IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Marin Postolachi
15/02/2025

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Summary of methodologies:
  - Exploratory Data Analysis
  - Visual Analytics and Dashboards
  - Machine Learning Prediction
- Summary of all results:
  - Determination of the corelation between different attributes of the dataset
  - Creation of prediction models

Introduction

- Project Background and Context
  - Economic Context:
    - SpaceX advertises Falcon 9 rocket launches at a cost of 62 million dollars.
    - Other providers charge upwards of 165 million dollars per launch.
    - The significant cost savings come from SpaceX's ability to reuse the first stage.
  - Project Goal:
    - Develop a machine learning pipeline to predict whether the first stage will land successfully.
    - This prediction can be used to estimate launch costs and give an edge to any company looking to bid against SpaceX.

- Key Questions to Address:
  - What factors determine whether the first stage will land successfully?
  - How do the interactions between various features influence the success rate?
  - What operating conditions must be in place to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was imported directly from SpaceX's official API.
  - This provided comprehensive and up-to-date information on Falcon 9 launches.
- Perform data wrangling:
  - Raw data was cleaned and transformed to ensure consistency and accuracy.
  - Data processing steps involved handling missing values, normalizing formats, and structuring data for analysis.

Methodology

- Perform exploratory data analysis (EDA) using visualization and SQL
    - Utilized visualization tools and SQL queries to uncover trends and insights.
    - Key variables and relationships were identified through detailed exploratory analysis.
- Perform interactive visual analytics using Folium and Plotly Dash
    - Created dynamic maps with Folium to visualize launch locations and landing zones.
    - Developed interactive dashboards using Plotly Dash for real-time data exploration.

Methodology

- Perform predictive analysis using classification models:
  - Built classification models to predict the success of first stage landings.
  - The modeling process included:
  - Building and tuning models.
  - Evaluating model performance to ensure reliable predictions.

Data Collection

- Accessing the Official API:
  - We connected to SpaceX's official API to retrieve data in JSON format.
- Cleaning and Structuring:
  - The data was filtered, cleaned for missing values, and transformed into a structured format.
- Storing and Preparing:
  - Cleaned data was stored systematically to support both exploratory analysis and predictive modeling.

```python
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
        if x:
            response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
            BoosterVersion.append(response['name'])
```

Python

```python
# Calculate the mean value of PayloadMass column

payload_mean = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value

data_falcon9.loc[:, 'PayloadMass'] = data_falcon9['PayloadMass'].fillna(payload_mean)
```

Python

Data Collection – SpaceX API

- Official API Access: Using SpaceX's REST API to obtain launch data.
- GET Requests & JSON: Retrieving data in JSON format via HTTP GET calls.
- Data Wrangling: Cleaning, filtering, and transforming raw JSON data.
- Structuring & Storage: Converting the JSON into a structured Pandas DataFrame for analysis.
- Supplementary API Calls: Using additional API endpoints (rockets, launchpads, payloads, cores) to enrich the dataset.
- https://github.com/marin1109/Applied-Data-Science-Capstone

1. Start
2. Connect to SpaceX API
3. Send GET Request to /launches/past
4. Retrieve Raw JSON Data
5. Data Wrangling & Filtering
6. Make Supplementary API Calls
7. Transform & Structure Data
8. Store Data for Analysis
9. End

Data Collection - Scraping

- HTTP GET Request: Retrieve the Wikipedia page snapshot using the Requests library.
- HTML Parsing: Use BeautifulSoup to parse the HTML content.
- Table Extraction: Locate and extract the launch records from the HTML table.
- Data Cleaning & Transformation: Parse table rows, extract column names and cell values, and convert data into a Pandas DataFrame.
- Export Data: Save the cleaned data as a CSV for further analysis.

1. Start
2. Send GET Request to Wikipedia URL
3. Receive HTML Response
4. Parse HTML with BeautifulSoup
5. Locate Target HTML Table
6. Extract Column Headers & Table Rows
7. Clean & Process Data
8. Convert Data into Pandas DataFrame
9. Export Data to CSV
10. End

Data Wrangling

- **Data Cleaning:**
  - Handling missing values, filtering irrelevant data, and correcting inconsistencies.
- **Data Transformation:**
  - Converting data types (e.g., dates), parsing strings, and structuring data into a usable format.
- **Feature Engineering:**
  - Creating new features (e.g., training labels) and extracting key metrics for analysis.
- **Exploratory Data Analysis (EDA):**
  - Analyzing distributions, counting occurrences, and summarizing data statistics.

EDA with Data Visualization

Flight Number vs. Payload Mass (with Outcome):

Purpose: To see how flight number and payload mass affect landing success.

Flight Number vs. Launch Site:

Purpose: To examine site performance trends.

Payload Mass vs. Launch Site:

Purpose: To explore payload distribution across launch sites.

Success Rate by Orbit (Bar Chart):

Purpose: To compare landing success rates among different orbits.

Flight Number vs. Orbit:

Purpose: To assess if flight number influences success in various orbits.

Payload Mass vs. Orbit:

Purpose: To reveal how payload mass impacts success across orbits.

Yearly Trend of Launch Success (Line Chart):

Purpose: To track improvements in landing success over time.

EDA with SQL

Unique Launch Sites:

Used SELECT DISTINCT to list all unique launch sites.

Launch Sites Starting with 'CCA':

Filtered records using LIKE 'CCA%' and LIMIT 5.

Total Payload Mass for NASA Boosters:

Applied SUM on payload mass for records where customer is NASA.

Average Payload Mass for F9 Boosters:

Calculated average payload mass using AVG for booster version F9.

Earliest Successful Ground Pad Landing Date:

Retrieved the minimum date using MIN for successful landings.

Counting Success vs. Failure Outcomes:

Used COUNT with CASE WHEN to tally successful and failed mission outcomes.

Ranking Landing Outcomes:

Employed RANK() over outcome counts to rank landing outcomes between specific dates.

Build an Interactive Map with Folium

Markers:

Folium Marker & DivIcon: Added markers at launch site coordinates to display site names as text labels.

Purpose: Quickly identify the precise location of each launch site on the map.

Circles:

Folium Circle: Placed circles around launch site coordinates to highlight their areas.

Purpose: Visually emphasize launch sites and provide a clear area of influence.

Marker Clusters:

MarkerCluster Plugin: Grouped markers for launch outcomes (success or failure) to manage overlapping points.

Purpose: Improve map readability by clustering markers when many launches occur at similar coordinates.

Lines (PolyLine):

Folium PolyLine: Drew lines between launch sites and nearby points of interest (e.g., coastlines, cities, railways, highways).

Purpose: Illustrate the distances between launch sites and key landmarks, aiding in geographical analysis.

Build a Dashboard with Plotly Dash

Success Pie Chart:

What: Displays launch success rates.

Interactions:

Dropdown Filter: When "All Sites" is selected, it shows the total success launches by each site. When a specific site is selected, it shows the proportion of successful vs. failed launches for that site.

Why: Helps users quickly gauge overall performance and compare individual site outcomes.

Payload-Success Scatter Plot:

What: Illustrates the relationship between payload mass and launch success.

Interactions:

Dropdown Filter: Filters data by launch site.

Range Slider: Adjusts the payload mass range displayed on the plot.

Why: Allows users to explore how payload mass impacts the success rate and see correlations by booster version category.

Predictive Analysis (Classification)

Data Preparation:

- Standardized the dataset using StandardScaler.
- Split the data into training and test sets (80% training, 20% testing).

Model Building & Tuning:

- Developed multiple classifiers: Logistic Regression, Support Vector Machine, Decision Tree, and K-Nearest Neighbors.
- Employed GridSearchCV with 10-fold cross-validation to tune hyperparameters for each model.

Model Evaluation:

- Assessed performance using accuracy scores and confusion matrices on the test set.
- Compared results across models to determine which provided the best predictive performance.

Model Selection:

- Selected the best performing model based on the highest test accuracy and error analysis from confusion matrices.

Results

Exploratory Data Analysis Results:

- Identified key correlations between flight number, payload mass, launch site, and landing success.
- Visualized success rates across different orbits and launch sites, revealing geographical and operational patterns.

Interactive Analytics Demo (Screenshots):

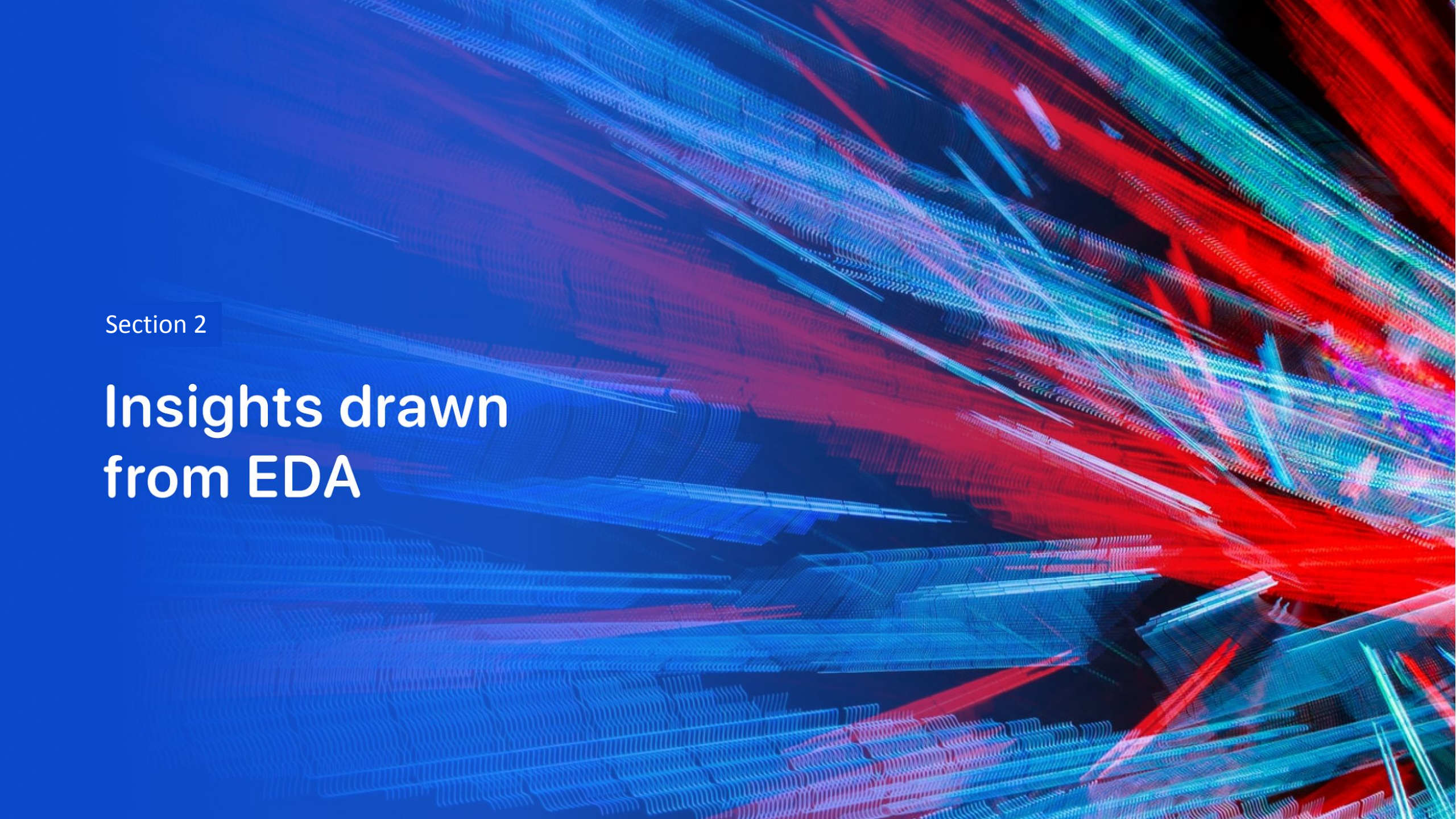
- Developed interactive maps with Folium displaying launch sites and their proximities (coastlines, railways, cities).
- Created dynamic dashboards with Plotly Dash that allow users to filter and explore launch records by site and payload.

Predictive Analysis Results:

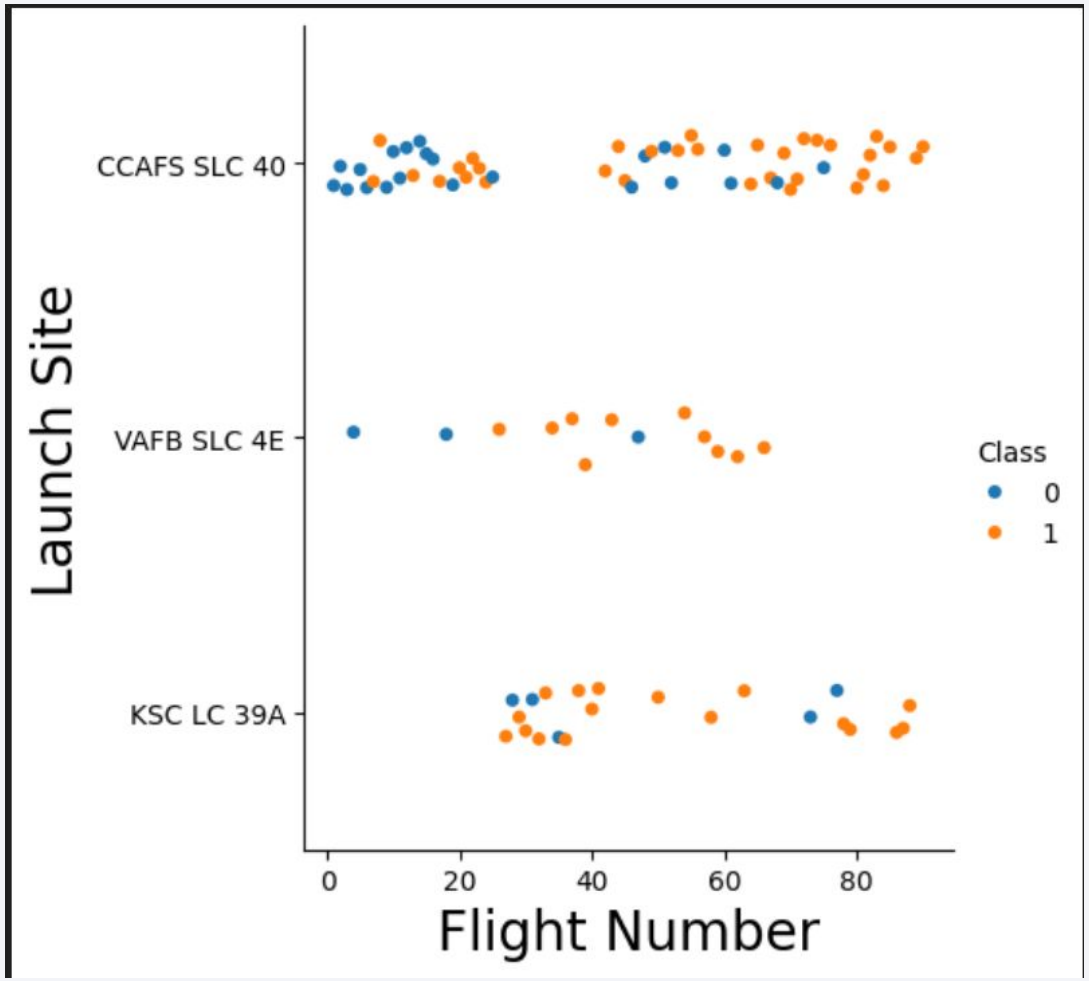
- Built and tuned multiple classification models (Logistic Regression, SVM, Decision Tree, KNN) using GridSearchCV with cross-validation.
- Selected the best performing model based on test accuracy and confusion matrix analysis, effectively predicting Falcon 9 first stage landing success.

Section 2

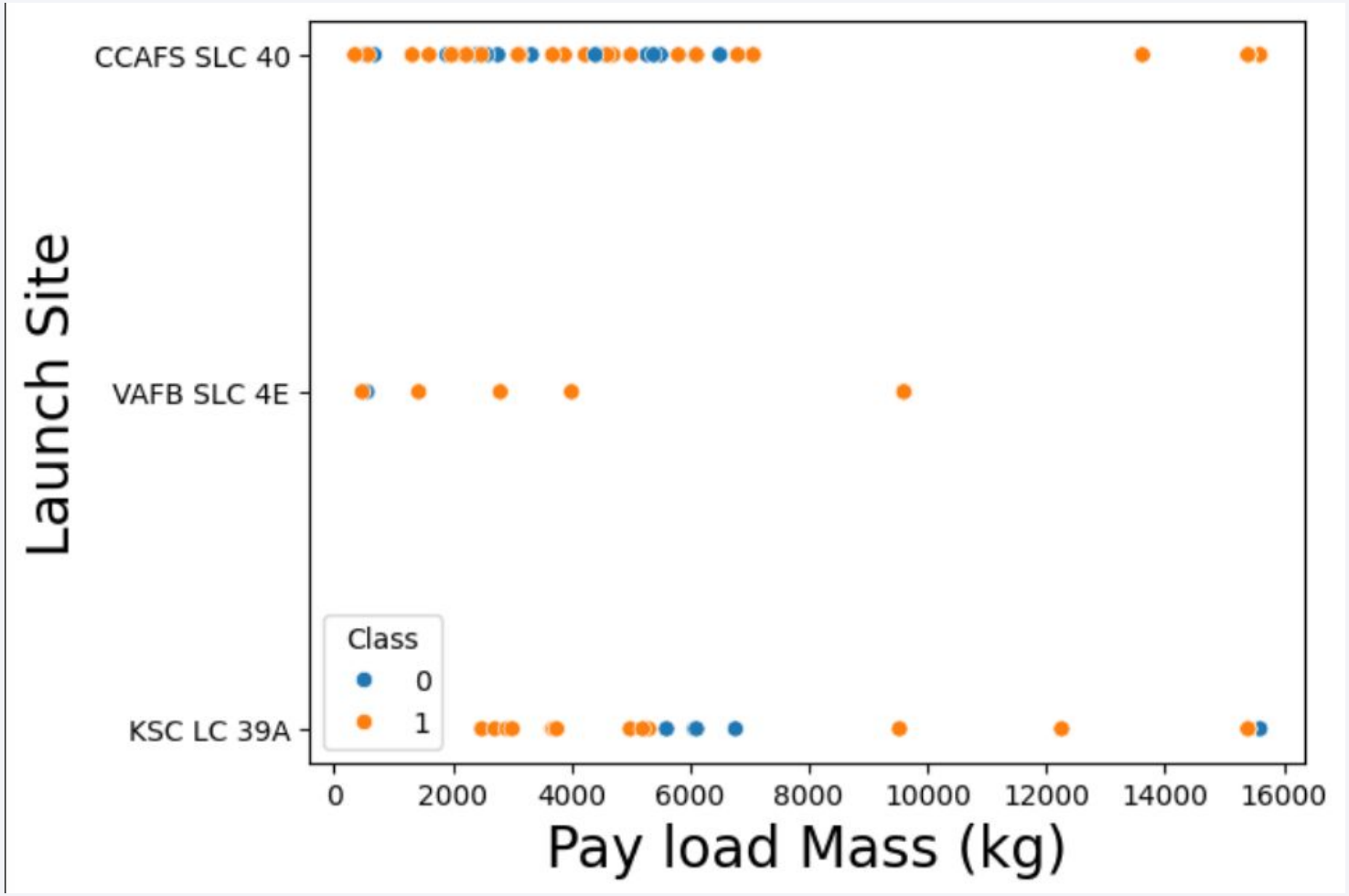
Insights drawn from EDA



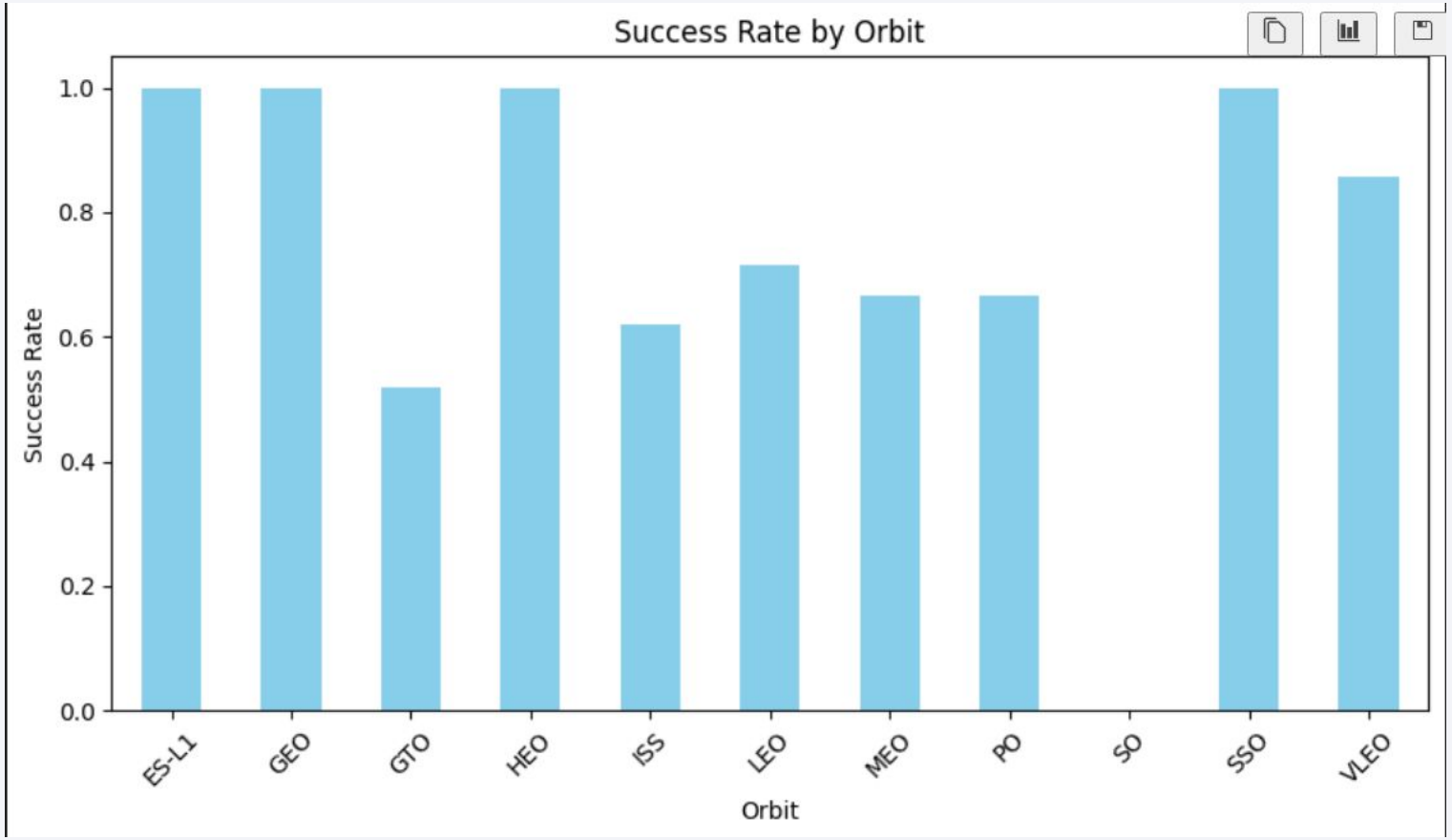
Flight Number vs. Launch Site



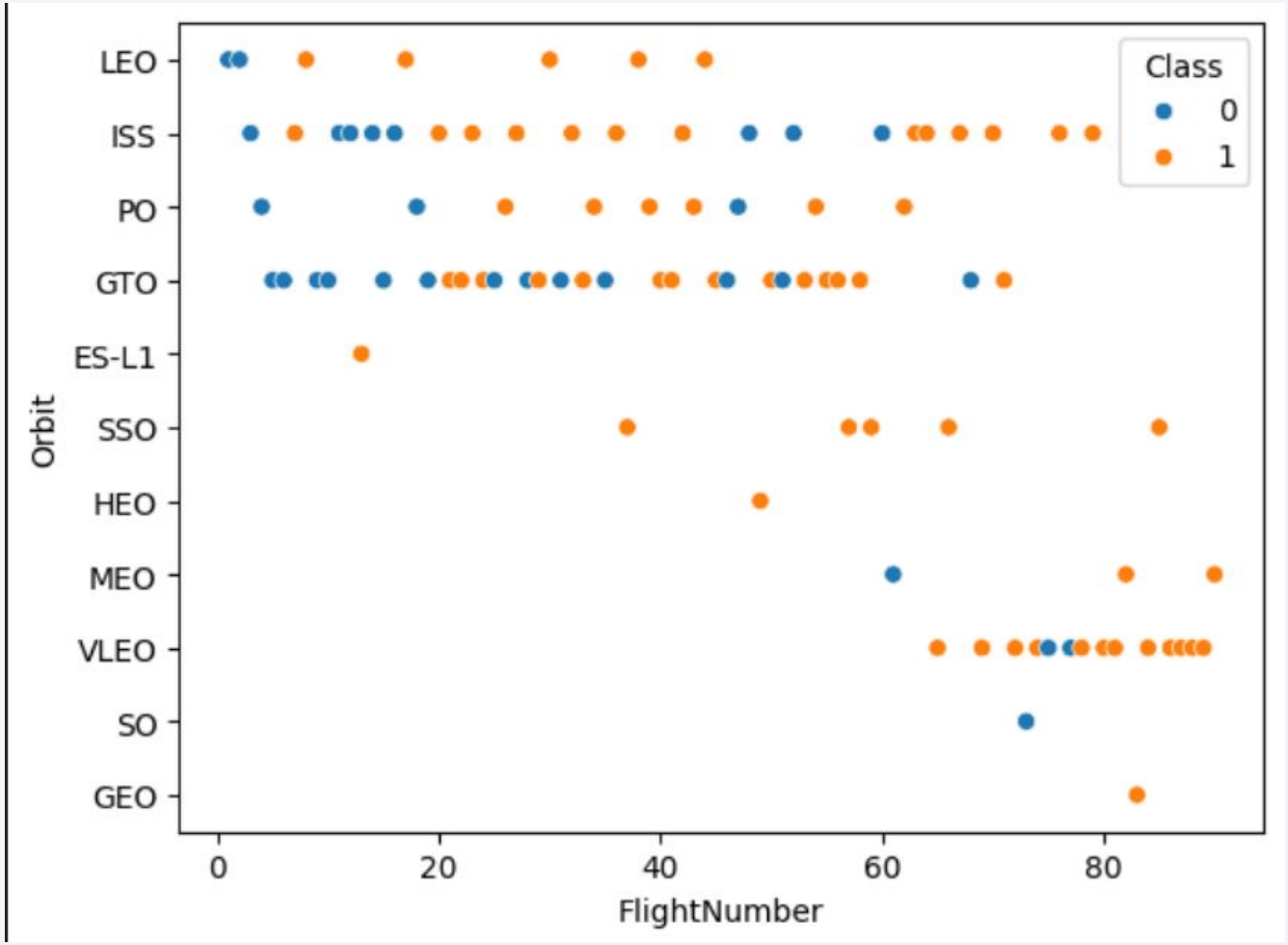
Payload vs. Launch Site



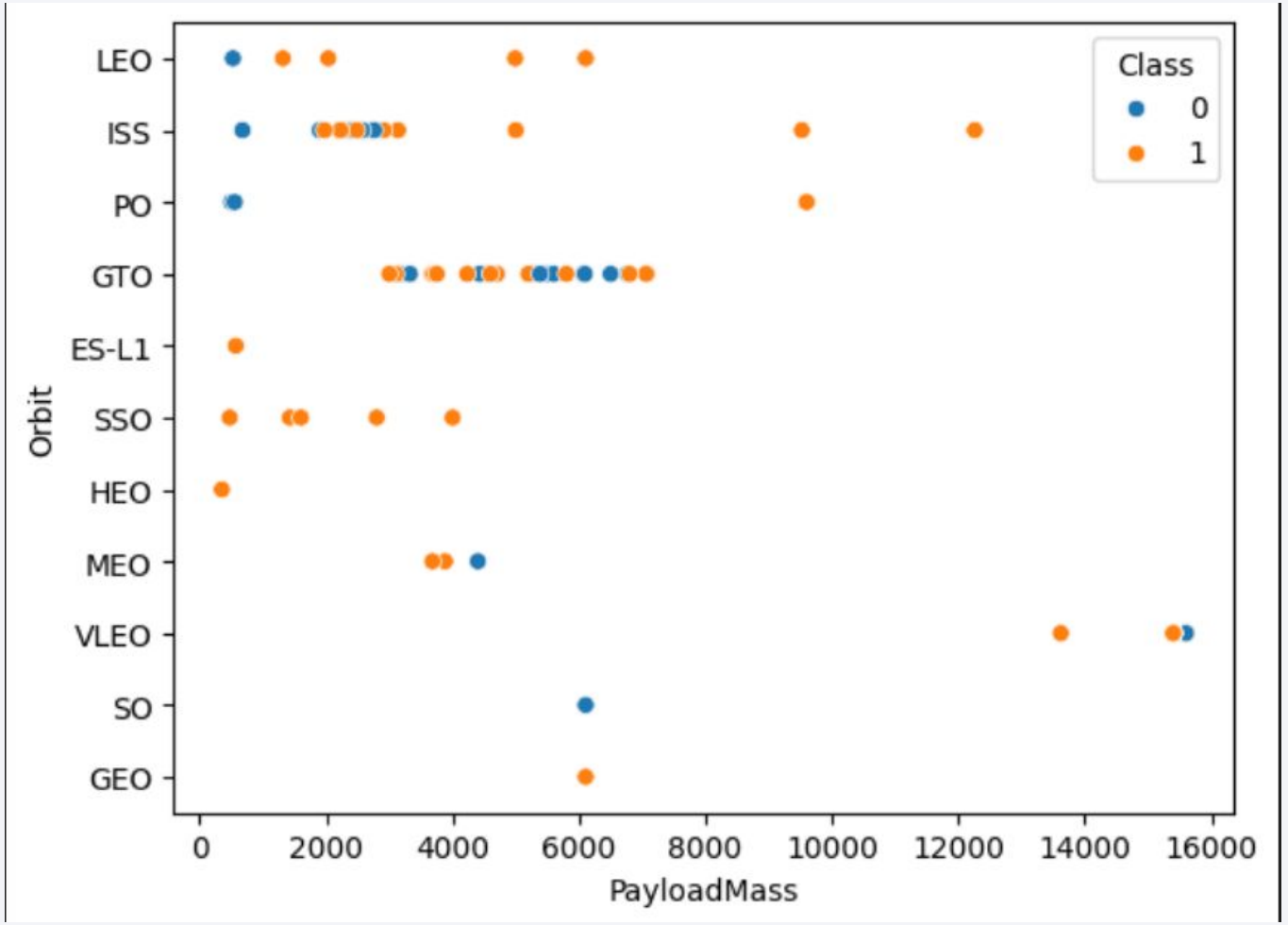
Success Rate vs. Orbit Type



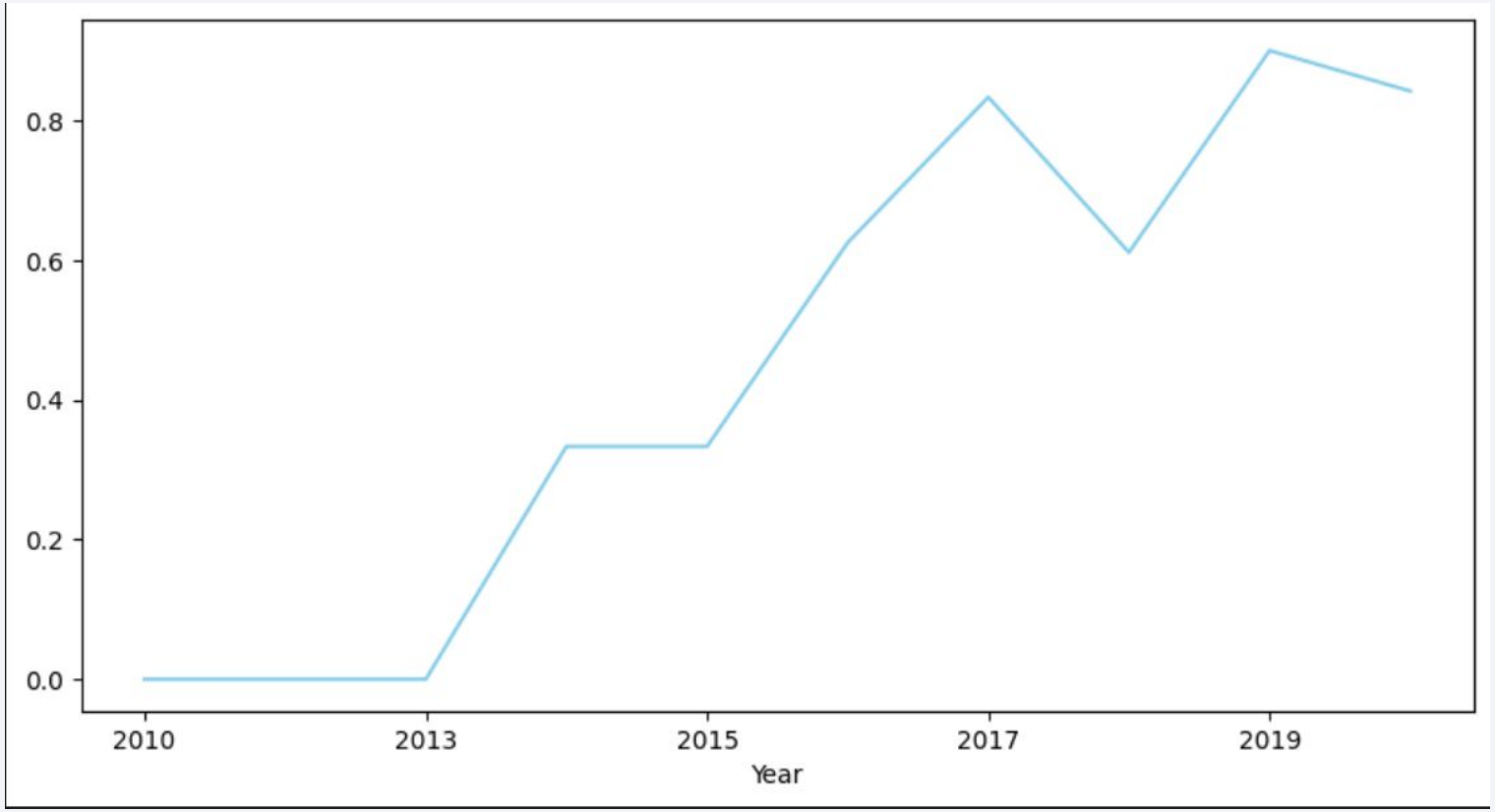
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

Launch Site Names Begin with 'CCA'

```
%sql SELECT Launch_Site FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

0.0s

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer LIKE 'NASA%'
```

✓ 0.0s

```
 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
99980

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9%'
```

✓ 0.0s

```
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
6138.287128712871

First Successful Ground Landing Date

```
%sql SELECT MIN(Date) FROM SPACEXTABLE
```

✓ 0.0s

```
 * sqlite:///my_data1.db
Done.
```

MIN(Date)
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql SELECT * FROM SPACEXTABLE
WHERE Mission_Outcome LIKE 'Success' AND PAYLOAD_MASS__KG_ < 6000 AND PAYLOAD_MASS__KG_ > 4000
```

✓ 0.0s Python

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission
2014-08-05	8:00:00	F9 v1.1	CCAFS LC-40	AsiaSat 8	4535	GTO	AsiaSat	
2014-09-07	5:00:00	F9 v1.1 B1011	CCAFS LC-40	AsiaSat 6	4428	GTO	AsiaSat	
2015-03-02	3:50:00	F9 v1.1 B1014	CCAFS LC-40	ABS-3A Eutelsat 115 West B	4159	GTO	ABS Eutelsat	
2015-04-27	23:03:00	F9 v1.1 B1016	CCAFS LC-40	Turkmen 52 / MonacoSAT	4707	GTO	Turkmenistan National Space Agency	
2016-03-04	23:35:00	F9 FT B1020	CCAFS LC-40	SES-9	5271	GTO	SES	
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	
2017-								

Total Number of Successful and Failure Mission Outcomes

```
%%sql SELECT
    COUNT(CASE WHEN Mission_Outcome LIKE 'Success' THEN 1 END) AS Success_Count,
    COUNT(CASE WHEN Mission_Outcome NOT LIKE 'Success' THEN 1 END) AS Failure_Count
FROM SPACEXTABLE;
```

✓ 0.0s

```
 * sqlite:///my_data1.db
Done.
```

Success_Count	Failure_Count
98	3

Boosters Carried Maximum Payload

```
%%sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE
    WHERE PAYLOAD_MASS__KG_ IN (
        SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
)
```

✓ 0.0s Python

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

```
%%sql SELECT
    CASE
        WHEN substr(Date, 6, 2) = '01' THEN 'January'
        WHEN substr(Date, 6, 2) = '02' THEN 'February'
        WHEN substr(Date, 6, 2) = '03' THEN 'March'
        WHEN substr(Date, 6, 2) = '04' THEN 'April'
        WHEN substr(Date, 6, 2) = '05' THEN 'May'
        WHEN substr(Date, 6, 2) = '06' THEN 'June'
        WHEN substr(Date, 6, 2) = '07' THEN 'July'
        WHEN substr(Date, 6, 2) = '08' THEN 'August'
        WHEN substr(Date, 6, 2) = '09' THEN 'September'
        WHEN substr(Date, 6, 2) = '10' THEN 'October'
        WHEN substr(Date, 6, 2) = '11' THEN 'November'
        WHEN substr(Date, 6, 2) = '12' THEN 'December'
    END AS Month_Name,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM SPACEXTABLE
WHERE substr(Date, 0, 5) = '2015'
AND Landing_Outcome LIKE '%drone ship%';
```

✓ 0.0s

```
 * sqlite:///my_data1.db
Done.
```

Month_Name	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
June	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

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

```
%%sql SELECT
    Landing_Outcome,
    COUNT(*) AS Outcome_Count,
    RANK() OVER (ORDER BY COUNT(*) DESC) AS Rank
FROM SPACEXTABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY Outcome_Count DESC;
```

✓ 0.0s

```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Outcome_Count	Rank
No attempt	10	1
Success (drone ship)	5	2
Failure (drone ship)	5	2
Success (ground pad)	3	4
Controlled (ocean)	3	4
Uncontrolled (ocean)	2	6
Failure (parachute)	2	6
Precluded (drone ship)	1	8


Section 3
Launch Sites
Proximities Analysis

<Folium Map Screenshot 1>

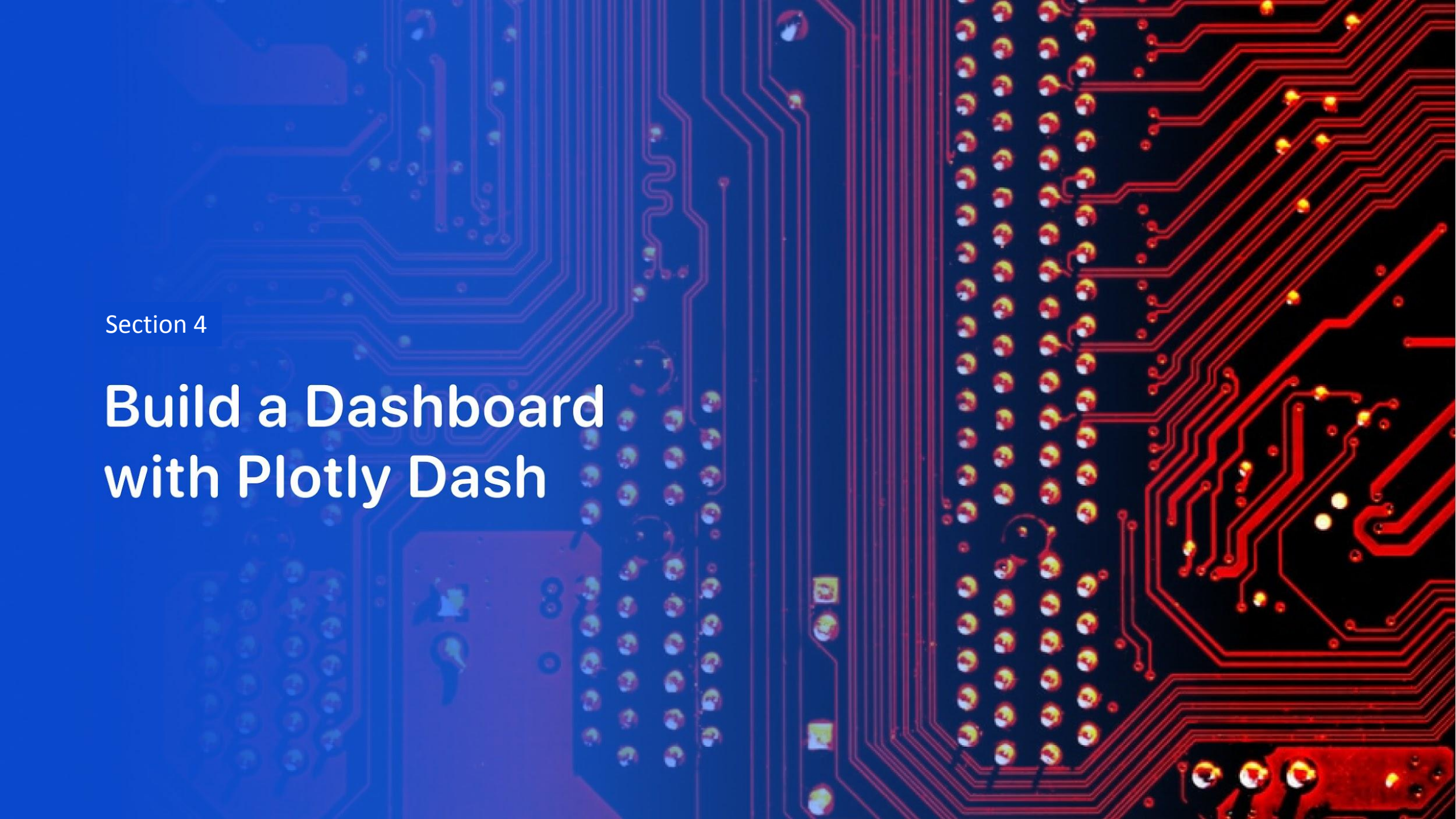
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

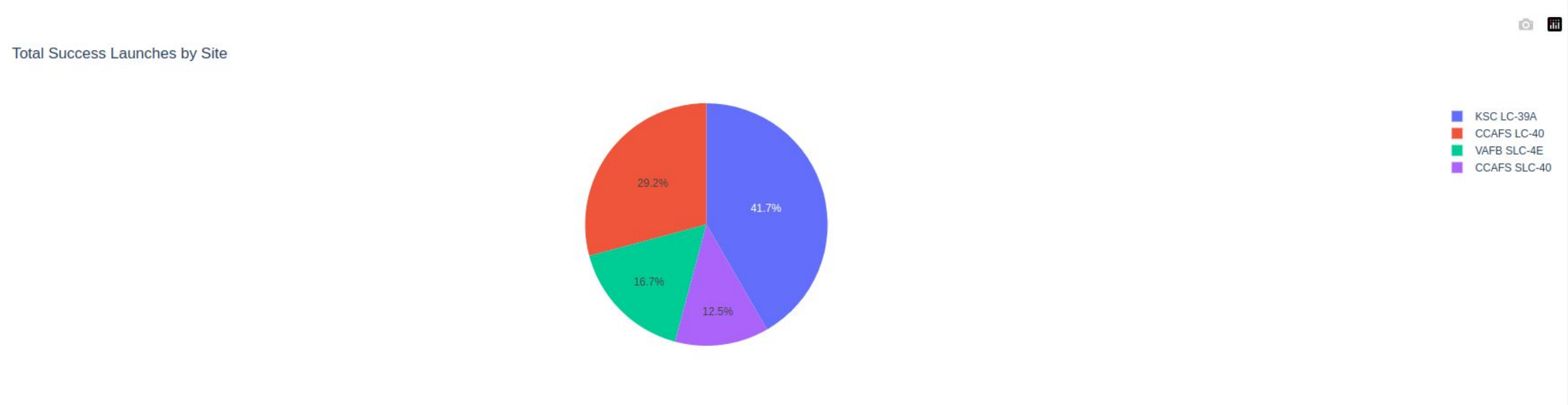
- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

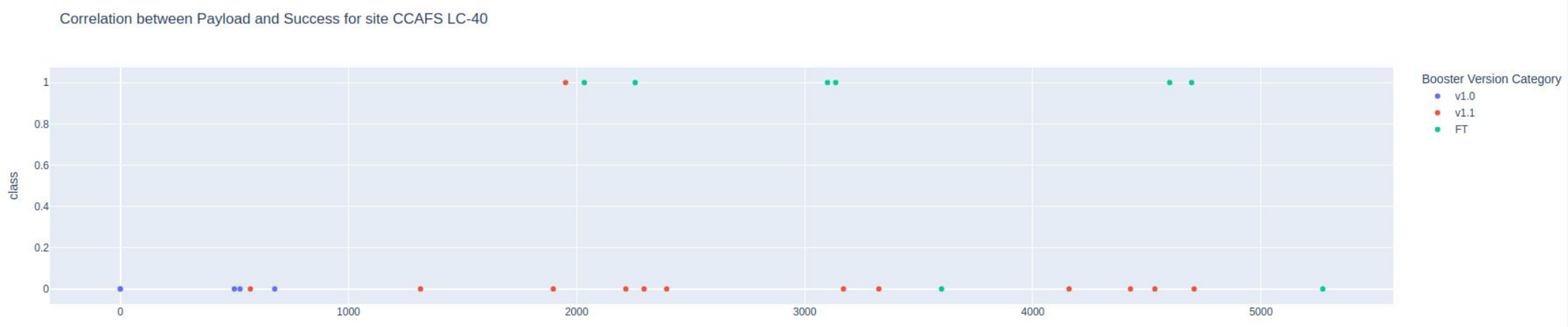
- Replace <Folium map screenshot 3> title with an appropriate title
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- Explain the important elements and findings on the screenshot


Section 4
Build a Dashboard with Plotly Dash

Dashboard: Launch Success Distribution by Site



Dashboard: Payload vs. Launch Outcome Scatter Plot



Conclusions

- **Data Insights:**
  - EDA revealed clear patterns between payload mass, flight number, and landing success, with certain launch sites and orbit types exhibiting higher success rates.
- **Interactive Analytics:**
  - Interactive maps and dashboards provided intuitive visualization of launch site locations and their proximities to key landmarks, enhancing understanding of geographical factors in launch success.
- **Predictive Analysis:**
  - Through model tuning and evaluation, the best performing classifier was identified, demonstrating the feasibility of predicting Falcon 9 first stage landing outcomes with high accuracy.
- **Overall Impact:**
  - The combined analysis and modeling approach can assist alternative companies in estimating launch costs and optimizing operational strategies by predicting landing success reliably.

Thank you!