IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

José Vivero
September 8, 2023

Outline

Executive Summary

- **Summary of methodologies**
  - Data Collection: API and web scrapping.
  - Exploratory Data Analysis : SQL, Pandas and Matplotlib.
  - Interactive Visualizations and Dashboard: Folium, Plotly and Dash.
  - Machine Learning: Scikit-learn.

- **Summary of all results**
  - We apply several classification Machine Learning algorithms to predict if the first stage of a Falcon 9 rocket will land successfully.

Introduction

- **Project background and context**

  The company Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars, which can be up to 60% more cost effective than other providers. Much of the savings is because Space X can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. In this project we will use historical data to create a machine learning pipeline to predict if the first stage will land.

- **Main Objective**
  - Predict future landing outcomes for Falcon 9 first stage using Machine Learning.

Section 1

Methodology

Methodology

Executive Summary

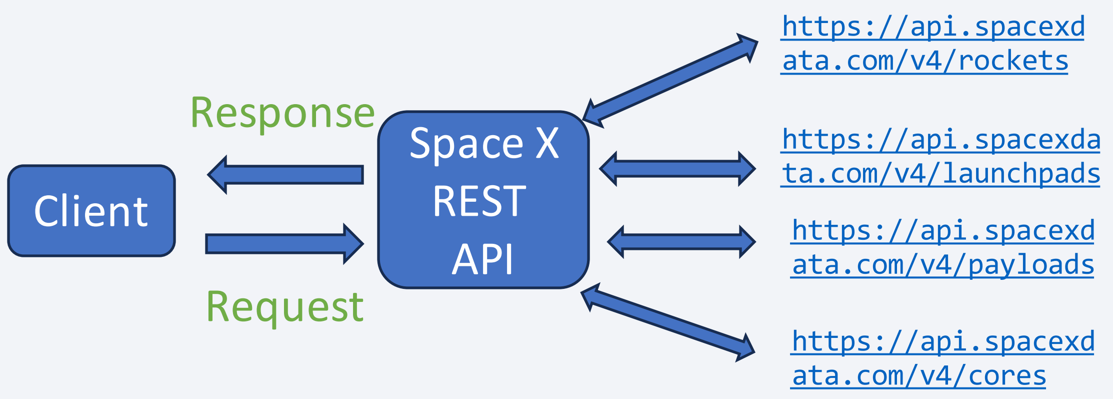
- **Data collection methodology:**
  - Describe how data was collected
- **Perform data wrangling**
  - Describe how data was processed
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Folium and Plotly Dash**
- **Perform predictive analysis using classification models**
  - How to build, tune, evaluate classification models

Data Collection – SpaceX API

- We collected historical Falcon 9 launch records by making a get request to Space X REST API.

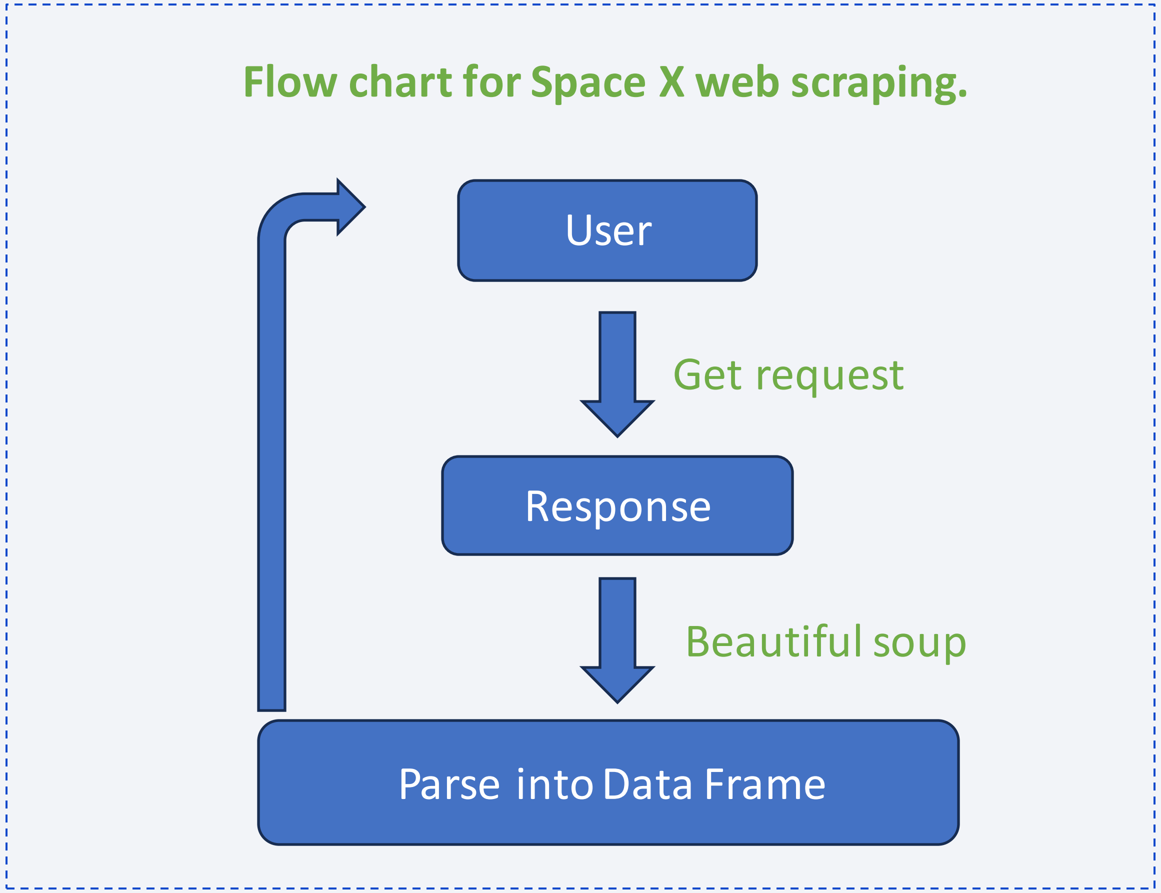
- Please find a link to the jupyter notebook with all the details here.

Flow chart for Space X REST API calls.


Response
Client
Request
Space X REST API
https://api.spacexdata.com/v4/rockets
https://api.spacexdata.com/v4/launchpads
https://api.spacexdata.com/v4/payloads
https://api.spacexdata.com/v4/cores

Data Collection - Scraping

- We also collected Falcon 9 historical launch records by web scraping a Wikipedia page.
- Please find a link to the jupyter notebook with all the details here.



Data Wrangling

- The first step was to analyze missing values.
  - The column 'PayloadMass' had 5 missing values which we substituted by the mean.
  - The column 'LandingPad' had 26 missing values, but since they can be interpreted as a non-attempted landing, we left them as such.

- The second step was to create a new variable 'Class' with the value of 1 if the outcome was favorable and 0 otherwise.

- For more details here are link to the relevant Jupyter notebooks.
  - Data_wrangling1 and Data_wrangling2.

EDA with Data Visualization

- Summary of plots and their use.
  - Scatter plots: Used to investigate the relation between two variables, for instance 'PayloadMass' and 'FlightNumber'.
  - Bar charts: Used to measure the relevance of different categories with respect to a variable, for example we plotted the success rate for each orbit using a bar chart.
  - Line plots: Used to visualize a trend over time, for example we plotted such a line to visualize success rate over the years.

Link to Jupyter Notebook.

EDA with SQL

- Summary of SQL queries:
  - SELECT: Used to select relevant information from the table.
  - WHERE: Used to specify selection criteria.
  - GROUP BY: Used to group data using values from a column.
  - HAVING: Used together with GROUP BY to specify selection criteria.
  - ORDER BY: Used to sort data using values from a column.

Link to Jupyter Notebook

Build an Interactive Map with Folium

- Summary of map objects and their use:
    - Circle: Used to create a circle at NASA Johnson Space Center's coordinates with a popup label showing its name.
    - Marker: Used to create a circle at NASA Johnson Space Center's coordinates with a popup label showing its name.
    - Marker Cluster: Used to create circles representing coordinates that are very close to each other.
    - Mouse Position: Used to obtain coordinates of any point in the map just by placing the mouse pointer on top of it.
    - PolyLine: Used to join two marked points by a straight line.

Link to Jupyter Notebook.

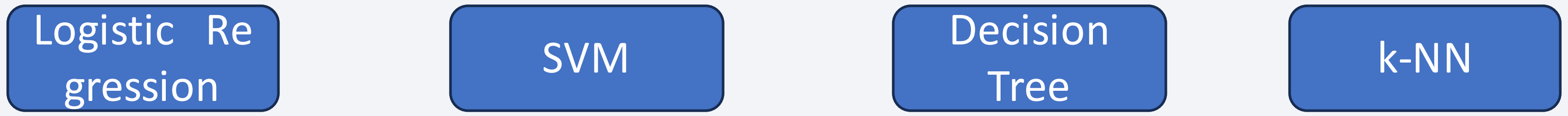
Build a Dashboard with Plotly Dash

- Summary of plots and interactions.
    - Dropdown: Used to select launch site.
    - Pie chart: Used to visualize success rate depending on launch site.
    - Slider: Used to select payload mass.
    - Scatter plot: Used to visualize correlation between payload and launch success.

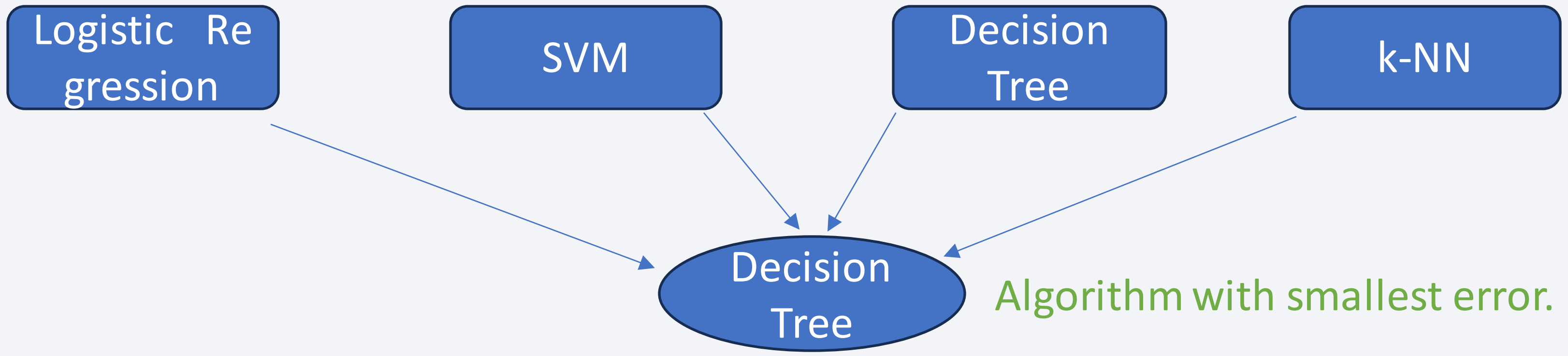
Link to Python code.

Predictive Analysis (Classification)

We used the following Machine Learning classification algorithms.



For each of them we measured the error with different metrics and selected the algorithm that performed best overall.



Algorithm with smallest error.

Link to Jupyter Notebook.

Results

In the following slides we will take a deeper look into our results. In particular,

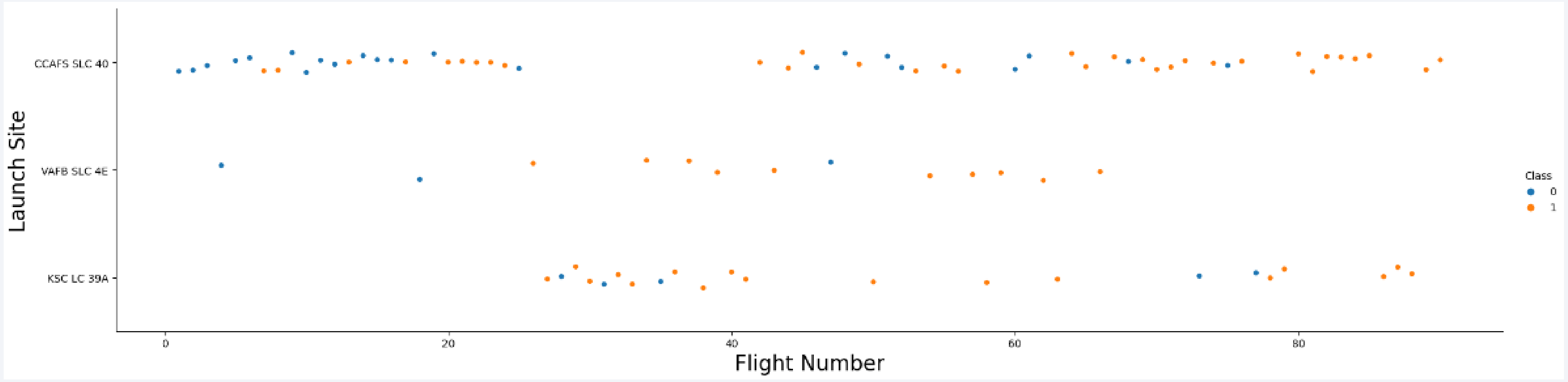
- Exploratory data analysis results.
- Interactive analytics demo in screenshots.
- Predictive analysis results.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

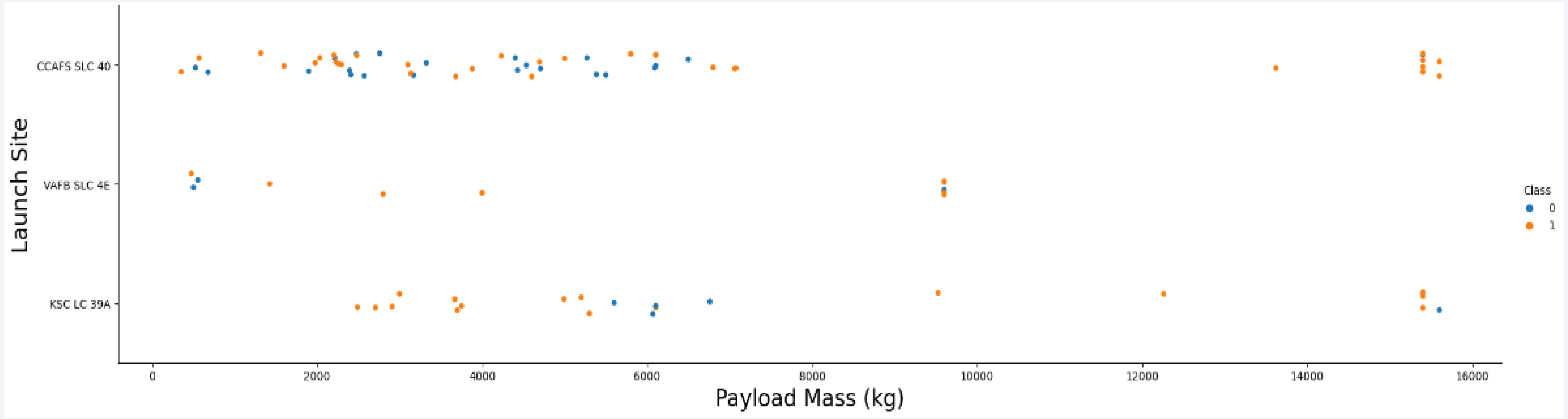
The following is a scatter plot of Flight Number vs. Launch Site.



We have color coded the success classification, which allows us to see that as the number of flights increases, also does the success rate across all launch sites.

Payload vs. Launch Site

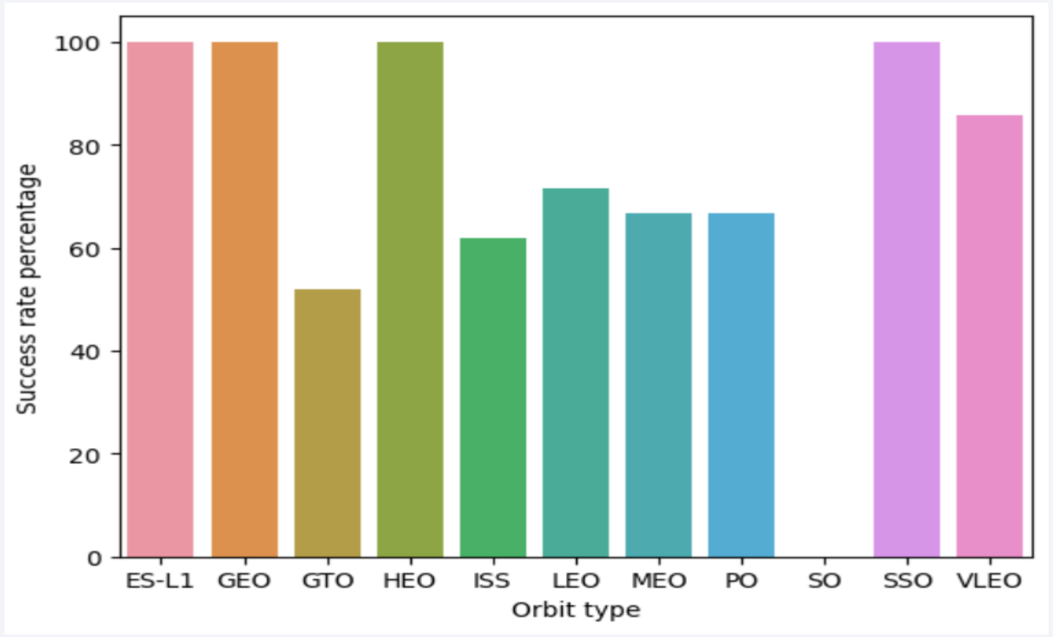
The following is a scatter plot of Payload vs. Launch site.



We observe that for the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000).

Success Rate vs. Orbit Type

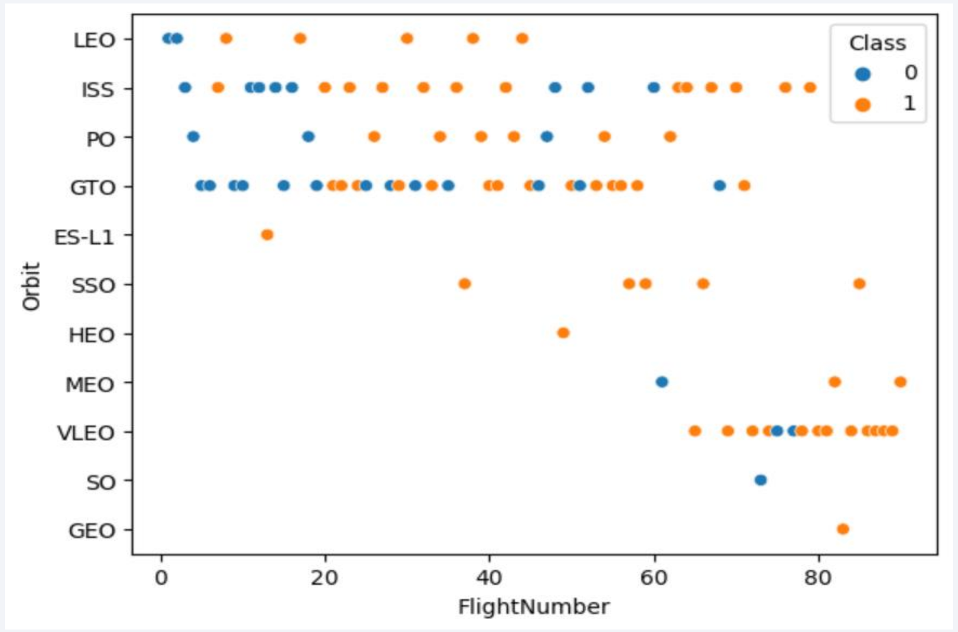
The following is a bar chart of Success Rate vs. Orbit Type.



We observe that four orbits have 100% success rate, but this can be deceiving since the amount of flights is quite low for those orbits as can be seen in the next chart.

Flight Number vs. Orbit Type

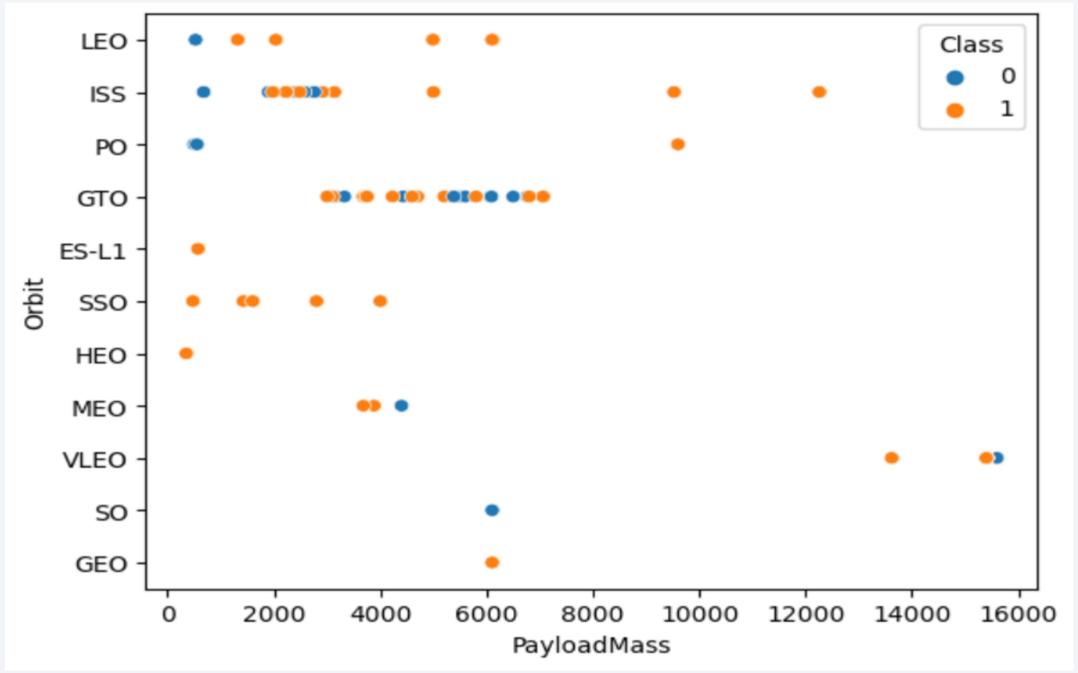
The following is a scatter plot of Flight Number vs. Orbit Type.



We observe that for some orbits, like LEO, 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

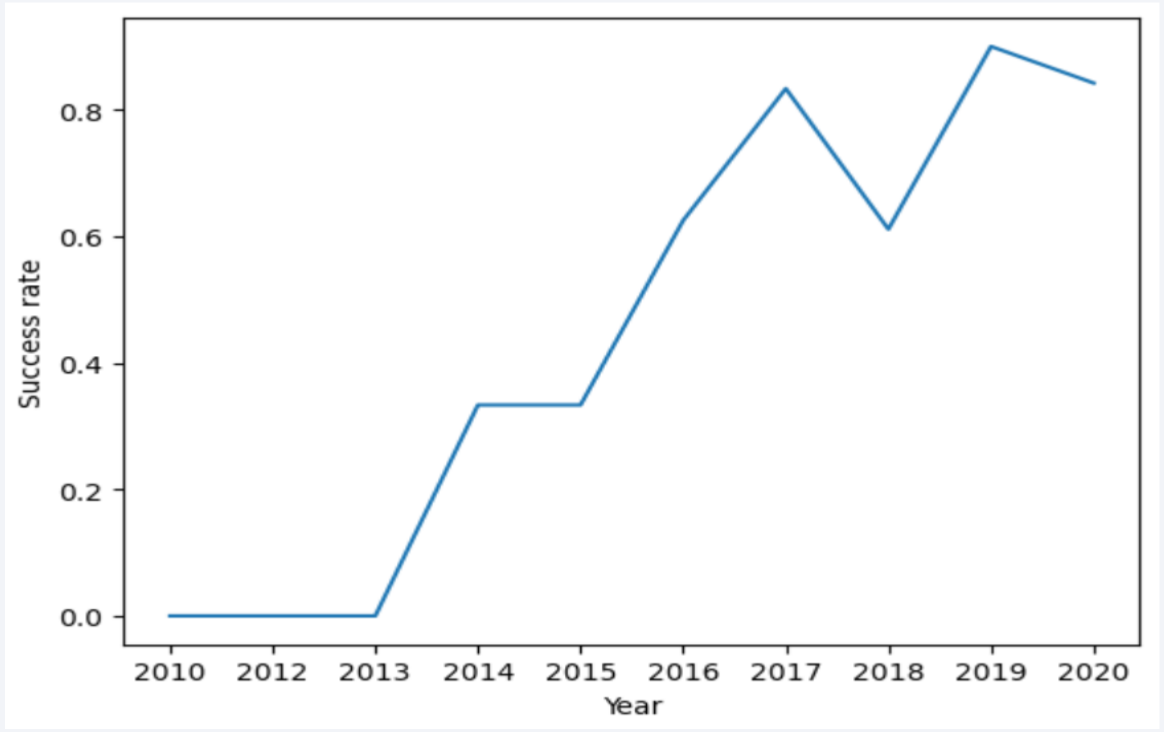
The following is a scatter plot of Payload vs. Orbit Type.



We observe that with heavy payloads the successful landing or positive landing rate are higher for Polar, LEO and ISS.

Launch Success Yearly Trend

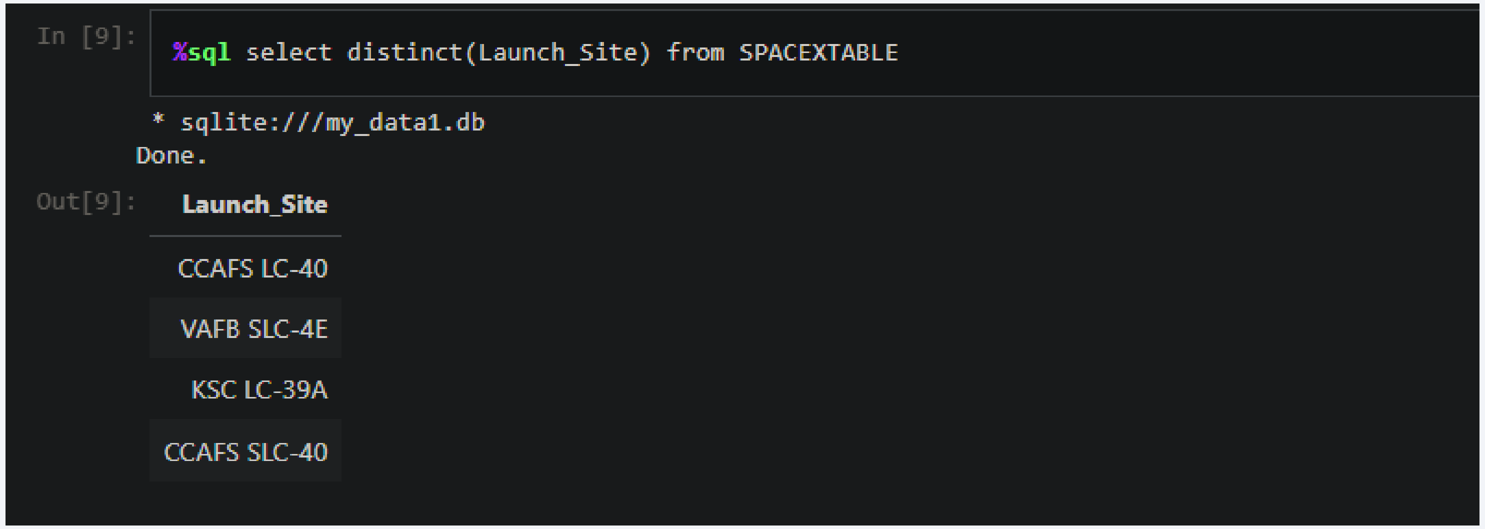
The following is a line plot showing Launch Success Yearly Trend.



We observe that from 2013 onwards the success rate has been increasing overall.

All Launch Site Names

Query to obtain the names of the unique launch sites:

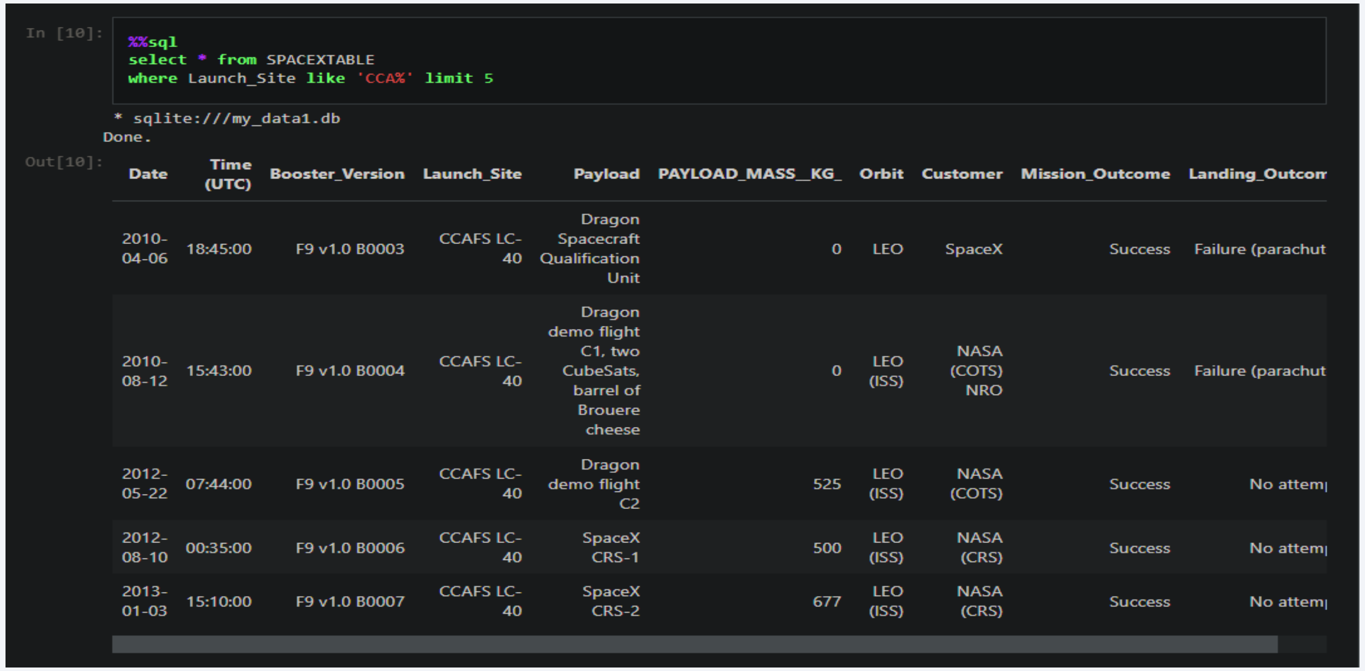


```
In [9]:  %sql select distinct(Launch_Site) from SPACEXTABLE

          * sqlite:///my_data1.db
         Done.
Out[9]:   Launch_Site
          CCAFS LC-40
          VAFB SLC-4E
          KSC LC-39A
          CCAFS SLC-40
```

Here we used the function *distinct* to only show the unique values for launch sites.

Launch Site Names Begin with 'CCA'

Query to obtain the 5 records where launch sites begin with the string 'CCA':



```
In [10]:
%%sql
select * from SPACEXTABLE
where Launch_Site like 'CCA%' limit 5
```

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

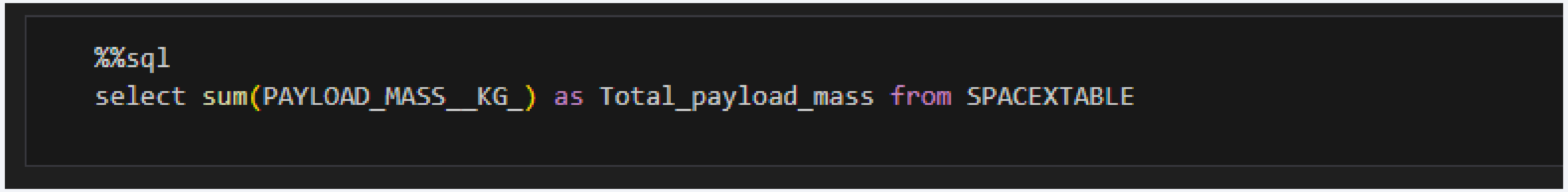
Out[10]:

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

Here we used the *where* statement to filter the data and *limit* statement to display only the desired number of rows.

Total Payload Mass

Query to obtain the total payload mass:

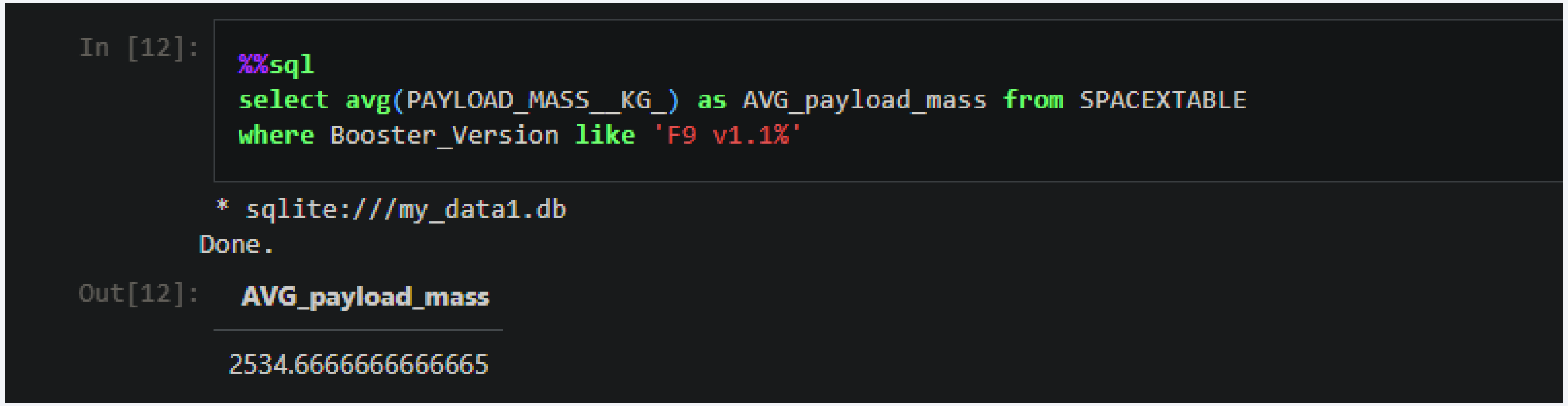


```
%%sql
select sum(PAYLOAD_MASS__KG_) as Total_payload_mass from SPACEXTABLE
```

Here we used the *sum* function to obtain the total payload mass.

Average Payload Mass by F9 v1.1

Query to obtain the average payload mass by F9 v1.1:

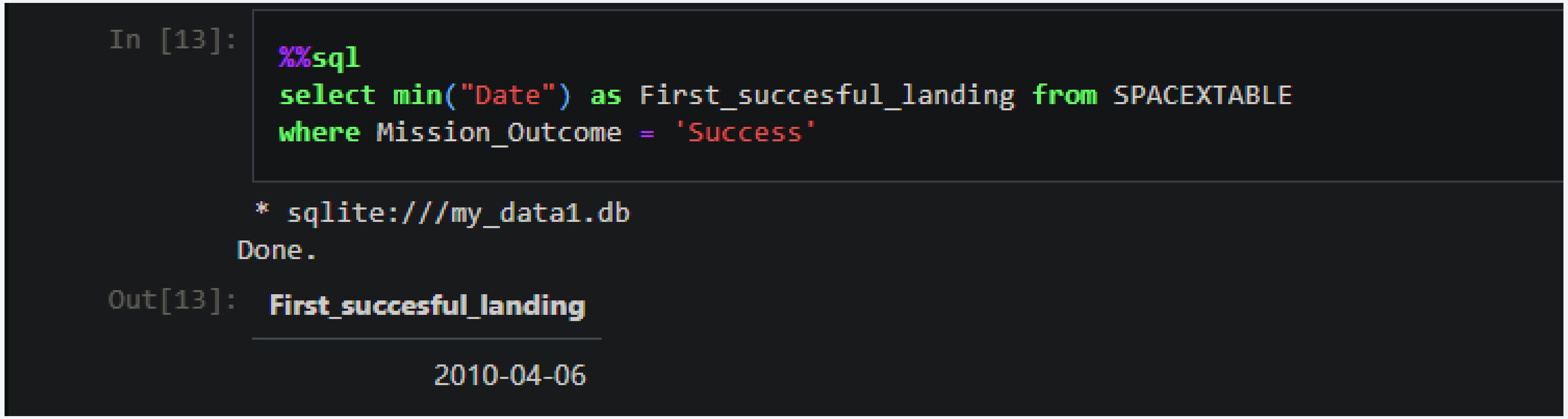


```
In [12]:  %%sql
          select avg(PAYLOAD_MASS__KG_) as AVG_payload_mass from SPACEXTABLE
          where Booster_Version like 'F9 v1.1%'

 * sqlite:///my_data1.db
Done.

Out[12]:  AVG_payload_mass
          2534.6666666666665
```

Here we used the *where* clause together with *like* to filter the data. The *avg* function calculates the average after filtering.

First Successful Ground Landing Date

Query to obtain the date of the first successful landing outcome:

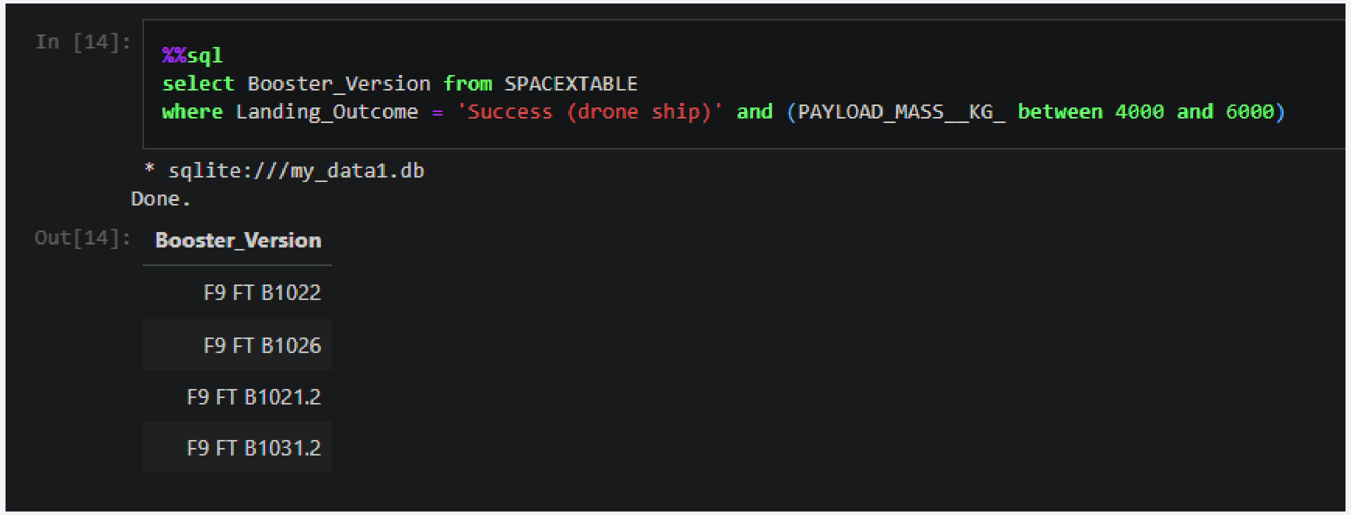


```
In [13]: %%sql
         select min("Date") as First_succesful_landing from SPACEXTABLE
         where Mission_Outcome = 'Success'

          * sqlite:///my_data1.db
         Done.

Out[13]: First_succesful_landing
                      2010-04-06
```

Here we used the *where* clause to filter the data.

Successful Drone Ship Landing with Payload between 4000 and 6000

Query to obtain Successful Drone Ship Landing with Payload between 4000 and 6000:

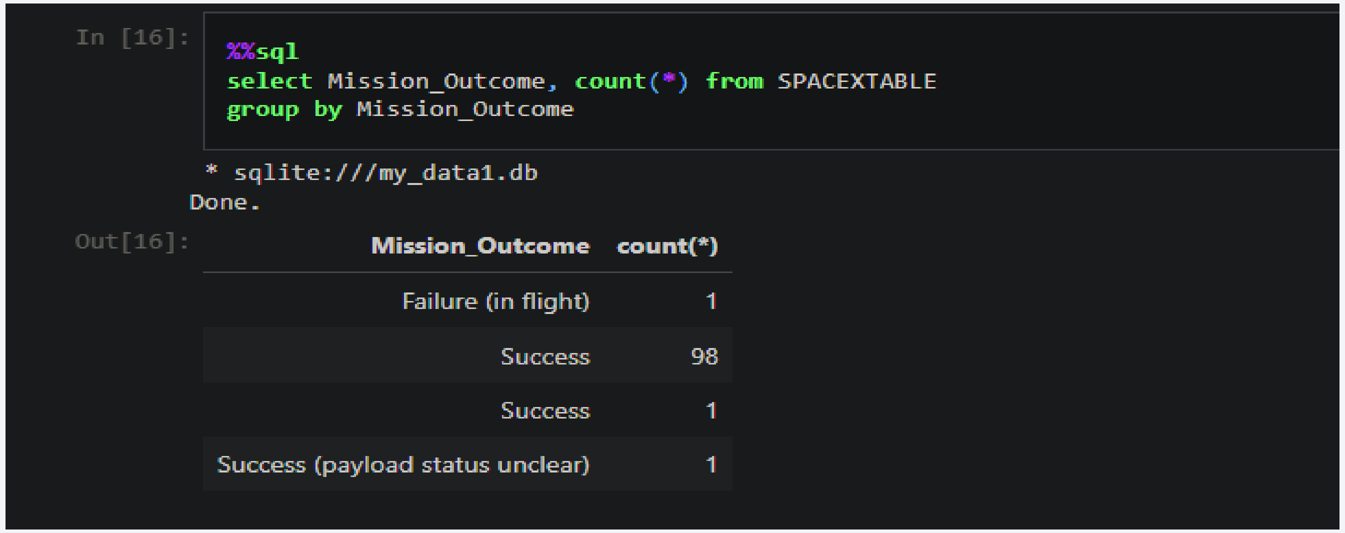


```
In [14]: %%sql
         select Booster_Version from SPACEXTABLE
         where Landing_Outcome = 'Success (drone ship)' and (PAYLOAD_MASS__KG_ between 4000 and 6000)

          * sqlite:///my_data1.db
         Done.

Out[14]: Booster_Version
           F9 FT B1022
           F9 FT B1026
         F9 FT B1021.2
         F9 FT B1031.2
```

Here we used the *where* clause to filter the data together with *between* to specify an interval of values.

Total Number of Successful and Failure Mission Outcomes

Query to obtain Total Number of Successful and Failure Mission Outcomes:



```
In [16]:  %%sql
          select Mission_Outcome, count(*) from SPACEXTABLE
          group by Mission_Outcome

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

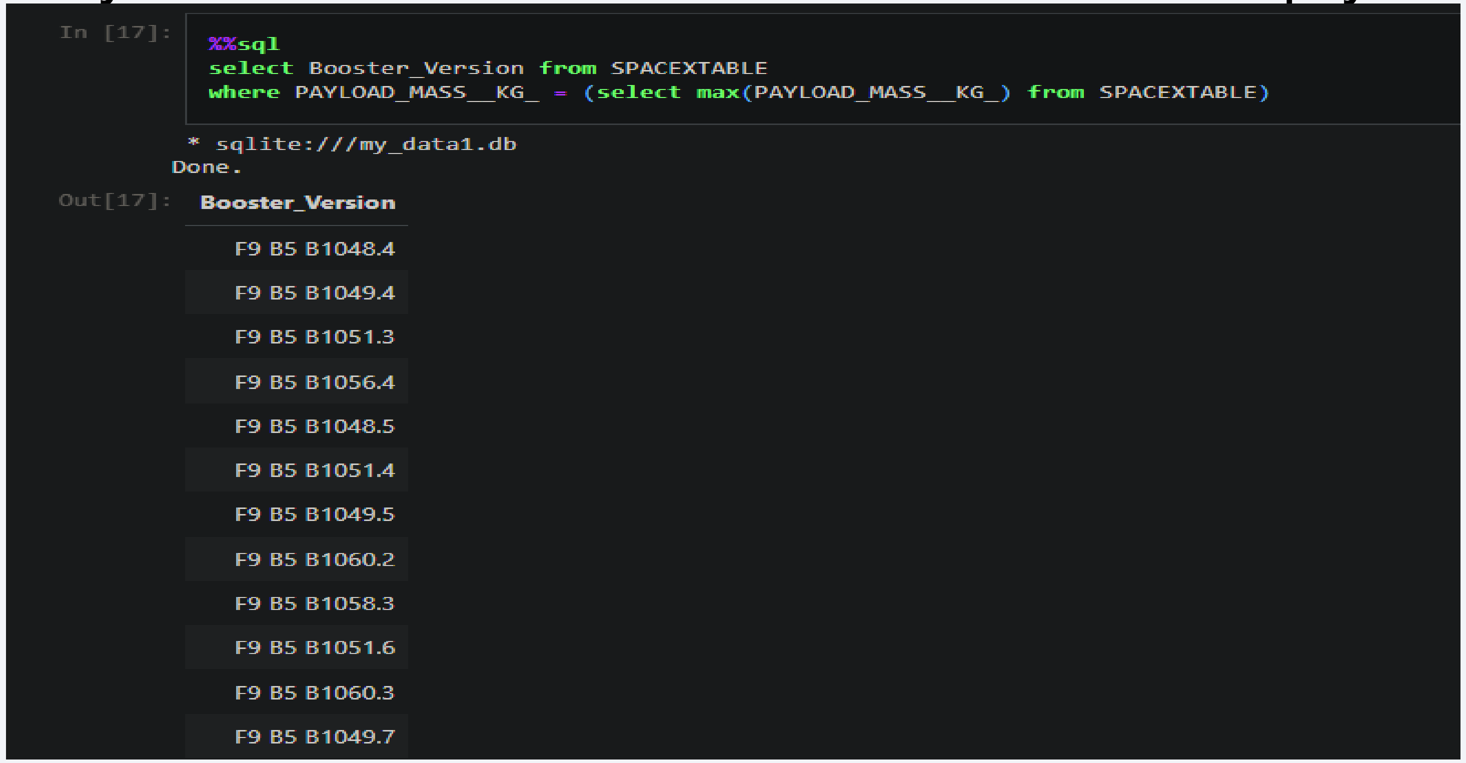
Out[16]:

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

Here we used the *group by* clause to filter the data together with *count* to obtain the number of successful and failure mission outcomes.

Boosters Carried Maximum Payload

Query to obtain the boosters that carried maximum payload:



```
In [17]: %%sql
         select Booster_Version from SPACEXTABLE
         where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)

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

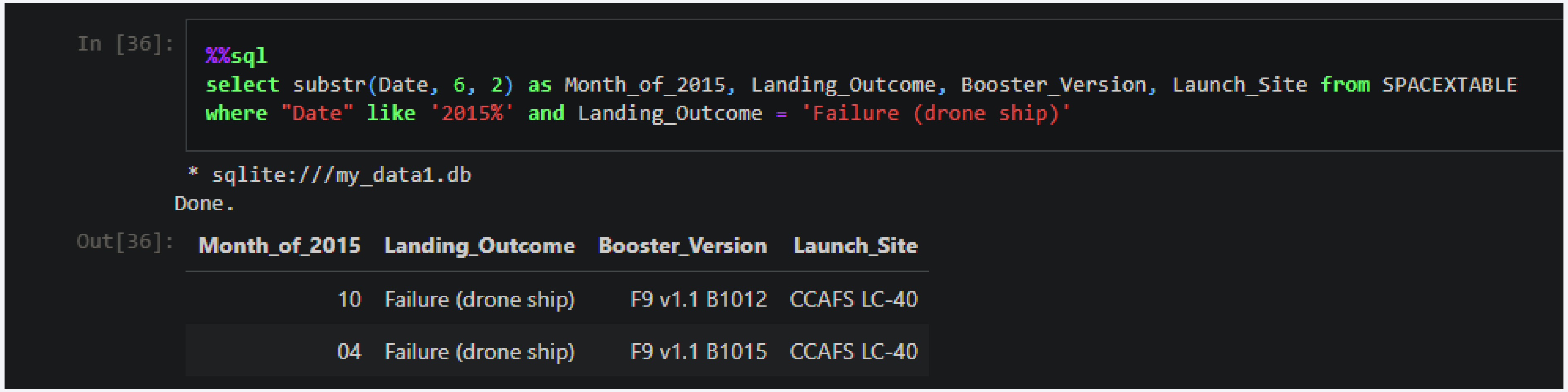
Out[17]:

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

Here we used the *where* clause to filter the data together with a subquery to select the maximum payload.

2015 Launch Records

Query to obtain the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015:



```
In [36]: %%sql
         select substr(Date, 6, 2) as Month_of_2015, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE
         where "Date" like '2015%' and Landing_Outcome = 'Failure (drone ship)'

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

Out[36]:

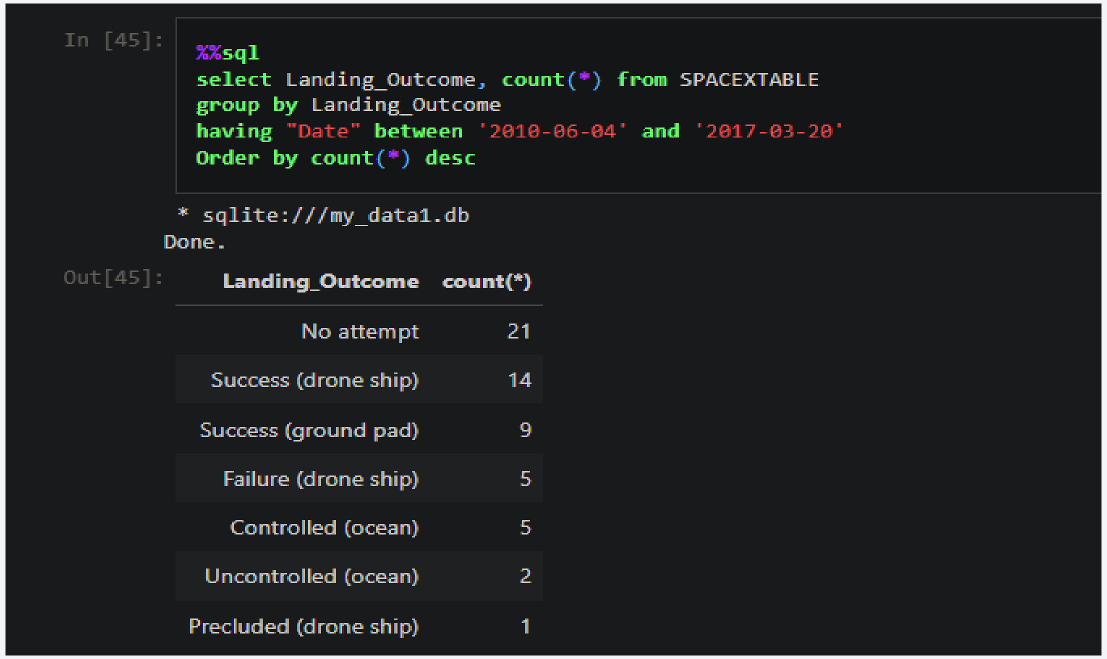
Month_of_2015	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Here we used the *where* clause to filter the data together with *like* command.

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

Query to obtain the ranking of Landing Outcomes Between

2010-06-04 and 2017-03-20:



```
In [45]: %%sql
         select Landing_Outcome, count(*) from SPACEXTABLE
         group by Landing_Outcome
         having "Date" between '2010-06-04' and '2017-03-20'
         Order by count(*) desc

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

Out[45]:

Landing_Outcome	count(*)
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Uncontrolled (ocean)	2
Precluded (drone ship)	1

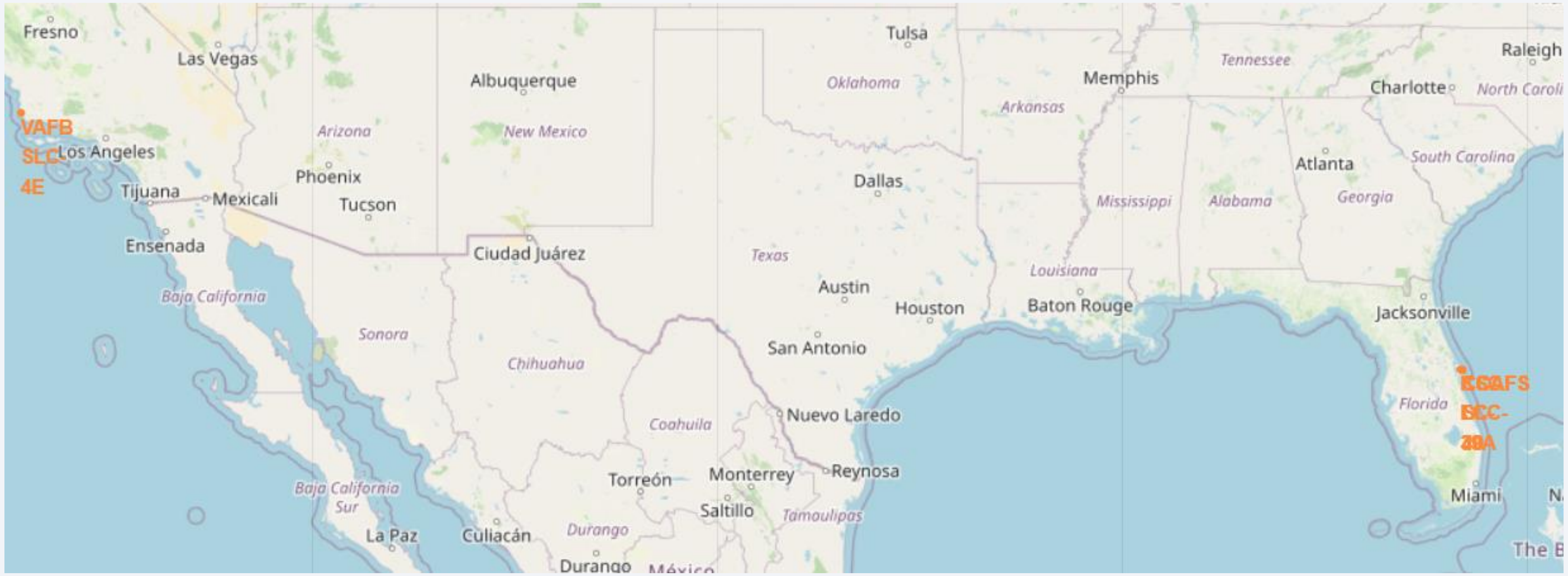
Here we used the *group by* clause to filter the data together with *having* clause. Finally, we used the *order by* clause to rank the outcomes.

Section 3

Launch Sites Proximities Analysis

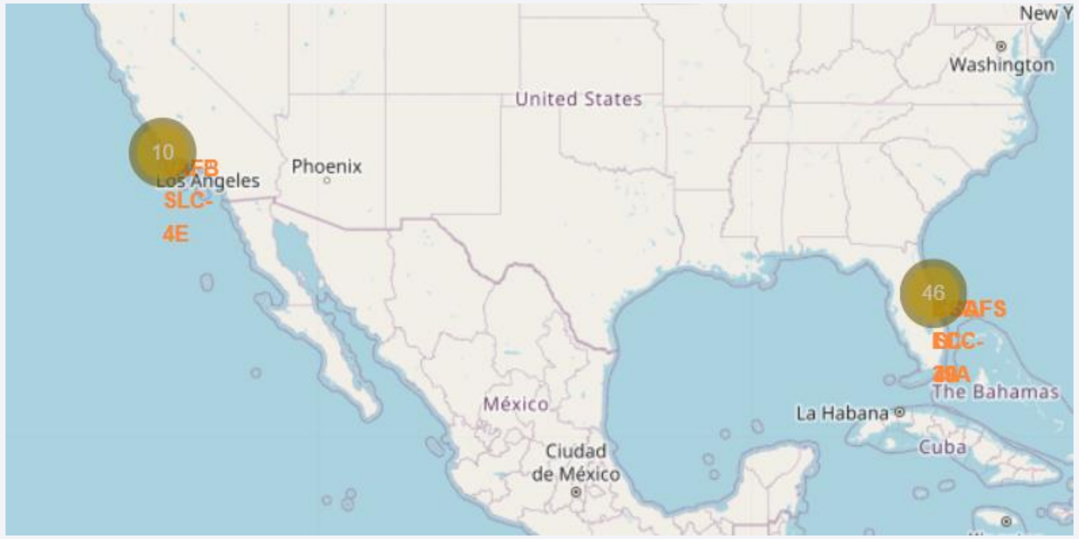
Map of Launch Sites

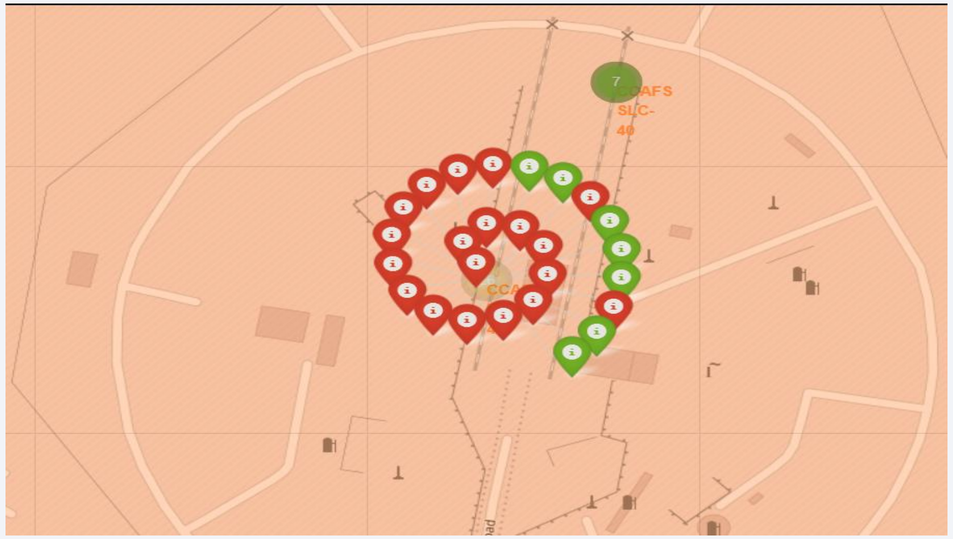
In the following map we have added *markers* to highlight the different launch sites.



Color Coded Markers

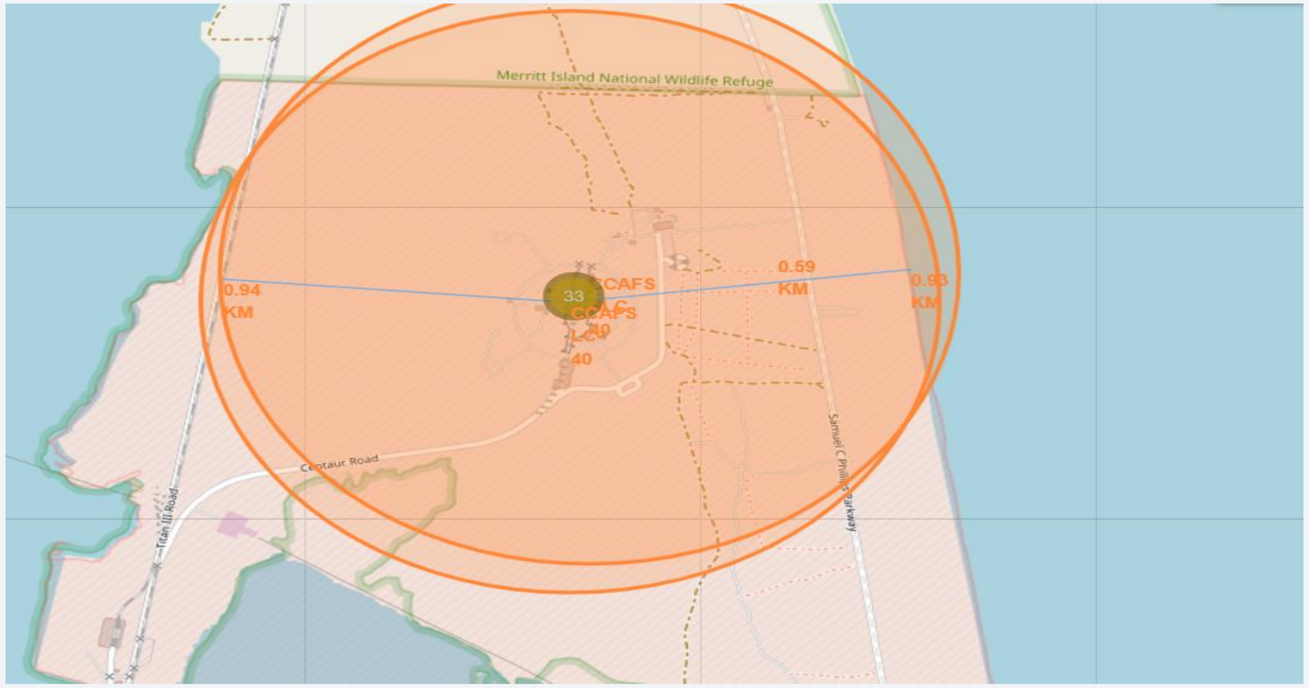
In the next maps we can see markers for each launch outcome. They are color coded: green means success and red means failure.

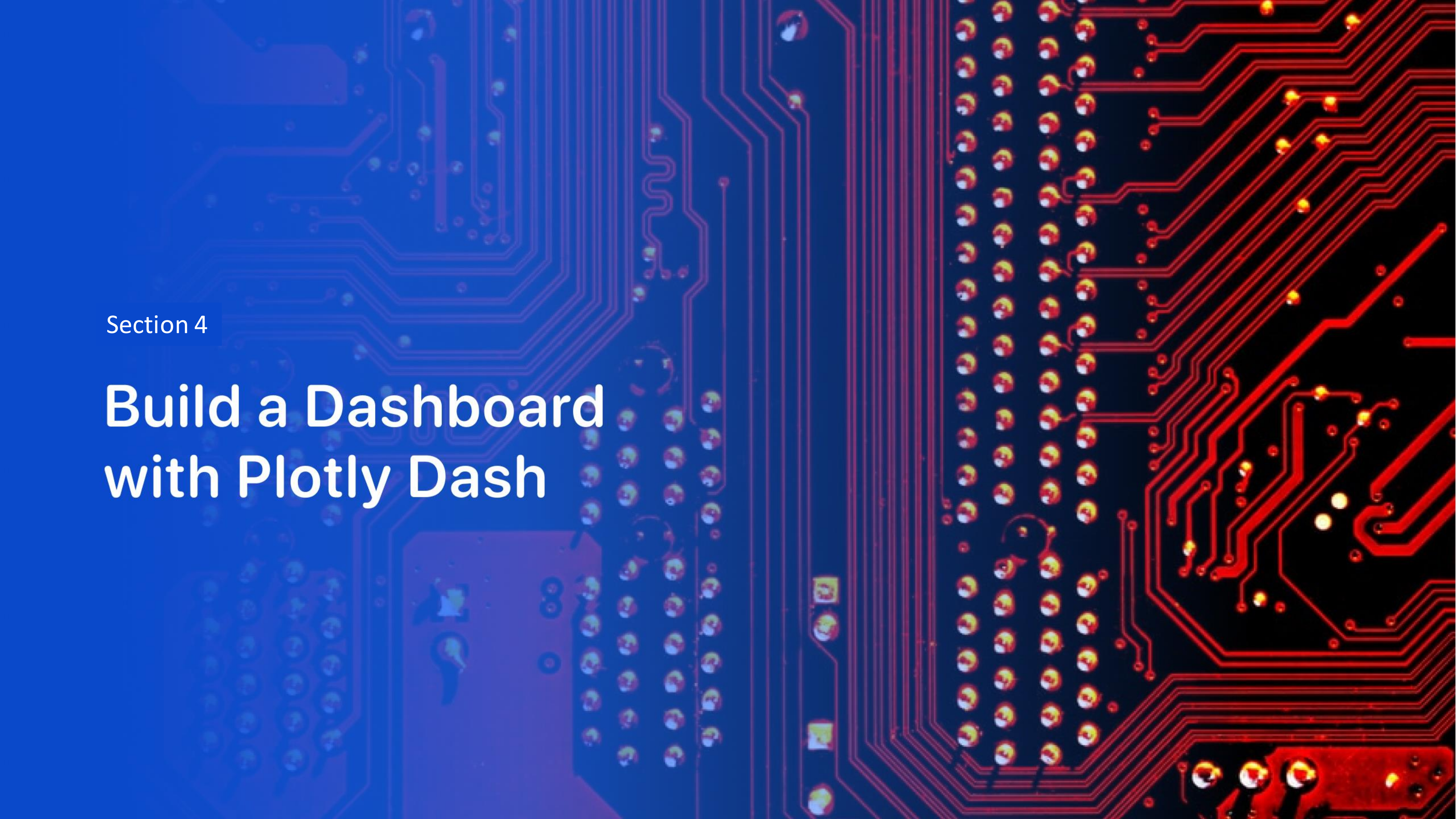




Distance to Railway and Coast

In the following map we have included the distance to the coast and to a railway in the proximity of one of the launch sites. We have also added a line joining the coordinates as well as the value of the distances.


Merritt Island National Wildlife Refuge
0.94 KM
33
CCAFS
CCAFS
LC-40
0.59 KM
0.93 KM
Centaur Road
Titan III Road
Samuel C Phillips Parkway

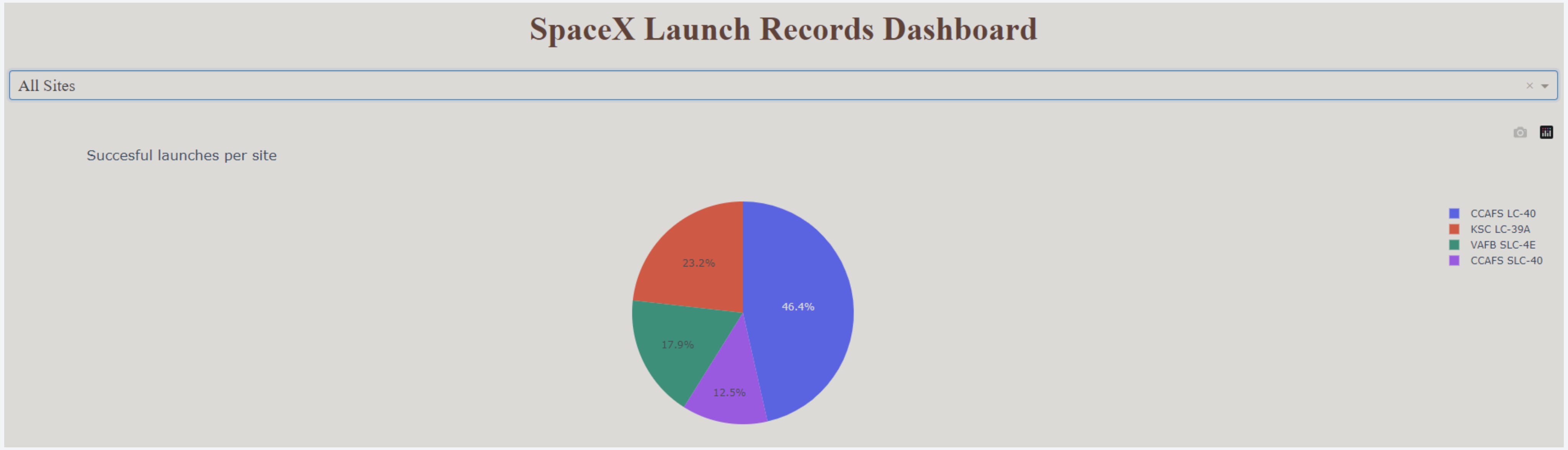


Section 4

Build a Dashboard with Plotly Dash

Successful Launches

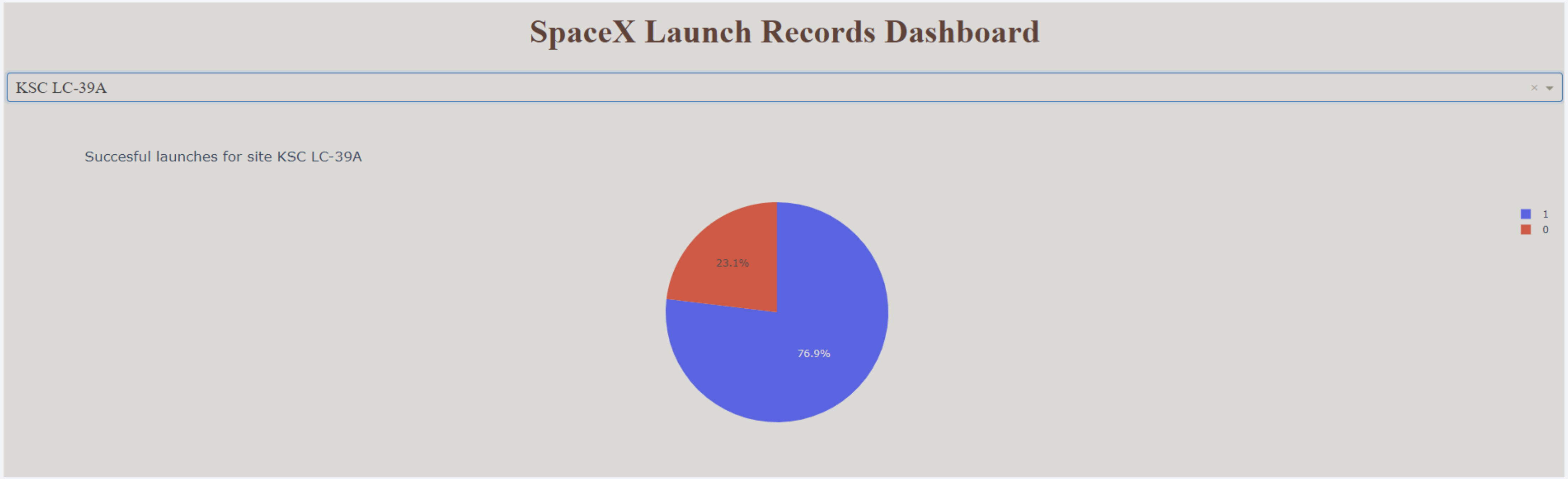
The following pie chart shows the success rate across all launch sites:



Here we can see the dropdown menu with the corresponding pie chart which depends on the site selection.

Site with highest launch success

The following pie chart shows the site with highest success rate:



SpaceX Launch Records Dashboard

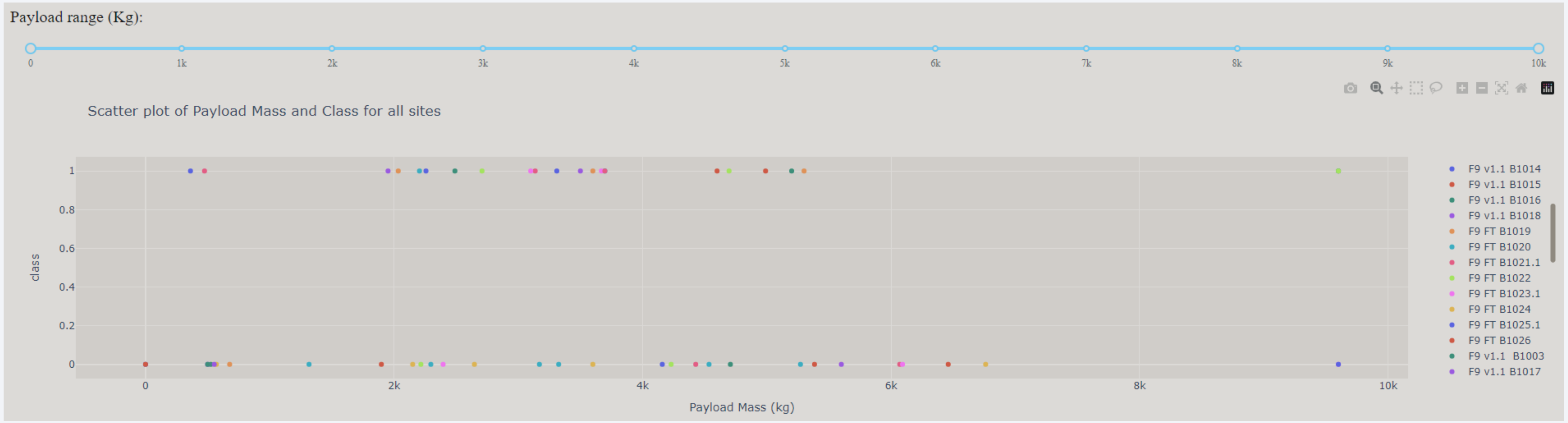
KSC LC-39A

Succesful launches for site KSC LC-39A

Here we can see the dropdown menu with the corresponding pie chart which depends on the site selection.

Payload vs. Launch Outcome scatter plot

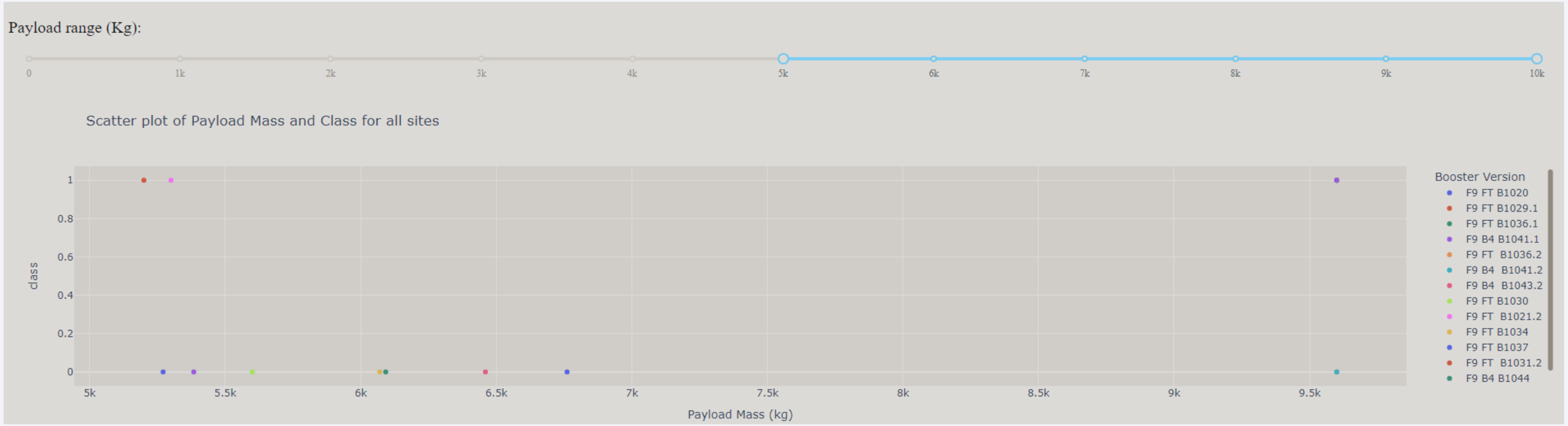
The following scatter plot shows the payload vs. Outcome for all sites:



Here we can see the slider with the corresponding scatter plot which depends on the payload selection. In this case we have selected the whole range of payloads and we can see that the booster with highest success rate is F9 FT.

Payload vs. Launch Outcome scatter plot

The following scatter plot shows the payload vs. Outcome for all sites:



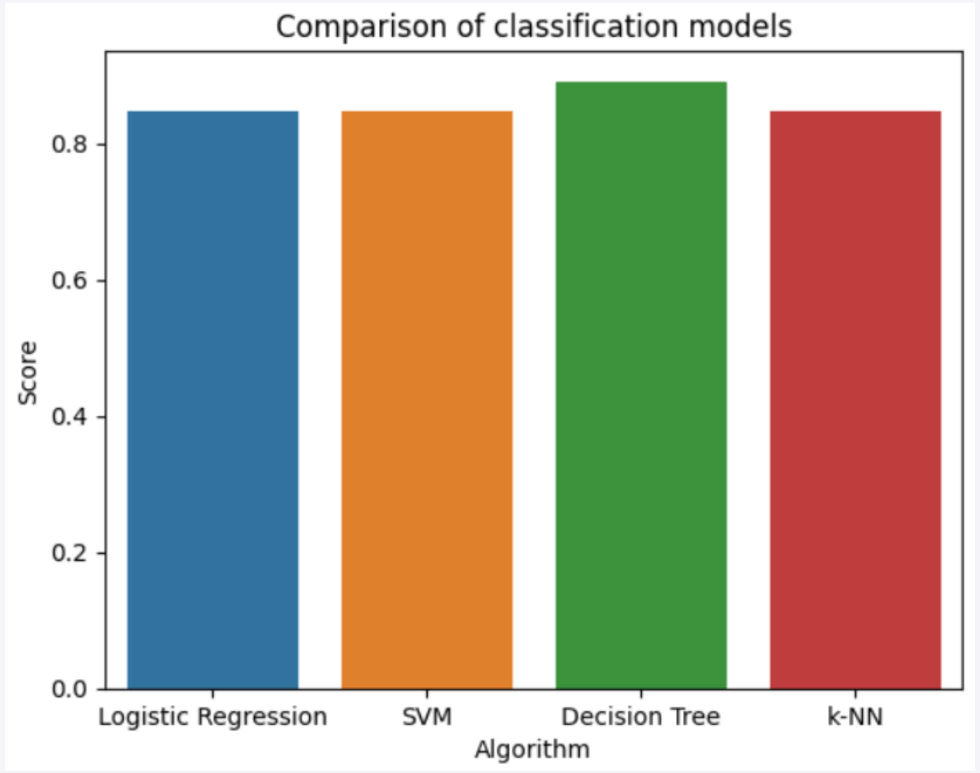
Here we have selected payloads larger than 5k. In this case we can see that still the booster with highest success rate is F9 FT, however the success rate has decreased with heavier loads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

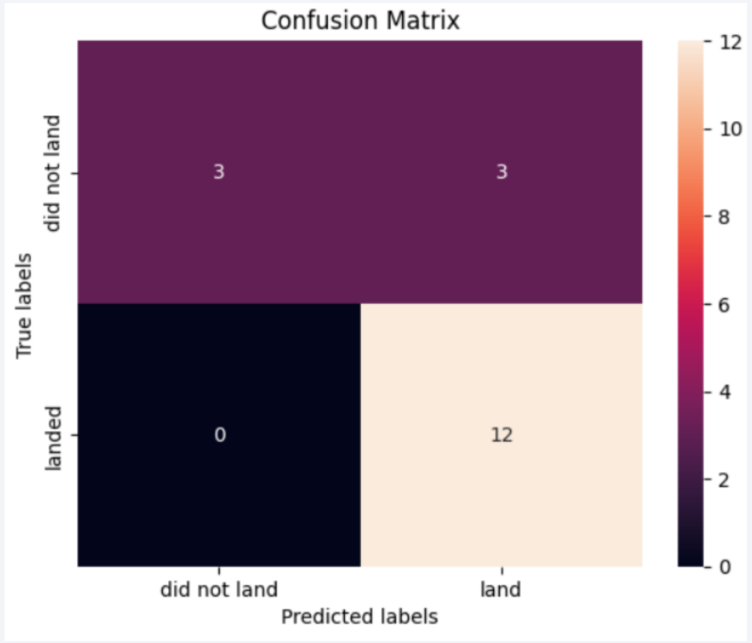
The following bar chart shows the scores for different models



We can see that the model with the highest score is Decision Tree.

Confusion Matrix

The following is the confusion matrix of the Decision Tree Classifier:



We can see that the algorithm predicts very well the cases that actually landed, however there are 3 false positives.

Conclusions

- We were able to gather and clean the necessary data sets.
- Using matplotlib and folium we have performed EDA and prepared the data for further analysis.
- Using Machine Learning classification algorithms we were able to select the best model to predict landing outcome.
- This results can have a powerful impact on the company's performance and help in the decision making process.

Thank you!