Flight Price Prediction

Group 23
Ravi Khimjibhai Patel
Shivam Ashokbhai Lalakiya

Overview and Problem Statement

Our project focuses on building a supervised machine-learning model for accurately predicting flight prices. Predicting flight prices is a complex task involving monitoring various indicators such as departure time, airline, flight duration, and seasonality. To address this challenge, we propose to use several machine learning algorithms, including Linear Regression, Polynomial Regression, Decision Tree, Regression Tree, and Gradient Boosting, to develop the model. By training and generalizing the model, we aim to achieve high accuracy in predicting flight prices. Additionally, we will conduct exploratory data analysis to identify relevant patterns and trends in the flight data. Feature importance analysis will be used to determine which variables have the most significant impact on flight prices. We will use metrics such as Accuracy, Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared to evaluate the model's performance. Our ultimate goal is to provide a reliable and accurate model to help airlines and travel companies make more informed pricing decisions and assist consumers in making informed decisions about air travel.

I. Background and Introduction

1.1 Background:

The airline industry has been growing steadily over the past few decades, and air travel has become an essential mode of transportation for many people worldwide. With the increased demand, airlines constantly change their pricing strategies to optimize their revenue. However, the variability of flight prices can make it challenging for travelers to plan their trips and stay within their budgets.

1.2 The Problem:

The variability of flight prices is a significant challenge for travelers, and predicting flight prices has become a crucial aspect of the travel industry. Traditional methods of predicting flight prices relied on historical data and expert opinions, which may need to be more accurate and up-to-date. With the availability of large datasets and advancements in machine learning algorithms, it is now possible to predict flight prices more accurately and efficiently.

1.3 The Solution:

This project aims to predict flight prices using machine learning algorithms, including Linear Regression, Decision Tree, Regression Tree, KNN, and Neural Network. We will use a dataset containing ten features that capture various aspects of a flight, including the airline, departure, and arrival cities, number of stops, duration, seat class, and booking and trip dates. Our target variable is the ticket price, which we will predict using the other features.

To predict flight prices, we will use machine learning algorithms, which are algorithms that learn patterns and relationships from data. Specifically, we will use supervised learning algorithms, which learn from labeled data, i.e., data where we know the input features and the corresponding output (in this case, the ticket price). We will split our dataset into training and testing sets, using the training set to train our models and the testing set to evaluate their performance.

Section 2 will describe the dataset and perform exploratory data analysis. In section 3, we will prepare the data for model training by handling missing values, encoding categorical features, and feature scaling. In section 4, we will train four machine learning models and evaluate their performance using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared. Finally, in section 5, we will summarize our findings and provide recommendations for future work, such as exploring other machine learning algorithms or adding more features to the dataset.

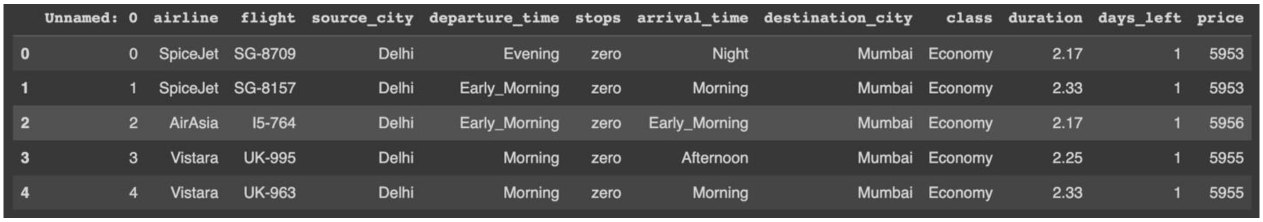
II. Data Exploration and Visualization

1.1 Data Description:

The dataset we will use in this project contains information on flight prices and their corresponding features. It includes ten features and 300K instances. Here is a description of each part:

i. **Airline**: The name of the airline company is stored in the airline column. It is a categorical feature having six different airlines.
ii. **Flight**: The flight stores information regarding the plane's flight code. It is a categorical feature.
iii. Source City: The city from which the flight takes off. It is a definite feature having six unique towns.
iv. **Departure** Time: This derived categorical feature is created by grouping periods into bins. It stores
v. information about the departure time and has six unique time labels.
vi. **Stop** A categorical feature with three distinct values that store the number of stops between the source and destination cities.
vii. **Arrival Time**: This derived categorical feature is created by grouping time intervals into bins. It has six different time labels and keeps the information about the arrival time.
viii. **Destination City**: City where the flight will land. It is a categorical feature having six unique cities.
ix. **Class**: A categorical feature that contains information on seat class; it has two distinct values: Business and Economy.
x. **Duration**: A continuous feature that displays the time it takes to travel between cities in hours.
xi. **Days Left**: This is a derived characteristic calculated by subtracting the trip date from the booking date.
xii. **Price**: Target variable stores information on the ticket price.

Here's the snapshot of the dataset's overview:

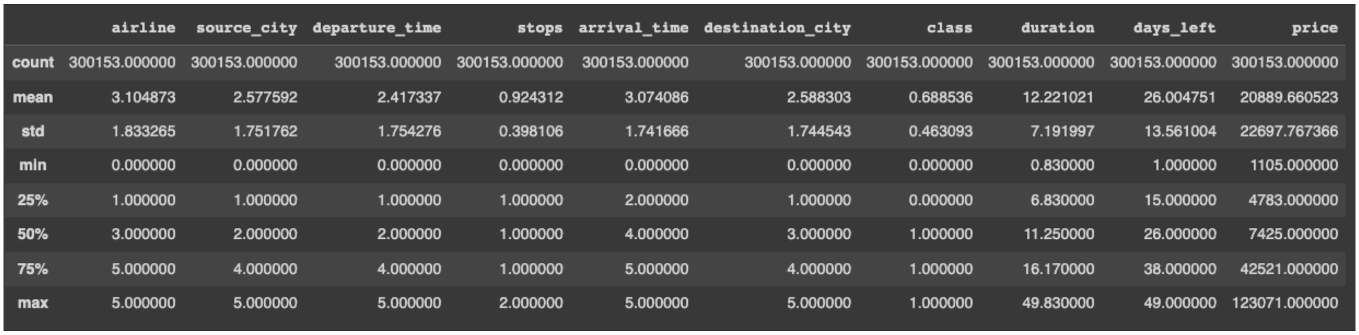


	Unnamed: 0	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
0	0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	5953
1	1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953
2	2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956
3	3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955
4	4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955

Figure 1 - Overview of Flight Price Data Frame

1.2 Data Visualization:

We analyzed the distributions of some attributes to get an idea about the data distribution.



	airline	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
count	300153.000000	300153.000000	300153.000000	300153.000000	300153.000000	300153.000000	300153.000000	300153.000000	300153.000000	300153.000000
mean	3.104873	2.577592	2.417337	0.924312	3.074086	2.588303	0.688536	12.221021	26.004751	20889.660523
std	1.833265	1.751762	1.754276	0.398106	1.741666	1.744543	0.463093	7.191997	13.561004	22697.767366
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.830000	1.000000	1105.000000
25%	1.000000	1.000000	1.000000	1.000000	2.000000	1.000000	0.000000	6.830000	15.000000	4783.000000
50%	3.000000	2.000000	2.000000	1.000000	4.000000	3.000000	1.000000	11.250000	26.000000	7425.000000
75%	5.000000	4.000000	4.000000	1.000000	5.000000	4.000000	1.000000	16.170000	38.000000	42521.000000
max	5.000000	5.000000	5.000000	2.000000	5.000000	5.000000	1.000000	49.830000	49.000000	123071.000000

Figure 2 - Data Summary

The next step is visualizing the data using charts and graphs such as scatter plots, histograms, and box plots. Visualization techniques help identify patterns and relationships that are not apparent through descriptive statistics alone. For example, a scatter plot can visualize the relationship between flight price and the number of days to departure. In contrast, a histogram can be used to identify the distribution of flight prices.

In addition to these basic visualization techniques, more advanced techniques such as heat maps and tree maps can be used to identify trends and patterns in the data. Heat maps can be used to visualize the distribution of flight prices by region, while treemaps can be used to determine the most popular airlines or destinations. The insights gained through data exploration and visualization can inform the development of a predictive model.

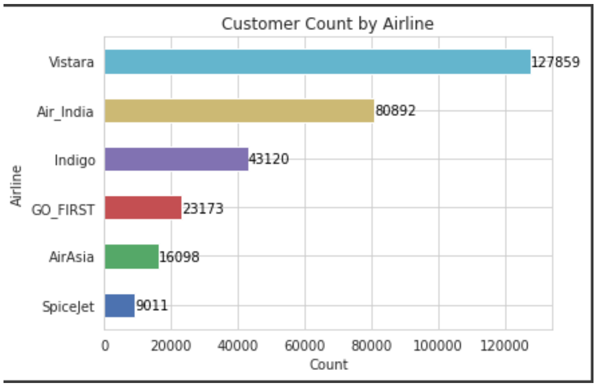


Figure 3 - Customer count for different Airlines

This bar graph (Figure 3) displays the customer count for each airline, with the horizontal or x-axis representing the number of customers and the vertical or y-axis representing the airlines. The graph shows that Vistara has the highest customer count, followed by Air India and Indigo.

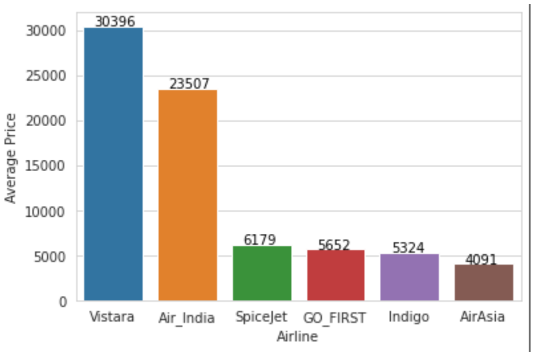


Figure 4 - Average flight ticket price for various airlines

This bar graph (Figure 4) displays the Average Flight Prices for each airline, with the horizontal or x-axis representing the airlines and the vertical or y-axis representing the average flight prices. The graph shows that Vistara has the highest average flight prices, followed by Air India.

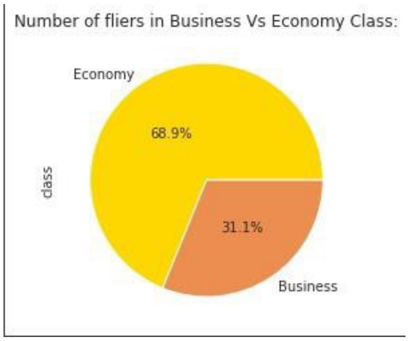


Figure 5 - Percentage of Business to Economy class passengers

This pie chart (Figure 5) displays all airlines' business and economy class percentages. There are 68.9% Economy class passengers (206,666), compared to 31.1% Business Class passengers (93,487).

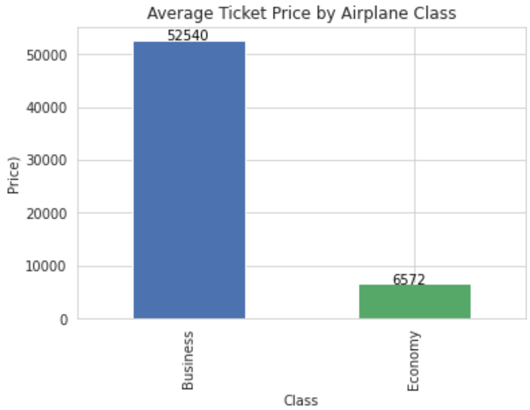


Figure 6 - Average ticket price for Business and Economy class

This bar chart (Figure 6) displays the average ticket price for all airlines' business and economy class passengers. The ticket price for Business class passengers is 52450 INR, while the price for Economy Class passengers is 6572 INR.

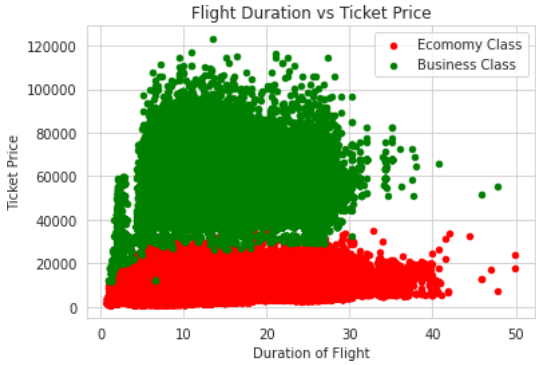


Figure 7 - Scatter plot for the duration of flight vs. ticket prices

The Scatter plot (Figure 7) compares the flight duration with ticket prices for the economy and business class. The key takeaway is that the prices depend more on style than flight duration, as we can observe two cluster types.

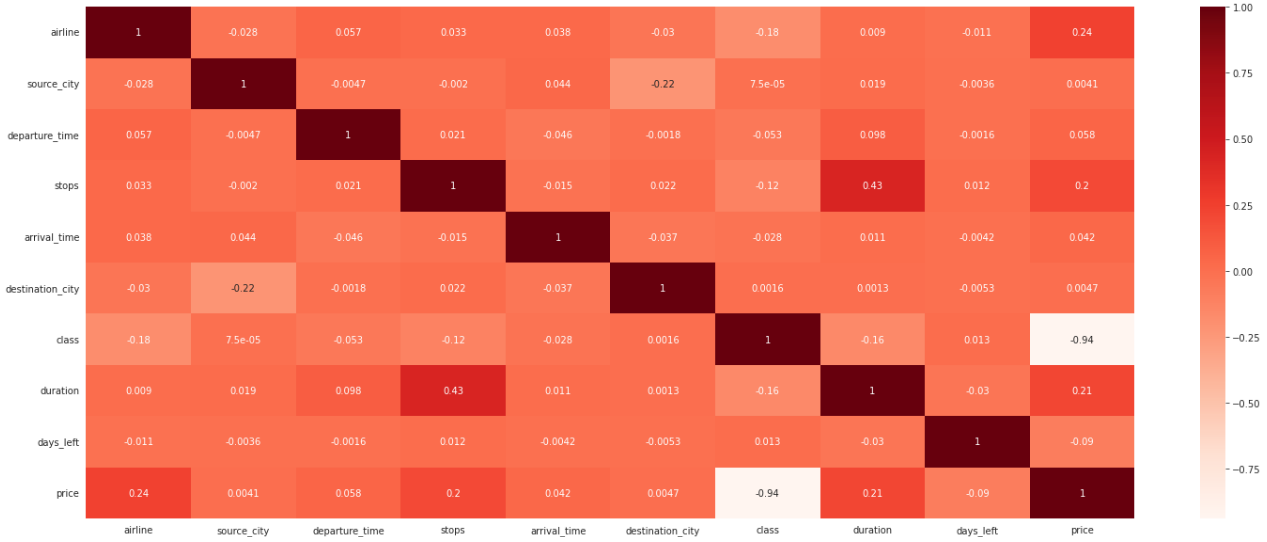


Figure 8 - Heatmap for the correlation between columns(features)

In Figure 8, we can observe that the Duration of a flight, airlines, and stops have a high positive correlation with prices, as prices will increase depending on these features. On the other hand, prices negatively correlate with class and days left before departure. Here, we labeled Business Class 0 and Economy Class 1. Therefore, we see a high negative correlation, as shown in Figure 6. Furthermore, the price will increase as fewer days are left during the departure.

In conclusion, a flight price prediction report's data exploration and visualization section is crucial in identifying patterns and relationships in the collected data. Descriptive statistics and visualization techniques help identify outliers and anomalies, visualize distributions, and identify trends and patterns. These insights can then be used to inform the development of a predictive model.

III. Data Preparation and Preprocessing

3.1 Data-Processing:

Before training our machine learning models, we must prepare and preprocess the dataset. In this section, we will perform the following steps:

1. **Check for missing values:** The first step in data preparation is to check for missing values. Missing values can cause errors in our machine-learning models or lead to incorrect predictions. We will use Panda's library to check for missing values in our dataset. If any missing values are found, we will either impute them or remove them.

2. **Remove unnecessary features:** We will remove unnecessary features irrelevant to our analysis. For example, the flight code feature may contribute little to predicting the flight price so we may remove it from our dataset.

3. **Encode categorical features:** Machine learning models require numerical data, so we need to encode our flat features into numerical values. We will use one-hot encoding to convert our categorical features into binary parts for our machine-learning models. This will allow our models to learn the relationships between different categories and predict flight prices accurately.

4. **Normalize continuous features:** Continuous features may have different scales, which can cause bias in our machine-learning models. To avoid this, we will normalize our constant features using the min-max scaling technique. This technique scales our continuous features to a range between 0 and 1, making them comparable and reducing bias in our models.

5. **Outlier Detection and Removal:** Outliers are data points that deviate significantly from the rest of the dataset. Outliers can cause bias in our machine-learning models, so we must identify and remove them. We will use box plots and the Z-score method to identify and remove outliers from our dataset. Box plots help us visualize the distribution of our data and identify any outliers. At the same time, the Z-score method calculates the number of standard deviations a data point is from the mean. If a data point's Z-score exceeds a certain threshold, we can consider it an outlier and remove it from our dataset.

IV. Data Mining Techniques and Evaluation

This section will apply several machine learning algorithms to our preprocessed dataset and evaluate their performance.

4.1 Machine Learning Models

We will apply the following algorithms:

1. **Linear Regression:**

   Linear Regression is a commonly used algorithm for regression problems whose goal is to predict a continuous output variable based on one or more input variables. It works by finding the best-fit line that minimizes the sum of the squared differences between the predicted and actual values. The algorithm assumes a linear relationship between the input and output variables, represented by a simple equation such as Y = aX + b. The coefficients a and b are estimated from the training data using ordinary least squares (OLS) or gradient descent methods. Linear Regression is a good starting point for simple regression problems and provides easy-to-interpret results, but it may not be suitable for complex data with non-linear relationships.

2. **Polynomial Regression:**

   Polynomial Regression extends Linear Regression, allowing for non-linear relationships between input and output variables. It works by adding polynomial terms to the linear equation to create a curve that better fits the data. For example, a second-order polynomial equation can be represented as Y = aX^2 + bX + c. The coefficients a, b, and c are estimated from the training data using OLS or gradient descent methods. Polynomial Regression is a good choice when more than a linear model is needed to capture the complexity of the data. Still, it may be prone to overfitting if the degree of the polynomial is too high.

3. **Decision Tree:**

   Decision Trees are an algorithm that can be used for classification and regression problems. They work by recursively splitting the data based on the input variables to create a tree-like structure. Each internal node of the tree represents a decision based on a particular input variable, and each leaf node represents a prediction for the output variable. The splitting criteria can be chosen based on various impurity measures, such as entropy or Gini impurity, to maximize the information gained at each step. Decision Trees are easy to interpret and can capture non-linear relationships, but they are prone to overfitting and unstable.

4. **Regression Tree:**

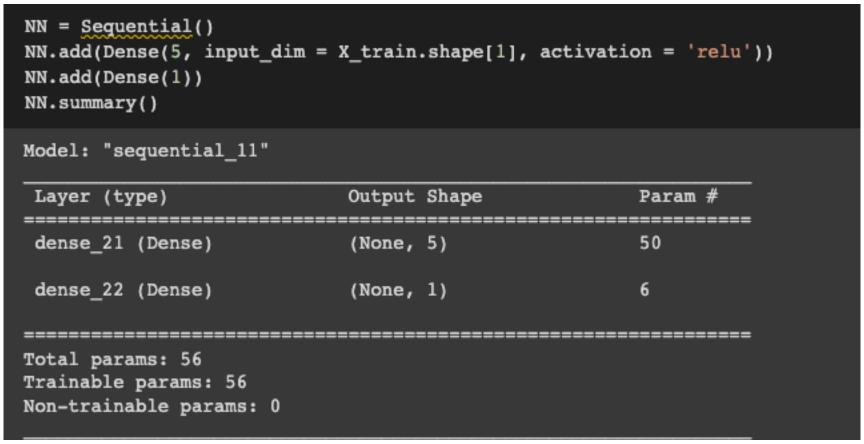
   Regression Trees are a specific type of Decision Tree used for regression problems. They work by recursively partitioning the data into smaller subsets based on the input variables and fitting a simple model (such as a mean or median) to the output variable for each subgroup. The splitting criteria can be chosen based on measures such as the variance reduction or the mean squared error. Regression Trees can handle non-linear relationships and are less prone to overfitting than Decision Trees, but they may not capture complex interactions between input variables. Ensemble methods such as Random Forest or Gradient Boosting can combine multiple Regression Trees with improving performance.

5. **KNN (K-Nearest Neighbors):**

   KNN (K-Nearest Neighbors) is a supervised machine learning algorithm that falls under instance-based learning. It is widely used for classification and regression tasks in various domains, such as image recognition, natural language processing, and bioinformatics. KNN operates by finding the K nearest data points from the training set to the input data point based on a distance metric, such as Euclidean distance or cosine similarity.

6. **Neural Networks (NN)**:

   Neural networks are a type of machine learning algorithm that is inspired by the structure and function of the human brain. They comprise multiple layers of interconnected nodes (neurons), which process and transform data as it passes through the network. Neural networks can be used for various machine-learning tasks, including classification, regression, and pattern recognition. The architecture used in this project is-



```
NN = Sequential()
NN.add(Dense(5, input_dim = X_train.shape[1], activation = 'relu'))
NN.add(Dense(1))
NN.summary()
```

```
Model: "sequential_11"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_21 (Dense)            (None, 5)                 50

 dense_22 (Dense)            (None, 1)                 6

=================================================================
Total params: 56
Trainable params: 56
Non-trainable params: 0
_________________________________________________________________
```

Figure 9 - NN architecture

4.2 Model Evaluation and Selection

We will split our preprocessed dataset into training, validation, and testing sets with a ratio of 60:25:15, respectively. We will train each of the above models on the training set and evaluate their performance on the validation set using various metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-Squared Score (R2). The algorithm with the lowest MSE and RMSE and the highest R2 score will be considered the best-performing algorithm for our dataset. And the best-performing algorithms will be used for the test set, and predictions and the final evaluation will be performed on it.
We will be evaluating these models on these parameters:

1. **Accuracy**:

   Accuracy is a metric used for classification problems and measures the proportion of correctly classified samples out of the total number of pieces. However, in regression problems, accuracy is not a suitable metric since it does not capture the magnitude of the errors.

2. **R-squared**:

   R-squared (or the coefficient of determination) is a metric used for regression problems that measure the proportion of the variance in the output variable explained by the model. It ranges from 0 to 1, with 1 indicating a perfect fit and 0 indicating that the model does not explain variance. R-squared is a valuable metric for comparing different models, but it can be biased towards complex models and may need to be revised for small datasets.

3. Absolute Error (MAE):

Mean Absolute Error (MAE) is a metric used to evaluate the performance of a regression model. It measures the average absolute difference between the predicted and actual values in the dataset. Specifically, it calculates the average fundamental differences between the predicted and actual values for each data point. One of the advantages of using MAE over other metrics, such as Mean Squared Error (MSE), is that it is less sensitive to outliers. Since absolute differences are used instead of squared differences, significant errors have less influence on the overall value of the metric.

4. Mean Squared Error (MSE):

MSE is a metric used for regression problems that measure the average squared difference between the predicted and actual values. It is calculated as the sum of the squared errors divided by the number of samples. MSE measures the average magnitude of the mistakes and helps evaluate the model's performance on the testing set. However, it can be sensitive to outliers and may need to provide a complete picture of the model's performance. Root Mean Squared Error (RMSE) is often used instead of MSE to report errors in the same units as the output variable]Instead, metrics such as R-squared, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) are commonly used to evaluate the performance of regression models. These metrics measure the average magnitude of the errors, and the proportion of the variance in the output variable that Stops generates.

5. **Gain chart**:

The gain chart is a widely used evaluation tool in regression analysis that helps evaluate a predictive model's performance. It is a graphical representation of the model's predictive power regarding how much it improves over a random baseline model. The gain chart plots the cumulative gains of the model against the number of observations in the test set, where the incremental increases measure the percentage of the target variable captured by the model at each point. In other words, it compares the ratio of the target variable correctly predicted by the model against the portion of the target variable that would have been captured by random guessing.

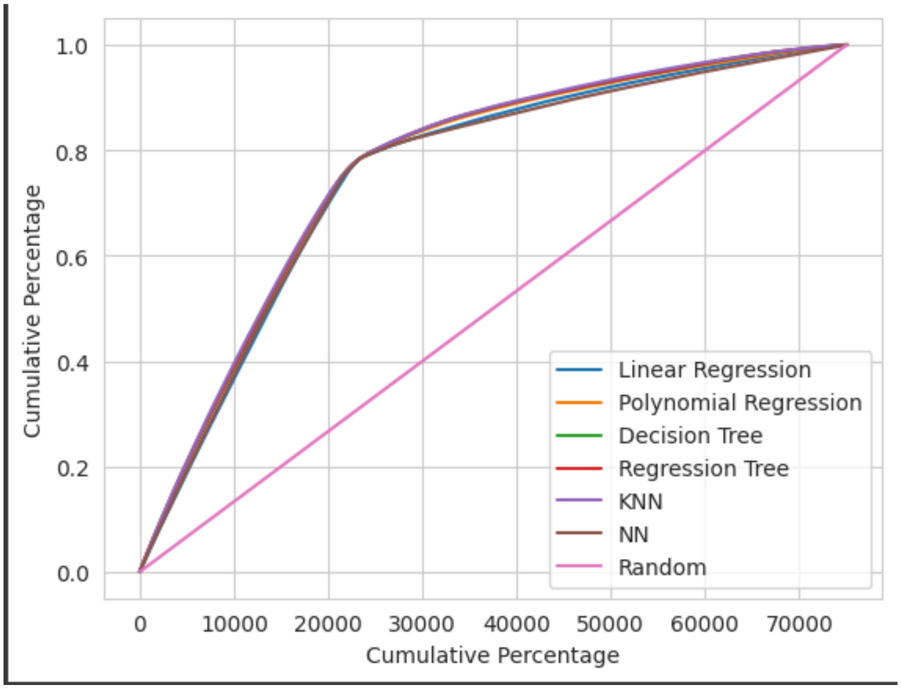
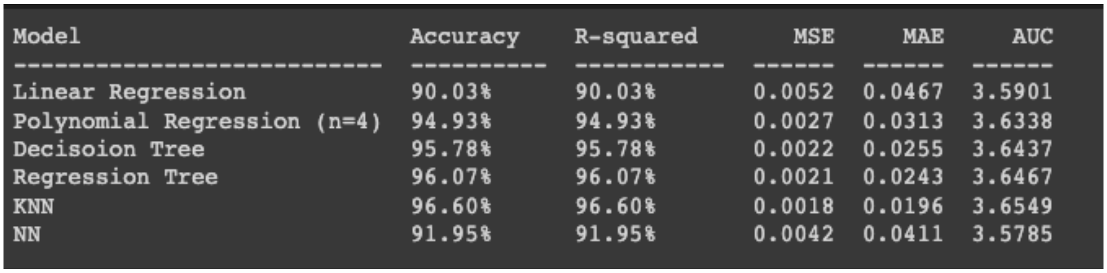


Figure 10 - Gain chart for implemented models

The training set is used to fit the model to the data. The validation set tunes the model's hyperparameters and evaluates its performance during training. The testing set is used to assess the final version of the model on unseen data. The split can be performed randomly or using a time-based approach, depending on the nature of the data. It is essential to ensure that the division is representative of the underlying distribution of the data to avoid bias in the model. Cross-validation techniques such as k-fold cross-validation can also be used to improve the robustness of the model evaluation.

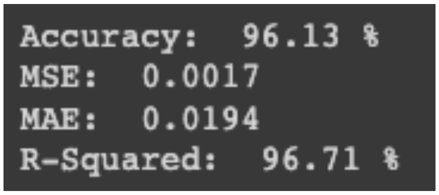
After Training these four models on the training set and validating on the validation set, we got the following results:



```
Model                        Accuracy    R-squared      MSE     MAE     AUC
---------------------------  ----------  -----------  ------  ------  ------
Linear Regression            90.03%      90.03%       0.0052  0.0467  3.5901
Polynomial Regression (n=4)  94.93%      94.93%       0.0027  0.0313  3.6338
Decisoion Tree               95.78%      95.78%       0.0022  0.0255  3.6437
Regression Tree              96.07%      96.07%       0.0021  0.0243  3.6467
KNN                          96.60%      96.60%       0.0018  0.0196  3.6549
NN                           91.95%      91.95%       0.0042  0.0411  3.5785
```

Figure 11 - Performance of various models

In Fig 9, we can observe that K-Nearest Neighbors (KNN) Regressor performs better than other models. Therefore, KNN will be selected and tested using the test set. Here’s the result obtained for the test set using Regression Tree-



```
Accuracy:  96.13 %
MSE:  0.0017
MAE:  0.0194
R-Squared:  96.71 %
```

Figure 12 - Results for test-set

V. Summary and Future Work

In this project, we used various regression techniques to predict flight ticket prices based on a set of features, including airline, flight code, source, and destination cities, departure and arrival times, number of stops, seat class, travel duration, days left to the trip, and more. We performed data exploration, preprocessing, and modeling and compared the performance of linear regression, decision tree, regression tree, KNN, and Neural Network regression models using various evaluation metrics.

Our results showed that KNN performed the best, with an R-squared value of 0.96, a mean absolute error (MAE) of 0.0196, and a mean squared error (MSE) of 0.0018. The feature importance analysis revealed that airline and number of stops were the most significant predictors of ticket prices, followed by the duration of the flight and the days left for the trip.

Several areas could be explored further to improve the model's accuracy in future work. First, additional data could be collected to include more features, such as weather conditions, airport congestion, or fuel prices, providing more information about the flight market and leading to better predictions. Second, different modeling techniques, such as more complex Neural Networks, or boosted gradient regression, could be explored to see if they provide better results than the current models. Finally, the model could be deployed online to help customers find the best flight deals based on their specific requirements and preferences.

VI. References

1. Bathwal, Shubham. (2021). Flight Price Prediction Dataset. Kaggle. Retrieved from (https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction)
2. Agrawal, A., & Thakkar, H. (2020). Flight price prediction using machine learning algorithms. International Journal of Emerging Technologies and Innovative Research, 7(3), 186-191.
3. Banik, S., Mitra, S., & Saha, S. (2021). A hybrid approach for flight fare prediction using machine learning and statistical analysis. Neural Computing and Applications, 33, 2001-2010.
4. Islam, M. A., Mahmud, M., & Islam, M. N. (2021). Flight ticket price prediction using machine learning algorithms. Journal of Engineering Science and Technology Review, 14(1), 11-16.
5. Kannan, D., & Shanthi, P. (2019). Predicting flight fare using machine learning techniques. International Journal of Engineering and Advanced Technology, 8(2), 65-69.
6. Naik, N., Nisar, T., & Patil, P. (2020). Machine learning approach for flight fare prediction. International Journal of Innovative Technology and Exploring Engineering, 9(1S2), 163-168.