EAST WEST UNIVERSITY

Department of Computer Science and Engineering
Fall-2022
Course Code: CSE366
Course Title: Artificial Intelligence

Mini Project Report

On

Spaceship Titanic

Submitted from:

Name: Abdullah al Tamim

ID: 2020-1-60-127

Name: Fatema Akter

ID: 2020-1-60-115

Instructor:

Redwan Ahmed Rizvee

Lecturer, Department of CSE

East West University

Project Description:

The *Spaceship Titanic* was an interstellar passenger liner launched in the year 2912. With almost 13,000 passengers on board, the vessel set out on its maiden voyage transporting emigrants from our solar system to three newly habitable exoplanets orbiting nearby stars.

While rounding Alpha Centauri en route to its first destination—the torrid 55 Cancri E—the unwary *Spaceship Titanic* collided with a spacetime anomaly hidden within a dust cloud. Sadly, it met a similar fate as its namesake from 1000 years before. Though the ship stayed intact, almost half of the passengers were transported to an alternate dimension!

To help rescue crews and retrieve the lost passengers, we are challenged to predict which passengers were transported by the anomaly using records recovered from the spaceship's damaged computer system.

Dataset Description:

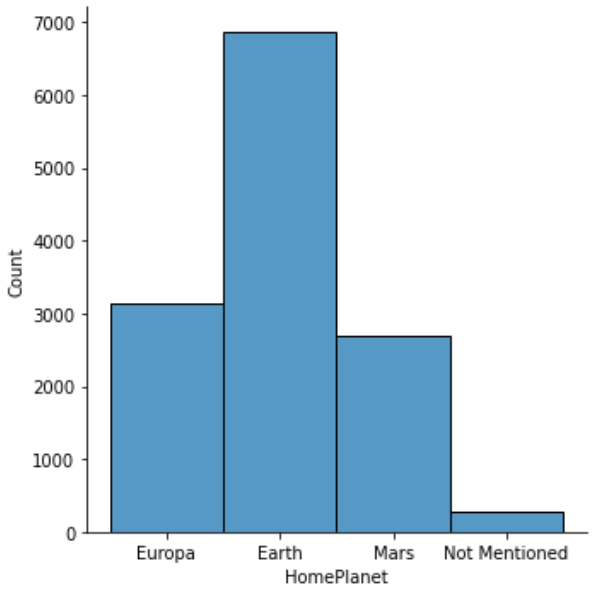
The problem is chosen from Kaggle Competitions. So, the training and test data are already splitted into two CSV files. The Training set has a total of 8693 rows and the testing set has a total of 4277 rows. The columns of the dataset are explained below:

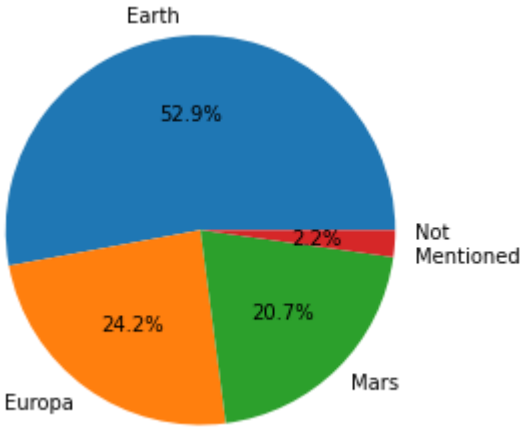
- `PassengerId` - A unique Id for each passenger. Each Id takes the form `gggg_pp` where `gggg` indicates a group the passenger is traveling with and `pp` is their number within the group. People in a group are often family members, but not always.
- `HomePlanet` - The planet the passenger departed from, typically their planet of permanent residence.
- `CryoSleep` - Indicates whether the passenger elected to be put into suspended animation for the duration of the voyage. Passengers in cryosleep are confined to their cabins.
- `Cabin` - The cabin number where the passenger is staying. Takes the form `deck/num/side`, where the `side` can be either `P` for *Port* or `S` for *Starboard*.
- `Destination` - The planet the passenger will be debarking to.

- `Age` - The age of the passenger.
- `VIP` - Whether the passenger has paid for special VIP service during the voyage.
- `RoomService`, `FoodCourt`, `ShoppingMall`, `Spa`, `VRDeck` - Amount the passenger has billed at each of the *Spaceship Titanic*'s many luxury amenities.
- `Name` - The first and last names of the passenger.
- `Transported` - Whether the passenger was transported to another dimension. This is the target, the column you are trying to predict.

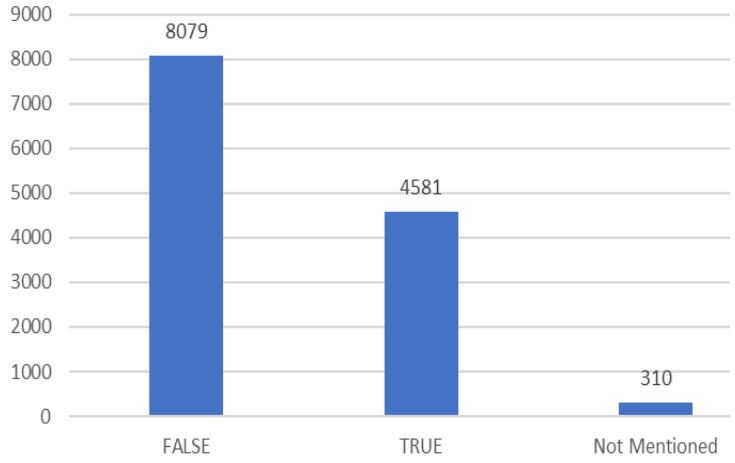
Visualizations of the given data:

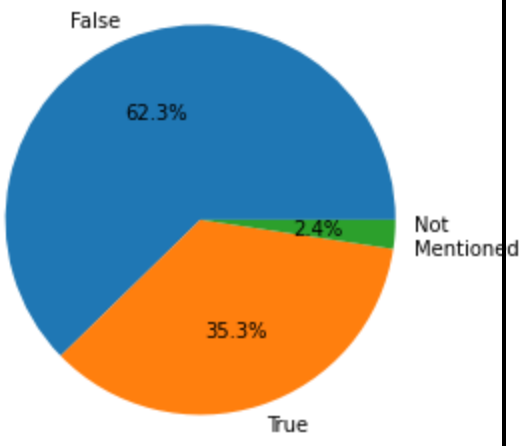
- **HomePlanet:** Histogram and Piechart for HomePlanet are given below,



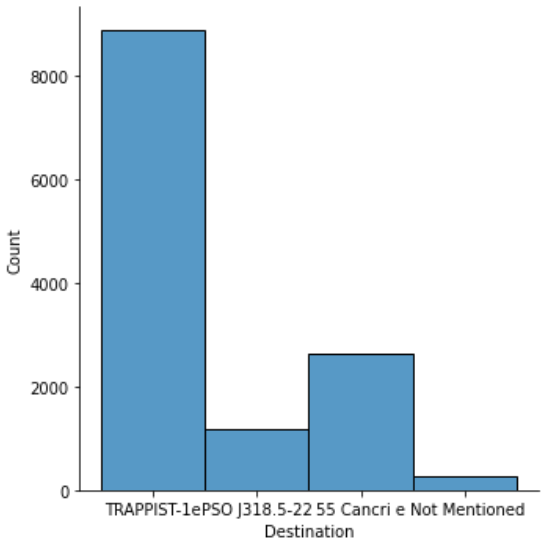


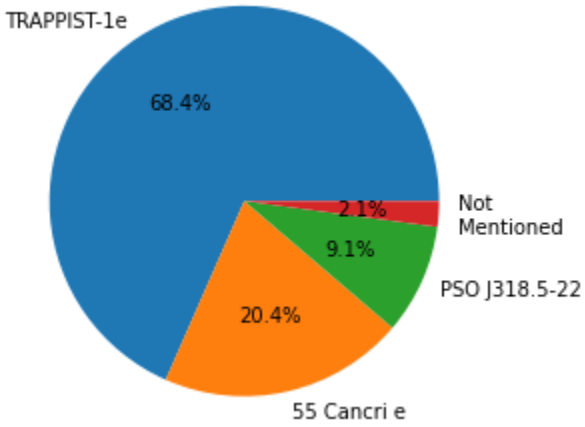
- **CryoSleep:** Histogram and Piechart for CryoSleep are given below,



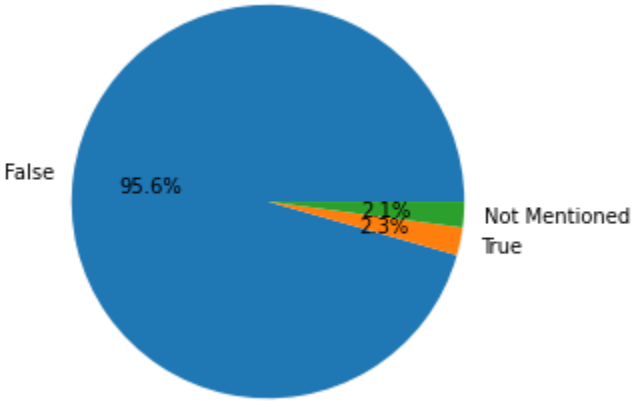


- **Destination:** Histogram and Piechart for Destination of the passengers are given below,

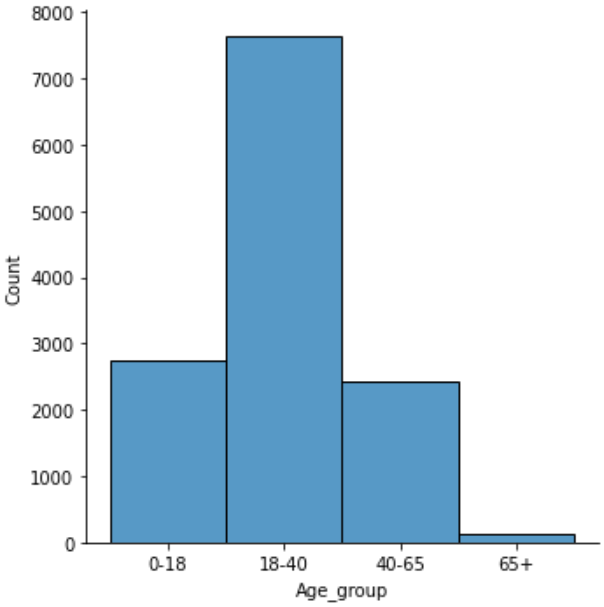




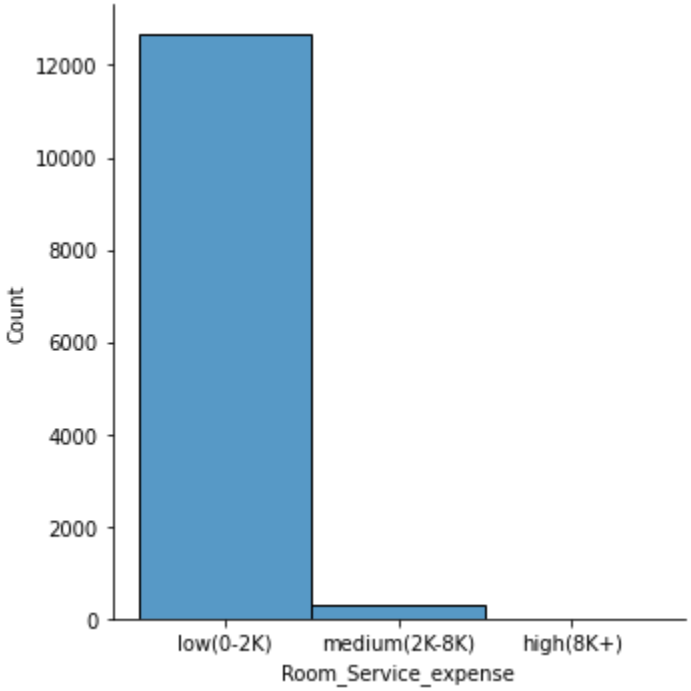
- **VIP:** Piechart for VIP passengers are given below,



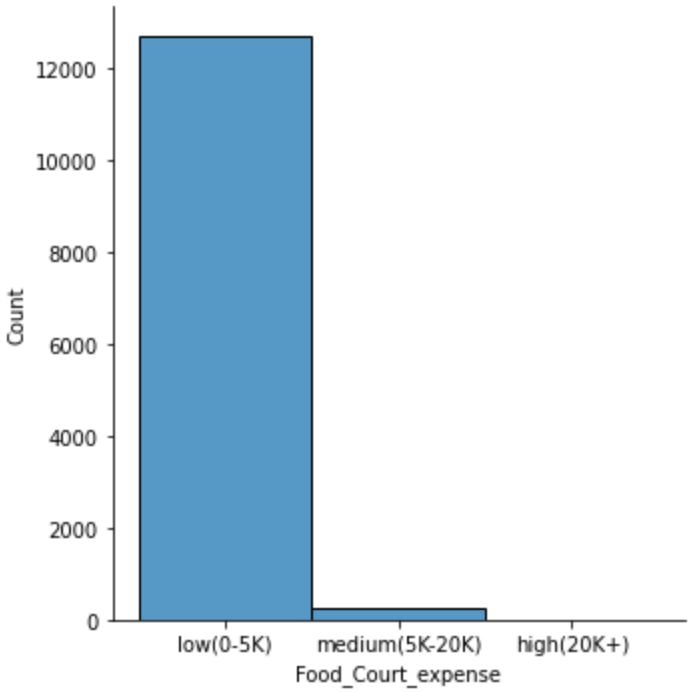
- **Age:** Histogram for age of the passengers is given below,



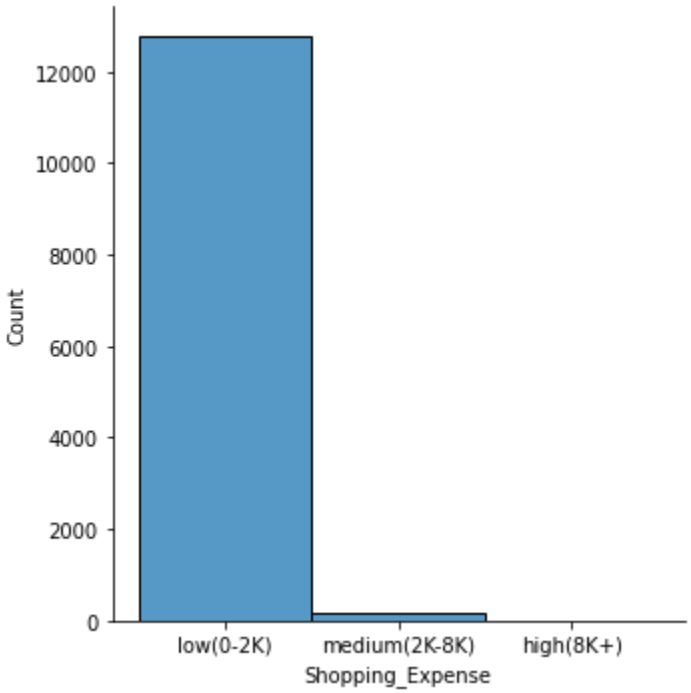
- **Room Service:** Histogram for Room service expense is given below,



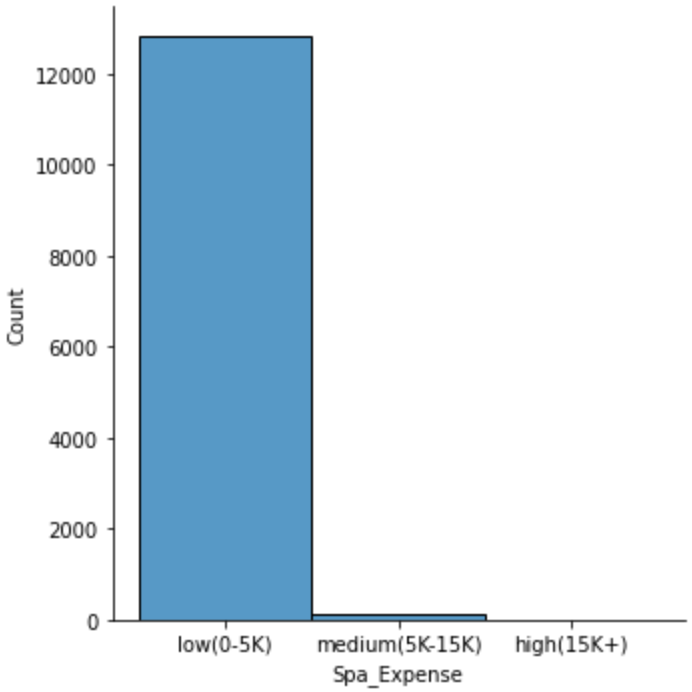
- **Food Court:** Histogram for Food Court expense is given below



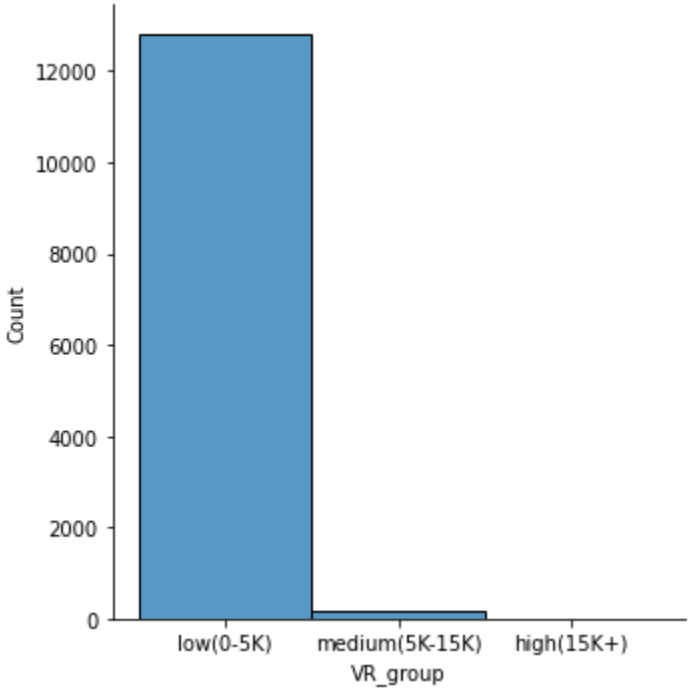
- **Shopping Mall:** Histogram for Shopping Mall expense is given below



- **Spa:** Histogram for Spa expense is given below



- **VR deck:** Histogram for VR deck expense is given below



Data Cleaning and Preprocessing:

- **Feature Selection:** Feature selection is used to make the process more accurate. It also increases the prediction power of the algorithms by selecting the most critical variables and eliminating redundant and irrelevant ones. We have selected 'Homeplanet', 'CryoSleep', 'Destination', 'Age', 'VIP', 'RoomService, FoodCourt, ShoppingMall, Spa, VRDeck', and 'Transported' as features. We didn't take 'PassengerId', 'Name', and 'Cabin' as features because we think that these three are not that relevant to our final result.

- **Replacing empty cells:** Replacing empty cells is very important otherwise, it would decrease the accuracy of our classification models. Since in our data sets we have many empty cells that's why we had to replace the empty cells with a value. We have used two strategies for replacing empty cells. For categorical data, we have used the most frequent value for empty cells, and for numerical data, we have used the mean value for empty cells. To do so we have used `SimpleImputer` class.
- **Encoding**: Encoding is a technique of converting categorical variables into numerical values so that they could be easily fitted to a machine learning model. Also, machine learning models can only work with numerical values. For this reason, it is necessary to transform the categorical values of the relevant features into numerical ones. We have many columns that contain categorical data That's why we encoded those columns. To do so we have used `LabelEncoder` class.
- **Scaling**: Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. Scaling the data makes it easy for a model to learn and understand the problem. We have used data scaling for taking every data into the same range. To do so we have used one type of scaler which is `MinMaxScaler` class.

Description of the classification algorithms:

We have applied four algorithms for predictions. Details descriptions about those algorithms are given below,

- **Decision Tree Classifier:** The decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. Decision Tree is one of the easiest and most popular classification algorithms to understand and interpret. We used entropy to measure the impurity of a node.
- **Logistic Regression:** Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. In logistic regression, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- **GaussianNB:** Gaussian Naive Bayes is a probabilistic machine learning algorithm. Gaussian Naive Bayes is the extension of naive Bayes. This algorithm only needs the mean and the standard deviation from training data.
- **Random Forest Classifier:** Random forests is a supervised learning algorithm. It can be used both for classification and regression. Random forests create decision trees on randomly selected data samples, get a prediction from each tree and select the best solution by means of voting. And, if it has more trees, the result will be more accurate.

Performance analysis:

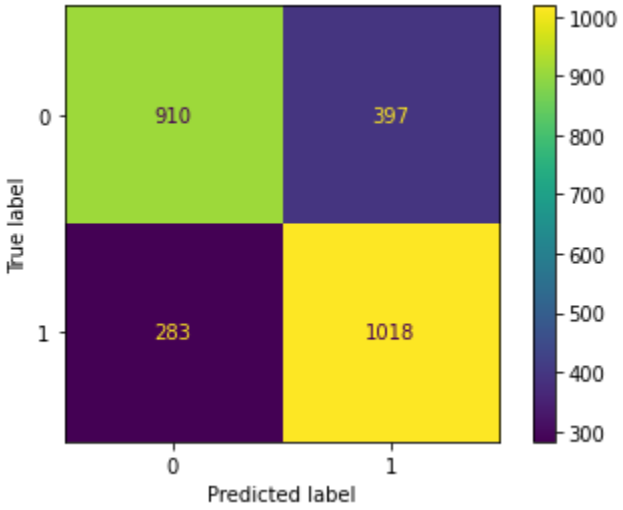
To measure the performance we split the train dataset from Kaggle into two parts, train, and test set with an 80:20 ratio. And then we trained the model using that train set and measured the performance for the test set using `sklearn.metrics` module.

- **For Decision Tree Classifier:**
  Accuracy Score: 0.73926
  Precision Score: 0.71943

Recall Score: 0.78248

F1 Score: 0.74963

Kaggle Score: 0.73696



- **For Logistic Regression:**

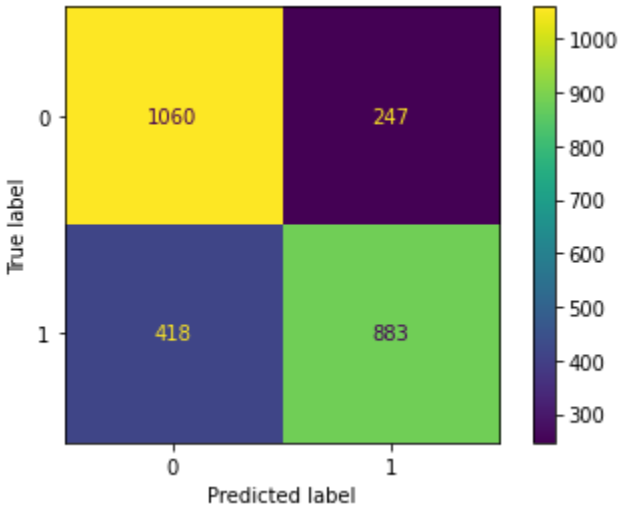
  Accuracy Score: 0.74502

  Precision Score: 0.78142

  Recall Score: 0.67871

  F1 Score: 0.72645

  Kaggle Score: 0.76642



- **For Naive Bayes:**

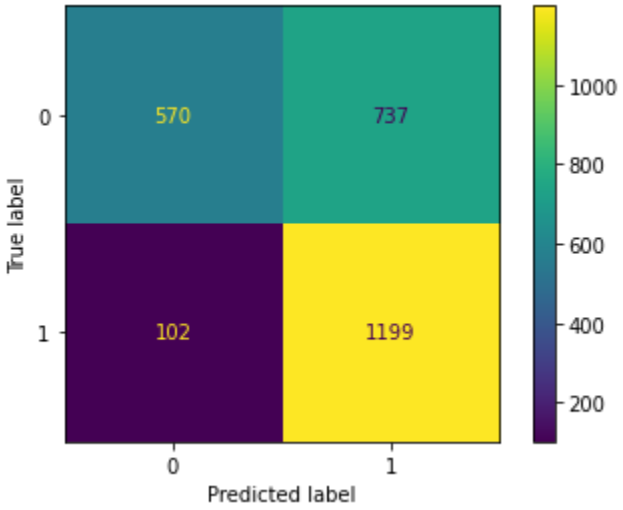
  Accuracy Score: 0.67830

Precision Score: 0.61932

Recall Score: 0.92160

F1 Score: 0.74081

Kaggle Score: 0.71989



- **For Random Forest:**

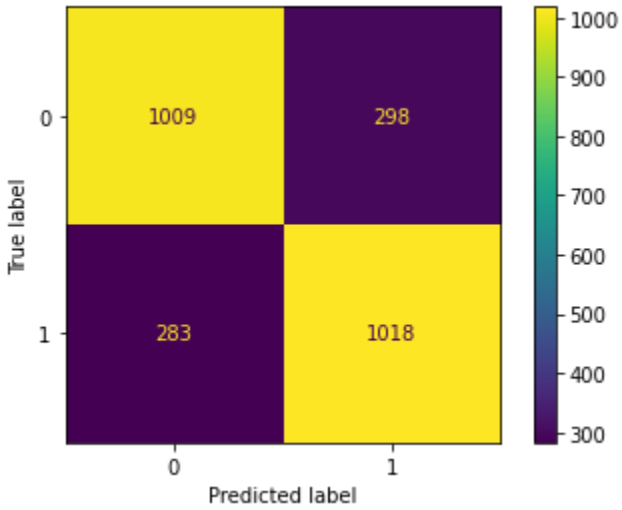
  Accuracy Score: 0.77722

  Precision Score: 0.77356

  Recall Score: 0.78248

  F1 Score: 0.77799

  Kaggle Score: 0.77671



So from the above scores, we can say that all four algorithms predicted quite good results. But among them, Random Forest Classifier had the highest score on Kaggle.

Conclusion:

Data cleaning and preprocessing was the most challenging part of the project. We faced many problems with different data types (Pandas DataFrame and Numpy ndarray) during these processes. But finally, we learned how to deal with these data types. Throughout this project, we have also learned how to apply machine learning algorithms to predict some results. Working in Kaggle was really interesting for us as well as this can be beneficial for our future as data scientists.