IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ankur Karmacharya
17-10-2025

Outline

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

Executive Summary

- **Summary of methodologies**
  - SpaceX Data Collection using SpaceX API
  - SpaceX Data Collection with Web Scraping
  - SpaceX Data Wrangling
  - SpaceX Exploratory Data Analysis using SQL
  - Space-X EDA DataViz Using Python Pandas and Matplotlib
  - Space-X Launch Sites Analysis with Folium-Interactive Visual Analytics and PlotyDash
  - SpaceX Machine Learning Landing Prediction
- **Summary of all results**
  - EDA results
  - Interactive Visual Analytics and Dashboards
  - Predictive Analysis(Classification

Introduction

- Project background and context

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. Therefore 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.

- Problems you want to find answers

Can we predict if the Falcon 9 first stage will land successfully using data from Falcon 9 rocket launches advertised on its website?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data extraction through SpaceX API
  - Web scraping data through Wikipedia page
- Perform data wrangling
  - Perform EDA to find some patterns
  - Determine what would be the label for training supervised model
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Compare logistic regression model, support vector machine tree decision classifier, KNN by using GridSearchCV to select the best fit model

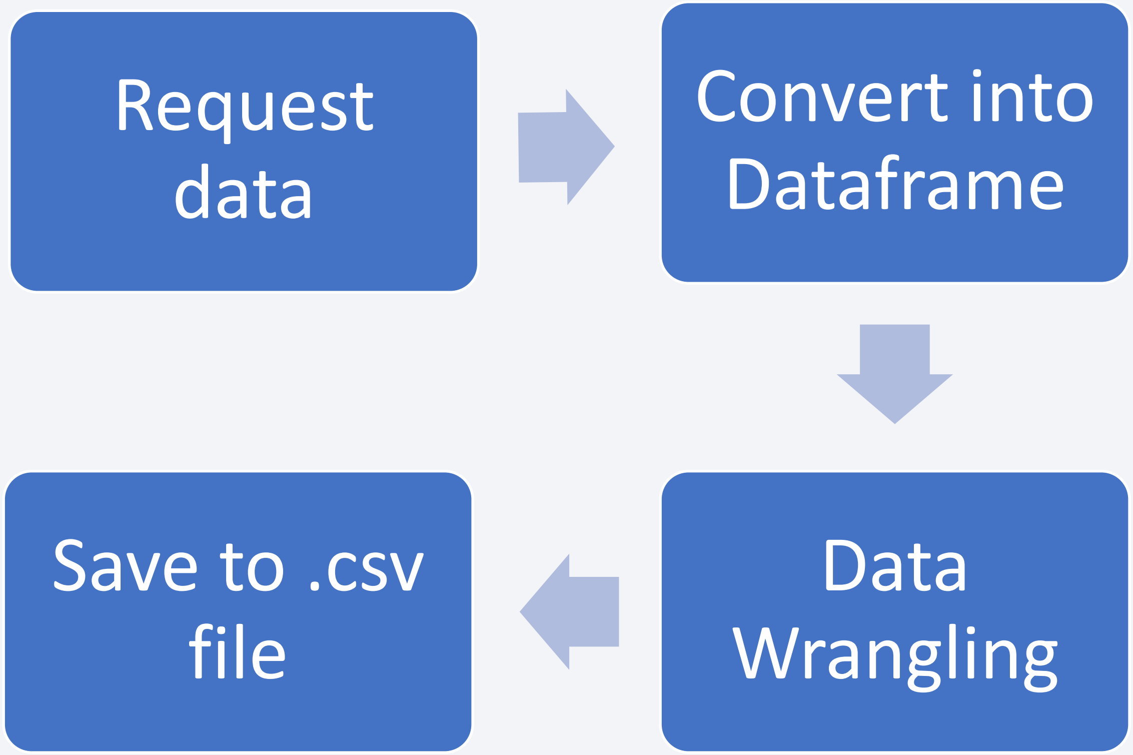
Data Collection

- **Description of how SpaceX Falcon9 data was collected**
  - Data was first collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API. This was done by first defining a series helper functions that would help in the use of the API to extract information using identification numbers in the launch data and then requesting rocket launch data from the SpaceX API URL.
  - Finally to make the requested JSON results more consistent, the SpaceX launch data was requested and parsed using the GET request and then decoded the response content as a Json result which was then converted into a Pandas data frame.
  - Also performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches of the launch records are stored in a HTML. Using BeautifulSoup and request Libraries, I extract the Falcon 9 launch HTML table records from the Wikipedia page, Parsed the table and converted it into a Pandas data frame

Data Collection – SpaceX API

- Request data from SpaceX API
- Convert the json result into a dataframe
- Filter dataframe to only Falcon 9` launches and data wrangling
- Export to csv
- Link:

https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/jupyter-labs-spacex-data-collection-api.ipynb


Request data

Convert into Dataframe

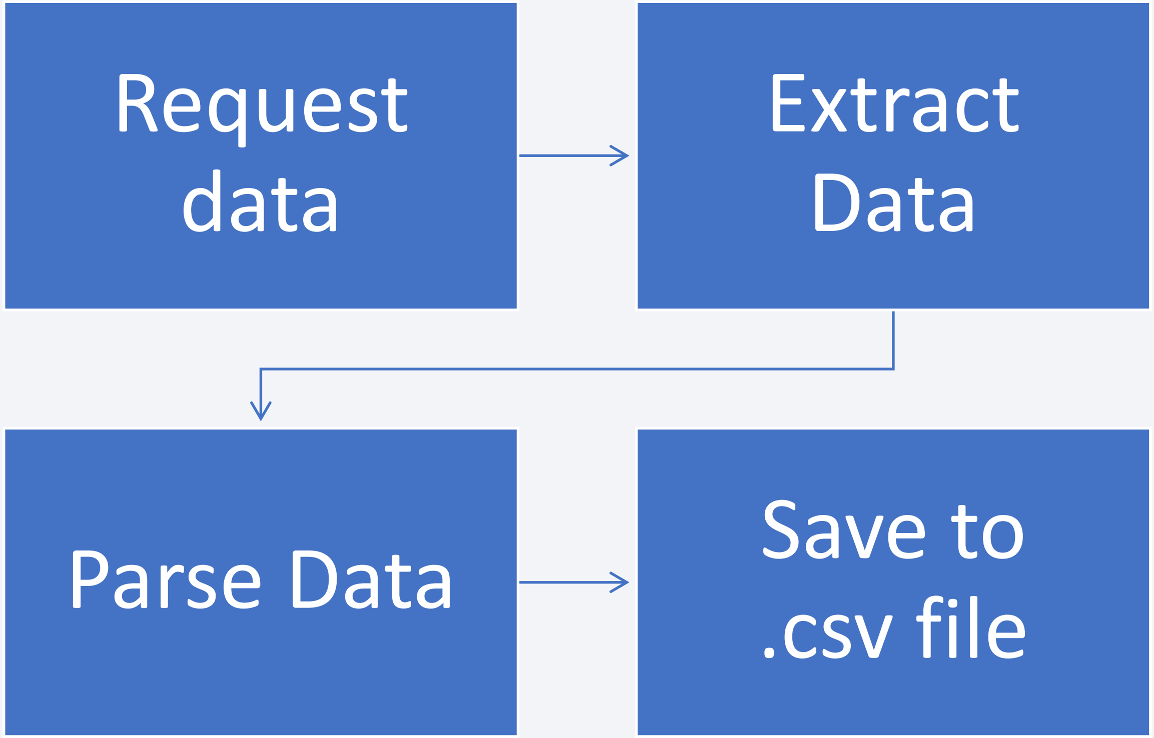
Data Wrangling

Save to .csv file

Data Collection - 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
- Export to csv
- Link :

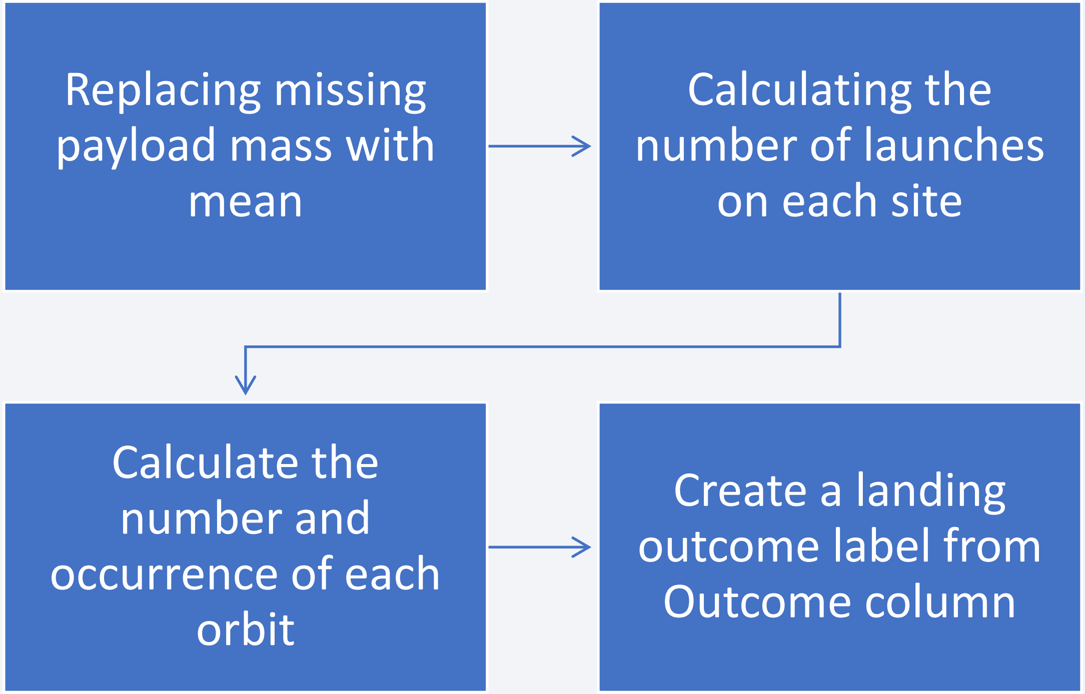
https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/jupyter-labs-webscraping.ipynb



Data Wrangling

- Replaced the missing value of payload mass with the mean of the column
- Used the method .value_counts() to determine the number and occurrence of each orbit in the column Orbit
- Used the method .value_counts() on the column Outcome to determine the number of landing_outcomes. Then assigned it to a variable landing_outcomes.
- Using the Outcome, created a list where the element is zero if the corresponding row in Outcome is in the set bad_outcome; otherwise, it's one. Then assigned it to the variable landing_class.
- Link:

https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb



Replacing missing payload mass with mean

Calculating the number of launches on each site

Calculate the number and occurrence of each orbit

Create a landing outcome label from Outcome column

EDA with Data Visualization

- The relationship between Flight Number and Launch Site -> scatter plot
- The relationship between Payload and Launch Site -> scatter plot
- The relationship between success rate of each orbit type -> bar plot
- The relationship between Flight Number and Orbit type -> scatter plot
- The relationship between Payload and Orbit type -> scatter plot
- The launch success yearly trend -> line chart

❑ The scatterplot is used best to describe the relationship between two categorical data.

❑ The bar plot is used best to compare several categorical data.

❑ The line plot is used best to show the time series data.

- **Link:**

https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/jupyter-labs-eda-dataviz-v2.ipynb

EDA with 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 successful landing outcome in ground pad was achieved.
- 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.
- 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 successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
- **Link :**

https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Mark all launch sites on a map
- Mark the success/failed launches for each site on the map
- Calculate the distances between a launch site to its proximities
  - ➢ Whether it is close to the coast
  - ➢ Whether it is close to the railway
  - ➢ Whether it is close to the highway
  - ➢ Whether it is close to the city

- The markers where created to gives geographical information about the launch site like which site near to the coast, railway, highway or city etc.
- Link:

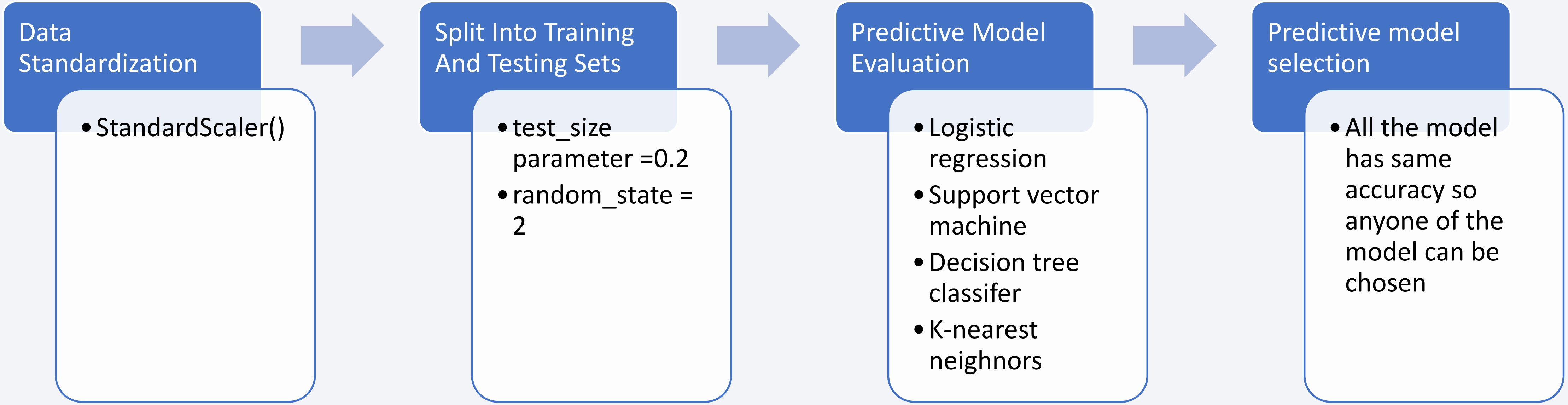
  https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Built an interactive dashboard application with Plotlydash by:
  - Adding a Launch Site Drop-down Input Component
  - Adding a callback function to render success-pie-chart based on selected site dropdown
  - Adding a Range Slider to Select Payload
  - Adding a callback function to render the success-payload-scatter-chart scatter plot

- Link:
  https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/analytics%20with%20plotly.py

Predictive Analysis (Classification)



Data Standardization

- StandardScaler()

Split Into Training And Testing Sets

- test_size parameter =0.2
- random_state = 2

Predictive Model Evaluation

- Logistic regression
- Support vector machine
- Decision tree classifer
- K-nearest neighnors

Predictive model selection

- All the model has same accuracy so anyone of the model can be chosen

- The data was first standardized using StandardScaler()
- The data was split into training and testing sets with test_size parameter =0.2 random_state = 2

Predictive Analysis (Classification)

- In order to find the best ML model that would performs best using the test data between SVM, Classification Trees, k nearest neighbors and Logistic Regression;
  - An object was created for each of the algorithms then created a GridSearchCV object and assigned them a set of parameters for each model.
  - For each of the models under evaluation, the GridsearchCV object was created with cv=10, then fit the training data into the GridSearch object for each to Find best Hyperparameter.
  - After fitting the training set, we output GridSearchCV object for each of the models, then displayed the best parameters using the data attribute best_params and the accuracy on the validation data using the data attribute best_score.
  - Finally using the method score to calculate the accuracy on the test data for each model and plotted a confusion matrix for each using the test and predicted outcomes
- Link:

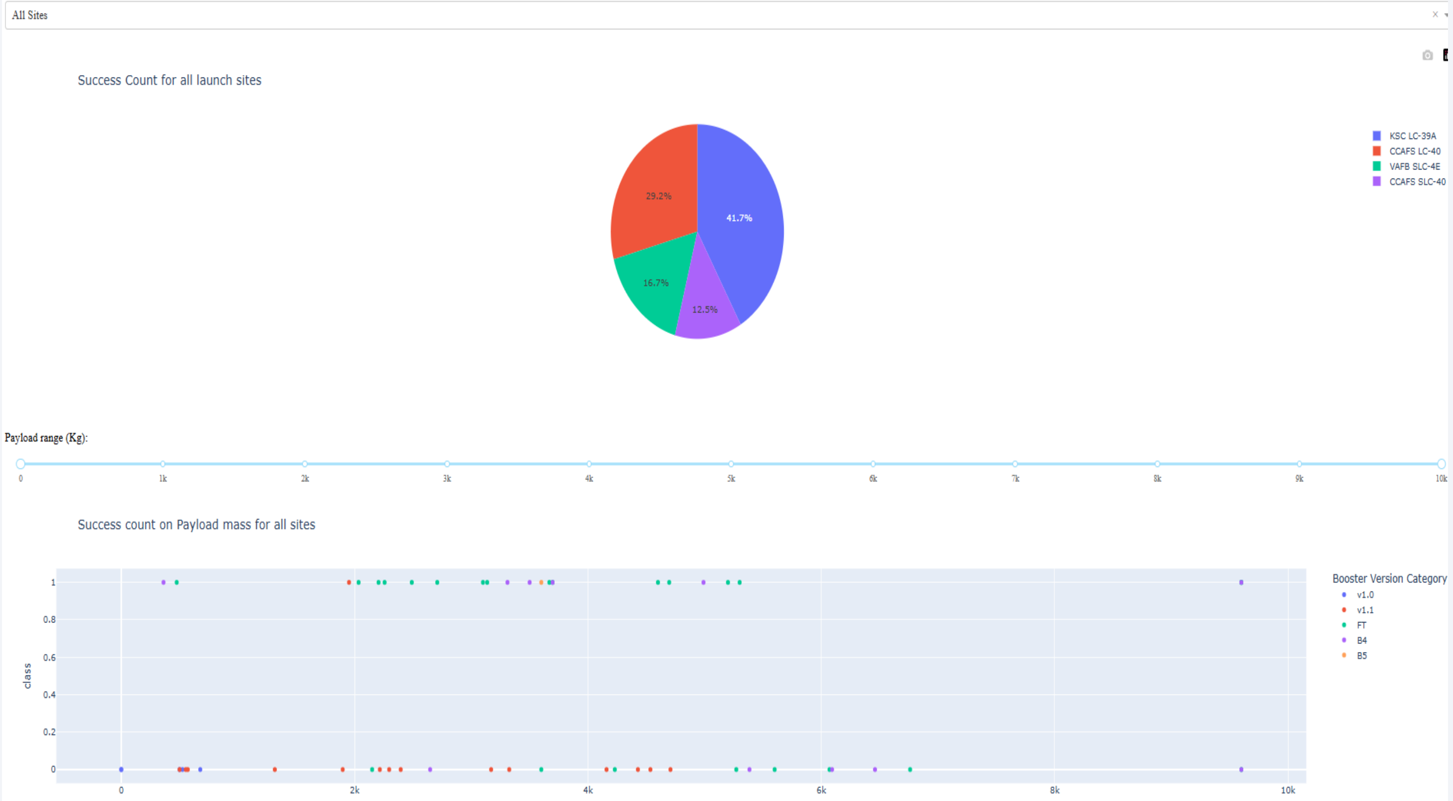
  https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

EDA

- KSC LC-39AandVAFB SLC 4E has a success rate of 77%
- •VAFB SLC 4E has no payload above 10000 kg
- •In the LEO orbit the Success appears related to the number of flights
- •With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS
- •The success rate since 2013 kept increasing till 2020

Interactive analytics



All Sites

Payload range (Kg):

Results

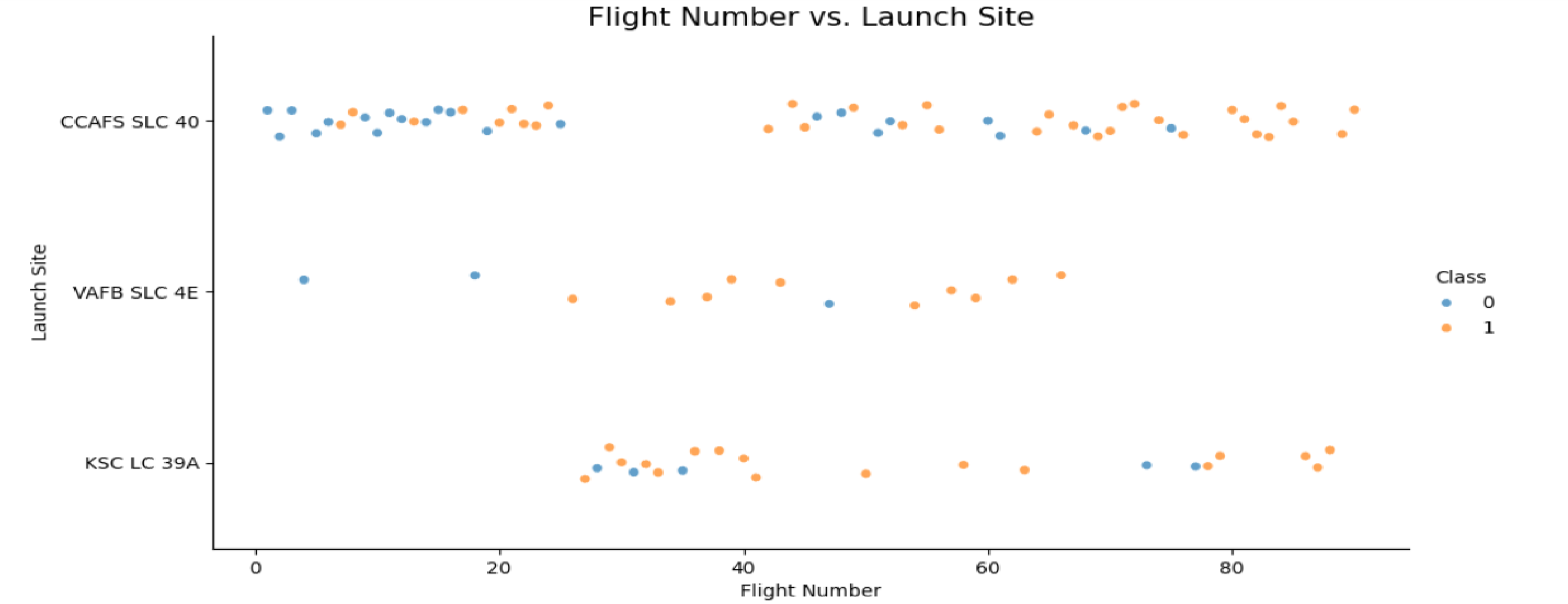
Predictive analysis

- The accuracy of all model are same i.e. 83.33% so anyone of the model can be used for classification.
- The confusion matrix shows us 3 true positive , 0 false positive , 3 false negative and 12 true negative.

Section 2

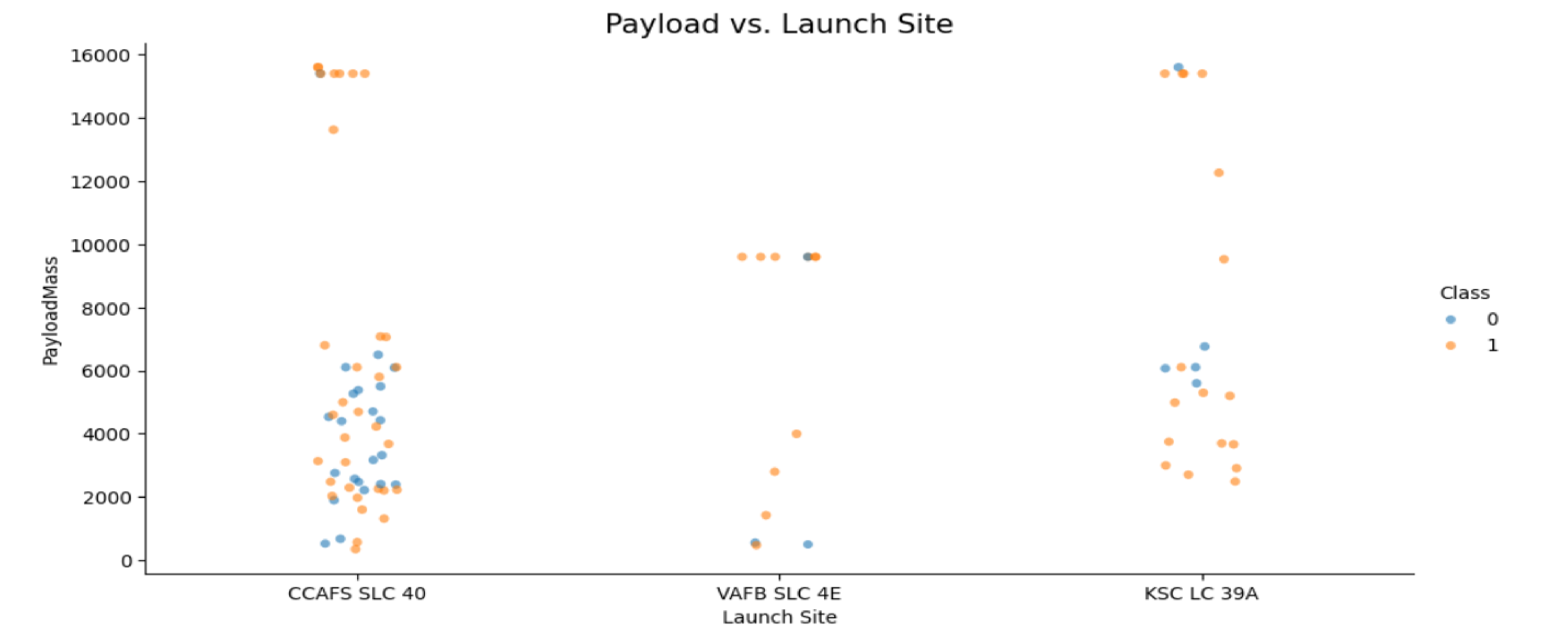
Insights drawn from EDA

Flight Number vs. Launch Site



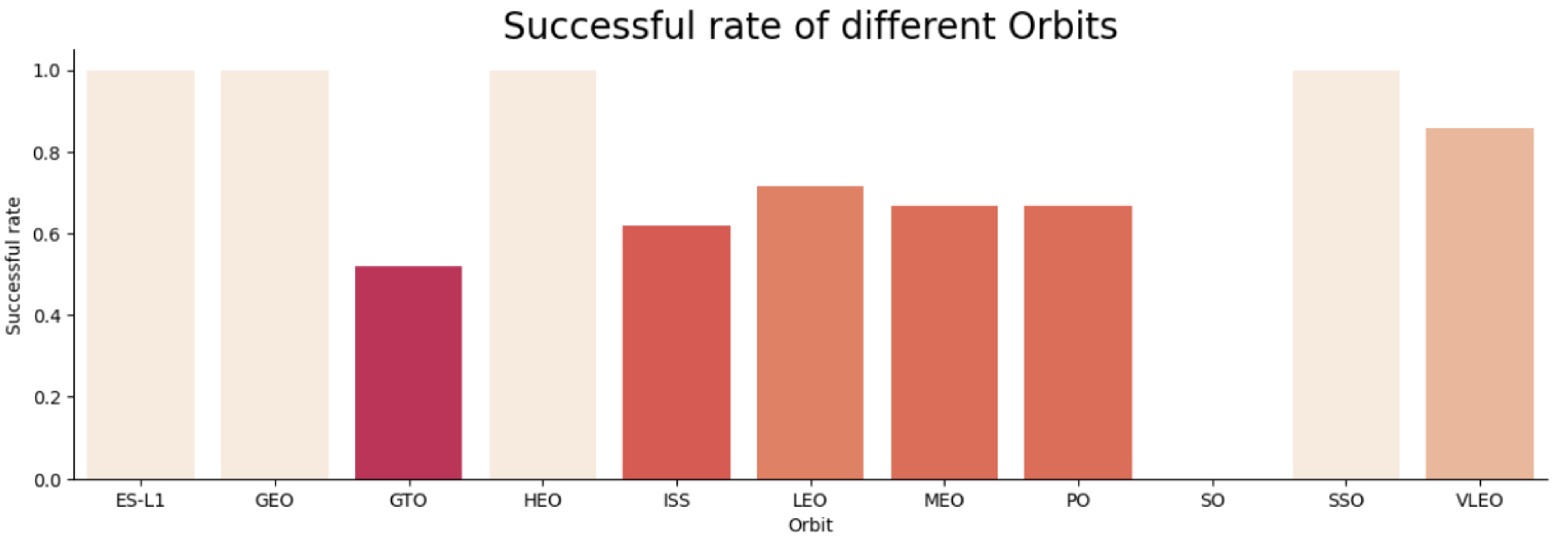
KSC LC 39A has the highest success rate where as CCAFS SLC 40 has the lowest success rate

Payload vs. Launch Site



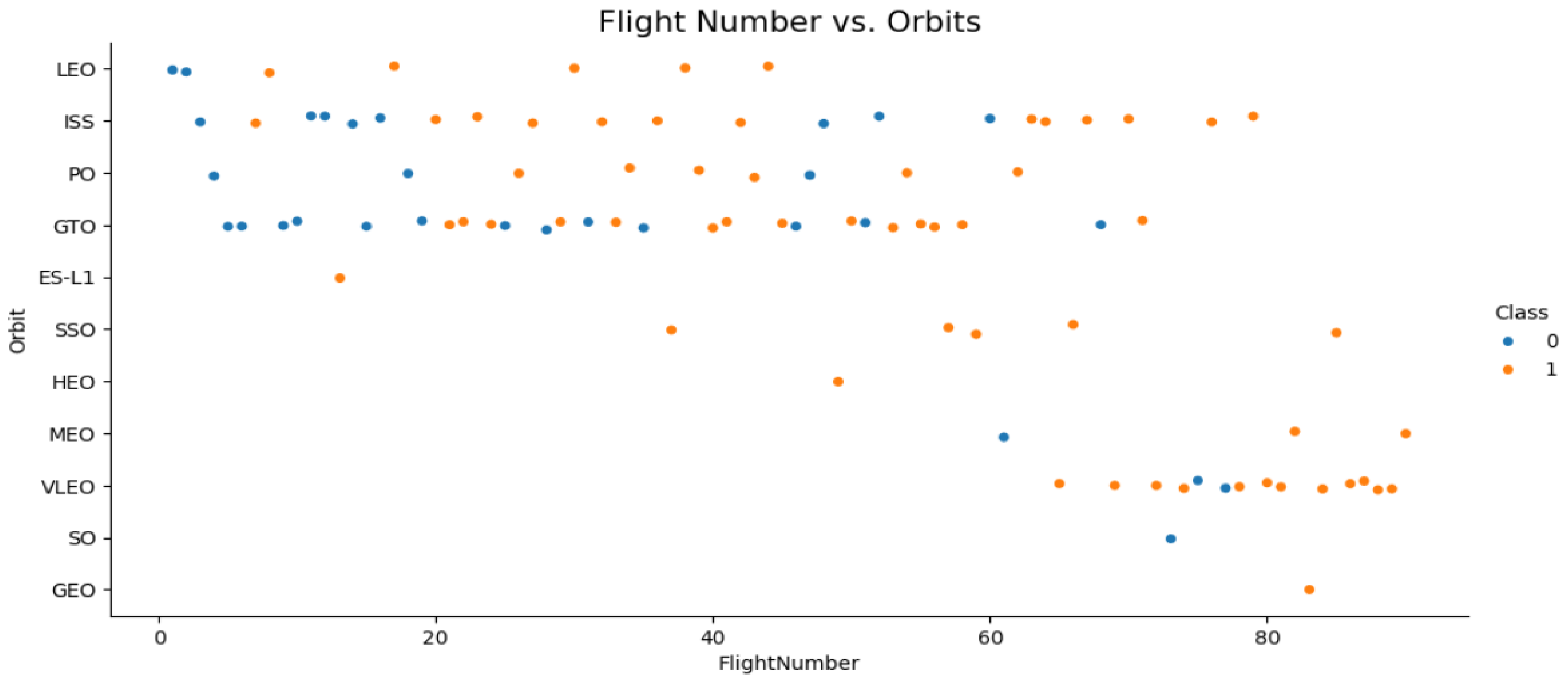
VAFB-SLC 4E has no payload above 10000 kg

Success Rate vs. Orbit Type



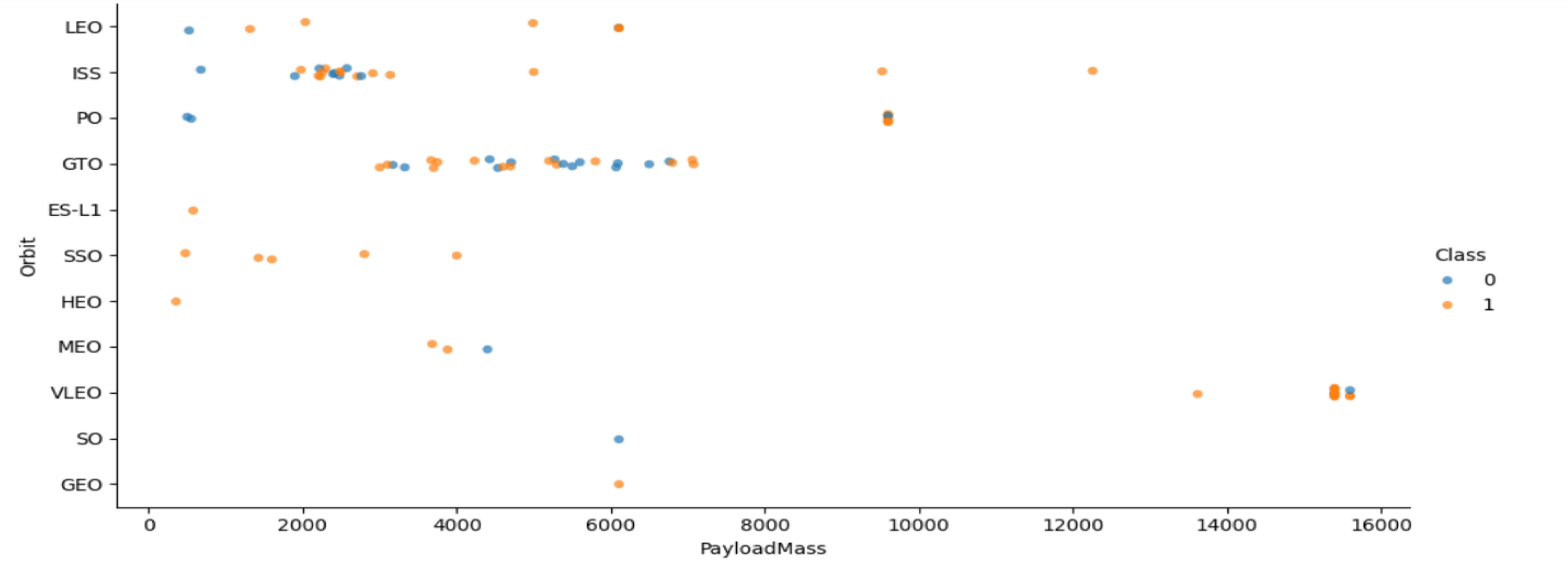
ES L1, GEO, HEO, SSO have the highest success rate

Flight Number vs. Orbit Type



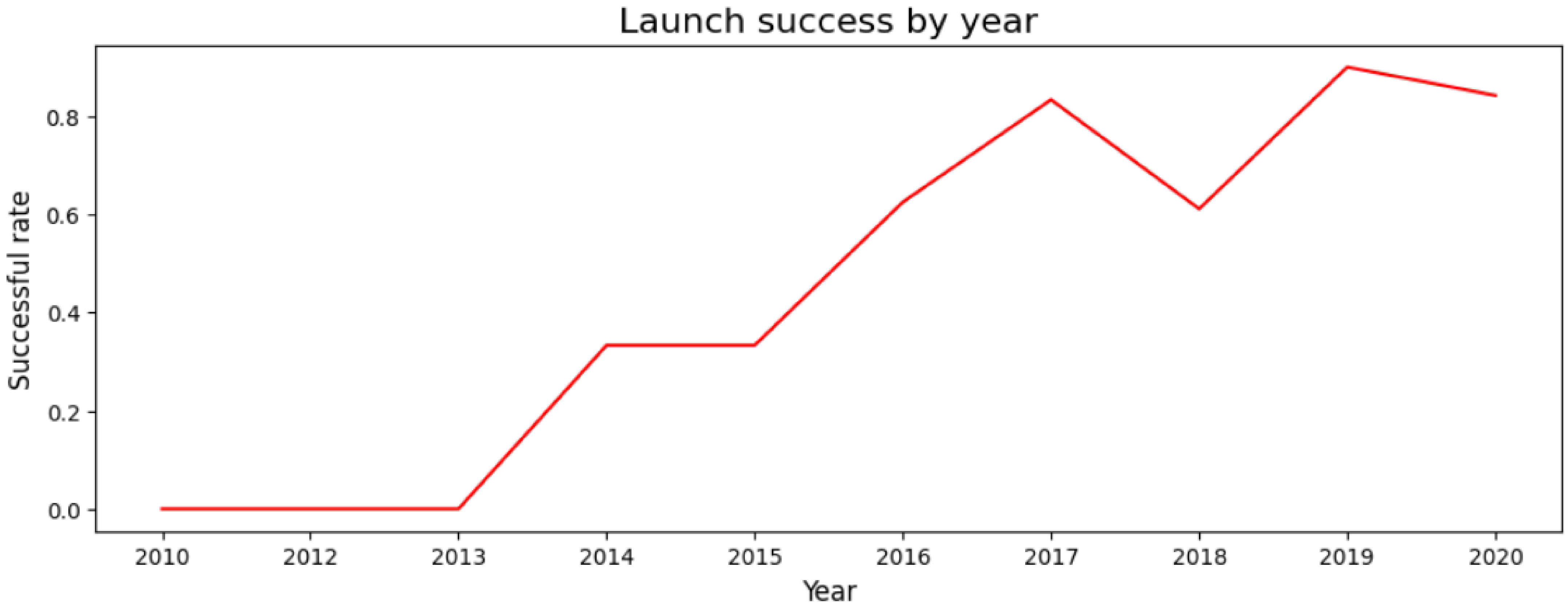
In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



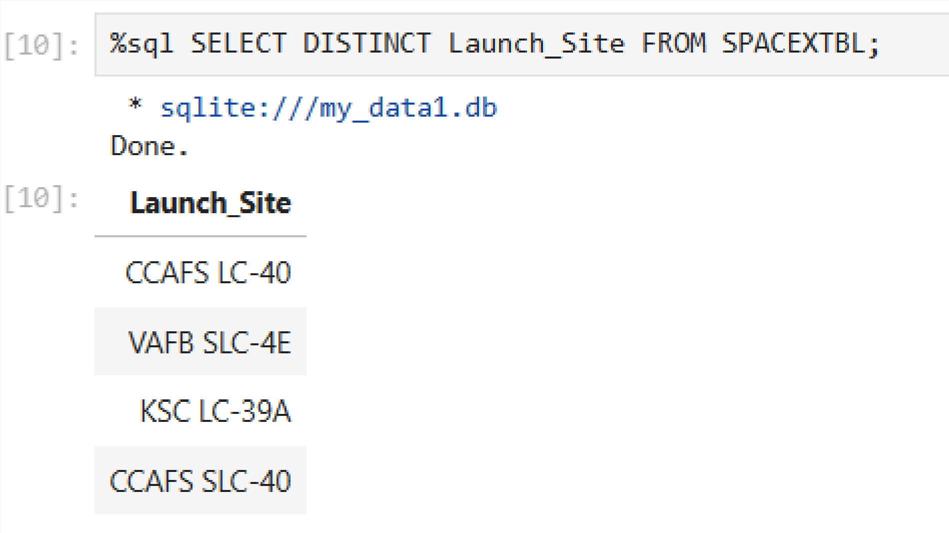
With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



The success rate started increasing from 2013 until 2020

All Launch Site Names



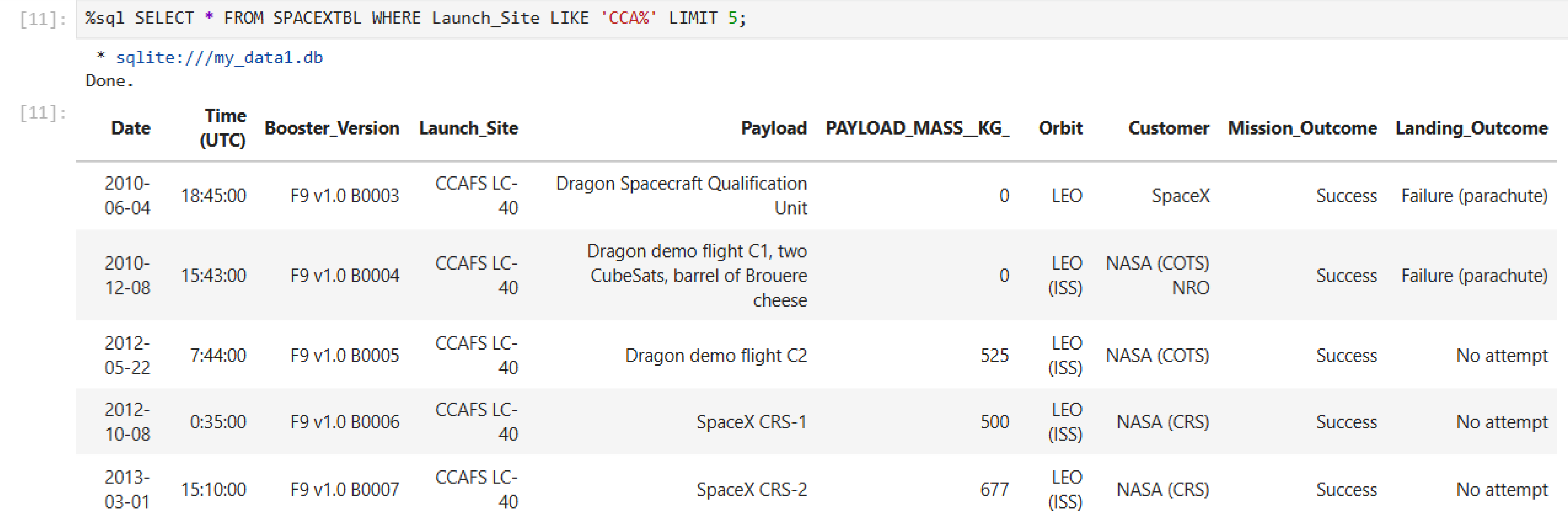
```
[10]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;

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

[10]:

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

- There are total 4 different launch sites

Launch Site Names Begin with 'CCA'



```
[11]: %sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

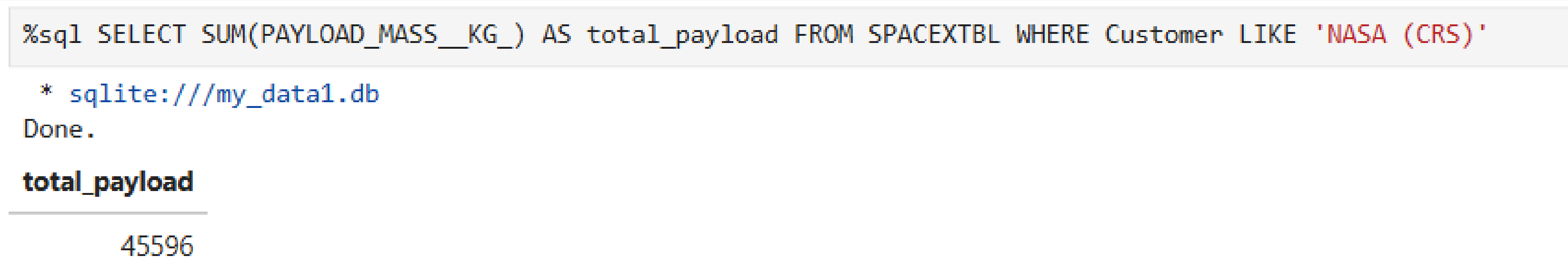
```
 * sqlite:///my_data1.db
Done.
```

[11]:

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

- All landing outcome are failures.

Total Payload Mass

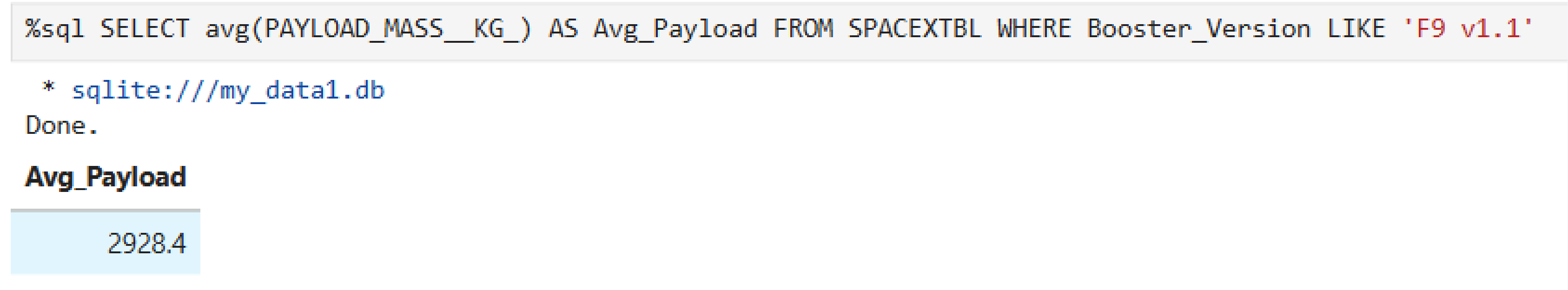


```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
total_payload
        45596
```

The total payload mass for NASA is 45,596 kg

Average Payload Mass by F9 v1.1



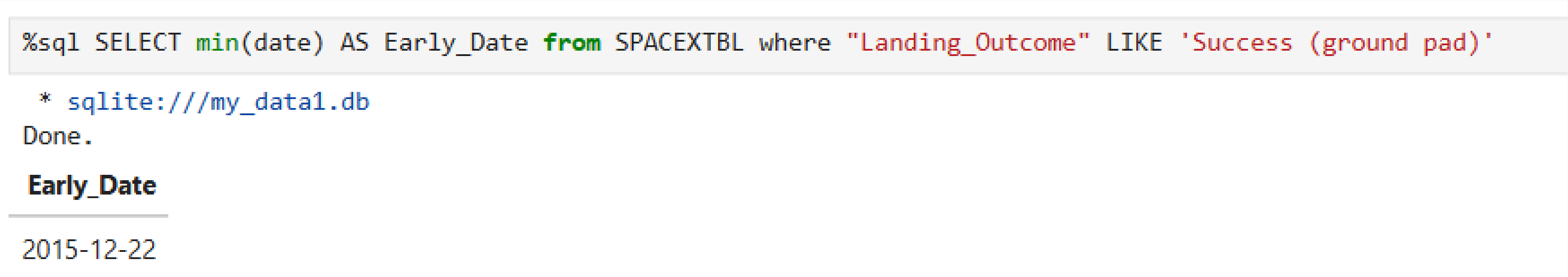
```
%sql SELECT avg(PAYLOAD_MASS__KG_) AS Avg_Payload FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1'
```

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

Avg_Payload
2928.4

- The average payload mass carried by booster version F9 v1.1 is 2,928.40 kg

First Successful Ground Landing Date



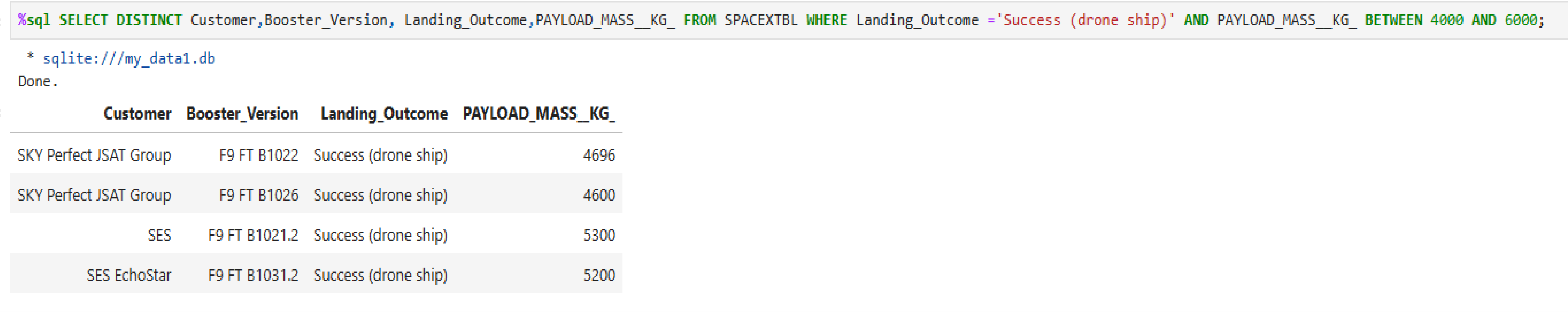
```
%sql SELECT min(date) AS Early_Date from SPACEXTBL where "Landing_Outcome" LIKE 'Success (ground pad)'
```

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

Early_Date
2015-12-22

- The first ground landing successful is on 22.12.2015

Successful Drone Ship Landing with Payload between 4000 and 6000



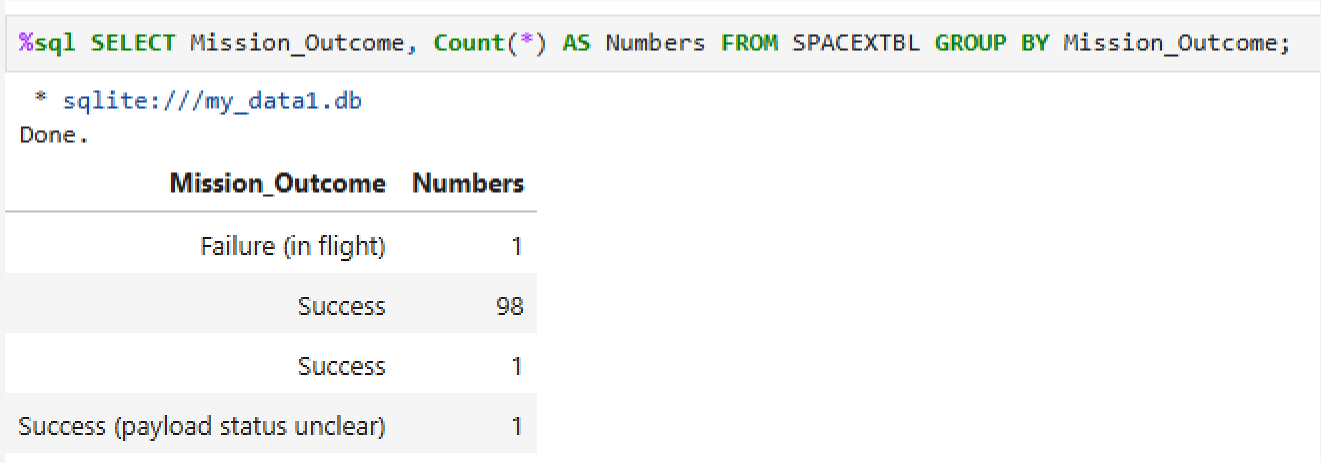
```
%sql SELECT DISTINCT Customer,Booster_Version, Landing_Outcome,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE Landing_Outcome ='Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

Customer	Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
SKY Perfect JSAT Group	F9 FT B1022	Success (drone ship)	4696
SKY Perfect JSAT Group	F9 FT B1026	Success (drone ship)	4600
SES	F9 FT B1021.2	Success (drone ship)	5300
SES EchoStar	F9 FT B1031.2	Success (drone ship)	5200

- There are 4 booster version which have success in drone ship and have payload mass greater than 4000 but less than 6000 .

Total Number of Successful and Failure Mission Outcomes



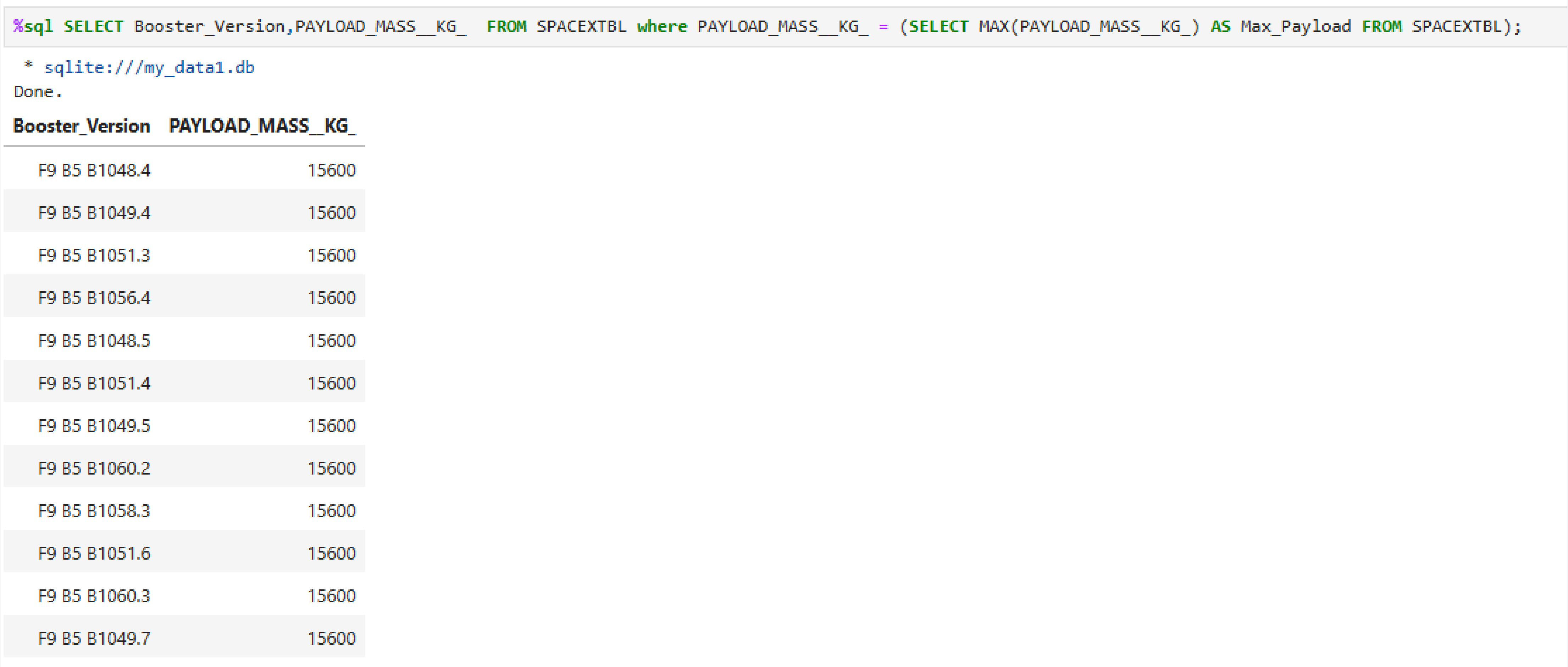
```
%sql SELECT Mission_Outcome, Count(*) AS Numbers FROM SPACEXTBL GROUP BY Mission_Outcome;
```

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

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

- There are 1 failure in flight, 99 successes and 1 success with unclear payload status.

Boosters Carried Maximum Payload



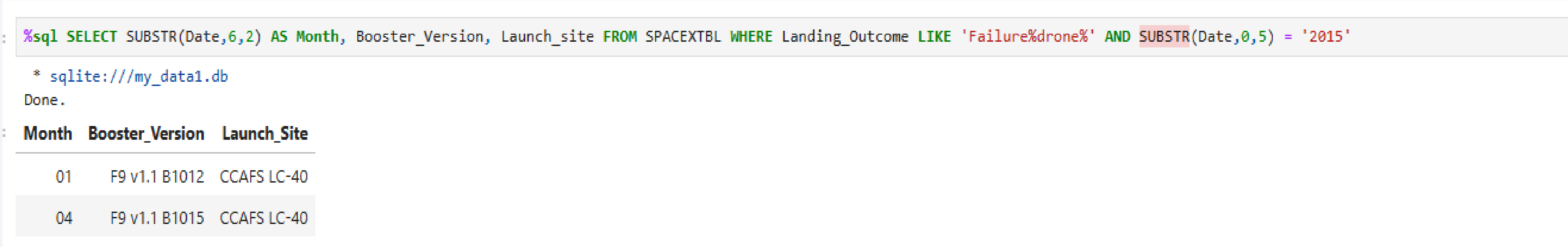
```
%sql SELECT Booster_Version,PAYLOAD_MASS__KG_  FROM SPACEXTBL where PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) AS Max_Payload FROM SPACEXTBL);
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

- There are 12 booster that carried maximum payload mass 15,600 kg

2015 Launch Records



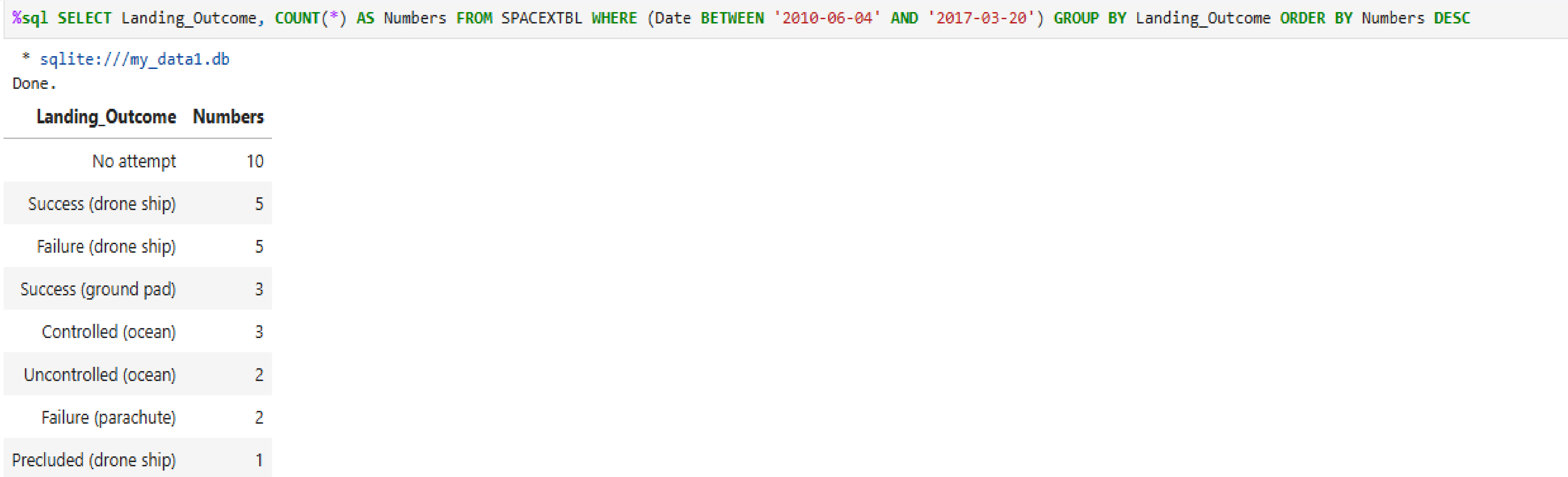
```
%sql SELECT SUBSTR(Date,6,2) AS Month, Booster_Version, Launch_site FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Failure%drone%' AND SUBSTR(Date,0,5) = '2015'
```

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

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

- In January and April, 2015 there are launch failure by booster B1012 and B1015.

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



```
%sql SELECT Landing_Outcome, COUNT(*) AS Numbers FROM SPACEXTBL WHERE (Date BETWEEN '2010-06-04' AND '2017-03-20') GROUP BY Landing_Outcome ORDER BY Numbers DESC
```

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

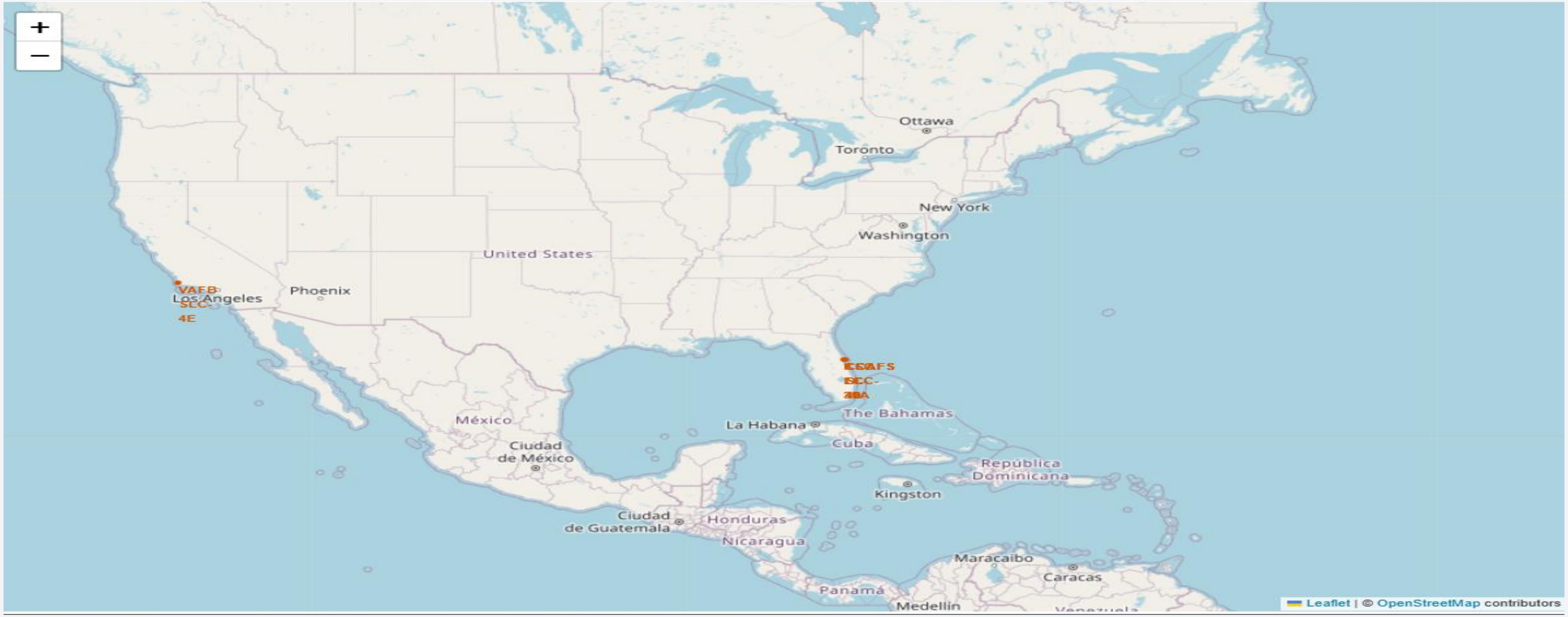
Landing_Outcome	Numbers
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- Between 04-06-2010 and 20-03-2017, there were 10 no attempt, 5 Success (drone ship), 5 Failure (drone ship), 3 Success (ground pad), 3 Controlled (ocean), 2 Uncontrolled (ocean), 2 Failure (parachute) and 1 Precluded (drone ship).

Section 3

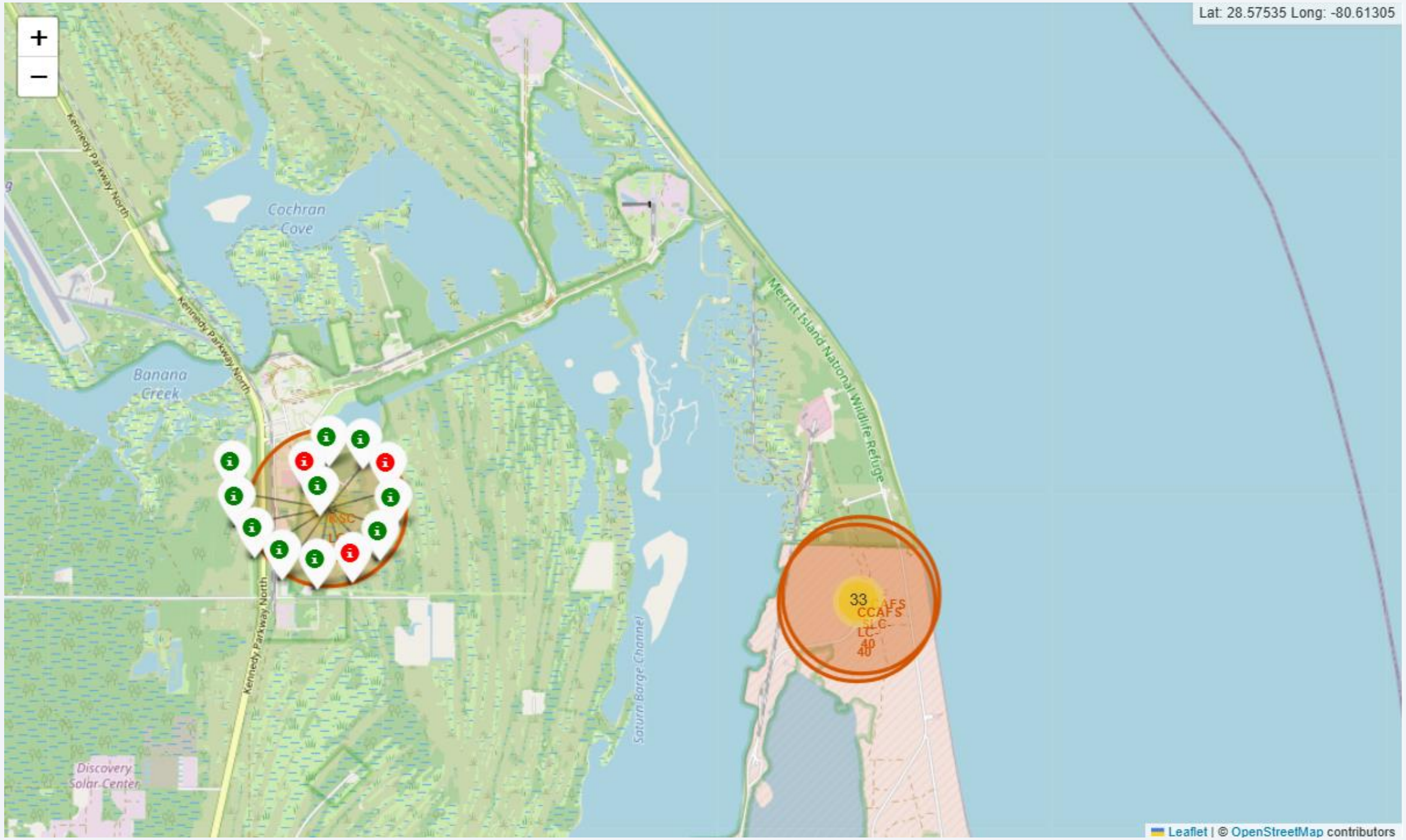
Launch Sites Proximities Analysis

Markers of all launch sites on global map



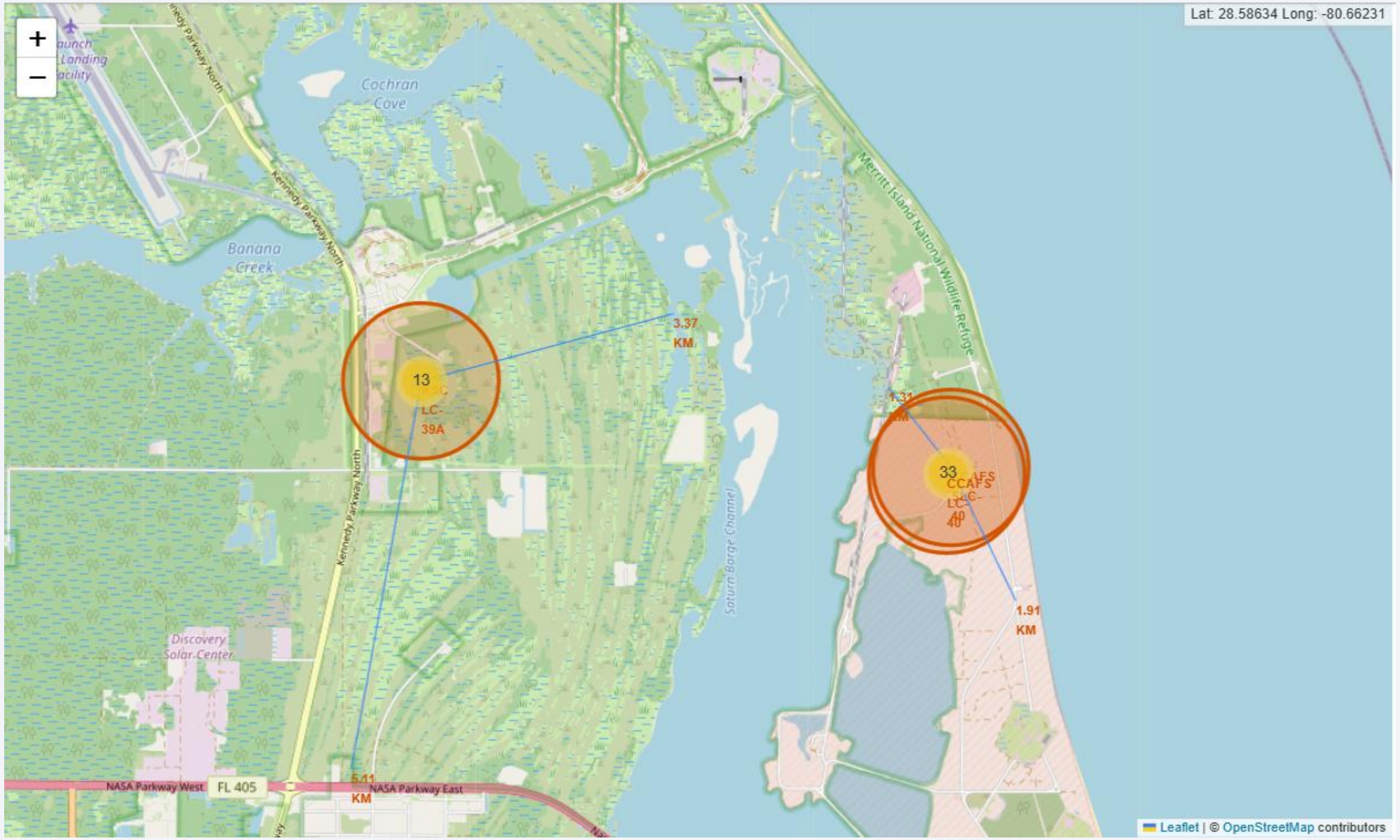
- All launch sites are in proximity to the Equator line.
- All launch sites are very close to the coast.

Launch outcome of different site

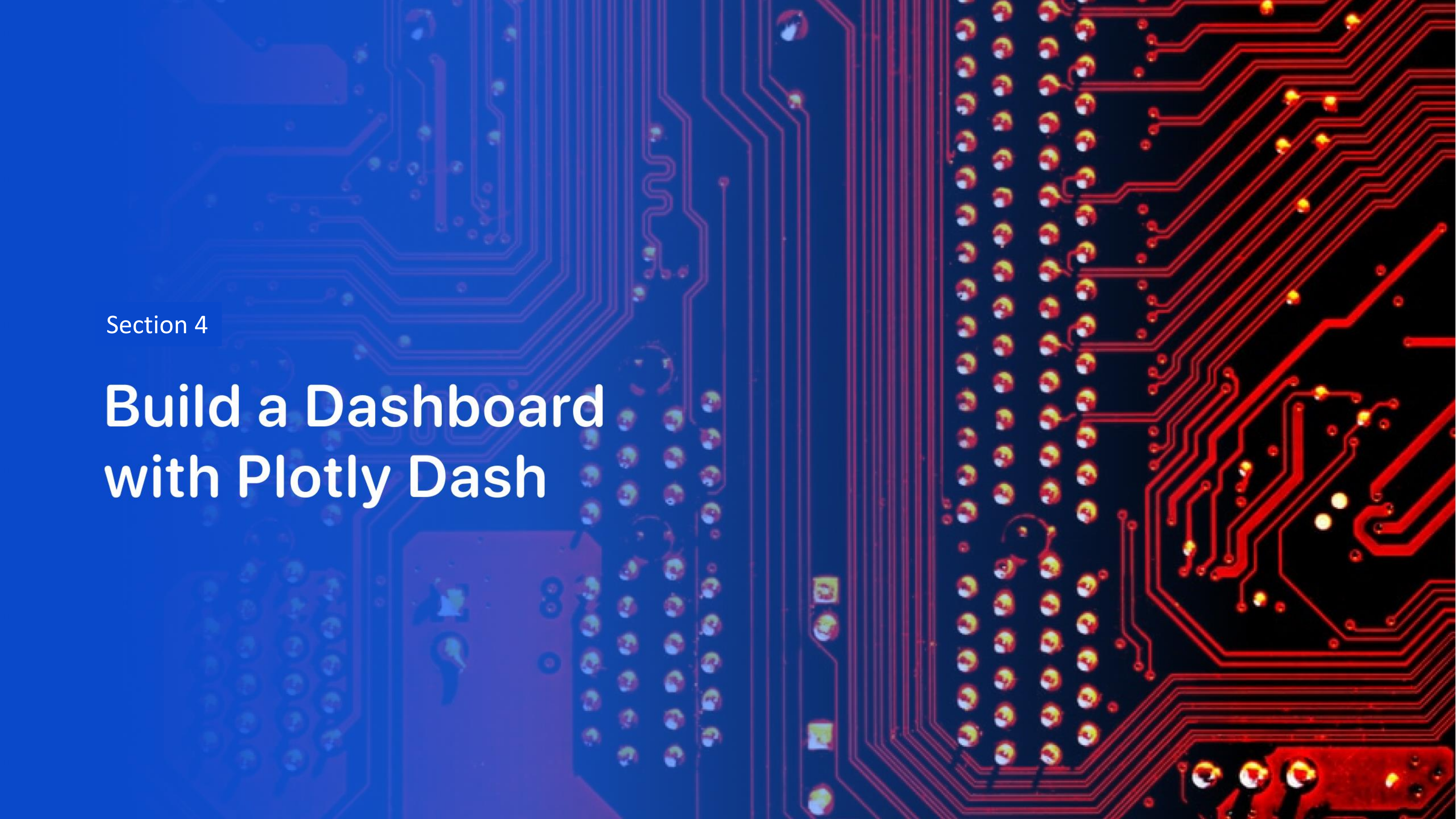


- Left coast site has 10 trails and right coast site has 46 trails

The proximity of the launch sites



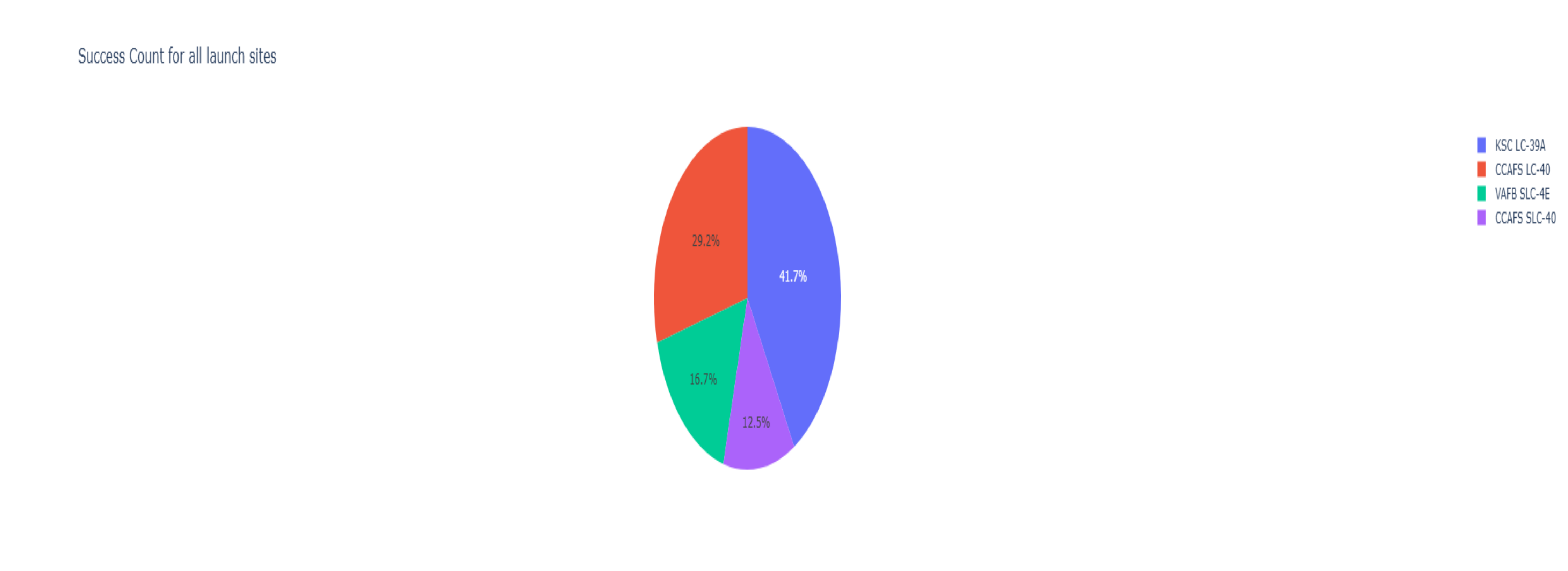
- KSC LC-39 A is 3.37 km far from the coast, and 5.11 km from the city
- CCAFS LC-40 is1.91km from the highway and 1.34km from the railway



Section 4

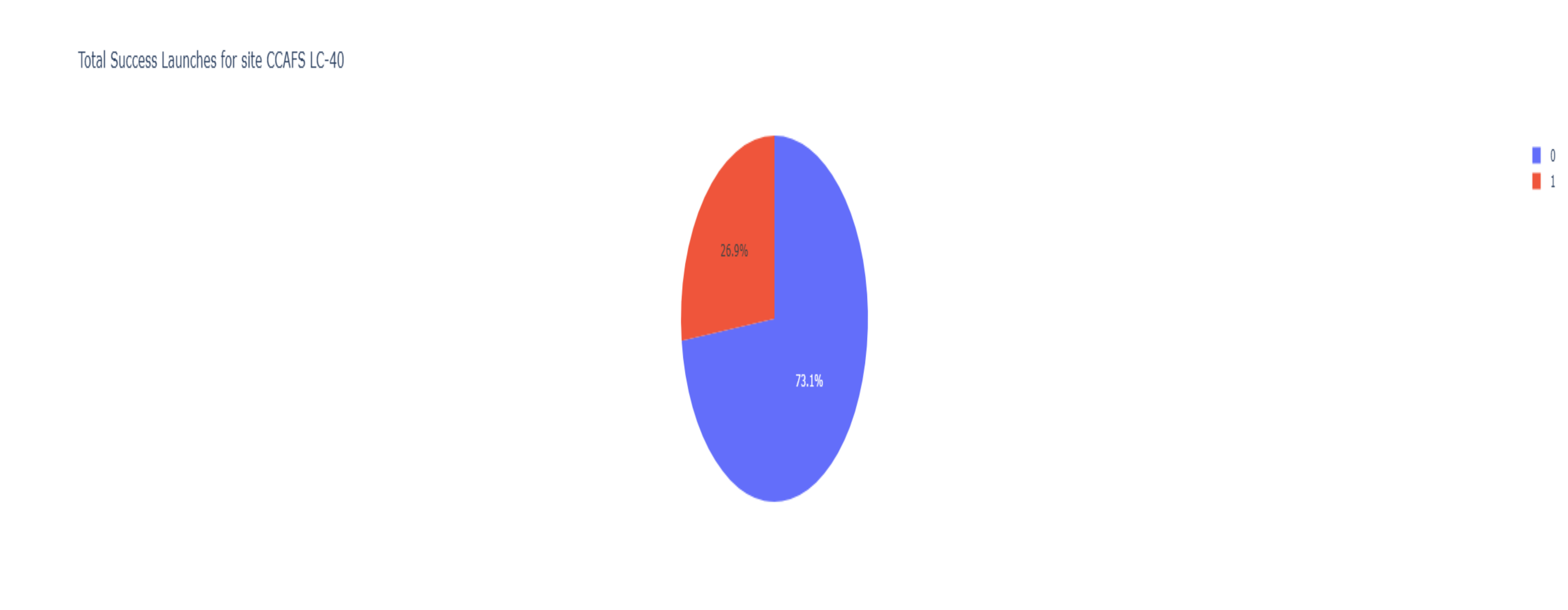
Build a Dashboard with Plotly Dash

Pie-Chart for launch success count for all sites



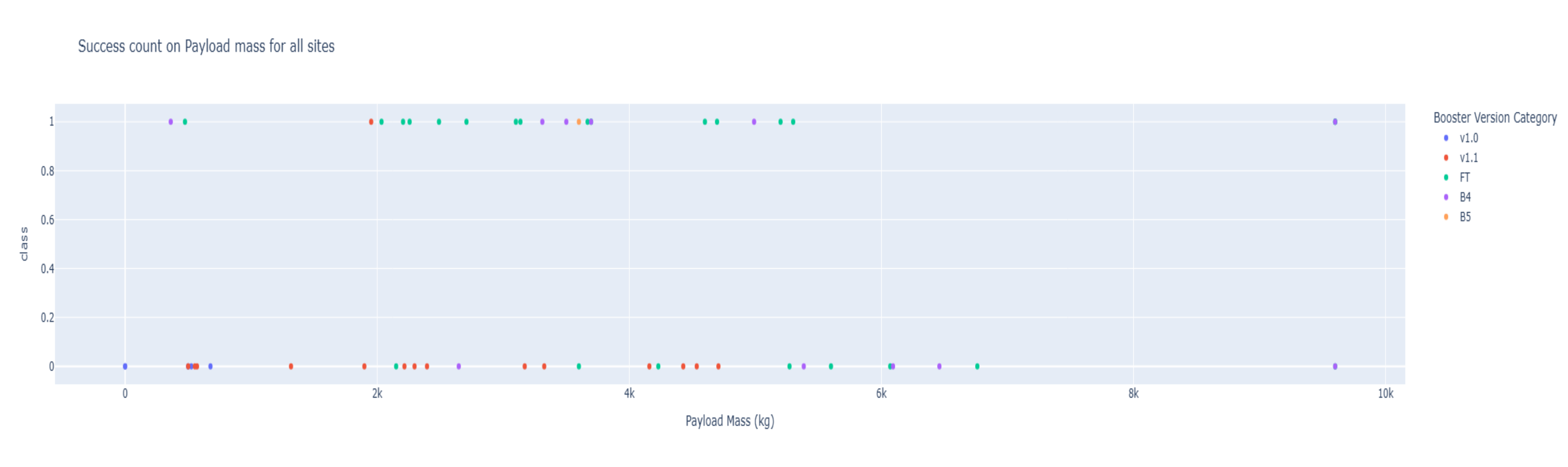
- Launch site KSC LC-39A has the highest launch success rate at 42% followed by CCAFS LC-40 at 29%, VAFB SLC-4E at 17% and lastly launch site CCAFS SLC-40 with a success rate of 13%

Pie chart for the launch site with highest launch success ratio



- Launch site CCAFS LC-40 had the highest success ratio of 73% success against 27% failed launches

Payload vs. Launch Outcome scatter plot for all sites

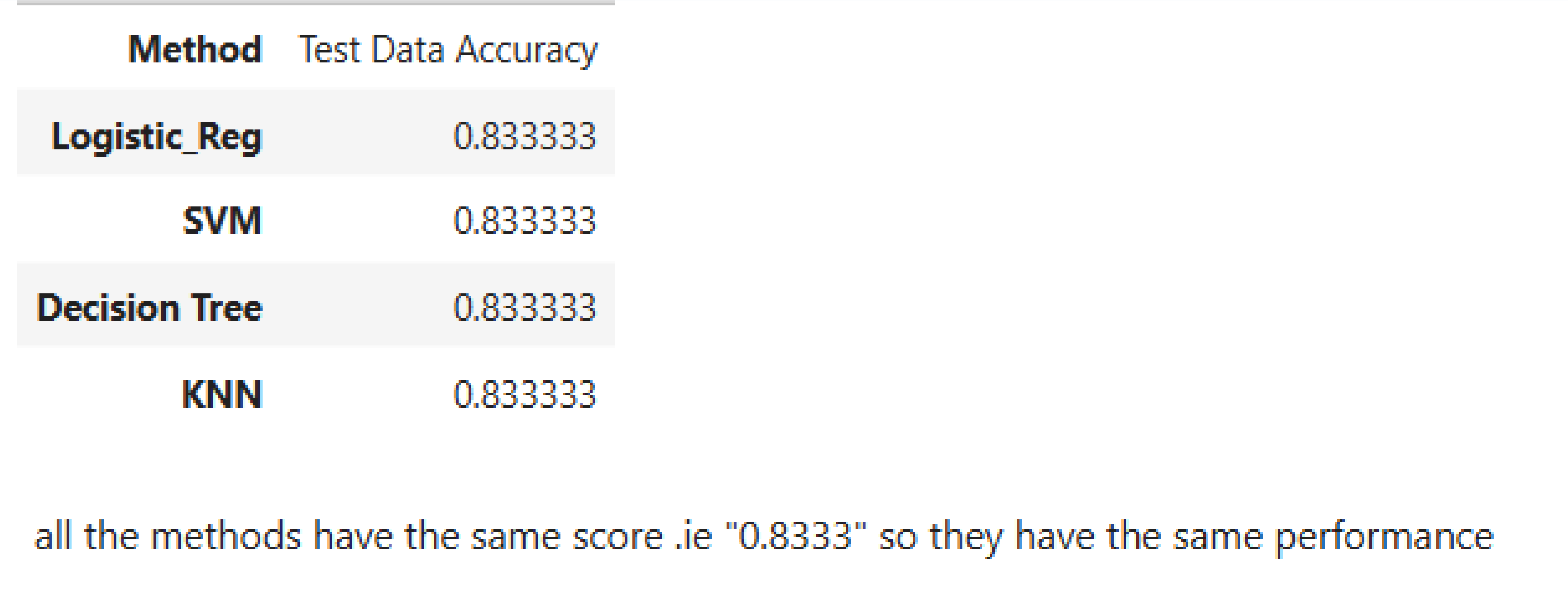


- V1.0 can take heaviest payload
- Most success landing happens between payload from 2k to 5k
- FT has the highest success rate

Section 5

Predictive Analysis (Classification)

Classification Accuracy

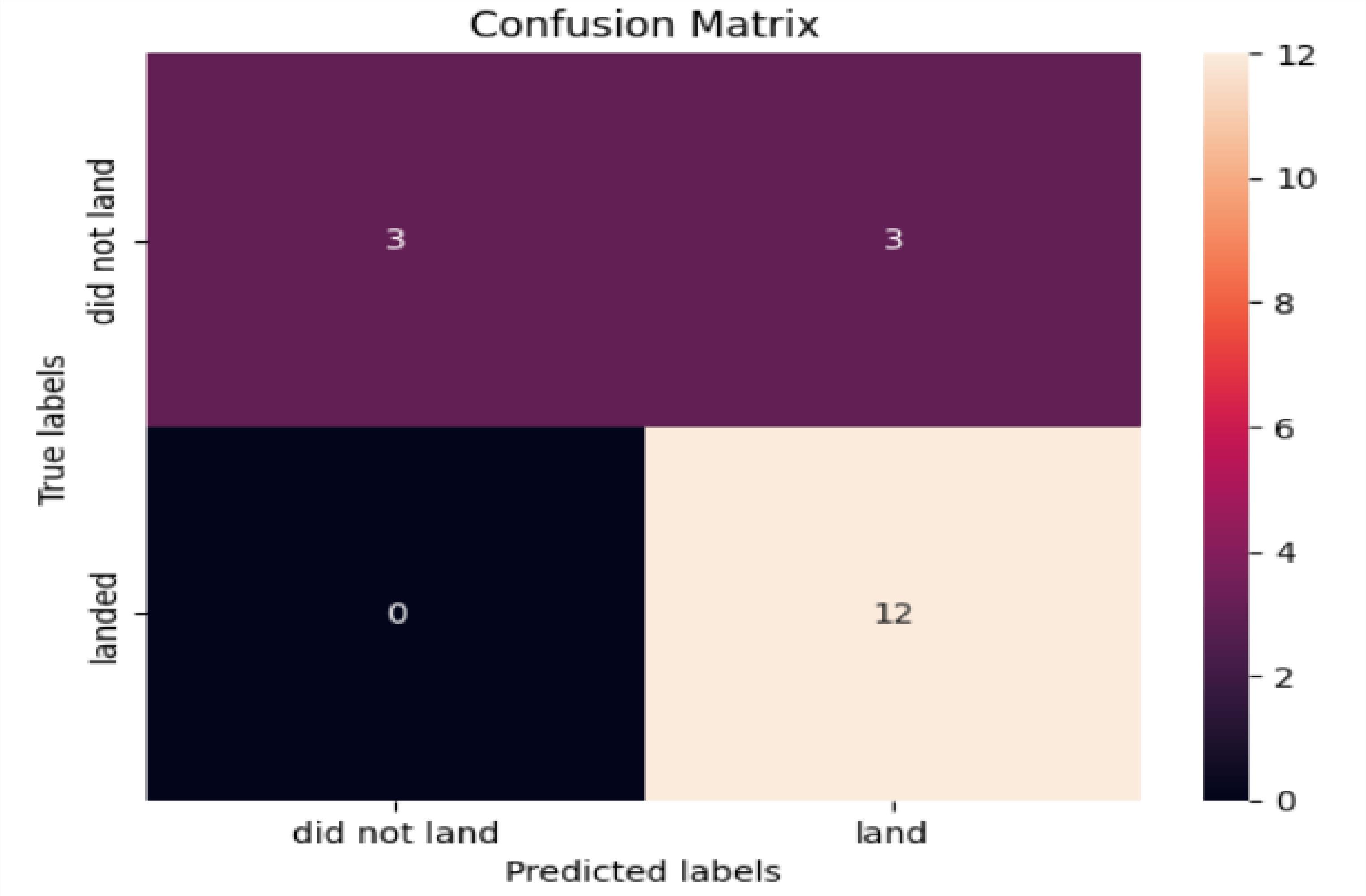


Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

all the methods have the same score .ie "0.8333" so they have the same performance

Confusion Matrix

- All the 4 classification model had the same confusion matrixes and were able equally distinguish between the different classes. The major problem is false positives for all the models.



Conclusions

- Different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
- There is a correlation between launch site and success rate Payload mass is also associated with the success rate.: the more massive the payload, the less likely the first stage will return
- For orbit type, SO has the least success rate while ES-L1, GEO, HEO and SSO have the highest success rate According to the yearly trend
- There has been an increase in the success rate since 2013 kept increasing till 2020
- The accuracy of all models are same(83.33 %) so anyone of them can be used for classification.

Appendix

- https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/dataset_part_1.csv
- https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/dataset_part_2.csv
- https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/dataset_part_3.csv
- https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/spacex_launch_dash.csv
- https://github.com/ankur013/IBM-Data-Science/blob/main/Capstone/spacex_web_scraped.csv

Thank you!