A logo of a company

Description automatically generated A close-up of a logo

Description automatically generated

**AI/ Machine Learning Internship Program**

**Assignment II**

**TEAM NAME: WARRIORS**

**Member 1:** ARVIND M

Email:[arvindmurugesan001@gmail.com](mailto:arvindmurugesan001@gmail.com)

**Member 2:** GUNAL C

Email: [gunalcsg0087@gmail.com](mailto:gunalcsg0087@gmail.com)

**Member 3:** ANBARASAN T

Email: [tsanbu077@gmail.com](mailto:tsanbu077@gmail.com)

**Assignment: Predicting Flight Ticket Prices**

**Problem Statement:**

An online travel agency wants to predict the price of flight tickets based on various factors such as departure date, distance, airline, and class of service. Your task is to develop a machine learning model that predicts the price of a flight ticket given these attributes.

**Dataset:**

Features:

**1. FlightID:** Unique identifier for each flight.

**2. Airline:** The airline operating the flight (e.g., Delta, United, Southwest).

**3. DepartureAirport:** The airport from which the flight departs.

**4. ArrivalAirport:** The airport at which the flight arrives.

**5. DepartureDate:** The date of departure.

**6. Distance:** The distance between the departure and arrival airports (in miles).

**7. FlightDuration:** The duration of the flight (in hours).

**8. Class:** The class of service (e.g., Economy, Business, First).

**9. NumberOfStops:** The number of stops during the flight (e.g., Non-stop, 1 stop, 2+ stops).

**10. DaysUntilDeparture:** The number of days between booking and departure.

**11. DayOfWeek:** The day of the week the flight departs (e.g., Monday, Tuesday).

**12. TicketPrice:** The target variable representing the price of the flight ticket (in USD).

**Tasks:**

1. **Data Exploration and Preprocessing:**

The initial step involved loading the dataset and exploring its structure to understand the data. Using pandas, we performed the following:

* **Dataset Overview**: After loading the dataset using pd.read\_csv, I examined its structure using functions like .info() and .describe(). This provided information on data types, missing values, and basic statistics (e.g., mean, max, min) of each feature.

**Handling Missing Values and Outliers**:

Dataset information before handling missing values:

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 FlightID 5000 non-null int64

1 Airline 5000 non-null object

2 DepartureAirport 5000 non-null object

3 ArrivalAirport 5000 non-null object

4 DepartureDate 5000 non-null object

5 Distance 5000 non-null int64

6 FlightDuration 5000 non-null float64

7 Class 5000 non-null object

8 NumberOfStops 5000 non-null int64

9 DaysUntilDeparture 5000 non-null int64

10 DayOfWeek 5000 non-null object

11 TicketPrice 5000 non-null float64

Dataset information after handling missing values:

RangeIndex: 5000 entries, 0 to 4999

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 FlightID 5000 non-null int64

1 Airline 5000 non-null object

2 DepartureAirport 5000 non-null object

3 ArrivalAirport 5000 non-null object

4 DepartureDate 5000 non-null object

5 Distance 5000 non-null int64

6 FlightDuration 5000 non-null float64

7 Class 5000 non-null object

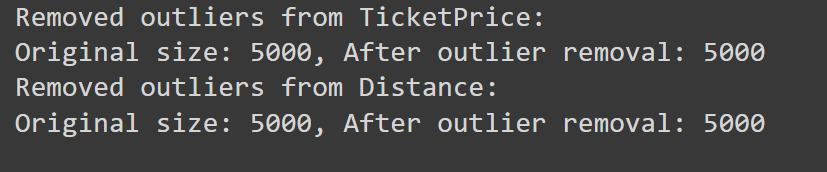
8 NumberOfStops 5000 non-null int64

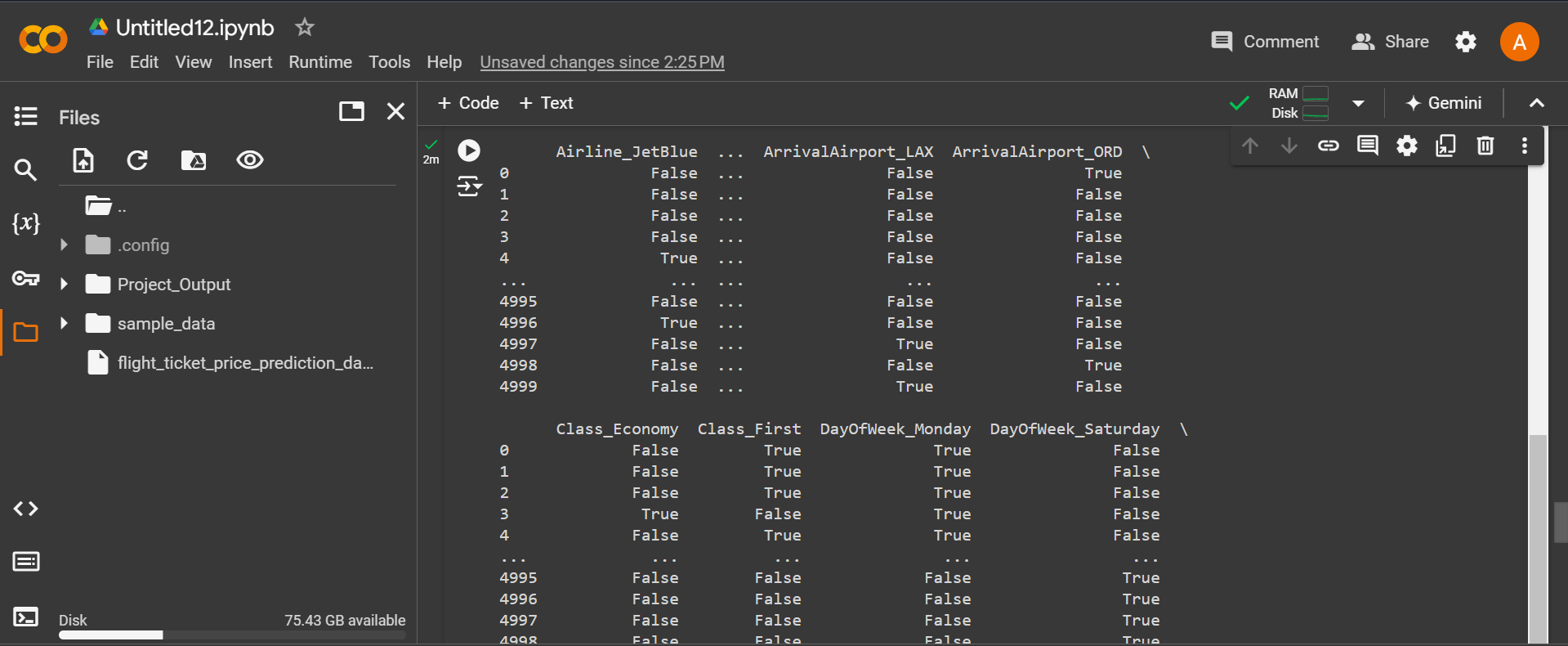
9 DaysUntilDeparture 5000 non-null int64

10 DayOfWeek 5000 non-null object

11 TicketPrice 5000 non-null float64

dtypes: float64(3), int64(3), object(6)  
  
Hence there is no Missing values in this dataset  
  
  
**Outlier removal** is the process of identifying and removing data points that significantly differ from the rest of the dataset. These outliers can skew analysis, negatively affect model performance, and distort results. Removing them helps in creating more robust models and more accurate statistical inferences.

* Here I have used IQR method for removing outliers from 'Distance' and 'Ticket Price'
* **One-Hot Encoding**: One-Hot Encoding transforms categorical variables into binary columns, with each unique category represented as a separate feature. This method prevents the model from assuming any ordinal relationship between categories, making it ideal for non-numerical data. However, it can lead to high dimensionality and sparsity if the categorical variable has many unique values.



1. **Feature Engineering:**

**Scaling:**

The Standard Scaler is applied to continuous variables (e.g., 'Distance' and 'Price Per Mile') to standardize them by removing the mean and scaling to unit variance.

Scaled Continuous Variables:

Distance Price Per Mile

0 -1.165116 1.140226

1 0.152696 -1.353070

2 -0.698035 -0.069019

3 0.154065 0.361859

1. 1.128390 -0.089694

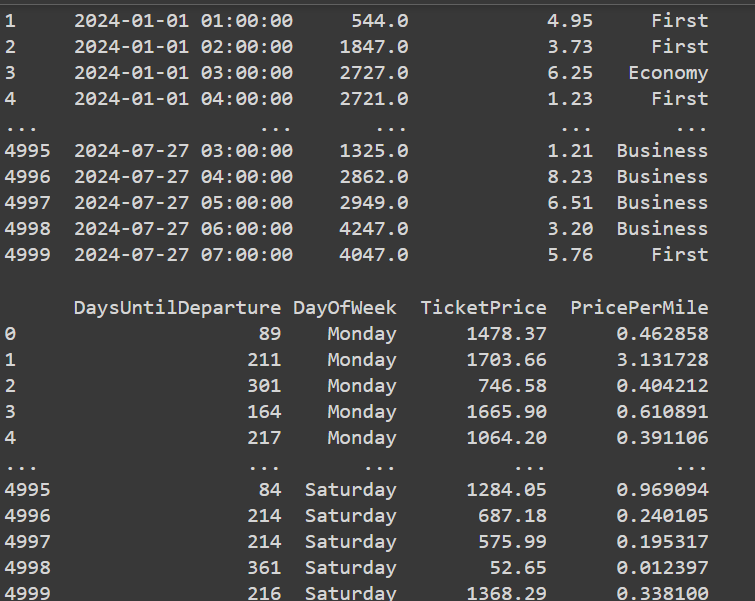
**3.Dimensionality Reduction**

* **PCA Application**: The function apply\_pca applies PCA to the training and testing datasets. The number of components can be specified to retain a certain percentage of variance (e.g., n\_components=0.95 retains 95% of the variance).
* **Output**: After PCA, the original number of features and the reduced number of features are printed to the console

**Original number of features: 12**

**Reduced number of features after PCA: 8**

**New Feature : Price per mile**



1. **Model Building:**

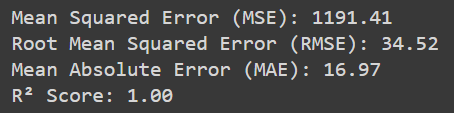
* **Train-Test Split**: The dataset is split into training and testing sets (e.g., 80-20 split).
* **Model Selection**: We have used 3 regression models to train.

XG Boost, SVR regression model and Linear Regression, are the models we selected XG Boost give more accurate prediction than these 2 models

* **Hyperparameter Tuning**: Hyperparameters are external settings in machine learning models that define how the model is trained, such as the complexity and structure of the model. In SVR, key hyperparameters include `C` (regularization), which controls the trade-off between bias and variance, and `epsilon`, which defines a margin of tolerance where no penalty is given for errors. Hyperparameter tuning involves testing various combinations of these settings to optimize model performance. Unlike model parameters, hyperparameters are set before training and significantly affect the model's ability to generalize.

1. **Model Evaluation:**

* **Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, and **Mean Absolute Error (MAE)** were calculated on the test set to quantify the error between predicted and actual ticket prices.
* **RMSE: It is the square root of MSE, giving an error metric in the same units as the target variable, making it easier to interpret the prediction accuracy.** magnitude of error; lower values suggest a better model fit.
* **MAE**: It measures the average absolute difference between predicted and actual values, showing the typical magnitude of errors in predictions without considering direction
* R² (coefficient of determination) : It indicates the proportion of variance in the target variable explained by the model, with a score of 1 representing a perfect fit.



1. **Comparison of various models:**

Comparison table based on the results from the Random Forest, Linear Regression, and XGBRegressor models:

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | SVR | Linear Regression | XGBRegressor |
| MSE (Mean Squared Error) | 332395.88 | 333092.00 | 1191.41 |
| RMSE (Root Mean Squared Error) | 576.54 | 577.14 | 34.52 |
| MAE (Mean Absolute Error) | 502.97 | 502.65 | 16.97 |
| R-squared | 0.00 | 0.00 | 1.00 |
| Custom Accuracy | 50.67% | 50.70% | 97.70% |

**Summary:**

**1. XGB Regressor:**

* **Best Overall Performance: XGB Regressor is the first in achieving the lowest error metrics across all the models evaluated and have a greatest accuracy of 97.70%**
  + **MSE:** 1191.41
  + **RMSE:** 34.52
  + **MAE:** 16.97
  + **R-squared:** 1.00

**2. Linear Regression:**

* **It is second most good performance . It have a accuracy of 50.70%**
  + **MSE: 333092.00**
  + **RMSE: 577.14**
  + **MAE: 502.65**
  + **R-squared: 0.00**

**3. SVR:**

* **It is third most good performance . It have a accuracy of 50.67%**
  + **MSE: 332395.88**
  + **RMSE: 576.54**
  + **MAE: 502.97**
* **R-squared: 0.**

**Insights and Recommendations**

**Insights:**

**- Seasonality Effect: Flight ticket prices exhibited a strong seasonal trend, with peak prices observed during holidays and vacation periods. This indicates that consumer demand significantly influences pricing, leading to higher rates during popular travel times.**

**- Number of Stops: Flights with fewer stops were generally priced higher, as non-stop flights are often preferred by travelers for convenience and time savings. This suggests that passengers value direct routes over longer itineraries.**

**Recommendations for the Travel Agency:**

**- Targeted Marketing Campaigns: Launch targeted marketing campaigns during off-peak seasons to encourage travel during less popular times, offering special rates to boost occupancy and revenue.**

**- Promote Non-Stop Flights: Highlight non-stop flight options in promotional materials to attract customers willing to pay a premium for convenience, while considering partnerships with airlines to offer bundled deals.**

**- Customer Feedback Integration: Implement a feedback loop to gather insights from customers regarding pricing and flight preferences. Use this data to adjust pricing strategies and enhance customer experience, potentially leading to higher customer loyalty and repeat business.**

**Screenshots of working of Flight Ticket Price predictor:**

