AIRLINE PRICE PREDICTION

Introduction

- **Project Overview**

  The Airline Price Prediction project aims to create a model that can predict the prices of airline tickets. This model is very important for travelers as they can use it to decide when to buy tickets to save money. Our model will focus on a few specific routes, mainly between Canada and the home countries of our classmates. This way, they can check ticket prices and find the best times to travel back home and return to Canada.

- **Aim**

  The main aim of the project is to use past airline ticket data to build a machine learning model that can predict future ticket prices.

- **Objectives**

  - To gather flight price data from various sources such as Skyscanner, Kayak, and Kaggle.
  - To clean and preprocess the data for accurate and effective analysis.
  - To engineer relevant features that will enhance the performance of predictive models.
  - To build and evaluate predictive models and deploy them for user accessibility.
  - To create an interactive dashboard for effective data visualization and user interaction.

  We will analyze data such as past ticket prices, flight dates, number of stops, and other relevant factors to train our model. This will help travelers plan their trips at the best times to save money.

Our data will include features such as:

Date of Journey: The date the flight is scheduled.
Booking date: The date the flight is booked.
Airline: The Airline operating the flight
Flight code: Unique code for each flight
Class: Travel class (e.g. economy class, first class, premium class)
Source and destination: The departure and arrival cities
Total Stops: The number of stops during the journey.
Days left: The numbers of days left until the flight at the time of booking
Fare: The price of the ticket (Target variable)

PROJECT MANAGEMENT

1. **Assigning Team Roles:**

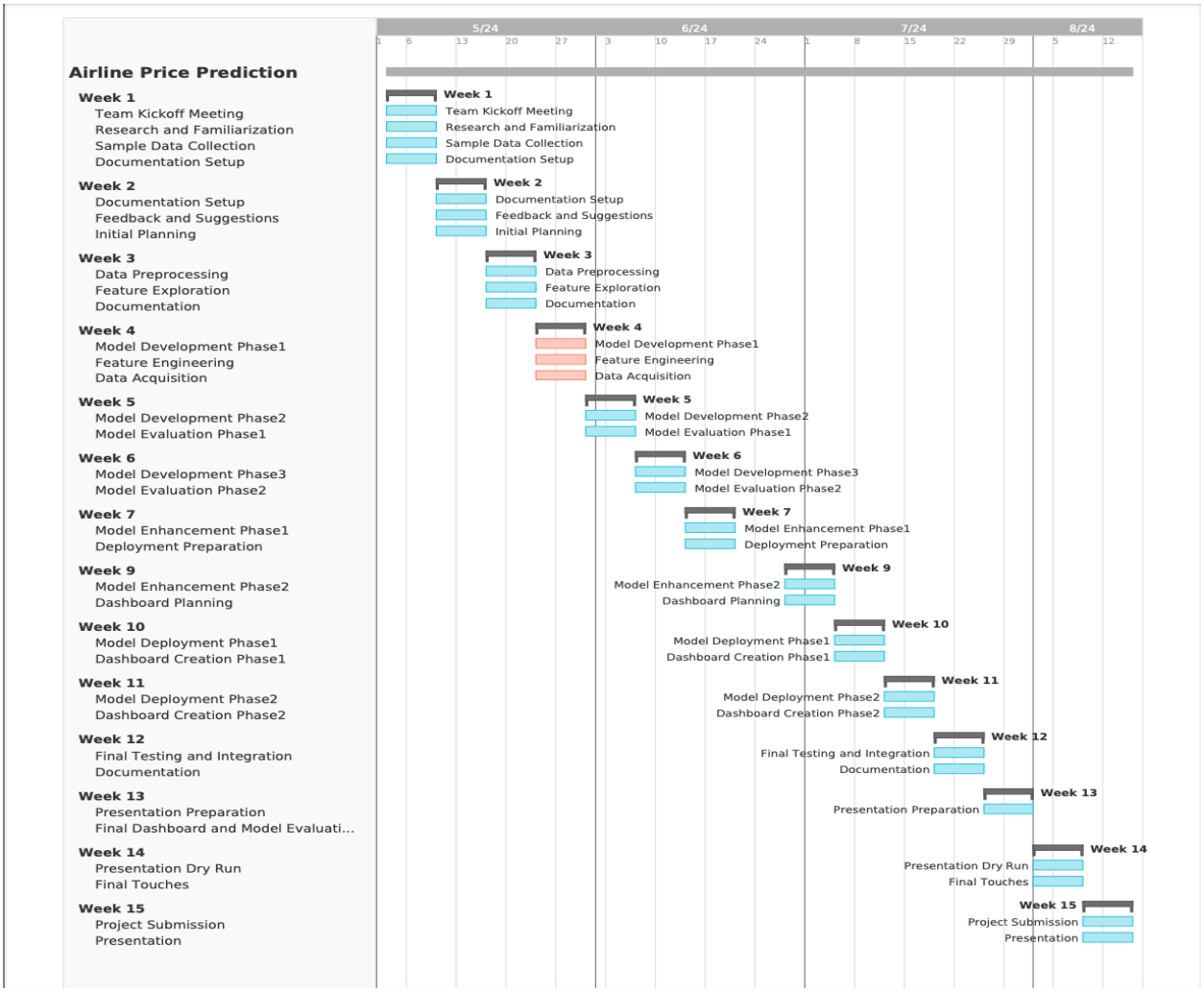
	A	B	C
1	**Roles**	**Name**	**Email**
2	Data Acquisition	Mohamed Maaz Rehan	mohamedmaazrehan@loyalistcollege.com
3	Data Preprocessing	Comfort Iroha Onuoha	comfortirohaonuoh@loyalistcollege.com
4	Feature Engineering	Devendra Singh Shekhawat	devendrasinghshek@loyalistcollege.com
5	Model Development	Gaurav Singh Swetha Tanikonda	gauravsingh3@loyalistcollege.com swethatanikonda@loyalistcollege.com
6	Model Evaluation	Jankiba Viralsinh Zala	jankibaviralsinhz@loyalistcollege.com
7	Data Warehousing	Gaurav Singh Rawat	gauravsinghrawat@loyalistcollege.com
8	Deployment	Fatemi Sadikbhai Lokhandwala Urjeet Parmar	fatemisadikbhailo@loyalistcollege.com urjeetparmar@loyalistcollege.com
9	Dashboard Creation	Tirth Patel Isha Savaliya	tirthpatel@loyalistcollege.com ishasavaliya@loyalistcollege.com

2. Reason for choosing this Project:

We considered several project ideas, such as Food Waste Reduction, Natural Disaster Prediction and Response, Building a Book Recommendation System, and many more. After conducting a brainstorming session and weighing the pros and cons of each idea, we ultimately decided to go with the Airline Price Prediction project. This decision was driven by the potential usefulness of the project for our classmates, who can benefit from accurate predictions of ticket prices for trips between Canada and their home countries.

3. Deciding on Timetable:

To ensure a structured and efficient workflow, we created a detailed Gantt chart that outlines the timetable for our project. This Gantt chart helps us visualize the timeline of various tasks and milestones, ensuring that we stay on track and meet our deadlines.



4. **Platforms Utilized in the Project:**

The following platforms are utilized for various tasks within the project:

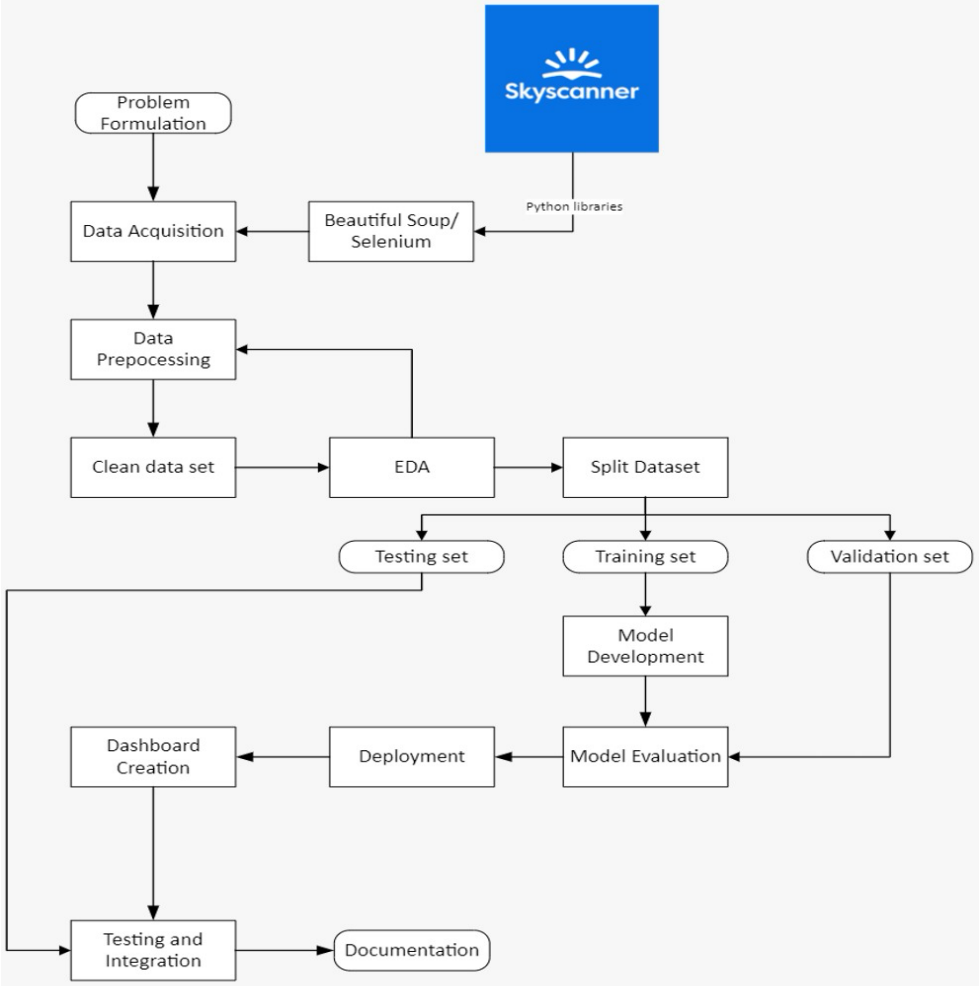
- Scraping, Data Cleaning, Model Building, and Evaluation: Python
- Model Deployment: Streamlit, AWS
- Data Storage: Google Cloud Platform
- Dashboard Building: Tableau or PowerBI

5. Data Features

- **Date and Time -** Departure date, booking date, day of the week, time of the day.
- **Flight Details -** Airline, flight duration, layovers, class (economy, business, etc.).
- **Route Information -** Source airport, destination airport, distance.
- **Price -** The target variable represents the cost of the flight.

6. Action Plan Development:

The following diagram illustrates the overall flow of our airline price prediction project. It captures each critical stage, from data acquisition to the final dashboard creation and documentation. This visual representation provides a clear and concise understanding of how data is processed and utilized throughout the project, showcasing the systematic approach we have adopted to develop an accurate and functional predictive model.



Our plan of action encompasses several crucial stages: data acquisition, preprocessing, feature engineering, model development, evaluation, deployment, and dashboard creation.

Initially, we collected sample data from various websites, including historical airline ticket prices, flight routes, dates, and other relevant factors. In the preprocessing stage, we clean and process this data to handle missing values and ensure consistency for reliable analysis.

For feature engineering, we identify and create key features to help our model understand factors affecting ticket prices, such as seasonality, demand, and external events. During model development, team members explore and test various machine learning models, like linear regression and decision trees, to find the most accurate predictors, using metrics like R-squared, MAE, and RMSE for evaluation.

Simultaneously, we are learning new tools and techniques to deploy the model for real-time use by airlines and travelers, setting up the necessary infrastructure for efficient operation. We are also creating an interactive dashboard to display predicted ticket prices and other useful information, aiding users in exploring price trends and making informed travel decisions.

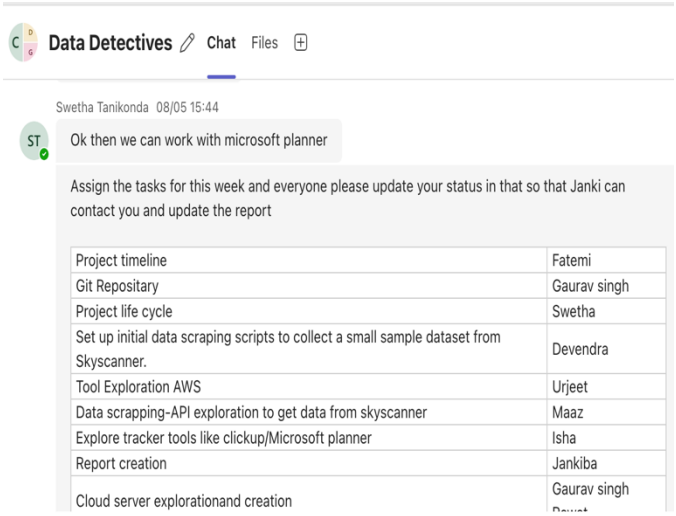
Each team member is actively contributing to these steps, ensuring a collaborative approach to building an accurate and effective airline price prediction model.

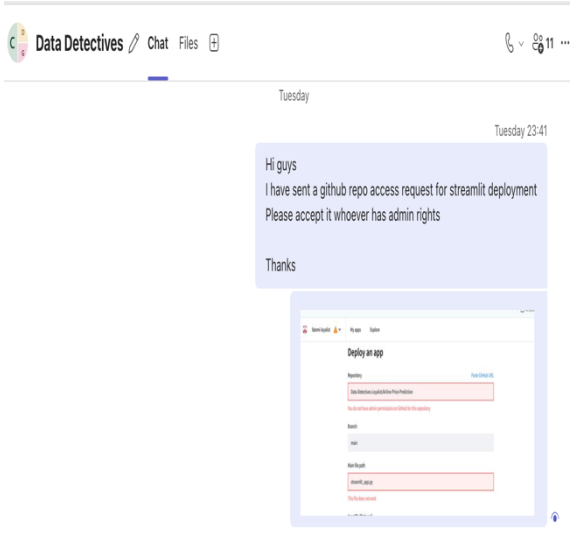
7. Project Discussion and Meeting Venues:

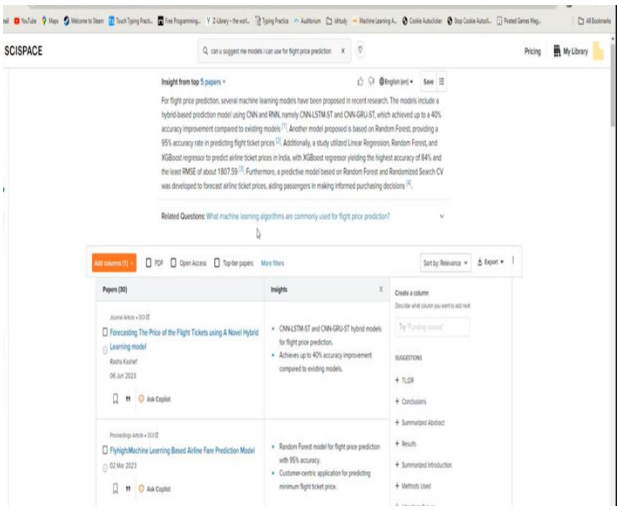
We discuss our project on Microsoft Teams, where we have created a dedicated project group. Every week, our tasks are allocated by our team leader, and we work on our respective parts. When any of us faces challenges in achieving our goals, we discuss them in the group channel, and everyone collaborates to resolve the issues or code errors. We also regularly conduct team meetings to check progress, discuss new ideas, and address any challenges faced by team members. Additionally, we record these meetings for future reference.

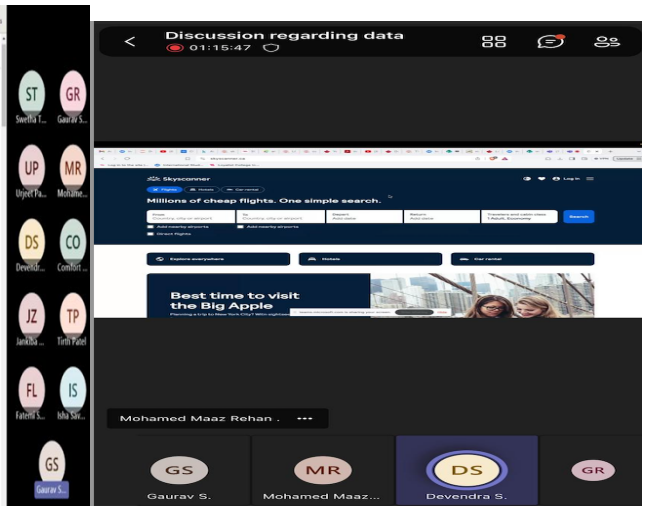
8. Proof of discussion:

Here are the screenshots of our Teams chat and Teams meeting, documenting our project discussions and decision-making processes.









SOLUTION FLOW

1. Data Set Sources:

Our team explored various websites to gather historical data, including Asana, Grepsr, Datarade, and Data.world, aiming to collect as many relevant features as possible. After evaluating these sources, we decided to use a sample dataset from Kaggle due to its comprehensiveness and ease of access. Additionally, we are working on fetching the latest data from Skyscanner using Selenium to ensure our model remains up-to-date and accurate.

2. Deciding on the type of analysis:

Our project employs several analytical techniques to ensure a robust airline price prediction model. We began with univariate analysis to understand the distribution and patterns of individual features like ticket prices and dates. Next, we performed bivariate analysis to explore relationships between pairs of features, such as how ticket prices vary with travel dates and routes.

We then focused on building our machine learning model, testing various algorithms including linear regression and decision trees. By evaluating these models using metrics like R-squared, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE), we identified the best algorithm for accurate price prediction. This comprehensive approach ensures a deep understanding of the data and optimal model performance.

3. Picking the output format

For the output format of our airline price prediction project, we focused on model deployment and dashboard creation to ensure the results are accessible and user-friendly. We deployed the model using Streamlit and AWS, which allows real-time predictions to be easily accessed by users.

Additionally, we created an interactive dashboard using tools like Tableau or PowerBI. This dashboard displays predicted ticket prices, historical trends, and other key insights in a visually appealing and easy-to-understand format. By integrating these tools, we provide a comprehensive platform where users can explore data trends and make informed travel decisions based on the model's predictions.

DATA MANAGEMENT

Our data management process entails gathering, storing, processing, and analyzing data to derive actionable insights. One of the key methods we use for data collection is web scraping, and we have chosen Selenium as our primary tool due to its robust capabilities in handling dynamic web content. This section outlines our approach, challenges, and best practices in using Selenium for web scraping.

Data Collection: In the data collection phase, we will collect relevant data from online sources to support our project’s analytical and modeling needs. We have decided to use Selenium to automate the extraction of data from websites that dynamically load content for accurate data collection.

Why are we using Selenium? Selenium is a powerful web automation tool that allows us to interact with web pages in a manner like a human user. It is particularly effective for scraping data from websites that utilize JavaScript to load content dynamically, which is a limitation for many traditional scraping tools. We will use selenium to open our target website Kayak and navigate through pages as needed.

Data Storage: We will convert the extracted data into Data Frames using pandas and store them in a database or CSV files for further processing.

Data Processing: We will clean and preprocess the data to ensure it is in a usable format for analysis. This includes handling missing values, correcting data types, and normalizing data.

Data Cleaning

- **Missing Values**: Addressed missing values using mean/mode imputation where appropriate, or discarded rows with significant missing data.
- **Duplicates**: Removed duplicate entries to maintain data integrity.
- **Outliers**: Identified and treated outliers using z-score analysis and interquartile range (IQR) methods to avoid skewed results.

Data Warehouse

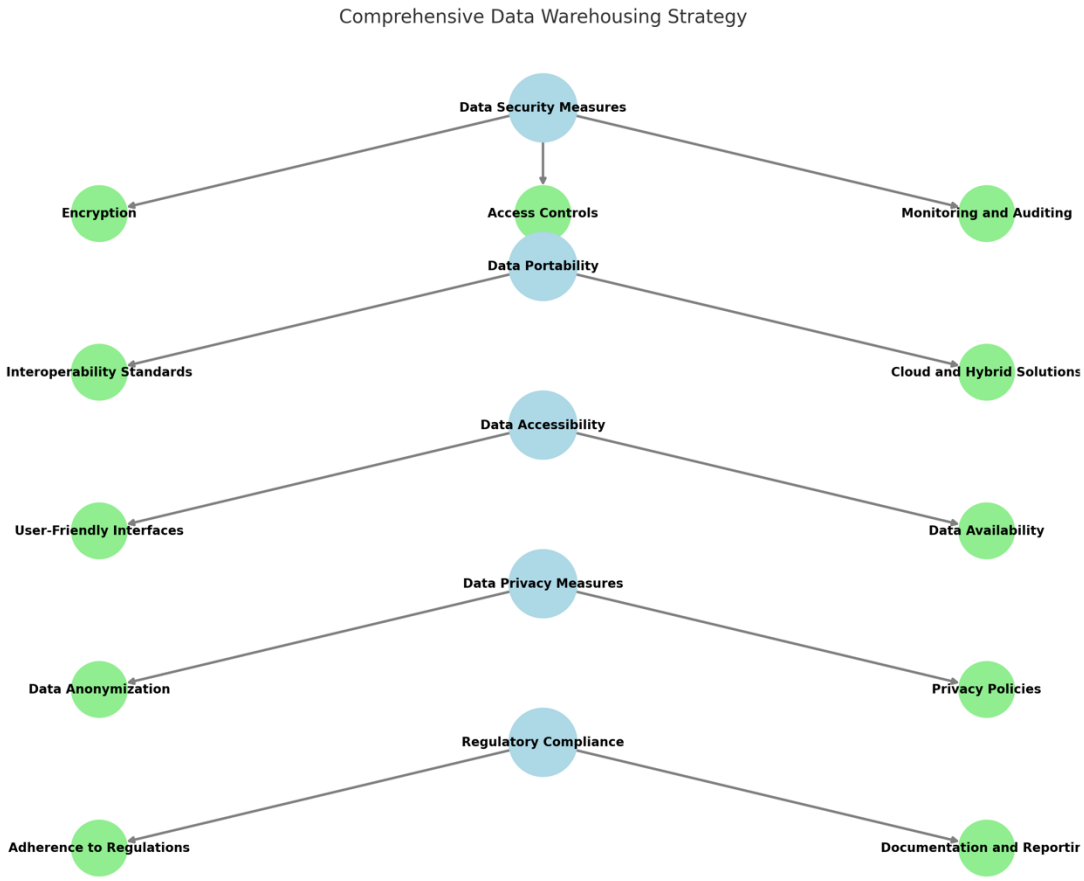
Security: Our data warehouse security approach will be to make sure we limit who has access to the data, making sure only the right people have access to it.

Portability: We will create a cloud-based data warehouse which will allow access to our data from anywhere.

Accessibility: We will use easy-to-use interfaces which will make it easy to use and very accessible to every team member.

Privacy: We will handle the data we scrap carefully and restrict any access to sensitive data.

Compliance: In building our data warehouse we will keep detailed records of everything we do and regularly check to make sure we're on track with what the law requires.



Data Quality Considerations: For data quality we will check for data completeness, if we have missing data, we will fill it in to maintain our data quality.

We will check for consistency of our dataset e.g: the Date_of_journey and journey_day columns will be verified to ensure that the day of the week matches the corresponding date.

For example: The departure and arrival times are categorized (e.g., "After 6 PM," "Before 6 AM"), we will ensure that these categories are used consistently throughout the dataset.

For accuracy, we will verify that the Duration_in_hours values correctly reflect the actual flight times between the Source and Destination.

The fare values will be examined for outliers in the preprocessing stage.

Since our project is a real time prediction model, we will ensure the data is scrapped in a timely manner to ensure accurate data collection.

We will ensure that the Flight code values are valid and correspond to existing flights.

DATA VALIDATION:

Our Data Validation procedure will be to regularly validate the data to ensure the scrapped data does not have duplicates, missing values and that we maintain a consistent datatype. The data ingestion pipeline will be designed to handle large volumes of data and scale with increasing data load.

DATA INGESTION

We will implement real-time or near-real-time data ingestion processes to ensure that the data warehouse is updated promptly with the latest information. During the data ingestion process, all necessary data transformation will be done to standardize and format the data to the needed datatype.

Data Analysis:

Our dataset contains flight information with the following columns:

Date of the journey, Day of the week of the journey, Airline name, Flight_code, Class, Source, Departure, Total_stops, Arrival, Destination, Duration_in_hours, Days_left, Fare.

Mathematics Used:

We will use Statistical Analysis like descriptive statistics and correlation analysis.

Descriptive Statistics: We will calculate the mean, median, standard deviation, and percentiles to summarize data distributions. This will help us to summarize our dataset, identify patterns, and make decisions.

Correlation Analysis: We will use the Pearson correlation coefficient to identify linear relationships between features. This will help us identify the strength of linear relationships between our features and help in our feature selection, data interpretation, and model building phase.

EVALUATION METRICS:

Our Evaluation metrics will be Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2). Each metric will provide us with valuable insights into different aspects of our model's predictive accuracy and effectiveness. Here's why we have chosen these specific metrics:

Why Mean Absolute Error (MAE)?

MAE measures the average magnitude of the errors in our price predictions, without considering their direction. This metric will help us understand how much, on average, our predicted prices deviate from the actual prices. MAE is easy to understand and interpret. It gives us a straightforward measure of the average prediction error in the same units as the data (currency). Since MAE treats all errors equally, it provides a clear picture of the average error magnitude.

In the context of airflight price prediction, knowing the average error magnitude is crucial for decision-making.

Why Root Mean Squared Error (RMSE)?

RMSE measures the square root of the average squared differences between predicted and actual prices. This metric emphasizes larger errors, making it useful for identifying models that minimize significant deviations. RMSE gives more weight to larger errors, making it a useful metric when we want to penalize larger discrepancies more heavily.

By taking the square root of the average squared errors, RMSE provides an error metric in the same units as the data, but with an emphasis on larger deviations.

In the airline industry, large price prediction errors can lead to significant financial implications. RMSE helps us ensure that our model is not just accurate on average, but also avoids large prediction errors that could disrupt pricing strategies and revenue management.

Why R-squared (R^2)?

R^2 indicates the proportion of the variance in airflight prices that is explained by our predictive model. This metric helps us understand how well our independent variables explain the variability in airflight prices. R^2 provides a measure of how well our model captures the underlying trends and patterns in the data. It allows us to compare different models and choose the one that best explains the variation in airflight prices.

For airflight price prediction, understanding the proportion of price variability explained by the model is crucial. A high R^2 value indicates that our model effectively captures the factors influencing airflight prices, leading to more reliable predictions.

Model Selection:

In our model selection process, we will consider an approach of using a simple model e.g. linear regression as a baseline model and use this to compare the effectiveness of more complex models.

Linear Regression: We will use linear regression to establish baseline performance metrics. This model will serve as a benchmark to compare the effectiveness of more complex algorithms.

Our Advanced Models:

Random Forests: To leverage the power of ensemble learning by averaging multiple decision trees.

Gradient Boosting Machines (GBM): To iteratively improve model performance by correcting the errors of previous models.

Hyperparameter Tuning: To optimize model performance, we will employ hyperparameter tuning techniques such as Grid Search and Random Search. These techniques

systematically explore a predefined space of hyperparameters to identify the combination that yields the best model performance.

Cross-Validation: To ensure our models generalize well to unseen data, we will use k-fold cross-validation. This technique involves dividing the dataset into k subsets, training the model on k-1 subsets, and validating it on the remaining subset. The process is repeated k times, with each subset serving as the validation set once. This approach provides a robust estimate of model performance.

ANALYSIS

In EDA, we will visualize based on the below:

Fare Distribution: The distribution of fares using histograms and box plots.

Fare by Airline: Analyze fare variations across different airlines using box plots and summary statistics.

Fare by Class: Investigate how fares differ across travel classes (Economy, Business, etc.).

Fare by Source and Destination: Examine fare patterns between different city pairs using heatmaps.

Correlation Analysis: Identify relationships between features and fares using correlation matrices.

Preliminary Results & Discussion

Airlines and Fares: Significant variation in fares across different airlines, with premium airlines having higher average fares.

Class and Fares: Higher travel classes (Business, First) have significantly higher fares compared to Economy.

Source-Destination Pairs: Certain routes have higher fares, likely due to demand and competition.

Preliminary Conclusions

Prediction Accuracy: Advanced models like Gradient Boosting provide better accuracy in fare prediction compared to simpler models.

Feature Importance: Key features influencing fare predictions include the airline, travel class, source-destination pair, and days left until the flight.

Practical Implications: The model can help travelers identify the best times to purchase tickets and select cost-effective routes, leading to significant savings.

Implementation and Deployment Plan

1. **Model Validation**: Finalize the model using the best-performing algorithm and validate its performance on a separate validation set.

2. **Web Application**: Develop a web-based application using Flask or Django to allow users to input flight details and get price predictions.

3. **API Integration**: Create an API endpoint for the prediction model to enable integration with other services.

4. **Monitoring and Maintenance**: Set up monitoring to track the model's performance in production and regularly update it with new data to maintain accuracy.

Recommendations

- **Travelers**: Use the model to identify the best times to book flights and save on travel costs.

- **Airlines**: Implement the model for dynamic pricing strategies to optimize revenue.

- **Future Enhancements**: Integrate real-time data and explore more advanced machine learning techniques like deep learning for improved predictions.

Next Steps

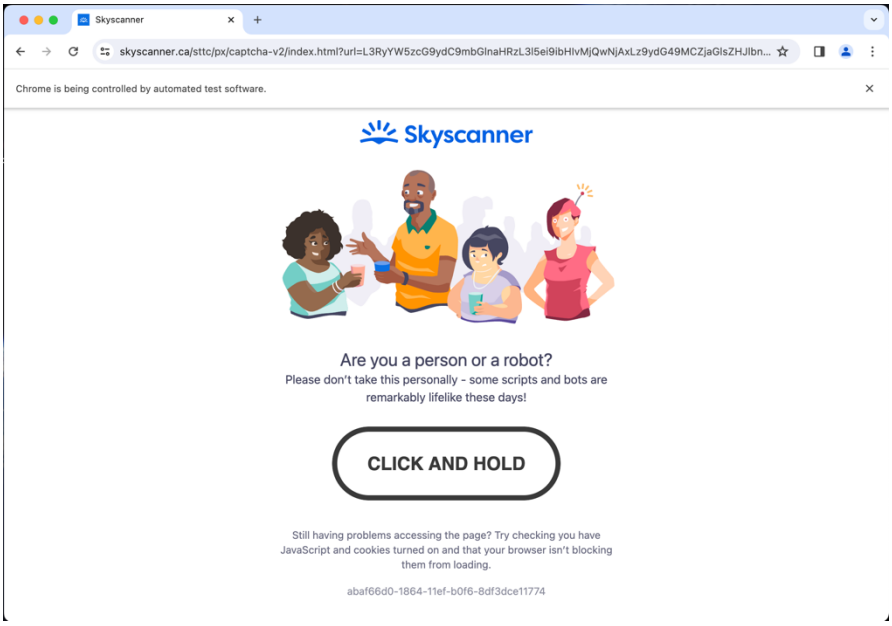
1. **Hyperparameter Tuning**: Perform grid search and cross-validation to fine-tune model parameters for optimal performance.
2. **Advanced Models**: Train and evaluate Gradient Boosting Machines (GBM) and Neural Networks to improve predictive accuracy.
3. **Feature Importance**: Analyze feature importance to refine the model by identifying the most influential features.
4. **Model Evaluation**: Conduct comprehensive evaluation using additional metrics such as RMSE and MAE on the test set.
5. **Integration**: Develop a user-friendly interface for travelers and airlines to use the prediction model.

FAILING

- Attempt with Skyscanner

Initially, we attempted to scrape flight data from the Skyscanner website. However, after hitting two to three URLs, we encountered a message asking, "Are you a person or robot?" This indicates that Skyscanner has robust anti-bot measures in place, making direct web scraping impractical.

```
In [8]: driver = webdriver.Chrome()
        #driver.implicitly_wait(3)
        driver.get("https://www.skyscanner.ca/transport/flights/yyz/blr/240601/?adultsv2=1&cabinclass=economy&childrenv2=&in

        try:
            elem = WebDriverWait(driver, 30).until(
                EC.text_to_be_present_in_element((By.XPATH, "//div[@class='SummaryInfo_itineraryCountContainer__NmJmM']"), '
            )
        finally:
            print("exit")
            #driver.quit()

        exit

        ---------------------------------------------------------------------------
        TimeoutException                          Traceback (most recent call last)
        Cell In[8], line 6
              3 driver.get("https://www.skyscanner.ca/transport/flights/yyz/blr/240601/?adultsv2=1&cabinclass=economy&child
        renv2=&inboundaltsenabled=false&outboundaltsenabled=false&preferdirects=false&ref=home&rtn=0")
              5 try:
        ----> 6     elem = WebDriverWait(driver, 30).until(
              7         EC.text_to_be_present_in_element((By.XPATH, "//div[@class='SummaryInfo_itineraryCountContainer__NmJ
        mM']"), 'results') # This is a dummy element
              8     )
              9 finally:
             10     print("exit")

        File ~/anaconda3/lib/python3.11/site-packages/selenium/webdriver/support/wait.py:105, in WebDriverWait.until(self,
        method, message)
            103     if time.monotonic() > end_time:
            104         break
        --> 105 raise TimeoutException(message, screen, stacktrace)

        TimeoutException: Message:
        Stacktrace:
        0   chromedriver                        0x0000000102f1e940 chromedriver + 4368704
        1   chromedriver                        0x0000000102f16dd4 chromedriver + 4337108
        2   chromedriver                        0x0000000102b3ac04 chromedriver + 289796
        3   chromedriver                        0x0000000102b7ce00 chromedriver + 560640
        4   chromedriver                        0x0000000102bb55ec chromedriver + 792044
        5   chromedriver                        0x0000000102b71ab4 chromedriver + 514740
        6   chromedriver                        0x0000000102b7250c chromedriver + 517388
```



- Exploring Skyscanner API

  Next, we explored the possibility of using Skyscanner's API for fetching the required data. we created a Python script to retrieve data using the API provided by RapidAPI (reference: RapidAPI Skyscanner API). Unfortunately, the API did not return the desired results, prompting us to look for alternative solutions.

- Issues with AWS Account Creation

  Another issue we faced was the inability to create a free AWS account. This posed a significant challenge as AWS is a crucial platform for deploying our model. We are currently exploring other cloud service providers or seeking assistance to resolve the AWS account creation problem.

RESOURCES

Software Platforms:

- Python for data scraping, cleaning, and model building
- Essential Python libraries: Pandas, NumPy, Scikit-learn, Matplotlib
- Streamlit and AWS for model deployment
- Tableau or PowerBI for dashboard creation

Hardware Requirements:

- Minimum of 16GB RAM
- Multi-core processor (Intel i5 or equivalent) for efficient data processing and model training tasks.