IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

D. Vasychev
15.12.2025

Outline

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

Executive Summary

- SpaceX Falcon 9 launch data were analyzed to predict first-stage landing outcomes. Data collection, exploratory analysis, interactive visualization, and classification models were applied.

- All evaluated machine learning models demonstrated comparable accuracy. The results support informed decision-making for launch cost estimation.

Introduction

- The cost advantage of Falcon 9 launches is largely driven by first-stage reusability.

- Predicting landing success enables estimation of launch costs and competitive bidding strategies.

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- Data were collected using the SpaceX REST API and web scraping of Wikipedia pages.

Data wrangling

- Missing values were handled using appropriate statistical methods.
- A binary target variable representing landing success was created.

Exploratory data analysis (EDA)

- Data relationships and success trends were analyzed using visualization techniques and SQL queries.

Methodology

Executive Summary

Interactive visual analytics

- Geographical launch data and success rates were visualized using Folium and Plotly Dash dashboards.

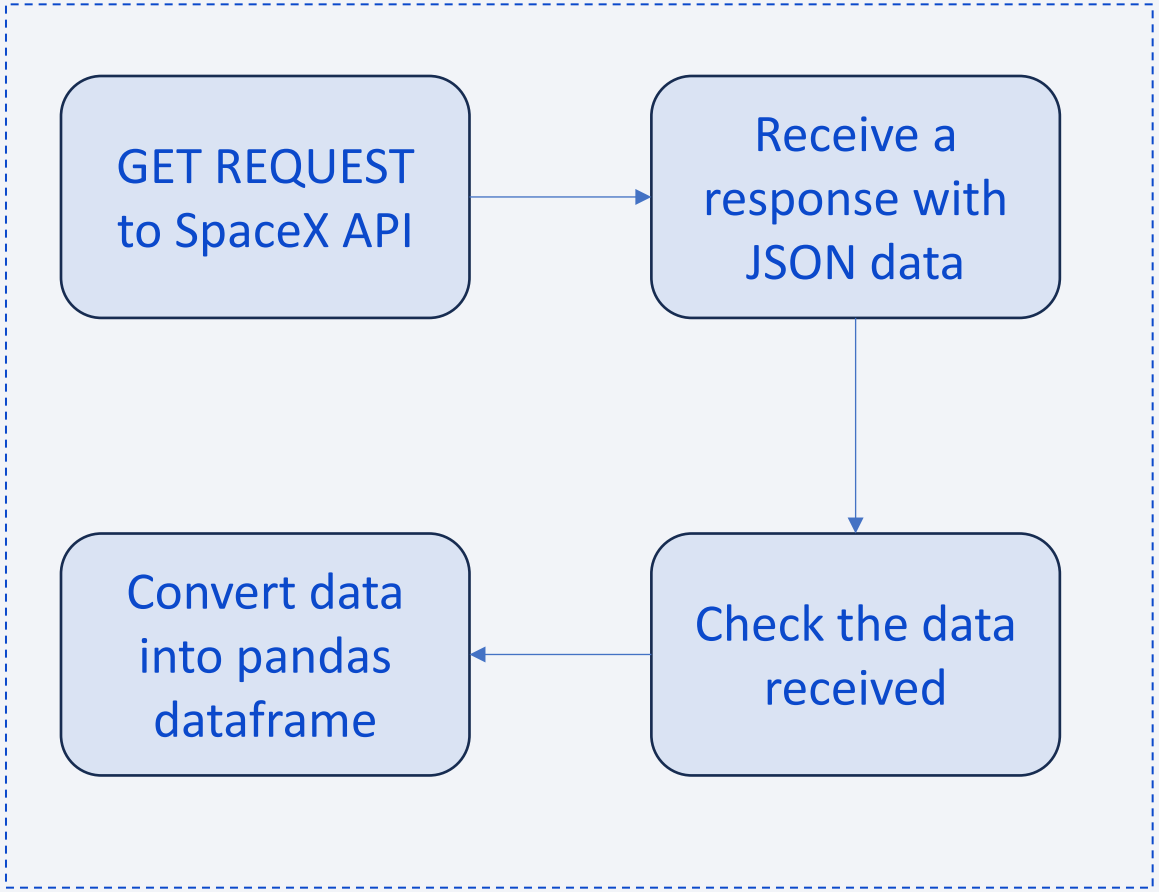
Predictive analysis

- Classification models including K-Nearest Neighbors, Support Vector Machine, Logistic Regression, and Decision Tree were trained and tuned using Grid Search.

Data Collection – SpaceX API

- Launch data were collected using the SpaceX REST API.
- The completed SpaceX API calls notebook can be found at GitHub repository with the following URL:

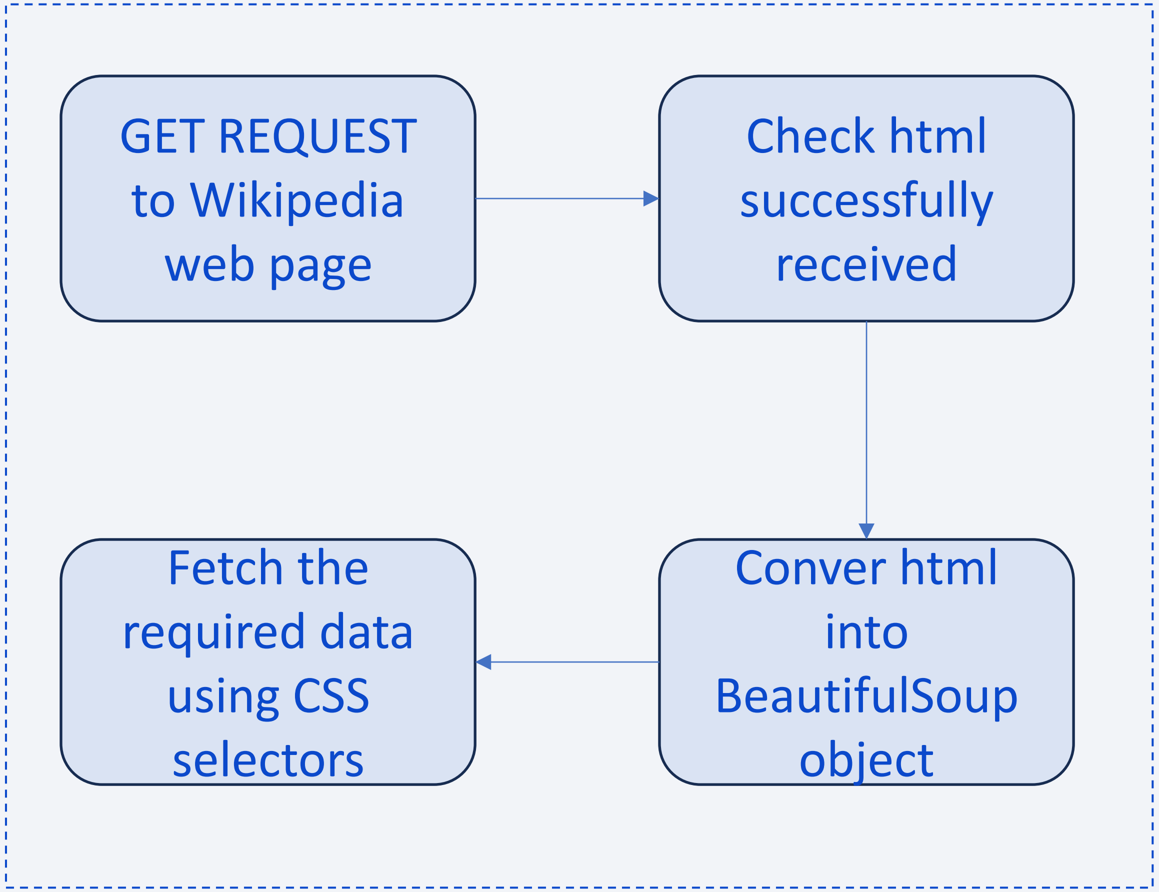
https://github.com/DennisVSIG/ibm_data_science_capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

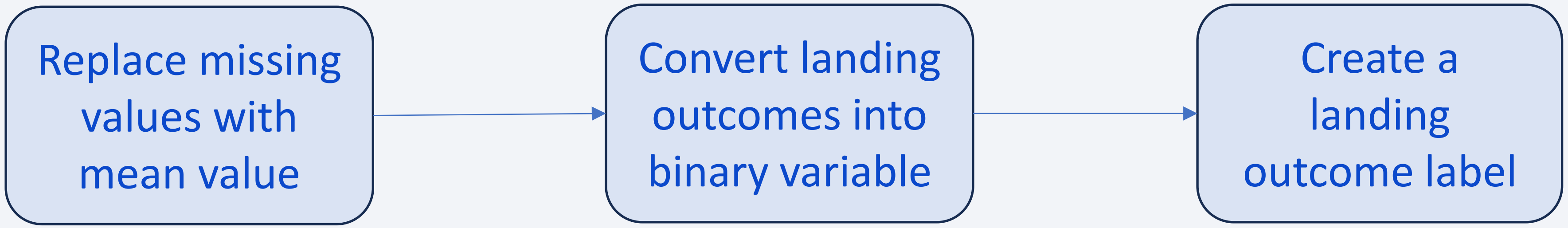
- The SpaceX data are supplemented with data obtained through web scraping of Wikipedia pages.
- The completed webscrapping notebook can be found at GitHub repository with the following URL:

https://github.com/DennisVSIG/ibm_data_science_capstone/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- The dataset was cleaned by handling missing values using appropriate statistical methods.
- A binary target variable representing successful or unsuccessful first-stage landing was created.



- The completed data wrangling notebook can be found at GitHub repository with the following URL: https://github.com/DennisVSIG/ibm_data_science_capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Scatter plots are used to visualize the relationship between launch features.
- Bar chart demonstrates success rate for each orbit.
- Line plots are built to show yearly success rate.

- The completed data visualization notebook can be found at GitHub repository with the following URL:

  https://github.com/DennisVSIG/ibm_data_science_capstone/blob/main/edadataviz.ipynb

EDA with SQL

- The following SQL queries were performed for exploratory data analysis:
- SELECT DISTINCT
- SELECT … WHERE … LIKE
- SELECT SUM(), SELECT AVG(), SELECT MIN(), SELECT MAX(), SELECT COUNT()
- SELECT DENSE_RANK()
- WITH … AS … FROM … SELECT; GROUP by, ORDER by

- The completed SQL exploratory data analysis notebook can be found at GitHub repository with the following URL: https://github.com/DennisVSIG/ibm_data_science_capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Launch site locations visualized on a map
- Success and failure outcomes marked with clustered points
- Distance lines added to cities, coastlines, and infrastructure
- Mouse position enabled for geographic reference

- The completed Folium notebook can be found at GitHub repository with the following URL: https://github.com/DennisVSIG/ibm_data_science_capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Pie chart shows success rate by launch site
- Dropdown allows selection of individual sites
- Scatter plot shows payload mass vs success outcome
- Color encoding represents booster versions

- The completed Plotly Dash lab can be found at GitHub repository with the following URL: https://github.com/DennisVSIG/ibm_data_science_capstone/blob/main/dash_interactivity.py

Predictive Analysis (Classification)

Predictive Modeling

- Models evaluated:
  - Logistic Regression
  - K-Nearest Neighbors
  - Support Vector Machine
  - Decision Tree
- Hyperparameters tuned using Grid Search

Predictive Analysis (Classification)

Model Evaluation

- All models demonstrated similar accuracy on test data
- No significant performance gap observed between classifiers
- Model selection prioritized interpretability and simplicity

- The completed Predictive Analysis notebook can be found at GitHub repository with the following URL: https://github.com/DennisVSIG/ibm_data_science_capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

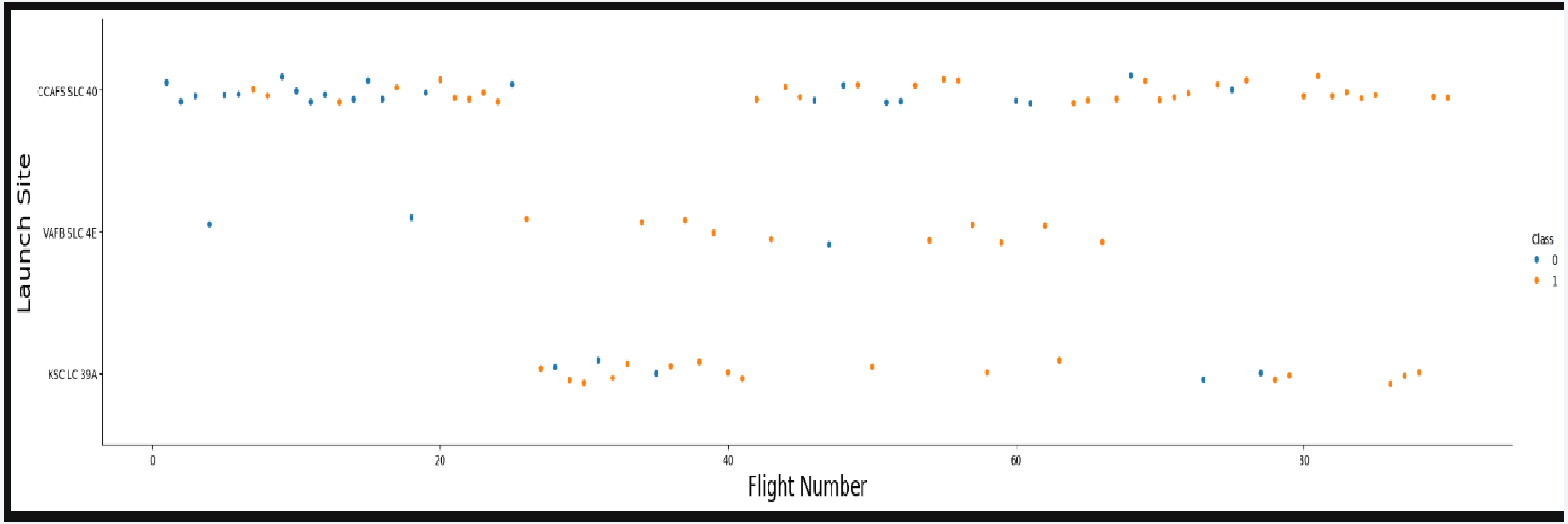
- Launch success rate increased over time
- Payload range impacts landing success
- Certain booster versions show higher reliability
- KSC LC-39A shows the highest number of successful launches

Section 2

Insights drawn from EDA

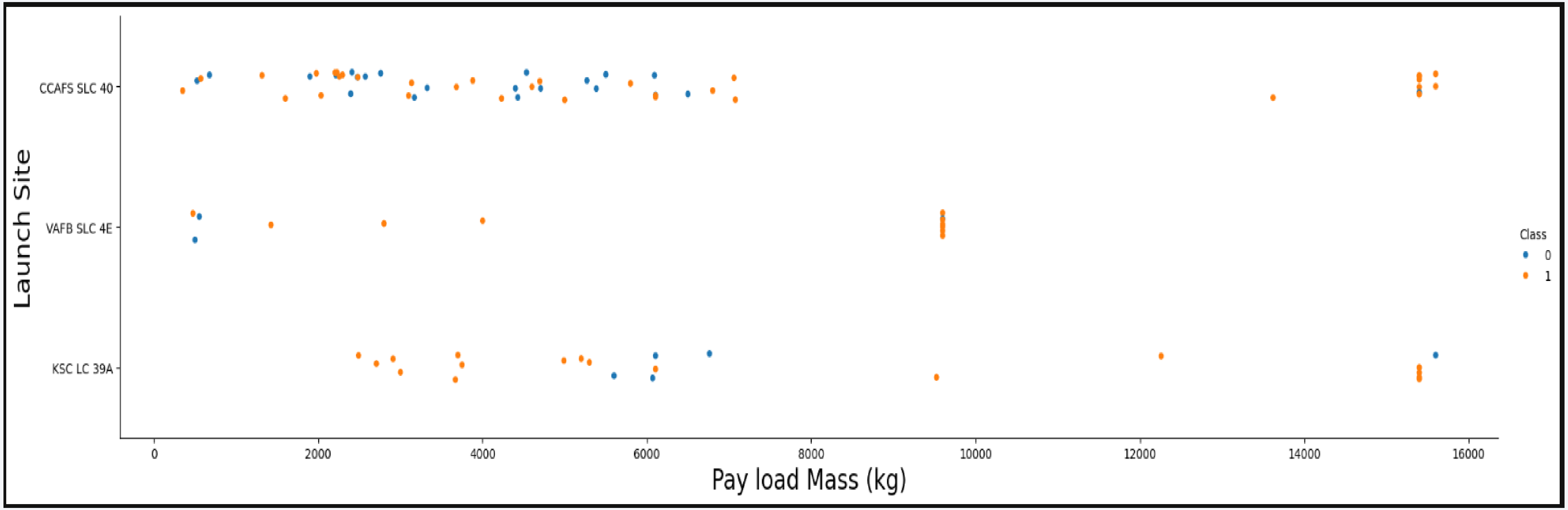
Flight Number vs. Launch Site

The relationship between Flight Number and Launch Site



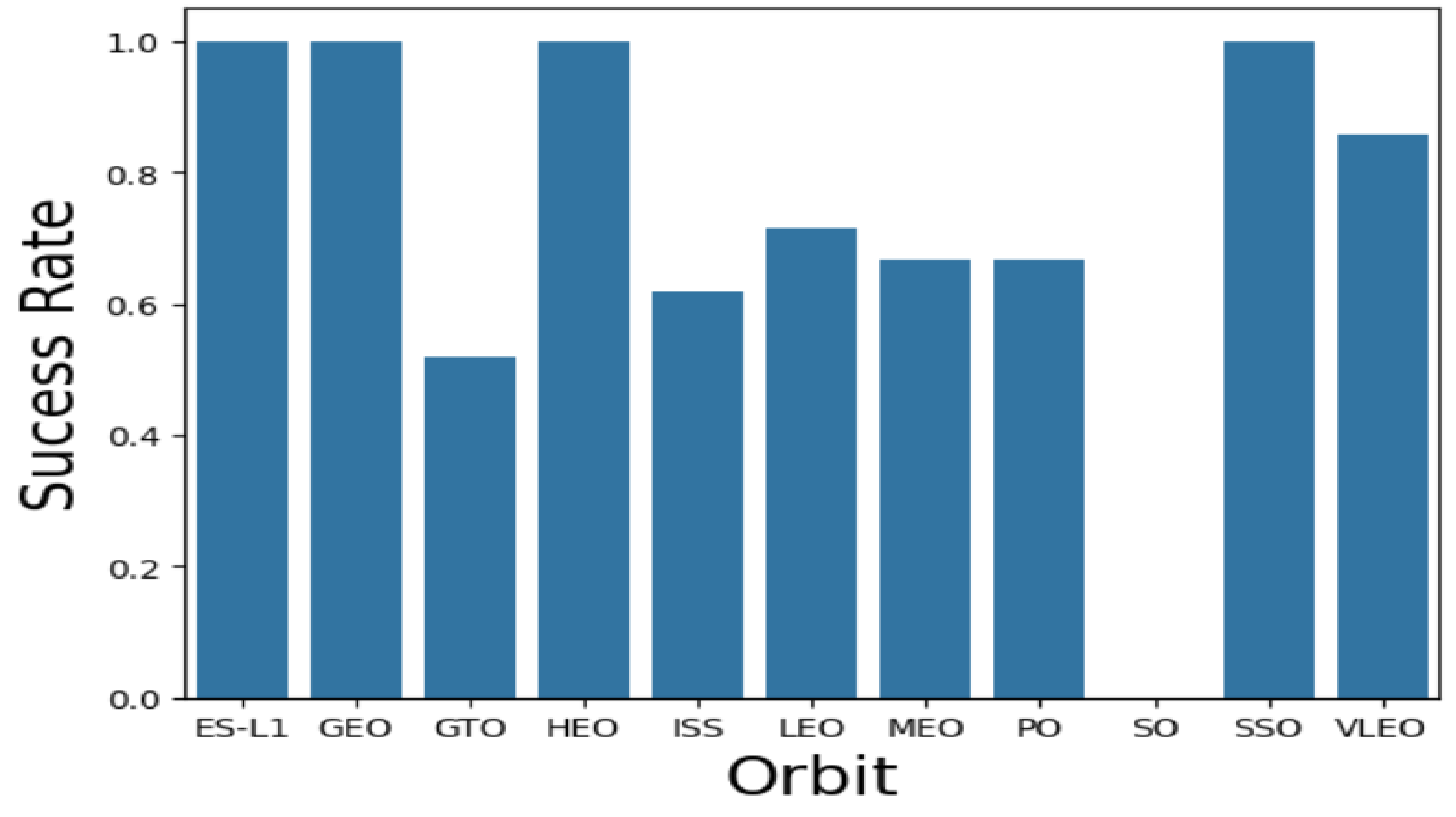
Payload vs. Launch Site

The relationship between Payload and Launch Site



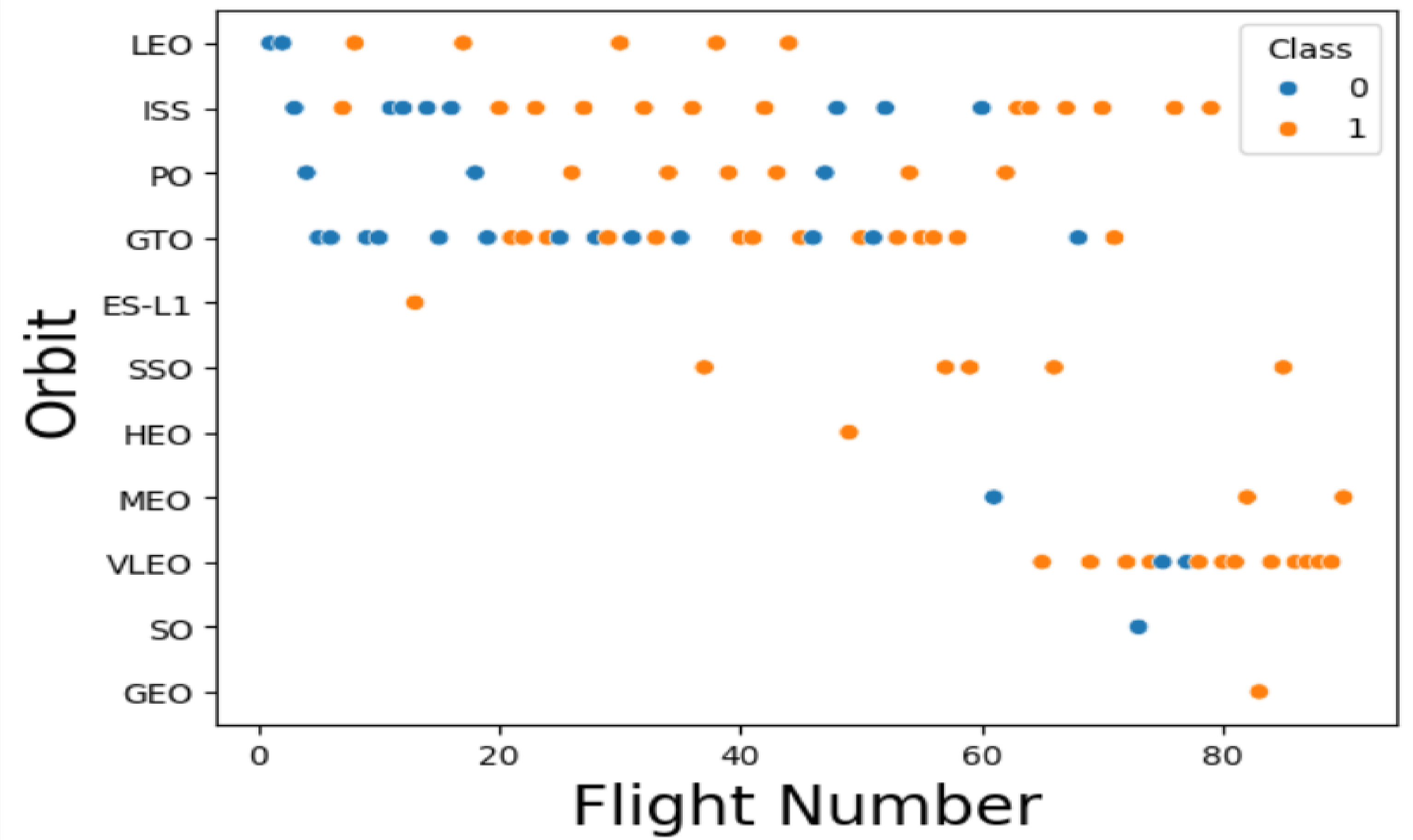
Success Rate vs. Orbit Type

Success rate of each orbit type



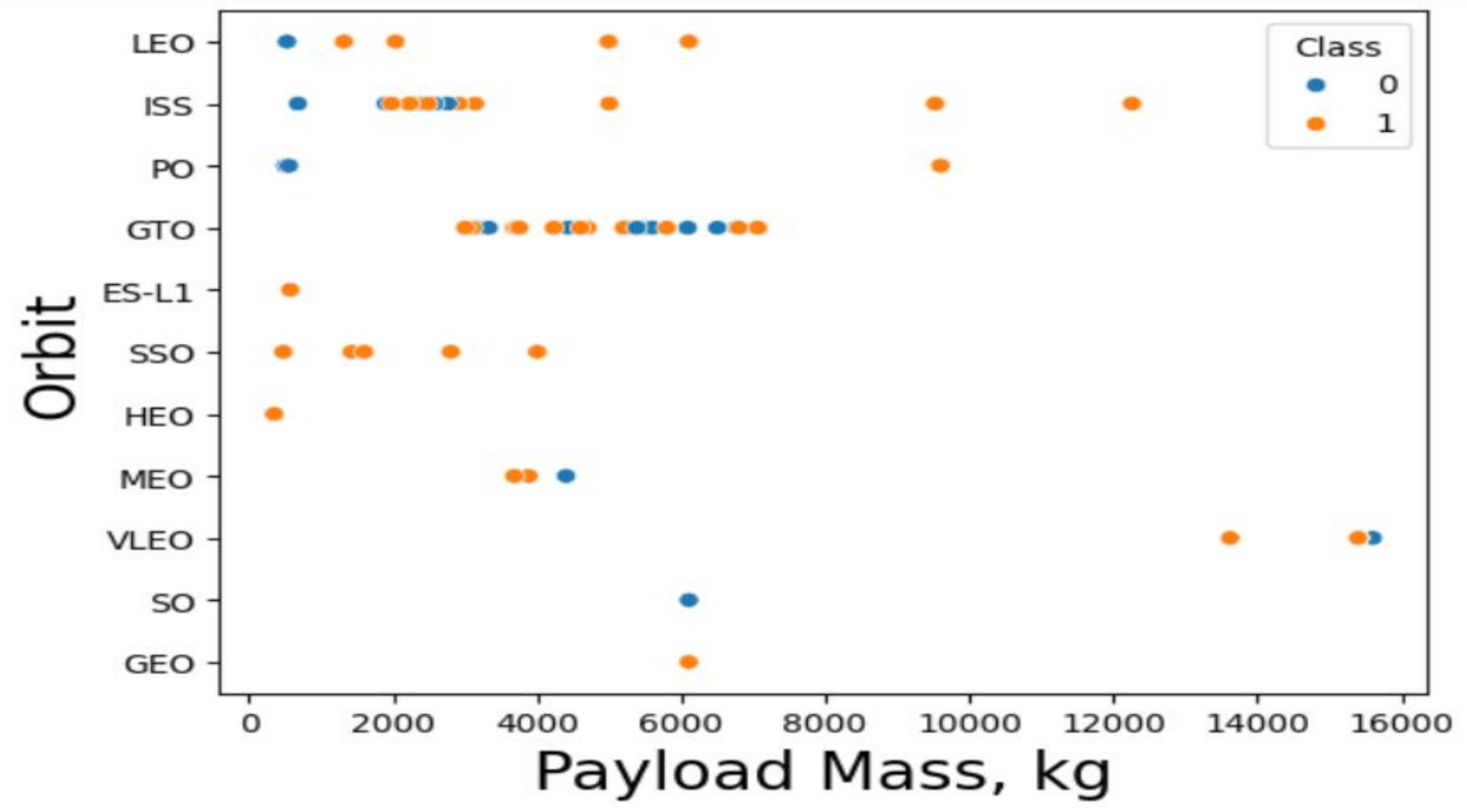
Flight Number vs. Orbit Type

The relationship between Flight number and Orbit



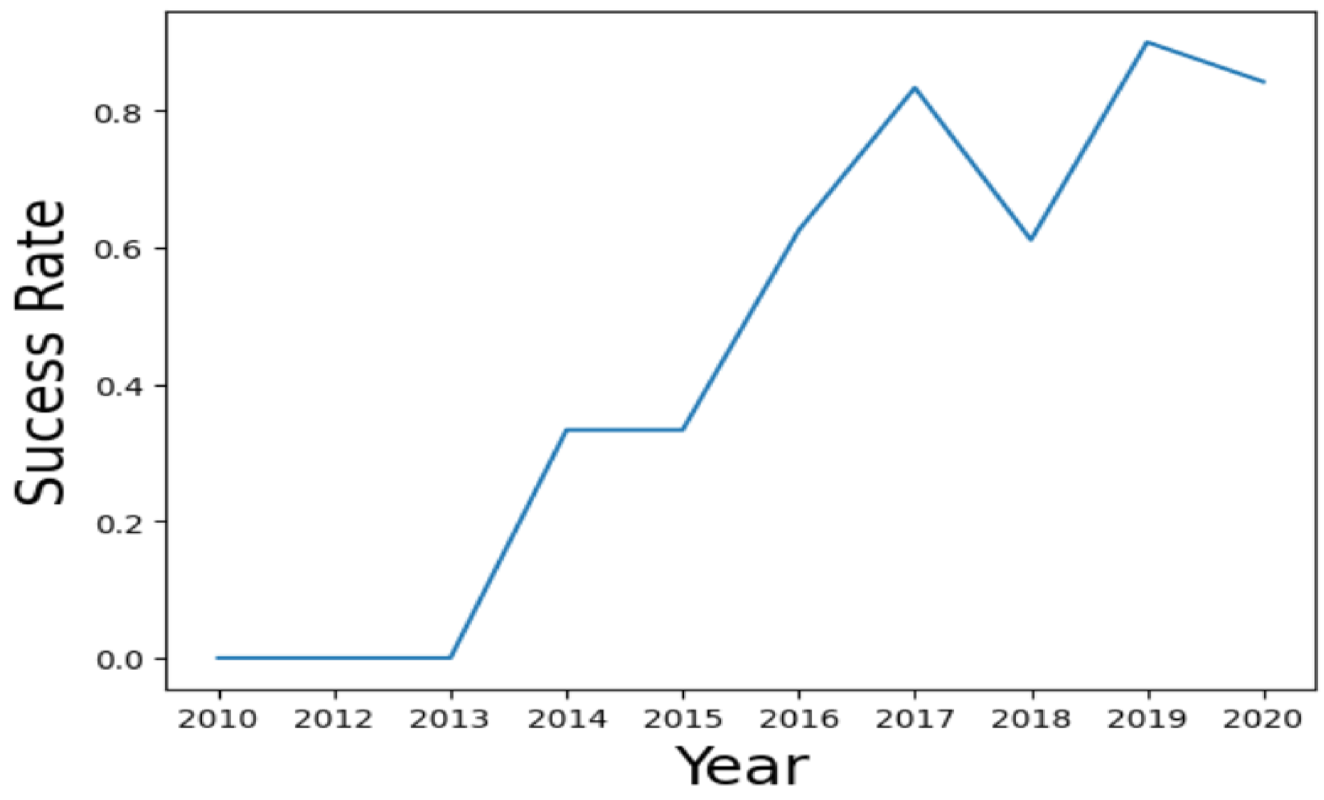
Payload vs. Orbit Type

Scatter plot of payload vs. orbit type



Launch Success Yearly Trend

Yearly average success rate



All Launch Site Names

SELECT DISTINCT query is used to find the names of the unique launch sites

```
[25]: %sql select distinct "Launch_Site" from SPACEXTBL;

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

[25]:

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

Launch Site Names Begin with 'CCA'

LIMIT and LIKE commands are used to find 5 records where launch sites begin with `CCA`

```
[27]: %sql select * from SPACEXTBL where "Launch_Site" like 'CCA%' limit 5;

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

[27]:

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

The SUM() and LIKE() functions are used to calculate the total payload carried by boosters from NASA (CRS)

Task 3

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

```
[33]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where "Customer" = 'NASA (CRS)'

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

[33]:

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- The AVG() function and WHERE clause are used to calculate the average payload mass carried by booster version F9 v1.1

Task 4

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

```
[37]: %sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where "Booster_Version" = 'F9 v1.1';
```

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

[37]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

The MIN() function is used to find the dates of the first successful landing outcome on ground pad

Task 5

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

Hint:Use min function

```
[40]: %sql SELECT MIN("Date") FROM SPACEXTBL where "Landing_Outcome" = 'Success (ground pad)'

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

[40]:

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- SELECT DISTINCT query with WHERE clause is used to list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 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

[37]:
```
%%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTBL
                        where "PAYLOAD_MASS__KG_" > 4000
                        AND "PAYLOAD_MASS__KG_" < 6000
                        AND "Landing_Outcome" = 'Success (drone ship)';
```

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

[37]:

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

Total Number of Successful and Failure Mission Outcomes

COUNT() and CASE WHEN are used to calculate the total number of successful and failure mission outcomes

Task 7 ¶

List the total number of successful and failure mission outcomes

[86]:
```
%%sql SELECT
COUNT(CASE WHEN "Mission_Outcome" LIKE 'Success%' THEN 1 ELSE NULL END) AS "SUCCESS",
COUNT(CASE WHEN "Mission_Outcome" LIKE 'Failure%' THEN 1 ELSE NULL END) AS "FAILURE",
COUNT("Mission_Outcome") AS "TOTAL" FROM SPACEXTBL;
```

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

[86]:

SUCCESS	FAILURE	TOTAL
100	1	101

Boosters Carried Maximum Payload

The IN() subquery with MAX() aggregating function is used to list the names of the booster which have carried the maximum payload mass

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

[84]:
```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" IN (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL) ;
```

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

[84]:

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

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

```
[106]: %%sql
SELECT DISTINCT STRFTIME('%m', "Date") AS month,
"Landing_Outcome", "Booster_Version", "Launch_Site"
FROM SPACEXTBL WHERE "Landing_Outcome" = "Failure (drone ship)" AND STRFTIME('%Y', "Date") = '2015';

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

[106]:

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

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.

[123…

```
%%sql
WITH counts_table AS (SELECT "Landing_Outcome", COUNT(*) AS "launches_count" FROM SPACEXTBL
        WHERE "Date" >= '2010-06-04' AND "Date" <= '2017-03-20' GROUP BY "Landing_Outcome"),
     ranked_counts_table AS (SELECT DENSE_RANK() OVER (ORDER BY "launches_count" DESC) AS "rank", * FROM counts_table)
SELECT * FROM ranked_counts_table ORDER BY "launches_count" DESC;
```

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

[123]:

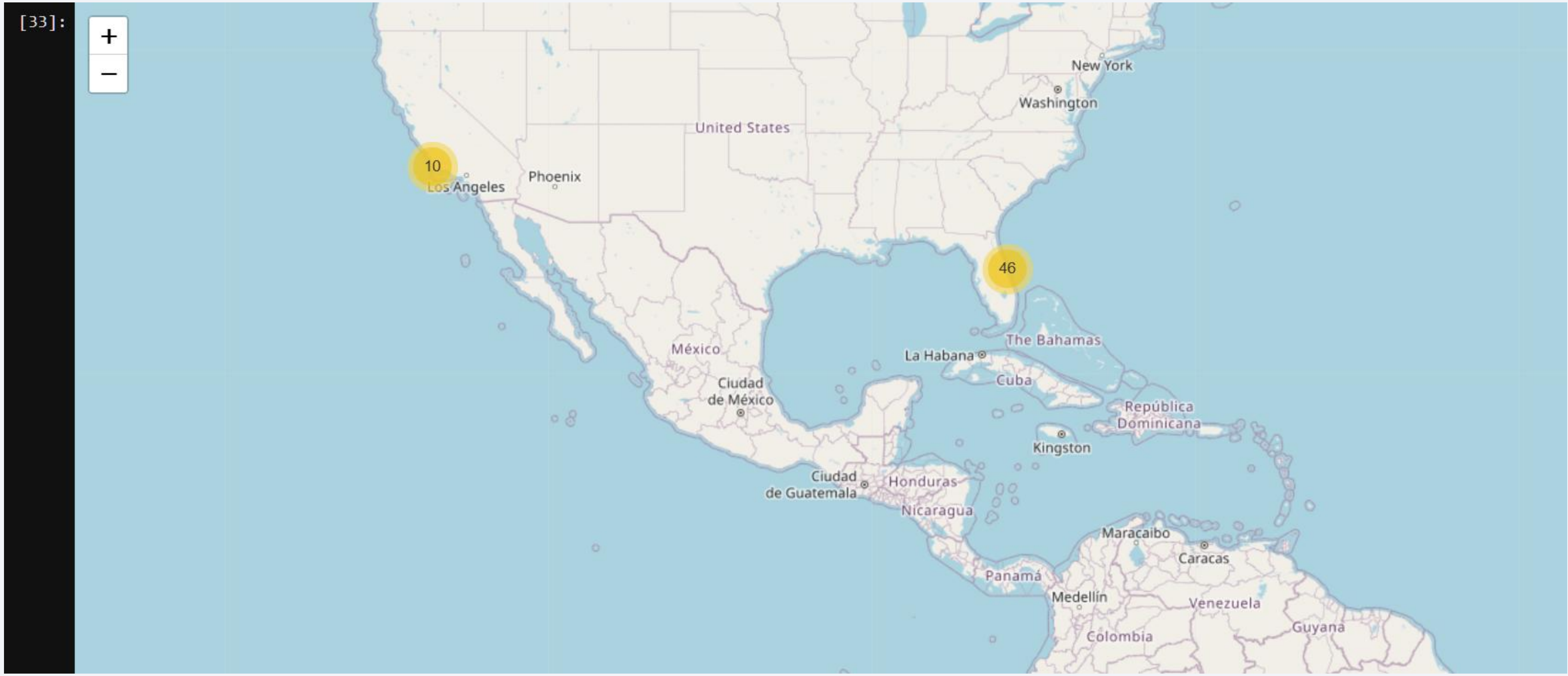
rank	Landing_Outcome	launches_count
1	No attempt	10
2	Failure (drone ship)	5
2	Success (drone ship)	5
3	Controlled (ocean)	3
3	Success (ground pad)	3
4	Failure (parachute)	2
4	Uncontrolled (ocean)	2
5	Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

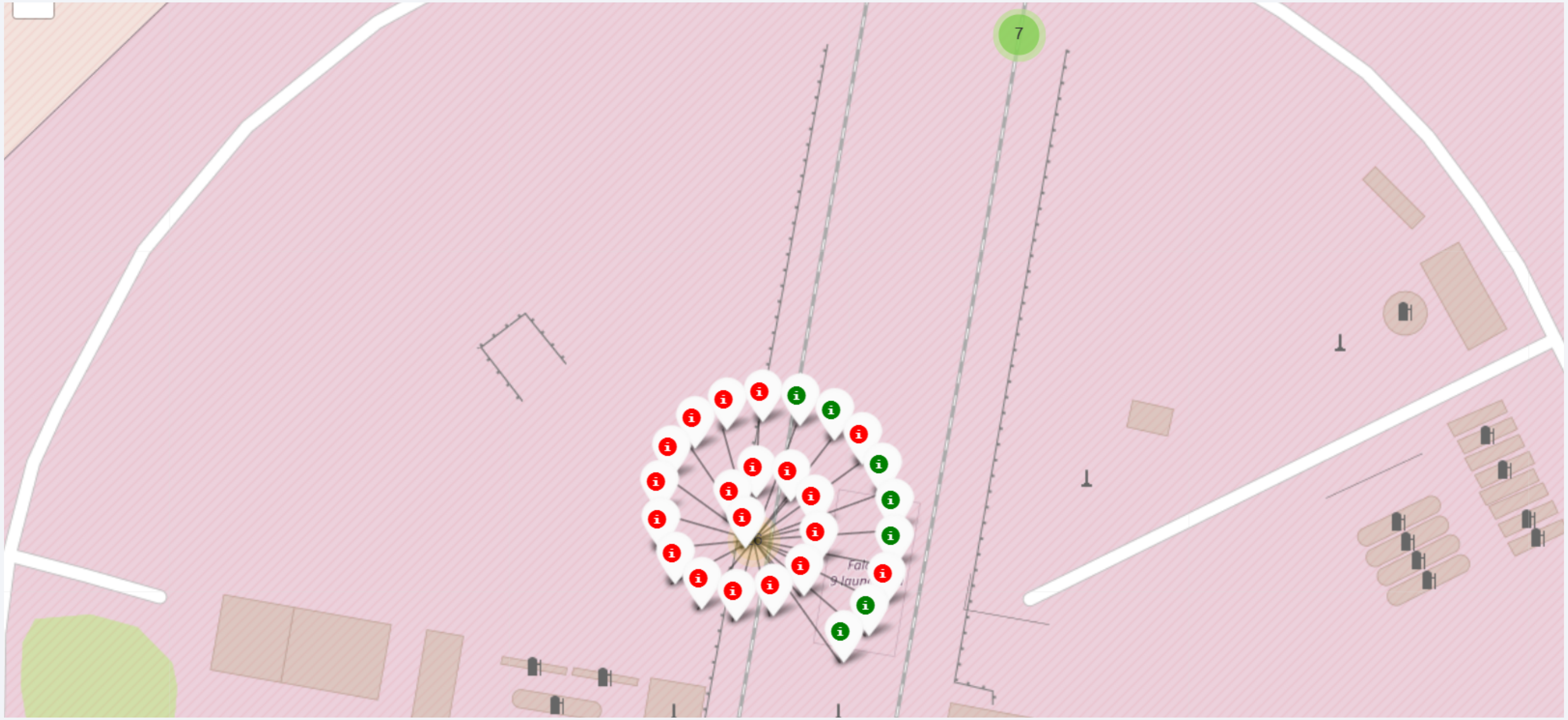
SpaceX launch site locations with landing outcomes

- This map displays launch site locations and landing outcomes.
- Clustering improves visualization clarity for overlapping points



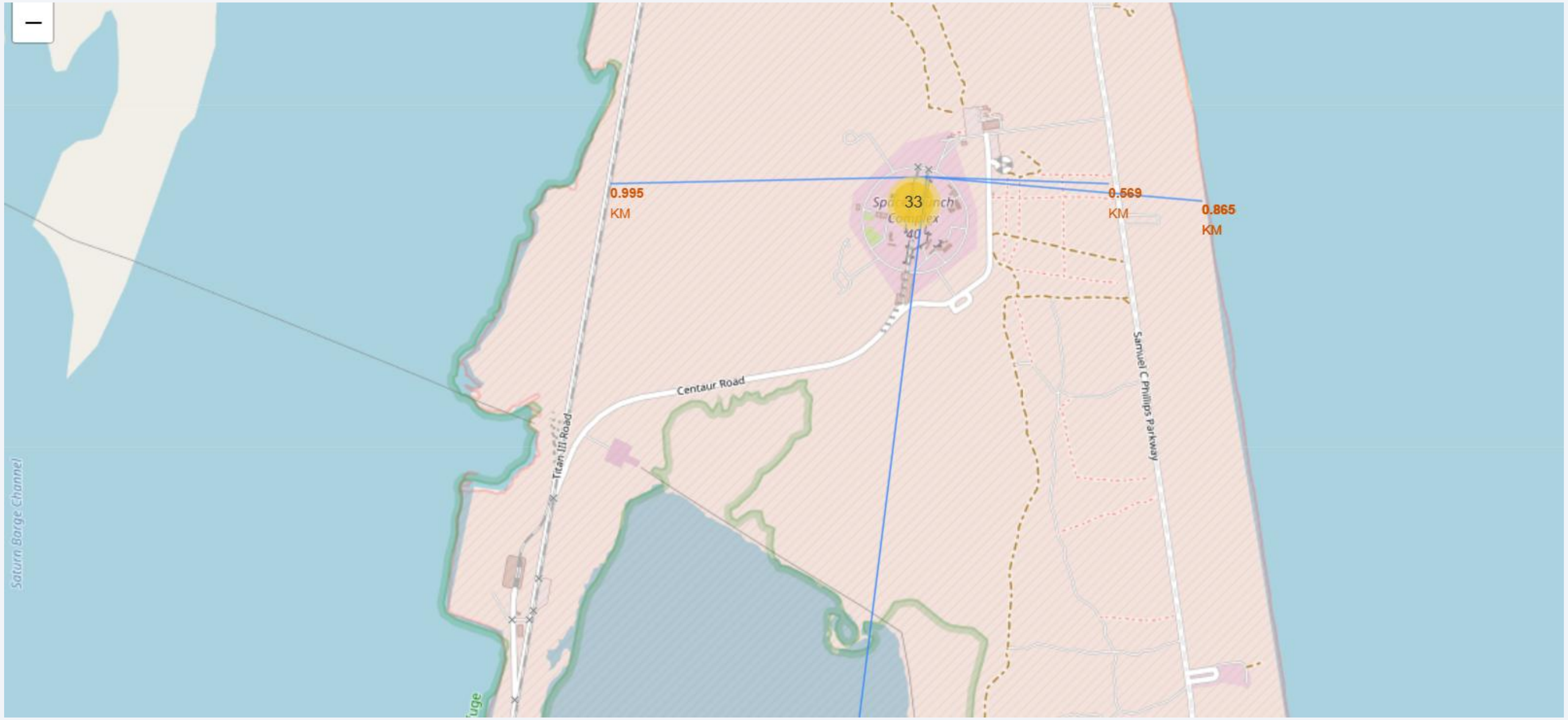
Folium Map with color-labeled launch outcomes

- Successful launch outcomes are marked with green color while failures with red



Launch site CCAFS SLC-40 and its proximities

- The launch site is located at short distance to railway, highway and coastline while staying at a considerable distance from the city.

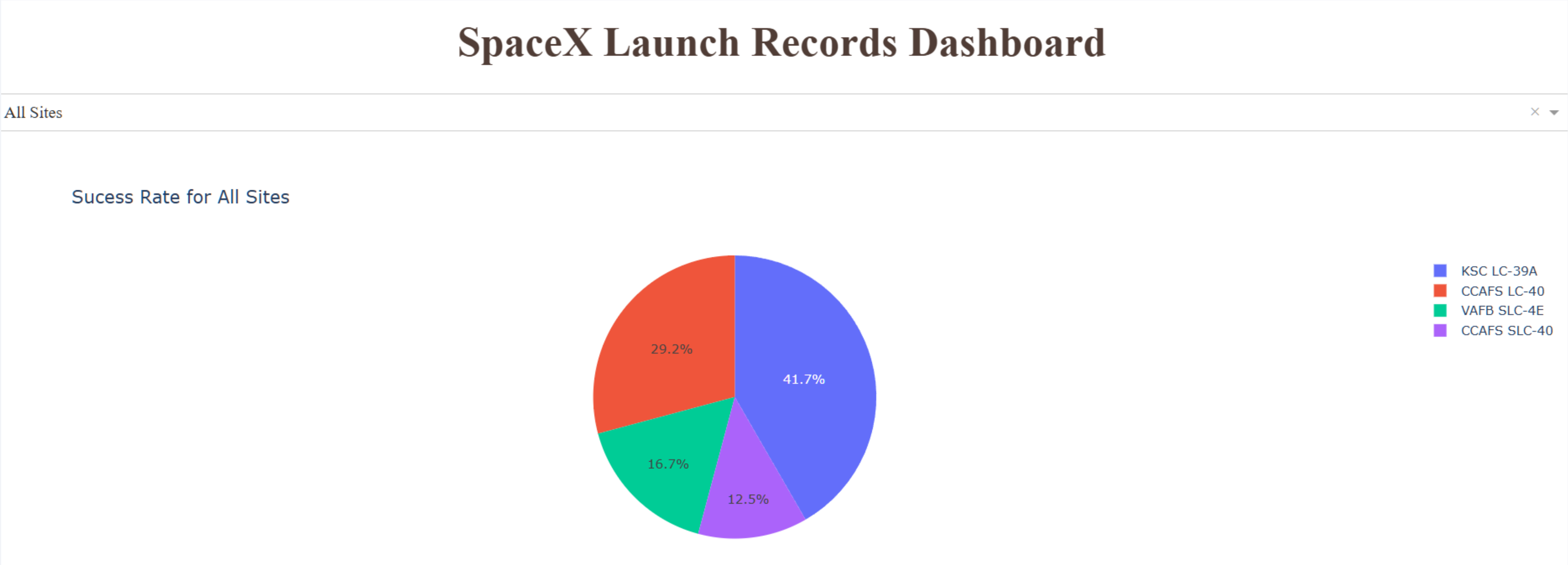


Section 4

Build a Dashboard with Plotly Dash

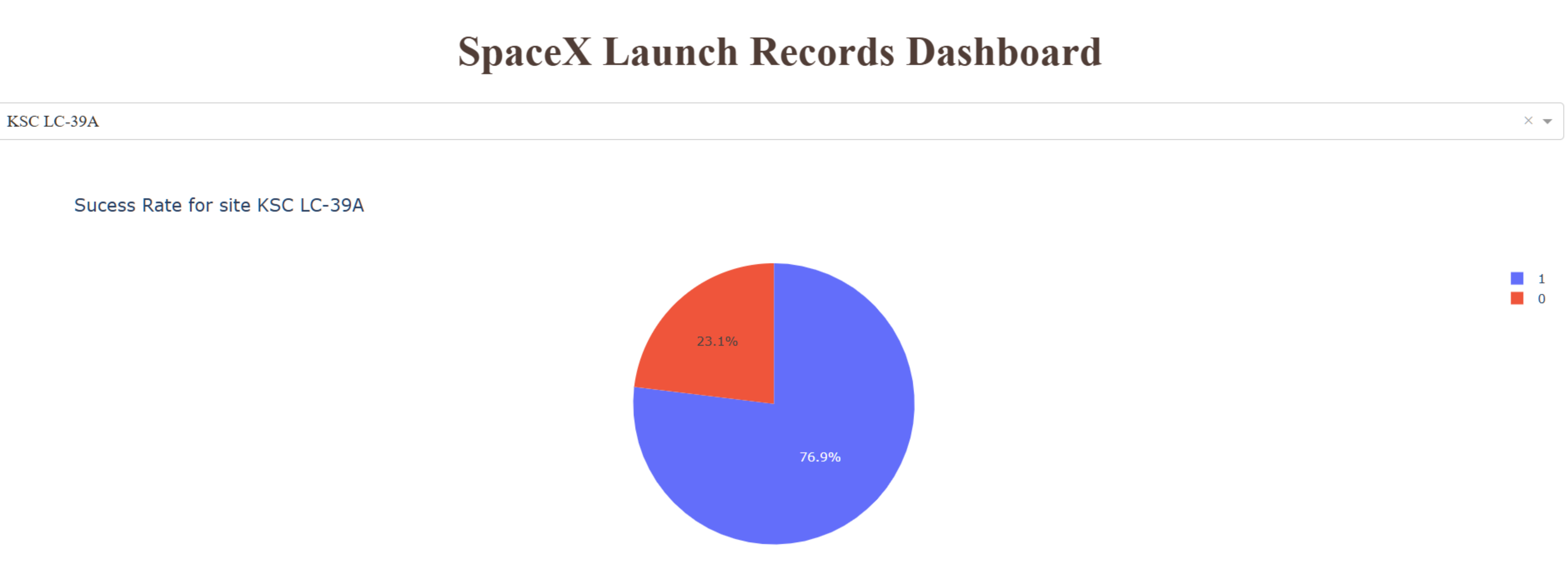
Success launches by site

- This pie chart shows success launches rate by all sites



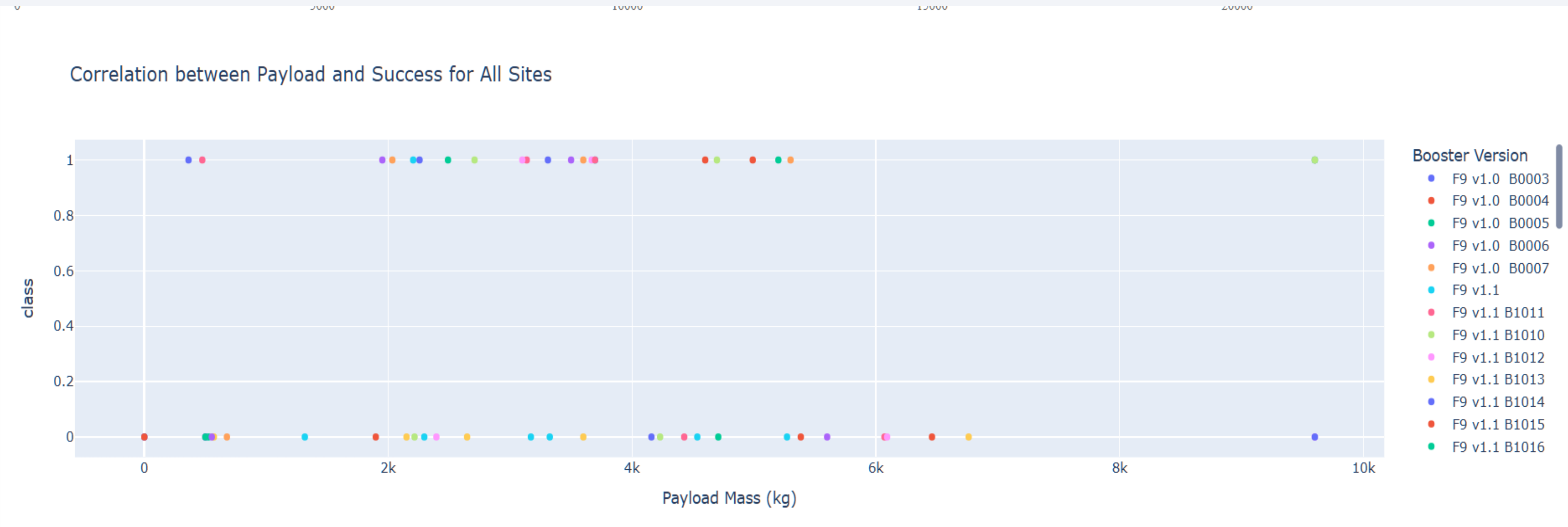
The launch site with the highest success ratio

- KSC LC-39A site has a success ratio of 76,9%



Payload vs. Launch Outcome

- Given the number and variety of booster versions there is no clear leading booster in terms of success rate.

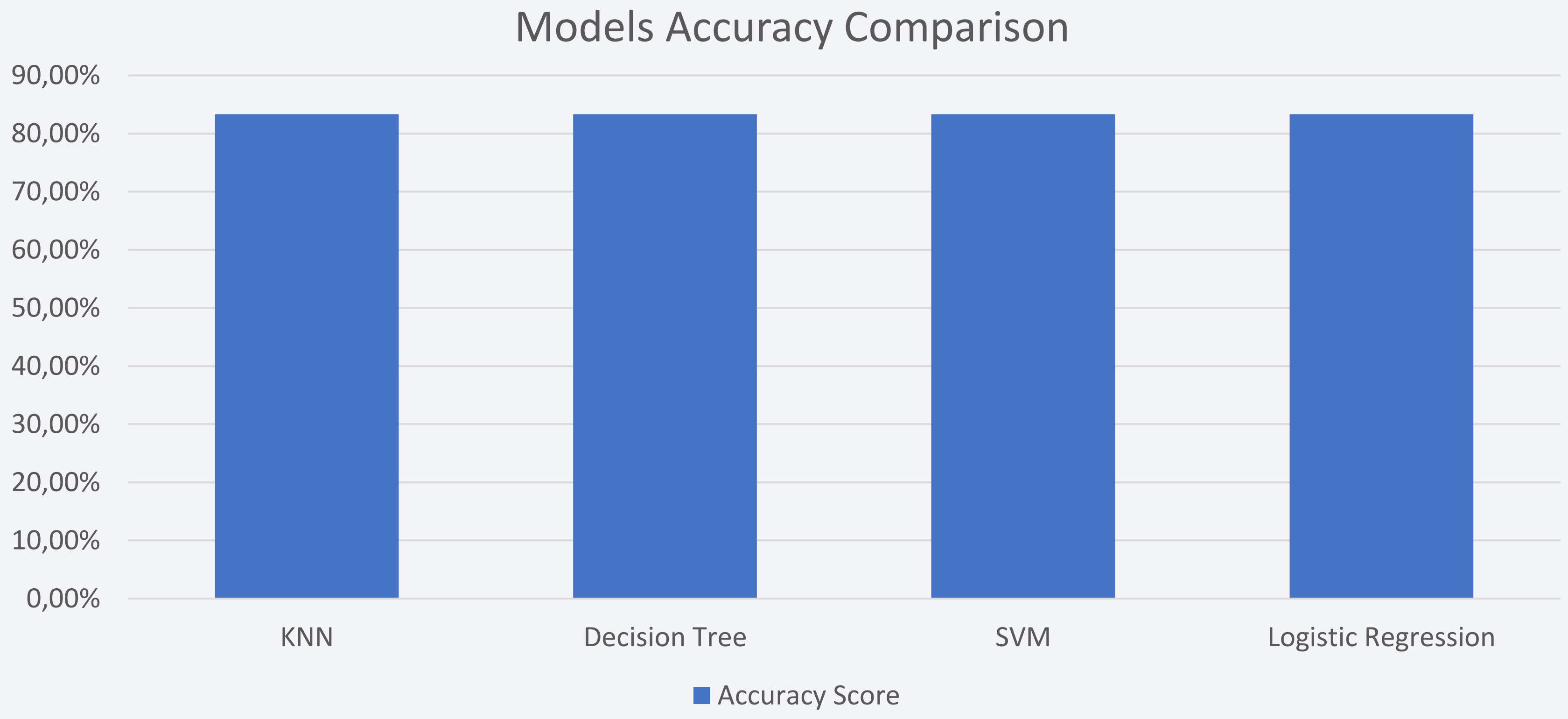


Section 5

Predictive Analysis (Classification)

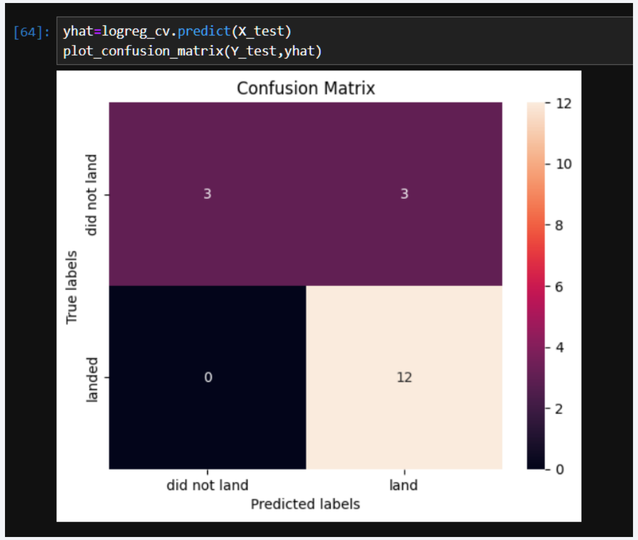
Classification Accuracy

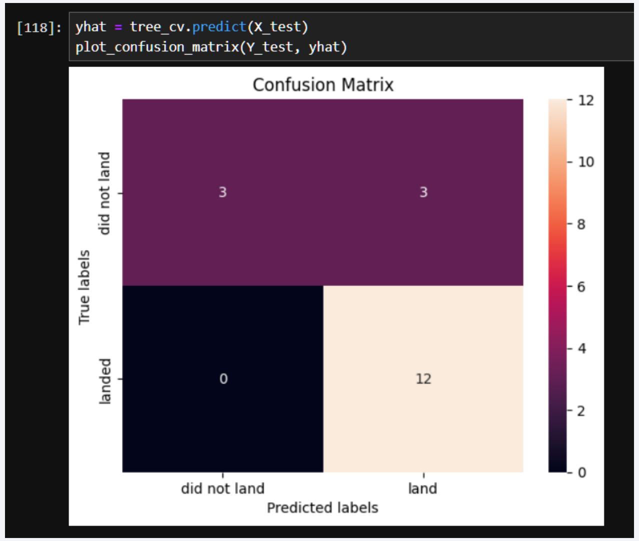
- All the models achieved similar classification performance

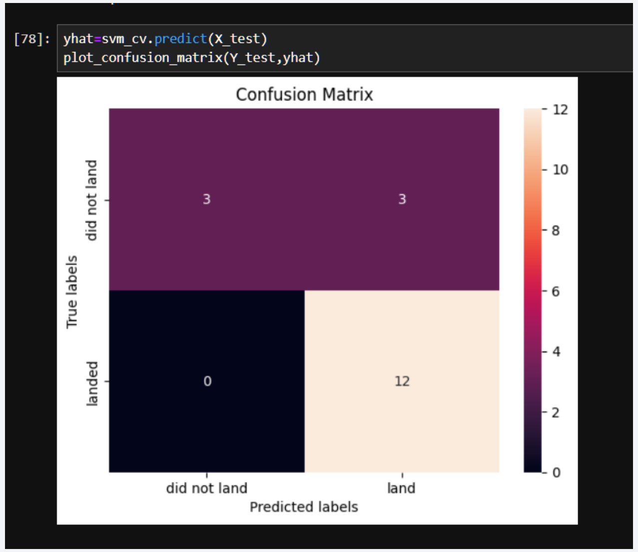


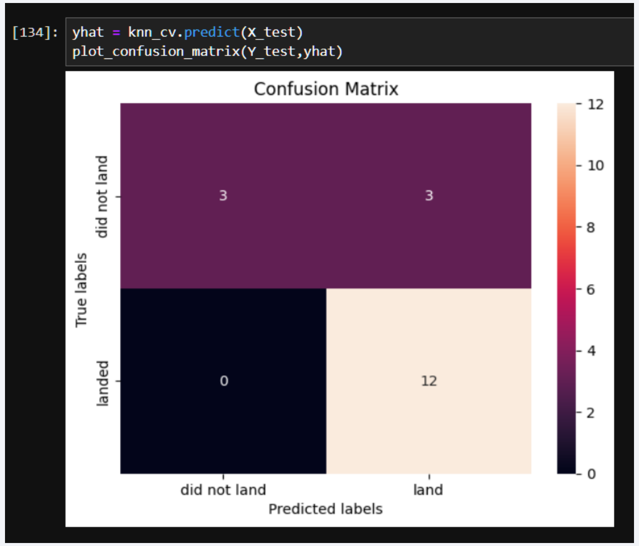
Confusion Matrix

- The evaluated models achieved comparable performance, suggesting no clear single best model for this dataset.









Conclusions

- The analysis confirms that first-stage landing success can be predicted using historical data.
- Machine learning models provide reliable classification performance.
- Data visualization reveals key operational patterns.
- The results support cost estimation for future launches.

Appendix

- The complete code, datasets and notebooks used for this project are available in the GitHub repository:

https://github.com/DennisVSIG/ibm_data_science_capstone

Thank you!