Artificial Intelligence + Machine Learning Project

TITANIC

Submitted by

Rishika Garg

CS, 2nd Year

BVCOE, New Delhi

Introduction

This project has been made using a dataset named ‘Titanic’ which contains the details of 1309 (possibly fictitious) passengers onboard the infamous Titanic- a luxury steamship which sank after sideswiping an iceberg during its maiden voyage in the Northern Atlantic in 1912.

The task at hand is to train the machine learning model such that based on all or some of the details given in the dataset, the model is able to predict the outcome of a passenger in terms of survival with as much accuracy as can be.

The datasets were processed and certain details or records had to be filtered or slightly changed in order to make the raw data fit for being worked upon by the model.

Various ML models had been deployed in the process, two of which were retained after filtering.

About the dataset

The raw dataset contained the following details:

1. PassengerID (A unique ID allotted to the passengers)
2. Ticket (Possibly a unique ticket ID)
3. Cabin (A cabin number, if allotted)
4. Name
5. Age
6. Embarked (Which port the passenger got on from)
7. Sex
8. SibSp (The number of the passenger’s siblings/spouse(s) aboard)
9. Parch (The number of parents/children aboard)
10. Pclass (The passenger-class)
11. Fare (The amount paid for the ticket)
12. Survived (Whether the passenger survived the tragedy or not)

These attributes or features were filtered out to keep only the ones that affected the output, i.e., the survival state of the passenger.

Features Chosen

The features considered while working with the model are:

- **Sex:** As it had been observed that though the number of female passengers aboard the ship were much less compared their male counterparts, their survival rate was higher
- **Embarked:** As it had been observed that embarkment ‘S’ had the highest number of passengers, but seemingly the lowest survival rate
- **SibSp:** As it had been observed that passengers having this value in the range 0-4 had a better chance of survival than those who didn’t
- **Parch:** As it had been observed that passengers having this value equal to or under 2 had a better chance of survival than those who didn’t
- **Pclass:** As it had been observed that passengers belonging to the higher classes had been clearly given priority over those belonging to the lower ones
- **Fare:** As it had been observed that the ones who paid more were prioritized over those who paid less or didn’t pay at all.

For the purpose of training and testing the model, all the residual features were discarded.

Data Prepping

Also known as data pre-processing, it is the step wherein we alter certain aspects of the data to make it fit for the machine to understand and analyse.

The stages involved in it were:

1. **Encoding –** Changes string data into numerical data.
   Library used: Pandas
   Type: One Hot Encoding
2. **Scaling –** Changes the numerical data with a much larger or smaller size than the rest of the data to be rescaled, such that it’s value is retained but it doesn’t confuse the machine in terms of priority.
   Library used: sklearn.preprocessing
   Type: Standard Scaler
3. **Splitting –** Changes the numerical data with a much larger or smaller size than the rest of the data to be rescaled, such that it’s value is retained but it doesn’t confuse the machine in terms of priority.
   Library used: sklearn.model_selection
   Function: train_test_split

Algorithms Used

Logistic Regression – A statistical model that basically uses a logistic function to model a binary dependent variable. It estimates the parameters of a logistic model (a form of binary regression).

Using this method, the accuracy I achieved was 81.56%.

Decision Tree Classification – A simple representation for classifying examples. It continuously splits data according to certain parameters.

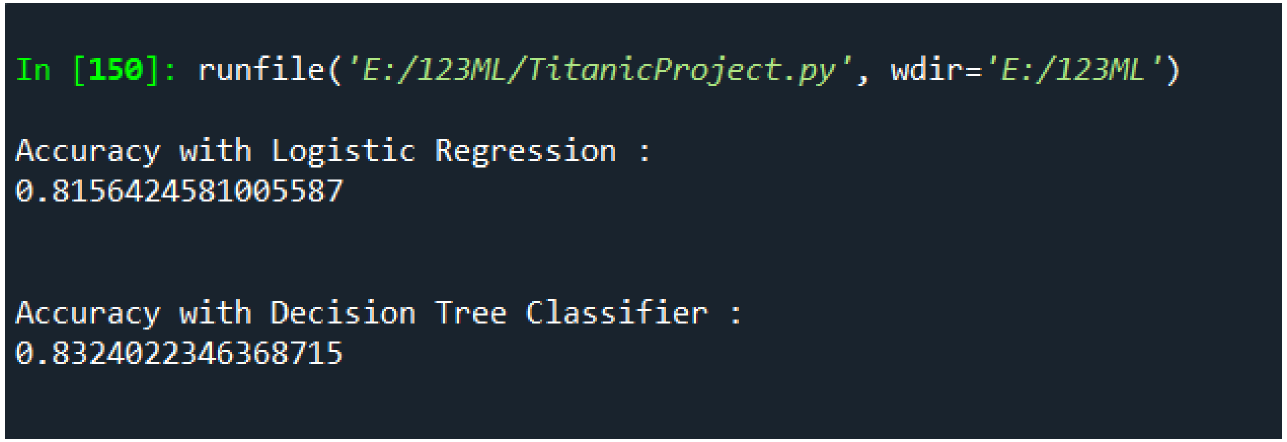
Using this method, the accuracy I achieved was 83.24%.

Output

Both the algorithms gave predictions with good accuracies, but Decision Tree Classification turned out to be better than Logistic Regression by 1.68%.

This could be due to Decision Boundaries, wherein:

Decision Trees bisect the space into smaller and smaller regions, whereas Logistic Regression fits a single line to divide the space exactly into two.



```
In [150]: runfile('E:/123ML/TitanicProject.py', wdir='E:/123ML')

Accuracy with Logistic Regression :
0.8156424581005587


Accuracy with Decision Tree Classifier :
0.8324022346368715
```