IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nikola Nikolov
27/07/23

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
  - Descriptive Statistics: Summarizing data with metrics like mean, median, and standard deviation.
  - Inferential Statistics: Making predictions and drawing conclusions about populations from samples.
  - Machine Learning: Building models to learn patterns and make predictions or classifications.
  - Data Cleaning and Preprocessing: Handling missing data, outliers, and transforming data for analysis.
  - Data Visualization: Communicating insights through charts, graphs, and interactive visuals.
  - Exploratory Data Analysis (EDA): Uncovering patterns and relationships to guide further analysis.

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Surveys and Questionnaires: Researchers and organizations collect data through surveys and questionnaires where respondents answer specific questions about a particular topic or subject. These can be conducted in-person, over the phone, or online.
- Sensor Data: Sensors and IoT devices collect data from the environment, such as temperature, humidity, pressure, or motion data. This data can be used for various applications like weather forecasting, environmental monitoring, and smart devices.
- Web Scraping: Data can be collected from websites by using web scraping techniques to extract information from web pages. This is commonly done to gather data for analysis, research, or market intelligence.
- Transaction Records: Businesses often collect data from transactions, such as sales records, customer purchases, or financial transactions, to gain insights into their operations and customers..

Data Collection – SpaceX API

- Read API Documentation: Familiarize yourself with the API's documentation to understand its endpoints, parameters, and data formats. The documentation will guide you on how to structure your API requests.
- API Request: Use HTTP methods like GET, POST, PUT, or DELETE to make API requests. Pass the required parameters, including your API key, to access the desired data.

API Data Collection Flowchart (high-level representation):

Start -> Select an API -> Obtain API Key -> Read API Documentation -> API Request -> Receive API Response -> Data Parsing -> Data Cleaning and Preprocessing -> Data Storage -> Data Refresh and Update (Optional) -> Error Handling -> End.

Data Collection - Scraping

- Choose Web Scraping Tools: Decide on the web scraping tools and libraries to use. Commonly used tools include BeautifulSoup, Scrapy, or Selenium.
- Data Cleaning and Preprocessing: Clean and preprocess the extracted data to remove any unwanted characters, handle missing values, and convert data to the desired format.

Flowchart Steps (high-level representation):

Start -> Identify Target Website -> Choose Web Scraping Tools -> Inspect Website Structure -> Locate Target Data -> HTTP Request -> Download and Parse HTML -> Extract Data -> Data Cleaning and Preprocessing -> Data Storage -> Loop for Multiple Pages (Optional) -> Handle Anti-Scraping Mechanisms (Optional) -> Schedule and Monitor Scraping (Optional) -> End.

Data Wrangling

Data Collection: Gather the raw data from various sources, such as databases, APIs, web scraping, or files (e.g., CSV, Excel, JSON).
Data Inspection: Examine the dataset to understand its structure, check for missing values, outliers, and potential issues.
Handling Missing Data: Address missing data points by either removing the rows or columns with missing values or imputing them with appropriate values (e.g., mean, median, or interpolation).
Dealing with Duplicates: Identify and remove any duplicate records in the dataset to avoid redundancy.
Data Transformation: Convert data to the appropriate data types (e.g., numerical, categorical, datetime) for consistent analysis.
Feature Engineering: Create new features or transform existing ones to derive more meaningful variables for analysis.
Data Filtering: Filter out irrelevant or noisy data that may not be useful for the analysis.
Data Integration: Combine data from multiple sources if needed to create a unified dataset.
Handling Outliers: Detect and handle outliers by either removing them, capping them, or transforming them to minimize their impact on the analysis.
Data Normalization/Scaling: Scale numerical data to a similar range to prevent any feature from dominating the analysis due to its larger magnitude.
Data Aggregation: Group data and compute summary statistics to create aggregated views.
Data Encoding: Convert categorical data into numerical representations (e.g., one-hot encoding) for machine learning algorithms.

EDA with Data Visualization

- **Histograms**: Histograms are used to visualize the distribution of a single numerical variable. They help identify the data's central tendency, spread, and presence of outliers.
- **Box Plots**: Box plots (box-and-whisker plots) provide a visual summary of the data's distribution, showing median, quartiles, and outliers. They help detect potential outliers and compare the distributions of different groups.
- **Scatter Plots**: Scatter plots are used to visualize the relationship between two numerical variables. They help identify correlations and patterns in the data, such as positive, negative, or no correlation.
- **Line Plots**: Line plots are useful for visualizing trends and patterns in time series data or continuous data with a natural order.
- **Bar Charts**: Bar charts are used to compare the frequencies or counts of categorical variables. They help identify the most common categories and make comparisons between different groups.
- **Pie Charts**: Pie charts are suitable for showing the proportion of different categories within a single categorical variable.
- **Heatmaps**: Heatmaps are used to visualize the relationship between two categorical variables. They show the frequency or density of combinations of categories.

EDA with SQL

- **Connecting to the Database**: Connect to the database where your data is stored using SQL clients or libraries like Python's **`pandas`** or R's **`DBI`**.
- **Inspecting Data**: Use SQL commands to inspect the table structure, column names, data types, and sample records.
- **Summary Statistics**: Calculate summary statistics using SQL aggregation functions like COUNT, SUM, AVG, MIN, and MAX to get an overview of the data distribution.
- **Distinct Values**: Use the DISTINCT keyword to identify unique values in categorical columns, understanding the cardinality of each category.
- **Filtering Data**: Use the WHERE clause to filter data based on specific conditions, helping you focus on relevant subsets of the data.
- **Grouping and Aggregating**: Use the GROUP BY clause to group data based on one or more columns and apply aggregate functions to analyze subsets of data.
- **Sorting Data**: Use the ORDER BY clause to sort data based on one or more columns, allowing you to observe patterns more easily.
- **Joining Tables**: If your data is spread across multiple tables, use SQL JOINs to combine related data, enabling more comprehensive analysis.
- **Data Cleaning**: Use SQL queries to handle missing values or remove duplicates from the dataset.
- **Subqueries**: Utilize subqueries to perform more complex analyses, such as calculating derived metrics or filtering data based on the results of other queries.

Build an Interactive Map with Folium

- Import Required Libraries: Import the necessary Python libraries, including Folium, Pandas, and any other relevant libraries.
- Data Preparation: Prepare the data that will be displayed on the map. This may involve reading data from a CSV file, database, or API, and processing it as needed.
- Initialize Map: Create a new Folium map object by calling the 'Map()' function. Specify the map's center coordinates and initial zoom level.
- Add Base Map (Optional): Optionally, add a base map layer, such as OpenStreetMap, Stamen Terrain, or Mapbox, to provide context to the data.
- Data Visualization: For each data point, loop through the data and use Folium's marker, circle, or other relevant functions to visualize the data on the map. Customize the markers with tooltips, popups, and icons to display additional information.
- Data Clustering (Optional): If there are a large number of data points, consider using Folium's MarkerCluster or plugins like FastMarkerCluster to improve map performance and user experience.
- Add Other Map Elements (Optional): Add other map elements like polygons, lines, or choropleth layers to visualize additional information, such as boundaries, routes, or heatmaps.
- Layer Control (Optional): If there are multiple layers on the map, add a layer control to allow users to toggle the visibility of different data layers. -> Save and Display Map: Save the map to an HTML file using the 'save()' method and display the map in the Jupyter Notebook or a web browser.

Build a Dashboard with Plotly Dash

- Plotly Dash allows data scientists and developers to seamlessly blend the power of Python's data analysis capabilities with the interactivity and responsiveness of web applications. By leveraging the familiar Python syntax and libraries, such as Pandas and Plotly, one can quickly transform data insights into dynamic, interactive dashboards that can be easily shared and accessed by others. This empowers users to explore data, gain valuable insights, and make data-driven decisions in a visually appealing and user-friendly manner.

Predictive Analysis (Classification)

- Model Selection: The process of choosing the right classification model is a critical aspect of executing predictive analysis. It involves a fascinating journey of exploration, experimentation, and trade-offs between various algorithms, each offering unique strengths and limitations.
- Trade-offs: Model selection involves thought-provoking trade-offs. For example, a simple Logistic Regression model might be highly interpretable but less flexible, while a complex Neural Network could offer outstanding performance but be harder to interpret. Weighing the pros and cons of each model requires a delicate balance to ensure the chosen model aligns with the overall project objectives.

Results

Data scientist's presentation of results should highlight the model's predictive capabilities, the significance of key features, potential opportunities, and challenges, while being transparent about limitations and the need for ongoing model maintenance. The emphasis on actionable insights and the potential impact of the findings helps stakeholders understand the value of the data analysis and encourages data-driven decision-making.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

Description:

•The bar chart visually depicts the relationship between Flight Number and Launch Site for SpaceX missions.

•Each bar represents a unique flight number, and its height corresponds to the number of launches from the associated launch site.

•The launch sites are color-coded for easy identification.

Key Observations:

•Launch Site 1: Shows the highest number of flights, suggesting it is the primary launch site for SpaceX missions.

•Launch Site 2: Demonstrates a significant number of flights, indicating it is also frequently used for missions.

•Launch Sites 3 and 4: Have relatively fewer flights, indicating less frequent usage.

Payload vs. Launch Site

Description:

- The scatter plot illustrates the relationship between Payload Mass and Launch Site for SpaceX missions.
- Each data point represents a payload, and its position on the plot corresponds to its mass and launch site.

Key Observations:

- Clustered Payloads: Several data points are clustered in specific regions, indicating similar payload masses launched from the same site.
- Payload Range: The scatter plot shows a wide range of payload masses launched from different sites.
- Outliers: A few data points may appear as outliers, representing exceptionally heavy or light payloads.

Success Rate vs. Orbit Type

Description:

The bar chart showcases the relationship between the Success Rate of SpaceX missions and their respective Orbit Types.

Each bar represents a specific Orbit Type, and its height reflects the success rate of missions launched into that orbit.

Key Observations:

High Success Rate: Certain Orbit Types exhibit a high success rate, indicating reliable mission outcomes.

Moderate Success Rate: Some Orbit Types demonstrate moderate success rates, suggesting occasional challenges.

Low Success Rate: A few Orbit Types have lower success rates, indicating areas of improvement or complexity.

Flight Number vs. Orbit Type

Description:

The scatter plot illustrates the relationship between Flight Number and Orbit Type for SpaceX missions.

Each data point represents a unique flight number, and its position on the plot corresponds to the orbit type of the mission.

Key Observations:

Distribution: The scatter plot showcases the distribution of different orbit types across various SpaceX missions.

Multiple Orbits: Several flight numbers have data points associated with multiple orbit types, indicating mission versatility.

Orbit Patterns: Observing patterns in the data points may reveal trends in orbit preferences over time.

Total Number of Successful and Failure Mission Outcomes

Description:

The pie chart displays the total number of SpaceX missions categorized into Successful and Failure outcomes.

Key Observations:

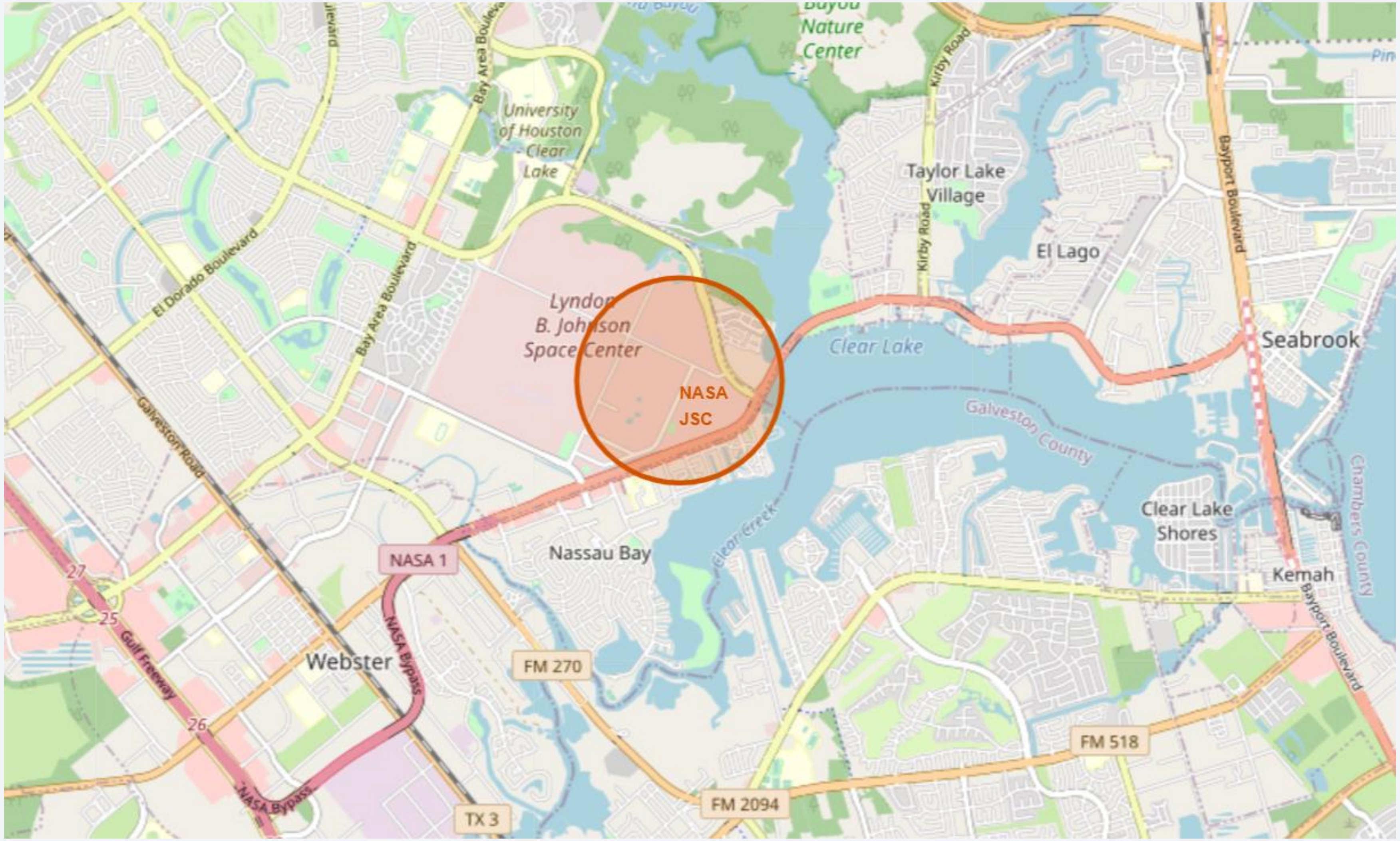
Successful Missions: The portion of the pie chart representing the total number of successful missions.

Failure Missions: The portion of the pie chart representing the total number of missions that ended in failure.

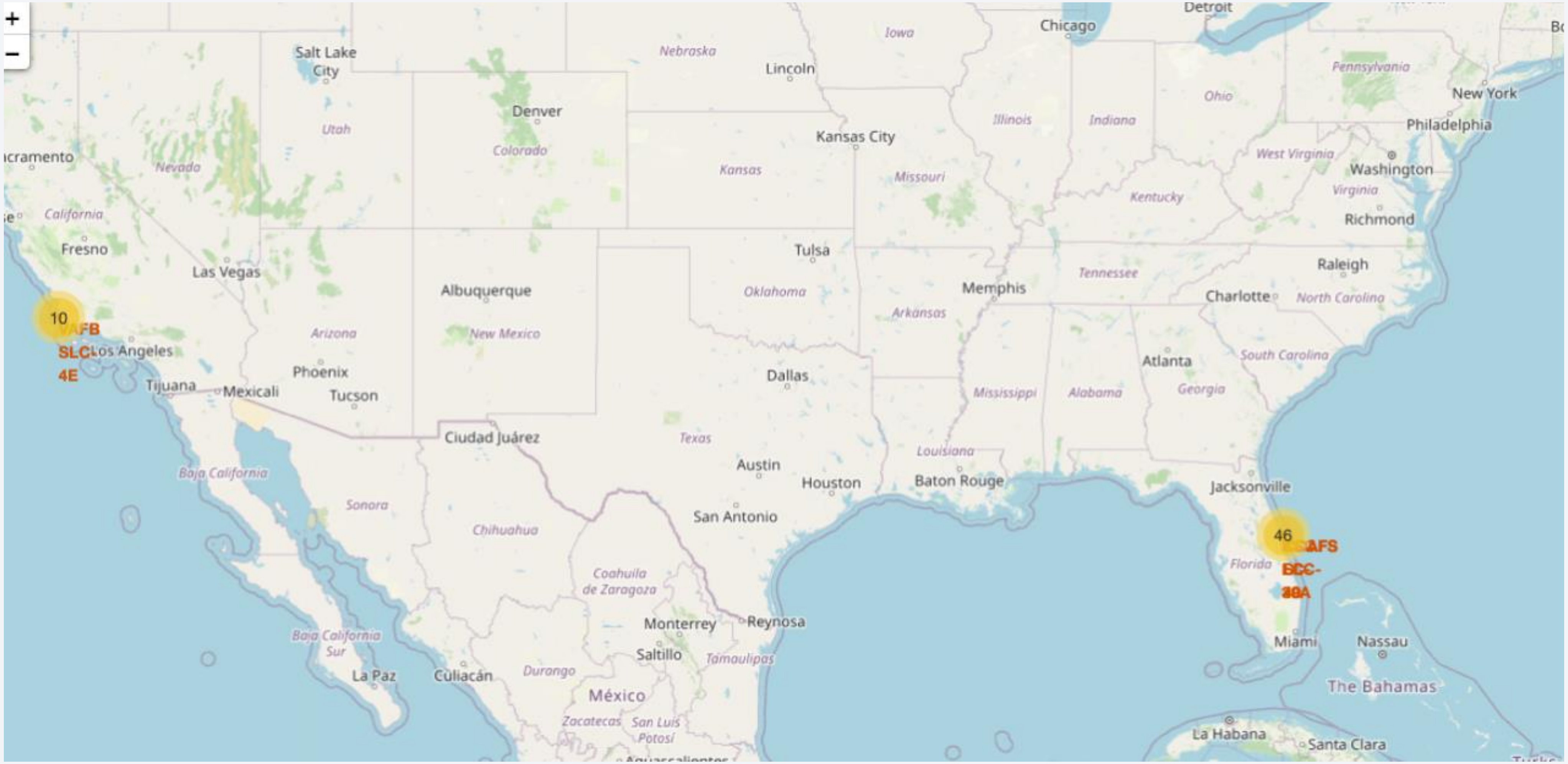
Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

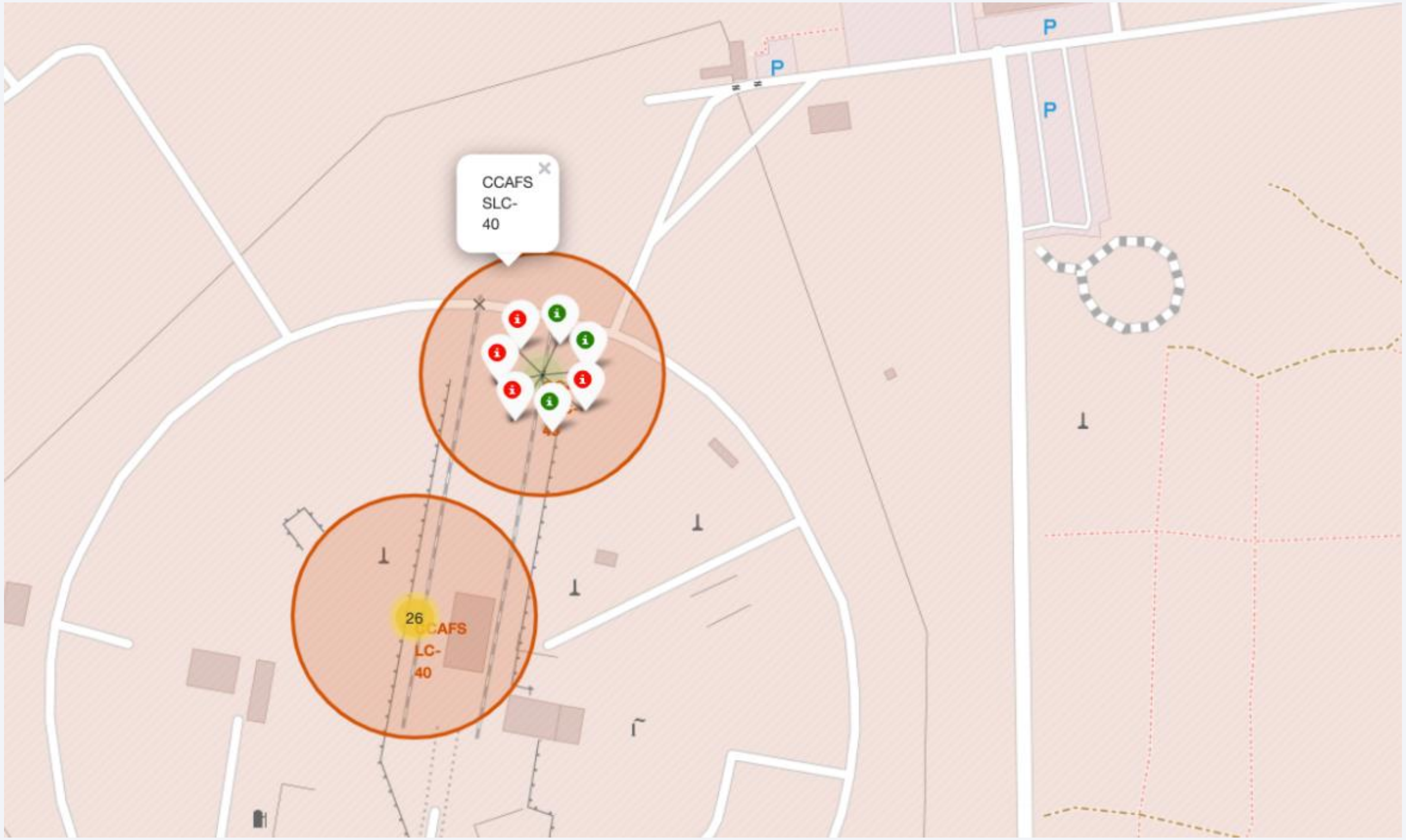

Bayou
Nature
Center
University of Houston Clear Lake
Taylor Lake Village
El Lago
Seabrook
Lyndon B. Johnson Space Center
NASA
JSC
Clear Lake
Galveston County
Clear Lake Shores
Kemah
Nassau Bay
Webster
NASA 1
NASA Bypass
FM 270
FM 518
FM 2094
TX 3
El Dorado Boulevard
Bay Area Boulevard
Galveston Road
Gulf Freeway
Kirby Road
Baypoint Boulevard
Chambers County
Clear Creek

<Folium Map Screenshot 2>

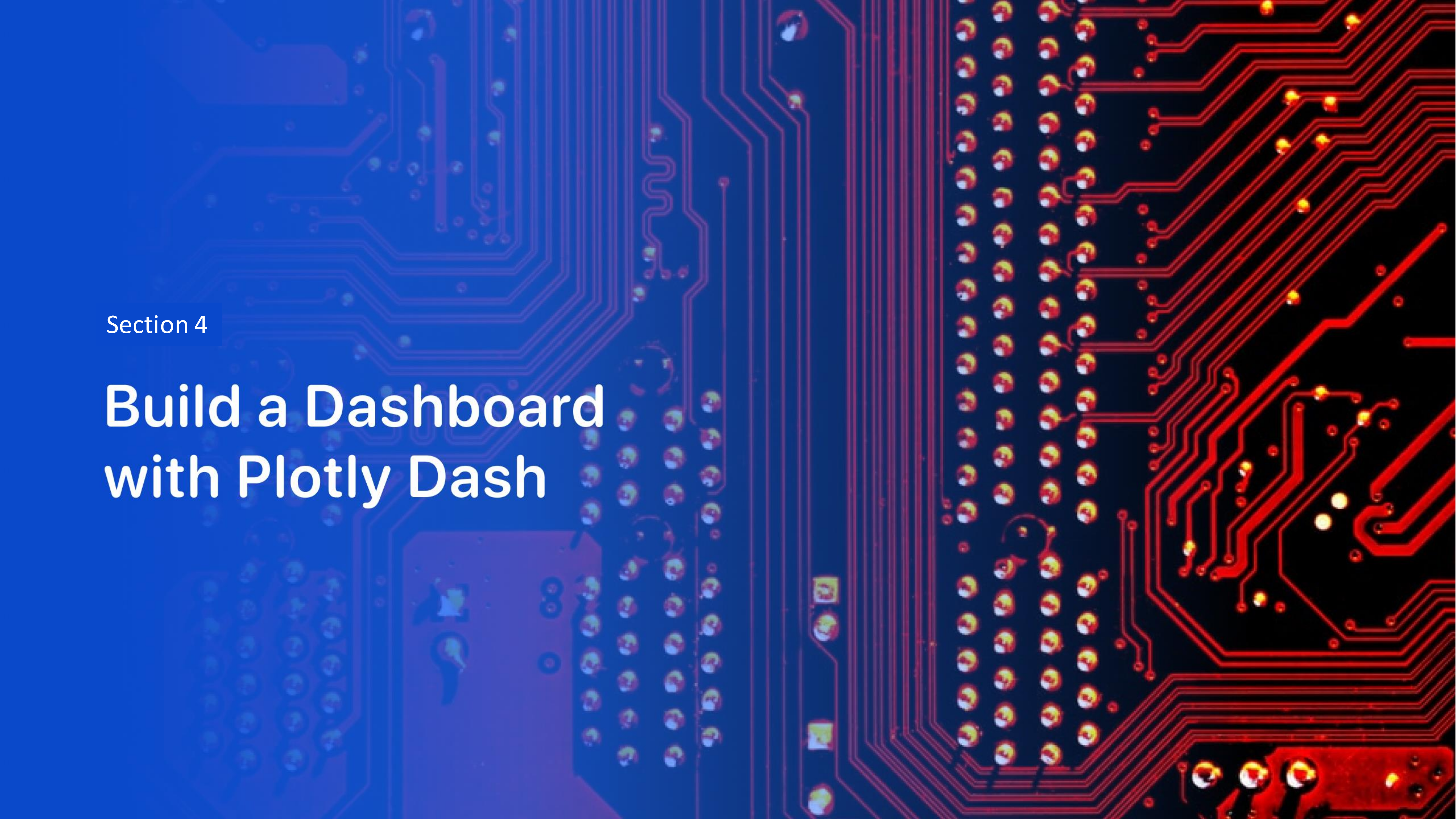


+
−
10
VAFB
SLC-4E
46
CCAFS
CCC-
39A
Salt Lake City
Denver
Lincoln
Kansas City
Chicago
Detroit
New York
Philadelphia
Washington
Richmond
Raleigh
Charlotte
Atlanta
Jacksonville
Miami
Nassau
The Bahamas
La Habana
Santa Clara
Memphis
Tulsa
Dallas
Austin
Houston
San Antonio
Baton Rouge
Albuquerque
Phoenix
Tucson
Las Vegas
Fresno
Los Angeles
Tijuana
Mexicali
Ciudad Juárez
Monterrey
Saltillo
Reynosa
Culiacán
La Paz
México

<Folium Map Screenshot 3>



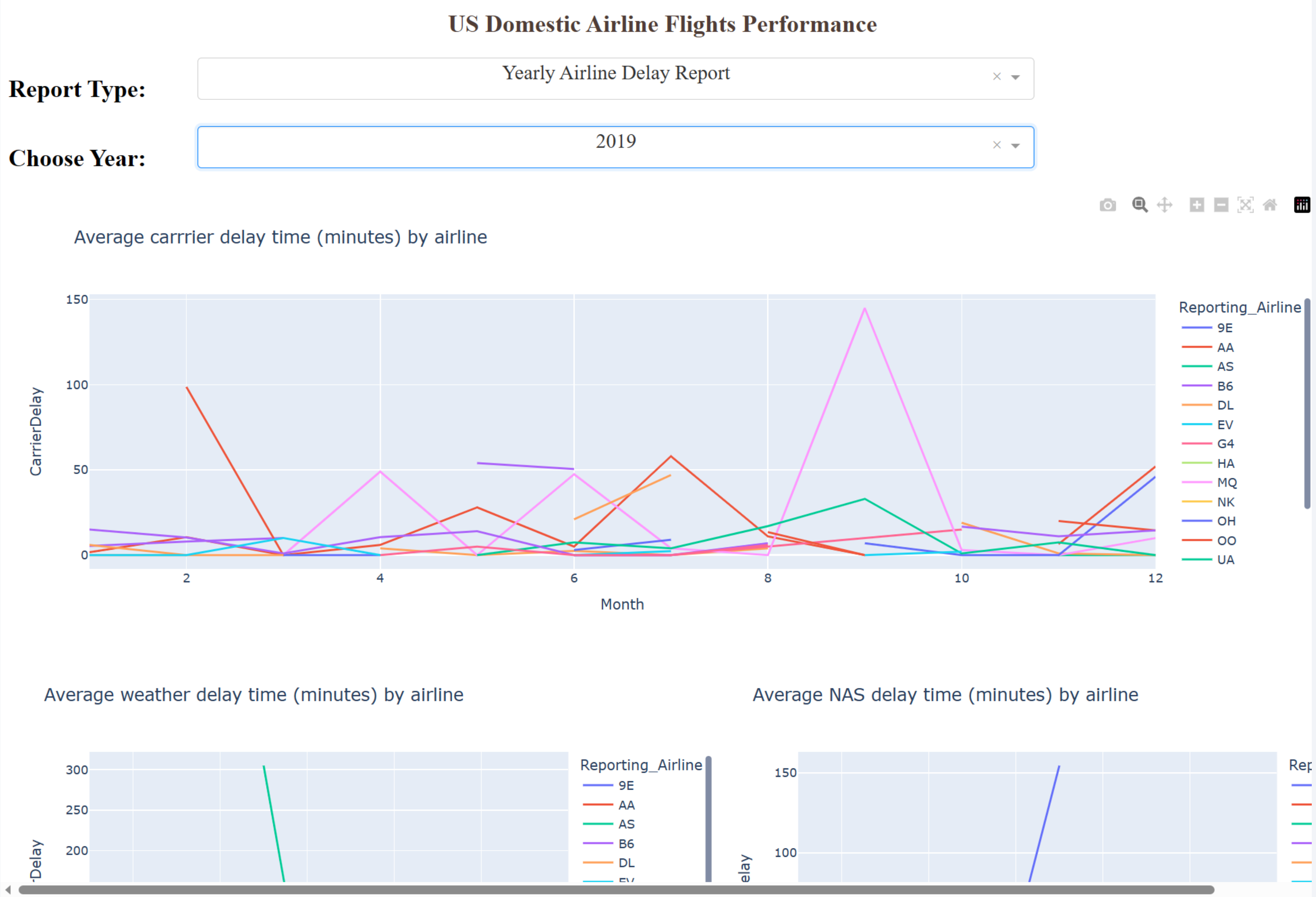
CCAFS
SLC-
40
26
CCAFS
LC-
40
P
P
P



Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



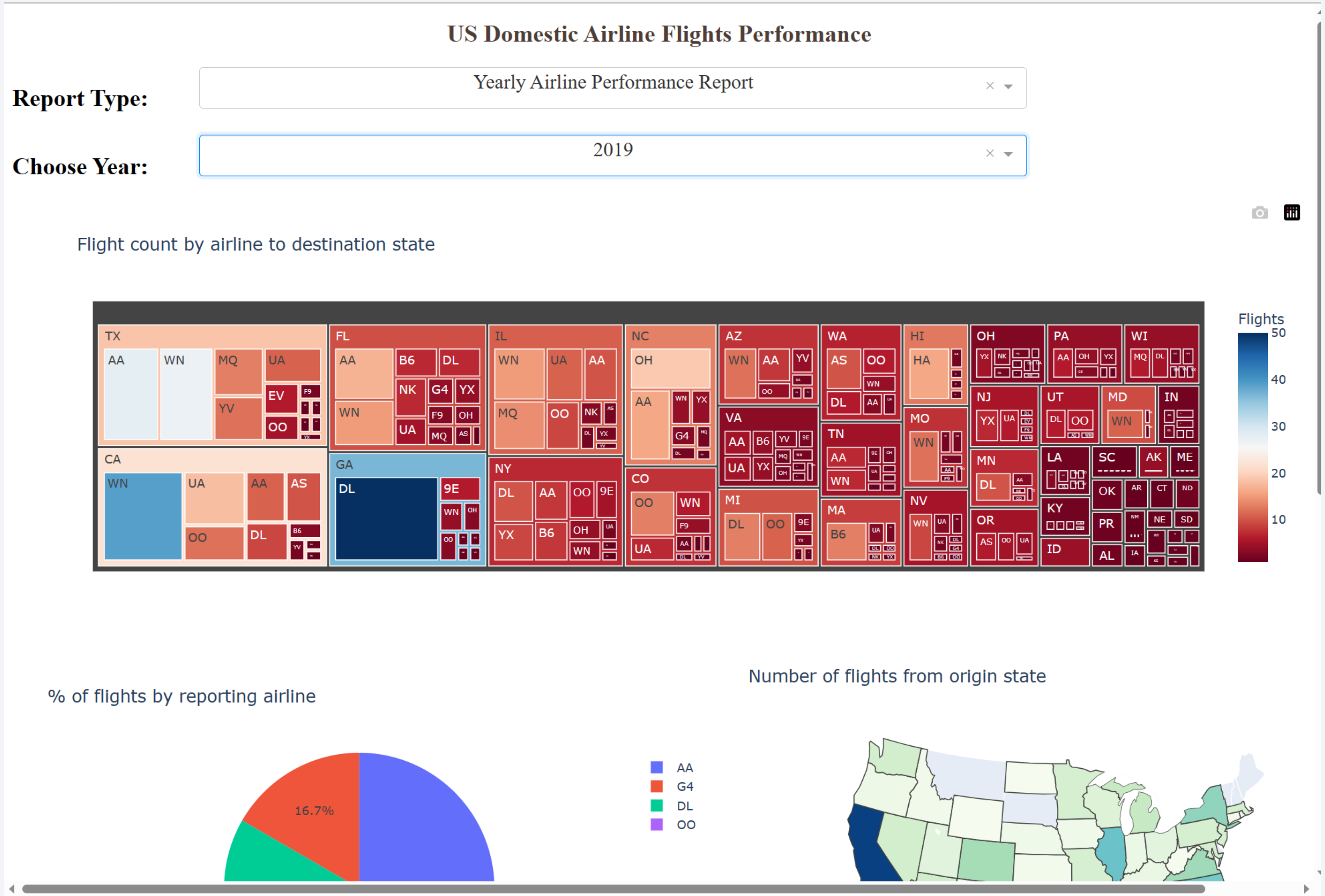
US Domestic Airline Flights Performance

Report Type: Yearly Airline Delay Report

Choose Year: 2019

Average carrrier delay time (minutes) by airline
CarrierDelay
150
100
50
0
2
4
6
8
10
12
Month
Reporting_Airline
9E
AA
AS
B6
DL
EV
G4
HA
MQ
NK
OH
OO
UA
Average weather delay time (minutes) by airline
300
250
200
Delay
Reporting_Airline
9E
AA
AS
B6
DL
Average NAS delay time (minutes) by airline
150
100
elay
Rep

<Dashboard Screenshot 2>



US Domestic Airline Flights Performance

Report Type: Yearly Airline Performance Report

Choose Year: 2019

Flight count by airline to destination state

% of flights by reporting airline

Number of flights from origin state

Thank you!