EXPERIMENT REPORT

Student Name	MAHJABEEN MOHIUDDIN
Project Name	Data Product with Machine Learning
Date	7-11-2023
Deliverables	**Notebook:** Group_10-Mohiuddin_Mahjabeen_24610507_AdaBoostRegressor_Pipeline_Model.ipynb **Model:** AdaBoost Regressor pipeline **GitHub Link :** https://github.com/rohanbrit/adv_mla_asgn3.git

• 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	Explain clearly what is the goal of this project for the business. How will the results be used? What will be the impact of accurate or incorrect results? ## Business Usecase: Approach 3: The fare for a flight with no layover and 1-3 layovers • The project aims to build a predictive model using a Supervised Machine Learning Algorithm to accurately predict the airline's fares that will help the residents of the USA efficiently estimate airfares on their travel. • The model also accurately estimates the fare price of non-stop flights and layover flights which helps end users plan their trip based on the fare prices. • This prediction helps the users decide whether to opt for direct flights or select layover flights based on their budget.

	• The wrong prediction leads to a loss to the user as the user may be charged extra fares.
1.b. Hypothesis	Present the hypothesis you want to test, the question you want to answer or the insight you are seeking. Explain the reasons why you think it is worthwhile considering it, • The fare's are extracted from the airlines websites using four main credentials and those are: • Destination airport • Departure date • Departure time • Cabin type (coach, premium, first, business) • These four main features play a major role in predicting a model. • **Alternative Hypothesis:** The Supervised Machine Learning Regressor Model will help the business produce accurate airline fares using the Pandas matrix module. • The objective of building the model is to analyse what features can be helpful for prediction and how to manipulate those features to get accurate results. • The most important factor to consider while building a prediction model using a pipeline is how well the data is fitted into the pipeline, which can help in predicting accurate predictions.
1.c. Experiment Objective	Detail what will be the expected outcome of the experiment. If possible, estimate the goal you are expecting. List the possible scenarios resulting from this experiment. • **Experient Expected Outcome**: The AdaBoost Regressor model, when data is fitted into the model using the label totalFare, will produce an accurate predictive model that will help airline business owners gain profit by strategizing the fare amount that makes customers comfortably travel within their budget.

- The model will produce the accurate predictions and to make sure model is neither underfitted or overfitted it is tested using the metrics "mean squared error" and "mean absolute error".

Resultant scenarios from the experiment:

- There is a chance that the model learns the data accurately from the lable totalFare.
- The model may become biassed.
- The model may underfit or overfit the data.

• 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	Describe the steps taken for preparing the data (if any). Explain the rationale why you had to perform these steps. List also the steps you decided to not execute and the reasoning behind it. Highlight any step that may potentially be important for future experiments. **Data Preparation:** • The dataset itineraries zip contains 16 subfolders with many numbers of zip files that contained csv's. • To begin the modelling, the initial step is to merge all the files present in the folders. • After merging files, the resultant rows and columns obtained are: 1351999 rows and 23 columns. • While merging the data, it was made sure that all the files were merged into the final data set. • Making a data frame for a dataset to work on building a model. • Making a copy of the dataframe so that if there is any loss on the data, it should not impact the original data. • Assessing the dataframe for null and duplicate values.

2.b. Feature Engineering	Describe the steps taken for generating features (if any). Explain the rationale why you had to perform these steps. List also the feature you decided to remove and the reasoning behind it. Highlight any feature that may potentially be important for future experiments.
	- There were many features that contained a variety of information in one single column. To access that data individually, feature engineering was performed on few of the columns. - The column "segmentsCabinCode" contains the cabin's information for three layovers, and to get the cabin type of flight, feature engineering was performed to extract cabin type data. - On the column "isNoneStop" one hot encoding was performed to encode "true" and "false" as 0 and 1 to analyse the layovers involved during the travel from origin to destination airport. - The column "segmentsDepartureTimeRaw" was split into hours, and minutes to extract the departure time. - The flightdate column is split into day, month, and year to easily use these columns during the training process. - The columns such as "segmentsDepartureTimeRaw", "Fightdate" were renamed as departureDay, departureMonth, departureYear, departureHours, and departureMinutes are used to making the model. - The rest of the remaining columns are dropped as they were not so important in training the model, and if they were incorporated into the model, it may lead to overfitting the data. Naming them explicitly here would occupy a lot of space. - The id's are dropped before training from the dataset; if the id is used to train the data, then the model will focus on learning specific data, but not all that amounts to inaccurate prediction results. **Model Serving:** - After preprocessing, the resultant dataframe has columns such as "fightDay","flightMonth","flightYear", "startingAirport","destinationAirport", "isNonStop","segmentCabinCode","departureHour", "departureMinute", and totalFare. - The variable X is assigned to all the columns except the lable. - The variable y is assigned to totalFare, which is the lable in the dataframe.

- The data is split into train, validation, test set using the scikit learn module train_test_split.
- The train set is assigned 60% of the data, and the model will learn all the patterns of the assigned data. To analyse whether the model is reading all the correct data, the unseen 20% of data is assigned to validation. The matrix test on this data shows whether the model is learning the patterns accurately or not. If the model gives results of overfitting or underfitting, it can be concluded that the model is not learning the entire data and therefore is going to predict inaccurate results.
- The test set with 20% of data does the evaluation on data to analyze the performance of model.
- **Pipeline:** The pipeline provides a way to modelurazie and organise the steps of a machine learning workflow. This makes it easier to understand, maintain, and modify the code.
- The pipeline process captures the sequence of data transformations.
- It also prevents data leakage by ensuring that data preprocessing is applied consistently to datasets.
- Modules such as Pipeline, StandrdScaler, TargetEncoder, are imported from Scikit-Learn.
- The dataset has numerical columns such as departureDay, departureMonth, departureYear, departureHours, and departureMinutes. and the categorical columns as "startingAirport","destinationAirport", "segmentCabinCode".
- The numerical columns and categorical columns are ingested into the pipeline, where numerical columns are passed through the Standard Scaler to scale the data, whereas categorical columns are passed to the Targetencoder to maintain its datatype while passing through the pipeline.
- The numerical columns and categorical columns are ingested into the Columntransformer. Then this is assigned to the variable named as "preprocessor".
- The preprocessor and model are added to the pipeline.
- Then the train data set is fitted into the pipeline, and prediction is performed on the train, validation set and test set to assess the model performance using the scikit-learn matrices mae and mse.
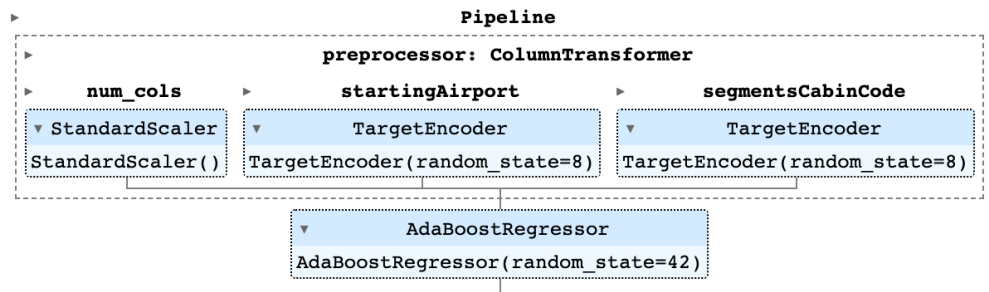
```
                                        Pipeline
        ┌───────────────────────────────────────────────────────────────────────┐
        │  ►          preprocessor: ColumnTransformer                            │
        │  ►   num_cols        ►   startingAirport        ►   segmentsCabinCode   │
        │ ▼ StandardScaler     ▼   TargetEncoder          ▼   TargetEncoder       │
        │ StandardScaler()   TargetEncoder(random_state=8)  TargetEncoder(random_state=8) │
        └───────────────────────────────────────────────────────────────────────┘
                          ┌────────────────────────────────┐
                          │ ▼       AdaBoostRegressor       │
                          │ AdaBoostRegressor(random_state=42) │
                          └────────────────────────────────┘
```

Figure : Model Pipeline process

Package:

• uts_mohiuddin_24610507.data.sets (package)

List of class and functions:

1. NullRegressor class
2. drop_nan_values
3. replace_null_with_Zero
4. mean_null
5. median_null
6. drop_target
7. random_split_train_val_sets
8. save_train_val_sets
9. load_train_val_sets
10. split_train_val_test_random
11. save_train_val_test_sets
12. load_train_val_test_sets
13. print_regressor_scores
14. assess_regressor_set
15. fit_assess_regressor
16. test

• **The Important Features:** Day of Flight, Origin Airport, Destination Airport, and TotalFare are important features that help in acquiring the customer's accurate fare of flights, and the flight date helps the customer bag pack and get ready for their travel.

2.c. Modelling	Describe the model(s) trained for this experiment and why you choose them. List the hyperparameter tuned and the values tested and also the rationale why you choose them. List also the models you decided to not train and the reasoning behind it. Highlight any model or hyperparameter that may potentially be important for future experiments.

Model 1:
AdaBoostRegressor:
The AdaBoost Regressor as an estimator fitting the regressor on the original airline dataset; along with this, it is also fitting additional copies of the regressor on the same dataset. Here, the weights of the instances are adjusted based on the error of the current prediction, and the subsequent regressor is focusing on more difficult cases. This process goes on until the error is reduced.

n_estimators=50, learning_rate=1.0, loss='linear', random_state=42, base_estimator='deprecated'

n_estimators :The n_estimators describes the maximum number of estimators at which boosting is terminated. If we want a model to fit perfectly, the learning procedure is stopped early.

learning_rate=1.0:
The learning rate is the weight applied to each regressor at each boosting iteration. A higher learning rate increases the contribution of each regressor.

loss ='linear':
The loss function to use when updating the weights after each boosting iteration.

base_estimator=*'deprecated'*
The boosted ensemble is built from the base estimator. If the base estimator is set to 'None' then the base estimator is DecisionTreeRegressor and it is initialized with max_depth=3.

Model 2:
Random forest Regressor :
Random forest algorithm is a user friendly and widely popular algorithm that has the ability to deal with the complex datasets.
Each tree is created from different sample of rows and at each note

different sample of features are selected for splitting.
Each of tree makes its own prediction and average of all predictions forms one single result.

Hyper parameters used :

n_estimators=100,min_samples_split=2, min_samples_leaf=1, max_depth=3, random_state=42

The mean squared error for **train** set is: 212.2357755752187
The mean absolute error for **train** set is: 151.34376002647525
ThThe mean squared error of **validation** set is : 214.22372577213747
The mean absolute error of **validation** set is: 151.81166417656698
The mean squared error of **test** set is : 212.75411016413557
The mean absolute error of **test** set is: 151.29873049009962

Model 3: Neural Network: with three hidden layers with regularization

```
 Sequential( ) # First layer
 layer1 = (Dense(70, activation='relu', input_shape=(13,) ,
kernel_regularizer=l2(0.01)))
   # Additional hidden layers
 layer2 = (Dense(64, activation='relu', kernel_regularizer=l2(0.01)))
 layer3 = (Dense(32, activation='relu', kernel_regularizer=l2(0.01)))
# Output layer
top_layer =(Dense(1))
fdsf
```

Model 4: Neural network with 4 hidden layers and 128 highest neurons and least are 38

```
 Model.sequential()
  layer1 = (Dense(128, activation='relu', input_shape=(15,)))
  layer2 =(Dense(70, activation='relu'))
  layer3 =(Dense(50, activation='relu'))
  layer4 =(Dense(38, activation='relu'))# Add the input layer with the
desired input shape
      top_layer=(Dense(1)) # Output layer
```

Mse: 30743.9707
Evaluation:32112.923828125

test prediction :553.176
y_test iloc [0] 594.7

Model 5: Neural networkwith 7 hidden layers, maximum of 140 neurons and minimum of 5
  Model.sequential()
  layer1 = (Dense(140, activation='relu', input_shape=(15,)))
  layer2 = (Dense(120, activation='relu')) # A hidden layer with 250 units and ReLU activation
   layer3 = (Dense(80, activation='relu'))
   layer4 = (Dense(50, activation='relu'))
   layer5 = (Dense(30, activation='relu'))
   layer6= (Dense(20, activation='relu'))
   layer7= (Dense(5, activation='relu'))# Add your output layer
   top_layer =(Dense(1)) # Output layer

Mse : 29818.6641
Evaluation: 30741.080078125
Prediction: 30741.080078125
y_test.iloc[0] 594.7

Models Not Trained:
The models that can be built using linear regression, logistic regression, classification algorithms, and clusters are not useful. The conclusion behind this decision is that linear regression may not produce the desired results, whereas classification algorithms and clusters are not useful here because the dataset has labels with discreated values in classification, whereas unsupervised algorithms do not use labels for predictions, but here in the data set there is a label.

EXPERIMENT RESULTS	
Analyse in detail the results achieved from this experiment from a technical and business perspective. Not only report performance metrics results but also any interpretation on model features, incorrect results, risks identified.	

3.a. **Technical Performance**	Score of the relevant performance metric(s). Provide analysis on the main underperforming cases/observations and potential root causes.

Baseline Model Results:
The mse score of **baseline** model is: 253.240
The mae score of **baseline** model is: 174.318

AdaBoost Regressor:
The hyperparameters used to train model are ***n_estimators=50, learning_rate=1.0, loss='linear', base_estimator='deprecated' random_state=42,***

The mean squared error for **train** set is: 183.40314808886214
The mean absolute error for **train** set is: 137.34291709321658
The mean squared error of **validation** set is : 183.53889261144298
The mean absolute error of **validation** set is: 137.32068331846247
The mean squared error of **test** set is : 183.65170672194114
The mean absolute error of **test** set is: 137.37440768679198

The above results give insight into that, the model beats the baseline.

Though the score slightly overfits the data, the model produces considerably good scores.

Streamlit app business use case non-stop or layover:

A Streamlit app is created to test the correctness of the fare price with layover model. The form is filled as shown in the above image, and its values are passed to the model. Then, the model predicts the fare price of the flight. Eventually, no layover and layover predicted fare price is returned and displayed on the app as shown above.

The results shows that the flight with no layover has less fare price compared to flight with layovers.

3.b. Business Impact	Interpret the results of the experiments related to the business objective set earlier. Estimate the impacts of the incorrect results for the business (some results may have more impact compared to others).

Results:

- The model results gave an insight that, with minor fluctuations in the learning of the data on the AdaBoost Regression model, the model does read the fitted data accurately and therefore has produced accurate predictions on airline total fares.

3.c. Encountered Issues	List all the issues you faced during the experiments (solved and unsolved). Present solutions or workarounds for overcoming them. Highlight also the issues that may have to be dealt with in future experiments. ## Issues face: • To analyze how to merge the files present in the zip folders. • Researched various websites to understand the processes of extracting these files and merging them without losing any data. • Learned neural network process and trained more than 10 models of it but could not proceed with it as Epoch process does not support pipeline process and without epoch model does not read the entire data set repeatedly. • Making stramlit app using various usecases.

• **FUTURE EXPERIMENT**	
Reflect on the experiment and highlight the key information/insights you gained from it that are valuable for the overall project objectives from a technical and business perspective.	
4.a. Key Learning	Reflect on the outcome of the experiment and list the new insights you gained from it. Provide rationale for pursuing more experimentation with the current approach or call out if you think it is a dead end. • The various number of regression models and neural network models were trained to get the least mse and mae values to obtain accurate predictions. The ranking can be given to the trained models based on scores are: 1. AdaBoost Regressor 2. Random Forest Regressor 3. Neural Network Forward Propogation • It can be concluded that the model built using AdaBoost Regressor has produced accurate results. The score of the model indicates that it will produce the airline ticket fare accurately. • Therefore, it can be concluded that no more models are required to be tested.

4.b. Suggestions / Recommendations	Given the results achieved and the overall objective of the project, list the potential next steps and experiments. For each of them assess the expected uplift or gains and rank them accordingly. If the experiment achieved the required outcome for the business, recommend the steps to deploy this solution into production. • The various models were trained to analyze their performance, and it can be concluded that the AdaBoost Regressor algorithm has given the best performance among all the models, as the score obtained is very low compared to other models. • The experiment has achieved the required outcome for the business; this model can be recommended for deployment in production.