Predicting Titanic Survivors using Machine Learning

By:

Satvik Gupta

Sabyasachi Bhoi

**Introduction**

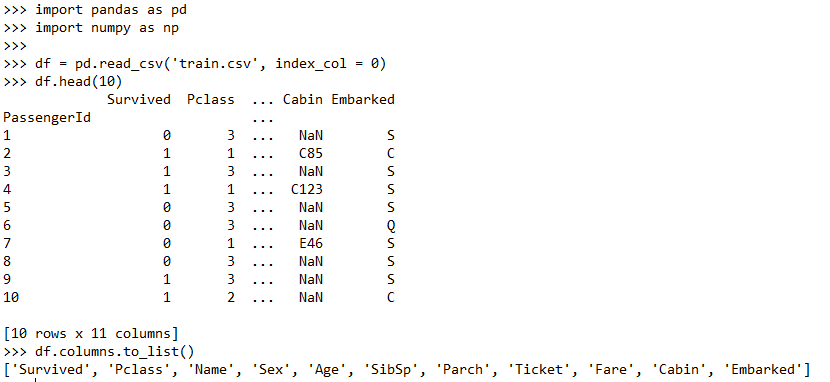
In this project, our goal is to essentially predict whether a person based on his information, would have survived the titanic. This task can be completed using Machine Learning Algorithms.

The data for these algorithms is collected from a website called Kaggle.com. This website provides a data set containing the information of actual people who sailed (and sank) in the titanic.

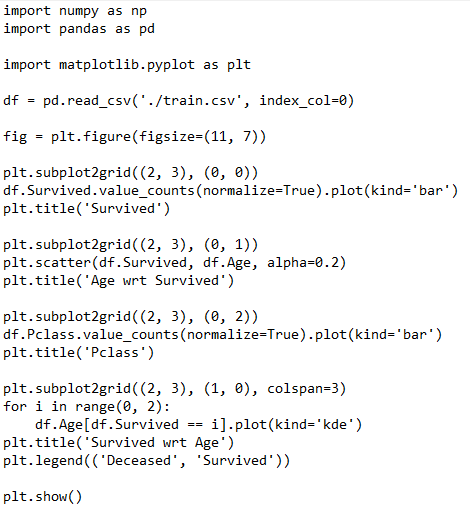
**Initial Analysis of the Data**

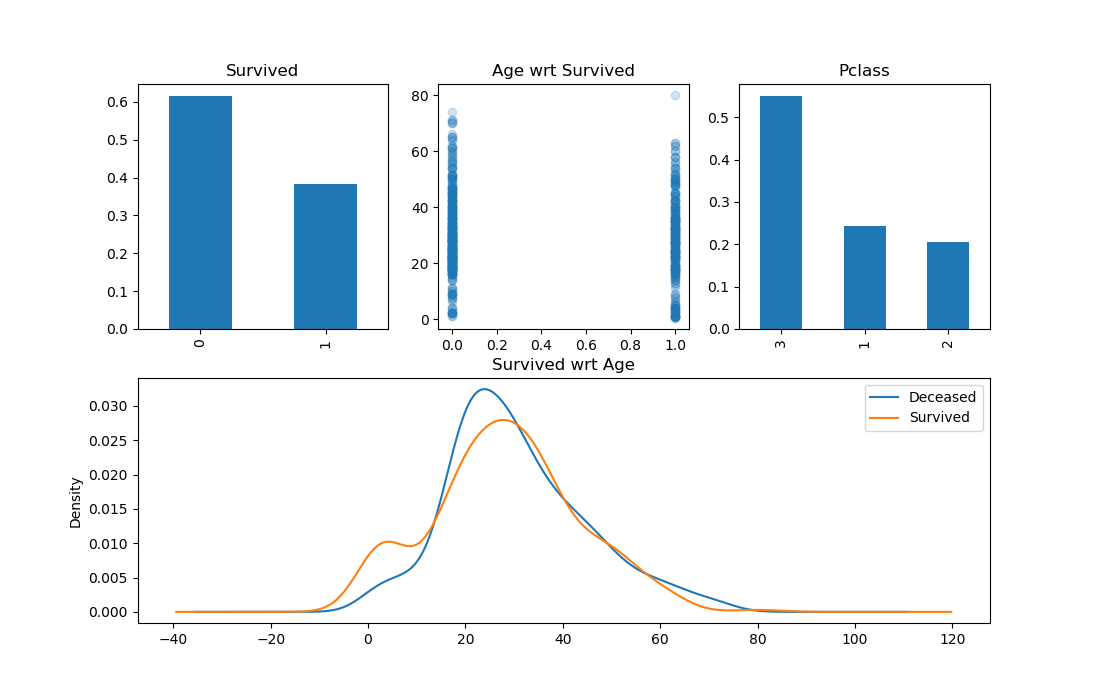
For analyzing the data, a python library called “Pandas” is used. This is a really efficient library used for handling and organizing large datasets.

We have the data stored in the form of a csv file called ‘train.csv’. Another csv file called ‘test.csv’ contains more data of the passengers; this file while later be used for testing our algorithms.



As it is observed that there is a total of 11 columns. It is interesting to note that not all of these parameters would be useful for tuning our algorithms.



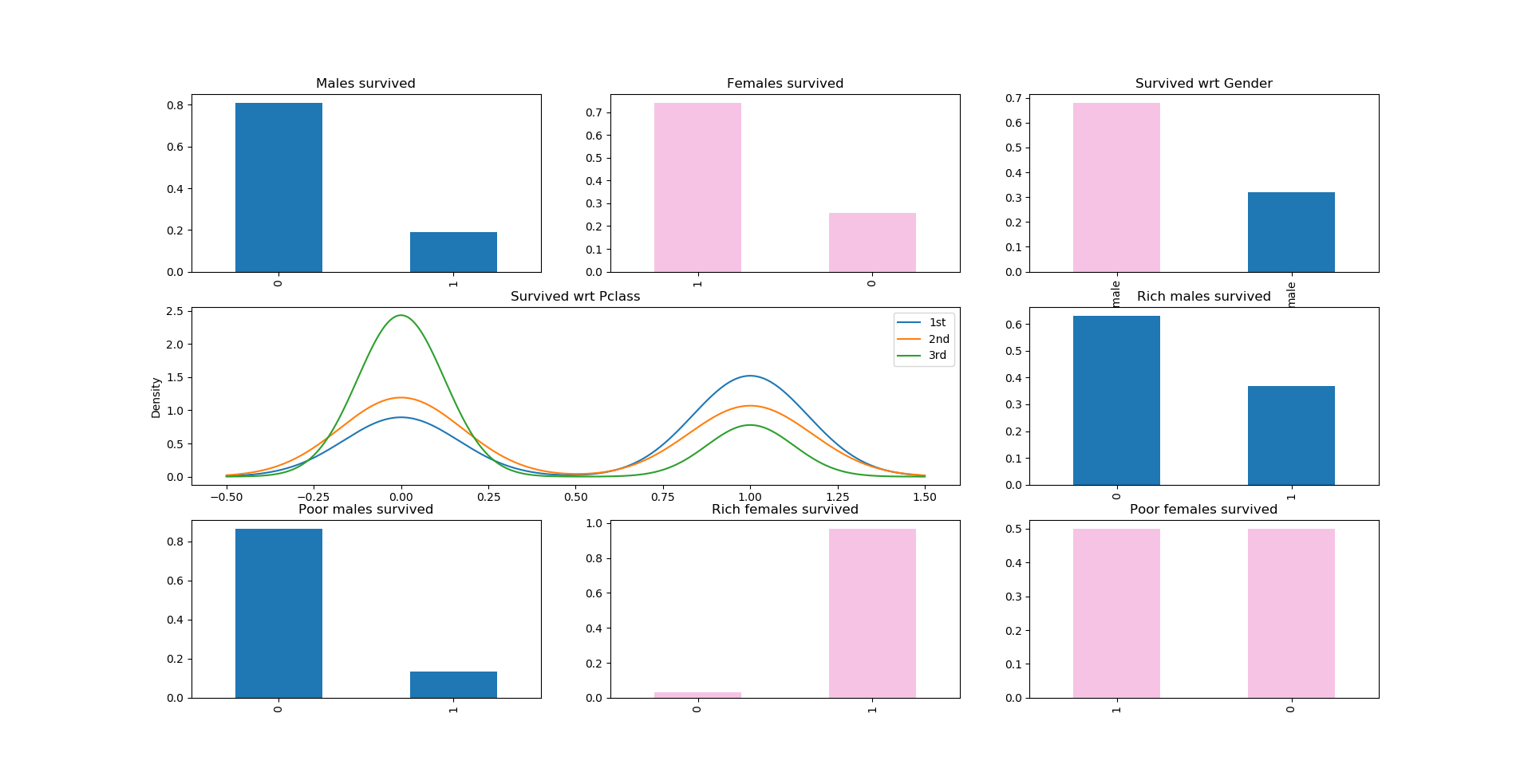


Using the Matplotlib library, various graphs are plotted.

The first plot is the overall survival rate (0 means deceased and 1 means survived).

**Note**: The normalize parameter is turned on in the value\_counts() function to give the percentage instead of the actual numbers.

The second plot as well as the final plot provide an overview of how the age of the passengers affected their mortality.

The Third plot provides an insight to how the Class that the passengers were travelling in affected their mortality. 

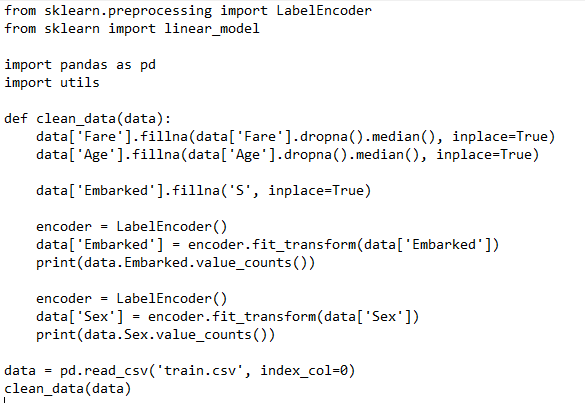
By plotting some more graphs, a clear relation between the gender of the passenger and his/her chances of survival is seen.

**Cleaning the Data**

Before making any predictions or writing algorithms, we need to clean the data. What that means is that we don’t want any null values in the data, because they can hinder our algorithms’ learning patterns.

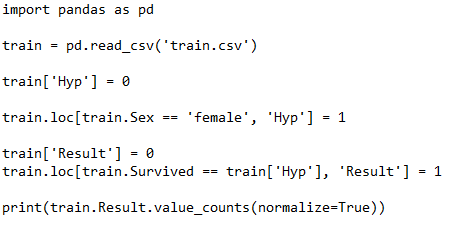
Furthermore, we wish to convert all text data like ‘male’ and ‘female’; ‘S’, ‘C’, ‘E’ to numerical values, as our algorithms only understand numbers, and not strings.

Here we have made use of the scikit-learn library. This is the most used machine learning library after Tensorflow.



**Basic Predictive Algorithm**

We’ve seen that the chances of survival of a female passenger is much greater than that of a male passenger. Therefore, we can create a simple algorithm which works on the following principle: “*If a person is female, then she will survive”.*



Output:

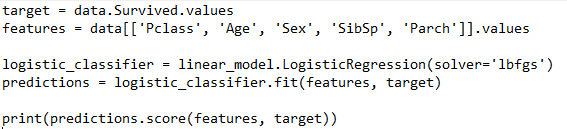


We see that the accuracy of this algorithm is 78%.

This shows the importance of initial analysis of data before making predictions.

**More Sophisticated Algorithms**

1. **Logistic Regression (with Linear Features)**



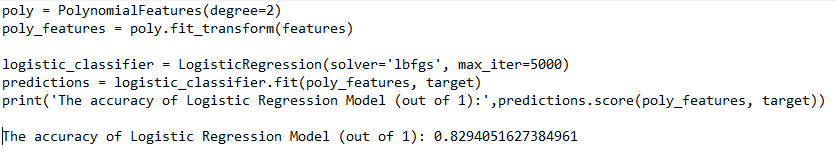
**Output:**



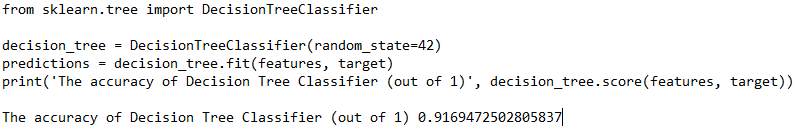
79% is pretty good. It is interesting to note that the increase in accuracy hasn’t been very high from our basic predictive algorithm.

1. **Logistic Regression (with Polynomial Features)**

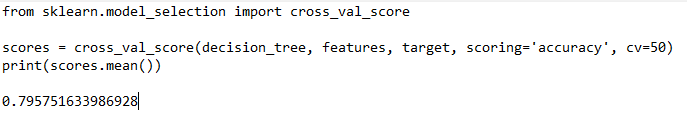
We can also transform the data to represent a polynomial dataset (let’s say of degree=2). It’s possible that the algorithm fits even better with higher degree datasets.



1. **Decision Tree Classifier**

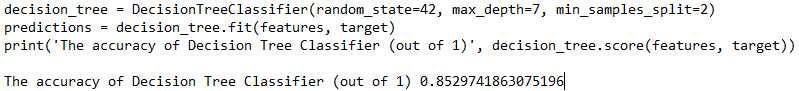


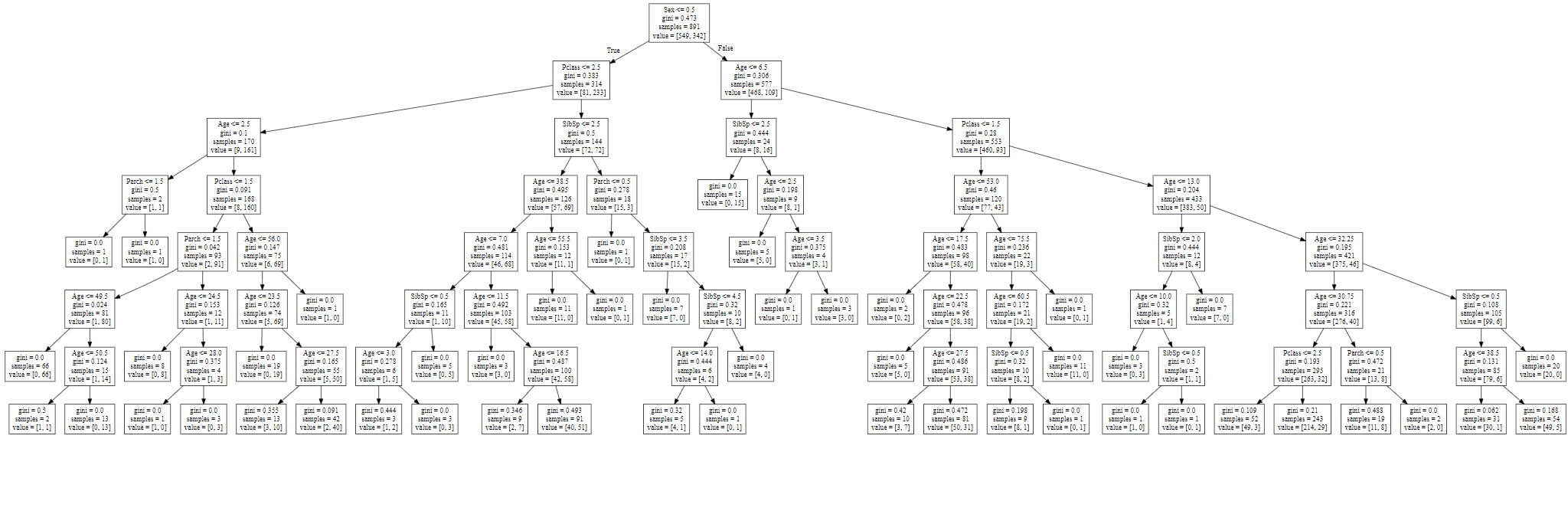
91% accuracy! This looks extremely accurate. However, this actually hides a lot of disparities. It is possible for the decision tree to actually overfit the data and give us such high accuracy.

Let’s Check its *real* accuracy using cross-validation

79.5% is the real accuracy. This clearly shows that the data has been excessively over fitted.

To solve this problem, we tweak the DecisionTreeClassifier





**Conclusion**

We get a maximum accuracy of 85% using our training algorithms on the data set.

It is always recommended to analyze the data before applying predictive algorithms.

**Sources**

* <https://www.kaggle.com/c/titanic>