IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Presentation by: Sibiya Kudzanai

07-Dec-2023

Outline

This report is structured as follows (in that order):

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

Executive Summary

Summary of methodologies

- Data Collection through SpaceX API, and Web Scrapping
- Data Wrangling and Cleaning – missing valves, consistent formatting of data types.
- Exploratory Data Analysis with SQL and Python Libraries – Descriptive statistics about the dataset.
- Data Visualization - interactive viz, using Folium.
- Machine Learning Prediction

Summary of all results

- Attached herein are screenshots and commentary of the meaning of the findings.

Introduction

Project background and context

This is a report on capstone project conducted on SpaceX dataset to predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of $62 million dollars/each. Other providers cost the same upward of $165 million/each, much of the savings is because SpaceX can reuse the first stage. If we can determine if the first stage will land, we can determine the cost of a launch. This information can be used by an alternate company that wants to bid against SpaceX for a rocket launch.

Problems you want to find answers

The project’s objective was to answer the below problem statements or questions

1. What are the factors that determine if a rocket landing is successful?
2. What is the interaction amongst features that determine the success rate of a successful rocket landing.
3. What operating conditions need to be in place for a successful rocket landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection was done through SpaceX API and web scraping from Wikipedia.
- The collected data was cleaned. This involved missing valves and formatting of same data for consistency.
- To understand the structure of the dataset, Exploratory Data Analysis (EDA) and data visualization were conducted.
- Folium and Plotly Dash were used to present interactive visuals.
- Lastly, predictive analysis using classification models was done.

Data Collection

- GET Request was used to collect data together with SpaceX API.
- Raw data from SpaceX was decoded a Json using .json() function call and was turned into a pandas Data Frame using.
- Data was then cleaned – checking for missing data, and fill in missing values where necessary – with averages.
- Web scraping from Wikipedia was also carried out to get Falcon 9 launch records. This was done using BeautifulSoup.

Data Collection – SpaceX API

Below is a link to the GitHub URL that has all of the details on how data was collected:

https://github.com/Sibiya-K/IBM-Applied-Data-Science-Capstone.git

1. Made a GET Request to the SpaceX API.
2. Received and inspected the data.
3. Data wrangling.
4. Data Formatting and cleaning.
5. Saving the cleaned data for further analysis.

Data Collection - Scraping

https://github.com/Sibiya-K/IBM-Applied-Data-Science-Capstone.git

1. Request the Falcon 9 Launch Wiki page from its URL
2. Extract all column names from the HTML table header
3. Find all tables on the wiki page
4. Create a data frame by parsing the launch HTML tables.

Data Wrangling

Carried out Exploratory Data Analysis to establish the training labels.

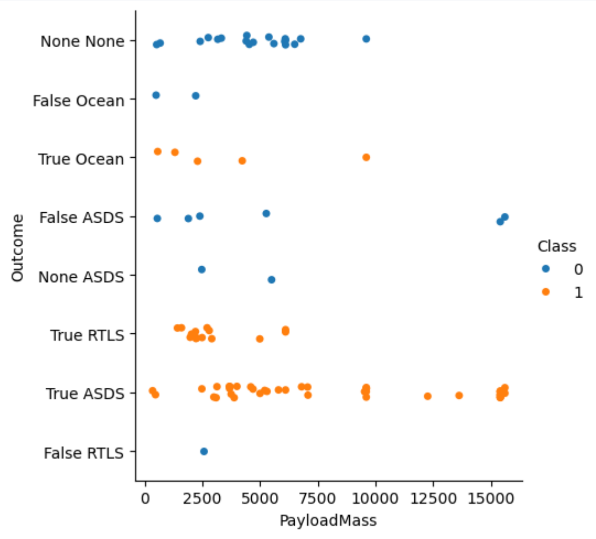
No. of launches at each site, and No. of occurrence of each orbits were calculated.

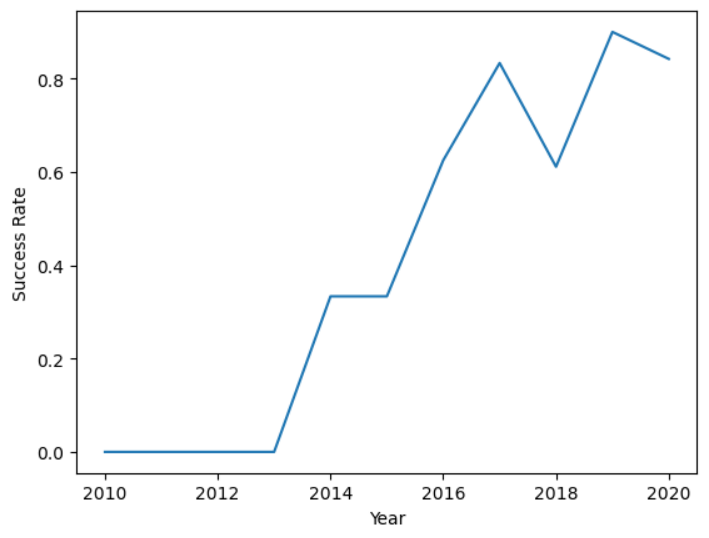
Created landing outcome label from outcome column and exported the results to csv file.

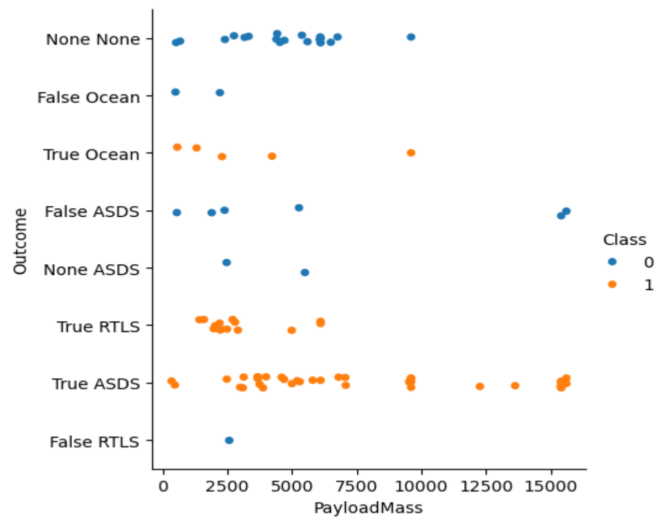
https://github.com/Sibiya-K/IBM-Applied-Data-Science-Capstone.git

EDA with Data Visualization

- Explore and visualized the relationship between; Flight Number vs. Launch Site, Payload and Launch Site, success rate of each orbit type, Flight Number and Orbit type, Payload and Orbit type and visualized the launch success yearly trend. **(https://github.com/Sibiya-K/IBM-Applied-Data-Science-Capstone.git)**







EDA with SQL

Used SQL in Jupyter notebook to analyze the SpaceX dataset. Queried the SpaceX dataset using SQL to answer the following;

- Display the names of the unique launch sites in the space mission,
- 5 records where launch sites begin with the string 'CCA'
- Total payload mass carried by boosters launched by NASA (CRS)
- The average payload mass carried by booster version F9 v1.1,
- The date when the first successful landing outcome in ground pad was achieved, the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- The total number of successful and failure mission outcomes and the names of the booster versions which have carried the maximum payload mass.
- The records on 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/Sibiya-K/IBM-Applied-Data-Science-Capstone.git)

Build an Interactive Map with Folium

- Marked all launch sites, added map objects/markers to mark the success or failure of launches for each site on the map.
- Assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, launch sites with relatively high success rate were identified.
- Calculated the distances between a launch site to its proximities, and my findings are about the launch sites are;

#Close proximity to coastline - to be able to fly over the ocean during launch.

#Close proximity to highways and railways - for easily transport required people, and heavy cargo that may be required.

#Not close proximity to cities - if close to cities the launch poses as a danger to the people.

(https://github.com/Sibiya-K/IBM-Applied-Data-Science-Capstone.git)

Build a Dashboard with Plotly Dash

- Built interactive dashboard with Plotly dash
- Plotted pie charts showing the total launches by a certain sites
- Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- **(https://github.com/Sibiya-K/IBM-Applied-Data-Science-Capstone.git)**

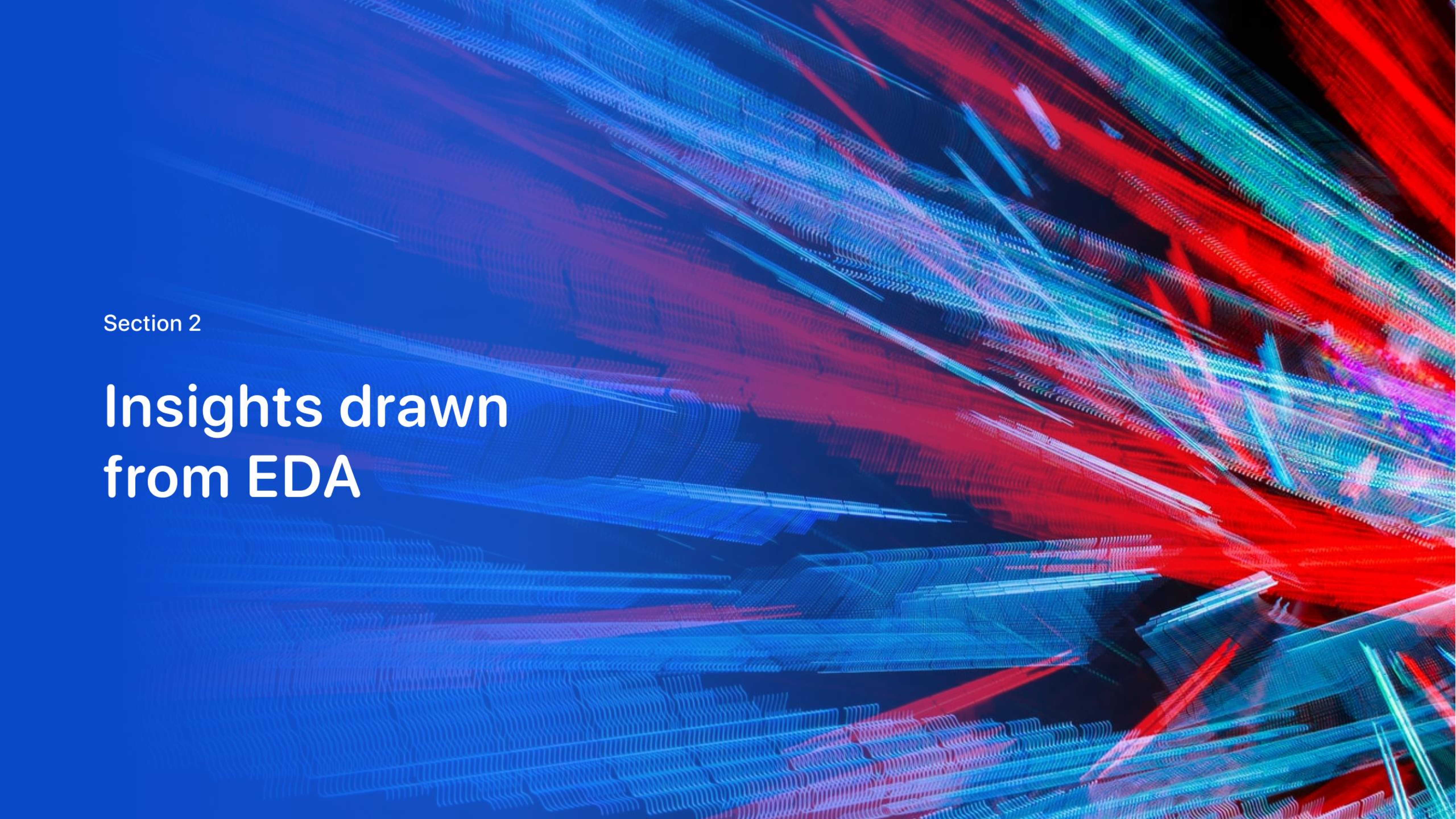
Predictive Analysis (Classification)

- Loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- Built different machine learning models and tune different hyperparameters using GridSearchCV.
- Used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- Found the best performing classification model.
- **(https://github.com/Sibiya-K/IBM-Applied-Data-Science-Capstone.git)**

Results

Covers insight from the below:

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

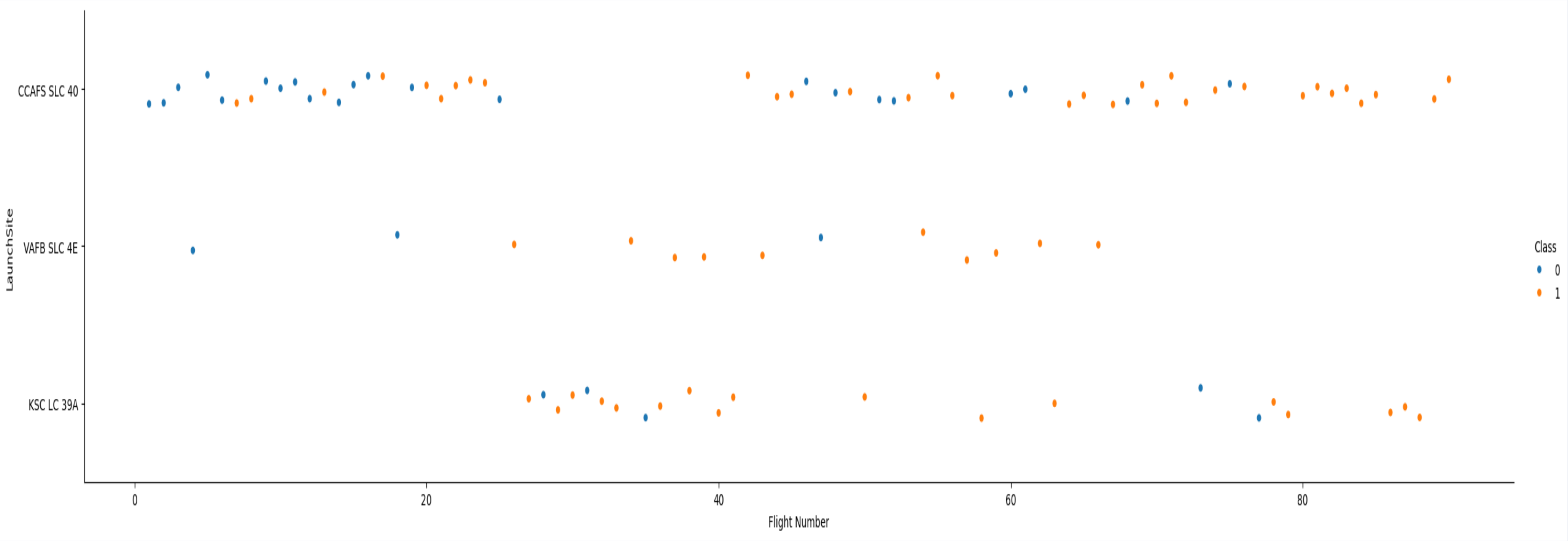


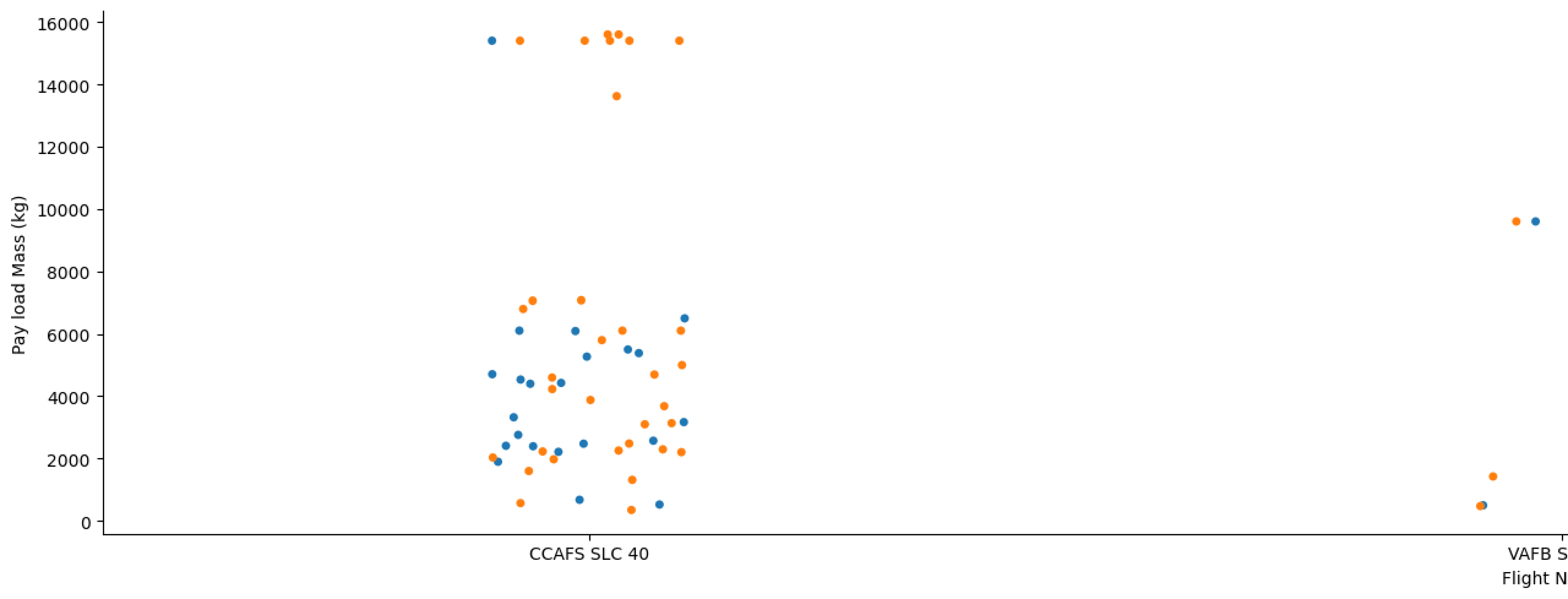
Section 2

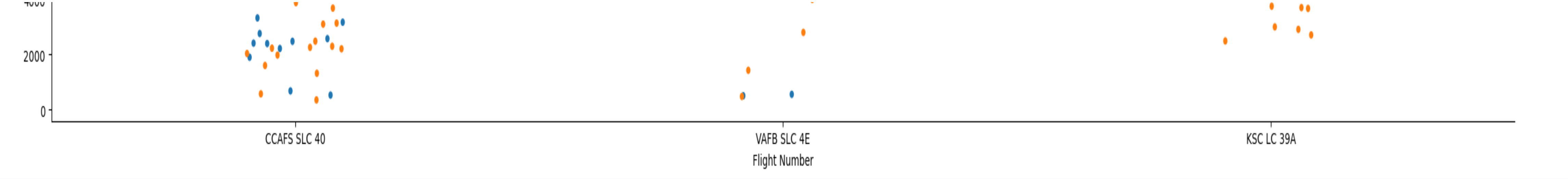
Insights drawn from EDA

Flight Number vs. Launch Site

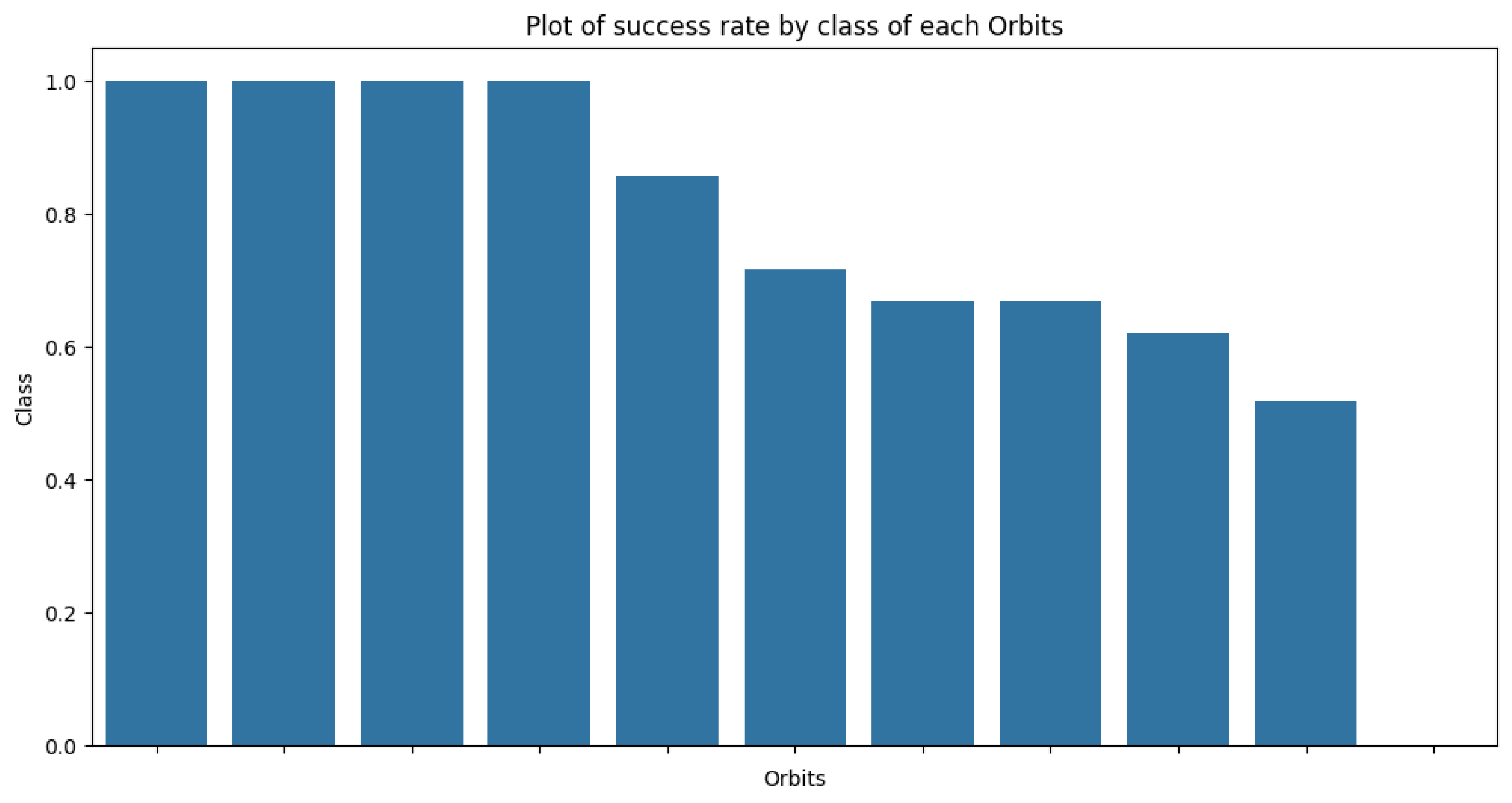
Scatter Plot Flight Number vs. Launch Site:



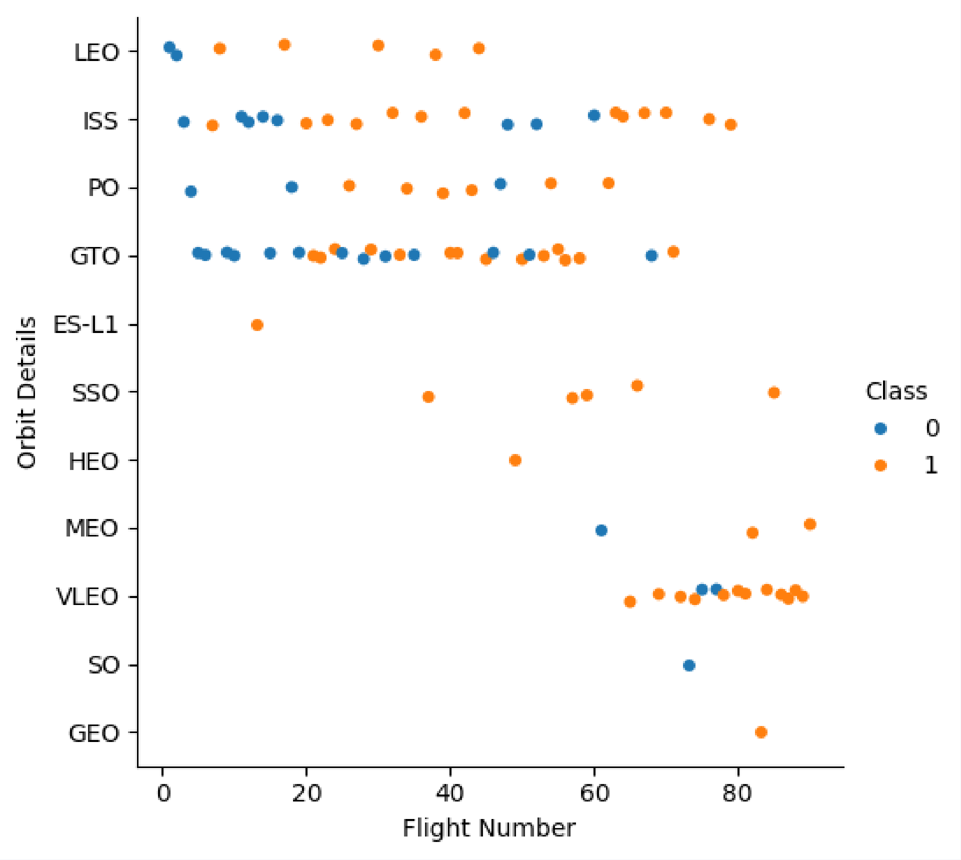

16000
14000
12000
10000
8000
6000
4000
2000
0
Pay load Mass (kg)
CCAFS SLC 40
VAFB S
Flight N


4000
2000
0
CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Flight Number

Success Rate vs. Orbit Type



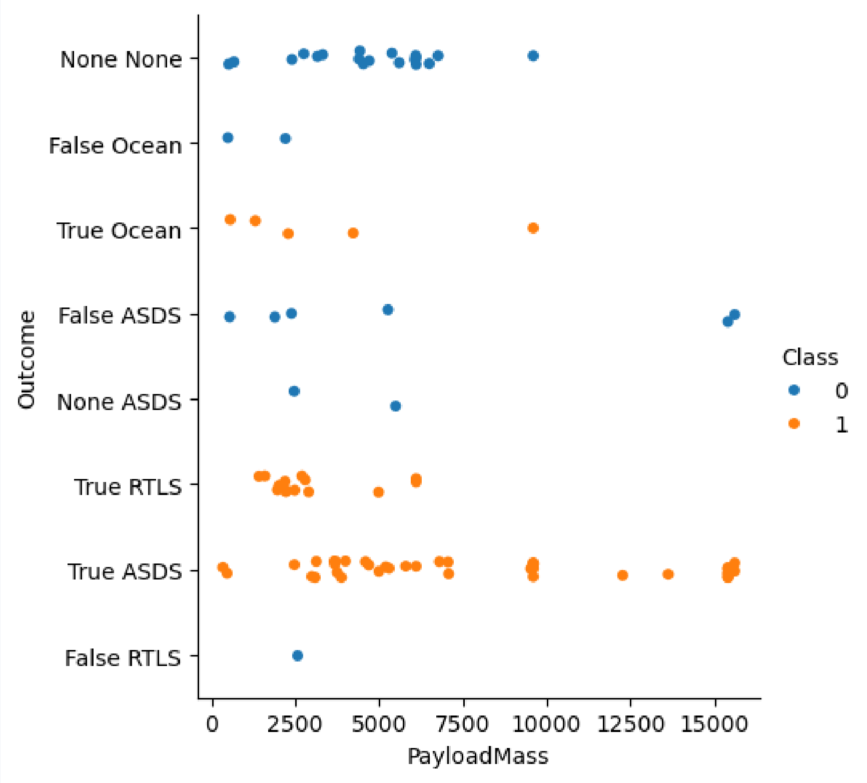
Flight Number vs. Orbit Type



LEO orbit success is related to the number of flights.

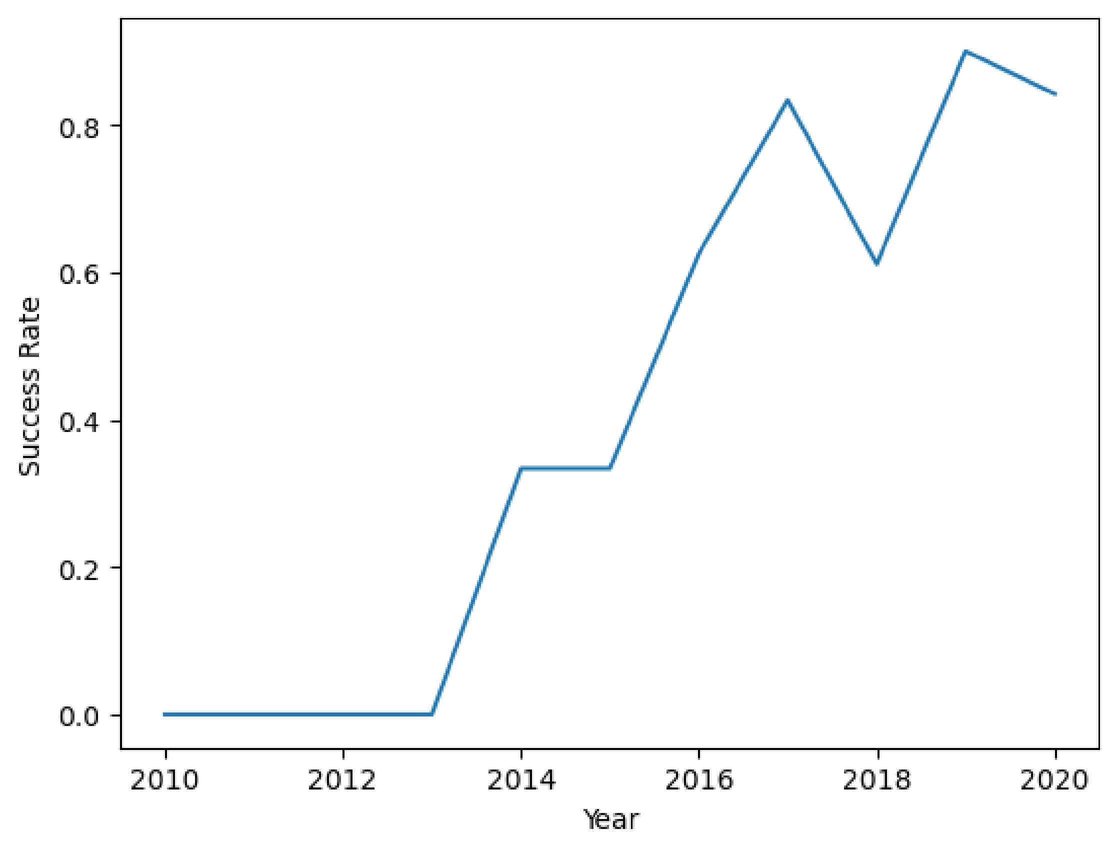
GTO orbit, does not seem to have no relationship between flight number and the orbit.

Payload vs. Orbit Type



Heavy payloads seem to be strongly result in successful landing.

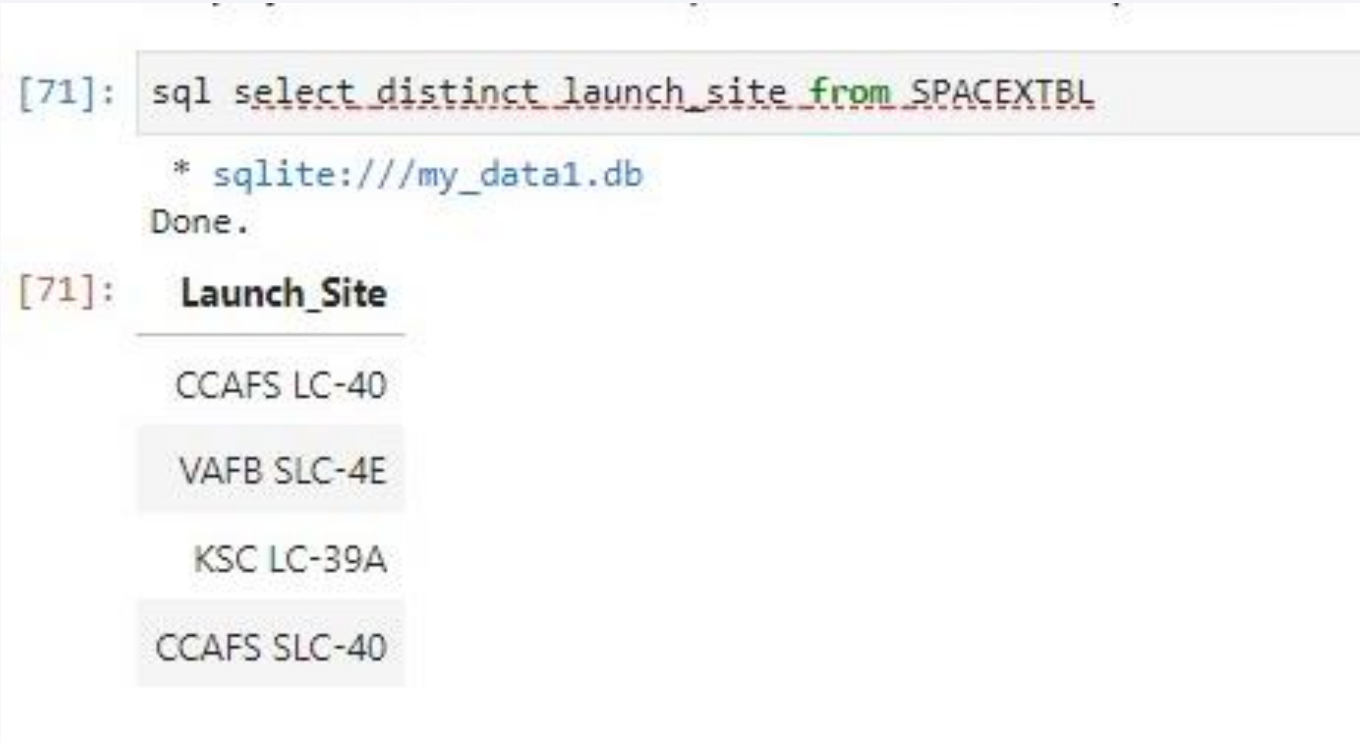
Launch Success Yearly Trend



Success rate for obit landing by SpaceX has been increasing from as low as 20% around 2013, and as high as 85% around 2019/2020.

All Launch Site Names

- Used key word **DISTINCT**.
- This will to show only unique i.e., launch sites from the SpaceX data in the dataset studied.



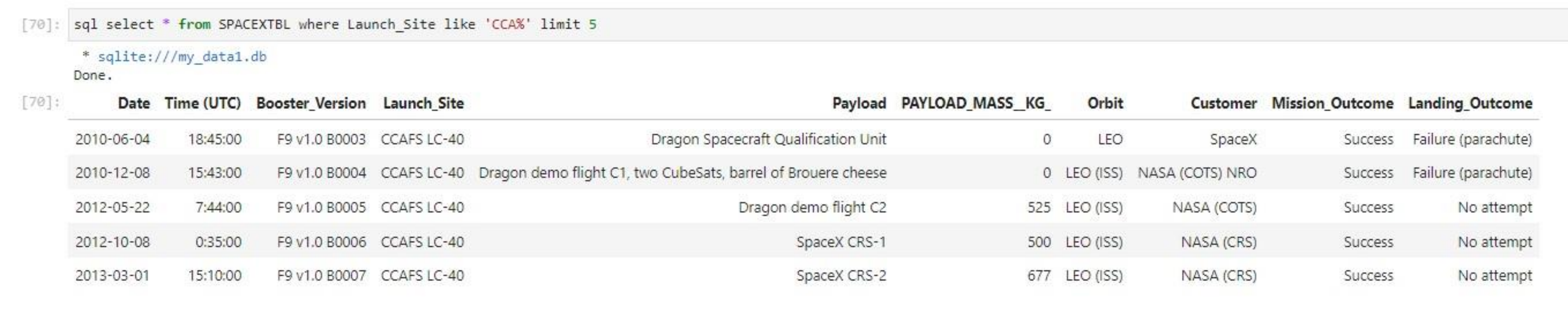
```
[71]: sql select distinct launch_site from SPACEXTBL

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

[71]:

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

Launch Site Names Begin with 'CCA'

Use the below, with a wild-card to find and display records where launch sites whose names begin with `CCA`



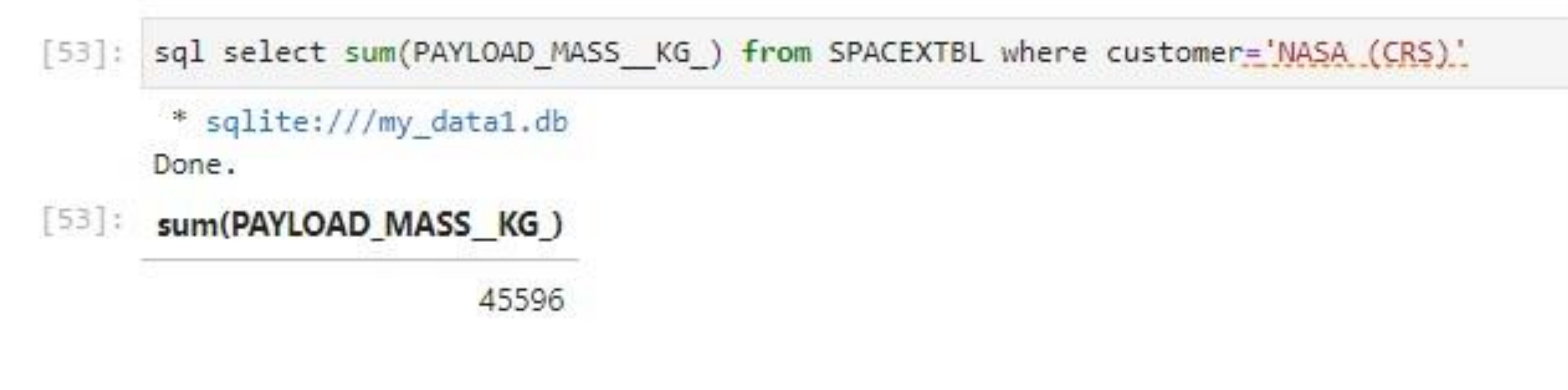
```
[70]: sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5

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

[70]:

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

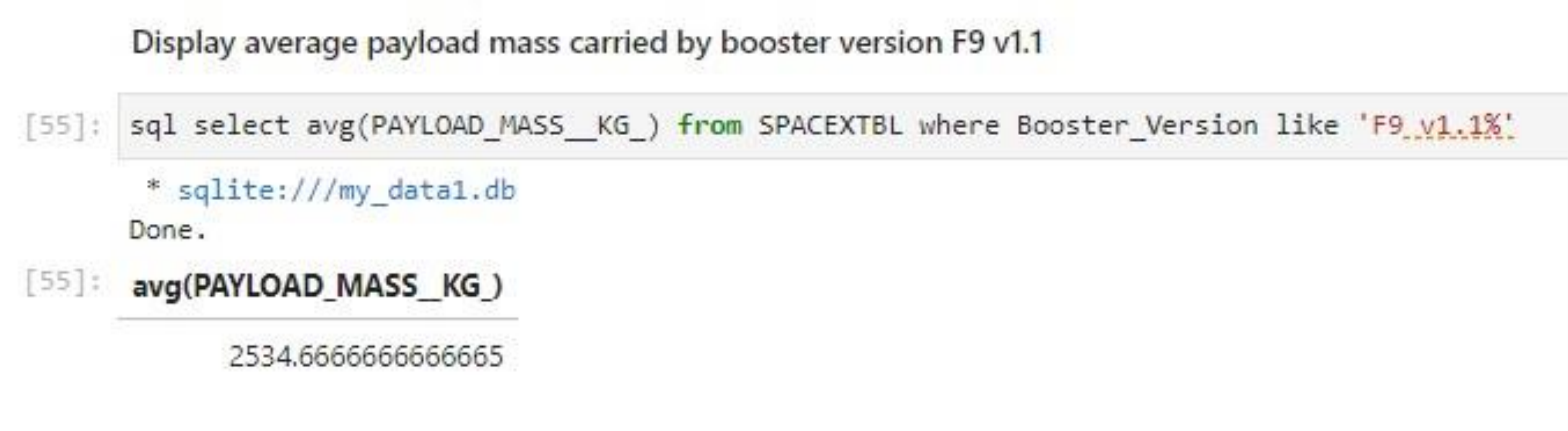
Used the WHERE clause to filter data, and Function ‘Sum’ to find the total payload, and the WHERE clause to filter, the payload carried by boosters from NASA



```
[53]: sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where customer='NASA (CRS)'

       * sqlite:///my_data1.db
      Done.

[53]: sum(PAYLOAD_MASS__KG_)
      ----------------------
                       45596
```

Average Payload Mass by F9 v1.1

- Used the AVG function to get the average payload mass carried by booster version F9 v1.1, and this was found to be 2,534.6

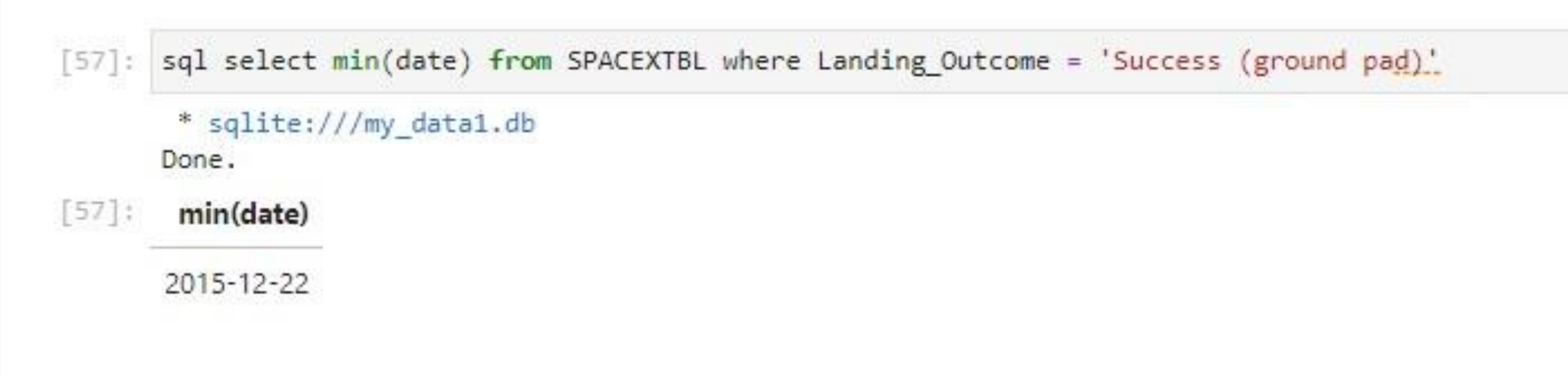


Display average payload mass carried by booster version F9 v1.1

```
[55]: sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version like 'F9 v1.1%'

 * sqlite:///my_data1.db
Done.

[55]: avg(PAYLOAD_MASS__KG_)
            2534.666666666665
```

First Successful Ground Landing Date

- The date of the first successful landing outcome on ground pad was 22 December 2015.



```
[57]: sql select min(date) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'

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

[57]:

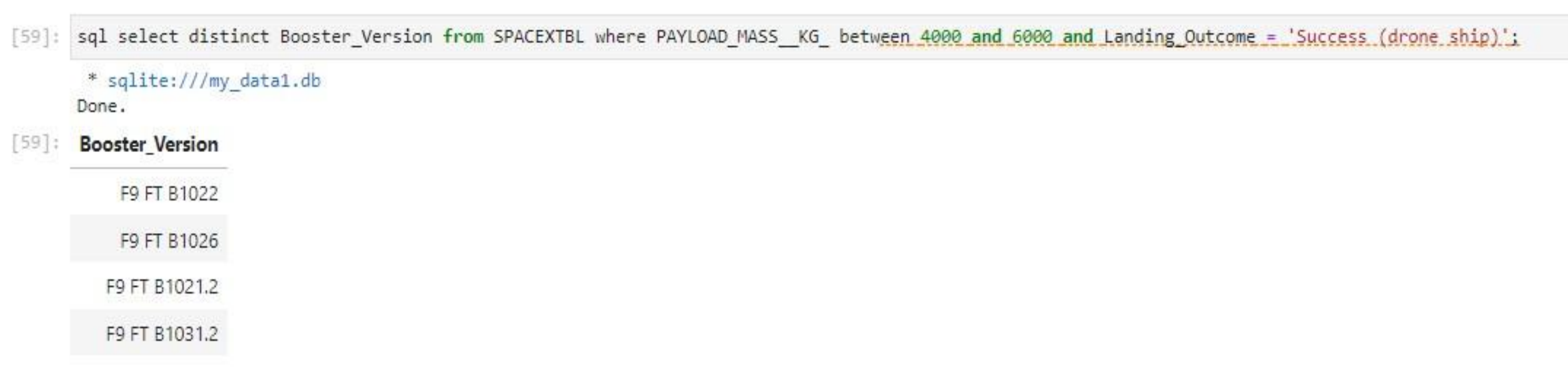
min(date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Use **WHERE** clause to filter for boosters which have successfully landed on drone ship.

Used the AND to get an intersection of the second condition, being.

Successful landing with payload mass greater than 4000 but less than 6000



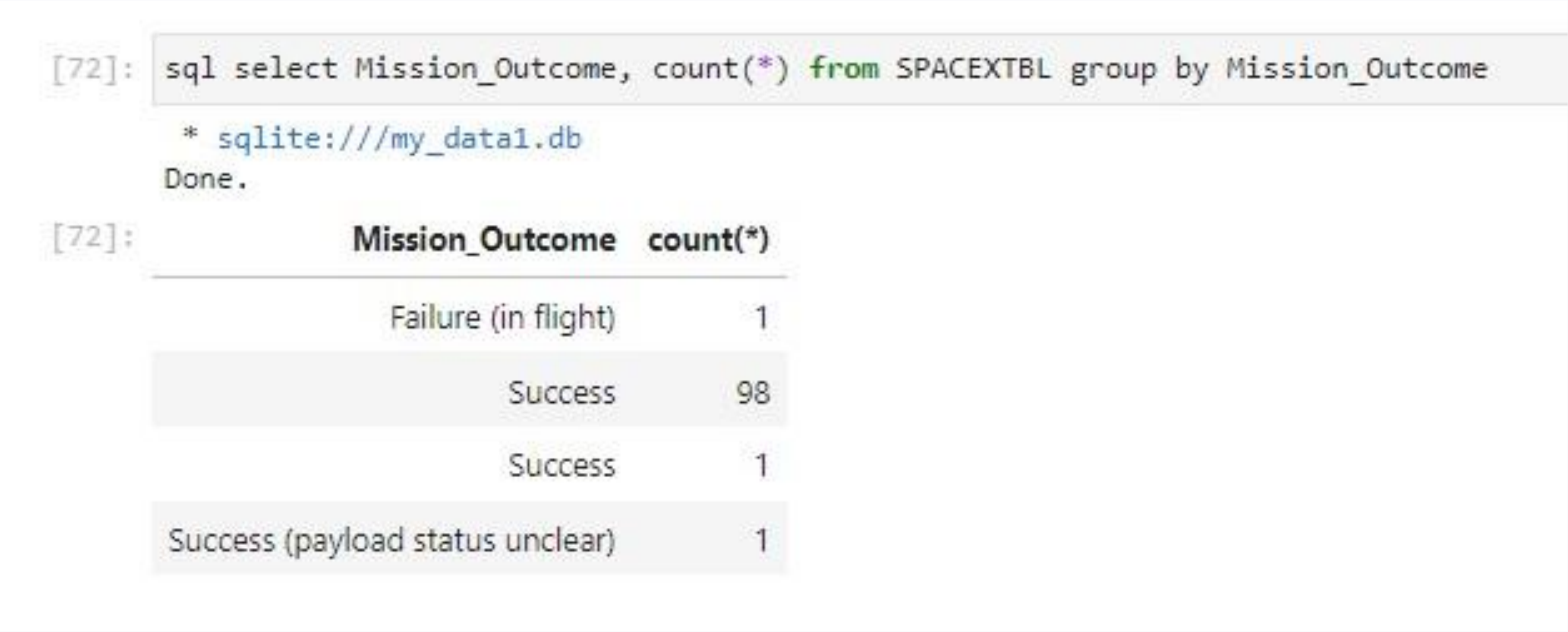
```
[59]: sql select distinct Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ between 4000 and 6000 and Landing_Outcome = 'Success (drone ship)';

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

[59]:

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Use the wildcard like ‘%’ to filter for **WHERE** MissionOutcome was a success or a failure.



```
[72]: sql select Mission_Outcome, count(*) from SPACEXTBL group by Mission_Outcome
```

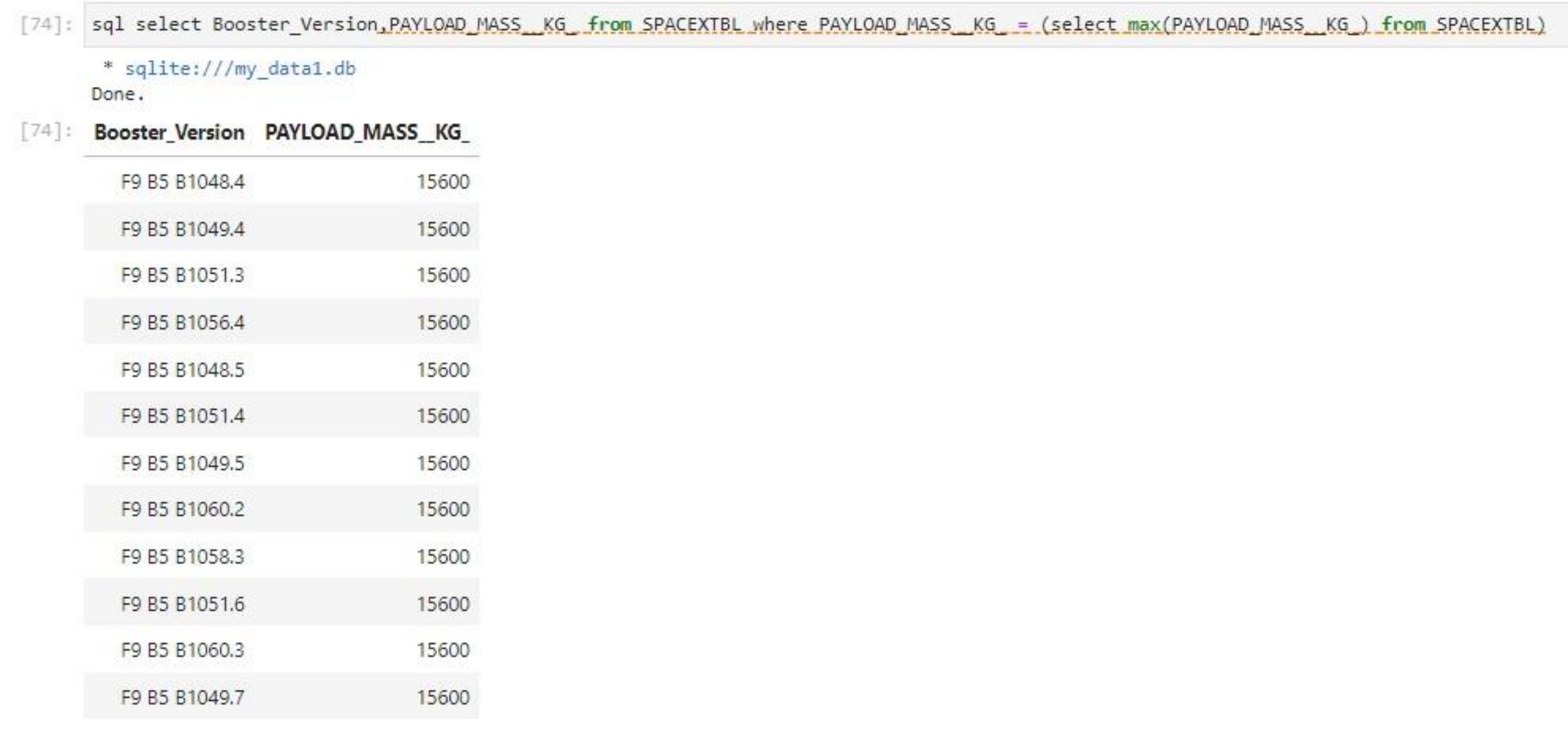
```
 * sqlite:///my_data1.db
Done.
```

[72]:

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

Boosters Carried Maximum Payload

- Found the booster that have carried the maximum payload.
- This was done through the use of a subquery in the **WHERE** clause and the **MAX()** function.

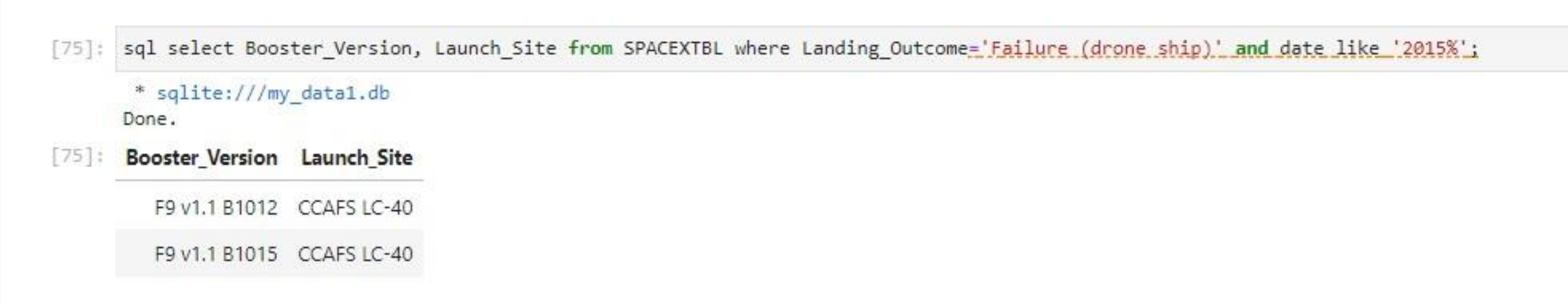


```
[74]: sql select Booster_Version,PAYLOAD_MASS__KG_ from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)

       * sqlite:///my_data1.db
      Done.
[74]:
```

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

2015 Launch Records

- Used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter the required.
- Being, the filtering of the failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015



```
[75]: sql select Booster_Version, Launch_Site from SPACEXTBL where Landing_Outcome='Failure (drone ship)' and date like '2015%';

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

[75]:

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

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

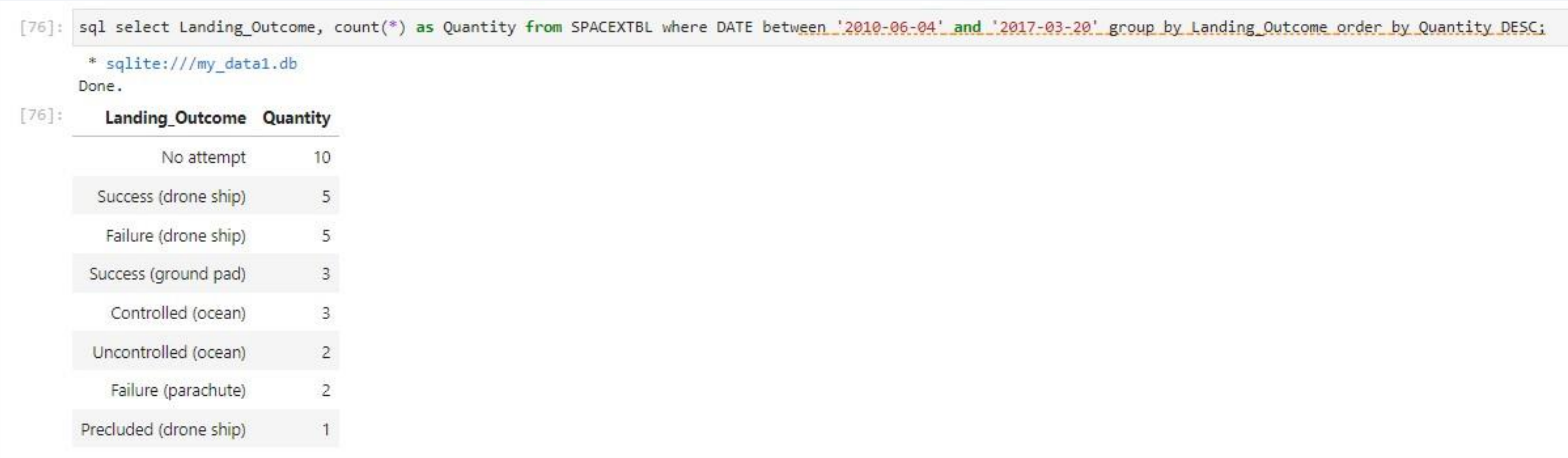
Used the COUNT, WHERE clause and the BETWEEN, GROUP by and ORDER BY to find the required.

COUNT of landing outcomes from the data.

WHERE clause to filter for landing outcomes **BETWEEN** to filter the by dates i.e. 2010-06-04 to 2010-03-20.

Applying the **GROUP BY** clause to group the landing outcomes.

Use the **ORDER BY** clause to order the grouped landing outcome in descending order.



```
[76]: sql select Landing_Outcome, count(*) as Quantity from SPACEXTBL where DATE between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by Quantity DESC;

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

[76]:

Landing_Outcome	Quantity
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

SpaceX Launch Sites

The SpaceX launch sites are mainly in the United State of America.

They are on the coastal region.

Closer to the seas.

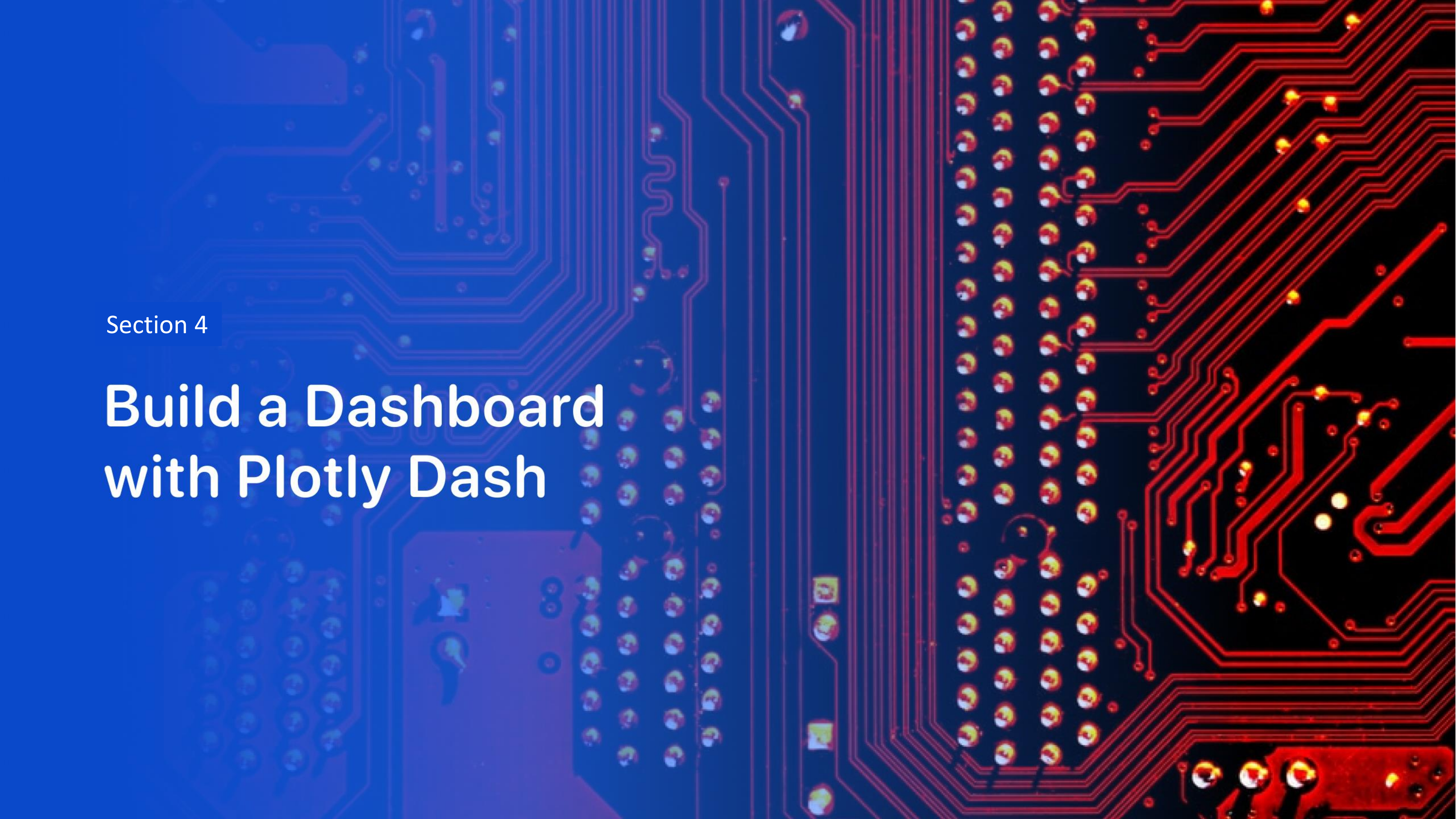
United States SpaceX Launch Sites

More successes were recorded under the Florida Launch.

SpaceX Launch Proximity

Launch sites are in close proximity to;

- Railway, highway and coastline – for easy access of transport and logistics of heavy equipment that maybe be required.
- They are far of main cities – as a risk management approach, to minimize fatality when they landing fail.

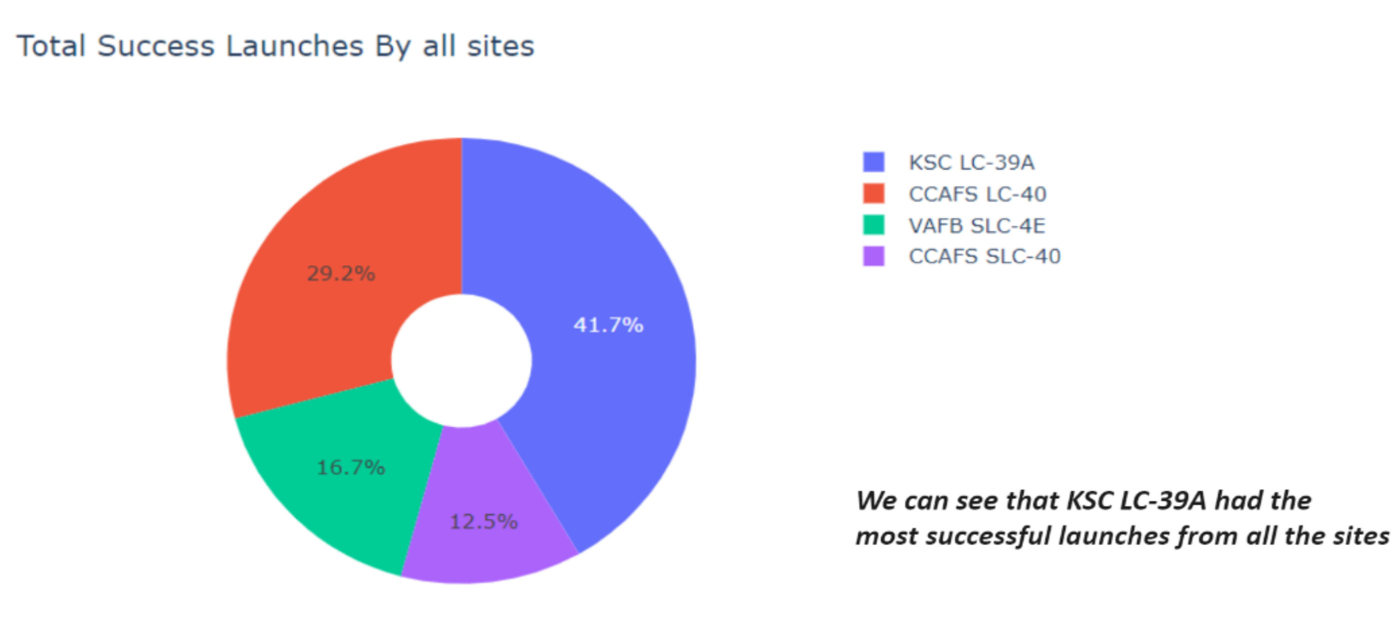


Section 4

Build a Dashboard with Plotly Dash

Successful Launches

- KSC LC-39A recorded the highest success rate for the period under study.

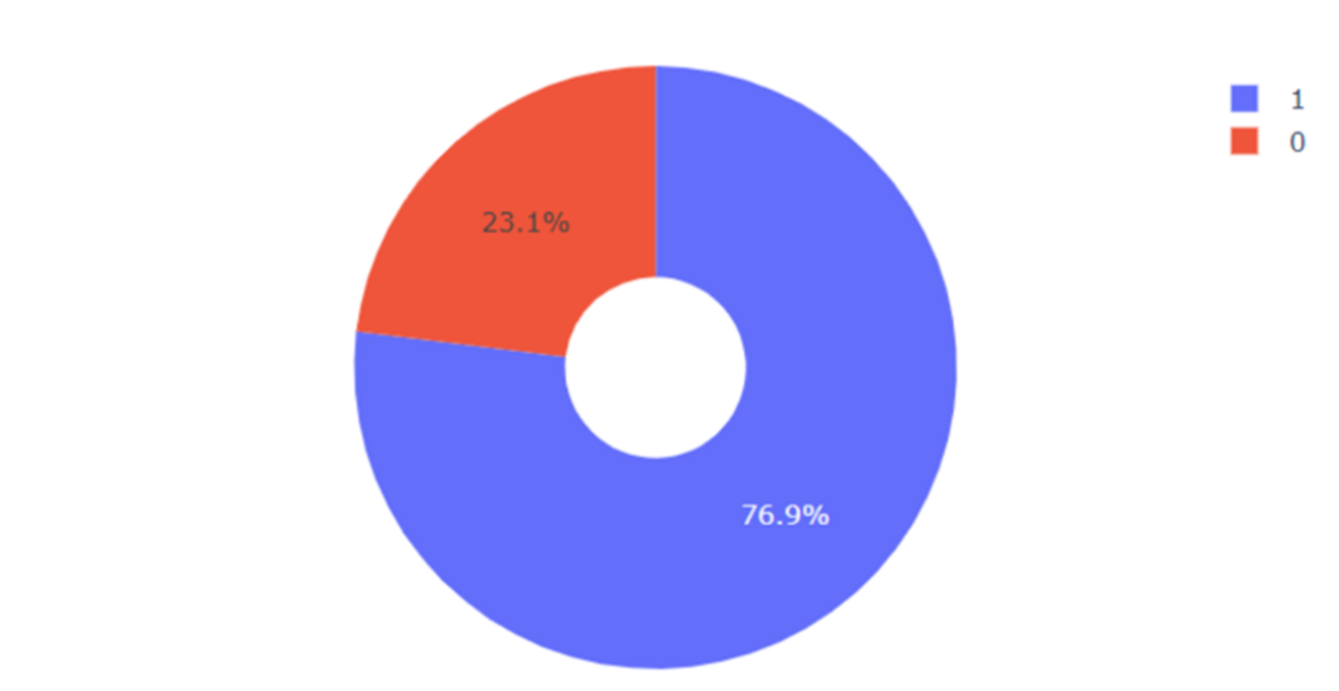


Total Success Launches By all sites

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

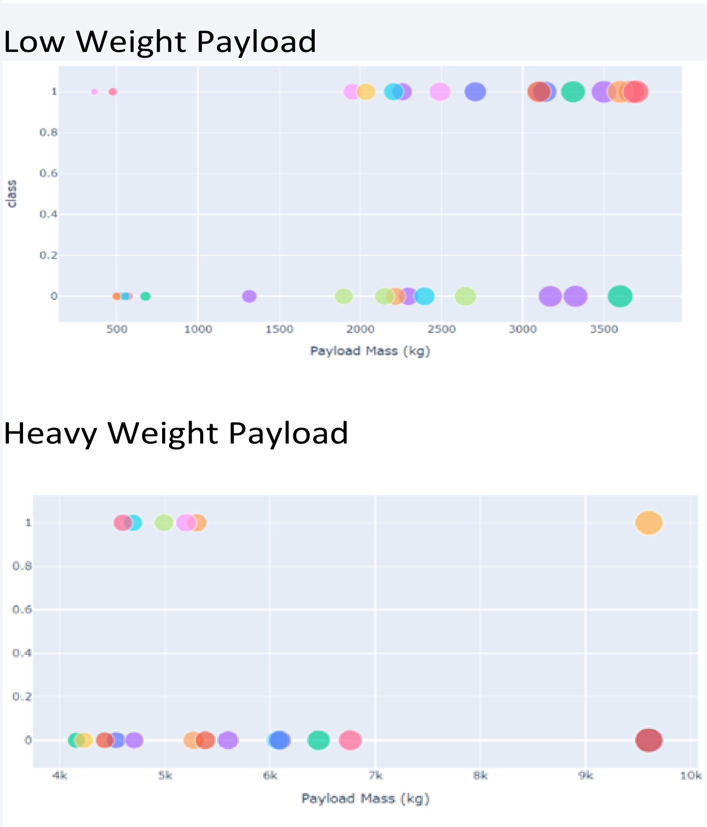
Success vs Failure Rate

KSC LC-39A had the highest success rate of 77% and 23 is failure rate.



Dashboard – Payload vs Launch Outcome

Success rates is high for low weight payloads compared to heavy weighted ones.



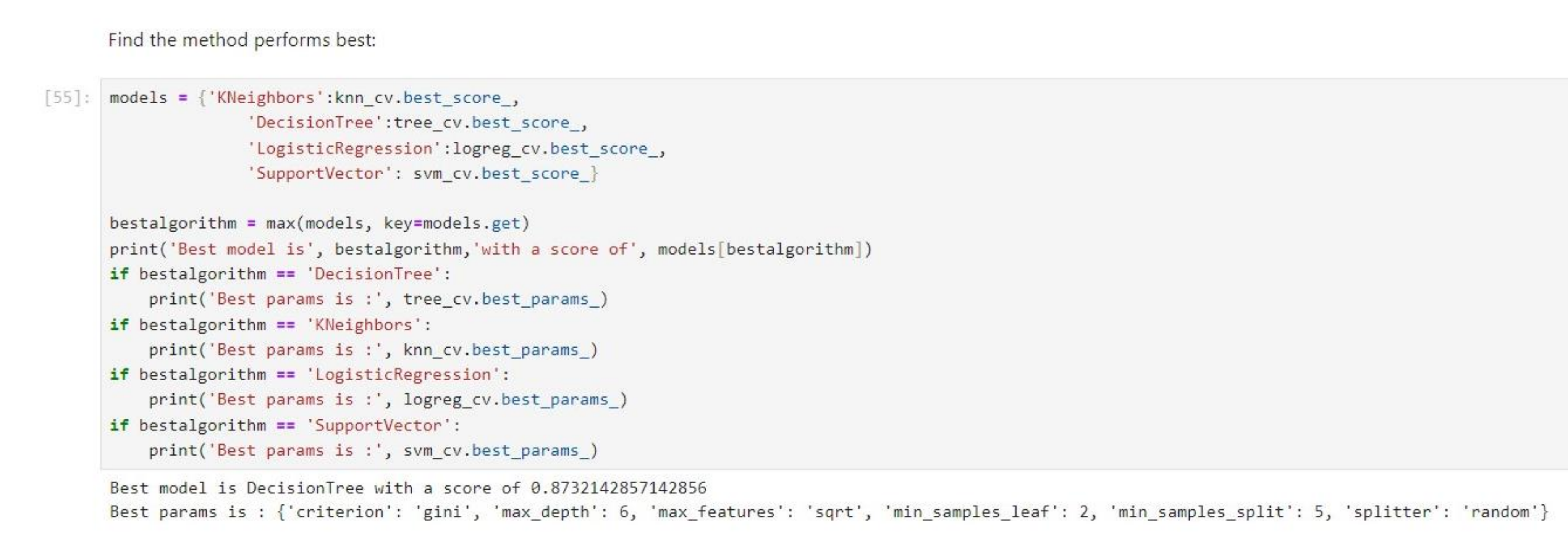
Low Weight Payload

Heavy Weight Payload

Section 6

Predictive Analysis (Classification)

Classification Accuracy



Find the method performs best:

[55]:

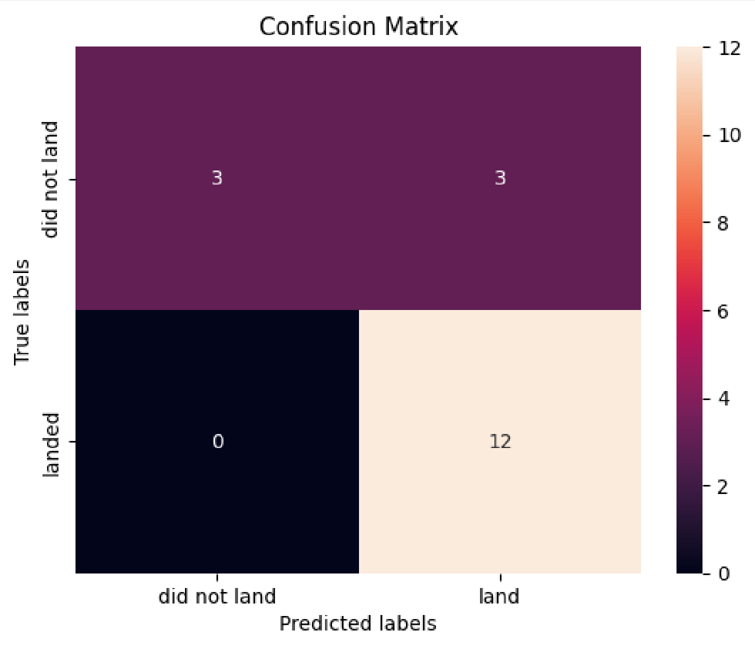
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

(https://github.com/Sibiya-K/IBM-Applied-Data-Science-Capstone.git)

Confusion Matrix

Confusion matrix for the decision tree classifier indicate that the classifier can distinguish between the different classes.



Conclusions

- Landing success increases as the flight counts increase at a launch site – this could be explained by the learning theory where each flight is improved from the previous success or failure.
- Landing launch success rate has been on an increase trend since 2013.
- Orbits with most success rates areES-L1, GEO, HEO, SSO and VLEO.
- KSC LC-39A had the most successful launches of all the other sites

Appendix & References

(https://github.com/Sibiya-K/IBM-Applied-Data-Science-Capstone.git)

Thank you!