Analysis of Logistic Regression

- Initially, I searched for a dataset on kaggle.com and found a dataset named "airline passenger satisfaction ". For this dataset, based on the independent feature I can apply logistic regression to the "Satisfaction" column as the target variable.

  Reference link for dataset: Airline Passenger Satisfaction | Kaggle
- Because the "Satisfaction" column is a categorical value with only two options, 'Neutral or Dissatisfied' and 'Satisfied'. As a result, we can use logistic regression to predict airline passenger satisfaction in the future.
- I used Google collab to perform operations on the dataset, and NumPy and Pandas were imported for scientific computations.
- I removed the 'ID' column because the values do not support future predictions and are simply numerical order representations of the customers.
- •I used the duplicate function to search the dataset for duplicates. Duplicates in the dataset will reduce the accuracy of future predictions. To avoid this, we must remove duplicates from the dataset. There are no duplicates to remove in this dataset.
- I also checked the dataset for null values. The 'Arrival Delay' column contains 393 null (blank) values. To avoid having our model affected by null values, I used the best estimate as the mean value of the 'Arrival Delay' instead of null values.
- •I also used Label Encoder to convert the dataset from 0 to the number of classes -1. To transform our dataset from an object to an integer data type. So that the machine learning algorithm can understand and perform the operations on the dataset.
- I also used the dataset's correlation function (Corr()) to examine the relationship between the variables.
- I created a heat map using correlation values. So that we can represent the values visually.
- I used seaborn and matplot.pyplot to create the heat map.
- According to the heat map, the 'Arrival Delay,' 'Departure Delay,' 'In-flight WiFi service,' and 'Ease of online booking' have high correlation values. As a result, dropping for a higher score value is a good idea. After cleaning the dataset considered, I had handled the dataset by splitting the data into trainset of x train and y train. Applying the data into the classifier after splitting the data of the compared measures of optimized model the best accuracy is analysed and predicted that the dataset used been trained with of various parameters of the classifier and analysed with of considered dataset.

Logistic Regression Development:

- Firstly, I used the s k learn import statement train test split to split the dataset into train and test, with 75% of the data for training and 25% for testing.
- After splitting the data, I used logistic regression importing from s k learn because customer satisfaction is the categorical value.
- Using logistic regression I have fitted the X_train and y_train and predicted the y_pred. Here I got the y_pred value is 0.81.
- After that I used a confusion matrix to determine the performance of the classification model of test data, which I imported from sklearn. And also used classification report to get precision, recall, accuracy and F1 score and printed the confusion matrix for testing data and predicted data, precision,recall, F1 score.
- Here I got 0.82 accuracy without applying any optimizers.
- After that I have applied standard scalar .This function standardizes features by removing the mean from them and scaling them to unit variance.
- I have used a pipeline .A Machine Learning pipeline is a method for automating the workflow of an entire machine learning task. It is possible to accomplish this by allowing a sequence of data to be transformed and correlated together in a model that can be analysed to get the output.
- I have fitted the data from x scaled and y using fit function and printed the score using score function.
- After the optimization the score is 0.86

Classifier Desired Results:

• As, of the below tabulation of the Confusion matrix the below scores were observed with of the dataset considered. As the classifier been predicted by F1-Score or that of the Accuracy is 82%.

Classification	Category	Precision	Recall	F1-Score	Support
Logistic Regression	0	0.85	0.83	0.84	18299
	1	0.78	0.81	0.79	14171
	Accuracy			0.82	32470
	Macro Avg	0.81	0.82	0.82	32470
	Weighted Avg	0.82	0.82	0.82	32470

• From, the overall view after applying the classifier to the dataset. I was clear that applying the different classifiers to the different dataset I was confine in applying the accurate classifiers that needs to train and test the data accordingly. By the similar learnings I also known of how data been cleaned, and which variables need to be considered to train the model and also I got to know where to apply the standard scaler to the dataset for fitting the data into the classifier.

• The obtained results from the dataset were compared to that of the optimized model were accuracy shown in better and made the data to trained to its maximum of predictive analysis.

- Comparing the model with and without optimizer, without optimizer I got the score of 0.81
- With an optimizer(using standard scalar and SGD classifier) I got a score of 0.86.
- There is an increase of 0.05 score after optimization.