IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pedro Marquez
06-Feb-20

Outline

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

Executive Summary

- Summary of Methodologies
  - Collected historical launch data from SpaceX using public APIs and web scraping techniques.
  - Cleaned, merged, and transformed datasets using Python and SQL to create an analysis-ready dataset.
  - Performed exploratory data analysis (EDA) with SQL queries and data visualizations to identify key relationships.
  - Built interactive visual analytics (maps and dashboards) to examine launch site performance and orbit distributions.
  - Trained and evaluated multiple machine-learning classification models to predict first-stage landing success.
  - Selected the best-performing model based on accuracy and validation performance.
- Summary of Results
  - Payload mass, orbit type, launch site, and booster version show strong influence on landing success probability.
  - Certain launch sites and orbit categories demonstrate consistently higher success rates.
  - Machine-learning models achieve strong predictive performance for landing success classification.
  - The final model demonstrates that historical mission parameters can reliably predict landing outcomes.
  - Findings support data-driven mission planning and cost-reduction strategies.

Introduction

- Project Background and Context
  - Reusable rocket technology is a key driver of cost reduction in the commercial space industry.
  - SpaceX pioneered routine first-stage booster recovery, enabling rapid reuse of launch vehicles.
  - Predicting whether a Falcon 9 first stage will successfully land is critical for mission planning and cost estimation.
  - Historical launch data provides an opportunity to apply data science techniques to understand and predict landing success.
- Problems You Want to Find Answers
  - Which factors most strongly influence first-stage landing success?
  - How do payload mass, orbit type, launch site, and booster version affect outcomes?
  - Are there observable patterns in launch success across different mission profiles?
  - Can machine-learning models accurately predict landing success using historical data?
  - How can these predictions support operational and financial decision-making?

Section 1

Methodology

Methodology

Executive Summary

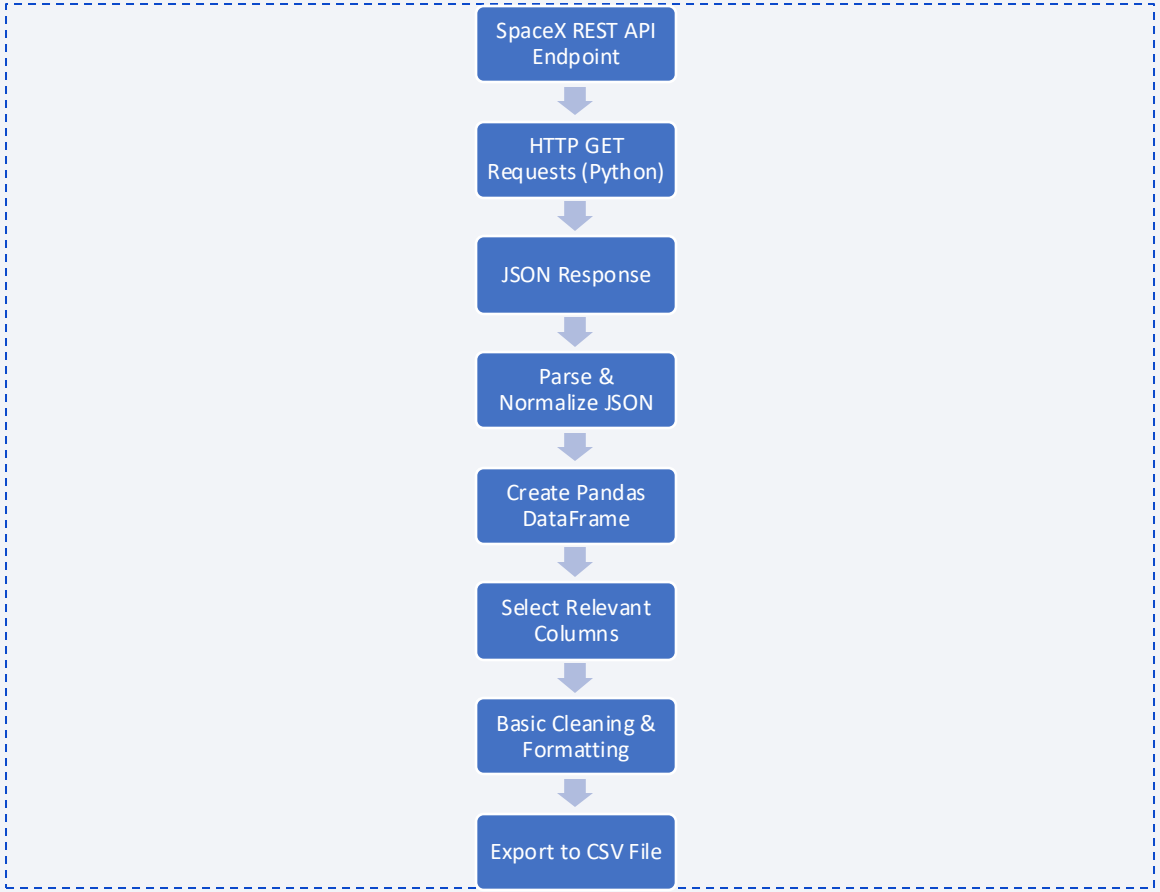
- Data collection methodology:
  - Collected historical launch data from SpaceX using public APIs and web scraping techniques.
- Perform data wrangling
  - Cleaned, merged, and transformed datasets using Python and SQL to create an analysis-ready dataset
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Trained and evaluated multiple machine-learning classification models to predict first-stage landing success.
  - Selected the best-performing model based on accuracy and validation performance.

Data Collection

- How the Data Sets Were Collected
  - Launch data retrieved from SpaceX REST API
  - Supplementary launch and booster information gathered via web scraping from public webpages
  - Data stored in CSV format and SQLite database
  - Multiple datasets merged into a unified master dataset
  - Data validated and cleaned for missing values and inconsistencies

Data Collection – SpaceX API

- SpaceX REST API Data Collection
  - Accessed SpaceX public REST API
  - Sent HTTP GET requests using Python (requests library)
  - Retrieved launch, rocket, payload, and landing outcome data
  - Parsed JSON responses into structured Pandas DataFrames
  - Performed basic validation and formatting
  - Exported cleaned data to CSV for downstream analysis
- https://github.com/pedromarquezv/PM_datascience course/blob/main/Part1.1_jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Web Scraping Process
  - Identify target webpages containing launch and landing information
  - Send HTTP requests using Python
  - Parse HTML content with BeautifulSoup
  - Extract tables and relevant fields
  - Clean and structure extracted data
  - Save results as CSV file
- https://github.com/pedromarquezv/PM_datasciencecourse/blob/main/Part 1.2_jupyter-labs-webscraping.ipynb

Data Wrangling

- Data Wrangling Process
  - Merge API and web-scraped datasets
  - Handle missing and null values
  - Filter relevant features
  - Convert data types
  - Create binary target variable (Landing Success)
  - Remove duplicates and inconsistencies
  - Generate final analysis-ready dataset
- https://github.com/pedromarquezv/PM_datasciencecourse/blob/main/Part1.3_labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- **Charts Plotted and Purpose**
- • **Bar charts** – Used to compare landing success rates across categorical variables (launch site, orbit type, booster version).
  • **Scatter plots** – Used to examine relationship between payload mass and landing success.
  • **Box plots** – Used to compare payload mass distributions for successful vs unsuccessful landings.
  • **Pie charts** – Used to visualize proportion of successful vs unsuccessful landings.
  • **Heatmaps / correlation matrices** – Used to identify relationships between numerical features.
  • **Histograms** – Used to understand distribution of payload mass and flight numbers.
- **Why These Charts Were Used**
- • To identify patterns and trends in landing success.
  • To detect relationships between mission parameters and outcomes.
  • To highlight important features for machine-learning modeling.
  • To support data-driven feature selection.
- https://github.com/pedromarquezv/PM_datasciencecourse/blob/main/Part2.2_edadataviz.ipynb

EDA with SQL

- **Summary of SQL Queries Performed**
- • Retrieve all launch records and preview dataset
  • Count total launches and total successful landings
  • Calculate landing success rate
  • Group launches by launch site and compute success rates
  • Group launches by orbit type and compute success rates
  • Identify unique launch sites and orbit types
  • Compute average payload mass by orbit type
  • Identify boosters with highest number of successful landings
  • Filter missions with heavy payloads (> specified threshold)
  • Order launches by flight number and payload mass
- **Purpose of SQL Analysis**
- • Quickly explore large datasets
  • Validate patterns found in visual EDA
  • Support feature selection for modeling
- https://github.com/pedromarquezv/PM_datasciencecourse/blob/main/Part2.1_jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- **Map Objects Created**
- • Markers for each launch site
  - • Colored markers for successful vs unsuccessful launches
  - • Circle markers indicating payload mass magnitude
  - • Lines showing distance from launch site to coast or nearby cities
  - • Marker clusters to group nearby launch points
- **Why These Objects Were Added**
- • Markers: visualize geographic locations of launch sites
  - • Colored markers: quickly distinguish landing success vs failure
  - • Circle markers: show relative payload size impact
  - • Lines: analyze spatial relationships between launch sites and infrastructure
  - • Clusters: improve readability when many points overlap
- https://github.com/pedromarquezv/PM_datasciencecourse/blob/main/Part3.1_lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- **Plots / Graphs Added**
- • Pie chart showing success vs failure for selected launch site
  • Scatter plot of payload mass vs landing success
  • Color-coded points by booster version or launch outcome
- **Interactions Added**
- • Dropdown menu to select launch site
  • Range slider to filter payload mass
  • Dynamic chart updates based on user selections
- **Why These Were Added**
- • Enable interactive exploration of launch performance
  • Compare success rates across launch sites
  • Analyze relationship between payload mass and landing success
  • Support user-driven pattern discovery
- https://github.com/pedromarquezv/PM_datasciencecourse/blob/main/Par3.2_spacex-dash-app.py

Predictive Analysis (Classification)

- **Key Phrases**
- • Select features from cleaned dataset
  • Encode categorical variables
  • Scale numerical features
  • Train/test split
  • Train multiple classifiers
  • Hyperparameter tuning
  • Evaluate and select best model

- **Flowchart (Bullet Version)**
- • Clean Dataset
  → Feature Selection
  → Encoding & Scaling
  → Train/Test Split
  → Train Models
  → Tune Hyperparameters
  → Evaluate
  → Best Model

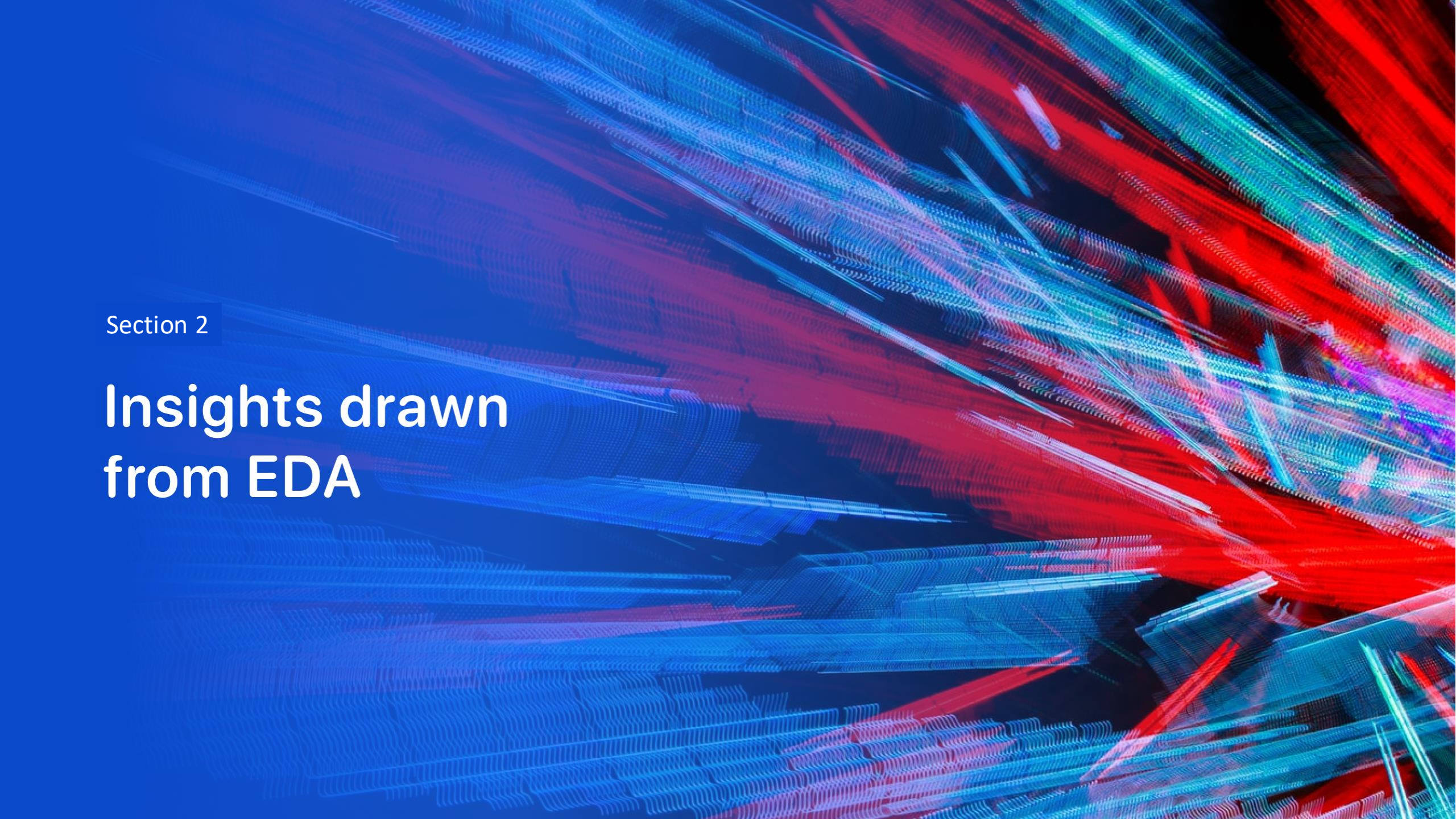
Best Model Selection

• Compared several classifiers
  • Used cross-validation
  • Selected model with highest test accuracy

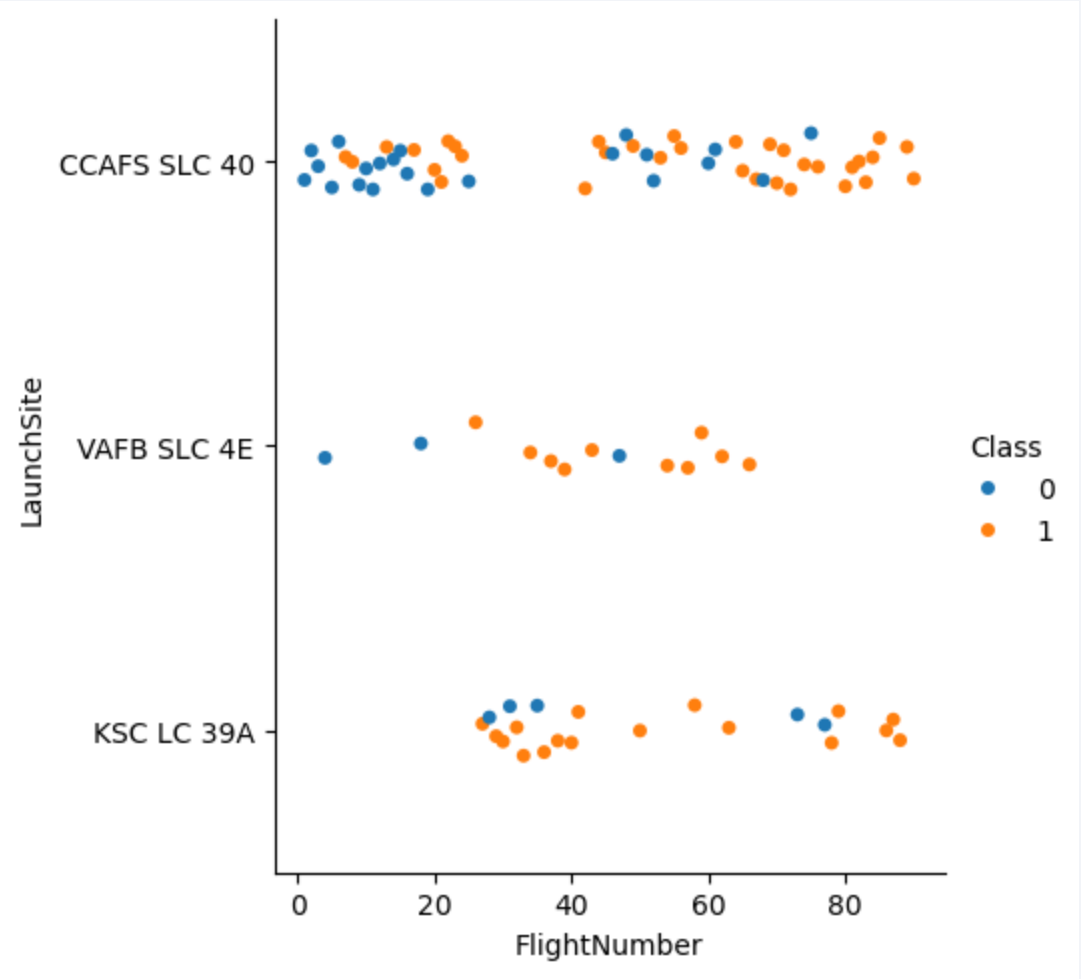
https://github.com/pedromarquezv/PM_datasciencecourse/blob/main/Part4_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

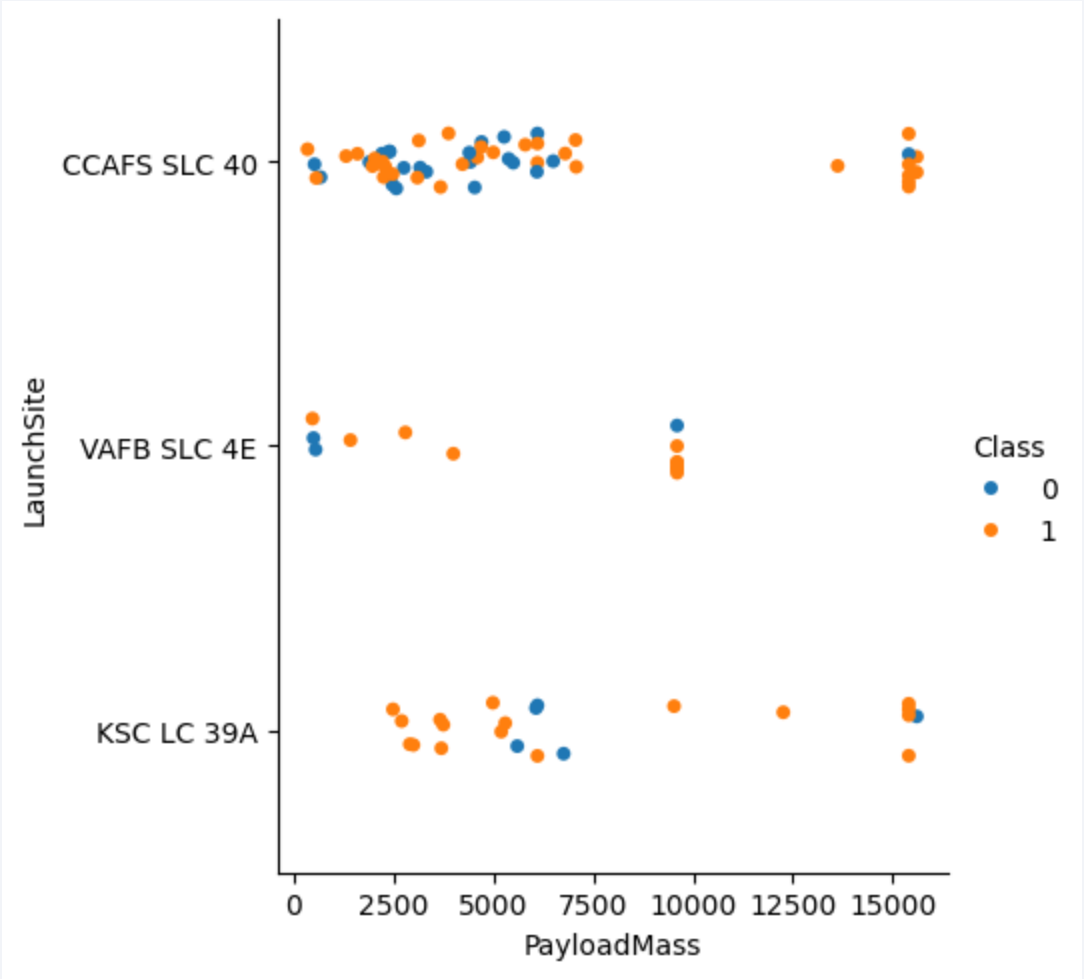
- Exploratory Data Analysis (EDA) Results
  - Landing success varies significantly by launch site and orbit type
  - Higher payload mass generally reduces landing success probability
  - Newer booster versions show higher success rates
  - Strong relationships identified between key mission parameters and outcomes
- Interactive Analytics (Demo Screenshots)
  - Interactive map shows geographic distribution of launch sites and outcomes
  - Dashboard enables filtering by launch site and payload range
  - Visual tools confirm EDA patterns through user-driven exploration
- Predictive Analysis Results
  - Multiple classification models trained and compared
  - Best model achieves strong accuracy in predicting landing success
  - Model confirms payload mass, orbit type, launch site, and booster version as most important features


Section 2
Insights drawn from EDA

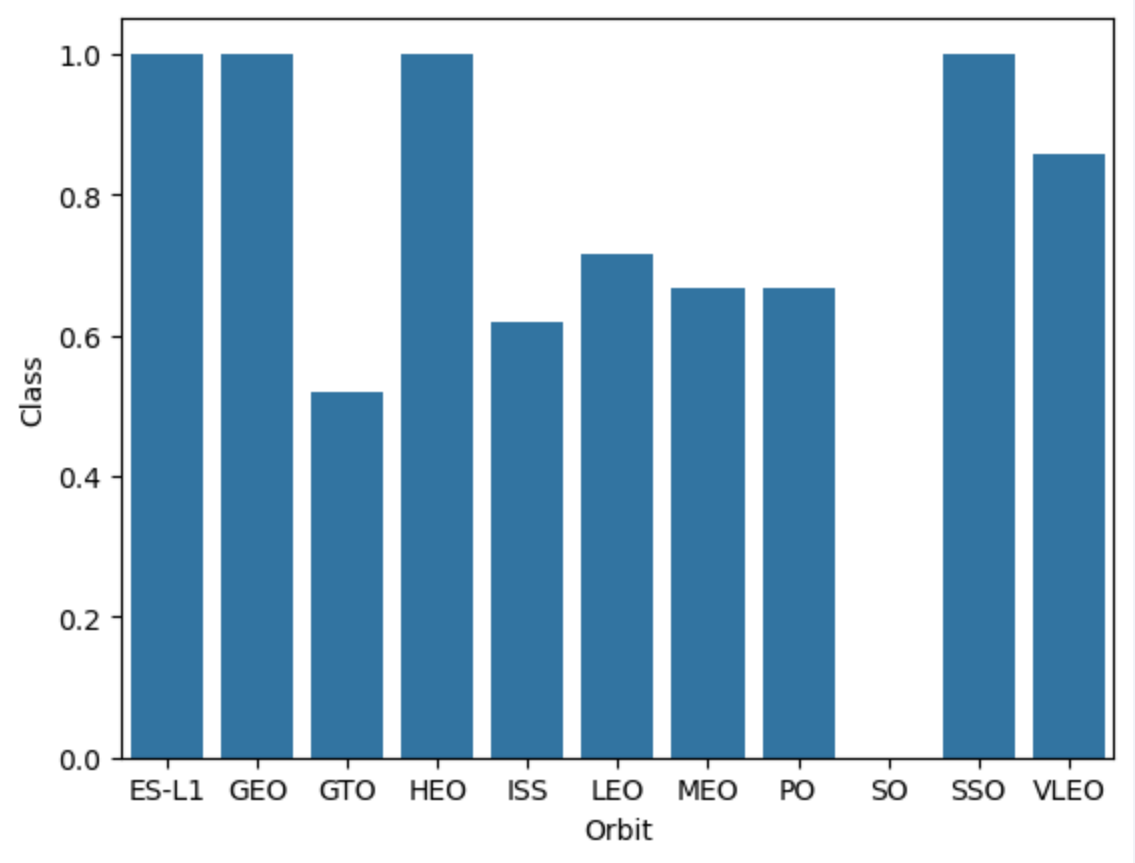
Flight Number vs. Launch Site



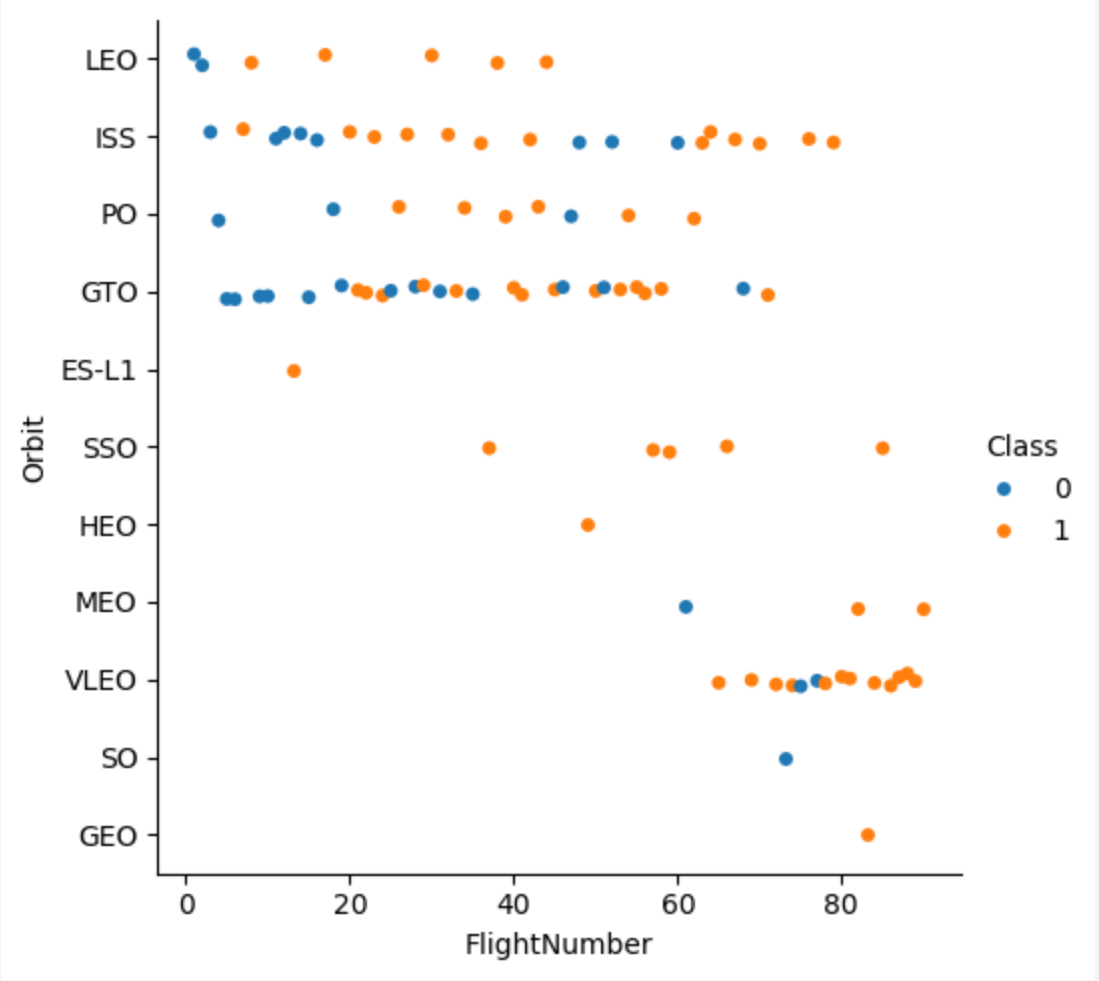
Payload vs. Launch Site



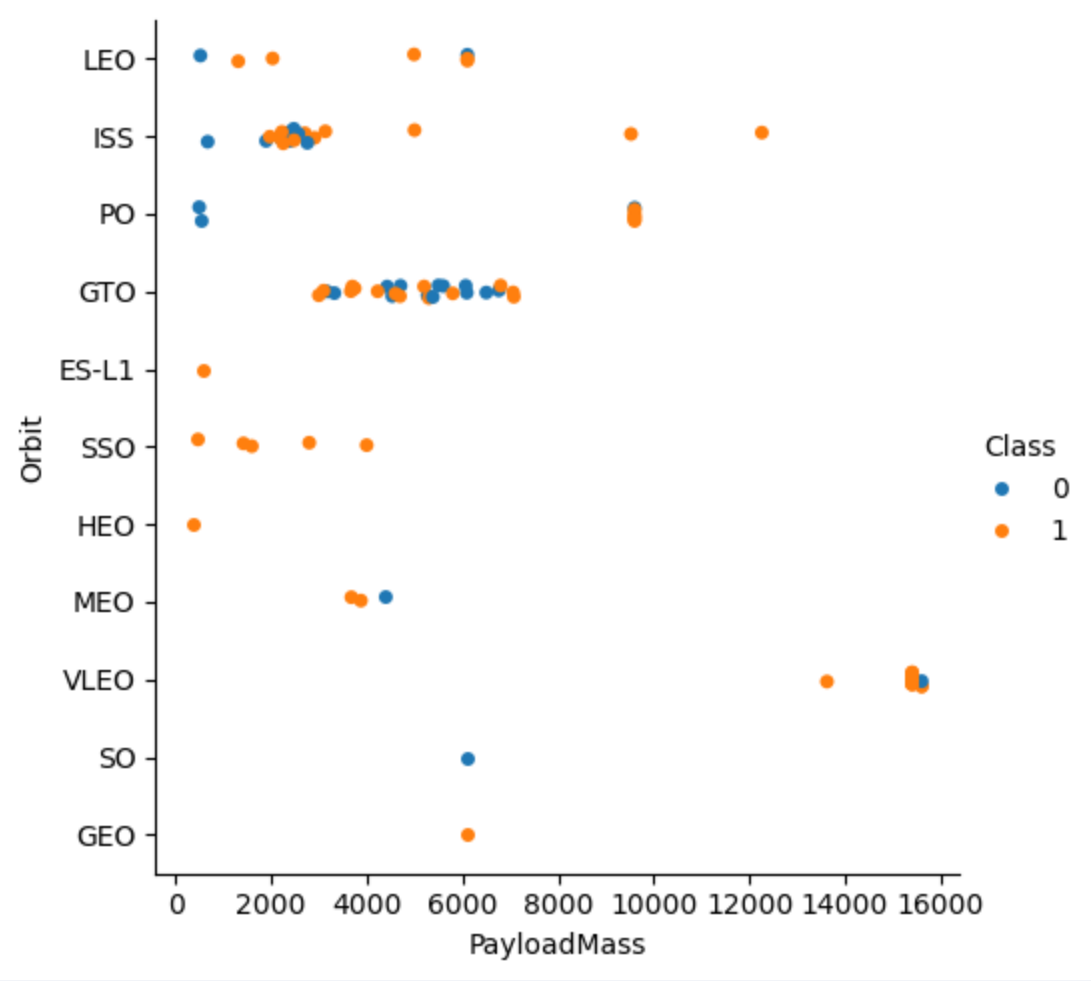
Success Rate vs. Orbit Type



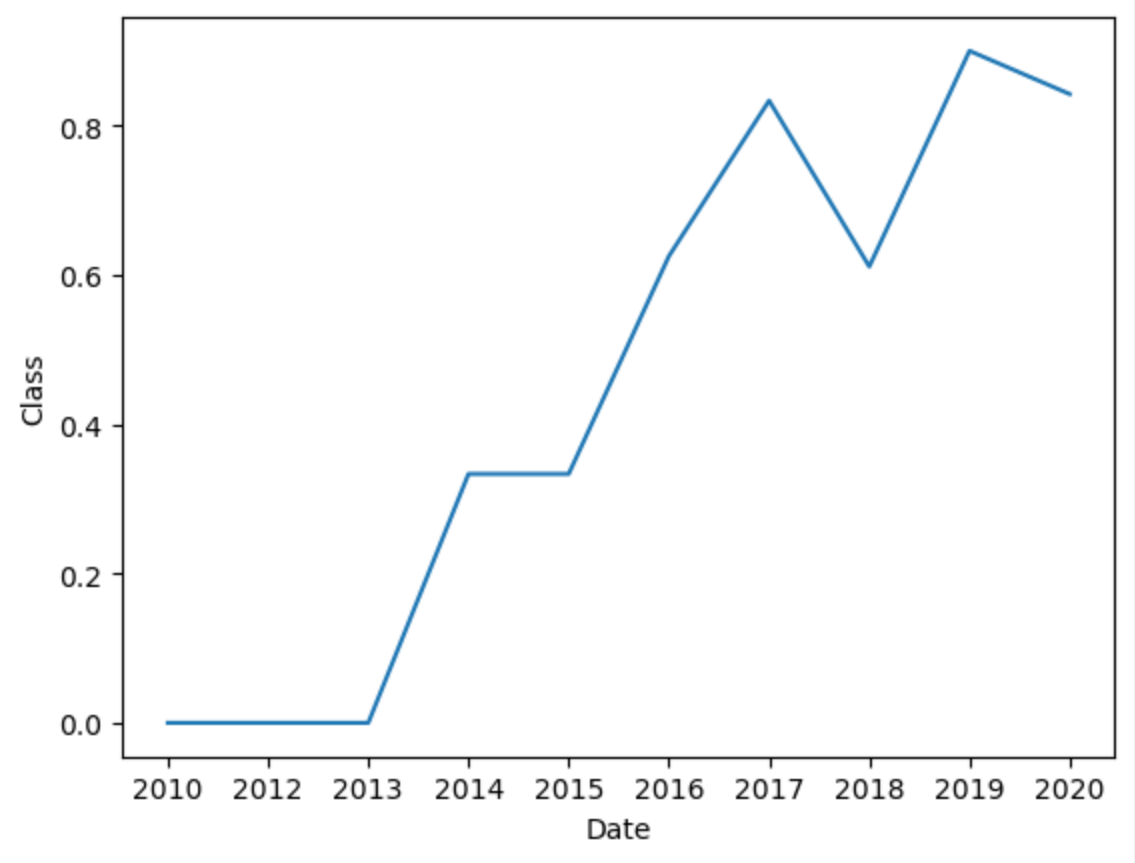
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

In [18]:
```
%sql select DISTINCT "Launch_Site" from SPACEXTBL
```

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

Out[18]:

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

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string 'CCA'

In [61]:
```
%sql select * from SPACEXTBL \
        where "Launch_Site" like "CCA%" limit 5
```

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

Out[61]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

In [37]:
```
%sql select SUM("PAYLOAD_MASS__KG_") from SPACEXTBL where Customer IN ("NASA (CRS)")
```

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

Out[37]:

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

Task 4

Display average payload mass carried by booster version F9 v1.1

In [38]:
```
%sql select AVG("PAYLOAD_MASS__KG_") from SPACEXTBL where "Booster_Version" IN ("F9 v1.1")
```

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

Out[38]:

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

In [45]:
```
%sql select min(Date) from SPACEXTBL where "Landing_Outcome" LIKE "Success (ground pad)"
```

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

Out[45]:

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [63]:
```
%sql select "Booster_Version" from SPACEXTBL \
    where "Landing_Outcome" like "Success (drone ship)" \
    and "PAYLOAD_MASS__KG_" >4000 and "PAYLOAD_MASS__KG_" <6000
```

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

Out[63]:

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [55]:
```
%sql SELECT \
  (SELECT COUNT("Mission_Outcome") \
   FROM SPACEXTBL \
   WHERE "Mission_Outcome" LIKE '%Success%') AS "Successful",\
  (SELECT COUNT("Mission_Outcome") \
   FROM SPACEXTBL \
   WHERE "Mission_Outcome" LIKE '%Failure%') AS "Failures";
```

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

Out[55]:

Successful	Failures
100	1

Boosters Carried Maximum Payload

Task 8

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

In [70]:
```
%sql select "Booster_Version" from SPACEXTBL \
    where "PAYLOAD_MASS__KG_" = \
    (select max("PAYLOAD_MASS__KG_") from SPACEXTBL)
```

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

Out[70]:

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

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [78]:
```
%sql select substr(Date, 6,2) as 'Month', "Landing_Outcome", "Booster_Version", "Launch_Site"\
    from SPACEXTBL where substr(Date,0,5) = '2015'\
    and "Landing_Outcome" like "Failure (drone ship)"
```

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

Out[78]:

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

Task 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.

In [91]:
```
%sql select "Landing_Outcome",count("Landing_Outcome") as 'Count' from SPACEXTBL\
    where Date>'2010-06-04' and Date<'2017-03-20'\
    group by "Landing_Outcome"\
    order by "Count" desc
```

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

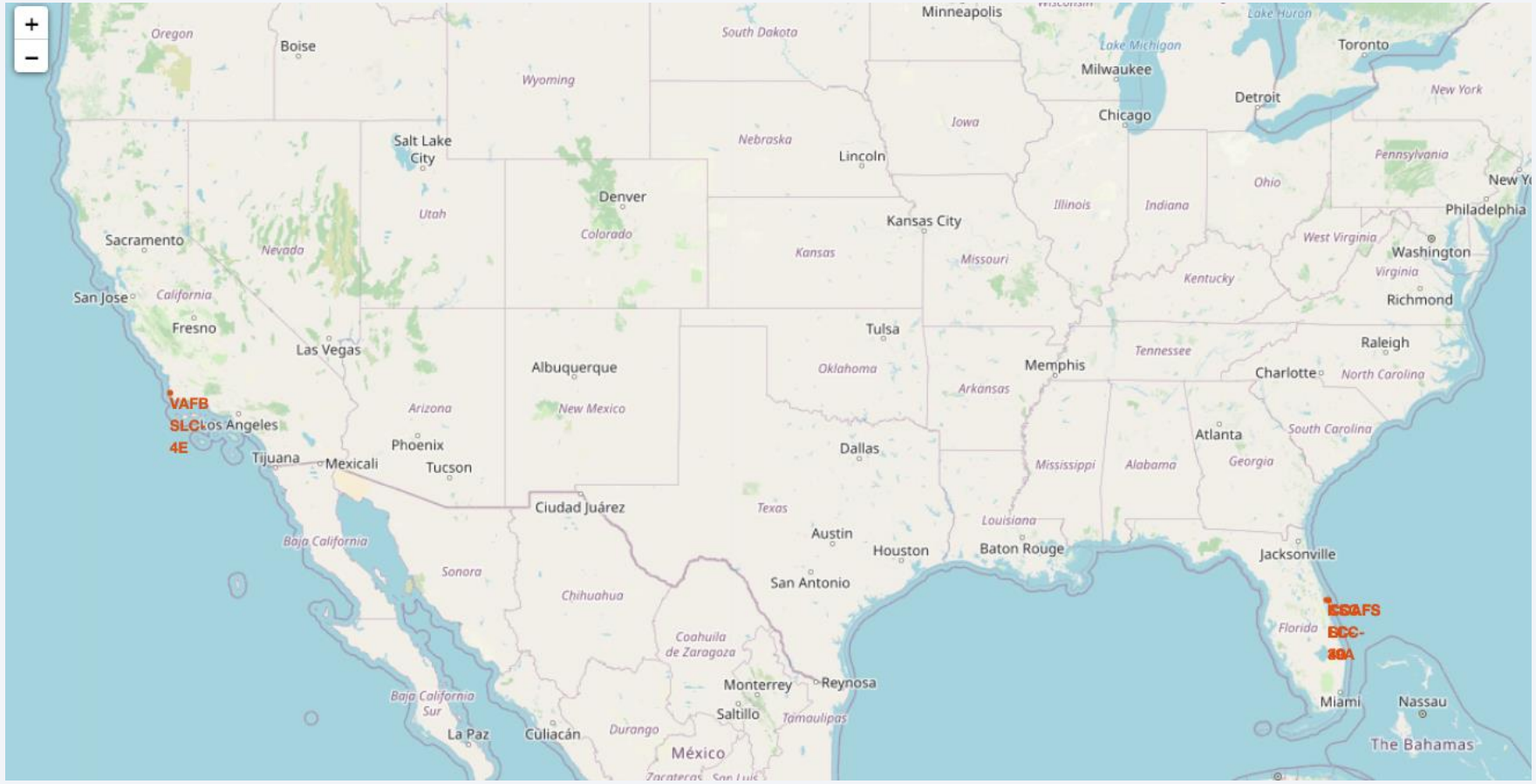
Out[91]:

Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

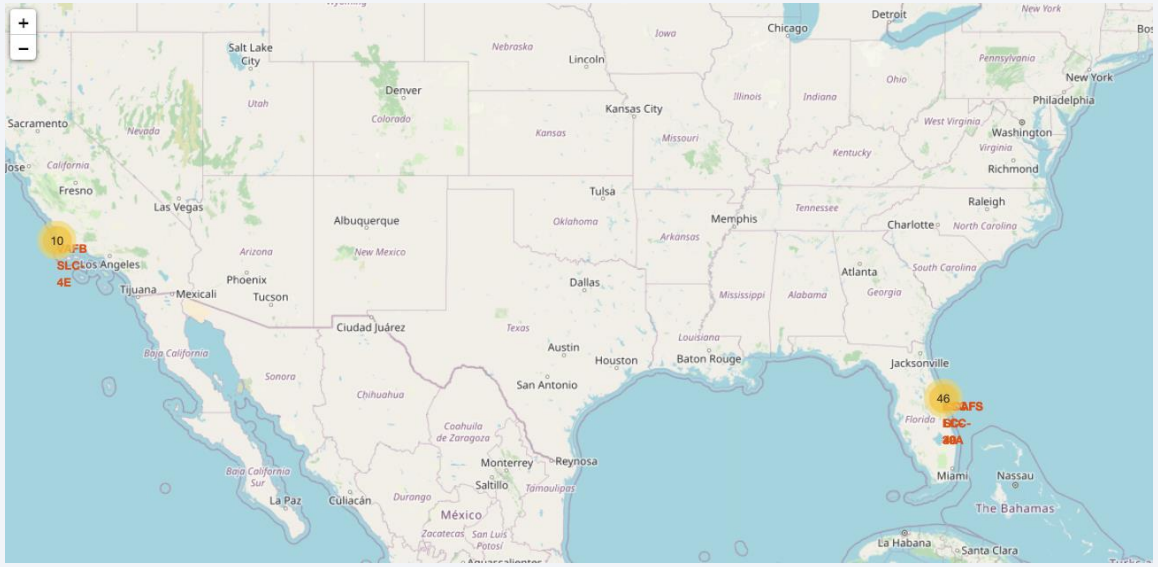
Section 3

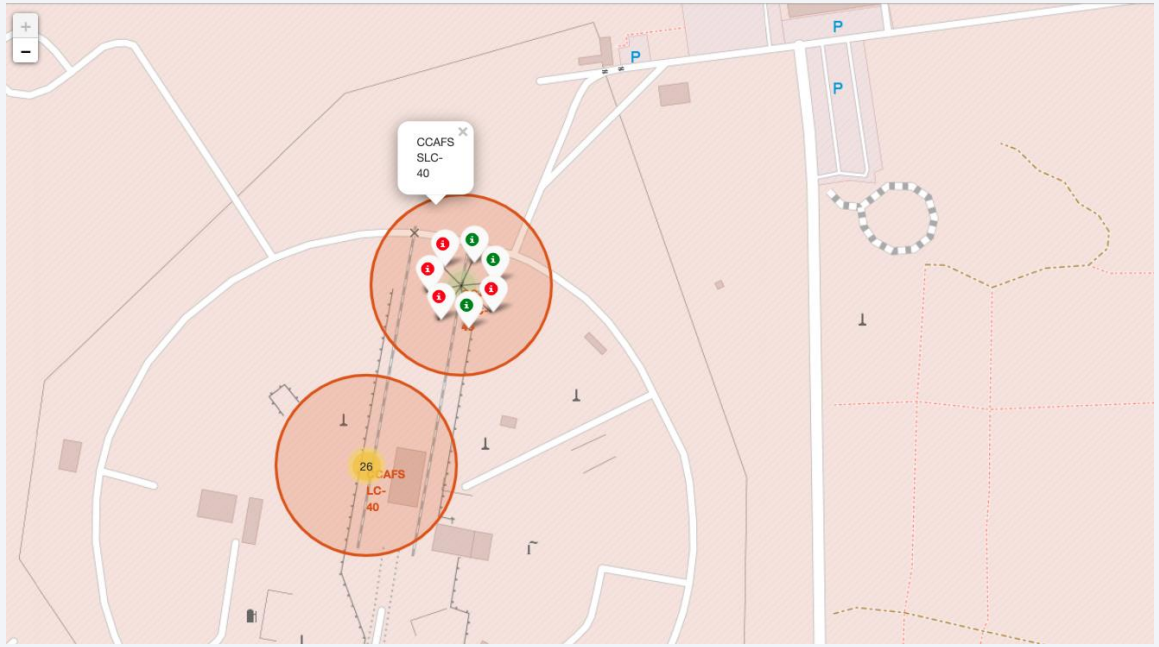
Launch Sites Proximities Analysis

Folium Map with Launch Sites

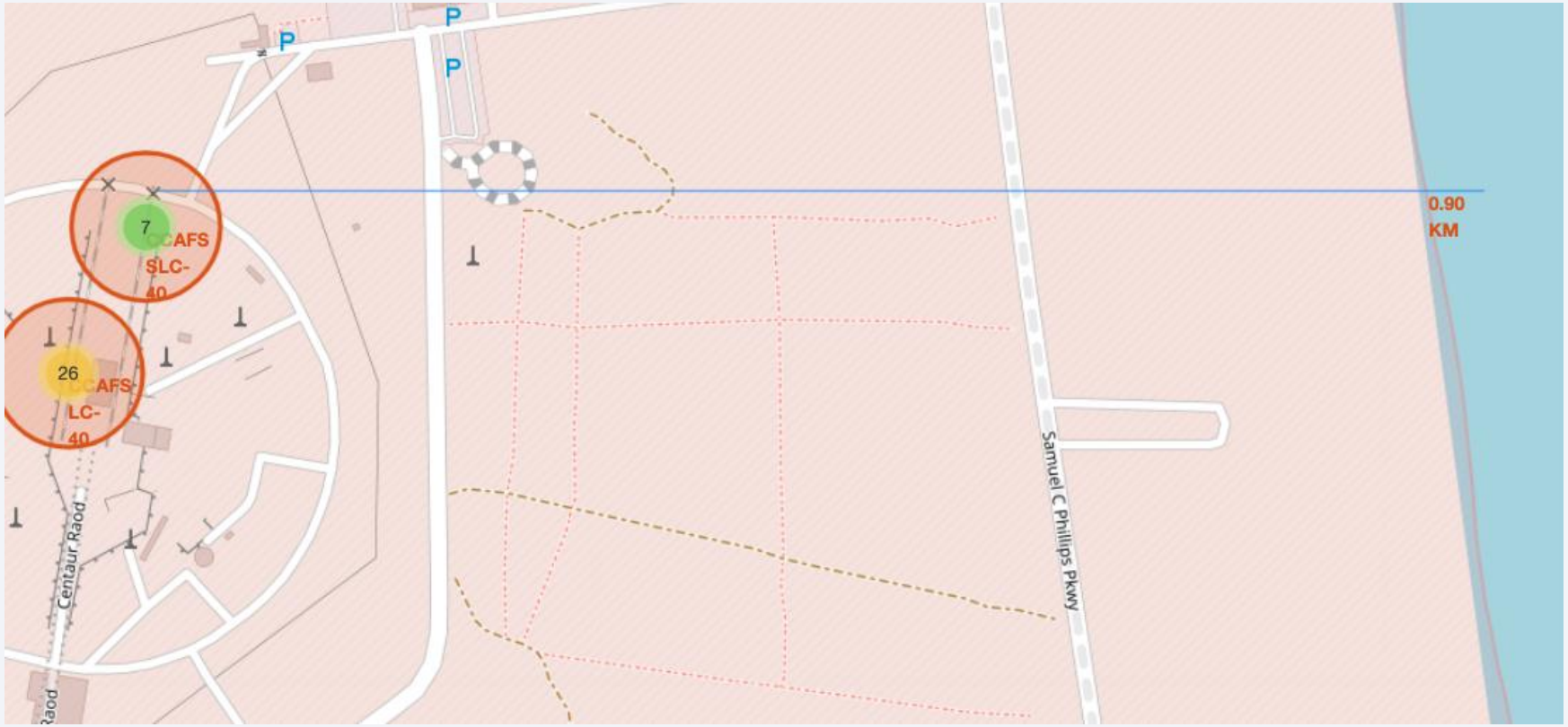


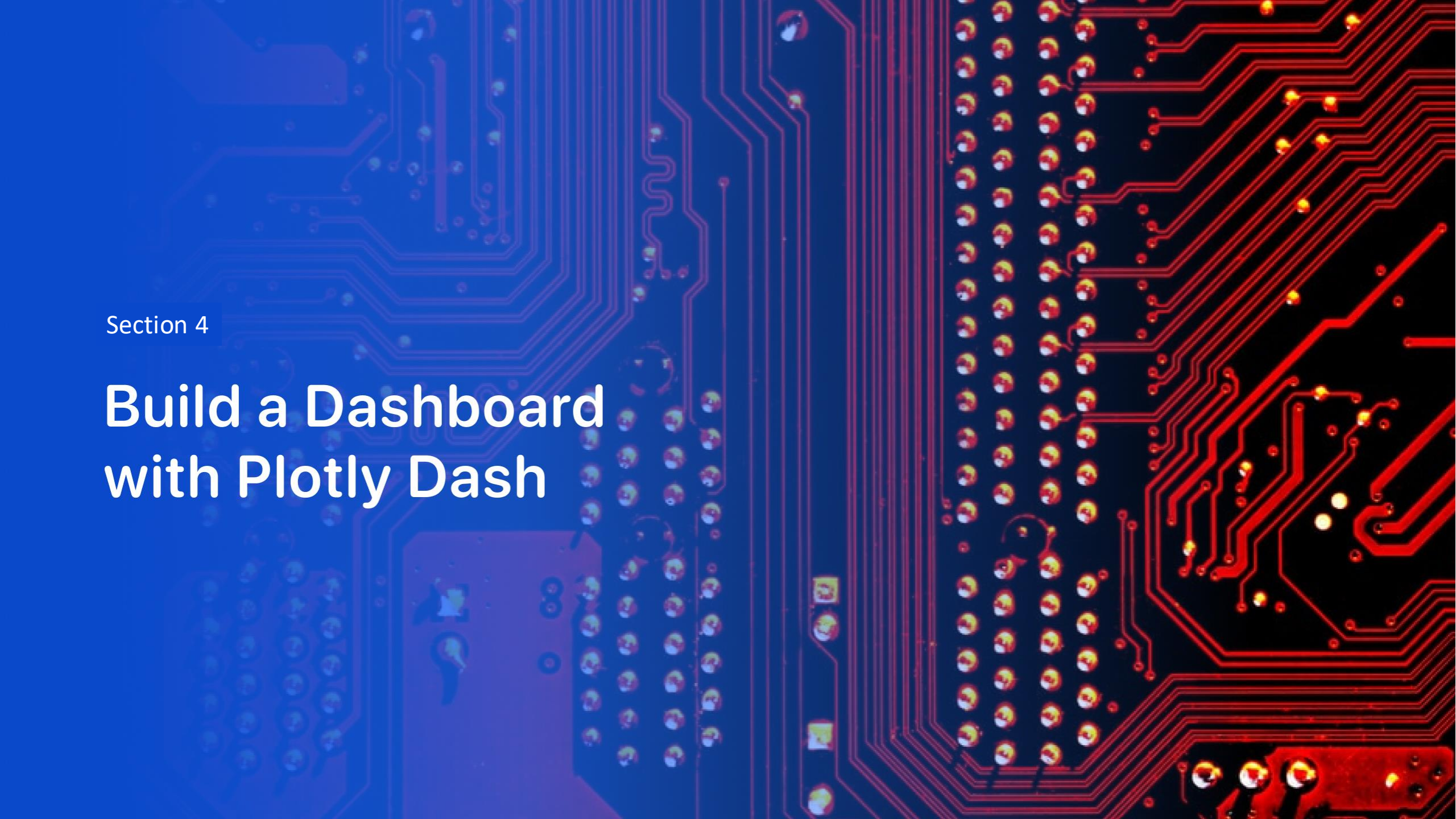
Folium Map with success/failed launches for each site



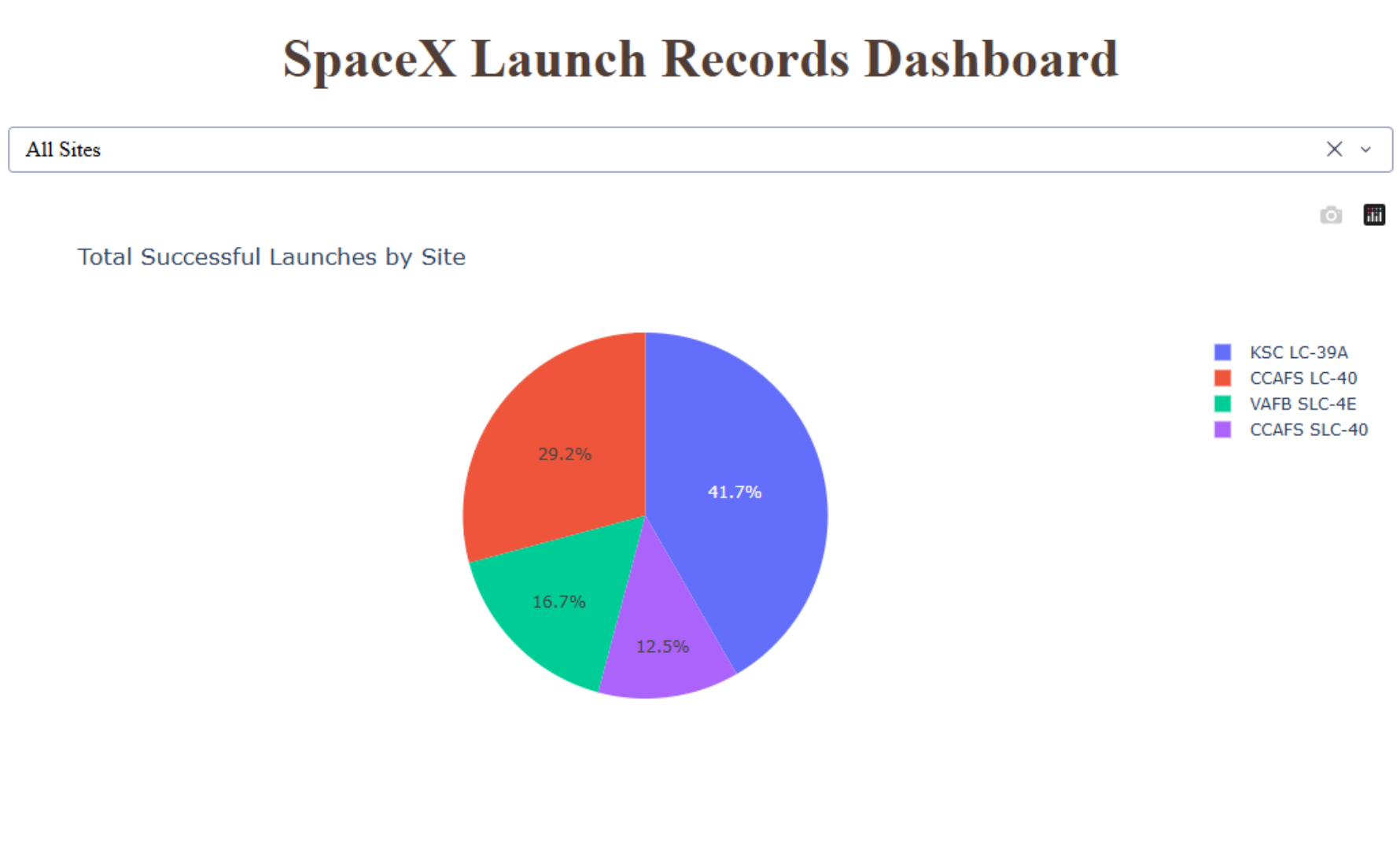


Folium Map with distance to coastline

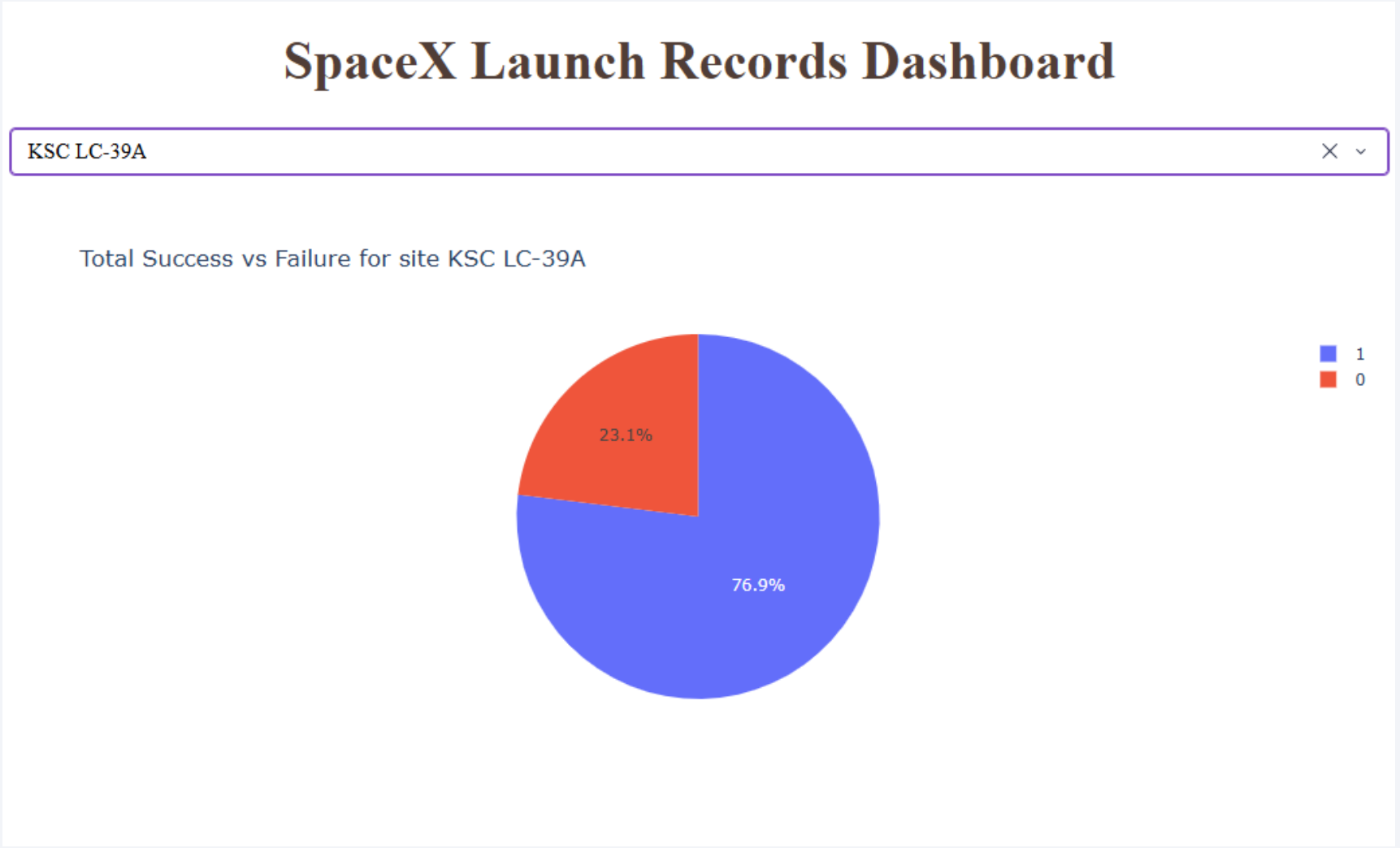



Section 4
Build a Dashboard with Plotly Dash

Dashboard launch success count for all sites



Dashboard launch site with highest success ratio



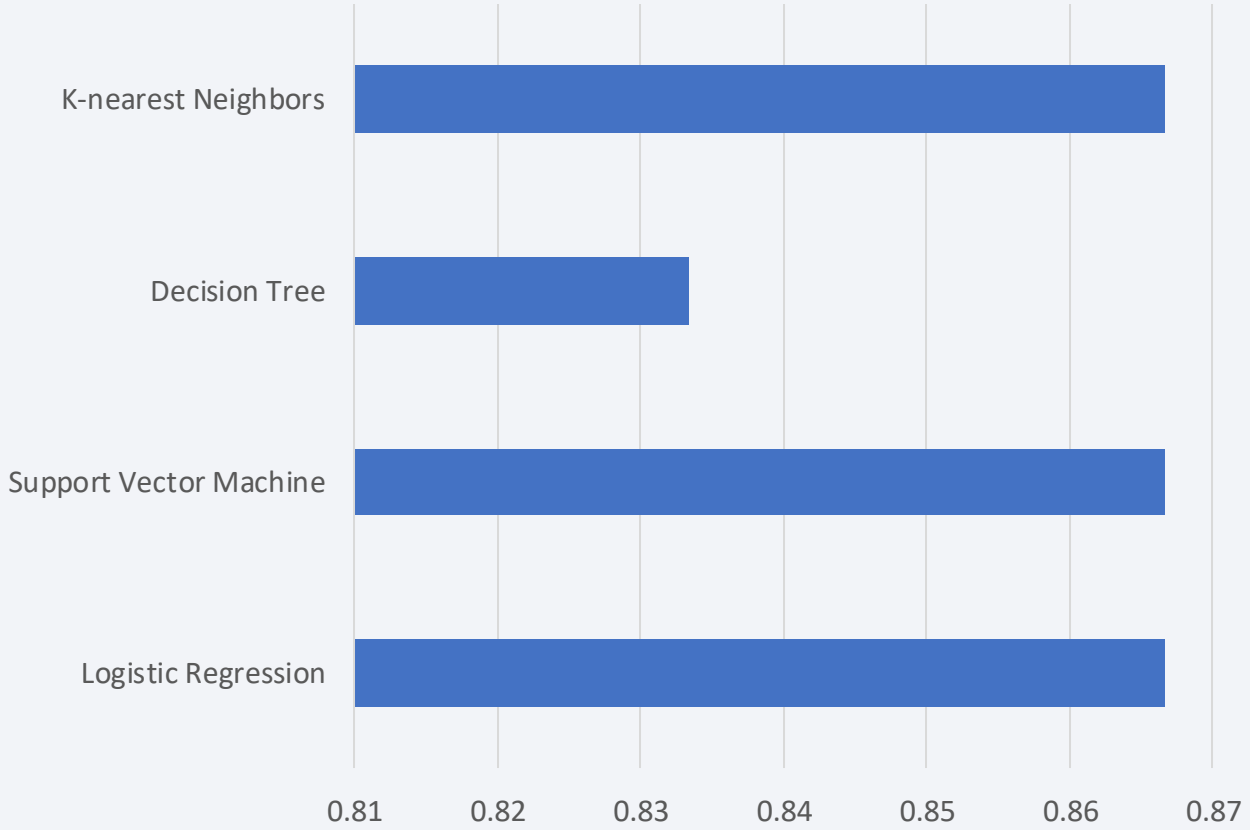
Dashboard Payload vs. Launch Outcome

Section 5

Predictive Analysis (Classification)

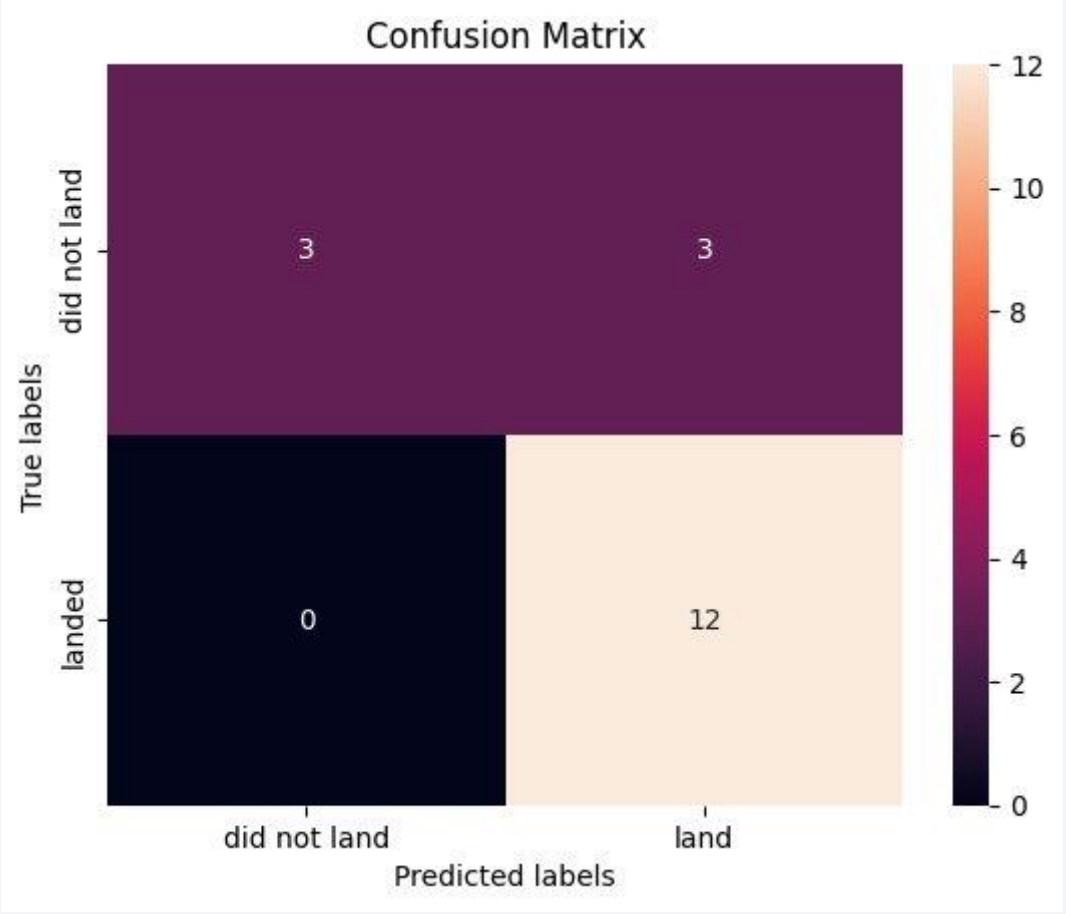
Classification Accuracy

All models except for Decision Tree have similar accuracy on the test dataset.



Confusion Matrix

- Best model confusion matrix: 12 True positives, 3 False positives



Conclusions

- • Machine-learning models can reliably predict first-stage landing success using historical launch data
  • Mission parameters such as payload mass, orbit type, launch site, and booster version are key predictors
  • The selected classification model demonstrates strong generalization performance
  • Predictive insights can support mission planning and cost optimization
  • Data-driven approaches add measurable value to launch decision-making

Thank you!