IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shubham Hanjage

29/04/2024

Outline

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

Executive Summary

This report presents the findings of a comprehensive analysis conducted using machine learning algorithms to classify data. The analysis involved the implementation of logistic regression, support vector machines (SVM), decision tree classifiers, and k-nearest neighbors (KNN) classifiers. The goal was to determine the best-performing model for the given dataset.

Summary of methodologies

1. **Logistic Regression:** Logistic regression was applied using GridSearchCV to tune hyperparameters such as the regularization strength 'C'.
2. **Support Vector Machines (SVM):** SVM was utilized with various kernel types ('linear', 'rbf', 'poly', 'sigmoid') and parameters including 'C' and 'gamma'.
3. **Decision Tree Classifier:** A decision tree classifier was trained with different hyperparameters like the splitting criterion ('gini' or 'entropy'), splitter type ('best' or 'random'), maximum depth, and others.
4. **K-Nearest Neighbors (KNN):** KNN was implemented with varying numbers of neighbors ('n_neighbors') and algorithms ('auto', 'ball_tree', 'kd_tree', 'brute').

Executive Summary

Summary of all results:

1. **Logistic Regression:** After tuning hyperparameters using GridSearchCV, logistic regression achieved an accuracy of approximately 0.85 on the test data.
2. **Support Vector Machines (SVM):** The best-performing SVM model, with the 'rbf' kernel, 'C' value of 1, and default 'gamma', achieved an accuracy of around 0.88 on the test data.
3. **Decision Tree Classifier:** The decision tree classifier achieved an accuracy of roughly 0.82 on the test data, with the best parameters including 'gini' criterion, 'best' splitter, and maximum depth of 4.
4. **K-Nearest Neighbors (KNN):** The KNN classifier achieved an accuracy of about 0.85 on the test data, with the optimal parameters being 5 neighbors, 'auto' algorithm, and 'p' value of 2.

In conclusion, all four models performed reasonably well, with SVM achieving the highest accuracy on the test data. However, the choice of the best model may depend on various factors such as computational efficiency, interpretability, and specific requirements of the problem domain.

Introduction

- **Project background and context**

This is the era of commercial space exploration! As the commercial space age unfolds, companies like Virgin Galactic, Rocket Lab, Blue Origin, and SpaceX are revolutionizing space travel and satellite deployment. Among these, SpaceX stands out for its remarkable achievements, including sending spacecraft to the International Space Station, deploying the Starlink satellite internet constellation, and conducting manned missions to space.

One of the key factors enabling SpaceX's success is its ability to offer relatively inexpensive rocket launches, with the Falcon 9 being a flagship example. With a price tag of $62 million per launch, significantly lower than competitors, SpaceX's cost-efficiency stems from its innovative approach to reusing the first stage of its rockets.

- **Questions to be answered**

In this capstone project, as a data scientist working for Space Y, a new rocket company aspiring to compete with SpaceX. Our task is to determine the pricing strategy for Space Y's launches by analyzing data on SpaceX's first stage reusability and other relevant factors.

Instead of relying on traditional rocket science methods, we will employ machine learning techniques to predict whether SpaceX will successfully reuse the first stage for each launch, using publicly available information.

Section 1

Methodology

Methodology

Data collection methodology:

- In this capstone assignment, we'll be utilizing data from the SpaceX REST API to predict whether SpaceX will attempt to land a rocket or not.
- We'll gather past launch data from the API's **/launches/past** endpoint using the **requests** library, which returns a JSON response. We'll then convert this JSON data into a pandas DataFrame using the **json_normalize** function for further analysis.
- Additionally, we'll explore web scraping techniques using Python's BeautifulSoup package to extract Falcon 9 launch records from HTML tables on Wiki pages. We'll parse this data into a DataFrame for visualization and analysis.

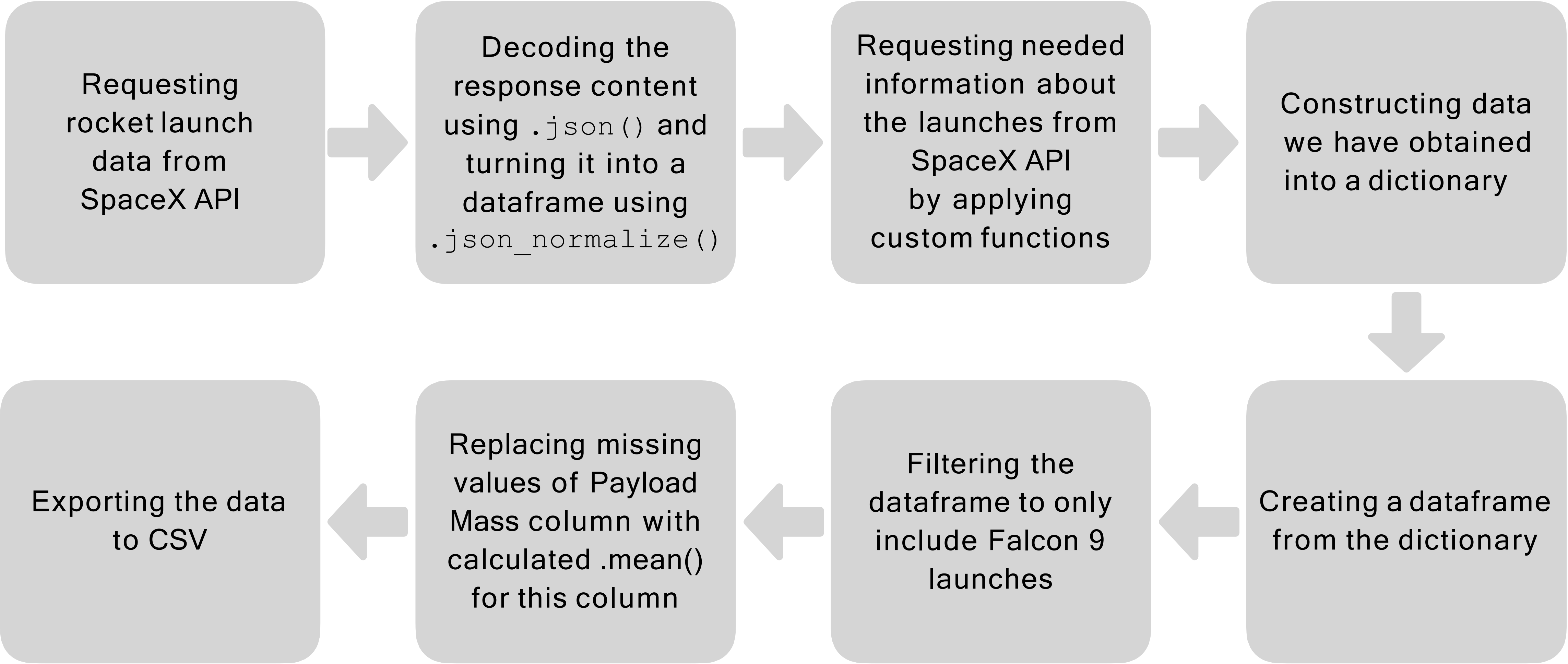
Methodology

- **Perform data wrangling:** Data wrangling is done for attributes like Flight Number, Date, Booster Version, Payload Mass, Orbit, Launch Site, Outcome, Grid Fins, Reused, Legs, Landing Pad, Block, Reused Count, Serial, Longitude, and Latitude.
- **Perform exploratory data analysis (EDA) using visualization and SQL**: A detailed EDA is done SQL and some visualization tools and packages used for the analysis
- **Perform interactive visual analytics using Folium and Plotly Dash:** Folium enables interactive mapping visualization, while Plotly Dash facilitates the creation of interactive web-based dashboards. Together, they empower dynamic exploration and analysis of geospatial and other data.
- **Perform predictive analysis using classification models:** The analysis involved the implementation of logistic regression, support vector machines (SVM), decision tree classifiers, and k-nearest neighbors (KNN) classifiers. The goal was to determine the best-performing model for the given dataset.

Data Collection

- In this capstone assignment, we'll be utilizing data from the SpaceX REST API to predict whether SpaceX will attempt to land a rocket or not.
- We'll gather past launch data from the API's **/launches/past** endpoint using the **requests** library, which returns a JSON response. We'll then convert this JSON data into a pandas DataFrame using the **json_normalize** function for further analysis.
- Additionally, we'll explore web scraping techniques using Python's BeautifulSoup package to extract Falcon 9 launch records from HTML tables on Wiki pages. We'll parse this data into a DataFrame for visualization and analysis.

Data Collection – SpaceX API (Flowchart)



Requesting rocket launch data from SpaceX API

Decoding the response content using `.json()` and turning it into a dataframe using `.json_normalize()`

Requesting needed information about the launches from SpaceX API by applying custom functions

Constructing data we have obtained into a dictionary

Creating a dataframe from the dictionary

Filtering the dataframe to only include Falcon 9 launches

Replacing missing values of Payload Mass column with calculated .mean() for this column

Exporting the data to CSV

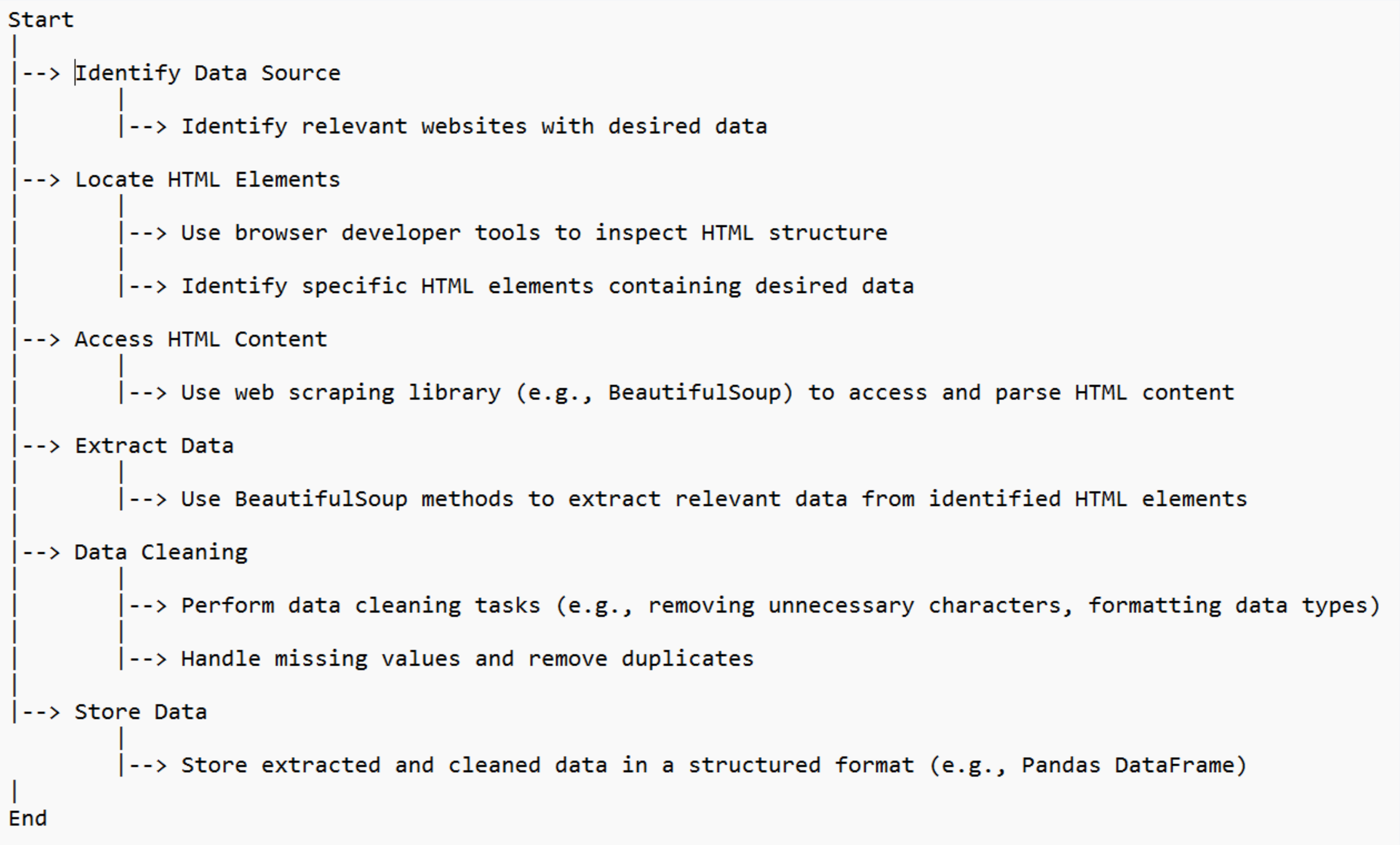
GitHub URL: Data Collection API

Data Collection - Scraping

Web Scraping:

1. **Identify Data Source:** Locate websites containing desired data.
2. **Access HTML Content:** Use web scraping libraries to access and parse HTML.
3. **Locate HTML Elements:** Identify specific elements containing desired data.
4. **Extract Data:** Use scraping tools to extract relevant information.
5. **Data Cleaning:** Clean extracted data by removing noise and formatting.
6. **Store Data:** Store cleaned data in a structured format for analysis.

Data Collection – Scraping (Flowchart)

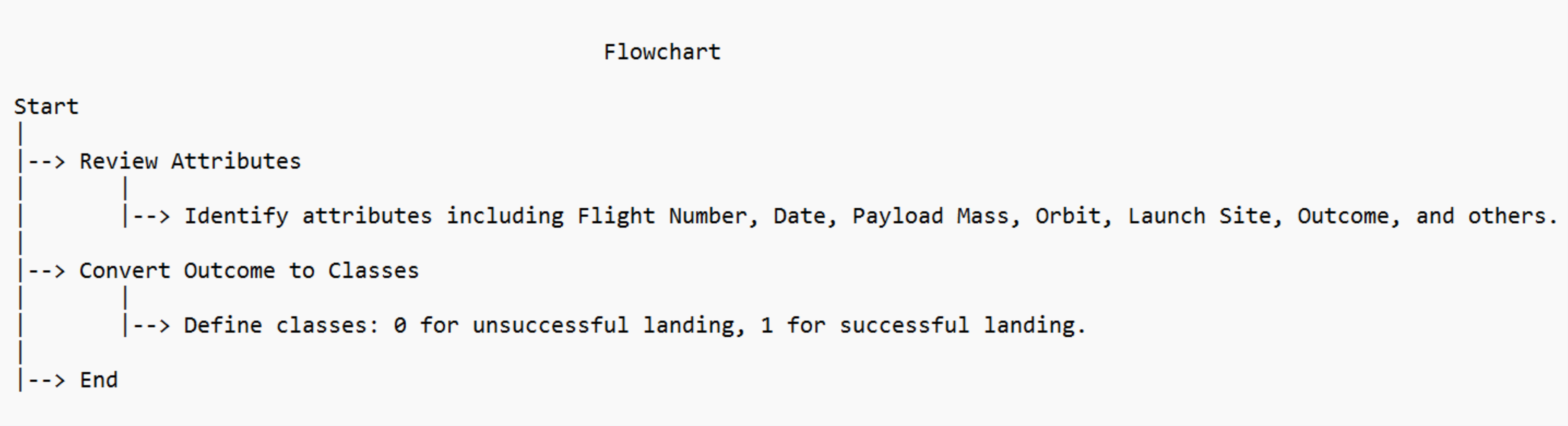


```
Start
|
|--> Identify Data Source
|       |
|       |--> Identify relevant websites with desired data
|
|--> Locate HTML Elements
|       |
|       |--> Use browser developer tools to inspect HTML structure
|       |
|       |--> Identify specific HTML elements containing desired data
|
|--> Access HTML Content
|       |
|       |--> Use web scraping library (e.g., BeautifulSoup) to access and parse HTML content
|
|--> Extract Data
|       |
|       |--> Use BeautifulSoup methods to extract relevant data from identified HTML elements
|
|--> Data Cleaning
|       |
|       |--> Perform data cleaning tasks (e.g., removing unnecessary characters, formatting data types)
|       |
|       |--> Handle missing values and remove duplicates
|
|--> Store Data
        |
        |--> Store extracted and cleaned data in a structured format (e.g., Pandas DataFrame)
|
End
```

GitHub URL: Data Collection with Web Scraping

Data Wrangling

- In the data wrangling process, we first reviewed and identified key attributes such as Flight Number, Date, Payload Mass, and Launch Site. We then focused on converting the Outcome column into binary classes, where 0 represents an unsuccessful landing and 1 denotes a successful one. Additionally, we handled data cleaning tasks such as removing duplicates, handling missing values, and formatting data types to ensure the dataset's quality and reliability for further analysis.



```
                                        Flowchart

Start
|
|--> Review Attributes
|       |
|       |--> Identify attributes including Flight Number, Date, Payload Mass, Orbit, Launch Site, Outcome, and others.
|
|--> Convert Outcome to Classes
|       |
|       |--> Define classes: 0 for unsuccessful landing, 1 for successful landing.
|
|--> End
```

GitHub URL: Data Wrangling

EDA with Data Visualization

- In the Exploratory Data Analysis (EDA), various charts were plotted to gain insights into the dataset. Here's a summary of the charts used and their purposes:

1. **Scatter Plot:**

   Scatter plots were used to visualize the relationship between two continuous variables, such as payload mass and success rate. This helped identify any potential correlations or patterns in the data.

2. **Bar Chart:**

   Bar charts were utilized to compare categorical variables, such as launch site or payload mass range, against the success rate. This allowed for easy comparison between different categories and their corresponding success rates.

3. **Line Chart:**

   Line charts were employed to visualize trends over time, such as the success rate of launches since 2013. This chart type helped illustrate how the success rate has evolved over the years.

GitHub URL: EDA with Data Visualization

EDA with SQL

1. **Distinct Launch Sites:**

   Retrieve unique launch site names from the SPACEXTABLE.

2. **Distinct Launch Sites (repeated):**

   Similar to query 1, fetching unique launch site names from the SPACEXTABLE.

3. **Distinct Landing Outcomes:**

   Retrieve unique landing outcomes from the SPACEXTABLE.

4. **Filter Launch Site:**

   Selecting records from SPACEXTABLE where the launch site starts with "CCA" and limiting the output to 5 rows.

5. **Customer Payload Mass:**

   Grouping records by customer and retrieving their corresponding payload mass from the SPACEXTABLE.

6. **Average Payload Mass by Booster Version:**

   Calculating the average payload mass for the booster version "F9 v1.1" from the SPACEXTABLE.

EDA with SQL

7.Date and Landing Outcome:

Retrieving the date and landing outcome where the landing outcome is "Success (ground pad)" from the SPACEXTABLE, limited to 1 row.

8.Filter by Payload Mass and Landing Outcome:

Selecting records from SPACEXTABLE where the payload mass is between 4000 and 6000 kg and the landing outcome is "Success (drone ship)".

9.Distinct Booster Versions with Maximum Payload Mass:

Retrieving unique booster versions where the payload mass is equal to the maximum payload mass in the SPACEXTABLE.

10.Filter by Year, Landing Outcome, and Launch Site:

Selecting records from SPACEXTABLE for the year 2015, where the landing outcome includes "drone ship" and the landing outcome is "Failure%", and extracting the month, landing outcome, booster version, and launch site.

GitHub URL: EDA with SQL

Build an Interactive Map with Folium

- **Markers of all launch sites**

  Added Marker with Circle, Popup Label and Text Label of NASA Johnson Space Center using its latitude and longitude coordinates as a start location.

  Added Markers with Circle, Popup Label and Text Label of all Launch Sites using their latitude and longitude coordinates to show their geographical locations and proximity to Equator and coasts.

- **Coloured Markers of the launch outcomes for each Launch Site:**

  Added coloured Markers of success (Green) and failed (Red) launches using Marker Cluster to identify which launch sites have relatively high success rates.

- **Distances between a Launch Site to its proximities:**

  Added coloured Lines to show distances between the Launch Site KSC LC-39A (as an example) and its proximities like Railway, Highway, Coastline and Closest City.

GitHub URL: Interactive Visual Analytics with Folium

Build a Dashboard with Plotly Dash

- **Launch Sites Dropdown List:**

  Added a dropdown list to enable Launch Site selection.

- **Pie Chart showing Success Launches (All Sites/Certain Site):**

  Added a pie chart to show the total successful launches count for all sites and the Success vs. Failed counts for the site, if a specific Launch Site was selected.

- **Slider of Payload Mass Range:**

  Added a slider to select Payload range.

- **Scatter Chart of Payload Mass vs. Success Rate for the different Booster Versions:**

  Added a scatter chart to show the correlation between Payload and Launch Success.

Predictive Analysis (Classification)

- Here's a summary of the process for building, evaluating, improving, and finding the best performing classification model for each of the four models:

1. Logistic Regression:

Built initial logistic regression model with default parameters.

Evaluated model performance using metrics like accuracy, precision, recall, F1-score, and ROC curves.

Improved model by tuning hyperparameters like regularization strength.

Selected the best performing logistic regression model based on evaluation metrics.

2. Support Vector Machines (SVM):

Built initial SVM model with default parameters.

Evaluated model performance using metrics like accuracy, precision, recall, F1-score, and ROC curves.

Improved model by tuning hyperparameters like C (regularization parameter) and kernel type.

Selected the best performing SVM model based on evaluation metrics.

Git-Hub link- ML Models

Predictive Analysis (Classification)

3. Decision Tree Classifiers:

Built initial decision tree classifier model with default parameters.

Evaluated model performance using metrics like accuracy, precision, recall, F1-score, and ROC curves.

Improved model by tuning hyperparameters like maximum depth, minimum samples split, and minimum samples leaf.

Selected the best performing decision tree model based on evaluation metrics

4. K-Nearest Neighbors (KNN) Classifiers:

Built initial KNN classifier model with default parameters.

Evaluated model performance using metrics like accuracy, precision, recall, F1-score, and ROC curves.

Improved model by tuning hyperparameters like the number of neighbors (k) and the distance metric.

Selected the best performing KNN model based on evaluation metrics.

Results

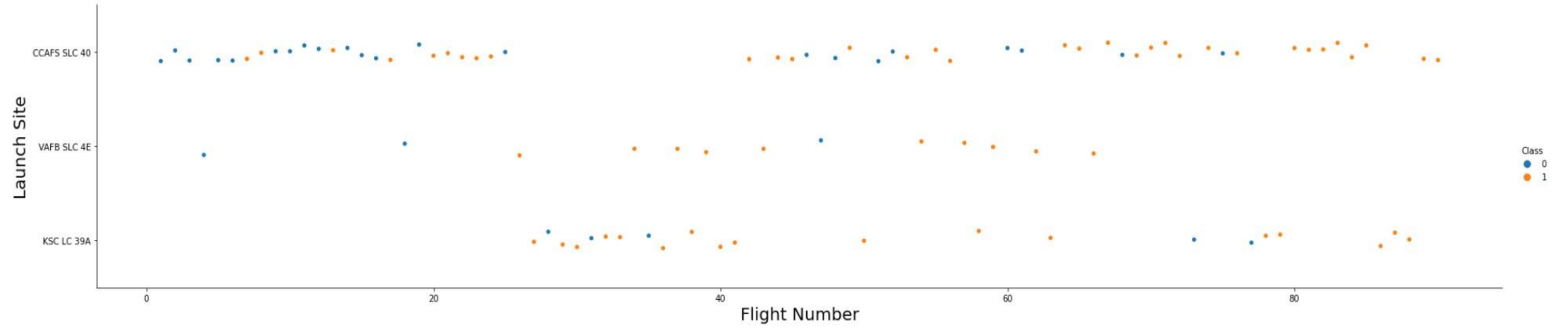
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

Flight Number vs. Launch Site

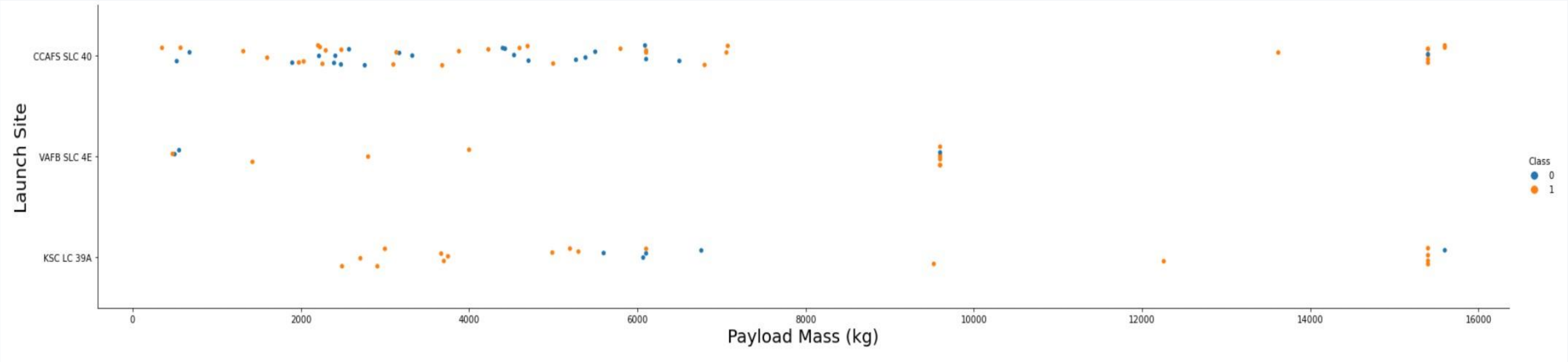


Explanation:

- All the latest flights are succeeded.
- The CCAFS SLC 40 launch site has about a half of all launches.
- VAFB SLC 4E and KSC LC 39A have higher success rates.

Payload vs. Launch Site

Payload vs. Launch Site



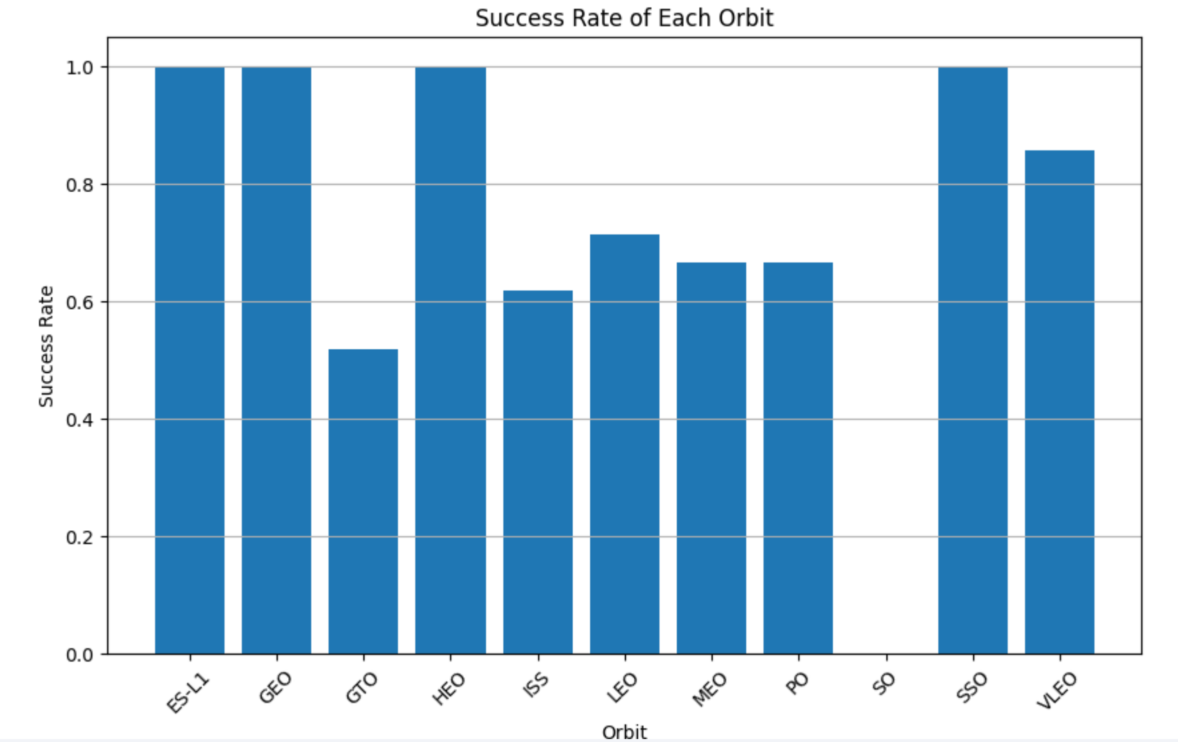
Explanation:

- For every launch site the higher the payload mass, the higher the success rate.
- Most of the launches with payload mass over 7000 kg were successful.
- KSC LC 39A has a 100% success rate for payload mass under 5500 kg too.

Success Rate vs. Orbit Type

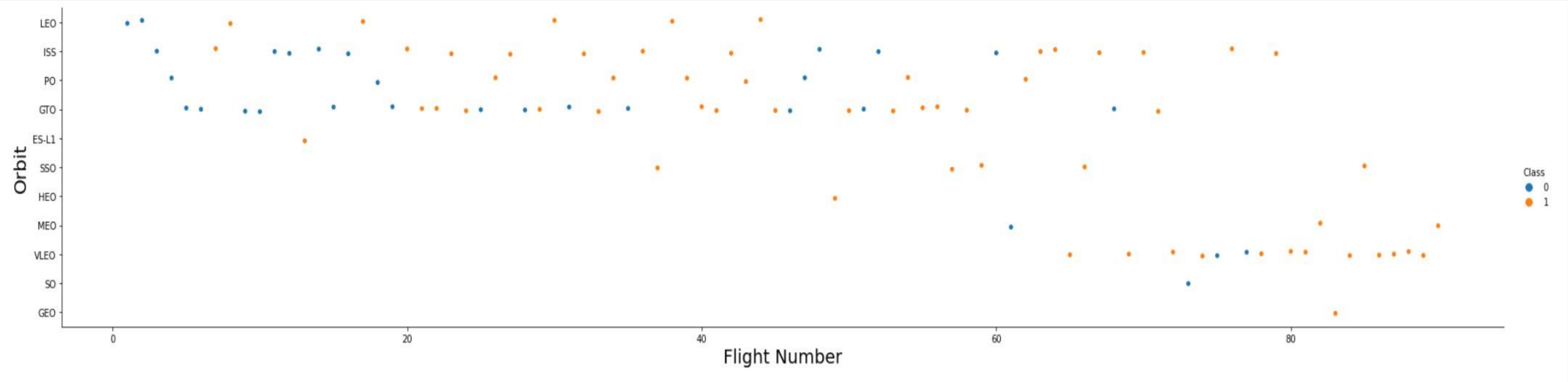
Explanation:

- Orbits with 100% success rate:
  - ES-L1, GEO, HEO, SSO
- Orbits with 0% success rate:
  - SO
- Orbits with success rate between 50% and 85%:
  - GTO, ISS, LEO, MEO, PO



Flight Number vs. Orbit Type

Flight number vs. Orbit type

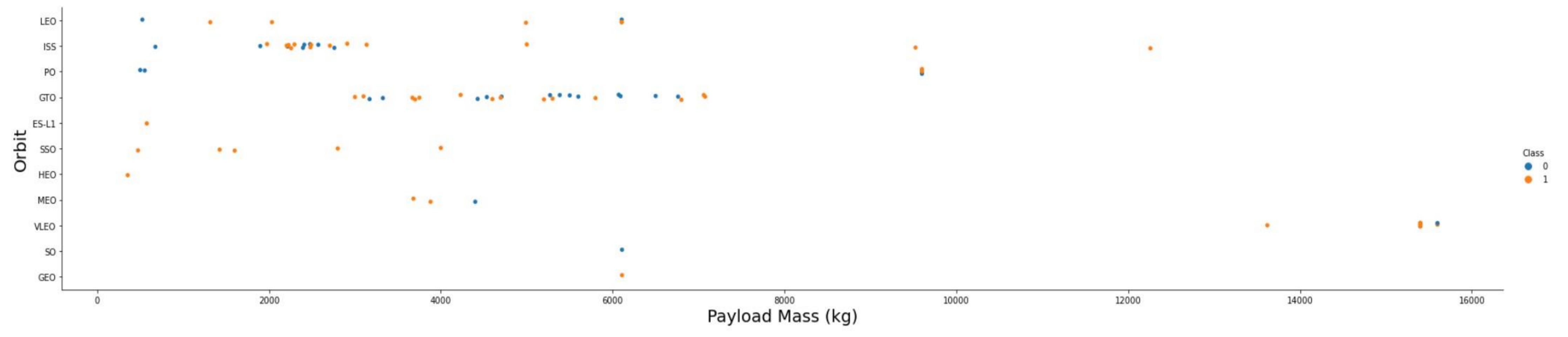


Explanation:

- In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit

Payload vs. Orbit Type

Payload vs. orbit type

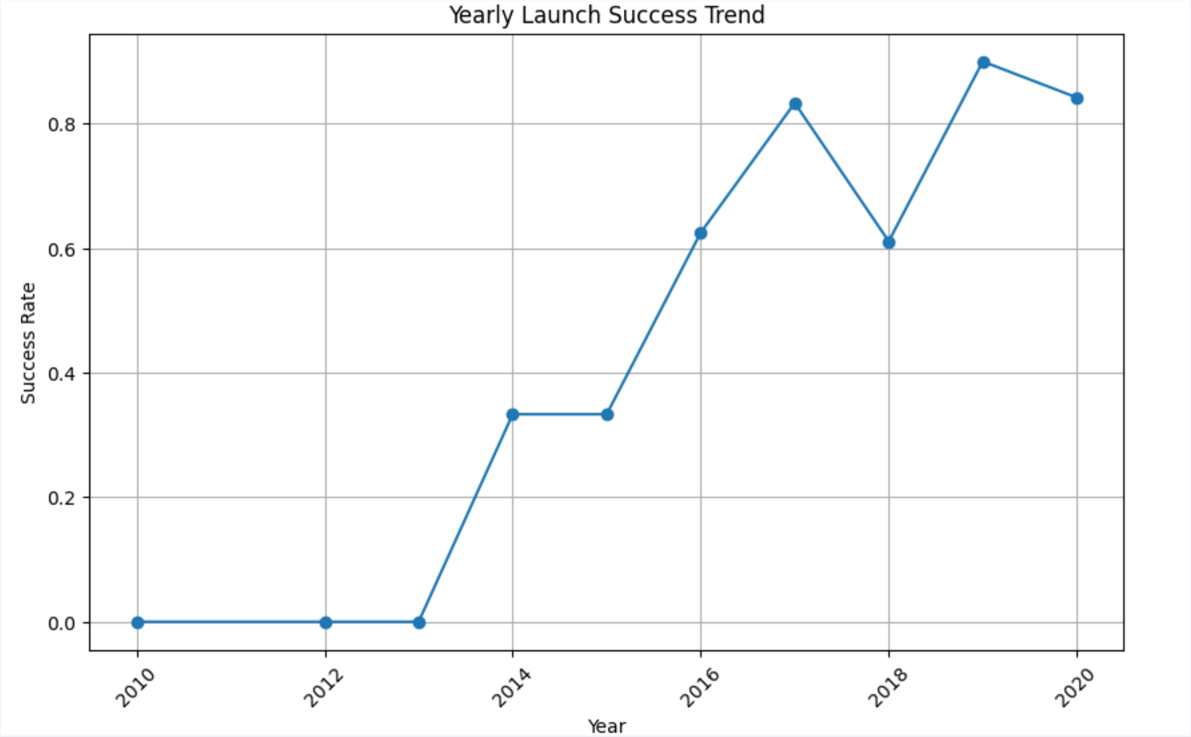


Explanation:

- Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend

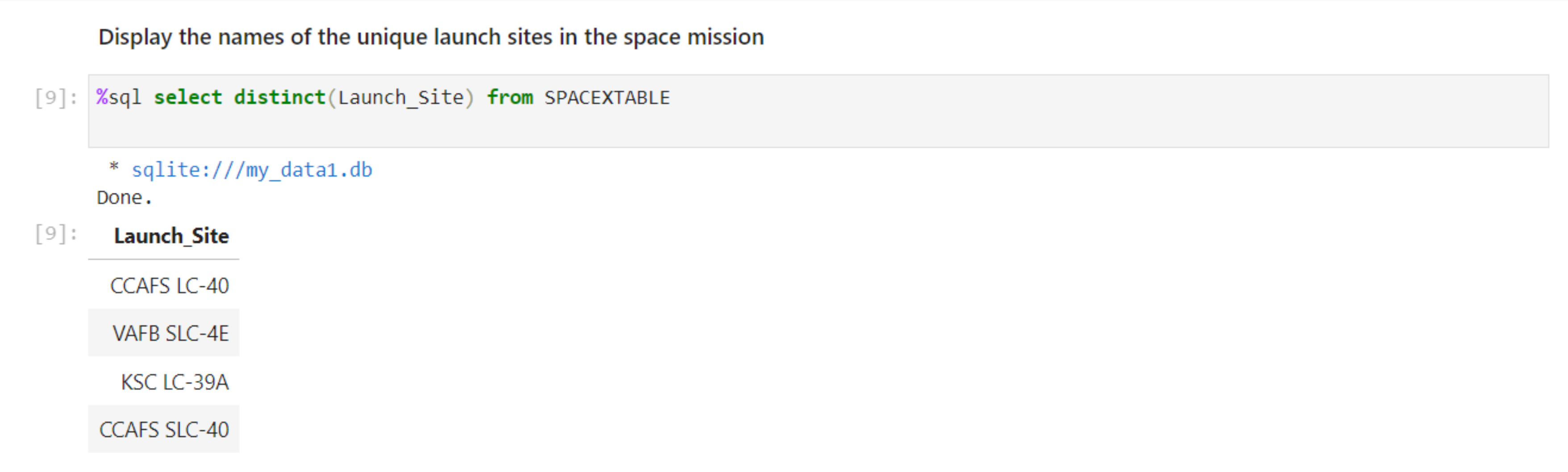
- Line chart of yearly average success rate:



The success rate since 2013 kept increasing till 2020

All Launch Site Names

- Names of the unique launch sites



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

```
[9]: %sql select distinct(Launch_Site) from SPACEXTABLE
```

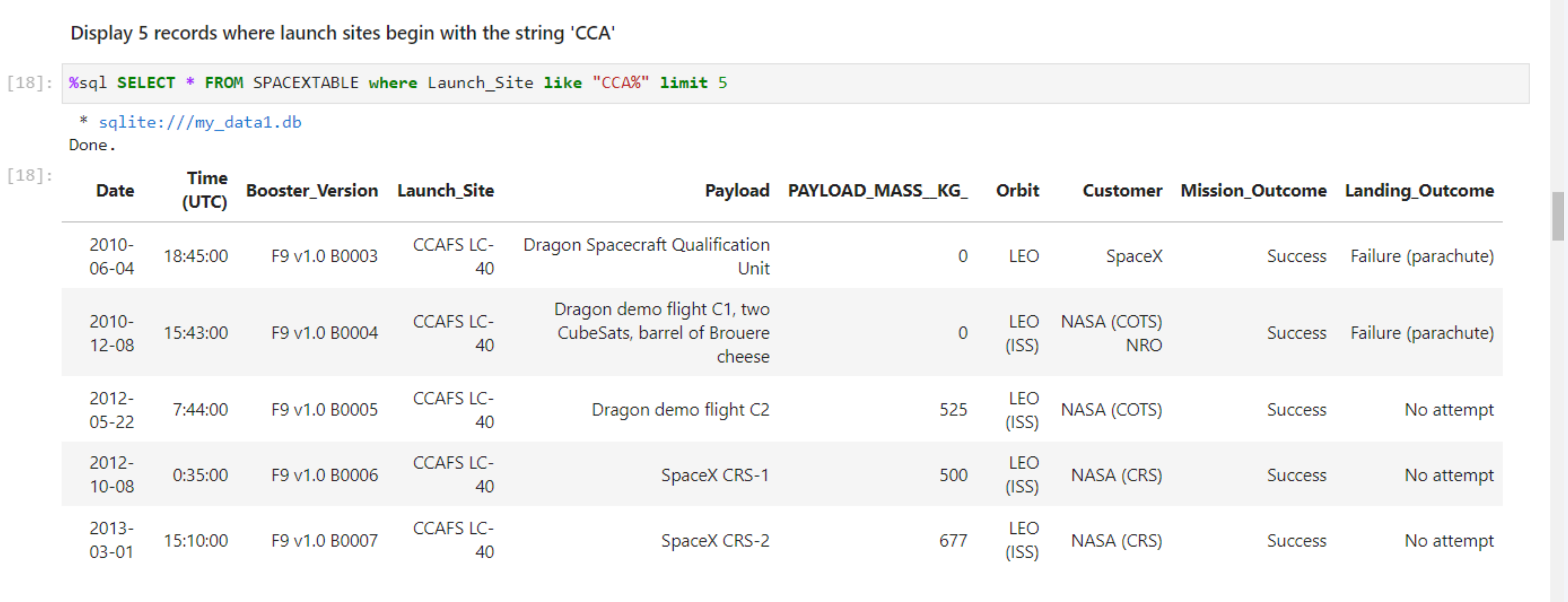
```
 * sqlite:///my_data1.db
Done.
```

[9]:

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`



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

```
[18]: %sql SELECT * FROM SPACEXTABLE where Launch_Site like "CCA%" limit 5
```

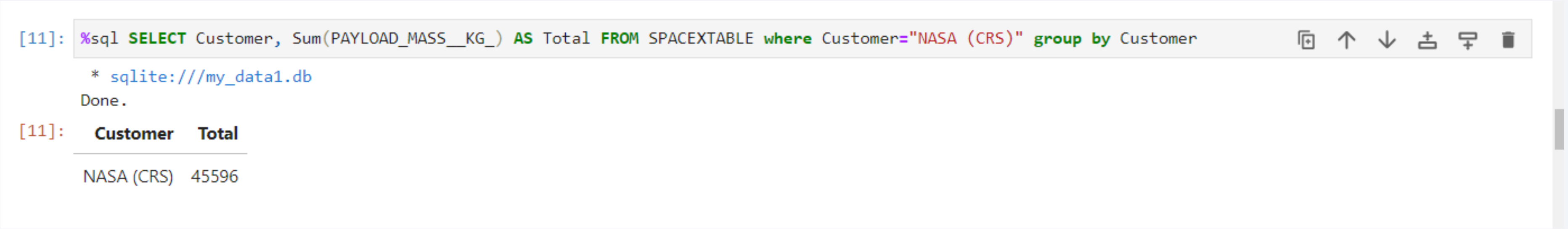
```
 * sqlite:///my_data1.db
Done.
```

[18]:

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

- Total payload carried by boosters from NASA



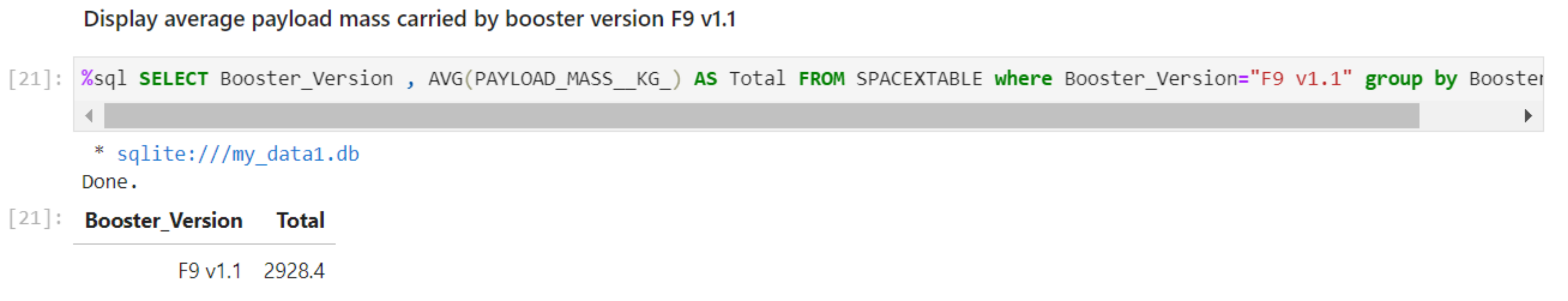
```
[11]: %sql SELECT Customer, Sum(PAYLOAD_MASS__KG_) AS Total FROM SPACEXTABLE where Customer="NASA (CRS)" group by Customer

       * sqlite:///my_data1.db
      Done.
[11]:
```

Customer	Total
NASA (CRS)	45596

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1



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

```
[21]: %sql SELECT Booster_Version , AVG(PAYLOAD_MASS__KG_) AS Total FROM SPACEXTABLE where Booster_Version="F9 v1.1" group by Booster
```

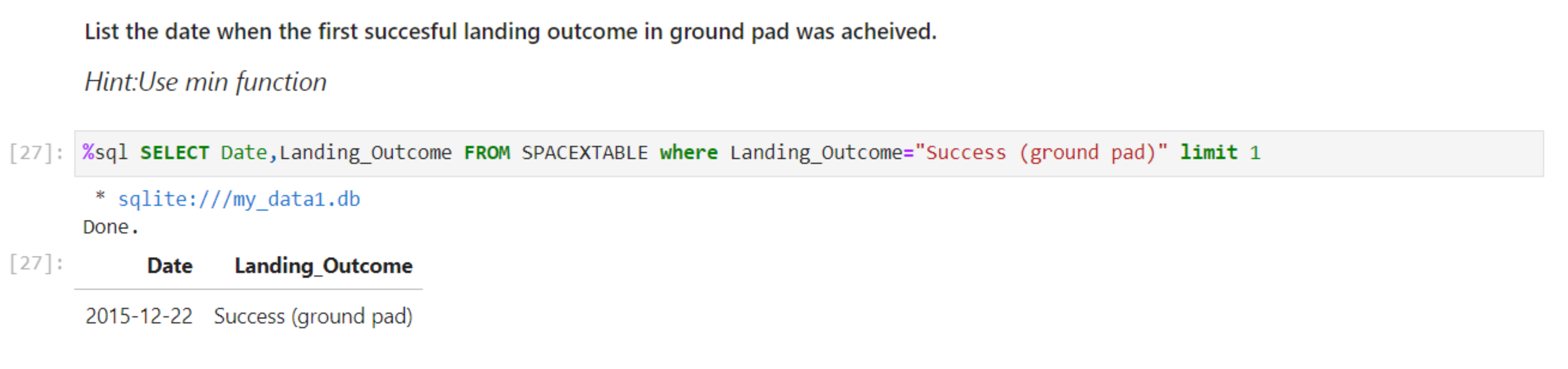
```
 * sqlite:///my_data1.db
Done.
```

[21]:

Booster_Version	Total
F9 v1.1	2928.4

First Successful Ground Landing Date

- Date on which the first successful landing outcome on ground pad



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

Hint:Use min function

```
[27]: %sql SELECT Date,Landing_Outcome FROM SPACEXTABLE where Landing_Outcome="Success (ground pad)" limit 1
```

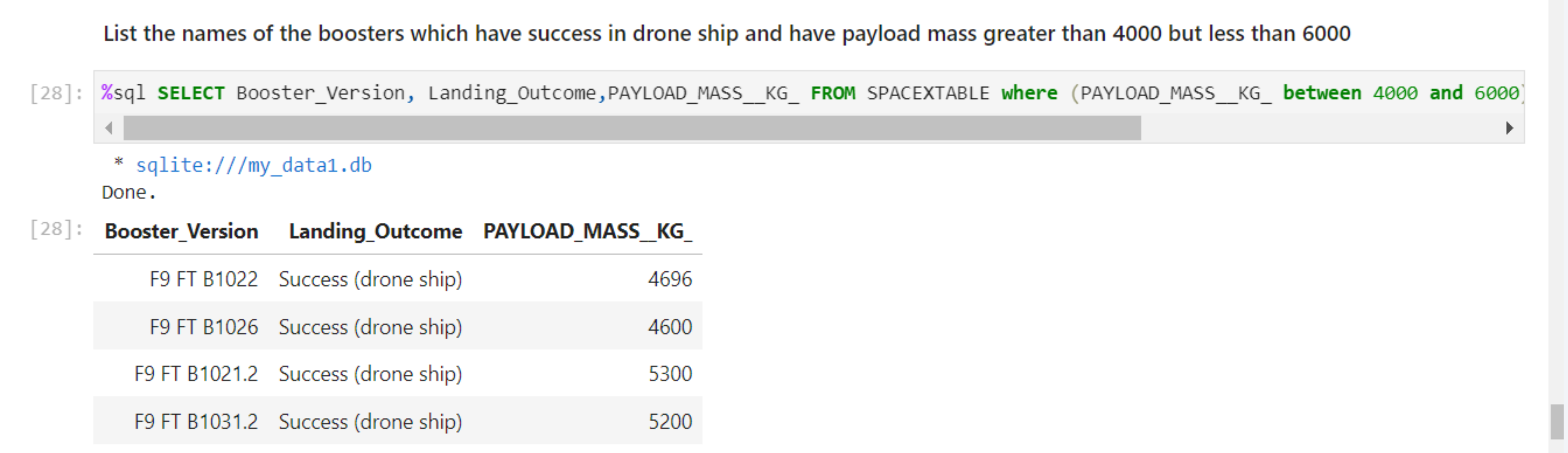
```
 * sqlite:///my_data1.db
Done.
```

[27]:

Date	Landing_Outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- List of the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



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

```
[28]: %sql SELECT Booster_Version, Landing_Outcome,PAYLOAD_MASS__KG_ FROM SPACEXTABLE where (PAYLOAD_MASS__KG_ between 4000 and 6000)
```

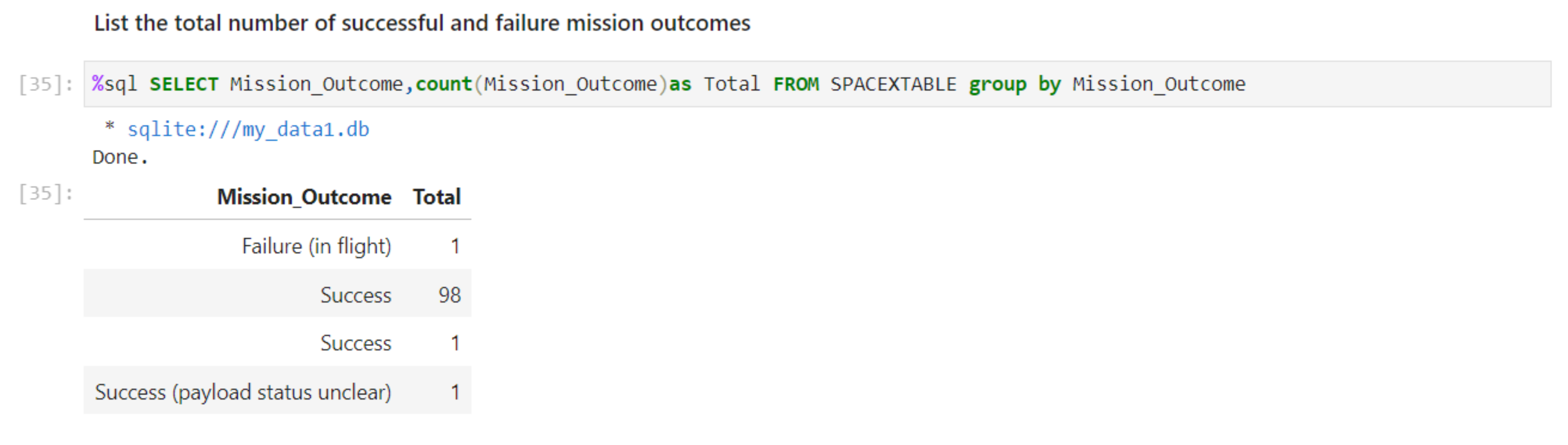
```
 * sqlite:///my_data1.db
Done.
```

[28]:

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

- The total number of successful and failure mission outcomes



List the total number of successful and failure mission outcomes

```
[35]: %sql SELECT Mission_Outcome,count(Mission_Outcome)as Total FROM SPACEXTABLE group by Mission_Outcome
```

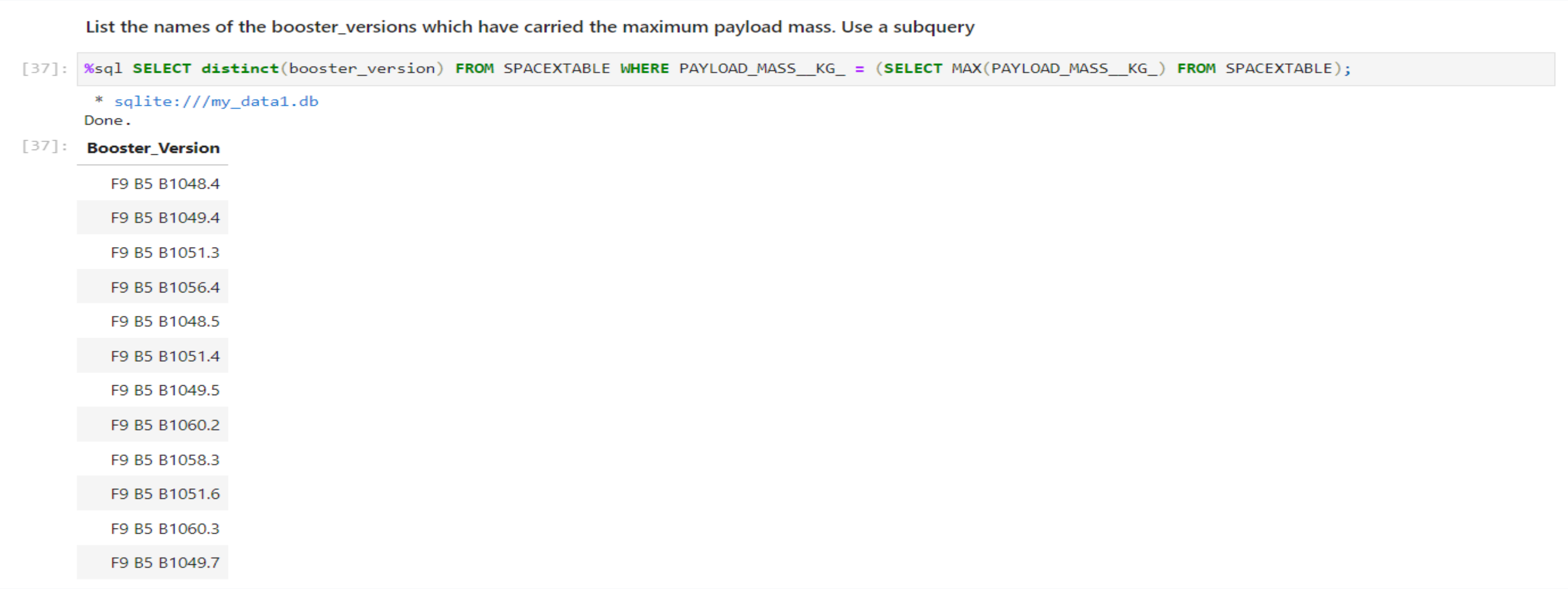
```
 * sqlite:///my_data1.db
Done.
```

[35]:

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

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass



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

```
[37]: %sql SELECT distinct(booster_version) FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

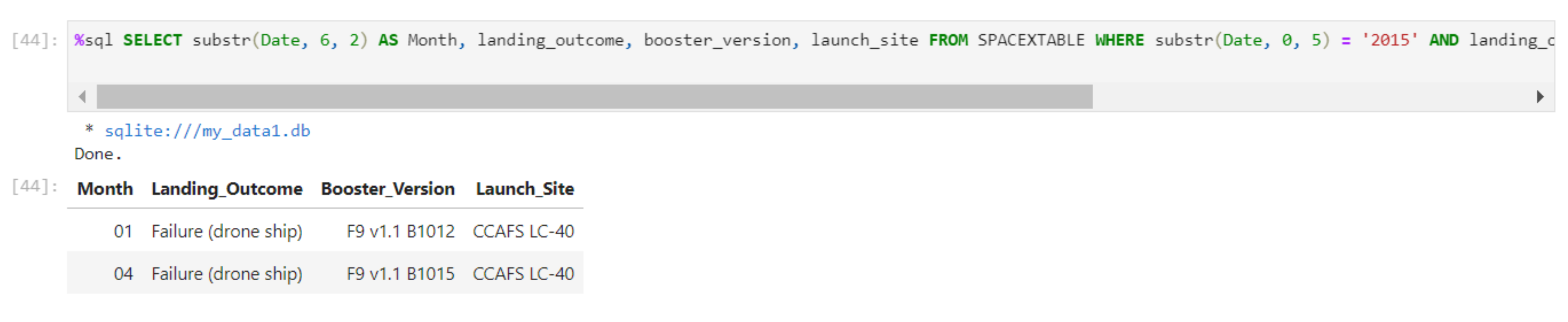
```
 * sqlite:///my_data1.db
Done.
```

[37]:

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



```
[44]: %sql SELECT substr(Date, 6, 2) AS Month, landing_outcome, booster_version, launch_site FROM SPACEXTABLE WHERE substr(Date, 0, 5) = '2015' AND landing_c
```

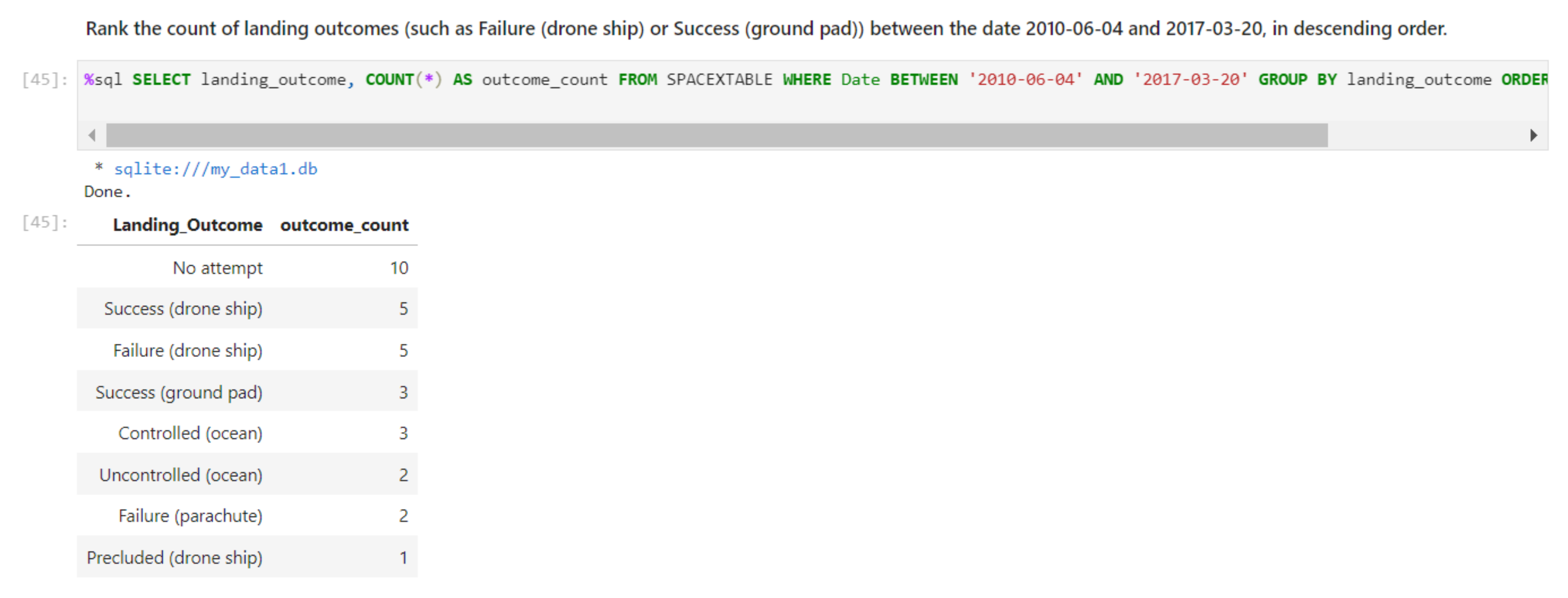
```
 * sqlite:///my_data1.db
Done.
```

[44]:

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.



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.

```
[45]: %sql SELECT landing_outcome, COUNT(*) AS outcome_count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing_outcome ORDER
```

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

[45]:

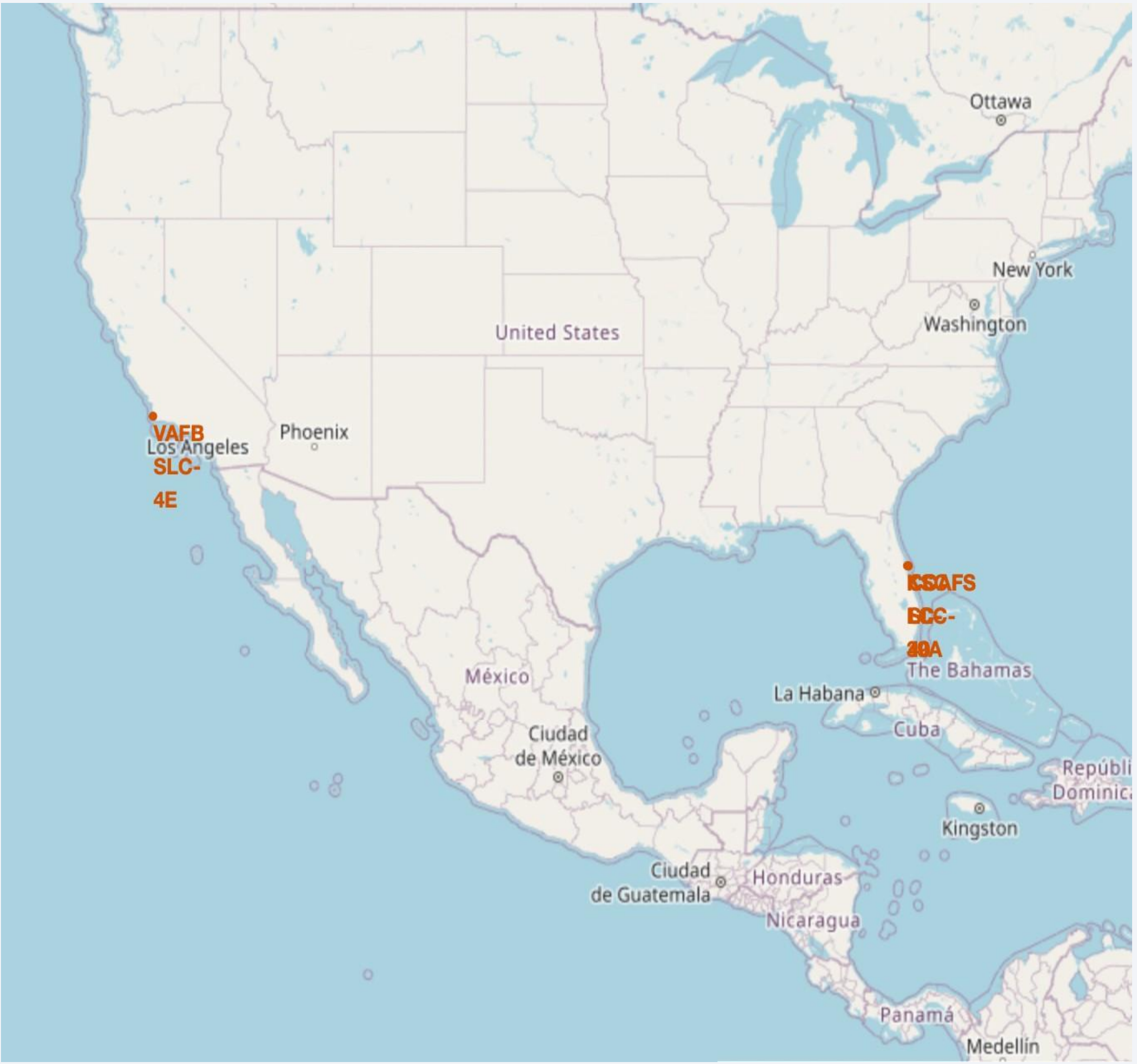
Landing_Outcome	outcome_count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

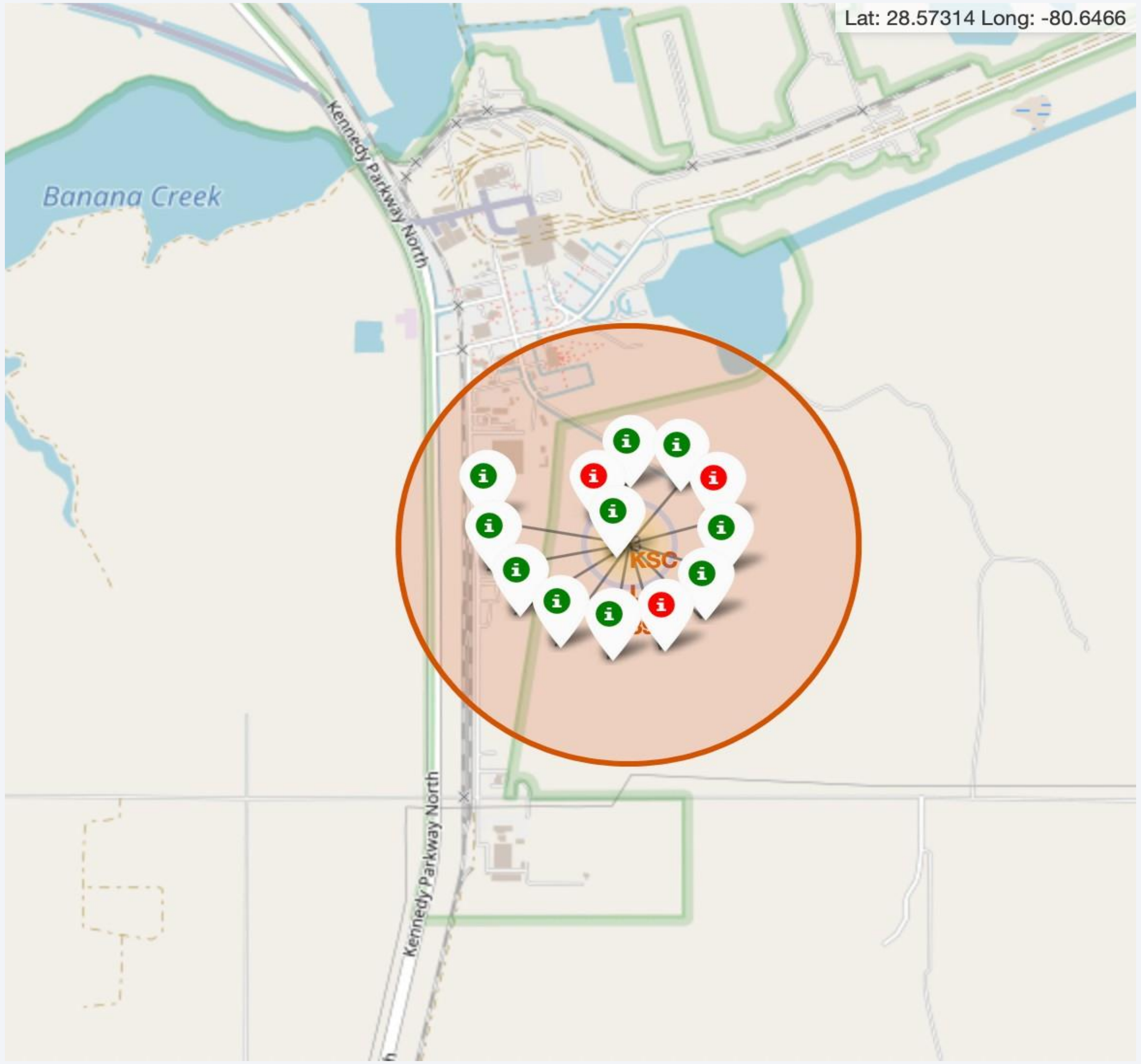
All Launch Sites Locations on Global Map

- All launch sites are in very close proximity to the coast,
- Launching rockets towards the ocean it minimises the risk of having any debris dropping or exploding near people.



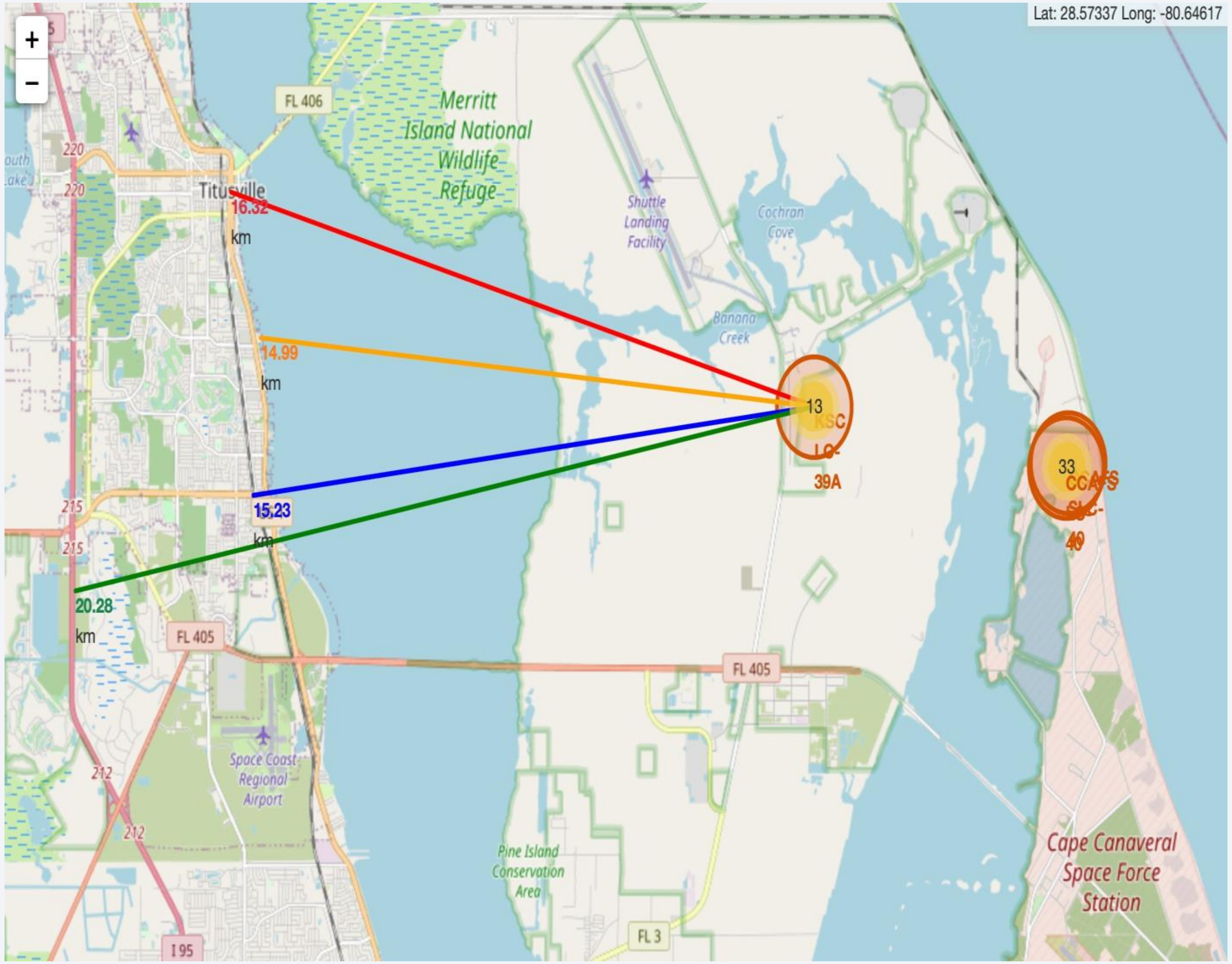
Colour Labeled Launch Records on Map

- Green Marker = Successful Launch
- Red Marker = Failed Launch
- Launch Site KSC LC-39A has a very high Success Rate



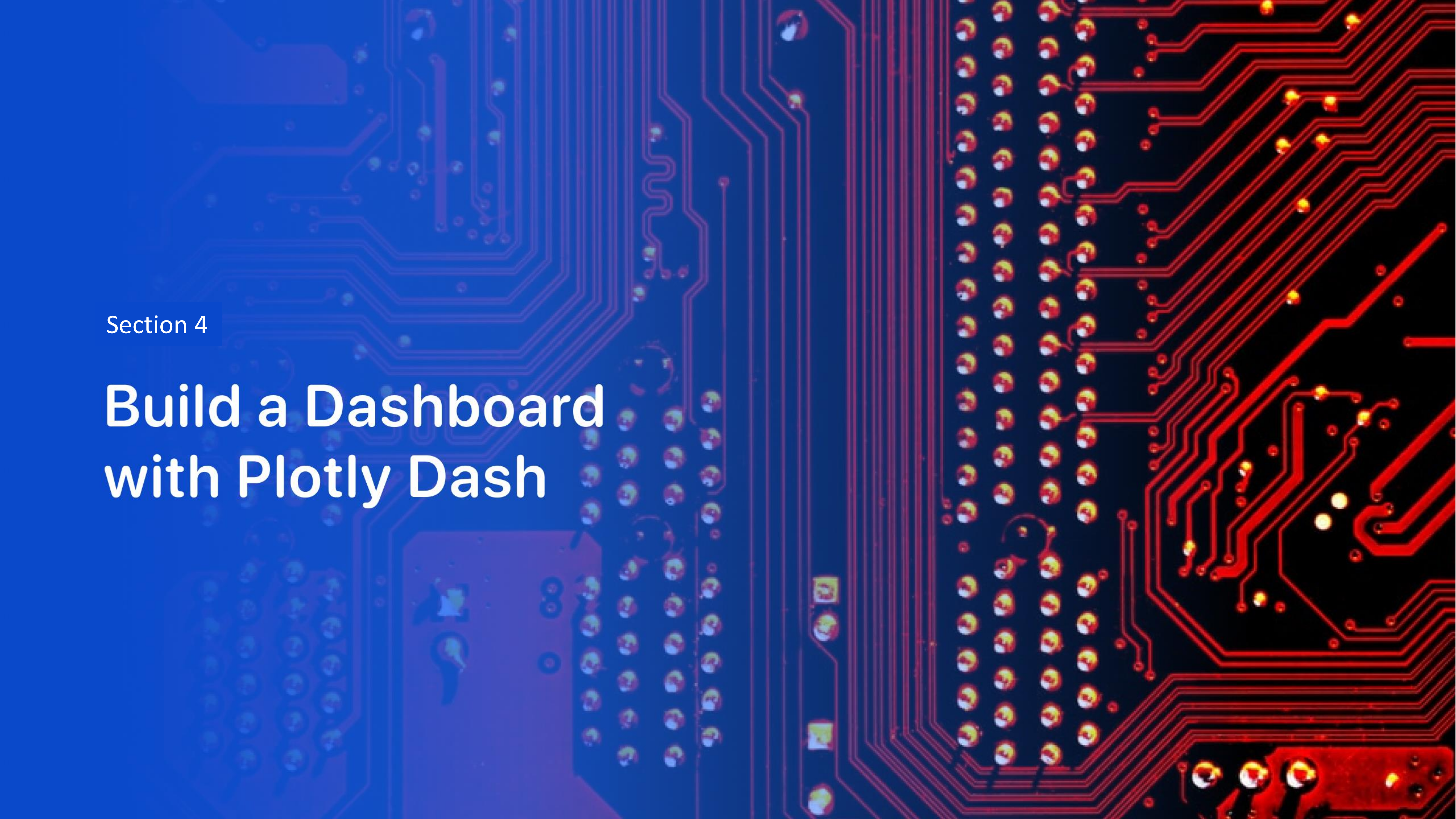
Distance from launch site KSC LC-39A of different locations

- From the visual analysis of the launch site KSC LC-39A we can clearly see that it is relative close to:
  Railway -15.23 km
  Highway -20.28 km
  Coastline -14.99 km
- Also the launch site KSC LC-39A is relative close to its closest city Titusville (16.32 km).
- Failed rocket with its high speed can cover distances like 15-20 km in few seconds. It could be potentially dangerous to populated areas.

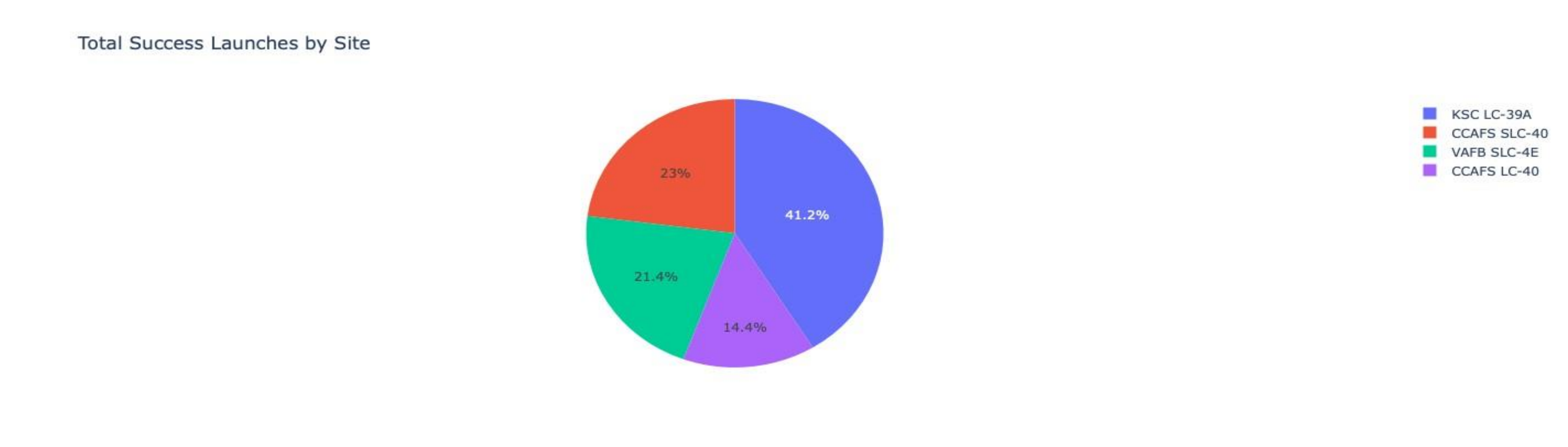

Lat: 28.57337 Long: -80.64617
+
-
FL 406
Merritt Island National Wildlife Refuge
Shuttle Landing Facility
Cochran Cove
Banana Creek
Titusville
16.32 km
14.99 km
15.23 km
20.28 km
13
KSC LC-39A
33
CCAFS SLC-40
FL 405
FL 405
Space Coast Regional Airport
Pine Island Conservation Area
FL 3
I 95
220
215
212
Cape Canaveral Space Force Station

Section 4

Build a Dashboard with Plotly Dash

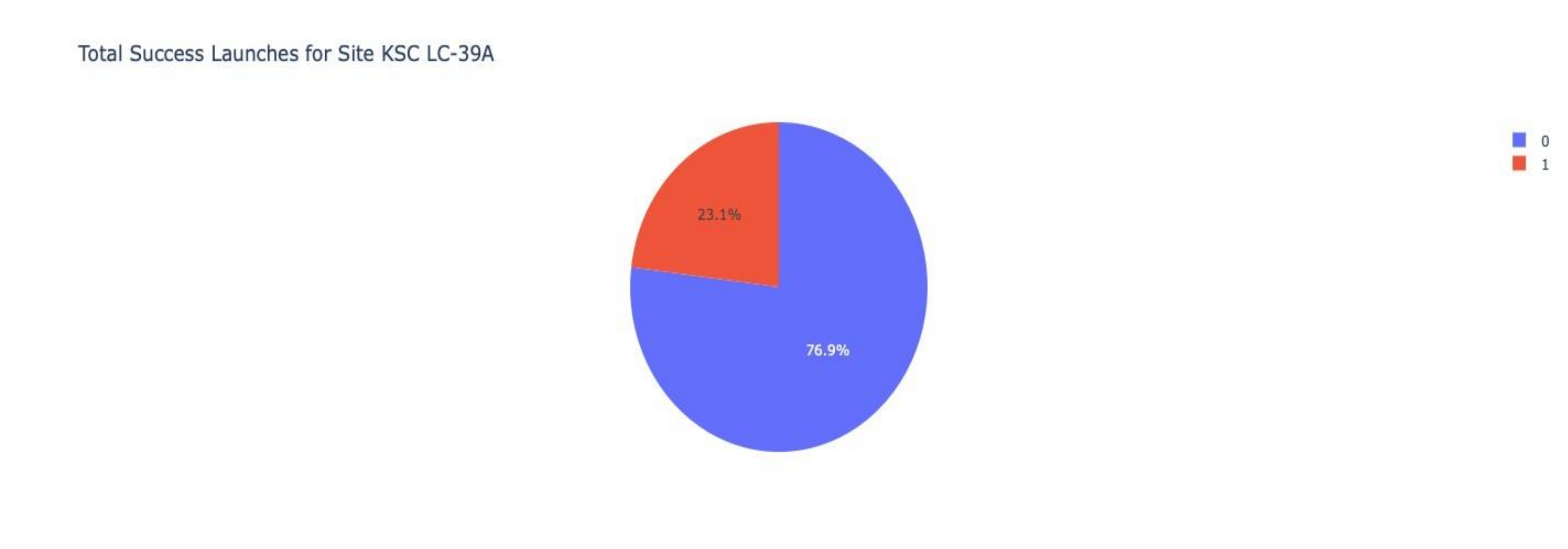


Launch Success for all the Sites



Explanation:
The chart clearly shows that from all the sites, KSC LC-39A has the most successful launches.

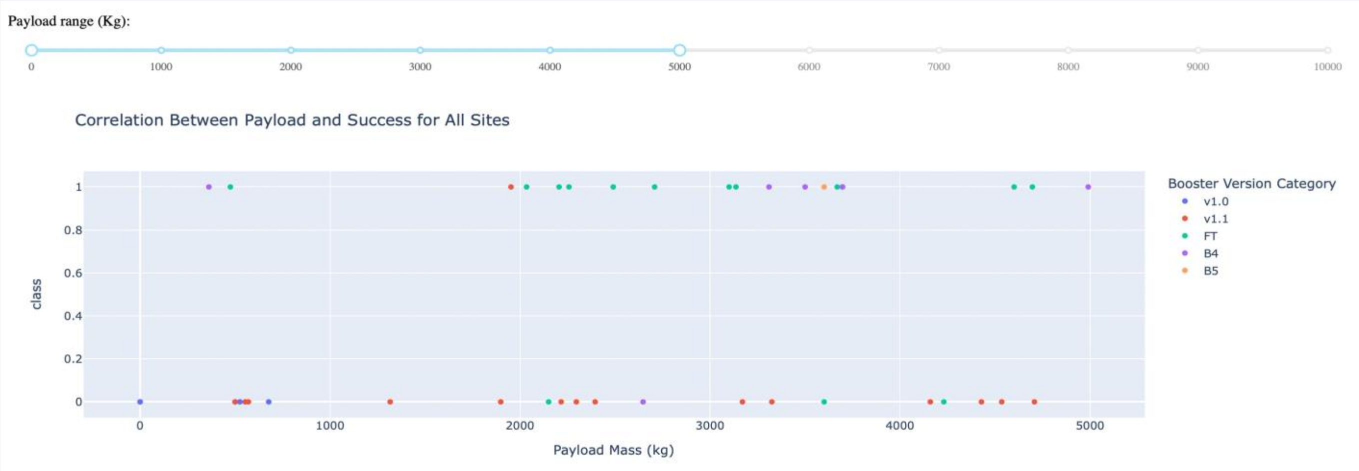
Launch Site with highest launch Success Ratio



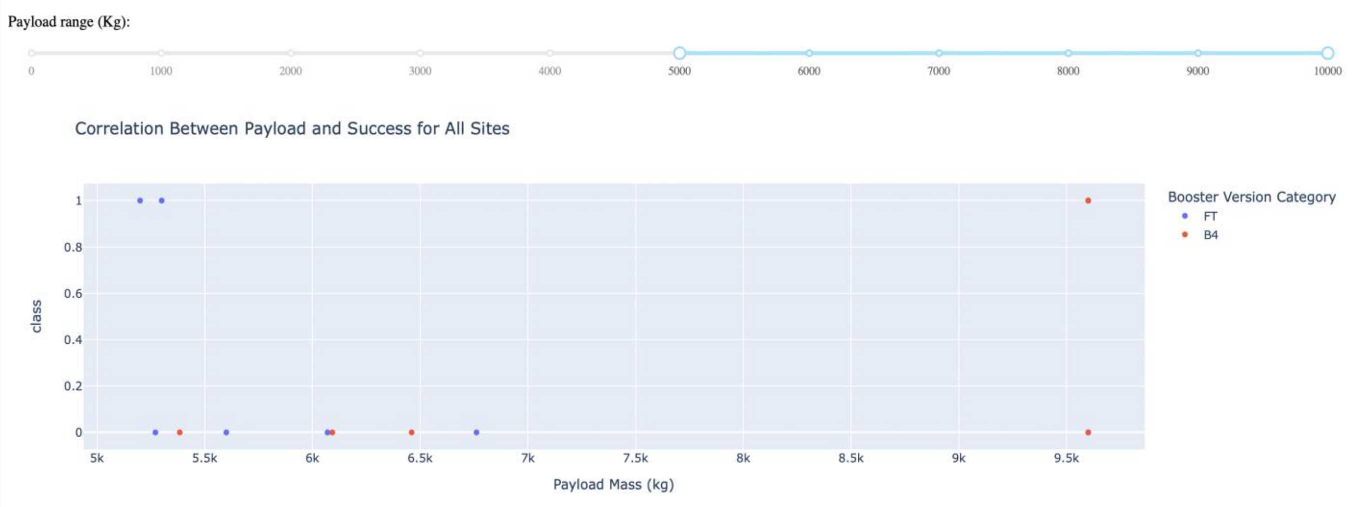
Explanation:
KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and only 3 failed landings.

<Dashboard Screenshot 3>

Explanation:
The charts show that payloads between 2000 and 5500 kg have the highest success rate



Payload range (Kg):



Payload range (Kg):

Section 5

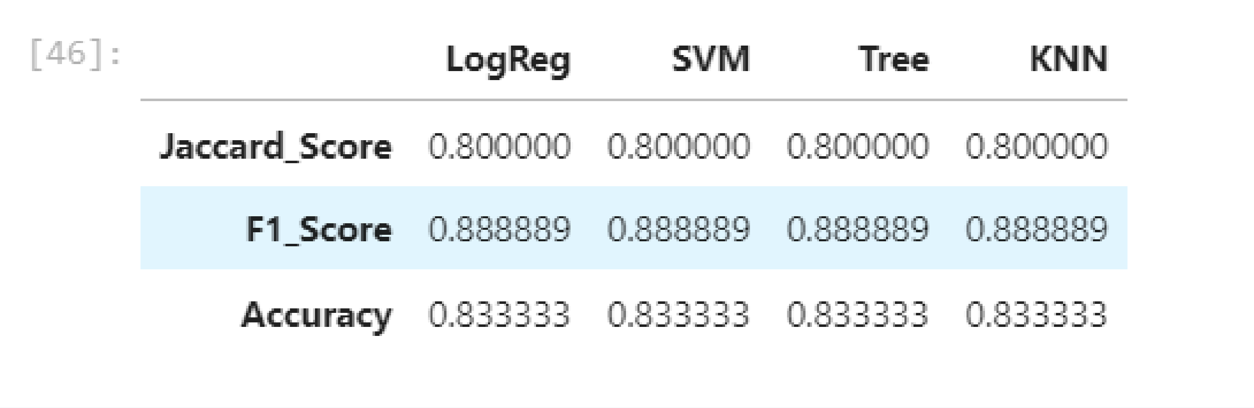
Predictive Analysis (Classification)

Classification Accuracy

Conclusion

1. All the methods are giving same f1 score and accuracy using test data.
2. This is due tosmall test sample size (18 samples).
3. When tested uding whole dataset it is seen that the Decision Tree Model performs better.

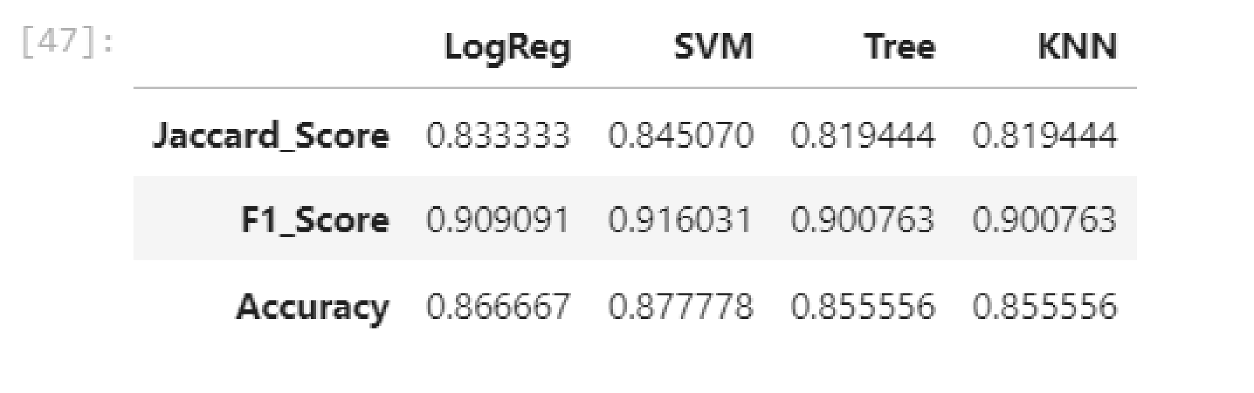
Scores and accuracy on test data set



[46]:

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

Scores and accuracy on whole data set

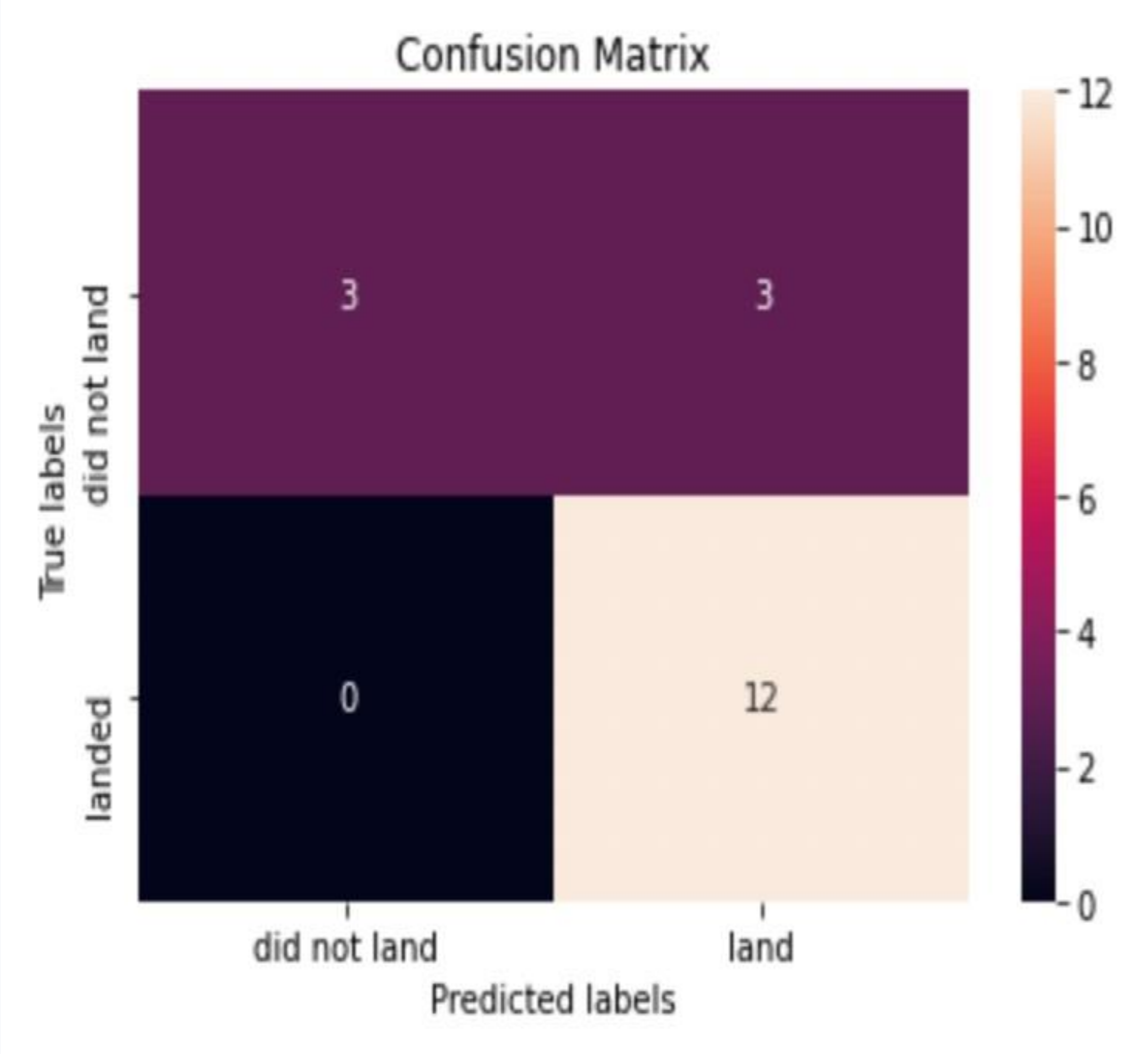


[47]:

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.819444	0.819444
F1_Score	0.909091	0.916031	0.900763	0.900763
Accuracy	0.866667	0.877778	0.855556	0.855556

Confusion Matrix

The confusion matrix analysis reveals logistic regression's ability to differentiate between classes, notably highlighting false positives as the primary challenge.



Conclusions

1.Decision Tree Model: Identified as the most suitable algorithm for this dataset due to its effectiveness.

2.Payload Mass Impact: Launches with lower payload masses demonstrate better outcomes compared to those with larger masses, indicating a significant influence on success rates.

3.Geographic Considerations:

Proximity to Equator: Most launch sites are situated near the Equator, leveraging the Earth's higher rotational speed for efficient launches.

Coastal Locations: All launch sites are positioned very close to the coast, reducing risks associated with debris near populated areas.

4.Success Rate Trends: Over the years, there's an observable increase in the success rates of launches, reflecting advancements in technology and processes.

5.Site and Orbit Performance:

KSC LC-39A: Noted for having the highest success rate among all launch sites, indicating superior operational efficiency.

Orbital Success: Orbits ES-L1, GEO, HEO, and SSO consistently achieve a 100% success rate, demonstrating reliability in achieving mission objectives.

Thank you!