IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Joseph Hurley

12/09/2024
https://github.com/JPHAnalytics/IBMDataScience

Outline

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

Executive Summary

- An examination of space mission data indicates a favorable trend in success rates over time.
- **Key insights include:**
  - The effectiveness of KSC LC-39A as a launch site.
  - The capability of CCAFS SLC-40 to accommodate heavier payloads.
  - The critical role of strategic mission planning, as demonstrated by consistent success rates in particular orbits.
  - A noticeable rise in success rates post-2013, attributed to technological advancements and improved operational practices.
- These findings emphasize the necessity for ongoing innovation and collaboration to ensure successful space missions.

Introduction

- In this capstone, we will predict if the Falcon 9 first stage will land successfully. 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.

- **If we can determine if the first stage will land, we can determine the cost of a launch**. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. In this module, you will be provided with an overview of the problem and the tools you need to complete the course.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - RESTFul APIs
  - Web scraping using BeautifulSoup library
- Perform data wrangling
  - Python code to conduct exploratory data analysis by manipulating data in a Pandas data frame
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - SVM
  - Classification Trees
  - Regression

Data Collection

- Data collection through an API involves retrieving structured data from a remote server using predefined endpoints and protocols.
- Key steps include accessing the API, sending a request with parameters specifying the desired data, receiving and parsing the response, and storing the data for further analysis.
- APIs provide advantages such as real-time data access, automation of retrieval processes, and interoperability, but users must adhere to terms of use and respect usage limits.

Data Collection – SpaceX API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

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

```
# Use json_normalize meethod to convert
response_json = response.json()
df = pd.json_normalize(response_json)
```

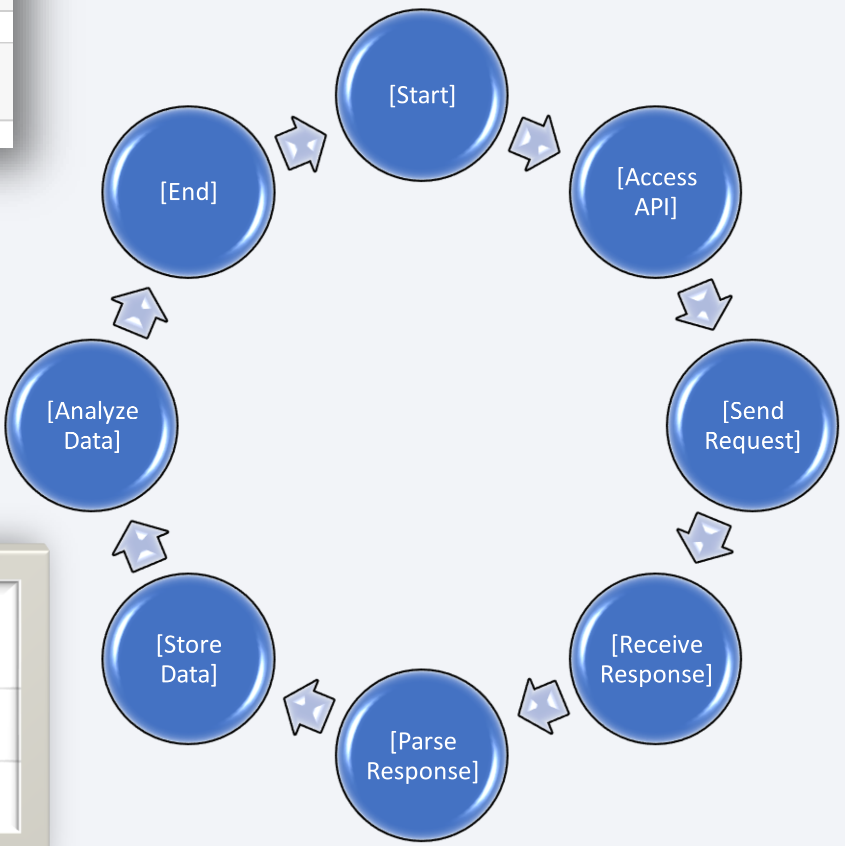
Using the dataframe `data` print the first 5 rows

```
# Get the head of the dataframe
df.head()
```

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

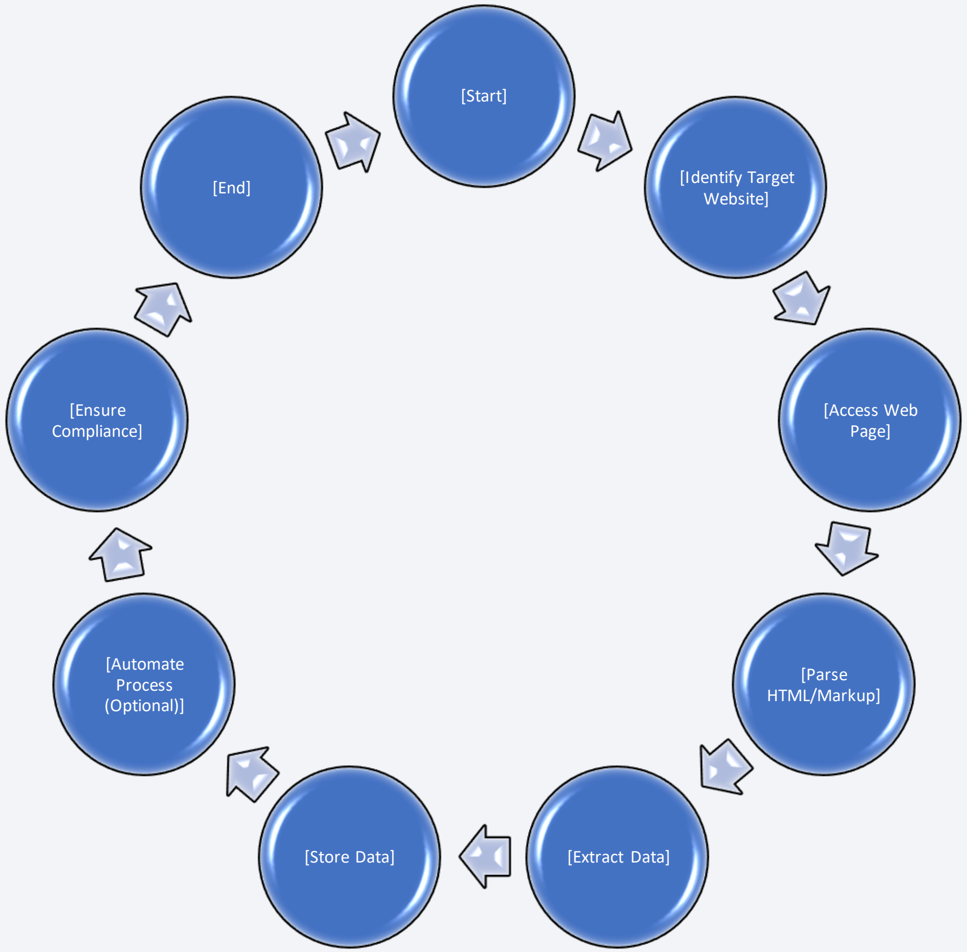
```
response.status_code
```

```
0]:  200
```



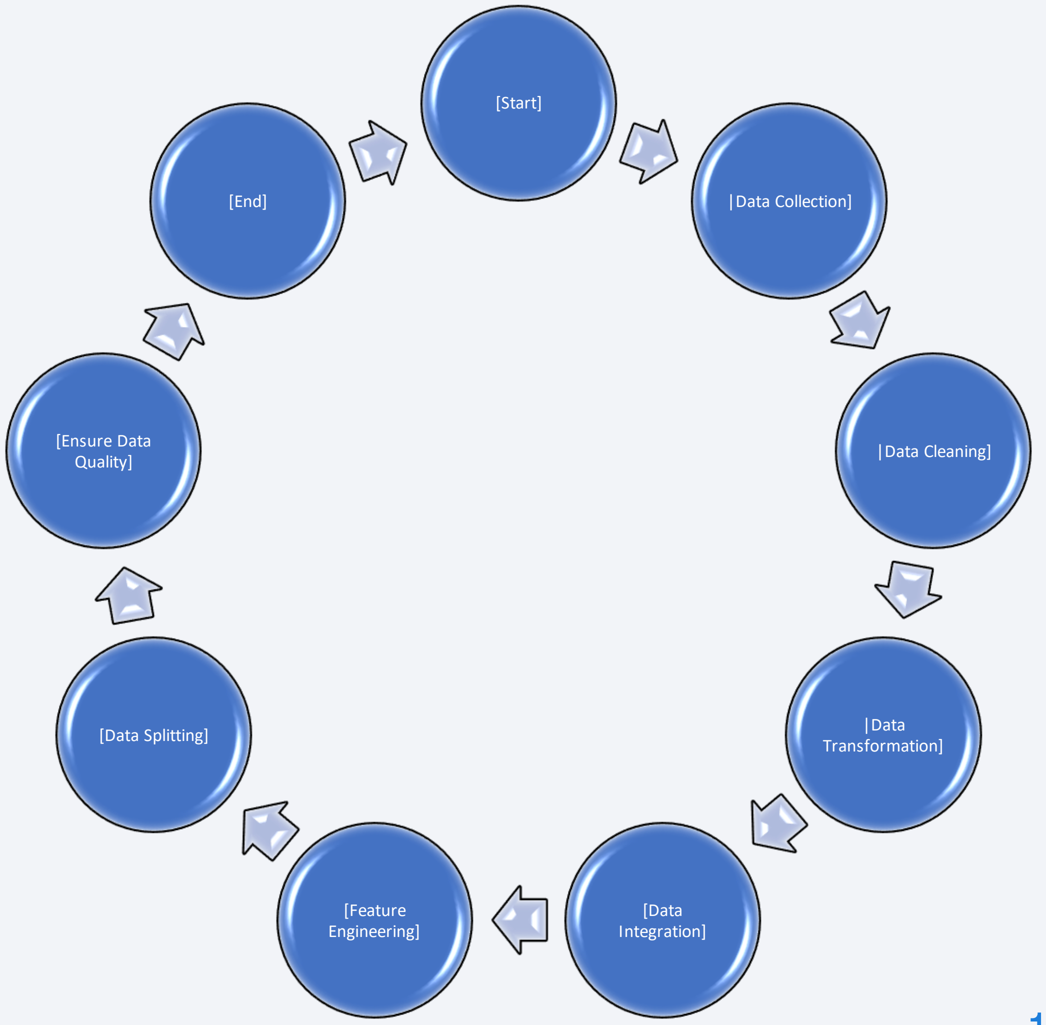
Data Collection - Scraping

- Web scraping is a powerful technique for programmatically extracting data from websites.
- It involves accessing web pages, parsing their HTML or other markup formats, and retrieving the specific information needed for analysis.
- Data scientists often leverage libraries and frameworks in languages like Python or JavaScript to facilitate this process, automate data collection, and efficiently parse HTML content.
- It's essential to remain mindful of legal and ethical considerations while implementing web scraping practices



Data Wrangling

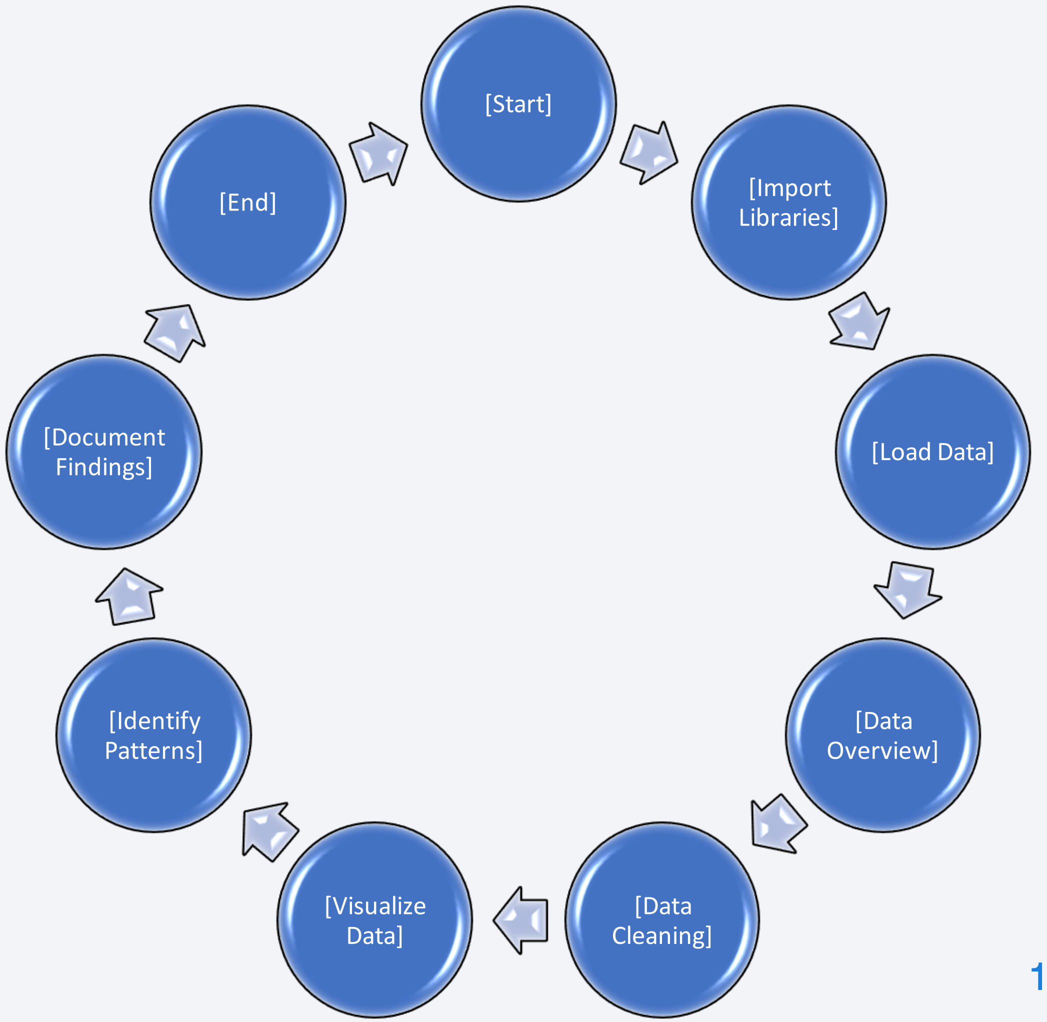
- Data wrangling, often referred to as data munging, is the process of cleaning, transforming, and preparing raw data for analysis.
- This critical phase includes several key steps: data collection, data cleaning, transformation, integration, feature engineering, and data splitting.
- Effective data wrangling ensures that the dataset is accurate, complete, and structured appropriately, enabling meaningful analysis and robust modeling.



EDA with Data Visualization

- Exploratory Data Analysis (EDA) was performed on the extracted SpaceX data to identify underlying patterns and insights.

- Essential libraries utilized in this process included pandas for data manipulation, matplotlib as a versatile plotting library, seaborn for advanced visualizations built on top of matplotlib, and numpy for efficient array operations.



EDA with SQL

- Exploratory Data Analysis (EDA) using SQL entails querying a database to directly explore and analyze datasets.
- This process includes tasks such as describing the data, sampling, managing missing values, examining relationships, and identifying patterns.
- While SQL is not primarily designed for visualization, it does offer some basic capabilities in this area.
- EDA with SQL provides an efficient means of gaining insights into large datasets within the database environment.

- Conducted EDA with SQL on SpaceX dataset.
- Tasks included showcasing unique launch site names and retrieving records with specific launch site strings.
- Calculated total payload mass for NASA's CRS program and average payload mass for booster version F9 v1.1.
- Identified date of first successful ground pad landing outcome.
- Listed boosters with successful drone ship landings within specified payload mass range.
- Tabulated total successful and failed mission outcomes.
- Identified booster versions with maximum payload mass using subquery.
- Compiled records to display month names, failure landing outcomes, booster versions, and launch sites for 2015.
- Ranked count of landing outcomes between specified dates

Build an Interactive Map with Folium

Folium facilitates interactive visual analytics in Python by generating dynamic maps for exploring geospatial data. Users can plot data points, customize map styles, and interact with various map elements such as markers and polygons. The integration of visualizations and widgets enables a deeper understanding of spatial patterns and relationships, thereby enhancing the analysis of geospatial data.

Using Folium maps, the following actions were performed to discover geographical patterns related to launch sites:

- Marking all launch sites on a map.
- Indicating the success or failure of launches for each site on the map.
- Calculating the distances between a launch site and its proximities.

Build a Dashboard with Plotly Dash

Plotly Dash enables users to build interactive web-based dashboards in Python without requiring any additional web development expertise. These dashboards include interactive elements such as dropdowns and sliders, responsive layouts, and customizable styling options.

A Plotly Dash application was developed to facilitate interactive visual analytics on real-time SpaceX launch data.

- Input components, including a dropdown list and a range slider, were integrated to enhance user interaction with the visualizations.
- Key tasks included adding a launch site dropdown and range slider, as well as implementing callback functions to render pie and scatter plots.
- The dashboard allows users to dynamically explore SpaceX launch data through its interactive features.
- Participants were guided through the process of building various components of the dashboard, enabling comprehensive data analysis

Predictive Analysis (Classification)

Machine learning prediction models leverage historical data to uncover patterns and forecast future outcomes or classify new data points. They are utilized across a wide range of domains for tasks such as stock price forecasting, predicting customer behavior, diagnosing diseases, and more. These models encompass various types, including regression, classification, clustering, and deep learning, each designed to address specific data characteristics and tasks.

The following tasks were accomplished:

- Conducted exploratory data analysis to identify training labels.
- Created a column for the class.
- Standardized the dataset.
- Divided the dataset into training and test sets.
- Determined the optimal hyperparameters for Support Vector Machines, Classification Trees, and Logistic Regression.
- Assessed the performance of each method using the test dataset.

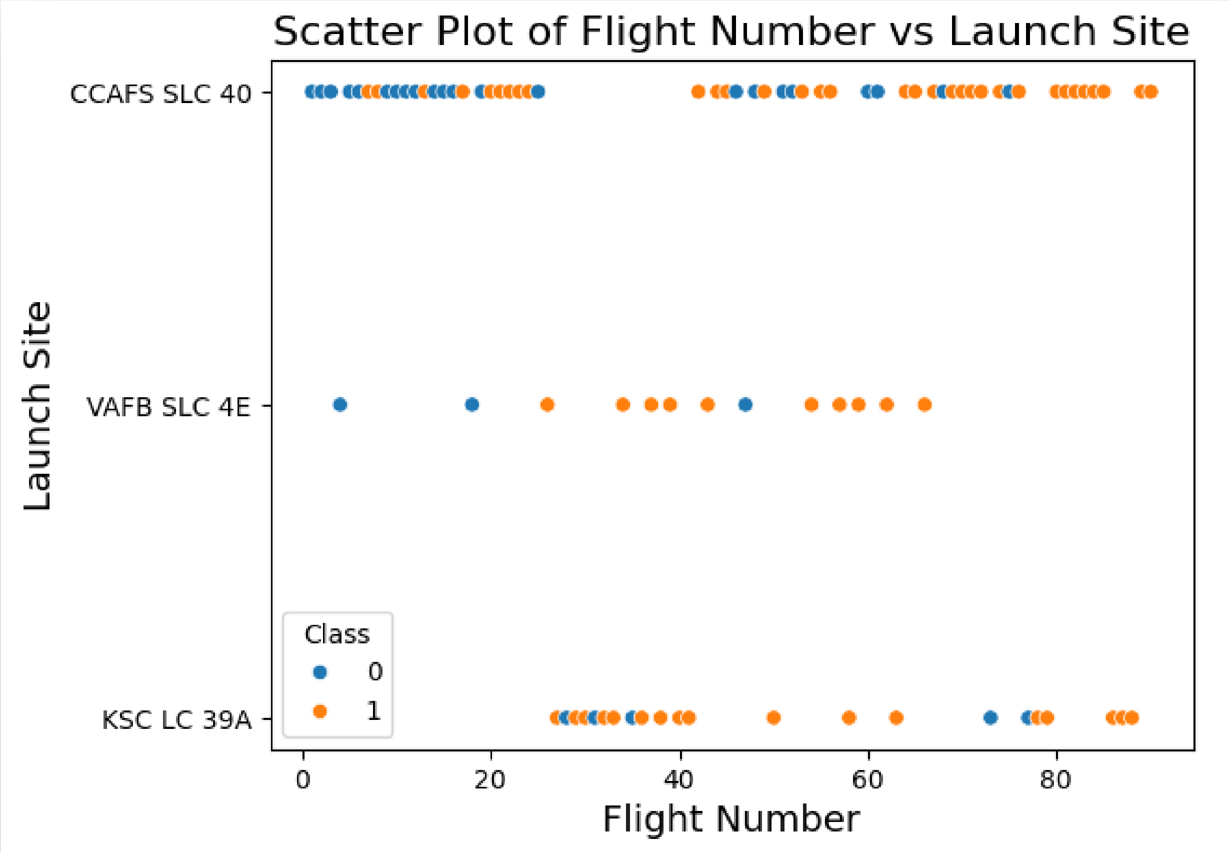
Results

- Exploratory data analysis (EDA) results
- Interactive analytics demo in screenshots
- Predictive analysis results

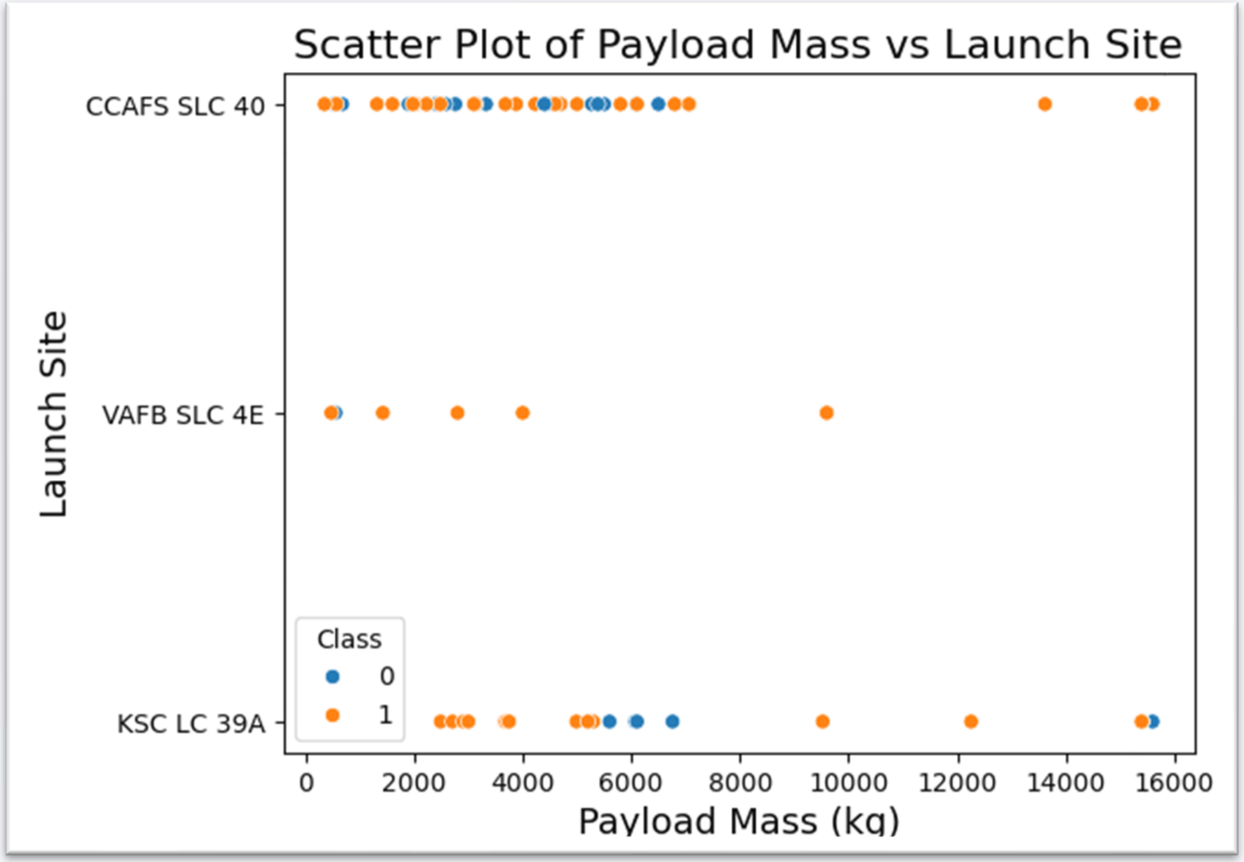
Section 2

Insights drawn from EDA

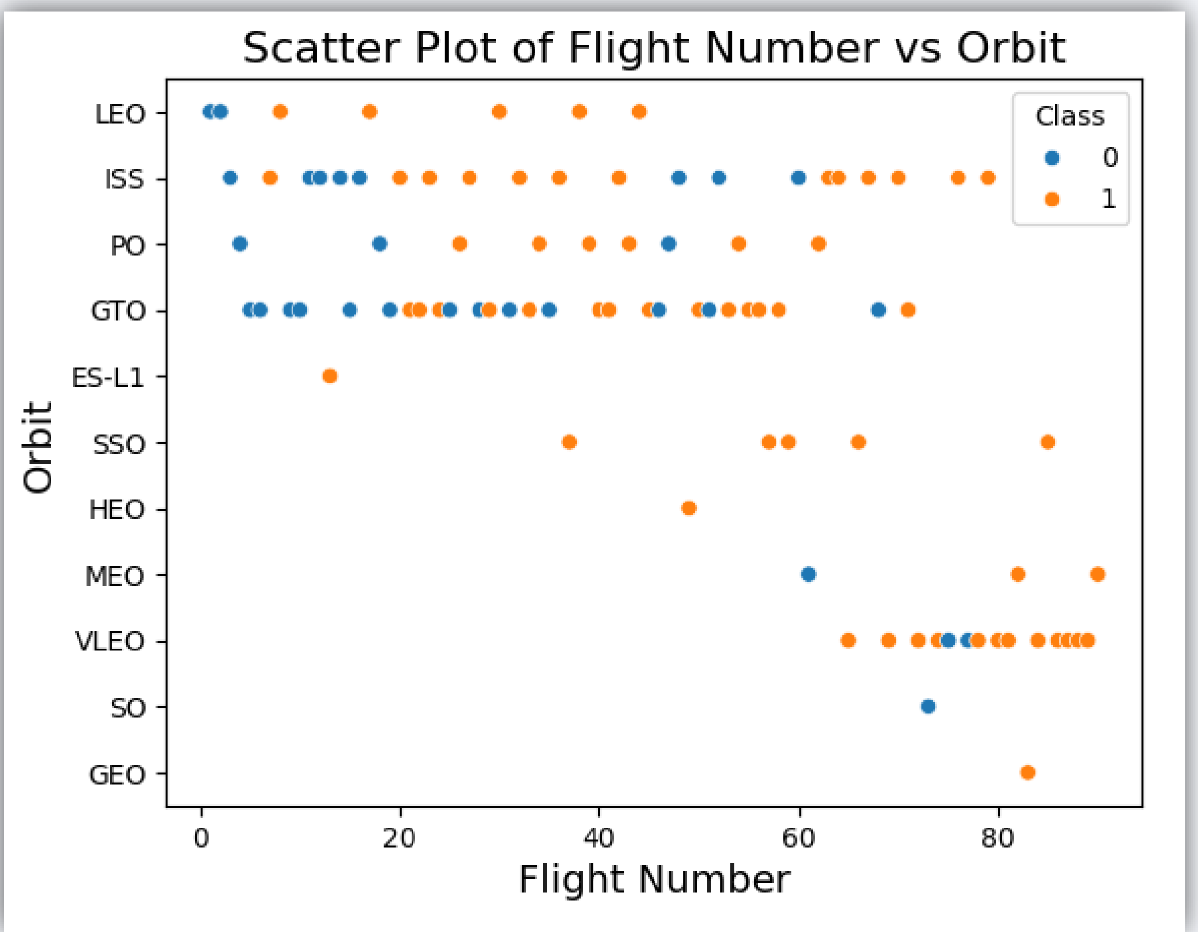
Flight Number vs. Launch Site



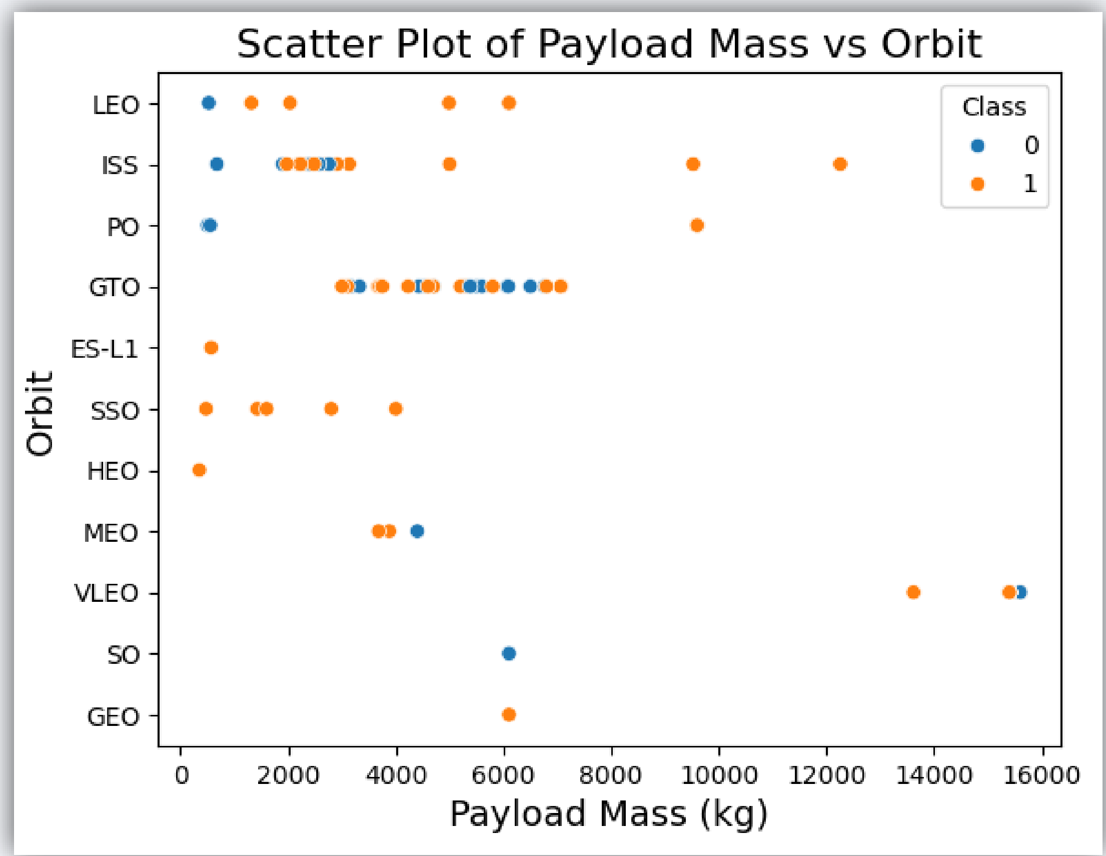
Payload vs. Launch Site



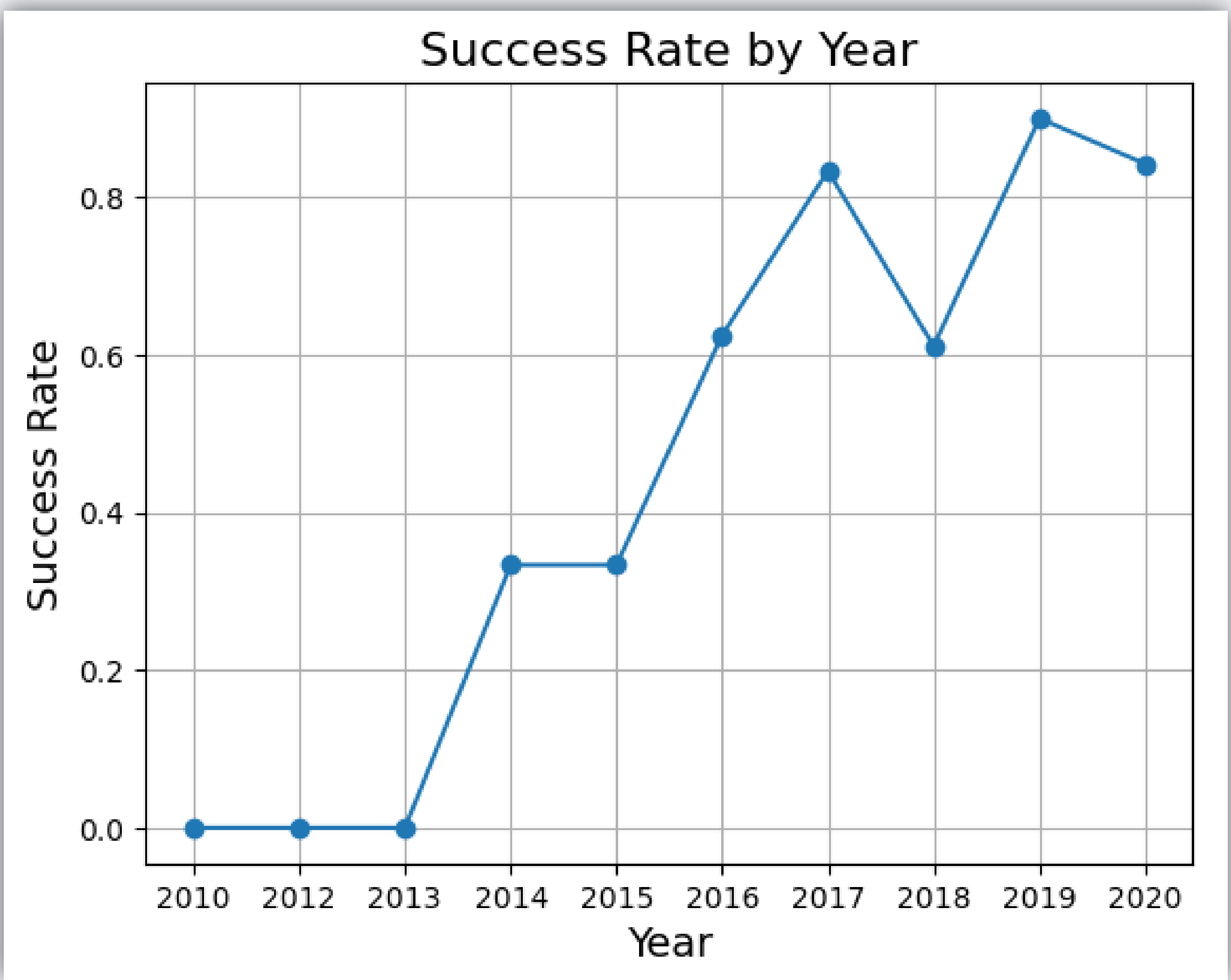
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

```
[10]:  %sql Select distinct Launch_Site from SPACEXTABLE

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

Out[10]:

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

Launch Site Names Begin with 'CCA'

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

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

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

5]:

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 - NASA

Task 3

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

]:

```
%%sql

SELECT SUM(PAYLOAD_MASS__KG_) AS TotalPayloadMass
FROM SPACEXTABLE
WHERE Customer LIKE '%CRS%';
```

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

t[14]:

TotalPayloadMass
48213

Average Payload Mass by F9 v1.1

Task 4

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

```
%%sql

SELECT AVG(PAYLOAD_MASS__KG_) AS AveragePayloadMass
FROM SPACEXTABLE
WHERE Booster_Version = 'F9 v1.1';
```

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

[15]:

AveragePayloadMass
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

```
]:   %%sql

     SELECT MIN(Date) AS FirstSuccessfulLandingDate
     FROM SPACEXTABLE
     WHERE Landing_Outcome LIKE 'Success%Ground Pad%';
```

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

:[18]:

FirstSuccessfulLandingDate
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql SELECT "Booster_Version", "Landing_Outcome","PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (drone ship)'
    and ("PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000)
```

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

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%%sql

SELECT Mission_Outcome, COUNT(*) AS TotalCount
FROM SPACEXTABLE
WHERE Mission_Outcome LIKE 'Success%' OR Mission_Outcome LIKE 'Failure%'
GROUP BY Mission_Outcome;
```

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

22]:

Mission_Outcome	TotalCount
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%%sql

SELECT Booster_Version
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

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

23]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4

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.

```
%sql SELECT SUBSTR("Date",6,2) AS Mon,SUBSTR("Date",0,5) AS Yr,"Landing_Outcome","Booster_Version","Launch_Site"
    FROM SPACEXTBL WHERE SUBSTR("Date",0,5)='2015' AND "Landing_Outcome" = 'Failure (drone ship)'
```

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

Mon	Yr	Landing_Outcome	Booster_Version	Launch_Site
01	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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.

```
%%sql

SELECT
    CASE substr(Date, 6, 2)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
```

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

MonthName	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

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.

```
%sql SELECT "Landing_Outcome",COUNT(*) AS TOT FROM SPACEXTBL WHERE "Date" > '2010-06-04' and "Date" < '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY COUNT(*) DESC
```

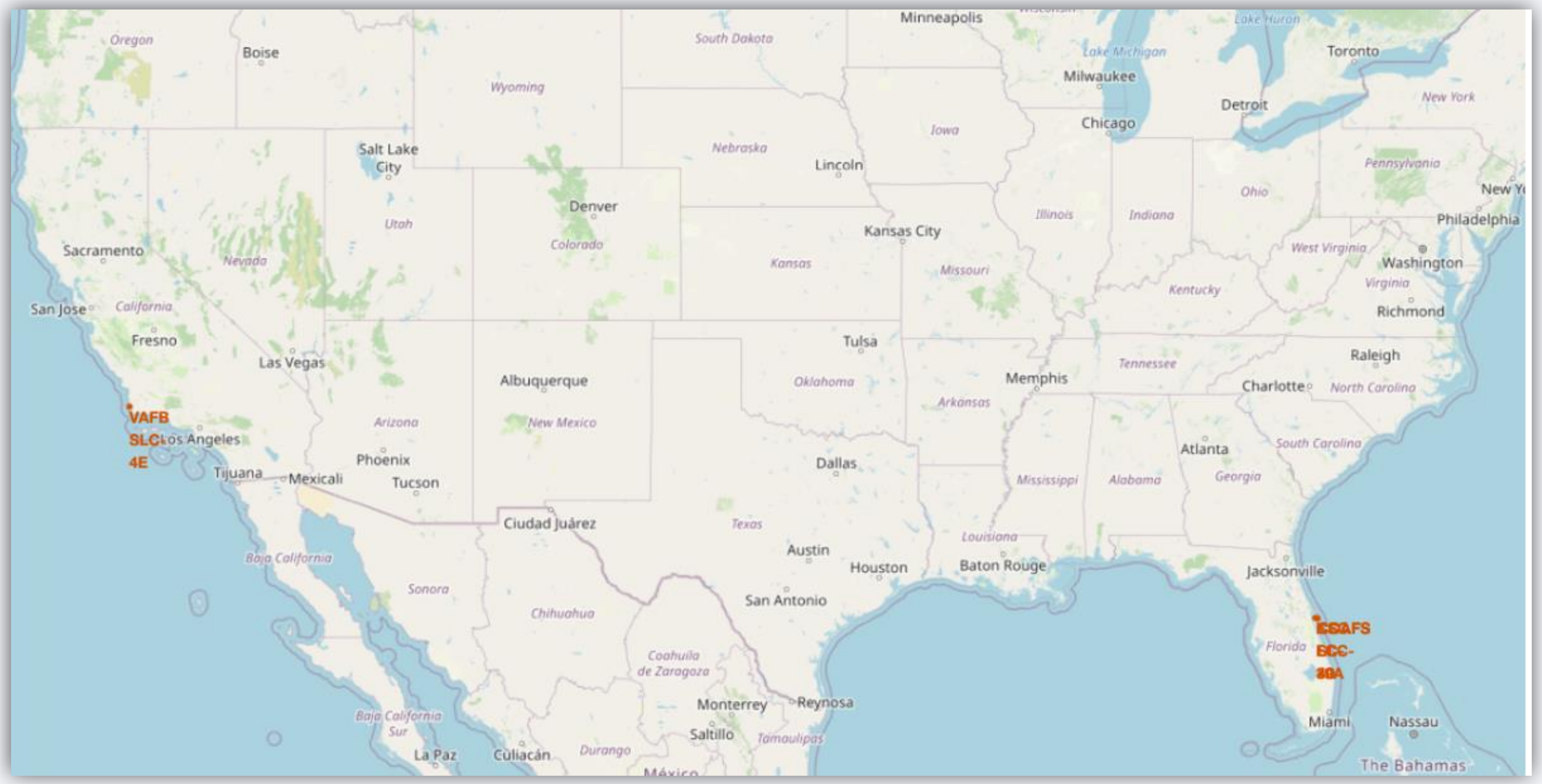
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	TOT
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

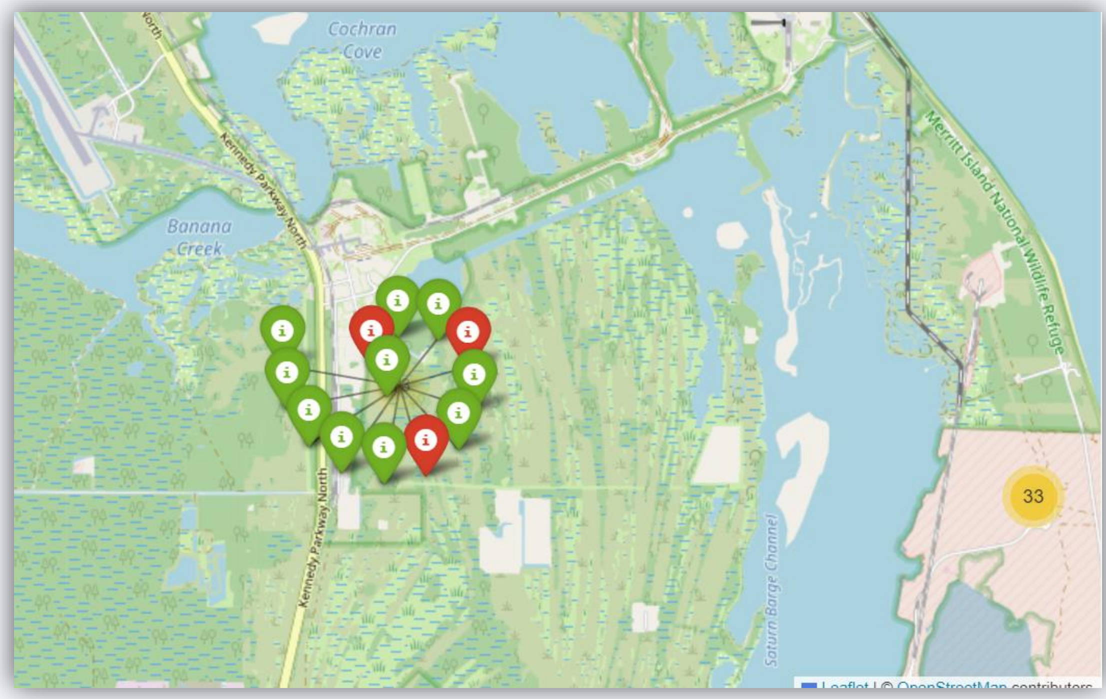
Section 3

Launch Sites Proximities Analysis

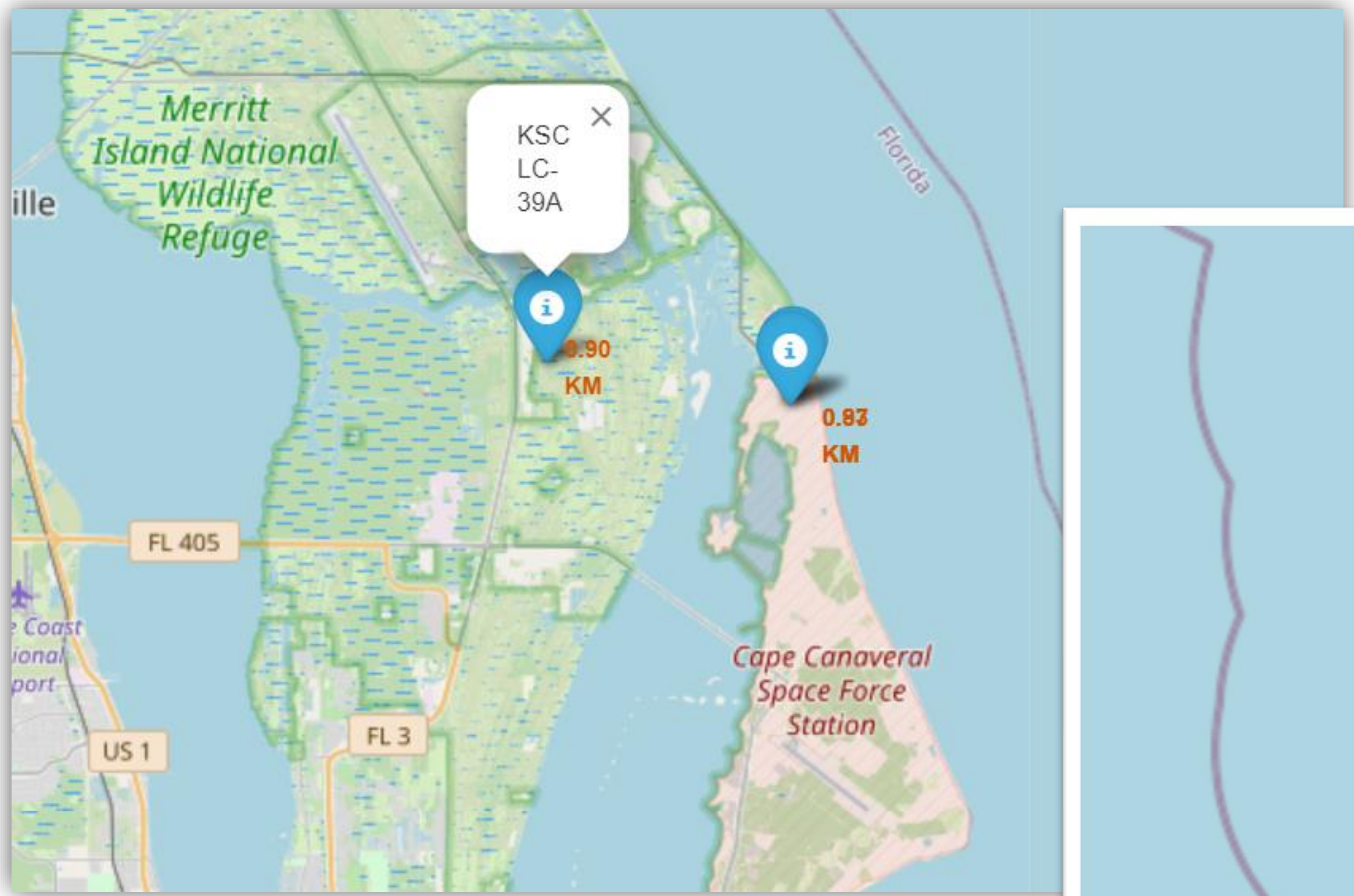
Launch Site Locations

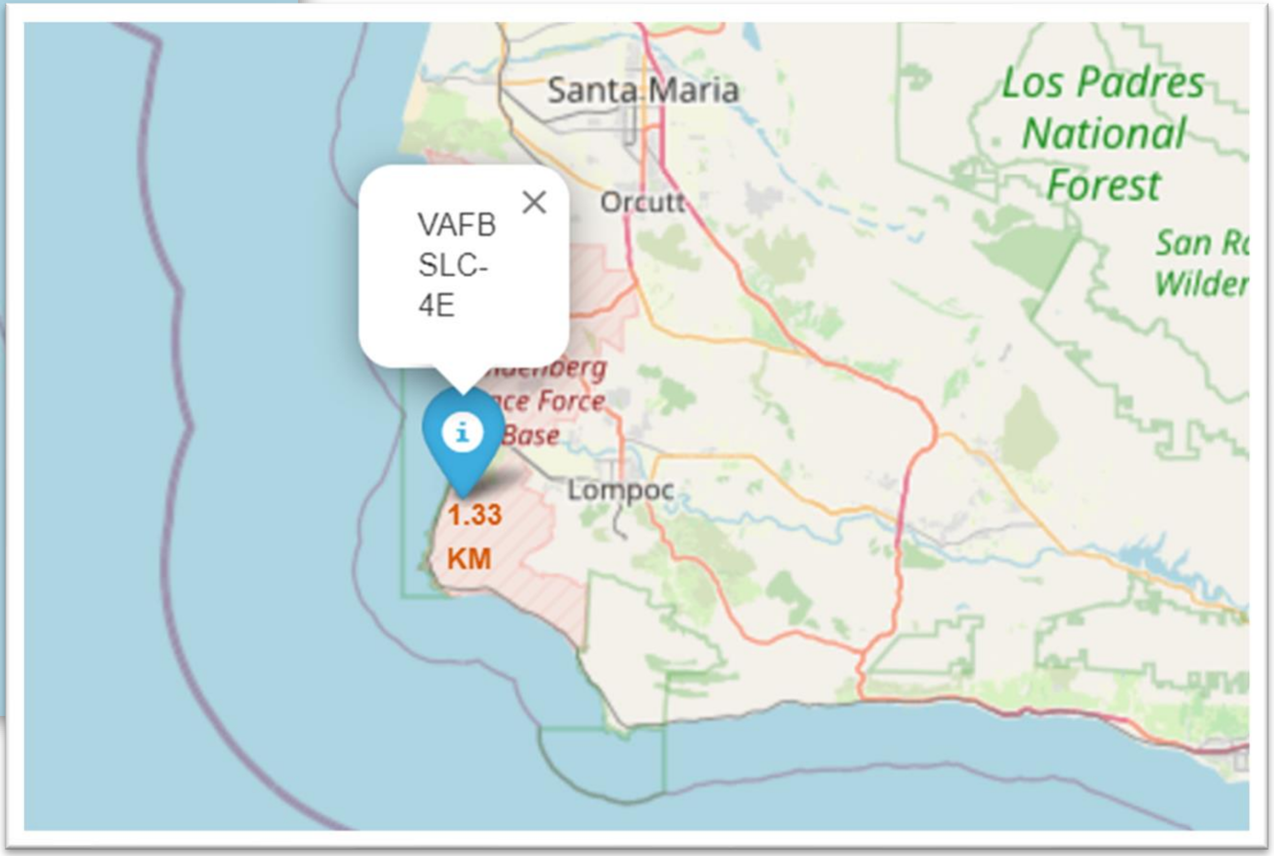


Launches Sites and Outcomes

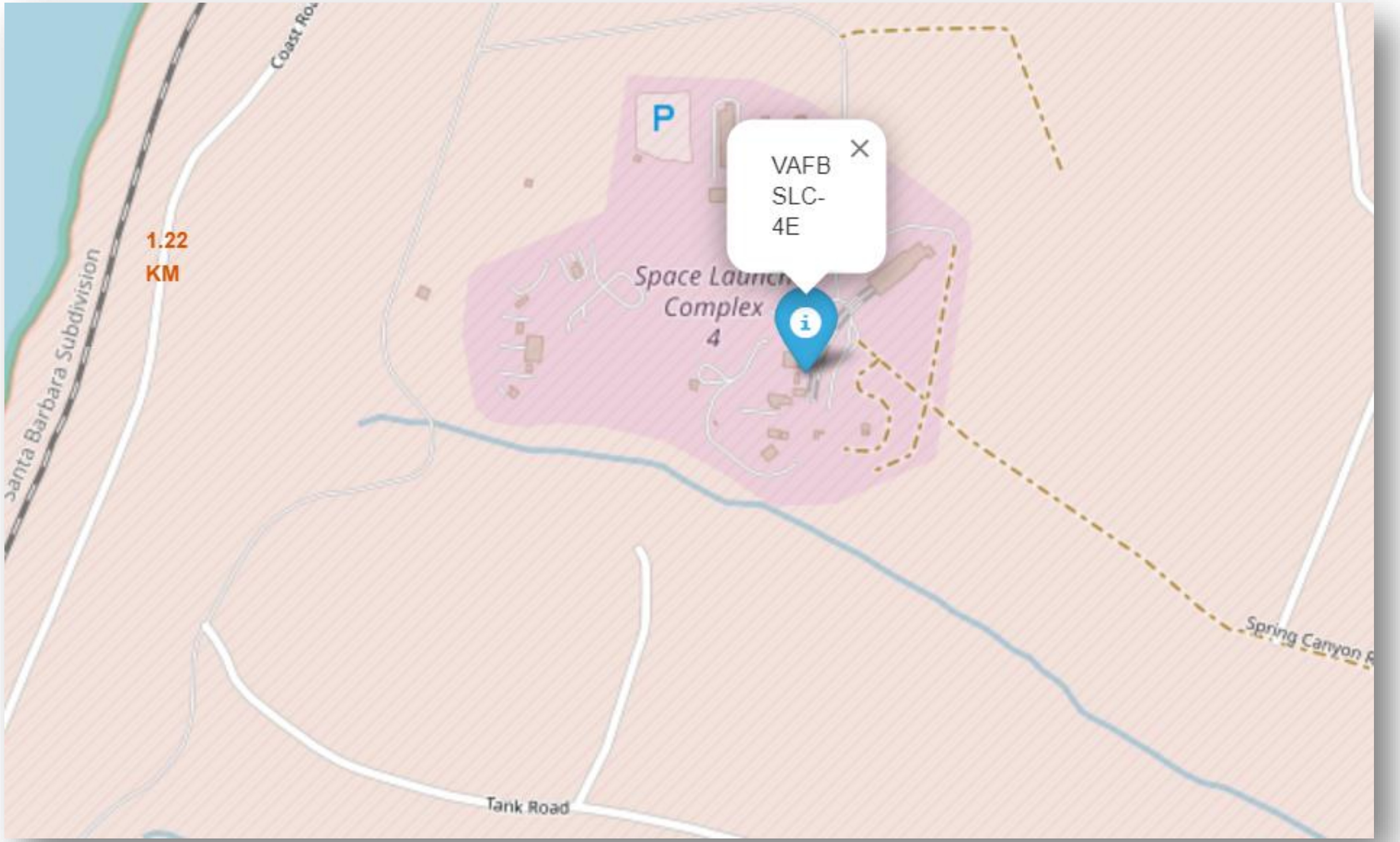


DISTANCE BETWEEN LAUNCH SITES AND NEAREST COASTLINE

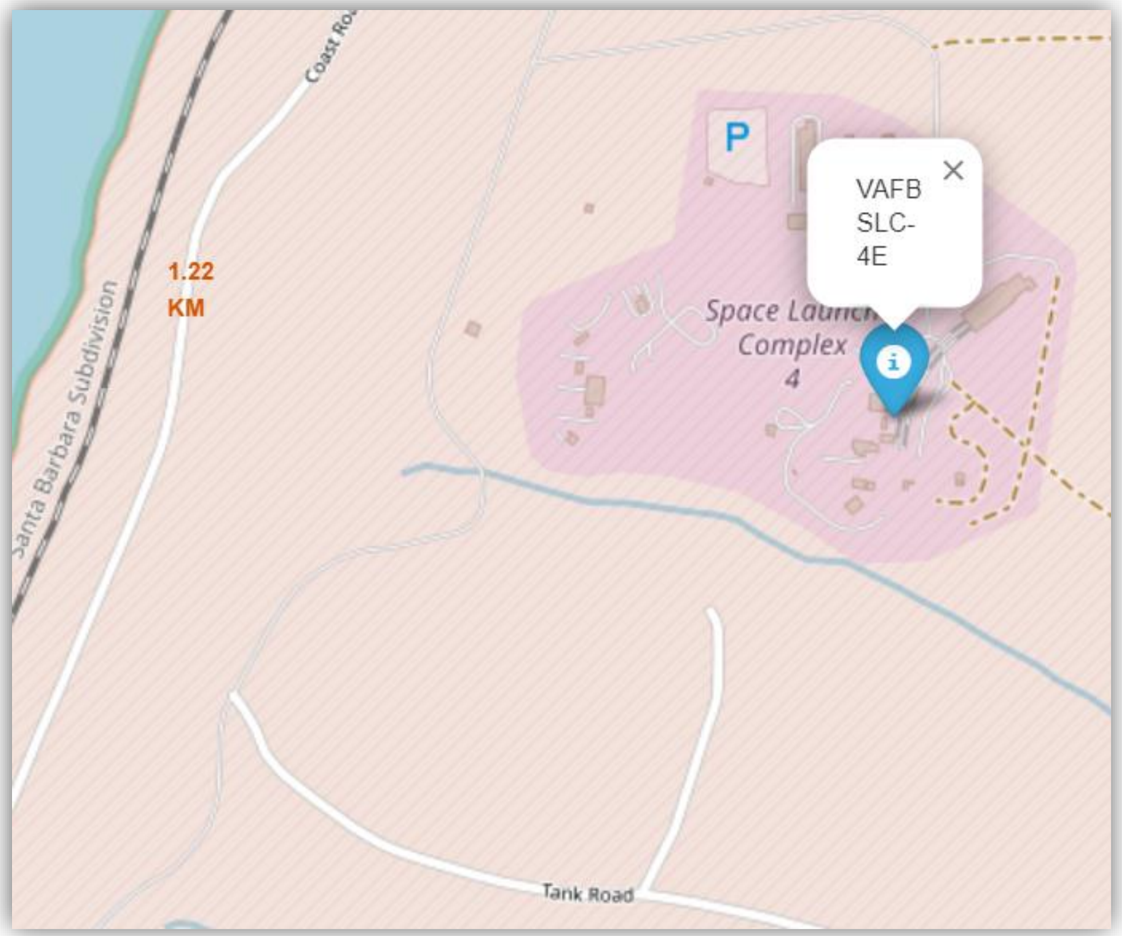


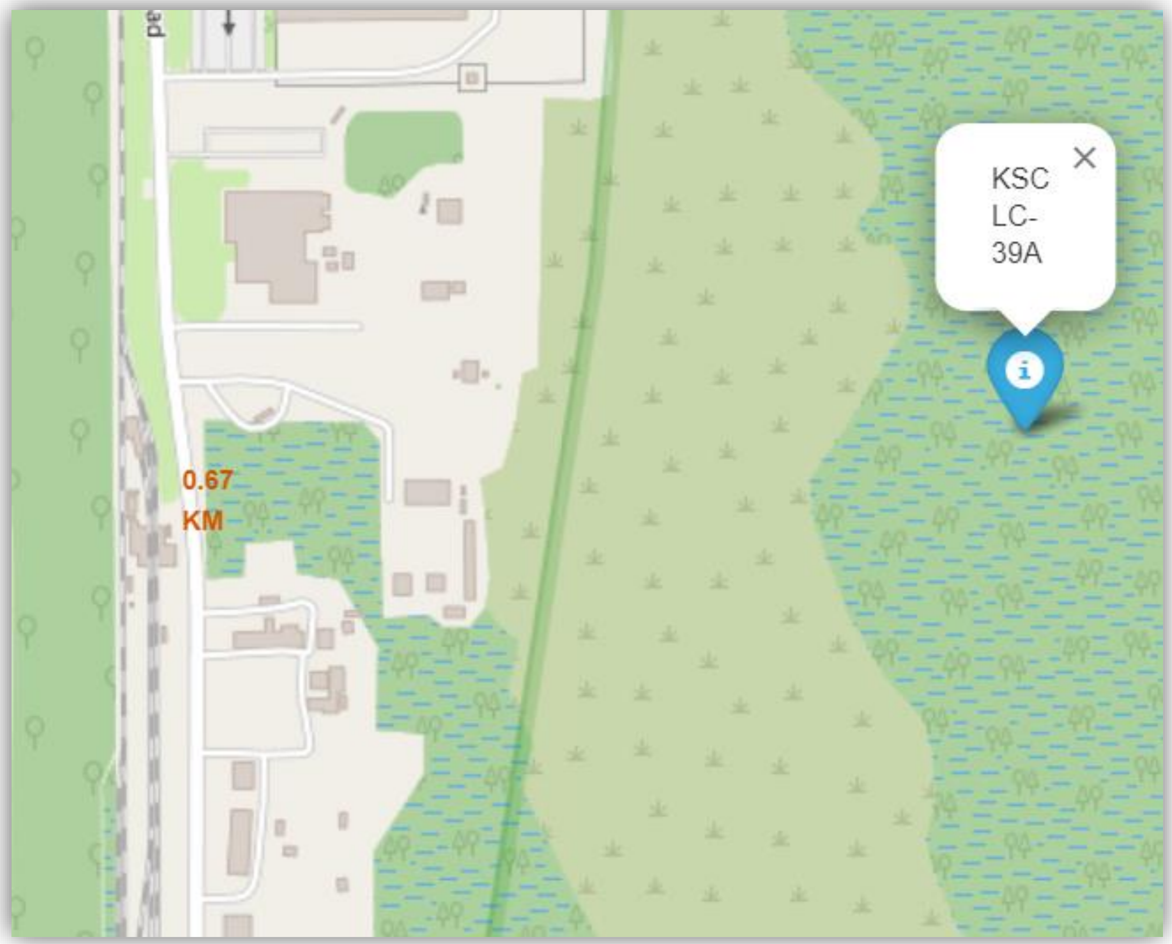


DISTANCE BETWEEN LAUNCH SITES AND NEAREST RAILWAY

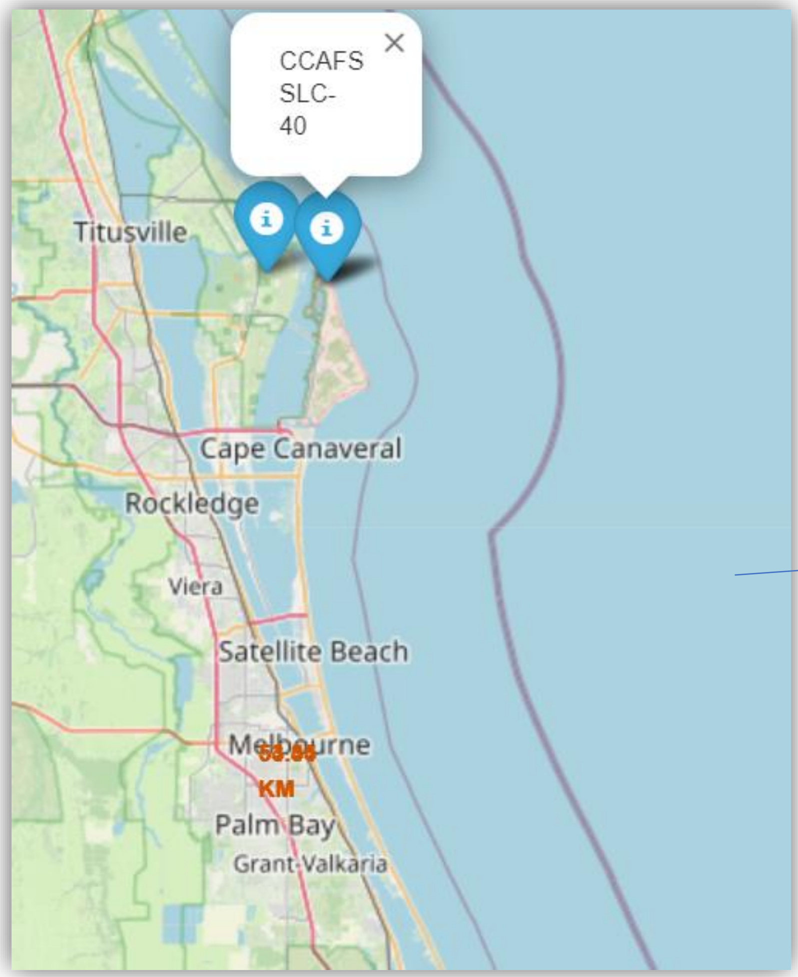


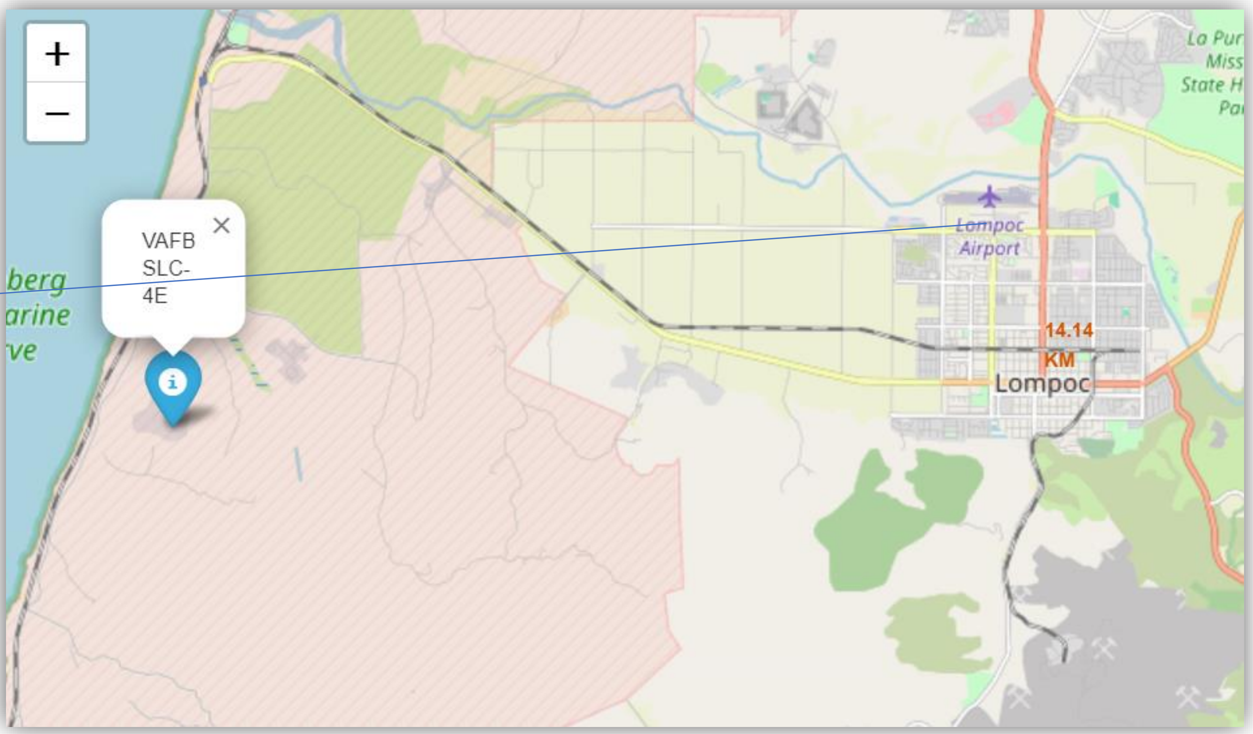
DISTANCE BETWEEN LAUNCH SITES AND NEAREST HIGHWAY



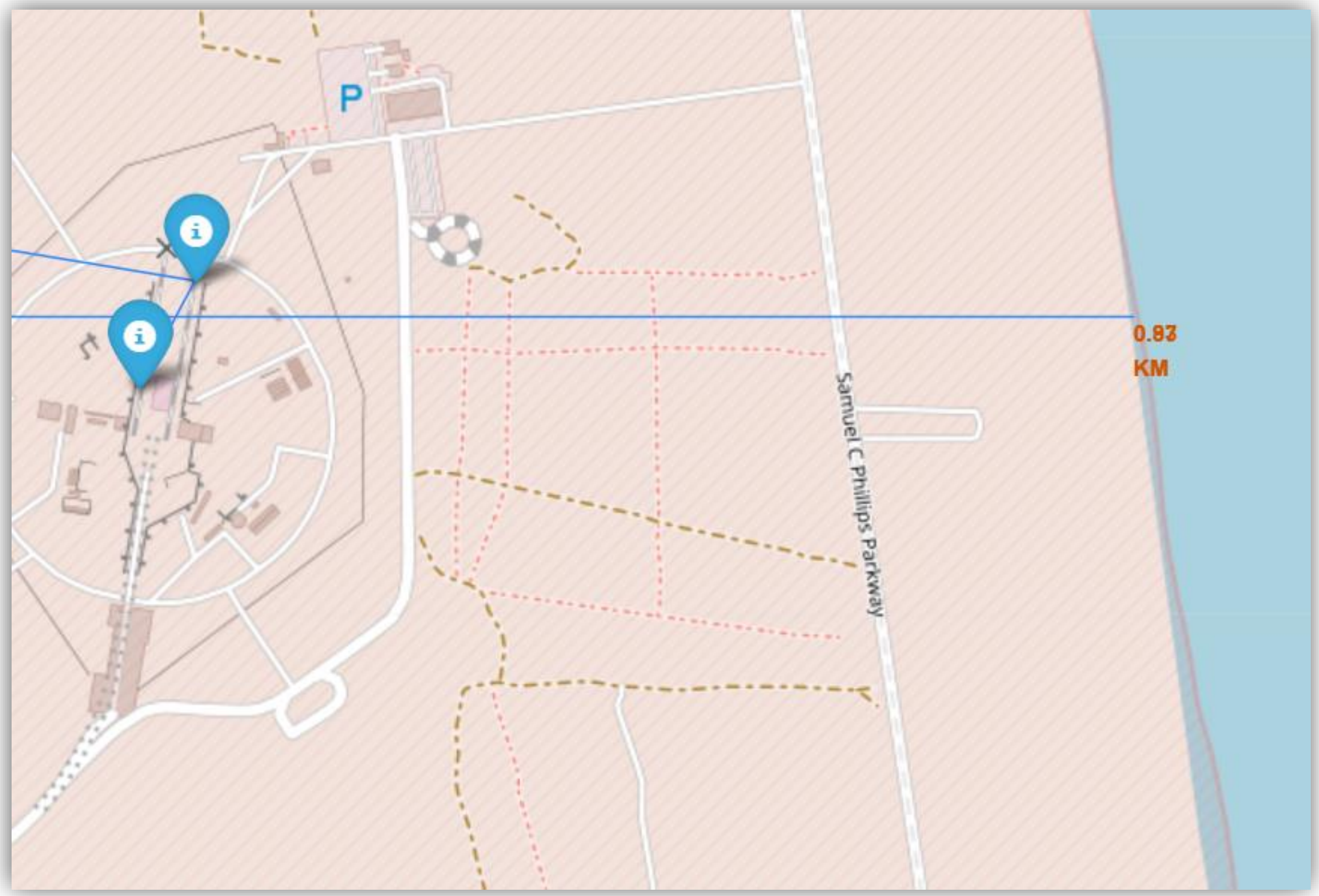


DISTANCE BETWEEN LAUNCH SITES AND NEAREST CITY AREA





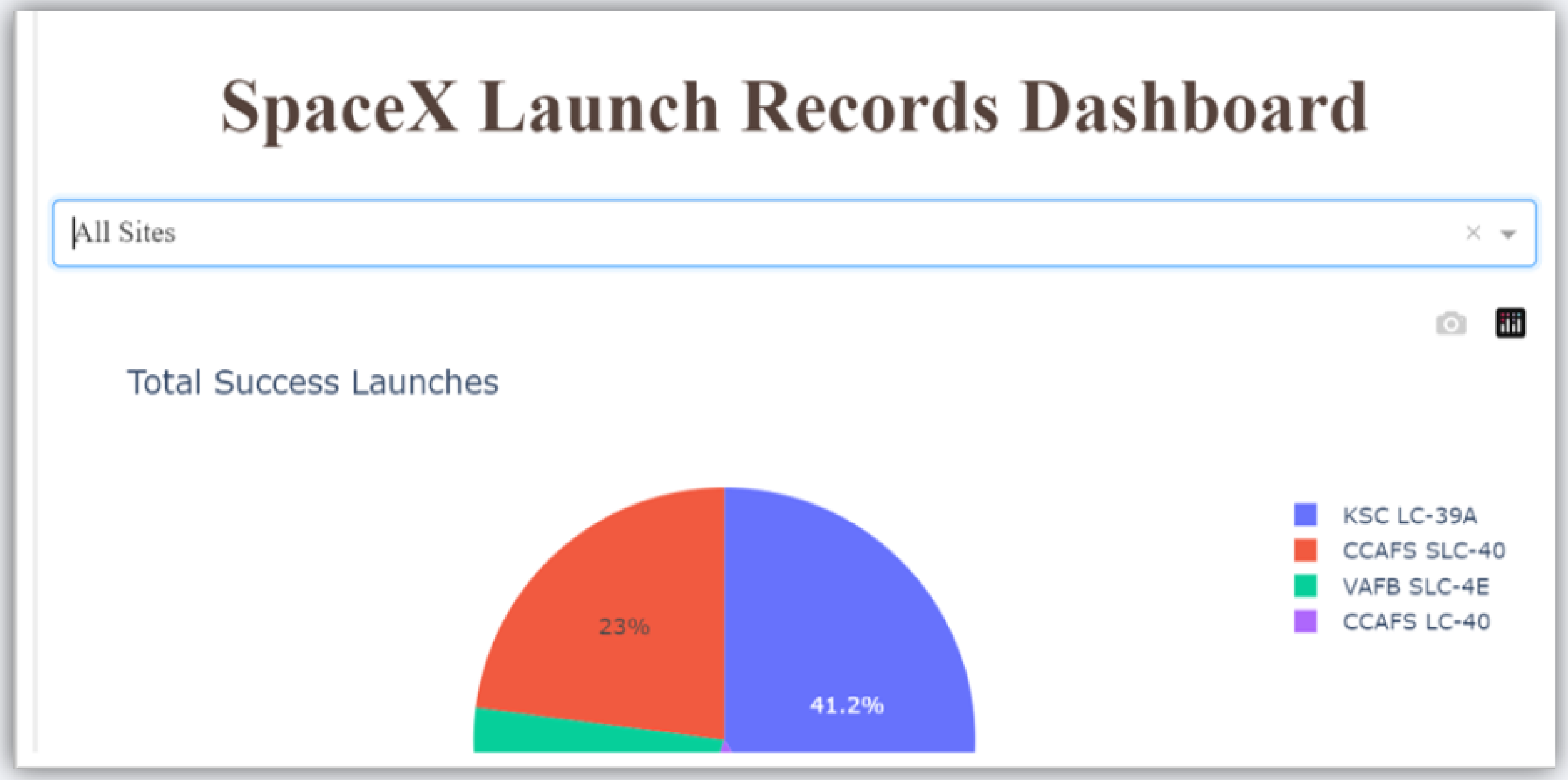
DISTANCE BETWEEN LAUNCH SITES AND NEAREST COAST



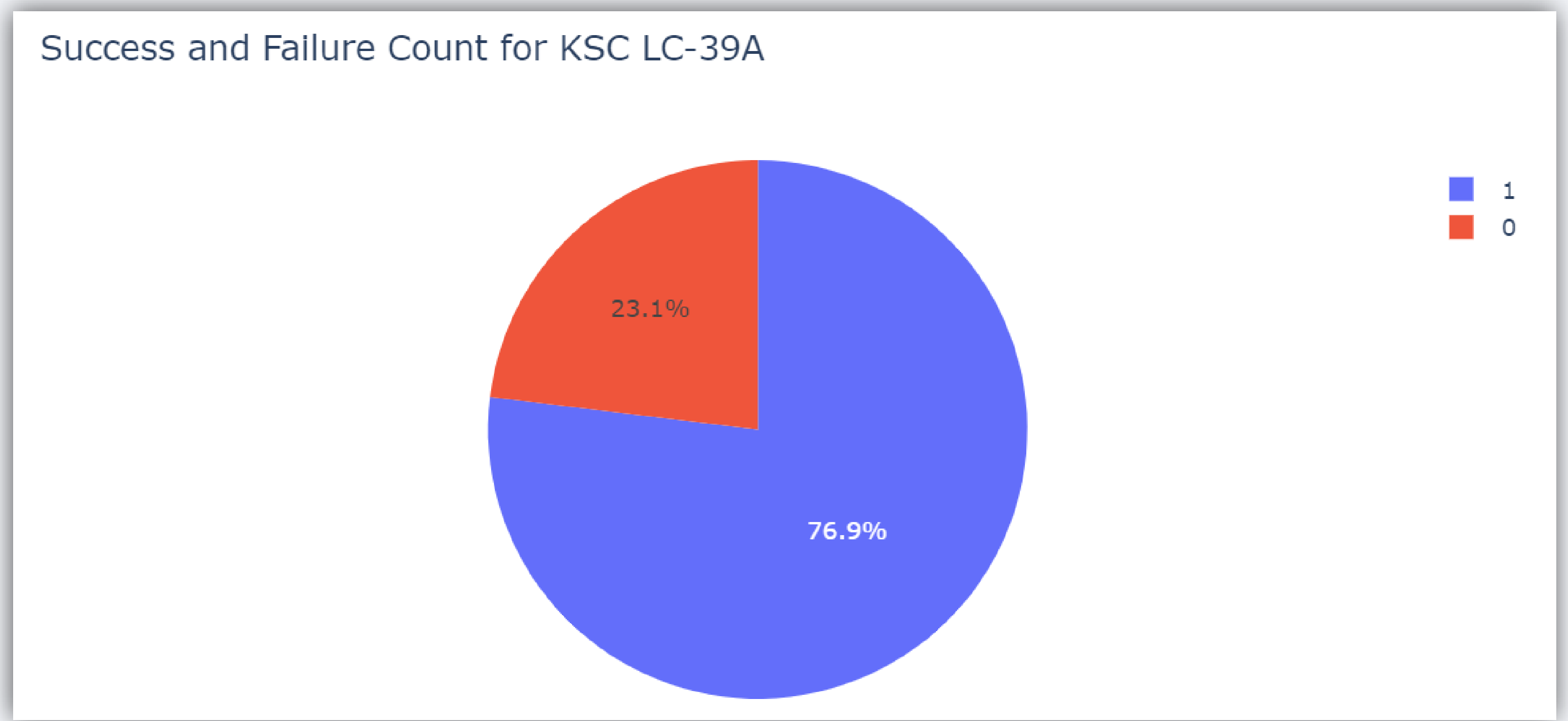
Section 4

Build a Dashboard with Plotly Dash

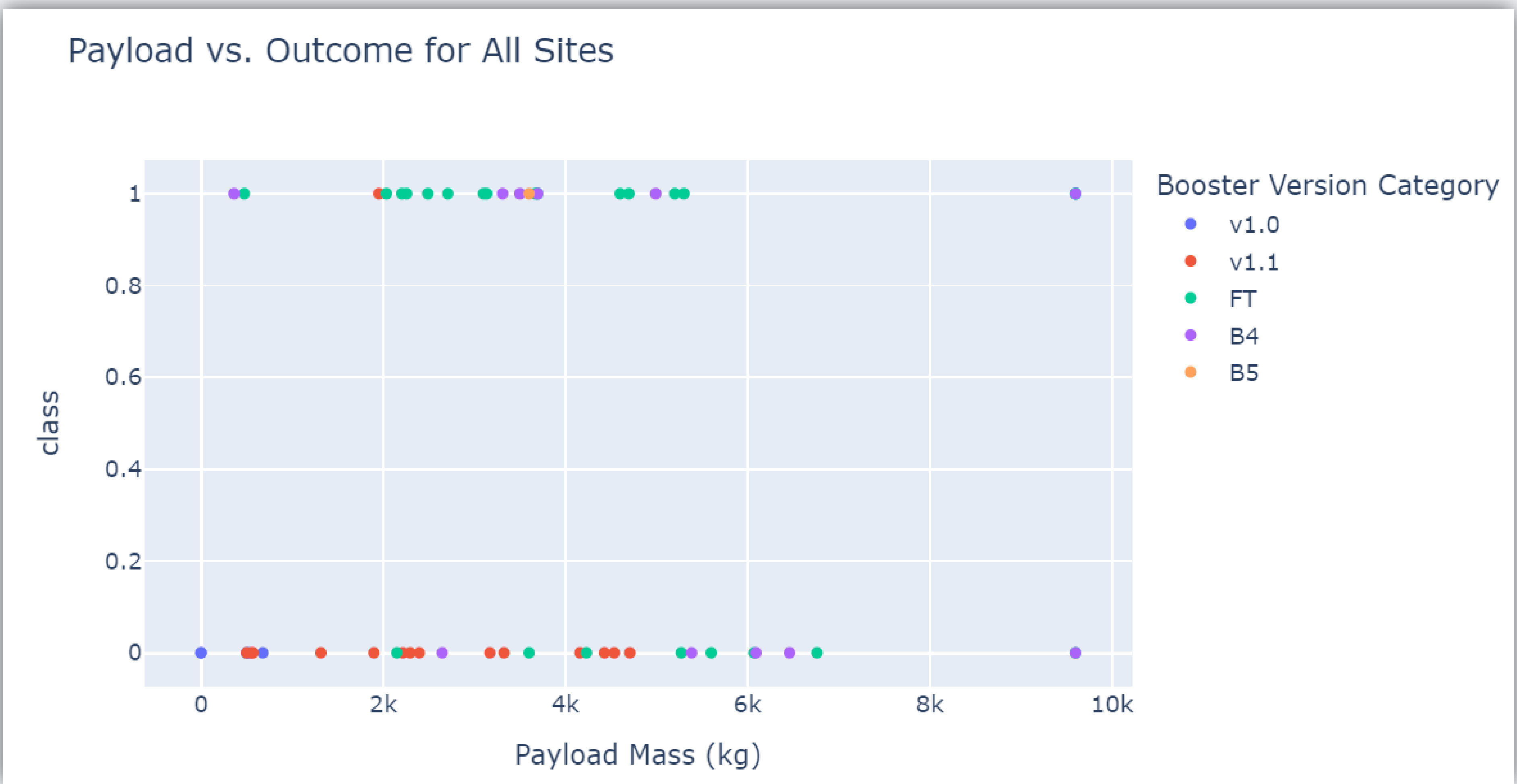
Successful Launches by Site



Launch Site with Highest Success Ratio



Payload vs. Launch Outcomes



Section 5

Predictive Analysis (Classification)

Classification Accuracy Across all models

TASK 12

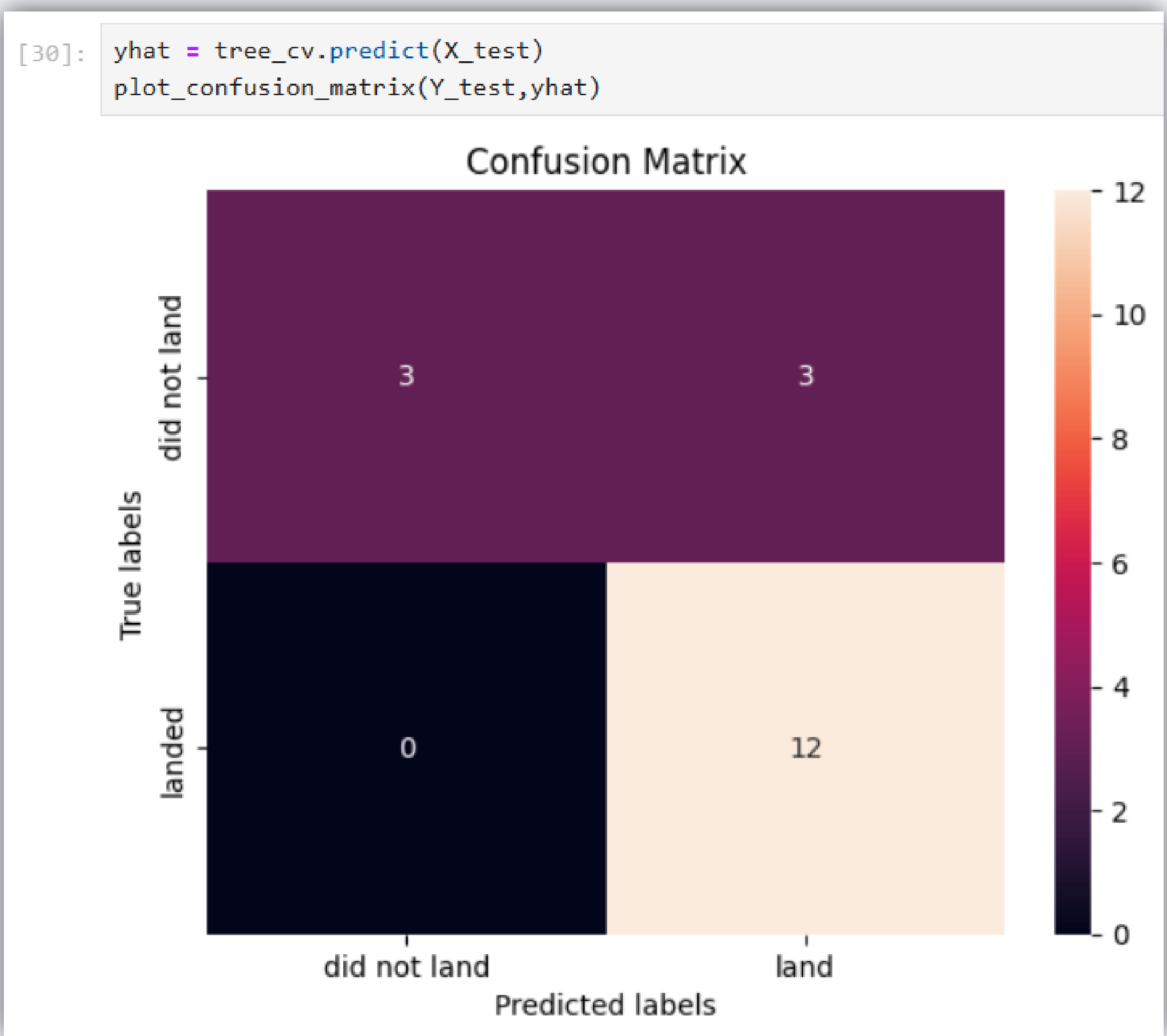
Find the method performs best:

[39]:

```
print(f"Log Regress Accuracy: {logreg_cv.score(X_test,Y_test)}")
print(f"SVM Accuracy: {svm_cv.score(X_test,Y_test)}")
print(f"Tree Accuracy: {tree_cv.score(X_test,Y_test)}")
print(f"KNN Accuracy: {knn_cv.score(X_test,Y_test)}")
```

```
Log Regress Accuracy: 0.8333333333333334
SVM Accuracy: 0.8333333333333334
Tree Accuracy: 0.8333333333333334
KNN Accuracy: 0.8333333333333334
```

Confusion Matrix – Decision Tree Model



Conclusions

- The analysis reveals a **positive trend** in the success rates of space missions, attributed to ongoing learning and the refinement of engineering practices.
- The increase in success rates post-2013 highlights advancements in technology and operational practices, reflecting the dynamic evolution of the aerospace industry
- **Kennedy Space Center Launch Complex 39A** (KSC LC 39A) stands out as a highly effective launch site, while Cape Canaveral Air Force Station Space Launch Complex 40 (CCAFS SLC-40) demonstrates strong potential for accommodating heavier payloads.
- The consistent success rates observed in specific orbits underscore the importance of strategic mission planning.

Thank you!