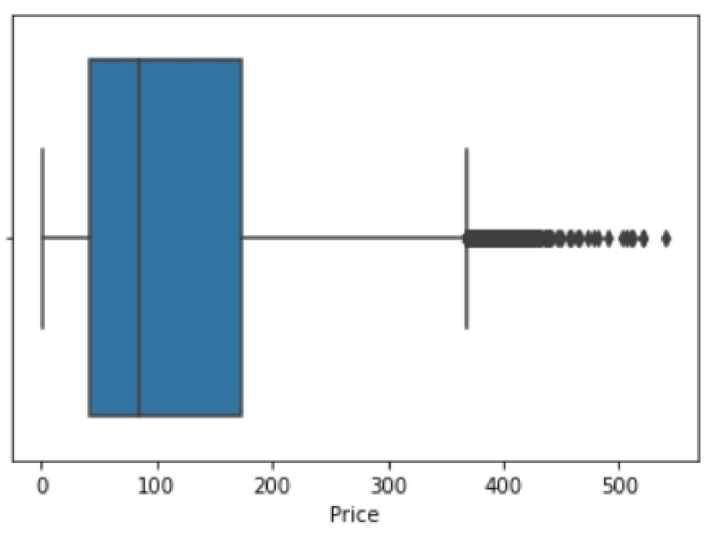
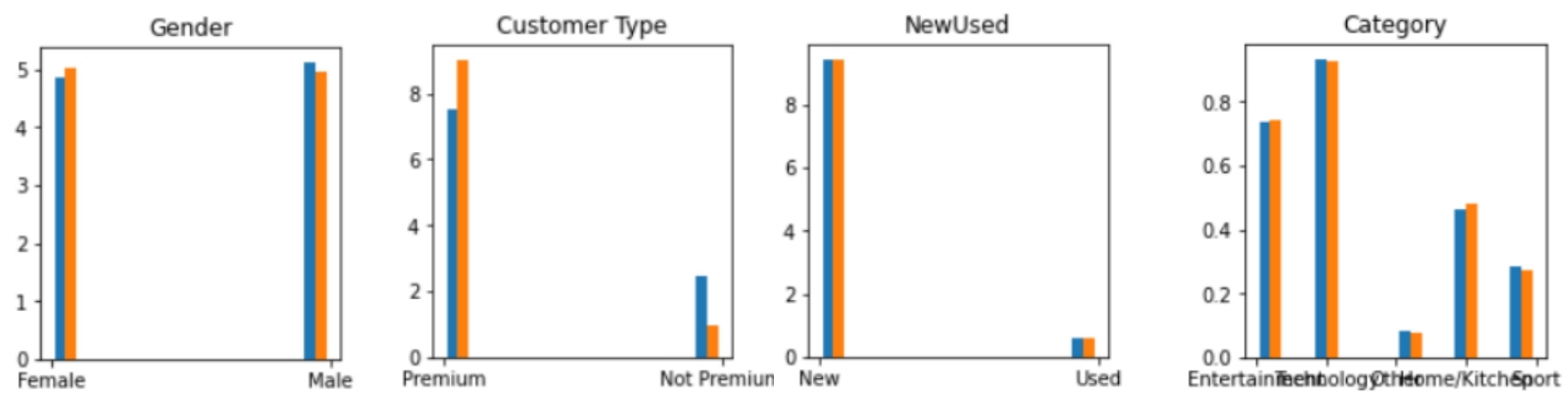
Introduction:
The aim of this project is to develop a customer satisfaction prediction system based on both the historical opinion of the customer and the characteristics of the purchases. The data for model is loaded from a .csv file. This data has 50000 samples of data in 19 categories like Id, Age, Customer Type, Satisfaction etc., which gives a brief idea about the customer and their preferences.

In our code, we started by importing the necessary libraries and reading the dataset. We checked for any duplicate or missing data. There was no duplicate data, but there was some missing data in the "Age" column. We decided to replace the missing age values with the mean value of the 'Age' column.

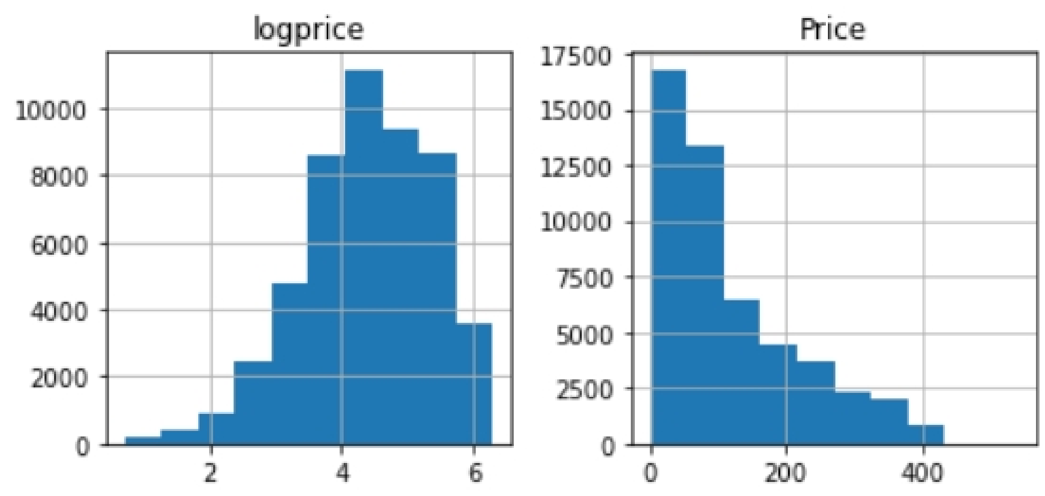
Outliers and missing data:
We searched to analyze if there were any outliers within our data. In most of the columns, the data was within the same range except in the "Price" column which some values were further away. But we decided not to remove the outlier values because they can be crucial for our model to make a precise prediction. In other words, the price of a product is one of the main factors for a customer to be satisfied or unsatisfied.



Categorical and Numerical data:
Once we had our complete dataset, we separated the numerical and the categorical data. We compared each difference between the Satisfied and unsatisfied customers and chose the most relevant ones to keep in our dataset. After comparing the Categorical data, we choose the columns Gender, Customer Type, and Category to keep in our dataset and remove the other data as it has no great significance for Satisfaction or dissatisfaction of the customers. We created a new table that transformed the Categorical data into 0s and 1s numerical using the dummies.

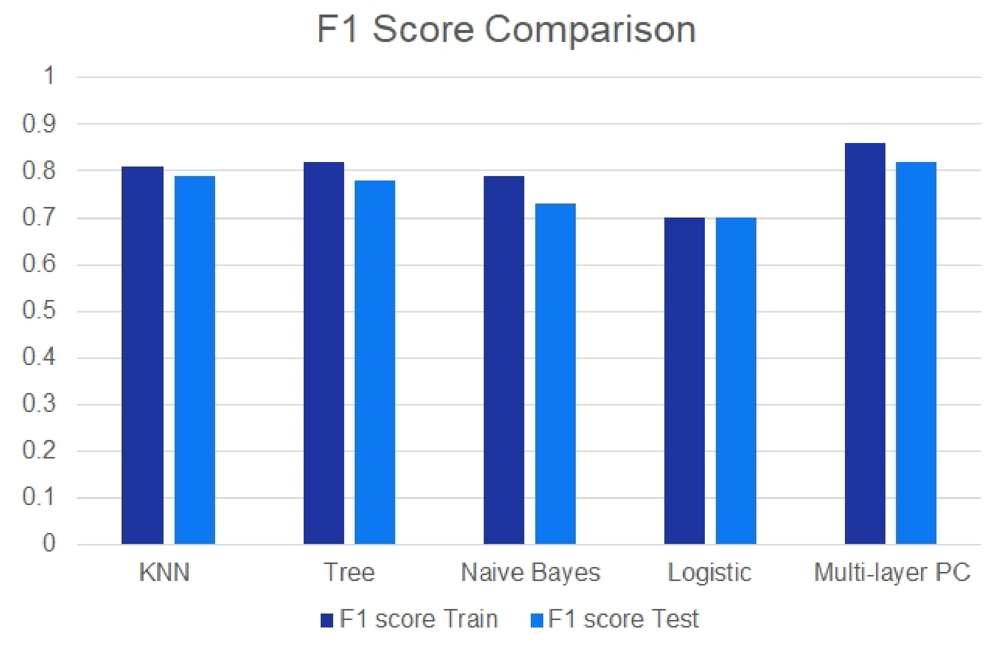


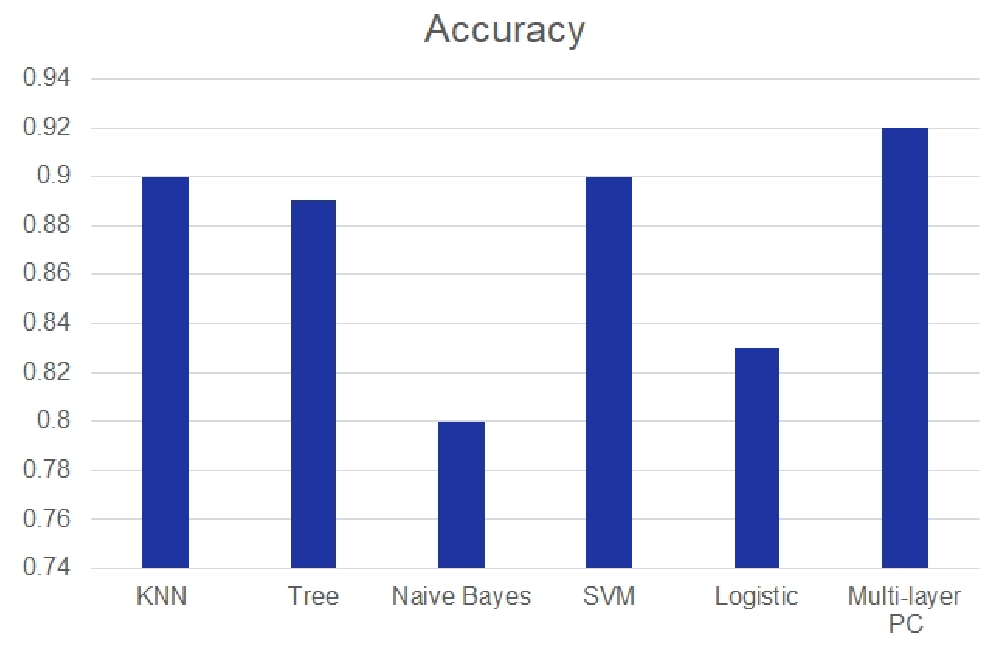
After plotting the histogram for the numerical categories, it looked like the data for the Price, and the delays are not equally divided. So we made the logarithmic and the division expression on them; for the delay days, the graph there not entirely changed, but the graph for the Price was different. After some trial and error and running the model with and without the logarithmic expression, we concluded that we would keep the data and not use the logarithmic one for the Price.



We scaled our numerical data using the standard scaler. We concatenated the scaled numerical dataset with the categorical dataset, which was transformed to numerical to have one whole numeric dataset, which we called the table X. The target which the model should predict has to be in 0s and 1s format(numerical). Still, since the target (satisfactory of the customer) is an object in our dataset, we separate the target column. With the help of the get_dummies, we can generate 0s and 1s for unsatisfied and satisfied customers, respectively. Once we have the all numeric data, we have separated the train set from the test set using 30 percent for the test set and 70 percent for the training set.

We have used different machine learning algorithms on our dataset to find the best f1 score and accuracy. We have used the KNN, Decision tree, Naive Bayes, Logistic, SVM, and Multi-layer Perceptron classifier algorithms to find the best model to use and predict our data. We have used the GridSearch with different arguments to find the best parameters for each algorithm and used the roc function to find the accuracy of our model.





After a long trial and error and using different parameters and observing the comparisons between each model, we concluded to use the Multi-layer Perceptron classifier as our model for our submission. Multilayer perceptron network is a neural network. Our dataset has 50000 data samples which is huge. Neural Networks work better for large amounts of data. We believe this is the reason why Multi Layer Perceptron Network performed well when compared to its counterparts.

Assignment by: Ali Shadman Yazdi