Machine Learning
Assignment 1
Report

1-Data Selection and Understanding

Source of the Dataset

The data is collected from **Flight records in Europe**. The csv file was downloaded from Kaggle.

Domain and Relevance

Domain of the dataset is transport .

Features and Records

There are a total of **1000250 records** / rows and **18 Columns** .

Problem Type (Regression/Classification)

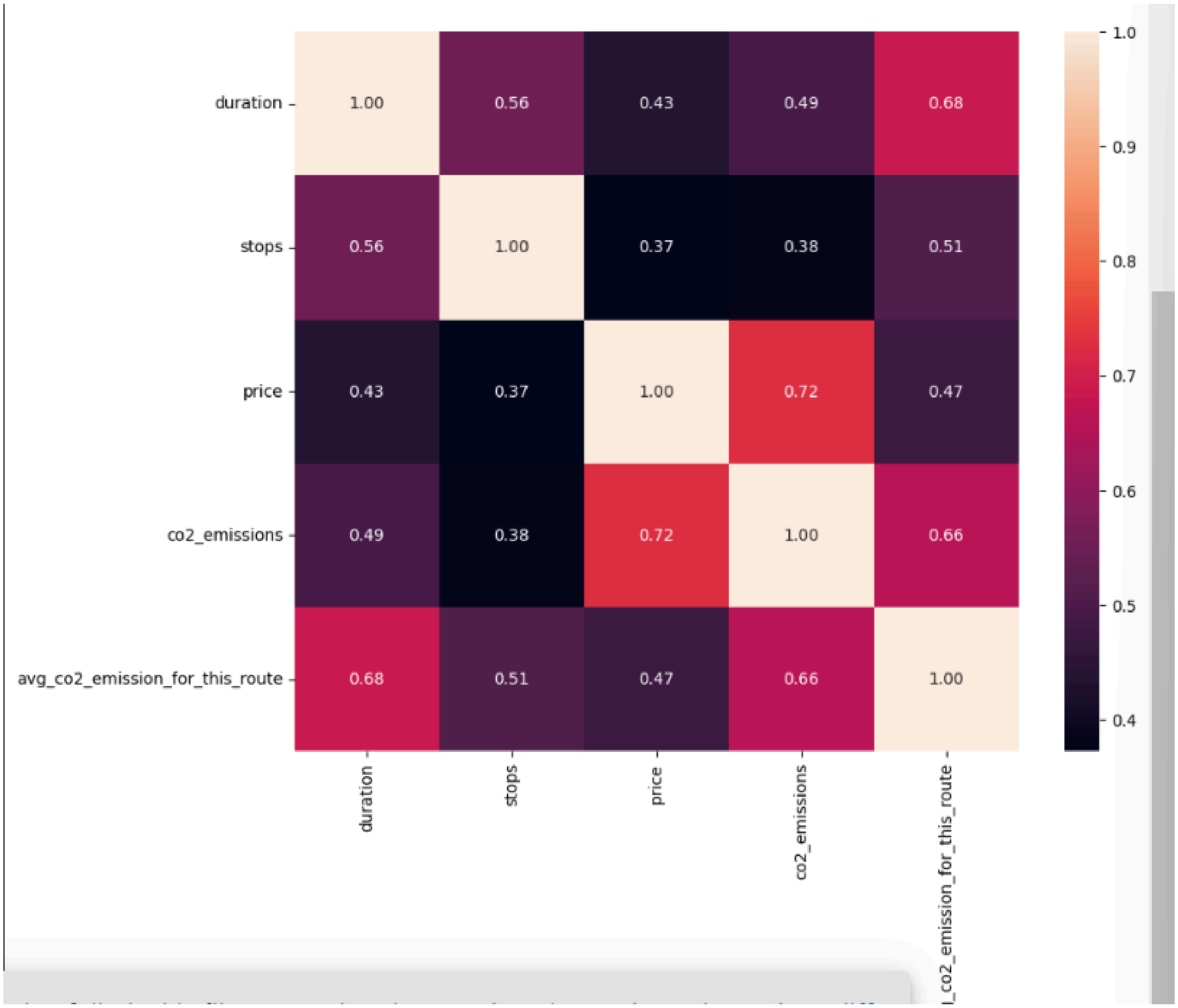
I am going to predict the **price of flight (regression)** hence my target variable based off of features for eg duration of flight , stops , seasons etc .

2-Exploratory Data Analysis (EDA)

Data Visualizations

Here are visualization that helped me the most in understanding the relations between my features and picking the most useful features from useless ones

Correlation Matrix :



	duration	stops	price	co2_emissions	avg_co2_emission_for_this_route
duration	1.00	0.56	0.43	0.49	0.68
stops	0.56	1.00	0.37	0.38	0.51
price	0.43	0.37	1.00	0.72	0.47
co2_emissions	0.49	0.38	0.72	1.00	0.66
avg_co2_emission_for_this_route	0.68	0.51	0.47	0.66	1.00

By analysing from the different EDA visualisations , correlation matrix helped the most in deciding which features are more important and how features are related to each other , I have also decided by analysing my correlation matrix that I can predict (regression) values for

- Flight Duration
- Price
- C02 Emission

I can conclude these relations from the correlation matrix

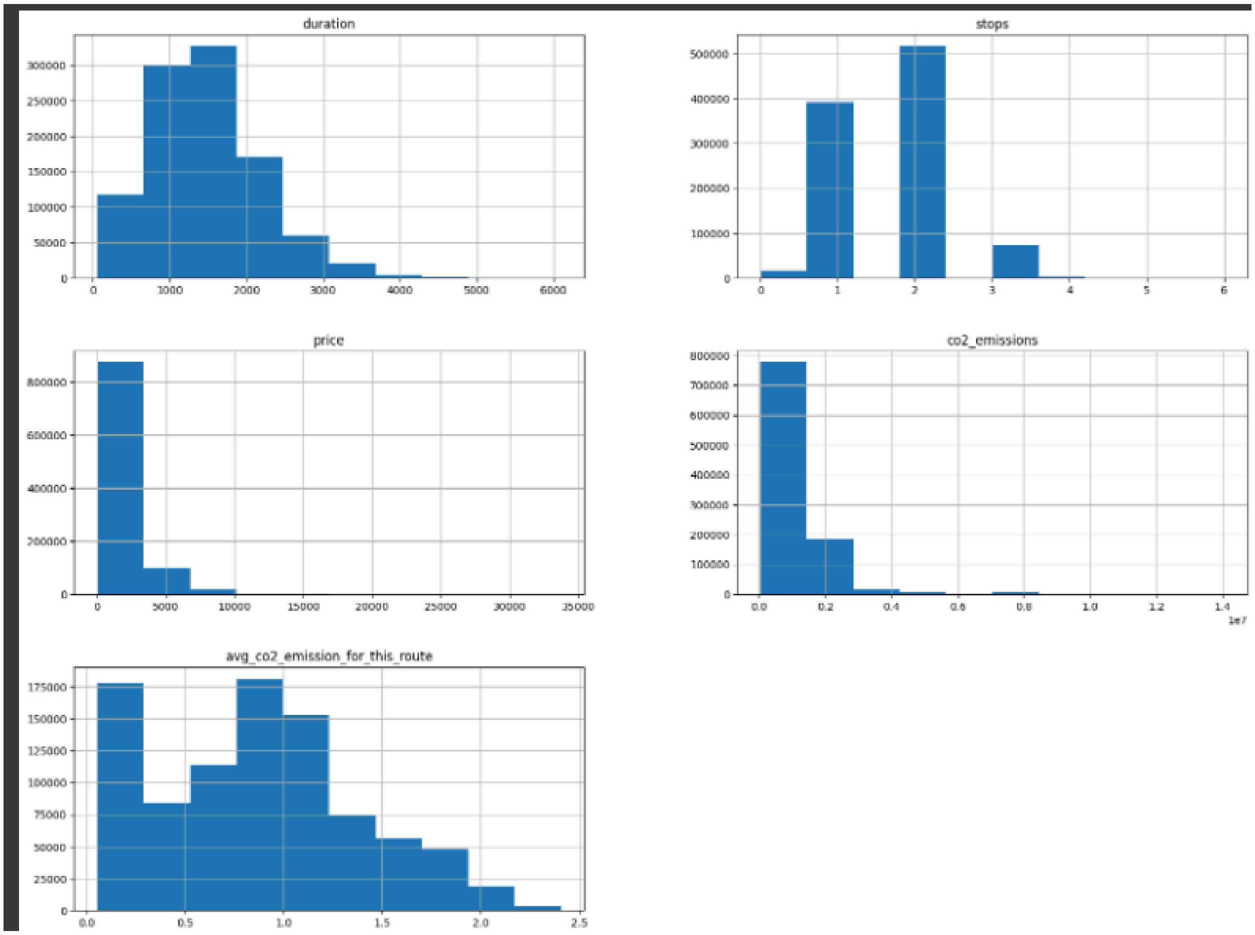
1. stops and co2 emission have moderate linear relation with flight duration and it makes sense too that more stops equal more fuel consumption and more duration

2. more fuel consumption equals more price and more co2 emission too and more duration too and vice versa

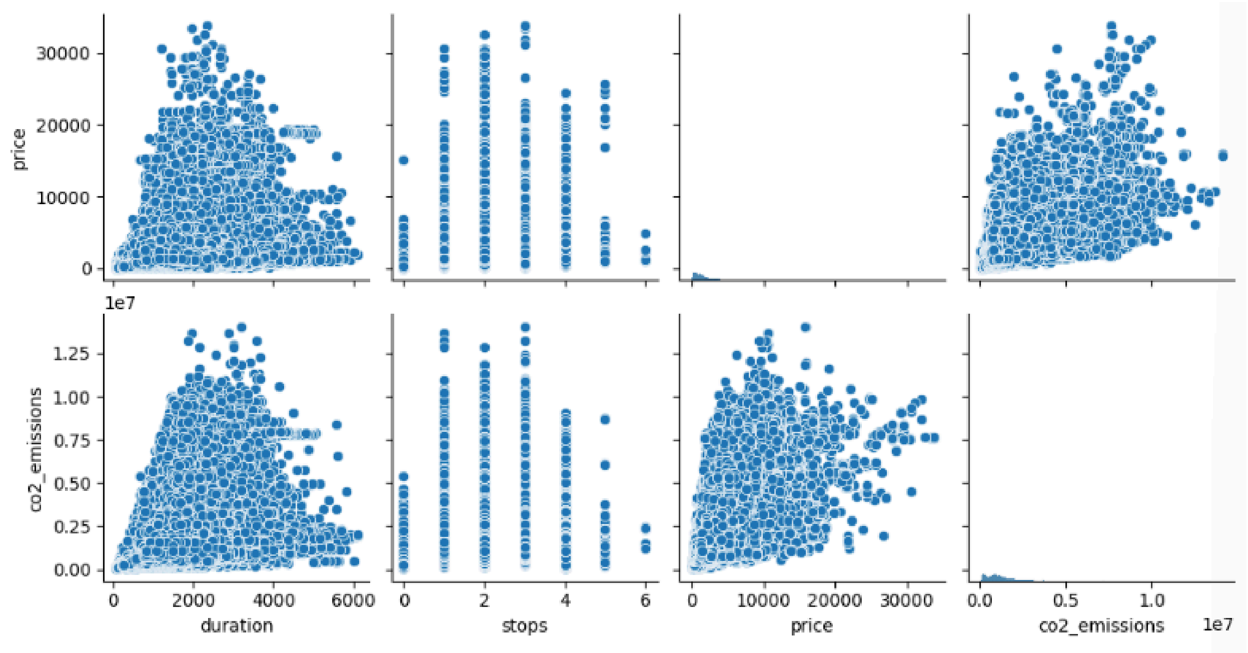
3. price and co2 emission have a direct high correlation , makes sense too because price increases by more fuel consumption

Histograms of Numeric Features

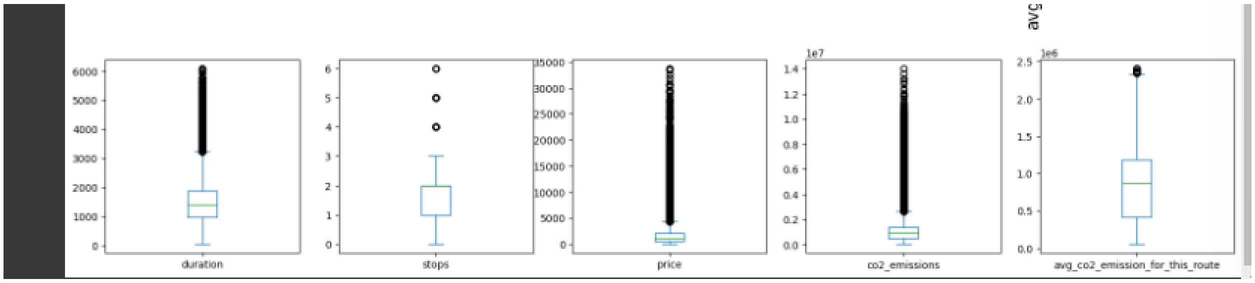
The histograms helped in analysing the frequency distribution of data



Scatter Plots



Boxplots for Outlier Detection

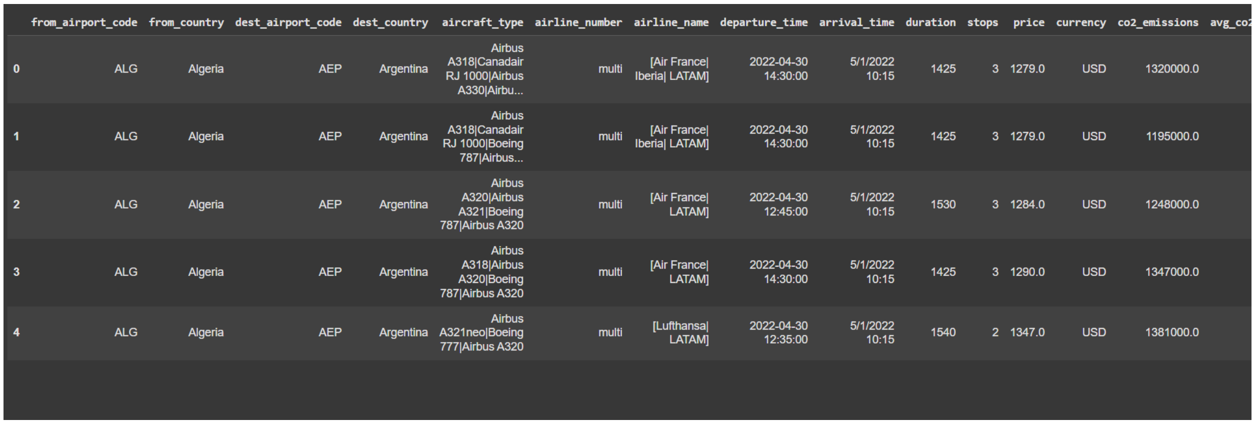


- Duration has a lot of outliers indicating many flights may have unusually long durations due to several reasons
- Stops have most values in the box plots; only a few 3 values were high .
- Price and Co2 emission are well concentrated in their box plots hence being normal

3- Key Findings from EDA

Data Preprocessing and Feature Engineering

Here is a summary of my columns in the dataframe before preprocessing:

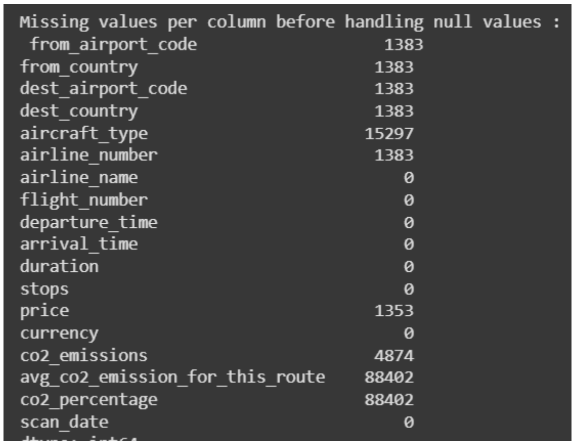


	from_airport_code	from_country	dest_airport_code	dest_country	aircraft_type	airline_number	airline_name	departure_time	arrival_time	duration	stops	price	currency	co2_emissions	avg_co2
0	ALG	Algeria	AEP	Argentina	Airbus A318|Canadair RJ 1000|Airbus A330|Airbu...	multi	[Air France| Iberia| LATAM]	2022-04-30 14:30:00	5/1/2022 10:15	1425	3	1279.0	USD	1320000.0	
1	ALG	Algeria	AEP	Argentina	Airbus A318|Canadair RJ 1000|Boeing 787|Airbus...	multi	[Air France| Iberia| LATAM]	2022-04-30 14:30:00	5/1/2022 10:15	1425	3	1279.0	USD	1195000.0	
2	ALG	Algeria	AEP	Argentina	Airbus A320|Airbus A321|Boeing 787|Airbus A320	multi	[Air France| LATAM]	2022-04-30 12:45:00	5/1/2022 10:15	1530	3	1284.0	USD	1248000.0	
3	ALG	Algeria	AEP	Argentina	Airbus A318|Airbus A320|Boeing 787|Airbus A320	multi	[Air France| LATAM]	2022-04-30 14:30:00	5/1/2022 10:15	1425	3	1290.0	USD	1347000.0	
4	ALG	Algeria	AEP	Argentina	Airbus A321neo|Boeing 777|Airbus A320	multi	[Lufthansa| LATAM]	2022-04-30 12:35:00	5/1/2022 10:15	1540	2	1347.0	USD	1381000.0	

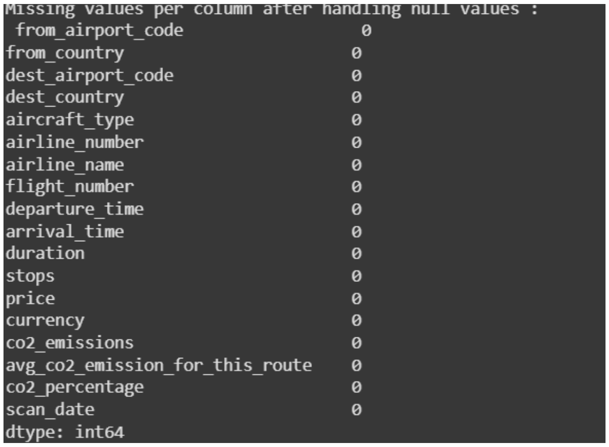
Handling Missing Values

We can replace numerical missing values with mean /mode/median , I choose mean .

For object datatype (categorical columns) I am filling in the most frequent value



```
Missing values per column before handling null values :
 from_airport_code                 1383
from_country                      1383
dest_airport_code                 1383
dest_country                      1383
aircraft_type                    15297
airline_number                    1383
airline_name                         0
flight_number                        0
departure_time                       0
arrival_time                         0
duration                             0
stops                                0
price                             1353
currency                             0
co2_emissions                     4874
avg_co2_emission_for_this_route  88402
co2_percentage                   88402
scan_date                            0
dtype: int64
```



```
Missing values per column after handling null values :
 from_airport_code                 0
from_country                      0
dest_airport_code                 0
dest_country                      0
aircraft_type                     0
airline_number                    0
airline_name                      0
flight_number                     0
departure_time                    0
arrival_time                      0
duration                          0
stops                             0
price                             0
currency                          0
co2_emissions                     0
avg_co2_emission_for_this_route   0
co2_percentage                    0
scan_date                         0
dtype: int64
```

Encoding Categorical Features

After analyzing data I have decided i will one hot encode my column airline number

I decided upon **one hot encoding** for this column because this column had less unique types and were the labels had no numerical/order wise relation between them

categorical_cols = ['airline_number']

Feature Scaling Techniques

These most useful numerical columns after EDA , i will scale them via **Standard Scaling**

In Standard Scaling as we studied in Stats course Z mean of data is 0 and it has a standard deviation of 1 means all the values in thousands etc are scaled bw [-1 and 1] with their mean being 0

This scaling introduces simplicity and computational ease to calculations

numerical_cols = ['price', 'duration', 'co2_emissions','stops']

New Features Created

I can see that in general flight costs , the fight costs escalate in seasons/days of week so what if i extract day of week from departure date and time given .The day of week feature could have relation to prices also the seasons (month) affect prices too.

Columns Dropped

Scan date is the date all these records were collected and hence ig same value for all rows and flight number is just a number , no real relation to anything

Note :After preprocessing When i did df.head() the columns were not one hot encoded and neither were numerical ones scaled.It is we did processing in pipeline and not on df.We setup the transformations and all , and they are relative to model to df

4-Model Selection and Training

Models Evaluated

I used pipelines of sklearn and used sklearn ML models to train my model for predicting flight prices:

- Linear Regression
- Neural Network
- Decision Tree

Model Comparisons

ML Model	**Root Mean Square Error**	**R2 Score**
Linear Regression	**1334.53725**	**0.54**
Neural Network	**1312.21204**	**0.56**
Decision Tree	**1399.1010**	**0.50**

RMSE and R2 Explanation :
Root Mean Squared Error , sum of y-y^ , and then take mean and then take root
RMSE score basically tells us how much error is between actual and predicted values , the smaller it is the better ,it is a mathematical measure and like tells how close to the best fit line you are .

R2 score is a measure of variance in your predicted and actual values, high variance == 1 means high correlation bw 2 values , while low means there is little to no relation between values
My R2 score indicates a moderately well performing linear regression model.

5-Hyperparameter Tuning

Parameters Tuned

For linear regression there really are not any hyper parameters that you could tune / change except for like learning rate , what we could do here in linear regression model fine tuning is that we fine tune the preprocessing steps , now like we did one hot encoding for categorical features and we also did standard scaling for numerical features and we know there are multiple ways to do scaling , weather centralizing values to a mean or not and hence many variations or even type of scaling like standard scaling or MinMax Scaling , hence we could train different models using these diff preprocessing variation and compare them to see the one that performs the best and keep that one for our use :)

For NN we trained different models with variating hyperparameters such as

- Size of hidden layer neurons
- Learning rate
- Number of iteration

For Decision Tree the variating hyper parameters are:

- Depth of Tree

- Leaf Nodes Count

  Decision Tree I have used here because it is a comparatively better and efficient ML algorithm , decision tree works by breaking down the classification/prediction form root node , so kind of like the model start from root node , it looks at the features like what was the duration of flight , what was the co2 emission of flight , how many stops were there in flight each of these decisions is a level of tree and the final decision is taken based off of decision at each step/level of tree .

Best Parameters Found

Parameters of a ML model are internal to a model and may change at every epochs like

-weights

-biases

-coefficients

Hyper parameters are the parameters external to a ML model and remain same for all model , like learning rate , k value in KNN , etc

For **Linear Regression:**

Results Of Hyper Parameter Tuning for linear regression indicate that when preprocessing numerical values via Standard Scaler if we scale values to a mean of 0 and a standard deviation of 1 would indeed improve performance more than scaling .

```
Best parameters: {'preprocessor__num__scaler__with_mean': False,
'preprocessor__num__scaler__with_std': True}
```

For **Neural Network**

```
Best Parameters: {'regressor__max_iter': 200,
'regressor__learning_rate_init': 0.01, 'regressor__hidden_layer_sizes':
(20,)}
```

For **Decision Tree**:

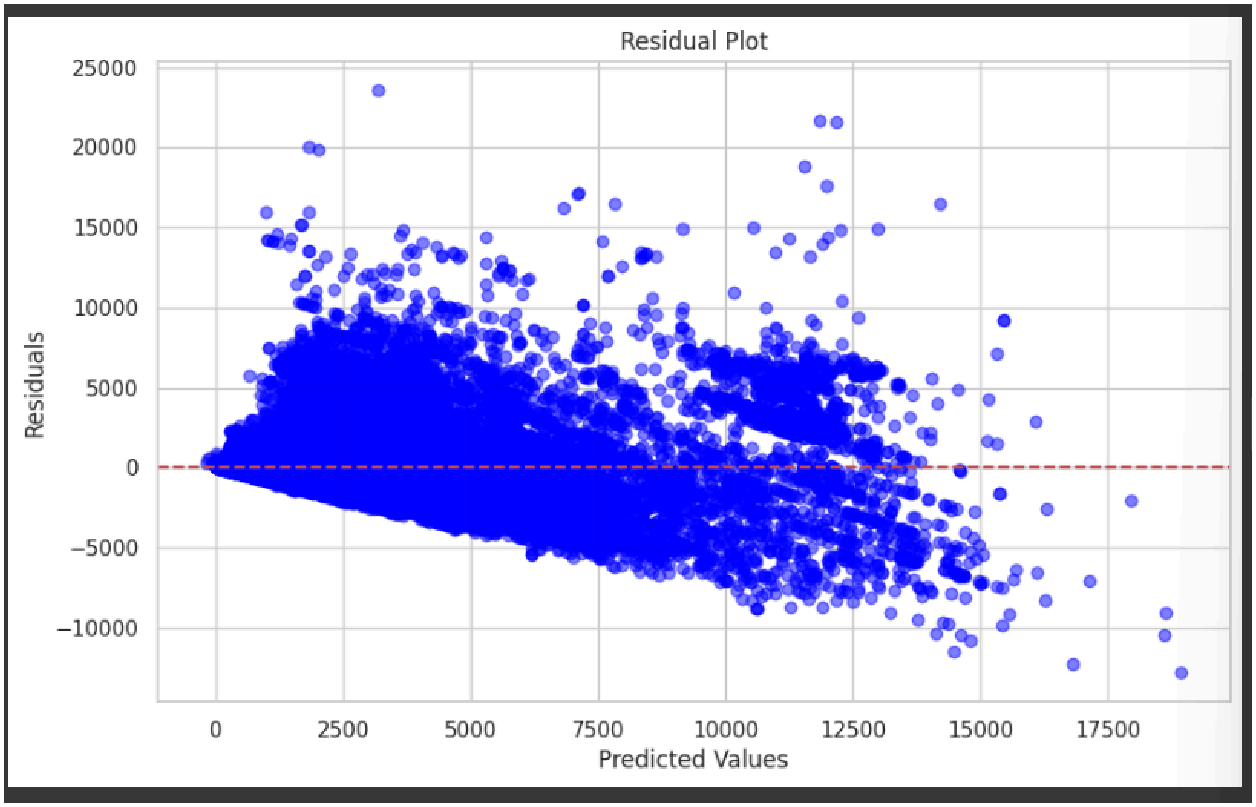
```
Best Parameters: {'regressor__min_samples_split': 10,
'regressor__min_samples_leaf': 5, 'regressor__max_depth': 10}
```

Performance Improvement

Best Tuned Model	Root Mean Square Error	R2 Score
Linear Regression	**1334.53700**	**0.547**
Neural Network	**1302.4010**	**0.569**
Decision Tree	**1245.769**	**0.60**

6-Final Model Evaluation

Visualizations (Residual Plots)



Analysis:

In a residual plot we keep on x- axis the predicted values and on y axis the diff bw predicated and actual values

My analysis of this Residual plot is that at higher values my predictions are too randomised hence ig not better but at small values the concentrated points indicate better evaluations.

There is non linearity in the predictions too.

7-Validation Techniques Comparison

Performance Analysis:

There is a slight improvement in the model results when k-fold cross validation across 5 folds was used .

```
Holdout RMSE: 1329.7048974663383
```

```
K-Fold RMSE: 1327.2101052828061
```

LOOCV never executed entirely because it trains on n-1 folds (training on n-1 samples and testing on 1 sample and then keep on looping) hence 1 million folds hence computationally insufficient on my laptop .

Bias-Variance TradeOff:

Holdout Validation is good computationally but K-fold validation does create a lot better balance between bias and variation and eventually leading to a lower bias and lower variance in data , the randomness increases in k fold hence model is better trained .Obviously LOOCV would yield most optimal results but I feel it could also overfit to the dataset .

Final Recommendation:

K-Fold because it is perfect balance between being computationally efficient and introducing randomness in dataset.

8-Stratified Sampling Analysis

Stratified Sampling makes sure that your target variable,s all classes like for eb both genders are equally represented in the dataset split . Since the prices column and my data did not have like such distinct classes I randomly split it into 10 classes so that any outlier / class is equally represented in data split.

Before stratifies sampling:`Holdout RMSE: 1329.7048974663383`

After Stratified Sampling :**`Holdout R2 Score: 0.5512618513254448`**

9-Handling Text and Categorical Attributes

Feature Encoding Techniques:

After analyzing data I have decided i will one hot encode my column airline number

I decided upon **one hot encoding** for this column because this column had less unique types and were the labels had no numerical/order wise relation between them

categorical_cols = ['airline_number']

Binary Format of One Hot Encoding:

The one hot encoding basically increases dimensionality which is one of its biggest cons because it creates a new column for each unique class and then gives a binary unique number to each row in the column .

Feature Scaling Techniques

These most useful numerical columns after EDA , i will scale them via **Standard Scaling**

In Standard Scaling as we studied in Stats course Z mean of data is 0 and it has a standard deviation of 1 means all the values in thousands etc are scaled bw [-1 and 1] with their mean being 0

This scaling introduces simplicity and computational ease to calculations

numerical_cols = ['price', 'duration', 'co2_emissions','stops']