IBM Developer
SKILLS NETWORK

IBM Data Science Capstone Project - SpaceX

Winning Space Race with Data Science

Nikhil Tore
25 January 2024

Outline

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

Executive Summary

- **Summary of methodologies**

✓ Data Collection

✓ Data Wrangling

✓ Exploratory Data Analysis with Data Visualization

✓ Exploratory Data Analysis with SQL

✓ Building an Interactive Map with Folium

✓ Building a Dashboard with Plotly Dash

✓ Predictive Analysis (Classification)

- **Summary of all results**

✓ Exploratory Data Analysis Results

✓ Interactive Analytics Demo in screenshots

✓ Predictive Analysis Results

Introduction

- **Project background and context**
  SpaceX stands out as the leading company in the era of commercial space exploration, achieving remarkable success in rendering space travel economically accessible to a wider audience. On its official website, SpaceX promotes Falcon 9 rocket launches at a price point of 62 million dollars, a significantly lower figure compared to other providers whose costs can soar to 165 million dollars per launch. This substantial cost reduction is largely attributed to SpaceX's innovative approach of reusing the first stage of its rockets. Consequently, by determining the likelihood of a successful first stage landing, we can ascertain the overall launch cost. Utilizing machine learning and leveraging publicly available information, we aim to develop a predictive model to assess whether SpaceX will opt to reuse the first stage in upcoming launches.
- **Problems you want to find answers**

✓ In what ways do factors such as payload mass, launch site, number of flights, and orbits influence the likelihood of a successful first stage landing?

✓ Is there an upward trend in the success rate of landings over the years?

✓ Which classification algorithm is most suitable for this scenario?

Section 1

Methodology

Methodology

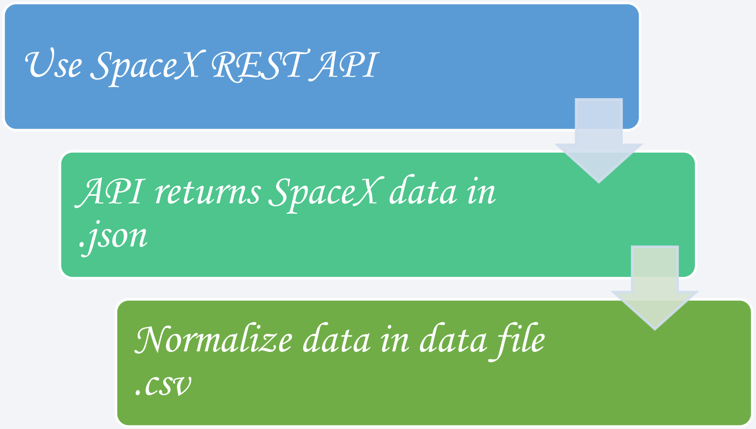
Executive Summary

- *Data collection methodology:*
  - ✓ *Using SpaceX Rest API.*
  - ✓ *Web Scraping from Wikipedia.*
- *Performed data wrangling*
  - ✓ *One Hot Encoding data fields for Machine Learning.*
- *Performed exploratory data analysis (EDA) using visualization and SQL*
  - ✓ *Plotted Scatter plots, Bar plots to show relationship between variables and to show patters of data.*
- *Performed interactive visual analytics using Folium and Plotly Dash*
- *Performed predictive analysis using classification models*
  - ✓ *How to build, tune, evaluate classification models*

Data Collection

- *The data collection process encompassed a dual approach, combining API requests from SpaceX's REST API with web scraping of a table on SpaceX's Wikipedia page. The utilization of both these methods was necessary to obtain comprehensive information about launches for a more thorough analysis.*
- *The REST API and Web Scraping yielded data on launches, encompassing details such as the rocket utilized, payload delivered, launch specifications, landing specifications, and the outcome of the landing.*

SpaceX REST API

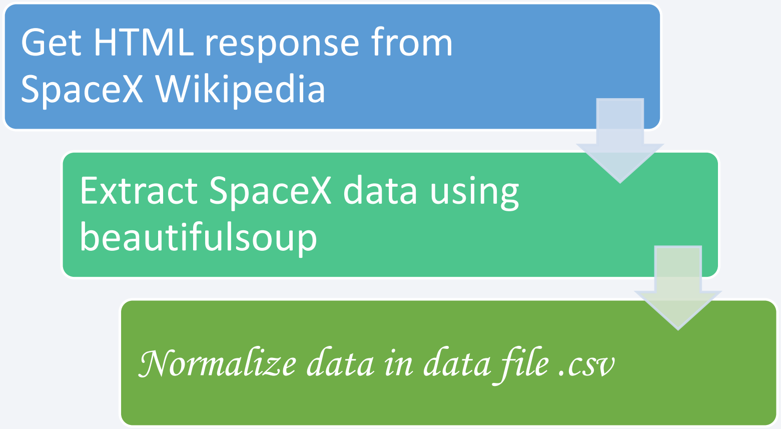


Use SpaceX REST API

API returns SpaceX data in .json

Normalize data in data file .csv

Web Scraping



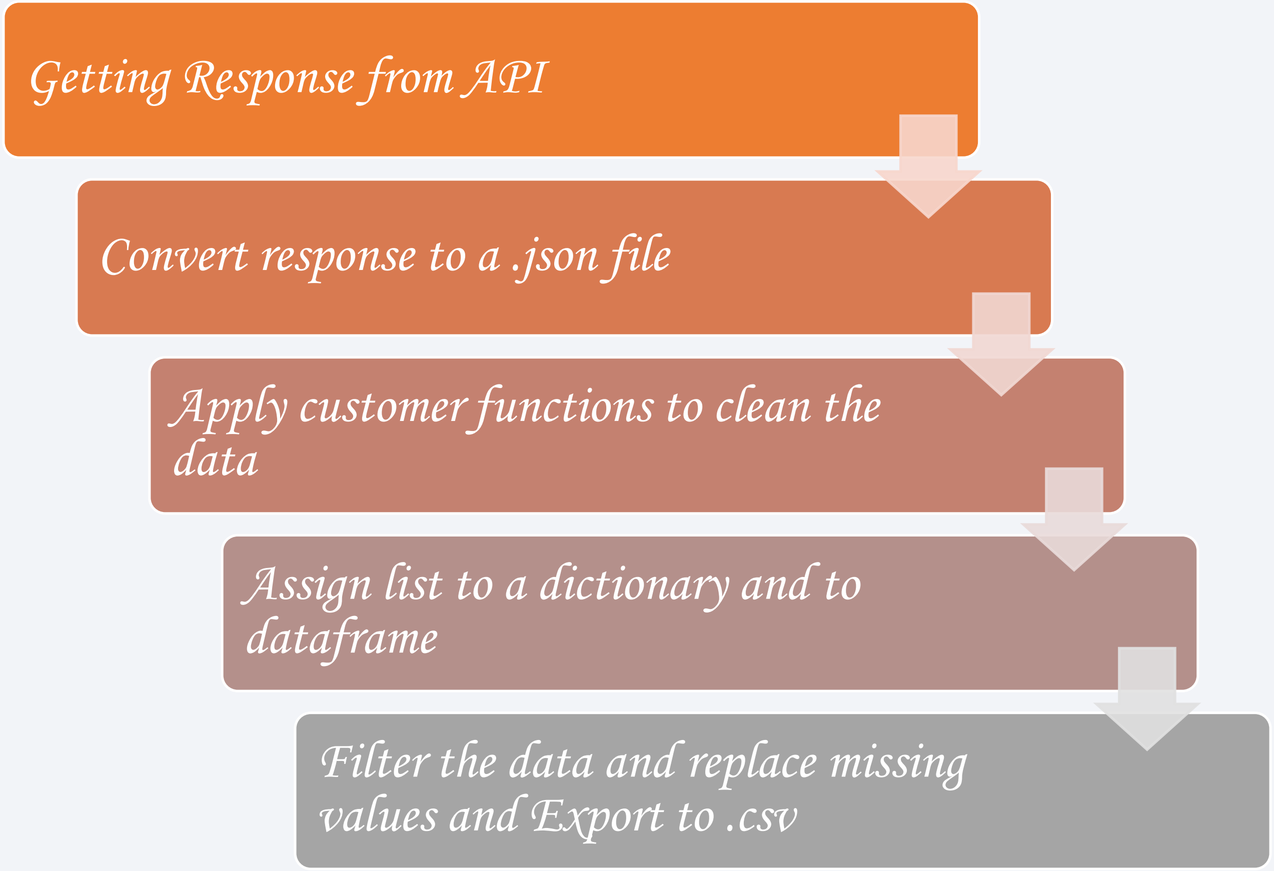
Get HTML response from SpaceX Wikipedia

Extract SpaceX data using beautifulsoup

Normalize data in data file .csv

Data Collection – SpaceX API

- Requesting Rocket Launch data from SpaceX API
- Converting the response content using .json() and normalizing it
- Using custom functions to clean the data to get the required information
- Constructing data into a dictionary and then into a dataframe
- Filtering data to obtain Falcon 9 launches, replacing the missing values
- Exporting the data into .csv
- https://github.com/nikhiltore/IBM-Data-Science-Capstone-Project---Final/blob/main/1.%20Spacex-Data-Collection-API.ipynb



Getting Response from API

Convert response to a .json file

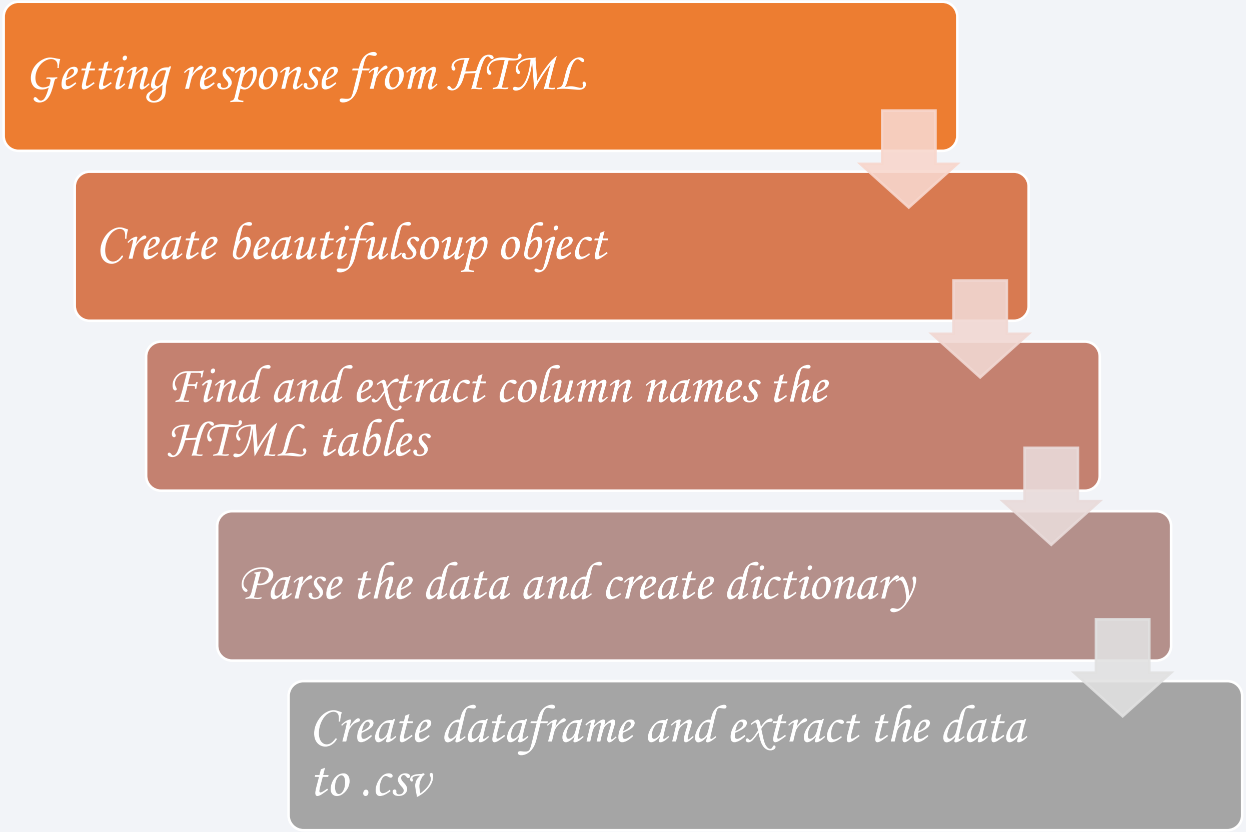
Apply customer functions to clean the data

Assign list to a dictionary and to dataframe

Filter the data and replace missing values and Export to .csv

Data Collection - Scraping

- Requesting Falcon 9 launch data from Wikipedia
- Creating beautifulsoup object from the HTML response
- Finding and extracting column names from the HTML tables
- Collecting the data by parsing the HTML tables
- Creating dictionary from the data collected
- Creating a dataframe from the dictionary
- Exporting the data to .csv
- https://github.com/nikhiltore/IBM-Data-Science-Capstone-Project---Final/blob/main/2.%20Spacex-webscraping.ipynb



Getting response from HTML

Create beautifulsoup object

Find and extract column names the HTML tables

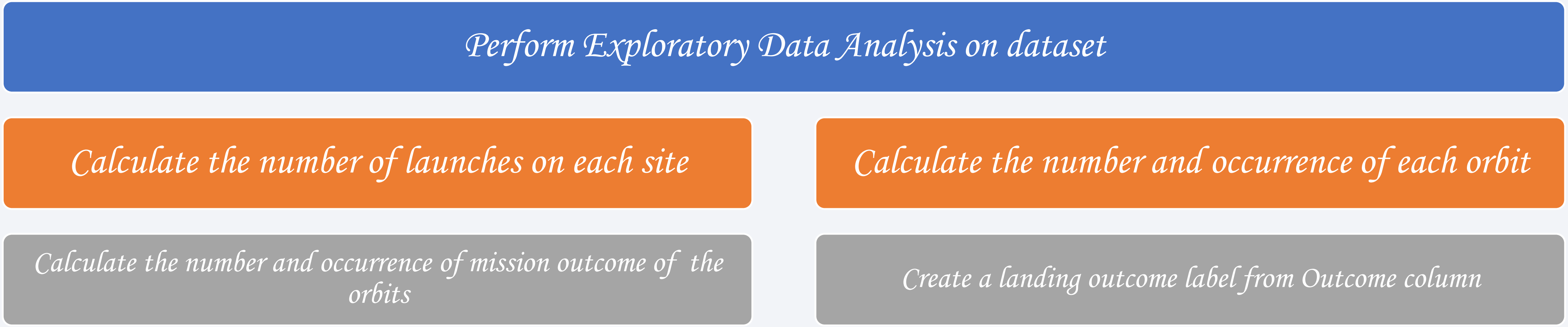
Parse the data and create dictionary

Create dataframe and extract the data to .csv

Data Wrangling

Introduction

- ✓ In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship. We will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.
- ✓ https://github.com/nikhiltore/IBM-Data-Science-Capstone-Project---Final/blob/main/3.%20Spacex-Data%20wrangling.ipynb



Perform Exploratory Data Analysis on dataset

Calculate the number of launches on each site

Calculate the number and occurrence of each orbit

Calculate the number and occurrence of mission outcome of the orbits

Create a landing outcome label from Outcome column

EDA with Data Visualization

- ***Scatter Plots:*** *Scatter plots show how much one variable is affected by another. The relationship between two variables is called correlation. If a relationship exists, they could be used in Machine learning model.*

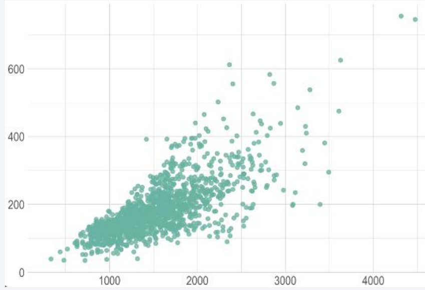
✓ *Flight Number vs. Payload Mass*

✓ *Flight Number vs. Launch Site*

✓ *Payload Mass vs. Launch Site*

✓ *Flight Number vs. Orbit*

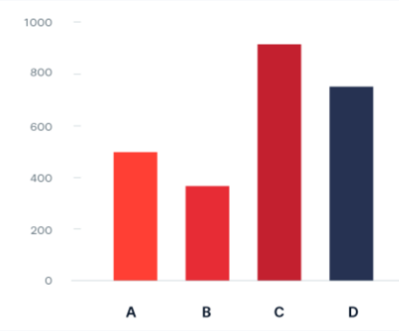
✓ *Orbit vs. Payload Mass*



- *https://github.com/nikhiltore/IBM-Data-Science-Capstone-Project---Final/blob/main/5.%20EDA-dataviz.ipynb*

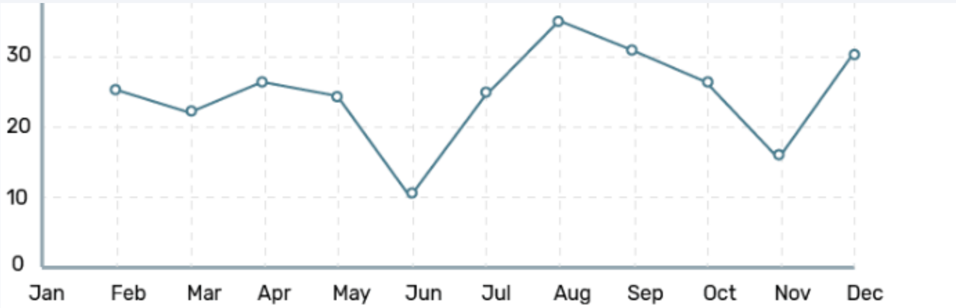
- ***Bar Plots:*** *Bar plots show comparisons among discrete categories. The goal is to show the relationship between the specific categories being compared and a measured value.*

✓ *Success Rate vs. Orbit Type*



- ***Line Plots:*** *Line plots show trends in data over time*

✓ *Success Rate vs. Year*



EDA with SQL

Performed SQL queries to gather information about the dataset:

- ✓ *Displaying the names of the unique launch sites in the space mission*
- ✓ *Displaying 5 records where launch sites begin with the string 'CCA'.*
- ✓ *Displaying the total payload mass carried by boosters launched by NASA (CRS).*
- ✓ *Displaying average payload mass carried by booster version F9 v1.1*
- ✓ *Listing the date when the first successful landing outcome in ground pad was achieved.*
- ✓ *Listing the names of the boosters, which have success in drone ship and have payload mass greater than 4000 but less than 6000.*
- ✓ *Listing the total number of successful and failure mission outcomes.*
- ✓ *Listing the names of the booster versions, which have carried the maximum payload mass.*
- ✓ *Listing the records, which will display the month names, failure-landing outcomes in drone ship, booster versions, launch site for the months in year 2015.*
- ✓ *Ranking 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.*
- ✓ *https://github.com/nikhiltore/IBM-Data-Science-Capstone-Project---Final/blob/main/4.%20EDA-sql.ipynb*

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 coordinates.*

✓ *Added markers with Circle, Popup label and text label of all launch sites using their coordinates to show their geographical locations and proximity to equator and coasts.*

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

✓ *Added colored markers of success (green) and failed (red) launches using marker cluster to identify which launch sites have relatively high success rates.*

✓ *Added colored lines to show distances between the launch site and its proximity like railway, highway, coastline, etc.*

- *https://github.com/nikhiltore/IBM-Data-Science-Capstone-Project---Final/blob/main/6.%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb*

Build a Dashboard with Plotly Dash

- The Interactive Dashboard was built using Plotly Dash.

✓ Added a dropdown list to enable Launch Site selection.

✓ Added a Pie Chart showing Success launches of All Sites / Certain Site i.e. it shows the total successful launches count for all sites and the success vs. failed counts for the site, if a specific launch site is selected.

✓ Slider to select Payload range

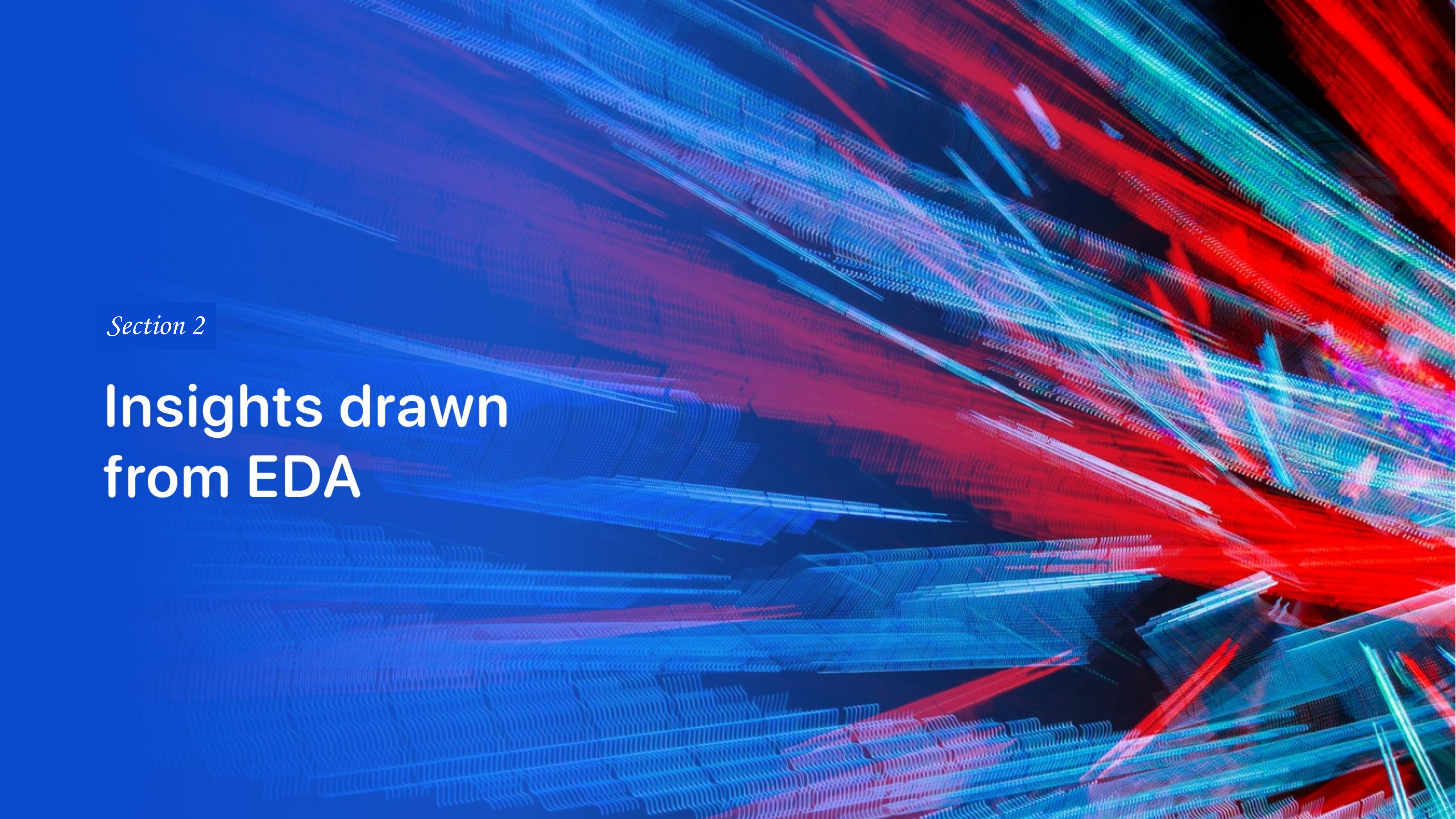
✓ Scatter plot showing the correlation between Payload and Launch Outcome for different Booster versions.

✓ https://github.com/nikhiltore/IBM-Data-Science-Capstone-Project---Final/blob/main/7.%20Spacex_dash_app.py

Predictive Analysis (Classification)

- Load the dataset into NumPy and Pandas.
- Standardizing the data with StandardScaler, fitting it and transforming it.
- Splitting the data into training and testing sets with train_test_split function.
- Creating GridSearchCV object and setting the parameters.
- Applying GridSearchCV on LogReg, SVM, Decision Tree and KNN Models.
- Calculating the accuracy on the test data using the method .score() for all the models.
- Plotting and examining the Confusion matrix for all models.
- Finding the best performing classification model by examining the jaccard_score and F1_score metrics.
- https://github.com/nikhiltore/IBM-Data-Science-Capstone-Project---Final/blob/main/8.%20SpaceX_Machine_Learning_Prediction.ipynb

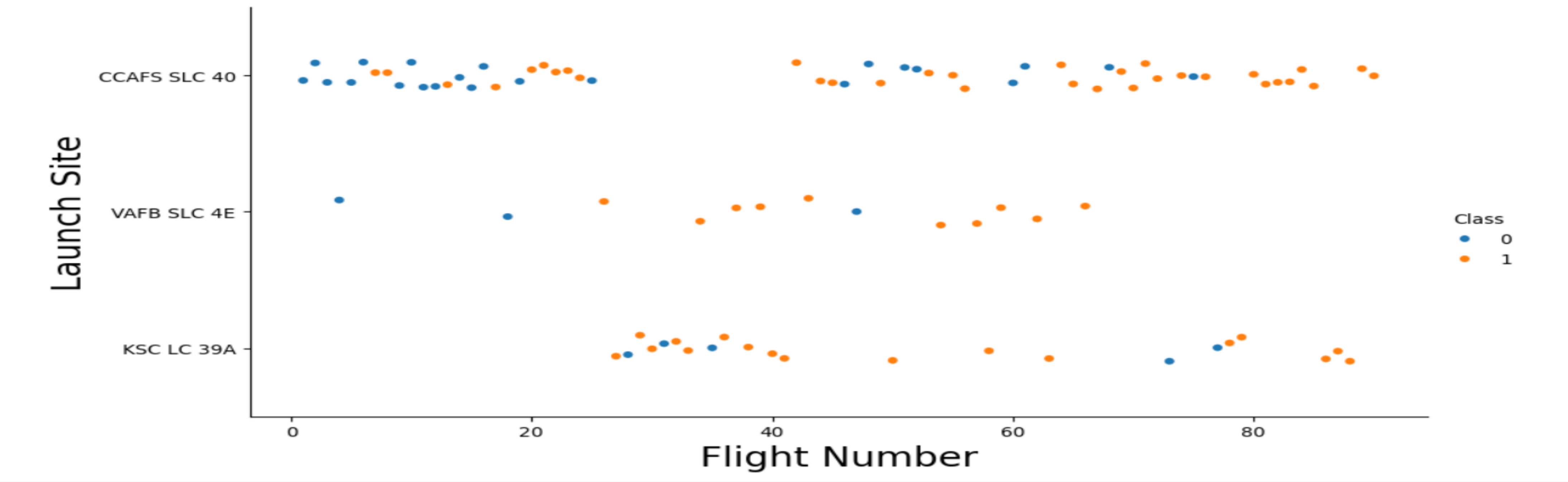
- *Exploratory data analysis results*
- *Interactive analytics demo in screenshots*
- *Predictive analysis results*



Section 2

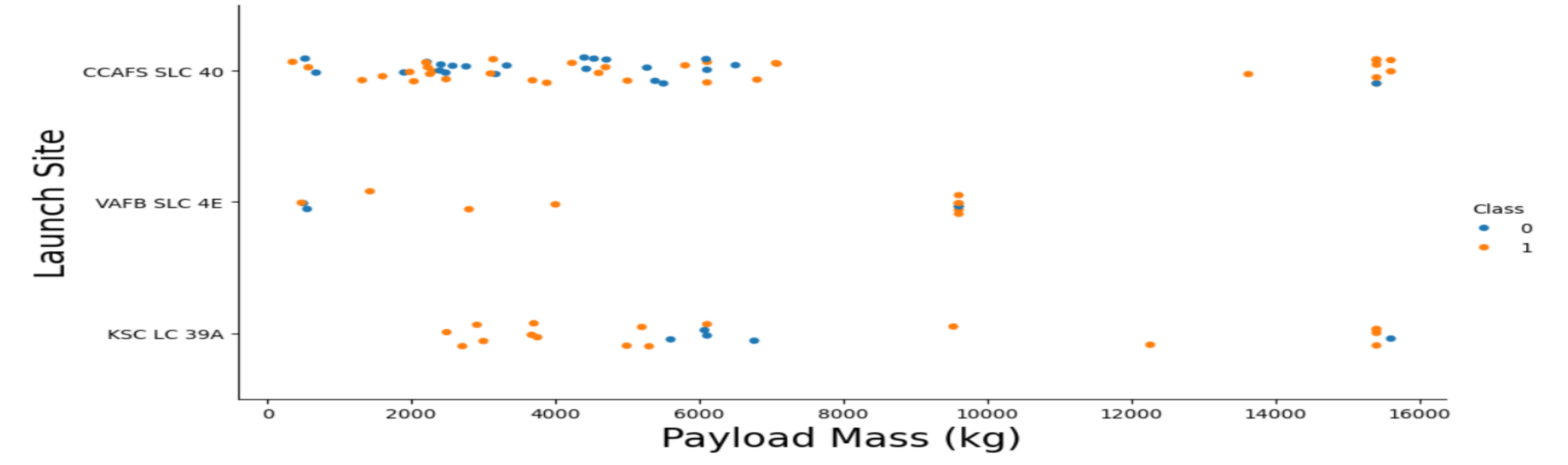
Insights drawn from EDA

Flight Number vs. Launch Site



- ✓ The more amount of flights at a launch site the greater the success rate at a launch site.
- ✓ The earliest flights all failed while the latest flights all succeeded.
- ✓ The CCAFS SLC 40 launch site has about a half of all the launches.
- ✓ VAFB SLC 4E and KSC LC 39A have higher success rates.
- ✓ It can be assumed that each new launch has a higher rate of success.

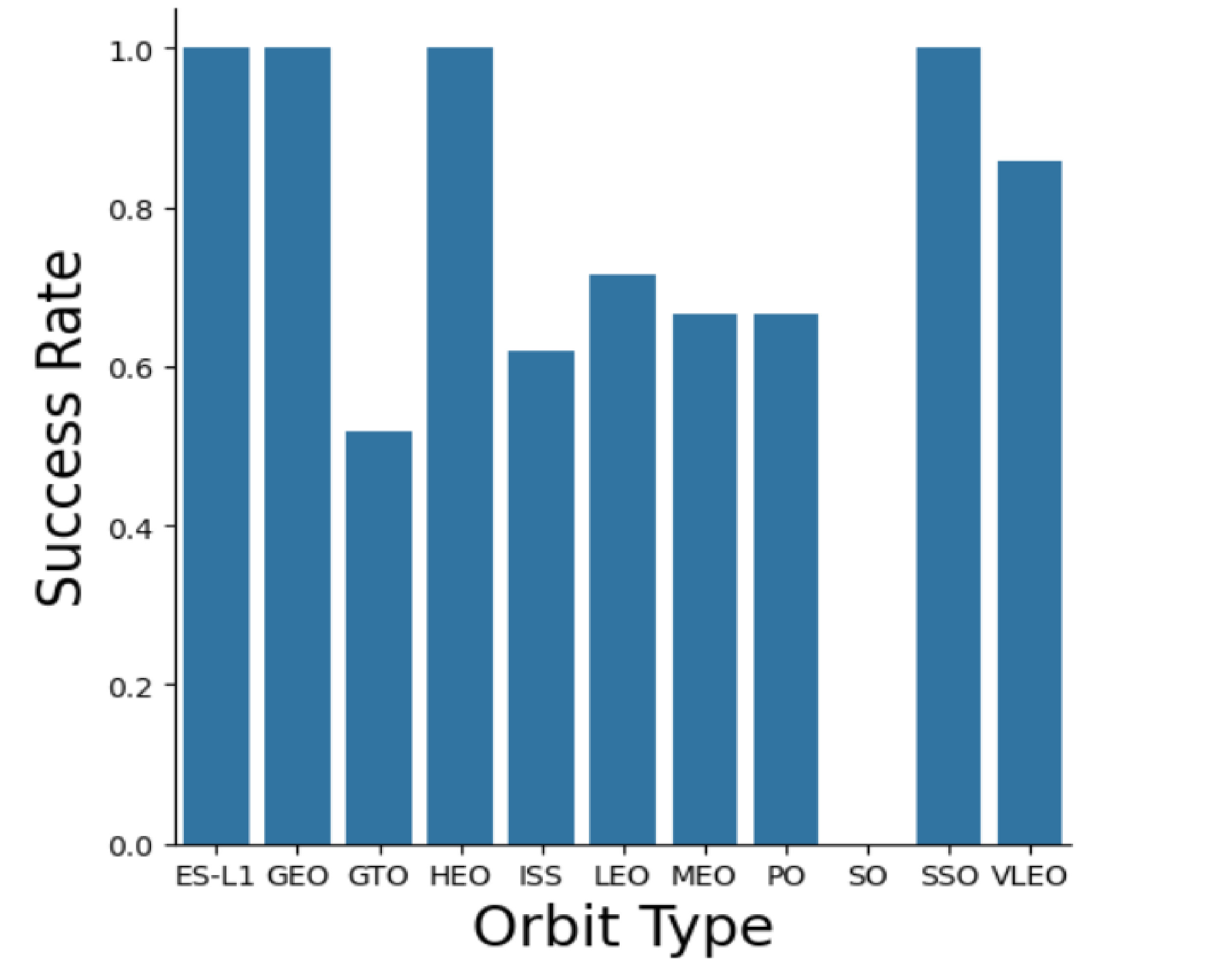
Payload vs. Launch Site



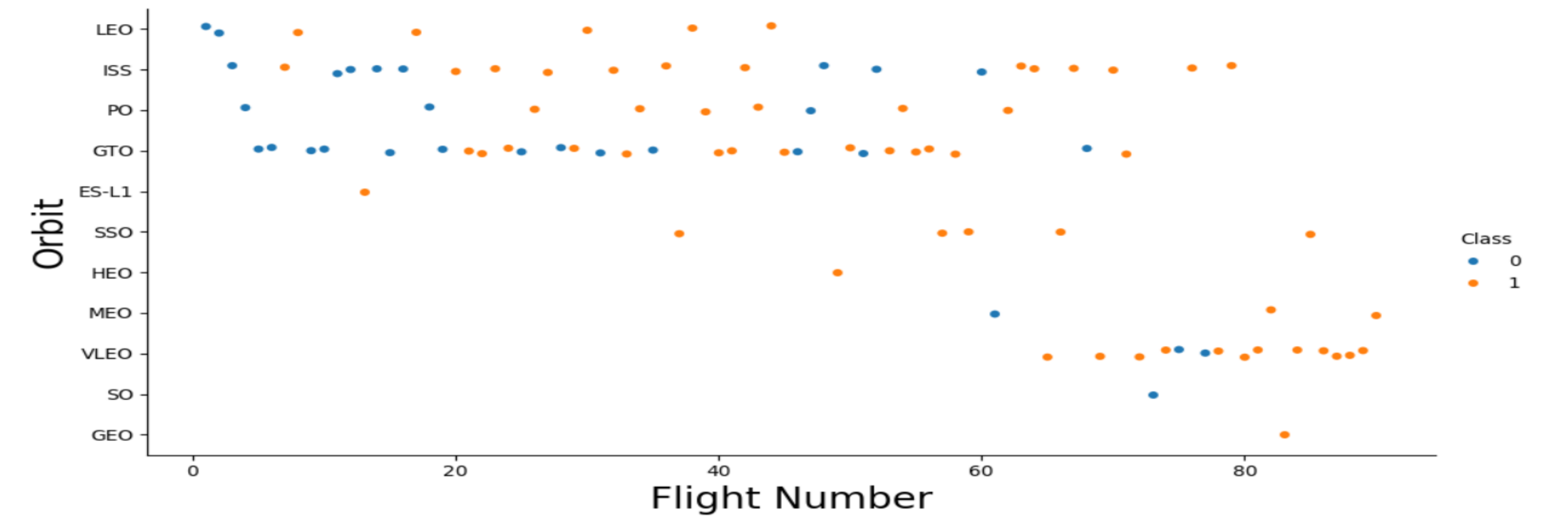
- ✓ *The greater the Payload mass for launch site CCAFS SLC 40, higher the success rate for the rocket.*
- ✓ *Most of the launches with Payload mass over 7000 kg were successful.*
- ✓ *KSC LC 39A has a 100% success rate for Payload mass under 5500 kg.*

Success Rate vs. Orbit Type

- ✓ *Orbits ES-L1, GEO, HEO, and SSO have the best Success rate of 100%.*
- ✓ *Orbit SO has the least Success rate of 0%.*
- ✓ *Orbits GTO, ISS, LEO, MEO, PO and VLEO have Success rate between 50% and 85%.*

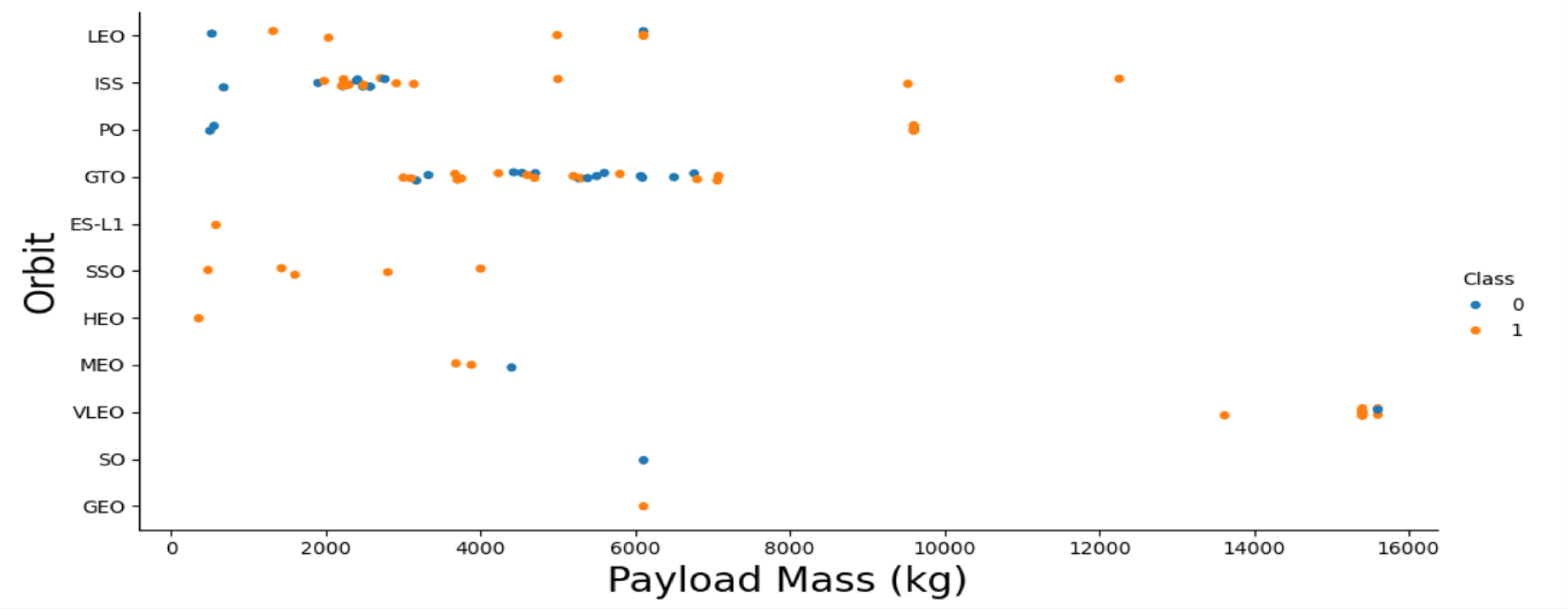


Flight Number vs. Orbit Type



- ✓ *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.*

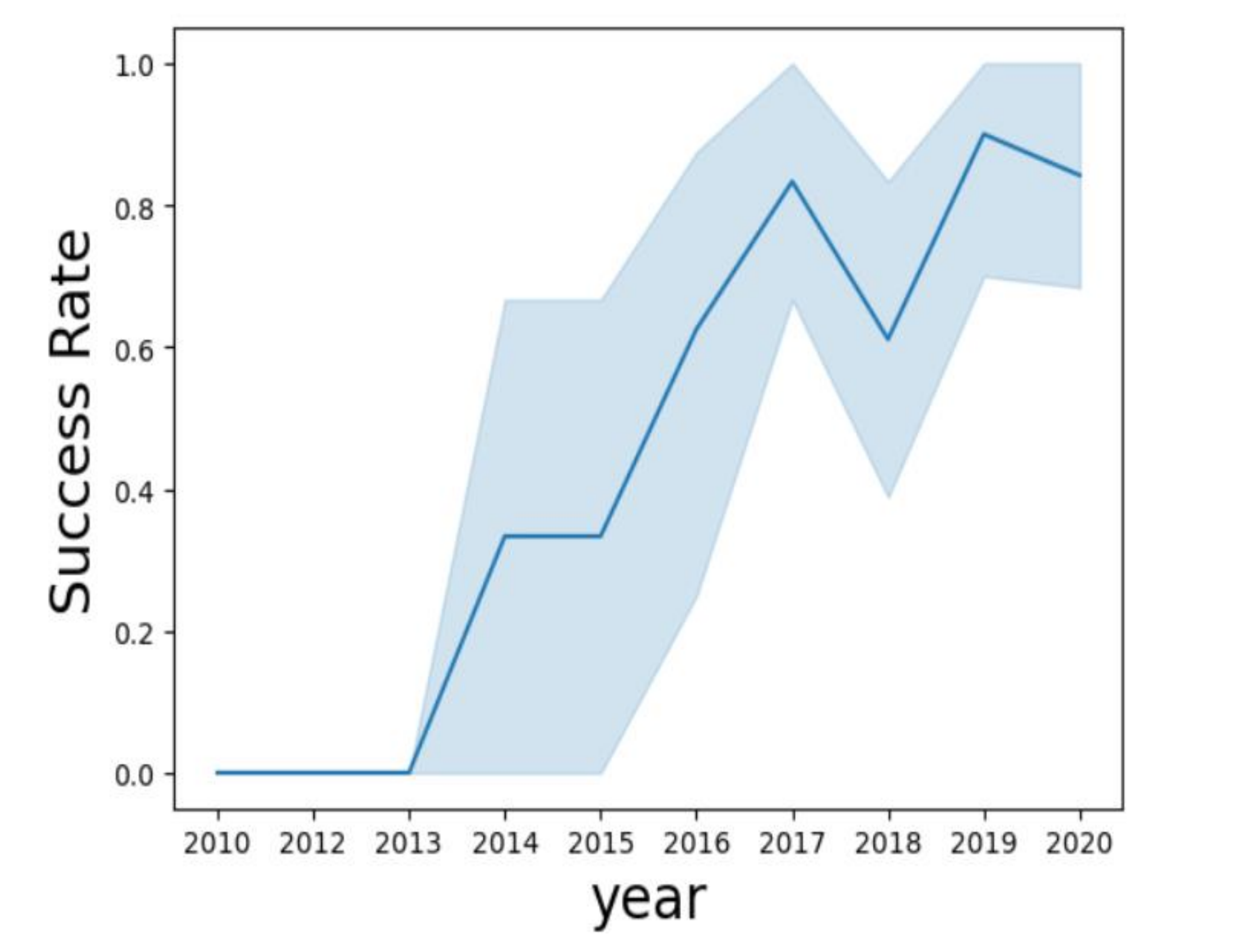
Payload vs. Orbit Type



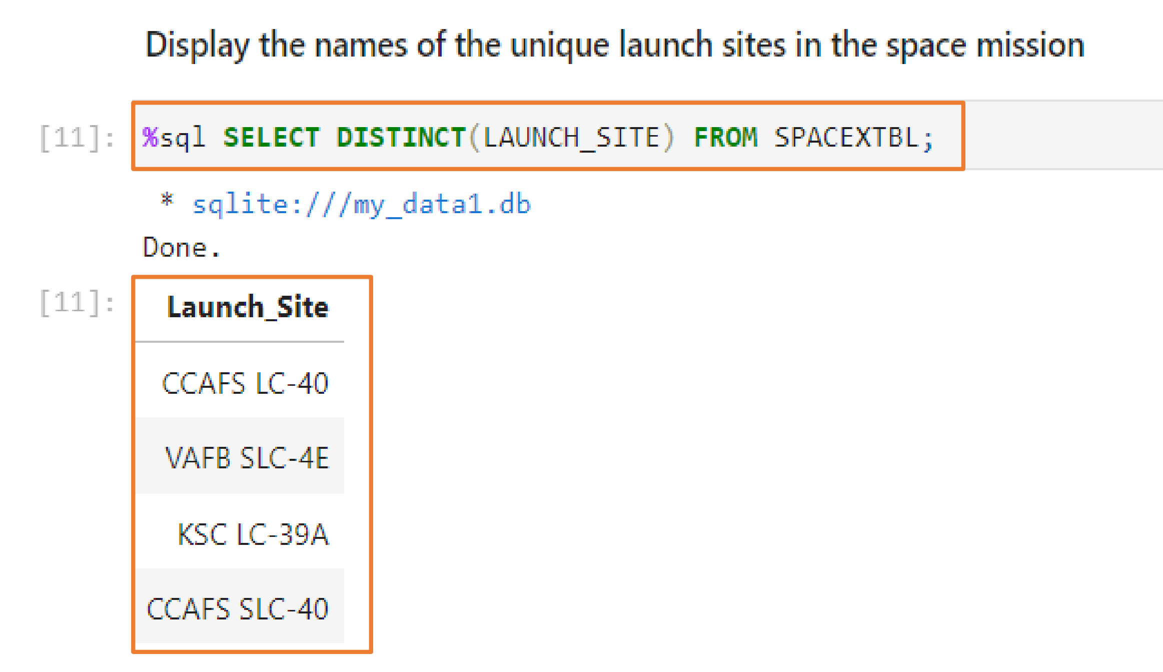
✓ *Heavy Payloads have a negative influence on GTO Orbits and positive influence on ISS Orbits.*

Launch Success Yearly Trend

- ✓ *Success Rate had an increase from the year 2013, saw a slight dip in the year 2018 and then went on to increase till the year 2020.*



All Launch Site Names



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

```
[11]: %sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;
```

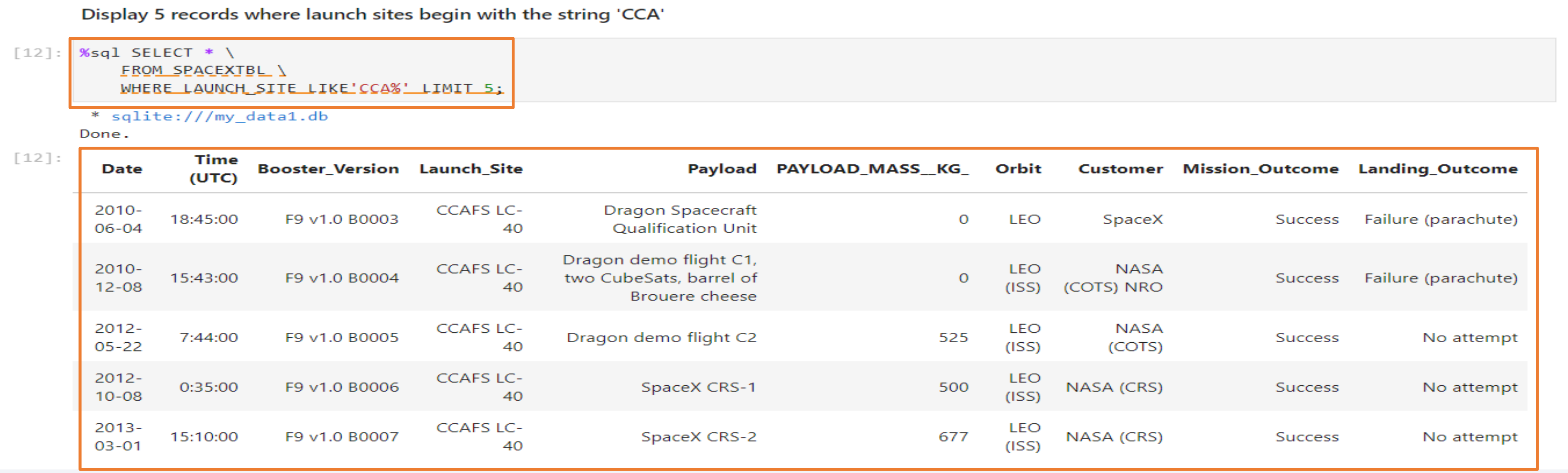
```
 * sqlite:///my_data1.db
Done.
```

[11]:

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

- ✓ *Using the clause DISTINCT in the query means that it will only show Unique values in the Launch_Site column from SpaceXtbl*

Launch Site Names Begin with 'CCA'



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

```
[12]: %sql SELECT * \
          FROM SPACEXTBL \
          WHERE LAUNCH_SITE LIKE'CCA%' LIMIT 5;
```

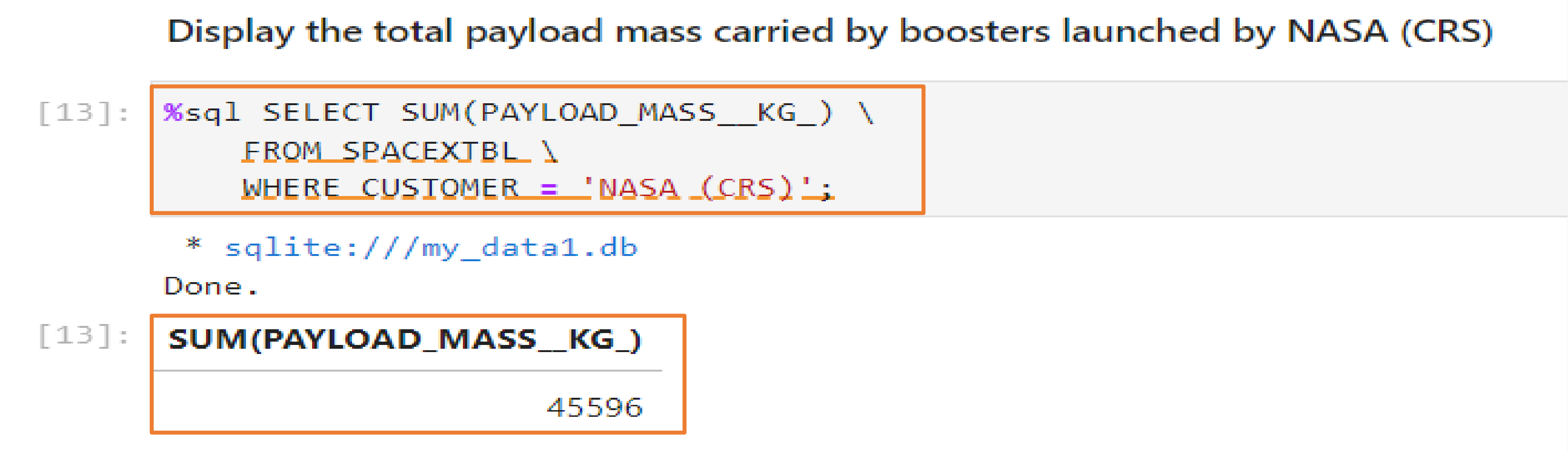
```
 * sqlite:///my_data1.db
Done.
```

[12]:

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

- ✓ *Using the LIMIT 5 clause, query will only show the Top 5 records from SpaceXtbl and LIKE keyword is a wild card with the word "CCA%", where % sign in the end suggests that Launch_Site name must start with 'CCA'.*

Total Payload Mass



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

```
[13]: %sql SELECT SUM(PAYLOAD_MASS__KG_) \
          FROM SPACEXTBL \
          WHERE CUSTOMER = 'NASA (CRS)';
```

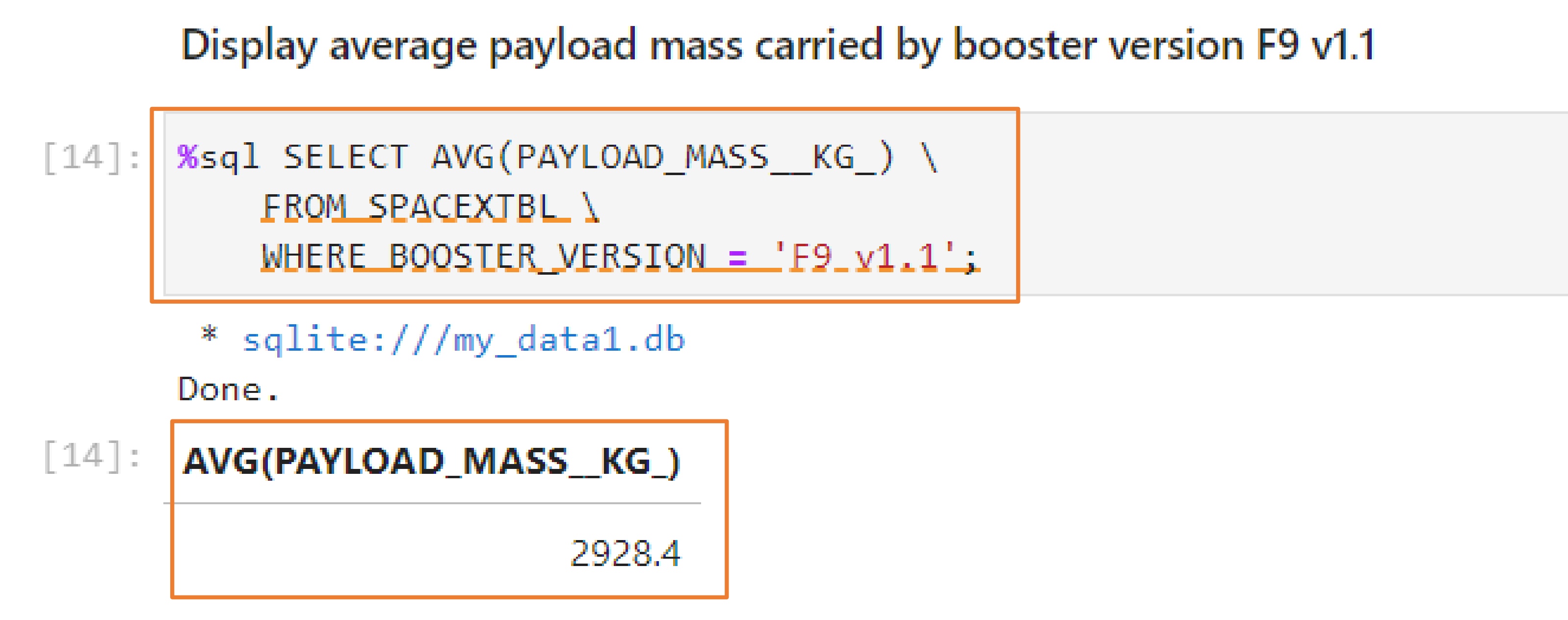
```
 * sqlite:///my_data1.db
Done.
```

[13]:

SUM(PAYLOAD_MASS__KG_)
45596

✓ *Using the aggregate SUM function, gives the total of the column Payload_Mass_kg_ from SpaceXtbl, and the WHERE clause filters the dataset to only perform summation on Customer NASA (CRS).*

Average Payload Mass by F9 v1.1



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

```
[14]: %sql SELECT AVG(PAYLOAD_MASS__KG_) \
          FROM SPACEXTBL \
          WHERE BOOSTER_VERSION = 'F9 v1.1';
```

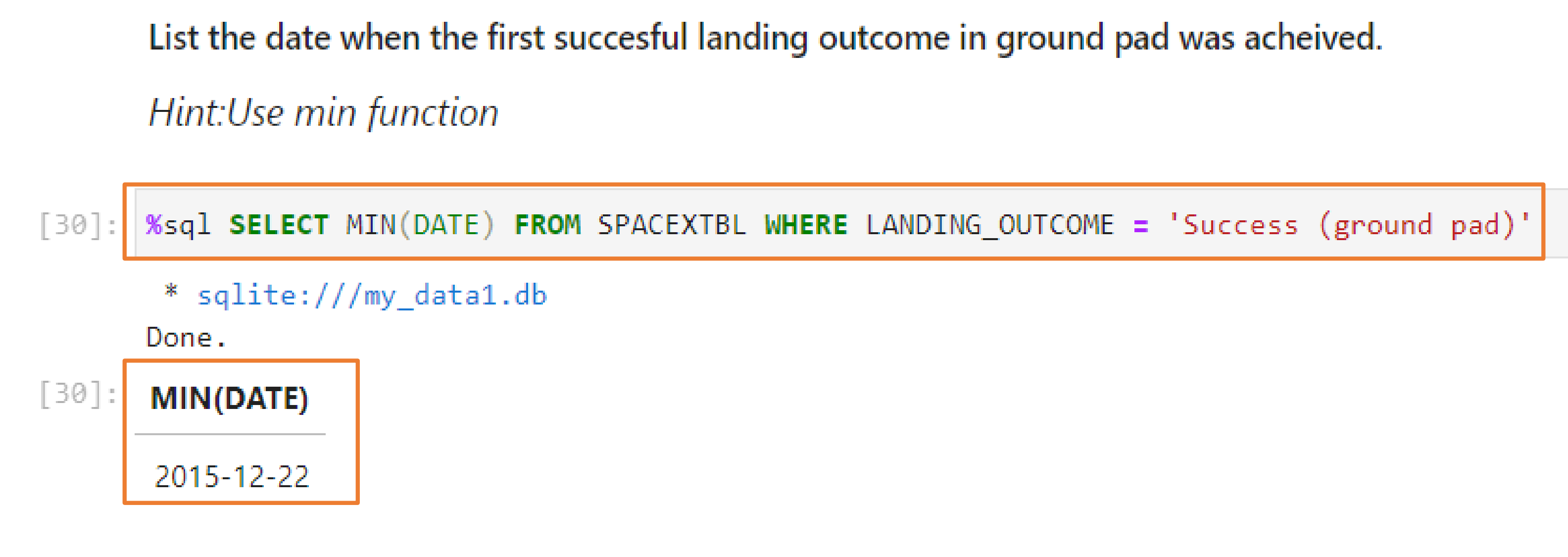
```
 * sqlite:///my_data1.db
Done.
```

[14]:

AVG(PAYLOAD_MASS__KG_)
2928.4

- ✓ *Using the aggregate AVG function, gives the average of the column Payload_Mass_kg_ from SpaceXtbl, and the WHERE clause filters the dataset to only perform average on Booster version F9 v1.1*

First Successful Ground Landing Date



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

Hint:Use min function

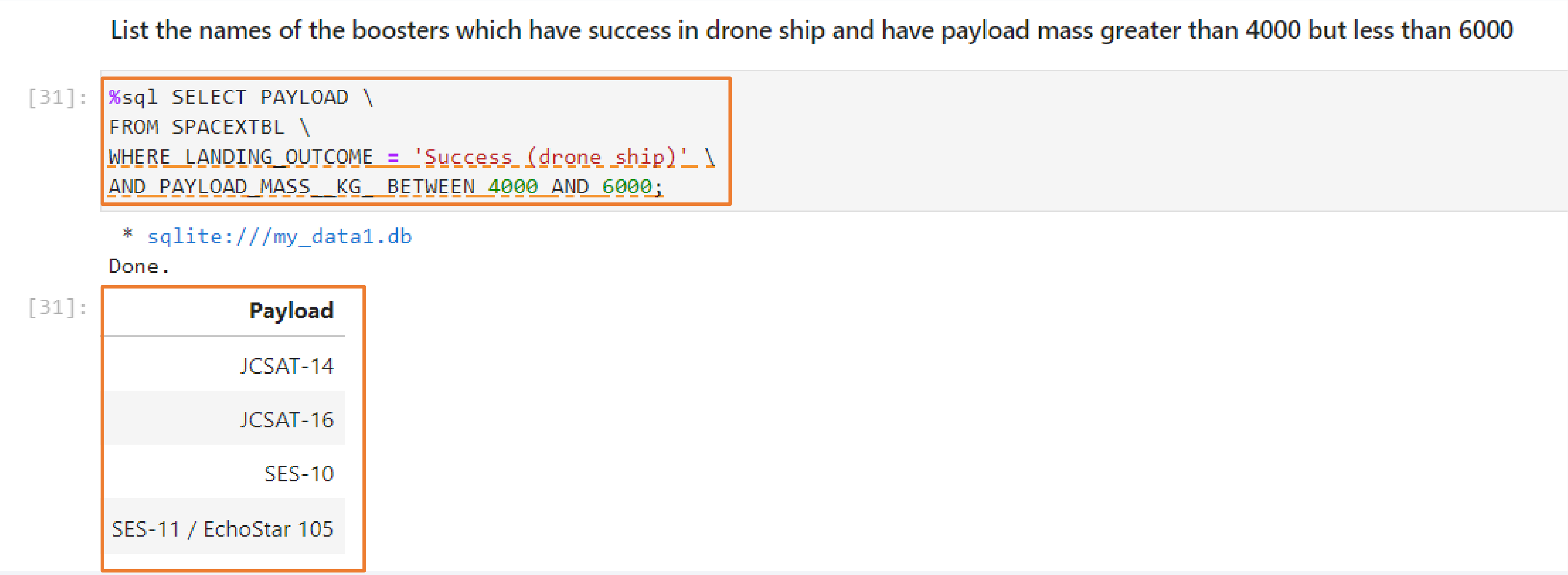
```
[30]: %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)'

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

[30]:

MIN(DATE)
2015-12-22

✓ *Using the aggregate MIN function, gives the first date of the column from SpaceXtbl, and the WHERE clause filters the dataset to provide result for Landing outcome equal to “Success (ground pad)”*

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

```
[31]: %sql SELECT PAYLOAD \
      FROM SPACEXTBL \
      WHERE LANDING_OUTCOME = 'Success (drone ship)' \
      AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

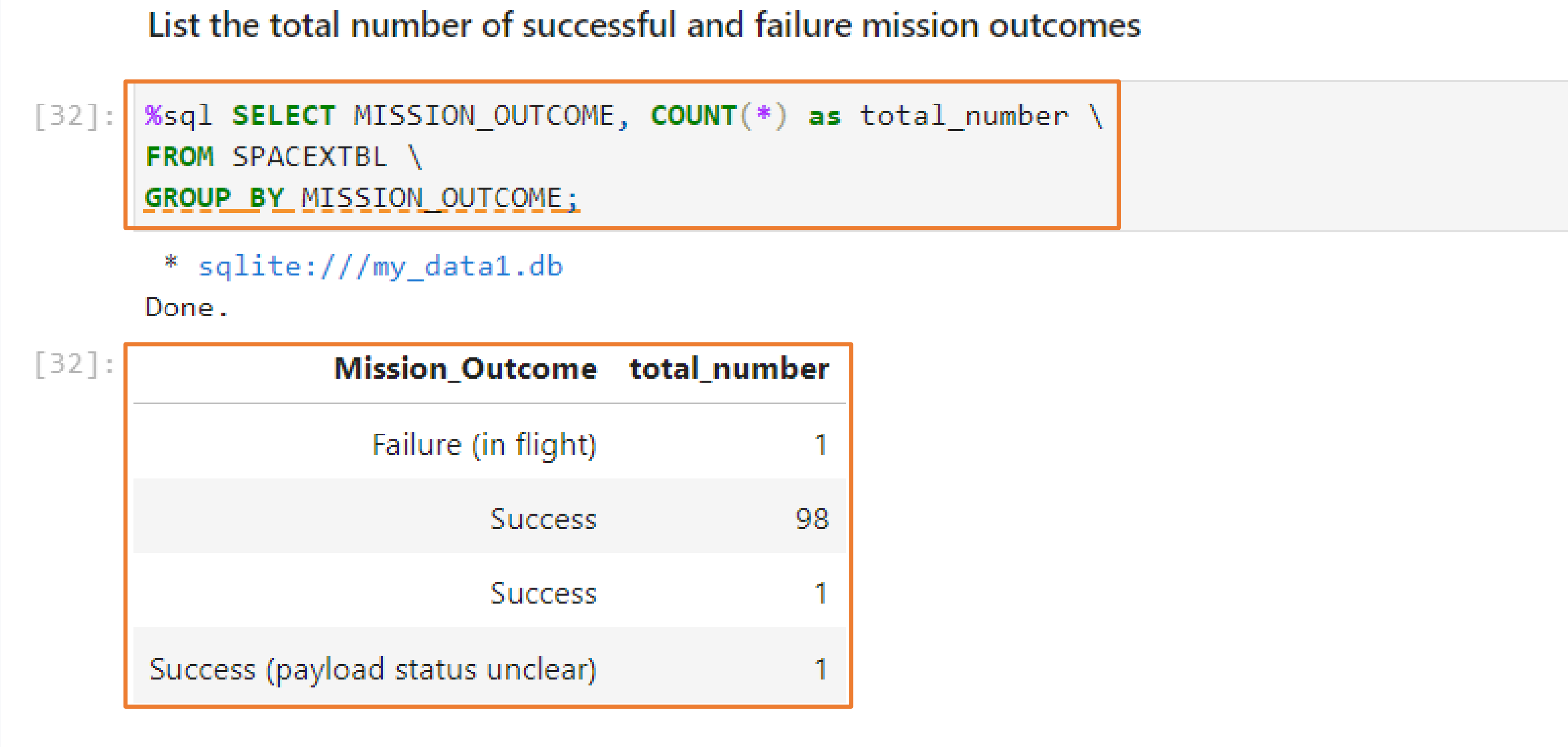
```
 * sqlite:///my_data1.db
Done.
```

[31]:

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

✓ *Selecting the names of the boosters from SpaceXtbl, using WHERE clause to filter the dataset where Landing Outcome is equal to "Success (drone ship)", AND clause adds additional filter of Payload mass between 4000 and 6000 kg.*

Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

```
[32]: %sql SELECT MISSION_OUTCOME, COUNT(*) as total_number \
      FROM SPACEXTBL \
      GROUP BY MISSION_OUTCOME;
```

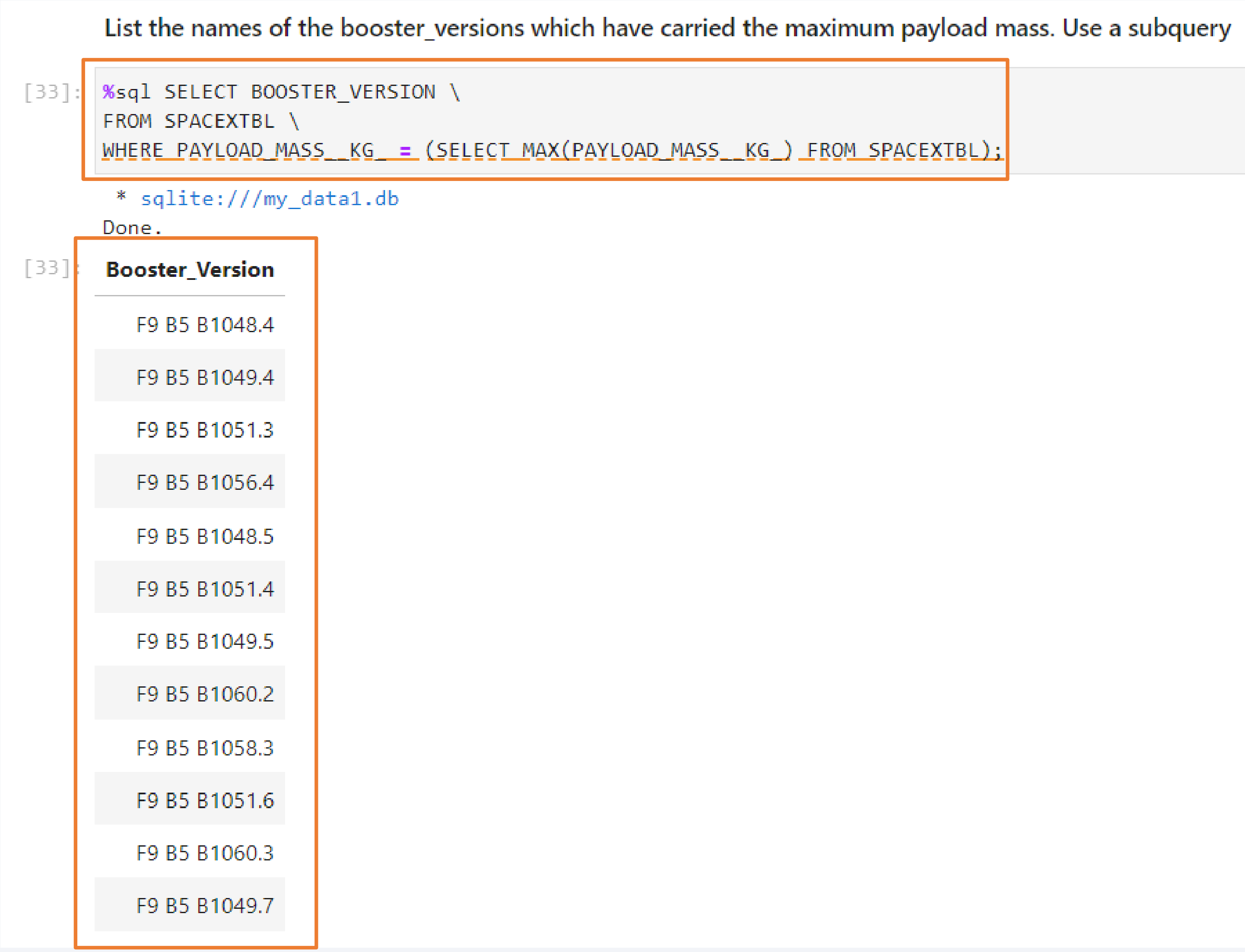
```
 * sqlite:///my_data1.db
Done.
```

[32]:

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

✓ *Selecting the Mission outcome and count of the mission outcome from SpaceXtbl and grouping them by mission_outcome using GROUP BY clause.*

Boosters Carried Maximum Payload



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

```
[33]: %sql SELECT BOOSTER_VERSION \
      FROM SPACEXTBL \
      WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

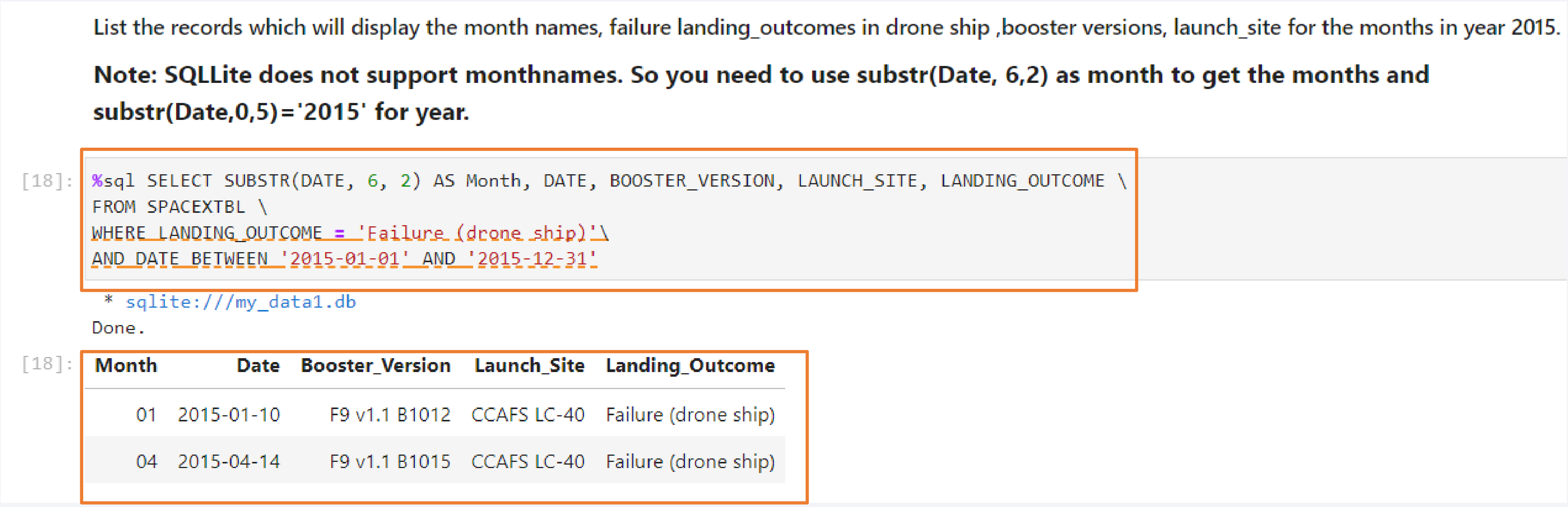
```
 * sqlite:///my_data1.db
Done.
```

[33]:

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

- ✓ *Selecting the Booster Versions which have carried the Maximum load from SPaceXtbl.*
- ✓ *Sub-query is used to select Maximum load, output of which is used in a WHERE clause to filter the main query*

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.

```
[18]: %sql SELECT SUBSTR(DATE, 6, 2) AS Month, DATE, BOOSTER_VERSION, LAUNCH_SITE, LANDING_OUTCOME \
      FROM SPACEXTBL \
      WHERE LANDING_OUTCOME = 'Failure (drone ship)'\
      AND DATE BETWEEN '2015-01-01' AND '2015-12-31'
```

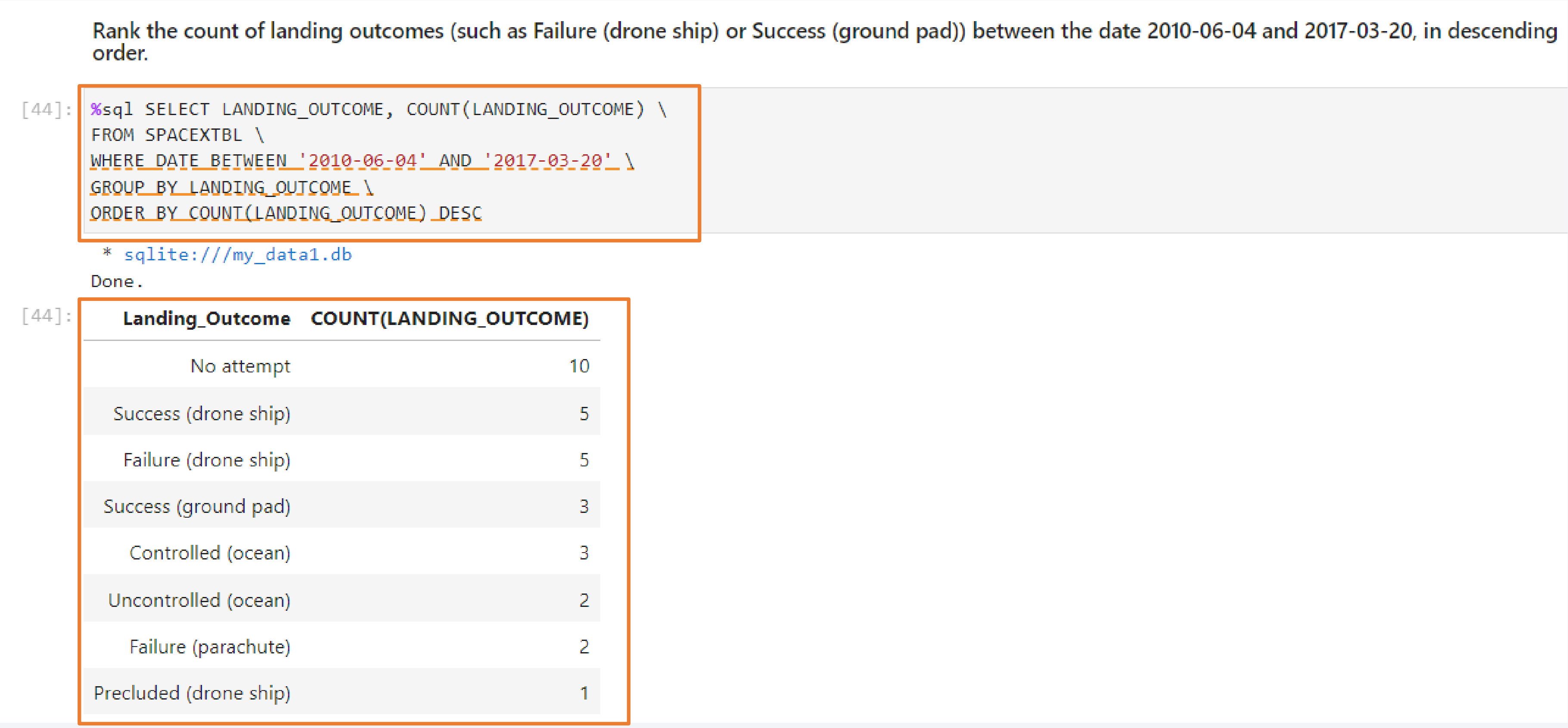
```
 * sqlite:///my_data1.db
Done.
```

[18]:

Month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

✓ *This query Selects Month, Date, Booster version, Launch Site and Landing Outcome from SpaceXtbl for the Year 2015, WHERE clause is used to filter the Failed Landing Outcomes.*

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.

```
[44]: %sql SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) \
      FROM SPACEXTBL \
      WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
      GROUP BY LANDING_OUTCOME \
      ORDER BY COUNT(LANDING_OUTCOME) DESC
```

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

[44]:

Landing_Outcome	COUNT(LANDING_OUTCOME)
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

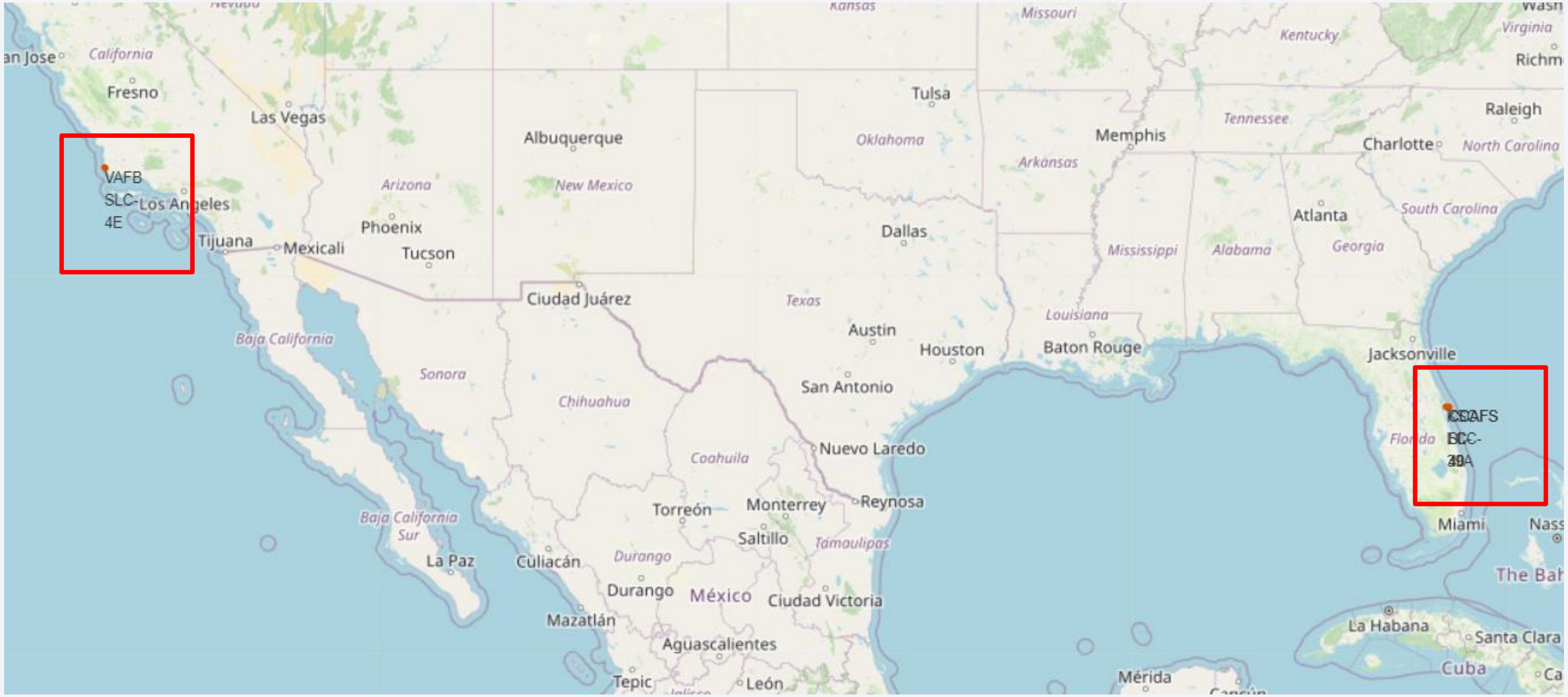
- ✓ *Selecting the Landing Outcome and Count of Landing Outcome from SpaceXtbl, using WHERE clause to filter Date between 2010-06-04 and 2017-03-20.*

- ✓ *Using GROUP BY clause to group the output result as per Landing Outcome and ORDER BY clause is used to Rank the various Landing Outcomes.*

Section 3

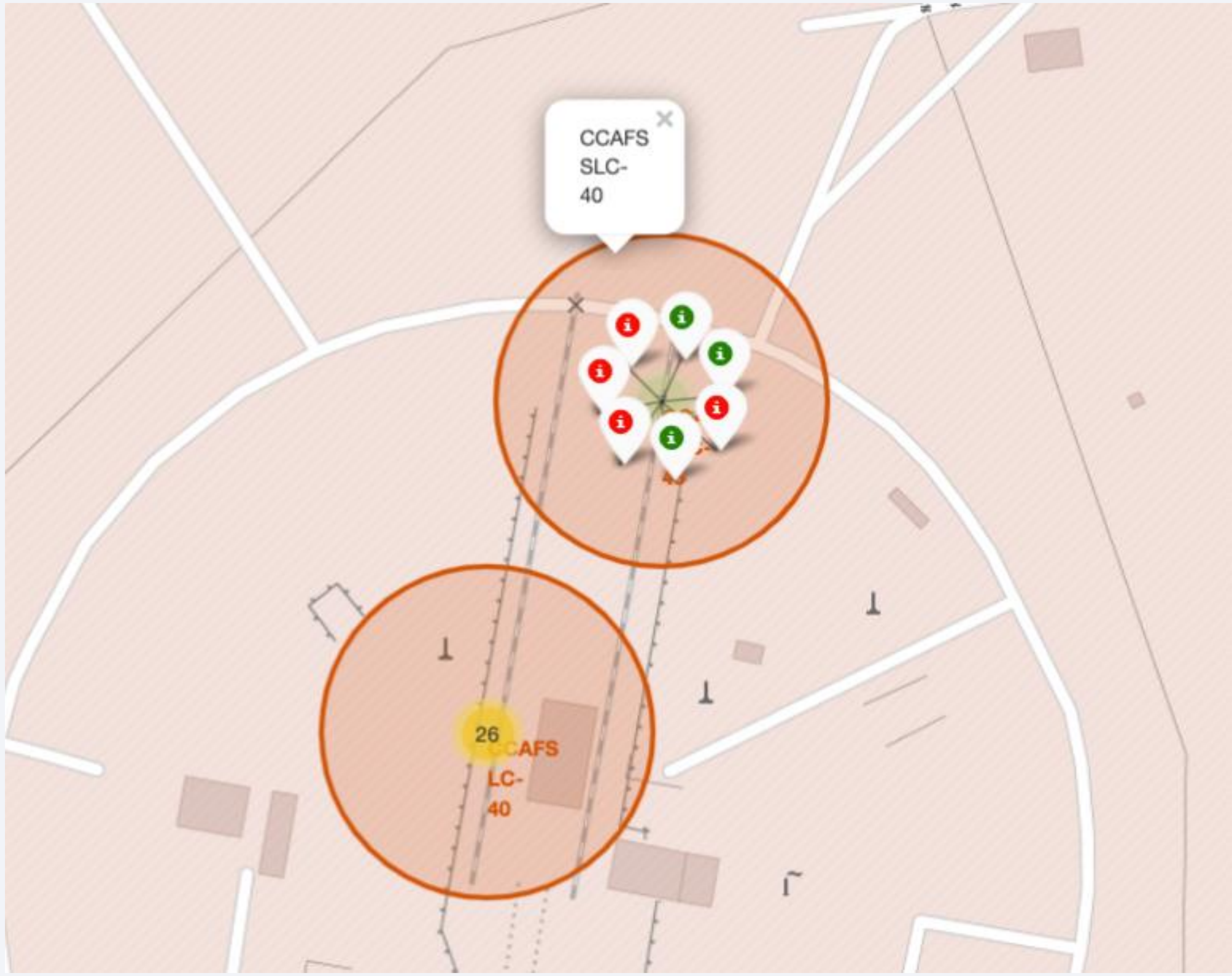
Launch Sites Proximities Analysis

All Launch Sites' location markers on Global map



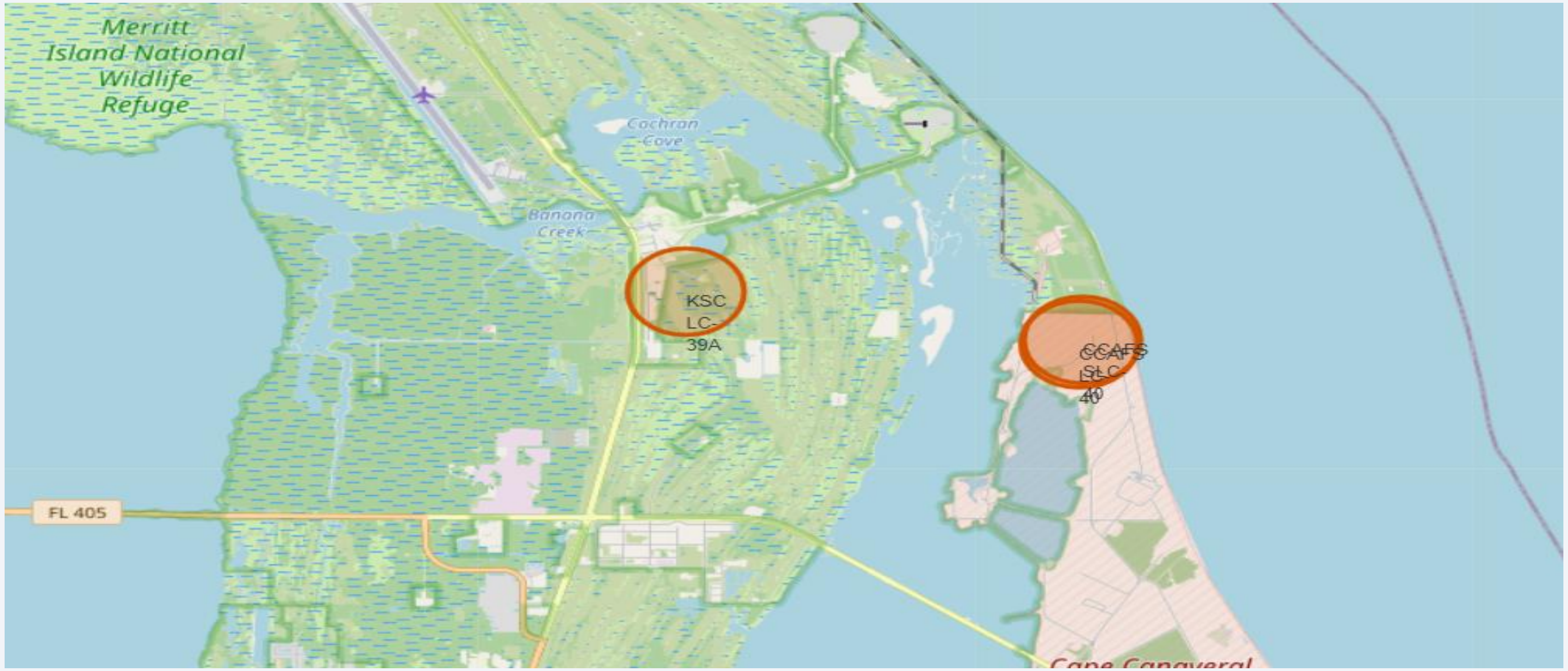
- ✓ *Most of the Launch sites are in proximity to the equator line.*
- ✓ *All the Launch sites are in very close proximity to the coast.*

Color Labelled Launch records on the Map

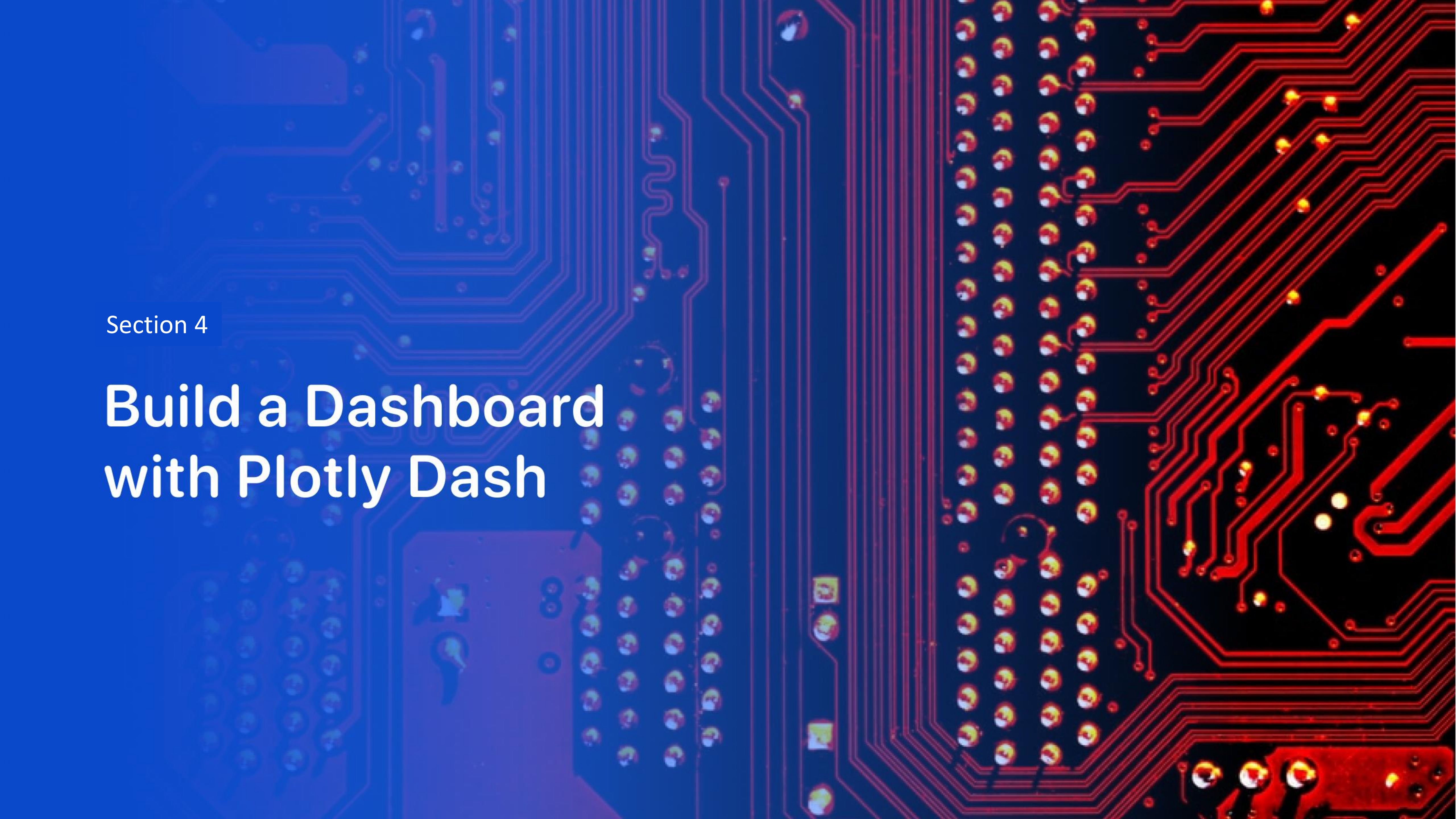


- ✓ *From the color labelled markers, we should be able to easily identify which launch sites have relatively high success rate.*
- ✓ *Green marker indicates successful launch, Red marker indicates failed launch.*

Distance from the Launch sites to its proximities



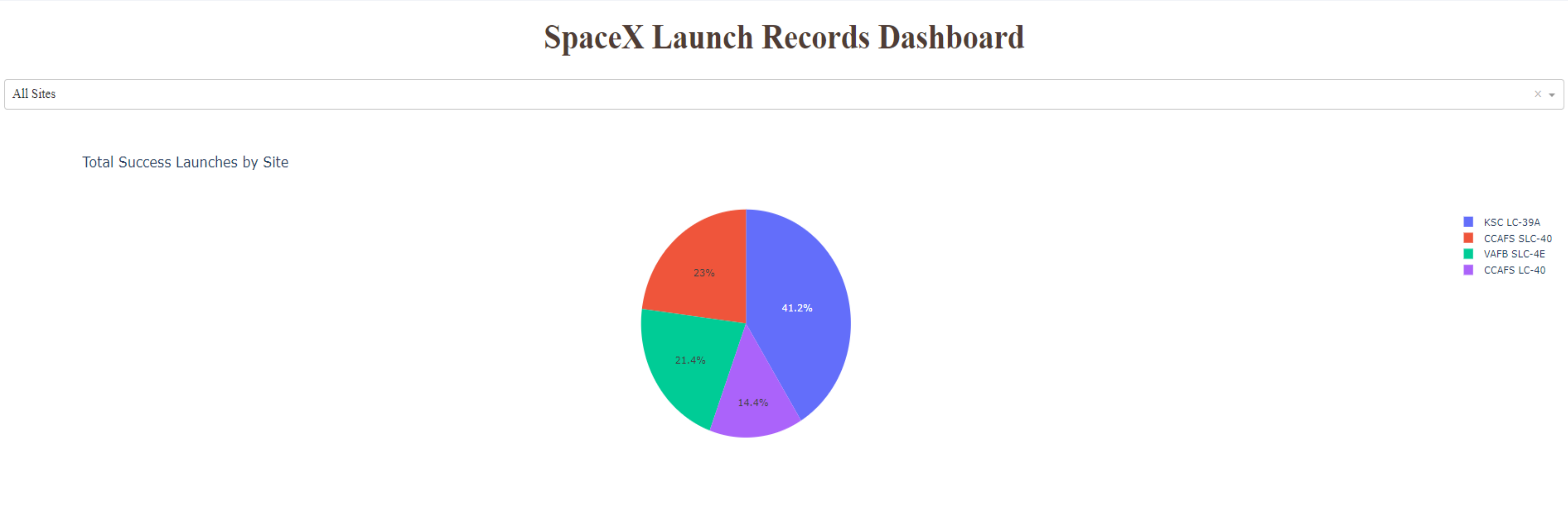
- ✓ *Launch sites are not in close proximity to the Railways.*
- ✓ *Launch sites are not in close proximity to the Highways.*
- ✓ *Launch sites are in close proximity to the Coastline.*
- ✓ *Launch sites keep certain distance away from the Cities.*



Section 4

Build a Dashboard with Plotly Dash

Launch success count for all Sites

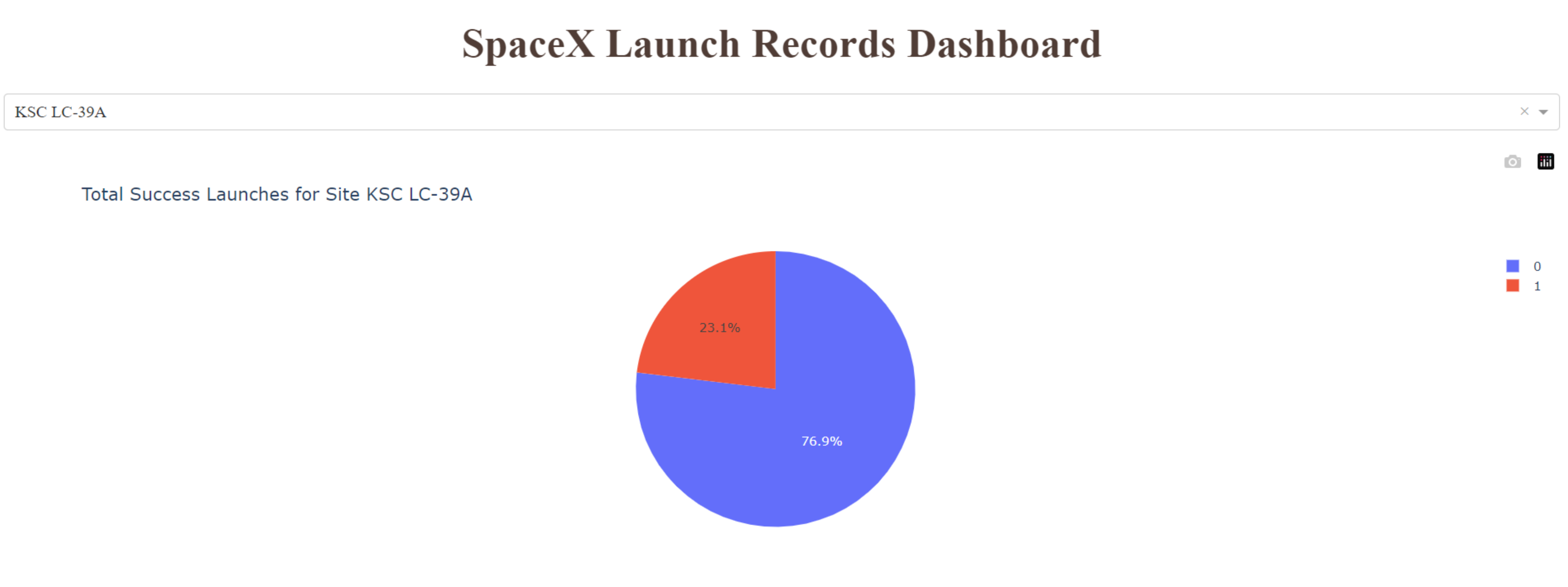


SpaceX Launch Records Dashboard

All Sites

✓ *We can see from the chart that KSC LC-39A had the most successful launches.*

Launch site with highest launch success ratio



SpaceX Launch Records Dashboard

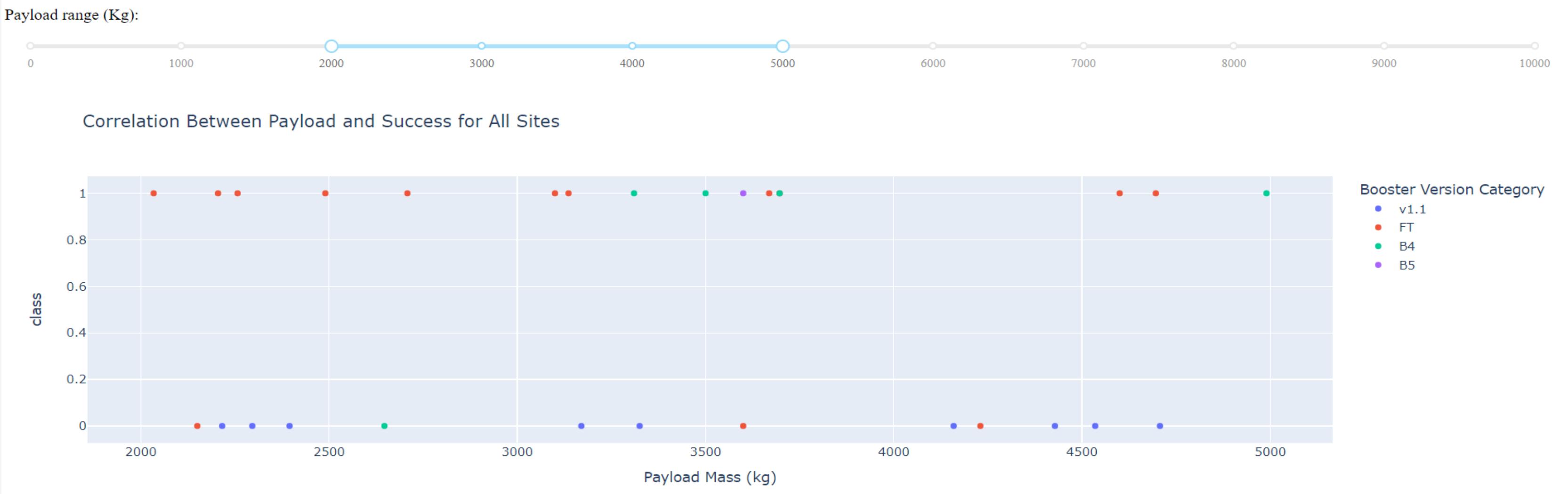
KSC LC-39A

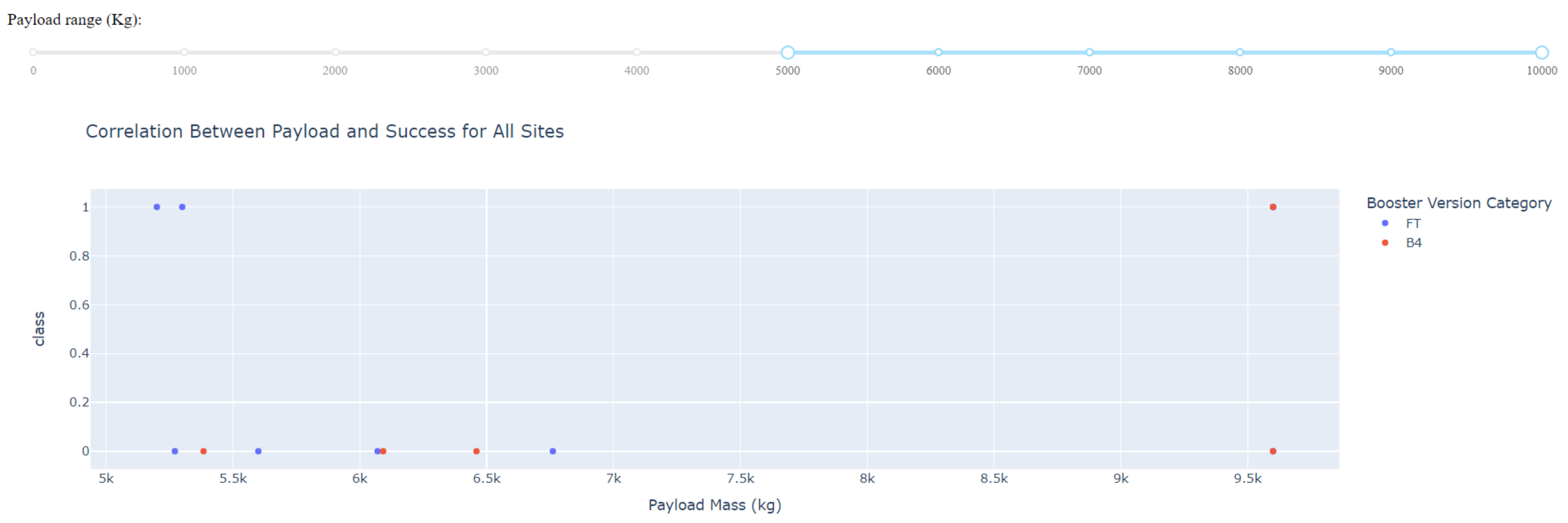
Total Success Launches for Site KSC LC-39A

✓ *KSC LC-39A has the highest launch success rate of 76.9% with 10 successful launches and only 3 failures.*

Payload Mass vs. Launch Outcome for All Sites

- ✓ *Launch with Payload between 2000 and 5000 kg. have higher success rate compared to Launches with Payload of more than 5000 kg.*



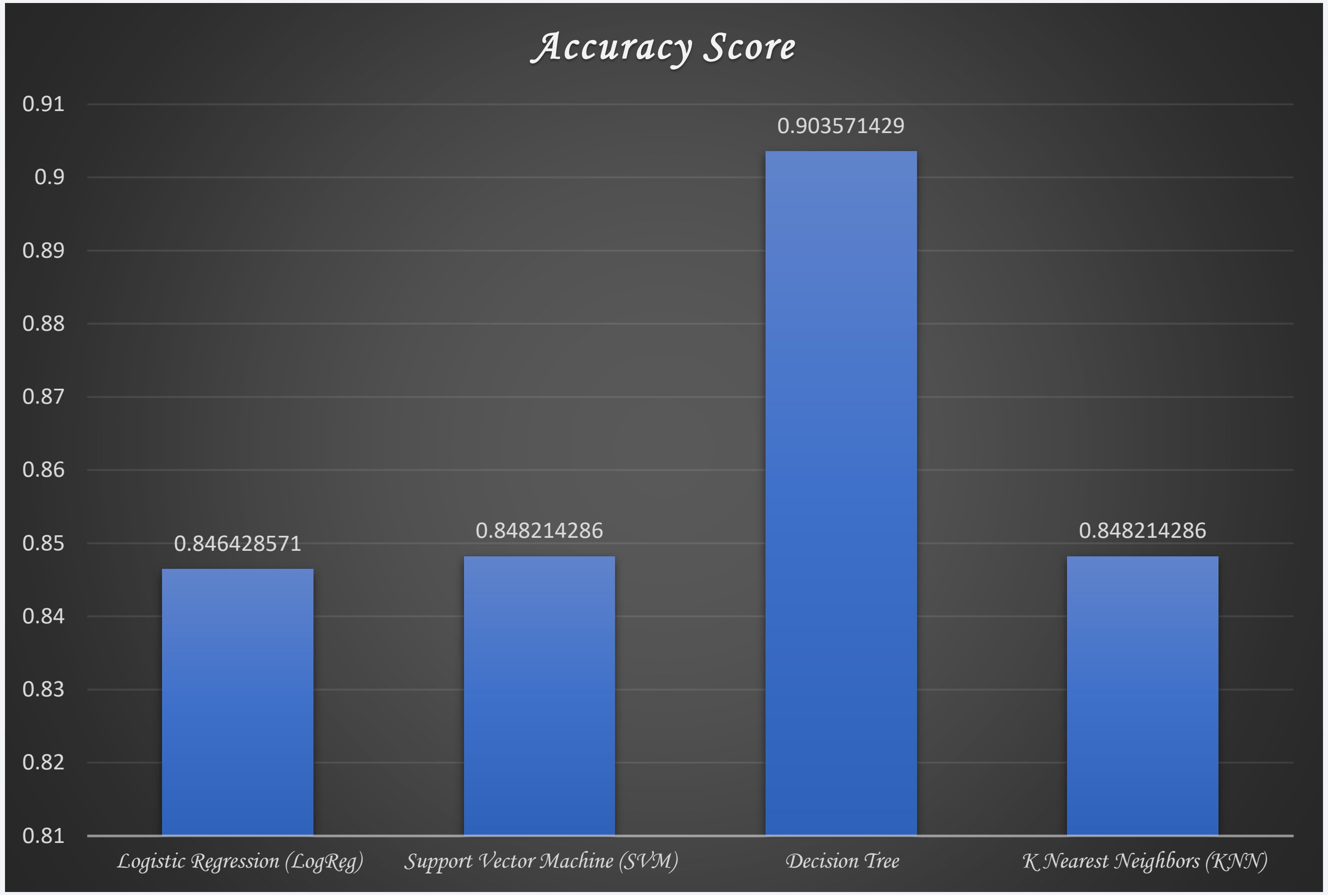


Section 5

Predictive Analysis (Classification)

Classification Accuracy

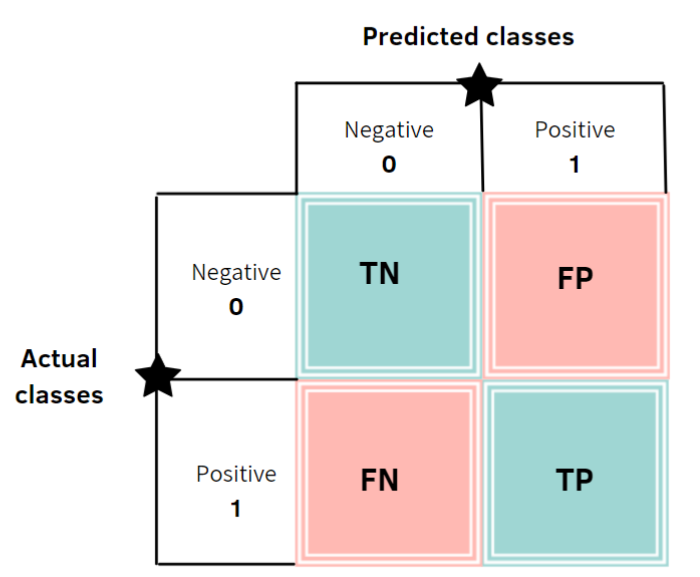
Algorithm	Accuracy Score
Logistic Regression (LogReg)	0.846428571
Support Vector Machine (SVM)	0.848214286
Decision Tree	0.903571429
K Nearest Neighbors (KNN)	0.848214286

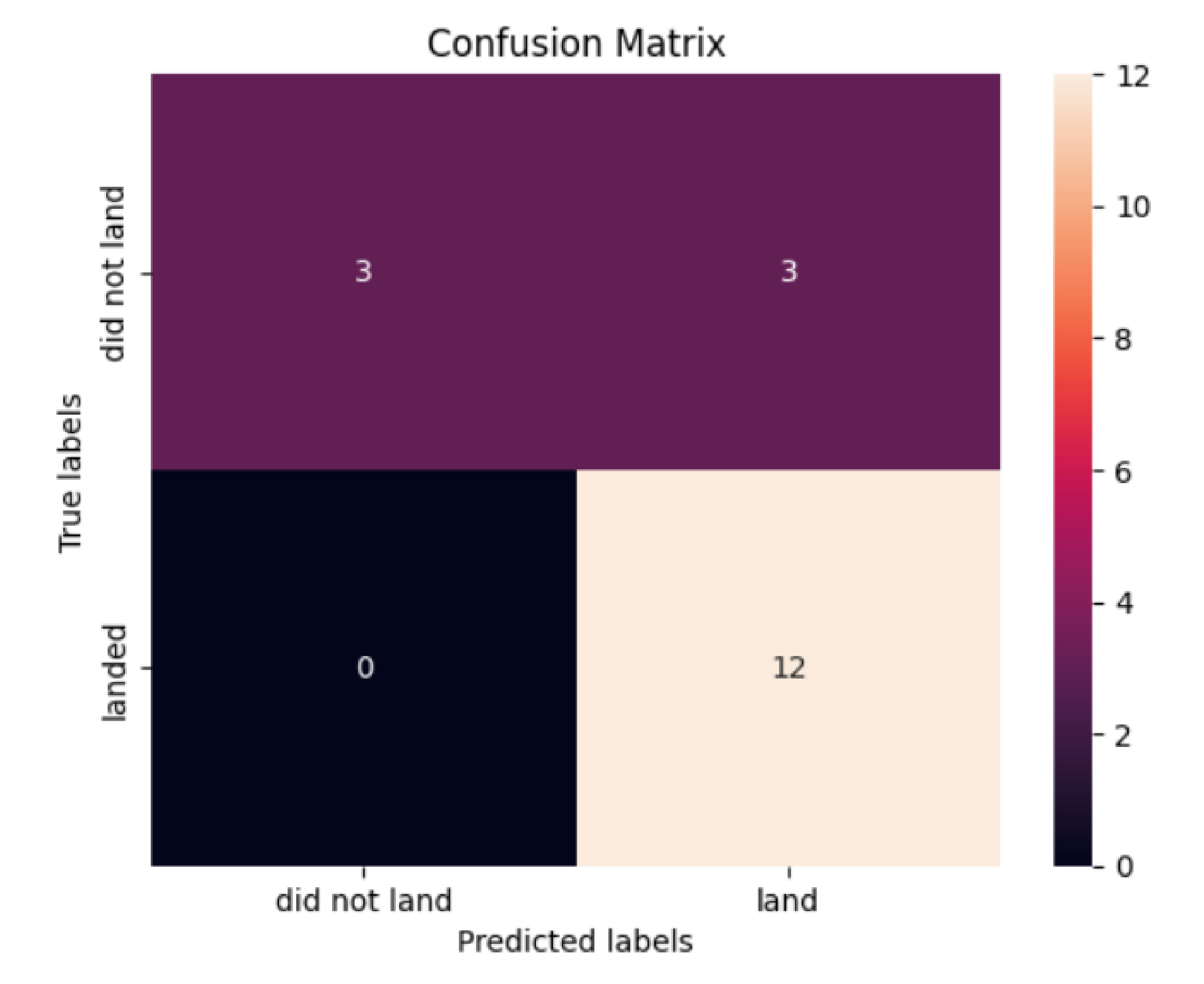


- ✓ After validating all the Models, we found out that the Best model is DecisionTree with a score of 0.9035714285714287 and Best params is : {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'random'}

Confusion Matrix

- ✓ *Examining the Confusion Matrix, we see that Decision Tree can distinguish between the different classes. We see that the major problem is False Positives.*





Conclusions

- ✓ Decision Tree is the best Machine Learning algorithm for this dataset.
- ✓ Launches with a low Payload mass show better results than launches with a larger Payload mass.
- ✓ Most of the Launch sites are in proximity to the equator and all the Launch sites are in very close proximity to the coastline.
- ✓ The success rate of SpaceX launches increases over the years.
- ✓ KSC LC-39A has the highest launch success rate of 76.9% with 10 successful launches and only 3 failures.
- ✓ Orbits ES-L1, GEO, HEO, and SSO have the best Success rate of 100%.

Thank you!