IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Batuhan Çağlar>
<13.09.2024>

Outline

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

Executive Summary

- Summary of methodologies
  - Data collected from SpaceX API as json then normalised.
  - Launch data collected from wikipedia using BeautifulSoup.
  - Data preprocessed for detailed analize.
  - Data analized detailed using SQL.
  - With seaborn relation between various variables inspected.
  - Launch sites and crash/landing sites marked on map an promixity to various landmarks calculated with the help of folium.
  - An interactive dashboard built with dash and plotly for further inspection.
  - LR, SVM, Desicion Tree, KNN methods applied for predicting if the spaceship will crush or sucessfully land.
- Summary of all results
  - Data analysis showed that with increasing payload mass spaceships less tend to crush.
  - Also after 2010 there is a remarkeble fall in crushes.
  - KNN method is the most suitable model for the prediction.

Introduction

- Project background and context
  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Problems you want to find answers
  - In this capstone, we will predict if the Falcon 9 first stage will land successfully using data from Falcon 9 rocket launches advertised on its website.

Section 1

Methodology

Methodology

Executive Summary

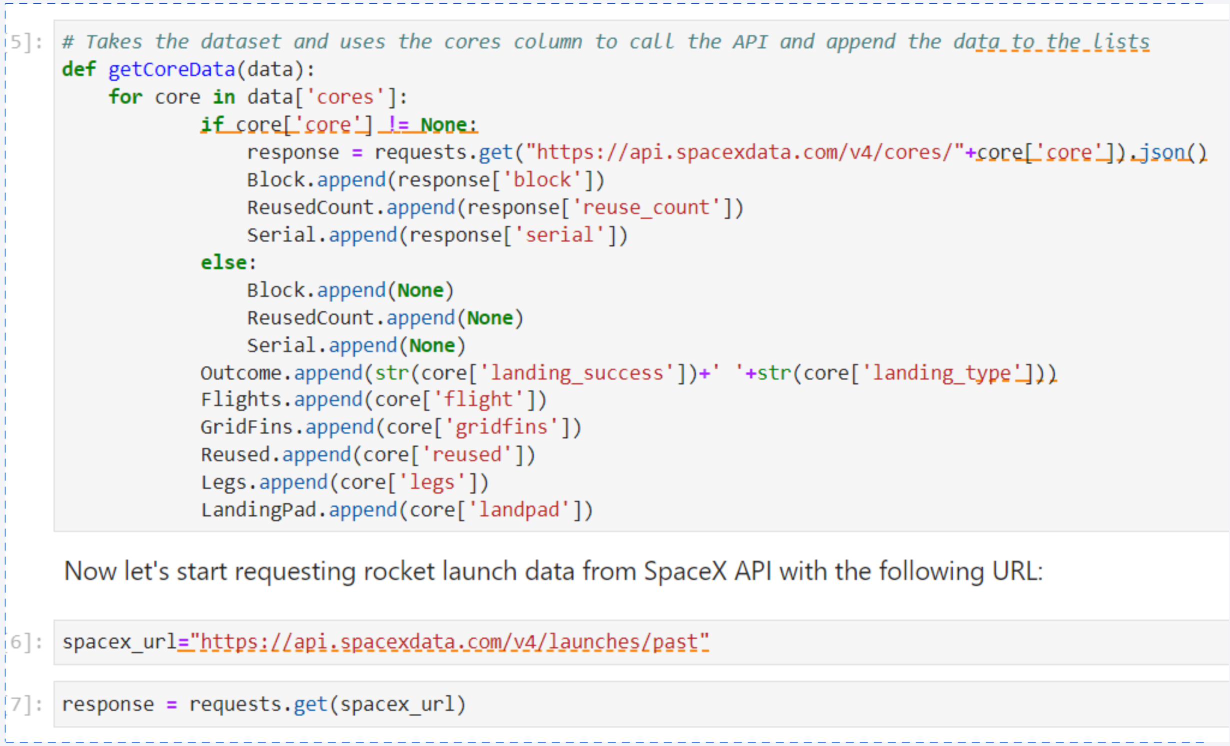
- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using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

Data Collection

- Description of how SpaceX Falcon9 data was collected.
  - Data was first collected using SpaceX API as json. With the help of some python libraries data was normalized and converted into a pandas dataframe.
  - From wikipedia webpage web scrapped and using beutifulsoup library, created a pandas dataframe with Falcon 9 launches.

Data Collection – SpaceX API

- Here is the link for data collection notebook.



```
5]: # Takes the dataset and uses the cores column to call the API and append the data to the lists
    def getCoreData(data):
        for core in data['cores']:
                if core['core'] != None:
                    response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                    Block.append(response['block'])
                    ReusedCount.append(response['reuse_count'])
                    Serial.append(response['serial'])
                else:
                    Block.append(None)
                    ReusedCount.append(None)
                    Serial.append(None)
                Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
                Flights.append(core['flight'])
                GridFins.append(core['gridfins'])
                Reused.append(core['reused'])
                Legs.append(core['legs'])
                LandingPad.append(core['landpad'])
```

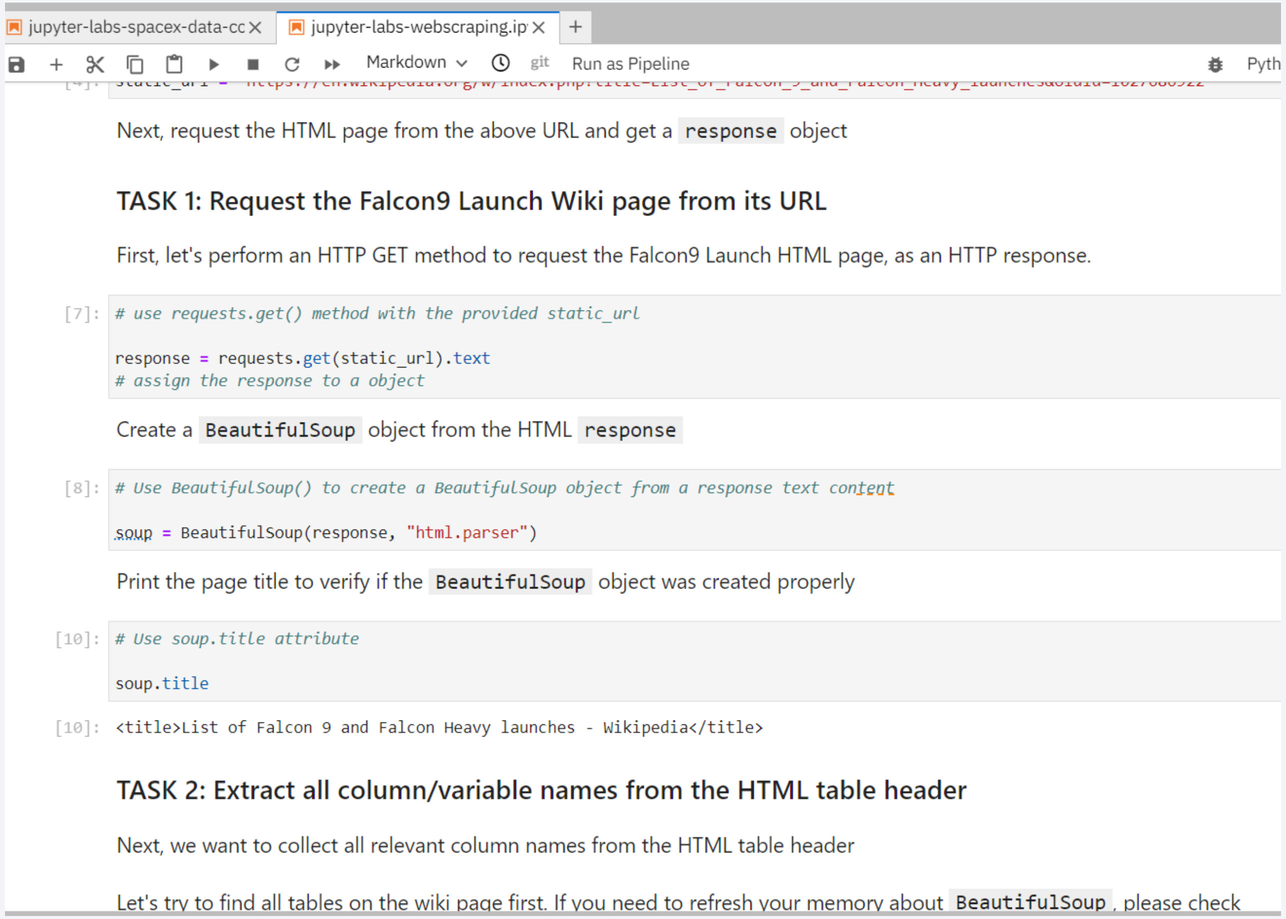
Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
6]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
7]: response = requests.get(spacex_url)
```

Data Collection - Scraping

- You can reach the notebook here.



```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
[7]: # use requests.get() method with the provided static_url

     response = requests.get(static_url).text
     # assign the response to a object
```

Create a `BeautifulSoup` object from the HTML `response`

```
[8]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content

     soup = BeautifulSoup(response, "html.parser")
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
[10]: # Use soup.title attribute

      soup.title

[10]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check

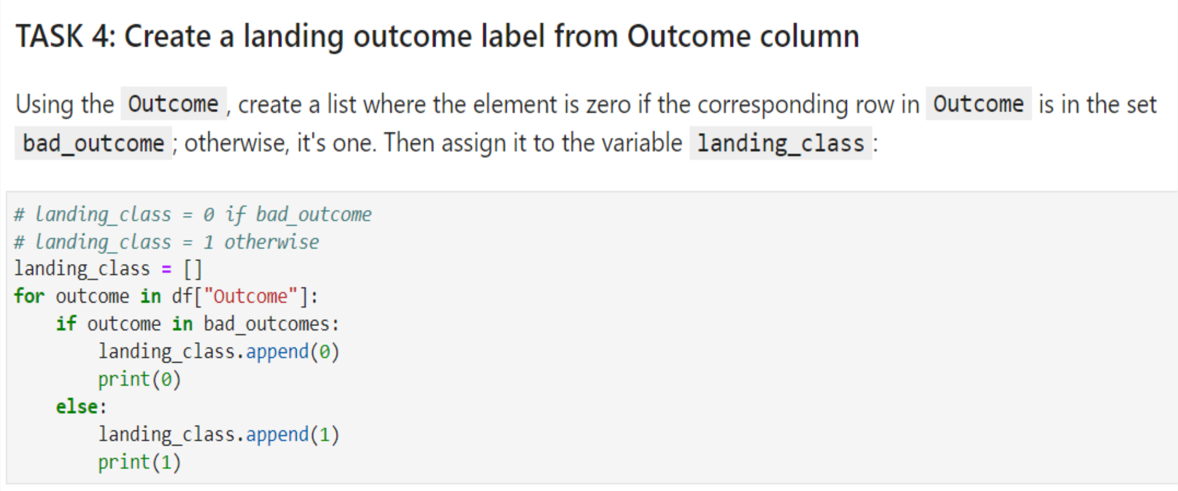
Data Wrangling

- Launches on each site, occurances in orbits and outcomes calculated.
- Outcomes classifed as 0 and 1

unsuccesfull and succesfull

respectively.

- You can find the notebook here.



TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`:

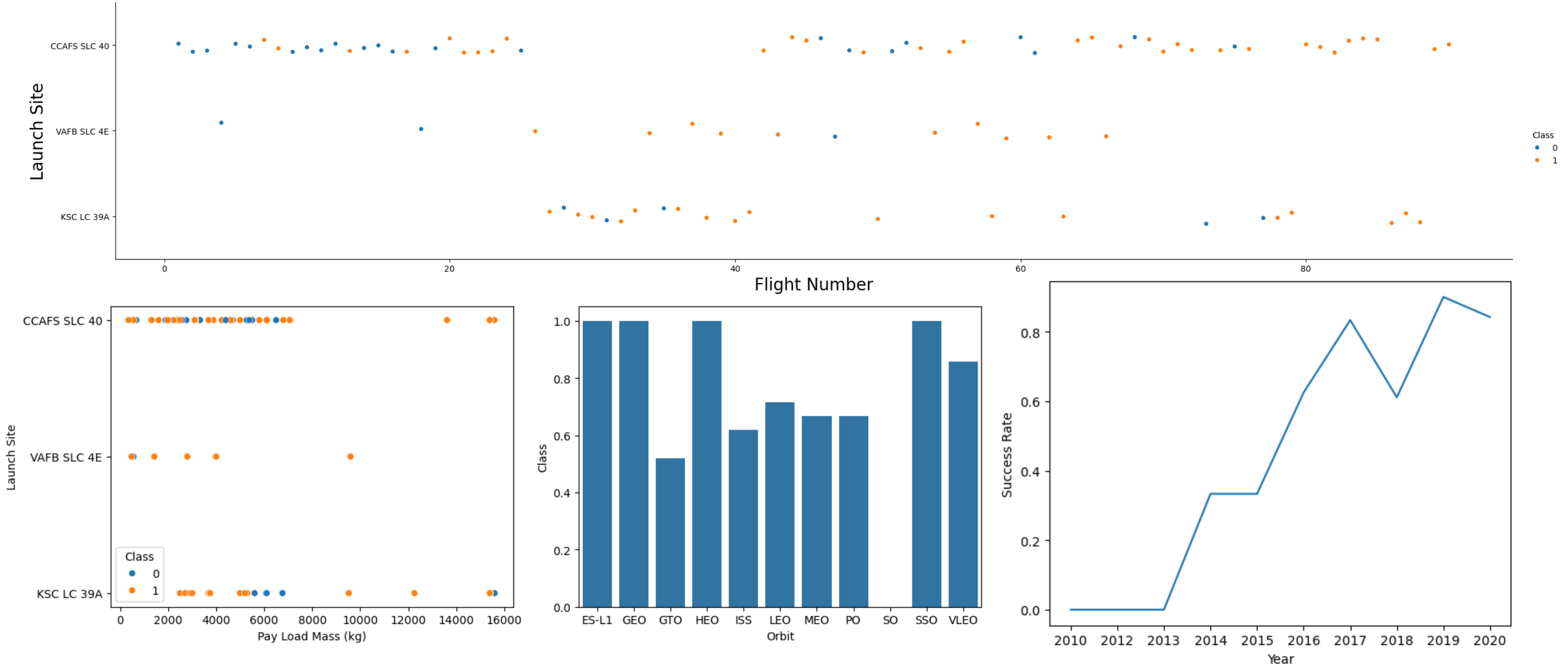
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for outcome in df["Outcome"]:
    if outcome in bad_outcomes:
        landing_class.append(0)
        print(0)
    else:
        landing_class.append(1)
        print(1)
```

EDA with Data Visualization

- Used scatter plots for:
  - FlightNumber vs. PayloadMas, FlightNumber vs. PayloadMas, PayloadMas vs. LaunchSite, FlightNumber vs. Orbit, PayloadMas vs. Orbit
- Used bar plots for:
  - Relationship between success rate of each orbit type
- Line plots for:
  - Launch success yearly trend

LINK for the notebook.

EDA with Data Visualization



LINK for the notebook.

EDA with SQL

- Names of unique launch sites displayed:
  - %sql SELECT DISTINCT(Launch_Site) FROM SPACEXTBL
- 5 records where launch sites begin with the string ‘CCA’ displayed:
  - %sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE "CCA%" LIMIT 5
- The total payload mass carried by boosters launched by NASA (CRS) displayed:
  - %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'

EDA with SQL

- Average payload mass carried by booster version F9 v1.1 displayed:
    - %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE ‘F9 v1.1’
- The total payload mass carried by boosters launched by NASA (CRS) displayed:
    - %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)’

EDA with SQL

- Date when the first succesful landing outcome in ground pad was acheived listed:
  - %sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = "Success (ground pad)"
- The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000 listed:
  - %sql SELECT Booster_Version, Payload FROM SPACEXTBL WHERE Landing_Outcome = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_< 6000

EDA with SQL

- The total number of successful and failure mission outcomes listed:
    - %sql SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTBL GROUP BY Landing_Outcome
- The names of the booster_versions which have carried the maximum payload listed by using sub querry:
    - %sql SELECt Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)

EDA with SQL

- Therecords which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015 listed:
  - %sql SELECT SUBSTR(Date,6,2) as Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTBL WHERE SUBSTR(Date,0,5)="2015" AND Landing_Outcome = "Failure (drone ship)"
- 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:
  - %sql SELECT COUNT(Landing_Outcome), Landing_Outcome FROM SPACEXTBL WHERE (Date BETWEEN "2010-06-04" AND "2017-03-20") GROUP BY "Landing_Outcome"

Notebook Link Is HERE

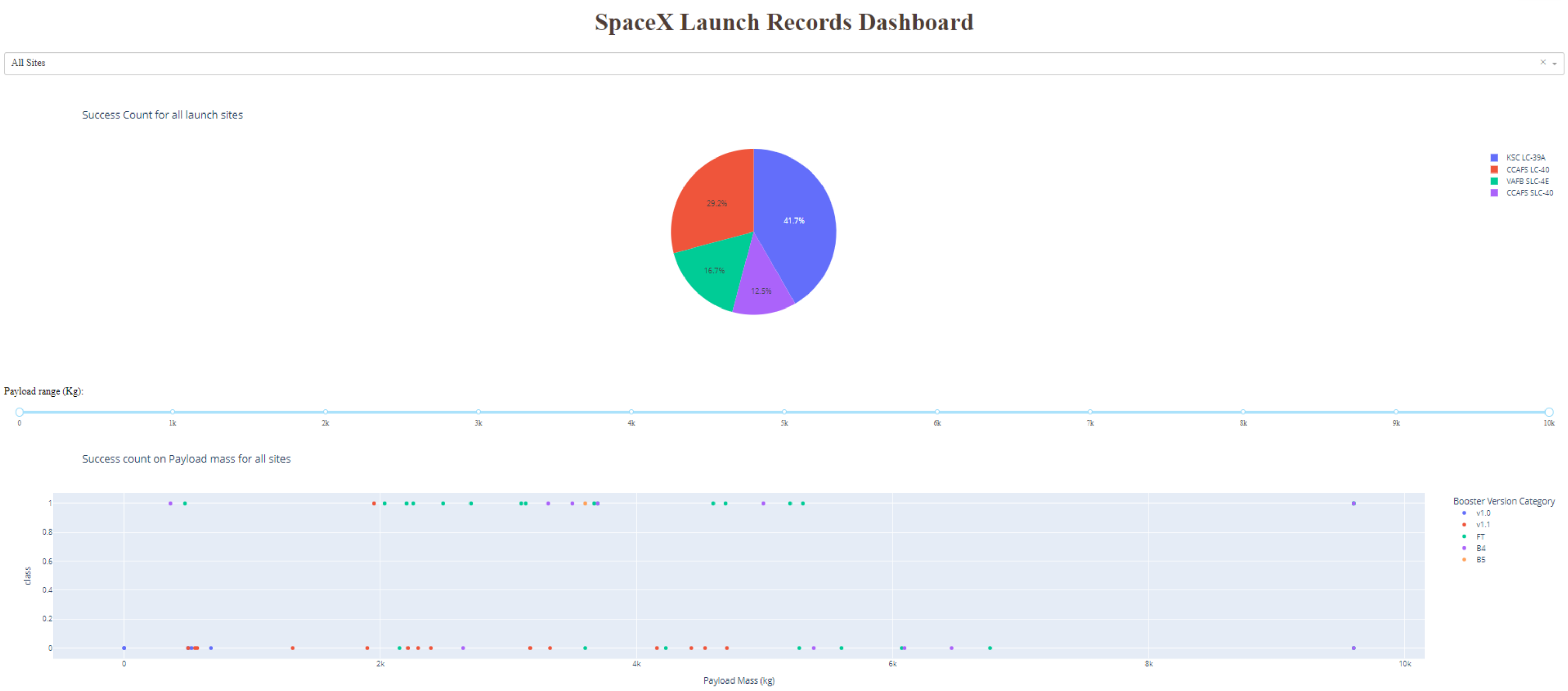
Build an Interactive Map with Folium

- Created folium map to marked all the launch sites, and created map objects such as markers, circles, lines to mark the success or failure of launches for each launch site.
- Defined a function for showing the lat and long of mouse icon.
- Created a launch set outcomes (failure=0 or success=1)
- Calculated the distance between launch sites and some landmarks like coasts, railroads etc.
- You can find the notebook here.

Build a Dashboard with Plotly Dash

- An interactive dashboard application built by Dash and Plotly
  - A pie chart using inputs from a drop-down menu for investigeting success rates.
  - A slider to chose the payload mass range for evaluating the effect of mass on success rate for different booster versions.
- Interactive dashboards are very easy to use for non-Professionals. For this reason we have built an dashboard for a comprehensive understanding.
  - A pie chart is the best for representing some rates depending on chatagories.
- You can find the codes here.

Build a Dashboard with Plotly Dash



SpaceX Launch Records Dashboard

All Sites

Success Count for all launch sites

Payload range (Kg):

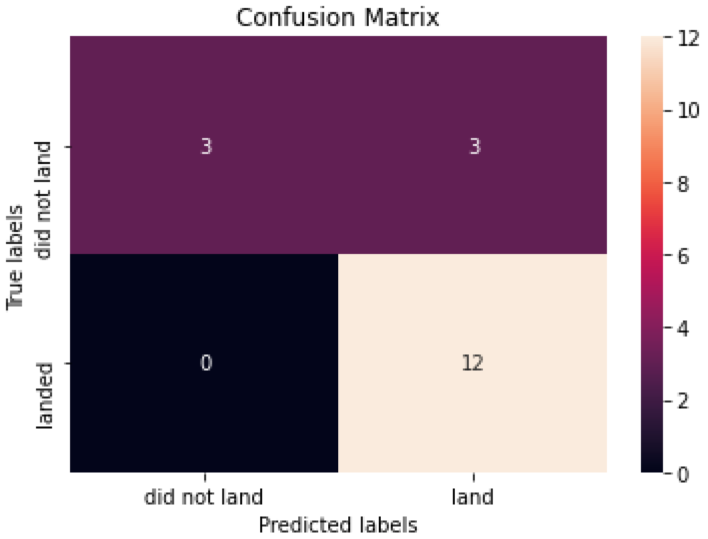
Success count on Payload mass for all sites

Predictive Analysis (Classification)

- A binary class column shows the results as 1 and 0, succesful and unsuccesful respectively. Since aim of this work to predict a launch would be successful or not a methods like linear or logistic regressions could not be applied. For this reason we applied logistic regression, desicion tree classifier an SVM.
  - First to column to be predicted which is Class column, arranged as array.
  - The other columns which represent X variable standardized with StandardScaler.
  - Then data split into x_train, x_test, y_train, y_test by using train_test_split with the test size of 0.2 and random state of 2 hyperparameters.
  - In order to find best parameters GridSearchCV used with cv=10 parameter.

Predictive Analysis (Classification)

- Best model was found for test sample as SVM with 0.833 score and best parameter were:
  - {'algorithm': 'auto', 'n_neighbors': 9, 'p': 1}
- Here is the link.



Results

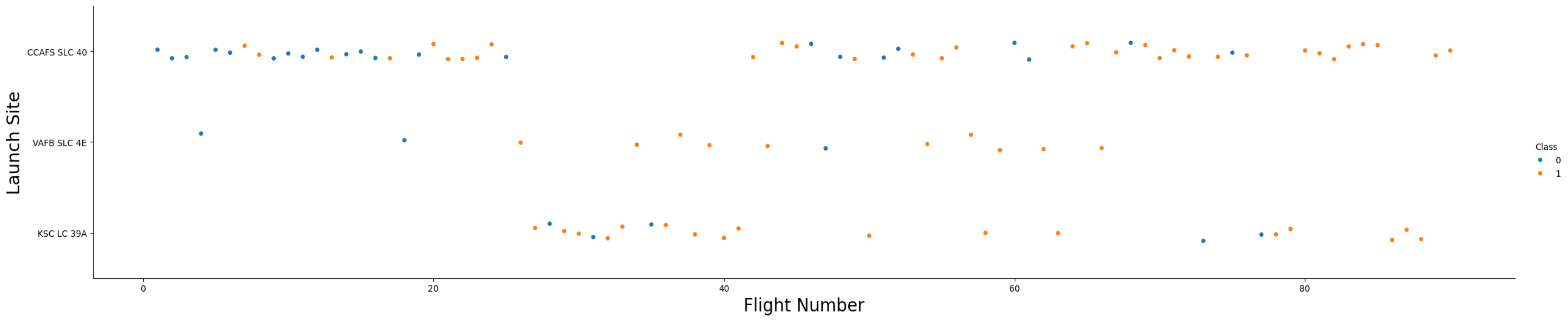
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

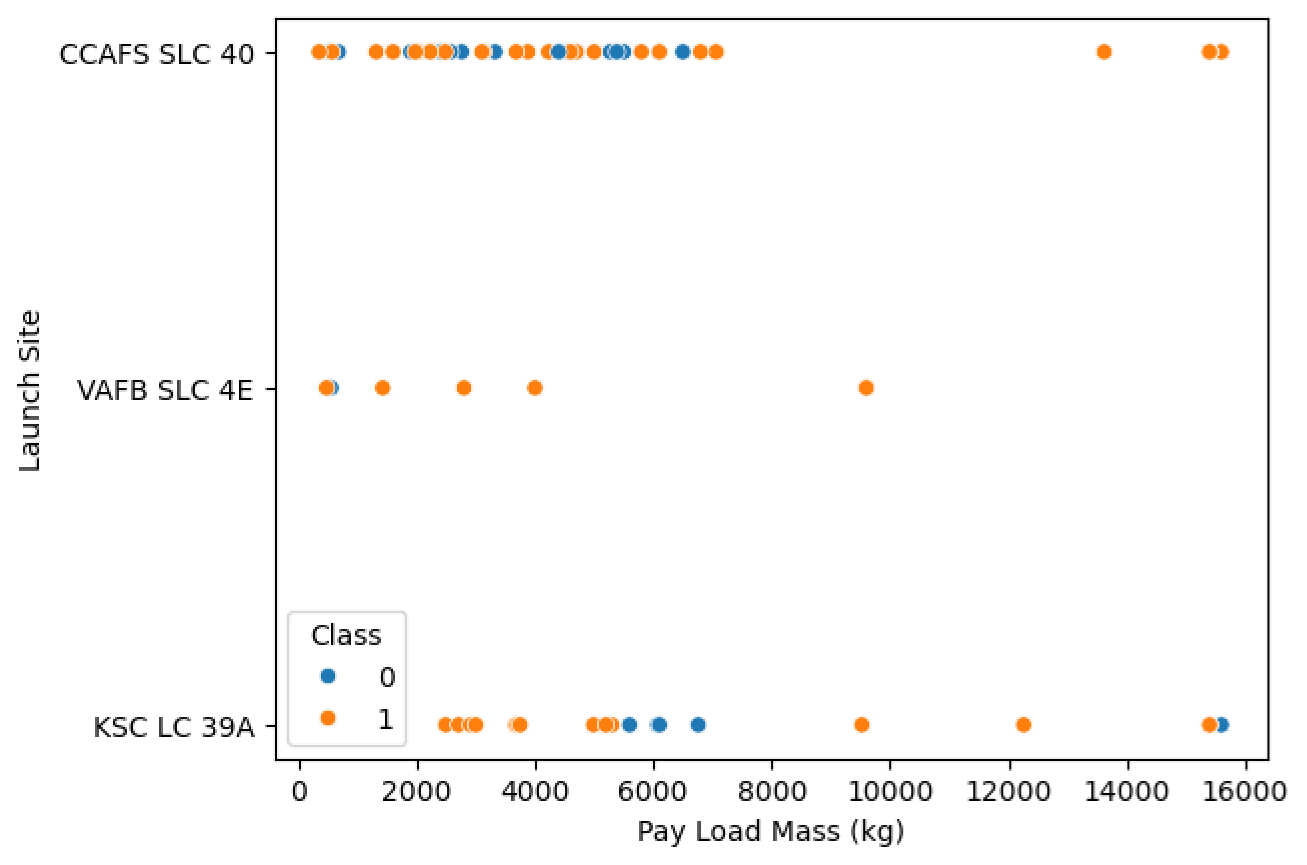
Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site



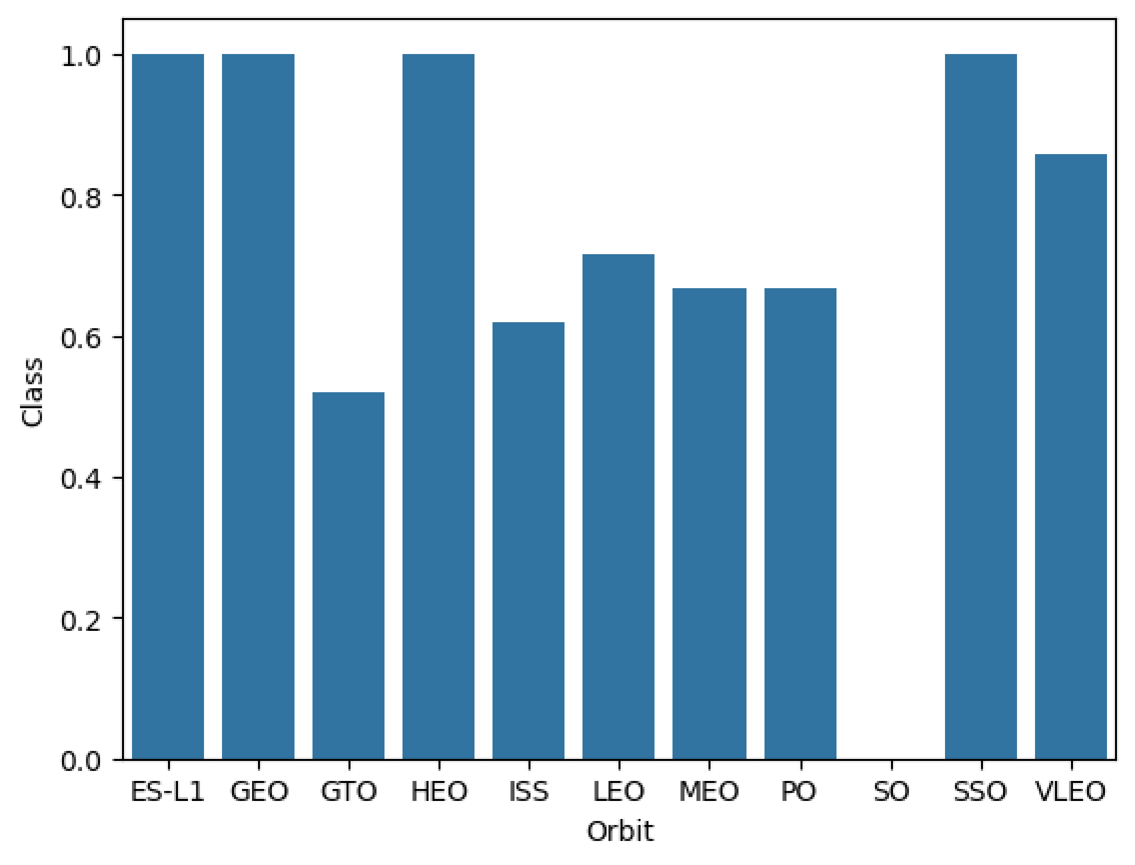
Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site



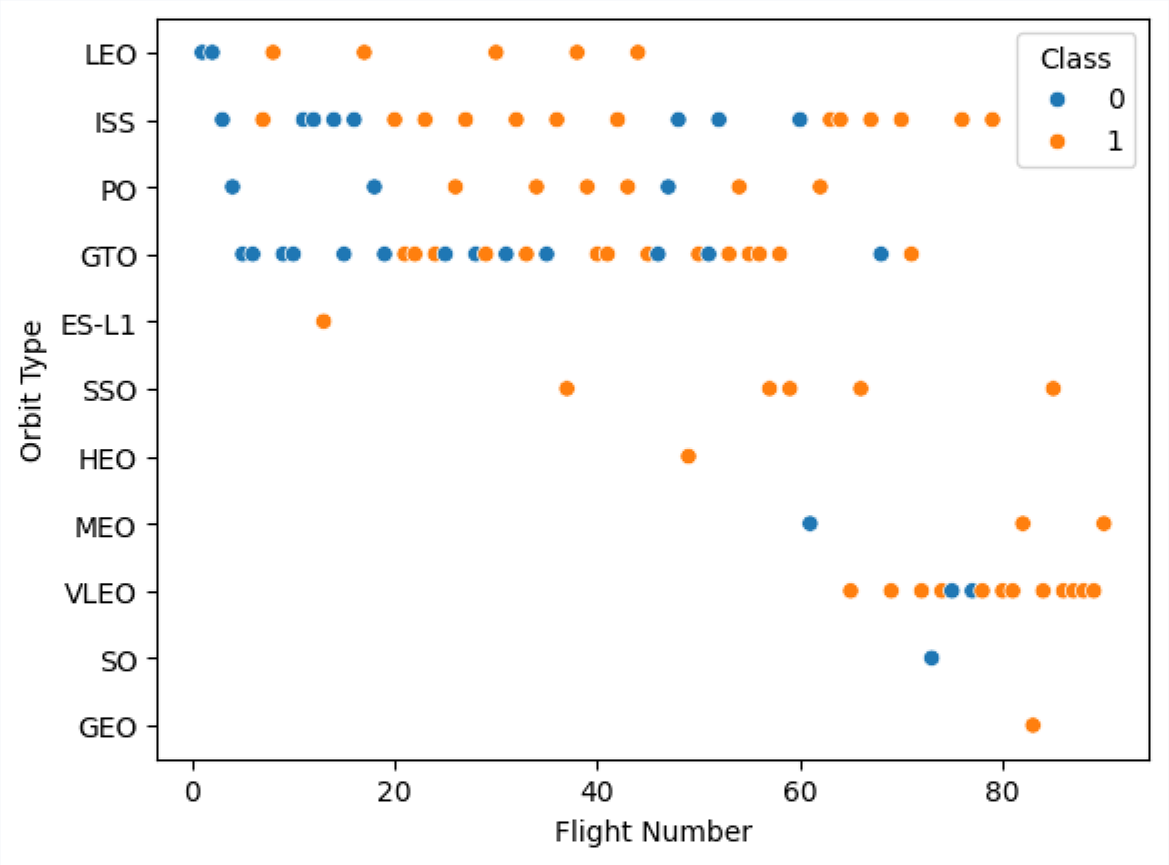
Success Rate vs. Orbit Type

- Bar chart for the success rate of each orbit type



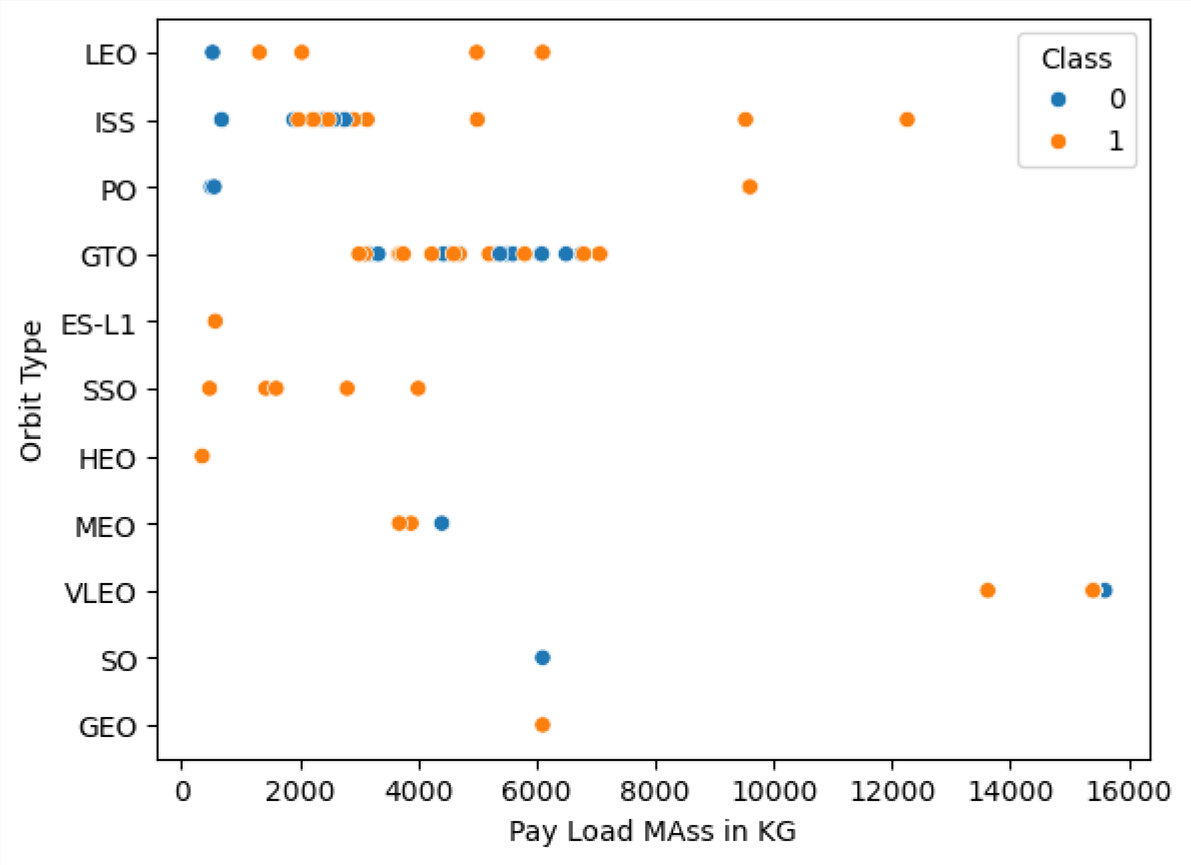
Flight Number vs. Orbit Type

- Scatter point of Flight number vs. Orbit type



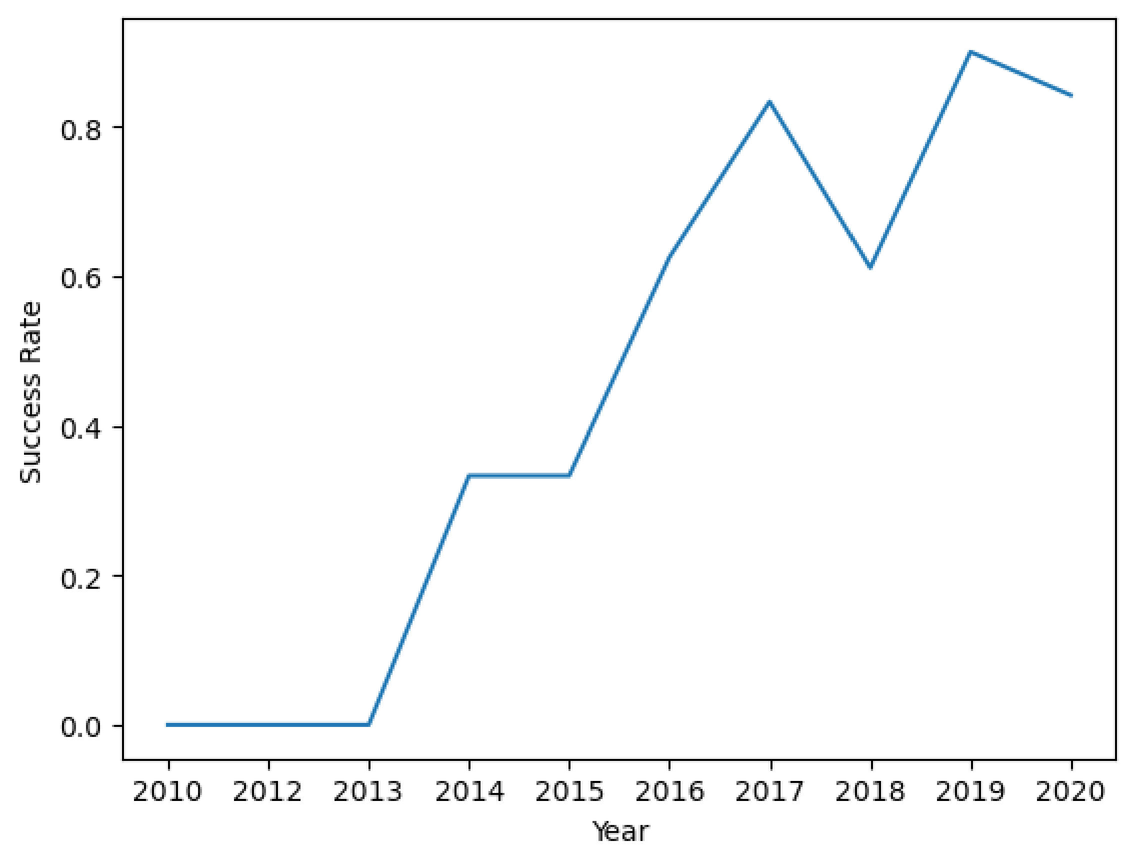
Payload vs. Orbit Type

- Scatter point of payload vs. orbit type



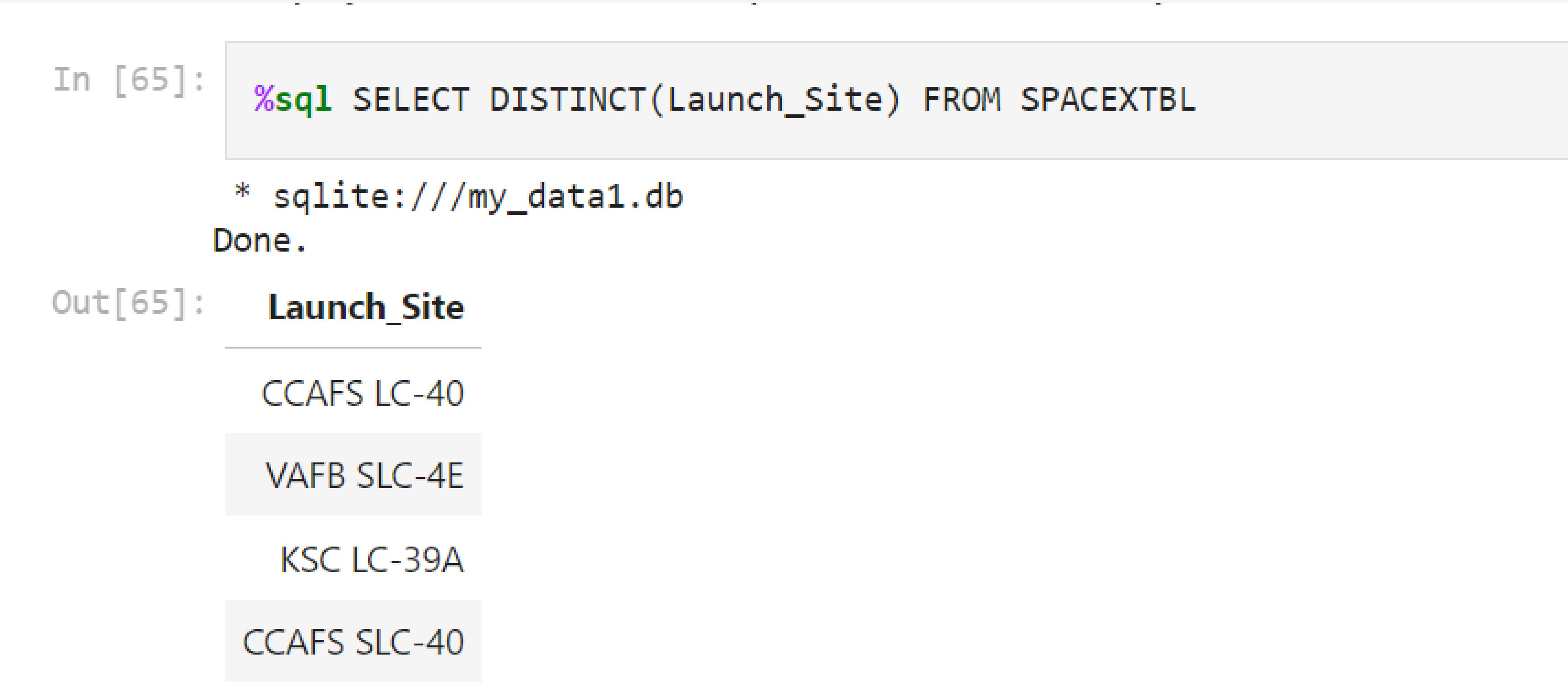
Launch Success Yearly Trend

- Line chart of yearly average success rate



All Launch Site Names

- Find the names of the unique launch sites



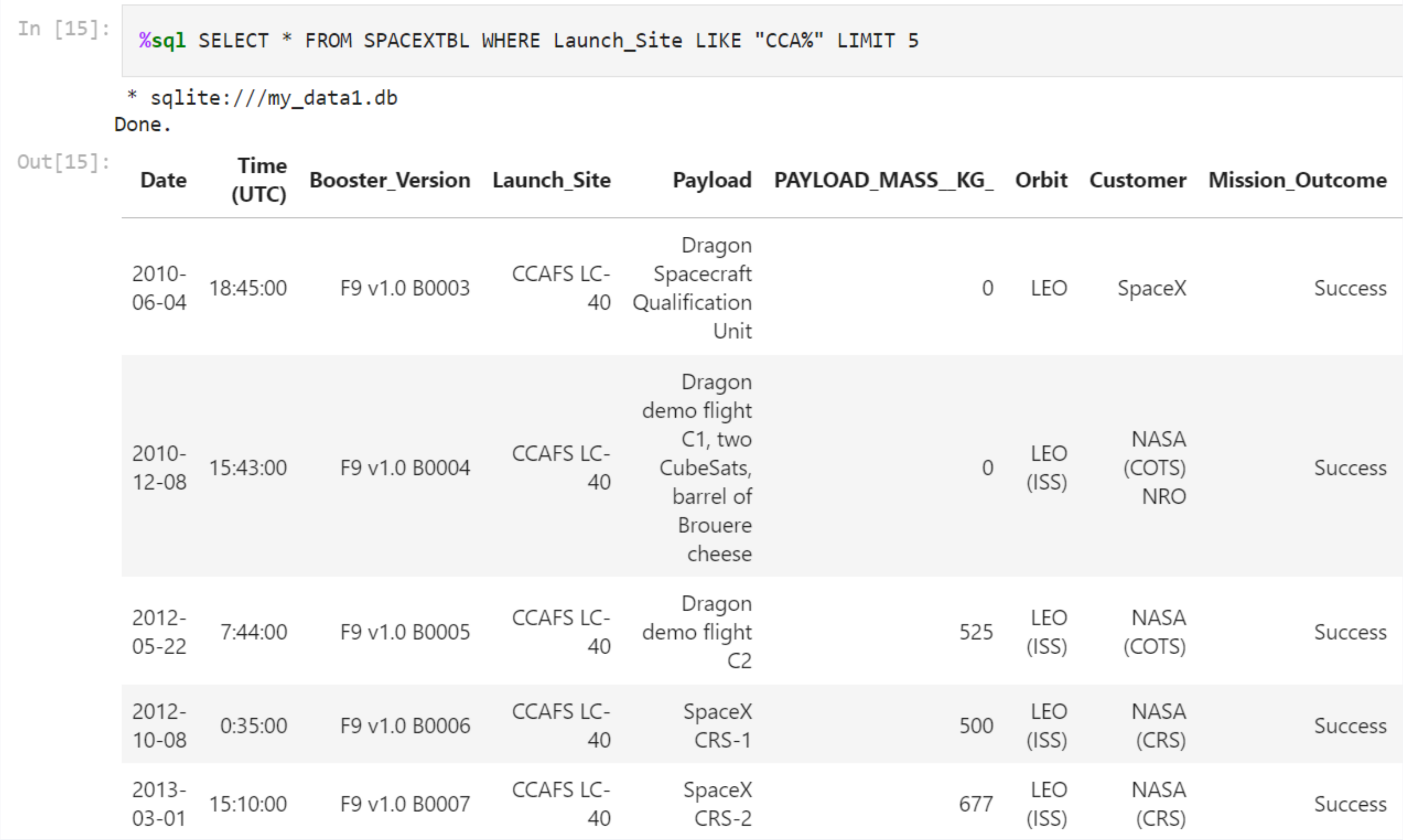
```
In [65]:  %sql SELECT DISTINCT(Launch_Site) FROM SPACEXTBL

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

Out[65]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`



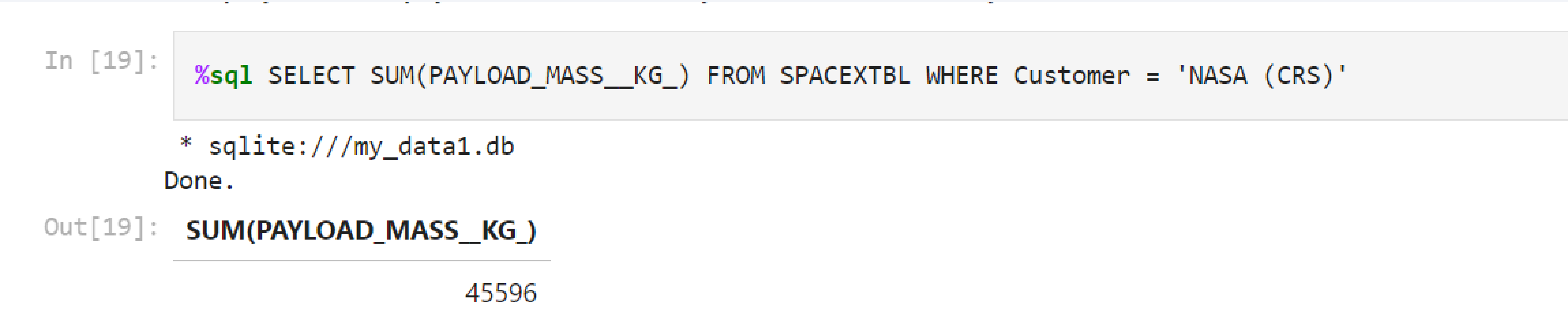
```
In [15]: %sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE "CCA%" LIMIT 5

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

Out[15]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

- Calculate the total payload carried by boosters from NASA



```
In [19]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

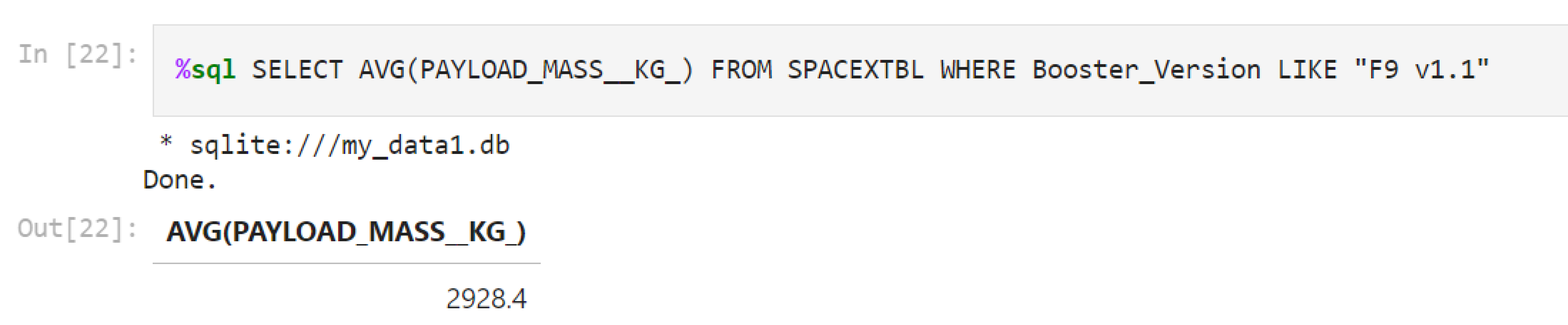
```
 * sqlite:///my_data1.db
Done.
```

Out[19]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

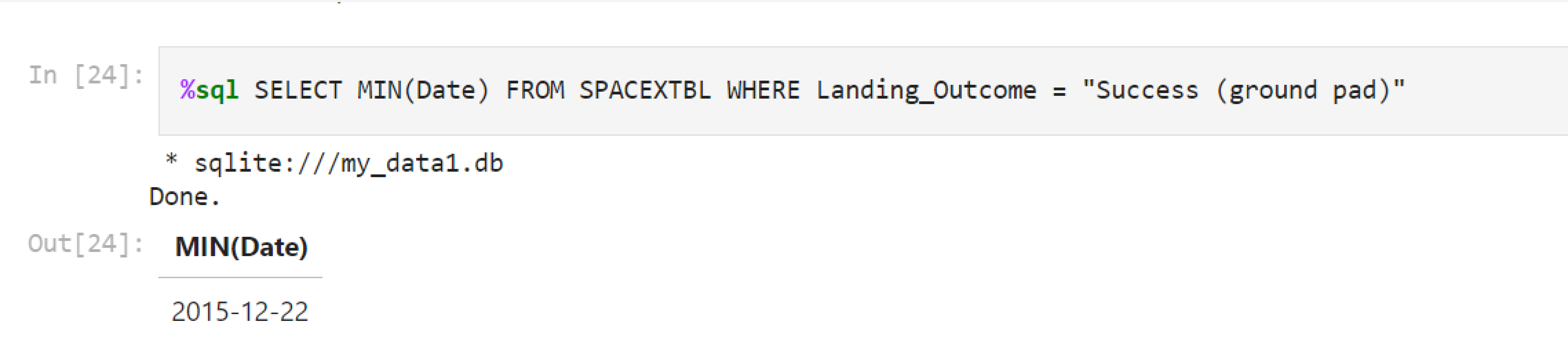
- Calculate the average payload mass carried by booster version F9 v1.1



```
In [22]:  %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE "F9 v1.1"

           * sqlite:///my_data1.db
          Done.

Out[22]:  AVG(PAYLOAD_MASS__KG_)
                            2928.4
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad



In [24]:
```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = "Success (ground pad)"
```

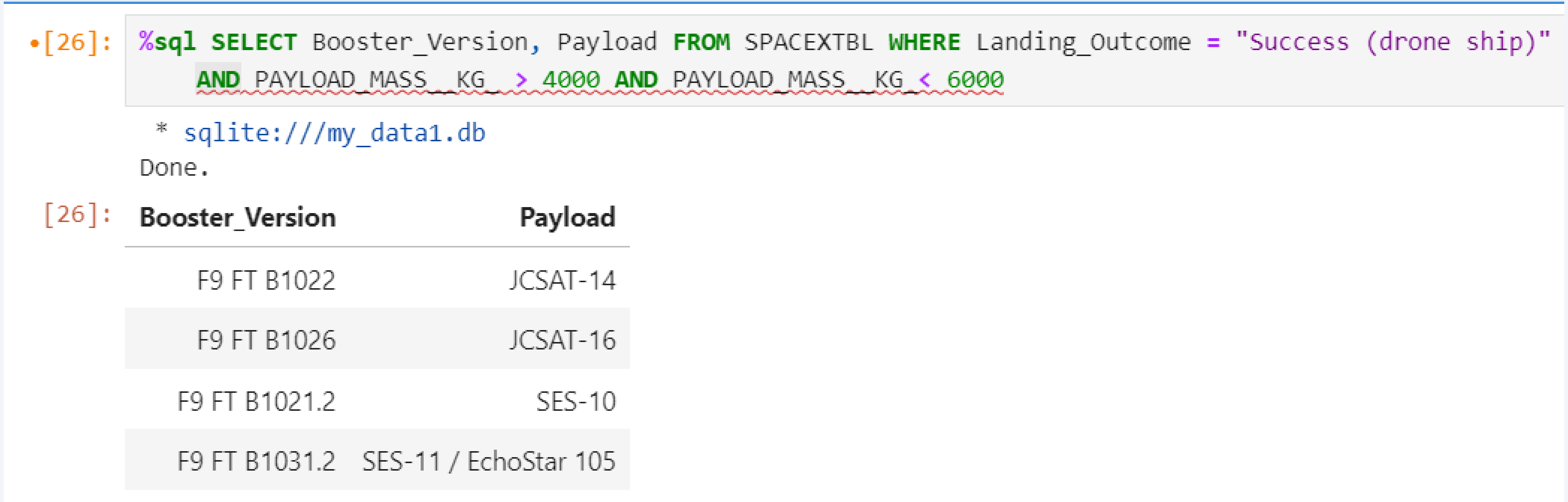
```
 * sqlite:///my_data1.db
Done.
```

Out[24]:

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



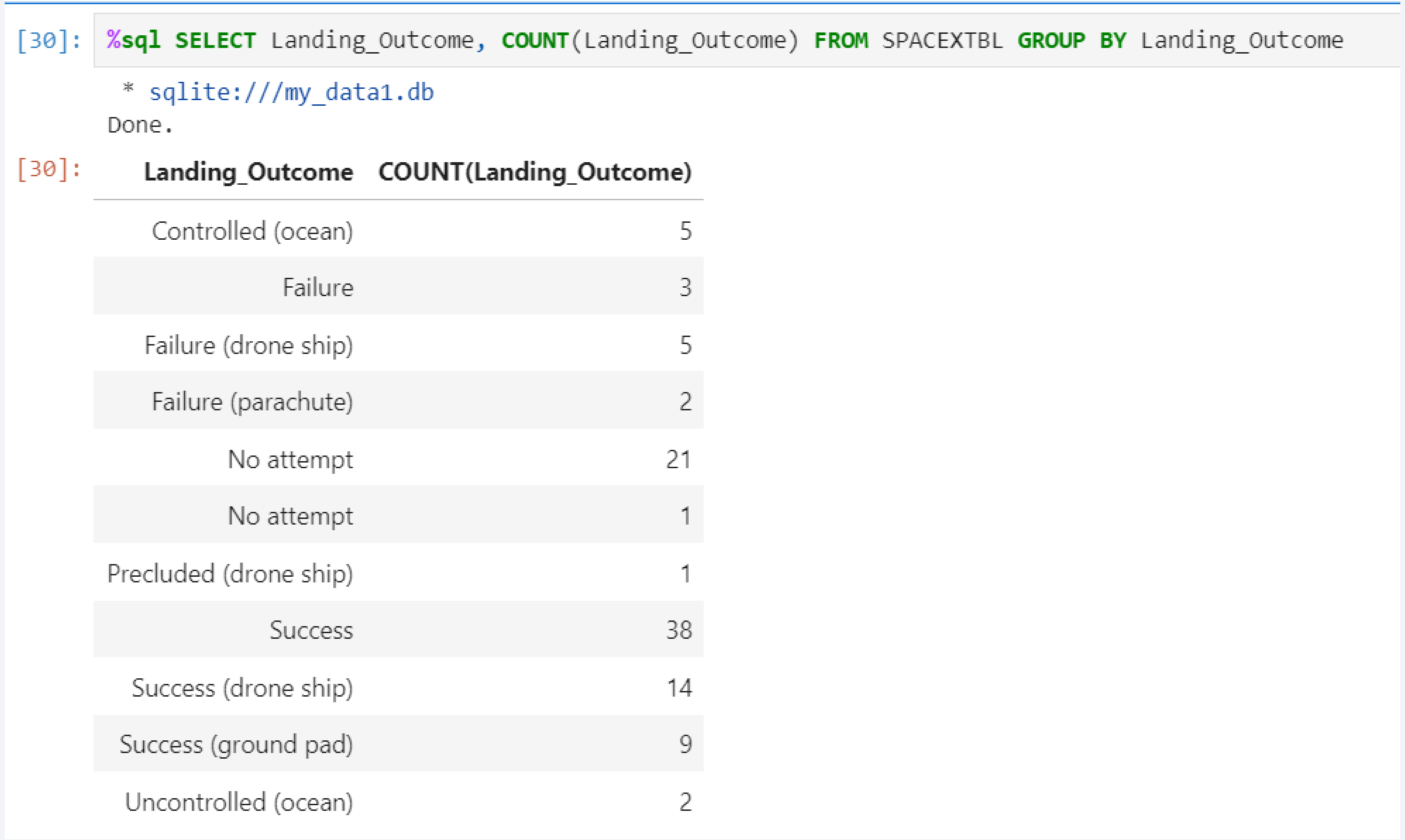
```
•[26]: %sql SELECT Booster_Version, Payload FROM SPACEXTBL WHERE Landing_Outcome = "Success (drone ship)"
           AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000

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

[26]:

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes



```
[30]: %sql SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTBL GROUP BY Landing_Outcome
```

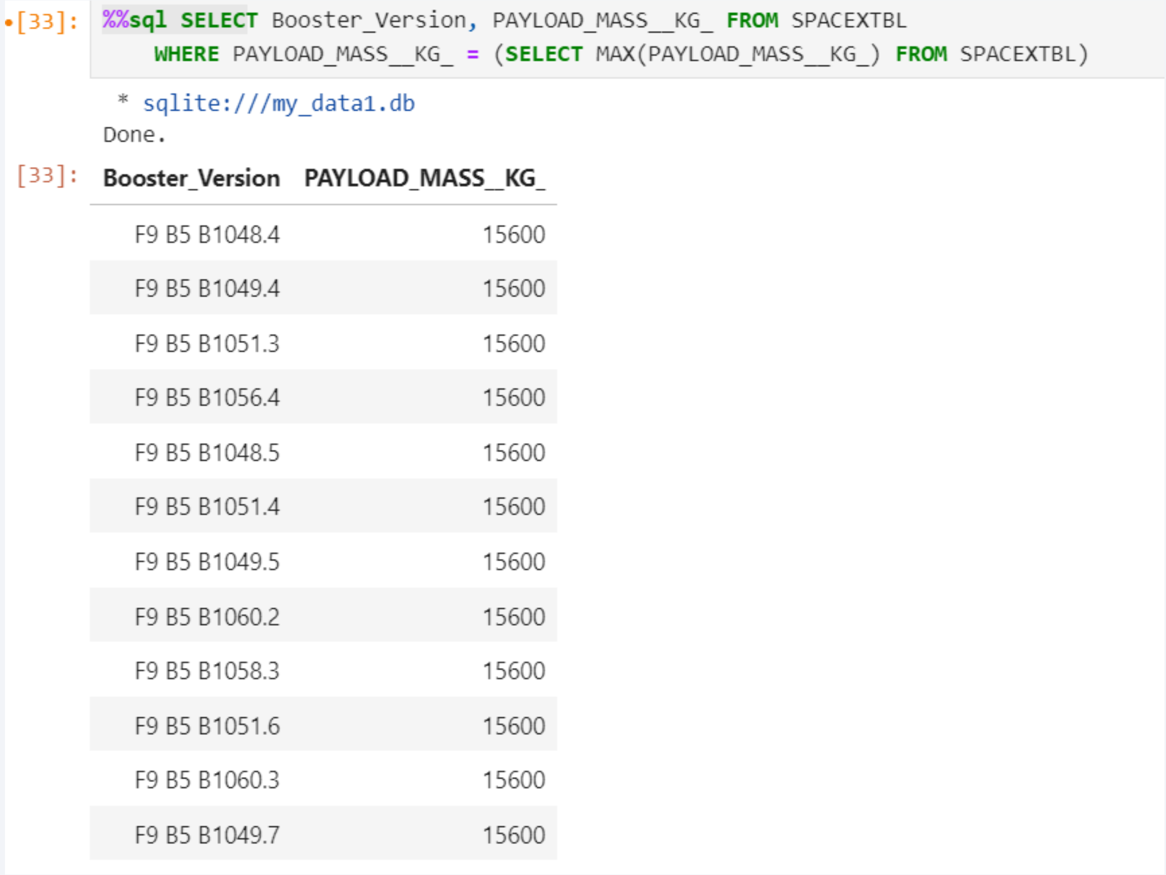
```
 * sqlite:///my_data1.db
Done.
```

[30]:

Landing_Outcome	COUNT(Landing_Outcome)
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass



```
[33]: %%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL
          WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

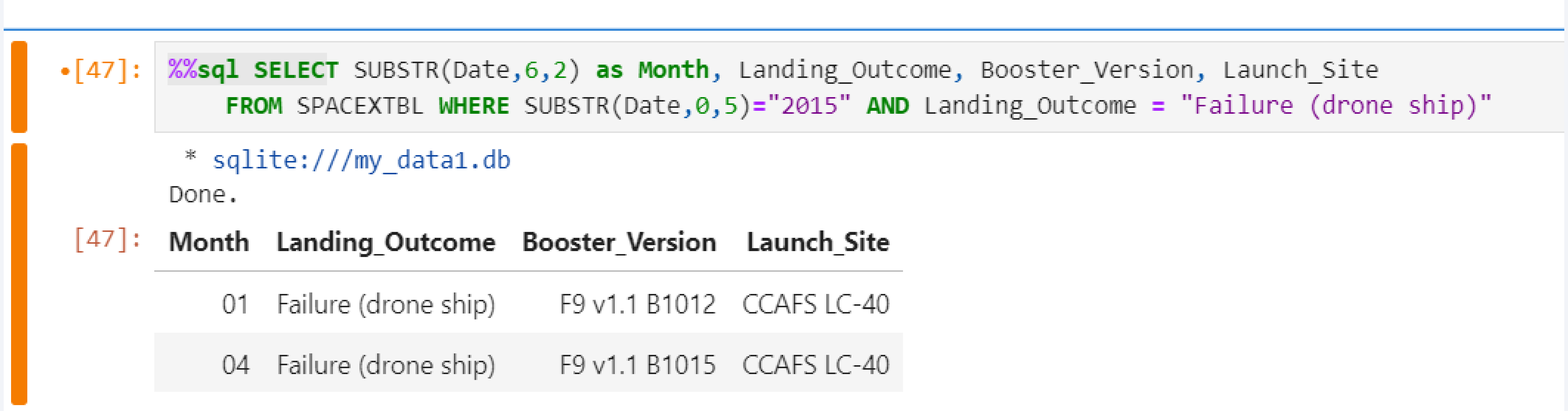
```
 * sqlite:///my_data1.db
Done.
```

[33]:

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015



```
[47]: %%sql SELECT SUBSTR(Date,6,2) as Month, Landing_Outcome, Booster_Version, Launch_Site
          FROM SPACEXTBL WHERE SUBSTR(Date,0,5)="2015" AND Landing_Outcome = "Failure (drone ship)"
```

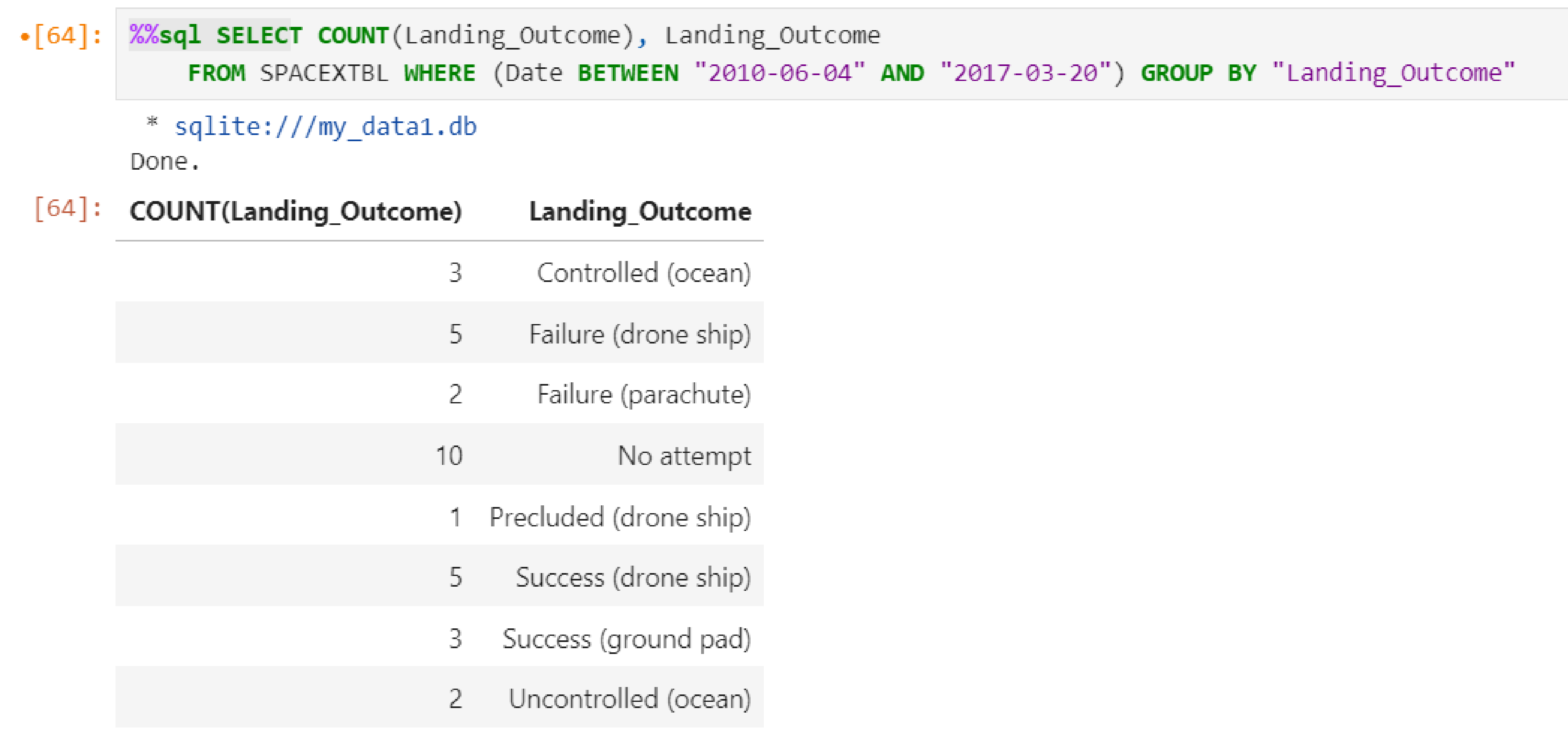
```
 * sqlite:///my_data1.db
Done.
```

[47]:

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

- 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



•[64]:

```
%%sql SELECT COUNT(Landing_Outcome), Landing_Outcome
    FROM SPACEXTBL WHERE (Date BETWEEN "2010-06-04" AND "2017-03-20") GROUP BY "Landing_Outcome"
```

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

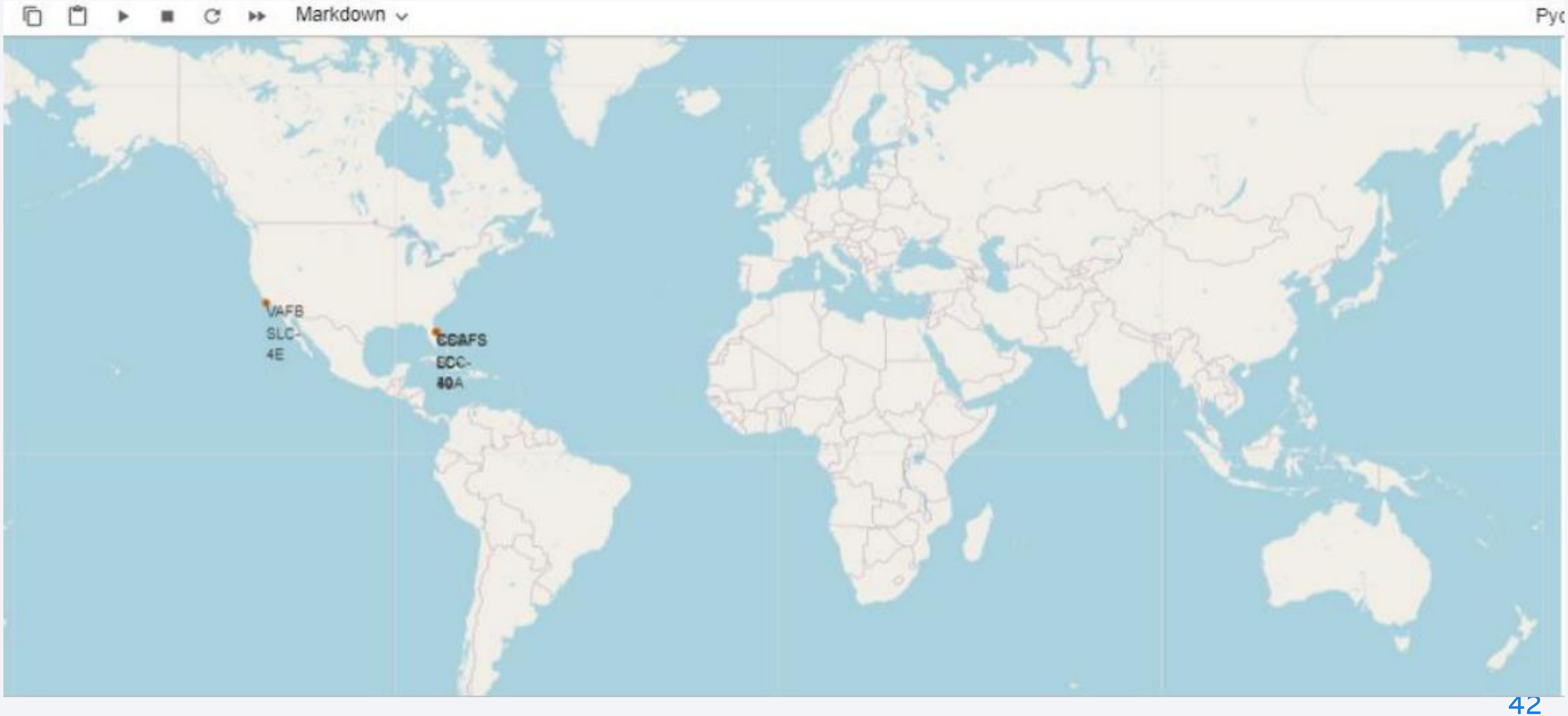
[64]:

COUNT(Landing_Outcome)	Landing_Outcome
3	Controlled (ocean)
5	Failure (drone ship)
2	Failure (parachute)
10	No attempt
1	Precluded (drone ship)
5	Success (drone ship)
3	Success (ground pad)
2	Uncontrolled (ocean)

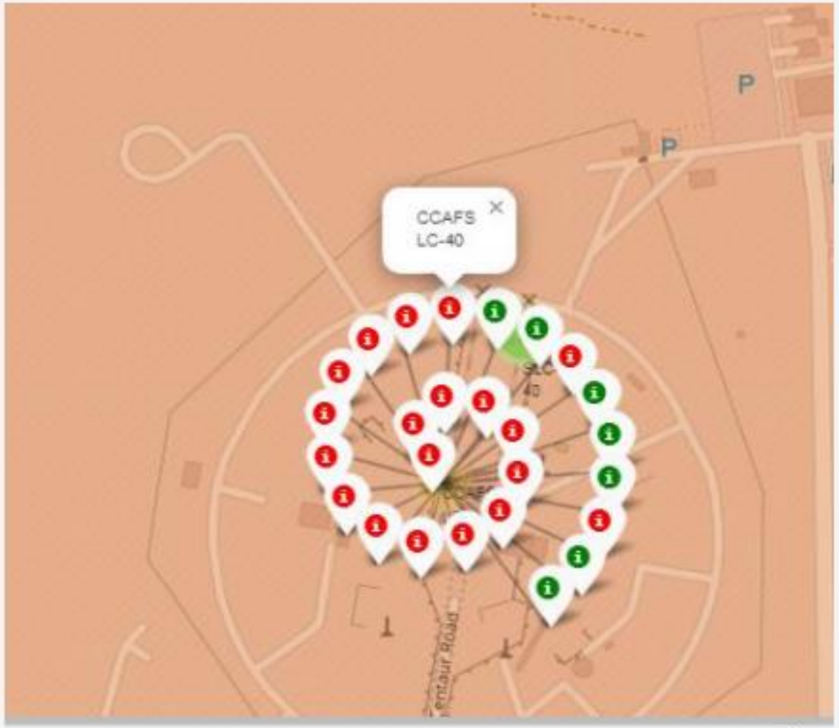
Section 3

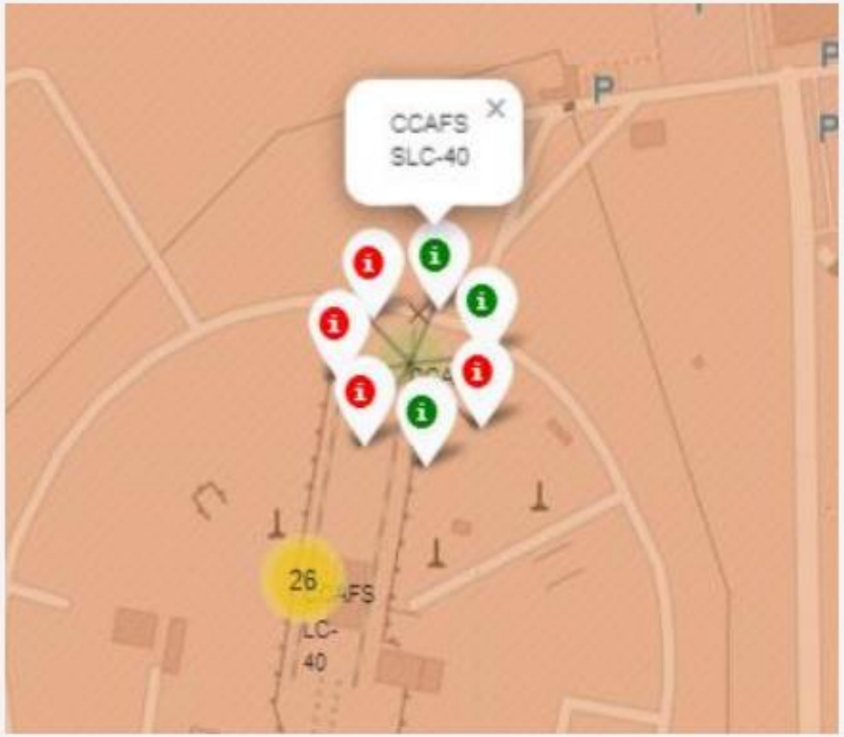
Launch Sites Proximities Analysis

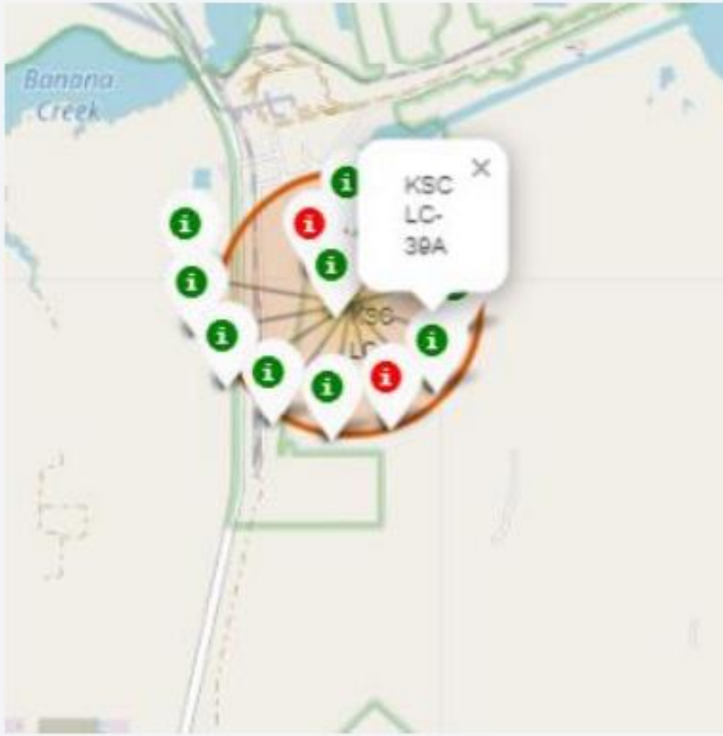
Markers of all launc sites



Launch Outcomes For Each City

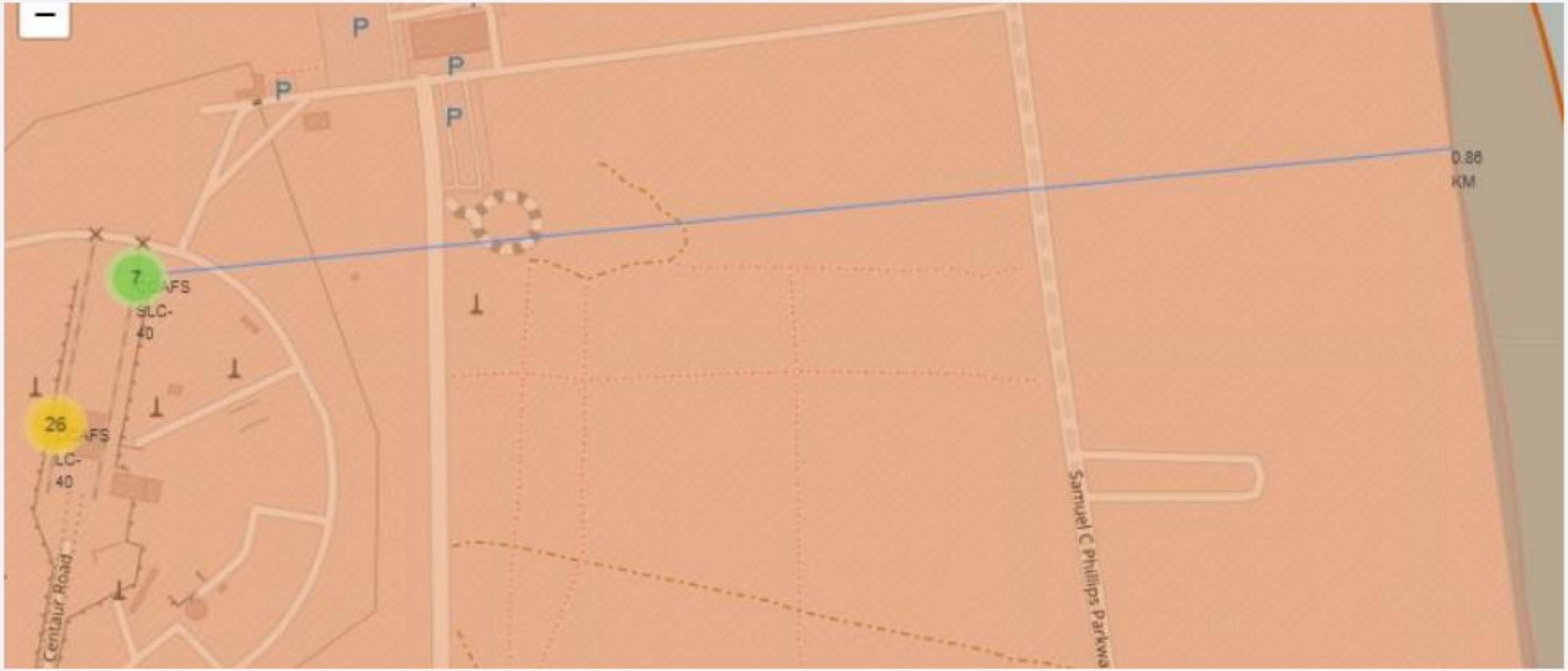






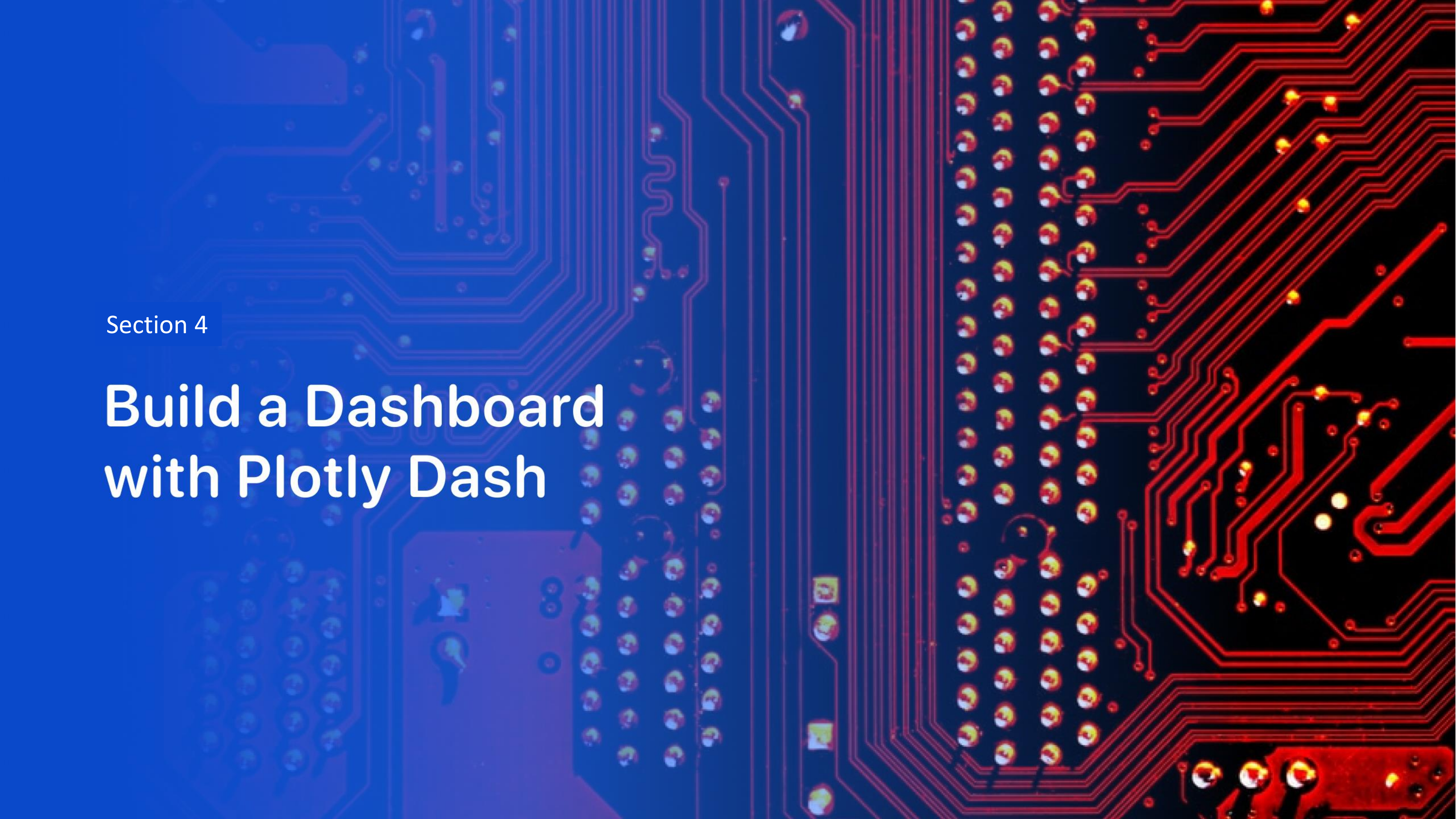
- Greens represent succesfull, reds represend unsuccessfull landings

Distance between launch sites and its proximities

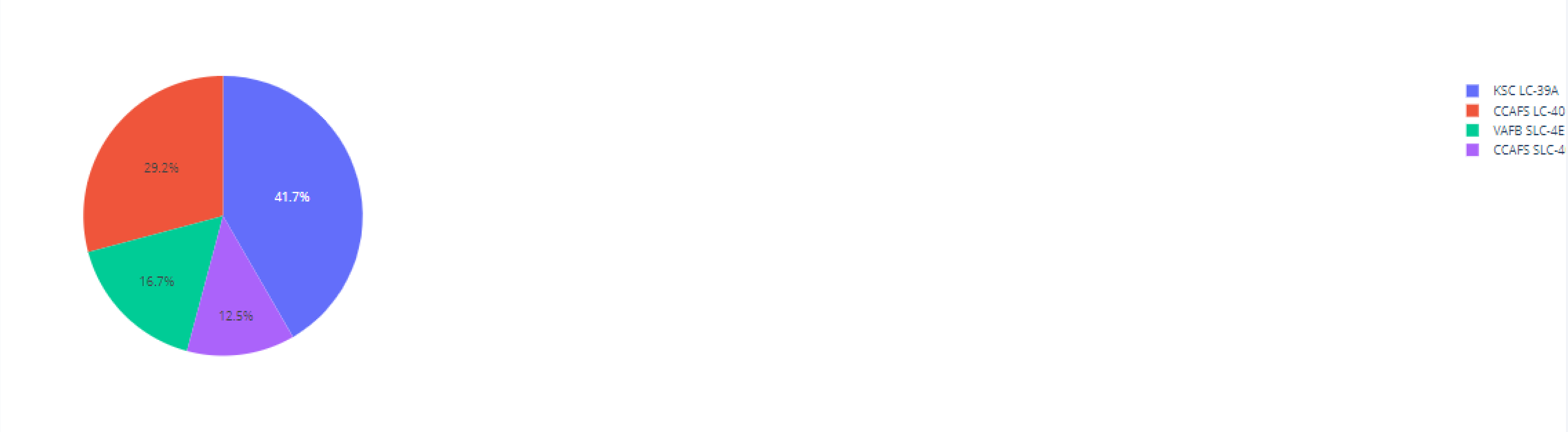


Section 4

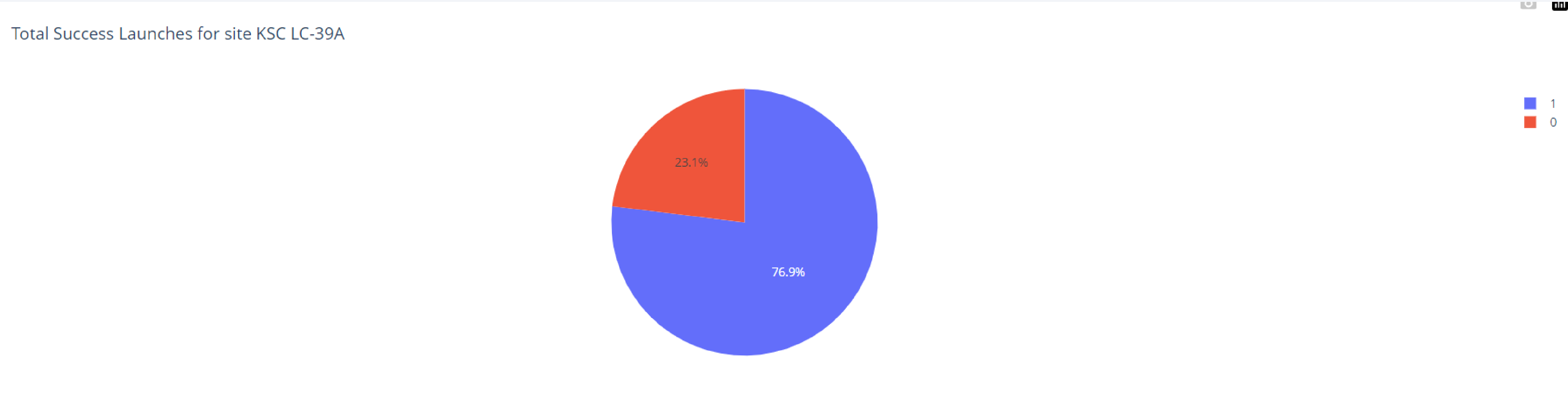
Build a Dashboard with Plotly Dash



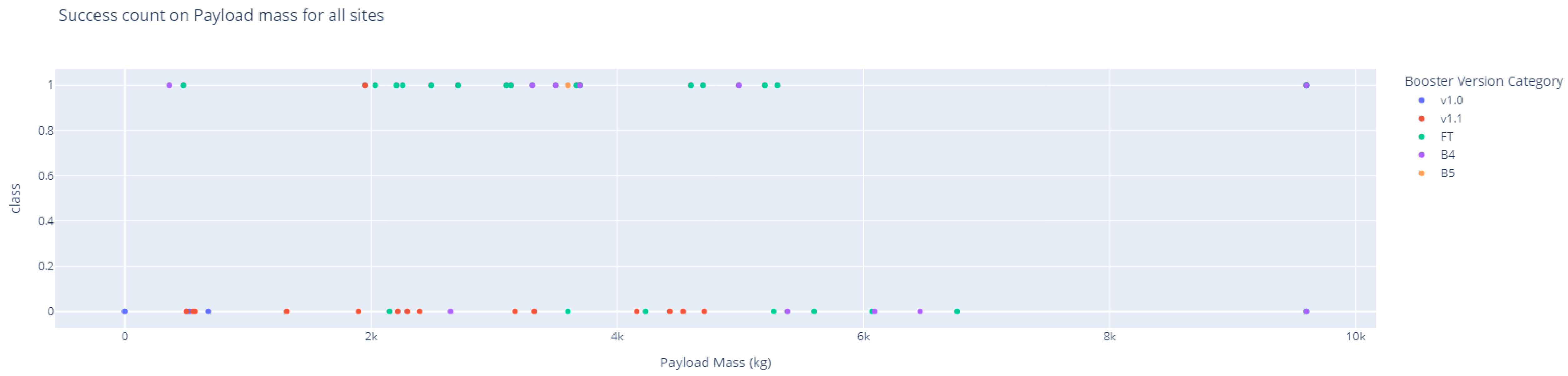
Success Rate of All Sites



Most Successful Launches are From KSC LC-39A



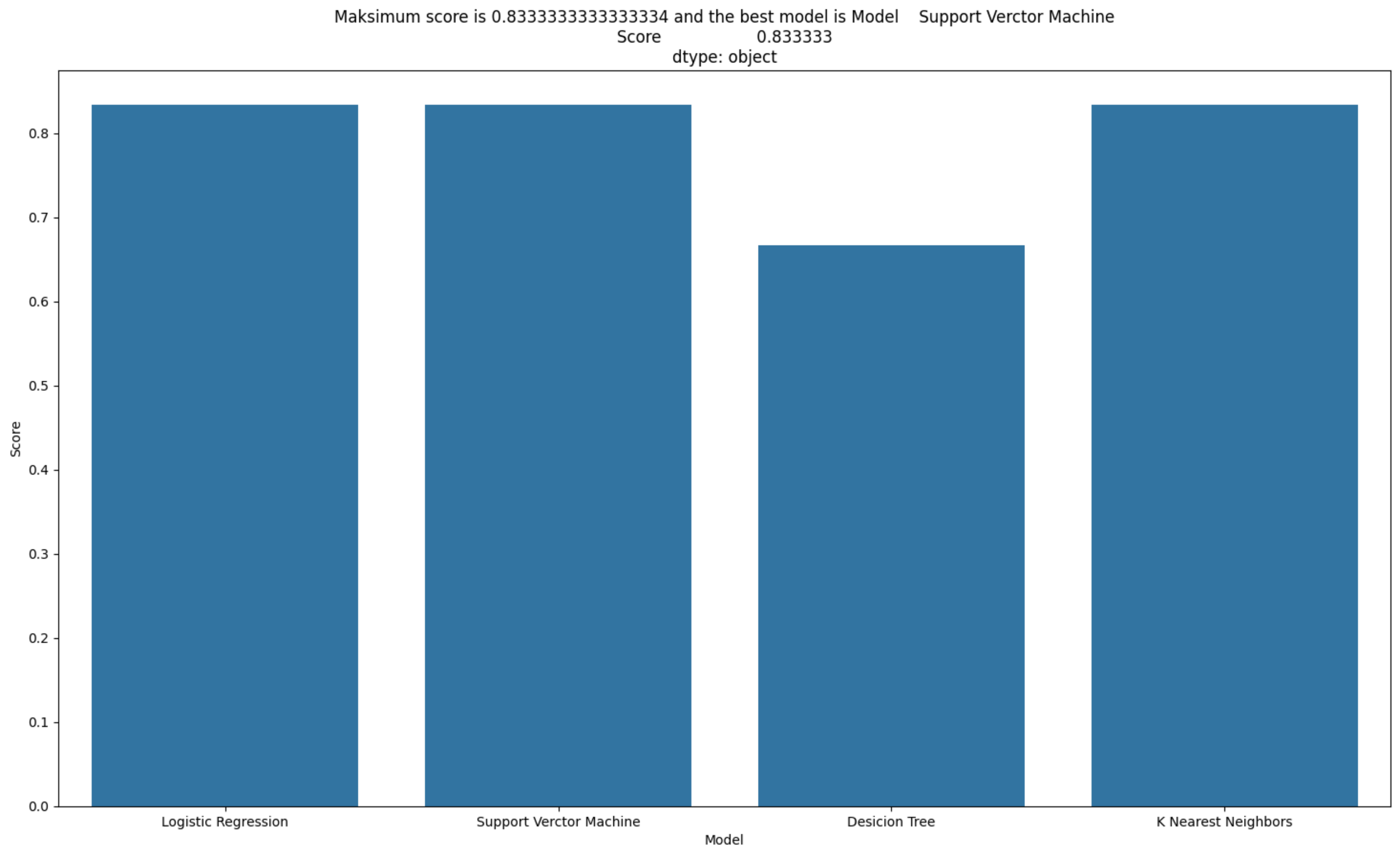
Success Count Oon Payload Mass for All Sites



Section 5

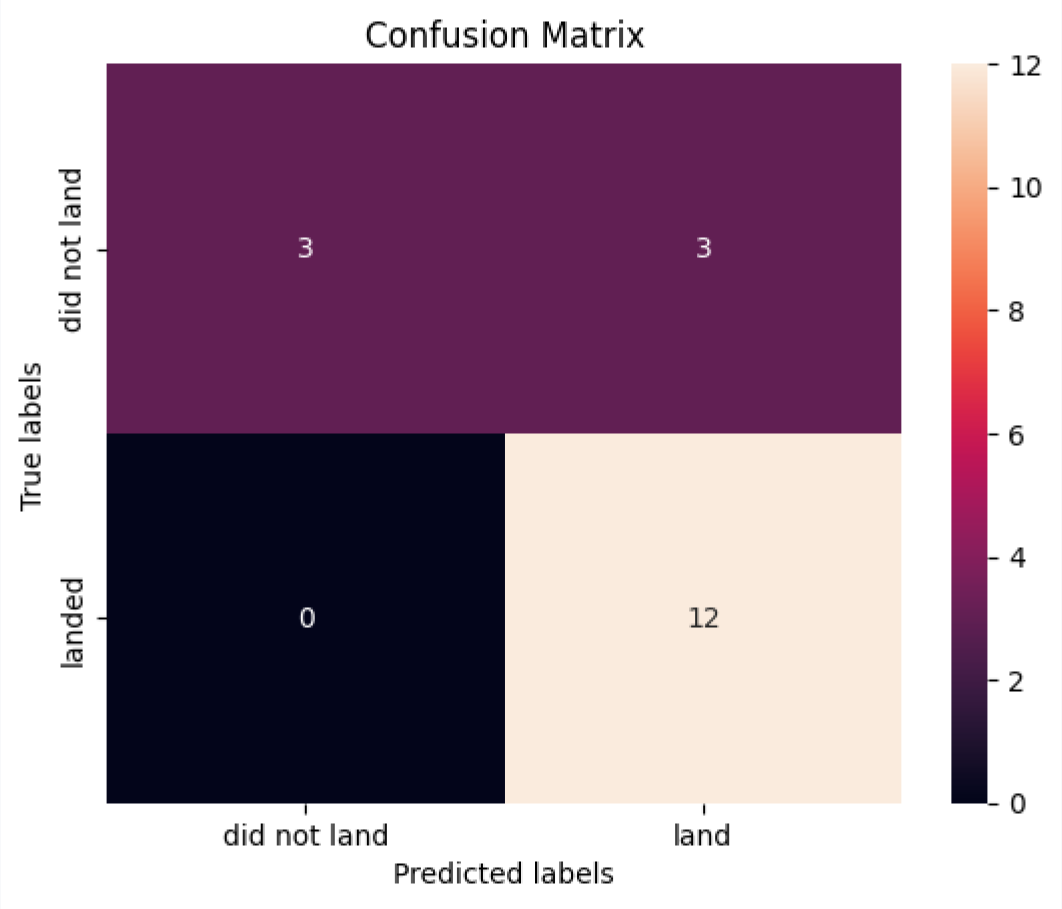
Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix of SVM

- As you can see in SVM confusion matrix true true predictions are all accurate but there are 3 true false predictions.



Conclusions

- Different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E having success rate of 77%.
- We can conclude that, as the flight number increases in each of the 3 launch sites, success rate also increases. If you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).
- Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%.
- 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

Thank you!