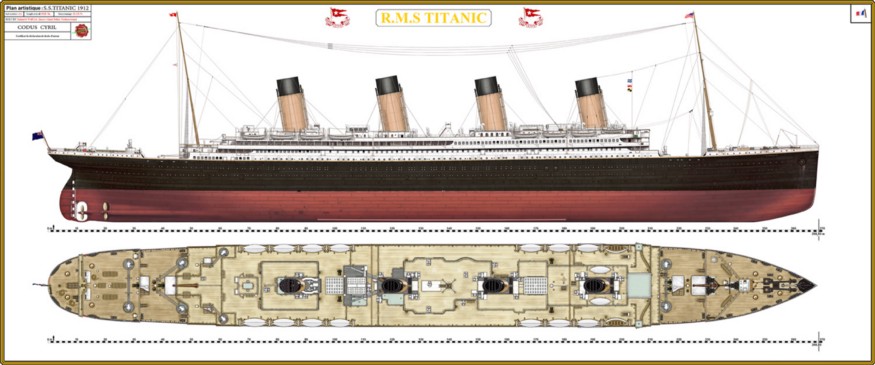
Predicting the Survival of Titanic Passengers



In this blog-entry, I will go through the entire course of making an AI model on the popular Titanic dataset, which is utilized by many individuals from one side of the planet to the other. It gives data on the destiny of travelers on the Titanic, summed up as per monetary status (class), sex, age and endurance.

RMS Titanic

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. The RMS Titanic was the largest ship afloat at the time it entered service and was the second of three Olympic-class Ocean liners operated by the White Star Line. The Titanic was built by the Harland and Wolff shipyard in Belfast. Thomas Andrews, her architect, died in the disaster.

# 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, 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.

The main aim of this project is to predict the survival state or status of the passengers based on the given features. With the help of this article, we will learn EDA Exploratory Data Analysis and Data Visualization, and survival prediction model building. we will learning different relationships between target data and independent data with the help of the given features.

# Importing the Libraries

# 

# Getting the Data

# 

# Data Exploration/Analysis

# 

# 

**<class 'pandas.core.frame.DataFrame'>**

**RangeIndex: 891 entries, 0 to 890**

**Data columns (total 12 columns):**

**# Column Non-Null Count Dtype**

**--- ------ -------------- -----**

**0 PassengerId 891 non-null int64**

**1 Survived 891 non-null int64**

**2 Pclass 891 non-null int64**

**3 Name 891 non-null object**

**4 Sex 891 non-null object**

**5 Age 714 non-null float64**

**6 SibSp 891 non-null int64**

**7 Parch 891 non-null int64**

**8 Ticket 891 non-null object**

**9 Fare 891 non-null float64**

**10 Cabin 204 non-null object**

**11 Embarked 889 non-null object**

**dtypes: float64(2), int64(5), object (5)**

**memory usage: 83.7+ KB**

Data is having 891 Rows and 12 columns.

Target: 'Survived'

Features: 'PassengerId', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'

Data is having 2 features float type, 5 features integer type and 5 features object type.

**Below I have listed the features with a short description:**

PassengerId - This gives the details of the ID no. of each passenger

Survived - This column tells the status of survival of a particular passenger. 1-denote survive and 0-denote not survive (Died)

Pclass - This column tells the Corresponding Ticket classes

Name - This column is having the details of all the passenger's name

Sex - This column tells the gender of the passengers

Age - By this column we can get the details of the age of passengers

SibSp - This column represents the no. of siblings / spouses aboard the Titanic

Parch- This column represents the no. of parents / children aboard the Titanic

Ticket- This represents the Ticket no.

Fare- This column represents the ticket amount which the passenger had paid

Cabin- This represents the Cabin no.

Embarked- This column represents the Port location of Embarkation of the passengers, C = Cherbourg, Q = Queenstown, S = Southampton

From the table above, we can take note of a couple of things. Above all else, that we wanted to change over a lot of features into numeric ones later on, so the machine learning algorithms can deal with them. Besides, we can see that the components have widely different ranges, that we should change over into generally a similar scale. We can likewise recognize some more components, that contain missing qualities (NaN = not a number), that small need to manage.



**PassengerId 0**

**Survived 0**

**Pclass 0**

**Name 0**

**Sex 0**

**Age 177**

**SibSp 0**

**Parch 0**

**Ticket 0**

**Fare 0**

**Cabin 687**

**Embarked 2**

**dtype: int64**

We can see data is having Null values.

For Age = 177 missing values

For Cabin= 687 missing values

For Embarked= 2 missing values

The Embarked feature has just 2 missing values, which can without much of a stretch be filled. It will be considerably more interesting, to manage the 'Age’ feature, which has 177 missing values. The ‘Cabin’ feature needs further examination, however it appears as though that we should drop it from the dataset, since 77 % of it are absent.

**What features could contribute to a high survival rate?**

To me it would be well if everything aside from 'PassengerId', 'Ticket' and 'Name' would be correlated with a high survival rate.

Let’s check the survival ratio of overall passengers on Titanic.



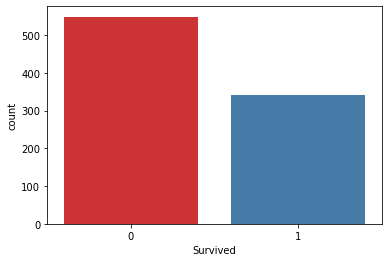
**0 549**

**1 342**

**Name: Survived, dtype: int64**



**<AxesSubplot:xlabel='Survived', ylabel='count'>**



Here we can see that 38% of the passenger survived on Titanic.

Let’s check the ratio of the gender on the Titanic.



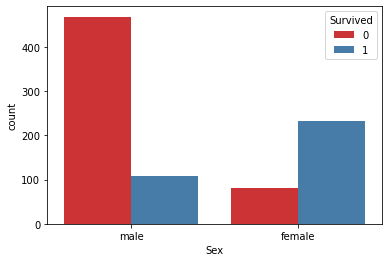
**male 577**

**female 314**

**Name: Sex, dtype: int64**



<AxesSubplot:xlabel='Sex', ylabel='count'>



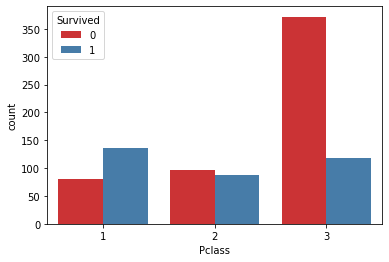
We can see 66 percent of passengers were male.

Here we can see the number of females survived more than the male passenger.

Let’s check the Pclass columns with respect to Survived.

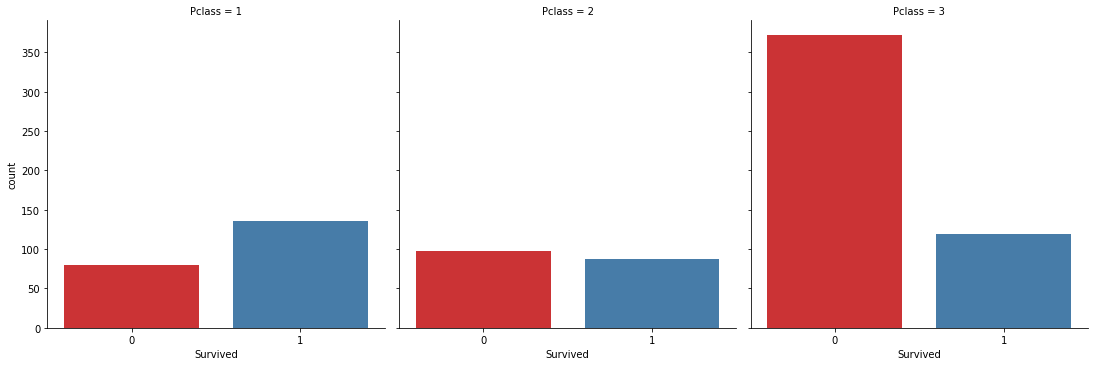


**<AxesSubplot:xlabel='Pclass', ylabel='count'>**



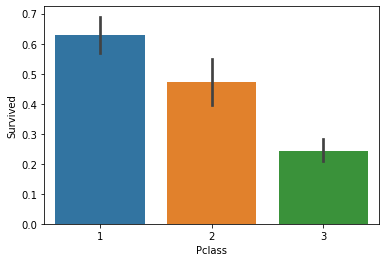


**<seaborn.axisgrid.FacetGrid at 0x15e279abd30>**





**<AxesSubplot:xlabel='Pclass', ylabel='Survived'>**



Here we can see Pclass playing a big role in determining survived case.

Passenger in Pclass=1 sutvived more than the other class.

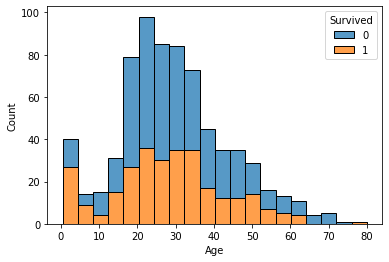
For Pclass= 2 the difference is less.

For Pcalss=3 survived is less as compared to not Survived.

Let’s talk about the Age column.



**<AxesSubplot:xlabel='Age', ylabel='Count'>**

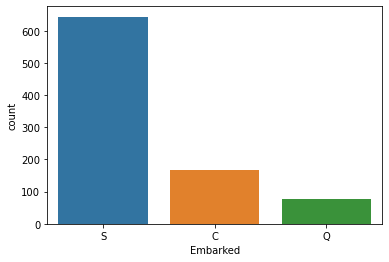


Here we can see age group (15-40) having high strength in the data. The number of Survived and Not Survived is almost same for these age group. and the data is distributed between 0.4 to 80 age group.

Let’s see Embarked having any relation or not.



**<AxesSubplot:xlabel='Embarked', ylabel='count'>**



Embarked implies where the passenger mounted from.

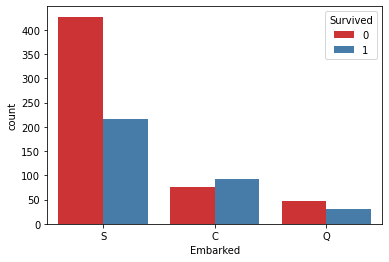
Here S= Southampton, C= Cherbourg and Q= Queenstown.

Most of the people boarded from Southampton.

Let’s see the survived relationship with Embarked.



**<AxesSubplot:xlabel='Embarked', ylabel='count'>**



If passenger board at Southampton are not likely to survive, according to Embarked data.

Let’s check the SibSp column. Which indicates Number of Sibling and spouse.



**0 608**

**1 209**

**2 28**

**4 18**

**3 16**

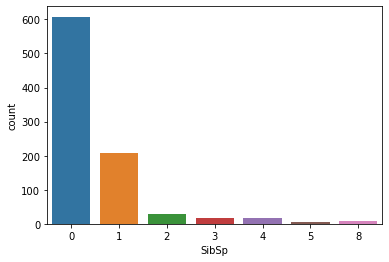
**8 7**

**5 5**

**Name: SibSp, dtype: int64**



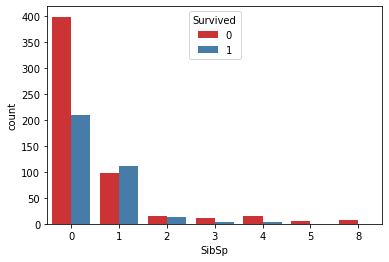
**<AxesSubplot:xlabel='SibSp', ylabel='count'>**



We can see 68 percent passenger traveled without any sibling or spouse and rest of the passenger were there with their spouse or siblings.



**<AxesSubplot:xlabel='SibSp', ylabel='count'>**



Passengers with no siblings or spouse are more likely to not Survived.

Let’s Visualize Parch columns. which indicates Parents and children.



**0 678**

**1 118**

**2 80**

**5 5**

**3 5**

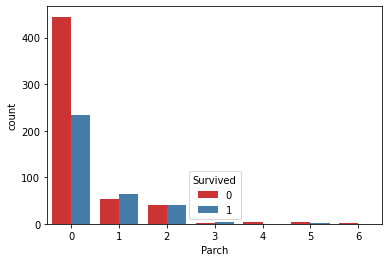
**4 4**

**6 1**

**Name: Parch, dtype: int64**



**<AxesSubplot:xlabel='Parch', ylabel='count'>**



Passengers without parents and children are more likely to not Survived.

# EDA Conclusion:

* The survival rate of the female passengers are more than the male passengers.
* Passenger class playing a big role in survival of passengers like Passenger in Pclass=1 survived more than the other class. For Pclass= 2 the difference is less. But for Pcalss=3 survived is less as compared to not Survived.
* Passengers who had no siblings or spouse are more likely to not Survived.
* If passenger board at Southampton are not likely to survive, according to Embarked data.
* Age group (15-40) having high strength in the data. The number of Survived and Not Survived is almost same for these age group. and the data is distributed between 0.4 to 80 age group

# Data-Pre-Processing:

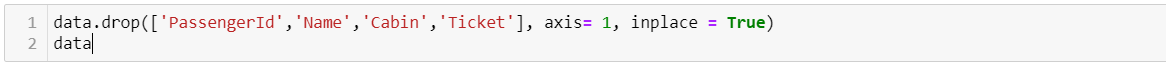
In this stage we will pre-process our data to make it relevant for model building. We will first drop all the columns which are not relevant. We will encode all the categorical feature columns using one hot encoder and target column using Label Encoder. After that we will check correlation, skewness and scale the feature data using Standard Scaler.

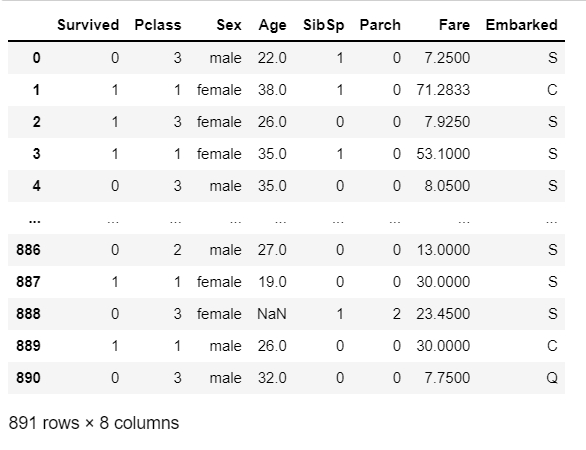
After analysis, we came to conclusion our Pclass, Sex, Age, SibSp, Parch, Fare and Embarked having relation with target column (Survived). On the other hand, PassengerId, Name, Ticket has no relevant relation with the Target Column.

So, we can drop the PassengerId, Name and Ticket column.

Cabin is having lots of missing value, instead of filling we will drop this column.

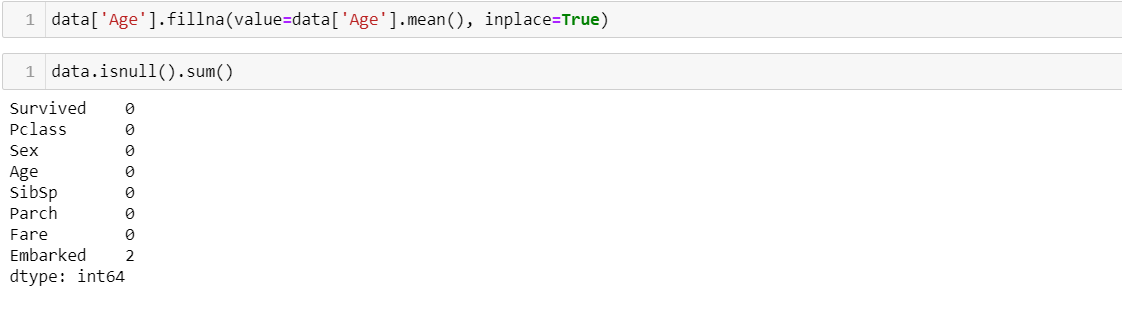
## Dropping Data:





Here we are having 177 Null values in Age columns and 2 Null values in Embarked.

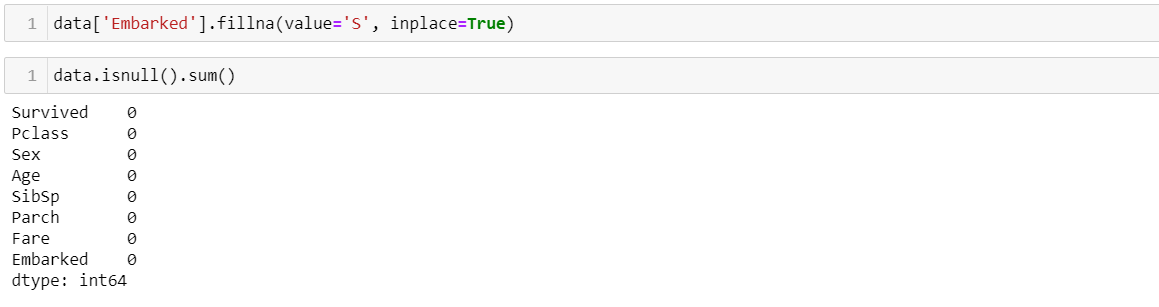
We will fill these Nan values first; we will fill our age column using mean of that column.



Let’s fill the Embarked column’s null values.



Here most common value is S. So, we will replace it with 'S'



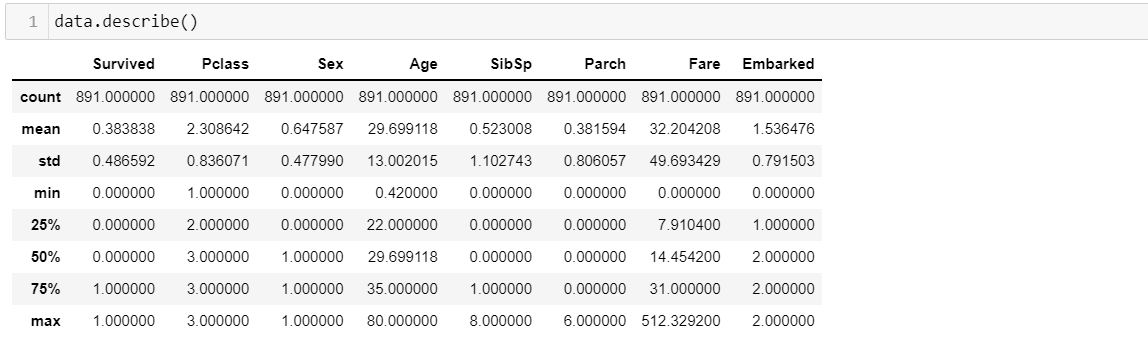
Data is not having Null values, its time to deal with the string data.

Sex and Embarked columns are having String type data.

We will use LabelEncoder to encode string values.



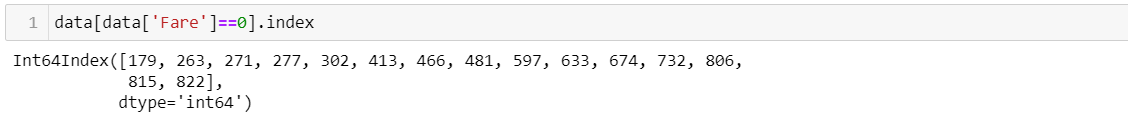




Here we can see data is having 891 counts in each column, so no null value is present.

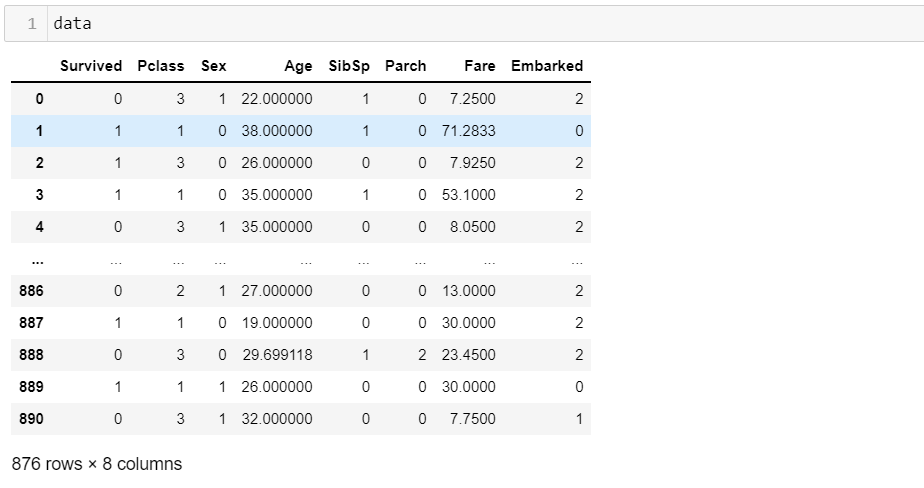
For Fare column Fare cannot be Zero.

Let’s check the index values of the Fare column having Zero values.



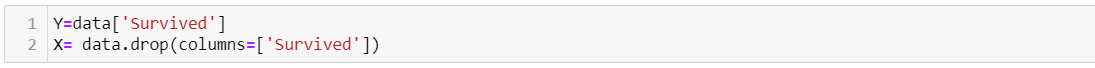
Above are the index values which are having zero values in the Fare columns.



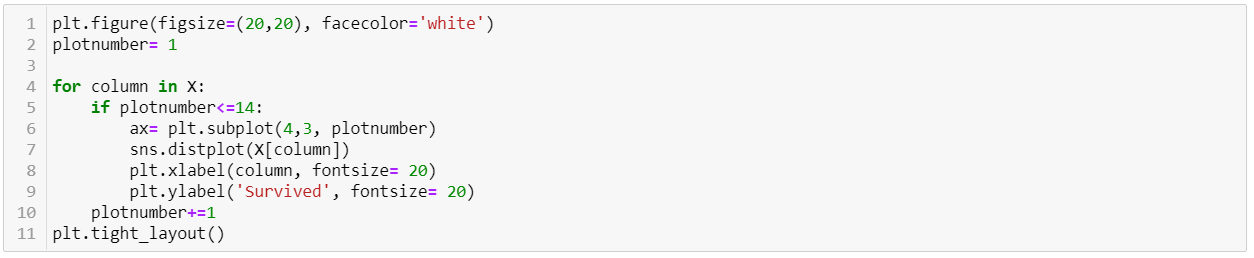


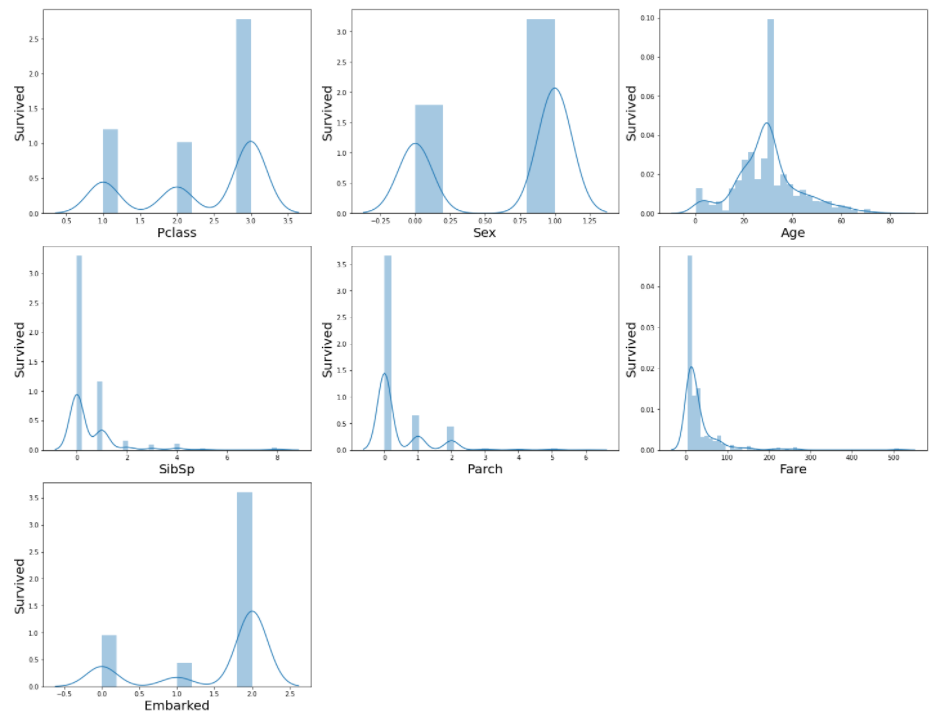
Now our final data after preprocessing having 876 rows and 8 columns.

Now we will divide the in to the independent columns and the dependent/target column

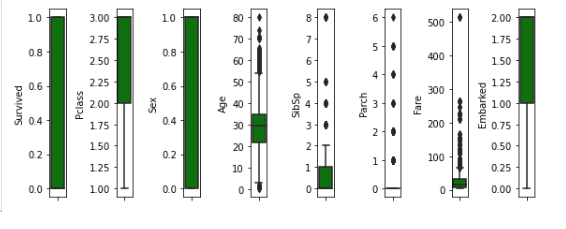


Before proceeding further first we will check the skewness and the outliers in the independent columns data by plotting distribution plot and boxplot.



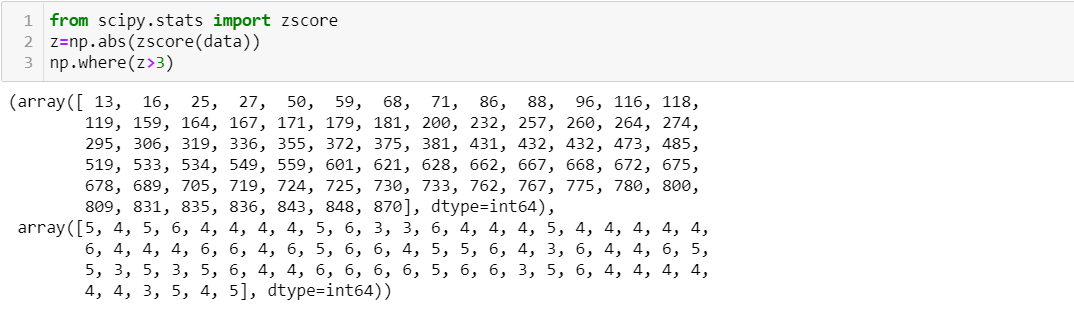




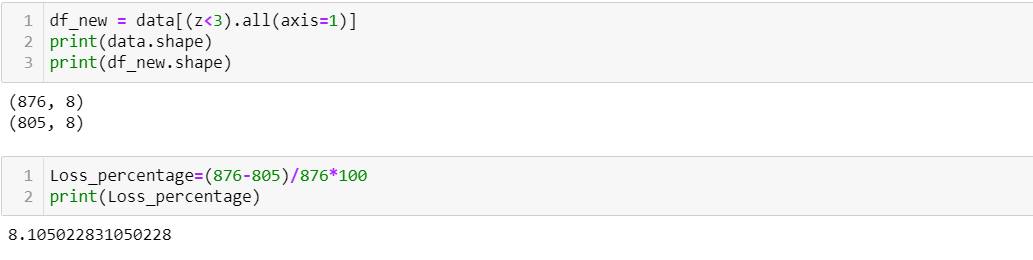


We can see in few Columns outliers are present, and data has skewness as well. So, we have to remove the skewness and the outliers as well by using the z-score and power transform method.

### Removing outliers



Before removing the outliers first, we will check the percentage of the data loss, if the data loss is more than the 10% then we will not remove the outliers and try another method to process the data but if the data loss is less than the 10 % then we will remove the outliers using this method.



Here Our data loss is less than 10 percent, hence we can proceed with df\_new. And now we will remove the skewness from the df\_new data.

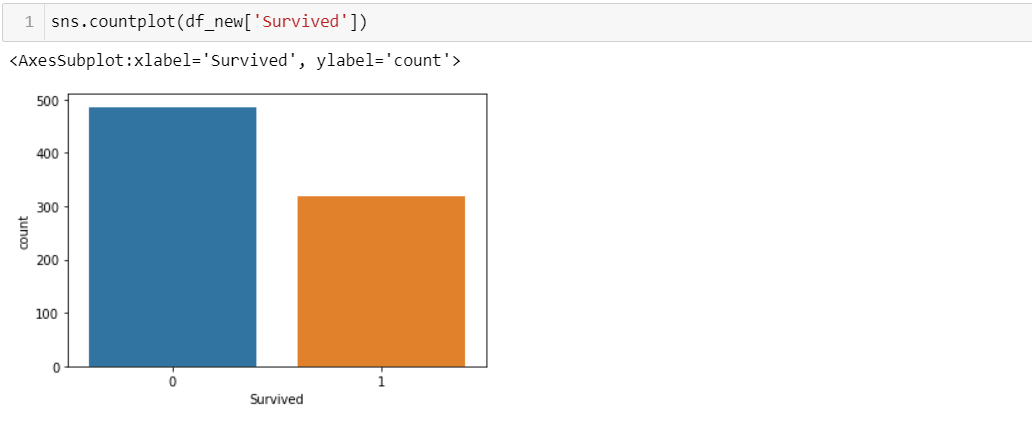
### Removing Skewness

Skewness is removed only for Features data so will divide our data into features and target.

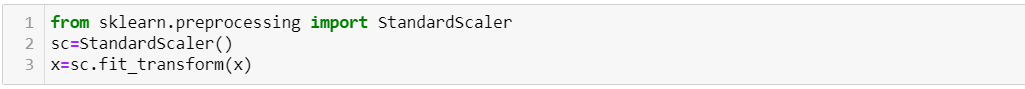


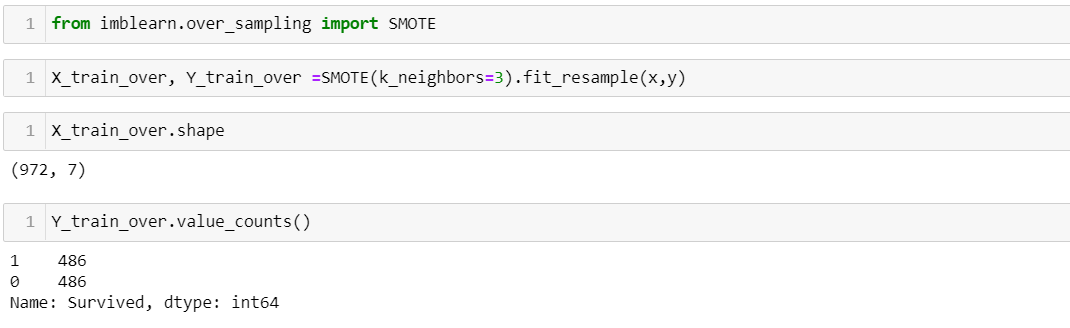
### Data Scaling

Now we will check that if our data is balance i.e., equally divided in to survived and non-survived



By the above graph we see that data is not properly balance so we will first, balance the data using the SMOTE method.

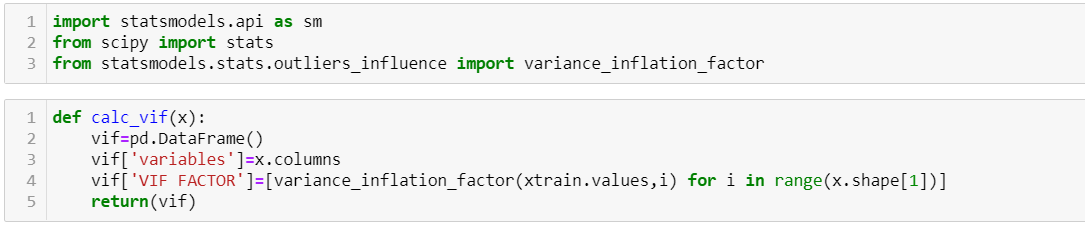


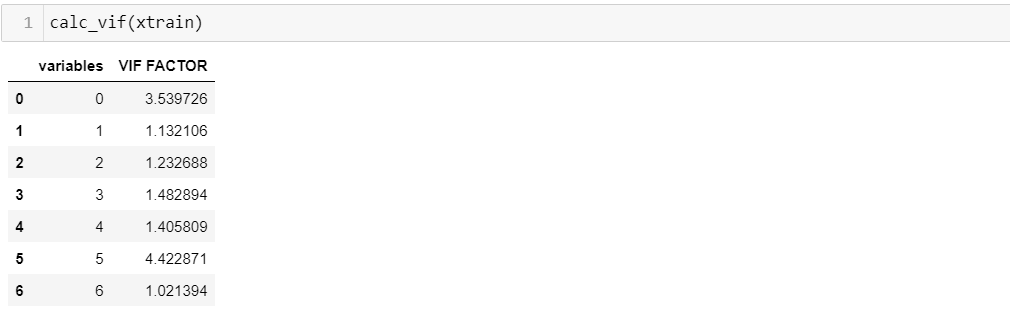




Now our data is balance. we will check the VIF (variance inflation factor).

### VIF calculation



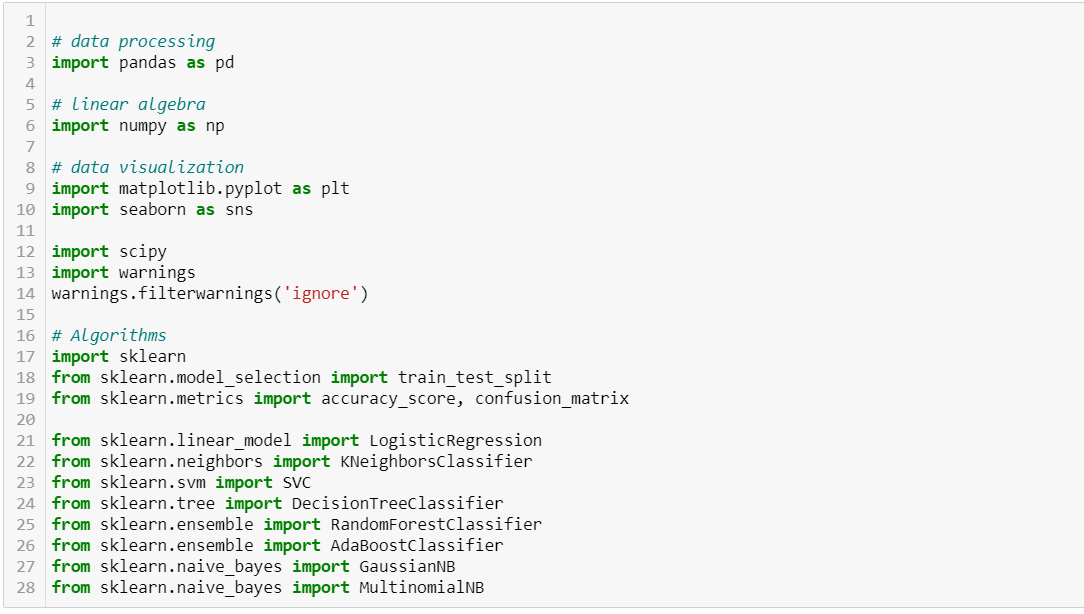


As we can see VIF is less than 10 we will not remove any columns, and proceed further.

**Building Machine Learning Models**

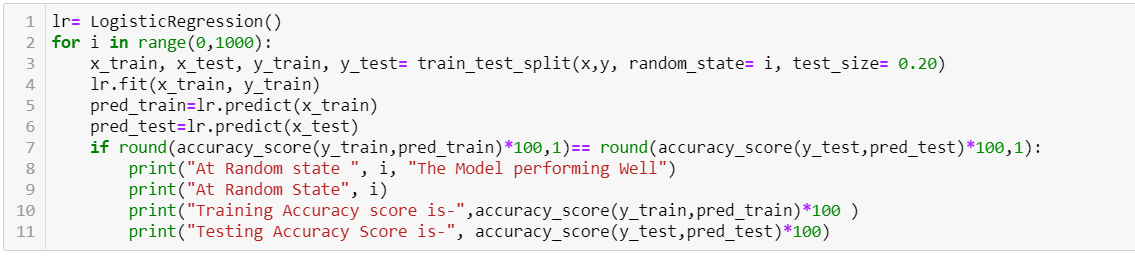
Since our output is having only two values "0" and "1", we will use binary classification model.





##### **Choosing Best Random State**

Now we will find the best random state at which model gives the best result.



At Random state 68 The Model performing Well

At Random State 68

Training Accuracy score is- 78.88198757763976

Testing Accuracy Score is- 78.88198757763976

At Random state 110 The Model performing Well

At Random State 110

Training Accuracy score is- 79.5031055900621

Testing Accuracy Score is- 79.5031055900621

At Random state 162 The Model performing Well

At Random State 162

Training Accuracy score is- 79.5031055900621

Testing Accuracy Score is- 79.5031055900621

At Random state 169 The Model performing Well

At Random State 169

Training Accuracy score is- 80.12422360248446

Testing Accuracy Score is- 80.12422360248446

At Random state 171 The Model performing Well

At Random State 171

Training Accuracy score is- 80.12422360248446

Testing Accuracy Score is- 80.12422360248446

At Random state 180 The Model performing Well

At Random State 180

Training Accuracy score is- 79.5031055900621

Testing Accuracy Score is- 79.5031055900621

At Random state 194 The Model performing Well

At Random State 194

Training Accuracy score is- 79.5031055900621

Testing Accuracy Score is- 79.5031055900621

At Random state 237 The Model performing Well

At Random State 237

Training Accuracy score is- 79.5031055900621

Testing Accuracy Score is- 79.5031055900621

At Random state 264 The Model performing Well

At Random State 264

Training Accuracy score is- 80.74534161490683

Testing Accuracy Score is- 80.74534161490683

At Random state 332 The Model performing Well

At Random State 332

Training Accuracy score is- 79.5031055900621

Testing Accuracy Score is- 79.5031055900621

At Random state 385 The Model performing Well

At Random State 385

Training Accuracy score is- 79.5031055900621

Testing Accuracy Score is- 79.5031055900621

At Random state 653 The Model performing Well

At Random State 653

Training Accuracy score is- 79.5031055900621

Testing Accuracy Score is- 79.5031055900621

At Random state 720 The Model performing Well

At Random State 720

Training Accuracy score is- 78.88198757763976

Testing Accuracy Score is- 78.88198757763976

At Random state 772 The Model performing Well

At Random State 772

Training Accuracy score is- 78.88198757763976

Testing Accuracy Score is- 78.88198757763976

At Random state 823 The Model performing Well

At Random State 823

Training Accuracy score is- 80.12422360248446

Testing Accuracy Score is- 80.12422360248446

At Random state 866 The Model performing Well

At Random State 866

Training Accuracy score is- 80.12422360248446

Testing Accuracy Score is- 80.12422360248446

At Random state 867 The Model performing Well

At Random State 867

Training Accuracy score is- 80.12422360248446

Testing Accuracy Score is- 80.12422360248446

At Random state 938 The Model performing Well

At Random State 938

Training Accuracy score is- 78.88198757763976

Testing Accuracy Score is- 78.88198757763976

At Random state 944 The Model performing Well

At Random State 944

Training Accuracy score is- 79.5031055900621

Testing Accuracy Score is- 79.5031055900621

At Random state 985 The Model performing Well

At Random State 985

Training Accuracy score is- 79.5031055900621

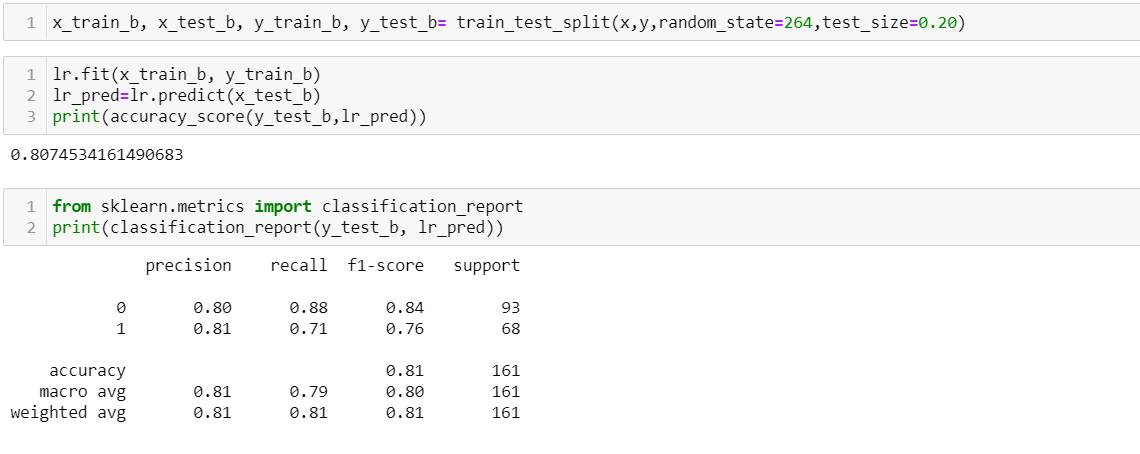
Testing Accuracy Score is- 79.5031055900621

At Random State 264

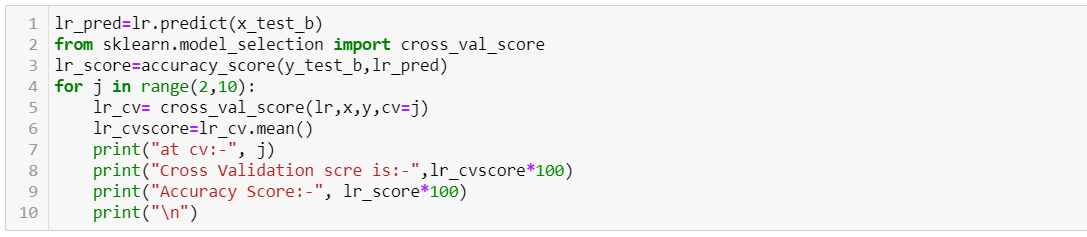
Training Accuracy score is- 80.74534161490683

Testing Accuracy Score is- 80.74534161490683

At Random state 264 The Model performing Well. So, we will train our model at this random state to get more accuracy in the output results.



#### **Cross Validation of the model**



at cv:- 2

Cross Validation score is:- 78.50882066096318

Accuracy Score:- 80.74534161490683

at cv:- 3

Cross Validation score is:- 79.13083282472397

Accuracy Score:- 80.74534161490683

at cv:- 4

Cross Validation score is:- 78.88281365449978

Accuracy Score:- 80.74534161490683

at cv:- 5

Cross Validation score is:- 79.1304347826087

Accuracy Score:- 80.74534161490683

at cv:- 6

Cross Validation score is:- 79.25557398194213

Accuracy Score:- 80.74534161490683

at cv:- 7

Cross Validation score is:- 79.13043478260869

Accuracy Score:- 80.74534161490683

at cv:- 8

Cross Validation score is:- 79.62623762376238

Accuracy Score:- 80.74534161490683

at cv:- 9

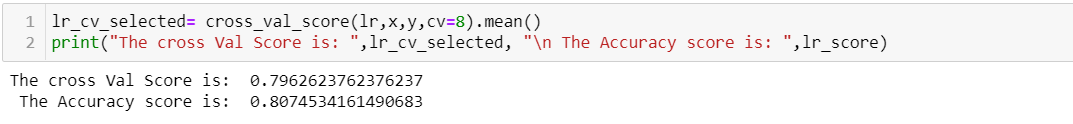
Cross Validation score is:- 79.61853239006797

Accuracy Score:- 80.74534161490683

At cv:- 8

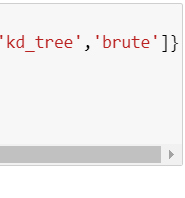
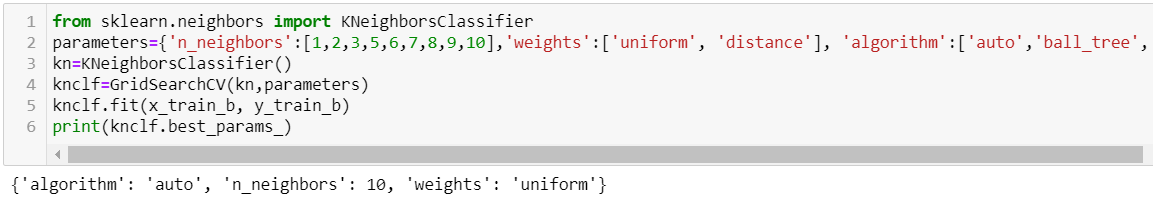
Cross Validation score is:- 79.62623762376238

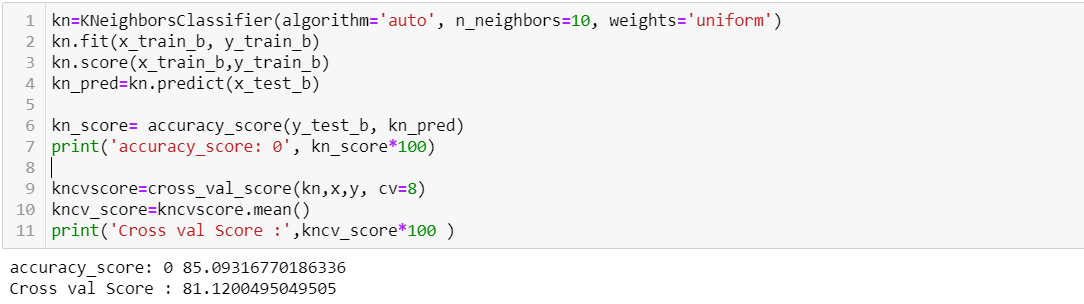
Accuracy Score: - 80.74534161490683

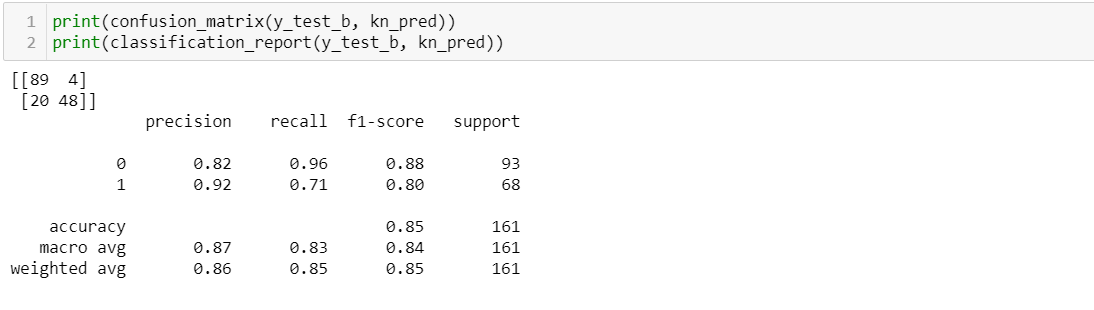


Logistic Regression giving accuracy score= 80.75 Let’s try another model to find best score.

### KNeighbors Classifier





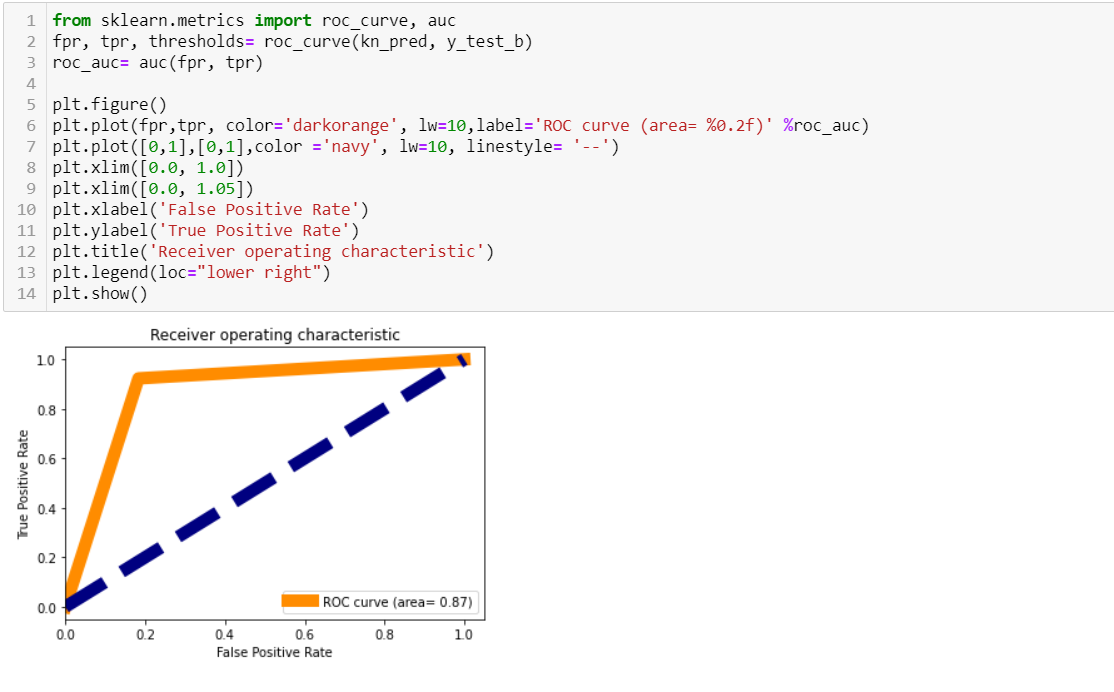


KNeighborsClassifier having better score than Logistic Regression. We can consider this as the best fit model.

# Concluding Remarks

After checking all the model, we came to conclusion that KNeighborsClassifier performing well. With almost equal accuracy score and cross validation score. Let’s save the best model, before that lets draw the AOC- ROC score.

### AUC ROC Curve:



### Saving Best Model



We can conclude that with help of machine learning or model data analytics we can predict the whether there will be chances of survival of passengers or not.