Flight Forecast

Q flight delay

Bruno
Fiona
Sushma

Background

Flight delays are a constant source of frustration for travelers, causing missed connections and disrupted schedules. Unpredictable weather and increasing air traffic make reliable information about potential delays even more critical.

In this project, our goal was to create a tool that leverages historical flight delay data and weather information to make informed predictions. By analyzing past trends and integrating with weather forecast APIs, BTS Flight Forecast assesses the likelihood of delays for a specific airport, date, and time.

Data Sources

- **Flight Delays:** Historical flight delay information is obtained from the Bureau of Transportation Statistics (BTS) https://www.transtats.bts.gov/Tables.asp?DB_ID=111.
- **Weather Data**: The tool incorporates weather data using WeatherWunder APIs. These return the weather corresponding to an airport code.
  - Historical Weather: https://api.weather.com/v1/location/K**SEA**:9:US/observations/historical.json
  - Real-time Weather Forecasts: https://api.weather.com/v3/wx/forecast/hourly/15day

Use cases

Use Case 1: Check Flight Delay Prediction

Travelers can enter an airport code, date, and time to see the predicted likelihood of a flight delay for that specific departure.

Use Case 2: Update Model with New Data (Admin Only)

Authorized users can upload new flight data to retrain the prediction model, ensuring its accuracy reflects the latest trends.

Components

- **Graphical User Interface:**

  This component provides two interfaces - one for travelers to easily select their flight details (date, time, airport) to see delay predictions, and another for authorized users to upload historical flight data and trigger model retraining.

- **Weather Utility:**

  This component fetches both historical and forecasted weather data relevant to a user's chosen airport and timeframe. It uses APIs to retrieve historical data for a specific period and forecasted data for a specific airport and time.

Components

- **Data Processor:**

  This component acts as a data cleaner and organizer. It takes historical flight data and weather data, cleanses any inconsistencies, and combines them into a usable format for training the predictive model.

- **Delay Predictor:**

  This component is the core of the application. It uses the processed data to train a machine learning model. Once trained, the model can predict the likelihood of flight delays based on the user's input (date, time, airport).

Component Diagram
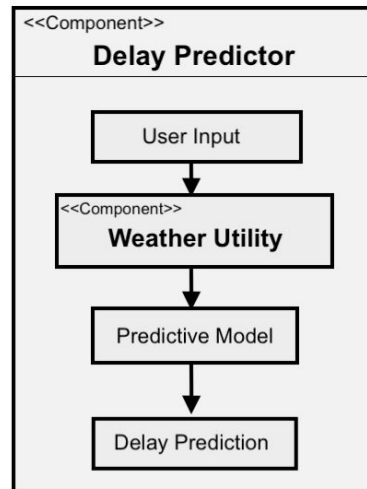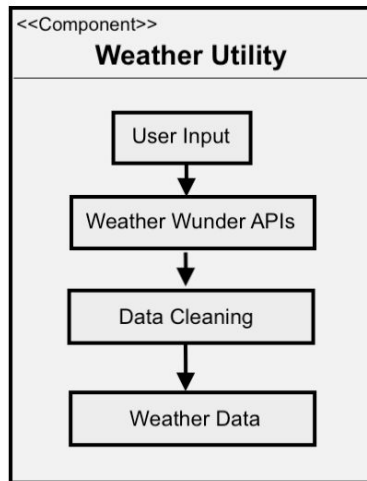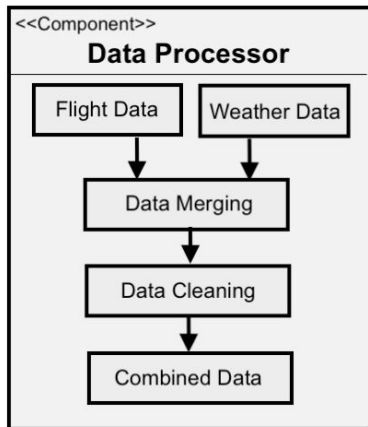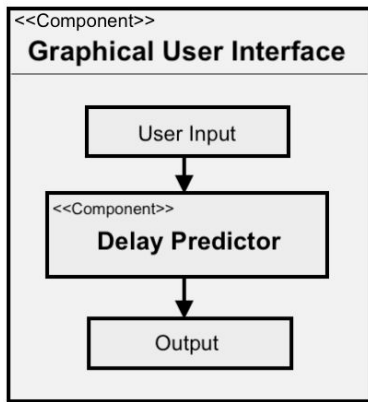
Lessons Learnt

- **Version control is key:** Track changes, collaborate effectively, and fix issues with Git.
- **Teamwork:** Collaborate with teammates and manage issues for smooth data science projects.
- **Virtual environments:** Isolate project dependencies and avoid conflicts with virtual environments.
- **Testing:** Rigorous testing throughout development ensures robust data science projects. It can take waaaaaay longer than you expect.

Future Work

- **Expand Airport Coverage**: Extend the tool's functionality to include all airports in the USA.
- **Incorporate Arrival Information**: Refine the model to consider not only the departure airport and time but also the arrival airport and arrival time. This will enable more comprehensive flight delay predictions.
- **Simplified User Input**: Enhance the user interface by allowing users to simply enter the flight number. The tool should then automatically fetch relevant departure and arrival information, including airport codes and times.