IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Paul-Édouard Lainé
2024/11/15

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- **Project Background and Context**

The commercial space age is here, with companies making space travel affordable for everyone. Notable players in this field include Virgin Galactic, Rocket Lab, Blue Origin, and SpaceX. SpaceX, in particular, has achieved significant milestones such as sending spacecraft to the International Space Station, launching the Starlink satellite internet constellation, and conducting manned missions to space. One of the key reasons for SpaceX's success is the relatively low cost of their rocket launches, primarily due to the reusability of the first stage of their Falcon 9 rockets. By determining whether the first stage will land successfully, we can estimate the cost of a launch. In this project, we will take on the role of a data scientist working for a new rocket company, Space Y, founded by billionaire industrialist Allon Musk. Our goal is to gather information about SpaceX, create dashboards for our team, and predict if SpaceX will reuse the first stage using machine learning models and public information

- **Problems You Want to Find Answers**

1. **Cost Estimation**: Determine the price of each rocket launch by analyzing various factors that influence the cost, such as payload, orbit, and mission parameters.
2. **First Stage Reusability**: Predict whether the first stage of SpaceX's Falcon 9 rocket will land successfully and be reused. This involves training a machine learning model using public information to make accurate predictions.
3. **Competitive Analysis**: Gather and analyze information about SpaceX's operations and performance to help Space Y compete effectively in the commercial space industry.
4. **Dashboard Creation**: Develop dashboards to visualize and communicate key insights and findings to the team, aiding in decision-making and strategic planning.

Section 1

Methodology

Methodology

Executive Summary

- **Data Collection Methodology**

1. **Describe How Data Was Collected:**
    1. Data was collected from various public sources, including SpaceX's official website, spaceflight databases, and other publicly available datasets. The data includes information on rocket launches, payloads, mission parameters, and outcomes.
2. **Perform Data Wrangling:**
    1. Data wrangling involved cleaning and transforming the collected data to ensure consistency and accuracy. This included handling missing values, correcting data types, and merging datasets from different sources.
3. **Describe How Data Was Processed:**
    1. The processed data was standardized and normalized to ensure that all features were on a similar scale. This step is crucial for machine learning models to perform effectively. Additionally, feature engineering was performed to create new features that could improve model performance.
4. **Perform Exploratory Data Analysis (EDA) Using Visualization and SQL:**
    1. EDA was conducted to understand the underlying patterns and relationships in the data. Visualization tools such as Matplotlib and Seaborn were used to create plots and charts. SQL queries were used to extract and analyze specific subsets of data.
5. **Perform Interactive Visual Analytics Using Folium and Plotly Dash:**
    1. Interactive visual analytics were performed using Folium and Plotly Dash. Folium was used to create interactive maps to visualize the geographical distribution of rocket launches. Plotly Dash was used to create interactive dashboards to visualize key metrics and trends.
6. **Perform Predictive Analysis Using Classification Models:**
    1. Predictive analysis was conducted using various classification models, including Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K Nearest Neighbors (KNN). These models were trained and evaluated to predict whether the first stage of the rocket would land successfully.
7. **How to Build, Tune, Evaluate Classification Models:**
    1. **Build:** Classification models were built using scikit-learn, a popular machine learning library in Python.
    2. **Tune:** Hyperparameter tuning was performed using GridSearchCV to find the best parameters for each model.
    3. **Evaluate:** The models were evaluated using metrics such as accuracy, precision, recall, and the confusion matrix. Cross-validation was used to ensure the models' robustness and generalizability.

Data Collection

- **Data Collection Methodology**

1. **Public Sources**: Data was collected from various public sources, including SpaceX's official website, spaceflight databases, and other publicly available datasets. These sources provided comprehensive information on rocket launches, payloads, mission parameters, and outcomes.
2. **APIs and Web Scraping**: APIs and web scraping techniques were used to gather data from online sources. For example, APIs provided by spaceflight tracking websites were utilized to fetch real-time data on rocket launches and their outcomes. Web scraping was employed to extract data from websites that did not offer APIs.
3. **Historical Data**: Historical data on rocket launches and their outcomes were obtained from spaceflight archives and databases. This data included detailed records of past missions, including launch dates, payloads, mission objectives, and whether the first stage was successfully recovered.
4. **Data Integration**: The collected data from various sources were integrated into a unified dataset. This involved merging data from different sources, ensuring consistency in data formats, and resolving any discrepancies.
5. **Data Validation**: The collected data was validated to ensure accuracy and reliability. This involved cross-referencing data from multiple sources, checking for inconsistencies, and correcting any errors.

Data Collection – SpaceX API

- SPACE X API

Request and parse the SpaceX launch data using the GET request

- Display the first few rows of the DataFrame
- Filter the dataframe to only include Falcon 9 launches
- Data Wrangling
- Dealing with Missing Values

Data Collection - Scraping

- WEB SCRAPING

Request the Falcon9 Launch Wiki page from its URL

Extract all column/variable names from the HTML table header

Create a data frame by parsing the launch HTML tables

Data Wrangling

- Data wrangling
- Data cleaning by removing duplicates, handling missing values and correct data types
- Data transformation by standardizing and normalizing the data to ensure consistency
- Feature engineering by creating new features to improve model performance.
- Data splitting into training and test sets fr model evaluation

TASK 1: Calculate the number of launches on each site

TASK 2: Calculate the number and occurrence of each orbit

TASK 3: Calculate the number and occurence of mission outcome of the orbits

TASK 4: Create a landing outcome label from Outcome column

EDA with Data Visualization

- EDA VISUALIZATION

Scatter graph

- The scatter plot was chosen to present the relation between the payload mass and the flight number. We can see the progression of the flights and the success or failure depending the mass. In a second graph, we can see what location has the most success as the number of flights goes up

Bar chart

- The bar chart was chosen to display what orbit type was the most successful

Linear graph

- The linear graph was to display the average success rate over the years

EDA with SQL

EDA_SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first succesful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- 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.
- 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.

Build an Interactive Map with Folium

- On the interactive maps, I added markers, lines and numbers for mutltiples reasons
- Markers: To see where are the launch sites and NASA
- Lines: To see how close are the launch sites from the coastlines and cities
- Numbers: Display the number of launches from launch sites that are in the same vicinity
- Map Link

Build a Dashboard with Plotly Dash

- The plots and charts used are the pie chart and scatter plot
- The pie chart: to display the number of successful and failed launches from either all sites or each sites separately
- The scatter plot: Dispalying what version of the Booster succeed or failed depending of the payload range with the aid of a slider that changes the slider ranging from 0 to 10000kg
- Dashboard
- Dash Git Hub

Predictive Analysis (Classification)

- We built and evaluated four classification models: Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K Nearest Neighbors (KNN). Each model was tuned using GridSearchCV to find the best hyperparameters. All models achieved the same test accuracy of 0.8333. Further analysis using additional metrics such as precision, recall, and the confusion matrix could provide more insights into the performance of each model. Overall, the models performed well, and any of them could be a suitable choice for this classification task.
- [Predictive Analysis]()

Create a NumPy array from the column Class in data,

Standardize the data in X then reassign it to the variable X

Use the function train_test_split to split the data X and Y into training and test data. Set the parameter test_size to 0.2 and random_state to 2.

Create a logistic regression object then create a GridSearchCV object with cv = 10.

Calculate the accuracy on the test data using the method

Create a support vector machine object then create a object with cv = 10

Calculate the accuracy on the test data using the method

Create a decision tree classifier object then create a object with cv = 10.

Calculate the accuracy of tree_cv on the test data using the method

Create a k nearest neighbors object then create a object with cv = 10

Calculate the accuracy of knn_cv on the test data using the method

Find the method performs best:

Results

- **Exploratory Data Analysis Results**

1. **Unique Launch Sites:** Identified the unique launch sites in the space mission dataset, which include 'CCAFS LC-40', 'VAFB SLC-4E', 'KSC LC-39A', and 'CCAFS SLC-40'.

2. **Launch Sites Starting with 'CCA':** Displayed 5 records where launch sites begin with the string 'CCA', showing details, such as launch date, booster version, payload, and mission outcome.

3. **Total Payload Mass by NASA (CRS):** Calculated the total payload mass carried by boosters launched by NASA (CRS), which amounted to 45,596 kg.

4. **Average Payload Mass by Booster Version F9 v1.1:** Calculated the average payload mass carried by booster version F9 v1.1, which was 2,928.4 kg.

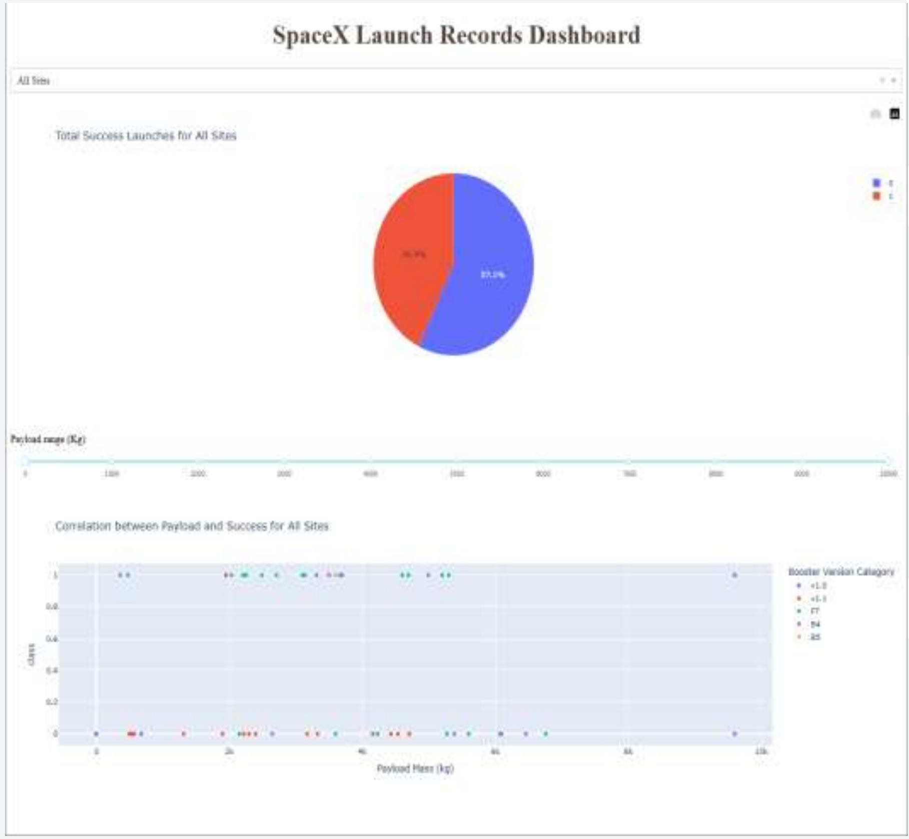
5. **First Successful Ground Pad Landing:** Identified the date of the first successful landing outcome on a ground pad, which was achieved on 2015-12-22.

6. **Boosters with Successful Drone Ship Landings and Specific Payload Mass:** Listed the names of boosters that had successful landings on a drone ship and carried payloads between 4,000 and 6,000 kg. The boosters included F9 FT B1022, F9 FT B1026, F9 FT B1021.2, and F9 FT B1031.2.

7. **Mission Outcomes:** Counted the total number of successful and failure mission outcomes, with 98 successes, 1 failure (in flight), 1 success (payload status unclear), and 1 success.

8. **Boosters with Maximum Payload Mass:** Listed the booster versions that carried the maximum payload mass, including F9 B5 B1048.4, F9 B5 B1049.4, F9 B5 B1051.3, and others.

9. **Failure Landing Outcomes in 2015:** Displayed records of failure landing outcomes on drone ships in 2015, including details such as month, booster version, and launch site.



SpaceX Launch Records Dashboard

All Sites

Logistic Regression

- **Best Parameters**: {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
- **Validation Accuracy**: 0.8464
- **Test Accuracy**: 0.8333

Support Vector Machine (SVM)

- **Best Parameters**: {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
- **Validation Accuracy**: 0.8482
- **Test Accuracy**: 0.8333

Decision Tree

- **Best Parameters**: {'criterion': 'gini', 'max_depth': 2, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'random'}
- **Validation Accuracy**: 0.8768
- **Test Accuracy**: 0.8333

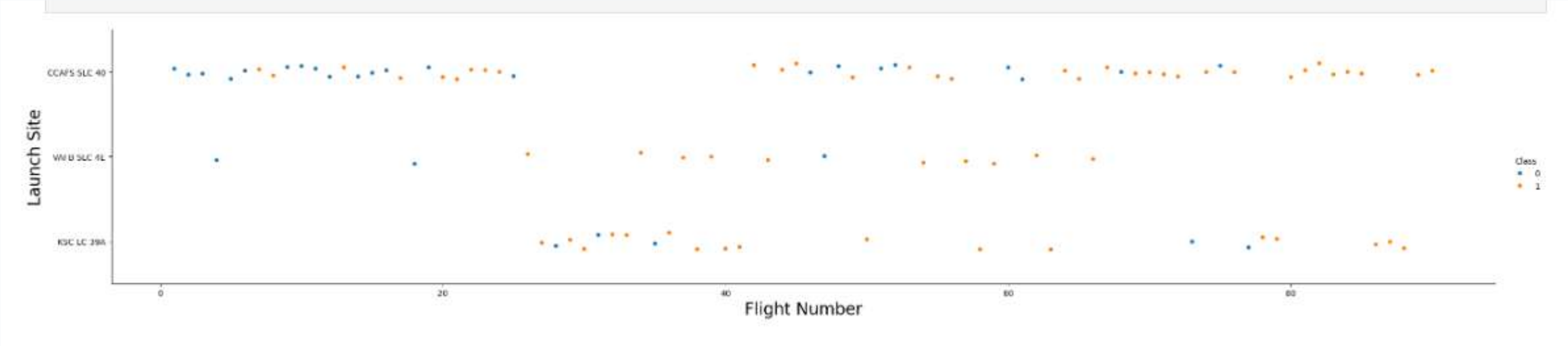
K Nearest Neighbors(KNN)

- **Best Parameters**: {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
- **Validation Accuracy**: 0.8482
- **Test Accuracy**: 0.8333

Section 2

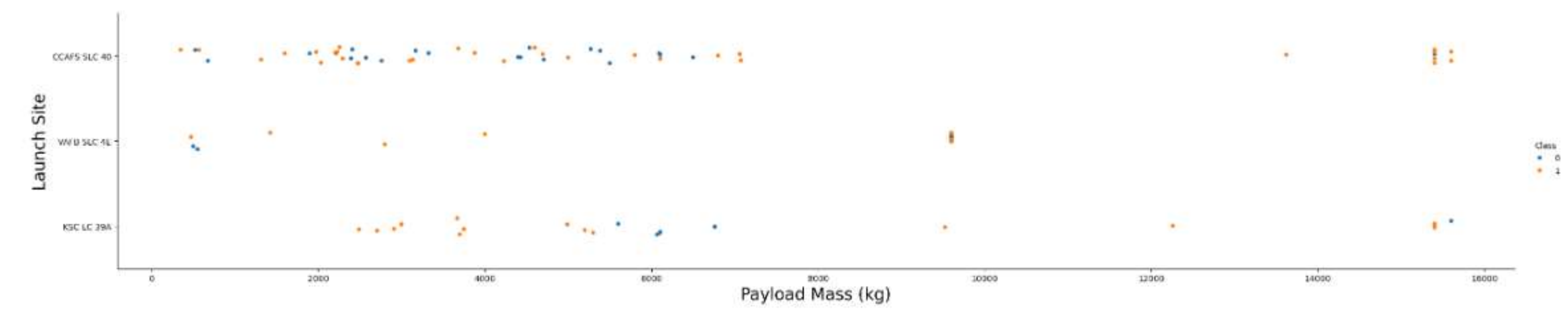
Insights drawn from EDA

Flight Number vs. Launch Site



The CCAF SLC 40 had the most launches therefore the most failed launches. We can see a stronger concentration of failures in the earlier flights

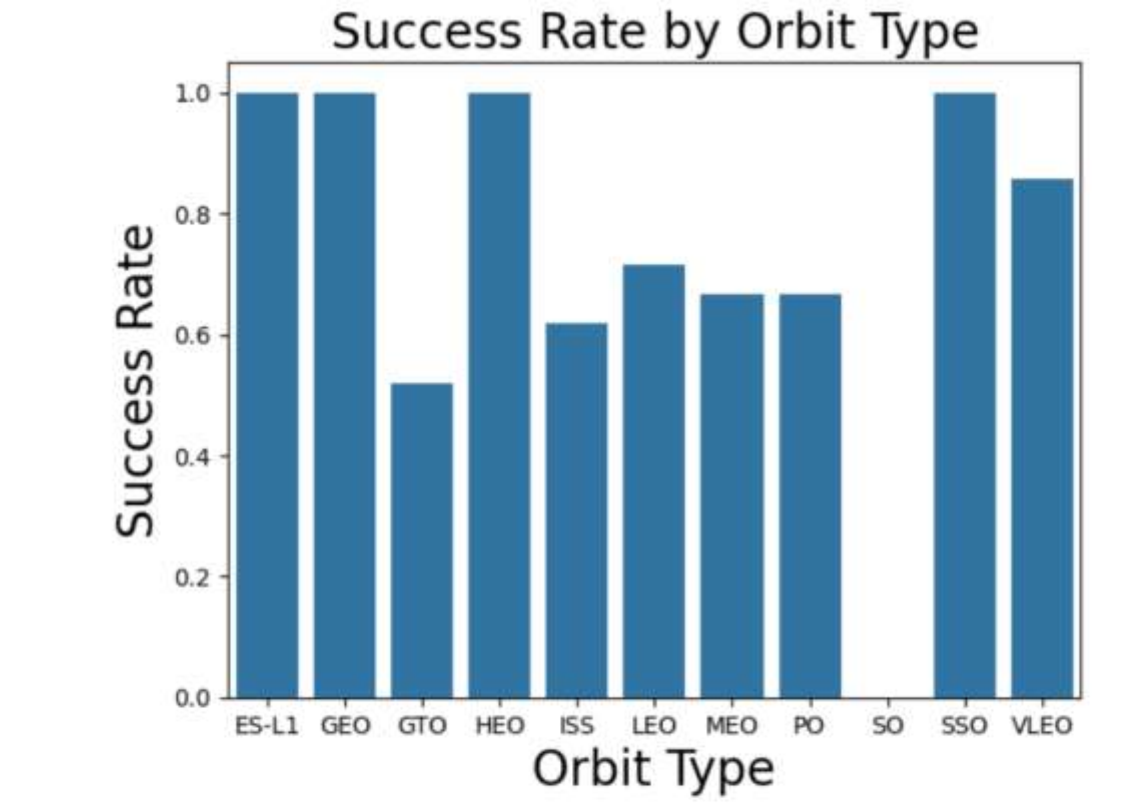
Payload vs. Launch Site



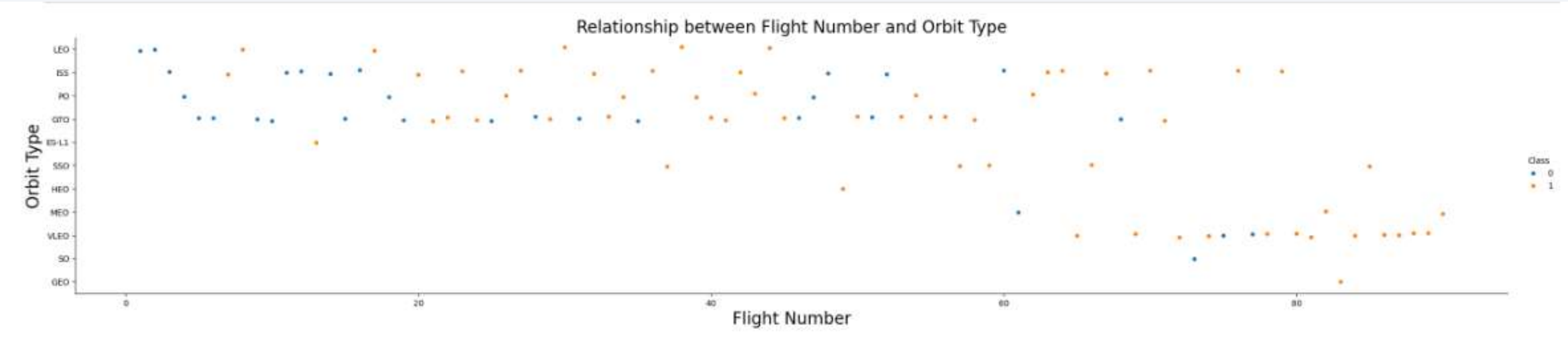
- It seems that when the payload is below 8000kg, it has higher chance to fail but it’s most likely tied to the fact that most flight were under that payload

Success Rate vs. Orbit Type

- Outside of GTO, most of them have over 60% success rate in launches

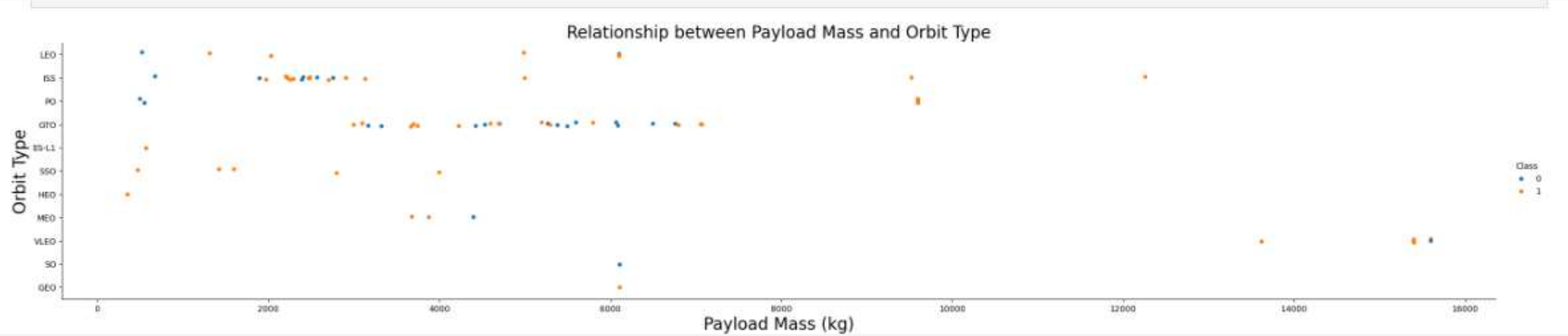


Flight Number vs. Orbit Type



- The orbit types LEO, ISS, PO, GTO have the most flights. It explains their higher failures compared to the other types of orbit.

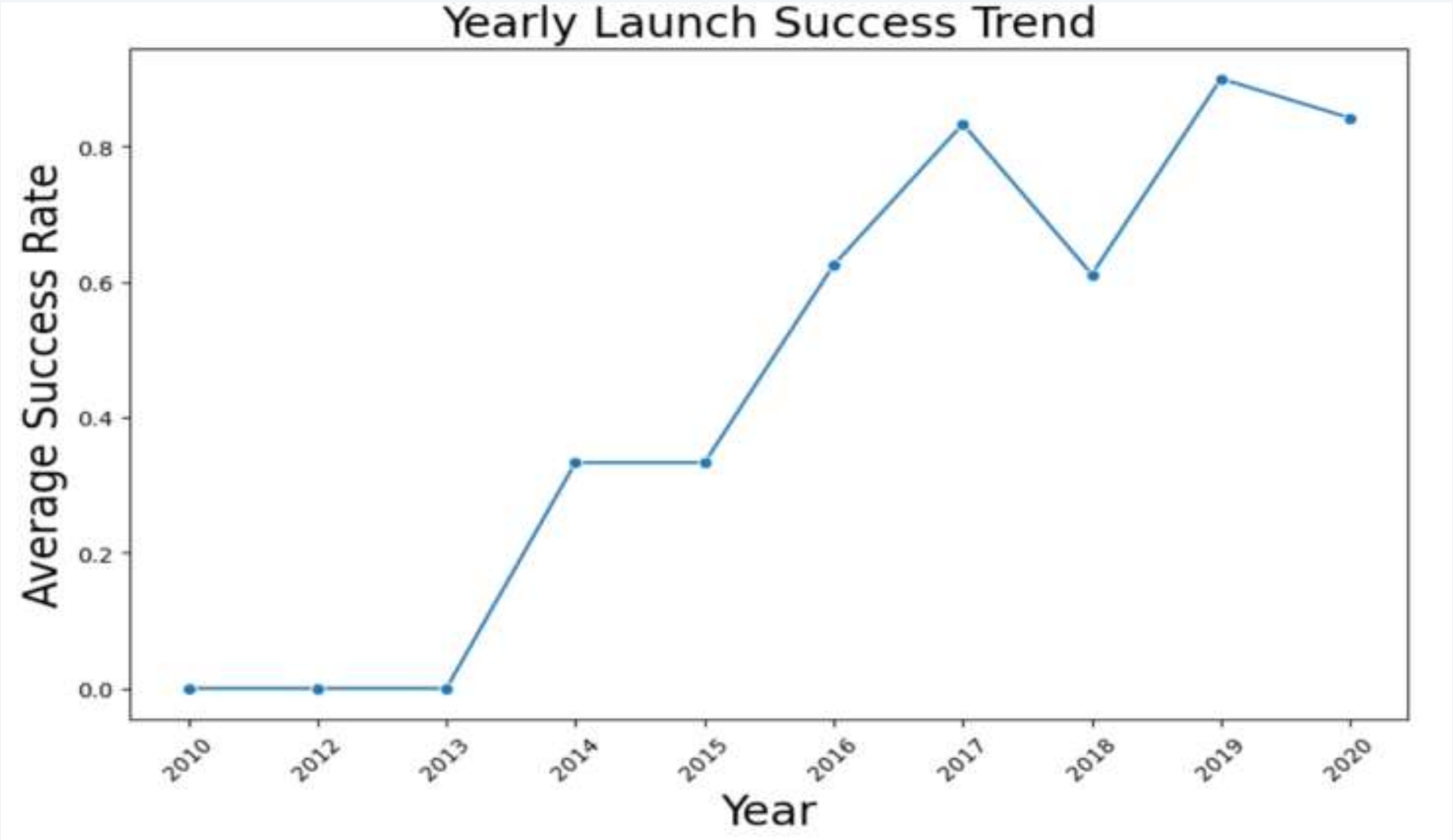
Payload vs. Orbit Type



- Most flights were under 8000KG payload. Between 0 and 2000kg, the orbit LEO, ISS and PO have mostly failed. The flights above 8000kg are very rare with much less failure.

Launch Success Yearly Trend

- The success rate goes up as the years goes by



All Launch Site Names

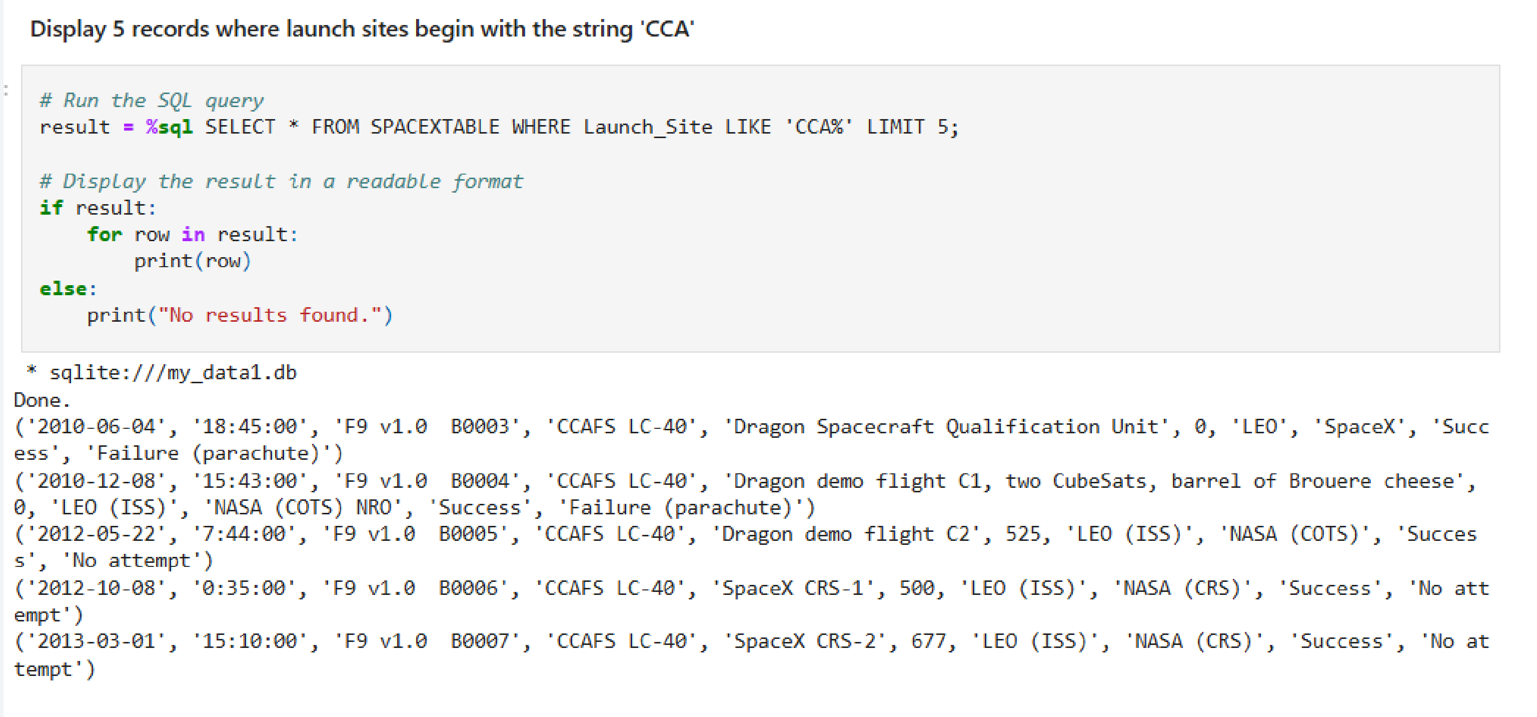
(

- 'CCAFS LC-40',) ('VAFB SLC-4E',) ('KSC LC-39A',) ('CCAFS SLC-40’,)
- This is the code I used to get the launch sites:

```
# Run the SQL query

result = %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;

# Display the result in a readable format

if result:

    for row in result:

        print(row)

else:

    print("No results found.")
```

The reason i used this code was to get unique launch sites in the list and not duplicates. I put a limit at 5.

Launch Site Names Begin with 'CCA'

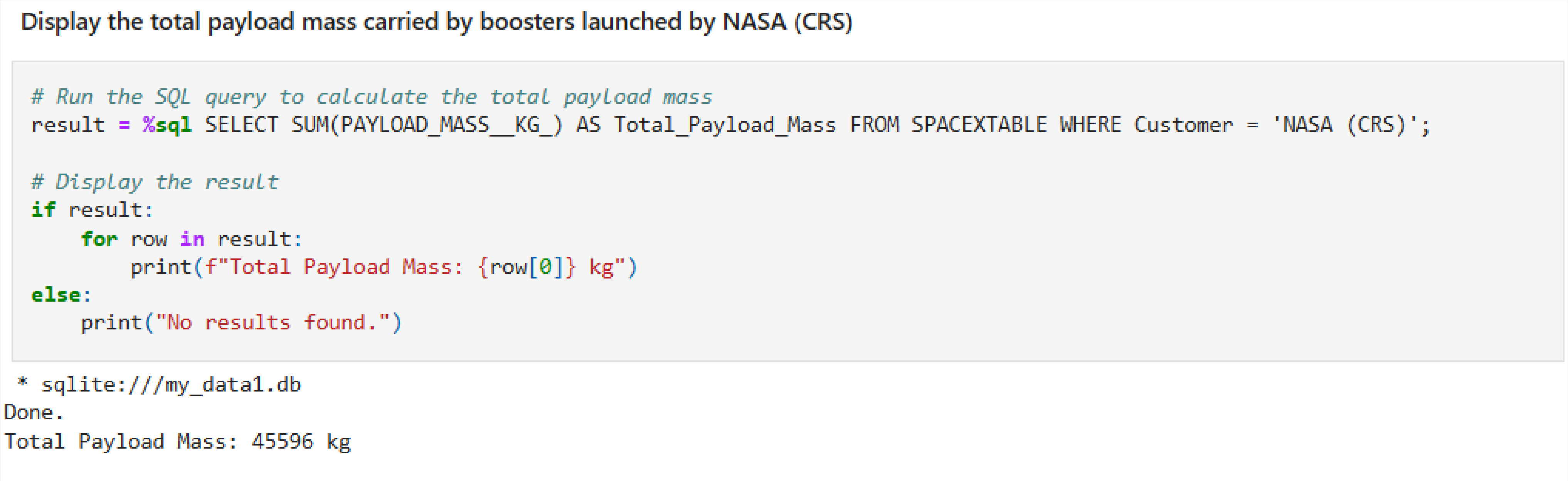


Display 5 records where launch sites begin with the string 'CCA'

```
# Run the SQL query
result = %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;

# Display the result in a readable format
if result:
    for row in result:
        print(row)
else:
    print("No results found.")
```

```
 * sqlite:///my_data1.db
Done.
('2010-06-04', '18:45:00', 'F9 v1.0  B0003', 'CCAFS LC-40', 'Dragon Spacecraft Qualification Unit', 0, 'LEO', 'SpaceX', 'Succ
ess', '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)', 'Succes
s', '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 att
empt')
('2013-03-01', '15:10:00', 'F9 v1.0  B0007', 'CCAFS LC-40', 'SpaceX CRS-2', 677, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'No at
tempt')
```

Total Payload Mass

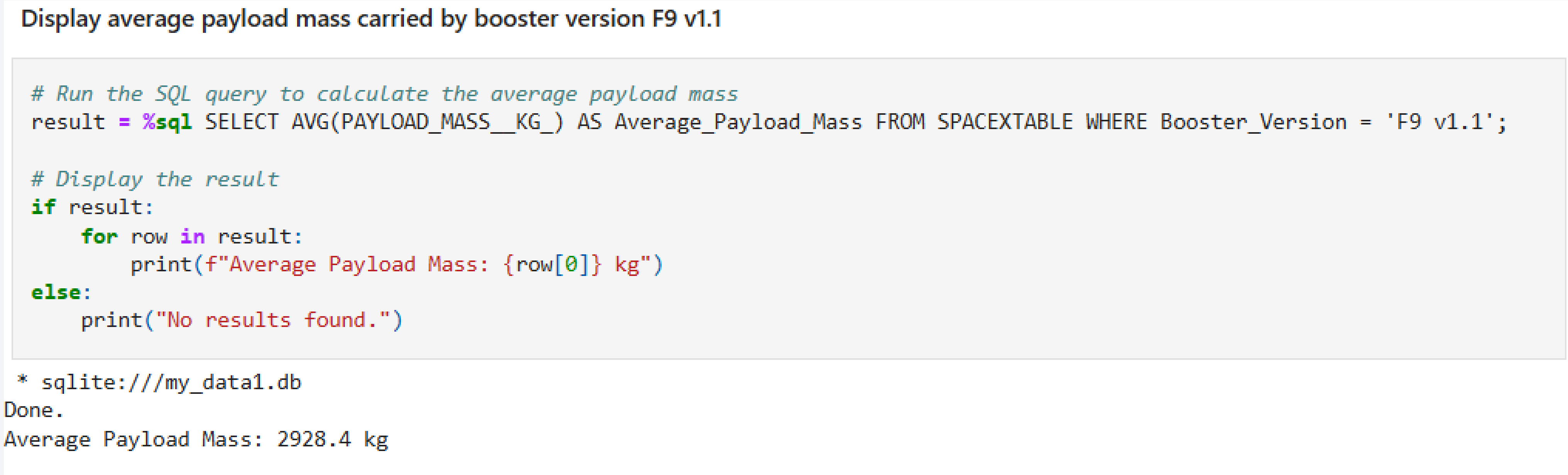


Display the total payload mass carried by boosters launched by NASA (CRS)

```
# Run the SQL query to calculate the total payload mass
result = %sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';

# Display the result
if result:
    for row in result:
        print(f"Total Payload Mass: {row[0]} kg")
else:
    print("No results found.")
```

```
 * sqlite:///my_data1.db
Done.
Total Payload Mass: 45596 kg
```

Average Payload Mass by F9 v1.1

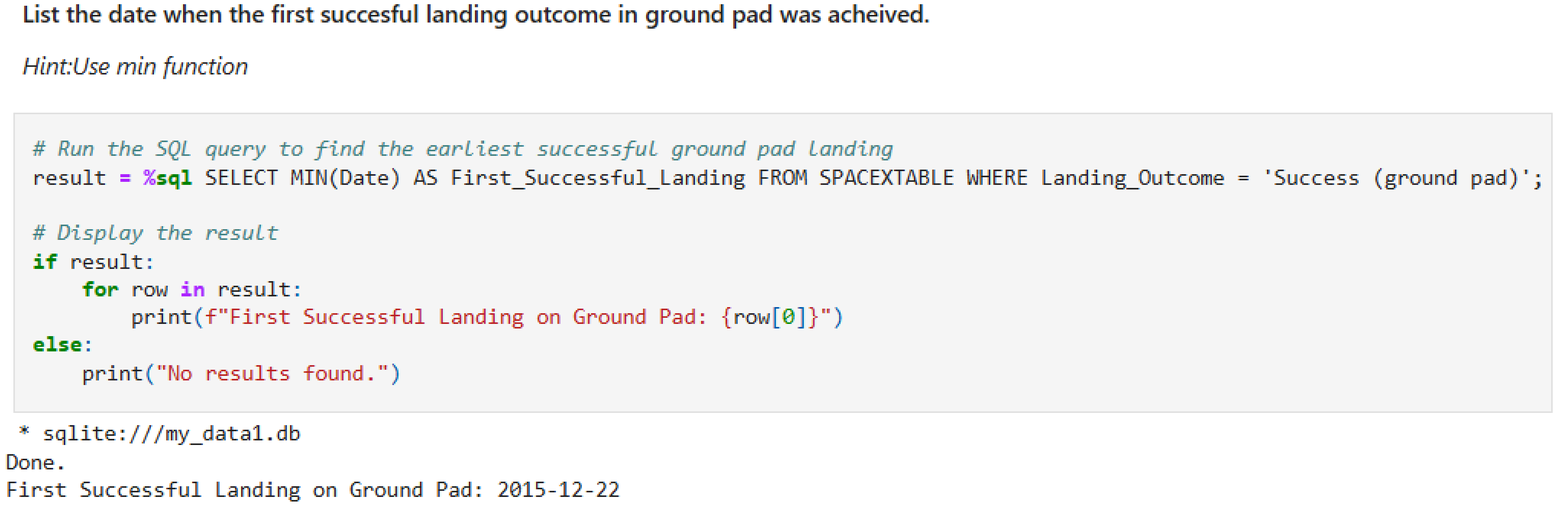


Display average payload mass carried by booster version F9 v1.1

```
# Run the SQL query to calculate the average payload mass
result = %sql SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload_Mass FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';

# Display the result
if result:
    for row in result:
        print(f"Average Payload Mass: {row[0]} kg")
else:
    print("No results found.")
```

```
 * sqlite:///my_data1.db
Done.
Average Payload Mass: 2928.4 kg
```

First Successful Ground Landing Date



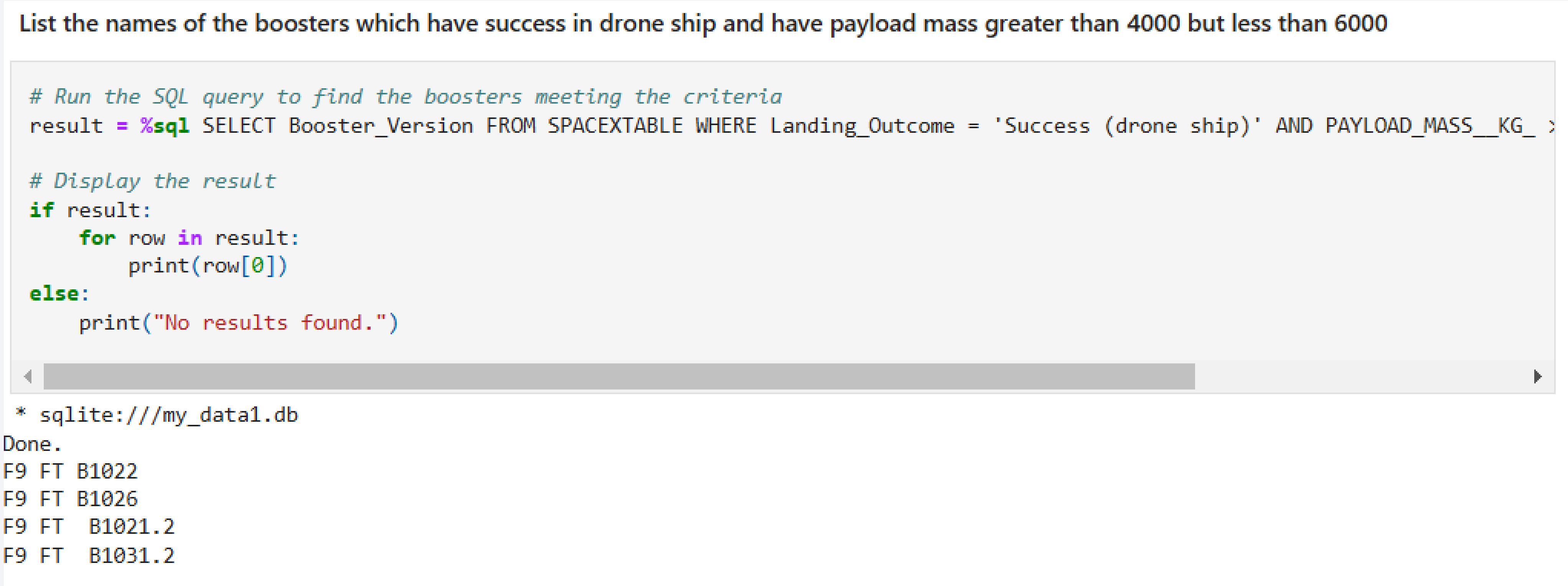
List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
# Run the SQL query to find the earliest successful ground pad landing
result = %sql SELECT MIN(Date) AS First_Successful_Landing FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';

# Display the result
if result:
    for row in result:
        print(f"First Successful Landing on Ground Pad: {row[0]}")
else:
    print("No results found.")
```

```
 * sqlite:///my_data1.db
Done.
First Successful Landing on Ground Pad: 2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

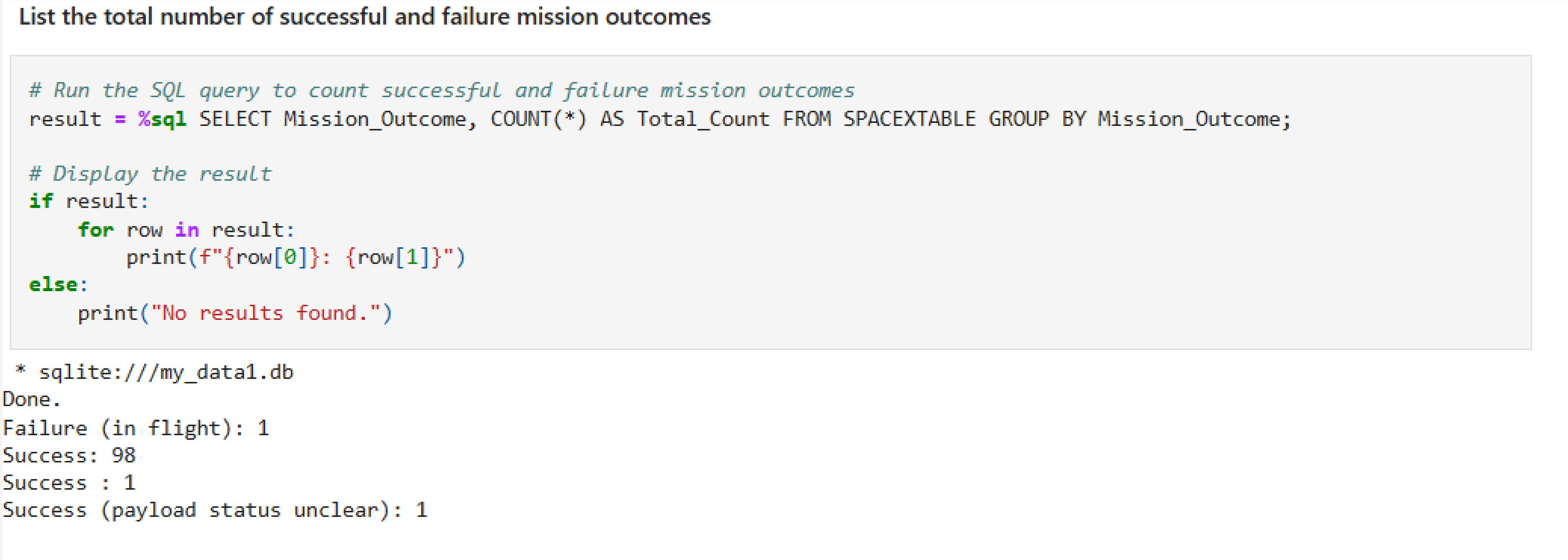


List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
# Run the SQL query to find the boosters meeting the criteria
result = %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ >

# Display the result
if result:
    for row in result:
        print(row[0])
else:
    print("No results found.")
```

```
 * sqlite:///my_data1.db
Done.
F9 FT B1022
F9 FT B1026
F9 FT  B1021.2
F9 FT  B1031.2
```

Total Number of Successful and Failure Mission Outcomes

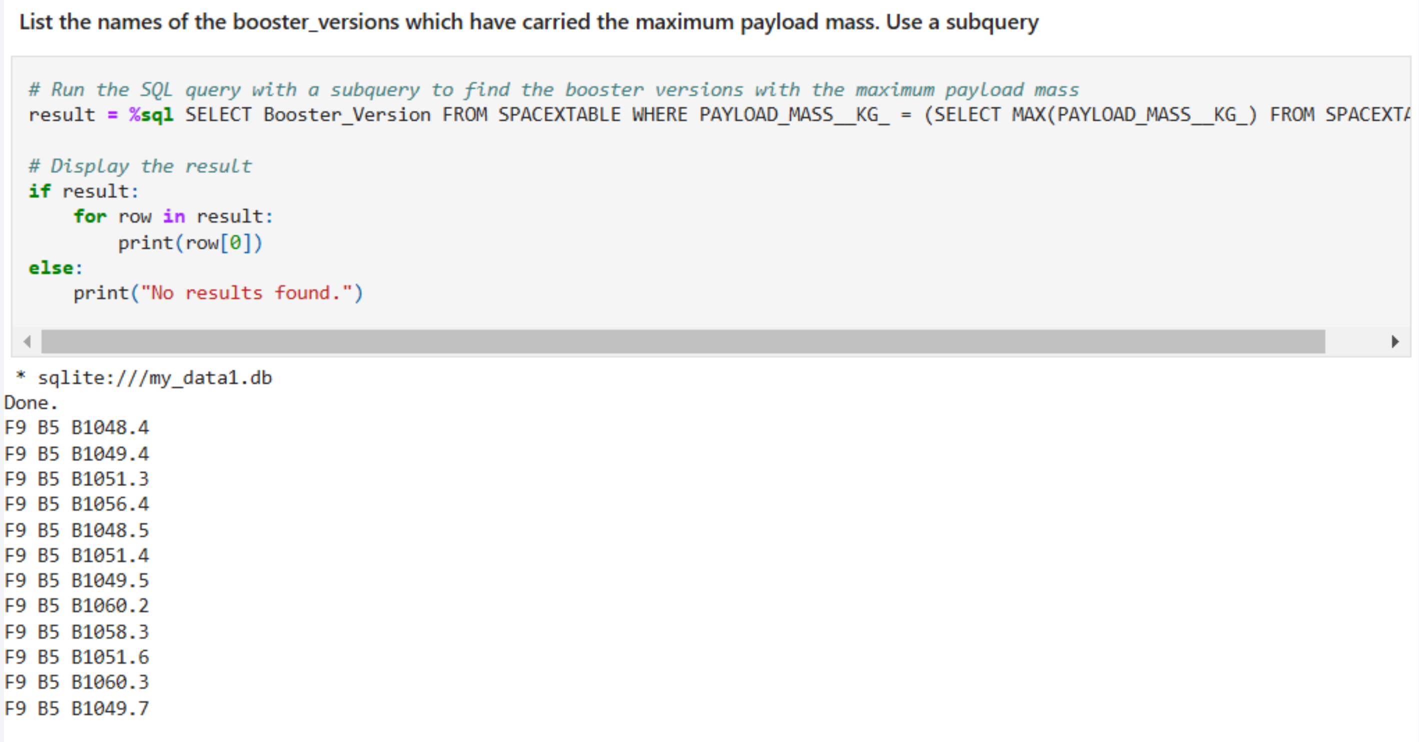


List the total number of successful and failure mission outcomes

```
# Run the SQL query to count successful and failure mission outcomes
result = %sql SELECT Mission_Outcome, COUNT(*) AS Total_Count FROM SPACEXTABLE GROUP BY Mission_Outcome;

# Display the result
if result:
    for row in result:
        print(f"{row[0]}: {row[1]}")
else:
    print("No results found.")
```

```
 * sqlite:///my_data1.db
Done.
Failure (in flight): 1
Success: 98
Success : 1
Success (payload status unclear): 1
```

Boosters Carried Maximum Payload

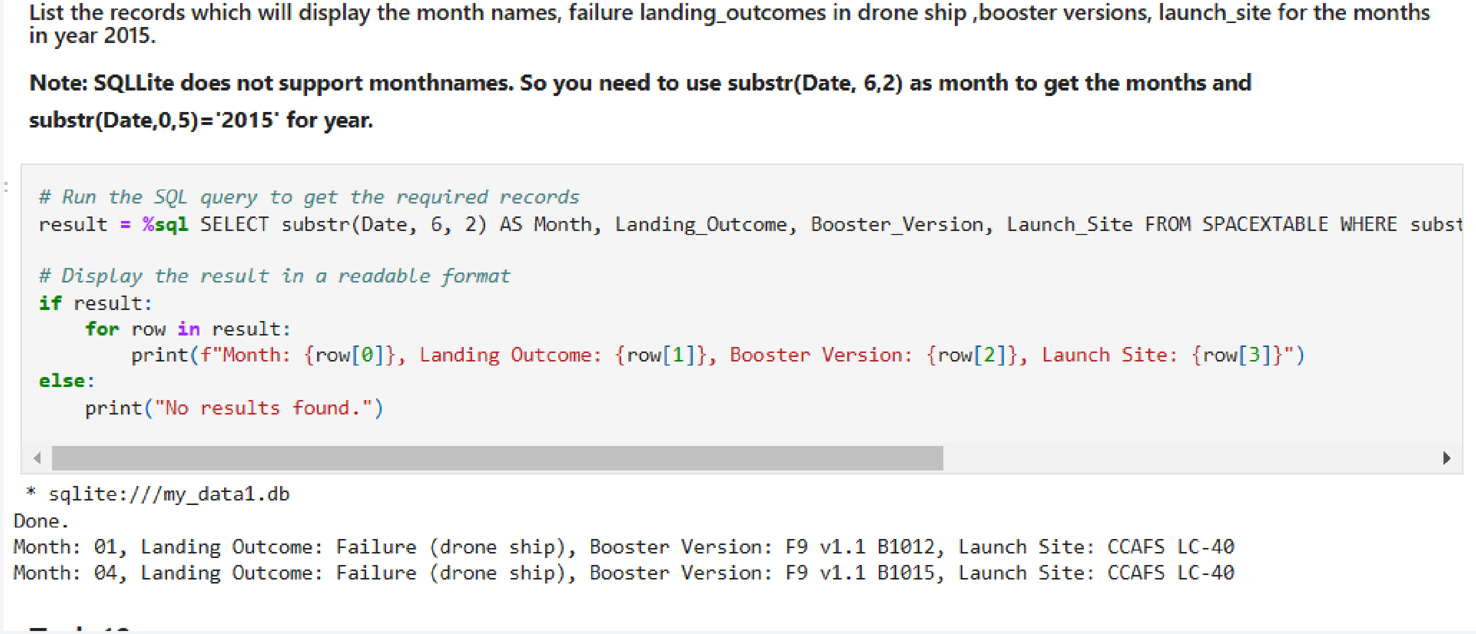


List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
# Run the SQL query with a subquery to find the booster versions with the maximum payload mass
result = %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTA

# Display the result
if result:
    for row in result:
        print(row[0])
else:
    print("No results found.")
```

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



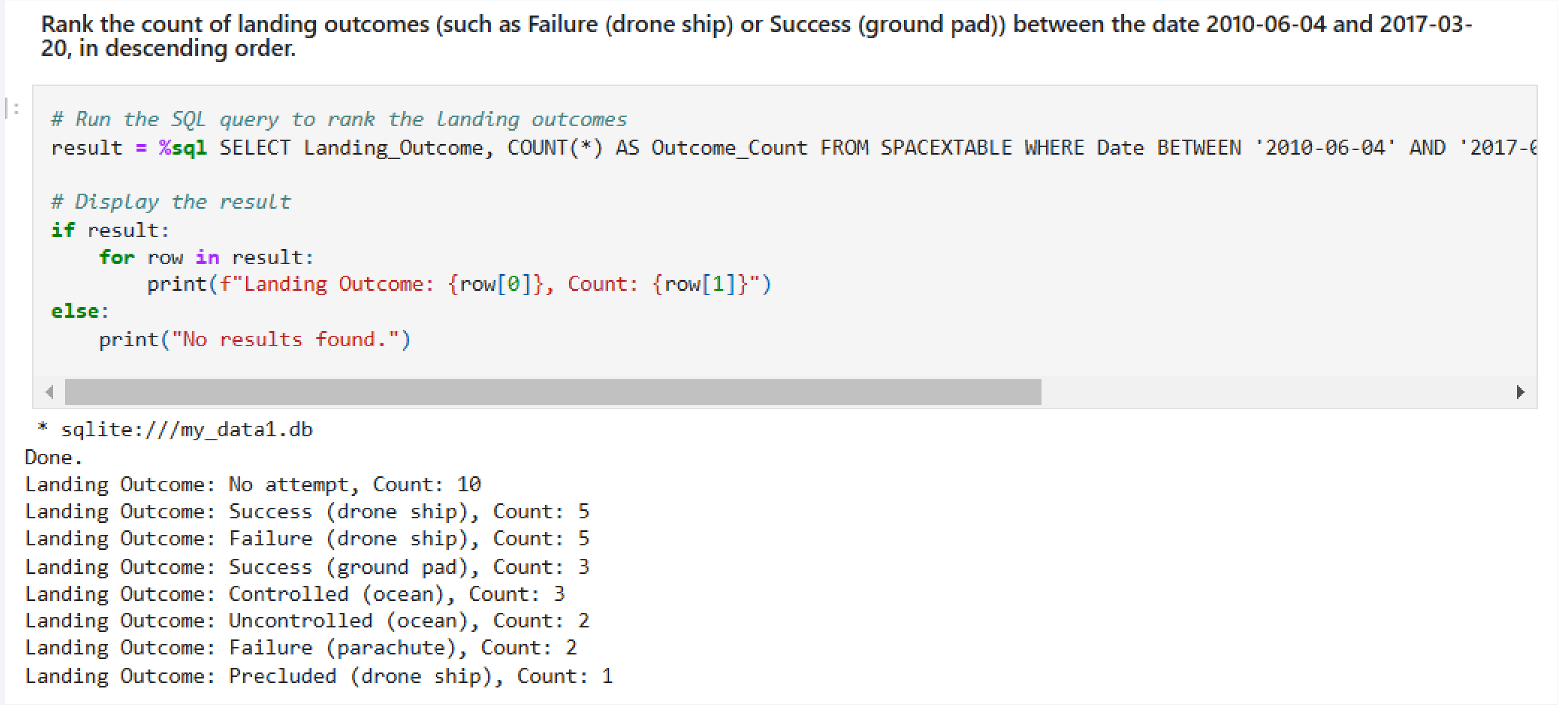
List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

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.

```
# Run the SQL query to get the required records
result = %sql SELECT substr(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE subst

# Display the result in a readable format
if result:
    for row in result:
        print(f"Month: {row[0]}, Landing Outcome: {row[1]}, Booster Version: {row[2]}, Launch Site: {row[3]}")
else:
    print("No results found.")
```

```
 * sqlite:///my_data1.db
Done.
Month: 01, Landing Outcome: Failure (drone ship), Booster Version: F9 v1.1 B1012, Launch Site: CCAFS LC-40
Month: 04, Landing Outcome: Failure (drone ship), Booster Version: F9 v1.1 B1015, Launch Site: 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.

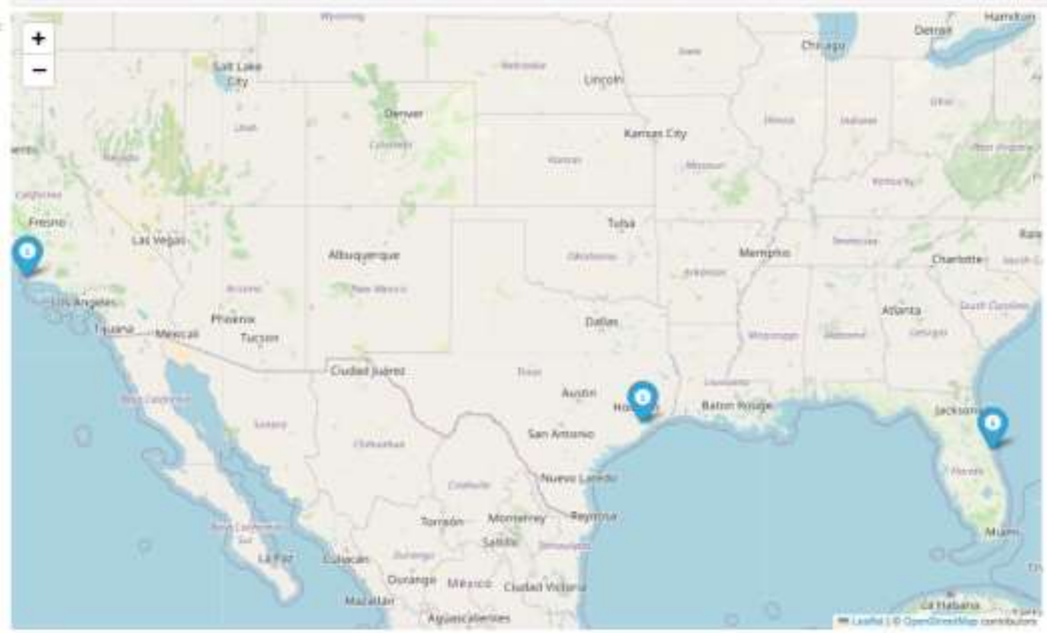
```
# Run the SQL query to rank the landing outcomes
result = %sql SELECT Landing_Outcome, COUNT(*) AS Outcome_Count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-0

# Display the result
if result:
    for row in result:
        print(f"Landing Outcome: {row[0]}, Count: {row[1]}")
else:
    print("No results found.")
```

```
 * sqlite:///my_data1.db
Done.
Landing Outcome: No attempt, Count: 10
Landing Outcome: Success (drone ship), Count: 5
Landing Outcome: Failure (drone ship), Count: 5
Landing Outcome: Success (ground pad), Count: 3
Landing Outcome: Controlled (ocean), Count: 3
Landing Outcome: Uncontrolled (ocean), Count: 2
Landing Outcome: Failure (parachute), Count: 2
Landing Outcome: Precluded (drone ship), Count: 1
```

Section 3

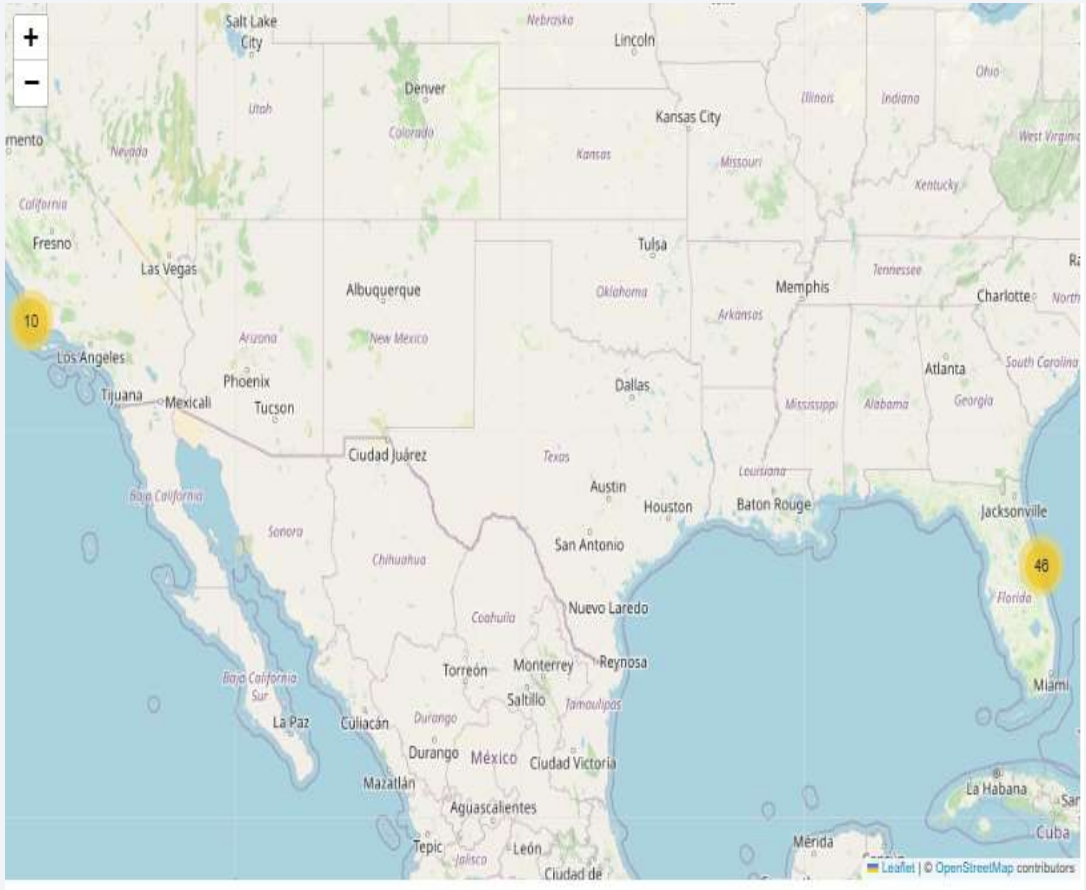
Launch Sites Proximities Analysis

Launch Sites Map Markers



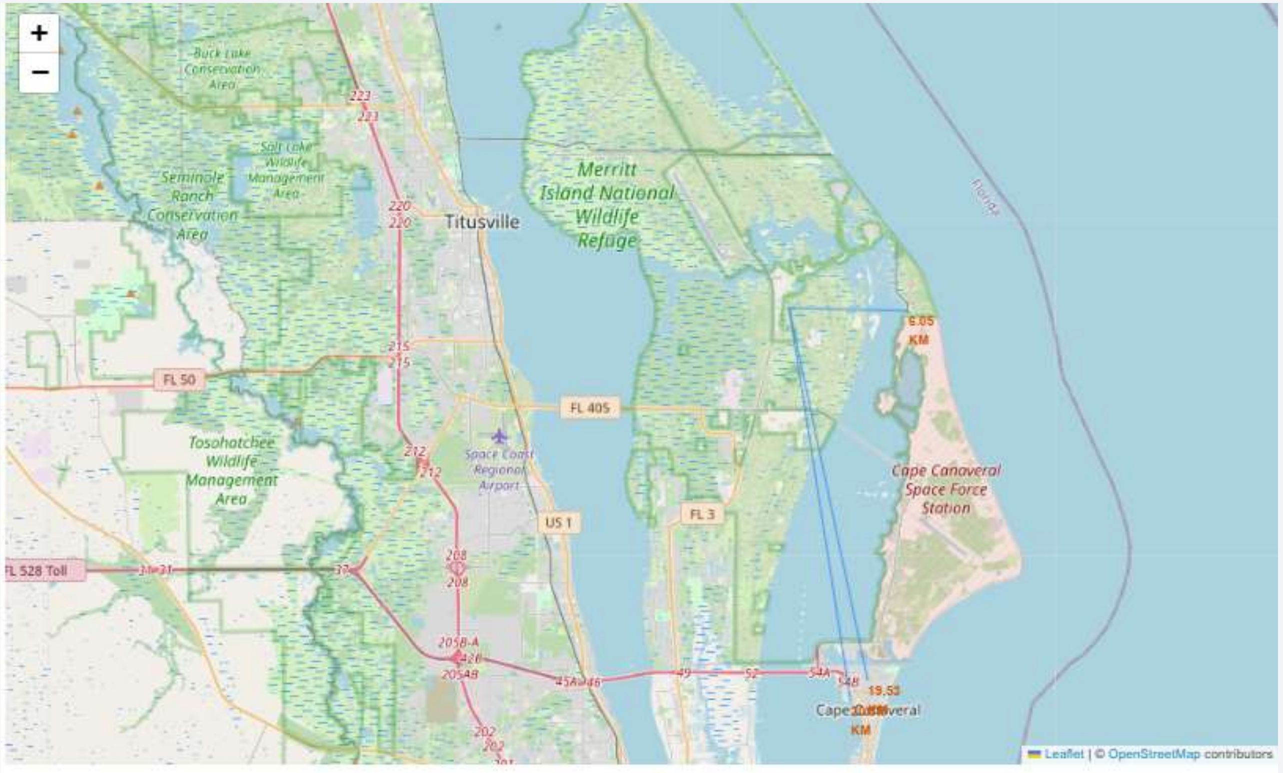
The markers on the map are the NASA and the launch sites in California and Florida

Map with successful and failed laucnhes

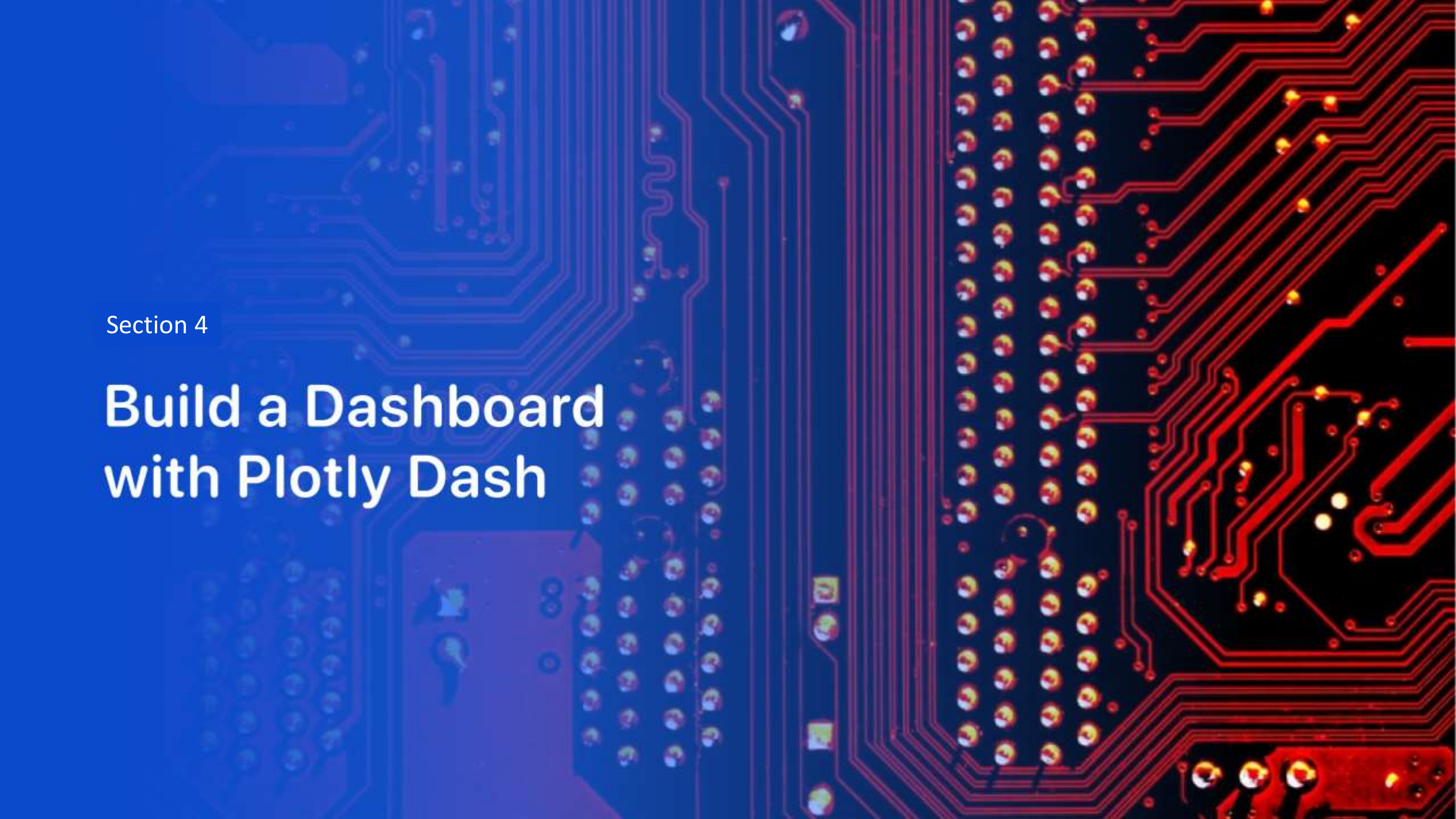


The numbers on the map are the numbers of launches from a certain radius. We can see that Florida has the most launches

Map with proximities


+
−
Buck Lake Conservation Area
Salt Lake Wildlife Management Area
Seminole Ranch Conservation Area
Titusville
Merritt Island National Wildlife Refuge
Florida
6.05 KM
FL 50
FL 405
Tosohatchee Wildlife Management Area
Space Coast Regional Airport
US 1
FL 3
Cape Canaveral Space Force Station
FL 528 Toll
19.53
Cape Canaveral
KM
Leaflet | © OpenStreetMap contributors

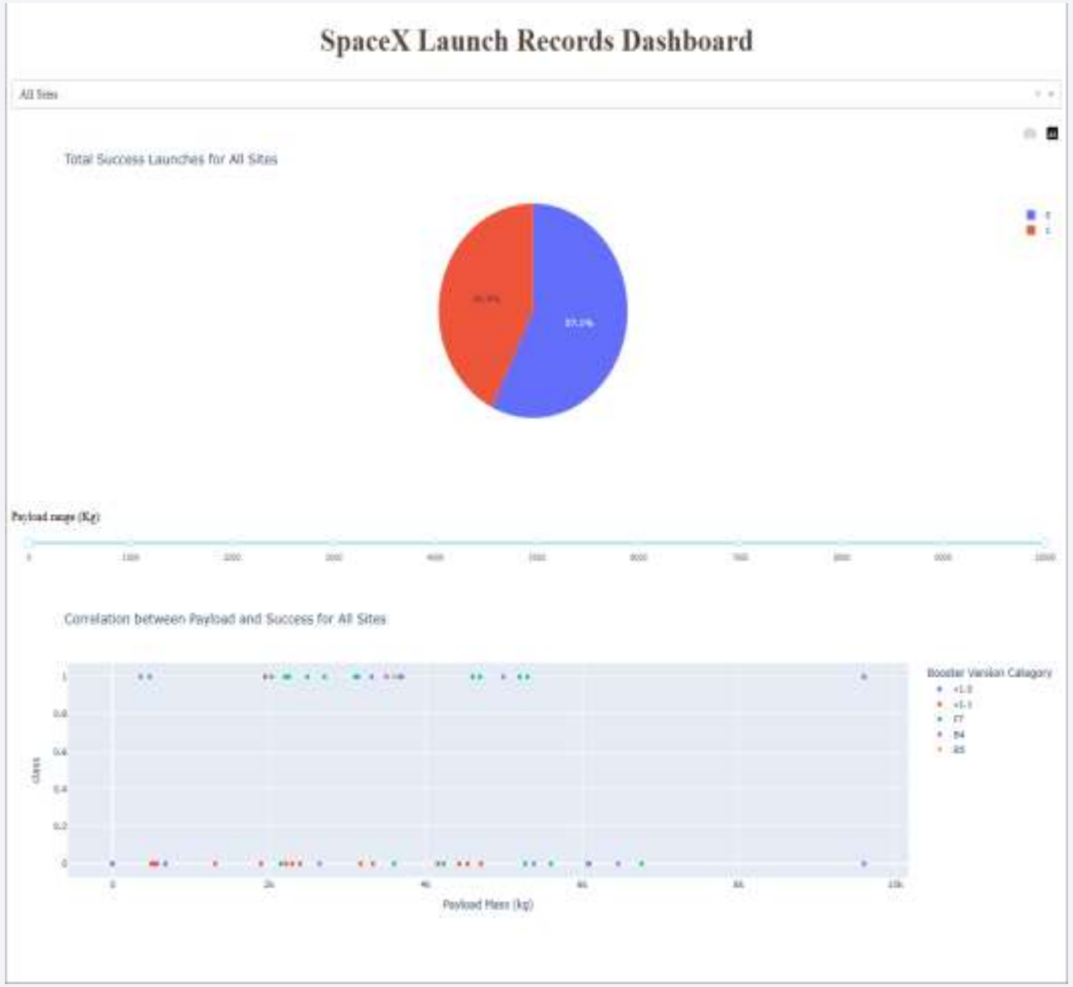
The lines are starting from Launch Site. They're connecting to the closest coastline and nearest city



Section 4

Build a Dashboard with Plotly Dash

Dashboard Pie Chart

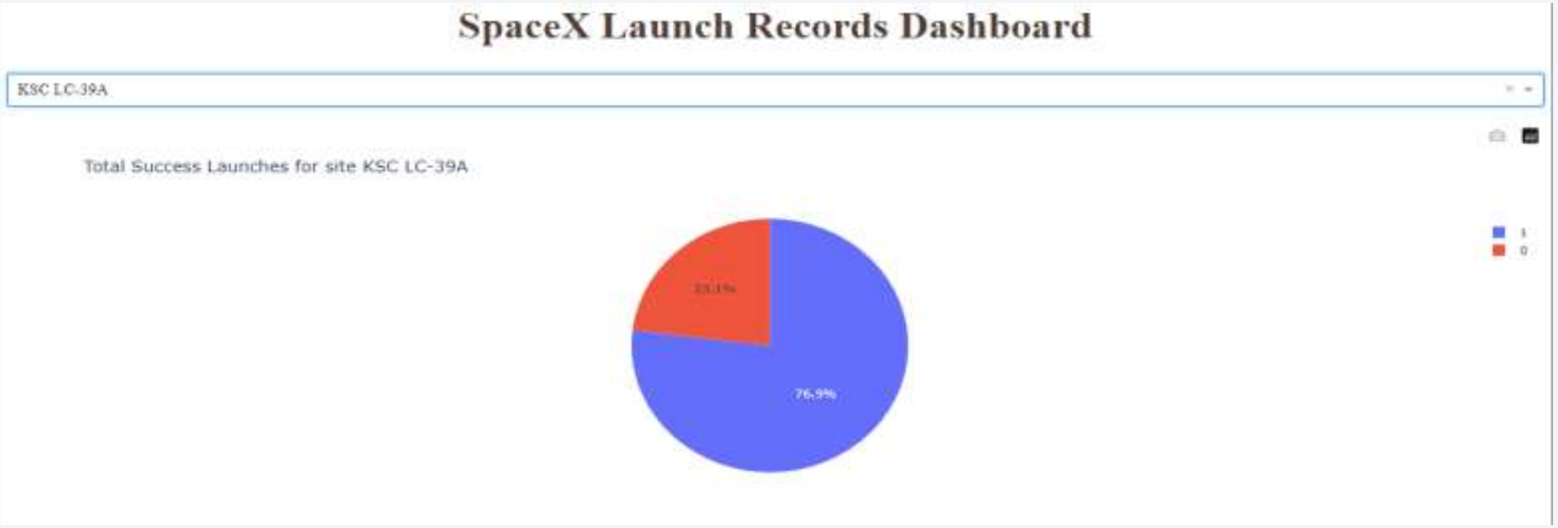


SpaceX Launch Records Dashboard

All Sites

As we can see, more than 50% of the launches succeeded.

Dashboard Pie Chart highest success rate site



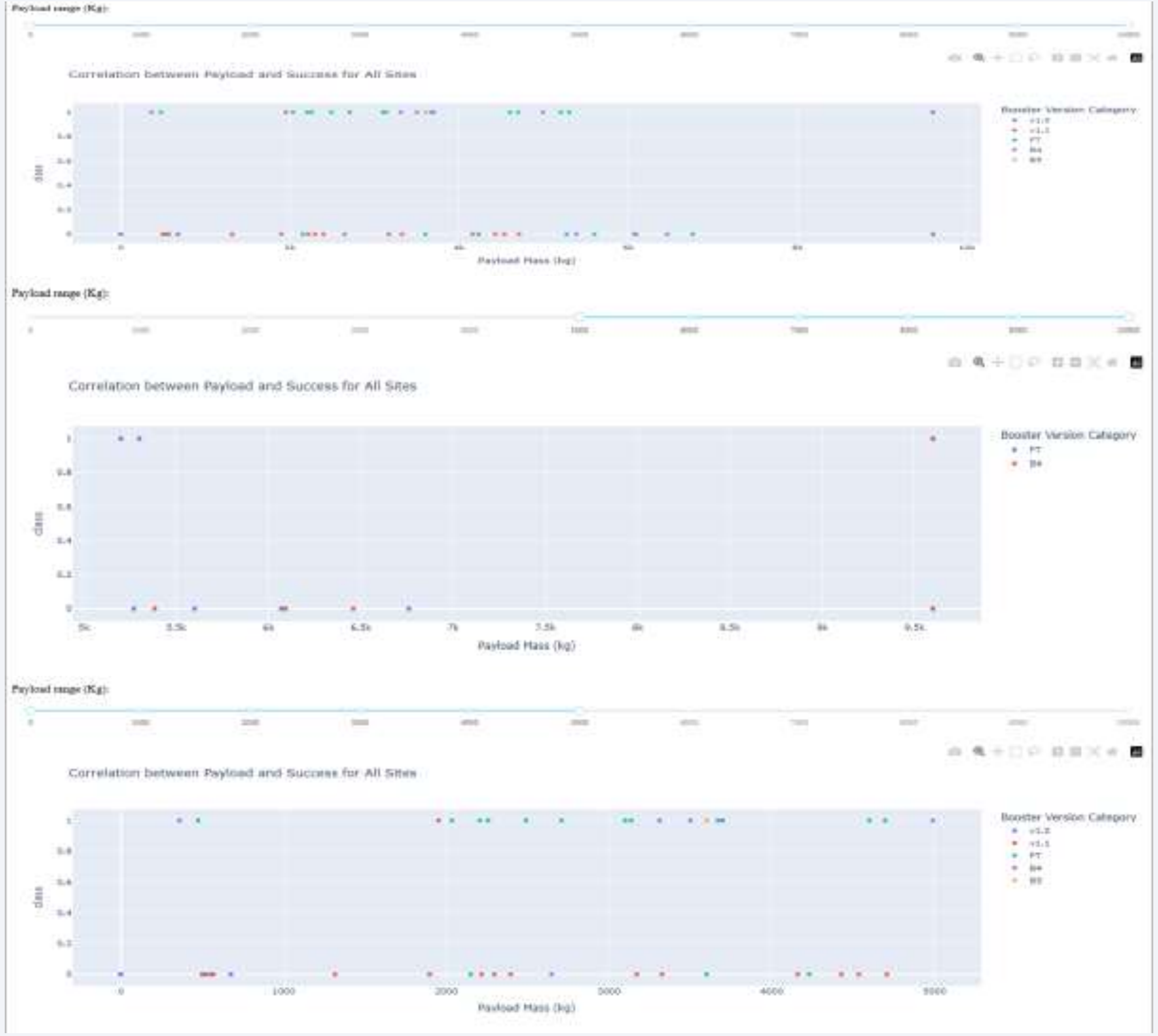
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success Launches for site KSC LC-39A

The KSC LC 39A is the launch site with the highest success rate compared to the other sites. Almost 80%

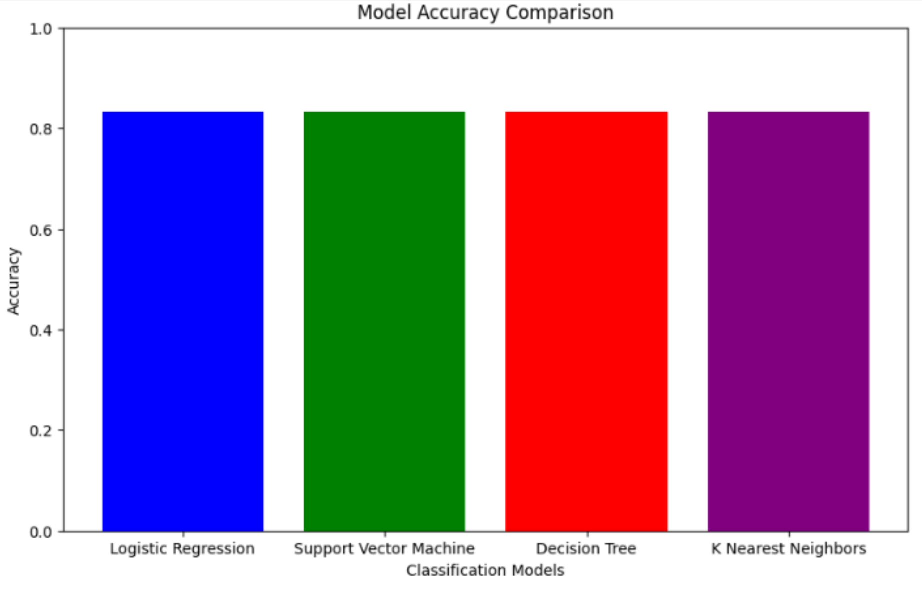
Dashboard Scatter Plot payload vs launch outcome

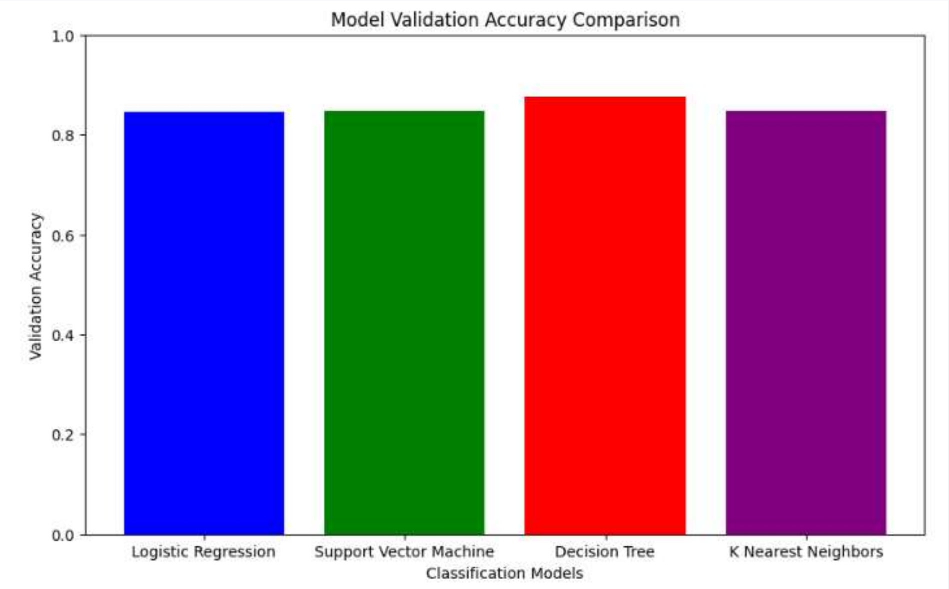


Section 5

Predictive Analysis (Classification)

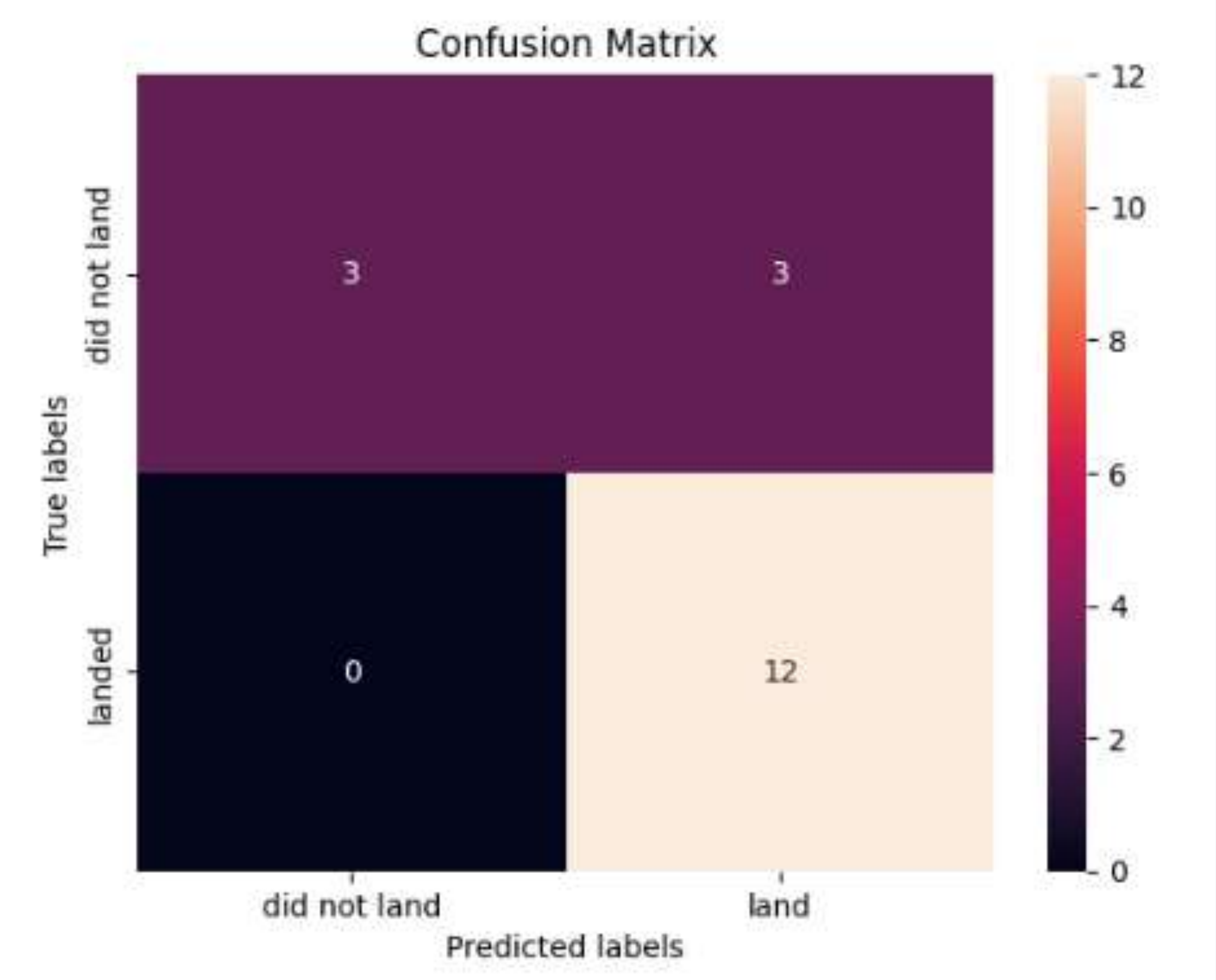
Classification Accuracy





The model accuracy is all the same, all at 83%. The model validation accuracy is quite similar but the decision tree is slighly higher than the others

Confusion Matrix



Conclusions

- The launches success went higher as the years goes by
- The success depend on the payload and the orbit type
- The models are very similar with high accuracy
- The launch sites are fairly far from cities, railrays and close to coastlines

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

```
import matplotlib.pyplot as plt# Model names and their accuraciesmodels = ['Logistic Regression', 'Support
Vector Machine', 'Decision Tree', 'K Nearest Neighbors']accuracies = [0.8333, 0.8333, 0.8333, 0.8333]# Create a
bar chartplt.figure(figsize=(10, 6))plt.bar(models, accuracies, color=['blue', 'green', 'red',
'purple'])plt.xlabel('Classification Models')plt.ylabel('Accuracy')plt.title('Model Accuracy Comparison')plt.ylim(0,
1)plt.show()
```

Thank you!