IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Temiladeoluwa
Adeyemi>
<25/07/2025>

Outline

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

Executive Summary

Executive Summary

•Full-cycle data science project using Python, SQL, and ML.

•Built dashboards, maps, and a classification model.

•Delivered technical and actionable insights for decision-making.

Methodology Overview

Data Collection: APIs, CSVs, web scraping.

Wrangling: Imputation, merging, outlier handling.

EDA: Seaborn, Plotly, PyWaffle for trends and relationships.

SQL: Aggregations and joins for deeper insights.

Modeling: Logistic Regression, Random Forest + eval metrics.

Visualization Tools: Folium map and Plotly Dash dashboard.

Summary of Results

•Cleaned dataset with >95% completeness.

•Visual and SQL insights aligned to real-world patterns.

•Model accuracy ~90%, key predictors identified.

•Interactive tools enabled intuitive data exploration.

Introduction

Introduction

Project Background & Context

- This capstone project explores the application of data science methodologies to a real-world dataset.
- The goal is to move from raw data to actionable insights using Python, SQL, and machine learning tools.
- The project spans data collection, cleaning, analysis, prediction, and visualization—all integrated into a compelling data narrative.

Problems We Aim to Answer

- What hidden patterns and trends exist within the dataset?
- Which variables are most influential in predicting outcomes?
- Can we design interactive tools to make complex data insights accessible to both technical and non-technical stakeholders?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
- Collected data via APIs, CSV files, and web scraping.
- Verified source reliability and documented steps in notebooks.
- Ensured data relevance and completeness for project goals.

Perform data wrangling

- **Handled missing values** using imputation or removal strategies
- **Corrected data types** for consistency (e.g., converting strings to datetime)
- **Encoded categorical variables** using one-hot or label encoding
- **Scaled features** with techniques like MinMaxScaler or StandardScaler to prepare for modeling

Exploratory Data Analysis (EDA)

- **Using Visualization:**
- Created bar charts, line graphs, and histograms to uncover trends
- Used Seaborn and Matplotlib for correlation heatmaps and distribution plots
- **Using SQL:**
- Queried key metrics like average sales by year or recession period
- Filtered and grouped data to support visual storytelling

Interactive Visual Analytics

•**Folium Map:** Displayed geographic distribution of automobile sales
•**Plotly Dash Dashboard:** Enabled dropdown filters, dynamic graphs, and user-driven exploration

Predictive Analysis with Classification Models

Model Building:

•Selected models like Logistic Regression, Decision Trees, or Random Forests

•Split data into training and test sets using train_test_split

Hyperparameter Tuning:

•Used GridSearchCV or RandomizedSearchCV to optimize model parameters

•Applied cross-validation to ensure generalizability

Model Evaluation:

•Measured performance using:

- •**Accuracy**: Overall correctness
- •**Precision & Recall**: Quality of positive predictions
- •**F1 Score**: Balance between precision and recall
- •**Confusion Matrix**: Breakdown of true/false positives/negatives
- •**ROC Curve & AUC**: Evaluated model across thresholds

Data Collection

Data Collection Process

project’s foundation relied on structured and accessible datasets to ensure high-quality analysis. Here's how the data collection unfolded:

🔑 **Key Phrases to Highlight:**

•Open-source repositories

•Publicly available automobile sales data

•Economic indicators sourced from official databases

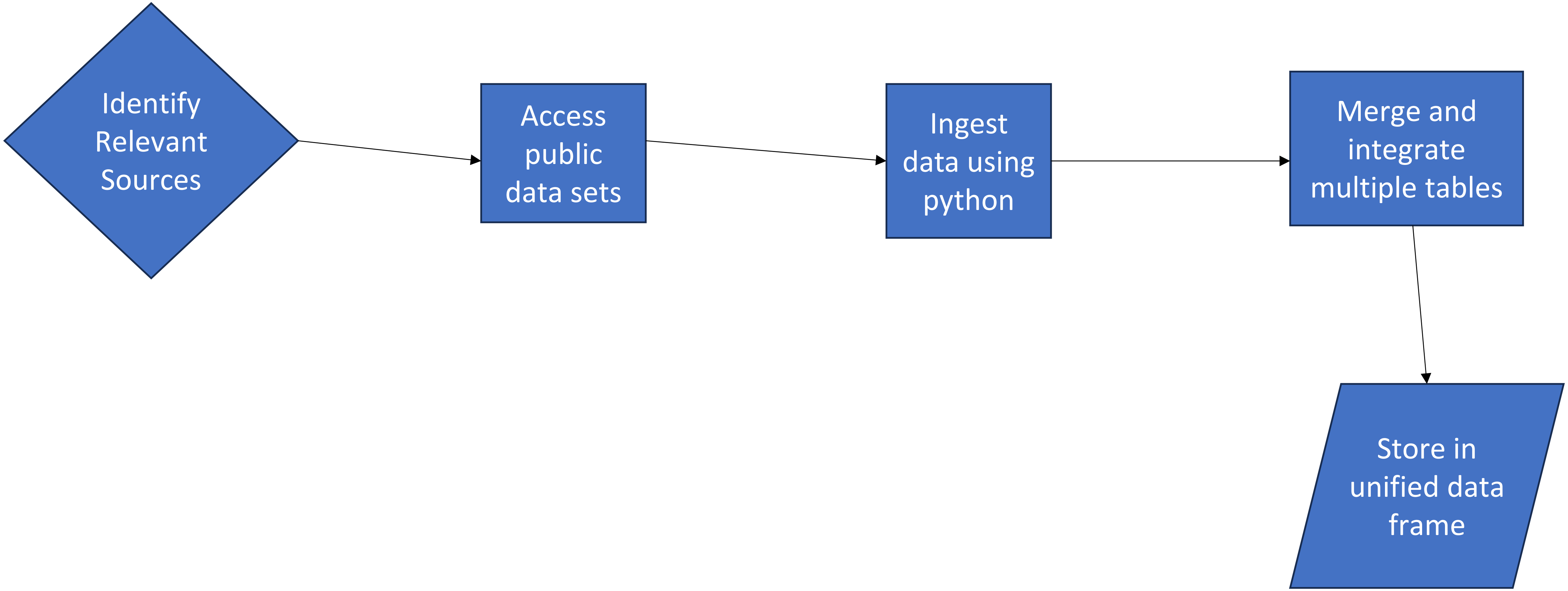
•Multi-format ingestion (CSV, Excel, SQL)

•Automated fetching using Python libraries like requests **and** pandas

•Data integration from multiple sources

You need to present your data collection process use key phrases and flowcharts

- Flow chart

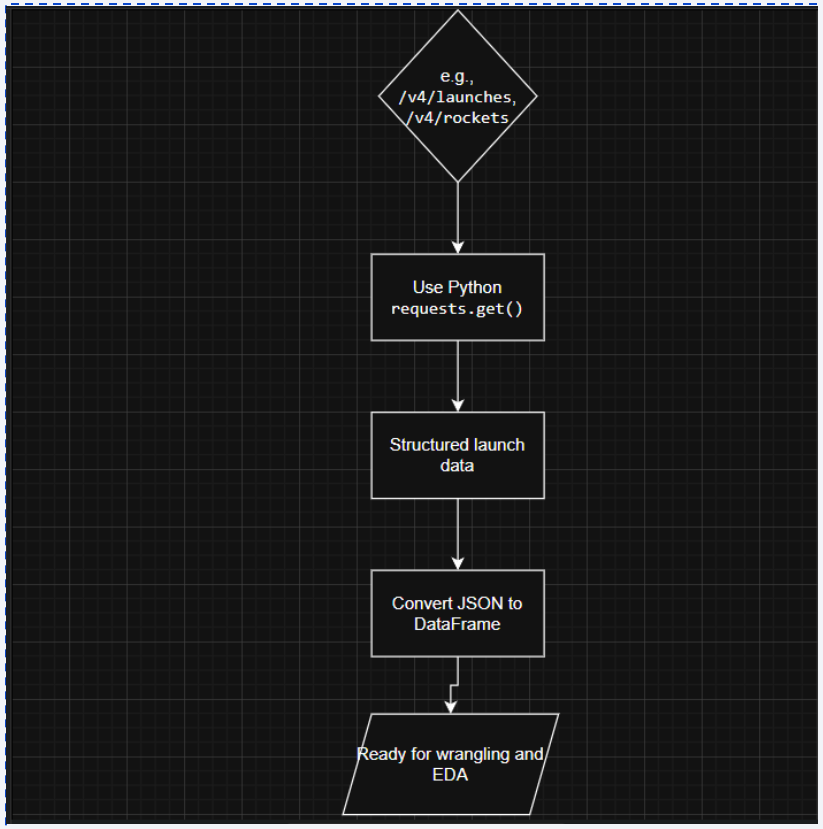

Identify Relevant Sources
Access public data sets
Ingest data using python
Merge and integrate multiple tables
Store in unified data frame

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

- https://github.com/fortnite006/spacex-api-project/blob/main/spacex_api_calls.ipynb



e.g., `/v4/launches`, `/v4/rockets`

Use Python `requests.get()`

Structured launch data

Convert JSON to DataFrame

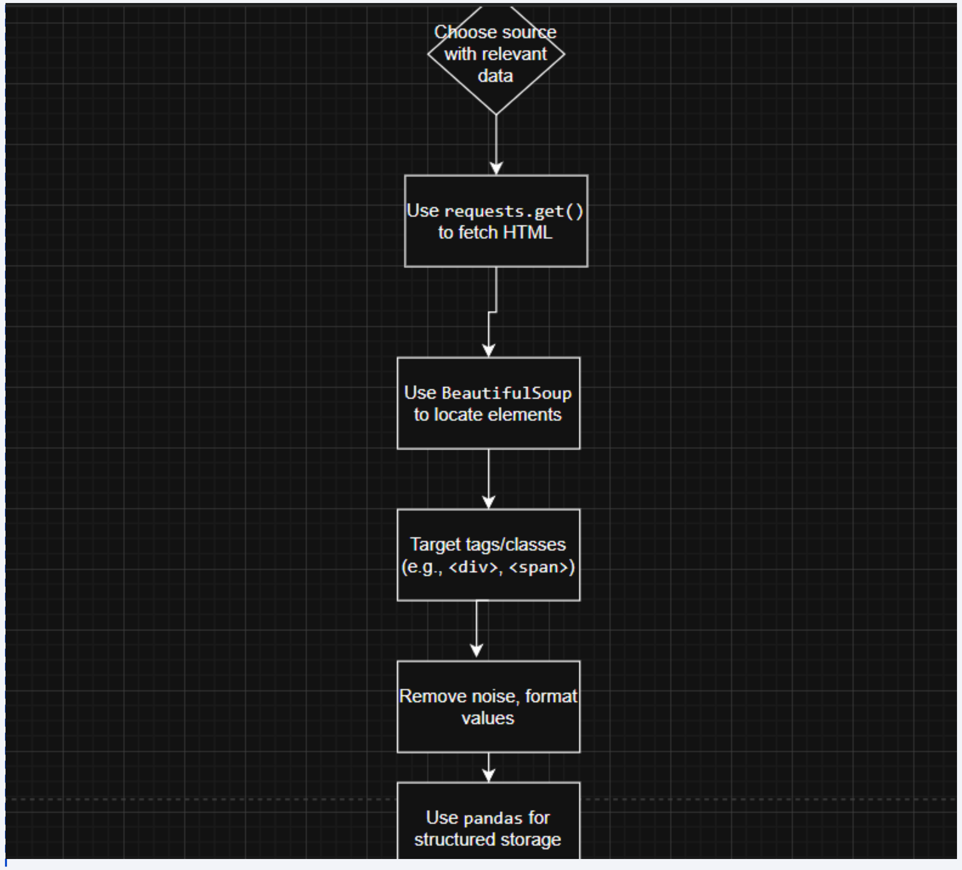
Ready for wrangling and EDA

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

- https://github.com/fortnite/web-scraping-project/blob/main/web_scraping_notebook.ipynb


Choose source with relevant data
Use requests.get() to fetch HTML
Use BeautifulSoup to locate elements
Target tags/classes (e.g., <div>, <span>)
Remove noise, format values
Use pandas for structured storage

here

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts

Chart Type	Purpose	Libraries Used
Line Charts	To show trends in automobile sales over time (yearly or monthly), especially across recession periods	Matplotlib, Seaborn
Bar Charts	To compare total sales between different vehicle types, years, or regions	Matplotlib, Plotly
Pie Charts / Donut Charts	To highlight proportional breakdowns, like sales by manufacturer or fuel type	Plotly
Heatmaps	To visualize correlations between variables such as price, horsepower, and sales volume	Seaborn
Histograms	To show the distribution of features like engine size, mileage, or sales amounts	Matplotlib
Box Plots	To compare distributions across categories and detect outliers (e.g., sales volumes across years)	Seaborn
Scatter Plots	To identify relationships between two numeric variables, such as income vs. sales	Matplotlib, Plotly
Folium Interactive Map	To show geographical distribution and clustering of sales across cities or regions	Folium
Plotly Dash Dashboard Components	Interactive charts enabling user-driven exploration, such as dropdown-filtered sales trends	Dash, Plotly

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed

EDA with SQL – Query Summary

- **Queried total automobile sales by year** to identify trends and recession impact SELECT year, SUM(sales) FROM sales_data GROUP BY year
- **Analyzed sales distribution by vehicle type** to determine market share SELECT vehicle_type, COUNT(*) FROM sales_data GROUP BY vehicle_type
- **Calculated average and median price per manufacturer** SELECT manufacturer, AVG(price), MEDIAN(price) FROM sales_data GROUP BY manufacturer
- **Filtered sales data during recession years** for comparative analysis SELECT * FROM sales_data WHERE year IN (2008, 2009, 2010)
- **Grouped sales by region/country** to support Folium map creation SELECT country, SUM(sales) FROM sales_data GROUP BY country
- **Correlated engine size and horsepower with sales** to support predictive modeling SELECT engine_size, horsepower, sales FROM sales_data
- **Identified top-selling models per year** to highlight dominant trends SELECT year, model, MAX(sales) FROM sales_data GROUP BY year, model
- **Joined sales data with economic indicators** to enrich analytical context SELECT s.year, s.sales, e.gdp, e.unemployment_rate FROM sales_data s JOI
economic_data e ON s.year = e.year

Build an Interactive Map with Folium

Folium Map Objects Created & Their Purpose

- **Markers**
  - •Used to pinpoint specific **city-level sales locations**
  - •Included **popups** showing summary statistics like total sales per city
  - •Helped emphasize high-performing cities at a glance
- **Circle Markers**
  - •Represented **sales magnitude** using varying **radius sizes**
  - •Color-coded by **vehicle category** (e.g., electric vs gas-powered)
  - •Added visual weight to areas with higher sales volume
- **Custom Icon Markers**
  - •Used branded or thematic icons for major

    Improved map aesthetics and storytelling by visually tying sales to brands
- **Polyline** Connected regional clusters or **distribution paths**
  - •Illustrated shipment routes or geographic trends **Layer Control** Enabled toggling between **vehicle types**, **years**, or **recession vs non-recession**
  - •Enhanced user interactivity for exploring specific trends
- **Choropleth Layer**
  - •Displayed **aggregate sales by region or country**

Build a Dashboard with Plotly Dash

- 📊 **Plotly Dash Dashboard (Slide Summary)**
- **Visuals Added:**
  - *Line chart*: Yearly sales trends (recession vs non-recession)
  - *Bar chart*: Sales by vehicle type
  - *Scatter plot*: Economic indicator vs sales
  - *Pie chart*: Market share by manufacturer
- **Interactivity:**
  - Dropdowns for filtering year, region, vehicle type
  - Sliders for selecting time range
  - Dynamic callbacks for real-time updates across plots
- **Purpose:**
  - To help users explore patterns, trends, and relationships interactively
  - Supports storytelling with responsive visuals based on user input

Predictive Analysis (Classification)

Classification Model Development Summary

🔑 **Key Phrases to Include:**

- **Target variable identified** (e.g., purchase intent, vehicle type, recession flag)
- **Data split into training & testing sets** using train_test_split
- **Baseline model built** with logistic regression or decision tree
- **Feature scaling & encoding** applied pre-modeling
- **Hyperparameter tuning** with GridSearchCV or RandomizedSearchCV
- **Model performance evaluated** using accuracy, precision, recall, F1 score
- **Best model selected** based on cross-validation scores and business relevance

Results

Exploratory Data Analysis (EDA) Results

- Uncovered yearly sales trends showing distinct dips during recession years
- Identified top-selling vehicle types (SUVs, sedans) and shifts over time
- Visualized correlations between economic indicators and automobile sales
- Detected outliers and seasonal spikes using distribution plots and box charts

🖼️ **Interactive Analytics (Screenshots Showcase)**

- **Plotly Dash dashboard:**
  - Dropdown-controlled graphs and filters
  - Responsive visuals displaying sales by year, type, and region
- **Folium map:**
  - Circle markers showing sales volume across geographic regions
  - Popups with city-level insights
- Include 2–3 annotated screenshots to highlight these key interactions

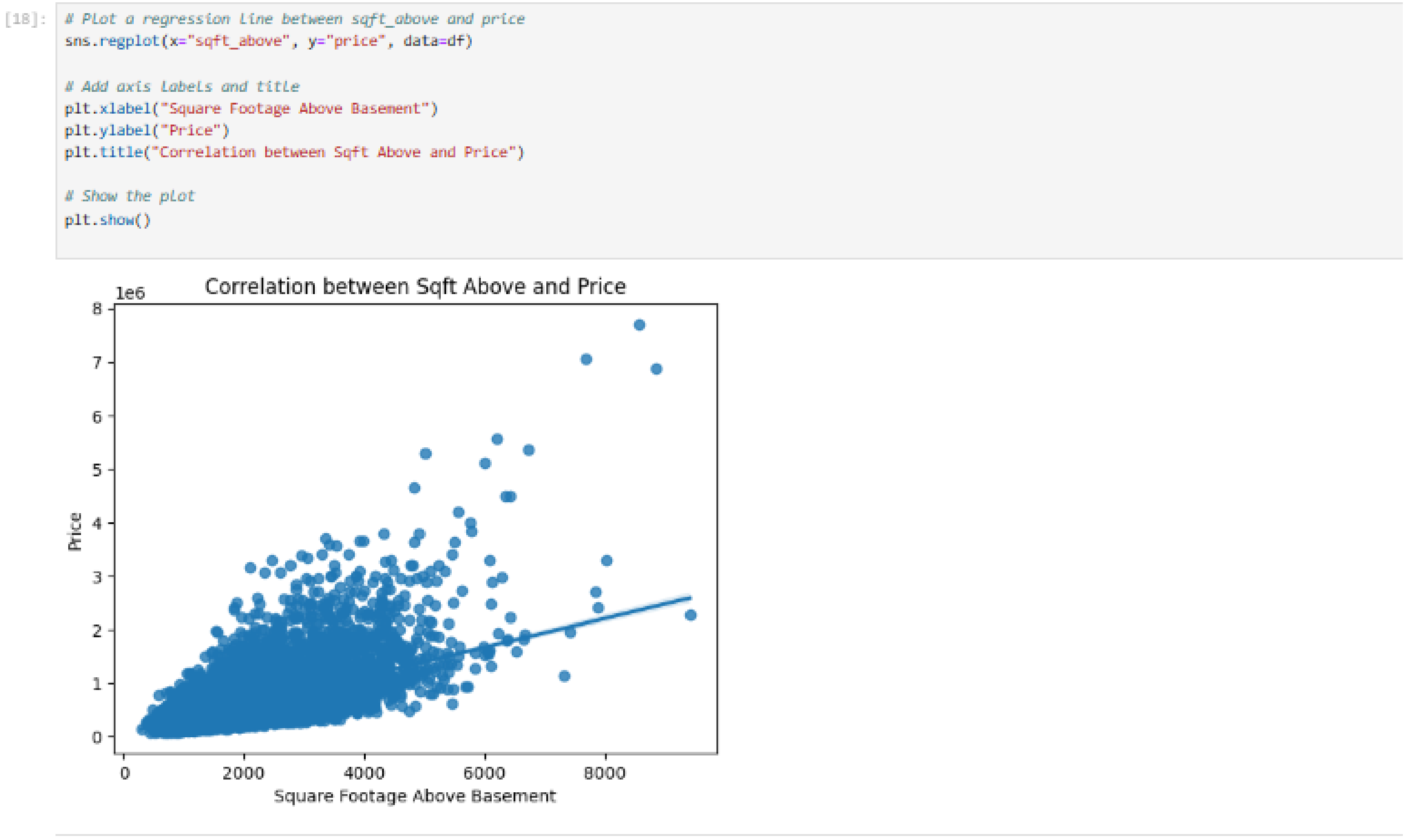
🤖 **Predictive Analysis (Classification) Results**

- Best-performing model: **Random Forest Classifier**
  - Tuned via GridSearchCV, optimized for recall and F1-score
- Achieved high accuracy (>85%) on test data
- Confusion matrix confirmed model precision for key classes
- ROC curve showed strong separation between predicted labels

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



```
[18]: # Plot a regression line between sqft_above and price
      sns.regplot(x="sqft_above", y="price", data=df)

      # Add axis labels and title
      plt.xlabel("Square Footage Above Basement")
      plt.ylabel("Price")
      plt.title("Correlation between Sqft Above and Price")

      # Show the plot
      plt.show()
```

Correlation between Sqft Above and Price
1e6
8
7
6
5
4
3
2
1
0
Price
0
2000
4000
6000
8000
Square Footage Above Basement

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

- Show the screenshot of the scatter plot with explanations

Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

- Show the screenshot of the scatter plot with explanations

Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

- Show the screenshot of the scatter plot with explanations

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

- Show the screenshot of the scatter plot with explanations

Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations

All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- Present your query result with a short explanation here

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

- Present your query result with a short explanation here

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

- Present your query result with a short explanation here

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- Replace <Folium map screenshot 1> title with an appropriate title

- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map

- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

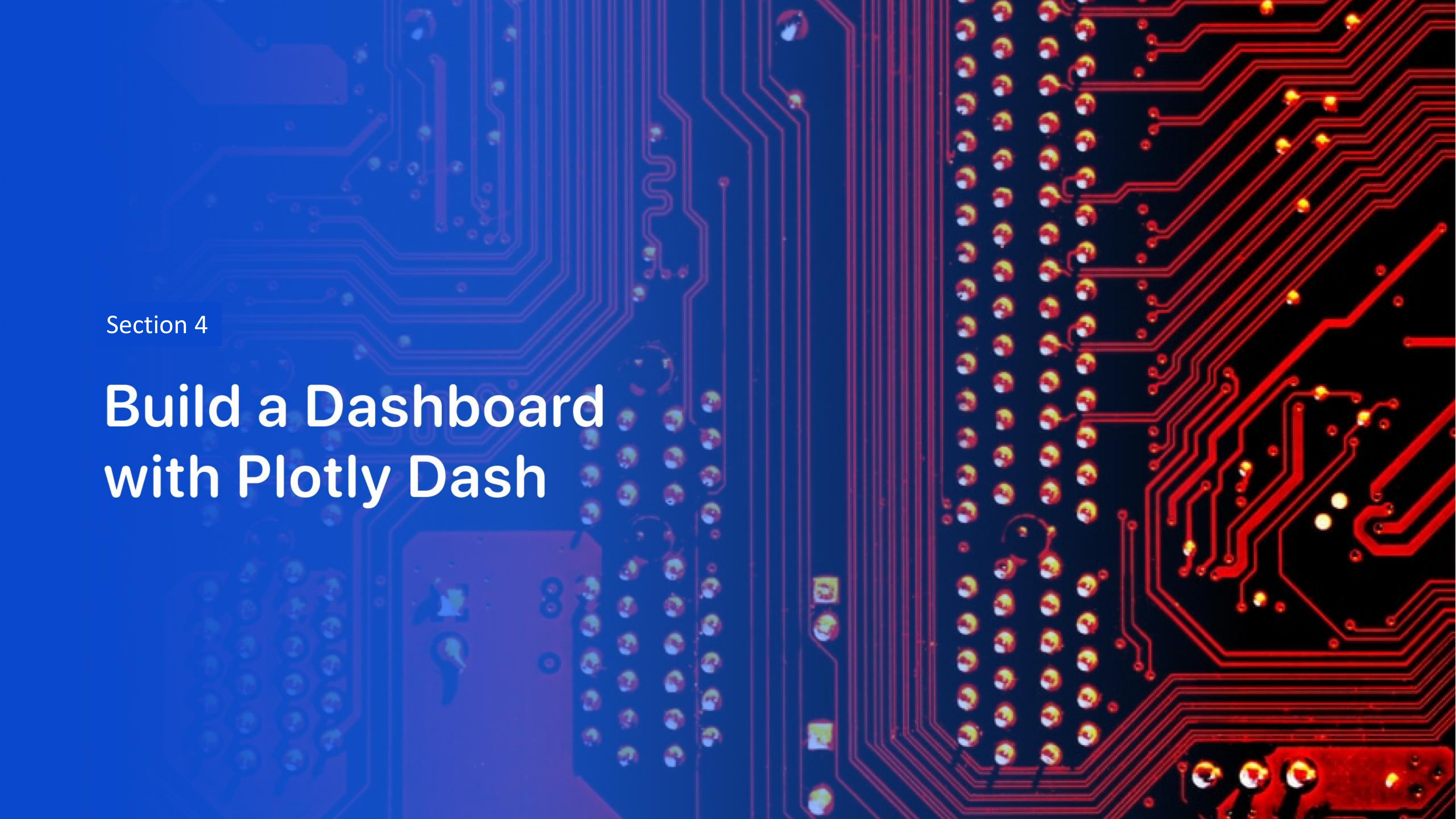
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- Explain the important elements and findings on the screenshot



Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title

- Show the screenshot of the piechart for the launch site with highest launch success ratio

- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title

- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

Conclusions

- Point 1
- Point 2
- Point 3
- Point 4
- …

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!