

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
|  | Predicting The Survival of Titanic Passengers | | | |  | |
|  | | | |  | |
|  | |  |  | | | |
|  | | Shohaib Shaikh |  | | | |
|  | | 11th February 2023—Titanic Survival Prediction—Guide - Akash Maurya |  | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  |  | | |  |
|  | THE TITANIC | | | | | | |  |
|  |  | | |  |  | | |  |
|  |  | |  | | |  | |  |
|  |  |  | **Titanic**, a British luxury passenger liner that sank on April 15, 1912, en route to New York from Southampton, England, on its maiden voyage.  The largest and most luxurious ship afloat, the Titanic had a double-bottomed hull divided into 16 watertight compartments. Because four of these could be flooded without endangering their buoyancy, it was considered unsinkable. | | |  |  |  |
|  | | | | |
|  |  |  |  |
|  |  |



**INDEX**

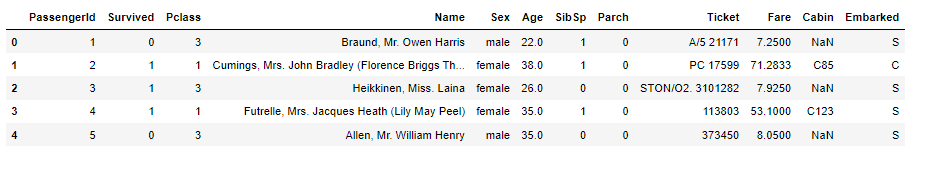
|  |  |
| --- | --- |
| **Content** | **Page No** |
| 1. Objective | 1 |
| 1. About the dataset | 1 |
| 1. Libraries and packages used | 5 |
| 1. Data understanding | 5 |
| 1. Data preprocessing | 6 |
| 1. Exploratory Data Analysis | 7 |
| 1. Splitting the data into new train and test data | 9 |
| 1. Splitting the new train data | 9 |
| 1. Model selection | 10 |
| 1. Final Model selection | 13 |
| 1. Prediction verification by giving random passenger inputs to model | 13 |
| 1. Loading the model to pickle file | 13 |
| 1. Creating web application using streamlit | 14 |
| 1. How to use run the web application | 14 |
| 1. Real time use of the project | 15 |
| 1. References | 15 |

1. **Objective:** 
   1. To build a stable model which is able to predict whether the passenger will survive or not based on different passenger details.
2. **About the Dataset:** 
   1. The dataset consists of mainly two CSV files train.csv and test.csv
   2. train.csv contains all the variables including the dependent/target and independent variables. Whereas the test.csv consists of only independent variables.
   3. The different features and the description of the features are as follows:

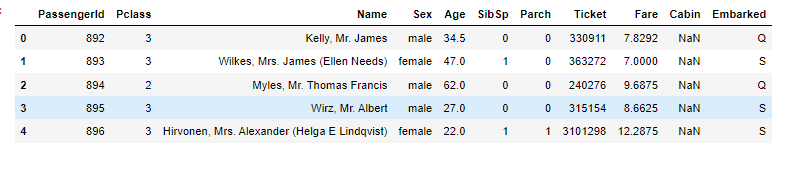
|  |  |
| --- | --- |
| **Feature Names** | **Description** |
| PassengerId | Unique Id of each passenger. |
| Pclass | Ticket class of the passenger. |
| Name | Passenger name. |
| Sex | Gender. |
| Age | Age. |
| SibSp | Number of sibling or spouses travelling. |
| Parch | Number of parents or childrens travelling. |
| Ticket | Ticket number of passenger. |
| Fare | Passenger fare price. |
| Cabin | Cabin number. |
| Embarked | Port of Embarkation. |
| Survived | Survival of passenger. |

* 1. In this dataset, we have categorical as well as numerical variables the categorical variables consist of Pclass, Gender, Age, SibSp, parch, Cabin, Embarked, and Survived. Whereas numerical variables consist of PassengerId, and Fare.

1. **Libraries and Packages Used:** 
   1. Numpy, Pandas, Matplotlib, Seaborn, Sci-kit Learn, Statistics, Pickle.
2. **Data Understanding:**



**Train Data**



**Test Data**

We can observe from the data that train data has the dependent/target categorical variable while the test data does not have the dependent/target variable. Hence our interest is to predict this target variable which has a binary output of whether the passenger is dead or alive.

We will mainly carry out data preprocessing, Exploratory data analysis, splitting the data, training the data, evaluating the data, hyperparameter tuning if required, and finally loading the data as a pickle file so that we can deploy the model.

1. **Data Preprocessing:** 
   1. First of all we will concatenate the train data and test data into a single data so that whatever the preprocessing is done is applied to the whole data hence we will save the time required for preprocessing of the train and test data separately.
   2. Position of the survived column has been changed so that we can easily differentiate between the variables.
   3. **eliminating unnecessary variables:** The are a few variables like passenger ID, Name, and Ticket which have unique records in these variables due to which we will be unable to draw much inference from the data hence these variables can be dropped as the as this will not contribute much in our model building.
   4. **Null Values:** Variables such as Age, Fare, Cabin, and Embarked has null values of 20.09%, 0.08%, 77.46%, and 0.15% respectively. Since Cabin has the highest number of null values which contributes to around 77.46% of data, this variable can be dropped. As it will have no use of this column in our analysis
   5. **Null Values Treatment:** We are left with Age, Fare, and Embarked having null values as Age and Fare are continuous numeric values we can treat these columns with the mean whereas the Embarked column is categorical in nature hence we will treat this column with the mode(frequency).
   6. **Changing Data type of columns:** We can observe that there are two categorical variables Age and Embarked which are in string data type we need to convert it into the numerical data type since our model only understands the numeric data hence we need to convert the data type from string to numeric.
   7. We can observe the mean age of the passengers travelling is 29 years old with a mean fare of 33 dollars with 1309 total passengers onboard.
2. **Exploratory Data Analysis:**
   1. Out of a total of 2240 passengers on board we have data of 1309 passengers after preprocessing hence we will do exploratory data analysis on this number of passengers.
   2. We can observe that most numbers of passengers belong to 3rd class (refer fig 1).
   3. Passengers travelling are more with gender male compared to females (refer fig 2).
   4. Most passengers have embarked from Southampton (refer fig 3)

Finally, we can observe the count of passengers who survived and wo was unable to make it (refer fig 4)

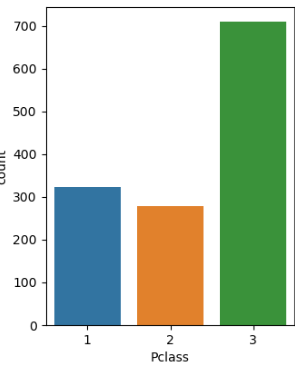
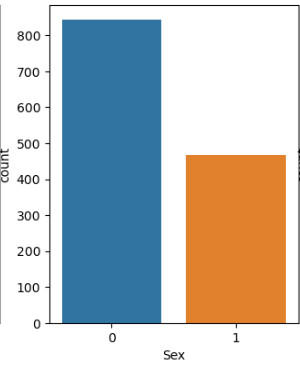
 

Fig 1 Fig 2

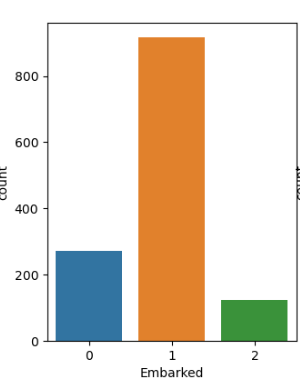
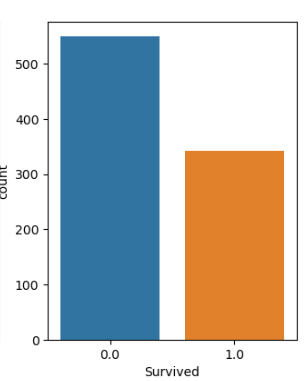
 

Fig 3 Fig 4

* 1. The number of passengers on board was between 20-40 years (refer fig 5)

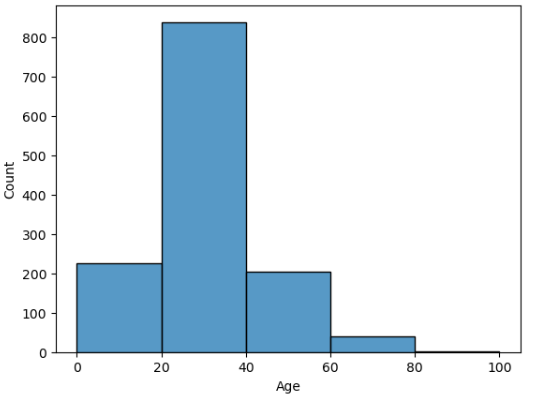
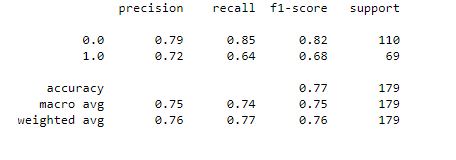
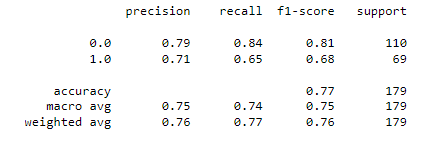


Fig 5

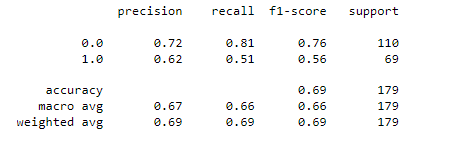
1. **Splitting the processed data into new train and test data:**
   1. Now we have processed and analyzed the data we are ready to split this processed data into new train and test data.
   2. This new train data will be fed to the model to learn the data and once we have found the stable model we may introduce the test data to get see the performance on the unseen data i.e., test data.
2. **Splitting the new train data:**
   1. We are going to split the train data into train and validation set and get results.
   2. Once we have found the results we will try and improve the model using hyperparameter tuning if required.
   3. The reason for splitting the train data into train and validation set is that if we introduce the test data to the model to learn, it will understand few of the data from the test data which may affect the model performance and lead to data leakage.
3. **Model Selection:**
   1. In this Project, since we have a categorical target variable, we are mainly using Logistic Regression, Decision Tree, and KNN (You can explore more algorithms and compare the results).
   2. **Logistic Regression:**
      1. We will import logistic regression from the sci-kit learn library.
      2. Fitting the train and validation set to the model and finding the accuracy on the train set, validation set and finding the standard deviation between these two sets of data. Let’s set the standard deviation as 5% for our evaluation.
      3. We have found out that the train set has an accuracy of 80.62% approximately and the validation set with an accuracy of 76.54% approximately with a standard deviation of 4.08%.
      4. We can observe that the standard deviation between the train and validation set has a standard deviation of 4.08% approximately which is less than our standard deviation.
      5. The f1 score, recall and precision are as follows:



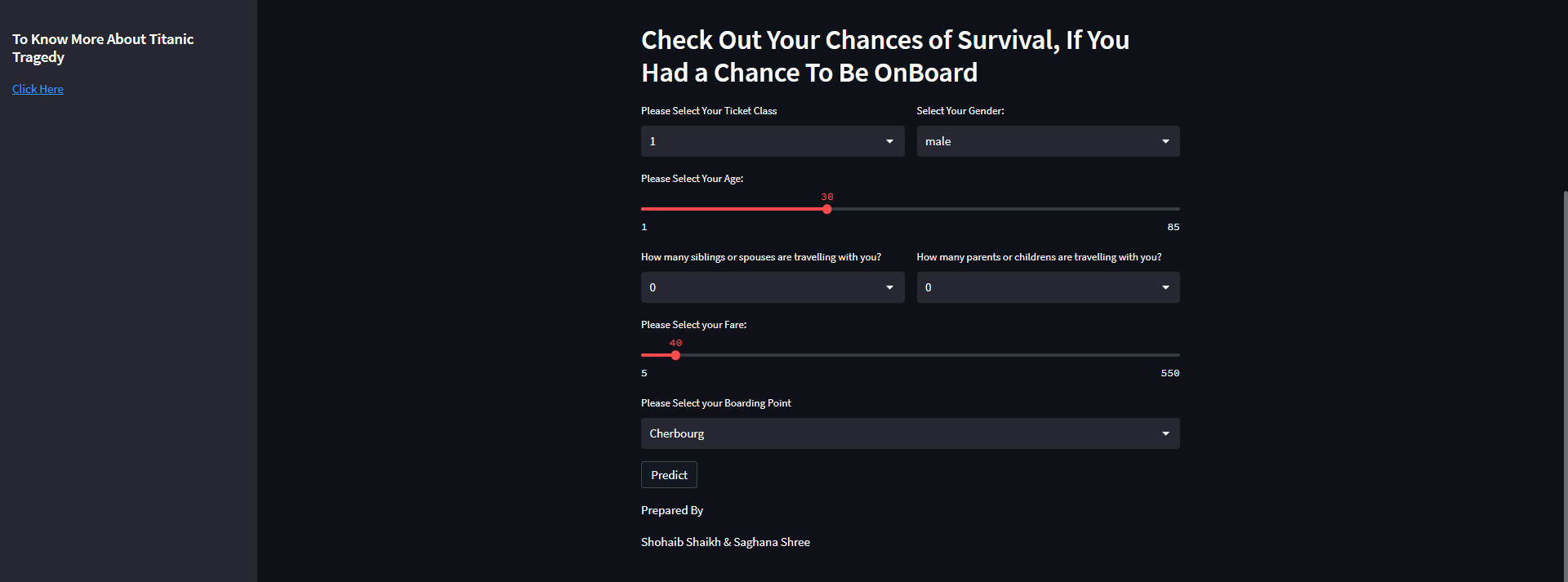
* 1. **Decision Tree:**
     1. As for logistic regression we have followed the same steps for this algorithm with the same evaluation metrics
     2. We have found out that the accuracy for the train set is 98.6% approximately and the validation set 76.54% respectively with a standard deviation of 22.06% approximately.
     3. From this evaluation we can understand that the train set is understood very good on train set and when it comes to the validation set the model performs badly hence the deviation between the train and validation set is more which means when we introduce the validation set the model has a prediction accuracy between 98.6% - 54.48% which is not a good sign for any model to perform.



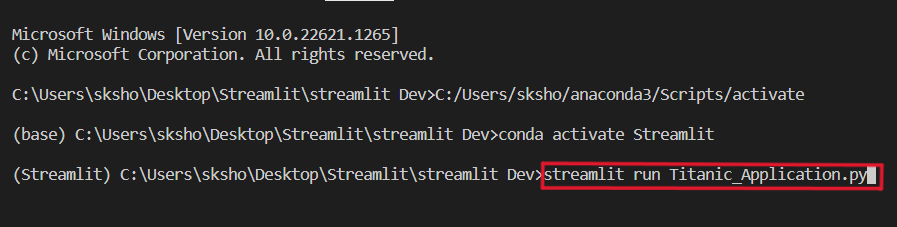
* 1. **KNN:**
     1. As for logistic regression we have followed the same steps for this algorithm with the same evaluation metrics.
     2. We have found out that the accuracy for the train set is 80.76% approximately and the validation set 69.29% respectively with a standard deviation of 11.46% approximately.
     3. From this evaluation we can understand that the train set is understood very good on train set and when it comes to the validation set the model performs badly hence the deviation between the train and validation set is more which means when we introduce the validation set the model has a prediction accuracy between 80.75% - 57.83% which is not a good sign for any model to perform.



1. **Final model selection:**
   1. Now comparing all three model evaluation metrics we can see that the logistic regression is the best fit for our problem statement
   2. So out of these three models we finalize the logistic regression as the stable model and we will see how the unseen test data performs using this algorithm.
2. **Prediction verification by giving random passenger inputs to the model:**
   1. We have predicted the results for the test data i.e., the unseen data.
   2. Also we have provided the input data of a passenger and we have found out that it is able to predict the survival of the passenger.
3. **Loading the model to pickle file:**
   1. After finalizing the logistic regression we have loaded the model into a pickle file using pickle package.
   2. Now we can use this file to deploy in real-time.
   3. We will deploy this model by creating a web application using streamlit which will predict the passenger’s survival by different input parameters of passengers.
4. **Creating web application using streamlit:**
   1. To create a web application we will be using streamlit and IDE used is VS code
   2. Important libraries imported are streamlit, pandas, pickle.
   3. Firstly we have to load the pickle file in the IDE.
   4. We will like to create our main functions where we will use different UI-based switches for our input parameters like title, adding images, and header.
   5. Once we have created all the input parameters we will create a predict button which will show the prediction based on the user-entered input.
   6. Below image shows the UI of the web application created.



1. **How to use/run the web application:**
   1. **Open your IDE.**
   2. In your IDE open a new command prompt terminal.
   3. Type streamlit run [filename].py and press enter.
   4. If your file name is ‘Titanic\_Application’ then the command will be streamlit run Titanic\_ Application.py and press enter. Refer fig below.



1. **Real-time use of the project:**
   1. From the titanic dataset analysis we understood the survival chances of a passenger based on different inputs.
   2. If in future there is another such ship with specifications such as the built of the ship, and the route the ship follows. We can predict the survival chances of a passenger so that we can be prepared for the worst and take necessary precautions for the same.
2. **References:**
   1. <www.britannica.com/topic/Titanic>
   2. [www.stackoverflow.com](http://www.stackoverflow.com)
   3. <https://docs.streamlit.io/>
   4. <https://www.geeksforgeeks.org/>