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

- Summary of all results

  - Exploratory Data Analysis results

  - Predictive analysis results

Introduction

- Project background and context

We know that the first stage is larger and more expensive than the second stage, and it plays a crucial role in getting the rocket off the ground. SpaceX has achieved success in landing and reusing the first stage in some cases, but there are instances where it cannot be recovered. Aim is to determine the cost of a rocket launch, focusing on SpaceX and their ability to reuse the first stage of their rockets

- Problems you want to find answers

How do variables such as payload mass, launch site, number of flights, and orbits affect the success of the first stage landing?

Does the rate of successful landings increase over the years?

Section 1

Methodology

Methodology

Executive Summary

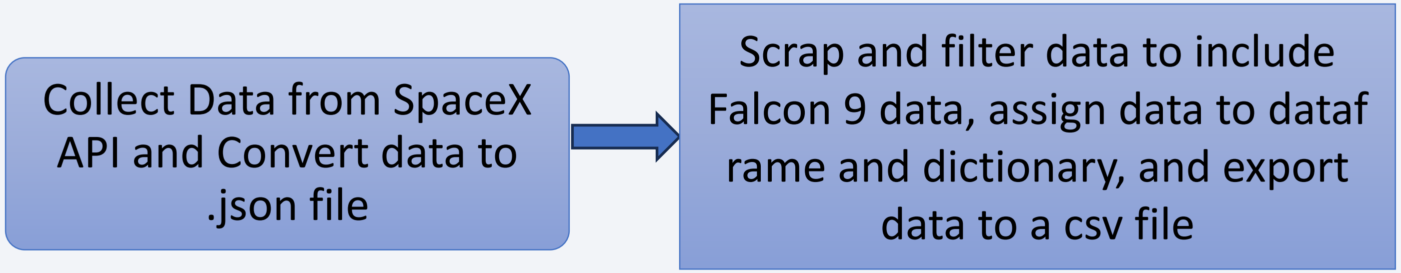
- Data collection methodology:
  - Using SpaceX Rest API
  - Using Web Scrapping from Wikipedia
- Perform data wrangling
  - Filtering the data
  - Dealing with missing values
  - Using One Hot Encoding to prepare the data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

Data collection process involved a combination of API requests from SpaceX REST API and Web Scraping data from a table in SpaceX's Wikipedia entry. We had to use both of these data collection methods in order to get complete information about the launches for a more detailed analysis.

Data Columns are obtained by using SpaceX REST API: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude Data Columns are obtained by using Wikipedia Web Scraping: Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection – SpaceX API

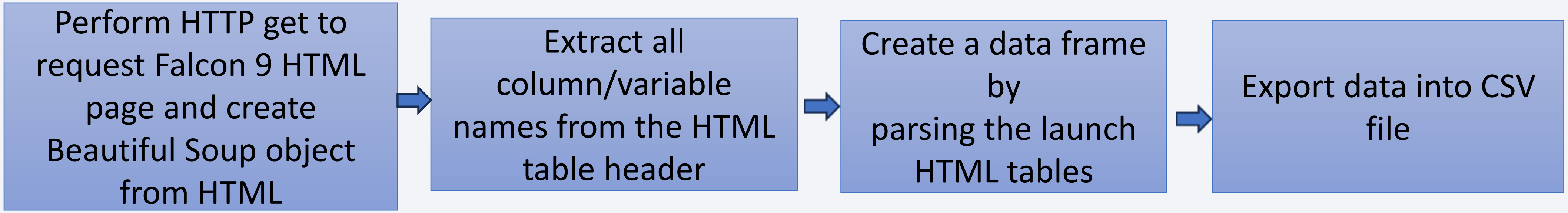


Collect Data from SpaceX API and Convert data to .json file

Scrap and filter data to include Falcon 9 data, assign data to dataframe and dictionary, and export data to a csv file

GitHub URL of the completed SpaceX API calls notebook :
https://github.com/sheebasyed/Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping



Perform HTTP get to request Falcon 9 HTML page and create Beautiful Soup object from HTML

Extract all column/variable names from the HTML table header

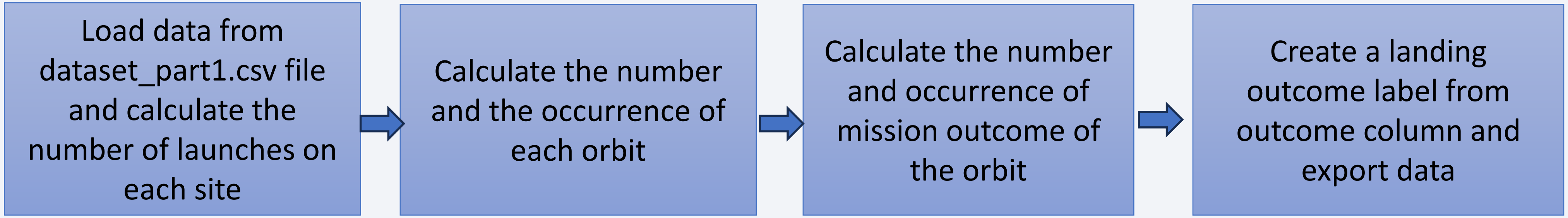
Create a data frame by parsing the launch HTML tables

Export data into CSV file

GitHub url: https://github.com/sheebasyed/Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- Performed exploratory data analysis and determined the training labels. Calculated the number of launches at each site, and the number and occurrence of each orbit and created landing outcome label from outcome column and exported the results to csv.



Load data from dataset_part1.csv file and calculate the number of launches on each site

Calculate the number and the occurrence of each orbit

Calculate the number and occurrence of mission outcome of the orbit

Create a landing outcome label from outcome column and export data

GitHub url: https://github.com/sheebasyed/Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-webscraping.ipynb

EDA with Data Visualization

Launch Site trends :

- Scatterplot to see mission outcome relationship split by Launch Site and Flight Number.
- Scatterplot to see mission outcome relationship split by Launch Site and Payload.

Orbit Type trends :

- Bar chart to see mission outcome relationship with Orbit Type.
- Scatterplot to see mission outcome relationship split by Orbit Type and Flight Number.
- Scatterplot to see mission outcome relationship split by Orbit Type and Payload.

Trends based on time :

- Line plot to see mission outcome trend by year

GitHub URL : https://github.com/sheebasyed/Applied-Data-Science-Capstone-Project/blob/main/edadataviz.ipynb

EDA with SQL

Queries were written to extract information about:

- Launch sites
- Payload masses
- Dates
- Booster types
- Mission outcomes

GitHub URL : https://github.com/sheebasyed/Applied-Data-Science-Capstone-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1, 0 for failure and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
    - Are launch sites near railways, highways and coastlines.
    - Do launch sites keep certain distance away from cities.

Build a Dashboard with Plotly Dash

- The input dropdown is used to select one or all launch sites for the pie chart and scatterplot.
- The pie chart displays :

For All Sites – the distribution of successful Falcon 9 first stage landings between the sites

For One Site – the distribution of successful and failed Falcon 9 first stage landings for that site

- The input slider is used to filter the payload masses for the scatterplot.
- The scatterplot displays the distribution of Falcon 9 first stage landings split by payload mass, mission outcome and by booster version category.

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.

GitHub URL: https://github.com/sheebasyed/Applied-Data-Science-Capstone-Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

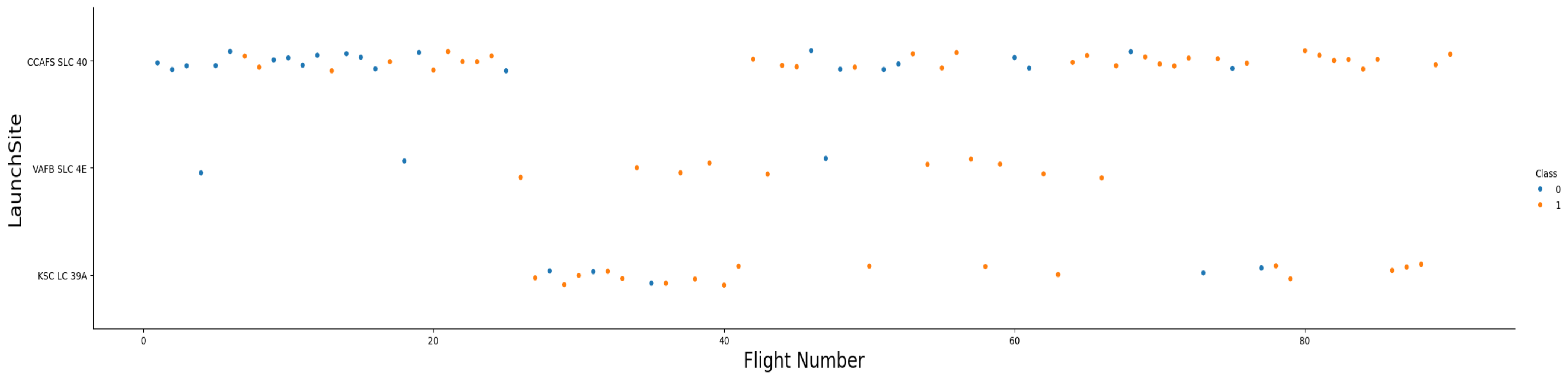
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

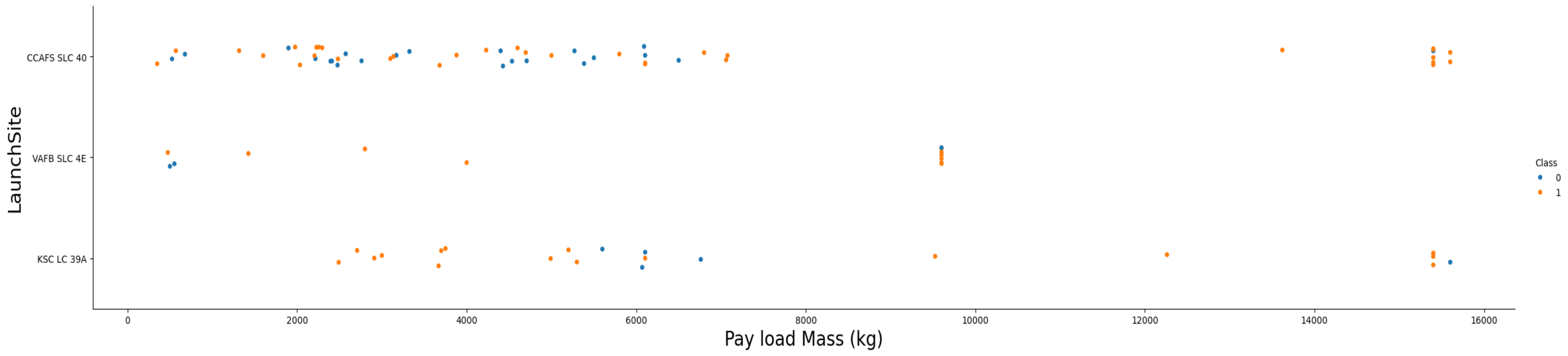
Flight Number vs. Launch Site

We can see that the larger the flight number at a launch site, the greater the success rate at a launch site.



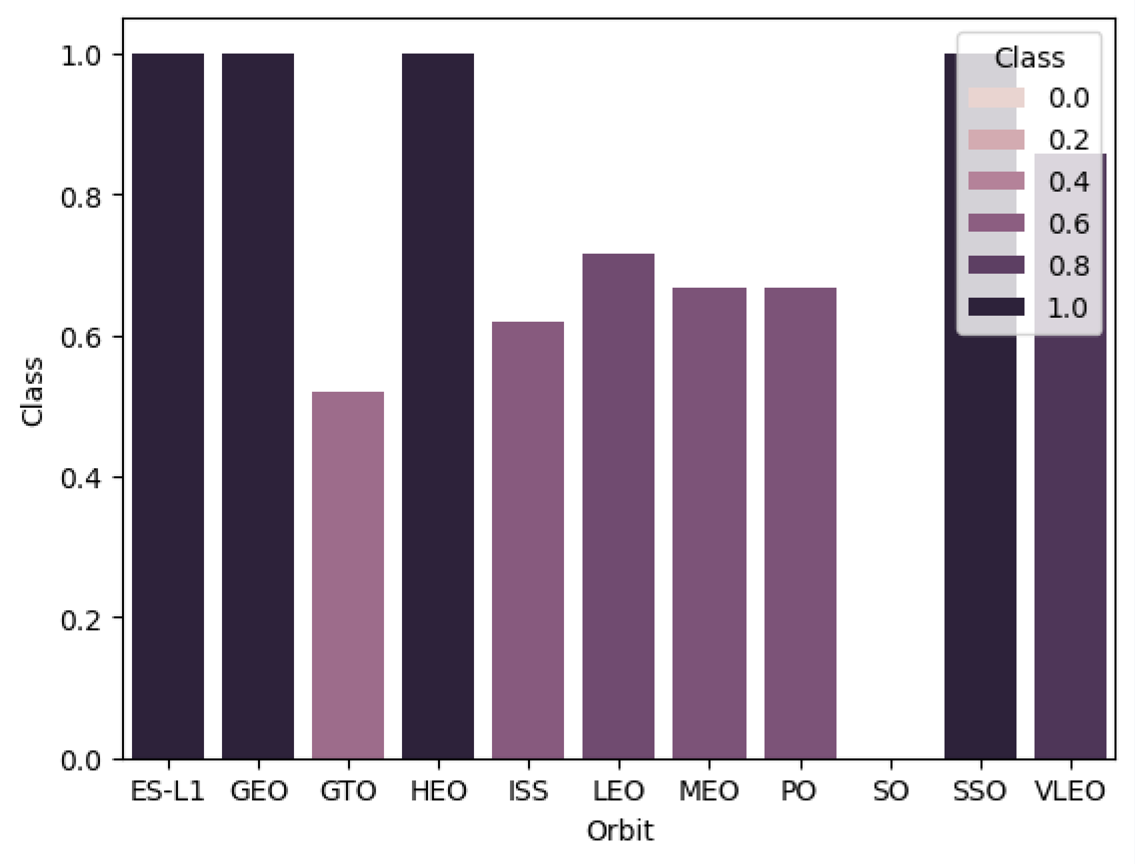
Payload vs. Launch Site

- For the CCAFS SLC 40 launch site, the payload mass and the landing outcome appear to not be strongly correlated



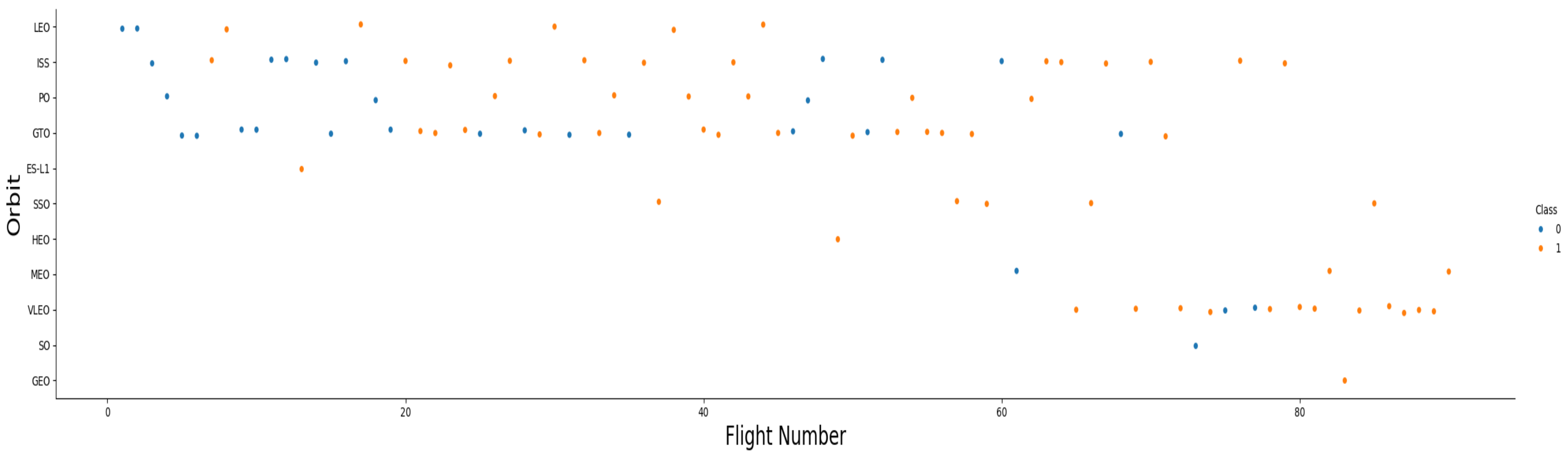
Success Rate vs. Orbit Type

- ES-L1, SSO, HEO and GEO orbits have no failed first stage landings.



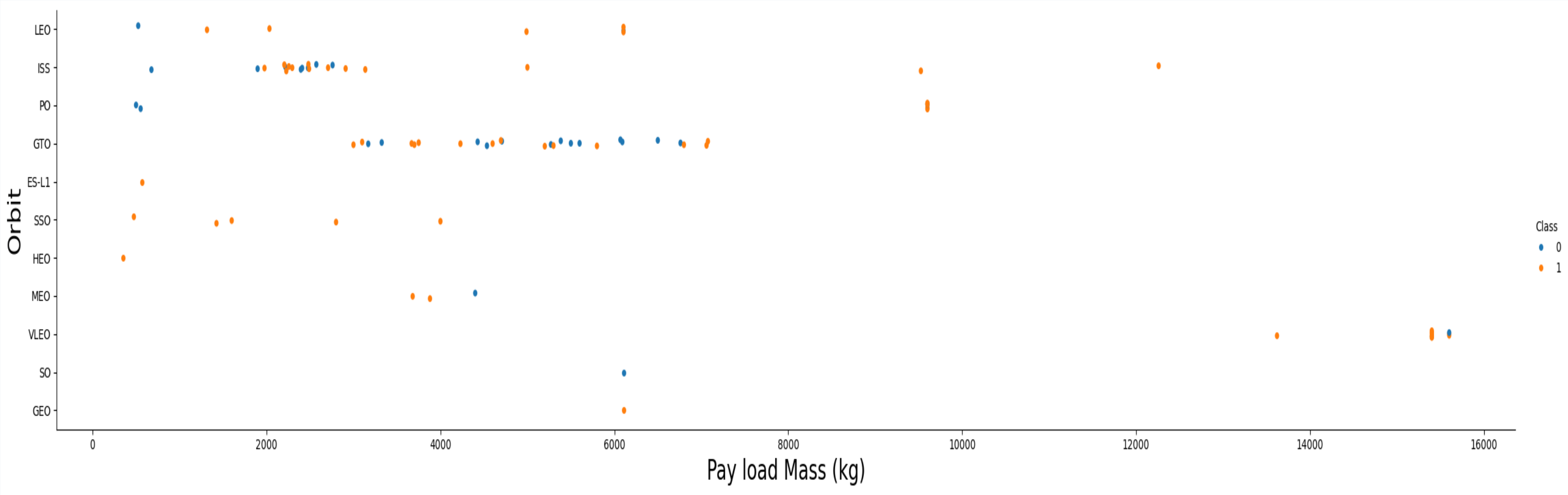
Flight Number vs. Orbit Type

- There is a correlation between flight number and success rate with larger flight numbers being associated with higher success rates.



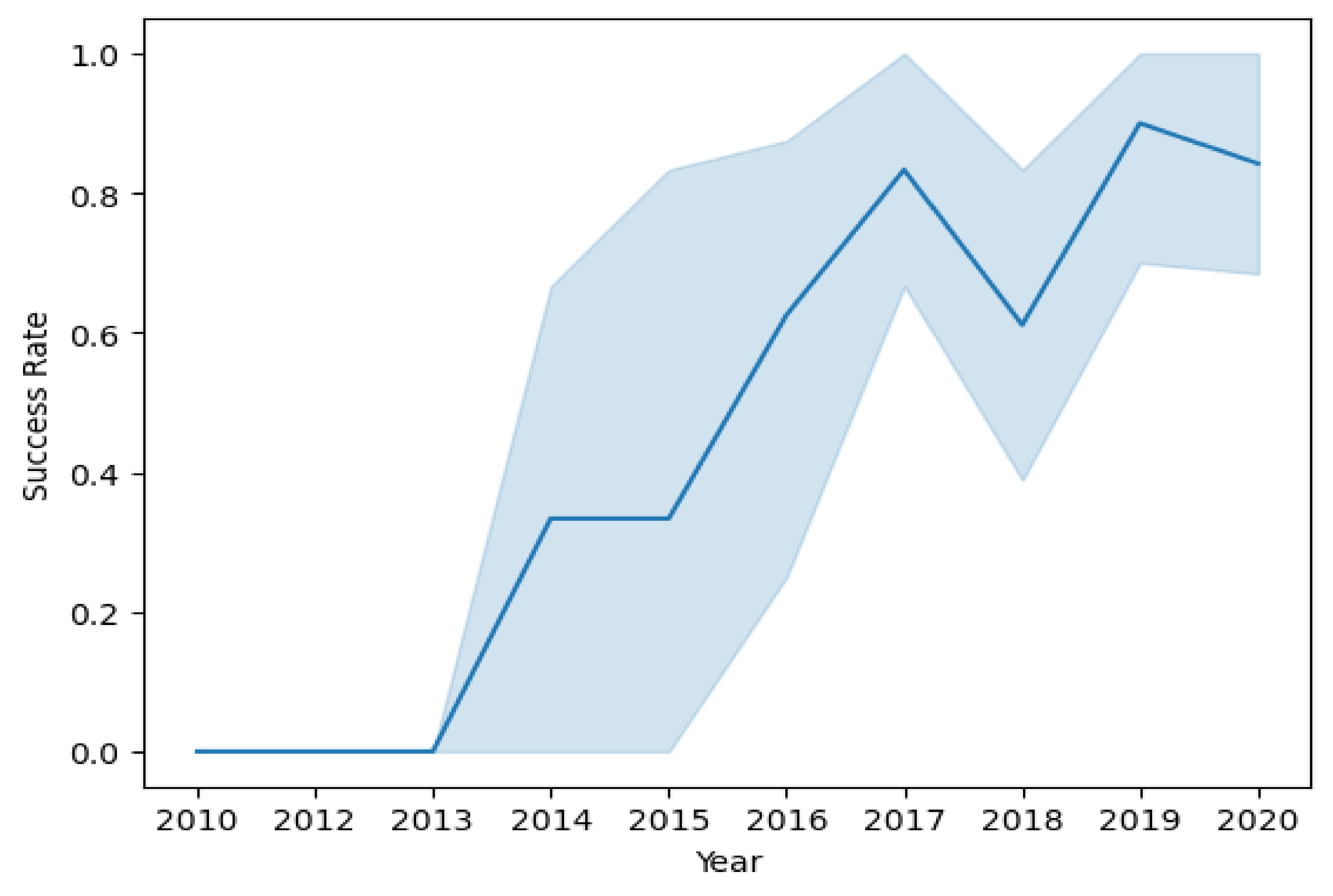
Payload vs. Orbit Type

- Success rate appears to have no obvious correlation with payload mass.



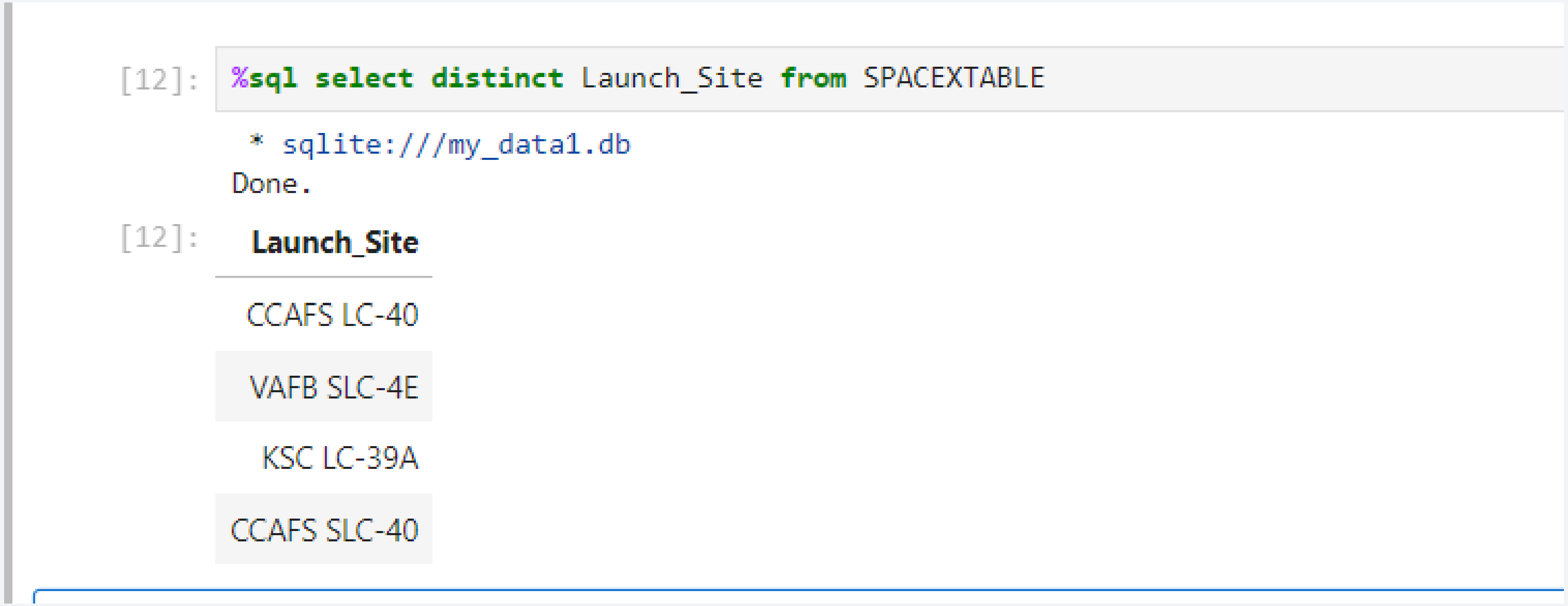
Launch Success Yearly Trend

- The success rate has increased significantly over the years.



All Launch Site Names

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



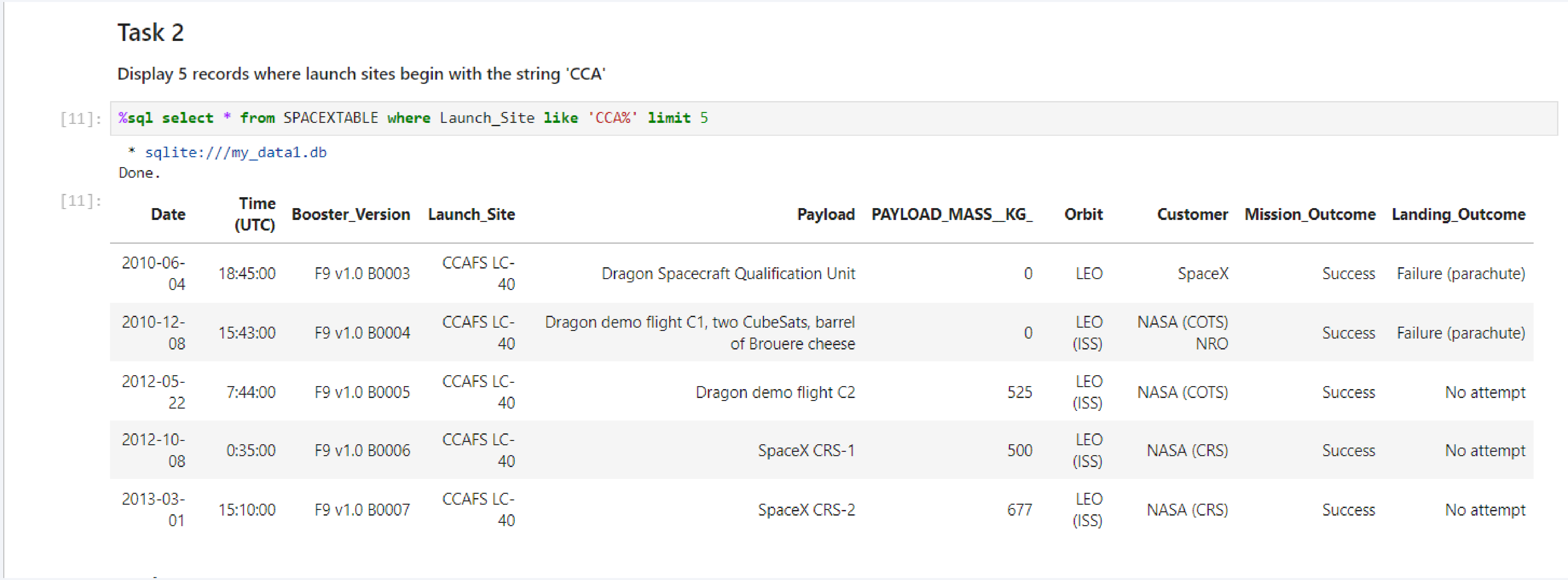
```
[12]: %sql select distinct Launch_Site from SPACEXTABLE

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

[12]:

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

Launch Site Names Begin with 'CCA'

- We used the query below to display 5 records where launch sites begin with `CCA`



Task 2

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

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

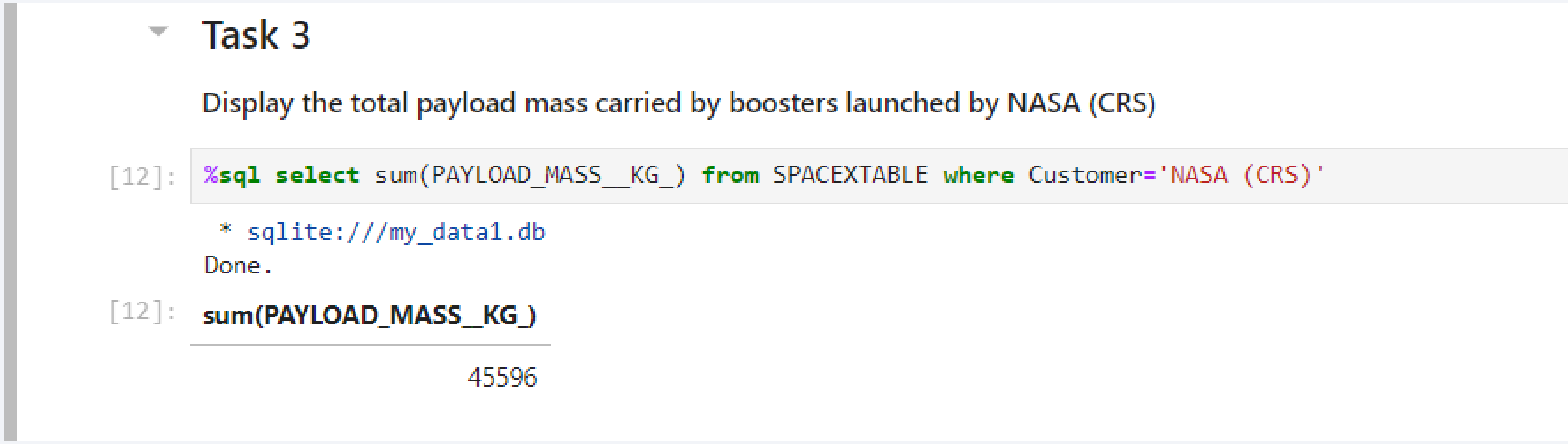
```
 * sqlite:///my_data1.db
Done.
```

[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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



Task 3

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

```
[12]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer='NASA (CRS)'
```

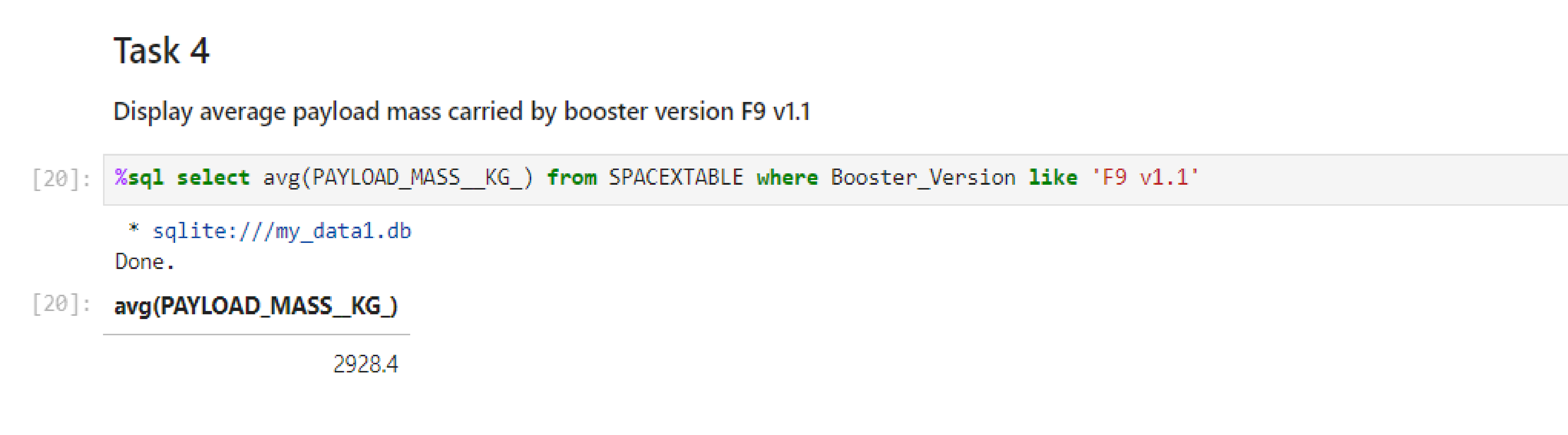
```
 * sqlite:///my_data1.db
Done.
```

[12]:

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



Task 4

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

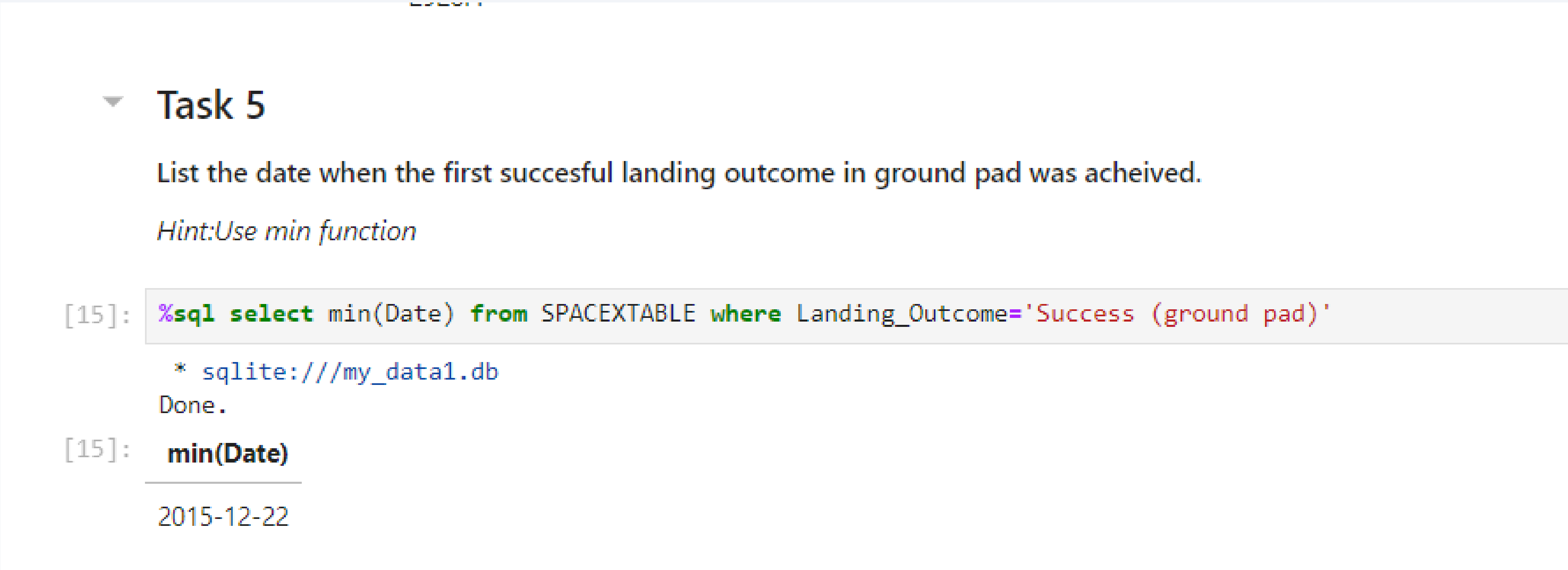
```
[20]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version like 'F9 v1.1'
```

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

[20]: 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



Task 5

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

Hint:Use min function

```
[15]: %sql select min(Date) from SPACEXTABLE where Landing_Outcome='Success (ground pad)'
```

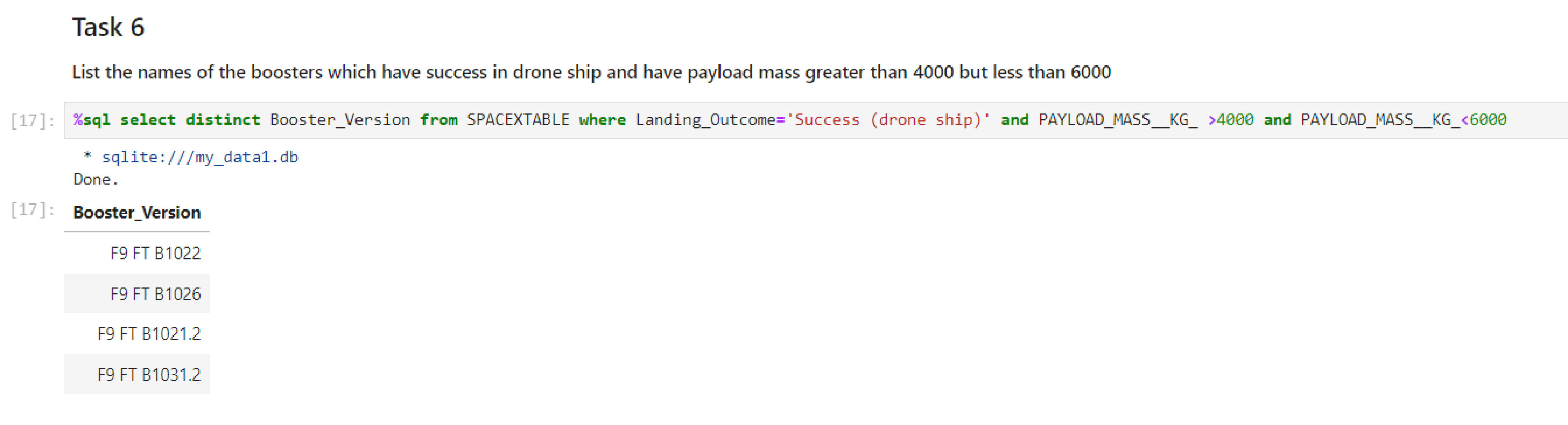
```
 * sqlite:///my_data1.db
Done.
```

[15]:

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000



Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
[17]: %sql select distinct Booster_Version from SPACEXTABLE where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ >4000 and PAYLOAD_MASS__KG_<6000
```

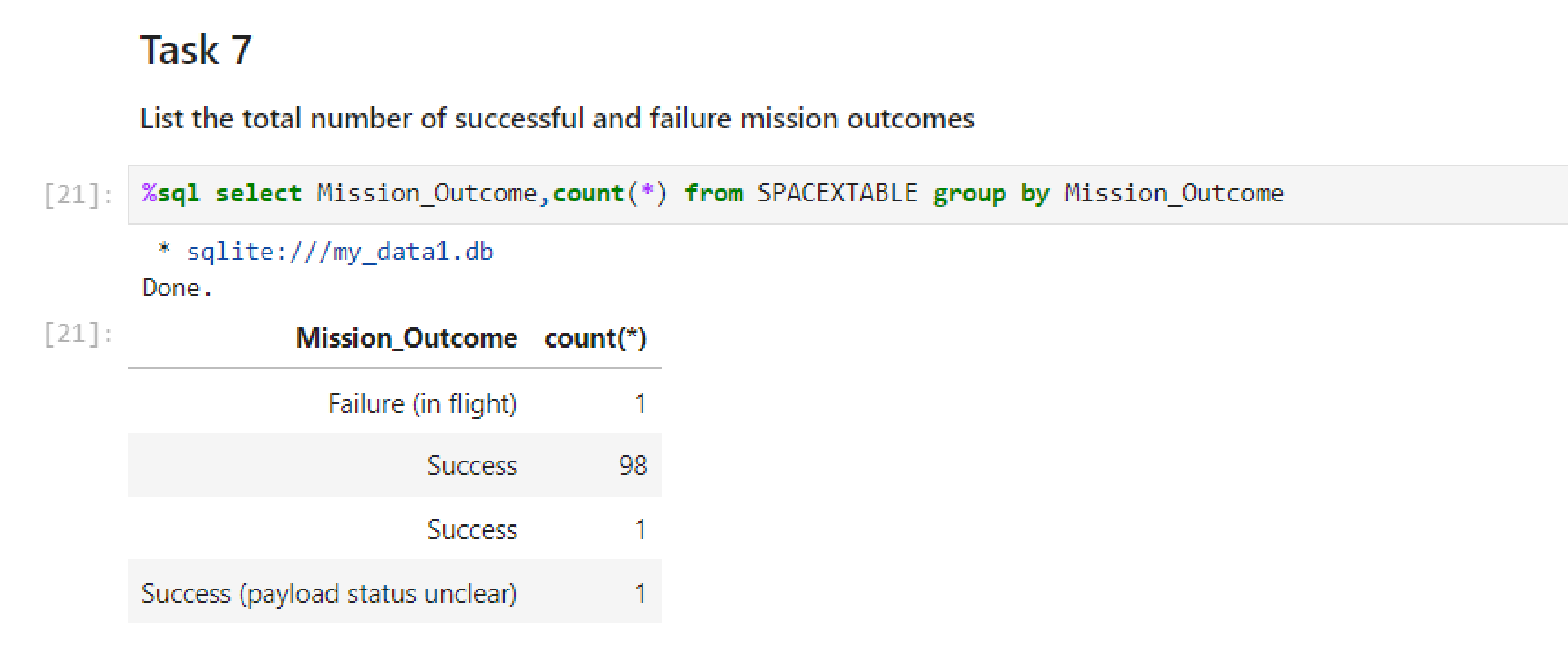
```
 * sqlite:///my_data1.db
Done.
```

[17]:

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

Total Number of Successful and Failure Mission Outcomes

- We used group by to get MissionOutcomes.



Task 7

List the total number of successful and failure mission outcomes

```
[21]: %sql select Mission_Outcome,count(*) from SPACEXTABLE group by Mission_Outcome
```

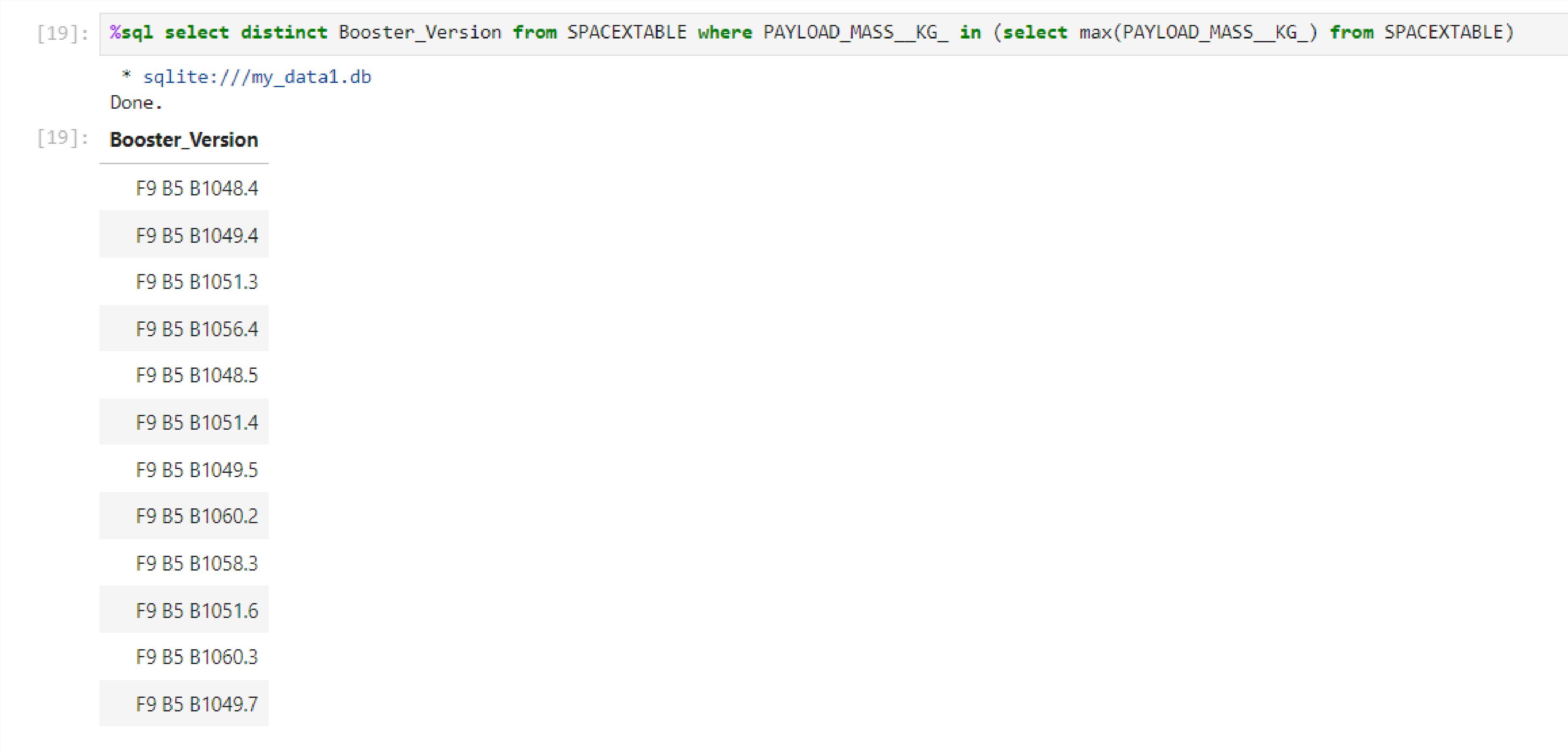
```
 * sqlite:///my_data1.db
Done.
```

[21]:

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



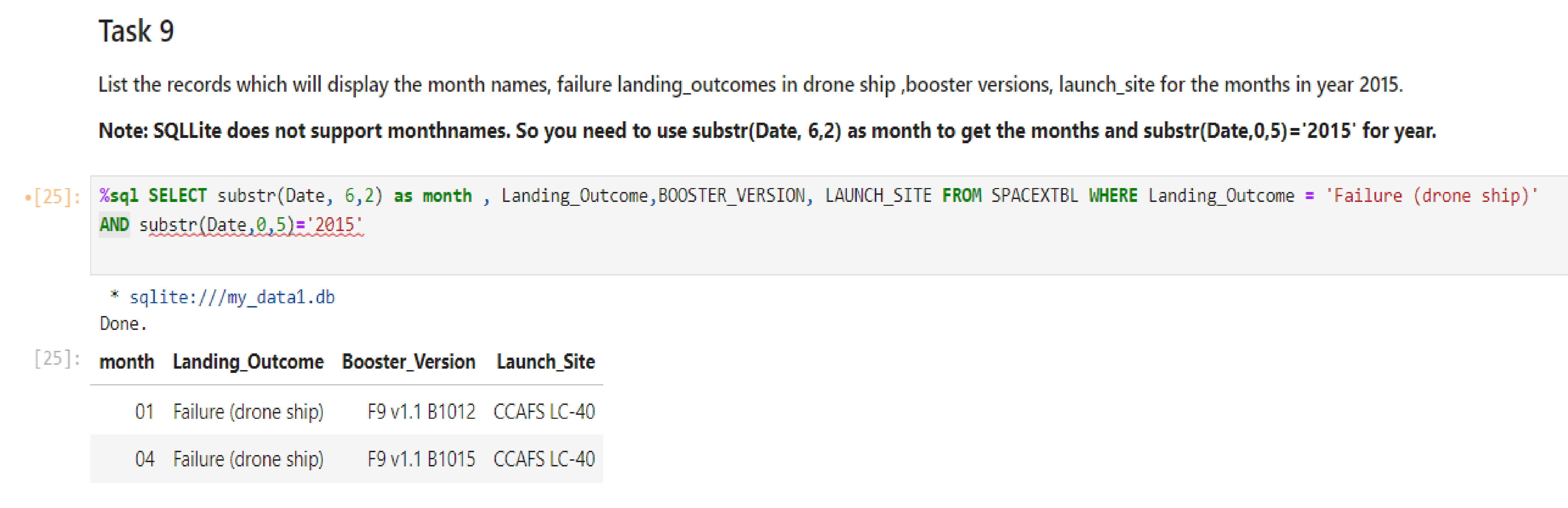
```
[19]: %sql select distinct Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)

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

[19]:

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 **WHERE** clause condition to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015



Task 9

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.

```
[25]: %sql SELECT substr(Date, 6,2) as month , Landing_Outcome,BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE Landing_Outcome = 'Failure (drone ship)'
      AND substr(Date,0,5)='2015'
```

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

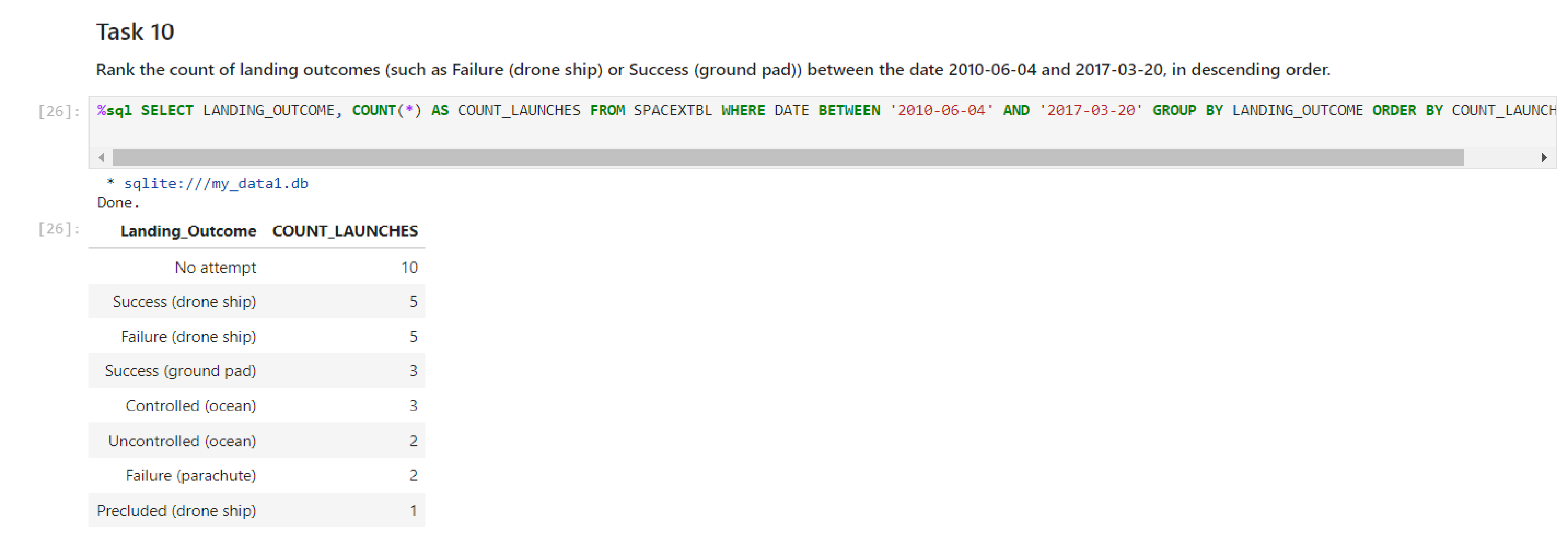
[25]:

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

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2017-03-20.

- 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.



Task 10

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.

```
[26]: %sql SELECT LANDING_OUTCOME, COUNT(*) AS COUNT_LAUNCHES FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING_OUTCOME ORDER BY COUNT_LAUNCH
```

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

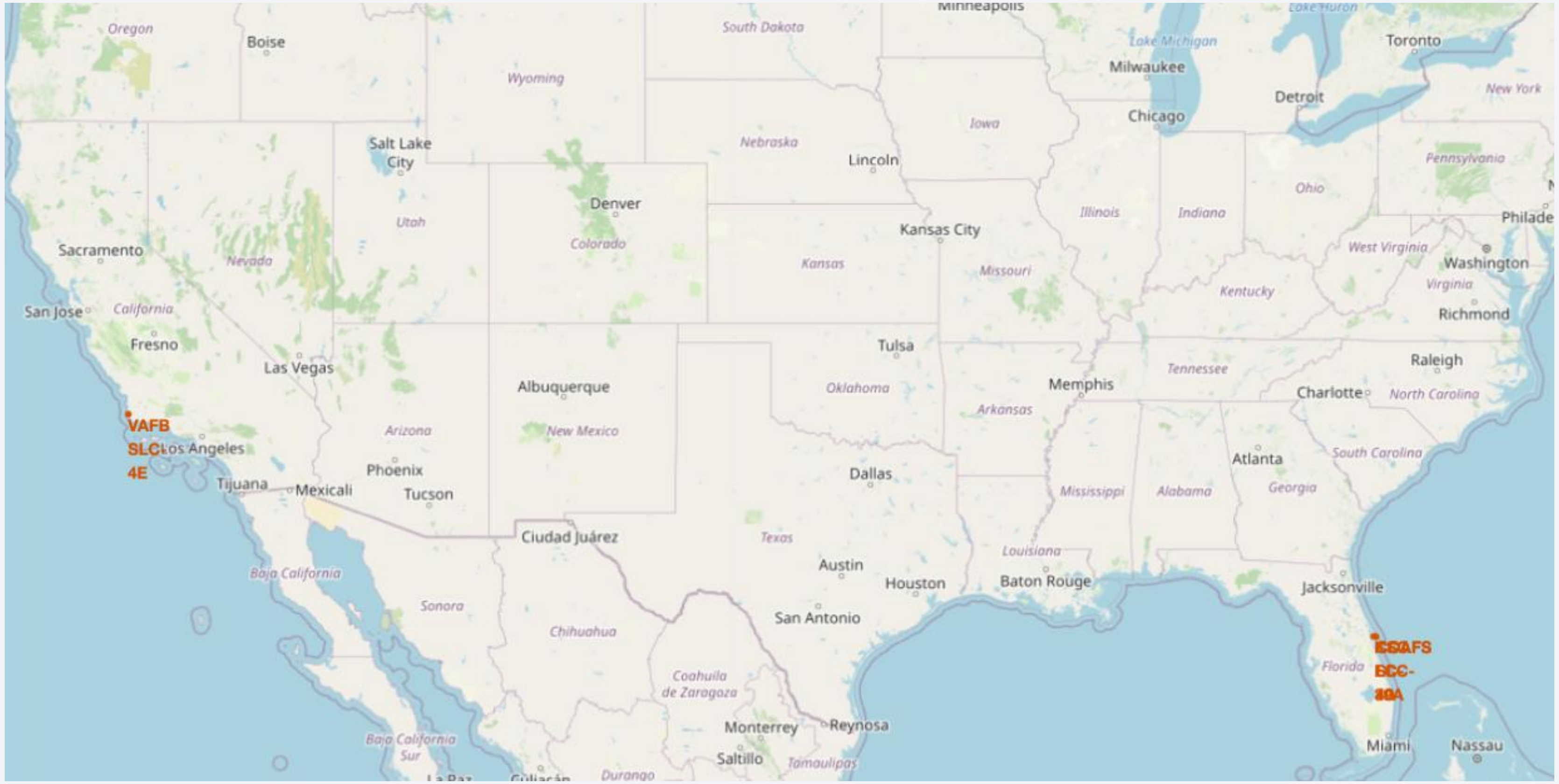
[26]:

Landing_Outcome	COUNT_LAUNCHES
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

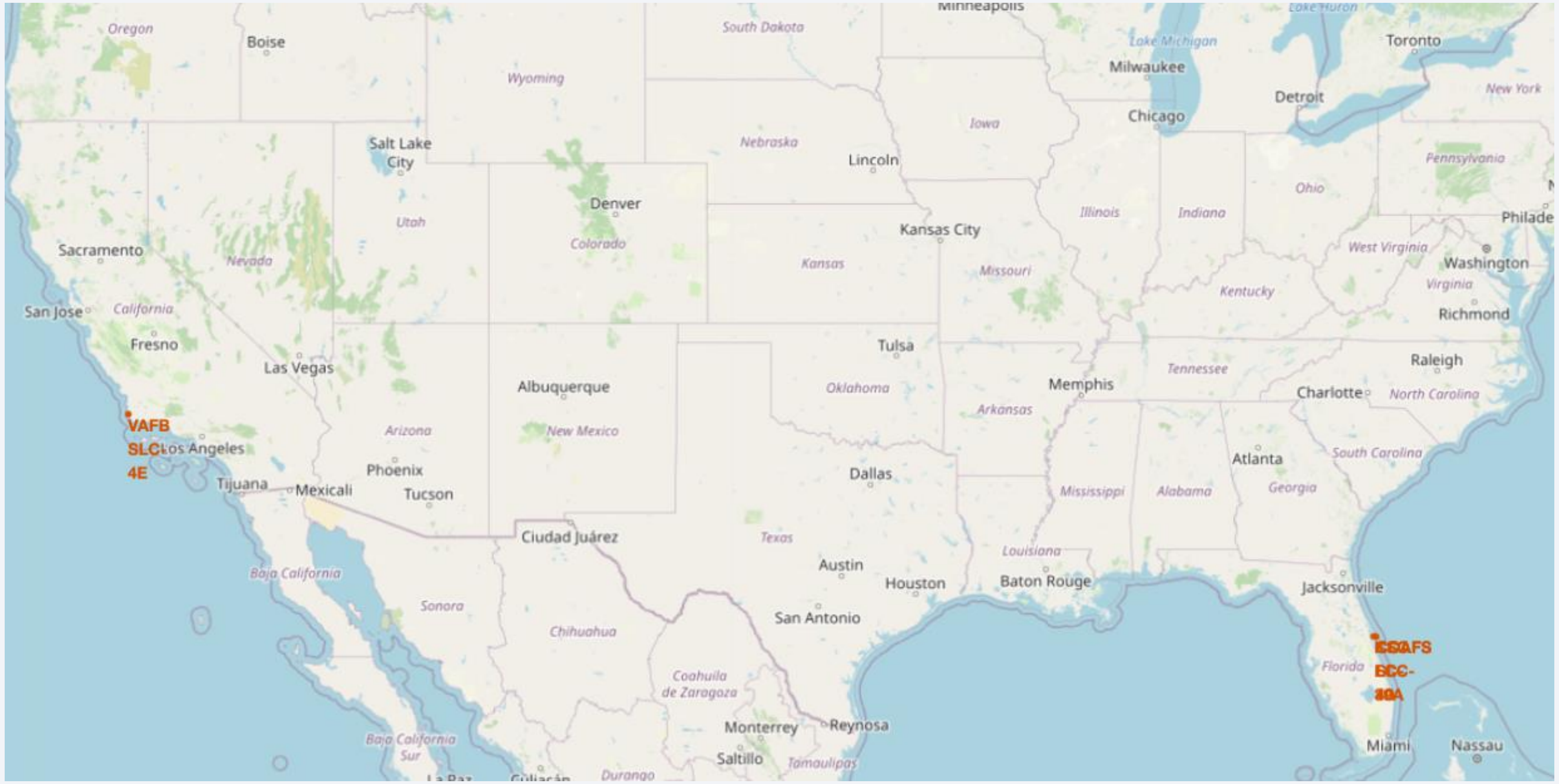
Section 3

Launch Sites Proximities Analysis

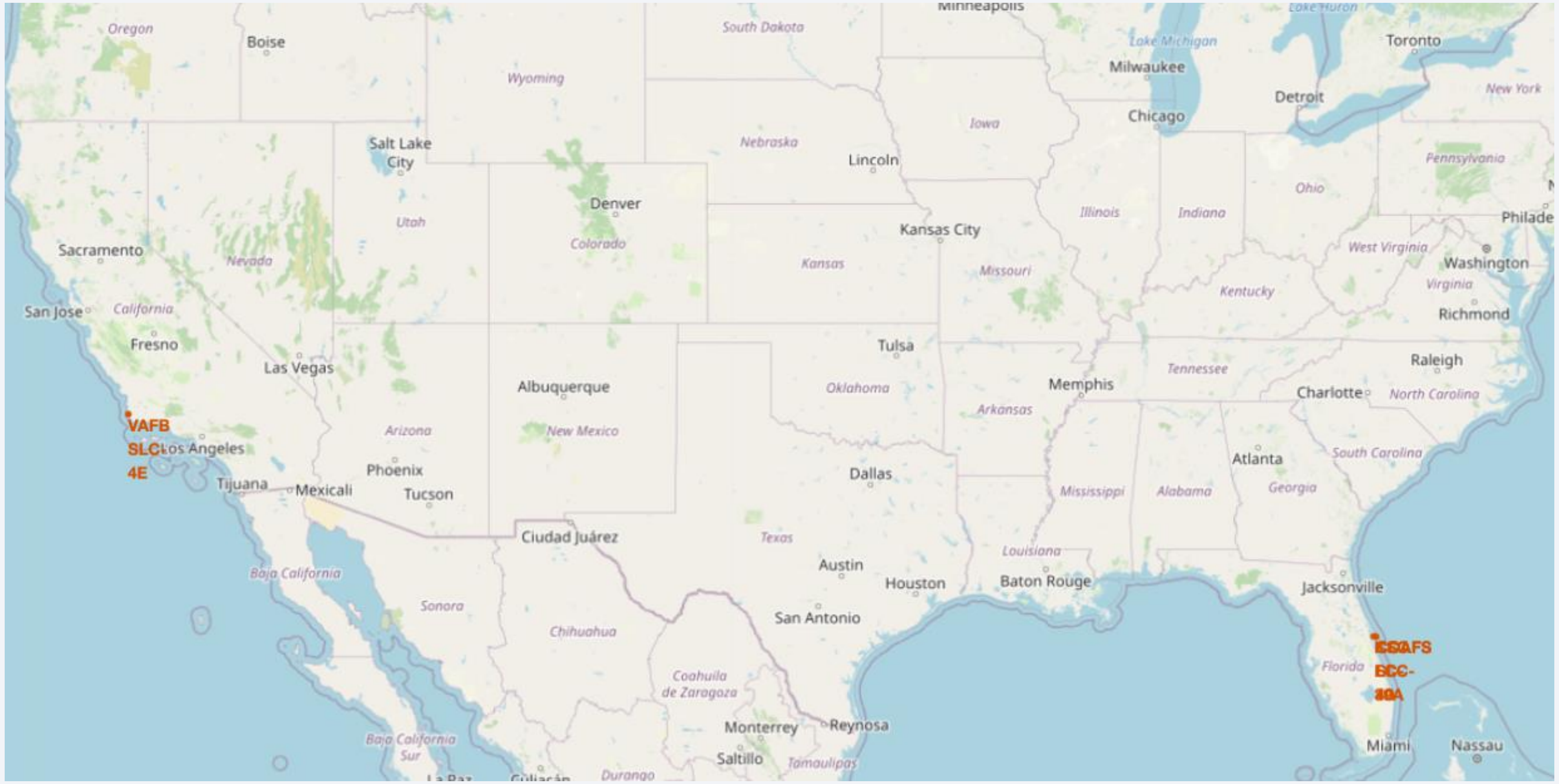
All launch sites global map markers



All launch sites global map markers



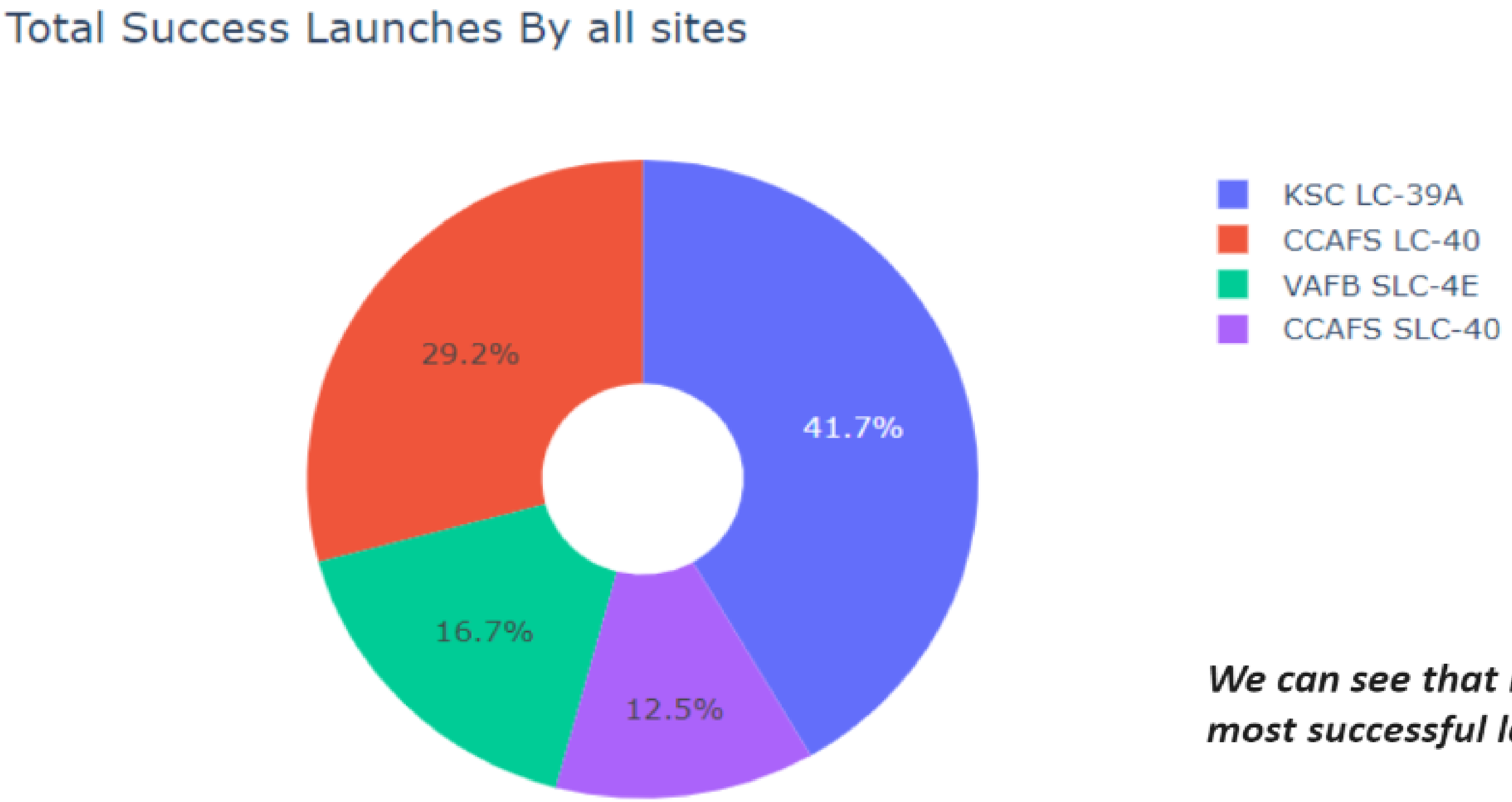
All launch sites global map markers



Section 4

Build a Dashboard with Plotly Dash

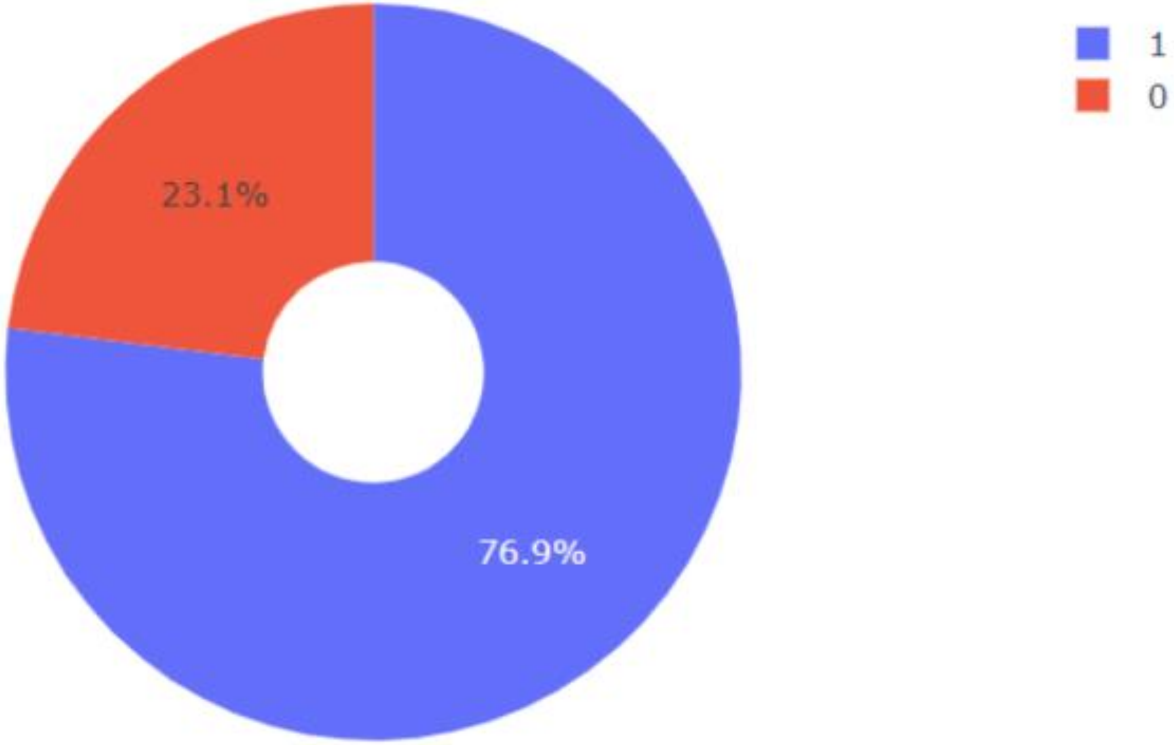
Pie chart showing the success percentage achieved by each launch site



Total Success Launches By all sites

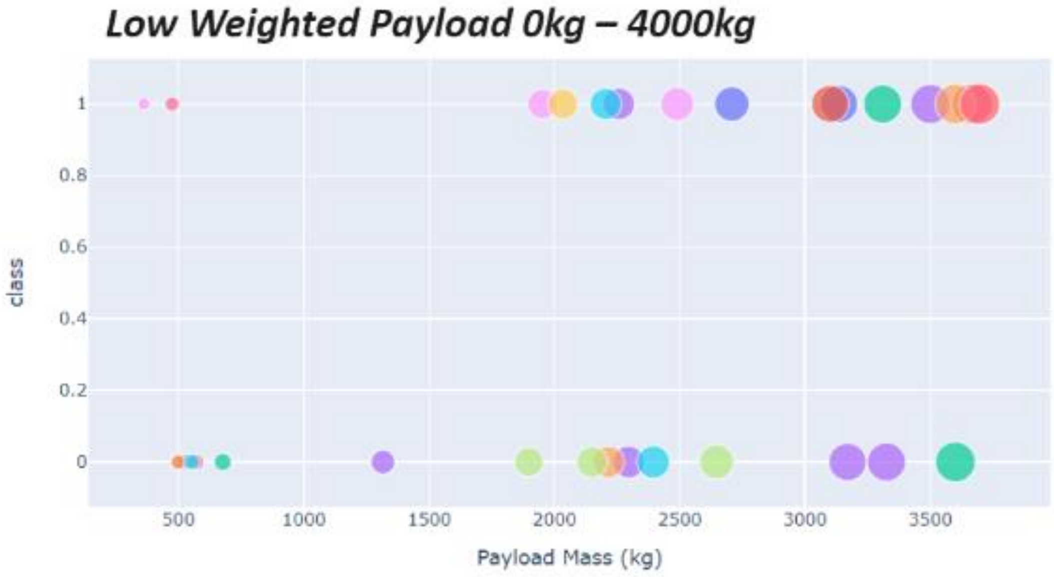
We can see that KSC LC-39A had the most successful launches from all the sites

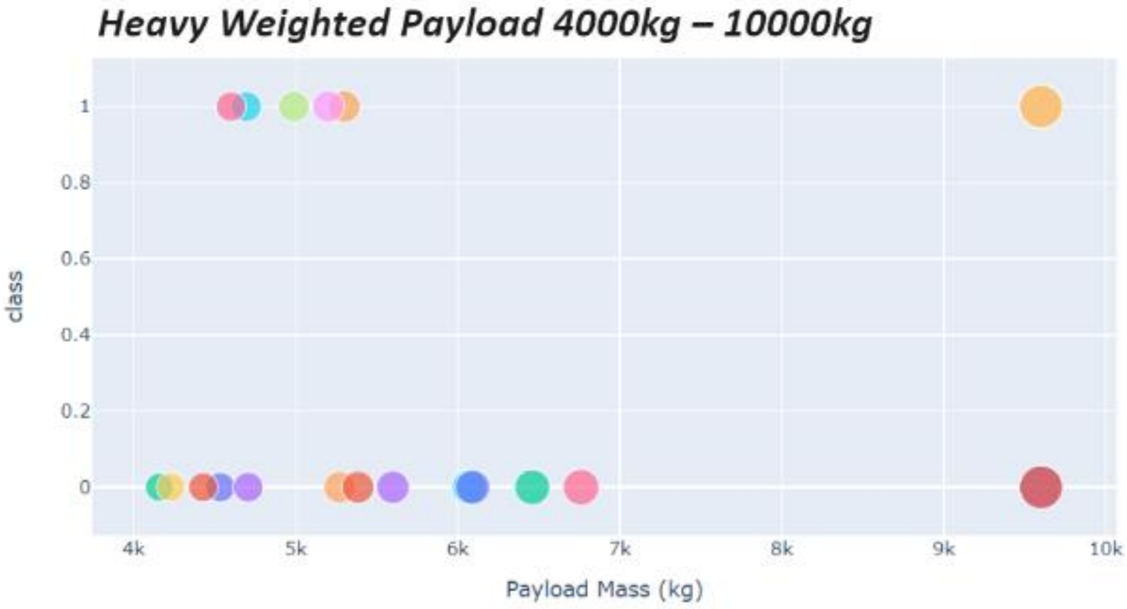
Pie chart showing the Launch site with the highest launch success ratio



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





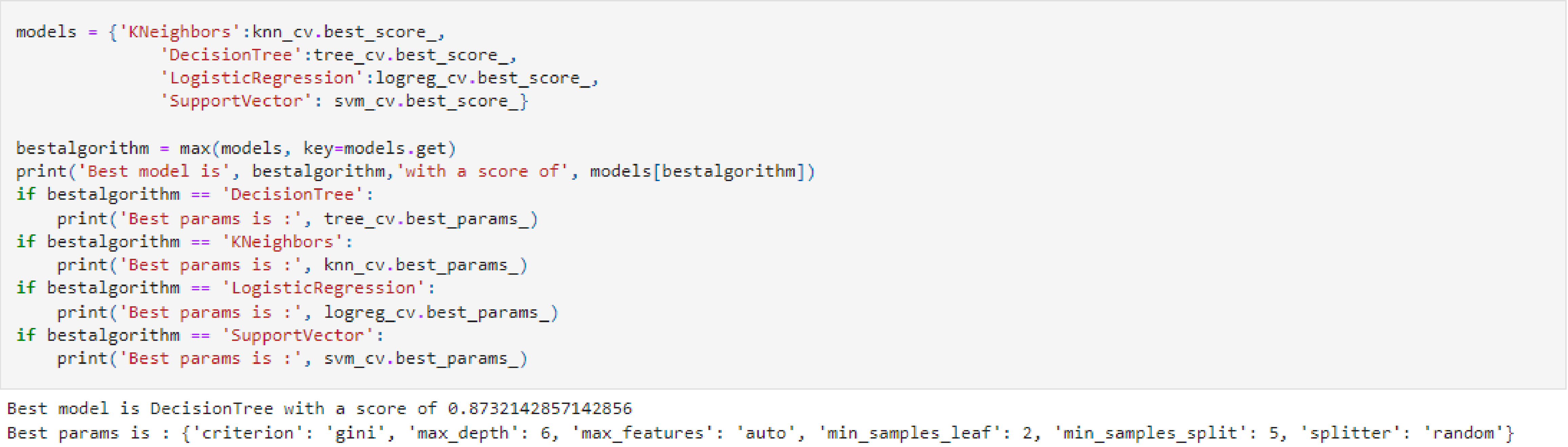
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy

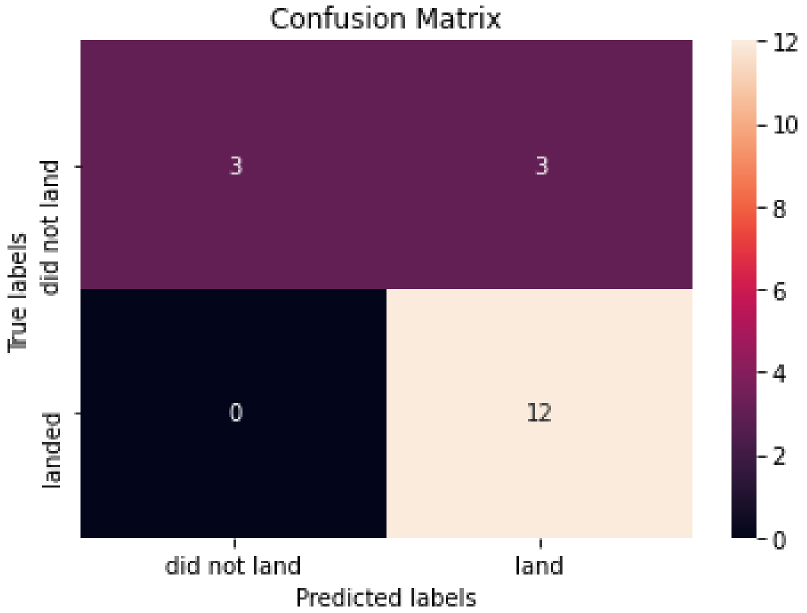


```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

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., unsuccessful landing marked as successful landing by the classifier.



Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.