AIRFARE PRICE PREDICTION

INTRODUCTION

In the contemporary landscape of the travel industry, accurate prediction of flight ticket prices is a critical capability that benefits both consumers and service providers. With fluctuating prices influenced by a myriad of factors such as time of booking, seasonal demand, and operational costs, there is a significant need for advanced analytical tools that can offer precise forecasts. The "Airfair Price Prediction System" aims to meet this demand by leveraging the power of machine learning technologies to predict flight ticket prices.

This project develops a comprehensive system using various machine learning models to analyze and predict prices based on historical data. The intent is to provide a robust predictive tool that can help consumers plan their travel budgets more effectively and enable airlines to optimize their pricing strategies dynamically. By integrating sophisticated data preprocessing techniques, feature engineering, and several state-of-the-art machine learning algorithms, the system seeks to achieve high accuracy in predicting airfair prices.

Our approach systematically breaks down the complexity of the data, which includes various features like airlines, flight numbers, departure and arrival times, and more, to uncover underlying patterns that affect ticket pricing. The Airfair Price Prediction System is not just a tool for cost estimation but also serves as a strategic asset for understanding market dynamics and enhancing operational decisions in the travel sector.

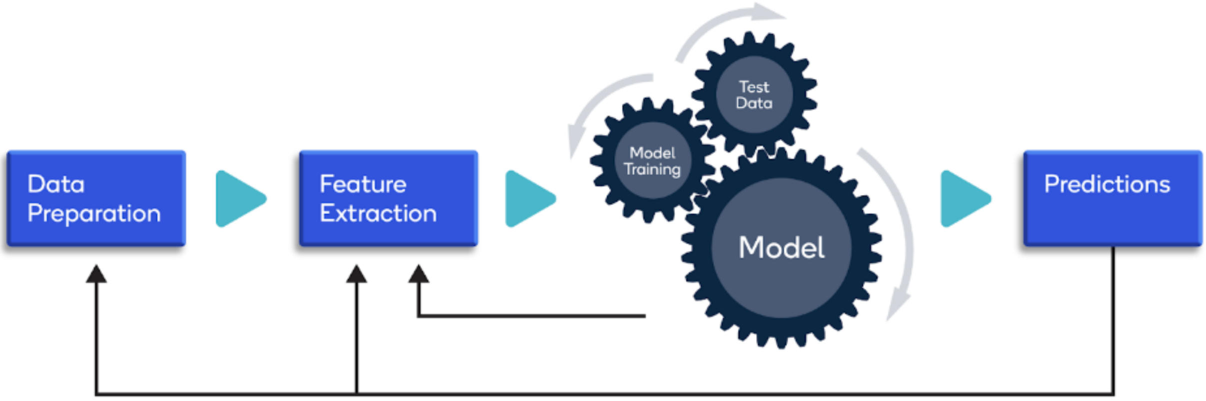
Through this project, we aim to bridge the gap between data-driven insights and practical applications, thus empowering stakeholders with the ability to make informed decisions that ultimately lead to enhanced customer satisfaction and business profitability.

STEPS INVOLVED

The steps involved in this were as follows-

1. Exploratory Data Analysis (EDA)
2. Handling Categorical Data
3. Feature Selection
4. Applying different Regression Model's
5. Evaluating Model's Performance

PROCESS



PRIMARY OBJECTIVES

The primary objective of the Airfair Price Prediction are:

1. It allows travelers to predict and plan their expenses more effectively by providing an estimate of flight costs well in advance.

2. Airlines and travel agencies can use predictive insights to optimize their pricing strategies, potentially maximizing their revenue through dynamic pricing based on demand, time of booking, flight schedules, and other factors.

3. Understanding how different variables such as time of day, the airline, the number of stops, and others affect flight prices can offer deeper insights into market dynamics and consumer preferences.

1. Exploratory Data Analysis (EDA)-

1.1 Data Acquisition:

The dataset for airfair price prediction was obtained, consisting of 300,153 observations with 12 features such as airline, flight, source_city, departure_time, etc.

In [3]:
```
df = pd.read_csv('./Flight_Price_Prediction/Clean_Dataset.csv')
df
```

Out[3]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
300148	300148	Vistara	UK-822	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	69265
300149	300149	Vistara	UK-826	Chennai	Afternoon	one	Night	Hyderabad	Business	10.42	49	77105
300150	300150	Vistara	UK-832	Chennai	Early_Morning	one	Night	Hyderabad	Business	13.83	49	79099
300151	300151	Vistara	UK-828	Chennai	Early_Morning	one	Evening	Hyderabad	Business	10.00	49	81585
300152	300152	Vistara	UK-822	Chennai	Morning	one	Evening	Hyderabad	Business	10.08	49	81585

300153 rows × 12 columns

1.2. Initial Data Assessment:

Using Python's Pandas library, we loaded the dataset and conducted an initial examination to understand the structure and content of the data. We performed checks for:

- Data types and formats.
- The presence of any missing values.
- Basic statistical descriptions.
- Checking outliers

In [5]:
```
df.dtypes
```

Out[5]:
```
airline             object
flight              object
source_city         object
departure_time      object
stops               object
arrival_time        object
destination_city    object
class               object
duration           float64
days_left            int64
price                int64
dtype: object
```

In [6]:
```
df.describe()
```

Out[6]:

	duration	days_left	price
count	300153.000000	300153.000000	300153.000000
mean	12.221021	26.004751	20889.660523
std	7.191997	13.561004	22697.767366
min	0.830000	1.000000	1105.000000
25%	6.830000	15.000000	4783.000000
50%	11.250000	26.000000	7425.000000
75%	16.170000	38.000000	42521.000000
max	49.830000	49.000000	123071.000000

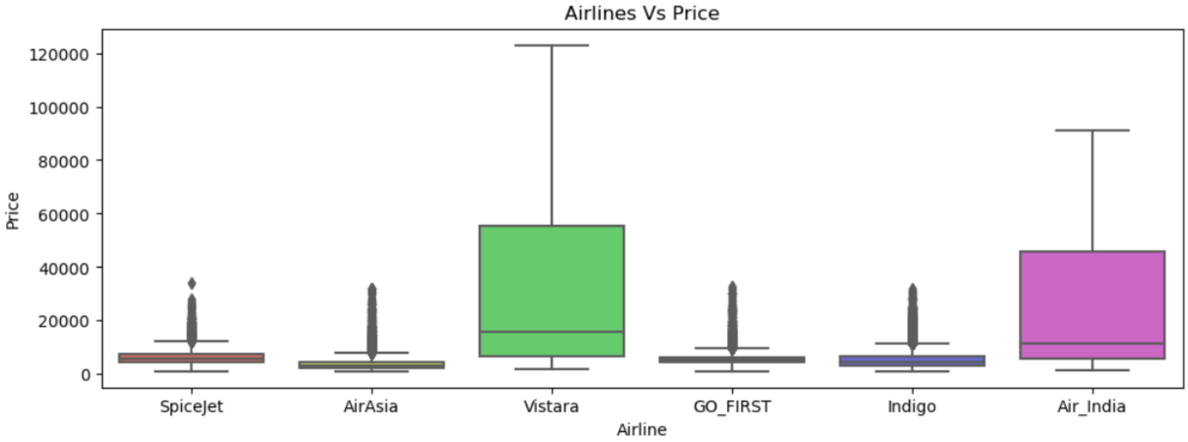
1.3. Data Cleaning:

Our initial assessment uncovered an unnecessary column, 'Unnamed: 0', which was promptly dropped. We employed the LabelEncoder from Scikit-learn to convert categorical text data into a model-readable numerical format.

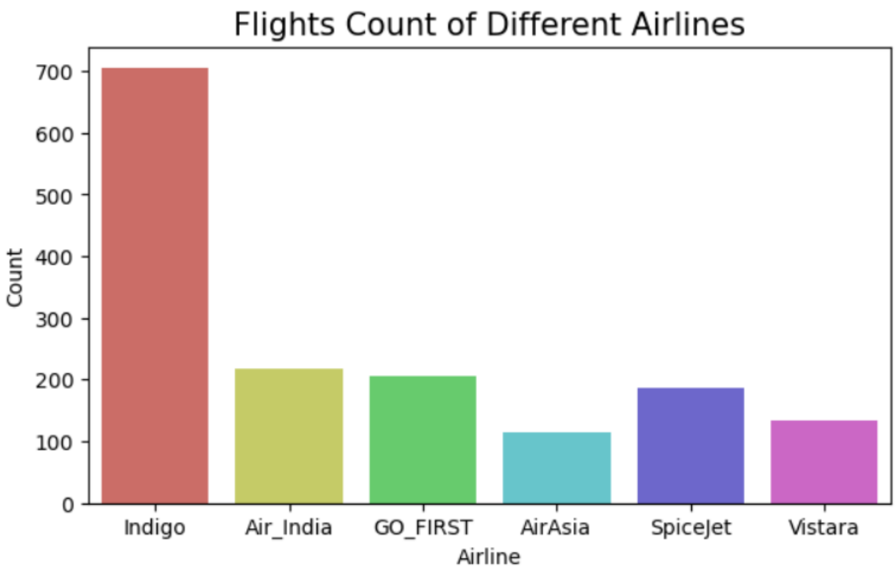
1.4. Data Visualization:

For visual exploration, we employed Seaborn and Matplotlib libraries to create:

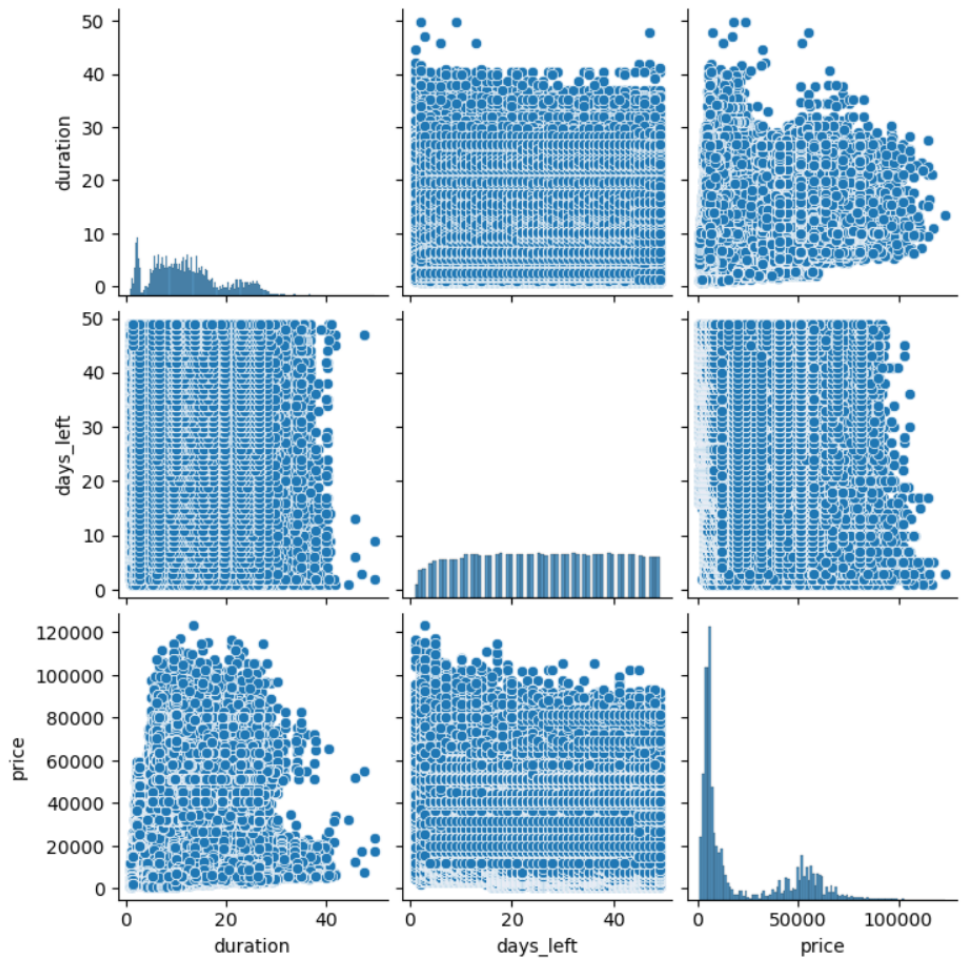
- Boxplot to visualize the distribution of ticket prices.



- Count plots to ascertain the frequency of flights for each airline, identifying Indigo as the most popular airline.



- Pairplot to see the distribution of the datapoints in each feature.



Duration and Price: There is a trend where shorter durations seem to have a wide range of ticket prices, whereas longer durations have increasingly higher prices. This suggests that longer flights are generally more expensive, but there's considerable variability for shorter flights.

Days Left and Price: The relationship between the number of days left to book a flight and the price appears to be fairly dispersed, indicating that the time left before the flight does not have a clear linear relationship with the price. However, there seems to be a slight concentration of lower prices when days left are higher, hinting at possible lower prices for earlier bookings.

Duration and Days Left: There doesn't appear to be a clear relationship between the duration of the flight and the number of days left before the flight. The points are evenly scattered, suggesting that these two variables don't influence each other significantly.

2. Data Preprocessing-

2.1. Dealing with Categorical Data:

We encountered several categorical variables such as airline, flight, source_city, departure_time, etc. Machine learning models require numerical input, so we used the **LabelEncoder** from Scikit-learn to convert these categories into numerical labels without introducing any ordinal relationship where inappropriate.

```
In [8]: from sklearn.preprocessing import LabelEncoder

        categorical_columns = ['airline', 'flight', 'source_city', 'departure_time', 'stops', 'arrival_time','destination_city', 'class']

        le = LabelEncoder()

        for i in categorical_columns:
            df[i] = le.fit_transform(df[i])

        df
```

Out[8]:

	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price
0	4	1408	2	2	2	5	5	1	2.17	1	5953
1	4	1387	2	1	2	4	5	1	2.33	1	5953
2	0	1213	2	1	2	1	5	1	2.17	1	5956
3	5	1559	2	4	2	0	5	1	2.25	1	5955
4	5	1549	2	4	2	4	5	1	2.33	1	5955
...	...	...	...	...	...	...	...	...	...	...	...
300148	5	1477	1	4	0	2	3	0	10.08	49	69265
300149	5	1481	1	0	0	5	3	0	10.42	49	77105
300150	5	1486	1	1	0	5	3	0	13.83	49	79099
300151	5	1483	1	1	0	2	3	0	10.00	49	81585
300152	5	1477	1	4	0	2	3	0	10.08	49	81585

300153 rows × 11 columns

2.2. Standardization:

The features had varying scales, and to ensure that our models treated all features equally, we used **StandardScaler** to normalize the feature space. This standardization ensured that variables with larger values didn't disproportionately influence the model.

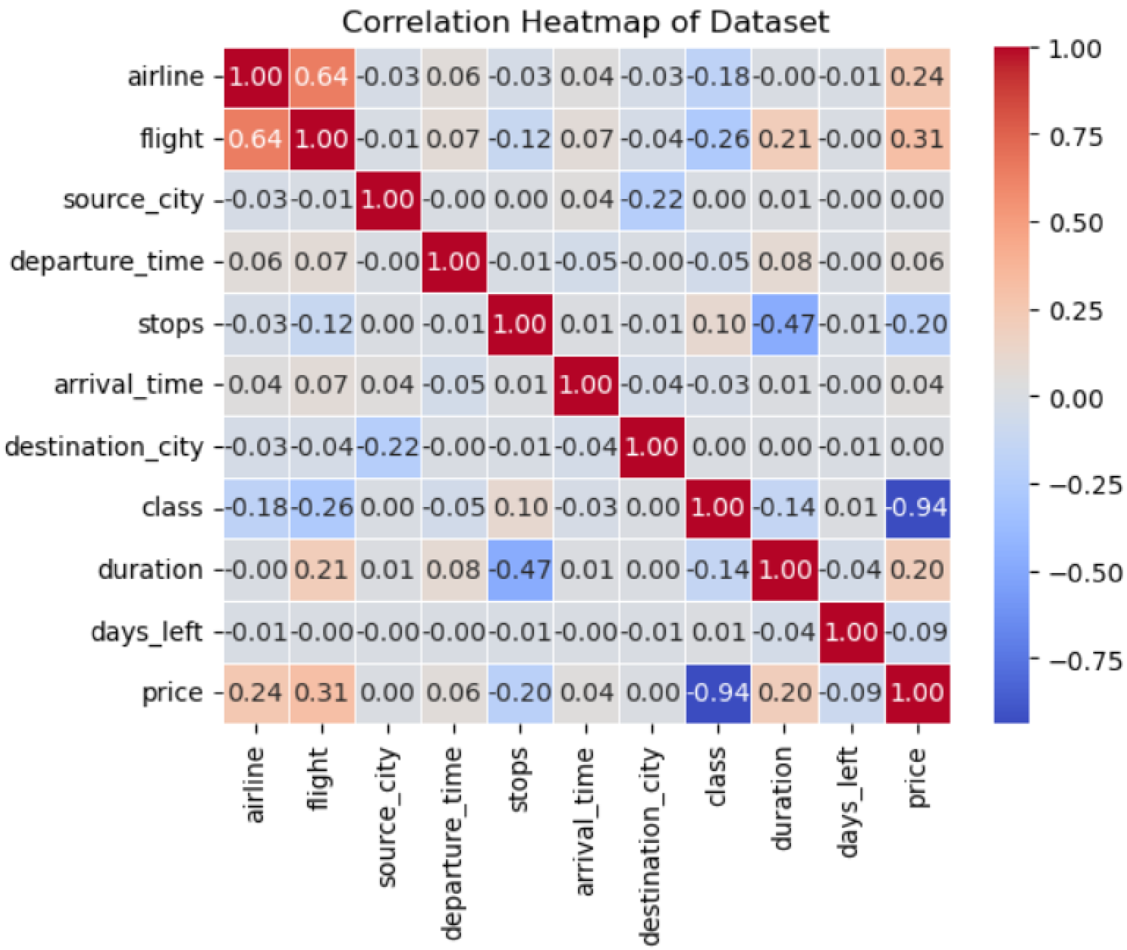
2.3. Handling Missing Values:

During the EDA phase, we established that the dataset had no missing values, thus allowing us to skip imputation strategies commonly used in preprocessing.

3. Feature Selection & PCA-

3.1. Correlation Analysis:

We created a heatmap to visualize the correlation between different features. This helped identify features that are highly correlated with the target variable 'price' and among themselves, which is crucial for the feature selection process.



3.2. Dimensionality Reduction:

We utilized Principal Component Analysis (PCA) to reduce the dimensionality of the data. This was after assessing the correlation heatmap to identify multicollinear features. PCA was applied to distill the information into a smaller number of principal components that capture the most variance in the data.

When we are considering 8 features we are getting 90% of information. So, proceeding using all 10 features.

4. Modelling-

4.1. Data Splitting:

The final step was to split the dataset into training and testing sets to evaluate the performance of our machine learning models objectively. We used an 80-20 split, maintaining a sizable amount of data for both training and evaluation.

4.2. Algorithms Used:

- Linear Regression
- K-Nearest Neighbors Regression (KNN)
- Random Forest
- XGBoost
- Neural Network

5. Results-

5.1. Performance Metrics:

Models were compared based on metrics like Mean Error (ME), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE).

5.2. Outcome:

XGBoost performed better in terms of both accuracy and error metrics compared to Linear Regression and KNN.

Model	MAE	MAPE	RMSE	R^2
Linear Regression	4622.187	43.444%	7013.558	0.90
K-Nearest Neighbors (KNN)	10747.308	99.32%	15681.941	0.52
Random Forest	4858.232	50.488%	6989.166	0.91
XGBoost	3019.499	23.095%	4930.694	0.95