LOGBOOK – PHASE3

GROUP#3

In this phase, our goal is to develop a predictive model to estimate prices based on specific flight characteristics. The dataset includes various attributes, such as airline, flight duration, departure and arrival cities, and the number of stops. By analyzing these features, we can gain insights into how they impact ticket prices and create a model capable of making accurate predictions.

Regression Task:

Target Variable Analysis:

First, we visualized the distribution of our target variable, **price**, and observed that it was right-skewed. Although there were some high-price outliers, we chose not to remove them because they represent real-world data that we need for accurate modeling and insights. **Figure 1**

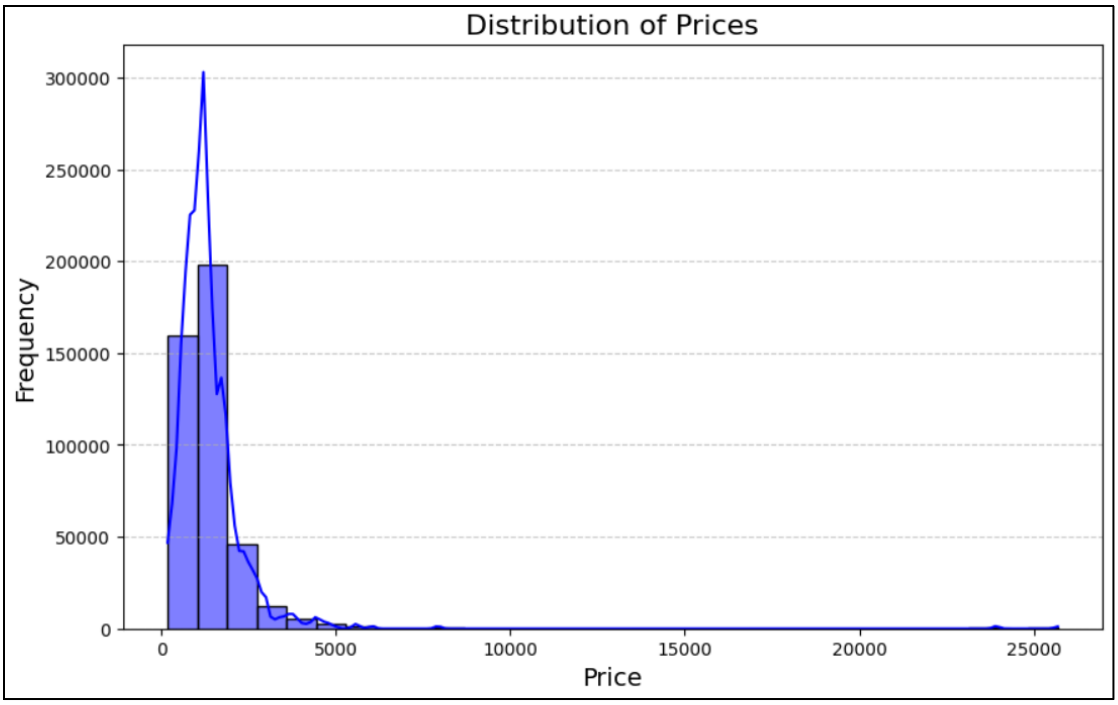


Figure 1

Data Preprocessing:

Next, we preprocessed the data as follows:

1. **Encoding Categorical Variables**: We used LabelEncoder to transform categorical columns into numerical values.
2. **Feature Scaling**: We standardized all features using MinMaxScaler to ensure they were on the same scale.
3. **Splitting Data**: The dataset was split into training and testing sets, with 70% used for training and 30% for testing.

Baseline Model and Model Comparison:

We began by training and evaluating a baseline model using **KNeighbors Regressor**. Then, we trained nine additional models to compare their performance:

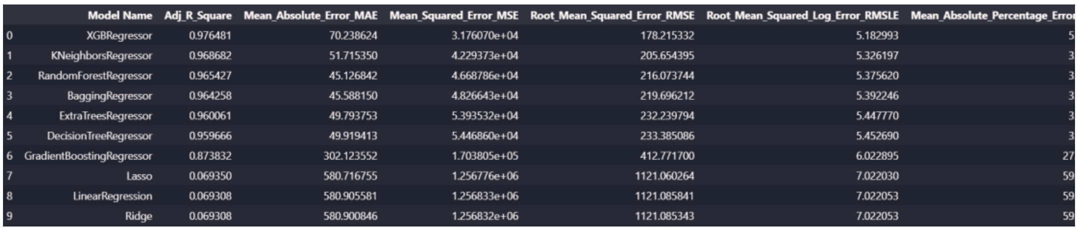
- Linear Regression
- Decision Tree Regressor
- Random Forest Regressor
- Extra Trees Regressor
- Gradient Boosting Regressor
- XGBoost Regressor (with default parameters)
- Bagging Regressor
- Ridge Regression
- Lasso Regression (with alpha=0.1)

The performance of these models was assessed using the following metrics:

- **Mean Absolute Error (MAE)**: To measure the average magnitude of errors in predictions.
- **Root Mean Squared Error (RMSE)**: To emphasize larger errors and evaluate overall model performance.
- **R-squared (R^2)**: To measure the proportion of variance in the target variable explained by the model.
- **Mean Absolute Percentage Error (MAPE):** Measures the accuracy of predictions as a percentage.
- **Mean Squared Error (MSE)**: Calculates the average squared difference between the predicted and actual values.
- **Root Mean Squared Log Error (RMSLE**): A metric that evaluates the ratio between predicted and actual values on a logarithmic scale.
- **Adjacent R-squared (R^2)**: A modified version of R-squared that adjusts for the number of predictors in the model. It provides a more accurate measure of model performance when adding more variables, preventing overfitting.

Results and Best Model:

After evaluating all models, **XGBoost Regressor** emerged as the best-performing model, achieving the highest R² score and the lowest RMSE values. This made it the most accurate and robust model for predicting ticket prices. **Figure 2**



	Model Name	Adj_R_Square	Mean_Absolute_Error_MAE	Mean_Squared_Error_MSE	Root_Mean_Squared_Error_RMSE	Root_Mean_Squared_Log_Error_RMSLE	Mean_Absolute_Percentage_Erro
0	XGBRegressor	0.976481	70.238624	3.176070e+04	178.215332	5.182993	5
1	KNeighborsRegressor	0.968682	51.715350	4.229373e+04	205.654395	5.326197	3
2	RandomForestRegressor	0.965427	45.126842	4.668786e+04	216.073744	5.375620	3
3	BaggingRegressor	0.964258	45.588150	4.826643e+04	219.696212	5.392246	3
4	ExtraTreesRegressor	0.960061	49.793753	5.393532e+04	232.239794	5.447770	3
5	DecisionTreeRegressor	0.959666	49.919413	5.446860e+04	233.385086	5.452690	3
6	GradientBoostingRegressor	0.873832	302.123552	1.703805e+05	412.771700	6.022895	27
7	Lasso	0.069350	580.716755	1.256776e+06	1121.060264	7.022030	59
8	LinearRegression	0.069308	580.905581	1.256833e+06	1121.085841	7.022053	59
9	Ridge	0.069308	580.900846	1.256832e+06	1121.085343	7.022053	59

Figure 2

Visualization:

To better understand the performance of our best model, we plotted the following graphs:

1. **Actual vs. Predicted Prices**: A scatter plot to visualize how closely the model's predictions align with the actual prices. **Figure 3**
2. **Residuals vs. Predicted Prices**: A plot to evaluate the distribution of errors and check for any patterns in the residuals. **Figure 4**

These analyses confirmed the effectiveness of the XGBoost Regressor and validated its selection as the final model for price prediction.

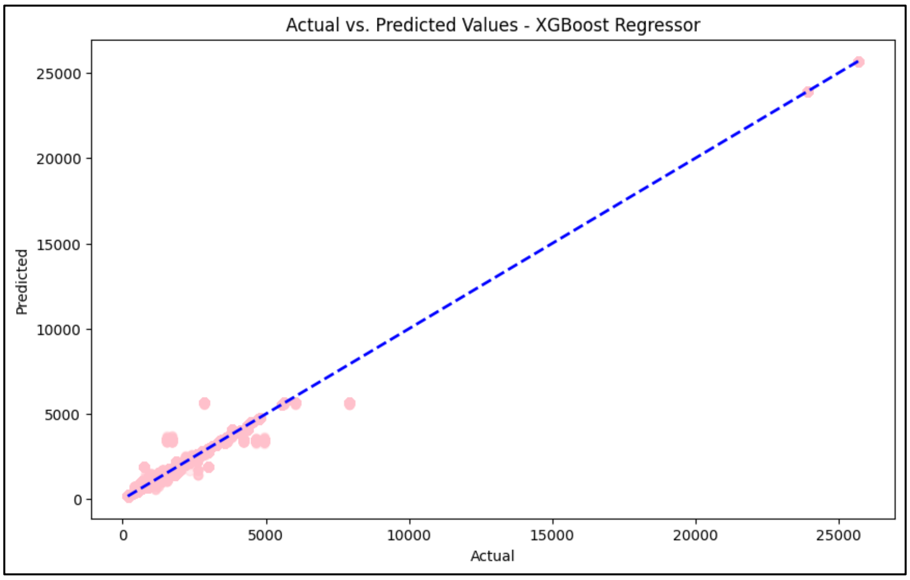


Figure 3

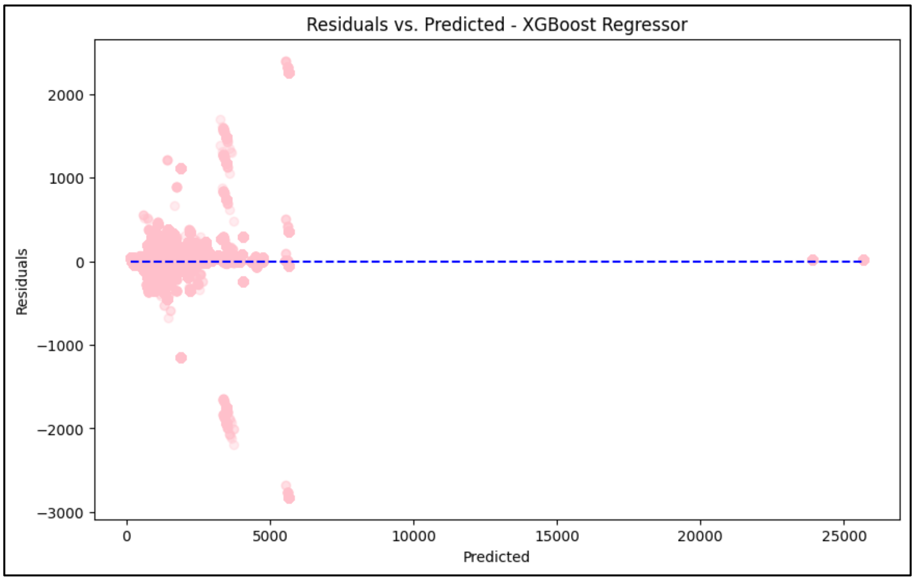


Figure 4

Clustering Task:

To analyze patterns and group flights with similar characteristics, we conducted a clustering task. This was performed separately from the regression task to avoid overlapping operations.

Dataset Preparation:

We started by copying the dataset into a new DataFrame variable to preserve the integrity of the original data. This allowed us to experiment with clustering-specific preprocessing without affecting the data used for regression.

Removing Unnecessary Columns:

Columns such as **price** (the target variable for regression) and **flight lands next day** (a Boolean) were removed. These features were deemed unnecessary for clustering since their data types (boolean and integer) do not align well with the goals of unsupervised learning, where clustering focuses on finding patterns in attributes that define similarities among data points.

Data Preprocessing:

1. **Standardization:** We used StandardScaler to standardize the numerical features, ensuring all attributes contributed equally to the clustering process.
2. **One-Hot Encoding:** Categorical columns were encoded using one-hot encoding to represent them as numerical data. This approach allowed us to retain the granularity of categorical variables during clustering.
3. **Dimensionality Reduction with PCA:** We applied Principal Component Analysis (PCA) to reduce the dimensionality of the dataset. This step helped minimize computation time while retaining the most important information, making clustering more efficient and less prone to overfitting.
4. **Sampling the Data:** Given the size of the dataset (400,000+ rows), we took a random sample to further optimize computational efficiency. This ensured that clustering algorithms could be run in a reasonable amount of time without compromising the results' reliability.

Clustering Methods:

We evaluated multiple clustering techniques to determine the best approach for grouping flights:

K-Means Clustering

- We performed K-Means clustering with different values of **k** (3, 5, 7, and 9).
- **Elbow Method:** The elbow point suggested the optimal number of clusters at **k = 5**, where the rate of decrease in within-cluster sum of squares leveled off. **Figure 5**
- **Silhouette Score:** Silhouette analysis indicated that **k = 9** produced the best cluster cohesion and separation. **Figure 6**

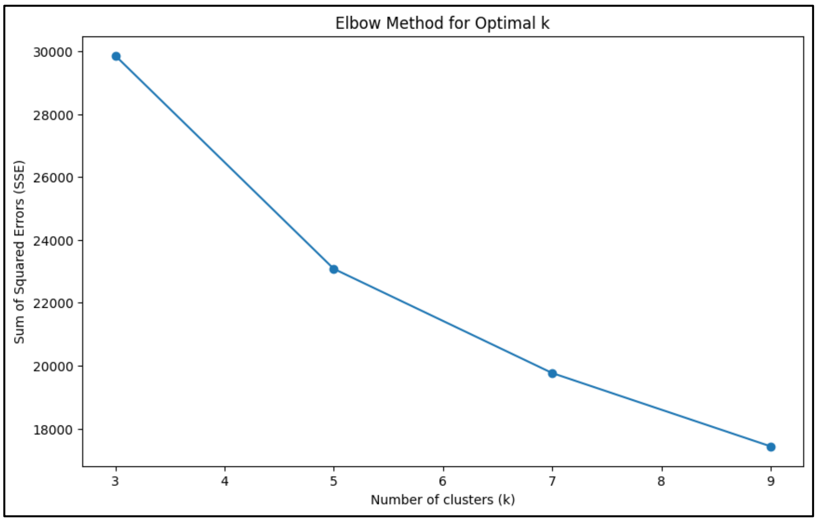


Figure 5

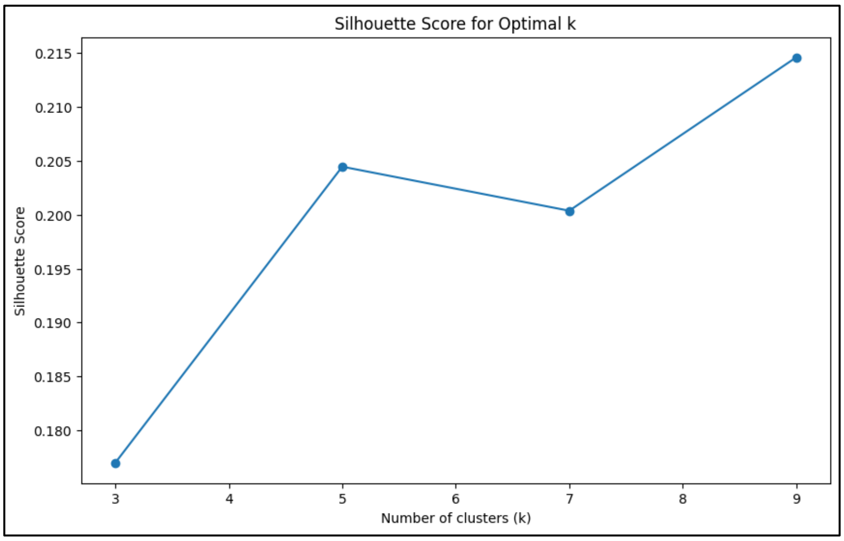


Figure 6

Gaussian Mixture Models (GMM):

We applied Gaussian Mixture Models to identify probabilistic cluster memberships. The **silhouette score** for GMM was **0.173**, indicating that while some degree of cluster cohesion and separation was achieved, the overall performance was relatively low. This suggests that GMM was able to capture some patterns in the data but struggled to define well-separated clusters compared to K-Means.

DBSCAN:

This model was explored for its ability to detect clusters of arbitrary shapes. However, the algorithm struggled with the dataset due to its increased dimensionality after preprocessing steps like **one-hot encoding** and despite applying **PCA for dimensionality reduction**. This challenge was reflected in the **silhouette score** of **-0.357**, indicating that DBSCAN failed to form meaningful clusters in this case.

Hierarchical Clustering:

The results of the hierarchical clustering method provided moderate insights into the structure of the data. While the technique successfully grouped similar data points into clusters, the results suggest that this approach may not be the most optimal for this dataset. The large and complex nature of the dataset, even after dimensionality reduction, posed challenges for hierarchical clustering, as evident in the uneven distribution and merging of clusters at higher distances. Although the method revealed some meaningful relationships, the results were less interpretable and cohesive compared to those obtained from K-Means, which performed better in terms of cluster quality and separation. Overall, hierarchical clustering provided some value in understanding the dataset's structure but was less effective than other clustering approaches like K-Means for this specific task. **Figure 7**

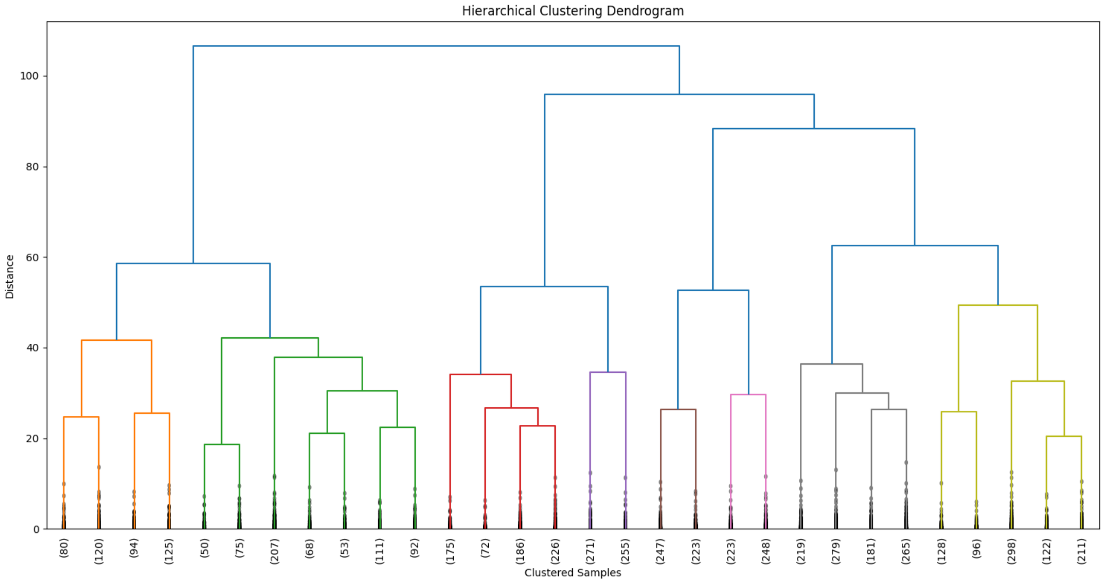


Figure 7

Results:

Among the clustering methods, K-Means performed the best:

- **k = 5** (optimal clusters according to the elbow method) provided well-separated groups suitable for further analysis.
- **k = 9** (optimal clusters according to the silhouette score) provided higher cohesion but at the cost of increased complexity.

Overall, K-Means clustering emerged as the most suitable technique for our dataset, effectively uncovering meaningful patterns in the flight data.