Space Titanic: Model Comparisons by Binary Brigade

Krit Gupta, Sai Adarsh Kasula, Himani Hooda, Vanshika Gurbani

Keywords Data Science, python, quarto

I. Introduction

The Spaceship Titanic was a giant space cruise ship that launched a month ago. It was on its first trip, carrying almost 13,000 people from our solar system to three new planets that people can live on around other stars.

As it passed by Alpha Centauri, heading to a hot planet called 55 Cancri E, it accidentally ran into a strange space-time disturbance hidden in a cloud of dust. Unfortunately, like the famous ship it was named after, it encountered a disaster. The ship didn't break apart, but nearly half of the passengers

II. About the Dataset

train.csv - Personal records for about two-thirds (~8700) of the passengers, to be used as training data.

Personal Records included basic info like name, age, passenger id, their home planet, their destination planet, their cabin number.

It also included their VIP status, and how much was there expenditure at space titanic's amenties such as Roomservice, Foodcourt, Shopping mall, Spa, VR deck.

test.csv - Personal records for the remaining one-third (~4300) of the passengers, to be used as test data. Your task is to predict the value of Transported for the passengers in this set

II. Data Cleaning

To start things off, we classified which features were numerical and which were catagorical and if there were any data types associated with it?!

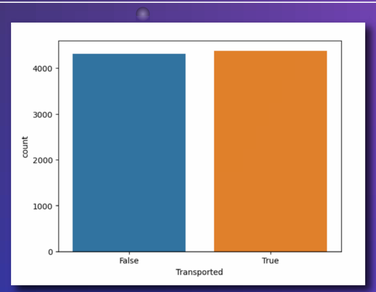
	% nulls
PassengerId	0.0%
HomePlanet	0.02%
CryoSleep	0.02%
Cabin	0.02%
Destination	0.02%
Age	0.02%
VIP	0.02%
RoomService	0.02%
FoodCourt	0.02%
ShoppingMall	0.02%
Spa	0.02%
VRDeck	0.02%
Name	0.02%
Transported	0.0%

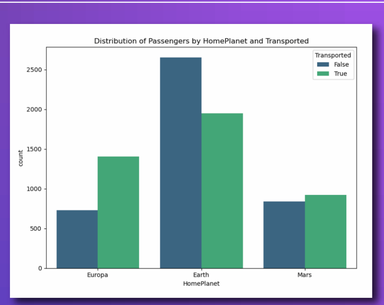
	% nulls
PassengerId	0.0%
HomePlanet	0.0%
CryoSleep	0.0%
Cabin	0.0%
Destination	0.0%
Age	0.0%
VIP	0.0%
RoomService	0.0%
FoodCourt	0.0%
ShoppingMall	0.0%
Spa	0.0%
VRDeck	0.0%
Name	0.02%
Transported	0.0%

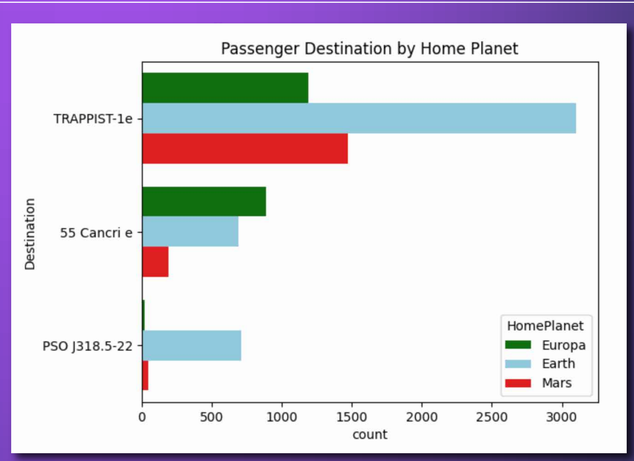
After discovering null values, the approach we took to handle null values was:

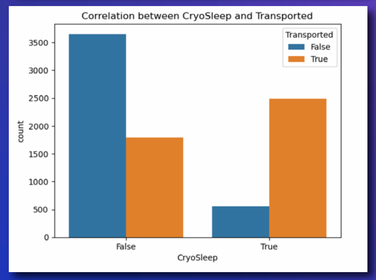
Categorical variables — Replace None values with the most common value in the column
Numerical variables — Replace the values with the mean

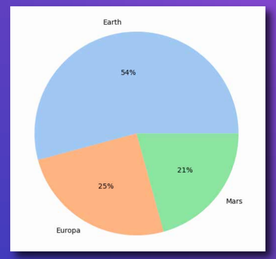
IV. Exploratory Data Analysis











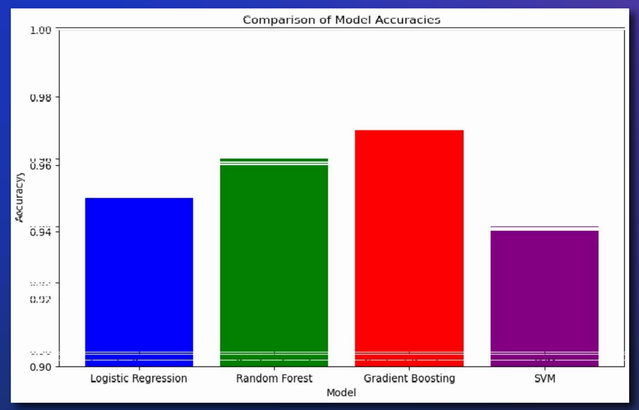
We performed several analysis to understand which features are important to us and which aren't.

Some of the plots are shown above.

V. Modelling

We then trained the test split of our data and fit 4 models:

Logistic Regression, Random Forest, Gradient Boosting. SVM



VI. Results

```
Logistic Regression - Confusion Matrix:
[[666 177]
 [185 698]]

Logistic Regression - Classification Report:
              precision    recall  f1-score   support

       False       0.78      0.79      0.79       843
        True       0.80      0.79      0.79       883

    accuracy                           0.79      1726
   macro avg       0.79      0.79      0.79      1726
weighted avg       0.79      0.79      0.79      1726


Logistic Regression - Accuracy: 0.7902665121668598
```

```
Random Forest - Confusion Matrix:
[[700 143]
 [220 663]]

Random Forest - Classification Report:
              precision    recall  f1-score   support

       False       0.76      0.83      0.79       843
        True       0.82      0.75      0.79       883

    accuracy                           0.79      1726
   macro avg       0.79      0.79      0.79      1726
weighted avg       0.79      0.79      0.79      1726


Random Forest - Accuracy: 0.7896871378910776
```

```
Gradient Boosting - Classification Report:
              precision    recall  f1-score   support

       False       0.80      0.76      0.78       843
        True       0.78      0.82      0.80       883

    accuracy                           0.79      1726
   macro avg       0.79      0.79      0.79      1726
weighted avg       0.79      0.79      0.79      1726


Gradient Boosting - Accuracy: 0.7920046349942063
```

```
SVM - Confusion Matrix:
[[671 172]
 [177 706]]

SVM - Classification Report:
              precision    recall  f1-score   support

       False       0.79      0.80      0.79       843
        True       0.80      0.80      0.80       883

    accuracy                           0.80      1726
   macro avg       0.80      0.80      0.80      1726
weighted avg       0.80      0.80      0.80      1726


SVM - Accuracy: 0.7977983777520278
```

We then printed out classification reports of all the models we fit to understand which one had higher precision, recall, fiscore and accuracy.