IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alain Daniel Godo Alonso
5/7/2023

Outline

1. Executive Summary
2. Introduction
3. Methodology
4. Results
5. Conclusion
6. Appendix

Executive Summary

Summary of methodologies

- Data Collection through API
- Data Collection with Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analytics (Folium and Ploty Dash)
- Machine Learning Prediction

Summary of all results

- Exploratory Data Analysis result
- Interactive analytics in screenshots
- Predictive Analytics result

Introduction

Project background and context

Space X 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 Space X 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 Space X for a rocket launch. The goal of the project is to create a machine-learning pipeline to predict if the first stage will land successfully.

Problems you want to find answers

- What factors determine if the rocket will land successfully?

- The interaction amongst various features that determine the success rate of a successful landing.

- What operating conditions need to be in place to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building different models: Logistic Regression, Decision Tree, SVM, KNN.

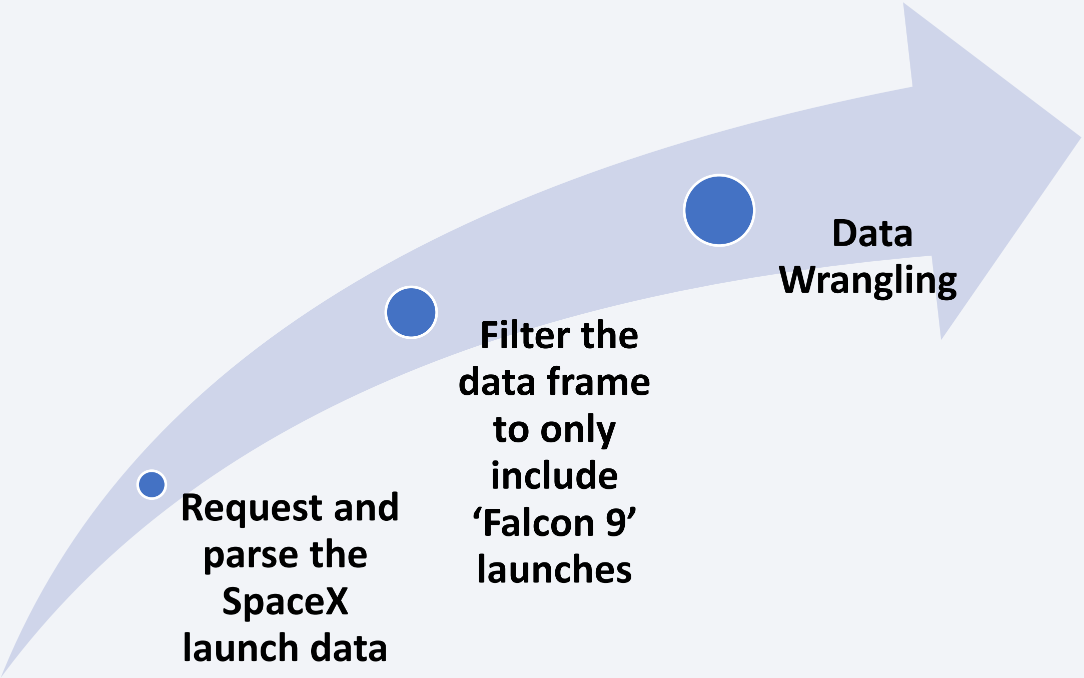
Data Collection

- Data was collected from two principal resources: SpaceX API and Wikipedia.
- Data from SpaceX API was cleaned, checked for missing values, and filled where was necessary.
- To get the data from Wikipedia performed a web scraping using the **BeautifulSoup** module in Python.
- All the data were processed using **Pandas** as the main framework in Python.

Data Collection – SpaceX API

- Link to the notebook in GitHub:

[IBM-CapstonProject/jupyter-labs-spacex-data-collection-api.ipynb at main · Alain-Godo/IBM-CapstonProject (github.com)]



Request and parse the SpaceX launch data

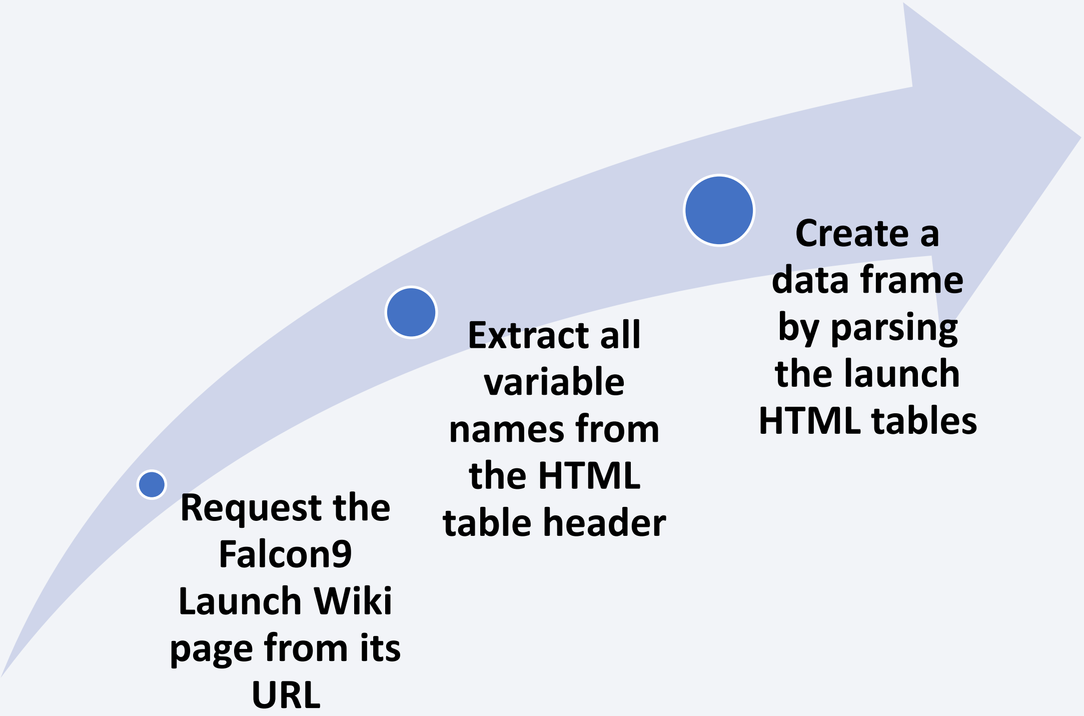
Filter the data frame to only include ‘Falcon 9’ launches

Data Wrangling

Data Collection - Scraping

- Link to the notebook in Github:

IBM-CapstonProject/jupyter-labs-webscraping.ipynb at main · Alain-Godo/IBM-CapstonProject (github.com)



Request the Falcon9 Launch Wiki page from its URL

Extract all variable names from the HTML table header

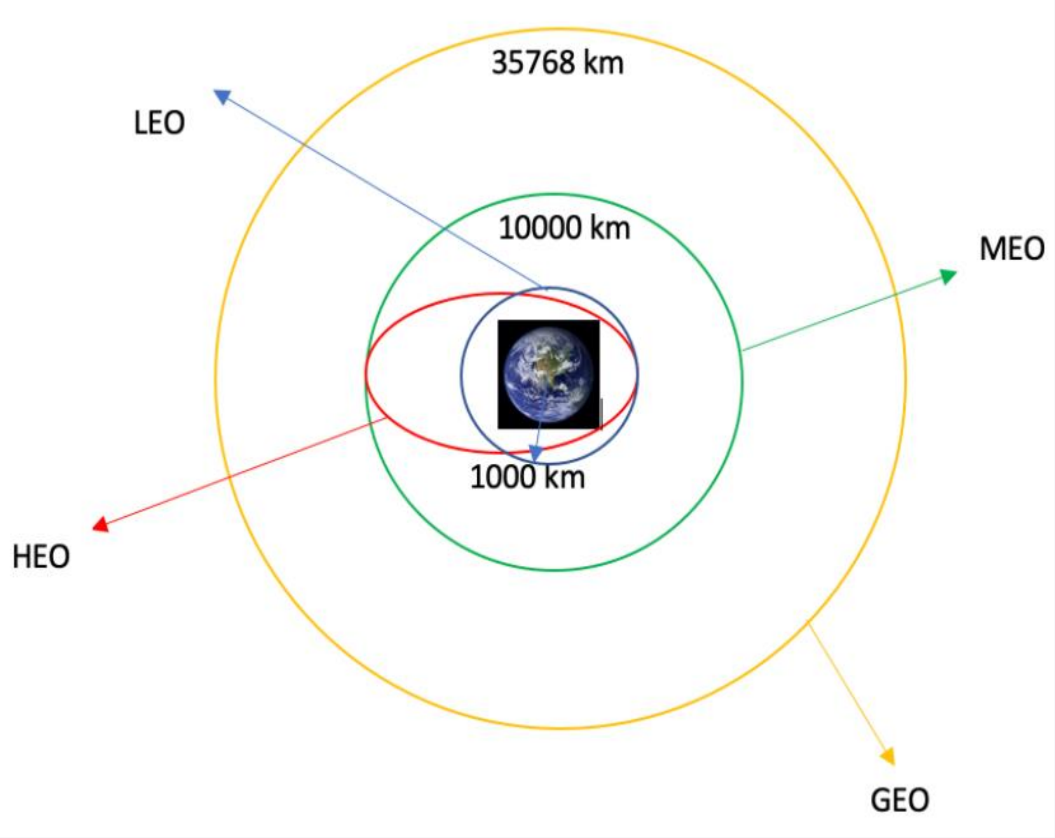
Create a data frame by parsing the launch HTML tables

Data Wrangling

In this step was performed some Exploratory Data Analysis (EDA). Found some patterns in the data and determined what would be the label for training supervised models.

Was calculated the number of launches on each site and the occurrences of each orbit.

The categorical features were converted to numerical ones.

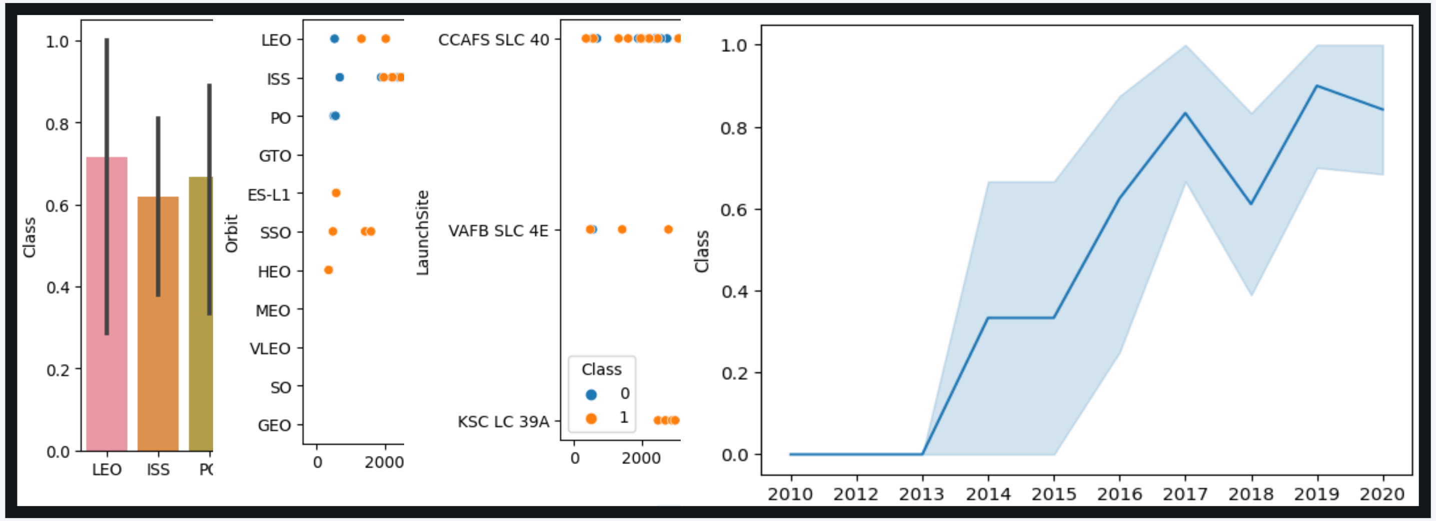


Link to the notebook in Github:

IBM-CapstonProject/labs-jupyter-spacex-Data wrangling.ipynb at main · Alain-Godo/IBM-CapstonProject (github.com)

EDA with Data Visualization

We analyzed the data by creating visualizations that displayed the correlation between flight number and launch site, payload, and launch site. We also examined the success rates of different orbit types, flight numbers, and orbit types, as well as the annual trend of launch successes.



Link to the notebook in Github:
IBM-CapstonProject/jupyter-labs-eda-dataviz.ipynb at main · Alain-Godo/IBM-CapstonProject (github.com)

EDA with SQL

In the Jupyter Notebook, we seamlessly uploaded the SpaceX dataset into a SQLite database. To gain insight from the data, we utilized SQL for exploratory data analysis. Through our queries, we were able to discover valuable information such as the unique launch sites involved in the space mission, the total payload mass of boosters launched by NASA (CRS), the average payload mass of booster version F9 v1.1, as well as the total number of successful and failed mission outcomes. Additionally, we were able to identify the names of launch sites, booster versions, and drone ships associated with failed landing outcomes.

IBM-CapstonProject/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · Alain-Godo/IBM-CapstonProject (github.com)

Build an Interactive Map with Folium

We have added markers, circles, and lines to a folium map to indicate the success or failure of launches from each launch site. We have also labeled these outcomes as either a 0 for failure or a 1 for success. By using color-coded marker clusters, we were able to identify launch sites with a high success rate. We have also assessed the distances between launch sites and their surroundings, answering questions such as whether they are located near railways, highways, coastlines, or cities.

IBM-CapstonProject/Step 6 at main · Alain-Godo/IBM-CapstonProject (github.com)

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash.
- We plotted pie charts showing the total launches by certain sites.
- We plotted a scatter graph showing the relationship between Outcome and Payload Mass (Kg) for the different booster versions.

IBM-CapstonProject/Step 7 at main · Alain-Godo/IBM-CapstonProject (github.com)

Predictive Analysis (Classification)

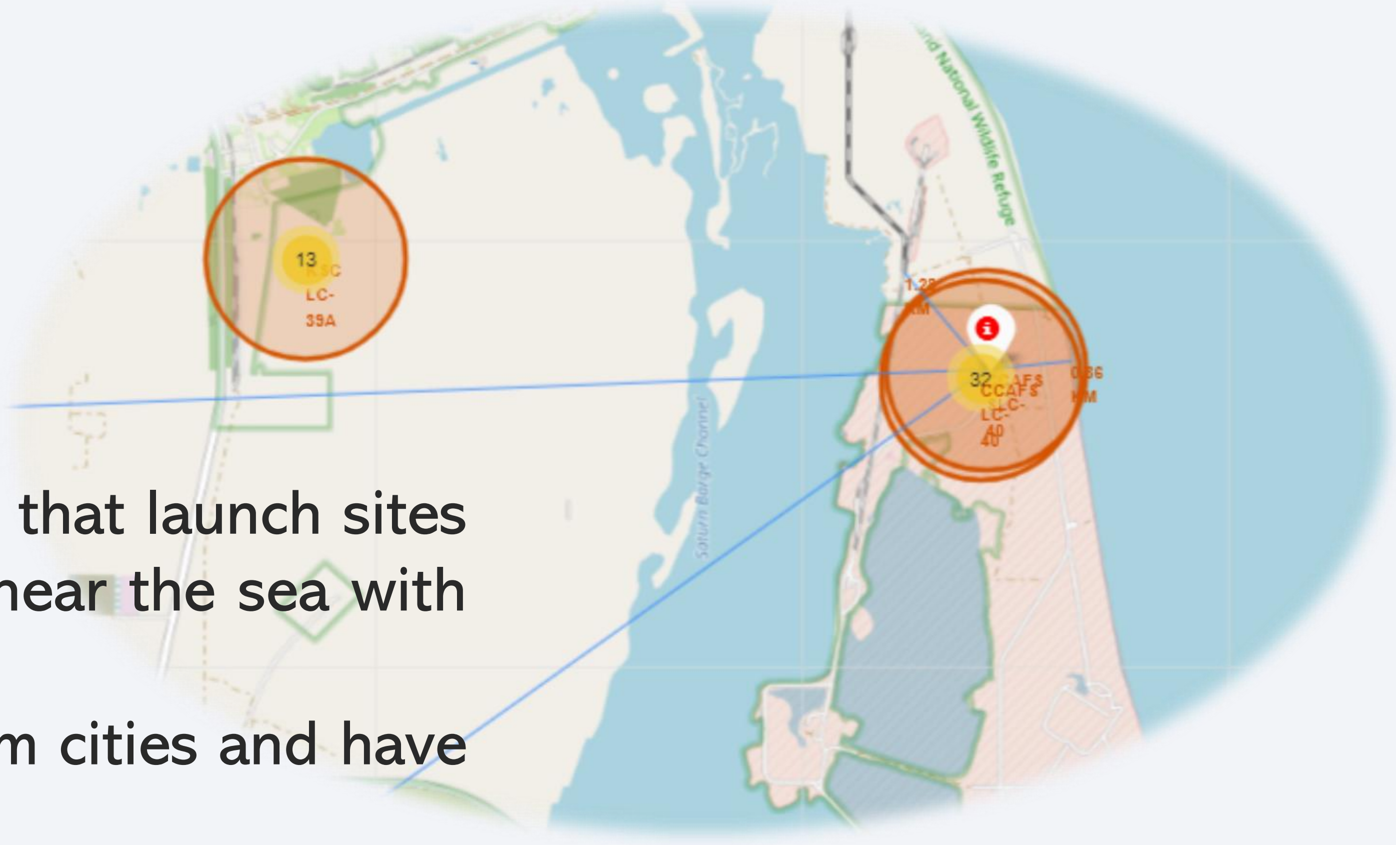
- The data was loaded and transformed using Numpy and pandas.
- The data was then split into training and testing sets.
- Various machine learning models were created and their hyperparameters were fine-tuned using GridSearchCV.
- Accuracy was chosen as the metric for the model and it was further improved through feature engineering and algorithm tuning.
- The highest-performing classification model was identified.

IBM-CapstonProject/Step 8 at main · Alain-Godo/IBM-CapstonProject (github.com)

Results

- Space X has launched from 4 different sites.
- The first launches were to Space X and NASA.
- The average payload of the F9 v1.1 booster is 2,928 kg.
- The first successful landing occurred in 2015, five years after the first launch.
- Many Falcon 9 booster versions have successfully landed on drone ships with payloads above the average.
- Almost all mission outcomes have been successful.
- Two booster versions failed to land on drone ships in 2015: F9 v1.1 B1012 and F9 v1.1 B1015.
- The number of successful landings has increased over time.

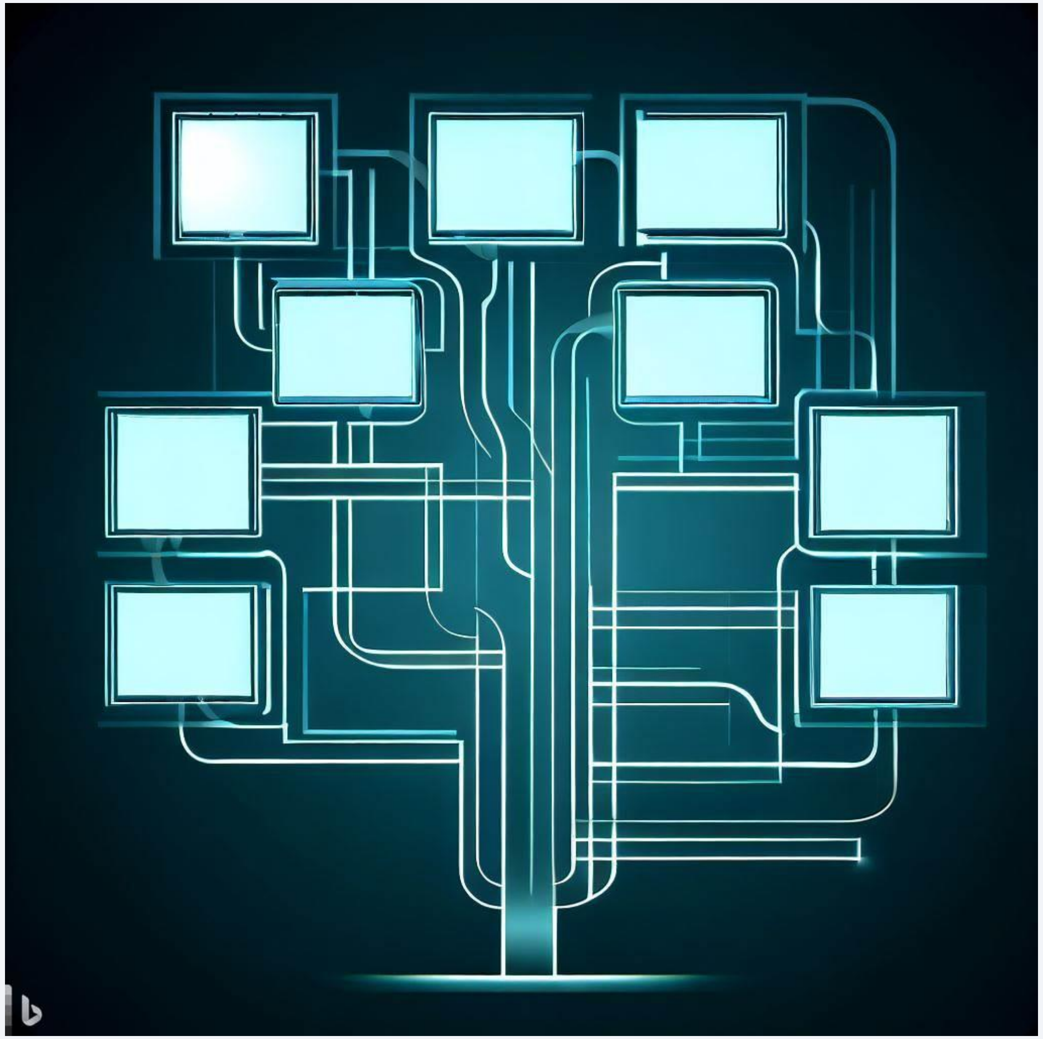
Results



- Interactive map analytics revealed that launch sites are typically located in safe areas near the sea with good logistical infrastructure.
- Most launches occur far away from cities and have railroads near.

Results

Based on the results of the Predictive Analysis, the Decision Tree Classifier model is the most effective in predicting successful landings. It has an accuracy rate of over 88% when tested.

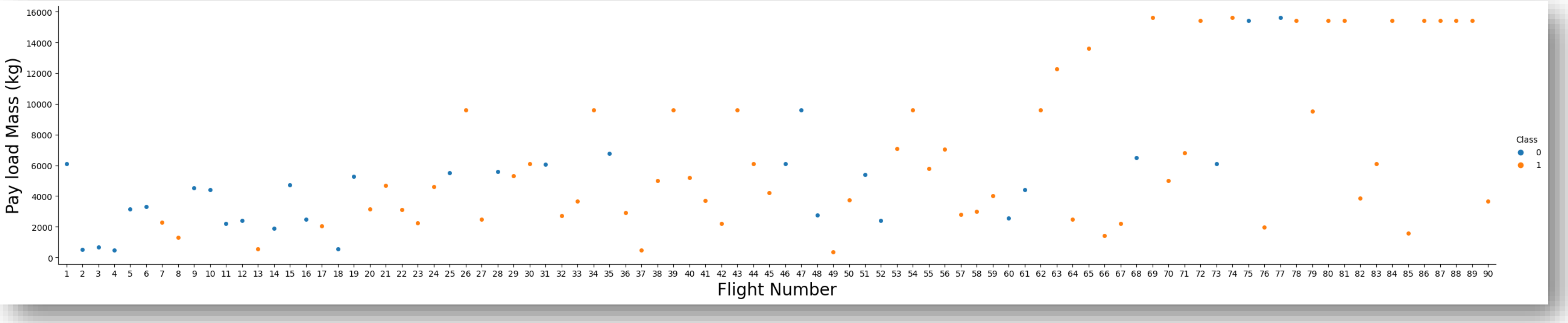


Section 2

Insights drawn from EDA

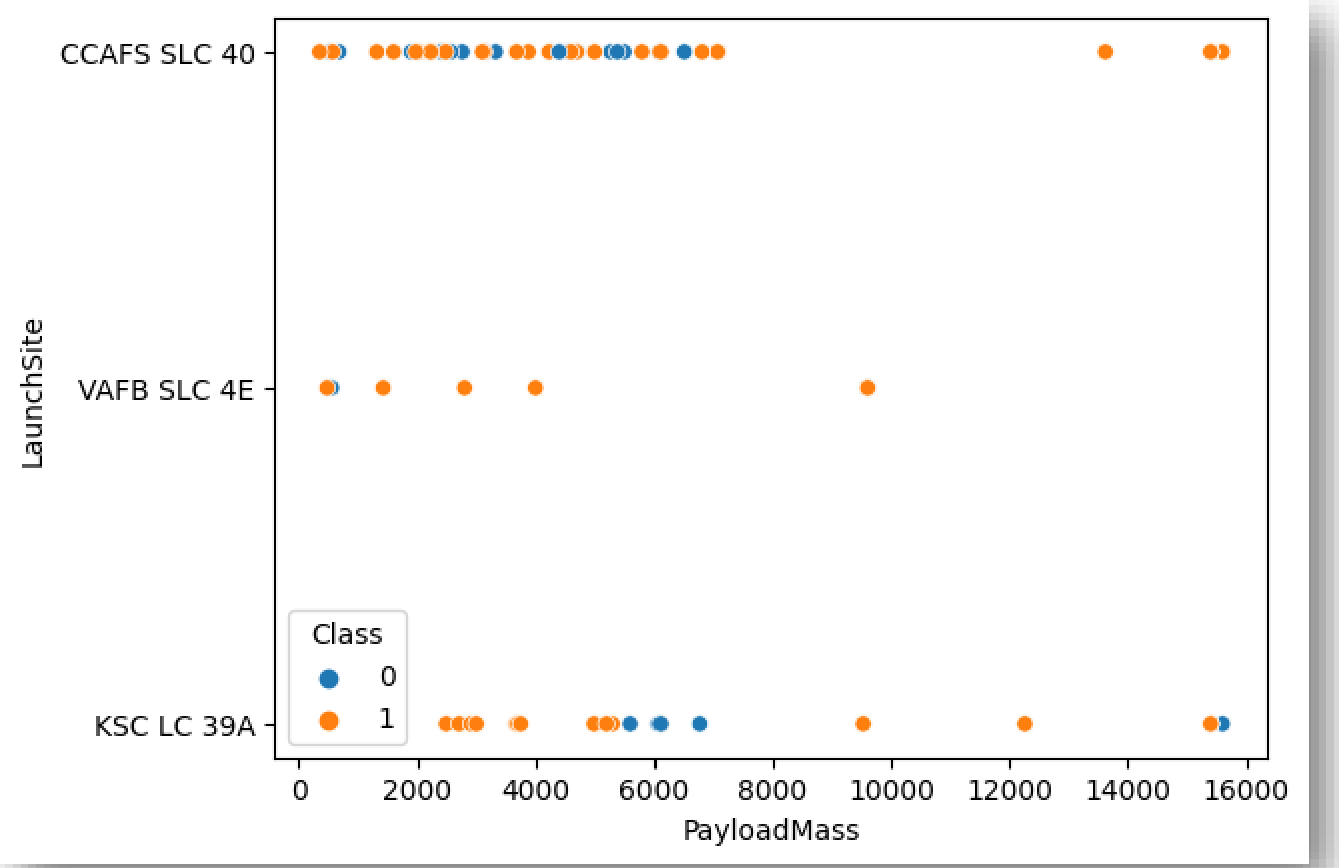
Flight Number vs. Launch Site

Based on the plot, we discovered that there is a positive correlation between the number of flights launched and the success rate at the launch site. In other words, as the flight amount increases, so does the success rate.



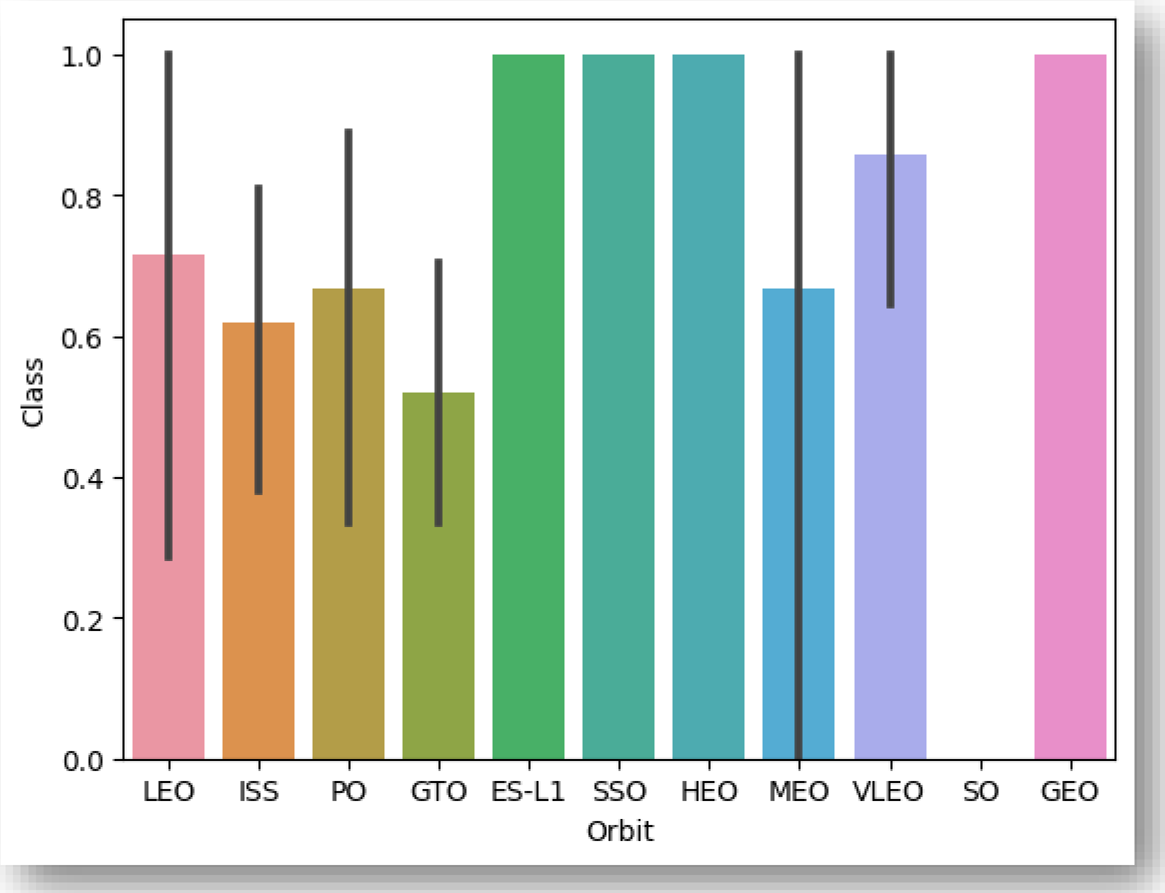
Payload vs. Launch Site

If you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).



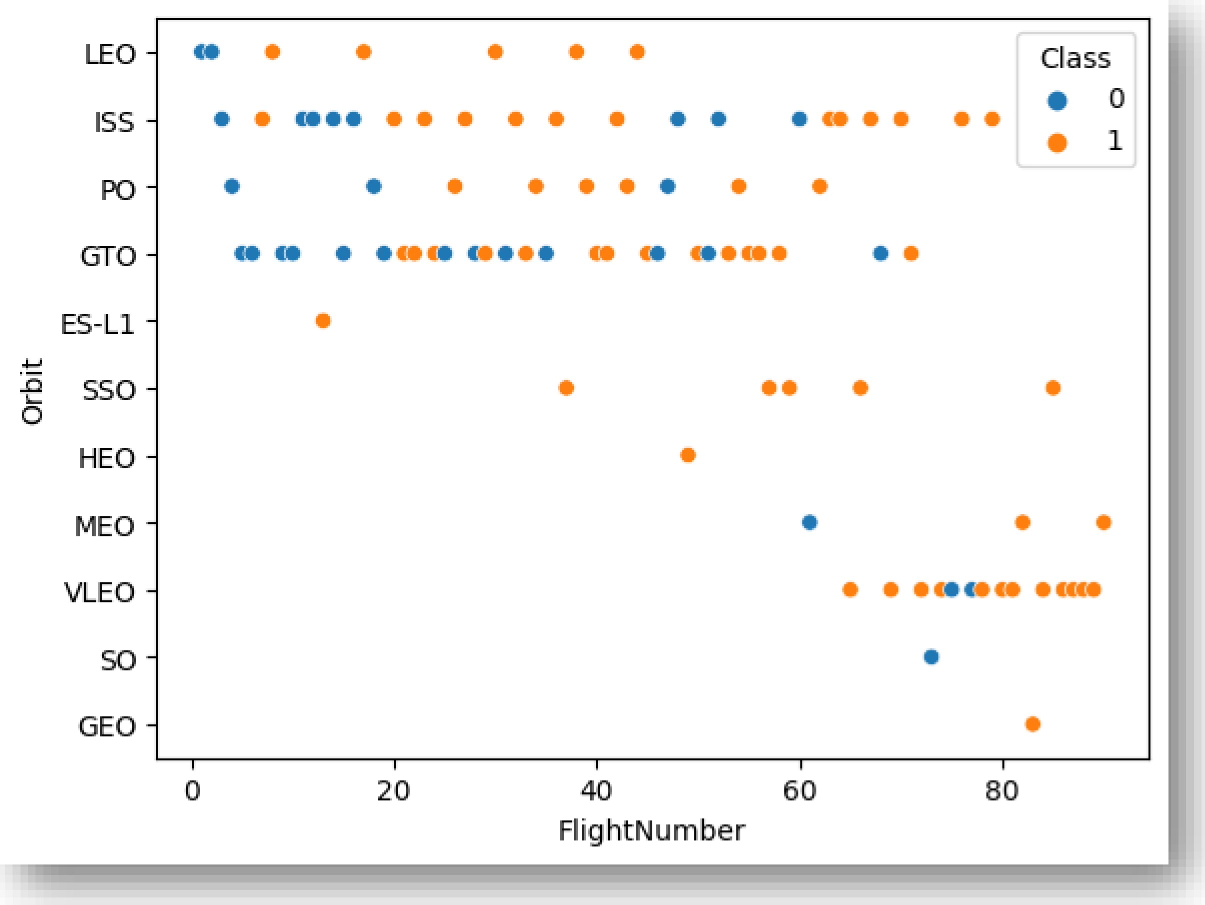
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



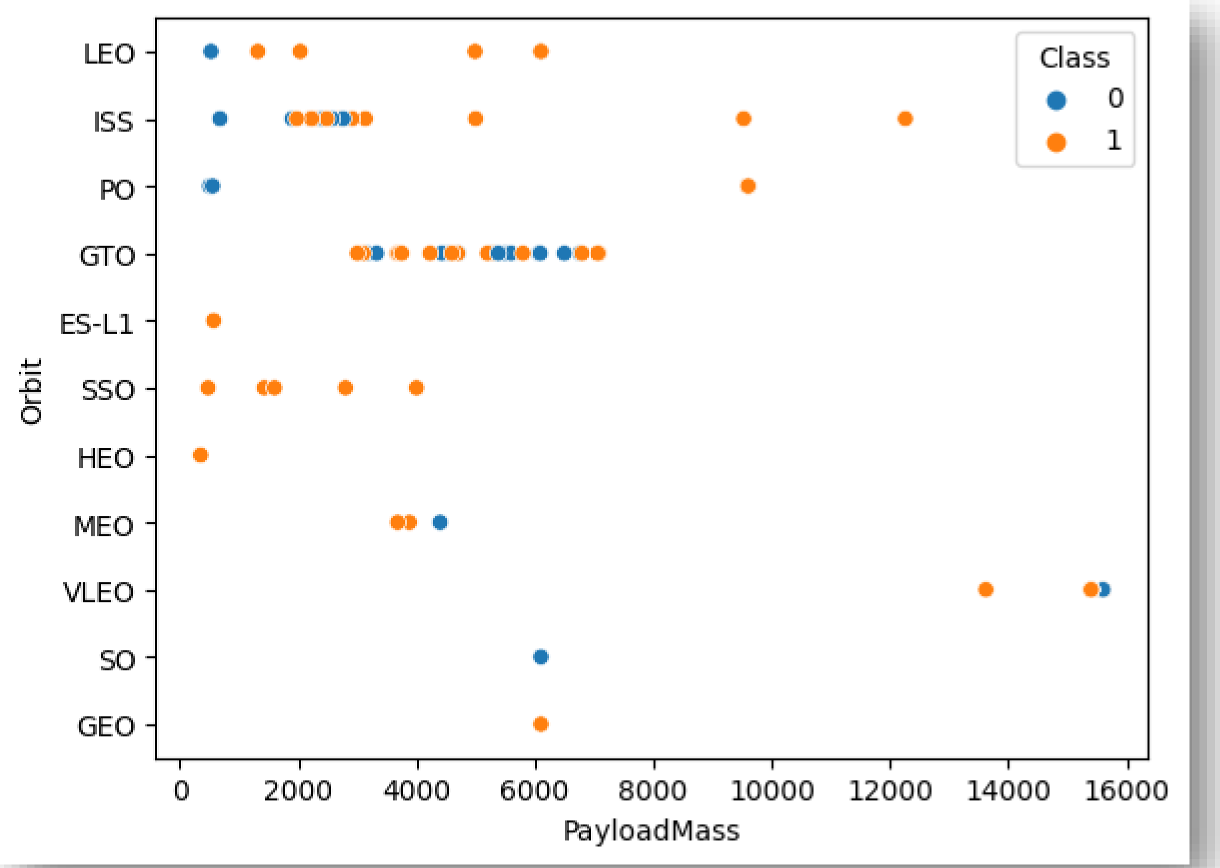
Flight Number vs. Orbit Type

It is observed that the number of flights in LEO orbit has a positive correlation with success, while there is no apparent correlation between flight number and success in GTO orbit.



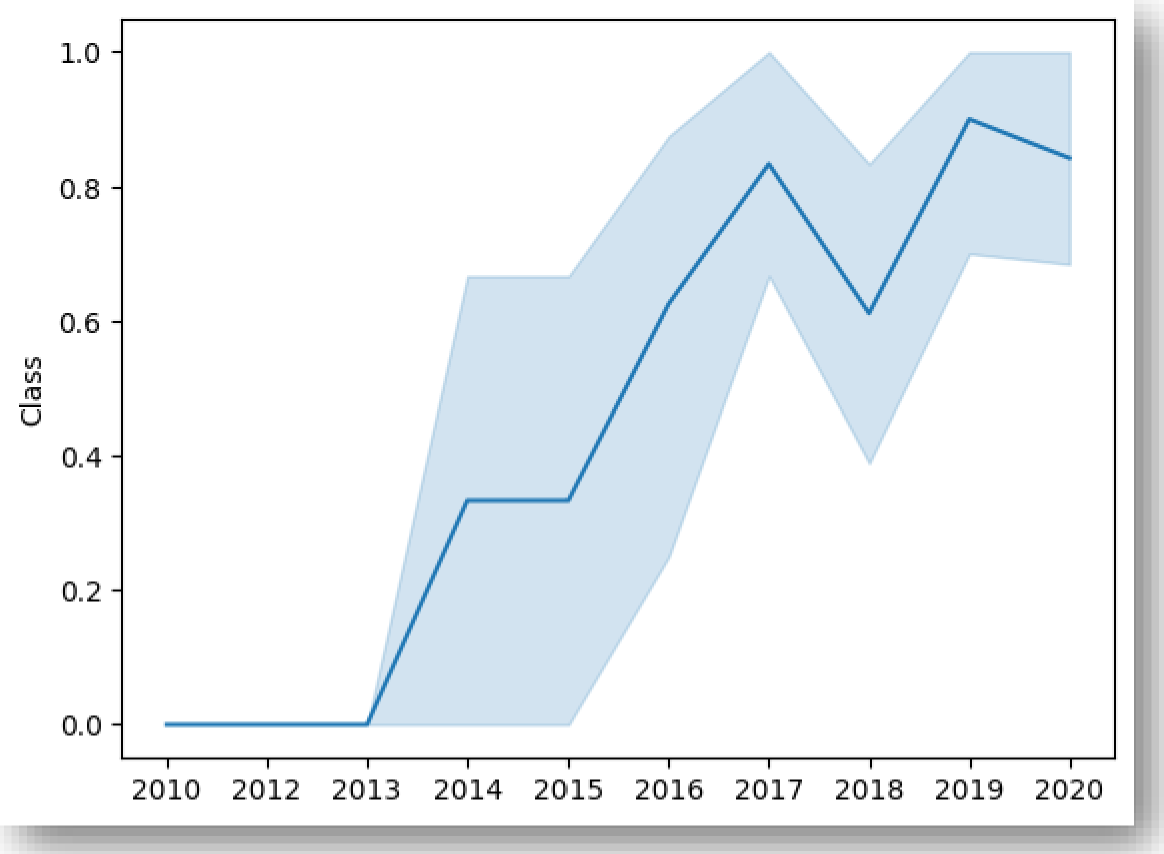
Payload vs. Orbit Type

We can observe that with heavy payloads, the successful landing is more for PO, LEO, and ISS orbits.



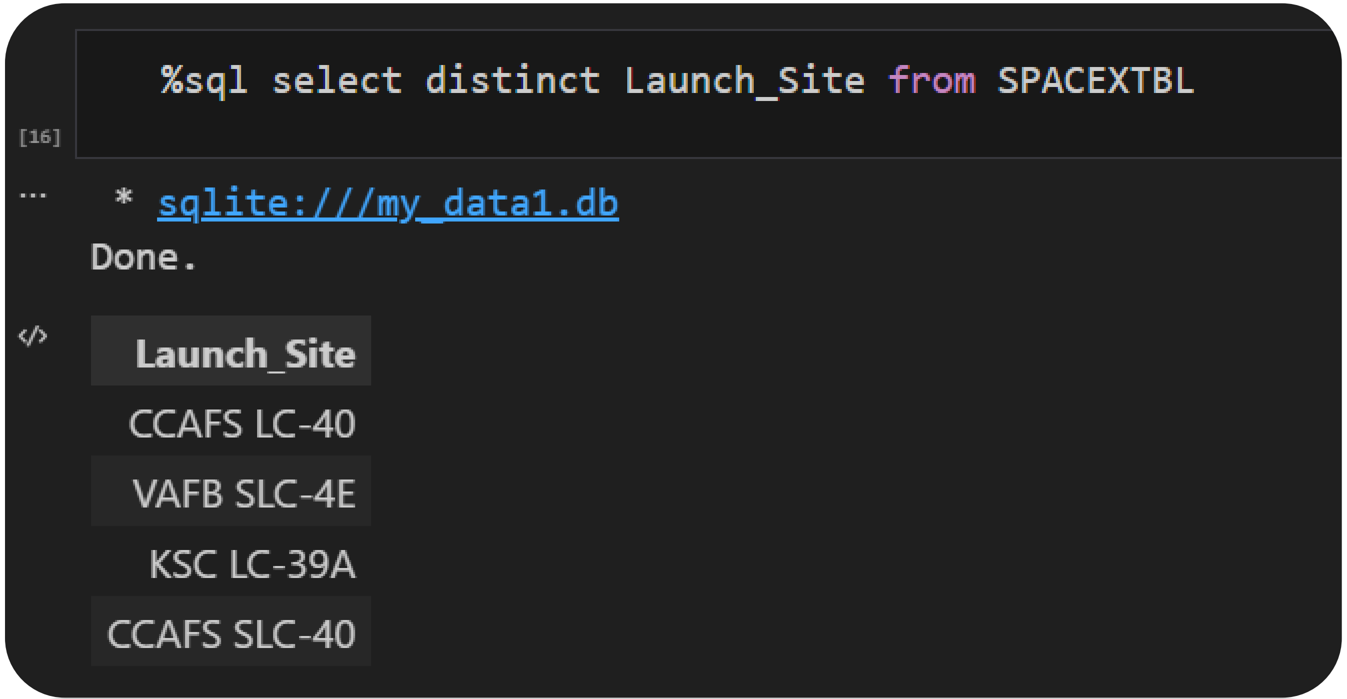
Launch Success Yearly Trend

Observing the plot, it is evident that the success rate has been consistently increasing from 2013 until 2020.



All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

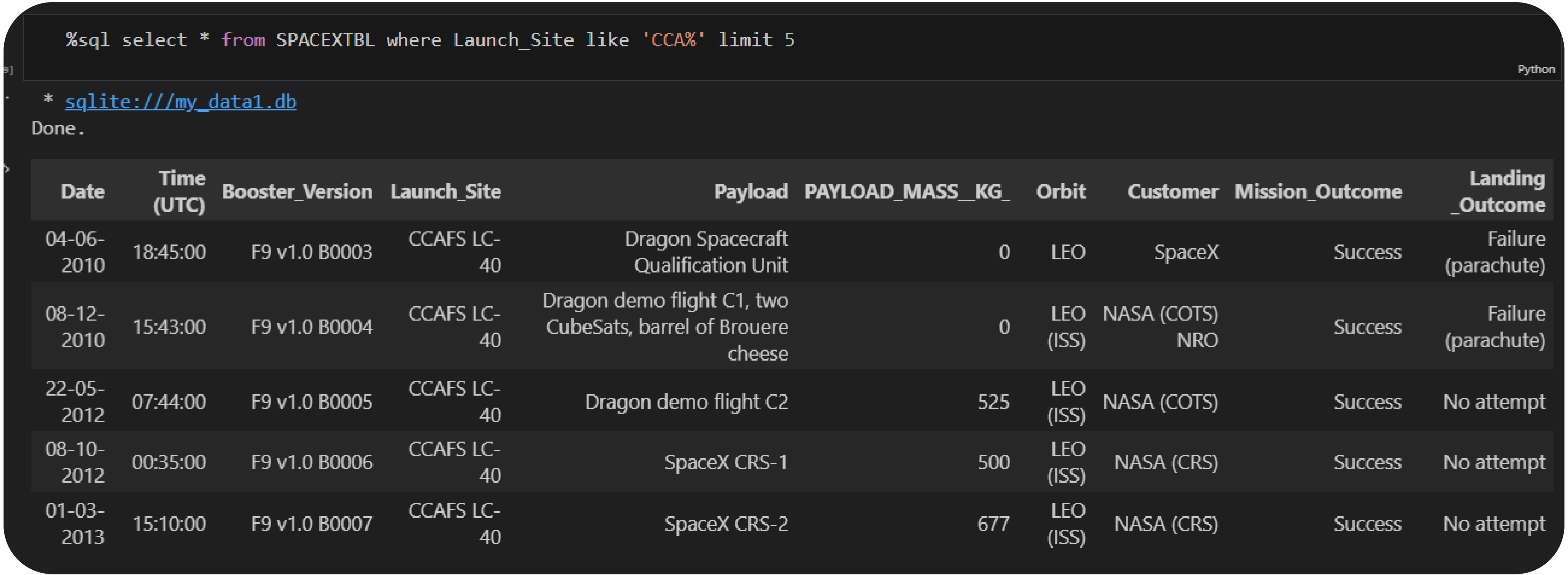


```
[16]  %sql select distinct Launch_Site from SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'



```
%sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5
```

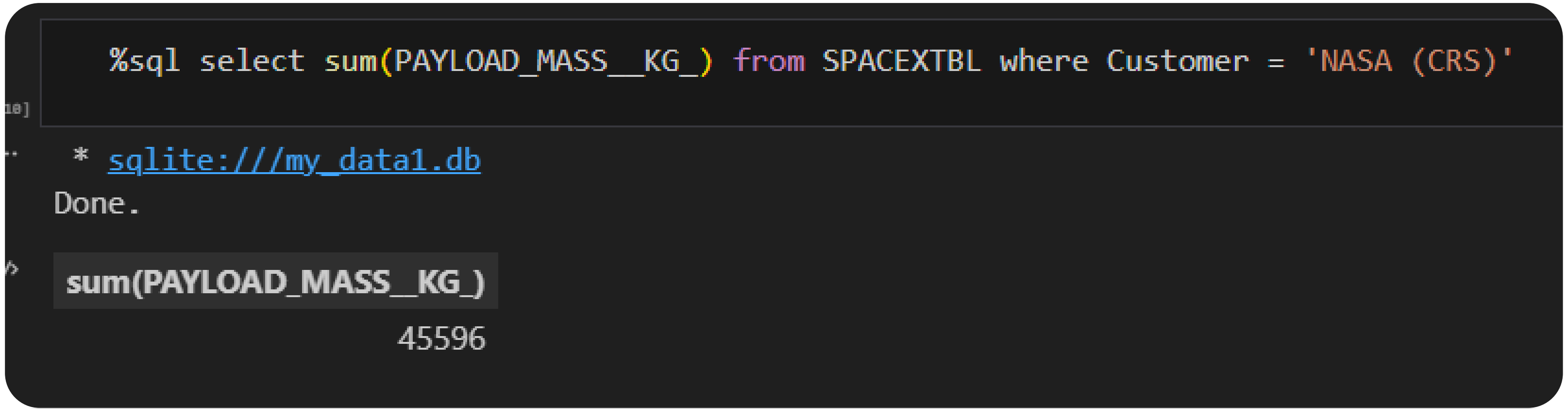
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

We used the query above to display 5 records where launch sites begin with ‘CCA’

Total Payload Mass

We calculated the total payload carried by boosters from NASA as 45596 using the query below

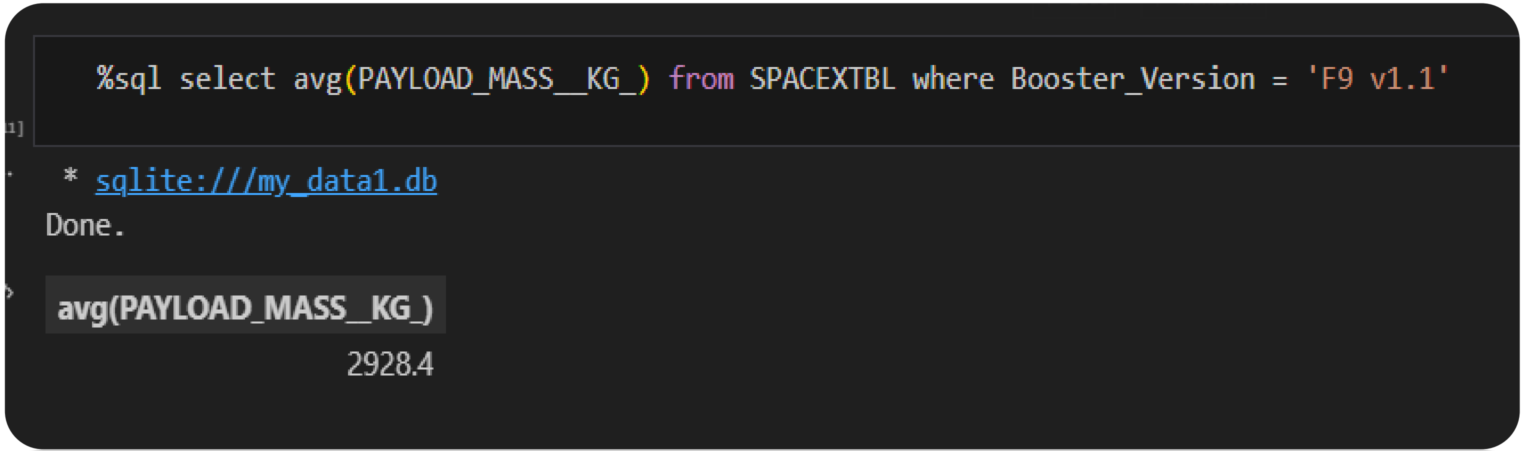


```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
                 45596
```

Average Payload Mass by F9 v1.1

We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4



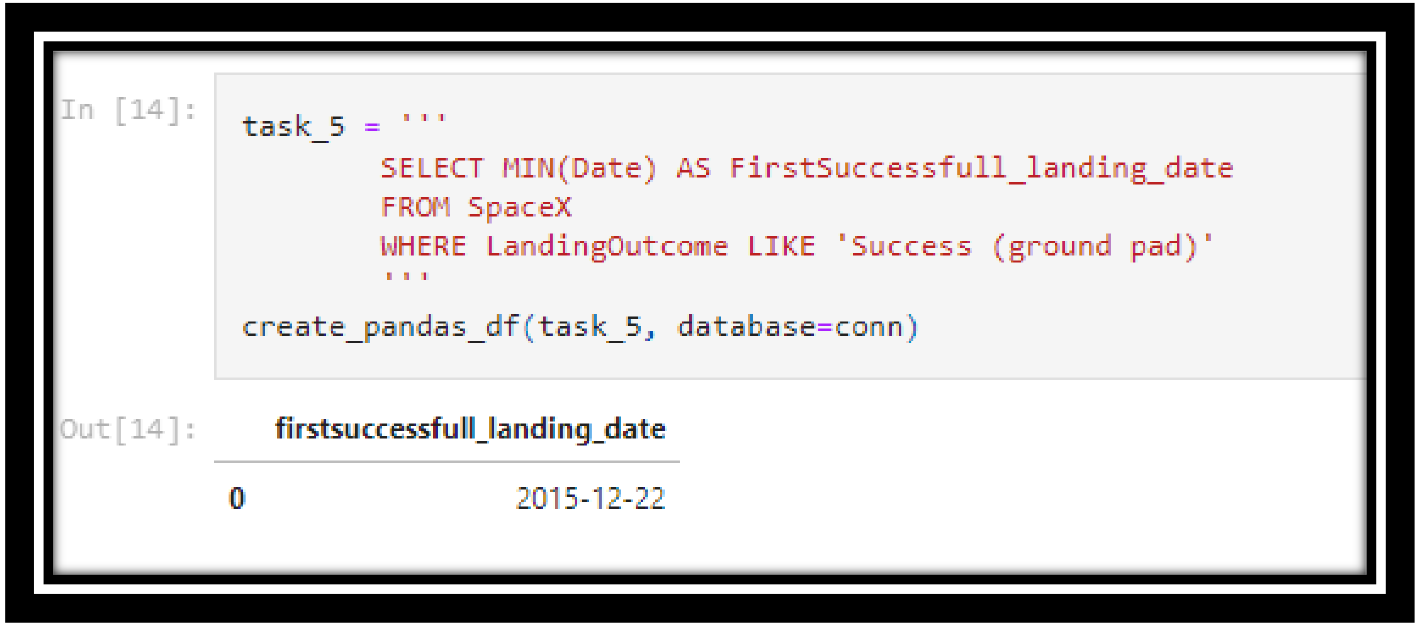
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

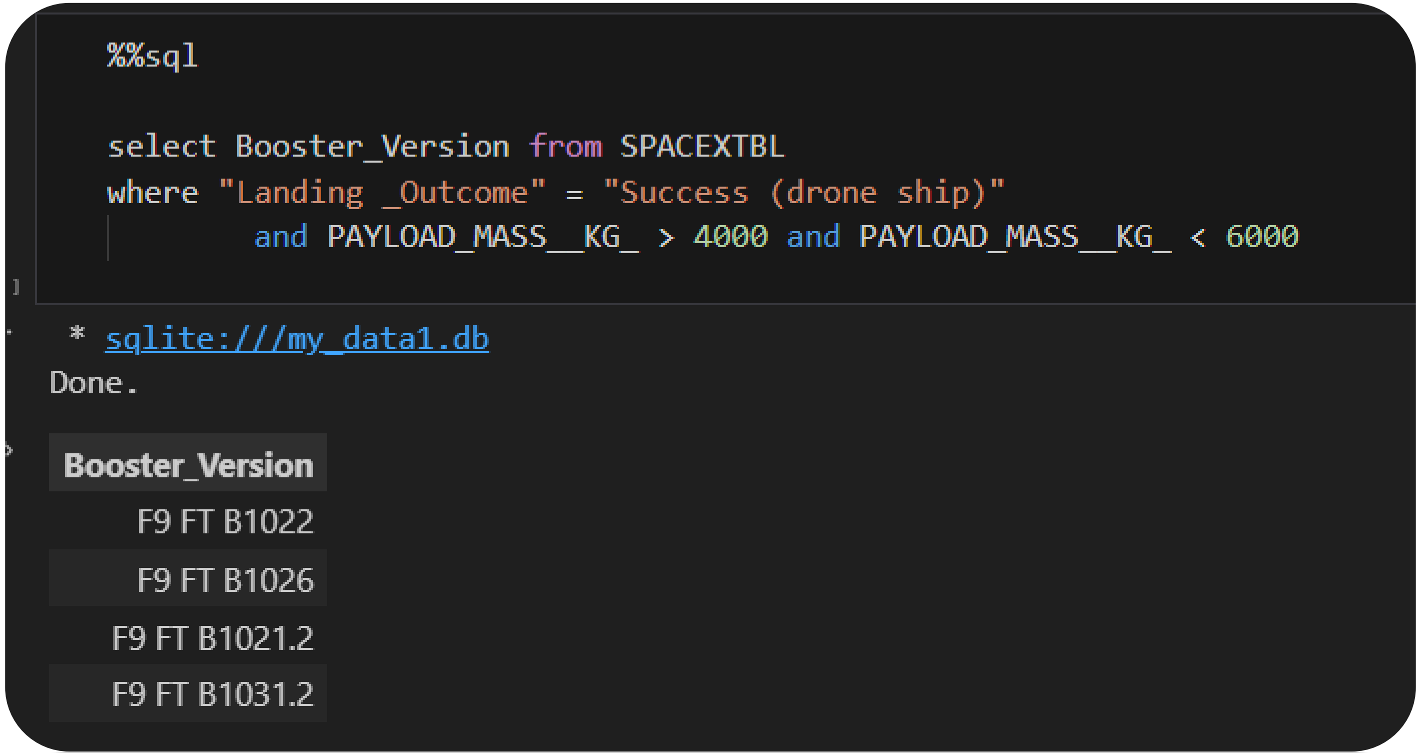
First Successful Ground Landing Date

We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015



```
In [14]:  task_5 = '''
                  SELECT MIN(Date) AS FirstSuccessfull_landing_date
                  FROM SpaceX
                  WHERE LandingOutcome LIKE 'Success (ground pad)'
                  '''
          create_pandas_df(task_5, database=conn)

Out[14]:    firstsuccessfull_landing_date
          0                    2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

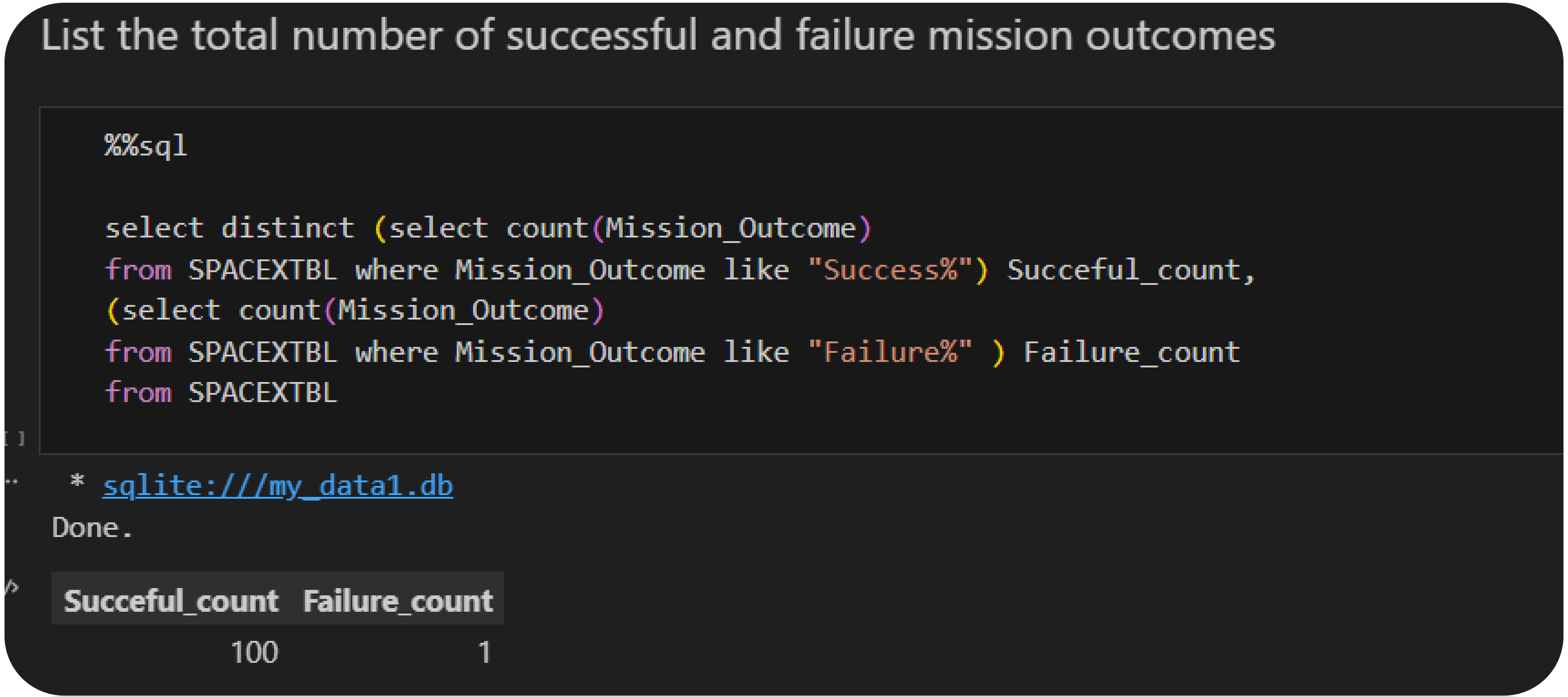


```
%%sql

select Booster_Version 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
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

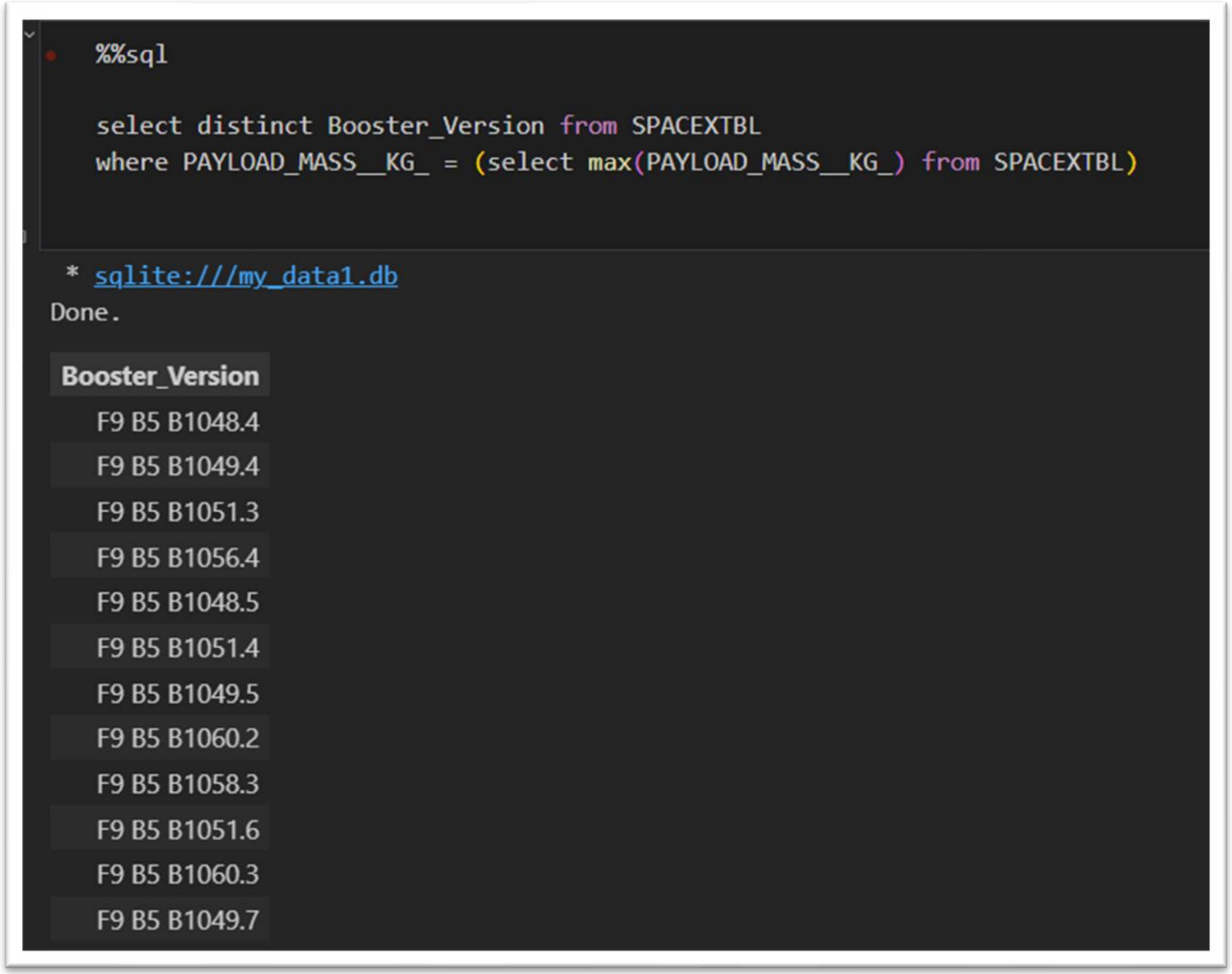
```
%%sql

select distinct (select count(Mission_Outcome)
from SPACEXTBL where Mission_Outcome like "Success%") Succeful_count,
(select count(Mission_Outcome)
from SPACEXTBL where Mission_Outcome like "Failure%" ) Failure_count
from SPACEXTBL
```

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

Succeful_count	Failure_count
100	1

Boosters Carried Maximum Payload

We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

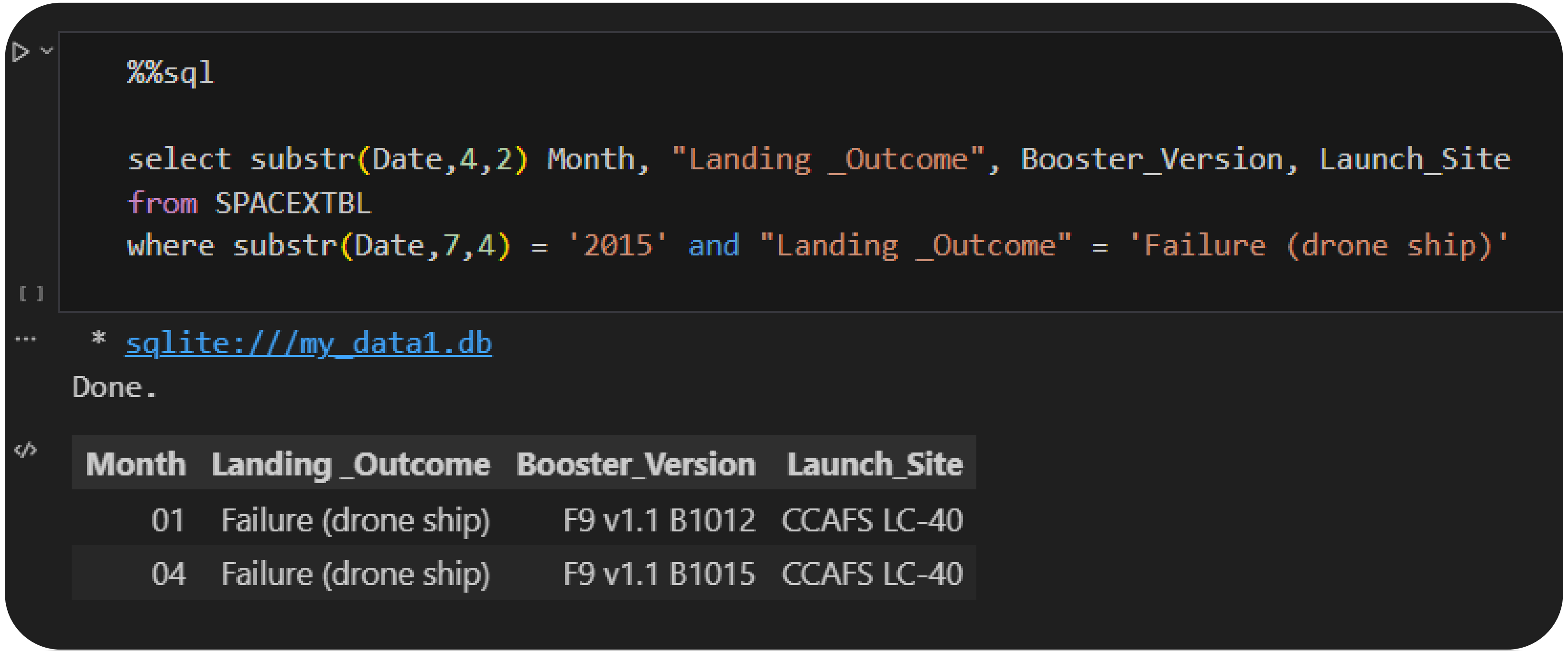


```
%%sql

select distinct Booster_Version from SPACEXTBL
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

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

We used a combination of the **WHERE** clause, **SUBSTR** function, **LIKE** and **AND** conditions to filter for failed landing outcomes in drone ships, their booster versions, and launch site names for the year 2015

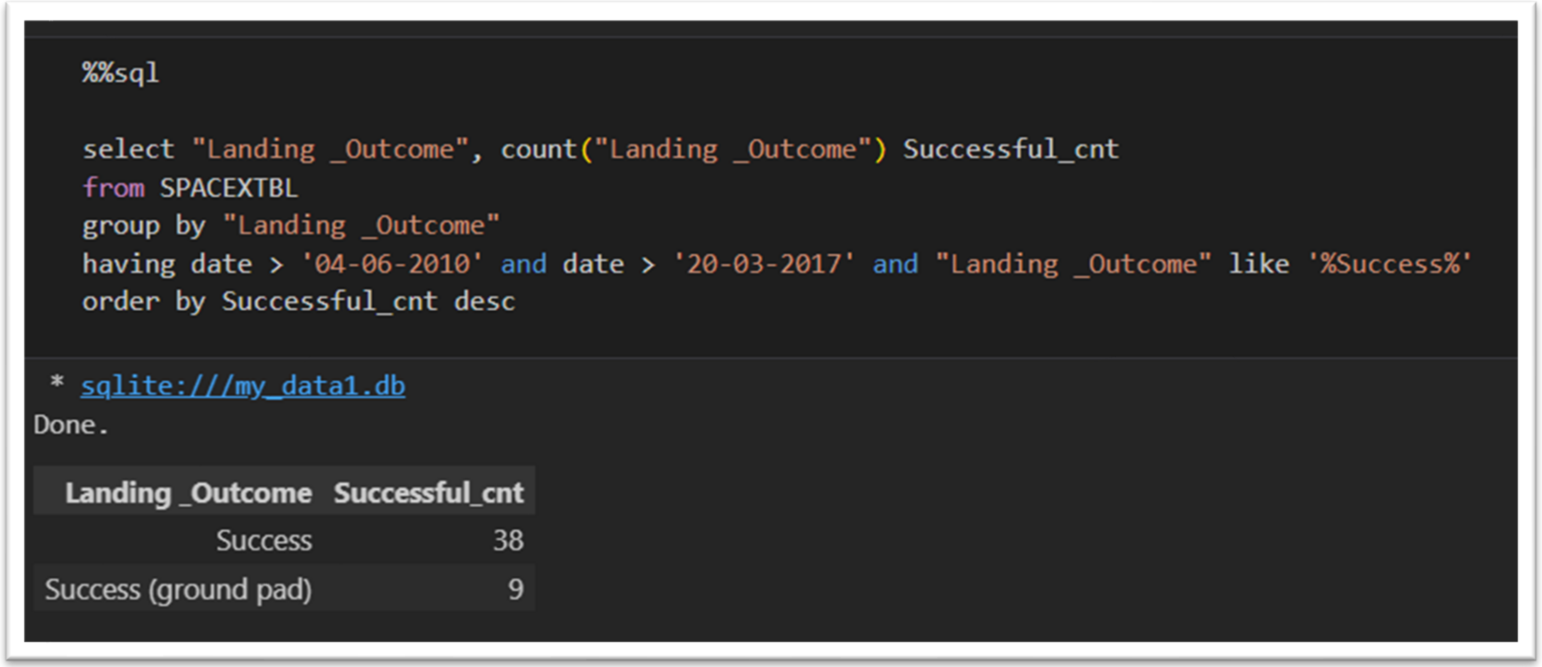


```
%%sql

select substr(Date,4,2) Month, "Landing _Outcome", Booster_Version, Launch_Site
from SPACEXTBL
where substr(Date,7,4) = '2015' and "Landing _Outcome" = 'Failure (drone ship)'
```

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

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



```
%%sql

select "Landing _Outcome", count("Landing _Outcome") Successful_cnt
from SPACEXTBL
group by "Landing _Outcome"
having date > '04-06-2010' and date > '20-03-2017' and "Landing _Outcome" like '%Success%'
order by Successful_cnt desc
```

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

Landing _Outcome	Successful_cnt
Success	38
Success (ground pad)	9

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **HAVING** clause to filter for landing outcomes between 2010-06-04 to 2010-03-20.

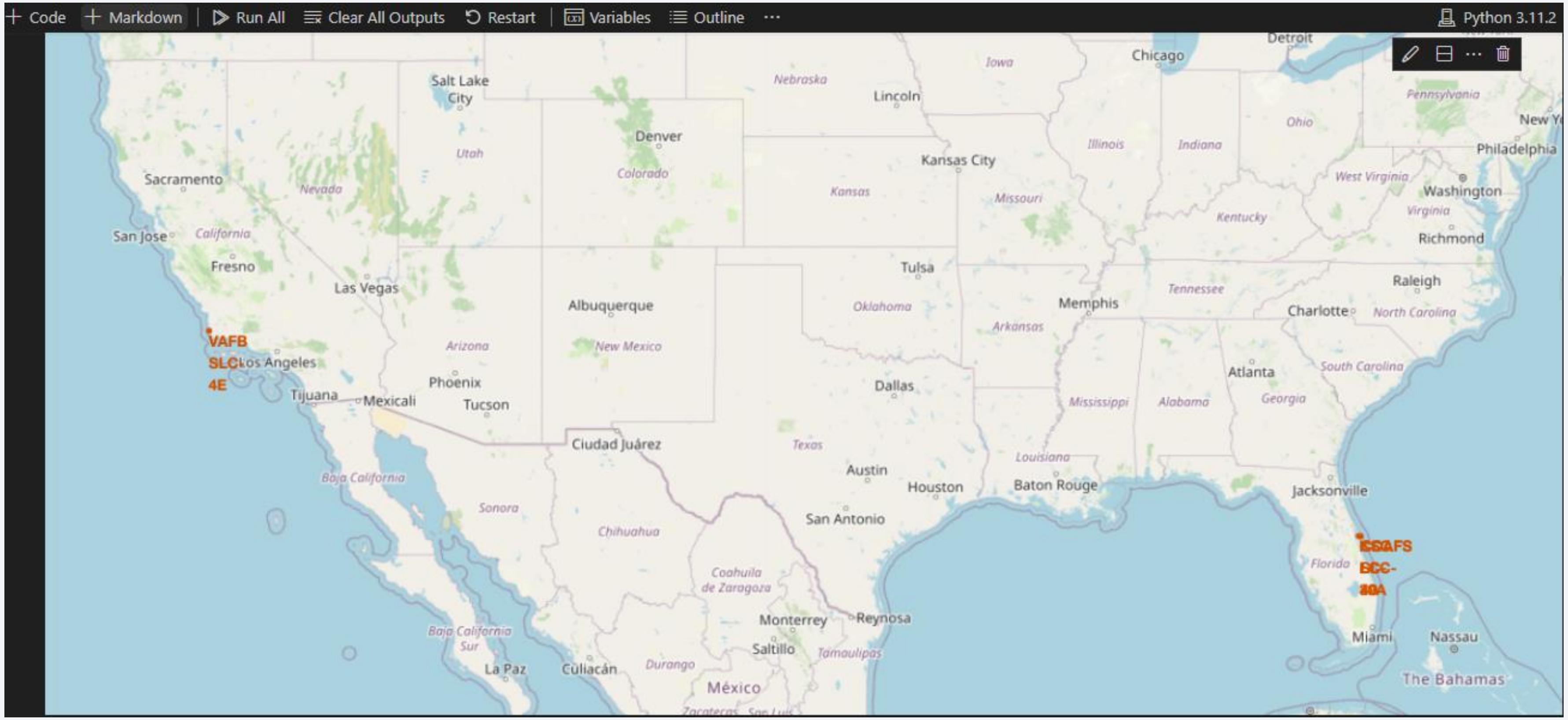
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

Section 3

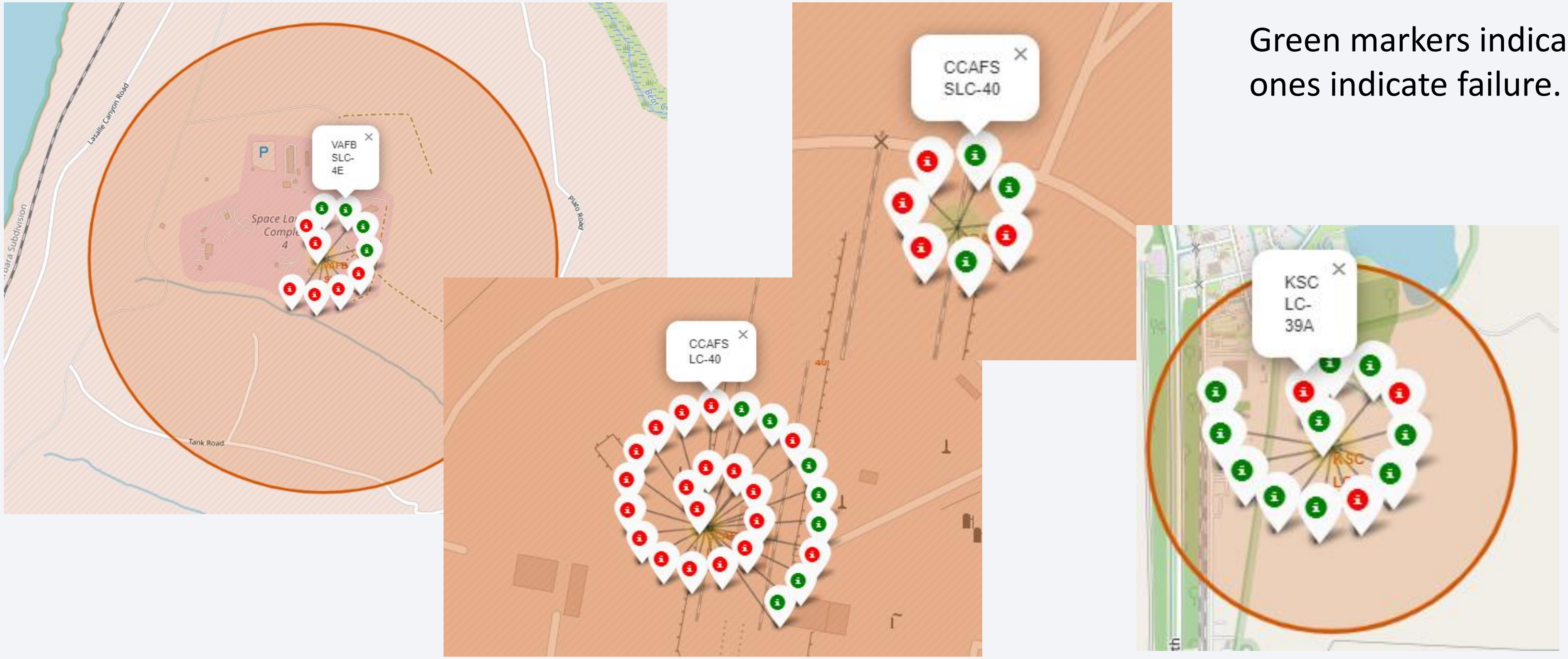
Launch Sites Proximities Analysis

All launch sites

The launch sites are located close to the sea, which is likely for safety reasons.

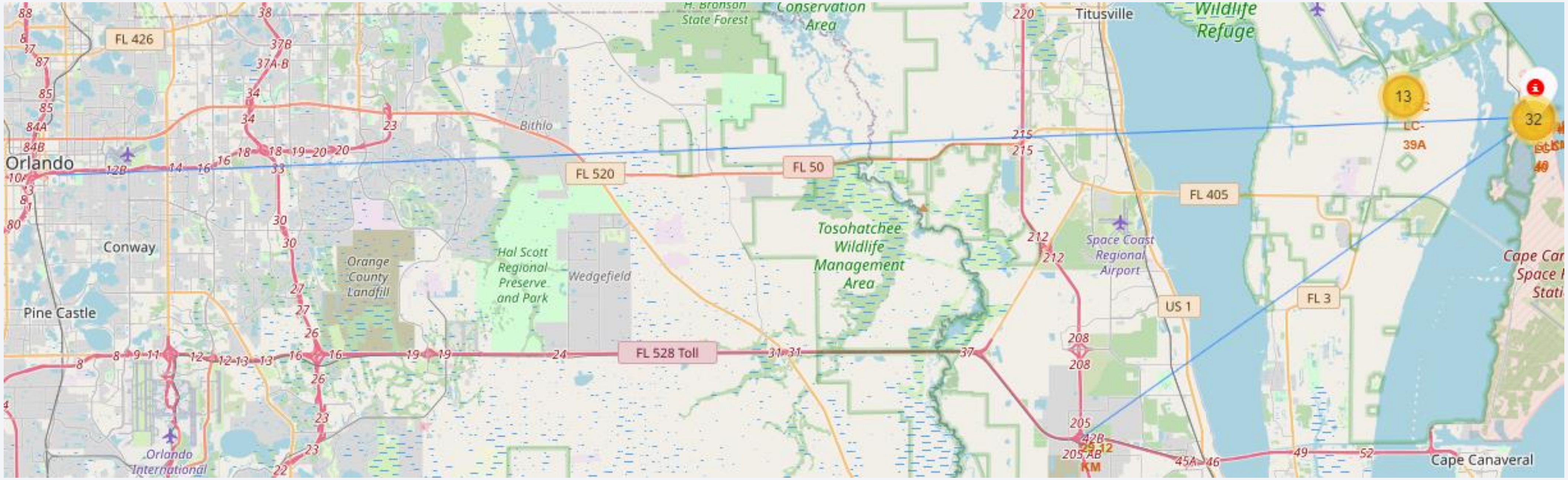


Launch Outcomes by Site

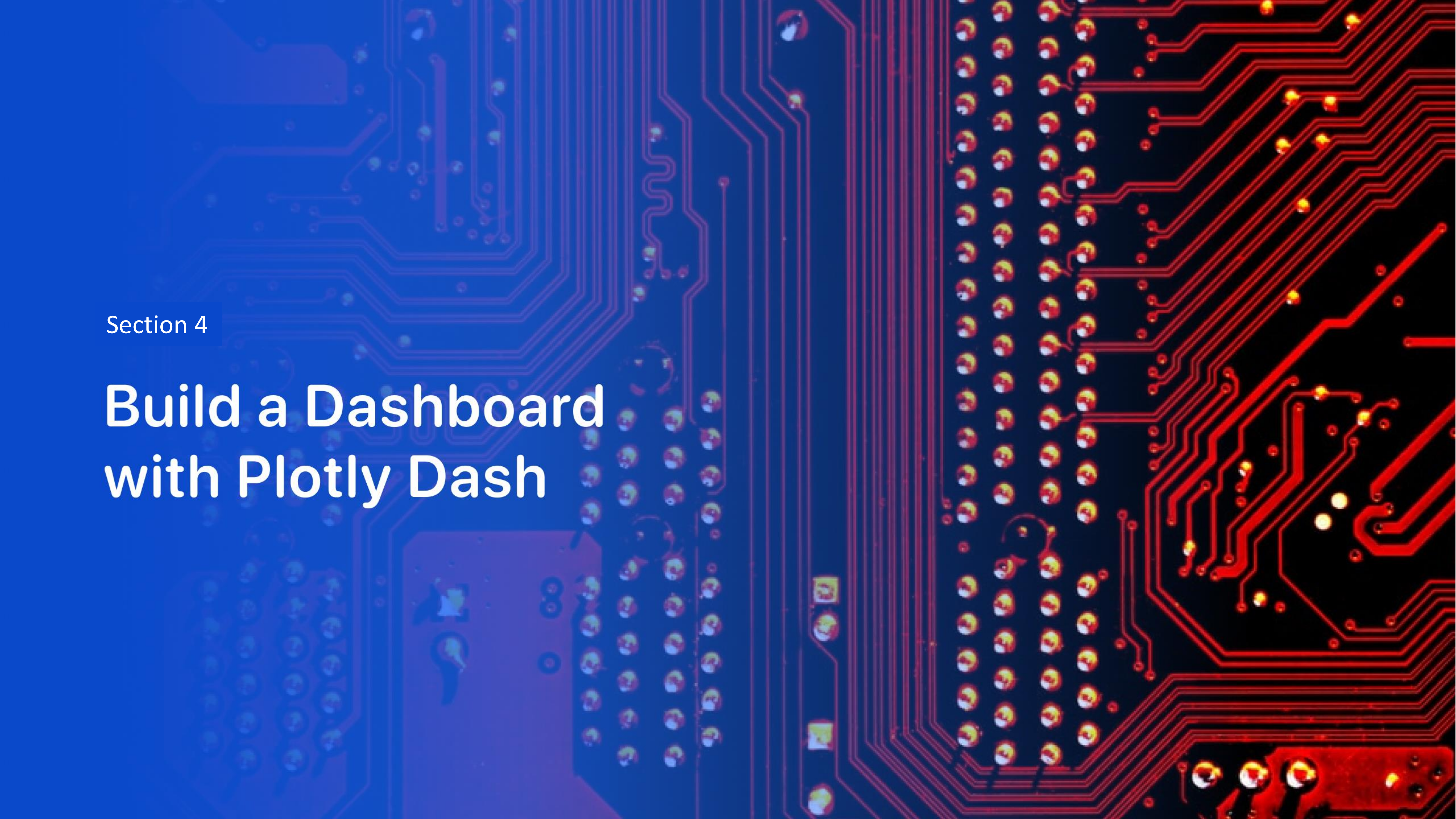


Green markers indicate success and red ones indicate failure.

Logistics



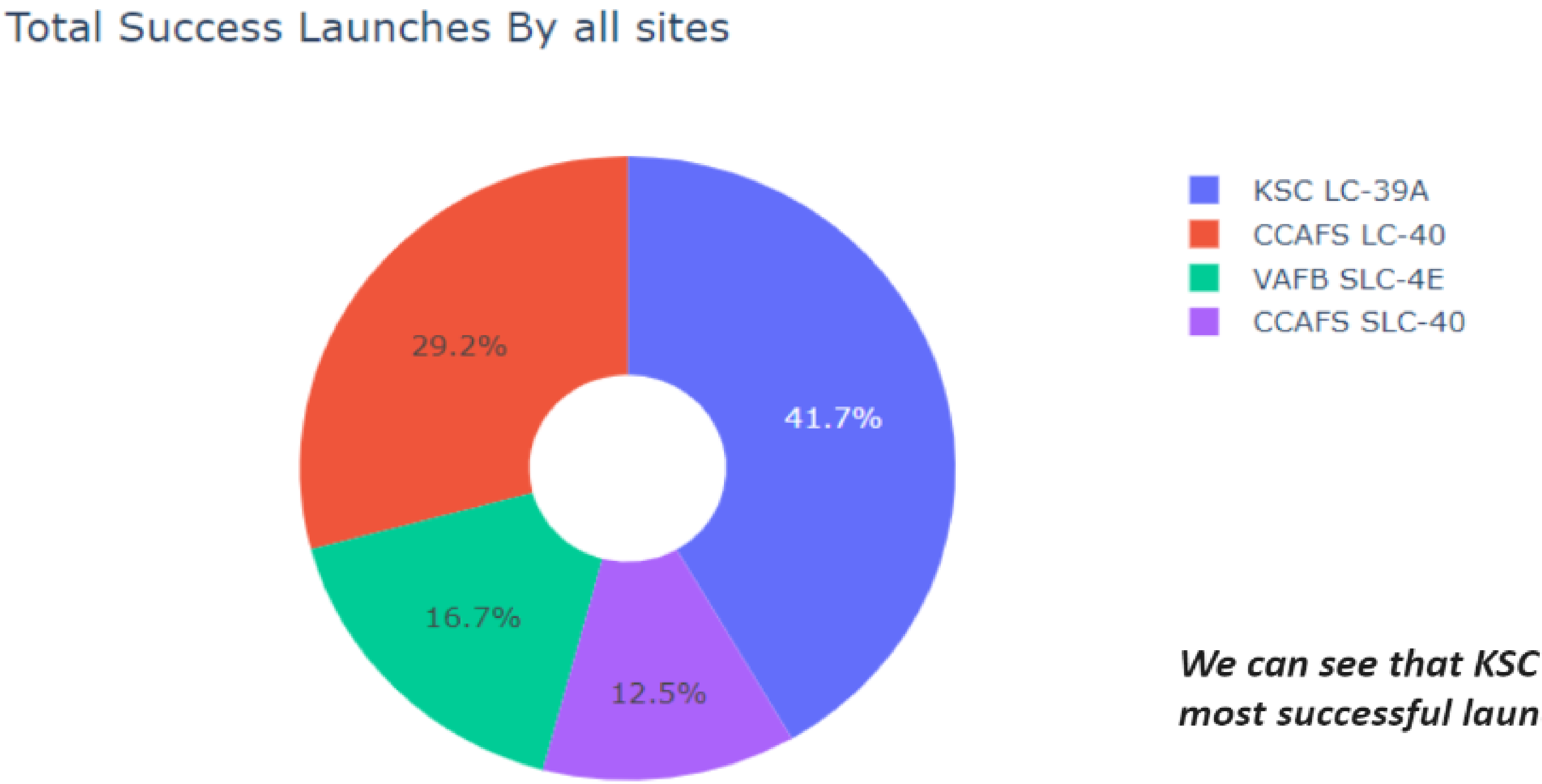
Launch sites have good logistics features, being near railroads and relatively far from habited areas.



Section 4

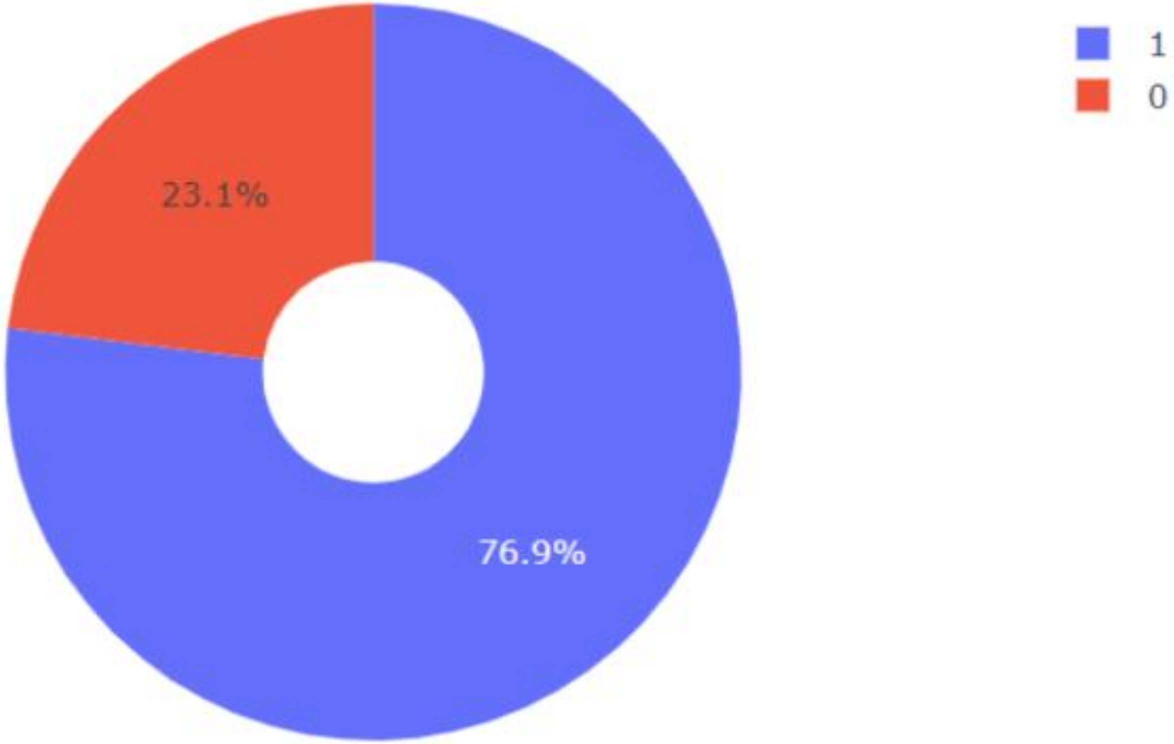
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site


Total Success Launches By all sites

We can see that KSC LC-39A had the most successful launches from all the sites

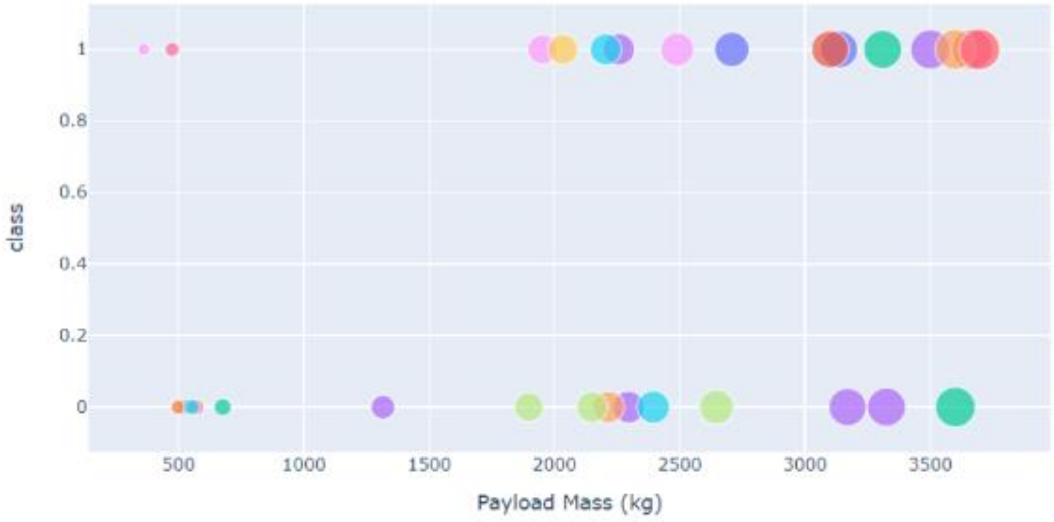
Pie chart showing the Launch site with the highest launch success ratio



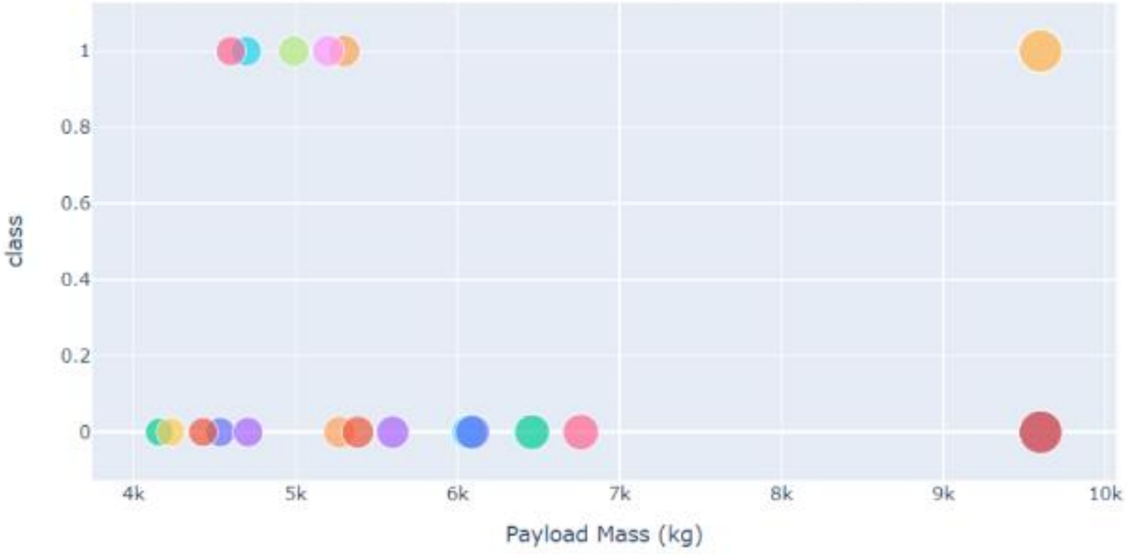
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

Low Weighted Payload 0kg – 4000kg



Heavy Weighted Payload 4000kg – 10000kg

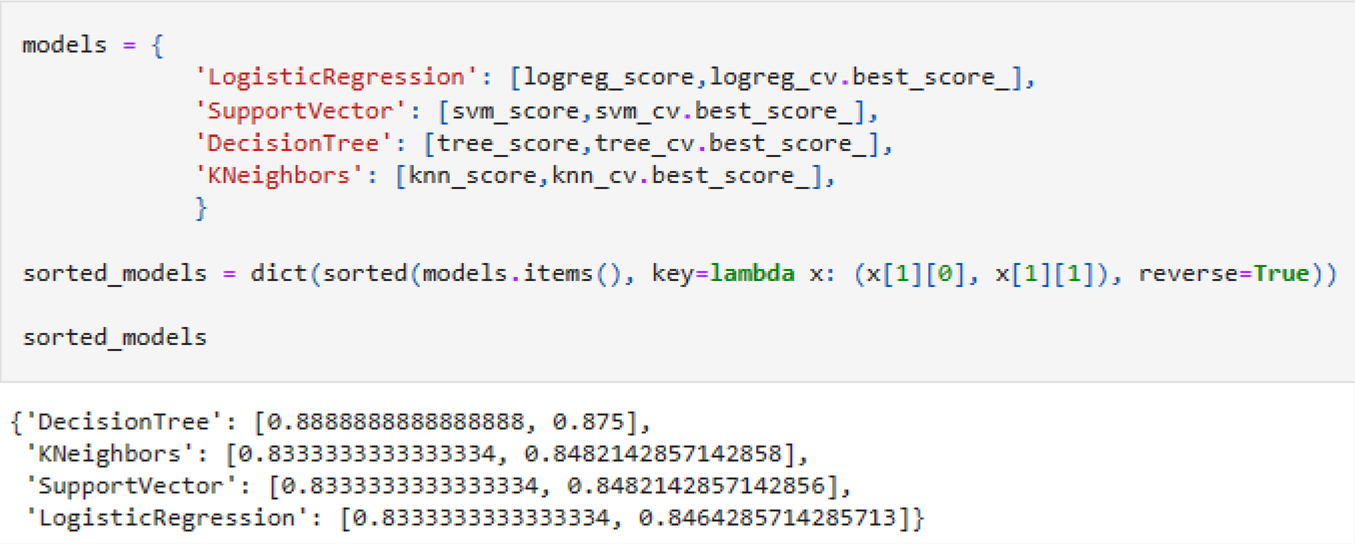


We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy



```
models = {
            'LogisticRegression': [logreg_score,logreg_cv.best_score_],
            'SupportVector': [svm_score,svm_cv.best_score_],
            'DecisionTree': [tree_score,tree_cv.best_score_],
            'KNeighbors': [knn_score,knn_cv.best_score_],
            }

sorted_models = dict(sorted(models.items(), key=lambda x: (x[1][0], x[1][1]), reverse=True))

sorted_models
```

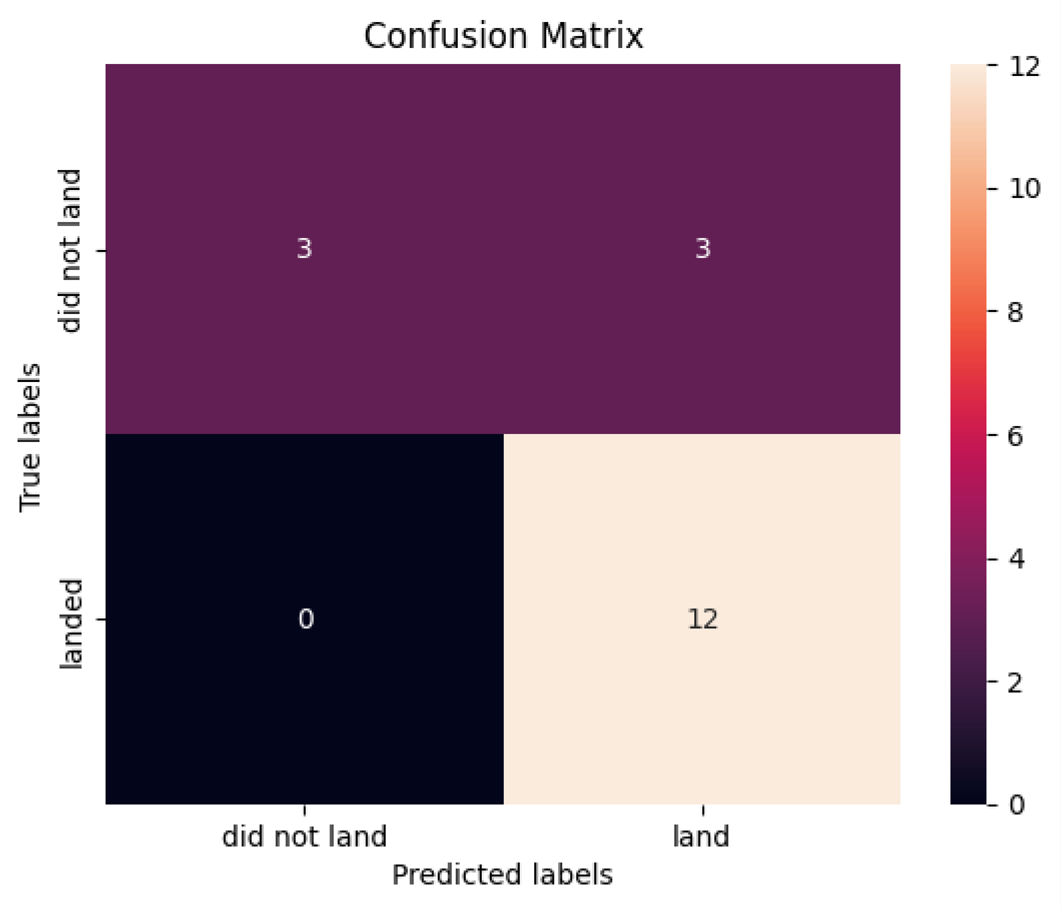
```
{'DecisionTree': [0.8888888888888888, 0.875],
 'KNeighbors': [0.8333333333333334, 0.8482142857142858],
 'SupportVector': [0.8333333333333334, 0.8482142857142856],
 'LogisticRegression': [0.8333333333333334, 0.8464285714285713]}
```

- Four classification models were tested, and their accuracies are shown.

- The model with the highest classification accuracy is the Decision Tree classifier, which has accuracies over 88% in the test set and 87.5% in the train set.

Confusion Matrix

The confusion matrix for the Decision Tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives. i.e., an unsuccessful landing is marked as a successful landing by the classifier.



Conclusions

- The best launch site is KSC LC-39A.
- Launches above 7,000kg are less risky.
- Although most of the mission outcomes are successful, successful landing outcomes seem to improve over time, according to the evolution of processes and rockets.
- Decision Tree Classifier can be used to predict successful landings and increase profits.

Thank you!