IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tewodros Hagos
05 June 2025

Outline

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

Executive Summary

- Summary of Methodologies:
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis
  - Interactive Visual Analytics
  - Predictive Analysis (Classification)

- Summary of Results:
  - Exploratory Data Analysis (EDA) results
  - Geospatial analytics
  - Interactive dashboard
  - Predictive analysis of classification models

Introduction

- Project background:
- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- Problems Statement:
- Predict if the first stage will land successfully as well as the cost of a launch in order to bid against SpaceX for a rocket launch.

Section 1

Methodology

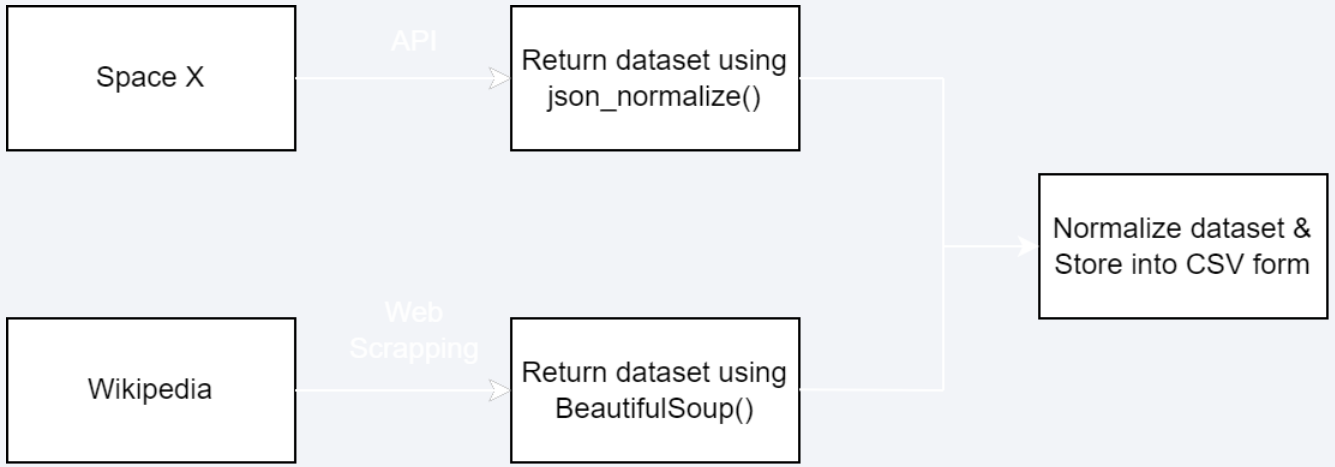
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Data cleaning, e.g., replacing null value with mean or median
  - One Hot Encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Model comparison : Linear Regression, Decision Tress, K-Nearest Neighbors (KNN),

Data Collection

- The data sets were collected through below two methods:
  - SpaceX API
  - Web Scrapping from Wikipedia
- Data collection process:



Data Collection – SpaceX API

- Request rocket launch data from SpaceX API
- Request and parse SpaceX launch data with a GET request and turning the JSON into a Pandas dataframe
- Use the functions created for using the API to extract information
- Construct dataset from data obtained from API and GET functions into a pandas dictionary with updating columns and rows.
- Replace null values with mean and export to CSV file

Place here

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

```
data.head()
```

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b6
1	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 301, 'altitude': 289, 'reason': 'harmonic oscillation leading to	Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown	[]	[]	[]	[5eb0e4b6b6

```
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].fillna(data_falcon9['PayloadMass'].mean())
data_falcon9.isnull().sum()
```

```
FlightNumber      0
Date              0
BoosterVersion    0
PayloadMass       0
Orbit             0
LaunchSite        0
Outcome           0
Flights           0
GridFins          0
Reused            0
Legs              0
LandingPad       26
Block             0
ReusedCount       0
Serial            0
Longitude         0
Latitude          0
dtype: int64
```

Data Collection - Scraping

- Request Falcon9 Launch HTML page HTTP method
- Create BeautifulSoup object
- Extract column name by iterate through each <th> element from HTML table header.
- Fill launch dictionary with launch records extracted from table rows with for loop to append into.
- Create dataframe from launch_dict with parsed launch values now added and export to CSV file.

```
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
```

```
soup = BeautifulSoup(response.text, 'html.parser')
```

```
for th in first_launch_table.find_all("th"):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(extract_column_from_header(th))
```

Check the extracted column names

```
print(column_names)
```

```
['Flight No.', 'Date and time ( )', 'Launch site', 'Payload', 'Payload mass', 'Orbit', 'Customer', 'Launch outcome']
```

Data Wrangling

- Load dataset from last section with read_csv function
- Calculate the number of column on each site with value_counts saperately.
- Use for loop to assign numbers to landing_outcomes, and list the failed landing outcomes
- Present list of outcomes with Class assigned as '0' or '1' for failure or success, respectively and export to CSV file.

EDA with Data Visualization

- Visualize the relationship between success rate of each orbit type
- Visualize the relationship between FlightNumber and Orbit type
- Visualize the launch success yearly trend
- Create dummy variables to categorical columns

EDA with SQL

The SQL queries performed on the data set were used to:

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string ‘CCA’
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display the average payload mass carried by booster version F9 v1.1
5. List the date when the first successful landing outcome on a ground pad was achieved
6. List the names of the boosters which had success on a drone ship and a payload mass between 4000 and 6000 kg
7. List the total number of successful and failed mission outcomes
8. List the names of the booster versions which have carried the maximum payload mass
9. List the failed landing outcomes on drone ships, their booster versions, and launch site names for 2015
10. 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

Build an Interactive Map with Folium

1. Mark all launch sites on a map

- Initialise the map using a Folium Map object
- Add a folium.Circle and folium.Marker for each launch site on the launch map

2. Mark the success/failed launches for each site on a map

- As many launches have the same coordinates, it makes sense to cluster them together.
- Before clustering them, assign a marker colour of successful (class = 1) as green, and failed (class = 0) as red.
- To put the launches into clusters, for each launch, add a folium.Marker to the MarkerCluster() object.
- Create an icon as a text label, assigning the icon_color as the marker_colour determined previously.

3. Calculate the distances between a launch site to its proximities

- To explore the proximities of launch sites, calculations of distances between points can be made using the Lat and Long values.
- After marking a point using the Lat and Long values, create a folium.Marker object to show the distance.
- To display the distance line between two points, draw a folium.PolyLine and add this to the map.

Build a Dashboard with Plotly Dash

1. Pie chart (px.pie()) showing the total successful launches per site

- This makes it clear to see which sites are most successful
- The chart could also be filtered (using a dcc.Dropdown() object) to see the success/failure ratio for an individual site

2. Scatter graph (px.scatter()) to show the correlation between outcome (success or not) and payload mass (kg)

- This could be filtered (using a RangeSlider() object) by ranges of payload masses
- It could also be filtered by booster version

Predictive Analysis (Classification)

1. Model Development

- Create a GridSearchCV object and a dictionary of parameters
- Fit the object to the parameters
- Use the training data set to train the model

2. Model Evaluation

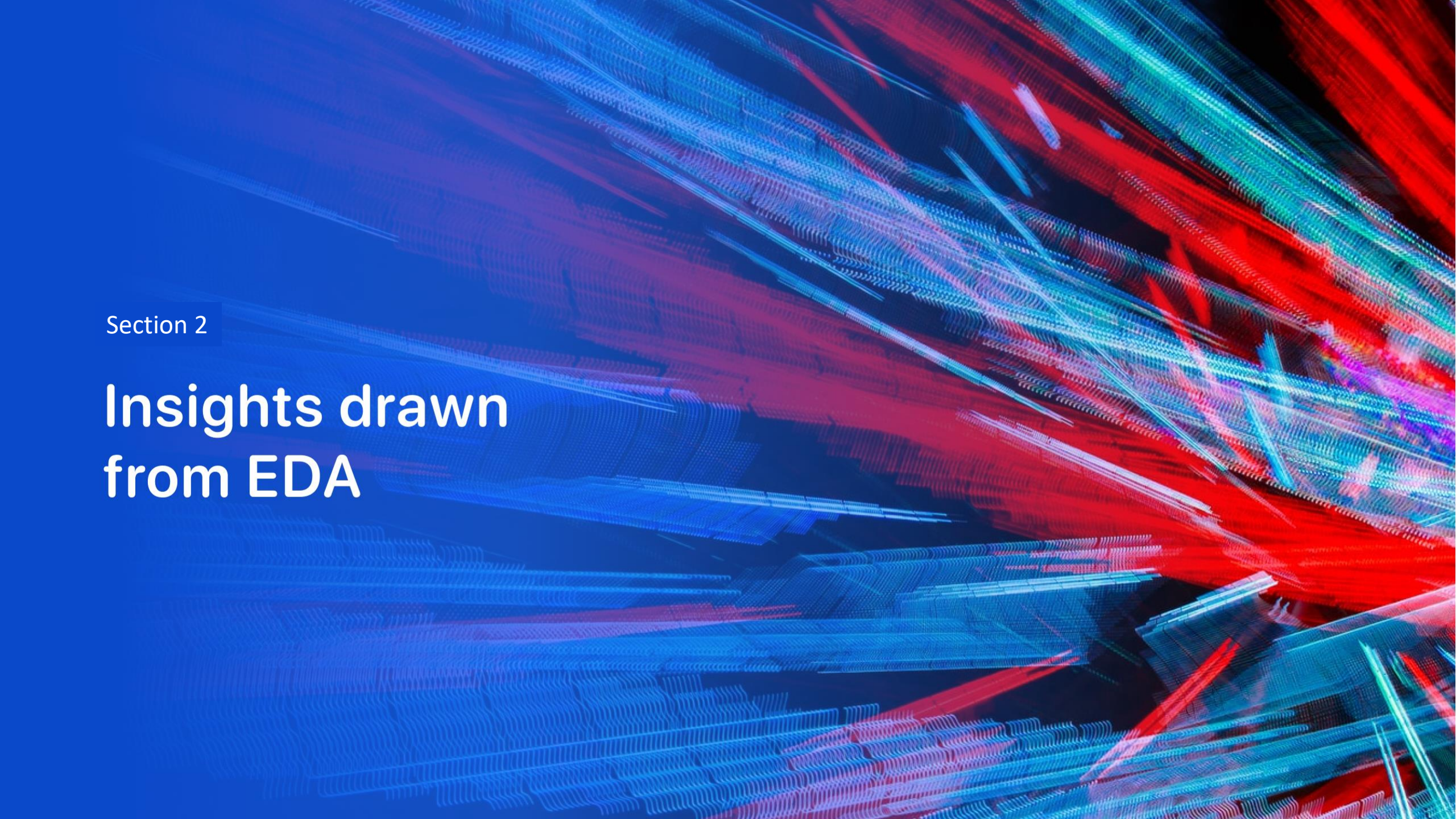
- Using the output GridSearchCV object
- Plot and examine the Confusion Matrix

3. Finding the Best Classification Model

- Review the accuracy scores for all chosen algorithms and record the highest one

Results

- Exploratory data analysis results
  - The success rate kept increasing since year 2013.
  - Orbit at ES-L1, GEO, HEO & SSO with a 100% success rate for rocket launching.
  - Based on scatter point chart, you will find that the VAFB-SLC launch site has no rockets
  - launched for heavy payload mass (greater than 10000).
- Interactive analytics demo in screenshots
  - KSC LC-39A had the most successful launches compared with the other sites.
- Predictive analysis results
  - Decision tree is the best model in terms of prediction accuracy in this dataset


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site

- Show the screenshot of the scatter plot with explanations

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>

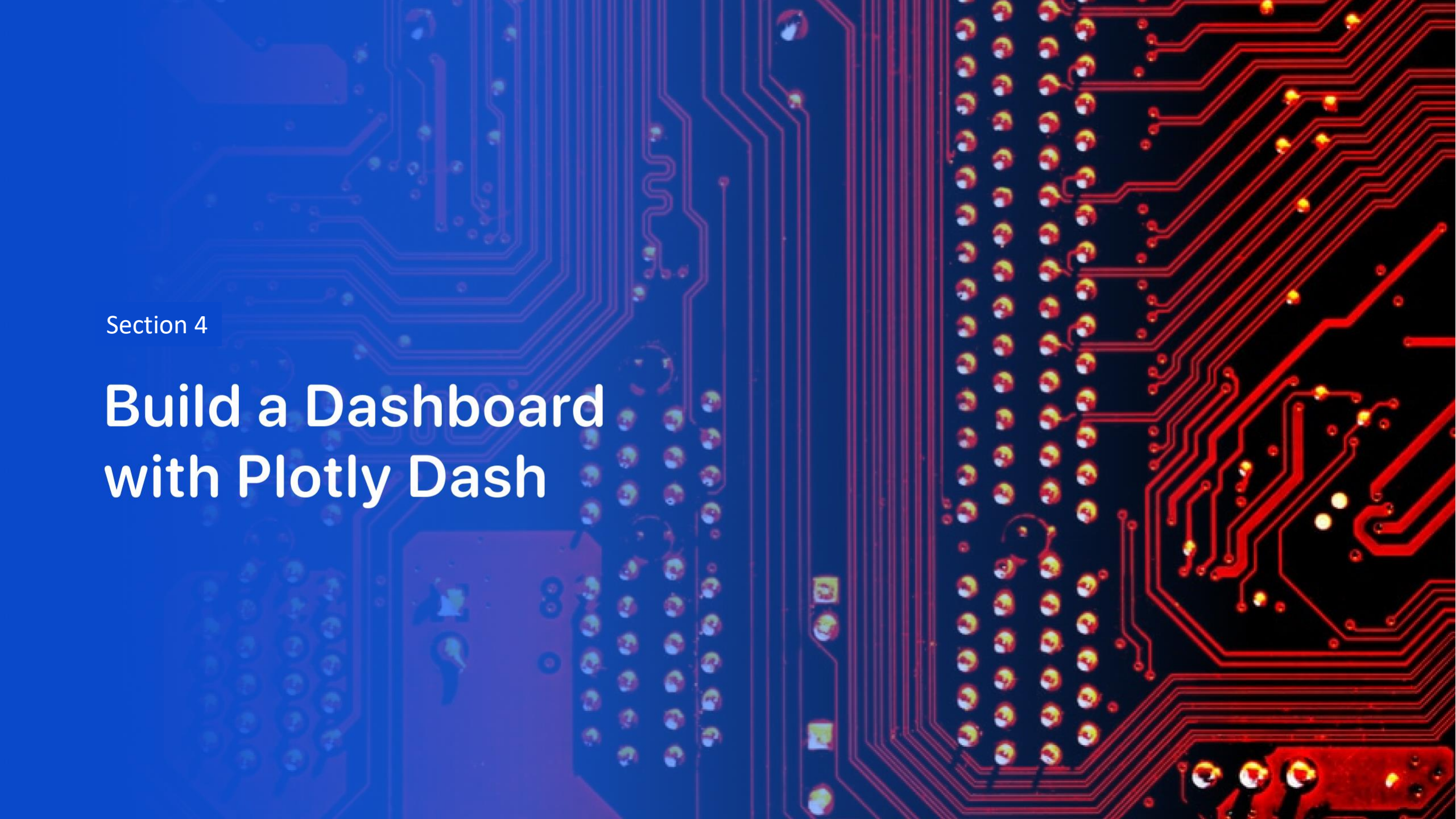
- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

<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!