Titanic Dataset

Shahriar Hossain

1712852642

Abstract

Titanic is one of the most famous factor in history, titanic dataset also famous for curiosity. I have also analyze titanic dataset by using machine learning methods, and algorithms to find out the survival chance.I used many machine learning techniques to improve accuracy of prediction, So that my project can get high accuracy.

Introduction

The goal of the project was to predict the survival of passengers based on a set of data from the titanic dataset.I used kaggle dataset which contains both training set and test set,so it makes it easier.For each passenger in the test set,I had to predict whether they survived or not.

At work I used the programming language Python and its libraries NumPy and SciKit-Learn.

I used many algorithm to find out best performance:

1. Logistic Regression

2. Naïve Bayes

3. SVM

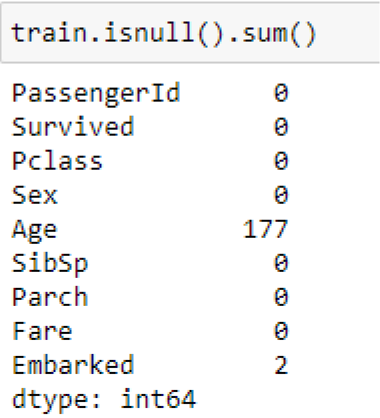
4. Decision Tree

5. Random Forest

I also performed feature engineering and analyzed correlation between different factors to get good accuracy.

WORKING

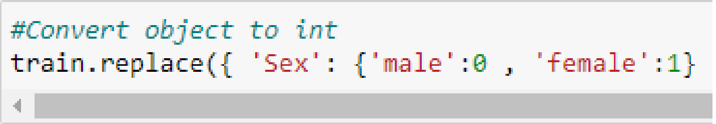
The titanic dataset contains Passenger ID, Passenger Class , Name , Sex, Age, embarked,Number of passenger's siblings and spouses etc.To building a prediction system we need feature engineering as we know the data can have missing fields, incomplete fields a crucial step in. so I checked for missing values and find out and fill them with mean values,



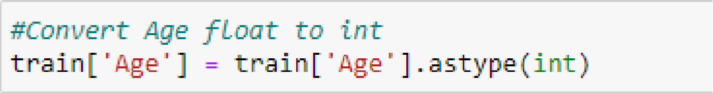
```
train.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Sex              0
Age            177
SibSp            0
Parch            0
Fare             0
Embarked         2
dtype: int64
```

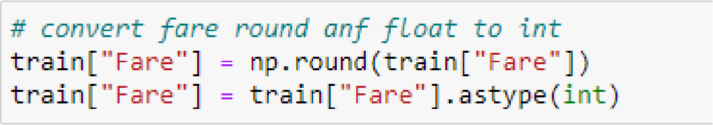
I also did data conversion object ,character and float type data to integer.so that I can fit the data to model.



```
#Convert object to int
train.replace({ 'Sex': {'male':0 , 'female':1}
```

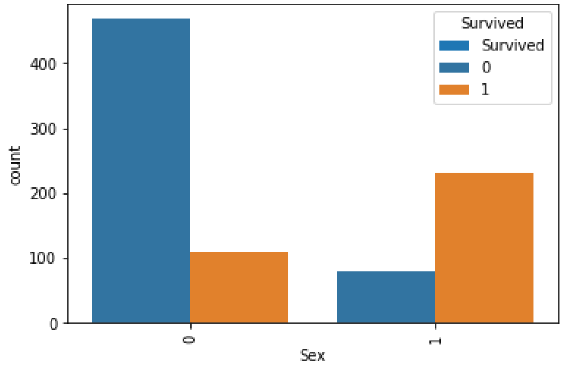


```
#Convert Age float to int
train['Age'] = train['Age'].astype(int)
```

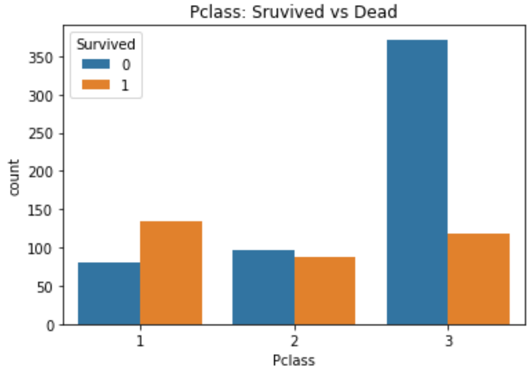


```
# convert fare round anf float to int
train["Fare"] = np.round(train["Fare"])
train["Fare"] = train["Fare"].astype(int)
```

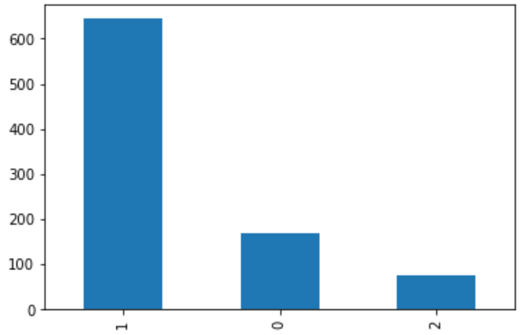
I also try to find correlation in features,



Sruvived difference in sex



Survived vs Dead



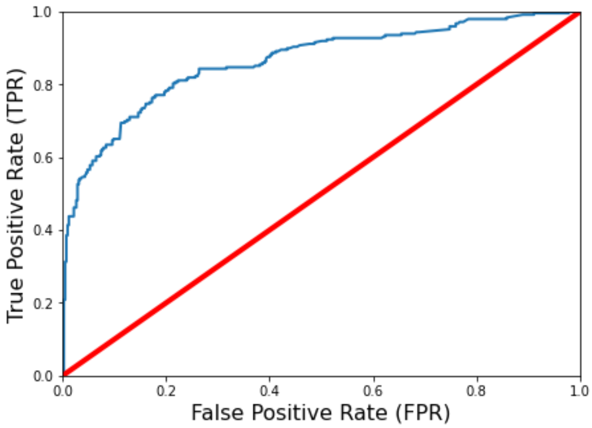
Embarked

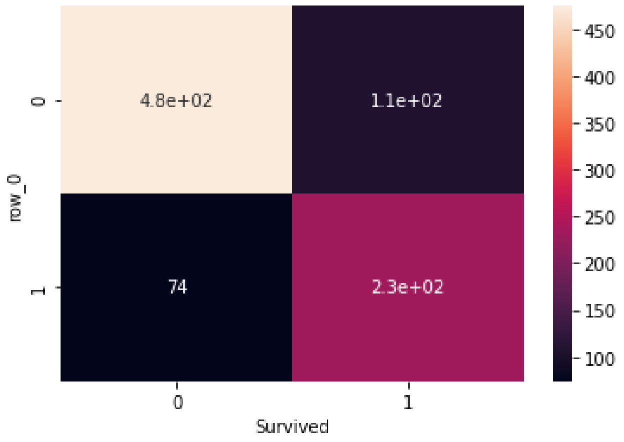
I also dropped an unnecessary column called Name & Tacket ,Then I fit the train data to models one by one to predict survival chances.

Algorithms	**Accuracy**
Logistic Regression	79.1%
Naïve Bayes	74.4%
SVM	84.7 %
Decision Tree	75.3%
Random Forest	75.2%

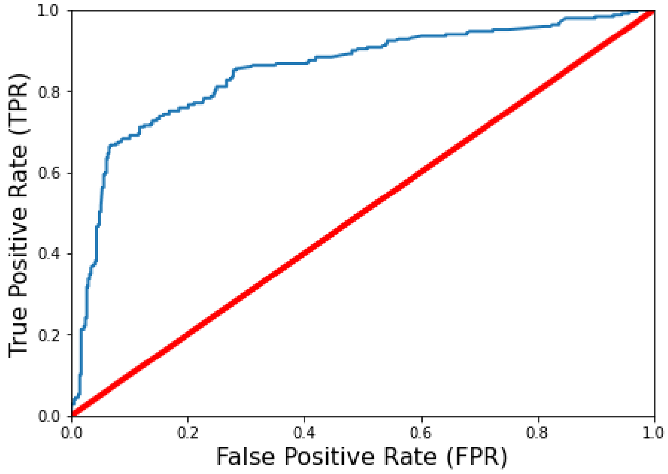
So it’s clear SVM gives me the highest accuracy in the prediction .Lets see the visuals for better understanding.

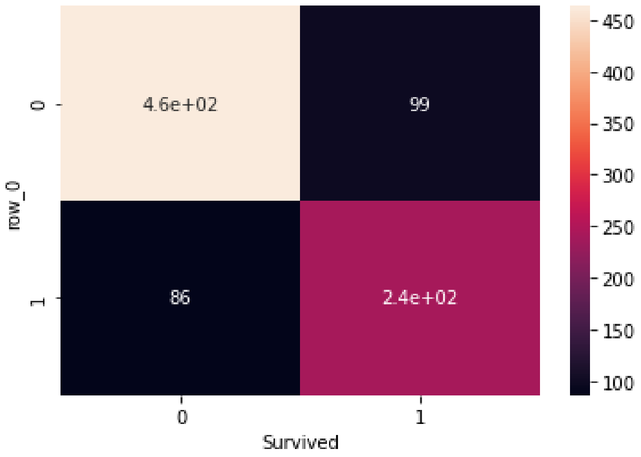
Logistic Regression



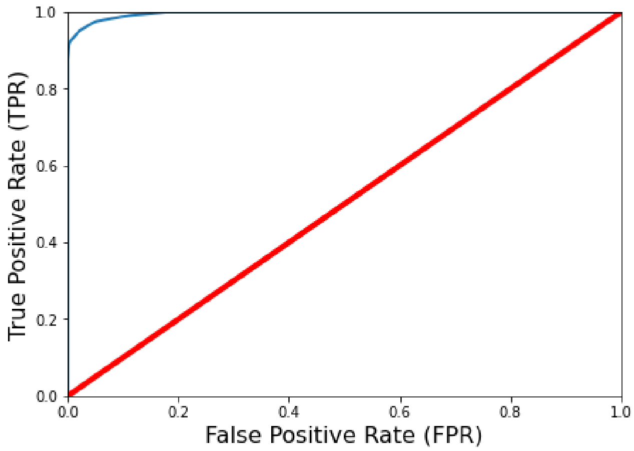

4.8e+02
1.1e+02
74
2.3e+02
0
1
row_0
0
1
Survived
450
400
350
300
250
200
150
100

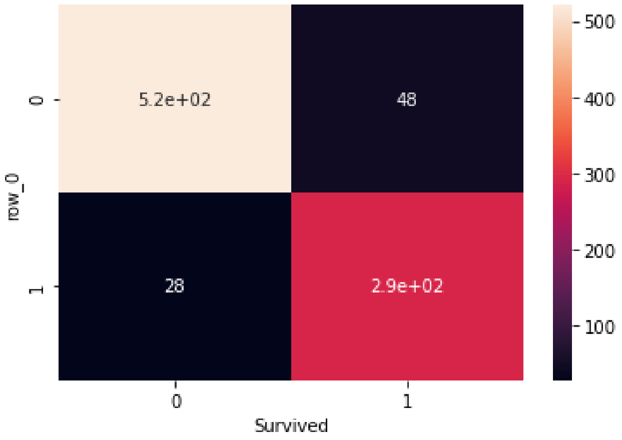
Naïve Bayes


1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate (TPR)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (FPR)

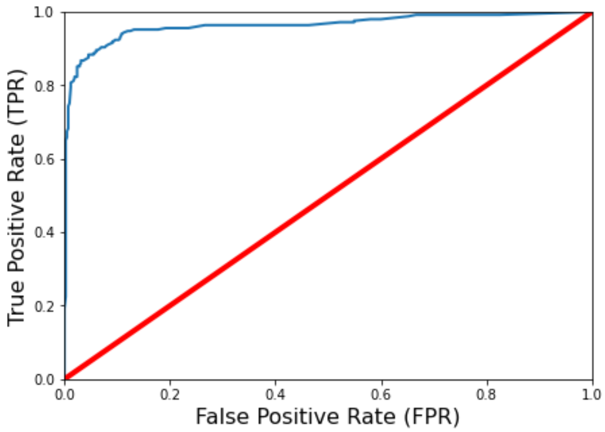

4.6e+02
99
86
2.4e+02
0
1
row_0
0
1
Survived
450
400
350
300
250
200
150
100

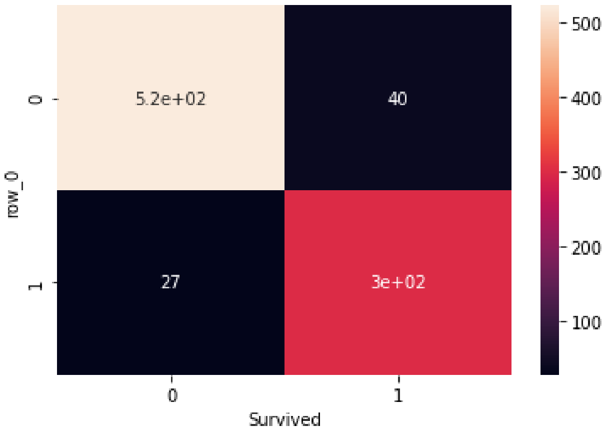
Decision Tree





Random Forest





CONCLUSION

In conclusion of my work I have gained good experience in building a prediction system and achieved 84.7 % accuracy with svm in predictiction of survival chance from Titanic dataset.