SMART BRIDGE PROJECT REPORT

Forecast Commuters Inflow for Airline Industry using Prophet Model

TEAM MEMBERS - REGISTRATION NUMBER

G.V. Sai Rakesh - 20BCD7049

M. Jahnavi - 20BCD7044

G. Vishwa - 20BCI7188

G. Dhanush - 20BCI7148

FORECAST COMMUTERS INFLOW FOR THE AIRLINE INDUSTRY USING PROPHET MODEL

1. INTRODUCTION

1.1 OVERVIEW

The airline industry plays a crucial role in the global economy, connecting people and facilitating trade across continents. Accurate forecasting of passenger demand is vital for airlines to optimize operations and plan for the future. Forecasting commuter inflow is particularly important as it helps airlines anticipate travel patterns, adjust capacity, optimize pricing strategies, and enhance overall efficiency.

This project report focuses on utilizing the Prophet model, a robust forecasting tool developed by Facebook's Core Data Science team, to predict commuters' inflow for the airline industry. The Prophet model has gained popularity for its ability to handle various time series data, incorporating trend, seasonality, and holiday effects.

1.2 PURPOSE

The primary objectives of this project are as follows:

a) To develop a reliable forecasting model to predict commuter inflow for the airline industry.
b) To analyze and understand the underlying patterns, trends, and seasonality in commuter data.
c) To provide insights into future demand patterns, enabling airlines to make informed decisions regarding capacity planning, marketing strategies, and revenue management.

This project enables airlines to achieve several benefits using the forecasting of commuters' inflow. It allows for improved capacity planning, optimizing resource allocation, and enhancing revenue management. By accurately predicting future demand patterns, airlines can implement effective marketing strategies, allocate resources efficiently, reduce costs, and gain a competitive advantage in the industry. Additionally, the project aids in minimizing risks associated with underutilization or overstaffing, optimizing flight schedules, and enhancing overall customer satisfaction. Overall, this forecasting project empowers airlines to make informed decisions, improve operational efficiency, and maximize profitability in the dynamic and competitive airline industry.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Existing approaches to solving the problem of forecasting commuters' inflow in the airline industry include traditional statistical methods and machine learning techniques. However, these approaches have their own limitations and problems:

1. Traditional Statistical Methods:
    - Moving Average (MA) and Exponential Smoothing (ES): These methods are simple and easy to implement but may not capture complex patterns and seasonality in the data. They often struggle with handling non-linear trends and abrupt changes in commuter behavior.
    - Autoregressive Integrated Moving Average (ARIMA): ARIMA models are widely used but assume linear relationships and stationary data. They may not capture non-linear trends or long-term dependencies in the commuter inflow data.
2. Machine Learning Techniques:
    - Regression Models: Linear regression or polynomial regression can be applied to predict commuter inflow. However, these models may not handle seasonality and non-linear relationships effectively.
    - Random Forests or Gradient Boosting: These ensemble methods can capture complex relationships but may not handle seasonality and may overfit the data if not tuned properly.
    - Long Short-Term Memory (LSTM): LSTM models can capture long-term dependencies in the data but may require significant computational resources and extensive data pre-processing.

Problems with existing approaches:

1. Lack of Seasonality Handling: Many traditional statistical methods and simple machine learning techniques struggle to effectively capture and incorporate seasonality in the commuter inflow data, leading to inaccurate predictions.
2. Inability to Handle Non-linear Trends: Linear models and traditional statistical methods assume linear relationships, which may not accurately represent the complex patterns and non-linear trends in commuter data.
3. Computational Complexity: Some machine learning techniques, such as LSTM, require extensive computational resources and longer training times, making them less feasible for real-time forecasting applications.
4. Lack of Interpretability: Some machine learning models, like random forests or deep learning models, are considered black-box models, providing limited insights into the underlying factors driving commuter inflow.
5. Sensitivity to Hyperparameters: Some machine learning techniques, such as random forests or gradient boosting, require careful tuning of hyperparameters to achieve optimal performance, which can be time-consuming and challenging.

To overcome these problems, the Prophet model, used in this project, addresses the limitations by explicitly handling seasonality, accommodating non-linear trends, providing interpretability, and offering a balance between accuracy and computational efficiency.

2.2 PROPOSED SOLUTION

The method or solution suggested by this project is to utilize the Prophet model for forecasting commuters' inflow in the airline industry. The Prophet model, developed by Facebook's Core Data Science team, is a powerful forecasting tool specifically designed to handle time series data with multiple factors, including trend, seasonality, and holiday effects.

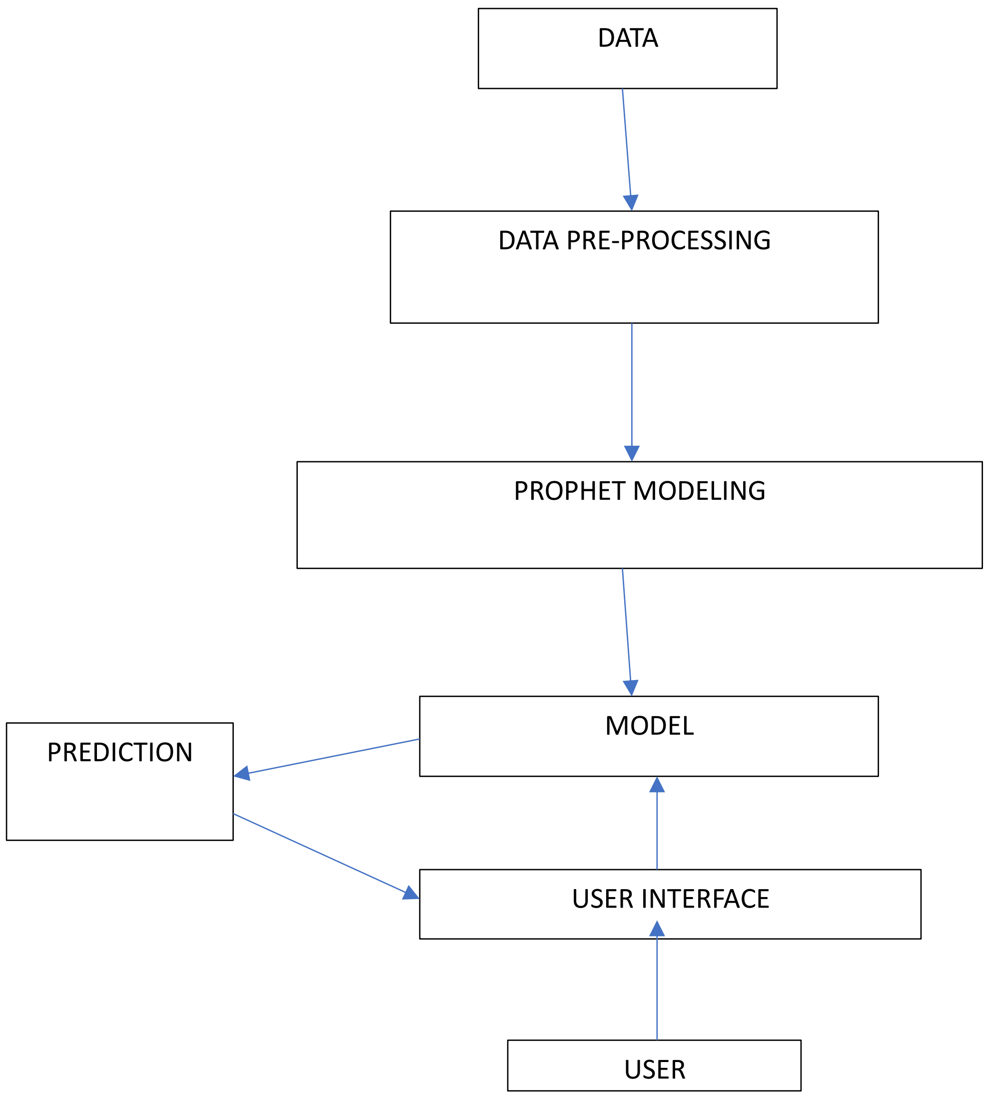
The Prophet model incorporates several innovative features:

1. Flexible Trend Modelling: The Prophet model can capture various types of trends, including non-linear trends, by utilizing flexible modelling techniques that adapt to the data.
2. Seasonality Handling: The model can effectively handle seasonality in the commuter inflow data by incorporating seasonal components and accounting for recurring patterns, such as daily, weekly, and yearly variations.

The approach aims to provide accurate predictions of commuter inflow, enable capacity planning, enhance revenue management, and optimize resource allocation in the airline industry.

3. THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE / SOFTWARE DESIGNING

Hardware Requirements:

1. Computer or Server: A reliable computer or server with sufficient processing power and memory to handle the data processing and modeling tasks efficiently.
2. Storage Space: Adequate storage space to store the historical data, intermediate files, and the forecasted results.
3. Internet Connectivity: A stable internet connection to access necessary data sources, libraries, and resources.

Software Requirements:

1. Operating System: Any popular operating system such as Windows, macOS, or Linux that supports the required software tools.
2. Python: The project requires Python programming language (preferably version 3.6 or higher) to run the necessary scripts and code.
3. Jupyter Notebook or Integrated Development Environment (IDE): An environment for running Python code, such as Jupyter Notebook, PyCharm, or Anaconda.
4. Prophet Library: Install the Prophet library developed by Facebook's Core Data Science team, which provides the necessary functions and capabilities for time series forecasting using the Prophet model.
5. Data Analysis and Visualization Libraries: Libraries like Pandas, NumPy, and Matplotlib or Seaborn for data preprocessing, exploratory data analysis, and visualizing the results.
6. Additional Libraries: Depending on specific project requirements, you may need additional libraries such as scikit-learn for machine learning tasks, statsmodels for statistical modeling, and any other libraries to handle data manipulation or specific analysis needs.
7. Data Storage: Depending on the project's needs, data storage solutions like SQL databases or cloud-based platforms (e.g., AWS S3, Google Cloud Storage) can be used to store and access the dataset.
8. Reporting Tools: Depending on the requirements, reporting tools such as Microsoft Excel, Google Sheets, or specialized data visualization libraries like Plotly can be used to present insights, recommendations, and forecasted results in a visually appealing and understandable manner.

4. EXPERIMENTAL INVESTIGATIONS

The analysis and investigation process involves various steps:

1. Data Quality Assessment: Before proceeding with the analysis, it is crucial to assess the quality and reliability of the data. This includes checking for missing values, outliers, and inconsistencies. Data cleansing techniques are applied to handle such issues, ensuring the dataset is suitable for analysis.

2. Exploratory Data Analysis (EDA): EDA is performed to gain insights into the characteristics of the commuter inflow data. Key components of EDA include data visualization, statistical analysis, and trend identification. This analysis helps identify patterns, seasonality, and any other underlying factors influencing the commuter inflow.

3. Trend and Seasonality Analysis: The EDA phase focuses on identifying and understanding the trends and seasonality patterns present in the data. This involves visualizing the data over time, decomposing the time series into trend, seasonality, and residual components, and analyzing the statistical significance of these components. Understanding the trends and seasonality helps in selecting appropriate modeling techniques.

4. Model Selection and Training: Based on the analysis conducted during the EDA phase, the Prophet model is chosen as the forecasting technique. The Prophet model is trained using historical commuter inflow data, and its hyperparameters are optimized to achieve the best performance. The training phase includes splitting the dataset into training and validation sets to assess the model's accuracy and adjust hyperparameters if necessary.

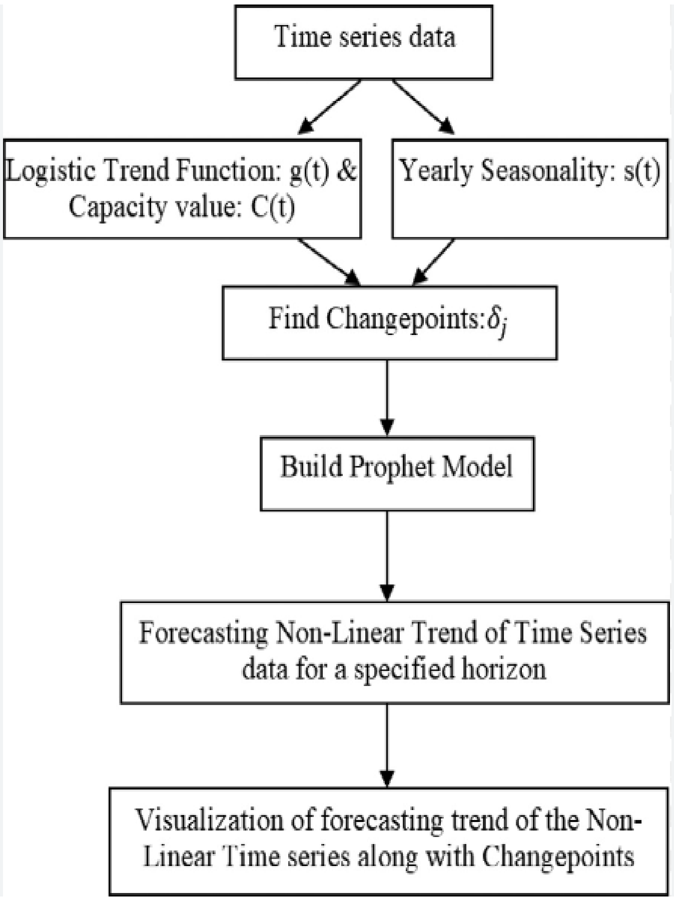
5. Forecasting and Evaluation: Once the Prophet model is trained, it is used to generate forecasts for future commuter inflow. The forecasted results are then evaluated by comparing them with the actual data. Metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and percentage errors are calculated to assess the model's accuracy and performance.

6. Sensitivity Analysis: Sensitivity analysis involves testing the model's performance under different scenarios and parameter variations. This analysis helps identify the model's robustness and provides insights into its limitations and potential sources of error.

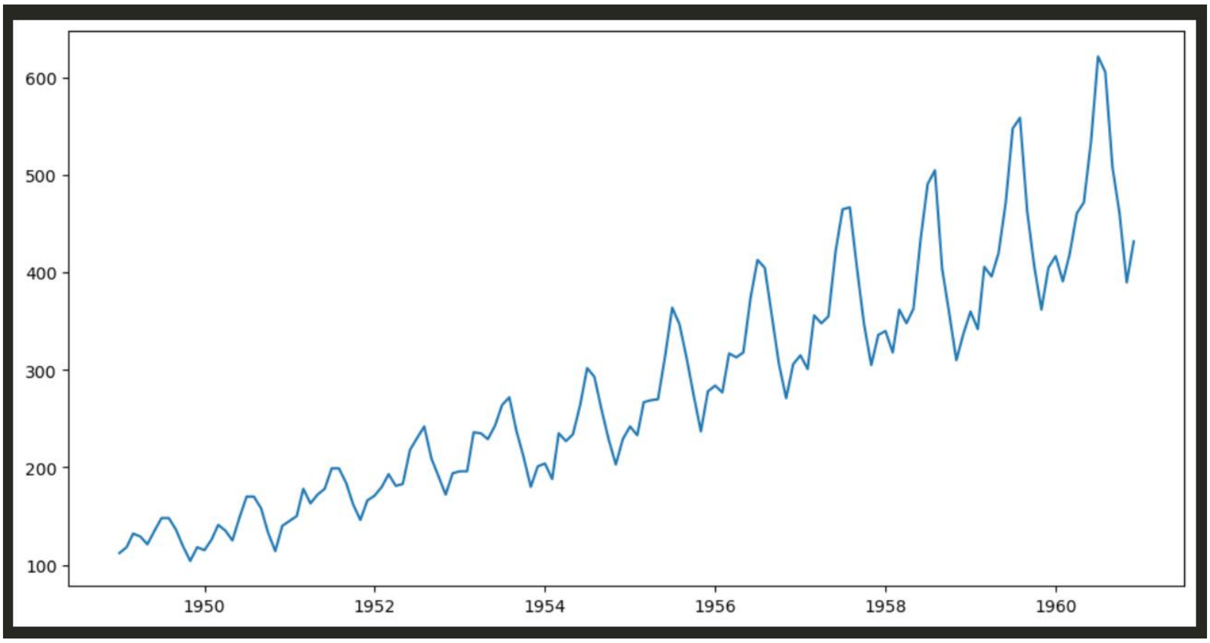
7. Interpretation of Results: The forecasted results are interpreted to derive meaningful insights and recommendations for the airline industry. The analysis may involve identifying factors influencing commuter inflow, understanding the impact of holidays and events, and providing actionable recommendations for capacity planning, revenue management, and marketing strategies.

By conducting these analyses and investigations, the project aims to provide accurate forecasts and valuable insights into commuter inflow, assisting airlines in making informed decisions and optimizing their operations in the dynamic and competitive airline industry.

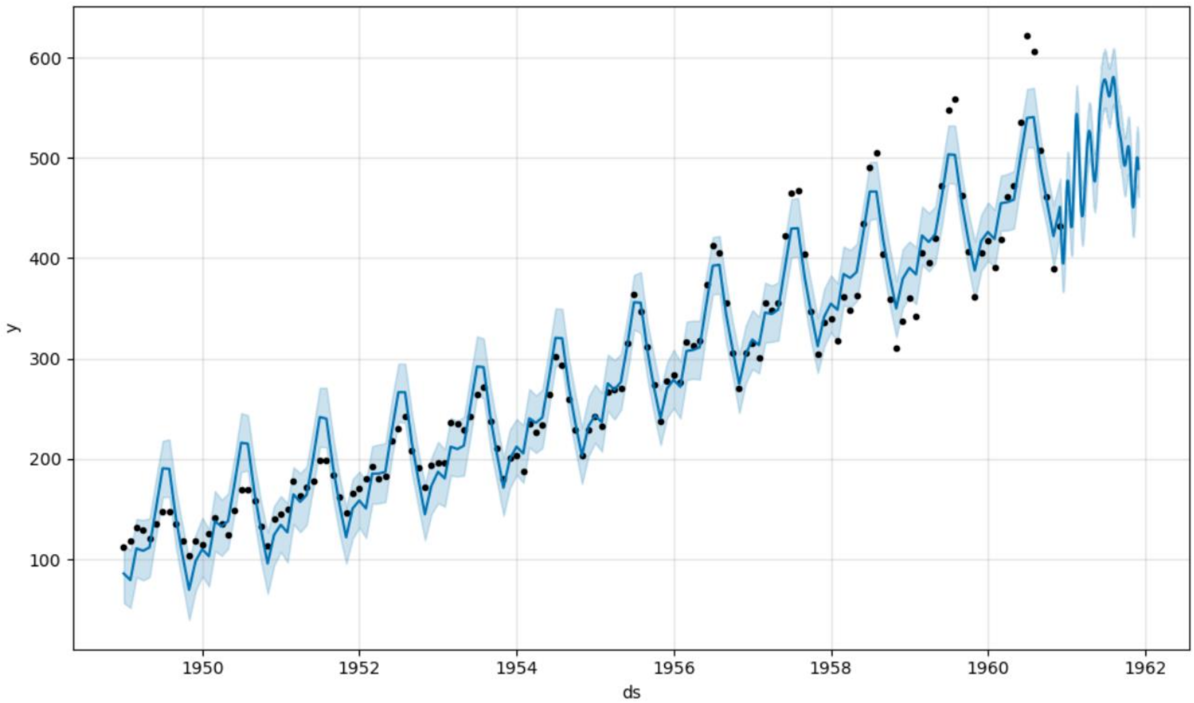
5. FLOWCHART



6. RESULTS



Plotting the values in the Data Set



Forecasted values using the Prophet model

7. ADVANTAGES AND DISADVANTAGES

Advantages of the Proposed Solution:

1. Accurate Forecasting: The Prophet model, utilized in the proposed solution, is specifically designed for time series forecasting. It incorporates trend, seasonality, and holiday effects, leading to accurate predictions of commuter inflow in the airline industry.
2. Handling Non-linear Trends: The Prophet model can effectively capture non-linear trends, allowing it to adapt to various patterns and fluctuations in the data. This enables better forecasting of commuter inflow, even in the presence of complex and changing trends.
3. Seasonality Handling: The model incorporates seasonality components, making it suitable for capturing recurring patterns in commuter demand, such as daily, weekly, and yearly variations. This helps in accurate forecasting during different periods of the year.
4. Robustness to Outliers and Missing Data: The Prophet model is robust to outliers and missing data, as it employs techniques to handle these issues gracefully. This enhances the reliability of the forecasts and reduces the impact of irregularities in the dataset.
5. Interpretable Results: The Prophet model provides interpretable results, allowing stakeholders to understand the underlying factors driving commuter inflow. This facilitates decision-making, as insights into trends, seasonality, and holiday effects can be derived from the model's output.

Disadvantages of the Proposed Solution:

1. Sensitivity to Model Parameters: Like any forecasting model, the performance of the Prophet model can be sensitive to its parameters. Selecting appropriate hyperparameters and ensuring proper tuning are necessary to achieve optimal results.
2. Data Requirements: The Prophet model typically requires a sufficient amount of historical data to make accurate predictions. Limited historical data may affect the model's performance and result in less reliable forecasts.
3. Computational Resources: While the Prophet model is computationally efficient compared to some complex machine learning models, it may still require substantial computational resources, especially when working with large datasets or conducting extensive parameter tuning.

8. APPLICATIONS

1. Demand Forecasting: Accurate forecasting of commuter inflow helps airlines estimate the demand for flights on different routes and plan their operations accordingly. It enables effective capacity planning, resource allocation, and scheduling to optimize flight frequencies, seat availability, and crew assignments.
2. Revenue Management: By accurately predicting commuter inflow, airlines can implement effective revenue management strategies. This includes optimizing pricing strategies, seat inventory management, and promotional offers to maximize revenue generation based on anticipated demand.
3. Operational Planning: Forecasting commuter inflow assists airlines in optimizing operational planning. It helps in resource allocation, such as aircraft deployment, crew scheduling, and ground handling, based on projected demand, ensuring efficient utilization of resources and minimizing operational costs.
4. Marketing Campaigns: Accurate demand forecasting enables airlines to design targeted marketing campaigns. By understanding the anticipated commuter inflow, airlines can tailor their marketing efforts to specific routes, customer segments, and periods of high demand, resulting in better campaign effectiveness and customer engagement.

9. CONCLUSION

In conclusion, the proposed solution of forecasting commuters' inflow for the airline industry using the Prophet model offers several advantages and applications. By leveraging the model's capabilities in capturing trends, seasonality, and holiday effects, airlines can make accurate predictions for capacity planning, revenue management, operational optimization, and marketing campaigns. While considering its limitations and data requirements, this solution provides valuable insights to enhance decision-making, improve resource allocation, and drive competitiveness in the dynamic airline industry.

10. FUTURE SCOPE

Here are some potential avenues for further development and expansion:

1. Integration of External Factors: Enhancing the forecasting model by incorporating external factors such as economic indicators, weather conditions, social events, and travel restrictions can provide a more comprehensive understanding of commuter inflow. This integration can lead to improved accuracy and robustness of the forecasts.
2. Advanced Modeling Techniques: Exploring and incorporating advanced forecasting techniques, such as machine learning algorithms, deep learning models, or hybrid approaches, can further enhance the accuracy and predictive capabilities of the solution. These techniques can capture complex patterns and non-linear relationships in the data, potentially improving forecast accuracy.
3. Real-time Forecasting: Developing real-time forecasting capabilities can enable airlines to adapt to dynamic changes in commuter inflow. By continuously updating the model with new data and leveraging real-time information, airlines can make proactive decisions and respond effectively to evolving market conditions.
4. Demand Sensing and Personalization: Integrating customer data, historical booking patterns, and behavioral analytics can enable demand sensing and personalized forecasting. Airlines can leverage this information to tailor their offerings, personalize marketing campaigns, and optimize pricing strategies based on individual customer preferences and demand patterns.
5. Collaboration and Data Sharing: Collaboration between airlines, airports, and industry stakeholders to share data and insights can lead to more accurate and holistic forecasting. By pooling data resources and utilizing collective intelligence, the industry can achieve a higher level of accuracy and generate comprehensive forecasts that consider broader market dynamics.

11. BIBILOGRAPHY

References:

https://www.kaggle.com/code/kriyeneekutbay/airline-passengers-time-series-forecasting

https://www.kaggle.com/code/sumitkant/time-series-with-keras

APPENDIX

Flask Code:

```
'''
    Import the NumPy library for numerical computations
    Import the Pandas library for data manipulation and analysis
    Import Flask for web application development
    Import the Pickle module for object serialization
'''

import numpy as np
import pandas as pd
from flask import Flask, render_template, request, jsonify
import pickle

# Create an instance of the Flask class

app = Flask(__name__)

# Load a pre-trained model using Pickle

model = pd.read_pickle('./Flask/air_passengers.pkl')

# Render the 'home.html' template when the root URL is accessed

@app.route('/')
def home():
    return render_template('home.html')

''' Check if the HTTP request method is POST
    Get the value of the 'Date' form field
    Create a dictionary with 'ds' as the key and 'dates' as the value
    Create a DataFrame from the dictionary
    Make predictions using the pre-trained model
```

```
    Print the predictions
    Extract the predicted value from the DataFrame
    Print the predicted value
    Render the 'home.html' template with the predicted value
    Render the 'home.html' template if the HTTP request method is not POST
'''

@app.route('/predict', methods=['POST'])
def y_predict():
    if request.method == "POST":
        dates = request.form["Date"]
        name = {"ds": [dates]}
        retain = dates
        dates = pd.DataFrame(name)
        prediction = model.predict(dates)
        print(prediction)
        output = round(prediction.iloc[0, 15])
        print(output)
        return render_template('home.html',
                               prediction_text="Commuters Inflow on {} is
{}.".format(retain,output))
    return render_template('home.html')

# Run the Flask application in debug mode

if __name__ == "__main__":
    app.run(debug=True)
```

Other codes available at: **Github**