CSE422 Lab Project Report

-

Table of contents

Introduction

The project focuses on predicting whether the deliveries of the e-commerce company reaches on time or not. Meaning being reaching on time and 0 being not reaching on time. We have used three specific models to achieve an accuracy of at least 70% combined. The models are: Naive bayes, logistic regression and neural network. This helps the e-commerce business to identify the errors and increase customer satisfaction.

Dataset description

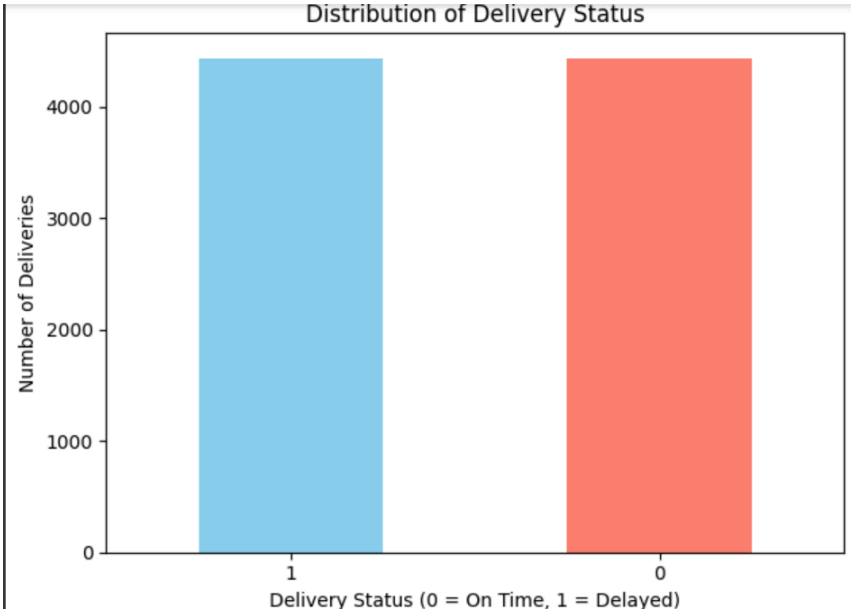
- Features: there are 12 features in the dataset. 11 excluding the ID column
- Classification/ regression problem: the target variable " Reached on time " has values 1s and 0s in respective entries. That means there are variations in the result and doesn't give the perfect accuracy. As classification has at least two or more distinct classes (1,0), it will be a classification problem.
- Datapoints: there are 10999 data points.
- Quantitative features: the columns having numerical values are the quantitative features. There are 8 quantitative features in the dataset (including ID).
- Categorical features: the columns having alphabetic or categorical values are categorical features. There are 4 categorical features in the dataset.
- Correlation of all the features: after applying the heatmap using the seaborn library, we found the pearson coefficient constant between all the features. After the correlation test we can say that the ID feature has the most negative correlation and the Discounted_offer feature has the most positive one. The

negative correlation with ID is unusual and demands further investigation. The zero correlation with Customer_rating is surprising and might indicate that customer satisfaction is truly not related or dependent on ID.

- Imbalanced Dataset: the output feature has two classes 1s and 0s. These variables are not balanced where 1s count are 6563 and 0s count are 4436. Using the following code we have balanced the number of classes by reducing the values of 1s.

```
reduce_1=dataset_1.sample(n=4436,replace=False)
dataset=pd.concat([reduce_1,dataset_0])
dataset=dataset.sample(frac=1)    # dataset shuffle
dataset['Reached.on.Time_Y.N'].value_counts()
```


The bar chart of the balanced classes are given below:



Dataset pre-processing

- Faults:
  - ➔ Null / Missing values and Categorical values: there were no null values in the dataset.
  - ➔ Feature Scaling: the variance values between the features shows significant difference. So feature scaling needed to be applied.
- Solutions

- ➔ Encoding(as required) [show cause]: the categorical values were transformed into numeric values using label encoding because the models don't accept non numeric values.
- ➔ Normalization: we used standard scaling to get the values in a closer range.

Dataset splitting:

splitting the dataset into training (70%) and testing (30%) sets, along with proper preprocessing, is essential for building robust machine learning models that perform well on unseen data. During this process we split the target variable and ID feature for avoiding getting a biased result.
When dividing data into training and testing sets, it's crucial to ensure that both subsets reflect the same distribution of the target classes. This is particularly important when dealing with imbalanced datasets, where one class significantly outnumbers the other. Without proper distribution, a model might become biased towards the majority class, leading to skewed predictions. By maintaining consistent class proportions through stratification, we help the model learn more effectively and make more accurate predictions on new, unseen data.

Model training & testing

We have used three models of machine learning to get better accuracy. After testing all the models except linear regression, we have chosen

- Logistic Regression
- Naive Bayes
- Neural Network

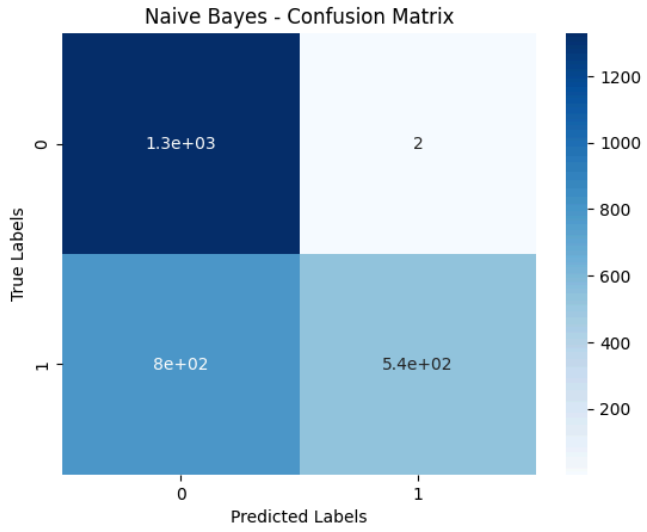
As we have a classification dataset, these three models helped us get as close as possible to accurately predict the outcome.

Model selection/Comparison analysis

- Bar chart showcasing prediction accuracy of all models:

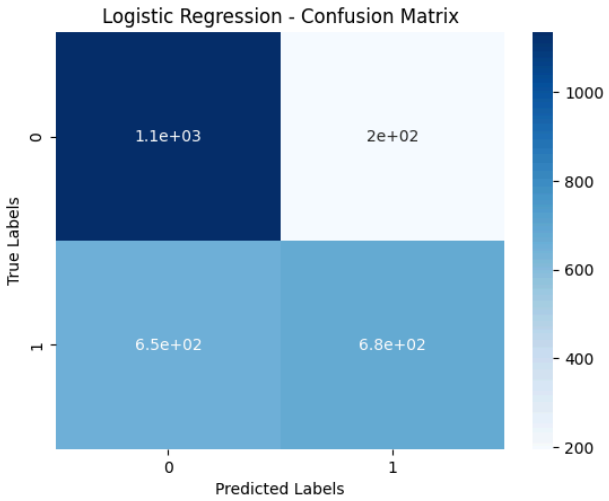
For Naive Bayes model,

1) Accuracy: 70%
2) Precision: 100%
3) Recall: 40%
4) AUC score: 0.71



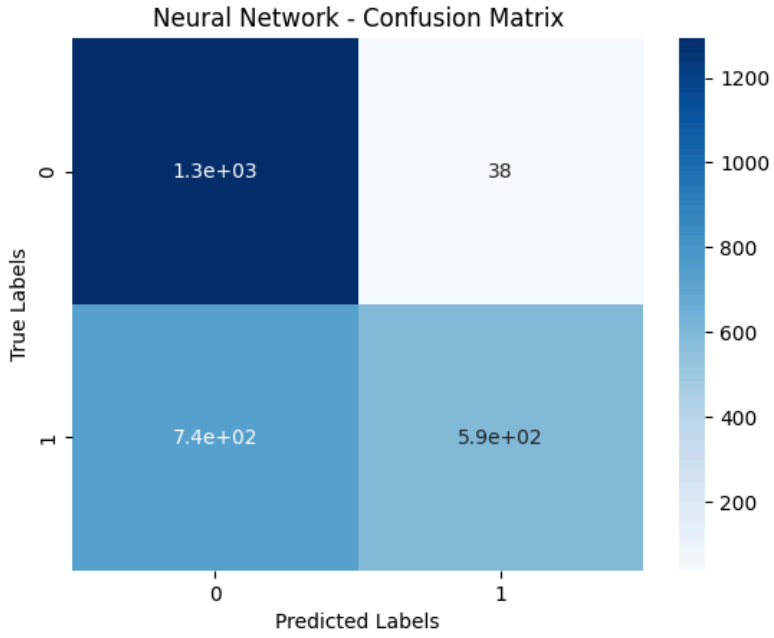
For Logistic Regression model,

1) Accuracy: 68%
2) Precision: 78%
3) Recall: 51%
4) AUC score: 0.71



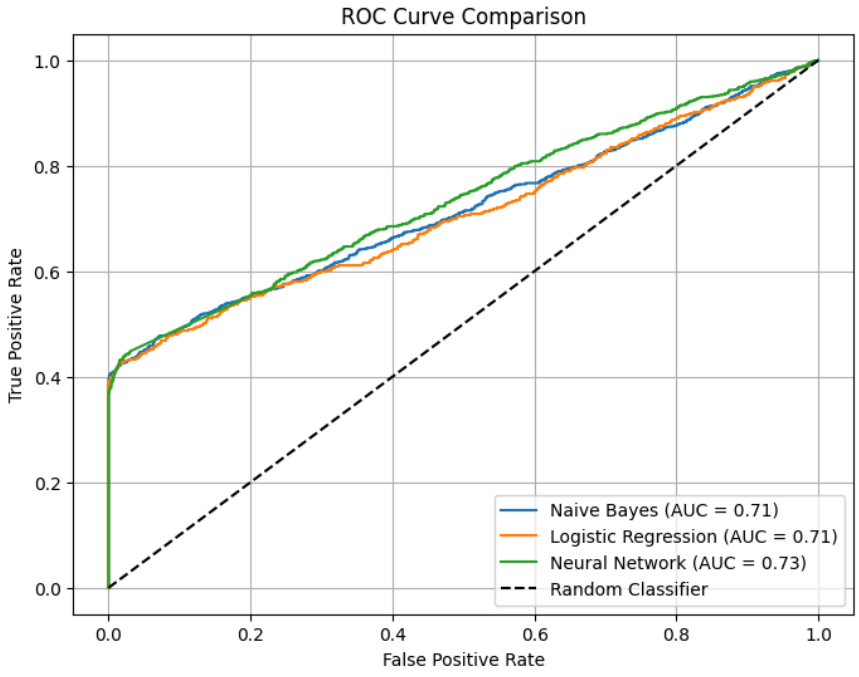
For Neural Network model,

1) Accuracy: 71%
2) Precision: 94%
3) Recall: 44%
4) AUC score: 0.73



Conclusion

After applying all three models we got the graph of accuracy,



From this result, it is clear that there is almost a 30% error in the final result. This happened due to the inconsistent values of 1s and 0s. But this helps us to identify the errors for future improvement. The biggest challenge during making this model was to understand and implement the model, neural network. Working with bars and plots was a bit challenging. Which

methods to work with was also confusing i,e, initially used 1-hot encoding for transforming the categorical values to numerical, later used label-encoding for this transformation.