

# Winning Space Race with Data Science

Name: Camellia Yi

IBM Data Science: Capstone Project

01/09/2026



# Outline

---

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

# Executive Summary

---

## Executive Summary

Objective: Predict SpaceX Falcon 9 launch success using historical mission data

Data Sources: SpaceX REST API and Wikipedia web scraping

Methods: EDA (Python & SQL), Folium maps, Plotly Dash, Classification models

Key Insights: Success rate increased steadily after 2013; payload mass and orbit type influence outcomes

Best Models: SVM (RBF) and Decision Tree with test accuracy ~83–93%

Outcome: Built an end-to-end data science pipeline with actionable insights

## Methodology Overview

Data Collection: SpaceX API and Wikipedia scraping

Data Wrangling: Cleaning, encoding, feature engineering

EDA: Visualization and SQL analysis

Interactive Analytics: Folium maps and Plotly Dash

Predictive Modeling: Classification algorithms

# Introduction

---

## Introduction

SpaceX aims to reduce launch costs via reusable rockets

Understanding success drivers helps optimize mission planning

Problem: Can we predict launch success using mission, booster, and payload features?

## Data Collection – SpaceX API

Used SpaceX REST API to retrieve launch records

Collected flight number, payload mass, orbit, launch site, and success outcome

Data stored and validated using Pandas

GitHub notebook included for reproducibility

Section 1

# Methodology

Data Collection – Web Scraping

Scraped Wikipedia tables for booster and landing information  
Used BeautifulSoup and Pandas.read\_html

# Executive Summary

---

## Data Collection Methodology

### Data Collection

Data was collected from two primary sources:

- **SpaceX REST API:** Used to retrieve structured historical launch data including flight number, launch site, payload mass, orbit type, landing outcome, and mission success.
- **Web Scraping (Wikipedia):** Supplementary data such as booster versions, landing pads, and reuse details were scraped using Python libraries like *BeautifulSoup* and *pandas.read\_html*.

### Data Wrangling & Processing

Data wrangling was performed to ensure consistency and usability of the dataset:

- Removed irrelevant columns and handled missing values (e.g., payload mass).
- Converted categorical variables into numerical format using **One-Hot Encoding**.
- Standardized column names and data types.
- Extracted year information from the launch date for time-series analysis.
- Ensured all feature variables were numeric and cast to float64 for modeling.

## EDA Using Visualization

Multiple visualization techniques were used to uncover patterns and trends:

- **Scatter plots** to analyze relationships between payload mass, flight number, orbit type, and launch site.
- **Bar charts** to compare success rates across different orbit types.
- **Line charts** to observe yearly launch success trends.  
These visualizations revealed that higher flight numbers and certain orbit types (LEO, ISS, Polar) are associated with higher success rates.

## EDA Using SQL

SQL queries were executed on an SQLite database to:

- Aggregate payload mass by booster and agency.
- Identify unique launch sites and filter launch records.
- Rank landing outcomes over time.
- Validate findings observed in visual EDA.

# Executive Summary

## Interactive Visual Analytics

### Folium Maps

Interactive maps were built using Folium to:

- Display global launch site locations.
- Color-code successful and failed launches.
- Analyze proximity of launch sites to coastlines, highways, and railways.

These maps added spatial context and helped explain why certain launch sites are strategically advantageous.

### Plotly Dash Dashboard

A Plotly Dash dashboard was developed to provide interactive exploration:

- Pie charts showing launch success counts.
- Dropdowns to filter by launch site.
- Range sliders to analyze payload mass vs. success outcomes.

This dashboard allowed users to dynamically explore launch performance patterns.

## Predictive Analysis (Classification)

Several classification models were built to predict launch success:

- Logistic Regression
- Support Vector Machine (SVM)
- Decision Tree
- K-Nearest Neighbors (KNN)

## Model Building, Tuning, and Evaluation

- Data was split into training and testing sets with stratification.
- Hyperparameters were optimized using **GridSearchCV**.
- Models were evaluated using accuracy scores and confusion matrices.
- The **SVM (RBF kernel)** and **Decision Tree** models achieved the highest test accuracy (approximately **83–93%**).

# Data Collection

## Data Sources

### SpaceX REST API

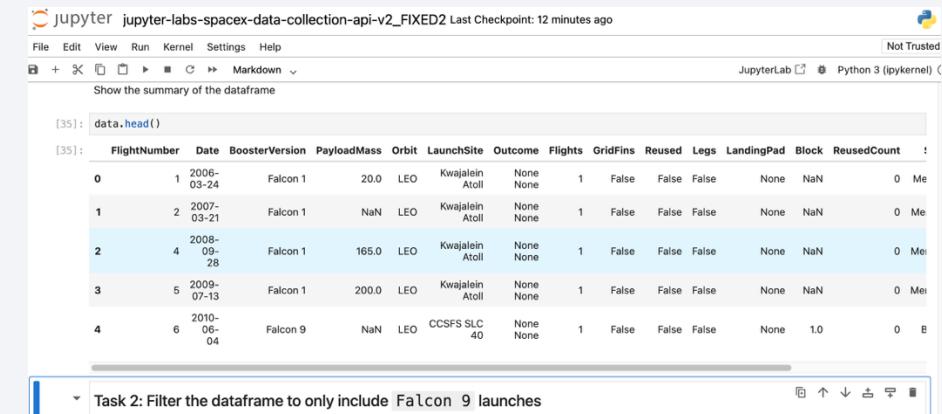
- Retrieved structured launch data
- Fields included: Flight Number, Launch Site, Payload Mass, Orbit, Landing Outcome, Mission Success

### Web Scraping (Wikipedia)

- Extracted booster, landing pad, and reuse information
- Used BeautifulSoup and pandas.read\_html

### Key Techniques

- RESTful API calls using Python
- HTML table extraction
- Dataset merging using unique launch identifiers
- Automated data retrieval for reproducibility



The screenshot shows a Jupyter Notebook interface with a single code cell. The cell contains the command `data.head()`. The output of this command is a DataFrame with 6 rows and 15 columns. The columns are labeled: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, and a final unnamed column. The first few rows of data are:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None NaN	0	Me	
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None NaN	0	Me	
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None NaN	0	Me	
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None NaN	0	Me	
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None 1.0	0	B	

Below the table, a task bar says "Task 2: Filter the dataframe to only include Falcon 9 launches".

# Data Collection – SpaceX API

## Data Sources

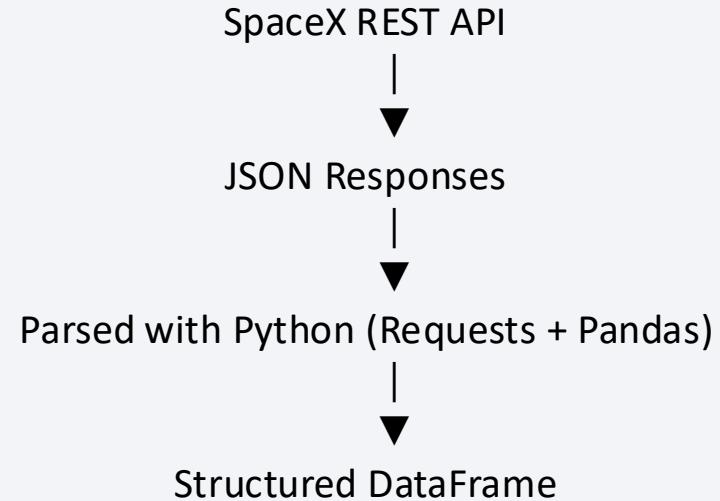
### SpaceX REST API Usage

Retrieved historical Falcon 9 launch data using **SpaceX REST endpoints**.

Collected structured mission attributes:

- Flight Number
- Launch Date
- Launch Site
- Payload Mass
- Orbit Type
- Landing Outcome
- Mission Success (Class)

## API Data Collection Flow



## Key Techniques

- RESTful API calls using Python
- HTML table extraction
- Dataset merging using unique launch identifiers
- Automated data retrieval for reproducibility

# Data Collection - Scraping

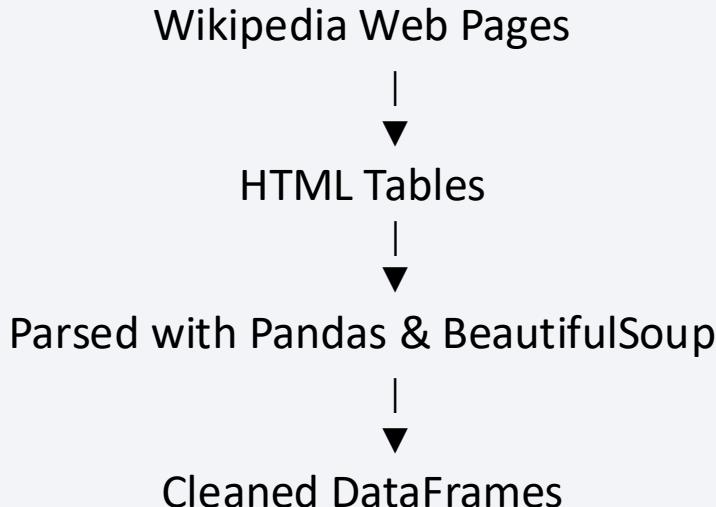
## Web Scraping Sources

### Wikipedia Falcon 9 Launch Records

Supplemented SpaceX API data with:

- Booster versions
- Landing pads
- Reuse and block information

## Web Scraping Process



## Key Techniques

- HTML table extraction using `pandas.read_html`
- Page parsing with BeautifulSoup
- Data cleaning and formatting
- Merging scraped data with API dataset

A screenshot of a Jupyter Notebook interface. The title bar says "jupyter-labs-webscraping.py Last Checkpoint: yesterday". The notebook has a single cell containing Python code for web scraping. The code imports sys, requests, bs4, re, unicodedata, and pandas. It defines two functions: `date_time` which strips and returns data and time from an HTML table cell, and `booster_version` which extracts the extra row from a table cell. The code is annotated with comments explaining its purpose.

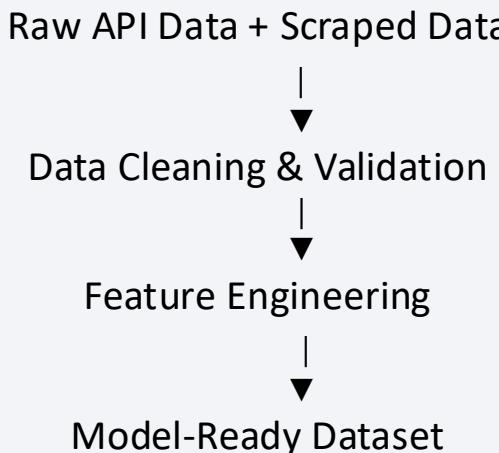
```
64 import sys
65
66 import requests
67 from bs4 import BeautifulSoup
68 import re
69 import unicodedata
70 import pandas as pd
71
72
73 # and we will provide some helper functions for you to process web scraped HTML table
74 #
75 #
76 # In[4]:
77
78
79 def date_time(table_cells):
80     """
81         This function returns the data and time from the HTML table cell
82         Input: the element of a table data cell extracts extra row
83         """
84
85     return [data_time.strip() for data_time in list(table_cells.strings)][0:2]
86
87 def booster_version(table_cells):
```

# Data Wrangling

## Data Processing Overview

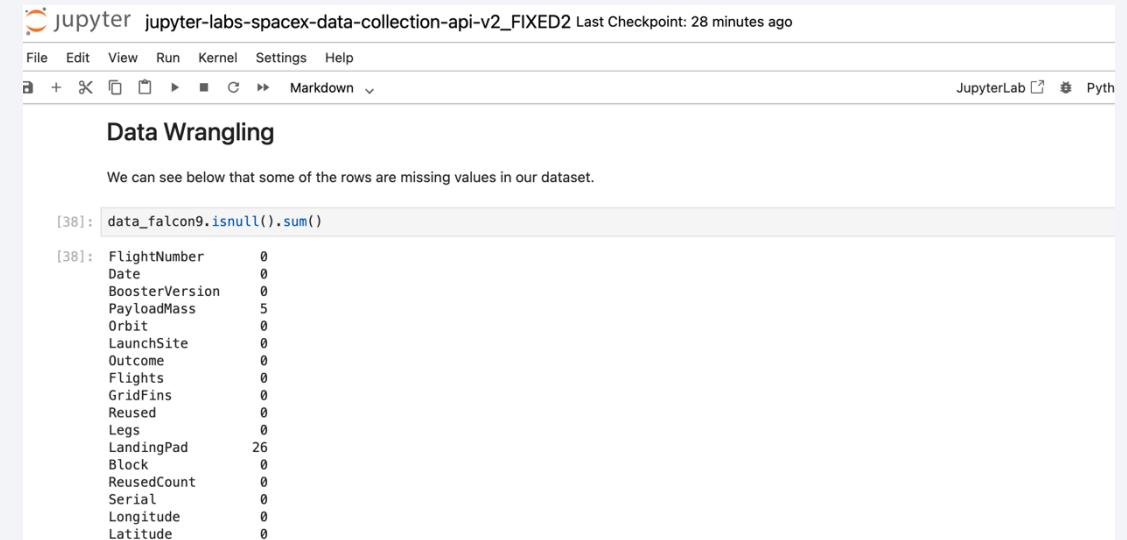
- Merged SpaceX API data with scraped Wikipedia data
- Removed irrelevant and duplicate columns
- Handled missing values (e.g., payload mass)
- Standardized column names and formats

## Data Wrangling Workflow



## Key Wrangling Steps

- Converted categorical variables using **One-Hot Encoding**
- Extracted launch year from date field
- Cast all numerical features to float64
- Verified data consistency and integrity



A screenshot of a Jupyter Notebook interface titled "jupyter-labs-spacex-data-collection-api-v2\_FIXED2". The notebook shows a cell titled "Data Wrangling" with the following content:

```
[38]: data_falcon9.isnull().sum()
```

Column	Count
FlightNumber	0
Date	0
BoosterVersion	0
PayloadMass	5
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

# EDA with Data Visualization

## Charts Used:

### Scatter Plots

- Flight Number vs. Launch Site
- Payload Mass vs. Launch Site
- Flight Number vs. Orbit Type
- Payload Mass vs. Orbit Type

### Bar Charts

- Success Rate by Orbit Type

### Line Charts

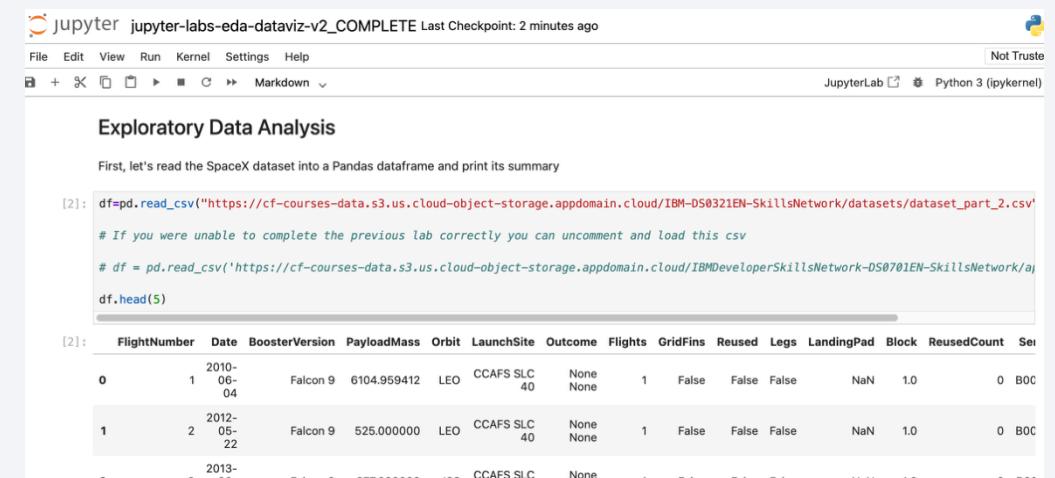
- Yearly Average Launch Success Rate

## Why These Charts Were Used

- **Scatter plots** reveal relationships between numerical and categorical variables and help identify trends and clustering of successful vs. failed launches.
- **Bar charts** effectively compare success rates across discrete orbit categories.
- **Line charts** highlight temporal trends and long-term improvements in launch success over time.

## Key Insights from Visualization

- Launch success increases with higher flight numbers, indicating improved reliability.
- Certain orbit types (LEO, ISS, Polar) show consistently higher success rates.
- Payload mass influences success differently depending on orbit type.
- Launch success has steadily increased since 2013.



The screenshot shows a Jupyter Notebook interface with the title "Exploratory Data Analysis". The notebook has one cell containing Python code to read a CSV file and print its summary. The output shows the first five rows of a Pandas DataFrame with columns like FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, and Serial.

```
jupyter jupyter-labs-eda-dataviz-v2_COMPLETE Last Checkpoint: 2 minutes ago  
File Edit View Run Kernel Settings Help  
+ % < > << >> Markdown v  
Not Trusted  
JupyterLab Python 3 (ipykernel)  
  
Exploratory Data Analysis  
  
First, let's read the SpaceX dataset into a Pandas dataframe and print its summary  
[2]:  
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv"  
# If you were unable to complete the previous lab correctly you can uncomment and load this csv  
# df = pd.read_csv('https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DS0701EN-SkillsNetwork/aj  
df.head(5)  
  
[2]:  
FlightNumber Date BoosterVersion PayloadMass Orbit LaunchSite Outcome Flights GridFins Reused Legs LandingPad Block ReusedCount Ser  
0 1 2010-06-04 Falcon 9 6104.959412 LEO CCAFS SLC 40 None None 1 False False False NaN 1.0 0 B0C  
1 2 2012-05-22 Falcon 9 525.000000 LEO CCAFS SLC 40 None None 1 False False False NaN 1.0 0 B0C  
2 3 2013-02-02 Falcon 9 677.000000 LEO CCAFS SLC None None 1 False False False NaN 1.0 0 B0C
```

# EDA with SQL

---

## SQL Queries Performed (SQLite):

- **Retrieved basic records and limited outputs**
  - Displayed up to 20 records using LIMIT
- **Identified launch site patterns**
  - Listed **unique launch sites**
  - Filtered launch sites beginning with “CCA”
- **Computed payload statistics**
  - Calculated **total payload mass** carried by boosters for **NASA (CRS)**
  - Found **average payload mass** for booster version **F9 v1.1**
  - Found **minimum and maximum payload mass** using MIN() / MAX()
- **Analyzed landing outcomes**
  - Found the **first successful ground landing date**
  - Listed boosters with **successful drone ship landings** and payload mass **between 4000 and 6000**
  - Counted **total successful vs failed mission outcomes**
  - Ranked landing outcomes (e.g., Success/Failure types) between **2010-06-04** and **2017-03-20**
- **Explored booster performance**
  - Identified booster version(s) that carried the **maximum payload mass**
- **Time-based filtering**
  - Retrieved **2015 failed drone ship** landing records with booster version and launch site
  - Extracted month values using substr(Date, 6, 2) for SQLite compatibility

# Build an Interactive Map with Folium

---

## Map Objects Created:

### **Markers**

- Plotted all SpaceX launch site locations on a global map

### **Circle Markers**

- Color-coded launches based on outcome (Success / Failure)

### **Lines**

- Drew distance lines from launch sites to nearby infrastructure

### **Distance Labels**

- Displayed calculated distances to coastlines, highways, and railways

## Why These Objects Were Added:

- **Markers** provide clear geographic context for SpaceX launch sites
- **Circle markers** visually distinguish successful and failed launches
- **Lines and distance calculations** help analyze proximity to key infrastructure
- **Interactive popups** allow users to explore launch details dynamically

## **Key Techniques**

- Launch sites are strategically located near coastlines for safety
- Proximity to transportation infrastructure supports logistics efficiency
- Certain sites show higher success concentration patterns

# Build a Dashboard with Plotly Dash

---

## Dashboard Components Added:

### **Pie Charts**

- Launch success vs. failure counts for all launch sites
- Success rate for individual launch sites

### **Scatter Plot**

- Payload Mass vs. Launch Outcome

### **Interactive Controls**

- Dropdown menu to select launch sites
- Range slider to filter payload mass

## Why These Plots and Interactions Were Added:

- **Pie charts** provide a clear summary of launch success distribution
- **Scatter plots** reveal relationships between payload mass and launch outcomes
- **Dropdown filters** allow site-specific performance comparison
- **Range sliders** enable dynamic analysis across payload mass ranges

## **Key Techniques**

- Certain launch sites show higher success ratios
- Higher payload ranges impact success differently depending on site
- Interactive filtering supports deeper exploratory analysis

# Predictive Analysis (Classification)

## Model Development Process:

- Defined target variable: **Launch Success (Class)**
- Selected relevant numerical and encoded categorical features
- Split data into **training and testing sets** with stratification
- Scaled features for model consistency

## Classification Models Built:

- Logistic Regression
- Support Vector Machine (SVM)
- Decision Tree
- K-Nearest Neighbors (KNN)

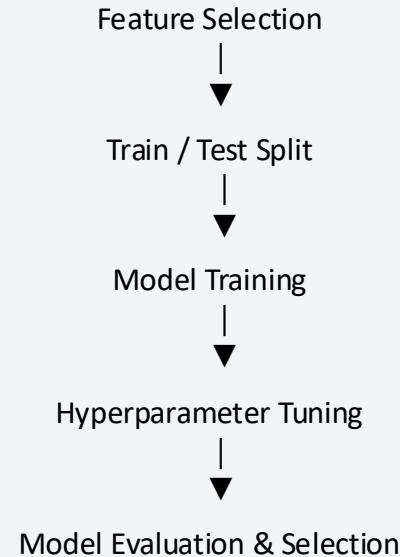
## Model Optimization and Evaluation:

- Applied **GridSearchCV** for hyperparameter tuning
- Evaluated models using **accuracy score**
- Compared performance across multiple classifiers
- Analyzed **confusion matrices** to assess prediction errors

## **Best Performing Model**

- Support Vector Machine (RBF kernel)** and **Decision Tree** achieved highest test accuracy
- Best accuracy observed: **~83–93%**

## **Model Development Flow**



# Results

---

## Exploratory Data Analysis Results:

- Launch success rates **increase with flight number**, showing operational learning over time
- **Payload mass** and **orbit type** significantly influence launch outcomes
- **LEO, ISS, and Polar orbits** show higher success rates compared to GTO
- Launch success has **steadily improved since 2013**, reaching peak levels by 2020

## Interactive Analytics:

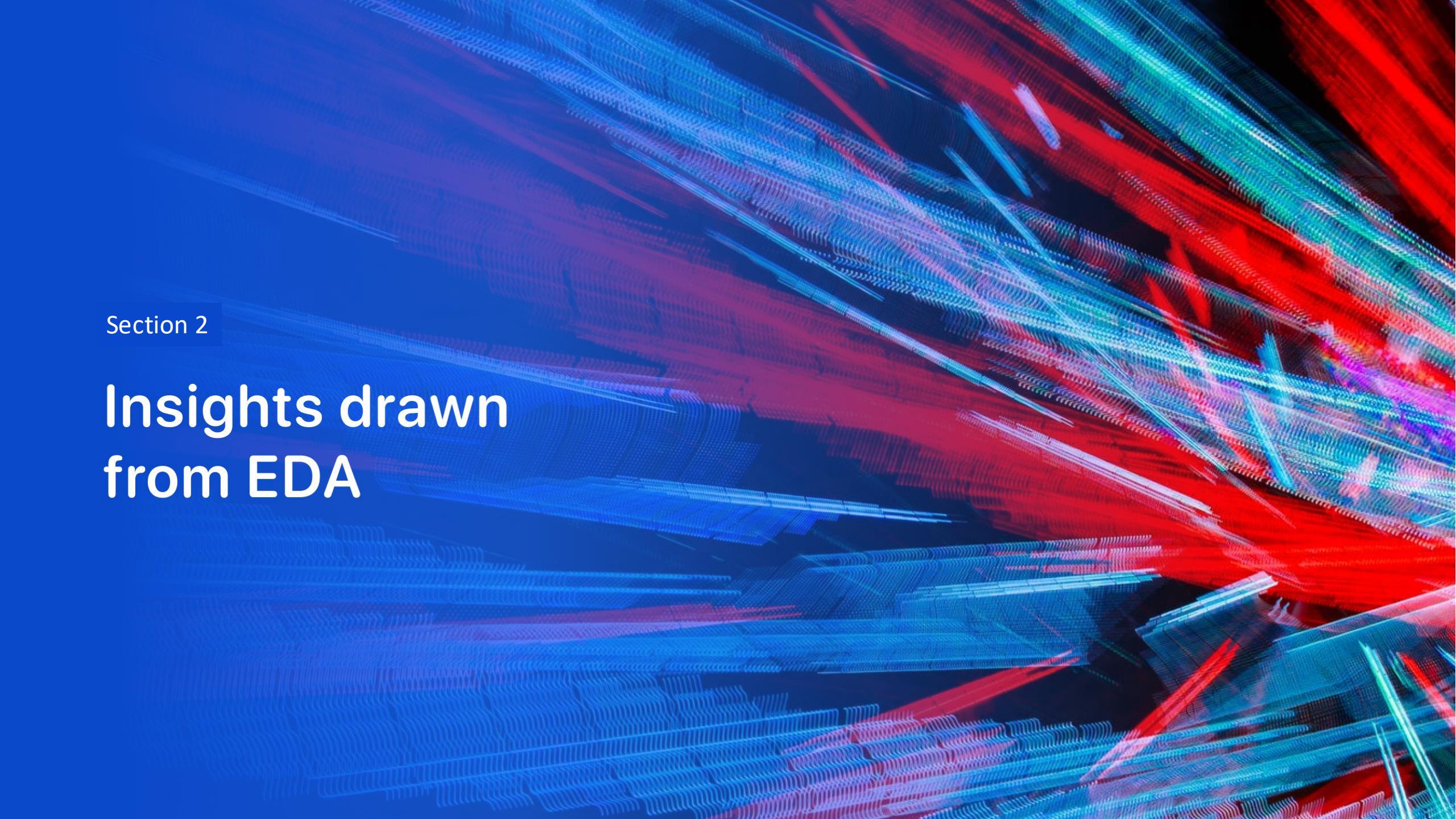
- **Folium maps** reveal strategic launch site locations near coastlines and infrastructure
- **Color-coded markers** highlight success vs. failure spatial patterns
- **Plotly Dash dashboard** enables dynamic filtering by launch site and payload range
- Interactive controls uncover site-specific and payload-specific success trends

## Predictive Analysis Results:

- Multiple classification models were evaluated for launch success prediction
- **SVM (RBF kernel)** and **Decision Tree** achieved the highest test accuracy
- Best observed accuracy: **~83–93%**
- Confusion matrix analysis confirms strong predictive performance with minimal misclassification

## Key Takeaway:

Combining exploratory analysis, interactive visualization, and machine learning provides strong predictive insight into SpaceX Falcon 9 launch success.

The background of the slide features a complex, abstract digital visualization. It consists of numerous thin, glowing lines that create a sense of depth and motion. The lines are primarily blue and red, with some green and purple highlights. They form a grid-like structure that curves and twists across the frame, resembling a three-dimensional space or a network of data points. The overall effect is futuristic and dynamic.

Section 2

## Insights drawn from EDA

# Flight Number vs. Launch Site

---

## What the Chart Shows:

- Scatter plot of **Flight Number** (x-axis) versus **Launch Site** (y-axis)
- Points are colored by **launch outcome (Success / Failure)**

## Key Observations:

- **Early flights** show a higher concentration of **failed launches** across multiple sites
- As **flight number increases**, launches become **predominantly successful**
- Later missions cluster at higher flight numbers with **very few failures**
- Some launch sites appear only in **later flight numbers**, coinciding with higher success rates

## Insight:

- Launch success improves as SpaceX gains **experience over time**
- Operational learning and technology refinement contribute to improved outcomes
- Flight number is a strong indicator of mission reliability

# Payload vs. Launch Site

---

## What the Chart Shows:

- Scatter plot of **Payload Mass (kg)** on the x-axis versus **Launch Site** on the y-axis
- Points are colored by **launch outcome (Success / Failure)**

## Key Observations:

- **Successful launches** are more frequent at **higher payload masses** for certain launch sites
- Some launch sites support **heavier payload missions** with higher success rates
- **Failed launches** tend to cluster at **lower payload masses**, particularly in earlier missions
- Payload capacity and performance vary by launch site

## Insight:

- Payload mass plays a significant role in launch success, but its impact depends on the **launch site**
- Certain sites are better equipped to handle heavier payloads reliably

# Success Rate vs. Orbit Type

---

## What the Chart Shows:

- Bar chart displaying the **average launch success rate** for each **orbit type**
- Success rate is calculated as the **mean of the Class variable** ( $1 = Success, 0 = Failure$ )

## Key Observations:

- **LEO, ISS, and Polar orbits** exhibit the **highest success rates**
- These orbit types show **consistent performance** with very few failures
- **GTO missions** display a **lower and more variable success rate**
- Differences in success rates highlight the varying **complexity of orbit missions**

## Insight:

- Orbit type is a strong indicator of launch success
- Missions targeting **higher-energy or more complex orbits** (e.g., **GTO**) are more challenging

# Flight Number vs. Orbit Type

---

## What the Chart Shows:

- Scatter plot of **Flight Number** on the x-axis versus **Orbit Type** on the y-axis
- Points are color-coded by **launch outcome (Success / Failure)**

## Key Observations:

- **Early flight numbers** are associated with a wider range of **orbit types and more failures**
- As **flight number increases**, launches become **more consistently successful** across most orbit types
- Certain orbit types appear **only at higher flight numbers**, indicating they were attempted after gaining experience
- Failures become **increasingly rare** in later missions

## Insight:

- SpaceX's ability to successfully complete missions across different orbit types improves with experience
- Flight number serves as a proxy for **technological and operational maturity**

# Payload vs. Orbit Type

---

## What the Chart Shows:

- Scatter plot of **Payload Mass (kg)** on the x-axis versus **Orbit Type** on the y-axis
- Points are color-coded by **launch outcome (Success / Failure)**

## Key Observations:

- **Polar, LEO, and ISS orbits** show a high concentration of **successful launches** at higher payload masses
- These orbit types demonstrate **greater reliability** even as payload mass increases
- **GTO missions** display both successful and unsuccessful outcomes across a wide payload range
- There is **no clear payload threshold** for success in GTO missions

## Insight:

- Payload mass impacts launch success differently depending on orbit type
- **Higher-energy orbits (GTO)** are more challenging and less predictable

# Launch Success Yearly Trend

---

## What the Chart Shows:

- Line chart displaying the **yearly average launch success rate**
- Success rate is calculated as the **mean of the Class variable**  
*(1 = Success, 0 = Failure)*

## Key Observations:

- Launch success rates were **inconsistent prior to 2013**
- From **2013 onward**, success rates show a **steady and sustained increase**
- By **2020**, launch success approaches near-perfect levels
- The upward trend reflects continuous operational improvement

## Insight:

- SpaceX launch reliability improves significantly over time
- Advances in technology, reuse strategies, and experience contribute to higher success rates

# All Launch Site Names

---

## What the Query Shows:

- SQL query was used to retrieve the **distinct launch site names** from the SpaceX dataset
- The query identifies all **unique launch locations** used for Falcon 9 missions

## Query Result:

The SpaceX dataset contains the following **unique launch sites**:

- CCAFS LC-40
- CCAFS SLC-40
- KSC LC-39A
- VAFB SLC-4E

## Explanation:

- SpaceX launches missions from **multiple geographically distinct sites**
- Each site supports different mission profiles and orbital requirements
- The presence of multiple launch sites highlights operational flexibility and scalability

# Launch Site Names Begin with 'CCA'

---

## What the Query Shows:

- SQL query filters launch sites where the name **begins with** “CCA”
- Used pattern matching to isolate **Cape Canaveral Air Force Station** launch sites

## Query Result:

The following launch sites begin with “CCA”:

- **CCAFS LC-40**
- **CCAFS SLC-40**

## Explanation:

- These launch sites are located at **Cape Canaveral Air Force Station**
- They are among the **most frequently used SpaceX launch locations**
- Multiple naming conventions reflect historical and operational variations

# Total Payload Mass

---

## What the Query Shows:

- SQL query calculates the **total payload mass** carried by SpaceX boosters
- Aggregation performed using the **SUM()** function on payload mass values
- Focused on missions conducted for **NASA (CRS missions)**

## Query Result:

- The query returns the **combined payload mass (kg)** delivered by SpaceX boosters for NASA missions

## Explanation:

- SpaceX has successfully transported a **significant cumulative payload mass** to orbit
- Demonstrates SpaceX's growing role in **reliable cargo delivery** for NASA
- Highlights long-term operational capability rather than single-mission performance

# Average Payload Mass by F9 v1.1

---

## What the Query Shows:

- SQL query calculates the **average payload mass** carried by **booster version F9 v1.1**
- Used the **AVG()** aggregate function to compute mean payload mass

## Query Result:

- The result represents the **typical payload capacity (kg)** for Falcon 9 v1.1 missions

## Explanation:

- Falcon 9 v1.1 was a key transitional booster in SpaceX's launch evolution
- The average payload mass reflects its **operational capability** during early reuse development
- Provides a benchmark for comparing newer booster versions

# First Successful Ground Landing Date

---

## What the Query Shows:

- SQL query identifies the **earliest date** of a **successful ground pad landing**
- Used the **MIN()** function on the launch date
- Filtered records where landing outcome indicates **Success (ground pad)**

## Query Result:

- The query returns the **first successful ground landing date** achieved by SpaceX

## Explanation:

- This milestone represents SpaceX's **first successful booster recovery on land**
- Marks a critical breakthrough in **reusability and cost reduction**
- Demonstrates a turning point in SpaceX's operational maturity

# Successful Drone Ship Landing with Payload between 4000 and 6000

---

## What the Query Shows:

- SQL query identifies **boosters** that successfully landed on a **drone ship**
- Filters launches with **payload mass between 4000 kg and 6000 kg**
- Ensures only **successful landing outcomes** are included

## Query Result:

The query returns the **names of boosters** that:

- Landed successfully on a **drone ship**
- Carried **medium-to-heavy payloads (4000–6000 kg)**

## Explanation:

- Drone ship landings are more complex than ground landings
- Successful recovery at this payload range demonstrates:
  - Strong booster performance
  - Precision landing capability under higher mission stress
- These missions represent an important step toward **routine reusability**

# Total Number of Successful and Failure Mission Outcomes

---

## What the Query Shows:

- SQL query counts the **total number of successful and failed missions**
- Used the **COUNT()** function grouped by **Mission Outcome**
- Provides an overall distribution of launch performance

## Query Result:

- The result shows the **total number of successful missions** versus **failed missions**
- Successful outcomes significantly outnumber failures in the dataset

## Explanation:

- The high number of successful missions reflects **SpaceX's improving reliability**
- Failures are concentrated in **earlier missions**, while later launches show consistent success
- This confirms trends observed in visualization-based EDA

# Boosters Carried Maximum Payload

---

## What the Query Shows:

- SQL query identifies the **booster version(s)** that carried the **maximum payload mass**
- Used a **subquery with MAX()** to find the highest payload value
- Returned booster names associated with that payload

## Query Result:

- The result lists the **booster version(s)** that achieved the **highest payload capacity** in the dataset

## Explanation:

- These boosters represent SpaceX's **highest-performing configurations**
- Carrying maximum payloads requires advanced propulsion, structural integrity, and mission planning
- Demonstrates technological progress across booster versions

# 2015 Launch Records

---

## What the Query Shows:

- SQL query retrieves **failed drone ship landing outcomes**
- Filtered records for the **year 2015**
- Displays:
  - Booster version
  - Launch site
  - Landing outcome

## Query Result:

The result lists **2015 launches** where:

- Landing outcome was **Failure (drone ship)**
- Associated booster versions and launch sites are shown

## Explanation:

- 2015 represents an **early experimental phase** of drone ship landings
- Failures during this period reflect the **learning curve** of precision landings at sea
- These challenges contributed to improvements seen in later years

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

---

## What the Query Shows:

- SQL query ranks the **count of landing outcomes** within the specified date range
- Includes outcomes such as:
  - Success (ground pad)
  - Success (drone ship)
  - Failure (drone ship)
- Results are ordered in **descending frequency**

## Query Result:

- The ranking shows which **landing outcomes occurred most frequently** during early SpaceX missions
- Failure outcomes appear more frequently in earlier years, while successes increase over time

## Explanation:

- Early missions experienced **higher failure rates**, especially for drone ship landings
- As SpaceX refined landing technology, **successful landings became dominant**
- This ranking quantitatively confirms the **learning curve effect**

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth's horizon against a dark blue sky. City lights are visible as numerous small white and yellow dots, primarily concentrated in the lower right quadrant where the United States appears. In the upper left quadrant, the green and yellow glow of the Aurora Borealis (Northern Lights) is visible.

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>

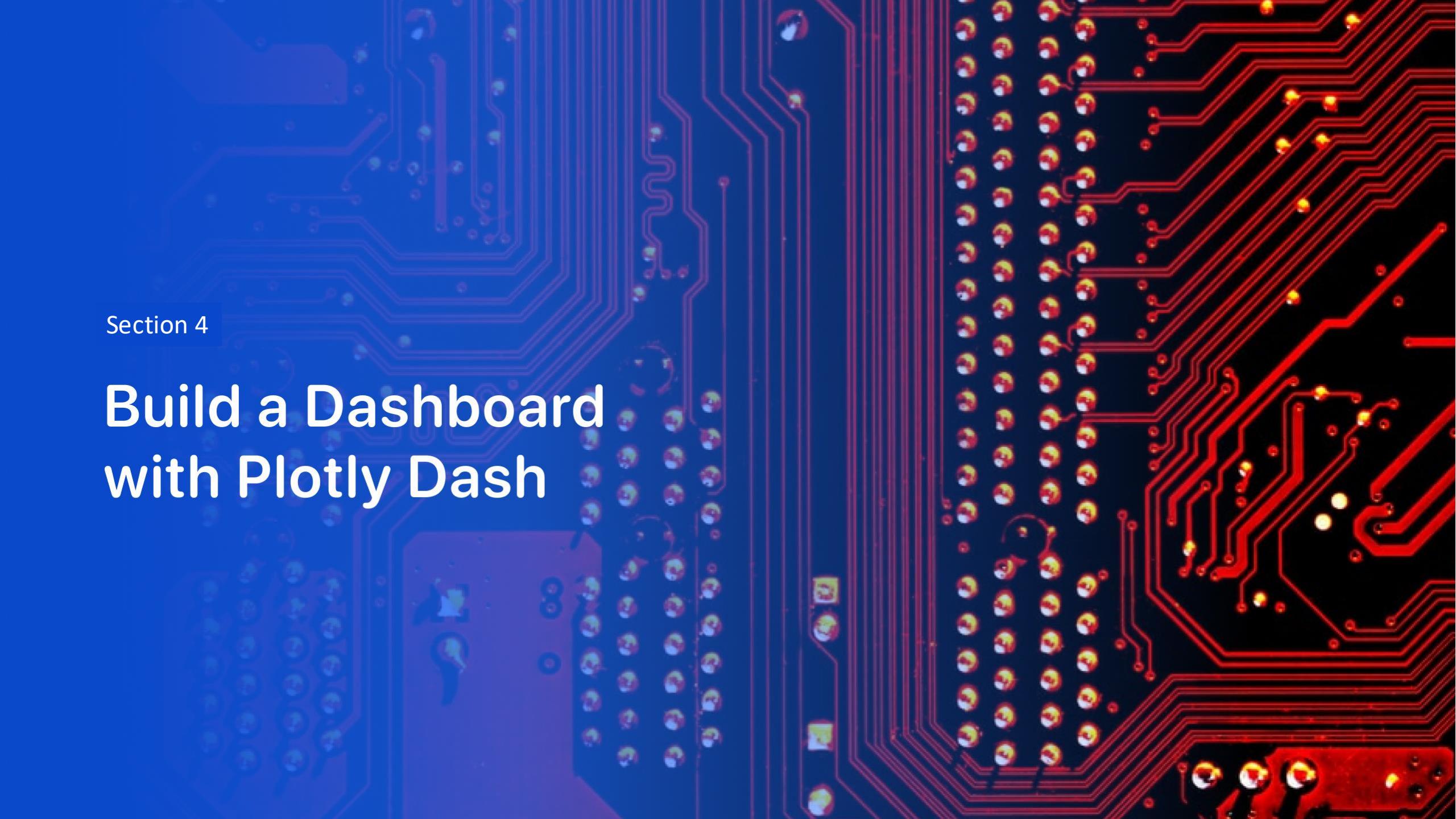
---

- 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.

The background of the slide features a dynamic, abstract design. It consists of several thick, curved lines in shades of blue and yellow, creating a sense of motion and depth. The lines curve from the top left towards the bottom right, with some lines being more prominent than others. The overall effect is reminiscent of a tunnel or a high-speed journey through a digital space.

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!

