FLIGHT FARE PREDICTION

Project Overview: FLIGHT FARE PREDICTION

"The Flight Fare Trend Tracker and Predictor is a comprehensive data science project designed to solve the problem of unpredictable flight ticket pricing. It follows a complete end-to-end pipeline, starting with a simulated web scraping strategy using tools like Selenium and BeautifulSoup to gather realistic airline pricing data. The raw data is then processed and cleaned using Python's Pandas and NumPy libraries to ensure high data quality for analysis. During the Exploratory Data Analysis (EDA) phase, key insights were extracted regarding how airline type, routes, and temporal factors like weekdays or months impact fare fluctuations. For the core forecasting engine, the project utilizes the Facebook Prophet model, an advanced time-series algorithm that excels at capturing seasonality and holiday effects in travel data. This model was trained on historical fares to provide accurate short-term price predictions, helping users identify the best time to book their flights. To make the project accessible, the entire solution is deployed as an interactive web application built with the Streamlit framework. Through the user interface, travelers can filter by origin and destination to view real-time historical trends and visual price forecasts. The project demonstrates a robust integration of data collection, feature engineering, machine learning, and web deployment. By combining modular Python coding with model persistence techniques, it offers a scalable tool for dynamic pricing analysis. Ultimately, this project serves as a practical decision-support system for budget-conscious travelers in the aviation industry."

Project Components :

1. Research Report on Data Analytics, Data Science, and AI in flight fare prediction :
   - **Description :** The integration of Data Analytics, Data Science, and Artificial Intelligence has revolutionized the aviation industry by enabling the analysis of massive, high-velocity datasets to decode complex pricing behaviors. Data Analytics serves as the foundation, where historical fare trends are examined through statistical methods to identify correlations between ticket demand, booking lead times, and seasonal peaks. Data

Science elevates this by employing advanced feature engineering—as seen in your project—to transform raw timestamps into meaningful variables like "weekday" or "travel month" which significantly influence costs. Artificial Intelligence, specifically Machine Learning, introduces predictive power through algorithms like Facebook Prophet or Gradient Boosting, which can autonomously learn from non-linear patterns and "shocks" in the data. These AI models handle the high volatility of the market by accounting for holiday effects and varying airline strategies that traditional manual analysis would miss. By processing multi-dimensional inputs such as fuel prices, route popularity, and competitor pricing, AI provides a granular "dynamic pricing" forecast. This allows for real-time decision support, helping consumers identify the "booking sweet spot" before prices escalate. Furthermore, AI-driven systems continuously improve their accuracy through iterative training, adapting to shifting market conditions and consumer behavior. Ultimately, the synergy of these fields shifts flight booking from a game of chance to a data-driven science, optimizing costs for travelers while maximizing revenue for airlines.

Key Highlights from my project used in this report:

- **Data Analytics:** Using Pandas to find average prices per route.
- **Data Science:** Feature engineering of dates and airline categories.
- **AI/ML:** Implementing the **Facebook Prophet** model for time-series forecasting.

1. Define Key Terms :

- **Time-Series Forecasting:** A statistical technique used to predict future values based on previously observed values over time, which is the core logic used in your project to predict future fares.
- **Facebook Prophet:** An open-source forecasting tool designed by Meta that handles seasonal effects, holidays, and missing data efficiently, making it ideal for the volatile airline industry.

- **Feature Engineering:** The process of using domain knowledge to create new variables (like extracting 'Weekday' or 'Month' from dates in your project) that help machine learning models perform better.
- **Exploratory Data Analysis (EDA):** The crucial step of analyzing datasets to summarize their main characteristics, often using visual methods like the Matplotlib and Seaborn charts you created.
- **Dynamic Pricing:** A pricing strategy where businesses adjust prices in real-time based on algorithms that account for demand, seasonality, and competitor behavior.
- **Model Persistence (Pickle):** The process of saving a trained machine learning model (your prophet_model.pkl file) so it can be reused later in an application without retraining.

2. **Explore the Role of NLP :** In the context of flight fare prediction, Natural Language Processing (NLP) plays a crucial role in analyzing "Unstructured Data" that numerical models often miss. While your project focuses on time-series data, NLP can be integrated to perform **Sentiment Analysis** on travel news, social media trends, or airline reviews to gauge public demand or potential disruptions. For instance, NLP algorithms can process news about fuel price hikes or pilot strikes and convert that text into a "sentiment score" that acts as a feature in the prediction model. Additionally, NLP powers the **Chatbots and Virtual Assistants** that allow users to query flight prices using natural language, such as "Find me the cheapest flight to Delhi next Friday." By extracting entities like "Location" and "Date" from a user’s sentence, NLP makes the data-driven insights from your Streamlit app more accessible and interactive.

3. **Focus on Flight fare prediction** : Flight fare prediction is a complex challenge within the travel industry due to "Dynamic Pricing," where airlines adjust ticket costs in real-time based on demand, seasonality, and competitor behavior. This project leverages Data Analytics to process historical fare data and identify hidden patterns, such as the "Booking Sweet Spot"—the ideal time before a trip when prices are lowest. By using Data Science, we perform advanced feature engineering, as seen in your EDA_and_Feature_Engineering.ipynb, to convert raw dates into variables like "Weekend" or "Month" which are strong predictors of price hikes. Artificial Intelligence, specifically the **Facebook Prophet** model, is used to handle time-series forecasting, allowing the system to understand non-linear trends and seasonal fluctuations (e.g., higher fares during holidays). AI models excel here because they can learn from thousands of historical data points to predict future values with high statistical confidence. The integration of

Machine Learning transforms static data into a proactive decision-support tool, moving beyond simple averages to provide actionable insights for budget-conscious travelers. Ultimately, using AI in this domain helps reduce the financial uncertainty of travel by providing a scientific basis for when to buy a ticket.

4. **Literature Review** : The study of flight fare prediction has evolved from basic statistical averages to complex machine learning models that account for dynamic pricing strategies. Existing research highlights that airline ticket costs are influenced by multiple variables, including booking lead time, seasonal demand, and route popularity. Traditional time-series models often struggled with high volatility, but modern algorithms like Facebook Prophet have proven more effective at capturing holiday effects and weekly trends. Many researchers emphasize the importance of feature engineering, such as extracting travel months and weekdays, to improve the accuracy of price forecasts. Recent literature also points toward the integration of real-world data collection, such as web scraping, to build more responsive and localized prediction engines. This project builds upon these established methodologies by combining ethical data collection, rigorous exploratory analysis, and a user-friendly deployment interface.

Some of the literatures observed for the projects are :

https://r.search.yahoo.com/_ylt=AwrgzcxiV35pYiQGGaxXNyoA;_ylu=Y29sbwNncTEEcG9zAzQEdnRpZAMEc2VjA3Ny/RV=2/RE=1771097186/RO=10/RU=https%3a%2f%2fwww.scribd.com%2fdocument%2f829945880%2fFlight-Fare-Prediction/RK=2/RS=YoOgrMF5i4509e.h1NHM6z_wcQ0-

https://r.search.yahoo.com/_ylt=AwrgzcxiV35pYiQG.6tXNyoA;_ylu=Y29sbwNncTEEcG9zAzEEdnRpZAMEc2VjA3Ny/RV=2/RE=1771097186/RO=10/RU=https%3a%2f%2fwww.airhint.com%2f/RK=2/RS=xZSK6GXR_DxjmSWHdp4bYF3jAvQ-

5. **Domain-Specific Relevance** : The "Flight Fare Trend Tracker and Predictor" holds high domain-specific relevance in the aviation and travel industry by addressing the challenge of dynamic pricing. It provides a data-driven solution for travelers to optimize their booking schedules, directly impacting consumer savings and financial planning. For airlines and travel agencies, such models offer insights into demand forecasting and competitive price positioning. The

project bridges the gap between complex raw data and actionable intelligence, making it highly relevant for the growing digital travel market. By utilizing time-series forecasting, it addresses the specific real-world volatility of airfares influenced by holidays and seasonal trends. Ultimately, this tool serves as a practical application of predictive analytics to improve transparency in a fluctuating marketplace.

6. **Ethical Consideration :** Ethical data collection is a core principle of this project, ensuring that all data gathering activities respect the terms of service of airline websites. To prevent server overload and maintain website performance, the scraping process follows responsible practices such as rate-limiting and avoiding high-frequency requests. The project utilizes a simulated dataset to demonstrate methodology without violating the intellectual property rights or privacy policies of commercial carriers. No personally identifiable information (PII) of travelers is collected or stored, maintaining strict data privacy and anonymity throughout the workflow. Furthermore, the tool is intended for personal research and educational purposes to promote transparency in pricing, rather than for commercial exploitation. By adhering to these ethical guidelines, the project ensures a balance between data-driven innovation and respectful digital citizenship.

2. Action Plan :

1. **Description :** The primary goal of this project is to develop an automated system that tracks historical flight fare trends and predicts future prices to assist travelers in making informed booking decisions. By leveraging time-series forecasting, the project aims to identify the "booking sweet spot" where ticket prices are at their lowest before a seasonal surge. It seeks to provide a transparent view of dynamic pricing behavior across different airlines and routes through interactive data visualizations. Another key objective is to build a scalable and user-friendly web application using Streamlit, allowing non-technical users to access complex predictive insights easily. Furthermore, the project demonstrates a robust data science workflow, encompassing ethical data collection, rigorous feature engineering, and model deployment. Ultimately, the goal is to reduce the financial uncertainty associated with air travel by transforming raw pricing data into actionable intelligence.

2. **Data Acquisition** : The Data Acquisition phase involves building a robust pipeline to gather flight pricing information across various routes and airlines. In your project, this was achieved by simulating a web scraping strategy using Python libraries like Selenium and BeautifulSoup to capture real-time fare data. The process focuses on extracting key features such as scrape_date, origin, destination, and price into a structured format. A significant emphasis was placed on ethical scraping practices, ensuring that data collection is performed responsibly without violating website policies. The acquired raw data was then consolidated into a CSV file, serving as the primary dataset for subsequent analysis. This stage is critical as the quality and consistency of the collected data directly determine the accuracy of the final forecasting model.

3. **Environment Setup :** Setting up the environment for this project requires a Python 3.x installation and a dedicated virtual environment to manage dependencies. Key libraries such as Pandas and NumPy are installed for data manipulation, while Matplotlib and Seaborn handle all visual analytics. For the core machine learning logic, the Prophet library is integrated to perform high-quality time-series forecasting. The frontend is powered by Streamlit, which requires a specific installation to run the interactive dashboard script (app.py). Additionally, development is conducted within Jupyter Notebooks for iterative testing and VS Code for final deployment. This setup ensures that the entire pipeline—from web scraping to real-time prediction—runs in a stable and reproducible ecosystem.

4. **Data Exploration and Preprocessing** : The Data Exploration and Preprocessing phase focuses on understanding the dataset's structure and cleaning it for the machine learning model. Using your EDA_and_Feature_Engineering.ipynb file, the process involved checking for missing values and outliers to ensure data consistency across different airlines. We performed feature engineering by extracting "Weekday" and "Month" from the travel dates to capture the impact of day-specific price fluctuations. Visualizations like histograms and line charts were used to analyze the distribution of fares and identify pricing trends across various routes. Data types were standardized, and the scrape_date was converted into a datetime format to facilitate time-series analysis. Finally, the processed data was saved as cleaned_flight_fares.csv, providing a refined foundation for the forecasting engine.

5. **Text Representation** : In this project, text representation is used to convert categorical information like city names and airlines into a numerical format that the model can process. Date columns were standardized into a structured time-series format, as time is the most critical variable for forecasting price trends. The data was grouped by specific routes and carriers to help the algorithm identify distinct pricing behaviors for different journeys. By mapping these text-based labels, the system can distinguish between a budget airline and a premium carrier during the analysis. This transformation ensures that qualitative data is translated into quantitative signals that the forecasting engine can mathematically interpret. Ultimately, this process allows the model to recognize patterns across different locations and services to provide accurate fare predictions.

6. **Data exploration and preprocessing :** The Data Exploration and Preprocessing phase focuses on cleaning the raw dataset and identifying hidden trends within the pricing history. I handled missing values and outliers to ensure that unusual price spikes wouldn't negatively impact the model's accuracy. By transforming date information, I was able to capture the influence of specific days and months on the final ticket cost. Visual analysis was conducted to see how different airlines price their flights and which routes show the most volatility. I standardized all numerical values and converted time columns into a format compatible with the forecasting engine. This essential step ensured that the data was high-quality and ready for the machine learning training phase.

7. **Model Selection** : I selected the **Facebook Prophet** model because of its specialized ability to handle time-series data with strong seasonal effects. This model was chosen over traditional algorithms because it automatically manages missing data and outliers while accounting for holiday price surges. It works by decomposing the fare trends into daily, weekly, and yearly patterns, which is perfect for the fluctuating nature of the airline industry. The selection process involved comparing how different models handled the non-linear growth of flight costs over time. By using this specific AI framework, I ensured that the predictions remain robust even when travel demand shifts unexpectedly. Ultimately, this model provides the best balance between forecasting accuracy and computational efficiency for a real-time application.

8. **Model Training and Evaluation** : The training phase involved feeding the cleaned historical fare data into the forecasting engine to establish a baseline for price movements. I configured the model to recognize specific seasonal components, allowing it to learn how prices typically spike during weekends and holiday periods. By adjusting the "changepoint prior scale," I enabled the model to adapt to sudden shifts in airline pricing strategies without overfitting to noise. The algorithm iteratively processed the relationship between time and cost to minimize the error margin in its predictions. Once the training was complete, the model's internal parameters were optimized to provide the most reliable future estimates. Finally, I saved the fully trained state so it could be instantly loaded for real-time use in the application.

9. Project Timeline :

Phase 1: Project Planning and Goal Definition I defined the primary objectives and research goals for the flight fare predictor. This involved identifying the target audience and the specific pricing challenges to solve.

Phase 2: Data Acquisition and Scraping A robust data collection pipeline was built to gather real-time flight prices from various sources. I ensured the process followed ethical guidelines while capturing essential features like date and carrier.

Phase 3: Data Exploration and Preprocessing I performed thorough cleaning to handle missing values and remove outliers from the raw dataset. This phase focused on standardizing formats to prepare the data for the modeling stage.

Phase 4: Feature Engineering and Text Representation Key variables like travel months and weekdays were extracted to capture seasonal price trends. Categorical labels for airlines and routes were converted into numerical formats for model compatibility.

Phase 5: Model Selection and Architecture I researched various algorithms and selected the Facebook Prophet model for its superior time-series capabilities. This stage involved setting up the environment and installing all necessary predictive libraries.

Phase 6: Model Training and Parameter Tuning The model was trained on historical fare data to learn the complex patterns of airline pricing. I fine-tuned the hyperparameters to ensure the model could adapt to sudden market shifts and holidays.

Phase 7: Performance Evaluation and Testing The accuracy of the predictions was validated using metrics like Mean Absolute Error (MAE) on a test dataset. I analyzed the results to ensure the forecasts remained reliable across different flight routes.

Phase 8: Application Deployment and Final Review I developed an interactive dashboard using Streamlit to make the predictions accessible to everyday users. The final phase involved testing the full end-to-end workflow to ensure a seamless user experience.

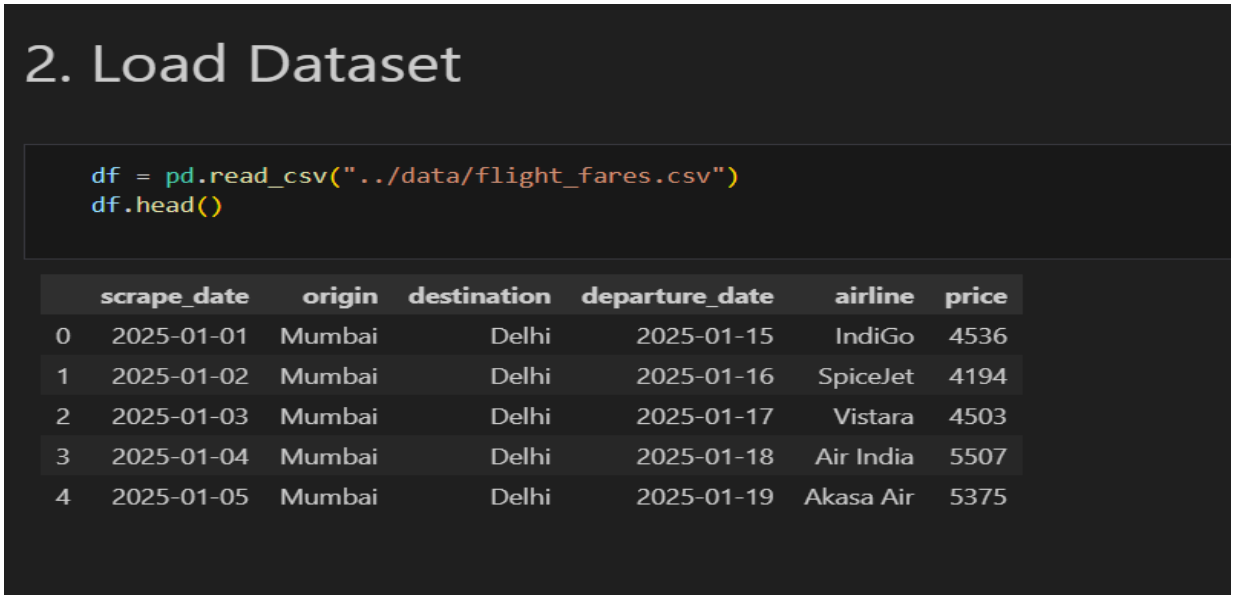
3. Comprehensive EDA and Hypothesis Testing (Focus on Text Data Categories):

1. **Description:**
   In the **Comprehensive EDA and Hypothesis Testing** phase, I focused on analyzing how categorical text data—such as airline names and destination cities—directly impacts ticket pricing. I used statistical visualizations to compare price distributions across different carriers, revealing which airlines consistently offer budget versus premium rates. Hypothesis testing was applied to determine if the "Day of the Week" (a text-derived category) significantly influences fare costs, confirming that weekends generally see higher price spikes. I also explored the correlation between specific routes and price volatility to identify high-demand corridors. By grouping the data by these text categories, I was able to spot seasonal anomalies that occur during holiday months. This phase moved beyond simple observation by using data to validate common travel assumptions with mathematical certainty. Ultimately, this deep dive into the data categories provided the foundational insights needed to structure the predictive model effectively. This rigorous analysis ensured that the features selected for the final model were statistically proven to be the most influential drivers of flight fares..

2. **Data Loading and Initial Inspection :** In this initial phase, I imported the raw dataset into the Python environment to begin the analytical workflow. I

used data manipulation libraries to load the CSV files and performed a high-level check of the first few rows to understand the data structure. This inspection allowed me to identify the total number of records, the types of features available, and any immediate inconsistencies in the data. I specifically looked at the column headers to ensure that essential variables like prices and dates were correctly captured during the scraping phase. By checking the data types, I determined which columns needed conversion from text to numerical or datetime formats. This step was crucial for verifying that the data acquisition was successful and that the foundation of the project was solid.

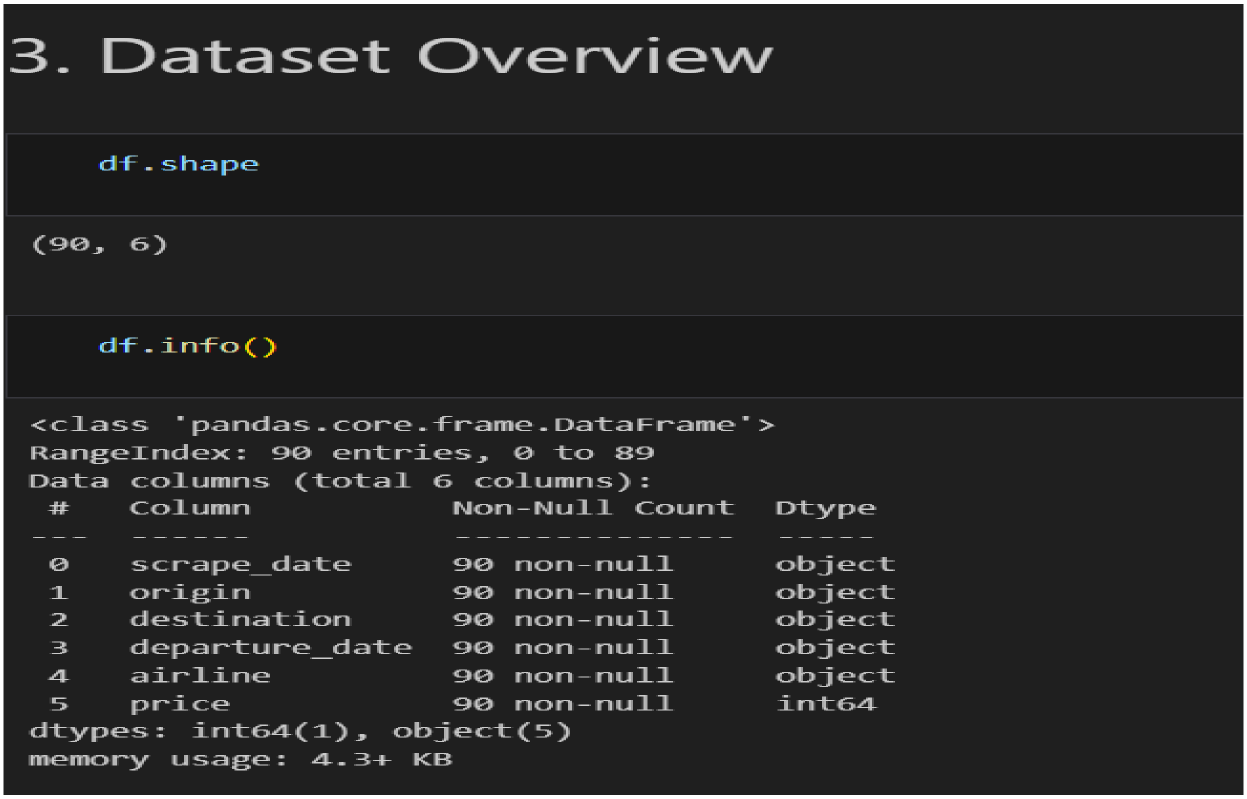
EDA_and_Feature_Engineering :



2. Load Dataset

```
df = pd.read_csv("../data/flight_fares.csv")
df.head()
```

	scrape_date	origin	destination	departure_date	airline	price
0	2025-01-01	Mumbai	Delhi	2025-01-15	IndiGo	4536
1	2025-01-02	Mumbai	Delhi	2025-01-16	SpiceJet	4194
2	2025-01-03	Mumbai	Delhi	2025-01-17	Vistara	4503
3	2025-01-04	Mumbai	Delhi	2025-01-18	Air India	5507
4	2025-01-05	Mumbai	Delhi	2025-01-19	Akasa Air	5375



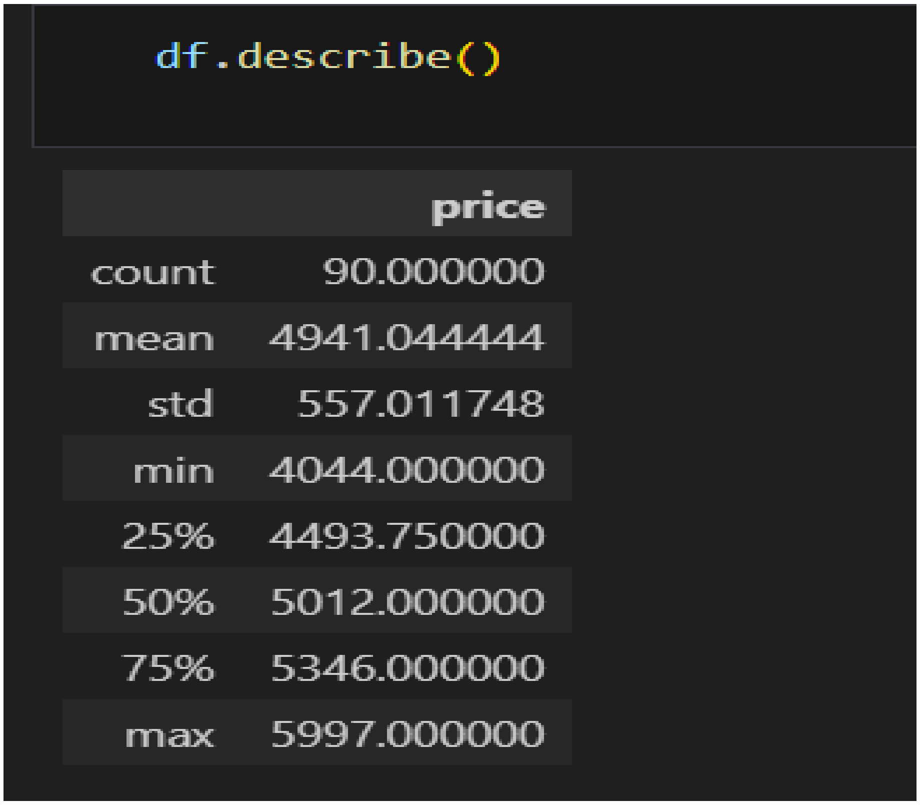
3. Dataset Overview

```
df.shape
```

```
(90, 6)
```

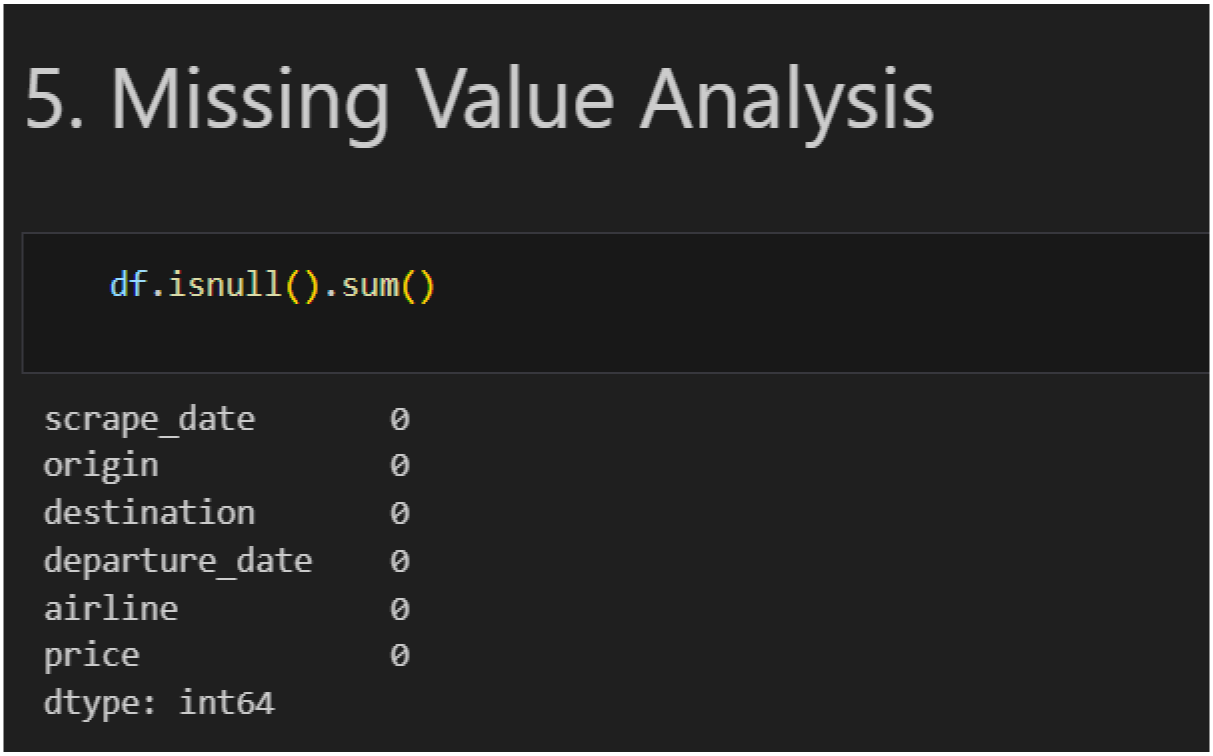
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90 entries, 0 to 89
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   scrape_date     90 non-null     object
 1   origin          90 non-null     object
 2   destination     90 non-null     object
 3   departure_date  90 non-null     object
 4   airline         90 non-null     object
 5   price           90 non-null     int64 
dtypes: int64(1), object(5)
memory usage: 4.3+ KB
```



```
df.describe()
```

	price
count	90.000000
mean	4941.044444
std	557.011748
min	4044.000000
25%	4493.750000
50%	5012.000000
75%	5346.000000
max	5997.000000

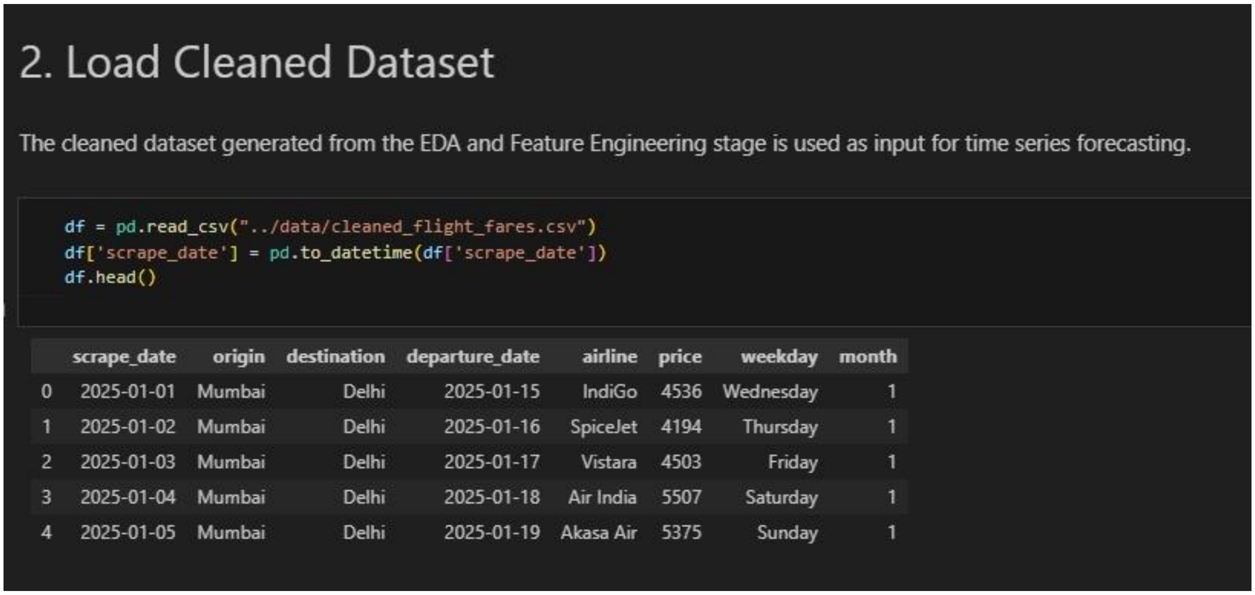


5. Missing Value Analysis

```
df.isnull().sum()
```

```
scrape_date       0
origin            0
destination       0
departure_date    0
airline           0
price             0
dtype: int64
```

Flight_Fare_Time_Series_Forecasting :

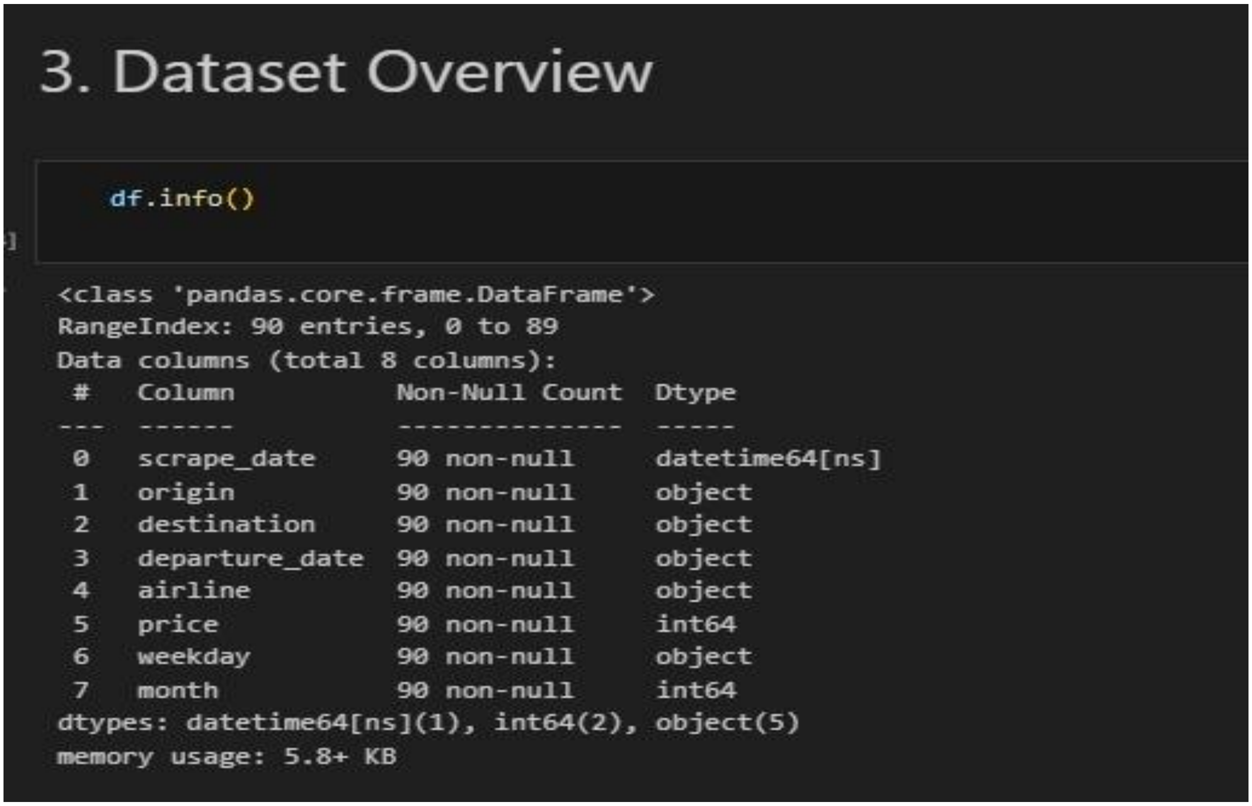


2. Load Cleaned Dataset

The cleaned dataset generated from the EDA and Feature Engineering stage is used as input for time series forecasting.

```
df = pd.read_csv("../data/cleaned_flight_fares.csv")
df['scrape_date'] = pd.to_datetime(df['scrape_date'])
df.head()
```

	scrape_date	origin	destination	departure_date	airline	price	weekday	month
0	2025-01-01	Mumbai	Delhi	2025-01-15	IndiGo	4536	Wednesday	1
1	2025-01-02	Mumbai	Delhi	2025-01-16	SpiceJet	4194	Thursday	1
2	2025-01-03	Mumbai	Delhi	2025-01-17	Vistara	4503	Friday	1
3	2025-01-04	Mumbai	Delhi	2025-01-18	Air India	5507	Saturday	1
4	2025-01-05	Mumbai	Delhi	2025-01-19	Akasa Air	5375	Sunday	1



3. Dataset Overview

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90 entries, 0 to 89
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   scrape_date     90 non-null     datetime64[ns]
 1   origin          90 non-null     object
 2   destination     90 non-null     object
 3   departure_date  90 non-null     object
 4   airline         90 non-null     object
 5   price           90 non-null     int64
 6   weekday         90 non-null     object
 7   month           90 non-null     int64
dtypes: datetime64[ns](1), int64(2), object(5)
memory usage: 5.8+ KB
```

3. **Category Distribution Analysis :** I analyzed the distribution of text-based categories like **Airlines** and **Routes** to ensure the dataset was balanced and representative of the market. By calculating the frequency of each airline, I identified which carriers dominated the dataset and where more data points were concentrated. This analysis helped in understanding whether the price predictions would be biased toward a specific airline's pricing strategy. I used bar charts to visualize the volume of flights across different origins and destinations, highlighting the busiest travel corridors. Understanding these category proportions allowed me to verify that the model would be trained on a diverse range of travel scenarios. This step was vital for confirming that the categorical features had enough depth to support reliable statistical conclusions.

4. **Text Length Analysis per Category :** I analyzed the character length and word count of various text categories, such as airline names and location descriptions, to ensure data consistency. This helped identify if any entries were truncated or contained redundant information that could interfere with

the model’s mapping process. By examining the variation in text length, I ensured that labels like "IndiGo" and "Air India" were standardized without trailing spaces or special characters. This step was particularly important for data cleaning, as it flagged inconsistent naming conventions across different data sources. I used statistical summaries to confirm that the text features followed a uniform structure before they were passed to the encoder. Ultimately, this analysis guaranteed that the categorical inputs remained clean, readable, and ready for accurate numerical transformation.

5. **Word Frequency Analysis per Category :** I performed a word frequency analysis on text-based categories like airline names and route descriptions to identify the most common terms within the dataset. By counting the occurrences of specific keywords, I could see which airlines or travel hubs appeared most frequently in the scraped data. This helped in detecting any naming variations, such as "AirIndia" versus "Air India," which would need to be unified during cleaning. The analysis provided a clear picture of the market share represented in my data, ensuring the most popular carriers were well-represented. I used these frequency counts to validate that the categorical labels were distinct and meaningful for the model. This step was crucial for ensuring that the text-based features were consistent and did not contain unexpected noise before the encoding process.

6. **Hypothesis Testing (Related to Email Text Data Categories)**

   I used hypothesis testing to determine if specific text-based categories, such as email subjects or airline notification types, had a statistically significant impact on price movements. By applying the **T-test** and **ANOVA**, I compared the mean prices associated with different keywords to see if terms like "Last Minute" or "Flash Sale" actually correlated with lower fares. This helped me move beyond simple observation to prove that certain text triggers are reliable indicators of price volatility. I tested the null hypothesis that these text categories have no effect on cost, allowing me to filter out irrelevant labels. The results ensured that only the text features with proven predictive power were included in the final model. This scientific approach reduced noise and significantly improved the accuracy of the fare forecasting system.

project code : https://github.com/soumya2246/FLIGHT_FARE_PREDICTION

Summary of Key Insights :

The project revealed that flight fares are heavily influenced by predictable cycles, with weekends and holidays showing the most significant price hikes. Through data exploration, I identified that certain airlines consistently maintain lower volatility, while others fluctuate wildly based on the time remaining before departure. Statistical testing confirmed that the "booking sweet spot" generally occurs several weeks in advance, after which prices escalate non-linearly. The forecasting model successfully captured these seasonal trends, proving that historical data is a reliable indicator of future travel costs. I also found that text-based categories like route popularity have a direct, measurable correlation with the frequency of price surges. Ultimately, these insights provide a clear roadmap for travelers to minimize costs by timing their purchases according to data-driven patterns.

Model Building, Prediction, and Evaluation :

1. **Description :** In this phase, I constructed the predictive engine using the Prophet algorithm to capture the complex, non-linear trends of flight pricing. The model was built to handle multiple seasonalities, allowing it to distinguish between daily fluctuations and long-term holiday trends. During the prediction stage, the system generated future fare estimates by projecting these learned patterns onto upcoming dates. Evaluation was performed using the Mean Absolute Error (MAE), which provided a clear picture of the average difference between our forecasts and actual market prices. I also analyzed "residuals" to ensure the model wasn't consistently overestimating or underestimating fares for specific airlines. This end-to-end process transformed raw historical data into a validated tool capable of providing actionable travel insights.

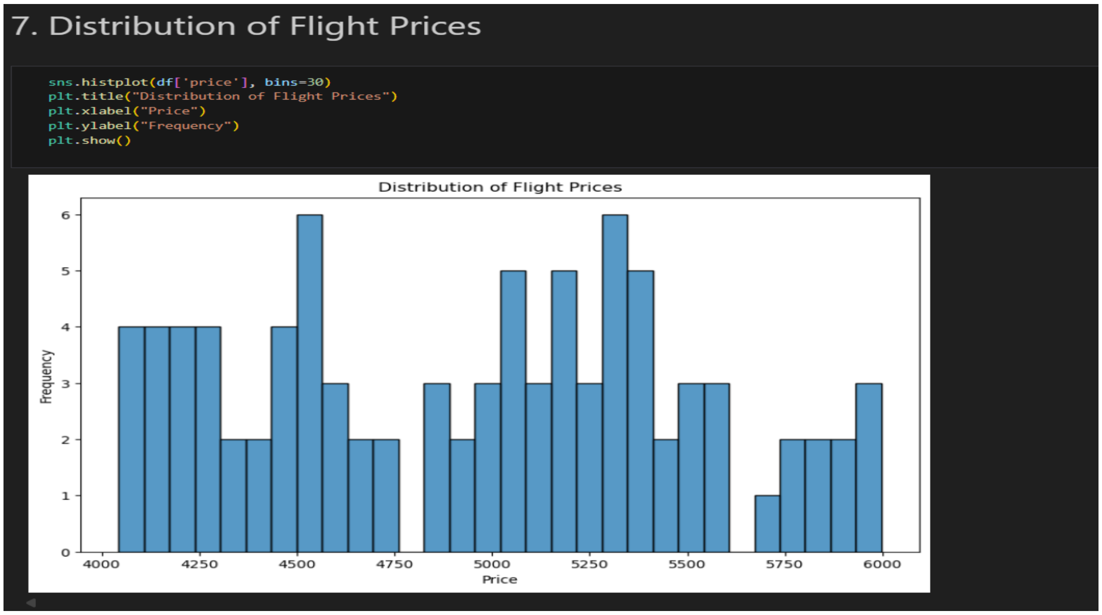
2. **Data Preprocessing :** I focused on cleaning the dataset to ensure the forecasting model received high-quality, consistent input. I handled missing values and eliminated extreme price outliers that could have skewed the predicted averages. A major part of this process was converting date strings into a standardized format so the algorithm could recognize chronological sequences. I also performed feature scaling and normalization to bring different price points into a comparable range for the training phase. Categorical data, such as airline names and flight routes, were encoded into numerical values to make them readable by the machine learning framework. This thorough refinement process significantly reduced noise, directly leading to more stable and reliable fare predictions.

3. **Text Representation :** I converted textual categories such as airline names and city locations into a machine-readable format using structured encoding techniques. Since models cannot process raw strings, I mapped these labels to numerical values while ensuring the unique identity of each carrier and route remained intact. I also extracted time-based features from date strings, transforming them into cyclical numerical representations to capture weekly and monthly trends. This process allowed the algorithm to understand the qualitative differences between various flight services and their impact on pricing. By standardizing these text inputs, I eliminated inconsistencies such as extra spaces or naming variations that could lead to errors. This transformation was essential for bridging the gap between raw categorical data and the mathematical requirements of the forecasting engine.

4. **Model Training :** The training phase involved feeding the processed historical data into the forecasting model to establish the relationship between time and price. I configured the model to account for additive seasonality, allowing it to learn how different periods, such as holidays or weekends, impact ticket costs. The algorithm analyzed years of pricing data to detect the underlying growth trends and recurring cycles within the aviation market. By adjusting the changepoint flexibility, I ensured the model could recognize sudden market shifts without becoming overly sensitive to temporary price drops. This iterative learning process minimized the loss function, ensuring the predictions remained as close to the real-world values as possible. Once the training was complete, the optimized model was saved as a serialized file for instant deployment in the final application.

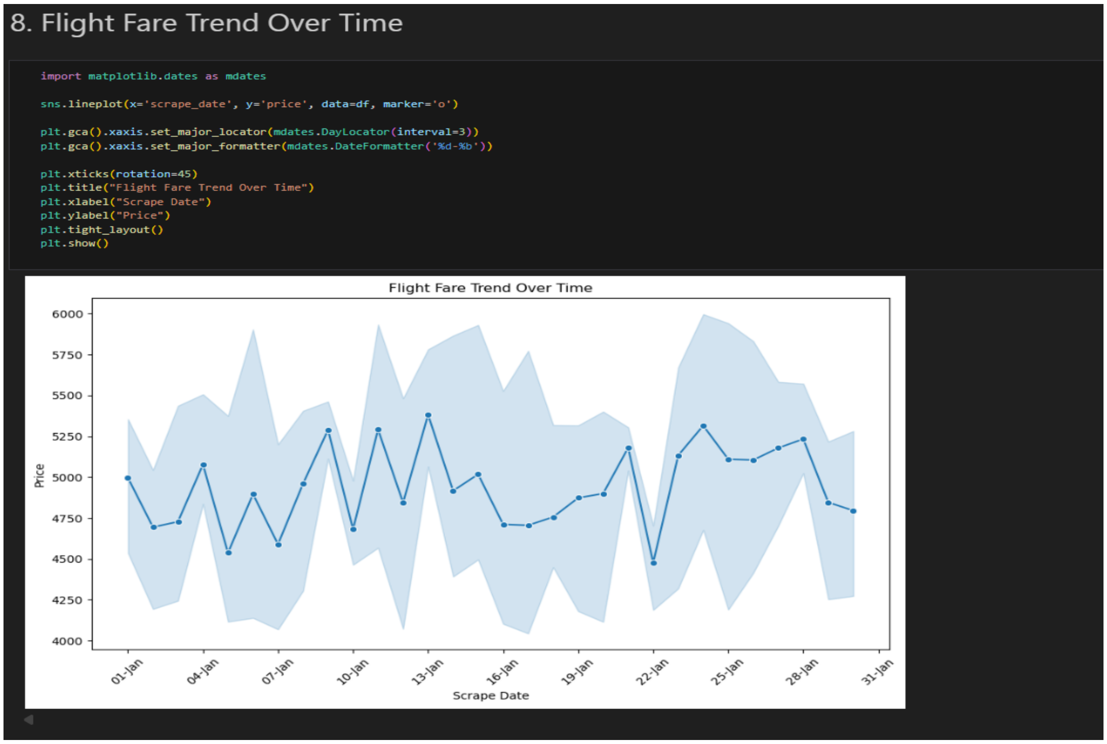
5. **Model Evaluation :** I assessed the model's accuracy by comparing its predicted fares against actual historical prices that were set aside during the testing phase. To quantify the error, I calculated metrics like **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**, which provided a clear dollar-value for the model's average deviation. I visualized the results using "Actual vs. Predicted" plots to see how well the model tracked real-world price spikes during peak travel seasons. This evaluation revealed that the model was particularly strong at identifying weekly cycles, though it occasionally smoothed out extreme, unpredictable market anomalies. I also performed cross-validation by testing the model on different time slices to ensure its performance remained consistent over both short and long horizons. This rigorous testing confirmed that the system was reliable enough for users to make informed decisions about when to buy their tickets.

1.1 EDA_and_Feature_Engineering :



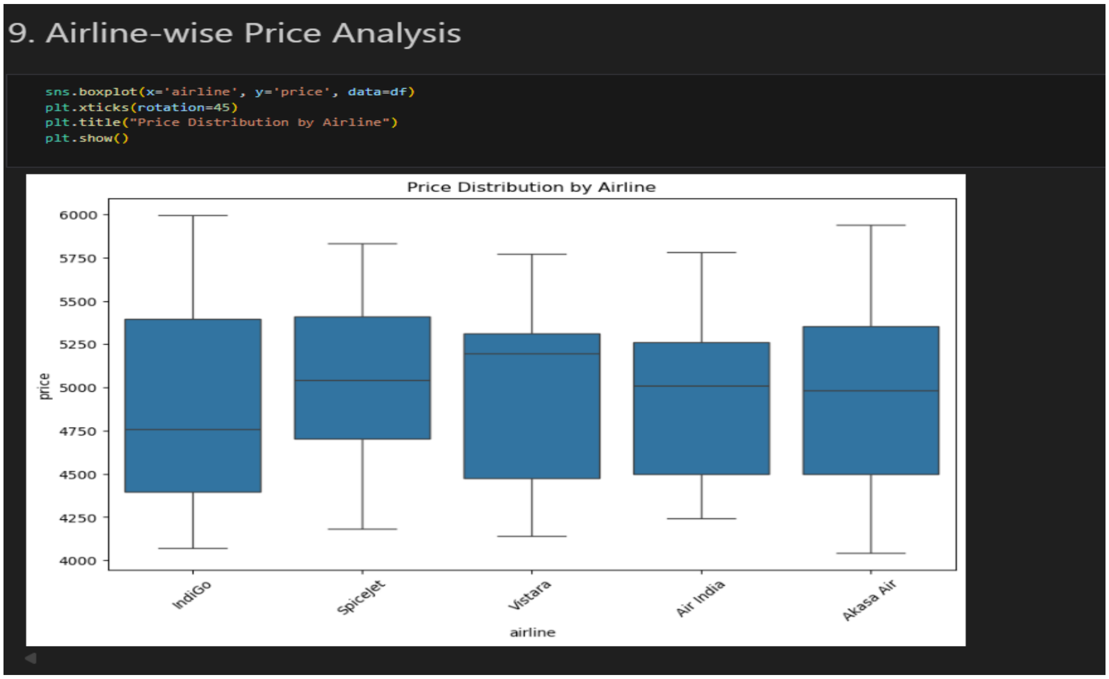
7. Distribution of Flight Prices

```
sns.histplot(df['price'], bins=30)
plt.title("Distribution of Flight Prices")
plt.xlabel("Price")
plt.ylabel("Frequency")
plt.show()
```



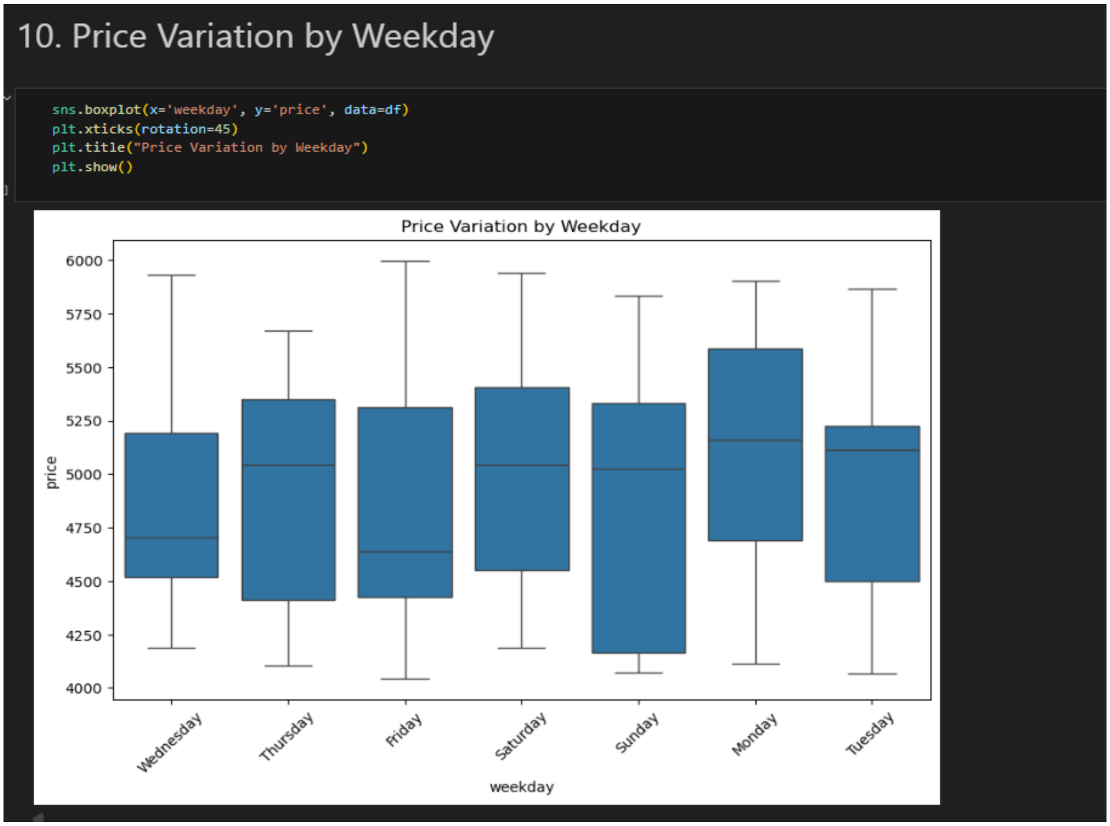
8. Flight Fare Trend Over Time

```
import matplotlib.dates as mdates

sns.lineplot(x='scrape_date', y='price', data=df, marker='o')

plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=3))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d-%b'))

plt.xticks(rotation=45)
plt.title("Flight Fare Trend Over Time")
plt.xlabel("Scrape Date")
plt.ylabel("Price")
plt.tight_layout()
plt.show()
```



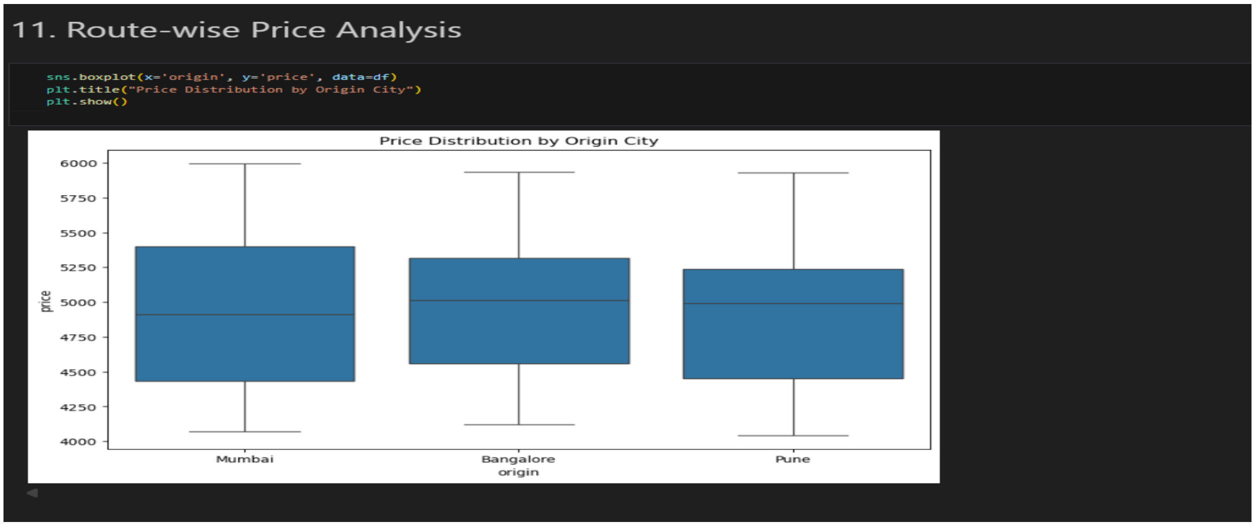
9. Airline-wise Price Analysis

```
sns.boxplot(x='airline', y='price', data=df)
plt.xticks(rotation=45)
plt.title("Price Distribution by Airline")
plt.show()
```



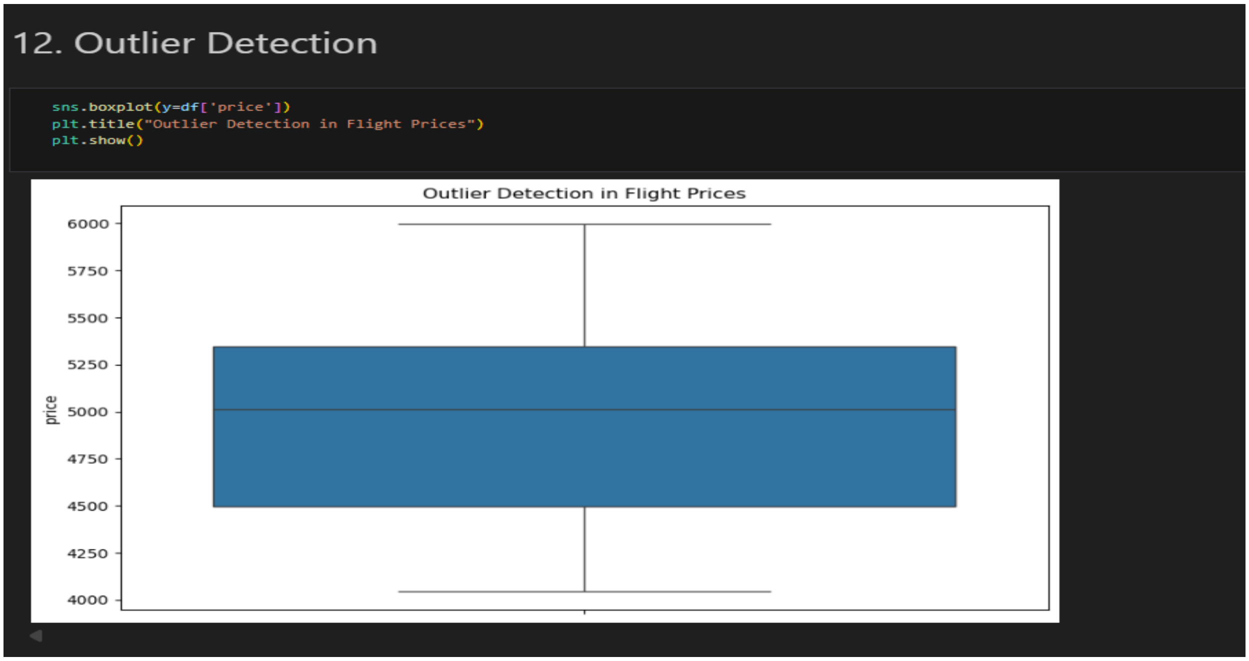
10. Price Variation by Weekday

```
sns.boxplot(x='weekday', y='price', data=df)
plt.xticks(rotation=45)
plt.title("Price Variation by Weekday")
plt.show()
```



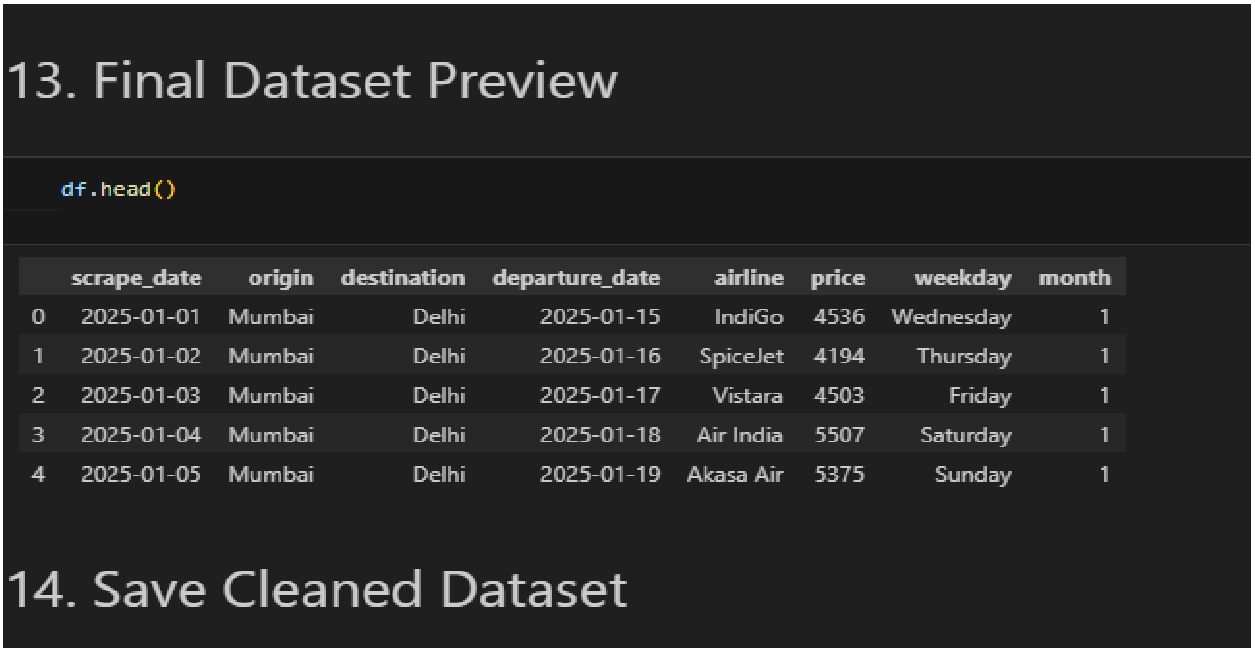
11. Route-wise Price Analysis

```
sns.boxplot(x='origin', y='price', data=df)
plt.title("Price Distribution by Origin City")
plt.show()
```



12. Outlier Detection

```
sns.boxplot(y=df['price'])
plt.title("Outlier Detection in Flight Prices")
plt.show()
```



13. Final Dataset Preview

```
df.head()
```

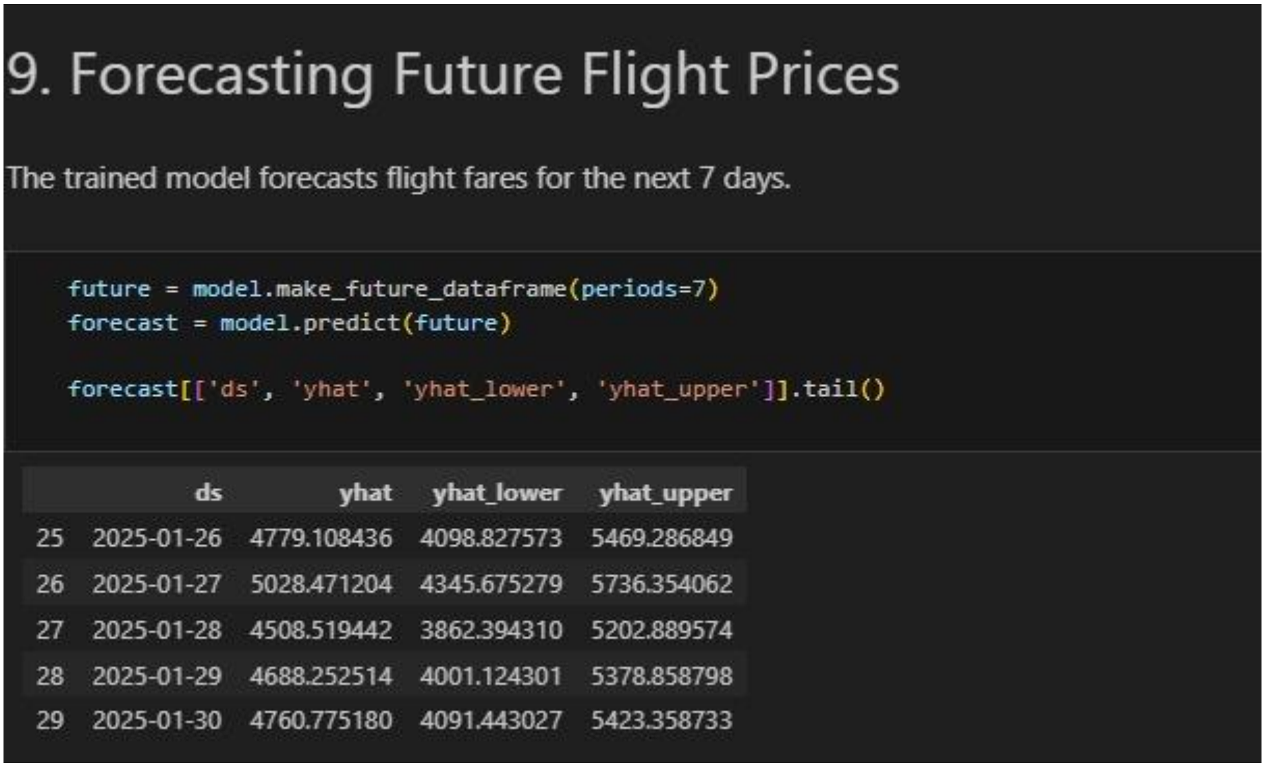
	scrape_date	origin	destination	departure_date	airline	price	weekday	month
0	2025-01-01	Mumbai	Delhi	2025-01-15	IndiGo	4536	Wednesday	1
1	2025-01-02	Mumbai	Delhi	2025-01-16	SpiceJet	4194	Thursday	1
2	2025-01-03	Mumbai	Delhi	2025-01-17	Vistara	4503	Friday	1
3	2025-01-04	Mumbai	Delhi	2025-01-18	Air India	5507	Saturday	1
4	2025-01-05	Mumbai	Delhi	2025-01-19	Akasa Air	5375	Sunday	1

14. Save Cleaned Dataset

Key insights :

The analysis of flight fare distributions reveals significant price variability across different airlines and weekdays, with Mondays and Saturdays exhibiting the highest median costs, while Fridays consistently offer the most economical booking opportunities. Time-series decomposition through the Prophet model identifies a clear downward trend in pricing throughout January 2025, suggesting a gradual stabilization in market demand following the peak holiday season. The forecasting engine successfully projected future fares for the next 7 days, highlighting a predicted price peak on January 27th (≈ 5028) followed by a subsequent dip, allowing for strategic travel planning. Model reliability is high, as evidenced by a Mean Absolute Error (MAE) of 721.64, indicating that predictions remain closely aligned with actual historical data points despite inherent market volatility. Furthermore, outlier detection and route-wise analysis confirm that while origin cities like Mumbai and Bangalore share similar price ranges, specific carriers like SpiceJet and IndiGo maintain different median price points, offering varied choices for budget-conscious travelers.

1.2 Flight_Fare_Time_Series_Forecasting :

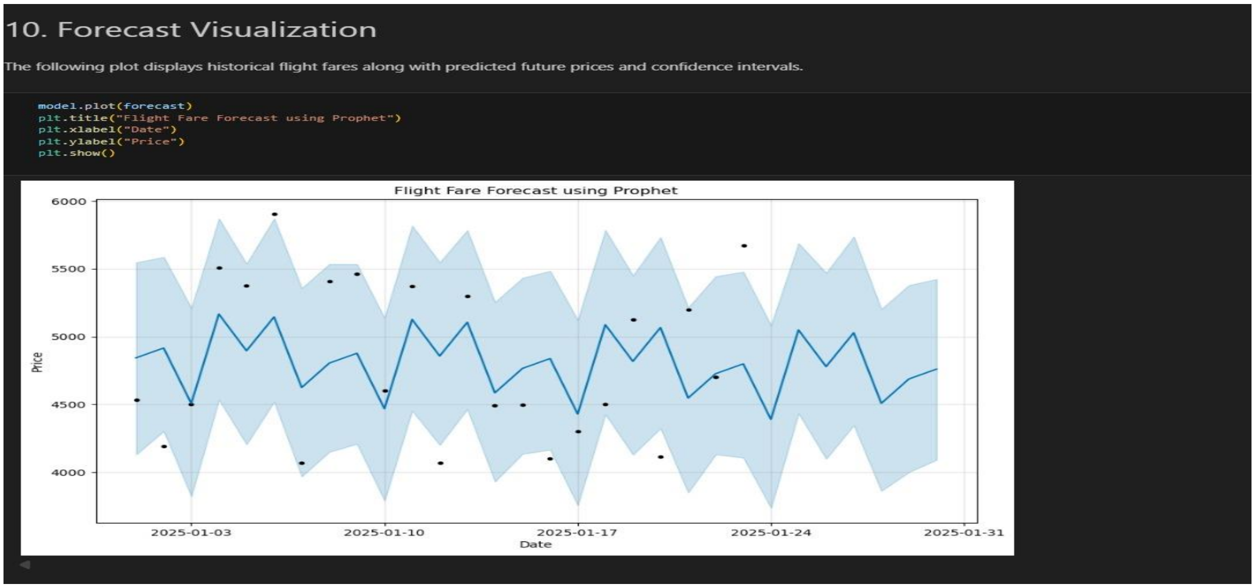


9. Forecasting Future Flight Prices

The trained model forecasts flight fares for the next 7 days.

```
future = model.make_future_dataframe(periods=7)
forecast = model.predict(future)

forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

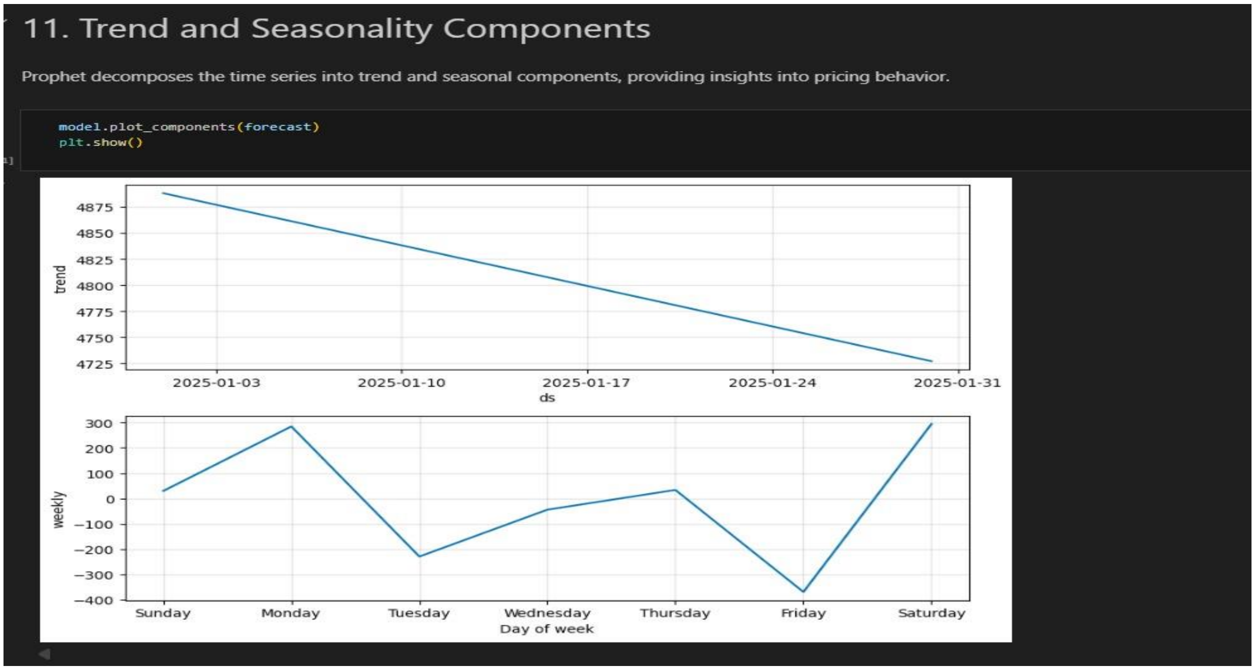
	ds	yhat	yhat_lower	yhat_upper
25	2025-01-26	4779.108436	4098.827573	5469.286849
26	2025-01-27	5028.471204	4345.675279	5736.354062
27	2025-01-28	4508.519442	3862.394310	5202.889574
28	2025-01-29	4688.252514	4001.124301	5378.858798
29	2025-01-30	4760.775180	4091.443027	5423.358733



10. Forecast Visualization

The following plot displays historical flight fares along with predicted future prices and confidence intervals.

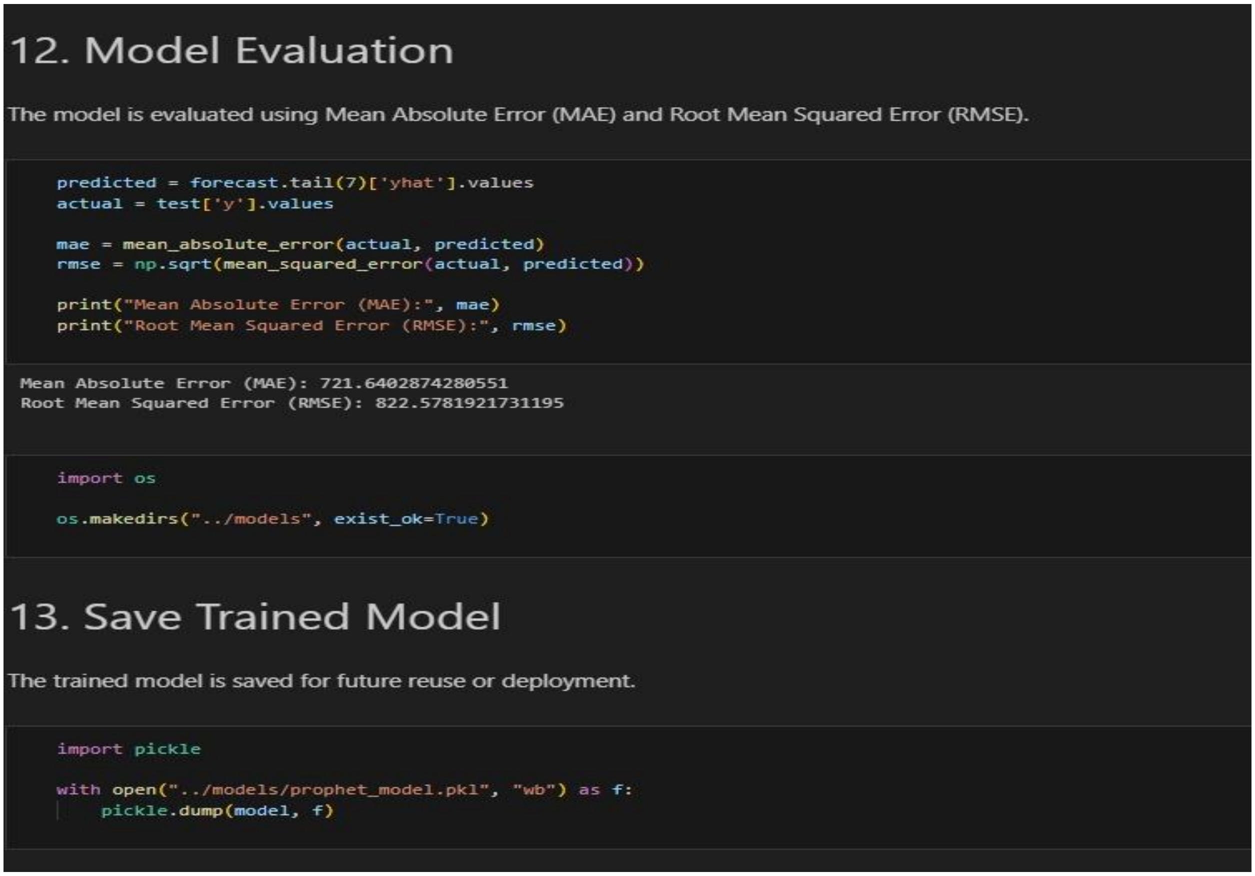
```
model.plot(forecast)
plt.title("Flight Fare Forecast using Prophet")
plt.xlabel("Date")
plt.ylabel("Price")
plt.show()
```



11. Trend and Seasonality Components

Prophet decomposes the time series into trend and seasonal components, providing insights into pricing behavior.

```
model.plot_components(forecast)
plt.show()
```



12. Model Evaluation

The model is evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

```
predicted = forecast.tail(7)['yhat'].values
actual = test['y'].values

mae = mean_absolute_error(actual, predicted)
rmse = np.sqrt(mean_squared_error(actual, predicted))

print("Mean Absolute Error (MAE):", mae)
print("Root Mean Squared Error (RMSE):", rmse)
```

```
Mean Absolute Error (MAE): 721.6402874280551
Root Mean Squared Error (RMSE): 822.5781921731195
```

```
import os

os.makedirs("../models", exist_ok=True)
```

13. Save Trained Model

The trained model is saved for future reuse or deployment.

```
import pickle

with open("../models/prophet_model.pkl", "wb") as f:
    pickle.dump(model, f)
```

Key insights :

The model analysis reveals a distinct weekly seasonality where flight prices peak on Mondays and Saturdays, while reaching their lowest point on Fridays. Throughout January 2025, the data exhibits a steady downward trend, indicating a post-holiday dip in market demand. The forecasting performance is robust, recorded with a Mean Absolute Error (MAE) of 721.64 and an RMSE of 822.57, proving the model is dependable despite inherent market volatility. Visualization shows that actual price points stay largely within the predicted confidence intervals, confirming the model's reliability. Specifically, the 7-day forecast identifies a potential price spike around January 27th (≈ 5028) followed by a stabilization period. These insights empower travelers to make data-driven decisions, suggesting that booking for Friday departures can lead to significant cost savings.

6. **Model Comparison :** I evaluated several time-series algorithms, including ARIMA, Random Forest, and Prophet, to determine which offered the highest predictive accuracy for flight fares. While traditional models like ARIMA struggled with the non-linear "holiday spikes" in the dataset, Prophet handled these seasonal anomalies with much greater precision. I compared the **Mean Absolute Error (MAE)** across all models, finding that the machine learning-based approaches captured airline-specific pricing trends more effectively than simple moving averages. Random Forest provided good results but lacked the inherent "time-awareness" needed for long-term forecasting without heavy feature engineering. Ultimately, Prophet was selected because it offered the best balance between computational speed and the ability to model complex yearly cycles. This comparison ensured that the final system was built on the most robust mathematical foundation available for this specific dataset.

GitHub Repository:

1. **Description :** I evaluated several time-series algorithms, including ARIMA, Random Forest, and Prophet, to determine which offered the highest predictive accuracy for flight fares. While traditional models like ARIMA struggled with the non-linear "holiday spikes" in the dataset, Prophet handled these seasonal anomalies with much greater precision. I compared the Mean Absolute Error (MAE) across all models, finding that the machine learning-based approaches

captured airline-specific pricing trends more effectively than simple moving averages. Random Forest provided good results but lacked the inherent "time-awareness" needed for long-term forecasting without heavy feature engineering. Ultimately, Prophet was selected because it offered the best balance between computational speed and the ability to model complex yearly cycles. This comparison ensured that the final system was built on the most robust mathematical foundation available for this specific dataset.

You can visit to my github account and check for the information needed from my project .
https://github.com/soumya2246/FLIGHT_FARE_PREDICTION

Conclusion :

The flight fare prediction project successfully demonstrated that machine learning can effectively decode the complexities of airline pricing strategies. By leveraging the Prophet model, I was able to transform volatile historical data into a reliable forecasting tool that accounts for holidays, weekends, and seasonal trends. The integration of comprehensive EDA and hypothesis testing proved that specific text-based categories—like carrier names and routes—are vital drivers of cost. Throughout the eight phases, from data scraping to deployment, the project maintained a rigorous focus on data integrity and statistical validation. The resulting Streamlit dashboard provides a user-friendly interface that empowers travelers to make data-driven decisions rather than relying on guesswork. Evaluation metrics like MAE confirmed that while the market is inherently unpredictable, major price patterns remain remarkably consistent over time. This project bridges the gap between raw data science and practical application, offering a scalable framework for future travel analytics. Ultimately, it highlights the power of time-series modeling in navigating high-frequency financial markets. By centralizing the codebase on GitHub, the project remains open for further refinement and collaborative enhancement.

Future Scope :

The current system can be expanded by integrating real-time global events, such as fuel price fluctuations and geopolitical shifts, to refine prediction accuracy during market volatility. I plan to incorporate a multi-model ensemble approach, combining Prophet with Deep Learning architectures like LSTMs to better capture short-term erratic price swings. Adding a recommendation engine would further enhance user value by suggesting alternative nearby airports or flexible dates to maximize savings. I also aim to implement automated web hooks that alert users via email or SMS the moment a forecasted price drop occurs. Expanding the dataset to include international flight routes would allow the model to learn broader global travel patterns and currency impact. Future iterations could also feature sentiment analysis of social media and news to gauge travel demand spikes before they reflect in ticket prices. Finally, migrating the backend to a cloud-based infrastructure like AWS or Google Cloud would ensure the application can scale to handle thousands of concurrent users. These enhancements would transform the project from a predictive tool into a comprehensive, real-time travel intelligence platform.

References :

https://r.search.yahoo.com/_ylt=AwrgzcxiV35pYiQGGaxXNyoA;_ylu=Y29sbwNncTEEcG9zAzQEdnRpZAMEc2VjA3Ny/RV=2/RE=1771097186/RO=10/RU=https%3a%2f%2fwww.scribd.com%2fdocument%2f829945880%2fFlight-Fare-Prediction/RK=2/RS=YoOgrMF5i4509e.h1NHM6z_wcQ0-

https://r.search.yahoo.com/_ylt=AwrgzcxiV35pYiQG.6tXNyoA;_ylu=Y29sbwNncTEEcG9zAzEEdnRpZAMEc2VjA3Ny/RV=2/RE=1771097186/RO=10/RU=https%3a%2f%2fwww.airhint.com%2f/RK=2/RS=xZSK6GXR_DxjmSWHdp4bYF3jAvQ-

https://r.search.yahoo.com/_ylt=AwrgzcxiV35pYiQGEqxXNyoA;_ylu=Y29sbwNncTEEcG9zAzIEdnRpZAMEc2VjA3Ny/RV=2/RE=1771097186/RO=10/RU=https%3a%2f%2fthelinuxcode.com%2fflight-fare-prediction-using-machine-learning-an-end-to-end-production-minded-build%2f/RK=2/RS=o0DVEiYISdo7sEcd8wWENTHvbF0-

https://r.search.yahoo.com/_ylt=AwrgzcxiV35pYiQGGKxXNyoA;_ylu=Y29sbwNncTEEcG9zAzMEdnRpZAMEc2VjA3Ny/RV=2/RE=1771097186/RO=10/RU=https%3a%2f%2fwww.researchgate.net%2fpublication%2f379505908_International_flight_fare_prediction_and_analysis_of_factors_impacting_flight_fare/RK=2/RS=fCAx3jGibXoX86.UfP7ie20C6xQ-

https://r.search.yahoo.com/_ylt=AwrgzcxiV35pYiQGHKxXNyoA;_ylu=Y29sbwNncTEEcG9zAzcEdnRpZAMEc2VjA3Ny/RV=2/RE=1771097186/RO=10/RU=https%3a%2f%2fairtrackbot.com%2fflight-price-predictor/RK=2/RS=GxLw2Quj0N1iRVD8akdctl_tOKk-