IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ryan Kendrick

7July 2023

Outline

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

Executive Summary

- Data sources: Web scraped Wikipedia tables, SpaceX API
- Outlier payloads below 2000kg and above 6000kg greatly reduce the chance of successful recovery of the first stage.
- The success of SpaceX launches has risen steadily over time.
- Launch sites are placed near railways and coastlines, and far from cities.
- The Kennedy Space Center Launch Complex 39 has seen the highest rate of successful launches by a large margin at 76.9% - this is partly explained by its use for mid-weight payloads.

Introduction

This project aims to analyse SpaceX launches to reveal insights that would help a competing company get started in the industry.

SpaceX is able to save money by recovering the first stage of the rocket for reuse in subsequent launches.

We will train a machine learning model to predict whether a launch will be successful. In the process, we will identify what factors make for a high chance of a successful launch.

We will look at parameters like payload mass, orbit type and launch site to identify potential pitfalls for a competitor.

Section 1

Methodology

Methodology

Executive Summary

- Data collection:
  - SpaceX REST API requests & web scraping of Wikipedia tables
- Data wrangling:
  - Missing payload data was replaced by the mean weight of payloads where data was available
  - Launch outcome data was converted to a numeric 0/1 binary classification for unsuccessful and successful launches respectively
- Exploratory data analysis (EDA) using visualization and SQL
- Interactive visual analytics using Folium and Plotly Dash
- Predictive analysis using classification models:
  - Classification model K-Nearest Neighbours chosen through cross-validation, grid search & scoring

Data Collection

Datasets were created using a combination of requests to the publicly available SpaceX REST API and by using the Beautiful Soup Python library for web scraping additional launch data contained in tables on Wikipedia.

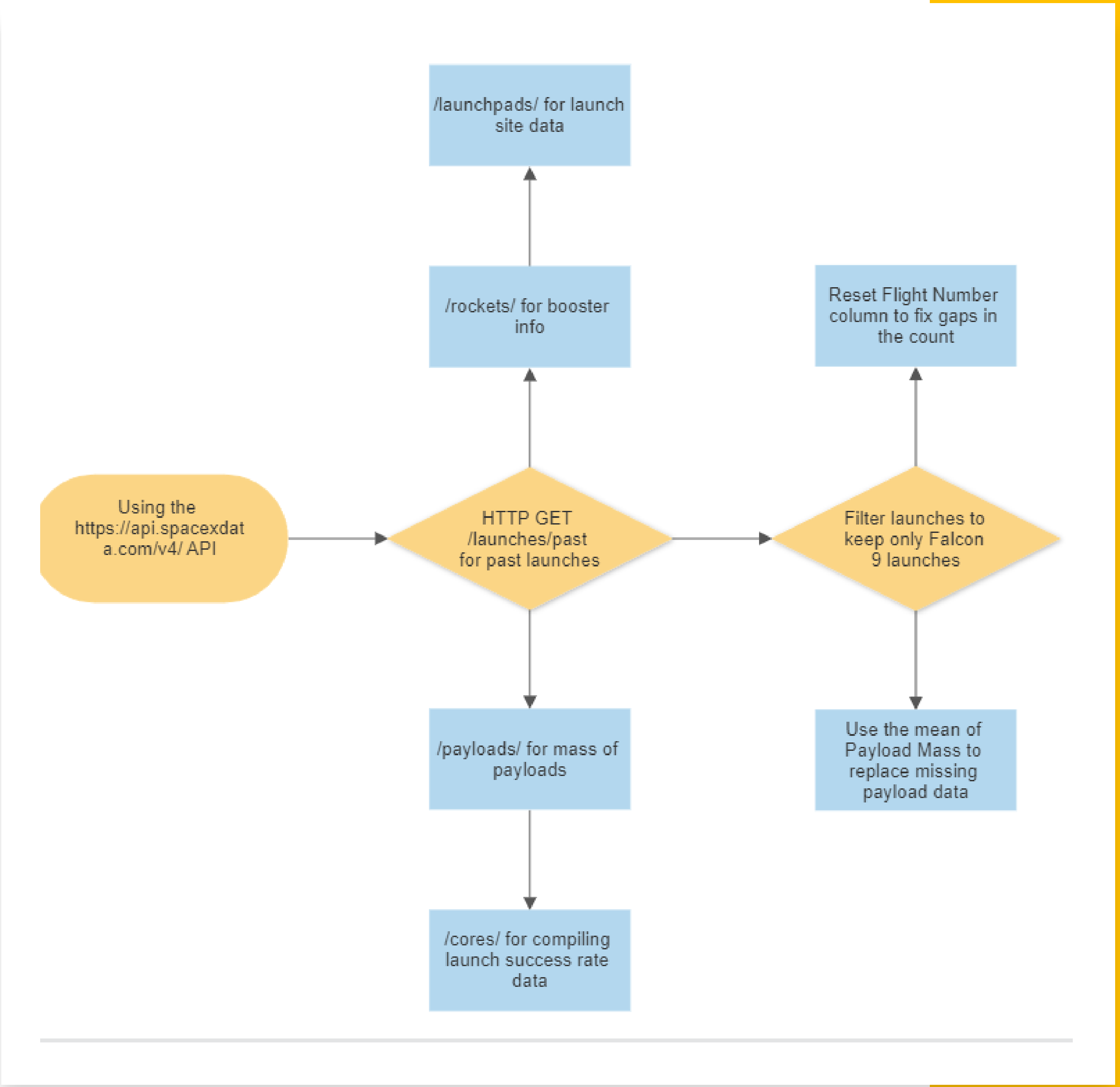
Collecting data from the SpaceX API involved a multi-step process of requesting information to request further information - i.e. requesting data on past launches to identify a rocket and then calling the /booster/ API to gathering information on the booster

Data Collection – SpaceX API

- Flowchart of the data collection process for the SpaceX API

- See the Jupyter Notebook where this was process was done here:

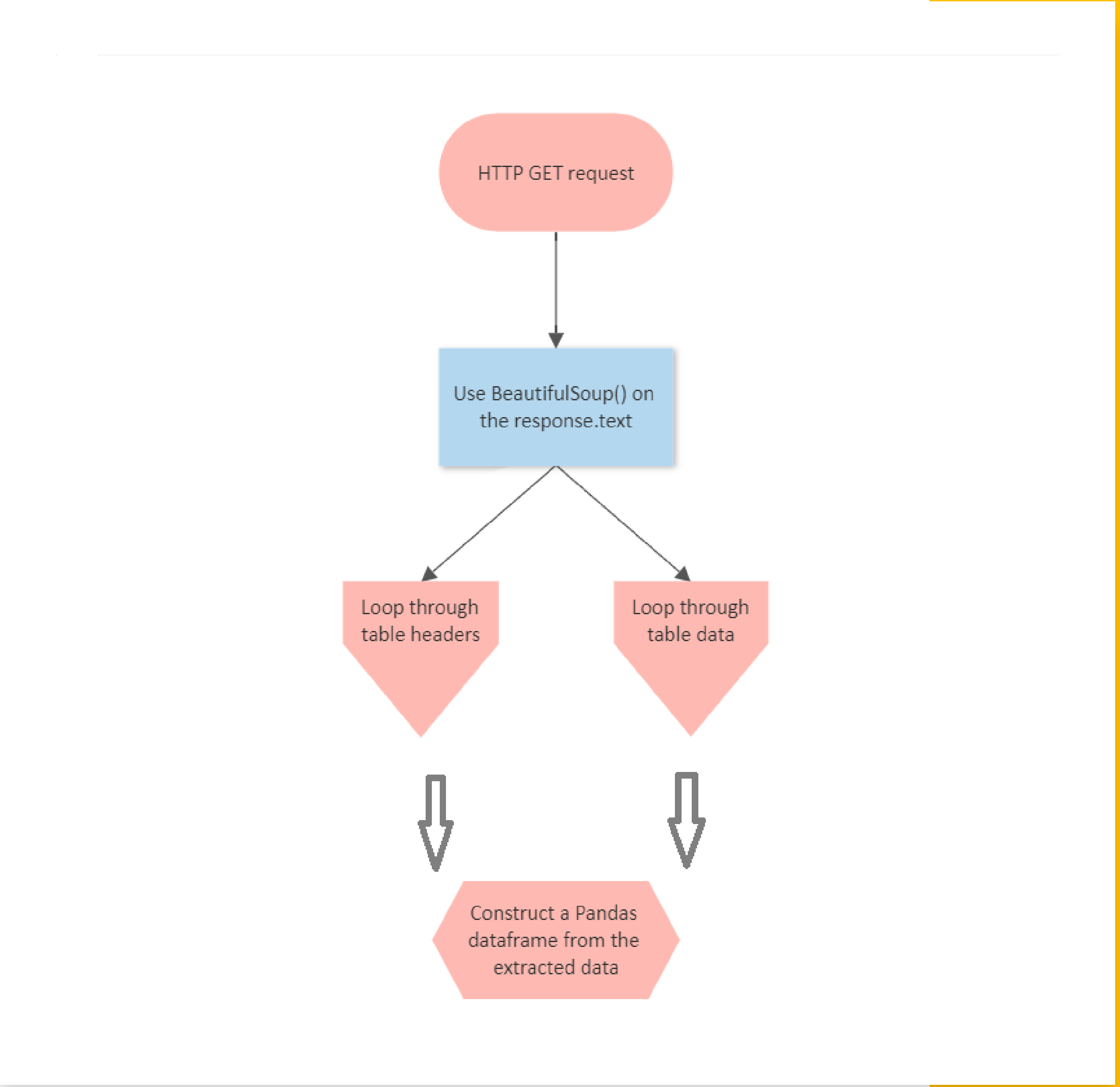
https://github.com/Ryan-Kendrick/ibm-final/blob/main/notebooks/jupyter-labs-spacex-data-collection-api.ipynb


Using the https://api.spacexdata.com/v4/ API
HTTP GET /launches/past for past launches
/rockets/ for booster info
/launchpads/ for launch site data
/payloads/ for mass of payloads
/cores/ for compiling launch success rate data
Filter launches to keep only Falcon 9 launches
Reset Flight Number column to fix gaps in the count
Use the mean of Payload Mass to replace missing payload data

Data Collection – Wikipedia Table Scraping

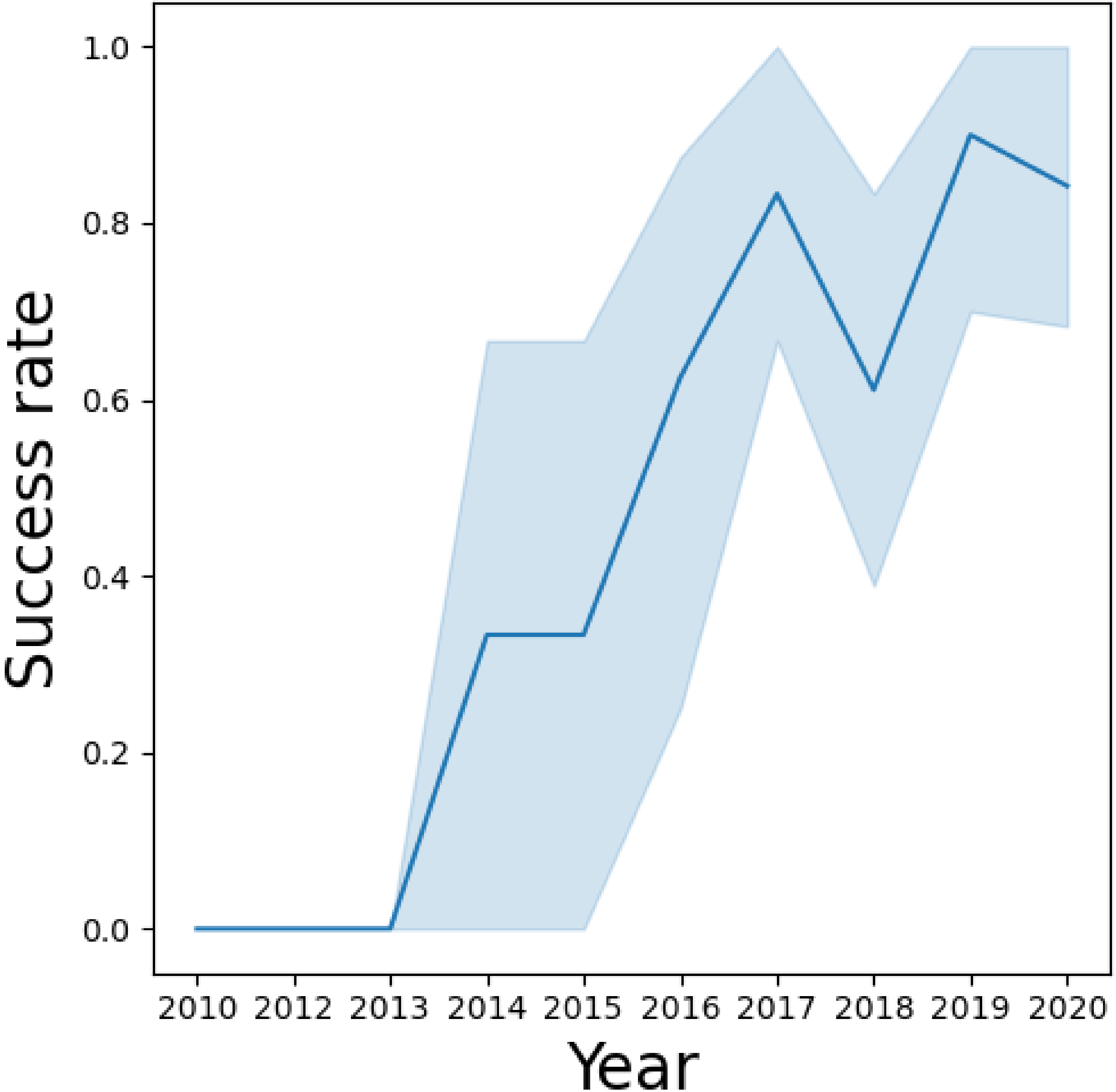
- Flowchart of the data collection process for web scraping tables

- See the Jupyter Notebook where this was process was done here:
- https://github.com/Ryan-Kendrick/ibm-final/blob/main/notebooks/jupyter-labs-spacex-data-collection-api.ipynb


HTTP GET request
Use BeautifulSoup() on the response.text
Loop through table headers
Loop through table data
Construct a Pandas dataframe from the extracted data

Data Wrangling

- The data collected gave landing outcomes as categorical data denoting on what type of landing pad the flight would land and if it was successful
- Data was wrangled to be simpler to work with by looping through the outcome dataframe and reassigning a value of 1 for successful and 0 for unsuccessful
- See the notebook here:
- https://github.com/Ryan-Kendrick/ibm-final/blob/main/notebooks/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb



EDA with Data Visualization

- A scatter plot was used to view correlations between payload mass and flight number and it was found that later flights were more successful than earlier flights.
- It was also found that the heavier the payload, the less likely the first stage will be successfully retrieved
- The line chart on the left shows the trend of increasing success with flights over time
- See notebook here:

https://github.com/Ryan-Kendrick/ibm-final/blob/main/notebooks/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

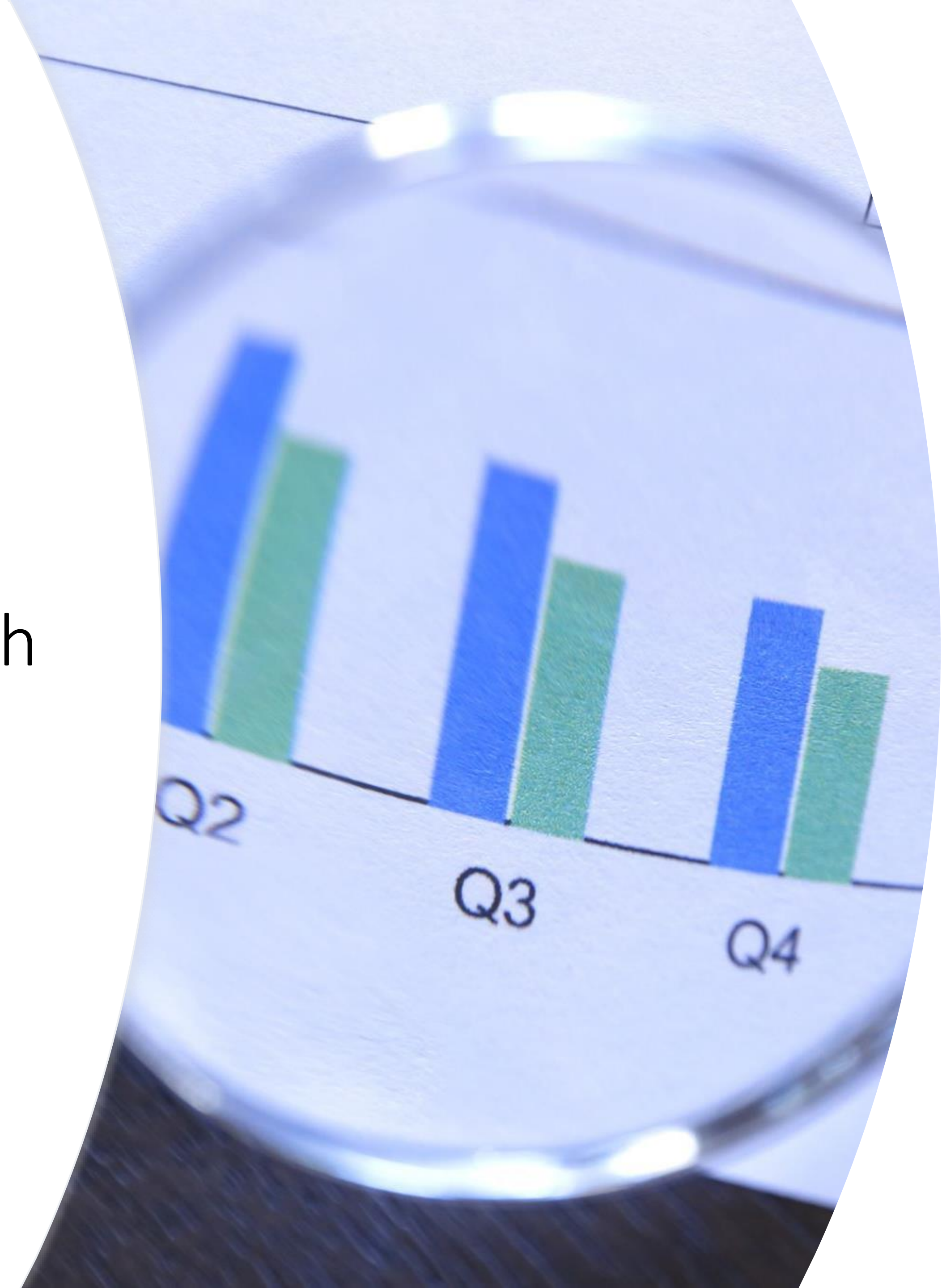
EDA with SQL

- SQL queries were used in one lab to extract data from specific date ranges with specific launch outcomes
- For example, in the last query, we return a count of landing outcomes between June 2010 and March 2017 grouped by type of outcome
- See the notebook here:
- https://github.com/Ryan-Kendrick/ibm-final/blob/main/notebooks/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Building an Interactive Map with Folium

- Circles were added to a Folium map indicating the location of each launch site
- Markers were used to indicate the success/failure of launches at the site
- Lines were added to indicate the nearest railway, city and coastline
- Highlighting these geographic features revealed that launch sites are located near railways and coastlines, and a sufficient distance from cities (~50km)
- See the notebook here:
- https://github.com/Ryan-Kendrick/ibm-final/blob/main/notebooks/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Building a Dashboard with Plotly Dash



- A Dash application was built for interactive data visualisation
- A pie chart was used to illustrate the number of successful launches at each launch site as a proportionate relationship to other launch sites
- Individual launch sites could be selected to visualise successful vs. unsuccessful launches at that site
- A scatter plot was made with an accompanying payload mass slider to get an idea of what happens to launch success when different ranges are excluded
- The python code for the app can be viewed here:
- https://github.com/Ryan-Kendrick/ibm-final/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Cross validation with GridSearchCV was used to find the optimal parameters for our prediction model
- The outcome column "class" was converted to a numpy array to be used as the Y variable and the collected data was standardised for the X variable
- Confusion matrices and the scikit-learn score method were used to identify the most accurate combination of model and parameters
- See the notebook here:

https://github.com/Ryan-Kendrick/ibm-final/blob/main/notebooks/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

EXPLORATORY DATA ANALYSIS RESULTS

INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

PREDICTIVE ANALYSIS RESULTS

Section 2

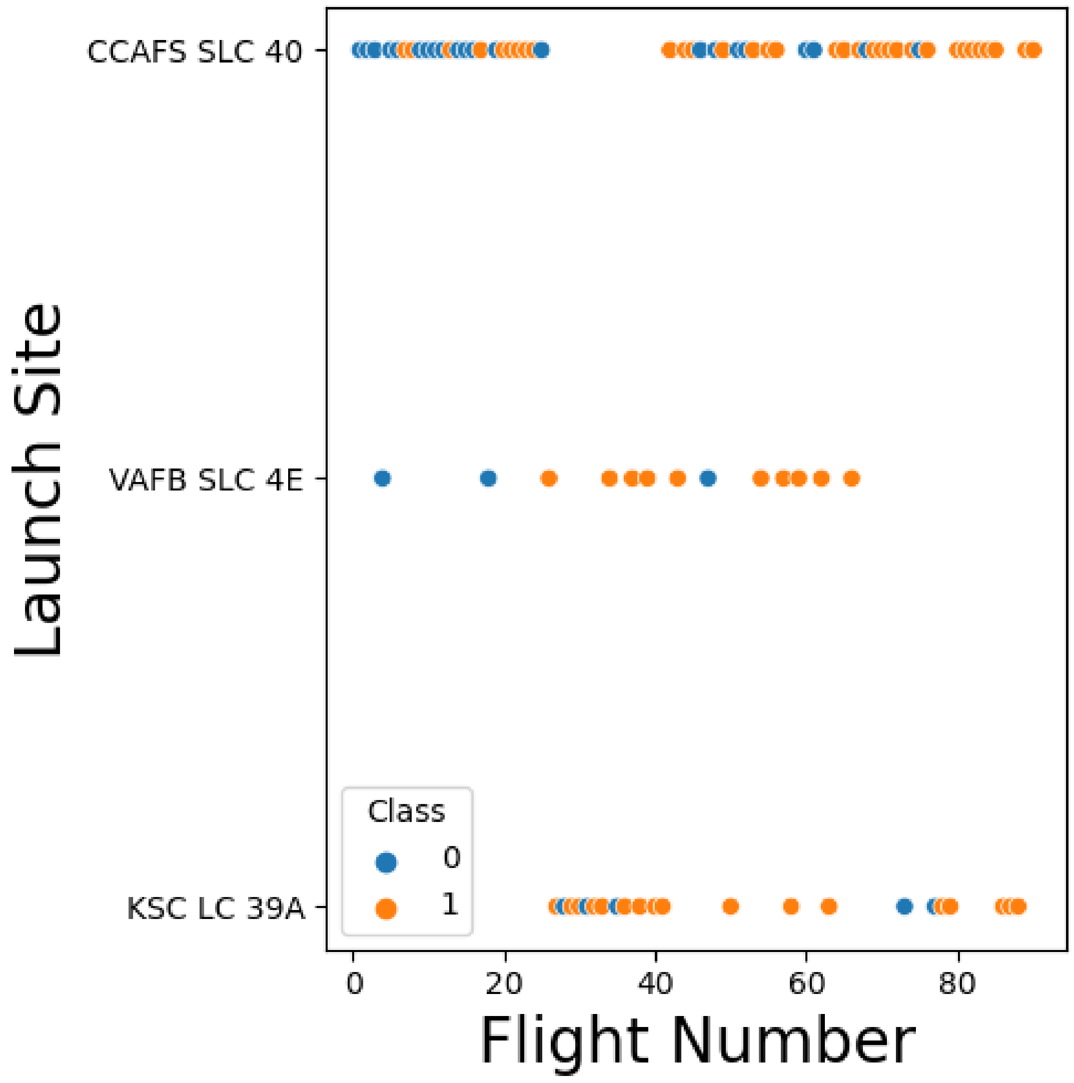
Insights drawn from EDA

Flight Number vs. Launch Site

- The Cape Canaveral Space Launch Complex 40 has been a popular launch site from the early failures to the late successes.

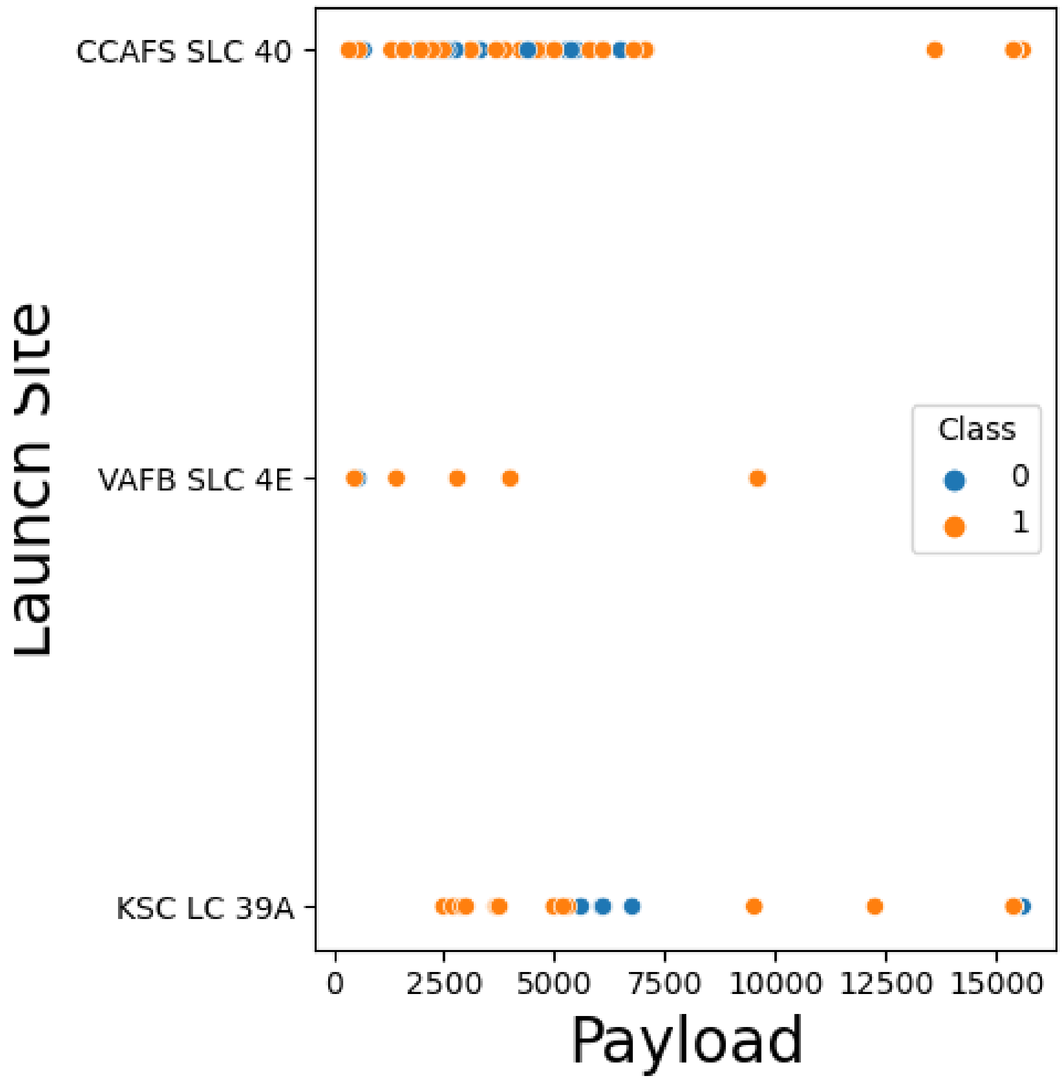
- The Vandenberg Space Launch complex was retired despite a string of successes.

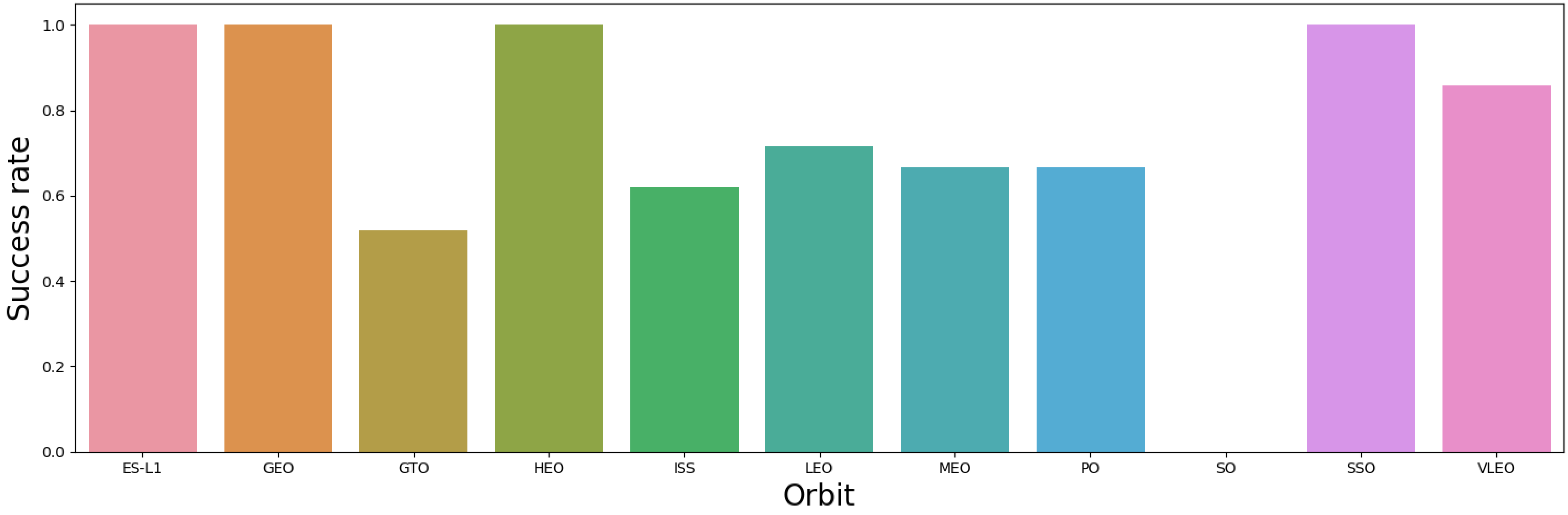
- The Kennedy Space Center Launch Complex 39A replaced CCAFS SLC 40 for a time before settling in as a regular launch site with an impressive success rate.



Payload vs. Launch Site

- The Cape Canaveral Space Launch Complex 40 has been a popular choice for outlier payload weights.
- Vandenberg has seen a strong bias towards lighter payloads.
- The Kennedy Space Centre has a bias towards mid-weight payloads.





Success Rate vs. Orbit Type

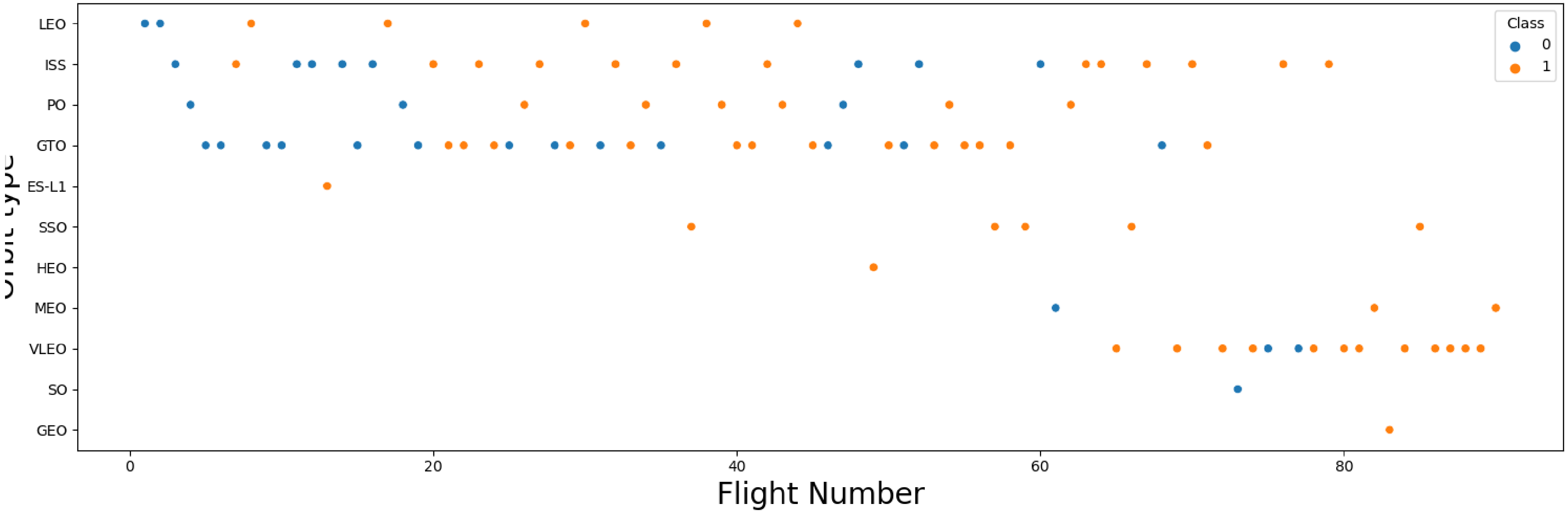
Notable findings from graphing orbit type vs. Success rate include the fact that L1-earth-sun, geosynchronous and highly elliptical orbits have a 100% success rate.

SSO appears to have a 100% success rate but this may be an inconsistency in the data due to SO orbits also referring to sun-synchronous orbits

Flight Number vs. Orbit Type

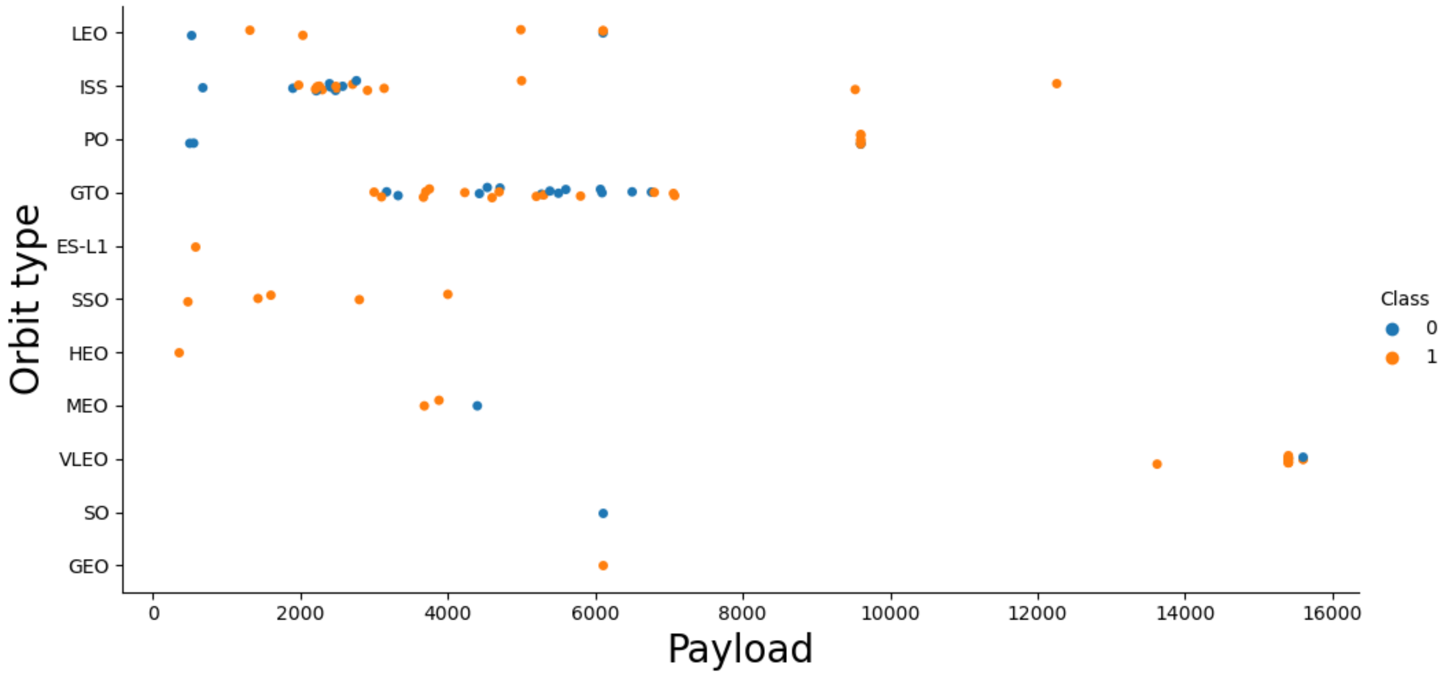
We found that early on, SpaceX worked to perfect LEO, ISS, PO & GTO orbit types.

Later, SpaceX specialised in very low earth orbits and infrequently experimented with other orbit types.

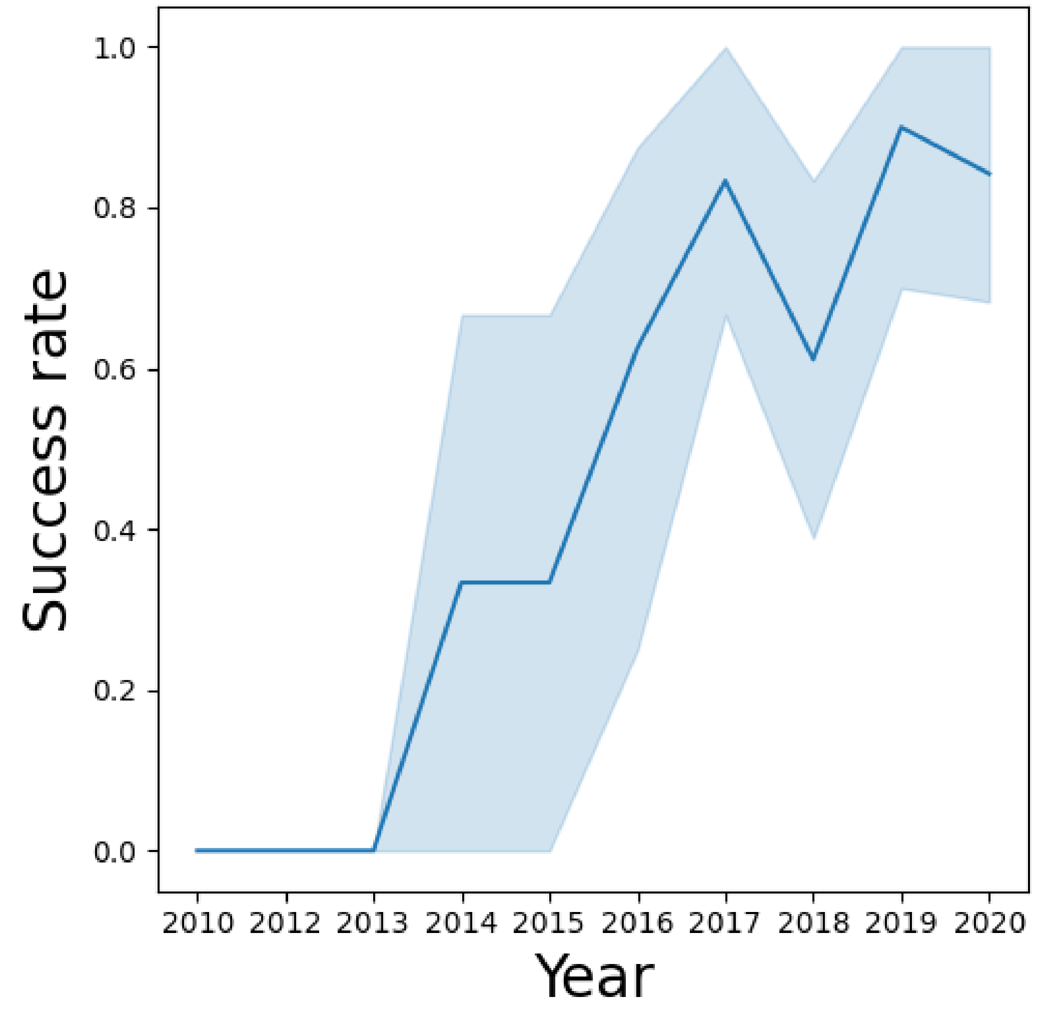


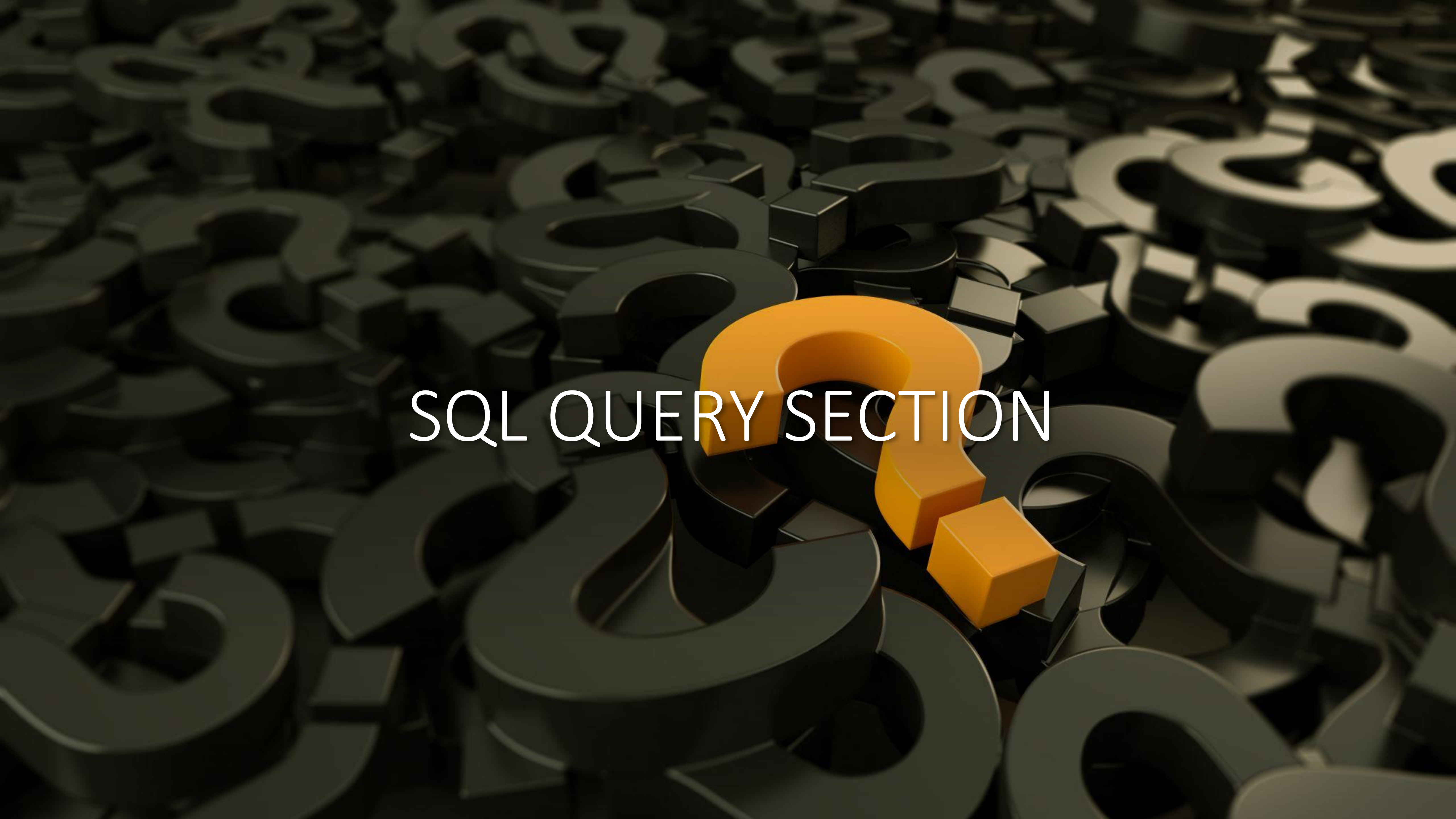
Payload vs. Orbit Type

- Very low earth orbit is the only orbit type SpaceX uses for extremely heavy payloads
- Lighter payloads are able to be sent on a wide variety of orbit types
- Geostationary transfer orbits are common for payloads on the light-to-medium end of the spectrum



Launch Success Yearly Trend



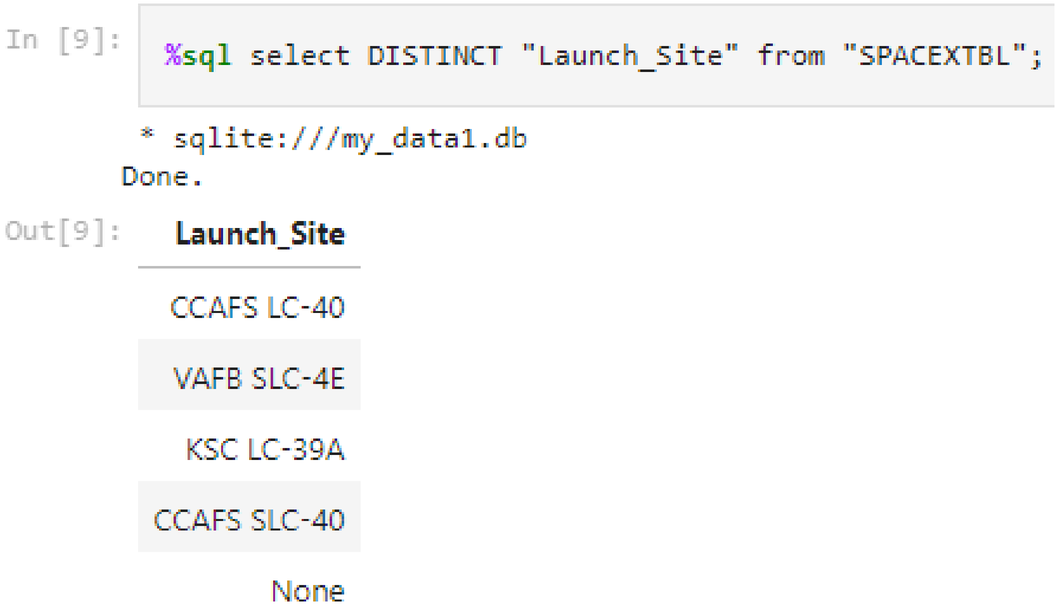


SQL QUERY SECTION

All Launch Site Names

Names of the unique launch sites.

A SQL query grabbing unique entries from the Launch_Site column of the SPACEXTBL database table.



```
In [9]:  %sql select DISTINCT "Launch_Site" from "SPACEXTBL";
```

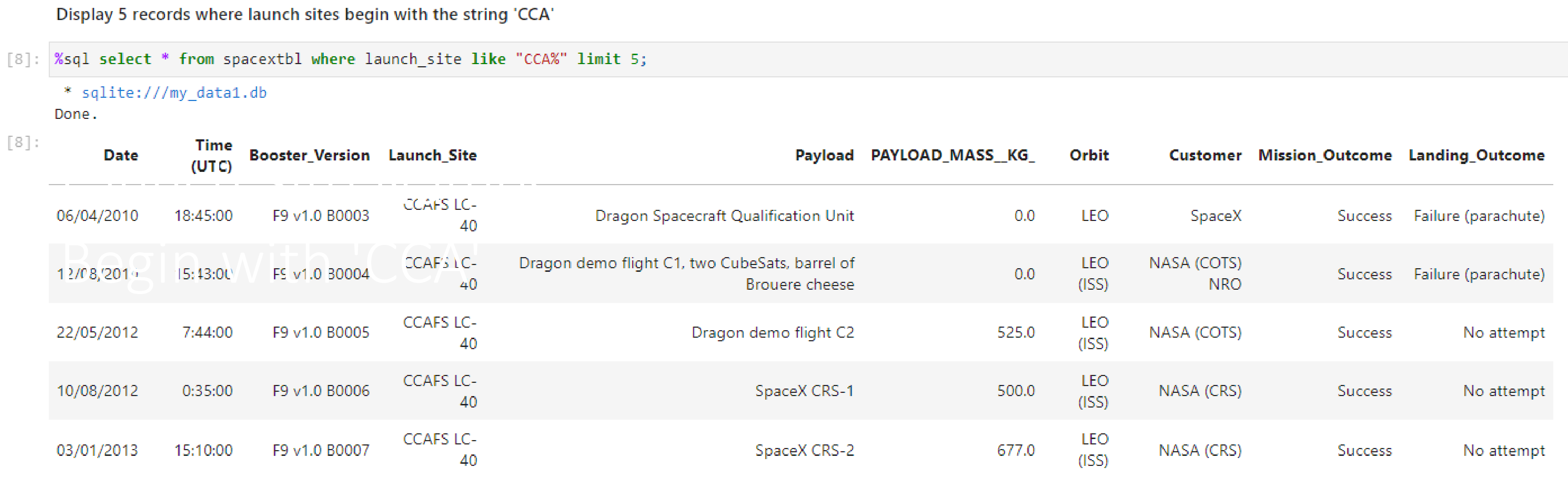
```
 * sqlite:///my_data1.db
Done.
```

Out[9]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Beginning with 'CCA'

Using the like operator to return CCAFS launch sites by finding those beginning with "CCA"



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

```
[8]: %sql select * from spacextbl where launch_site like "CCA%" limit 5;
```

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

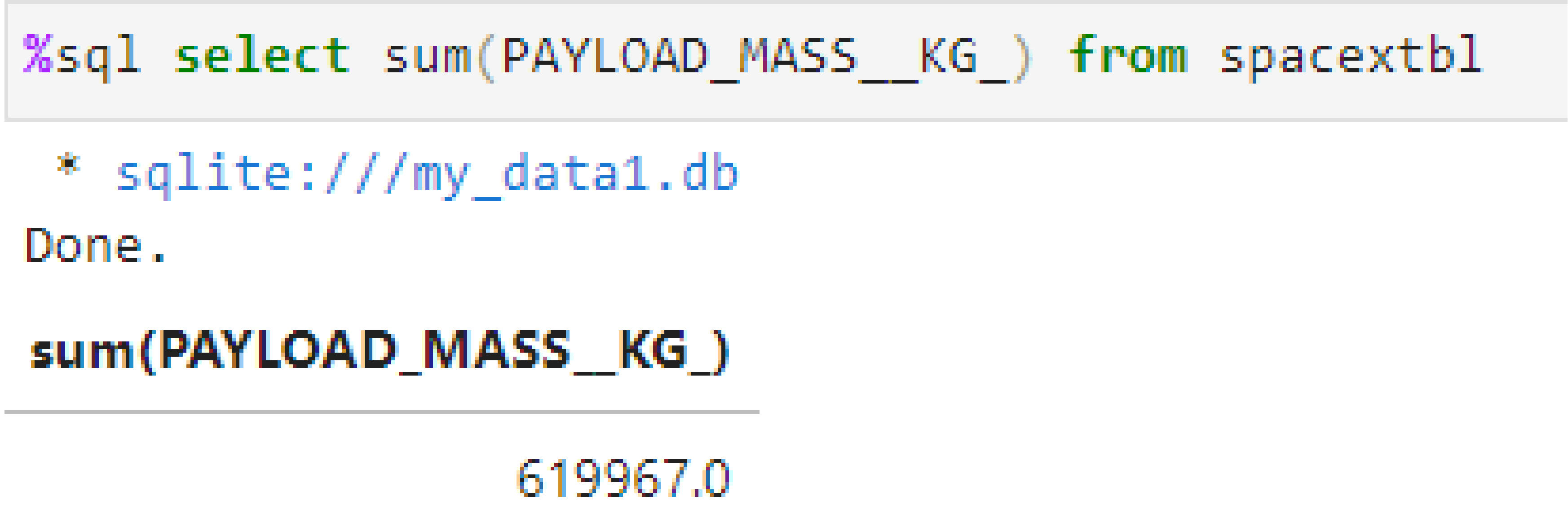
[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Total payload carried by boosters.

Using the sum function to add all payload mass entries.



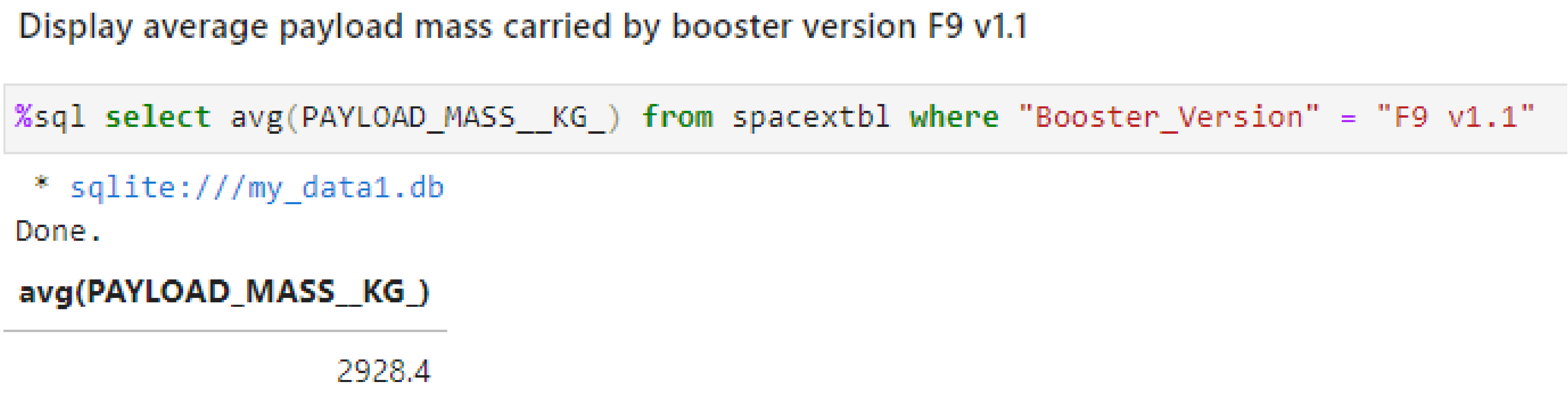
```
%sql select sum(PAYLOAD_MASS__KG_) from spacextbl
```

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

sum(PAYLOAD_MASS__KG_)
619967.0

Average Payload Mass by F9 v1.1

- Calculating the average payload mass carried by booster version F9 v1.1.
- Using the avg function with a select statement along with a where clause to return the average payload mass for F9 V1.1.



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

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

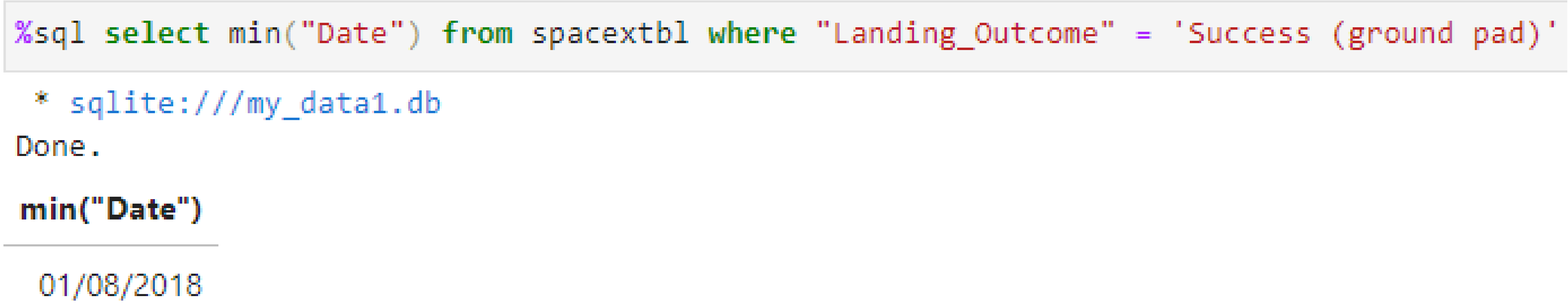
```
 * sqlite:///my_data1.db
Done.
```

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

Date of the first successful landing outcome on a ground pad.

Using the min() function to return the earliest date of a successful entry



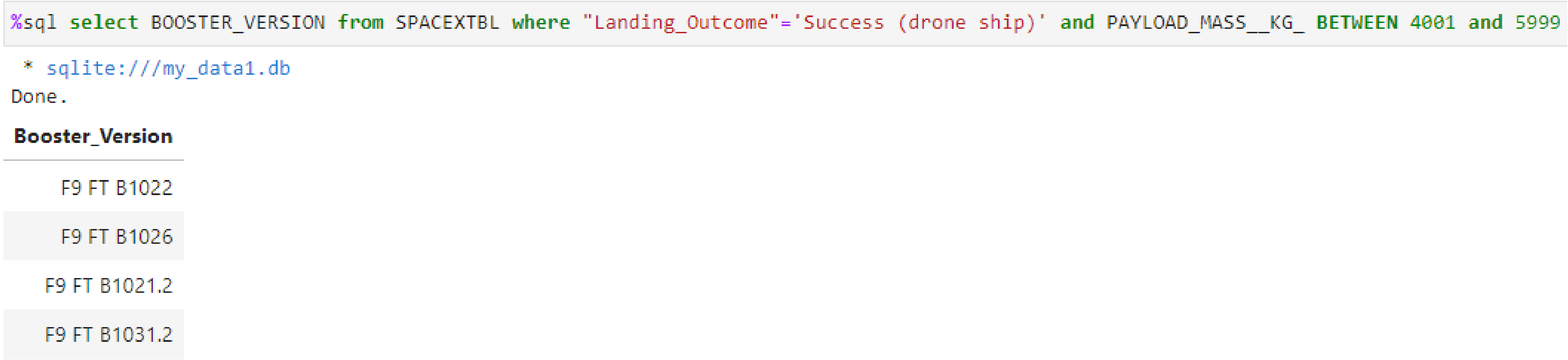
```
%sql select min("Date") from spacextbl where "Landing_Outcome" = 'Success (ground pad)'
```

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

min("Date")
01/08/2018

Successful Drone Ship Landing with Payload greater than 4000 and less than 6000

Using a more complex query containing a between comparison to return the names of boosters which have successfully landed on a drone ship and had payload mass greater than 4000 but less than 6000



```
%sql select BOOSTER_VERSION from SPACEXTBL where "Landing_Outcome"='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4001 and 5999
```

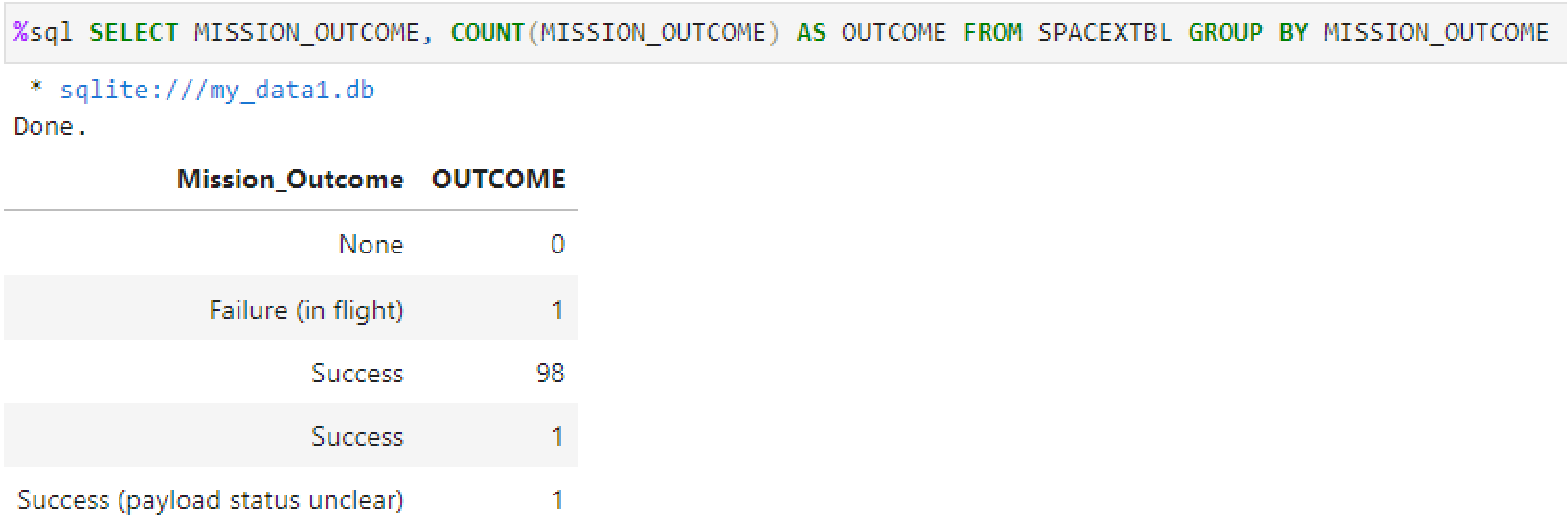
```
 * sqlite:///my_data1.db
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

Calculating the total number of success and failure mission outcomes

Using a "group by" to count mission outcomes



```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS OUTCOME FROM SPACEXTBL GROUP BY MISSION_OUTCOME
```

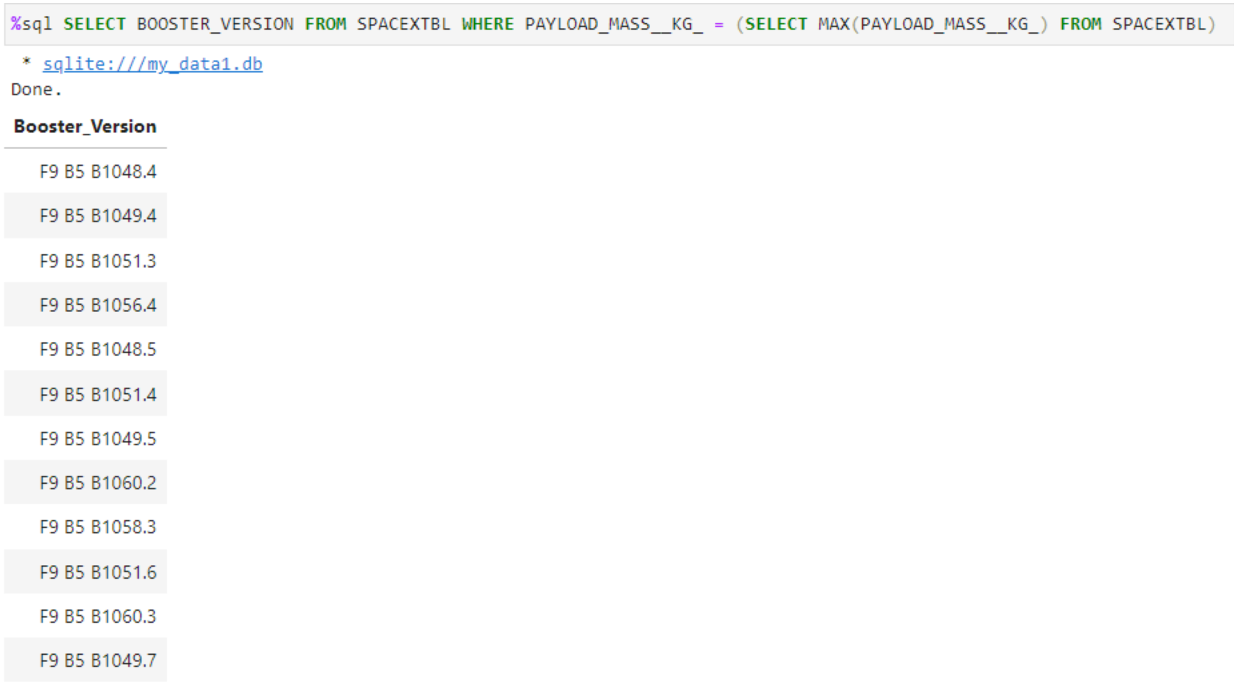
```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	OUTCOME
None	0
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carrying the Maximum Payload

Names of the boosters which have carried the maximum payload mass

Using a subquery to select only boosters for the heaviest payload



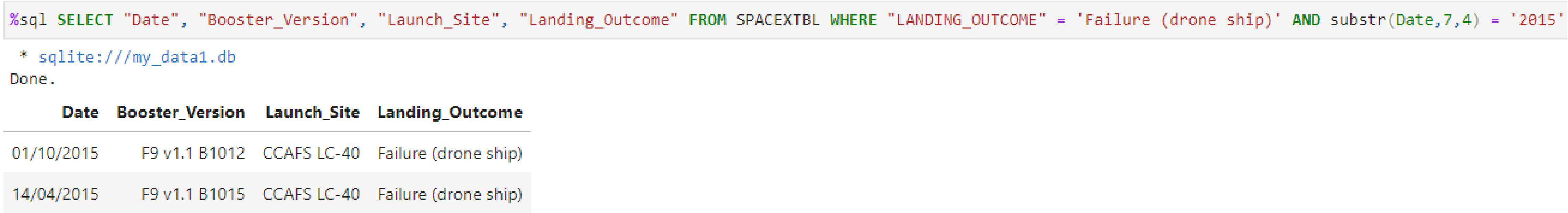
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * 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

Failed landing_outcomes for drone ships, their booster versions, and launch site names in the year 2015

Using a substring function to extract the year and compare it to the string '2015' to return only records for the year 2015



```
%sql SELECT "Date", "Booster_Version", "Launch_Site", "Landing_Outcome" FROM SPACEXTBL WHERE "LANDING_OUTCOME" = 'Failure (drone ship)' AND substr(Date,7,4) = '2015'
```

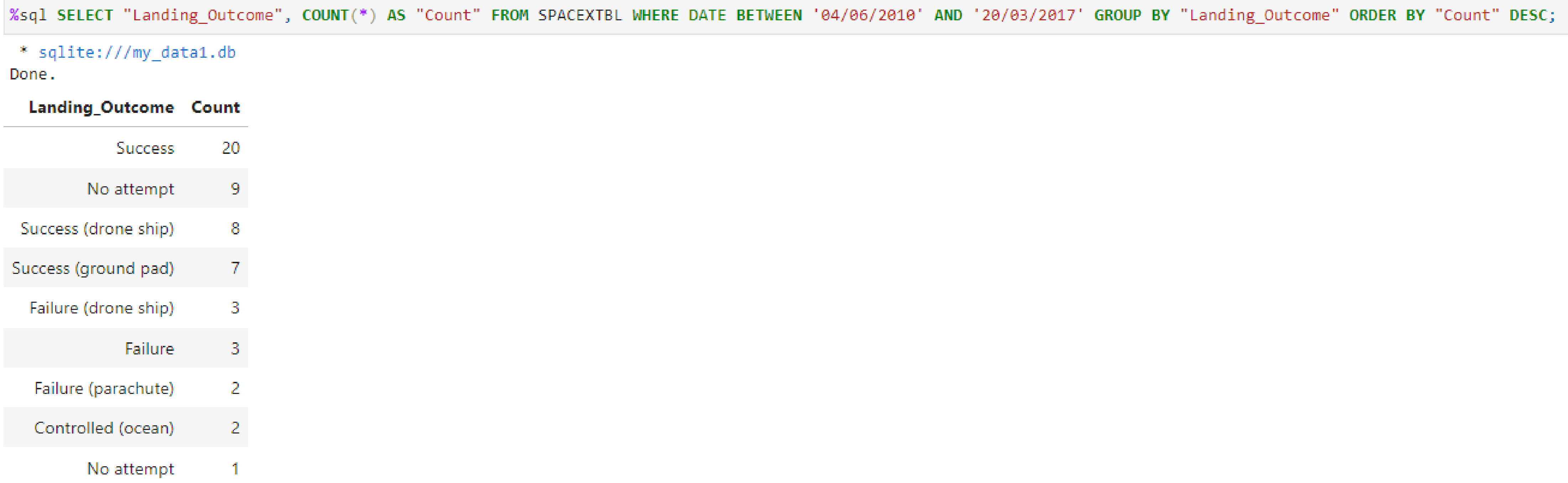
```
 * sqlite:///my_data1.db
Done.
```

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

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

- A complex query using count, where, date, between, group by, and order by operations to count landing outcomes for the provided date and list in descending order

- Adjustments had to be made from the course materials to account for local date formatting



```
%sql SELECT "Landing_Outcome", COUNT(*) AS "Count" FROM SPACEXTBL WHERE DATE BETWEEN '04/06/2010' AND '20/03/2017' GROUP BY "Landing_Outcome" ORDER BY "Count" DESC;
```

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

Landing_Outcome	Count
Success	20
No attempt	9
Success (drone ship)	8
Success (ground pad)	7
Failure (drone ship)	3
Failure	3
Failure (parachute)	2
Controlled (ocean)	2
No attempt	1

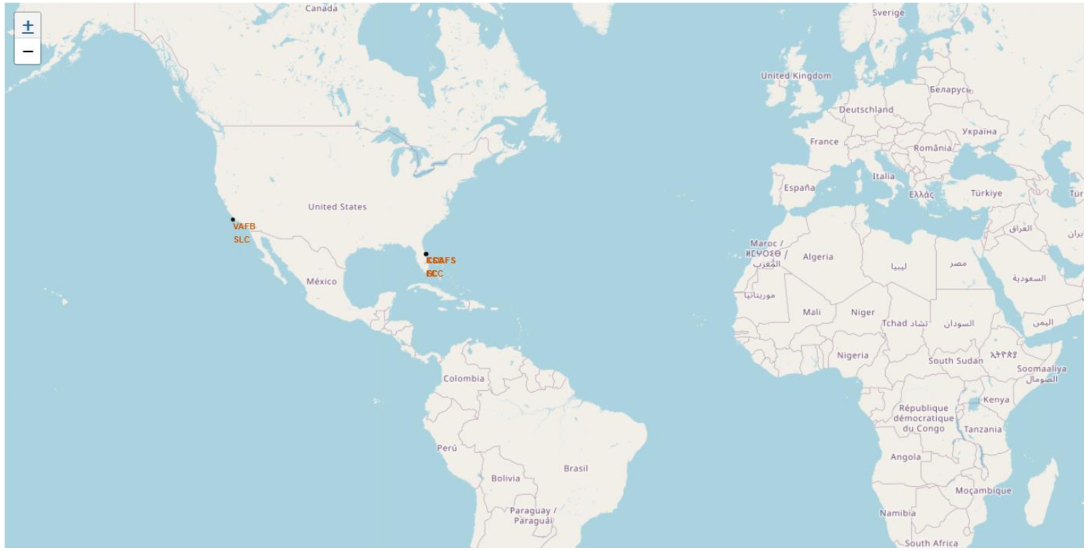
SQL QUERY SECTION ENDS

Section 3

Launch Sites Proximities Analysis

Launch site map

- The two CCAFS launch sites are located on the East Coast of the United States
- Adjacent is the KSC launch site
- On the West Coast sits the VAFB launch site.

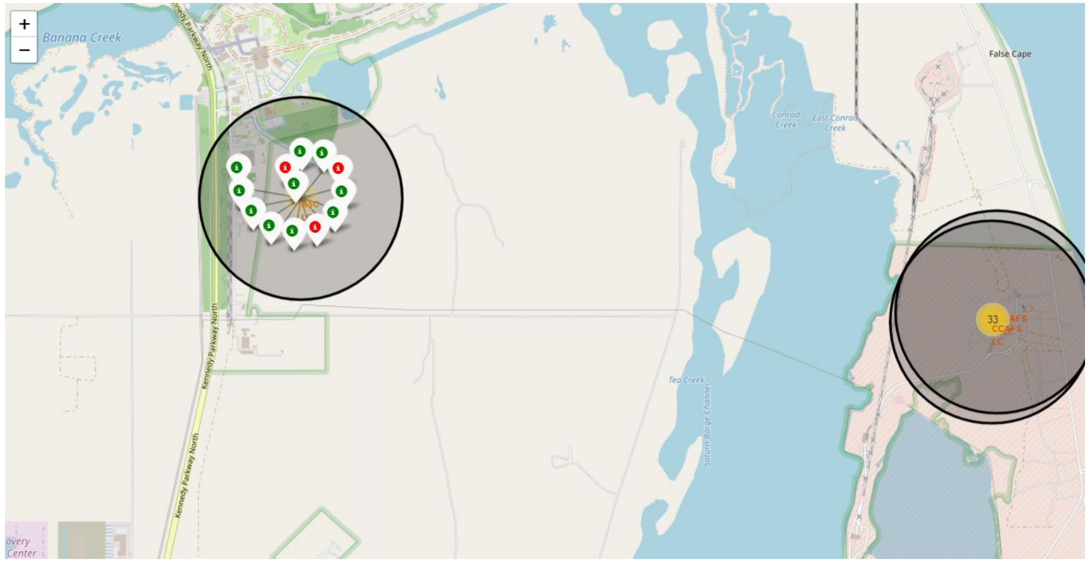


Kennedy Space Centre Launches

- The following map shows successful launches marked in green, and failed launches marked in red.

- The green markers indicate a launch from the Kennedy Space Centre where the first stage was successfully recovered.

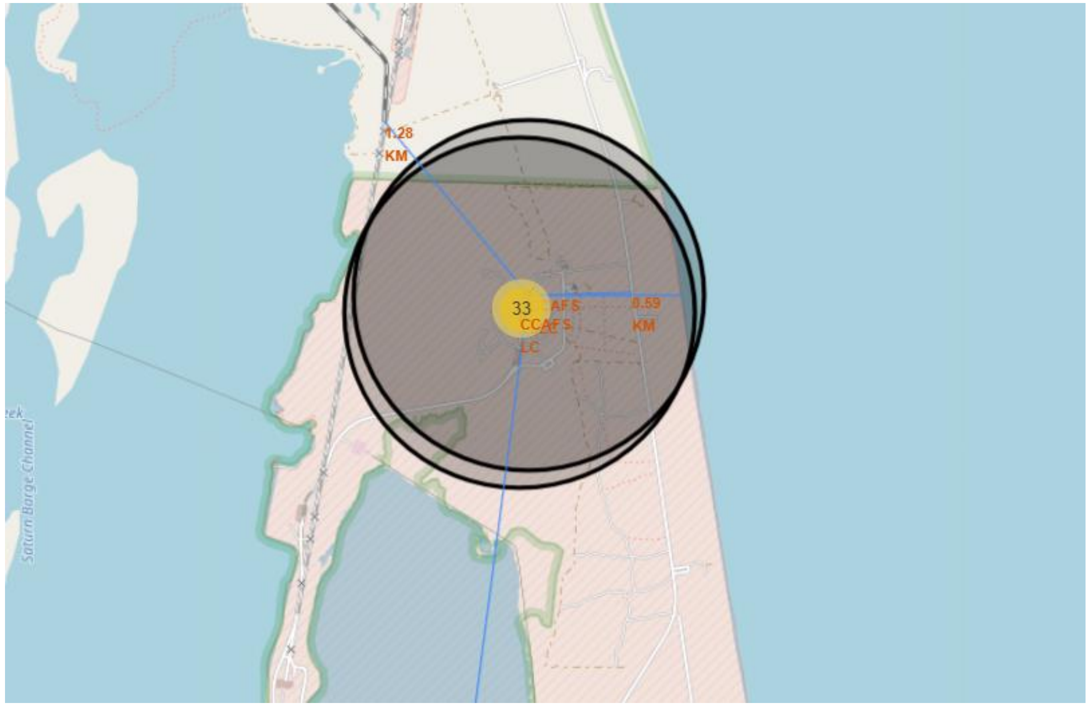
- The same functionality is available for the other launch sites.



Distance to the coast, railway & nearest city from CCAFS SLC

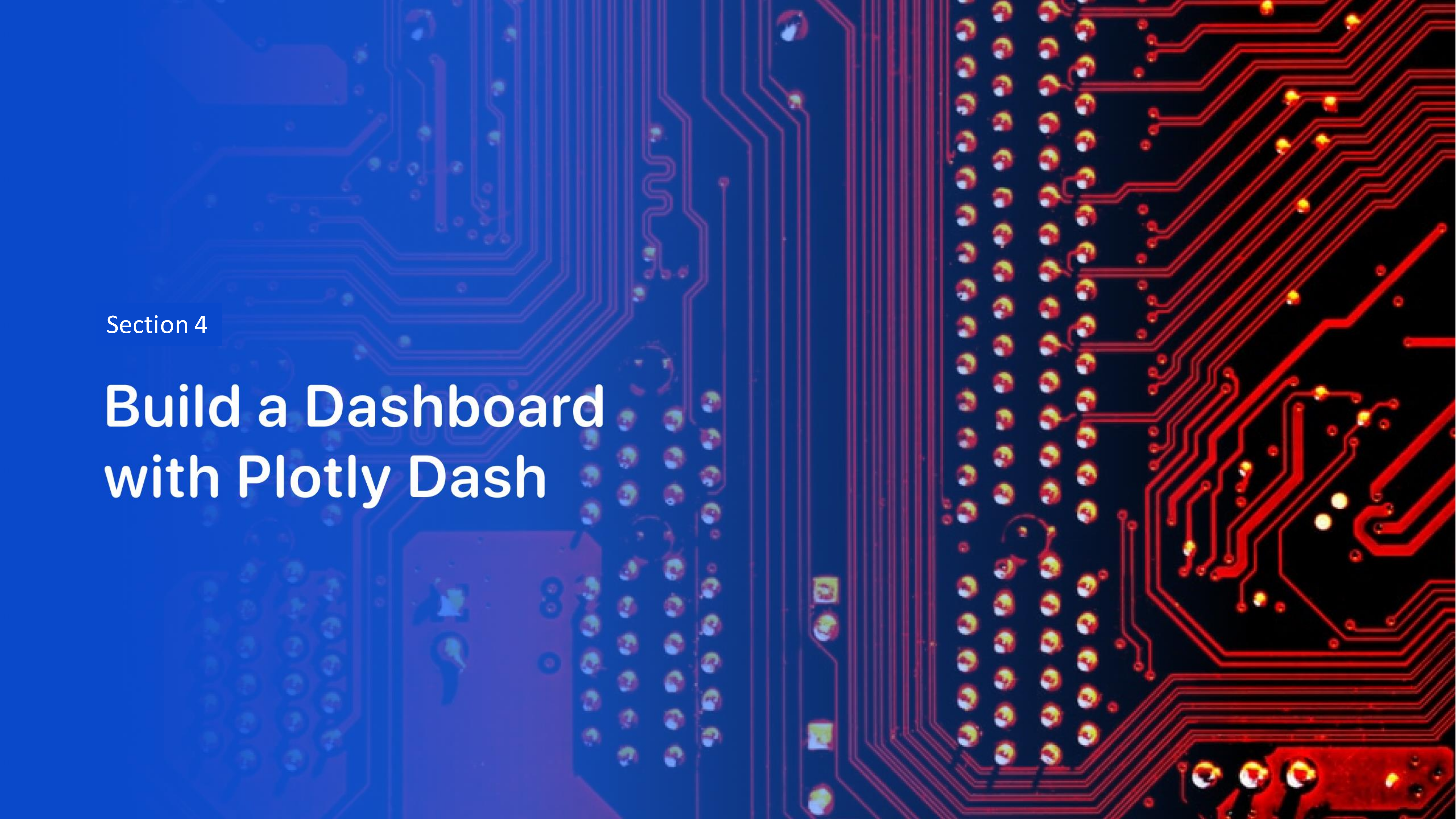
- The map shows the proximity to the nearest coast, railway and city (measured in km).

- The nearest city (not pictured here) is 50KM away from CCAFS SLC.



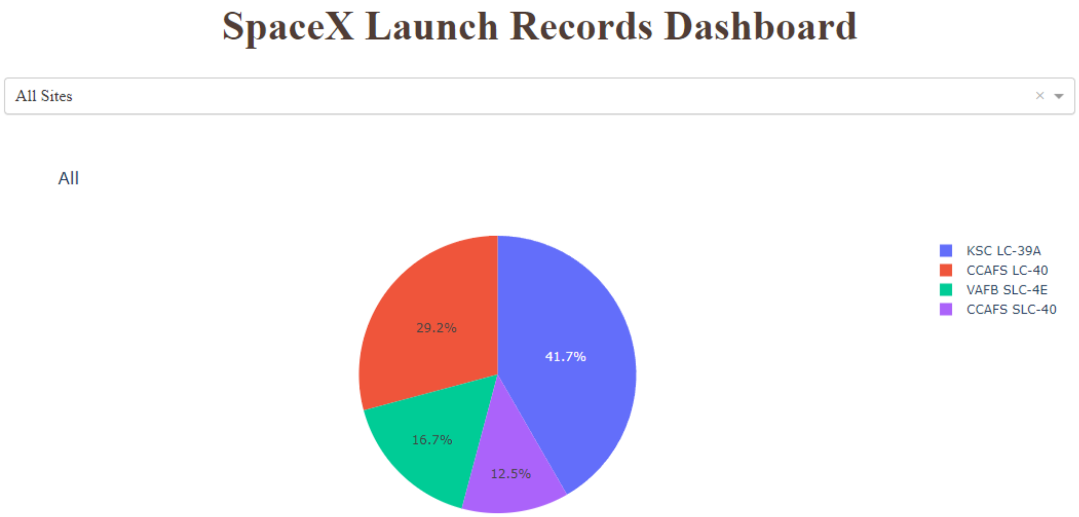
Section 4

Build a Dashboard with Plotly Dash



Launch sites as a proportion of successful launches

- The pie chart from the Dash dashboard shows the dominance of the Kennedy Space Center as a source of successful launches at 41.7%
- The CCAFS LC-40 provides 29.2% of successful launches, but as we have seen it is simply a source of many early launches.

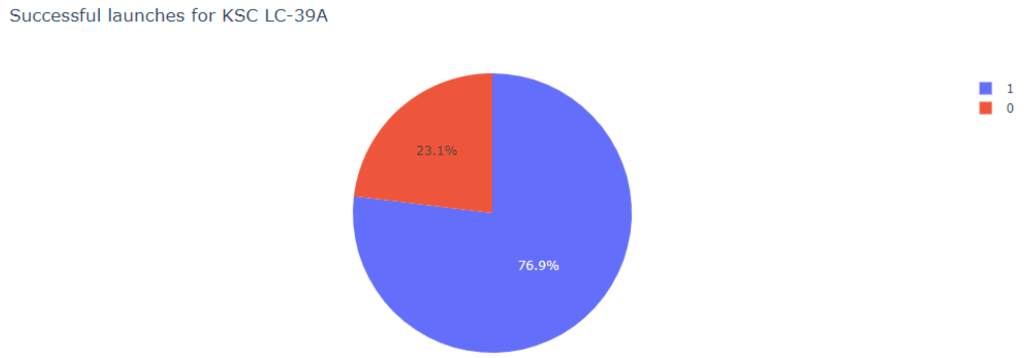


SpaceX Launch Records Dashboard

All Sites

Launch site with the highest success ratio

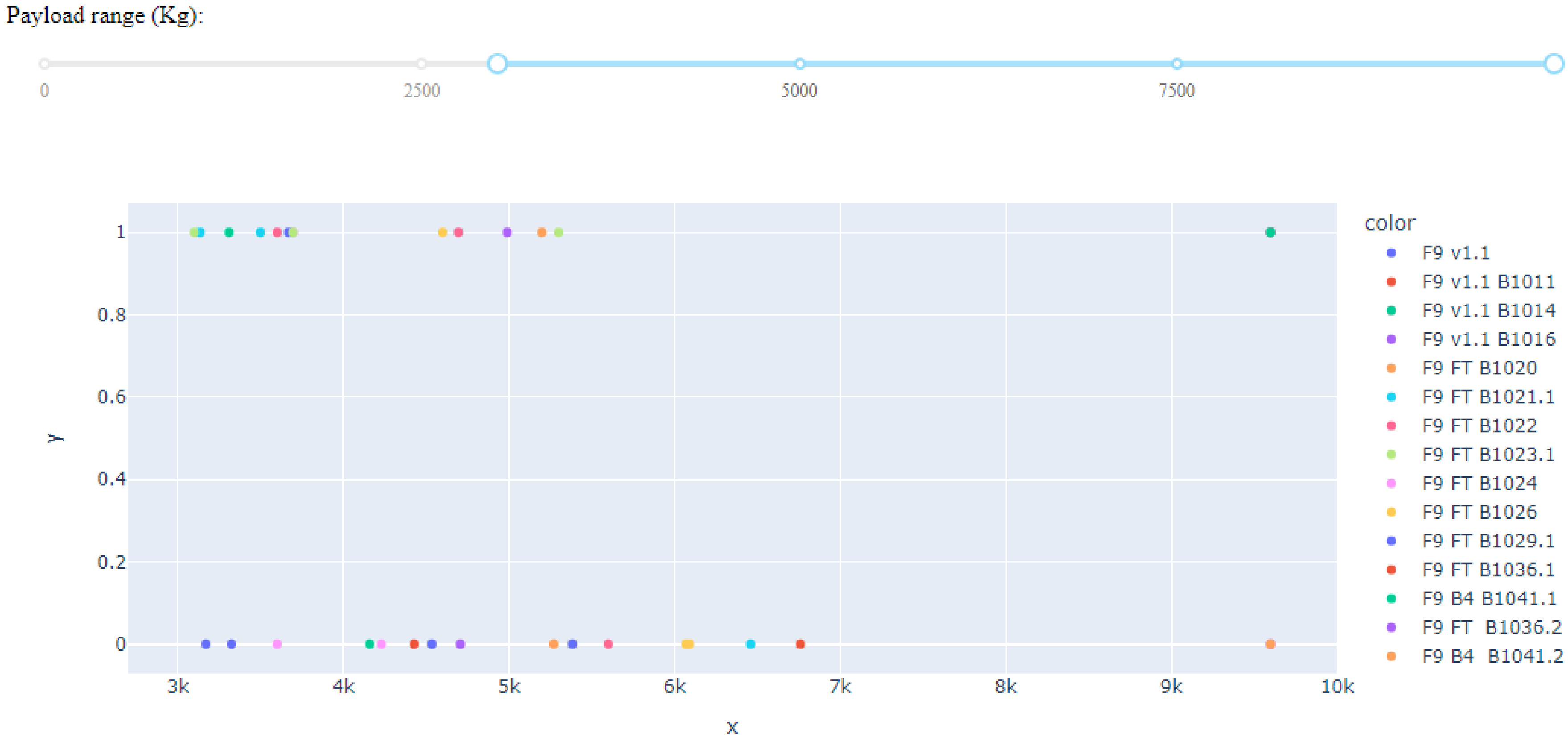
- Kennedy Space Center has by far the highest success rate at 76.9%
- CCAFS LC-40 has the lowest at 26.9%



Payload vs. Launch Outcome for all sites

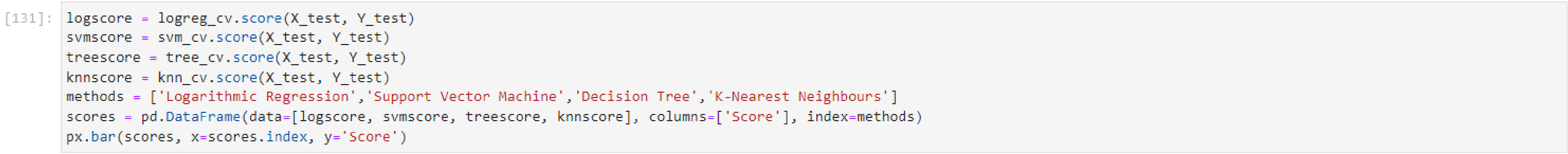
Outlier payload weights above 6000kg or below 2000kg tended to result in a lower rate of success.

The scatter plot shows how heavy payloads quickly result in a higher rate of failure.

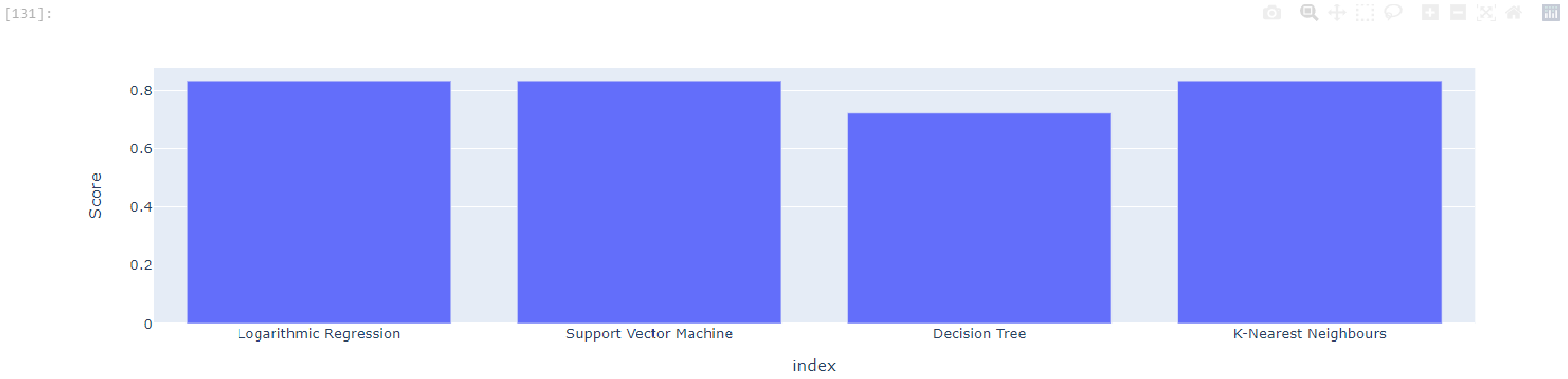


Section 5

Predictive Analysis (Classification)



```
logscore = logreg_cv.score(X_test, Y_test)
svmscore = svm_cv.score(X_test, Y_test)
treescore = tree_cv.score(X_test, Y_test)
knnscore = knn_cv.score(X_test, Y_test)
methods = ['Logarithmic Regression','Support Vector Machine','Decision Tree','K-Nearest Neighbours']
scores = pd.DataFrame(data=[logscore, svmscore, treescore, knnscore], columns=['Score'], index=methods)
px.bar(scores, x=scores.index, y='Score')
```

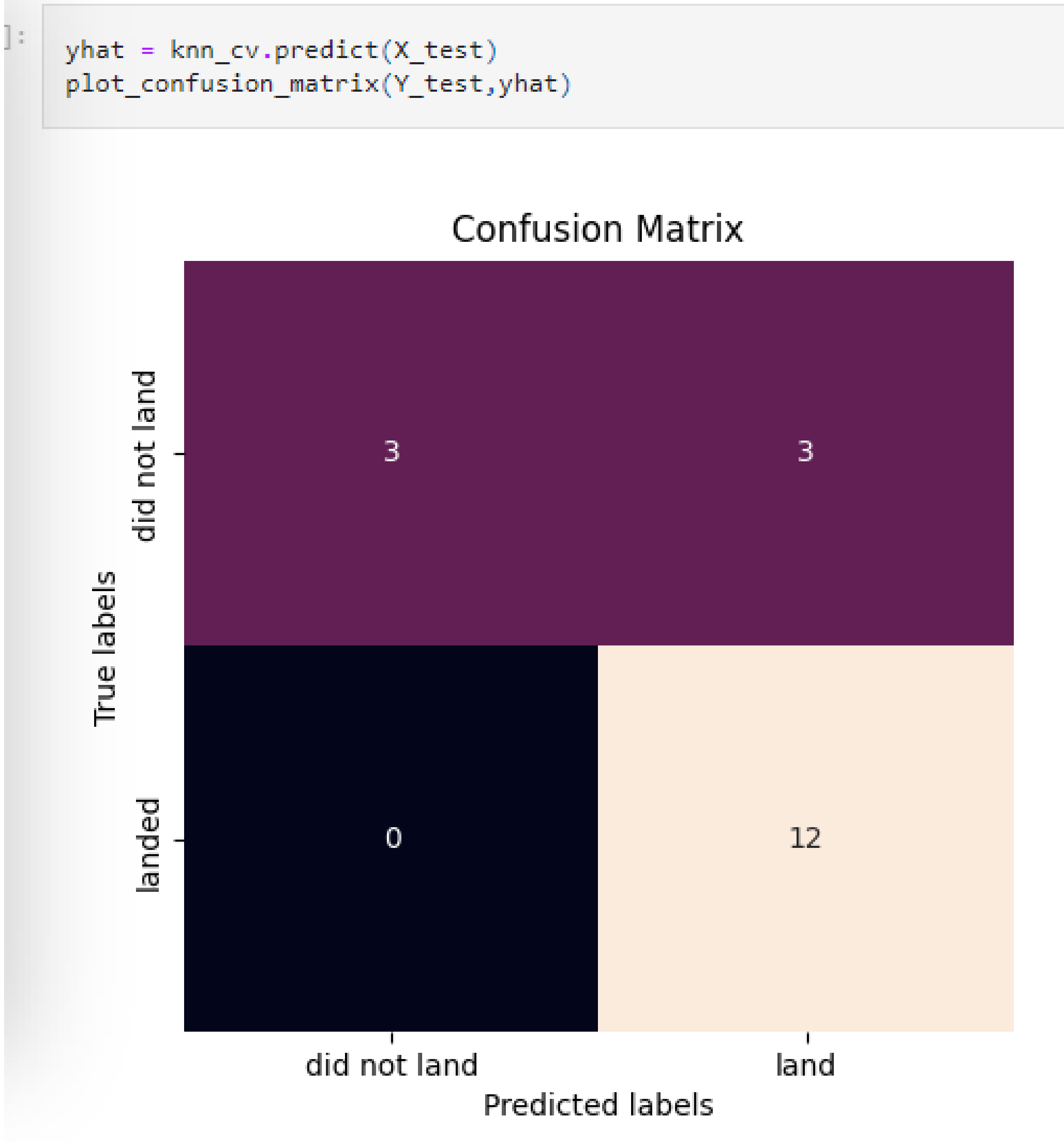


Classification Accuracy

- A three-way tie occurred with Logarithmic Regression, Support Vector Machine & K-Nearest Neighbours scoring 0.833

Confusion Matrix

- One of the best performing models was K-Nearest Neighbours
- A confusion matrix was constructed, showing 3 false positive predictions for the model



```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

Conclusions

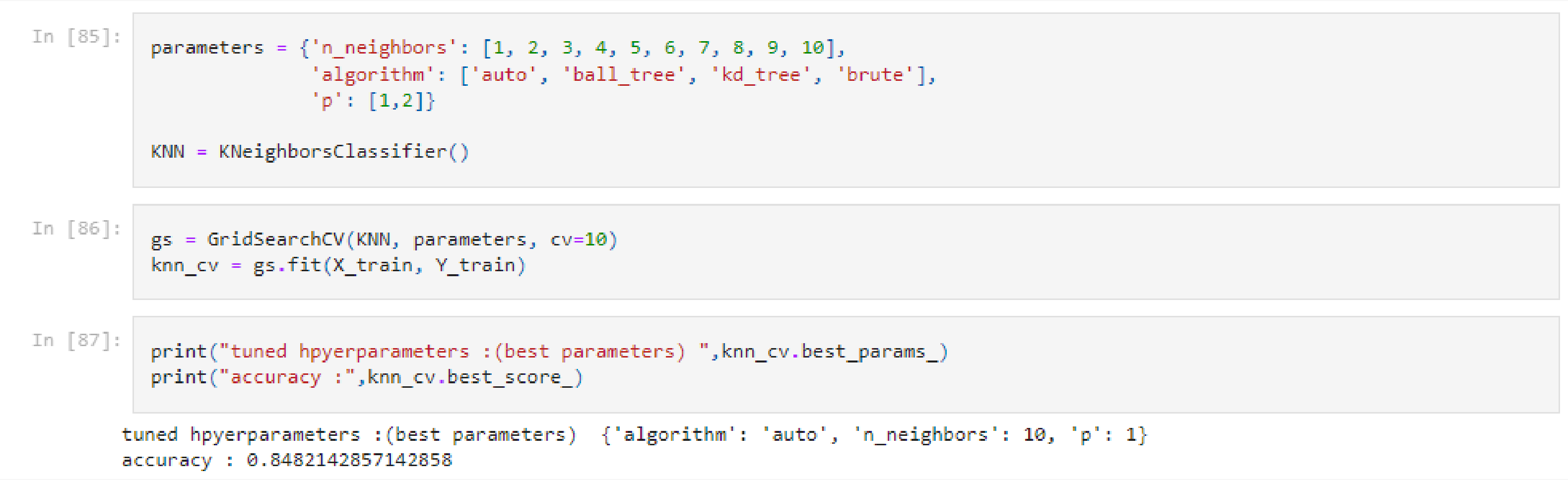
- Outlier payloads below 2000kg and above 6000kg greatly reduce the chance of successful recovery of the first stage.

- The success rate of SpaceX launches remained 0% from 2010 to 2013.

- After 2013, the success rate of launches gradually rose, rising above 80% in 2020.

- Launch sites are placed near railways and coastlines, and far from cities.

- The Kennedy Space Center Launch Complex 39 has seen the highest rate of successful launches by a large margin at 76.9% - this is partly explained by the bias towards mid-weight payloads.



```
In [85]:  parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
                        'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                        'p': [1,2]}

          KNN = KNeighborsClassifier()

In [86]:  gs = GridSearchCV(KNN, parameters, cv=10)
          knn_cv = gs.fit(X_train, Y_train)

In [87]:  print("tuned hpyerparameters :(best parameters) ",knn_cv.best_params_)
          print("accuracy :",knn_cv.best_score_)

        tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
        accuracy : 0.8482142857142858
```

Appendix

- Data collected from the SpaceX API:
    - https://api.spacexdata.com/v4/
- Additional data from Wikipedia:
    - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Thank you!