IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Nguyen Pham Ha Phuong>
<04/04/2024>

Outline

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

Executive Summary

- Summary of methodologies
  - SpaceX Data Collection using SpaceX API
  - SpaceX Data Collection with web scrapping
  - SpaceX Data Wrangling
  - SpaceX Exploratory Data Analysis using SQL
  - SpaceX EDA DataViz Using Python Pandas and Matplotlib
  - SpaceX Lauch Sites Analysis with Folium-Interactive Visual Analytics and Plotly Dash
  - SpaceX Machine Learning Landing Prediction
- Summary of all results
  - EDA results
  - Interactive Visual Analytics and Dashboards
  - Predictive Analysis (Classification)

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

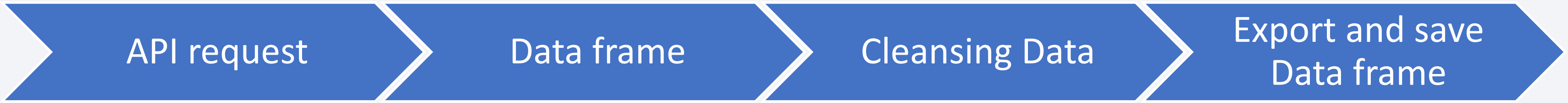
Methodology

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

- After a throughout research about SpaceX, I decided to use their **publicly free API using HTTP python library** to get the data in Json format. And then, I converted this result into a data frame to extract all useful information for my analysis.
- I filtered all records for “Falcon 1” to **keep only the Falcon 9 launches in my dataset**. I replaced all missing values (by mean value or mode value) at this early stage, leaving only LaunchingPad with nulls indicating when no LaunchPad was used during the launching process.
- Also performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page: Using BeautifulSoup and request libraries, I extracted HTML table of Falcon 9 launchs records from this page. Parsed the table and converted it into Pandas data frame table



API request

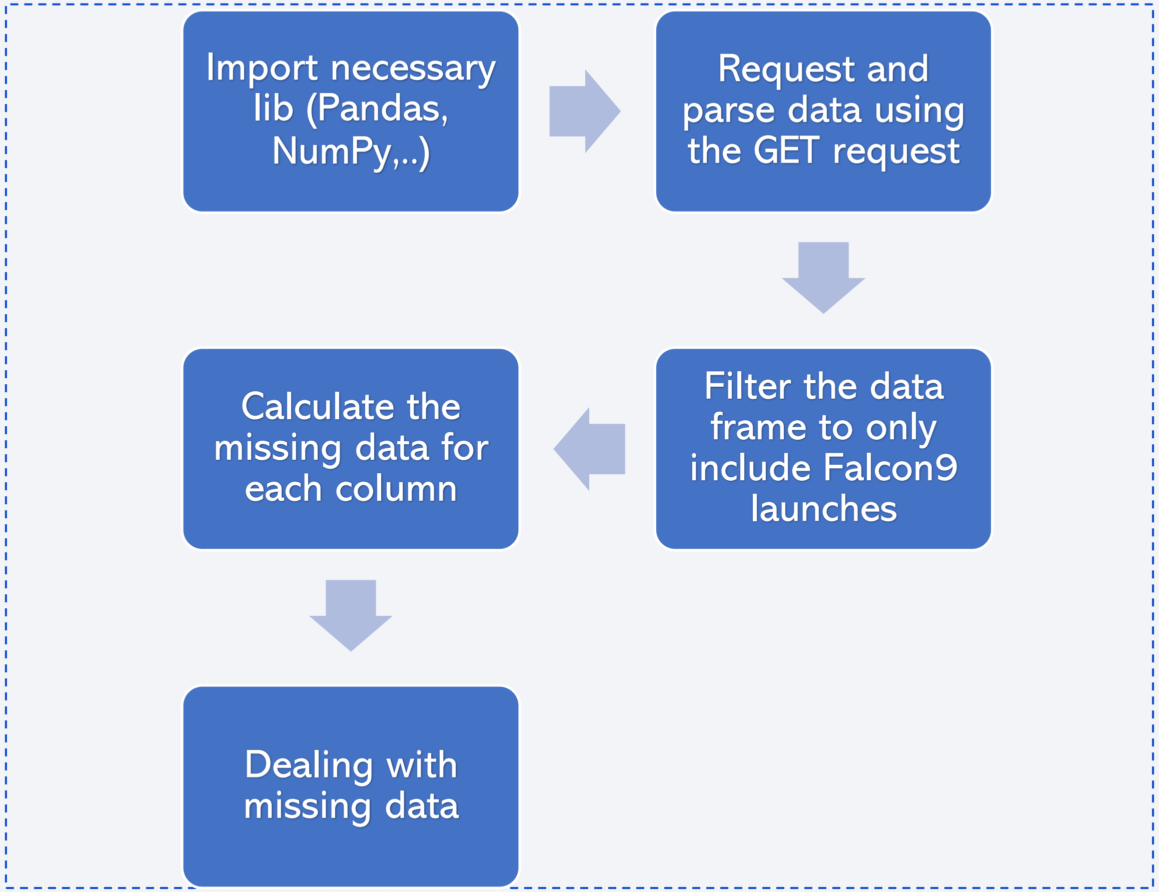
Data frame

Cleansing Data

Export and save Data frame

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean data, did some basic data wrangling and formatting
- Github Space X API calls notebook


Import necessary lib (Pandas, NumPy,..)

Request and parse data using the GET request

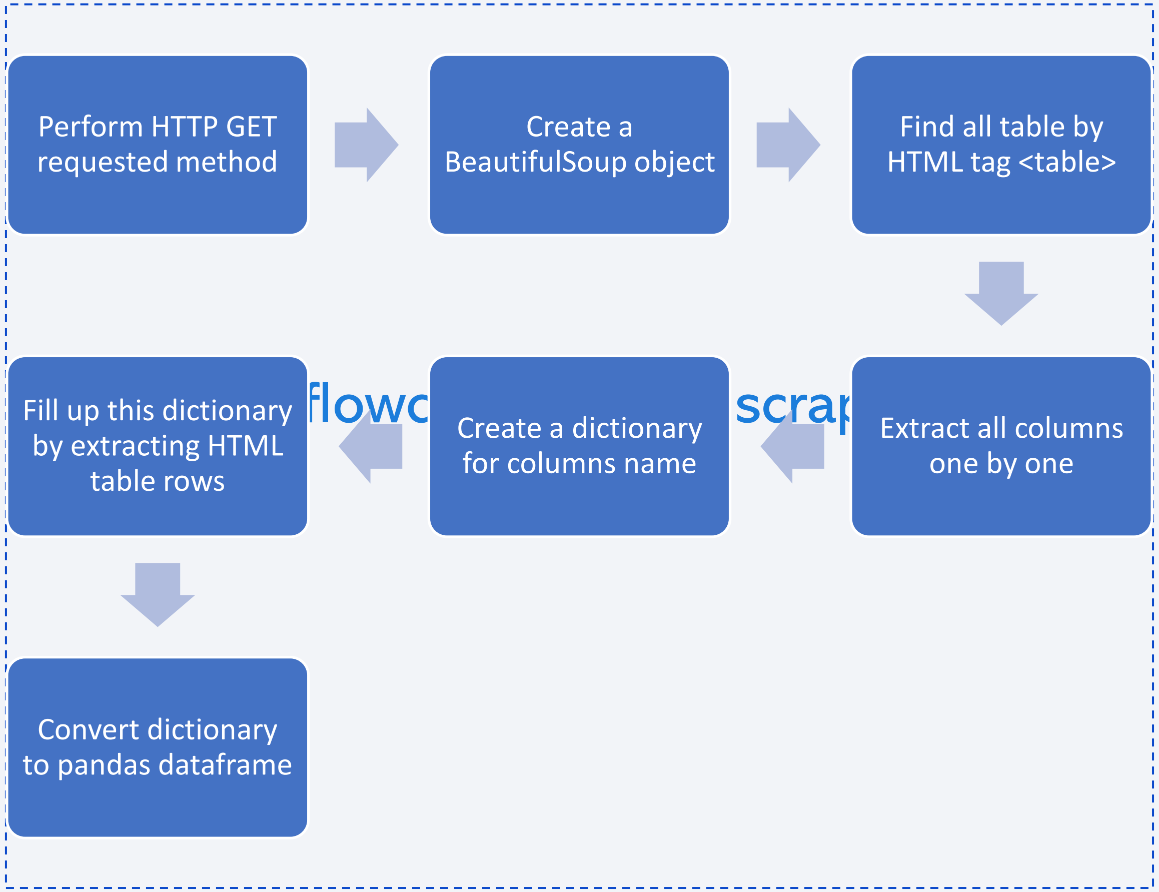
Filter the data frame to only include Falcon9 launches

Calculate the missing data for each column

Dealing with missing data

Data Collection - Scraping

- Applied web scrapping to webscrap Falcon9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe
- SpaceX Webscraping



Perform HTTP GET requested method

Create a BeautifulSoup object

Find all table by HTML tag <table>

Extract all columns one by one

Create a dictionary for columns name

Fill up this dictionary by extracting HTML table rows

Convert dictionary to pandas dataframe

Data Wrangling

- Training label
  - The outcome field details two components: ‘mission outcome’ and ‘landing location’
  - We want to create a training label ‘Class’ to indicate successful landing = 1; unsuccessful landing = 0
  - Value mapping:
    - Outcomes ‘True ASDS’, ‘True RTLS’, & ‘True Ocean’ – set Class to -> 1
    - Outcomes ‘None None’, ‘False ASDS’, ‘None ASDS’, ‘False Ocean’, ‘False RTLS’ – set Class to -> 0
- SpaceX Data Wrangling

EDA with Data Visualization

- Goal
  - Exploratory Data Analysis carried out on the variables ‘Flight Number’, ‘Payload Mass’, ‘Launch Site’, ‘Orbit’, ‘Class’ and ‘Year’, to investigate relationships between variables. ▪
- Charts plotted
  - Scatter charts: Flight Number VS. Payload Mass, Flight Number VS. Launch Site, Payload VS. Launch Site, Orbit VS. Flight Number, Payload VS. Orbit Type, Orbit VS. Payload Mass
  - Bar charts: Mean VS. Orbit
  - Line charts: Success Rate VS. Year
- EDA with data visualization notebook

EDA with SQL

- Goal
  - To better understand the data, the dataset is loaded into IBM DB2 Database and queried using SQL magic in python.
- Queries
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string ‘CAA’
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date where the successful landing outcome in drone ship was achieved
  - Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster versions which have carried the maximum payload mass
  - Listing the records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch site for the months in year 2015
  - Ranking the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order
- EDA with SQL notebook

Build an Interactive Map with Folium

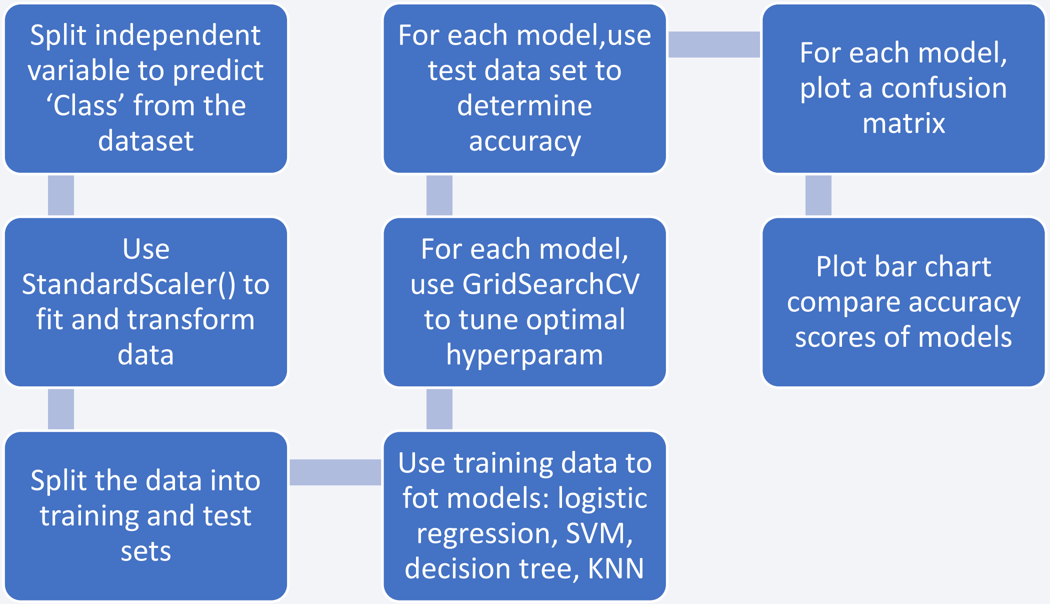
- Folium maps visualizes the launch data onto an interactive map. Using the latitude and longitude coordinates of each launch site, we added labelled circle markers at each launch site. Using MarkerCluster(), we indicate successful outcomes with green markers, and unsuccessful outcomes with red markers. We can calculate the distance to key locations on the map and mark a line on the map to visualize this. E.g. distance to nearest railway, highway, coast, city.
- Interactive map with Folium map notebook

Build a Dashboard with Plotly Dash

- Dashboard includes a pie chart and a scatter plot.
- Interactive pie chart used to visualise launch site success rate; showing distribution of successful landings across all launch sites or distribution of successful landings for specific individual launch site.
- Scatter plot used to visualise how success varies dependent on payload mass and booster version category.
- Plotly Dash lab

Predictive Analysis (Classification)

- Flowchart



Split independent variable to predict ‘Class’ from the dataset

Use StandardScaler() to fit and transform data

Split the data into training and test sets

Use training data to fot models: logistic regression, SVM, decision tree, KNN

For each model, use GridSearchCV to tune optimal hyperparam

For each model,use test data set to determine accuracy

For each model, plot a confusion matrix

Plot bar chart compare accuracy scores of models

- Predictive analysis lab

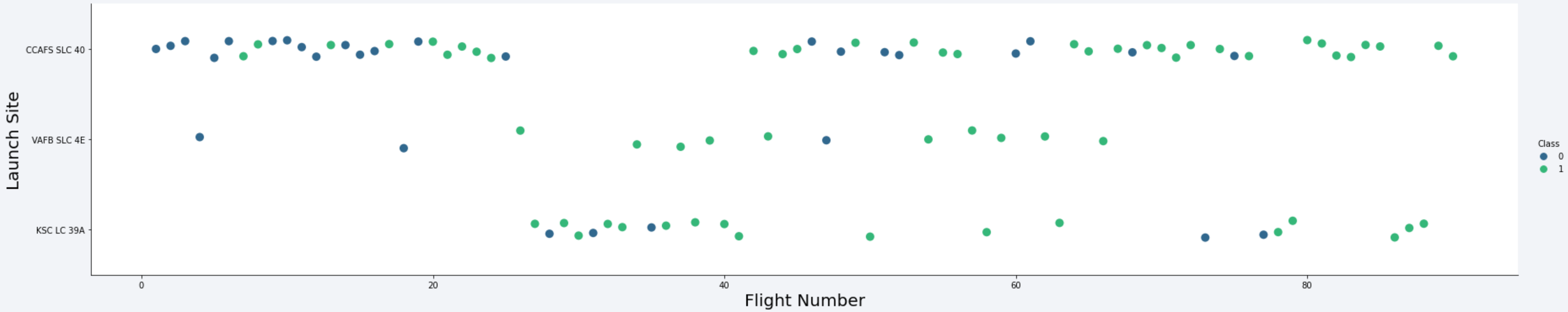
Results

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

Section 2

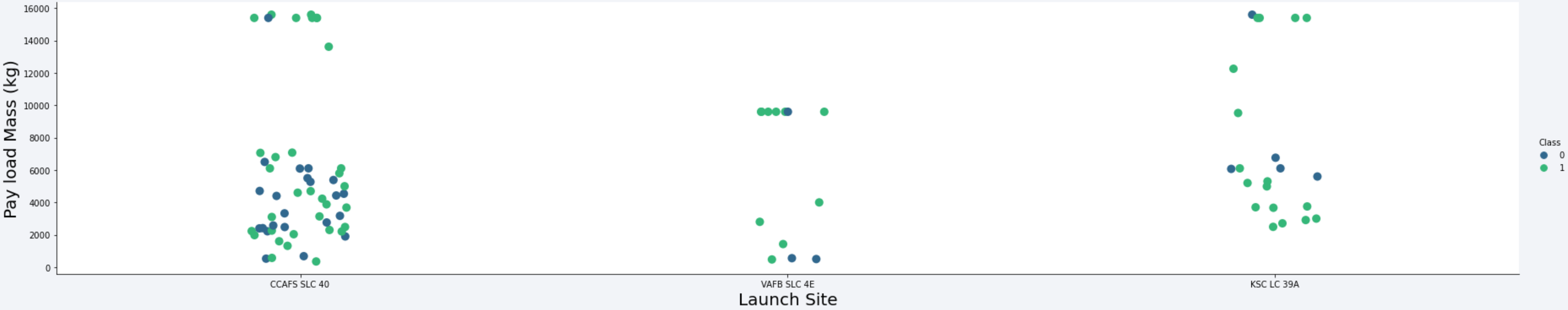
Insights drawn from EDA

Flight Number vs. Launch Site



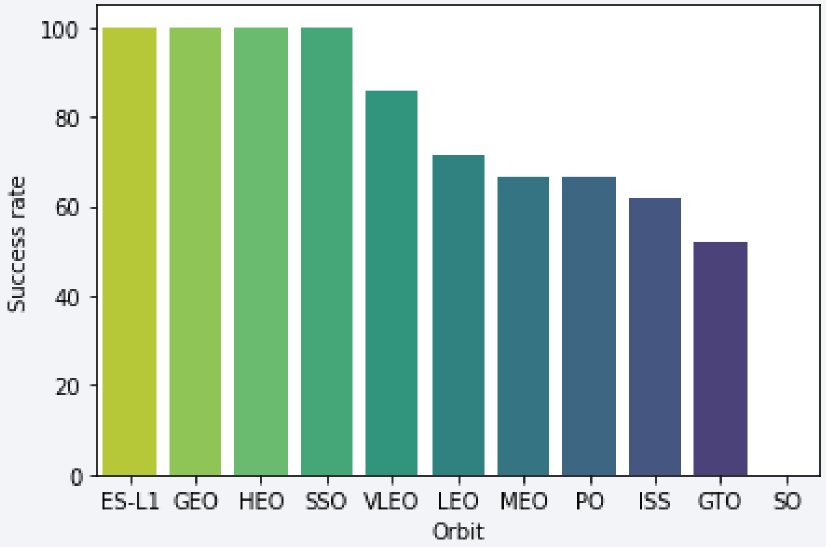
- Green indicates a successful launch. Purple indicates an unsuccessful launch
- Unsuccessful launches were more frequent in the early flight numbers, success rate has improved for more recent flights.

Payload vs. Launch Site



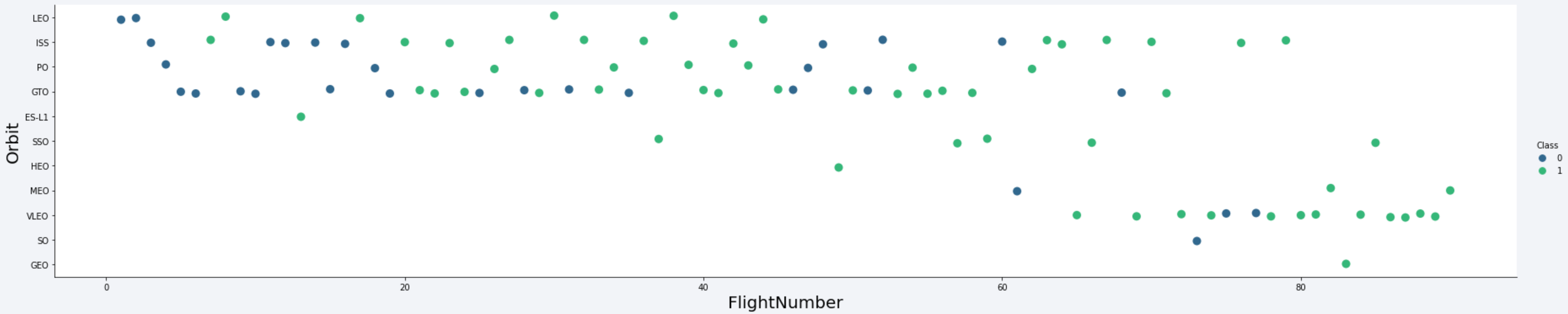
- Green indicates a successful launch. Purple indicates an unsuccessful launch.
- Unsuccessful launches are more frequent in flights with mid-lower pay load mass.

Success Rate vs. Orbit Type



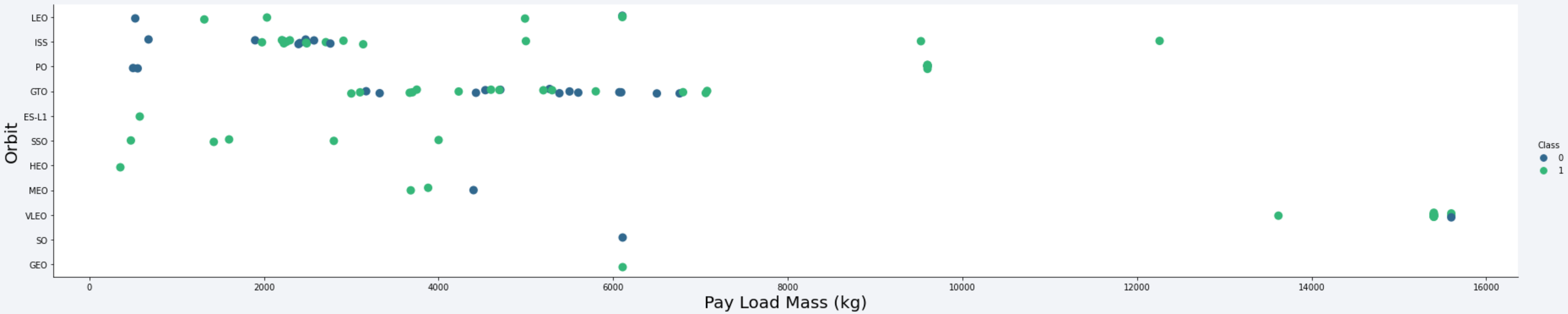
- ES-L1, GEO, HEO, SSO orbits have 100% successful launch rate.
- SO orbits have 0% successful launch rate.

Flight Number vs. Orbit Type



- Green indicates a successful launch. Purple indicates an unsuccessful launch
- Orbit preference appears to have changed over time. We see some correlation of a higher success rate with the more recent orbit preferences.

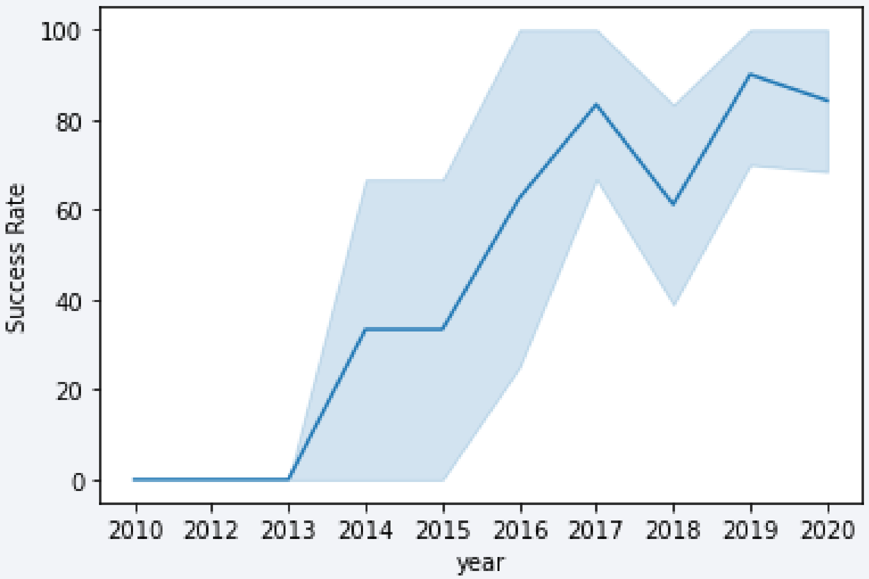
Payload vs. Orbit Type



- Show a scatter point of payload vs. orbit type

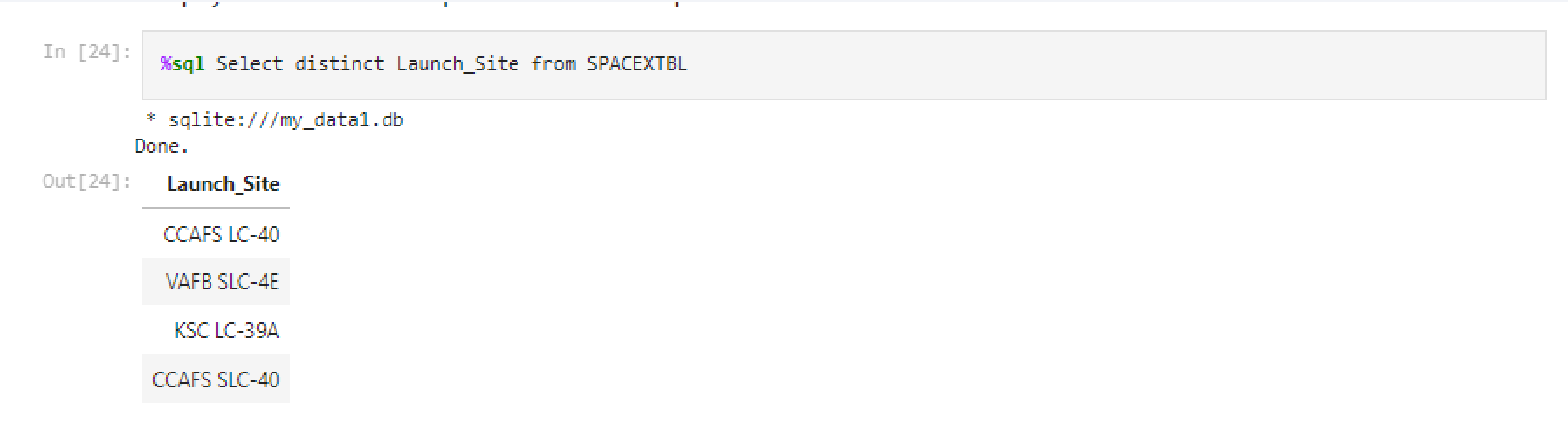
- Show the screenshot of the scatter plot with explanations

Launch Success Yearly Trend



- We can see that success rate has generally increased from 2013-2020, with a slight decrease in 2018, and the highest success rate so far being observed in 2019

All Launch Site Names

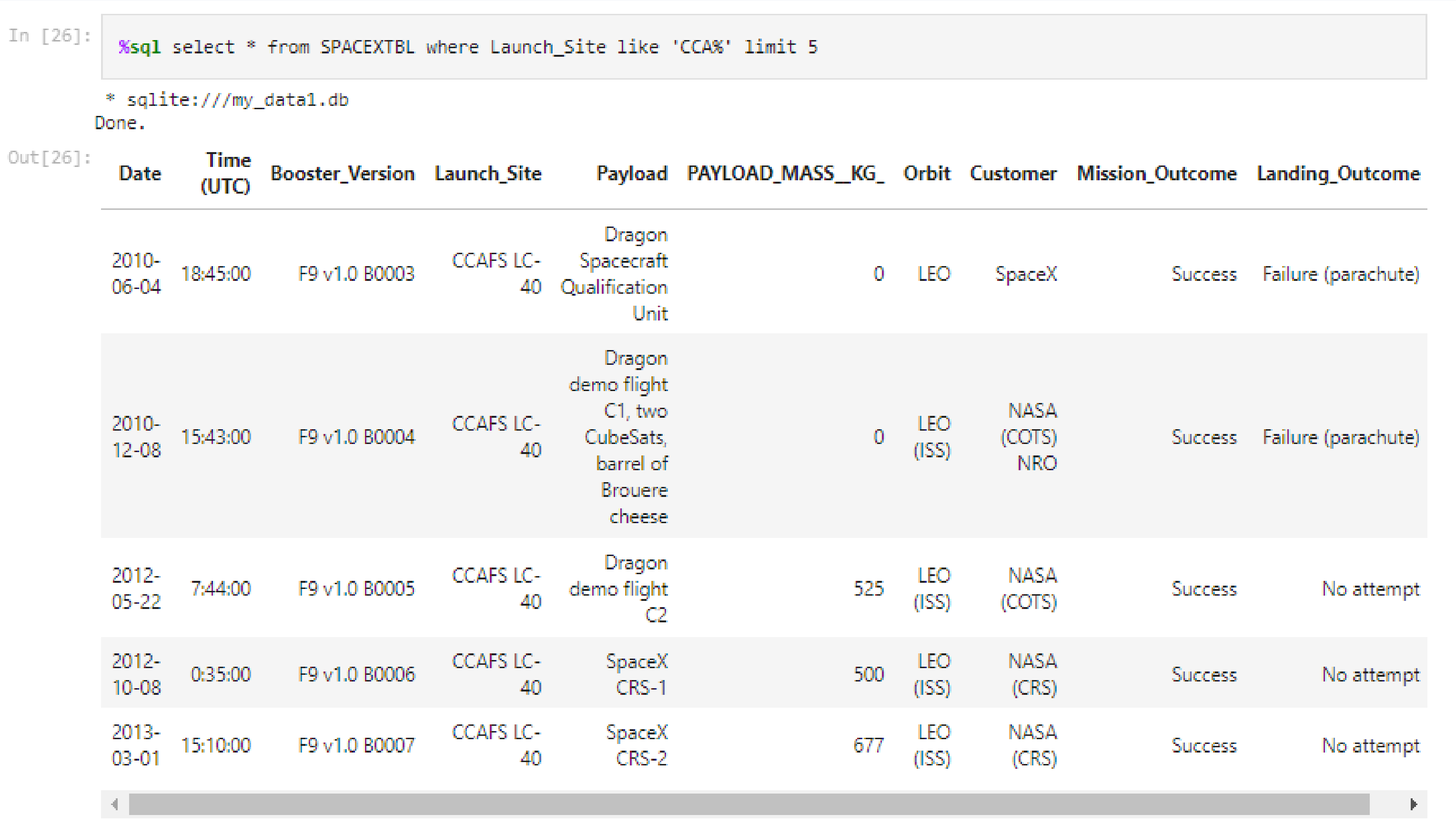


```
In [24]: %sql Select distinct Launch_Site from SPACEXTBL

 * sqlite:///my_data1.db
Done.
Out[24]:
```

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

- Return the names of the unique launch sites

Launch Site Names Begin with 'CCA'



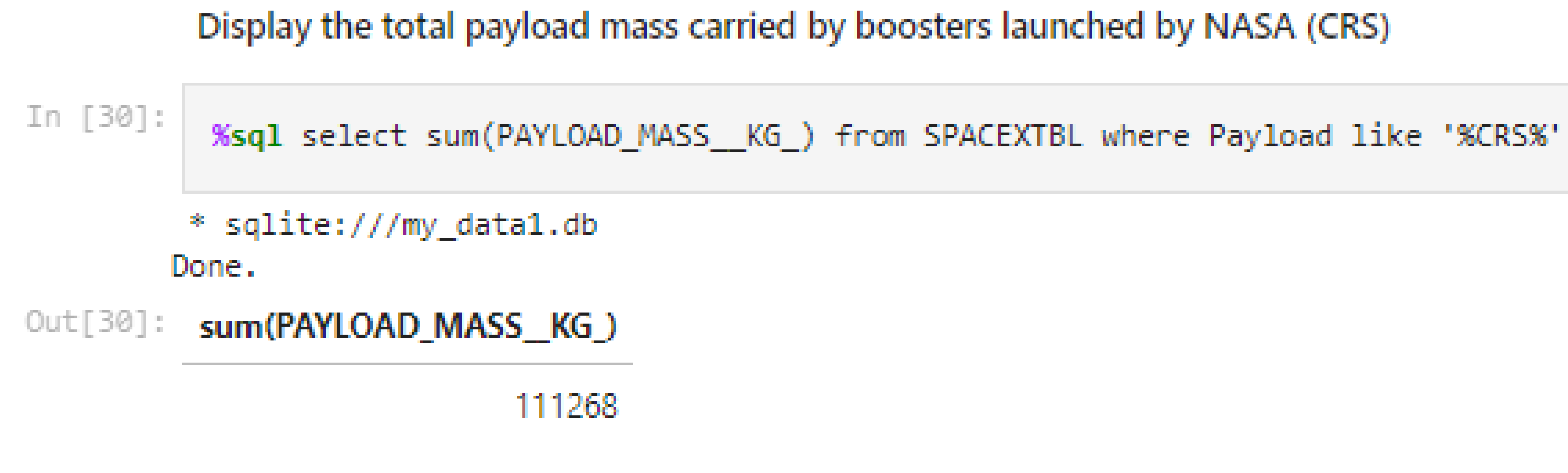
```
In [26]: %sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5

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

Out[26]:

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

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

Total Payload Mass



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

In [30]:
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Payload like '%CRS%'
```

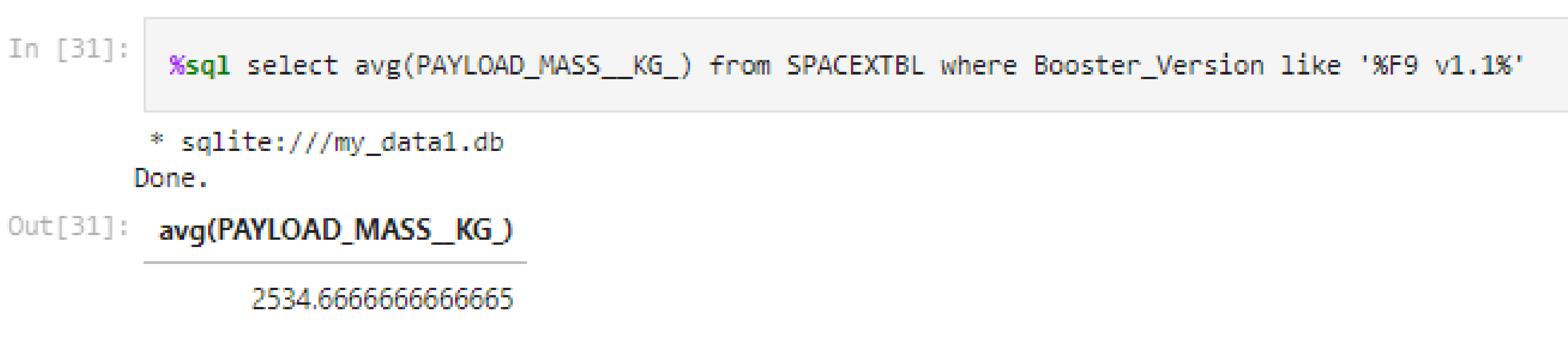
```
 * sqlite:///my_data1.db
Done.
```

Out[30]:

sum(PAYLOAD_MASS__KG_)
111268

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

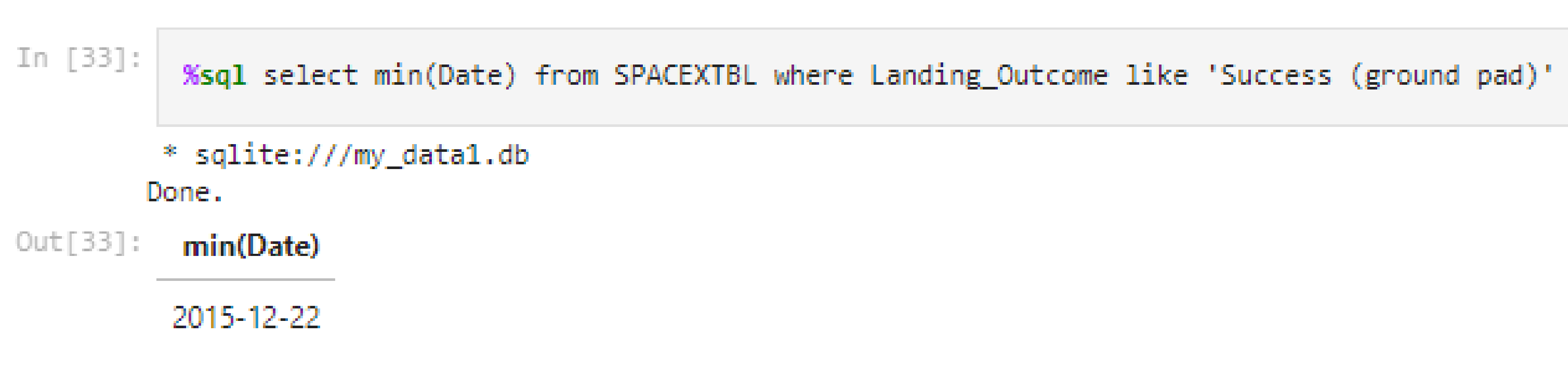
Average Payload Mass by F9 v1.1



```
In [31]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version like '%F9 v1.1%'

 * sqlite:///my_data1.db
Done.

Out[31]: avg(PAYLOAD_MASS__KG_)
         2534.6666666666665
```

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

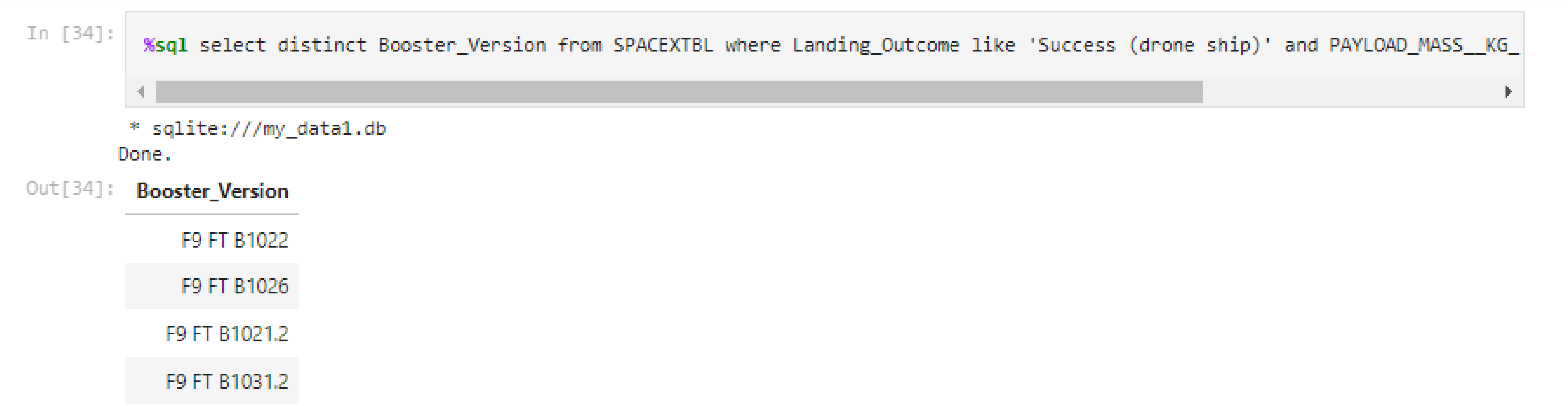
First Successful Ground Landing Date



```
In [33]:  %sql select min(Date) from SPACEXTBL where Landing_Outcome like 'Success (ground pad)'

           * sqlite:///my_data1.db
          Done.

Out[33]:  min(Date)

          2015-12-22
```

- Return the date of the first successful launch landing

Successful Drone Ship Landing with Payload between 4000 and 6000



```
In [34]: %sql select distinct Booster_Version from SPACEXTBL where Landing_Outcome like 'Success (drone ship)' and PAYLOAD_MASS__KG_
```

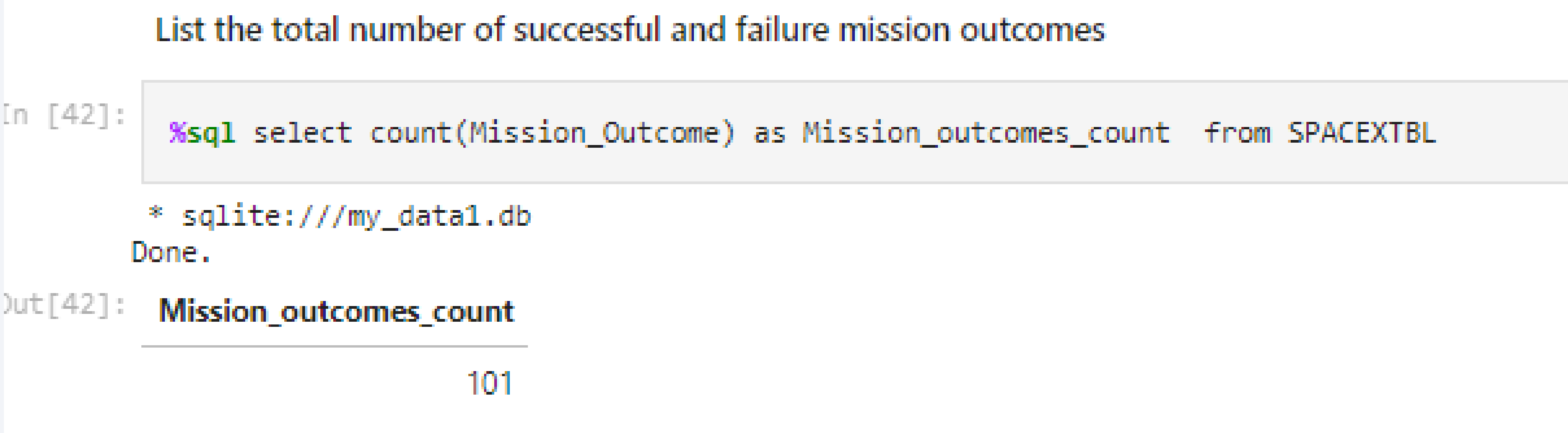
```
 * sqlite:///my_data1.db
Done.
```

Out[34]:

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

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

Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

```
In [42]:  %sql select count(Mission_Outcome) as Mission_outcomes_count  from SPACEXTBL
```

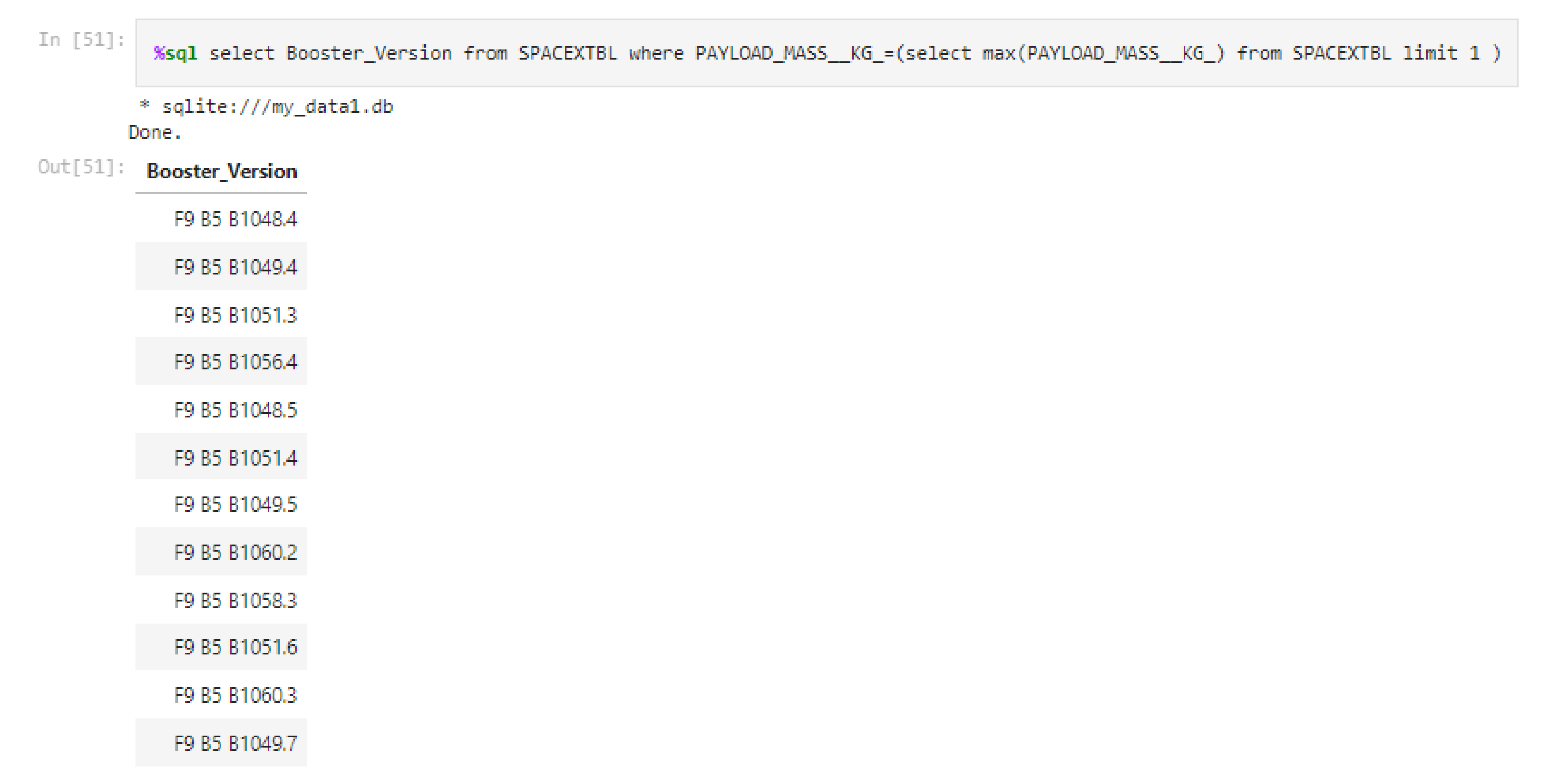
```
 * sqlite:///my_data1.db
Done.
```

Out[42]:

Mission_outcomes_count
101

- Return the total number of successful and failure mission outcomes

Boosters Carried Maximum Payload



In [51]:

```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL limit 1 )
```

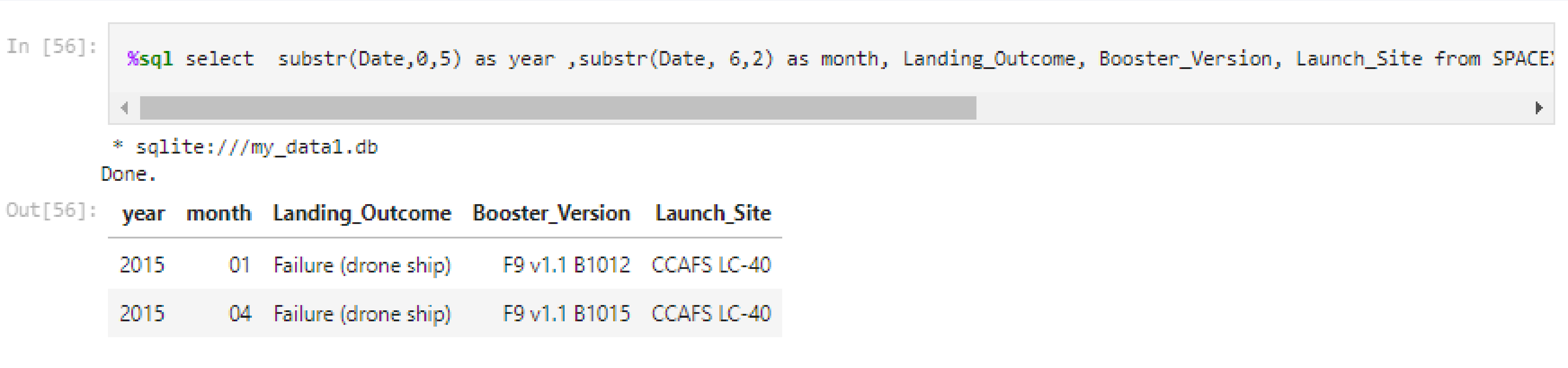
```
 * sqlite:///my_data1.db
Done.
```

Out[51]:

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

- Return all names of the booster_versions which have carried the maximum payload mass

2015 Launch Records



```
In [56]: %sql select  substr(Date,0,5) as year ,substr(Date, 6,2) as month, Landing_Outcome, Booster_Version, Launch_Site from SPACE
```

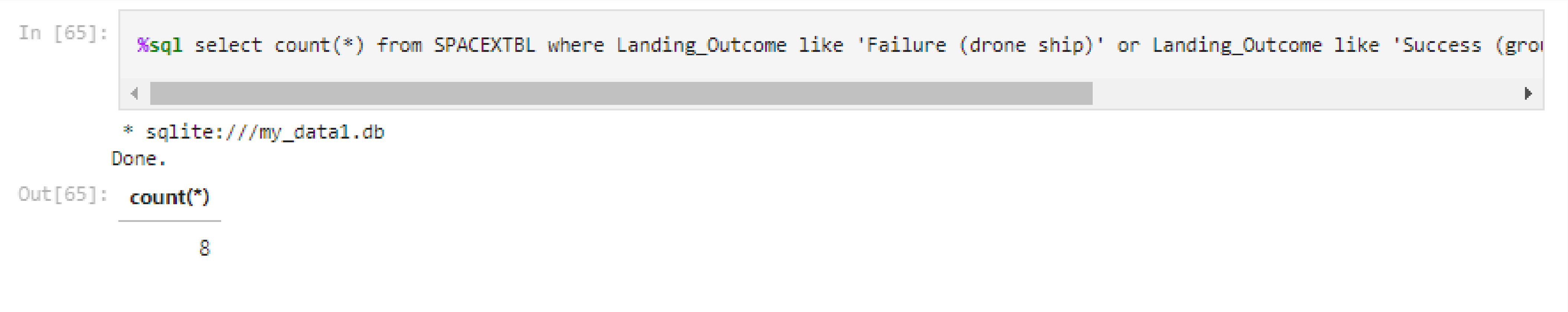
```
 * sqlite:///my_data1.db
Done.
```

Out[56]:

year	month	Landing_Outcome	Booster_Version	Launch_Site
2015	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Return all the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015

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



```
In [65]:  %sql select count(*) from SPACEXTBL where Landing_Outcome like 'Failure (drone ship)' or Landing_Outcome like 'Success (gro

 * sqlite:///my_data1.db
Done.
Out[65]:  count(*)
                 8
```

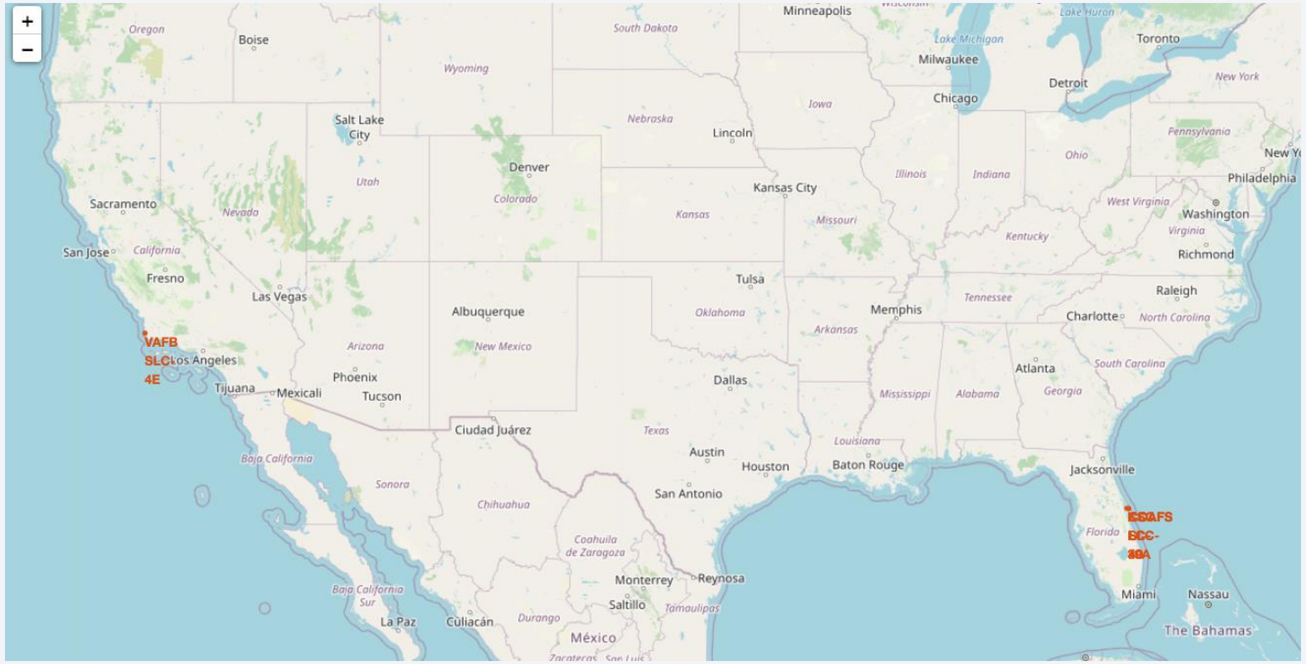
- Returns all landing outcome types and the number of occurrences for each, for successful landings between 04/06/2010 and 20/03/2017

Section 3

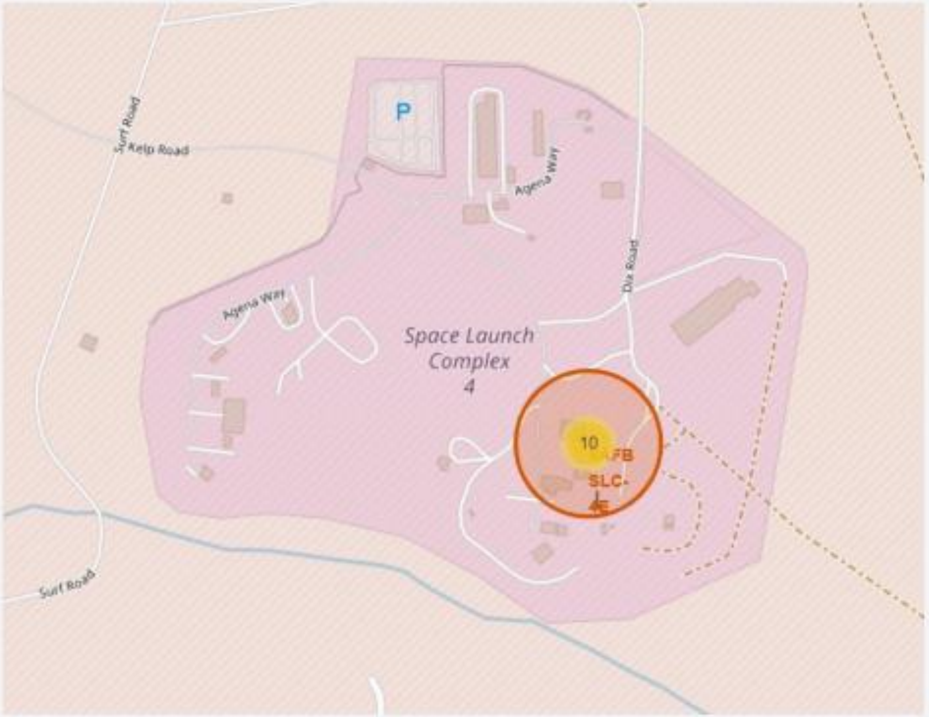
Launch Sites Proximities Analysis

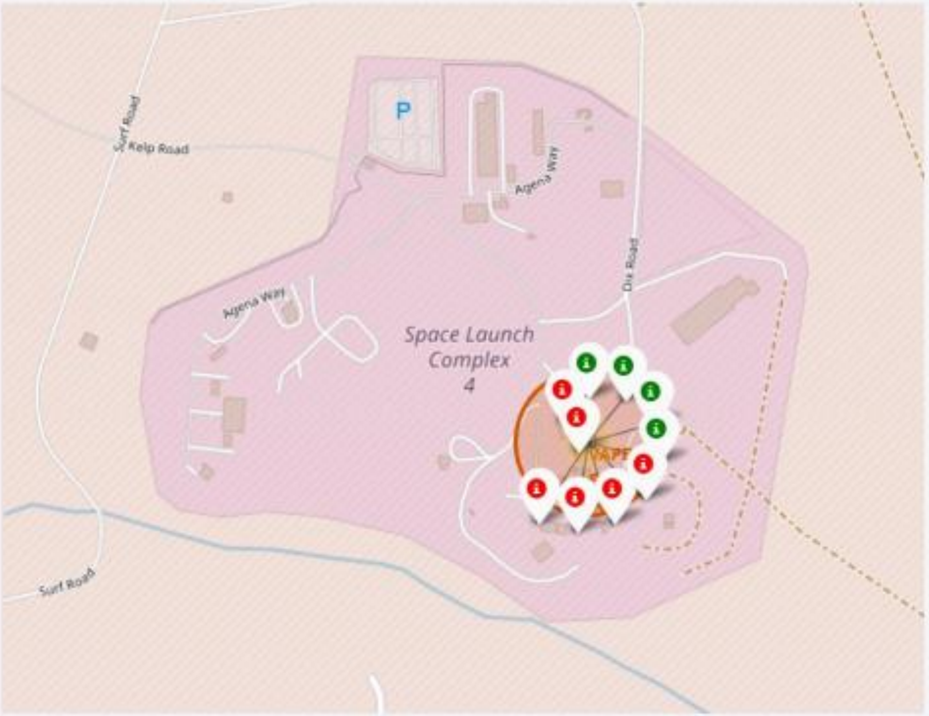
<Folium Map Screenshot 1>

- We can see that all launch sites are located in North America and that all launch sites are located near to coastlines, specifically the coasts of Florida and California



<Folium Map Screenshot 2>

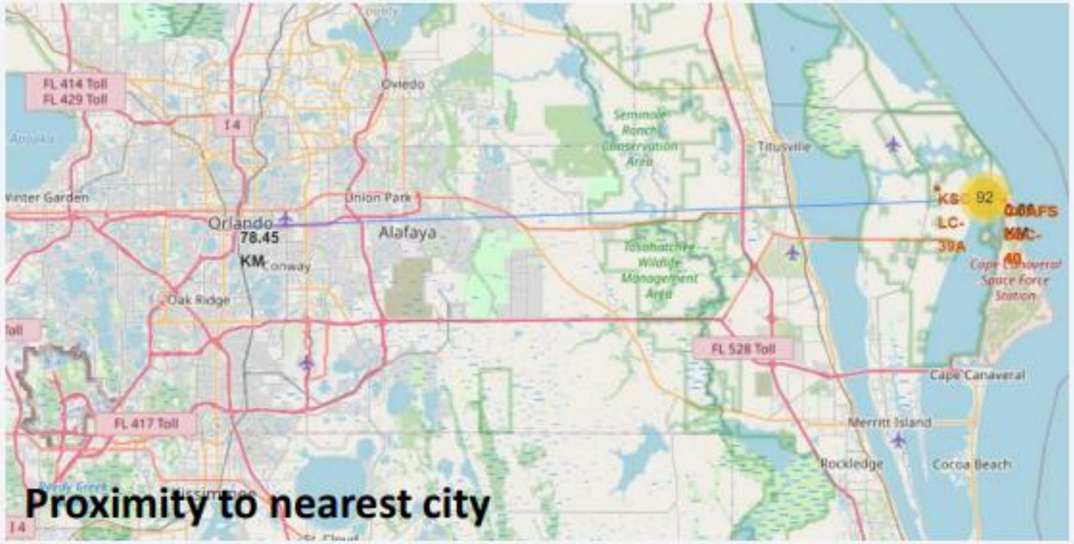




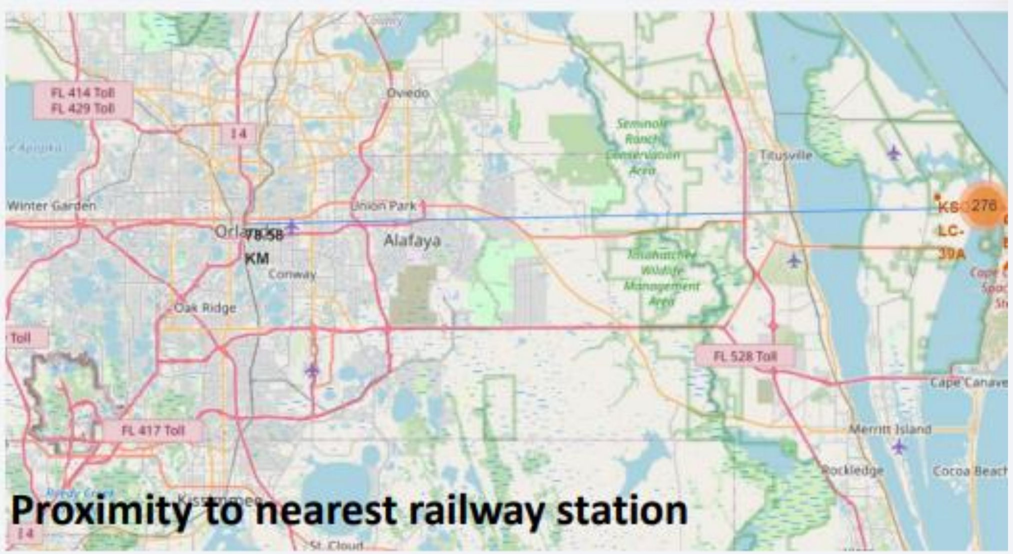
- **Insights**

- The left screenshot tells us that 10 launch records are clustered at this launch location (VAFB SLC-4E). We can drill down by clicking on the cluster, expanding the image like shown in the right screenshot. This tells us that there were 4 successful landings (green) and 6 unsuccessful landings (red).

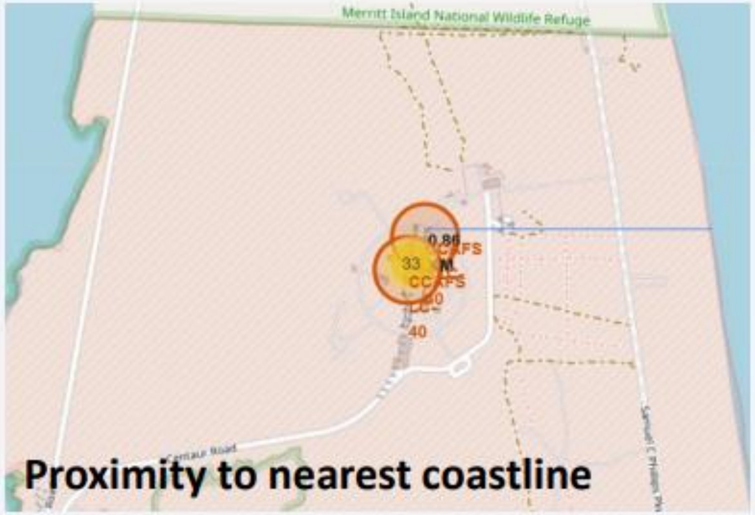
<Folium Map Screenshot 3>



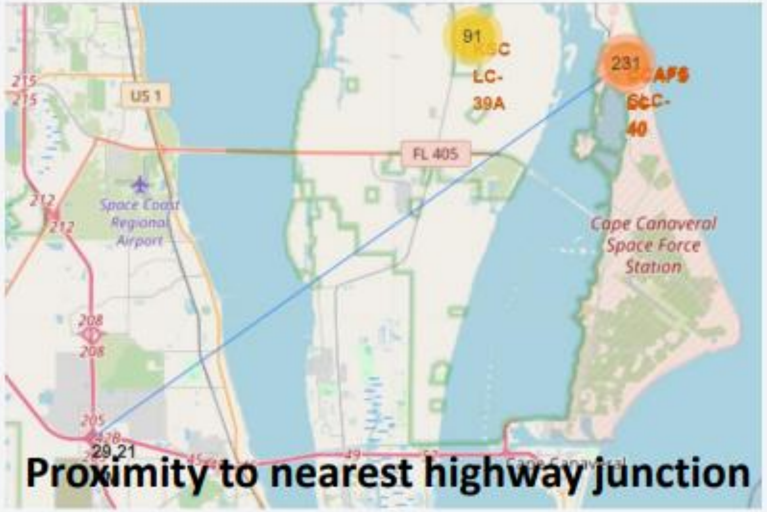
Proximity to nearest city



Proximity to nearest railway station



Proximity to nearest coastline

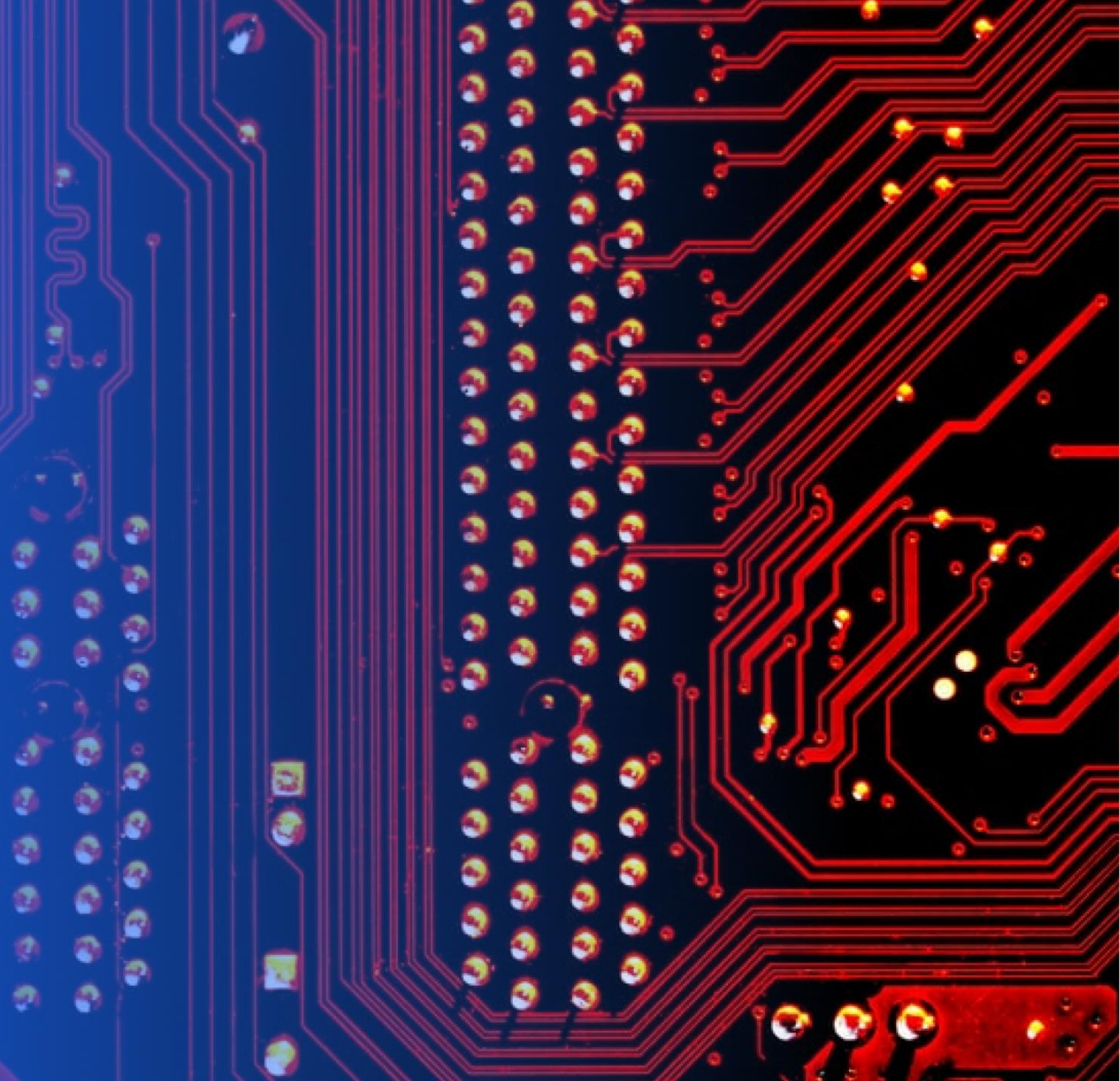


Proximity to nearest highway junction

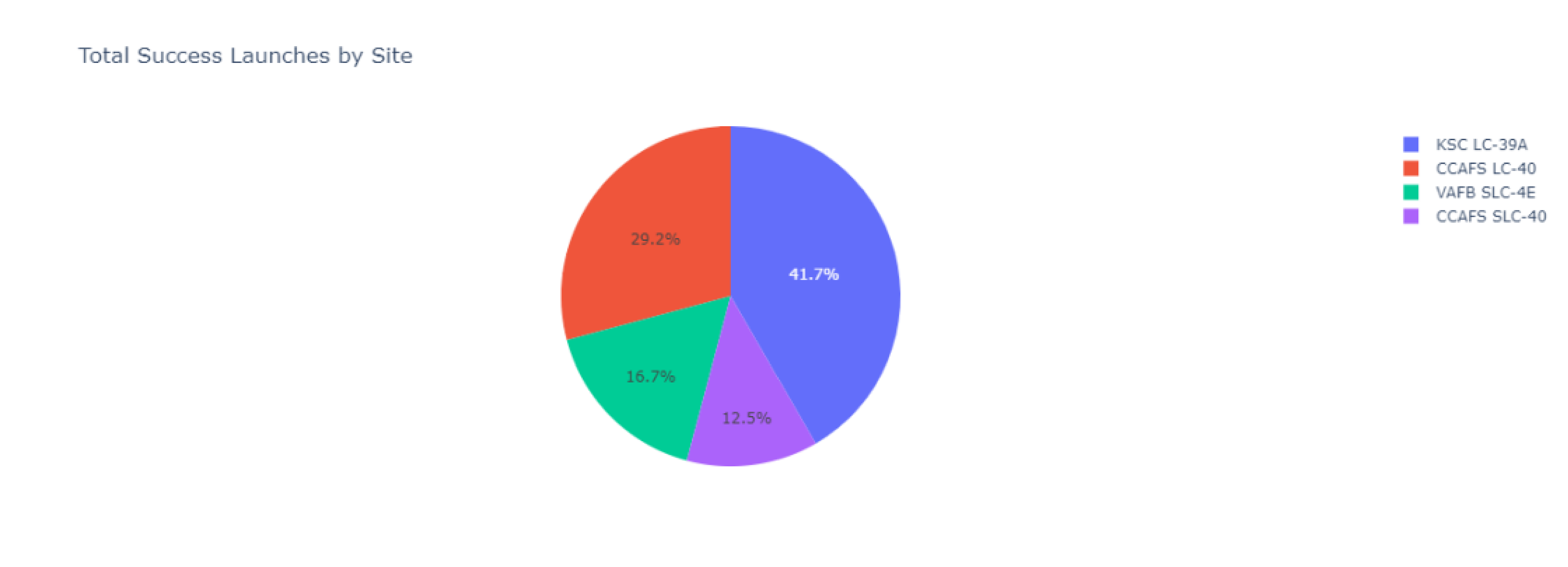
- **Insights**
  - Each screenshot shows proximity to key locations in km for launch site KSC LC-39A.

Section 4

Build a Dashboard with Plotly Dash



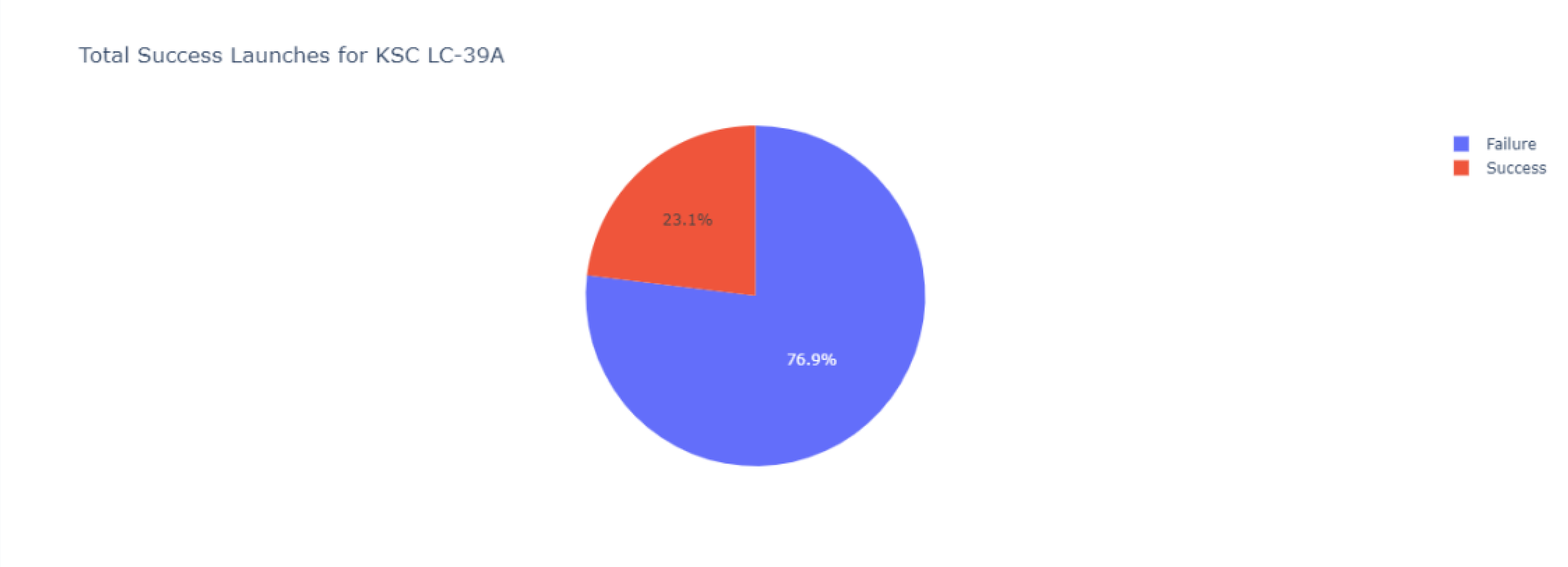
<Dashboard Screenshot 1>


Total Success Launches by Site

- **Insights**

- We can see that most successful landings were launches from KSC LC-39A. The least successful landings were launches from CCAFS SLC-40.

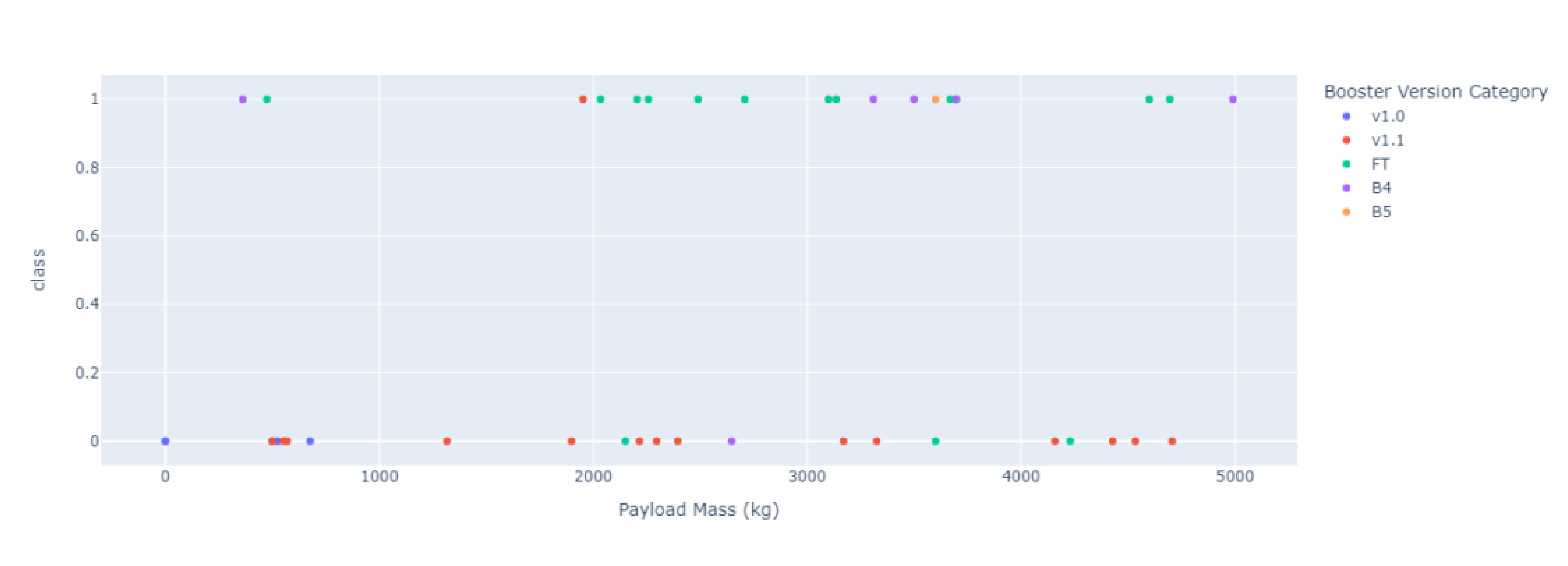
<Dashboard Screenshot 2>



- **Insights**

- Drilling into the most successful launch site, we can see the success vs. failure for KSC LC-39A. Even though many of the population successes are from this launch site, it actually has a low success rate.

<Dashboard Screenshot 3>



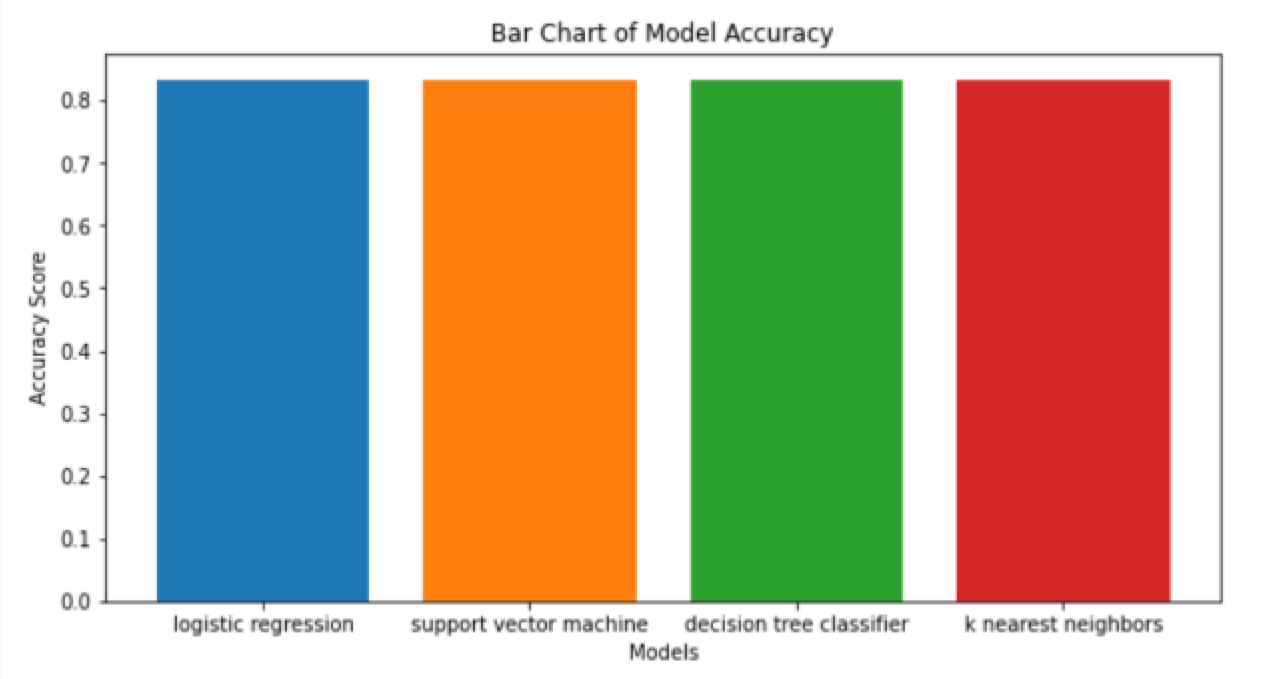
- **Insights**

- We can see that booster version category FT has many successes and few failures. In contrast, v1.1 has many failures and few successes. There are recorded failures with payload mass of 0kg, this may be a data entry error.

Section 5

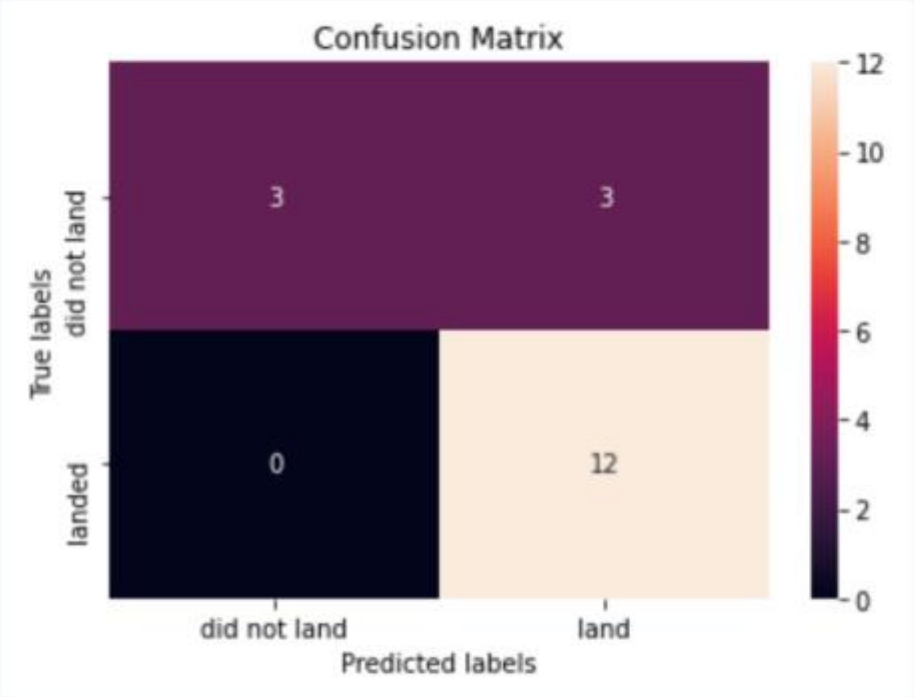
Predictive Analysis (Classification)

Classification Accuracy



- All models produced the same accuracy against the test data set (~83.33%).
- This is likely due to the limited data used, training and testing the models on larger data sets may produce more varied results.

Confusion Matrix



- All models generated the same confusion matrix.
- The models correctly predicted all successful landings as successful landings. The models only correctly predicted half of unsuccessful landings. The models over predict successful landings, they tend to give false positives.

Conclusions

- Our goal was to develop a machine learning model to predict if stage 1 will successfully land for a given launch.
- We developed four machine learning models, which all predicted successful landings with ~83.33% accuracy for some test data. The models tend to over predict successful landings, the models could be improved by using more data.
- In addition, we found that:
  - Success rate of stage 1 landings has improved over time.
  - ES-L1, GEO, HEO, SSO orbits have the best success rate.
  - Launch sites are typically located close to coastlines.

Appendix

- Github URL

Thank you!