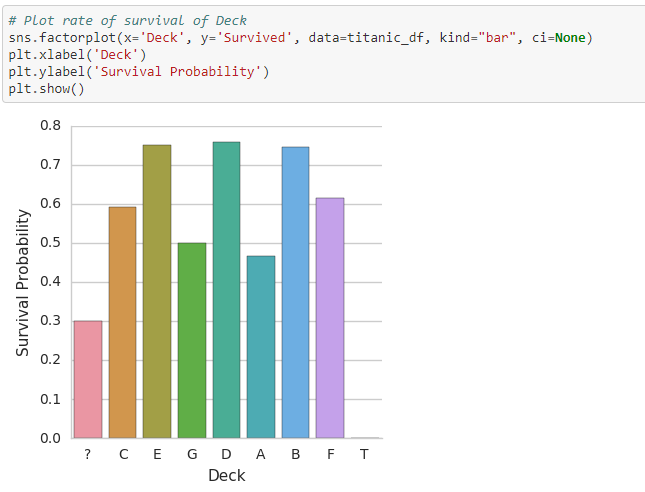
### Survivorship by Port of Embarkation

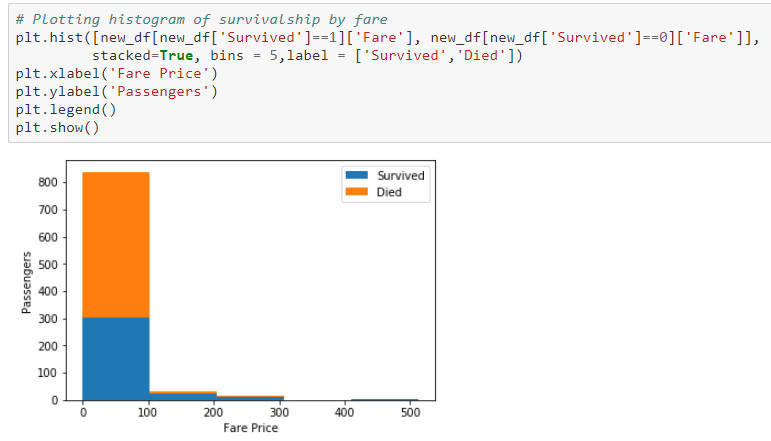


The above bar plot shows survivorship based on Port of Embarkation. Earlier we saw that a majority of passengers embarked from Southampton. Those passengers originating from Southampton also had the least likely probability of survival. Maybe if someone was from “Cherbourg” had a higher chance of surviving.

However, how much correlation is there with Port of Embarkation with survivorship? Or is it coincidence?

**Survivorship by Deck and Fare Price**





**Deck**: The above chart shows that the deck a passenger was assigned didn't have a significant effect of survival. However, notice that the passengers with no known decks had significantly lower survival rates.

**Fare**: Although it is difficult to see; the above histogram shows that those who paid a higher fare had a higher probability of survival. That is no surprise; as fare price directly relates with passenger class.

**EDA Concluding Remark:**

It appears that the greatest chance of surviving the titanic is dependent upon two main factors; sex and passenger class (Pclass). Age also shows it could be a factor in survivorship, but only when age is simply child vs. adult. Also, as we saw far above in the .info function, not all passengers in the dataset have an Age. It would be quite convoluted and suspicious to make concrete analysis based on a passenger property that is not complete.

Thus, it's good to be female or it's good to be a first-class passenger. But it's even better to be a first-class passenger **and** a child.

That’s all about data exploratory analysis. Now we have a good idea about our data. And one more thing to notice here is there are some features which have nothing to do with survival probability like PassengerId, Ticket number, Cabin number and also the name of the passenger. So, we can safely drop them before building our ml model. Moreover, we also need to handle missing values. So, all of these tasks come under Data Pre-processing.

**Step 4: Feature Engineering**

Feature engineering is the process of using domain knowledge of the data to create features (**feature vectors**) that make machine learning algorithms work.

We saw that our *ticket*and *cabin* data don’t really make sense to us, and this might hinder the performance of our model, so we have to simplify some of this data with feature engineering.

We can also use the Name feature to extract the Titles from the Name, so that we can build a new feature out of that.

Let’s start with **Name** Column

The 'Name' column in the titanic data frame appears to give **every** single passenger a title of some kind. This 'title' may become important later on, so let's create a new column that just holds a passenger's title.

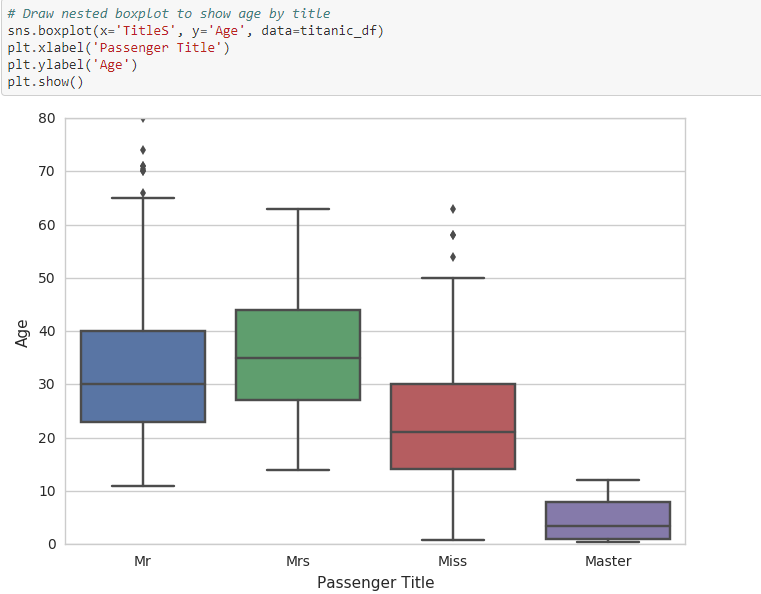


Now that the titles derived from the passenger names have been printed and plotted; we can see there's quite a bit of them. Some of them are standard types of titles; but some are obscure and few are rare.



Now the titles have been converted into standard format into (Mr, Miss, Mrs, Master).  
It looks like this is pretty much these titles could be the only data that could possibly be used to estimate the age of a passenger.

 Let's check and see if title has something to do with **Age** feature.



We can see how well the titles predict age by printing a box plot to summarize the features of quantitative variables (outliers, max/min, upper/lower quartiles, and median).

Now the passengers have been isolated into the four common titles of sex and age; Mr, Mrs, Miss, Master. So now, even though we do not have all the values filled in the 'Age' column; we do have some sort of column that can be used to predict age. So, I want to create another column ['MWC'] that dictates whether a passenger is a man, woman, or child.

Finding the men and women is simple. All those passengers with 'Mr' or 'Mrs' as their title would be considered Men or Women during the Titanic era. Finding male children is also simple; those with the title of 'Master' are male children.

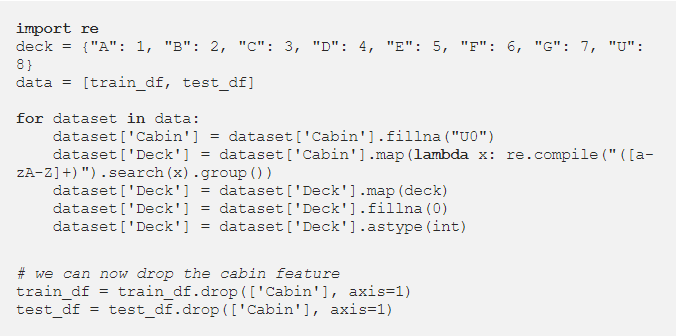
Finding female children is the difficult part; as the title of 'Miss' has quite the spread. In order to do the best attempt at classifying these Misses into Women or Children, this requires a bit of research into Victorian ages. Most historical research related to The Titanic describe children as those 14 years of age and younger. So, we will use that age. Notice, that this is very close to the lower quartile of the 'Miss' category in the box plot above.



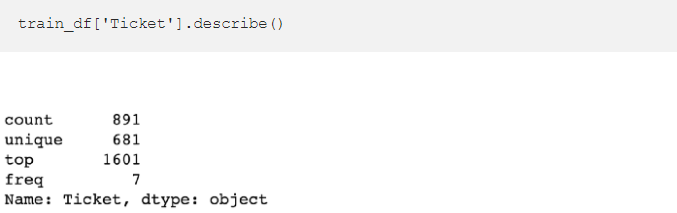
So now we have done the bulk of the work in dividing the passengers into the category of Man, Woman, or Child. Every title has been accounted for except for some still recorded as 'Miss'. These are the female passengers with the title of Miss who have an unknown Age.

We have to deal with Cabin (687), Embarked (2) and Age (177). First, I thought, we have to delete the ‘Cabin’ variable but then I found something interesting. A cabin number looks like ‘C123’ and the **letter refers to the deck**.

Therefore, we’re going to extract these and create a new feature, that contains a person’s deck. Afterwards we will convert the feature into a numeric variable. The missing values will be converted to zero. In the picture below you can see the actual decks of the titanic, ranging from A to G.



**TICKET**



Since the Ticket attribute has 681 unique tickets, it will be a bit tricky to convert them into useful categories. So, we will drop it from the dataset.

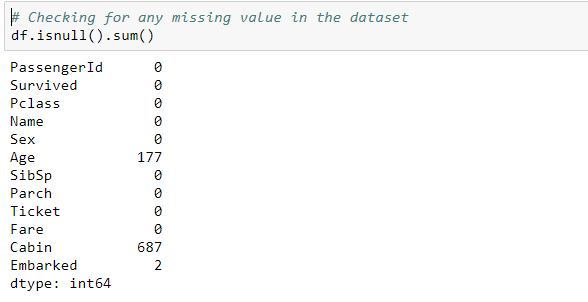
**Step 5: Data Pre-processing for the model**

Data pre-processing involves transforming raw data to well-formed data sets so that data mining analytics can be applied.

Data pre-processing involves both data validation and data imputation. The goal of data validation is to assess whether the data in question is both complete and accurate. The goal of data imputation is to correct errors and input missing values.

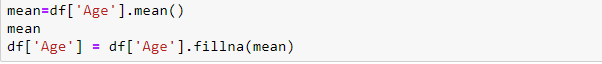
1. Check the missing values in all column.
2. Replace the missing values using appropriate methods wherever applicable.
3. Include only relevant data.
4. Use Encoding technique to transform categorical data into numerical data so that model can understand the data.
5. Check for Outliers and Skewness for numerical column.
6. Use standard scaler scale data 0-1 to bring the values to the same scale.

**Checking for the missing values in the data**



There are missing values in the Age, Cabin and Embarked Column out of which Age contributed for 19.86 % of total dataset, Cabin contain 77% of total data and Embarked result for only 0.22% of total data accordingly we will need to use methods or technique to handle the missing values.

Replacing the **Age** column with the **mean** values since the data is normally distributed



Now coming to **Embarked** features Since, there are 2 missing values in Embarked column we can replace by most frequent data.



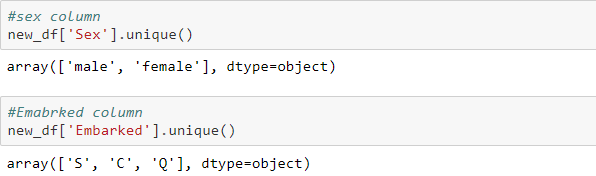
Data shows that there more than 77% percent NAN value for Cabin column as we don’t have sufficient data to analyse so it is better to drop the column.



Dropping all the other unnecessary column as well, which won't be useful in predicting the survival of person. Such as PassengerId, Name and Ticket.

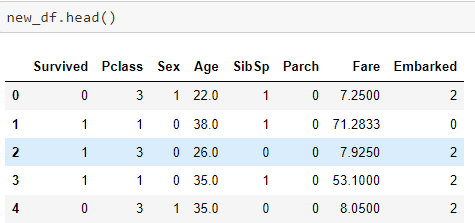
Now all the missing values and unnecessary data has been handled.

**Transform the categorical features into numerical feature using Label Encoding technique.**



There are two columns Sex and Embarked which contains categorical data so we will convert it into numerical ones.





**Label Encoding** refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated.

Now we can see that all the features are converted into numeric data.

**Check for Outliers and Skewness of the data**

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. In a sense, this definition leaves up to the analyst to decide what will be considered abnormal.

There are generally two methods to remove the outliers

1. Z Score Method

2. Quantile Method.

A skewed distribution is one that has a longer tail in one direction than the other. A very skewed distribution is likely to have outliers in the direction of the skew (but it is possible to have high skew without outliers). A symmetric distribution can also have outliers, but they have to be roughly symmetric.

In other way around we can say that the skewness is the result of outliers present in the dataset.

Skewness can be treated by checking either log transformation method or power transformation method whichever suits the best for any given data.

**Step 6: Building Machine Learning Models**

1. Split the dataset into independent features and dependent features.
2. Find the best random state.
3. Apply scaling technique to the feature columns.
4. Try out or test out different models on the dataset.
5. Apply hyperparameter tuning using Grid Search to the best accuracy model.
6. Save the best model for production.

Here we will simply deploy the various models with default parameters and see which one yields the best result. The models can further be tuned for better performance but are not in the scope of this one article. The models we will run are:

* Logistic Regression
* Decision Tree Classifier
* Support Vector Classifier
* Random Forest Classifier
* Gradient Boosting Classifier

Understanding some common terms in evaluation of metrics

**Confusion matrix**

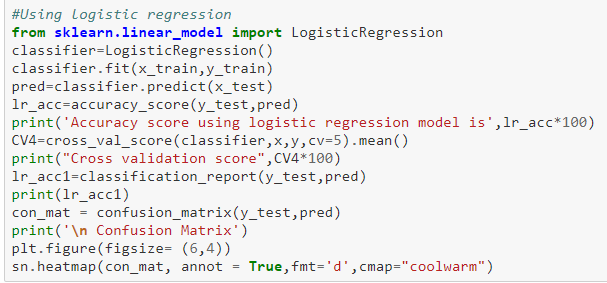
A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.

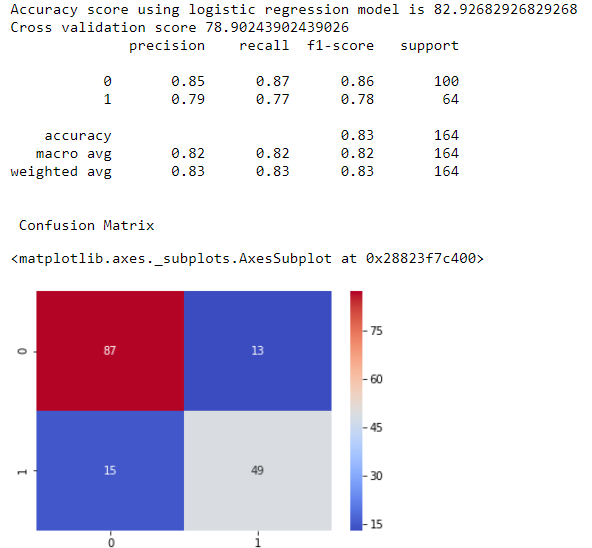
**Cross validation Score**

Cross validation is a statistical method used to estimate the skill of machine learning models.

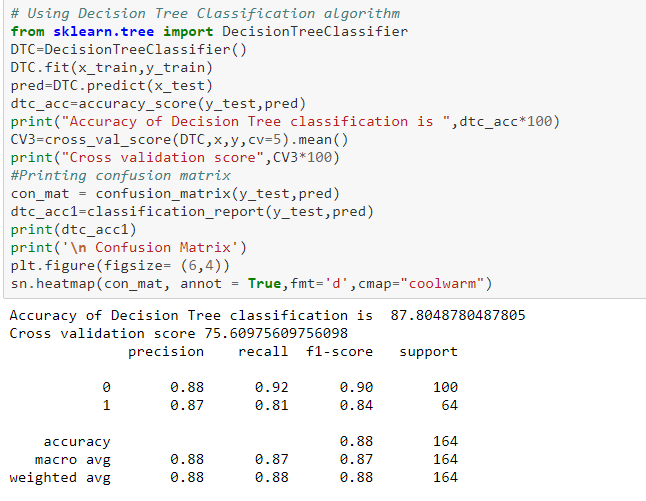
It is commonly used in applied machine learning to compare and select a model for a given predictive modelling problem because it easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than any other models.

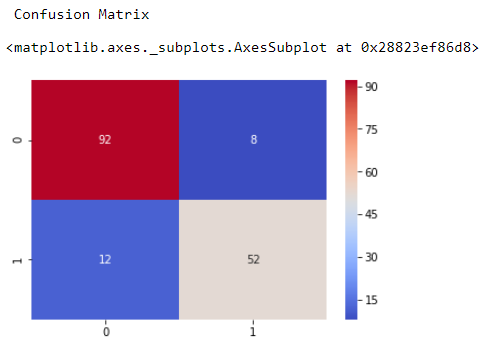
* **Logistic Regression**



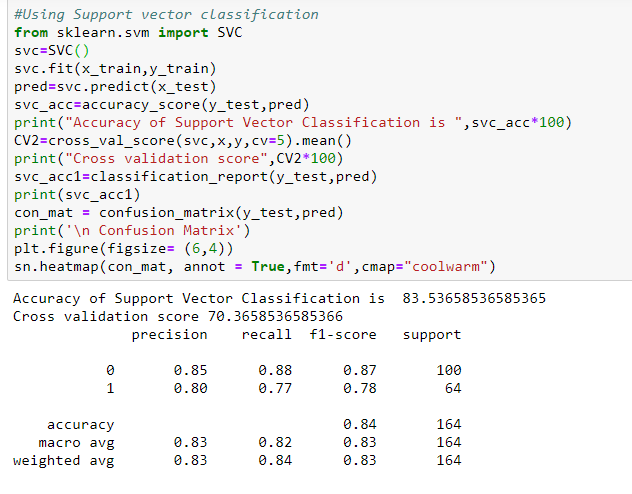


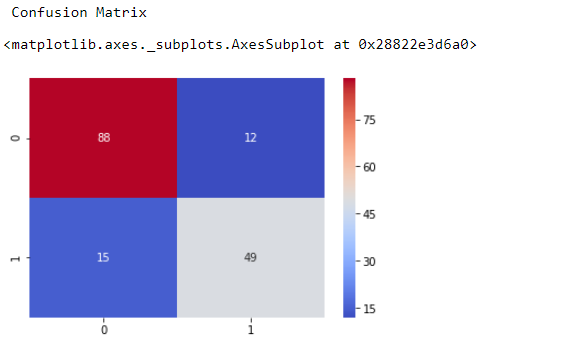
* **Decision Tree Classifier**



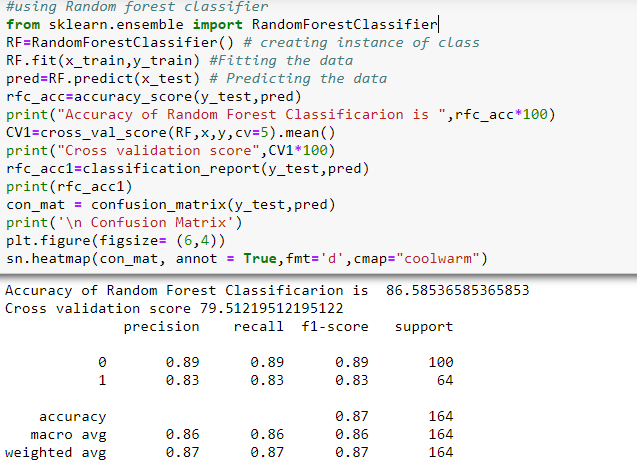


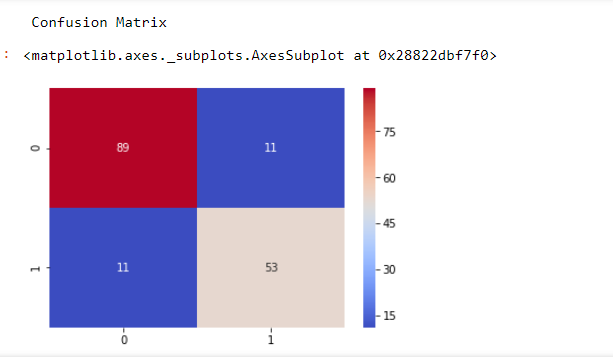
* **Support Vector Classifier**



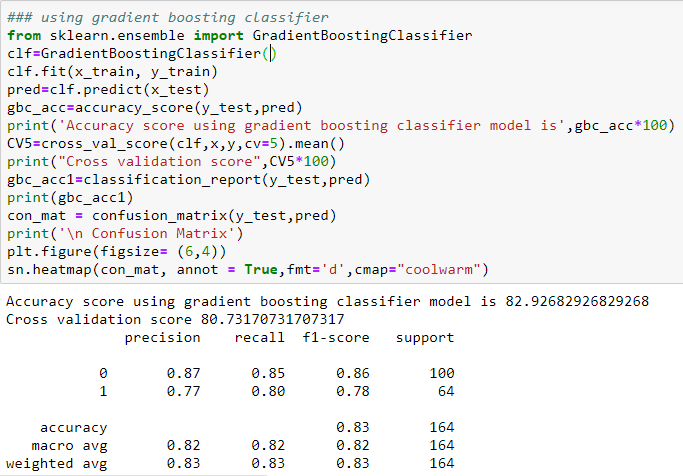


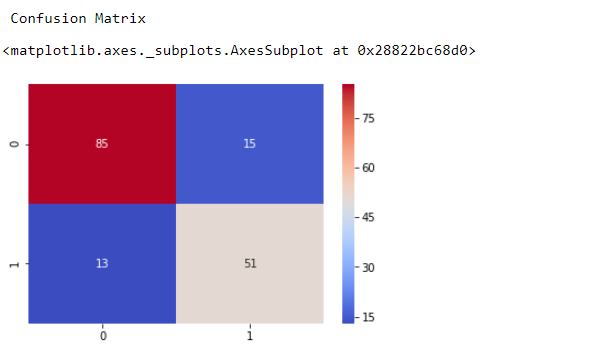
* **Random Forest Classifier**

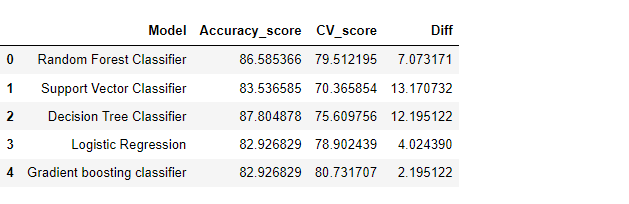




* **Gradient Boosting Classifier**







As we can see that we get decent accuracy with all models but the best accuracy score is of decision tree classifier but the difference between accuracy score and cross validation score is much higher than all of the model we have to look for the model which gives best performance or accuracy score and least cross validation score among all the models. Considering the above metrics for choosing the model then we can see that the Gradient Boosting Classifier gives accuracy score of nearly 83% and least cross validation score difference of 2. It can be considered as best model.

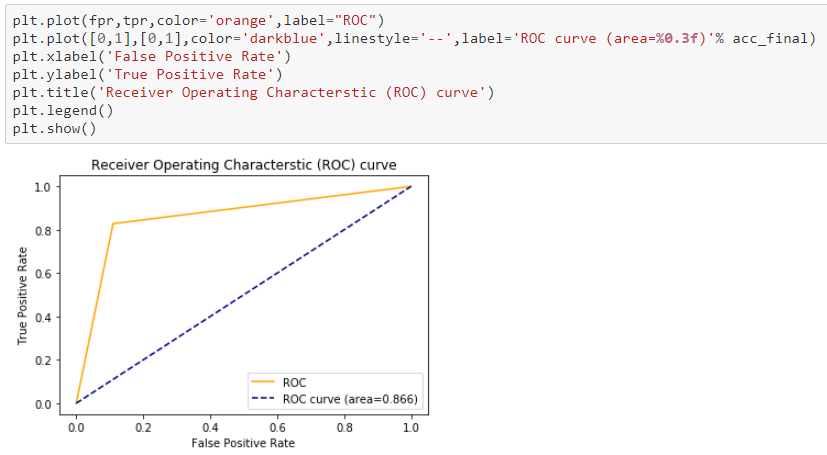
Since we got best model now, we can apply hyperparameter tuning to the best model



**With the help of hyperparameter tuning the accuracy has increased significantly from 83% to 86.58 %**

**ROC AUC CURVE**

Another way to evaluate and compare your binary classifier is provided by the ROC AUC Curve. This curve plots the true positive rate (also called recall) against the false positive rate (ratio of incorrectly classified negative instances), instead of plotting the precision versus the recall.



**Save the model for production**

In machine learning, while working with scikit learn library, we need to save the trained models in a file and restore them in order to reuse it to compare the model with other models, to test the model on a new data. The saving of data is called *Serialization*, while restoring the data is called *Deserialization*. When we need the same trained data in some different project or later sometime, to avoid the wastage of the training time, store trained model so that it can be used anytime in the future.   
There are two ways we can save a model in scikit learn:

1. **Pickle string**: The pickle module implements a fundamental, but powerful algorithm for serializing and de-serializing a Python object structure.
2. **Pickled model as a file using joblib**: Joblib is the replacement of pickle as it is more efficient on objects that carry large NumPy arrays. These functions also accept file-like object instead of filenames.

**Concluding Remarks**

We started with the data loading, data exploration where we got a feeling for the dataset, checked about missing data and learned which features are important. During this process we used seaborn and matplotlib to do the visualizations. During the data pre-processing part, we computed missing values, converted features into numeric ones, grouped values into categories and created a few new features.

Afterwards we started training 5 different machine learning models, picked one of them and applied hyperparameter tuning using grid search on it. Then we discussed how to choose best model based upon the different metrics, took a look at the importance it assigns to the different features and tuned its performance through optimizing it’s hyperparameter values. Lastly, we looked at its confusion matrix and computed the model’s precision, recall and f-score. Finally, we saved the model for production.