EXPERIMENT REPORT

Student Name	Archit Murgudkar
Project Name	Data Product using Machine LEarning
Date	10/11/2023
Deliverables	Model - XGBoost Regressor

---

1. EXPERIMENT BACKGROUND	
Provide information about the problem/project such as the scope, the overall objective, expectations. Lay down the goal of this experiment and what are the insights, answers you want to gain or level of performance you are expecting to reach.	
1.a. Business Objective	The primary objective of this project is to develop a data product wherein the users based in the USA will be able to estimate the local flight fare. The users will enter the details for certain fields such as starting airport, destination airport, cabin,etc and then the user-friendly web application will tell the users the flight ticket price. Users will be able to check for the flight fare based on what location and what time period they wish to travel.
1.b. Hypothesis	Users based in the USA will be able to get the flight ticket price for both cases, refundable ticket price and non-refundable ticket price. This will better help users in decision making for which type of flight to board based on their personal interests and situations and the flight prices in both the cases.
1.c. Experiment Objective	Using the web application, users will get to know the flight prices based on the journey details entered by the user. They will be able to check for the refundable flight price and the non-refundable flight price as well.

---

2. EXPERIMENT DETAILS	
Elaborate on the approach taken for this experiment. List the different steps/techniques used and explain the rationale for choosing them.	
2.a. Data Preparation	The data we had was present in multiple zip files. It was first merged together based on the columns and then the total number of rows turned out to be 13519999 across 23 columns. Later, I checked whether duplicates were present in the dataset and not a single duplicate row was found. The data had null values present in the 'totalTravelDistance', 'segmentsEquipmentDescription' and 'segmentsDistance', but these features were not used in the model building. Further, I did a check on the presence of outliers present in the 'totalFare' feature. There were around 75 cases where the price of flight fare was above $4000. This was possibly because of the premium cabin, total journey distance and the time when the user might have booked the flight. Then I created a dataframe with the following columns which are essential for the business use case; ● legId ● flightDate ● startingAirport ● destinationAirport ● isRefundable ● segmentsCabinCode ● segmentsDepartureTimeRaw ● totalFare The feature 'segmentsCabinCode' had a lot of distinct values as multiple journeys involved layovers in it and thus different cabins for different flights in that journey. I decided to lower its discreteness as I took a single value (say coach) wherein multiple flights had the same cabin in a single journey and kept the distinct records as it is.
2.b. Feature Engineering	I created time features such as day, month, year, week from the 'flightDate' variable and time and hour from the 'departureTime' feature. I converted the datatype of 'isRefundable' from boolean to int, indicating refundable as 1 and non-refundable as 0. It was surprising to see how few people had booked for refundable flight tickets as the number was just 191 out of total bookings of 13519999. I split data into training, validation and testing. As we have a lot of data to build a machine learning model, I decided to split the data into training and test in 7:3 ratio and further the training data divided into training and validation in 4:1 ratio. I decided to build a pipeline using the 'Pipeline' library from sklearn. I created a list of numerical and categorical columns which will be used in predicting the target feature. By creating a numeric transformer named num_transformer, I scaled all the numeric values to the same scale. Scaling numeric values to the same scale brings them into one range and thus the model performs with greater accuracy.

2.c. Modelling	The first essential step in modelling is to check the baseline performance of the model. I imported the accuracy metrics 'Mean Sqaured Error' (mse) and 'Mean Absolute Error' (mae). The mse and the mae achieved for the baseline performance was 207.481 and 154.9437 respectively. The baseline model acts as a benchmark and can be compared against the accuracy scores of the models developed using machine learning algorithms. Further, I decided to implement modeling using sklearn pipeline. I created a num_transformer which scaled all the numeric columns. Scaling all the numeric features ensures all the features are scaled on a single range, thus making machine learning models perform better. Then, I developed a cat_tranformer which encoded all categorical columns using target encoding. Later, I created a preprocessor which consisted of all the transformers defined. In the modeling pipeline, I defined the steps, first being the preprocessor which will convert all the data into numeric which will then be fed to ml model. I decided to implement an XGBoost regressor. The reasons behind this is XGBoost is a computationally efficient algorithm and performs efficiently on large datasets. After defining the model in the pipeline, I fit this pipeline on X_train, y_train. The mse and mae values for predicting y_train came out to be 129.5864 and 89.9654 respectively. For y_val, mse scored 129.6141 and mae scored 89.9654. This tells that the model did not overfit as it predicted well on the unseen validation set. The mse score achieved on the testing data using the XGBoost regressor came out to be 129.8985 and mae of 89.9323. After this using the joblib module, I saved the model for further usage.