IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Md Golam Kibria

January 17, 2024

Outline

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

Executive Summary

- Summary of methodologies
    - Data Collection through API
    - Data Collection with Web Scraping
    - Data Wrangling
    - Exploratory Data Analysis with SQL
    - Data Exploration and Preparation
    - Interactive Visual Analytics with Folium
    - Machine Learning Prediction
- Summary of all results
    - Exploratory Data Analysis Results
    - Interactive Analytics Results
    - Predictive Analytics Results

Introduction

- **Project background and context**

  The commercial space industry is thriving with companies like Virgin Galactic, Rocket Lab, Blue Origin, and notably SpaceX. SpaceX stands out for its achievements, including sending spacecraft to the International Space Station, launching Starlink, and conducting manned missions. The Falcon 9 rocket, advertised at $62 million per launch, owes its cost efficiency to the ability to reuse the first stage, a substantial and expensive component. Therefore, it is hypothesized that if SpaceX can successfully land the first stage, it can save substantial amount reusing the components used in the first stage. Therefore, the aim of this project is to determine whether or not the first stage will land usefully. To find the answer, this project creates a machine learning pipeline to predict if the first stage will land successfully.

- **Problems you want to find answers**
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

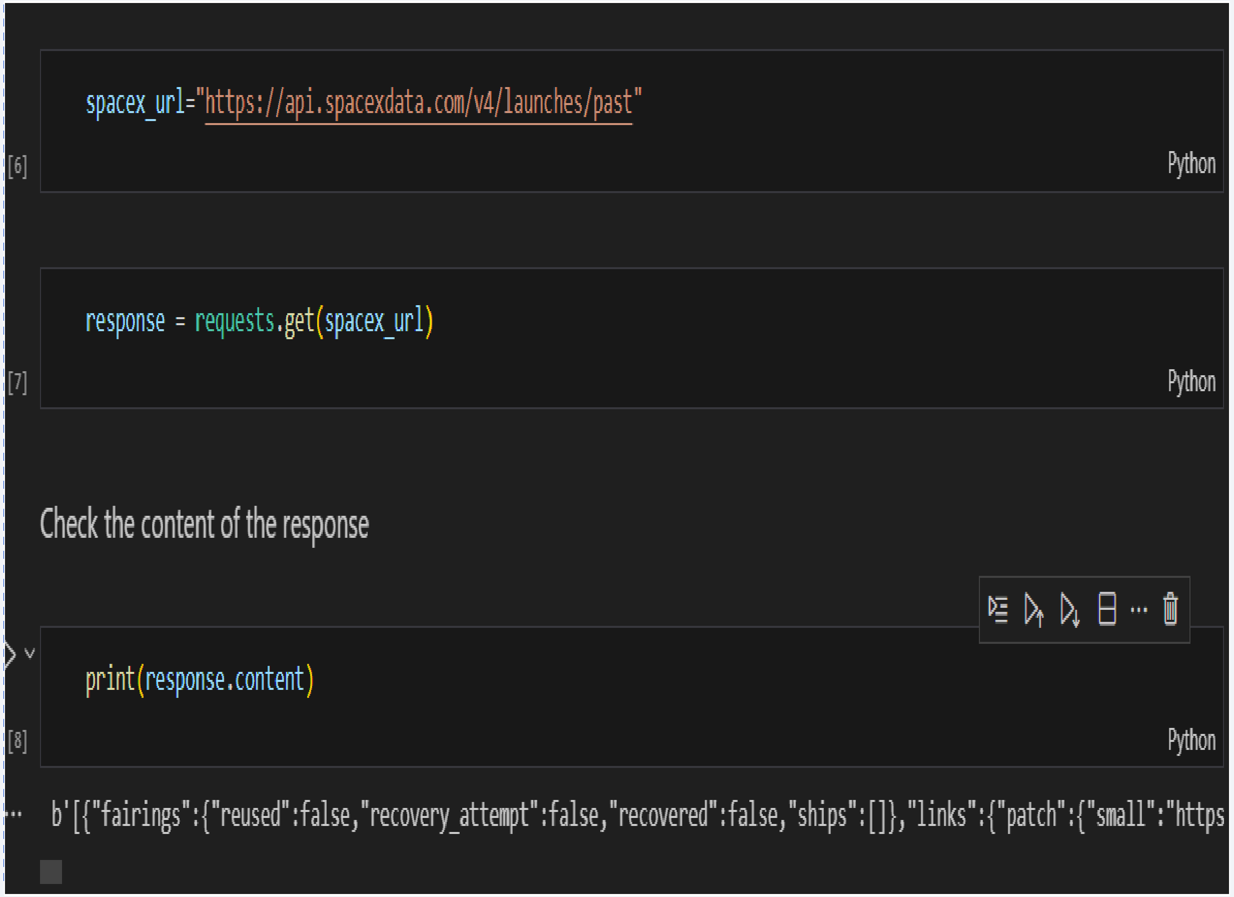
- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - Data was prepared for analysis using different data preparation technniques. For example, One-hot encoding was applied to categorical features.
- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using different classification models

Data Collection

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- https://github.com/mkibria2014/IBMCapstonProject/blob/main/Lab%201a%20-%20Data%20Collection%20using%20API.ipynb



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

[6] Python

```
response = requests.get(spacex_url)
```

[7] Python

Check the content of the response

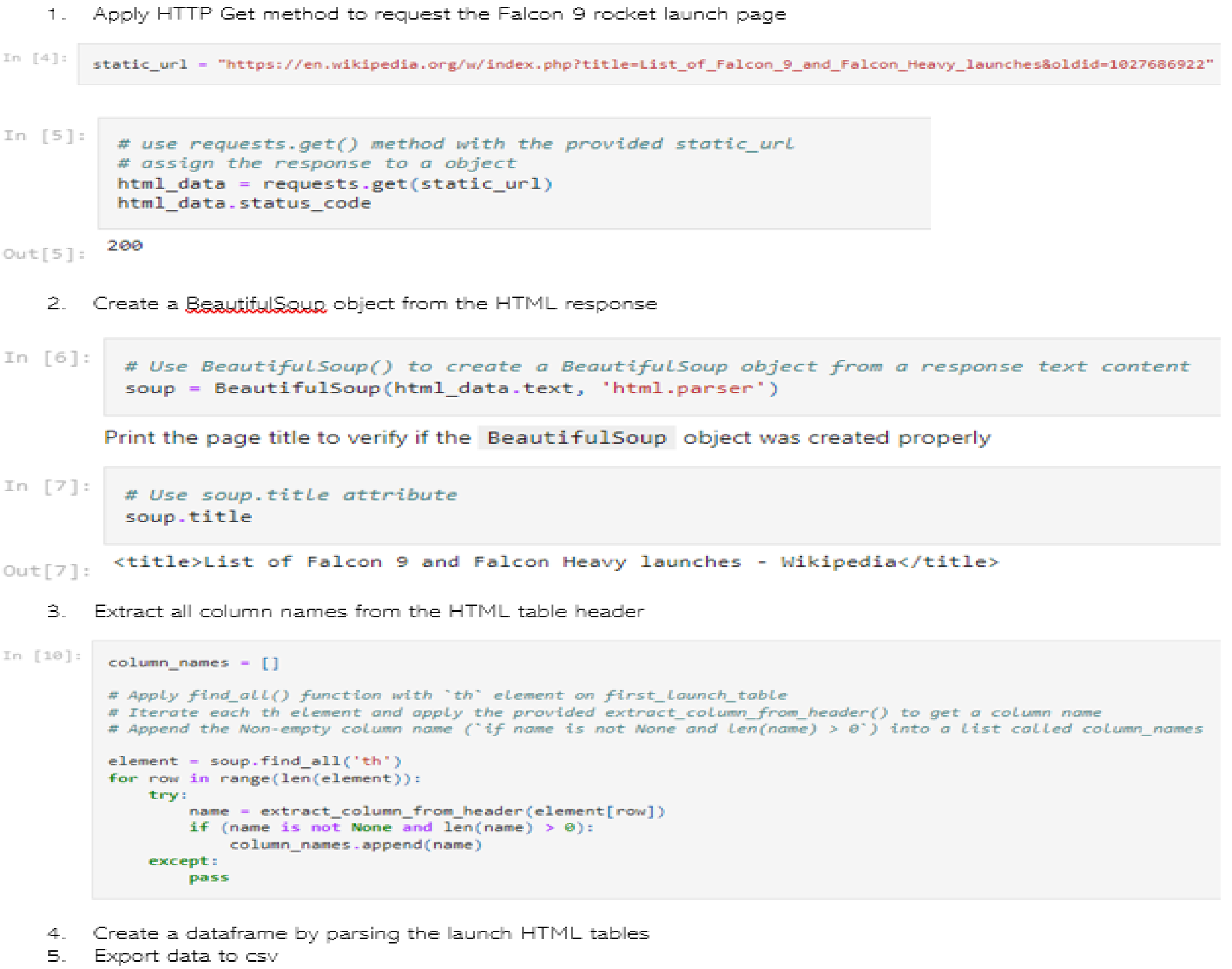
```
print(response.content)
```

[8] Python

```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https
```

Data Collection - Scraping

- Applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup. Then, parsed the table and converted it into a pandas dataframe.
- https://github.com/mkibria2014/IBMCapstonProject/blob/main/Lab%201b%20-%20Data%20Collection%20with%20Web%20Scraping.ipynb



1. Apply HTTP Get method to request the Falcon 9 rocket launch page

```
In [4]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
In [5]: # use requests.get() method with the provided static_url
        # assign the response to a object
        html_data = requests.get(static_url)
        html_data.status_code

Out[5]: 200
```

2. Create a BeautifulSoup object from the HTML response

```
In [6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
        soup = BeautifulSoup(html_data.text, 'html.parser')
```

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

```
In [7]: # Use soup.title attribute
        soup.title

Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Extract all column names from the HTML table header

```
In [10]: column_names = []

         # Apply find_all() function with `th` element on first_launch_table
         # Iterate each th element and apply the provided extract_column_from_header() to get a column name
         # Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

         element = soup.find_all('th')
         for row in range(len(element)):
             try:
                 name = extract_column_from_header(element[row])
                 if (name is not None and len(name) > 0):
                     column_names.append(name)
             except:
                 pass
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

Data Wrangling

- First, performed exploratory data analysis and determined the training labels.
- Then, calculated the number of launches at each site, and the number and occurrence of each orbits
- Finally, created landing outcome label from outcome column and exported the results to csv.

- https://github.com/mkibria2014/IBMCapstonProject/blob/main/Lab%202%20-%20Data%20Wrangling.ipynb

EDA with Data Visualization

- Explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- https://github.com/mkibria2014/IBMCapstonProject/blob/main/Lab%204%20-%20Exploring%20and%20Preparing%20Data.ipynb

EDA with SQL

- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
- https://github.com/mkibria2014/IBMCapstonProject/blob/main/Lab%203%20-%20Exploratory%20Data%20Analysis%20using%20SQL.ipynb

Build an Interactive Map with Folium

- 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.
- Assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- 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.
- https://github.com/mkibria2014/IBMCapstonProject/blob/main/Lab%205-Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

Predictive Analysis (Classification)

- Loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- Built different machine learning models and tune different hyperparameters using GridSearchCV.
- Used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- Found the best performing classification model.
- https://github.com/mkibria2014/IBMCapstonProject/blob/main/Lab%206%20-%20Machine%20Learning%20Prediction.ipynb

Results

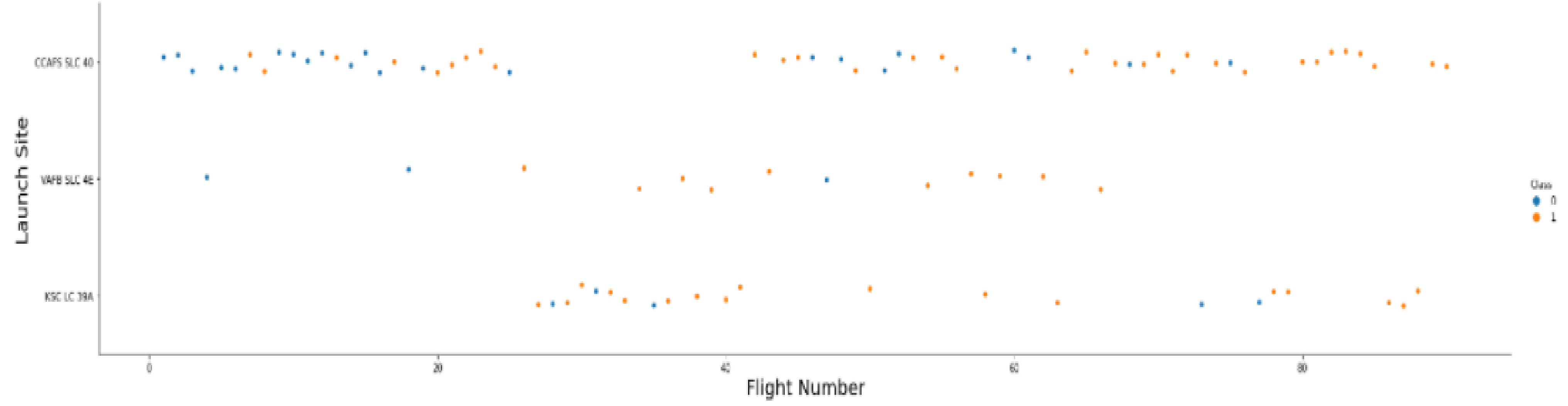
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

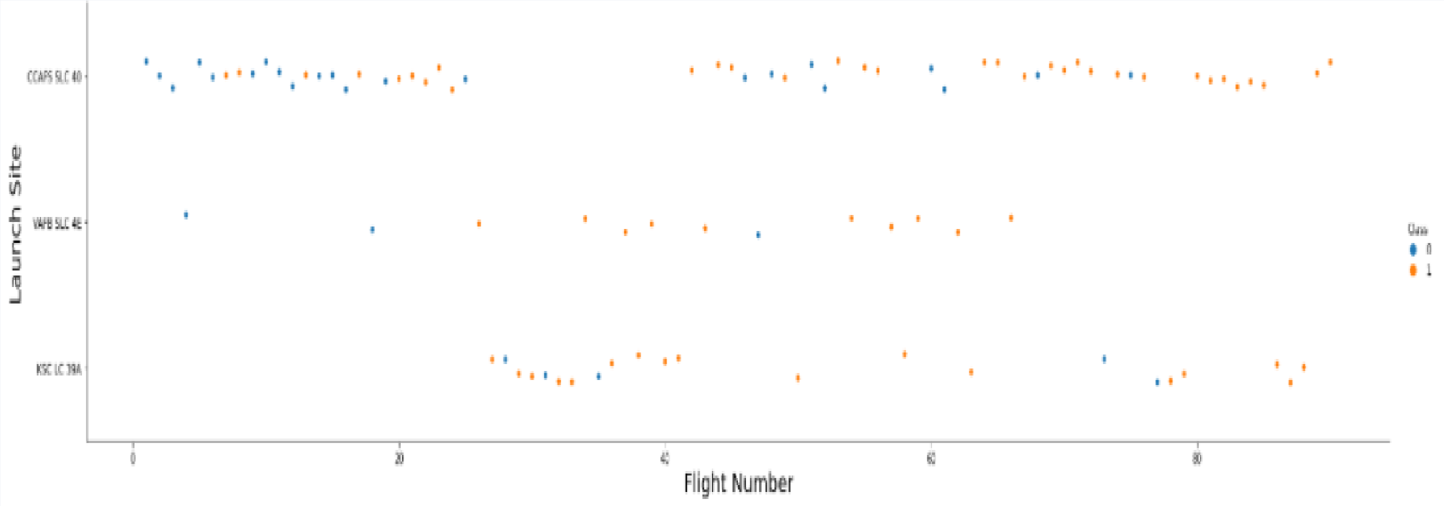
Insights drawn from EDA

Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



Payload vs. Launch Site

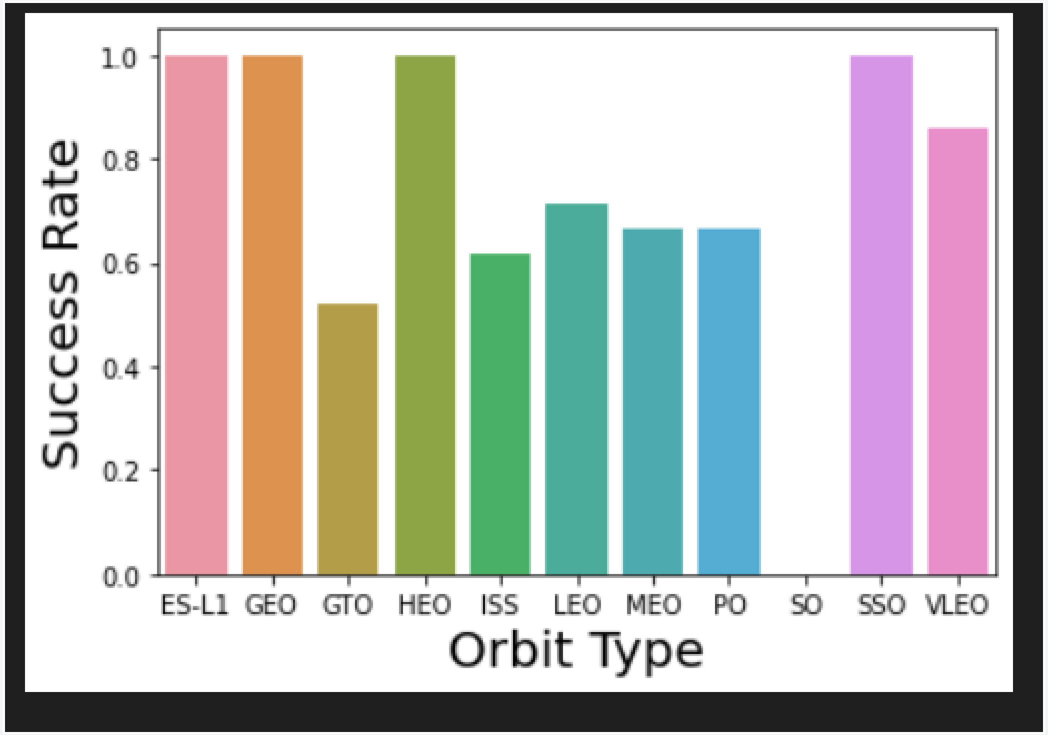


Explanation:

The greater the payload mass for launch site CCAFS SLC 40, the higher the success rate for the rocket

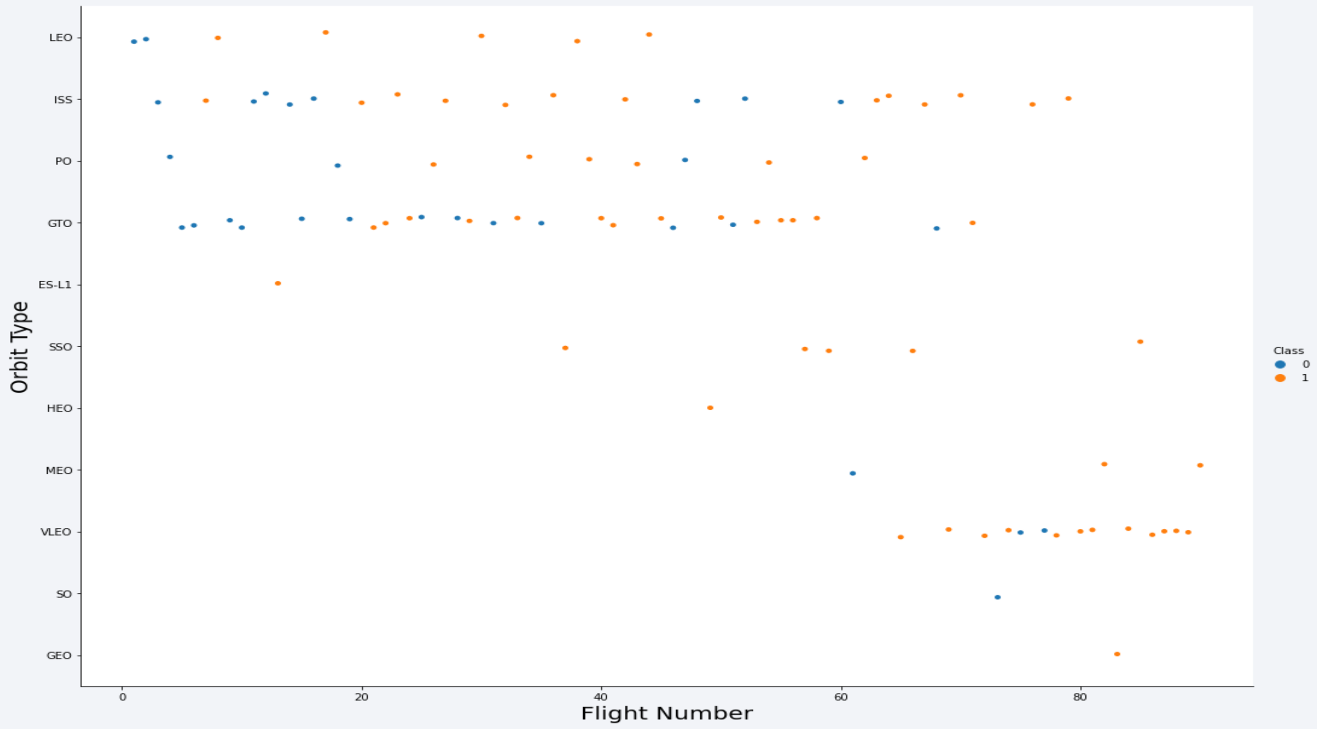
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, and SSO have same rate of success rate (1), followed by VLEO (more than 80%).



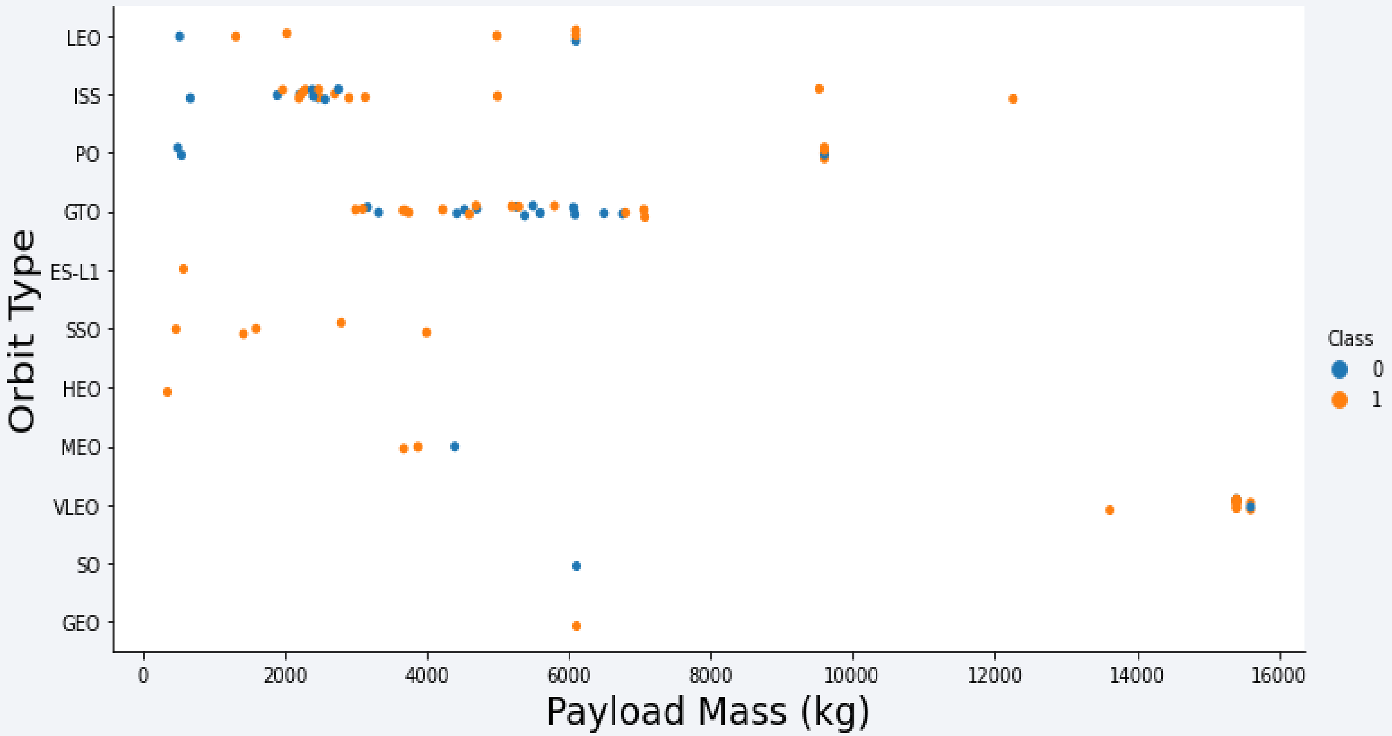
Flight Number vs. Orbit Type

- It is observed that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



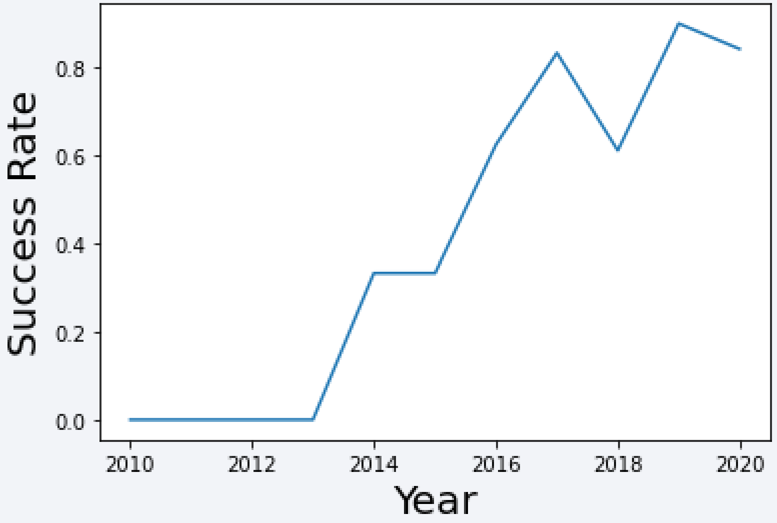
Payload vs. Orbit Type

- It is observed that most of successful landings have Payload Mass less than 8000.
- Also, heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



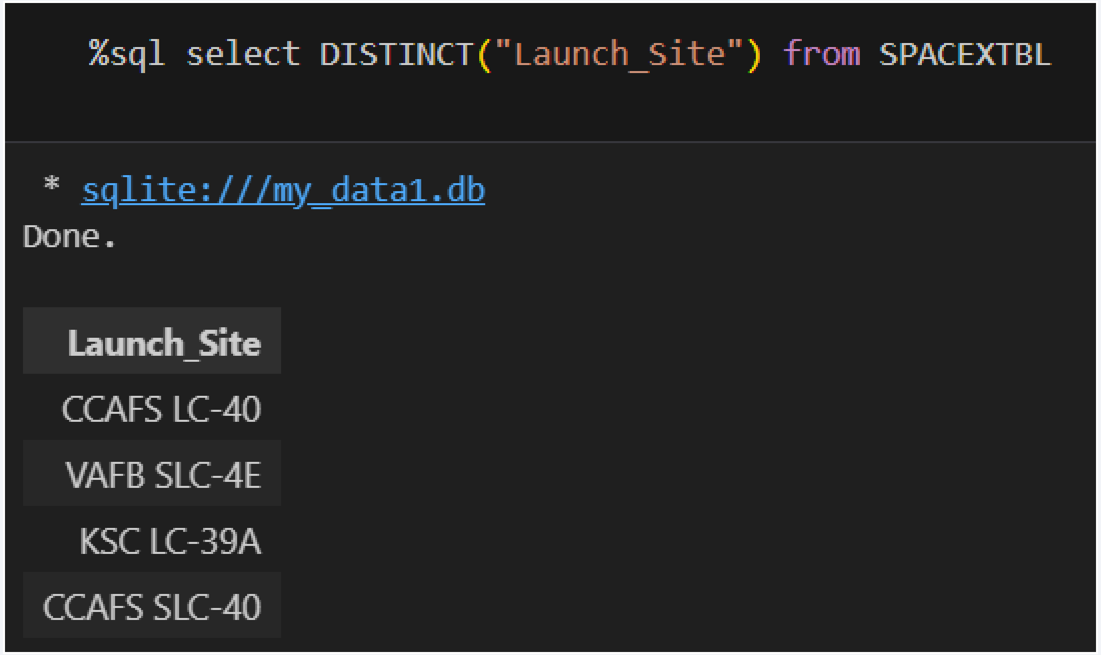
Launch Success Yearly Trend

- It is observed that the success rate constantly increased since 2013 to 2020, with a little bit drop in 2013.



All Launch Site Names

- **DISTINCT** keyword was used to find only unique launch sites from the SpaceX data.

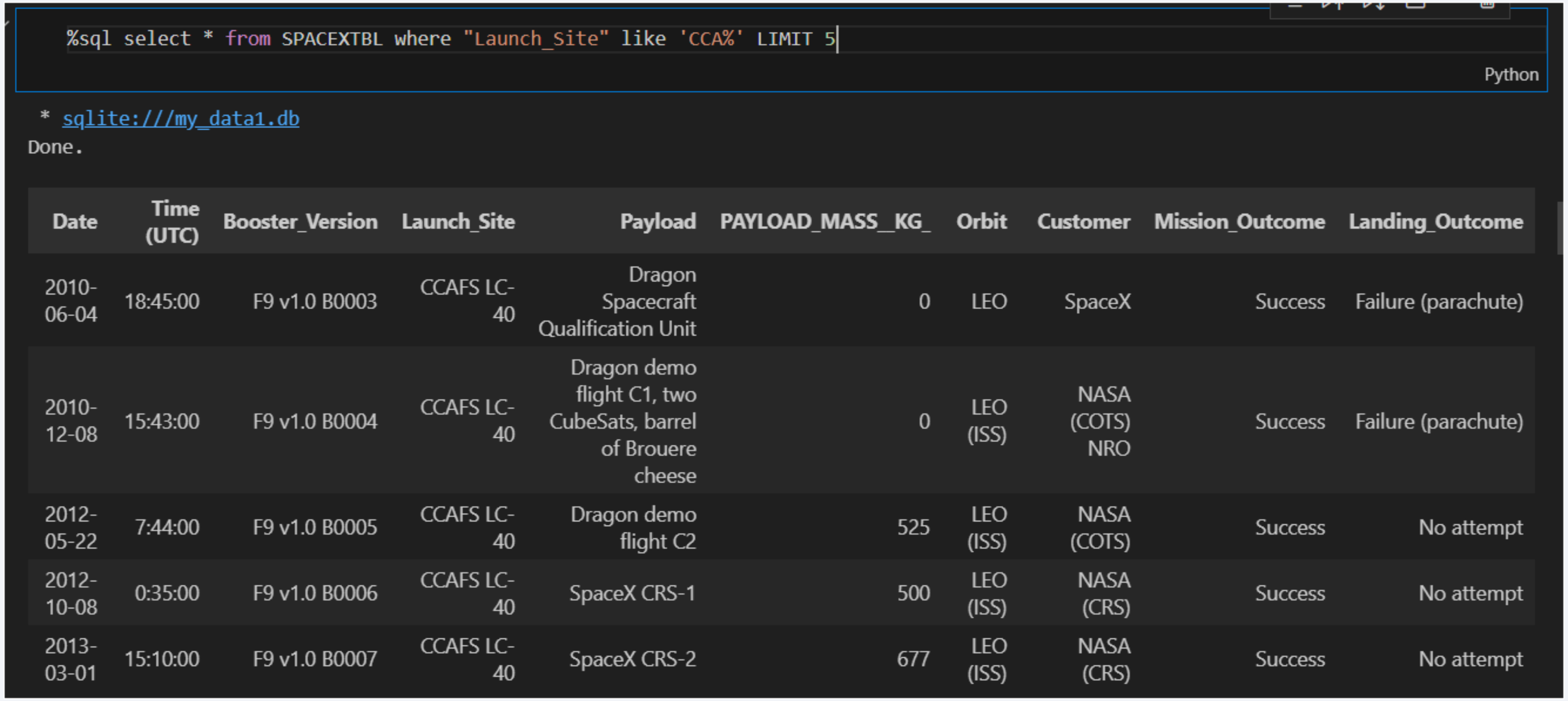


```
%sql select DISTINCT("Launch_Site") from SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'



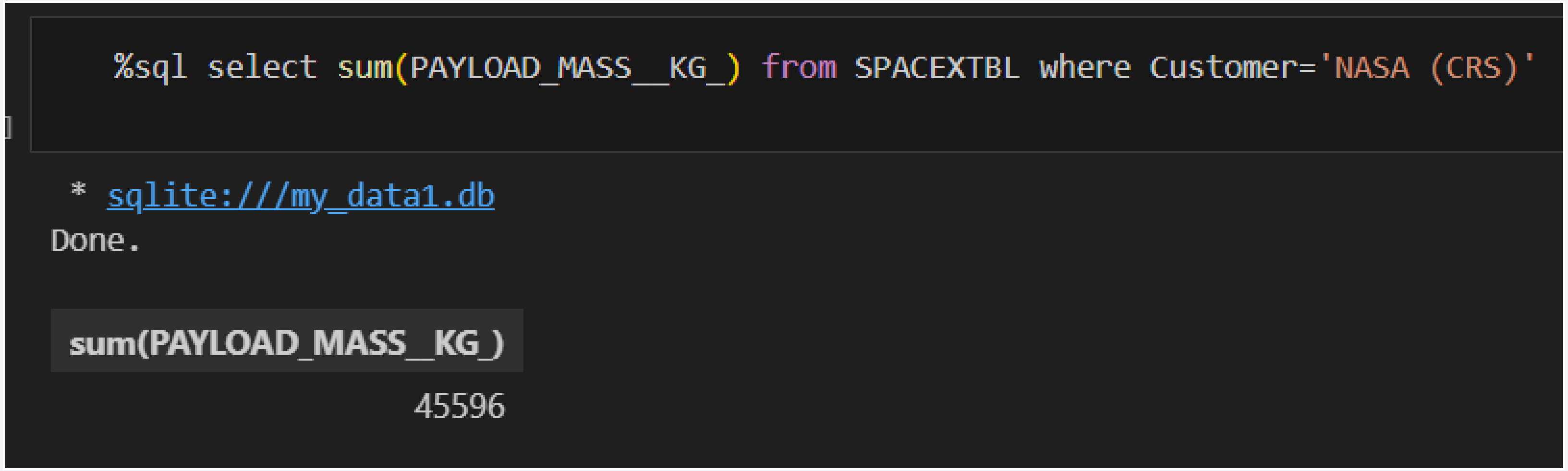
```
%sql select * from SPACEXTBL where "Launch_Site" like 'CCA%' LIMIT 5
```

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

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

We used the query above to display 5 records where launch sites begin with \`CCA\`

Total Payload Mass

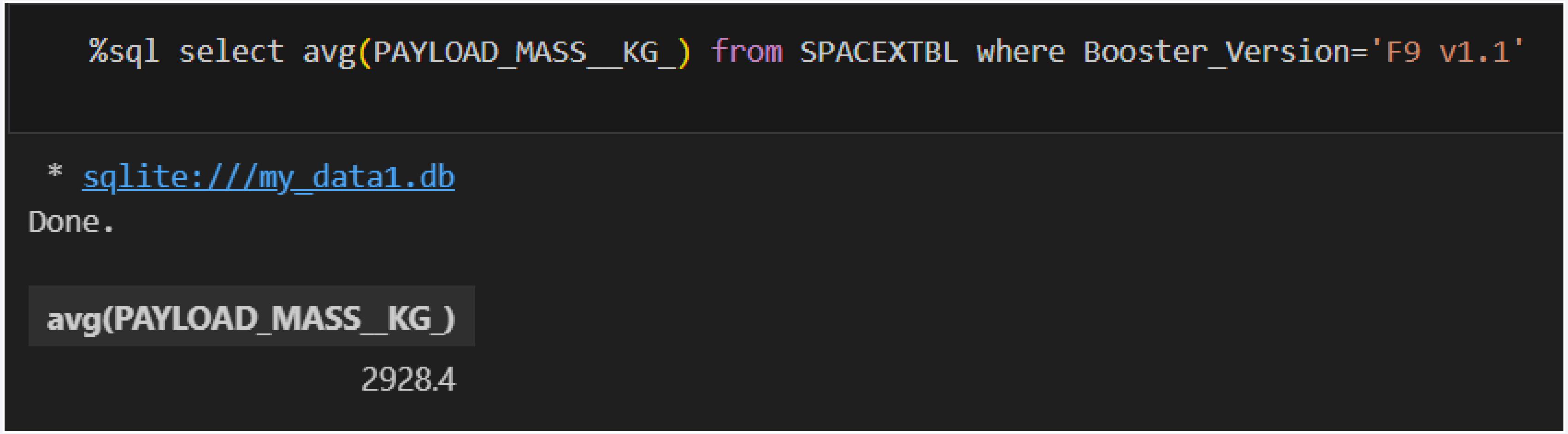


```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer='NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
                 45596
```

- The above query found the total payload mass as 45596 in KG.

Average Payload Mass by F9 v1.1



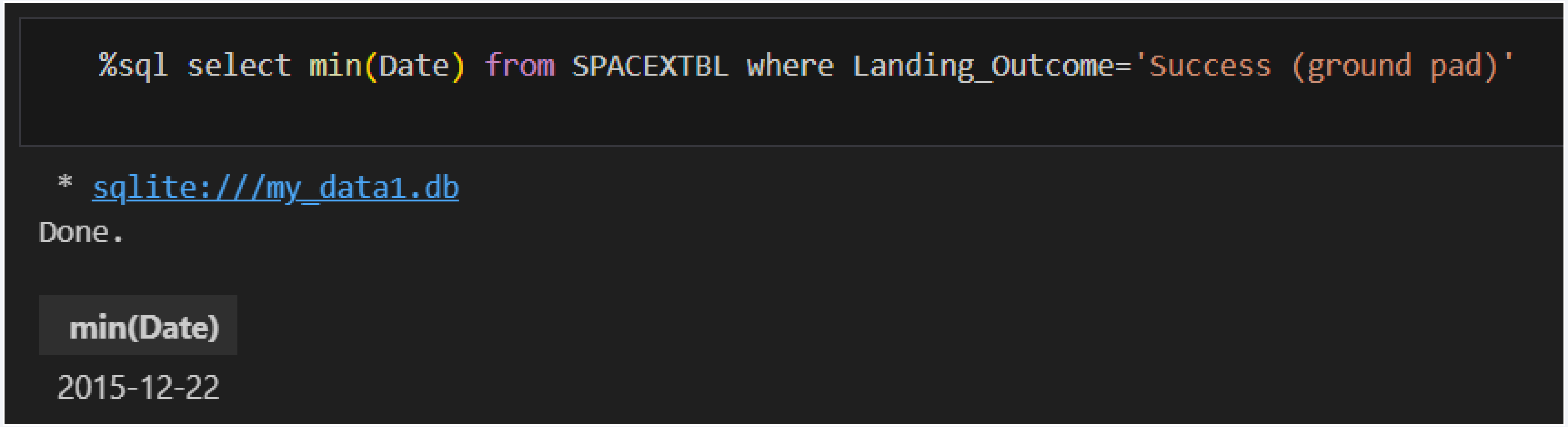
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version='F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

The Average payload mass = 2928.4

First Successful Ground Landing Date



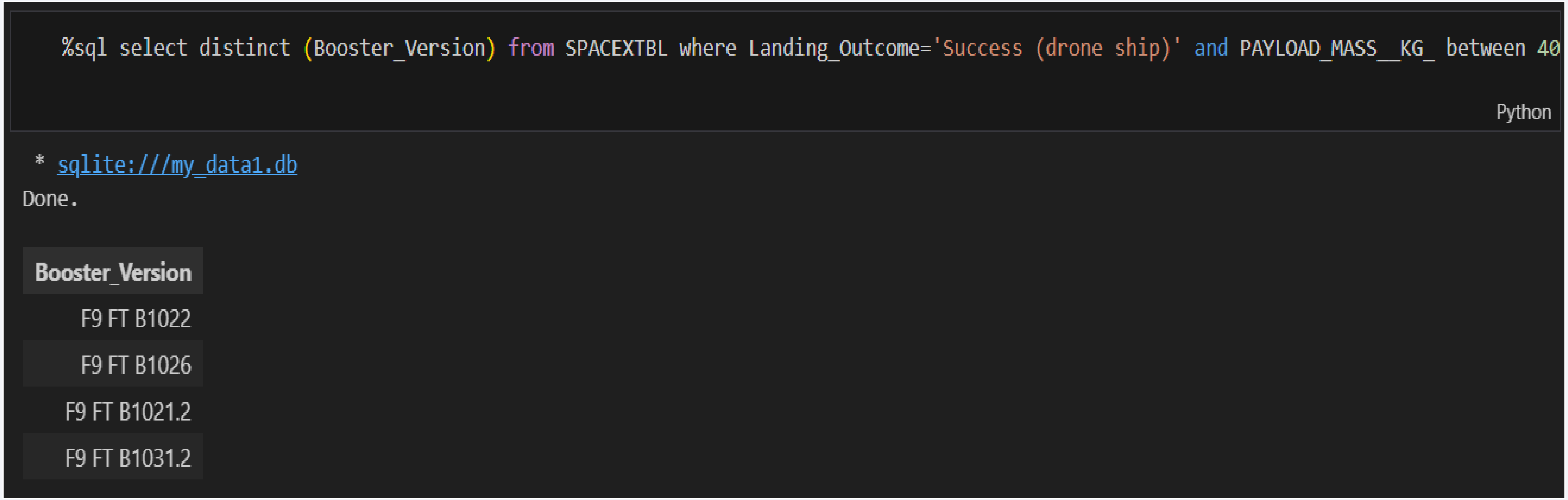
```
%sql select min(Date) from SPACEXTBL where Landing_Outcome='Success (ground pad)'
```

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

min(Date)
2015-12-22

- The rocket first landed on December 22, 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

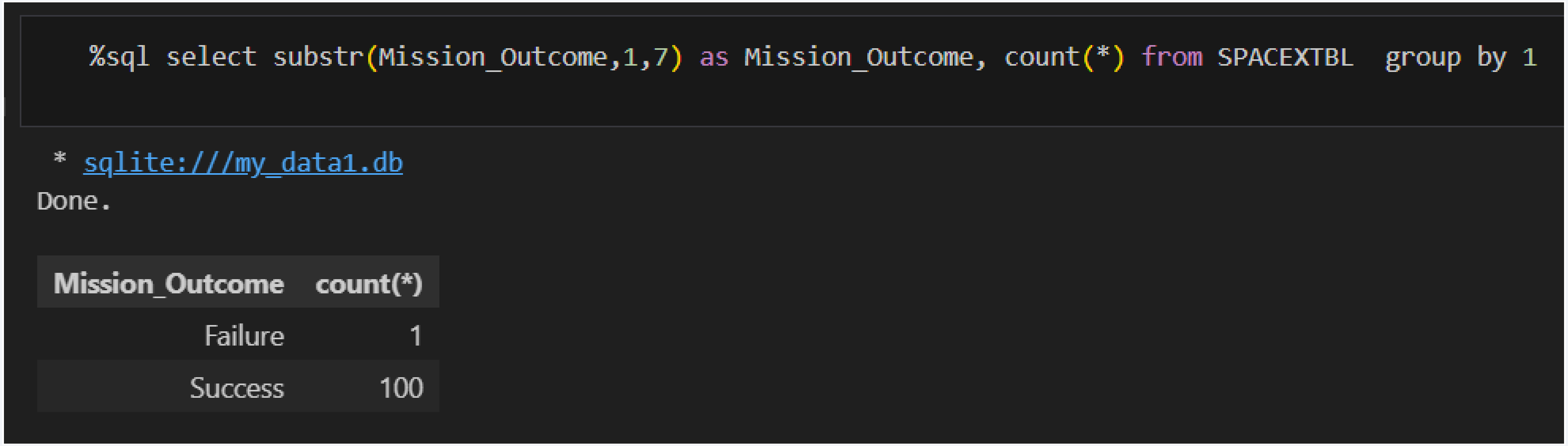


```
%sql select distinct (Booster_Version) from SPACEXTBL where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ between 40
```

```
 * 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



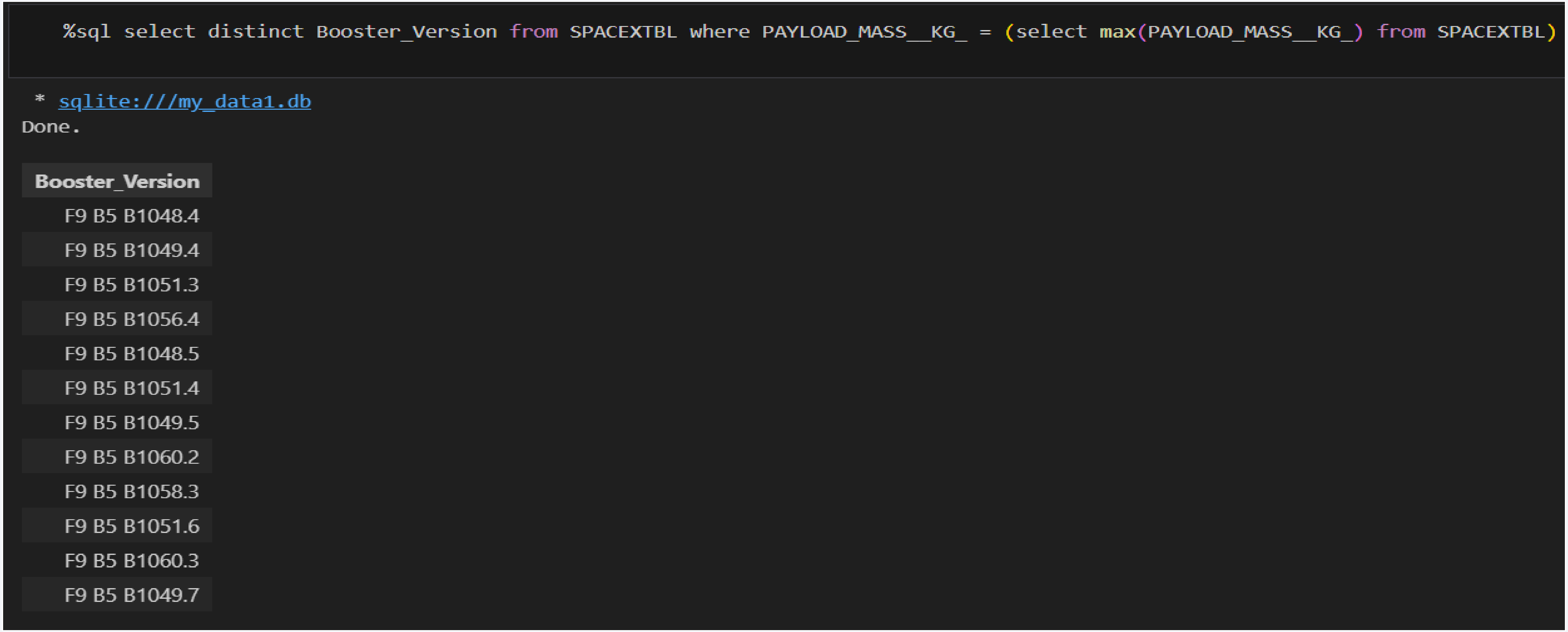
```
%sql select substr(Mission_Outcome,1,7) as Mission_Outcome, count(*) from SPACEXTBL  group by 1
```

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

Mission_Outcome	count(*)
Failure	1
Success	100

- The success mission outcomes were 100 out of 101, whereas the failure was 1.

Boosters Carried Maximum Payload

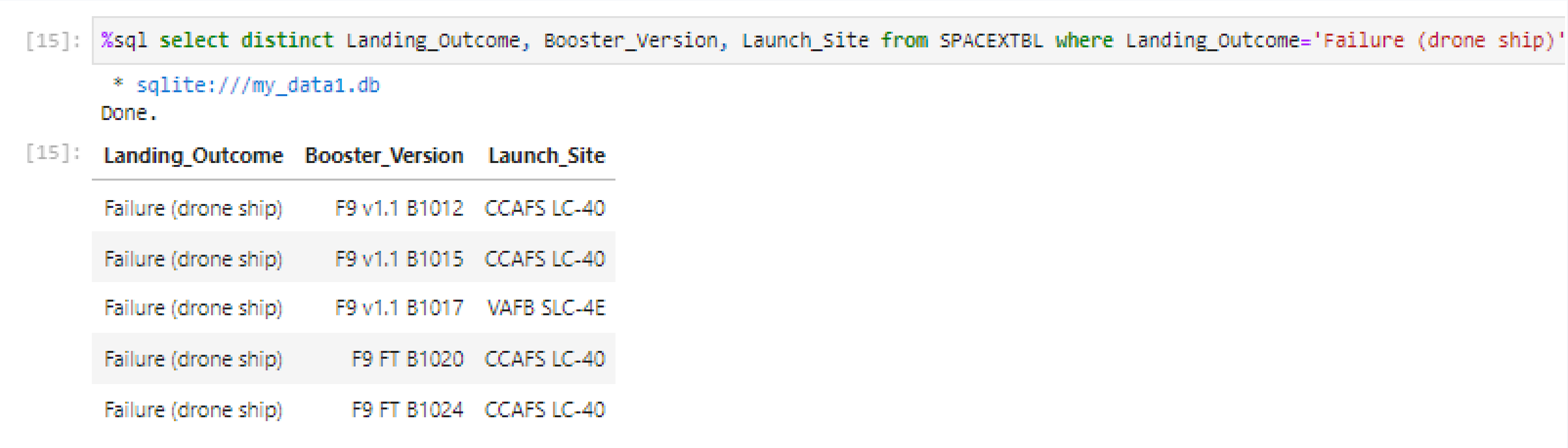


```
%sql select distinct Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * 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



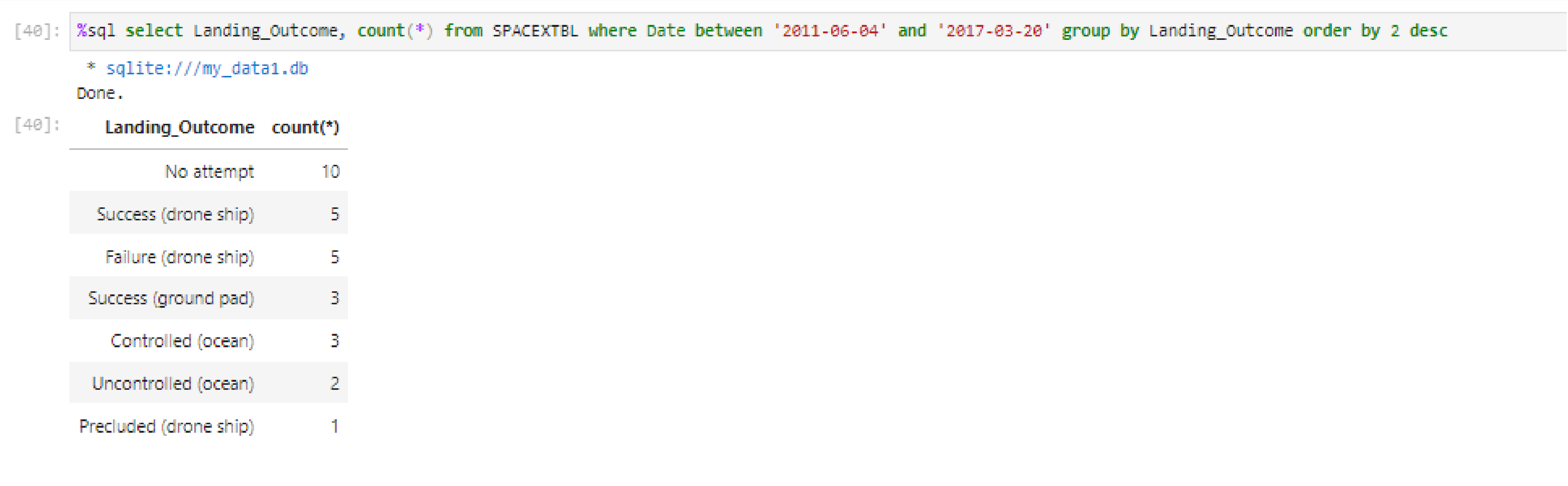
```
[15]: %sql select distinct Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL where Landing_Outcome='Failure (drone ship)'
```

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

[15]:

Landing_Outcome	Booster_Version	Launch_Site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1017	VAFB SLC-4E
Failure (drone ship)	F9 FT B1020	CCAFS LC-40
Failure (drone ship)	F9 FT B1024	CCAFS LC-40

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



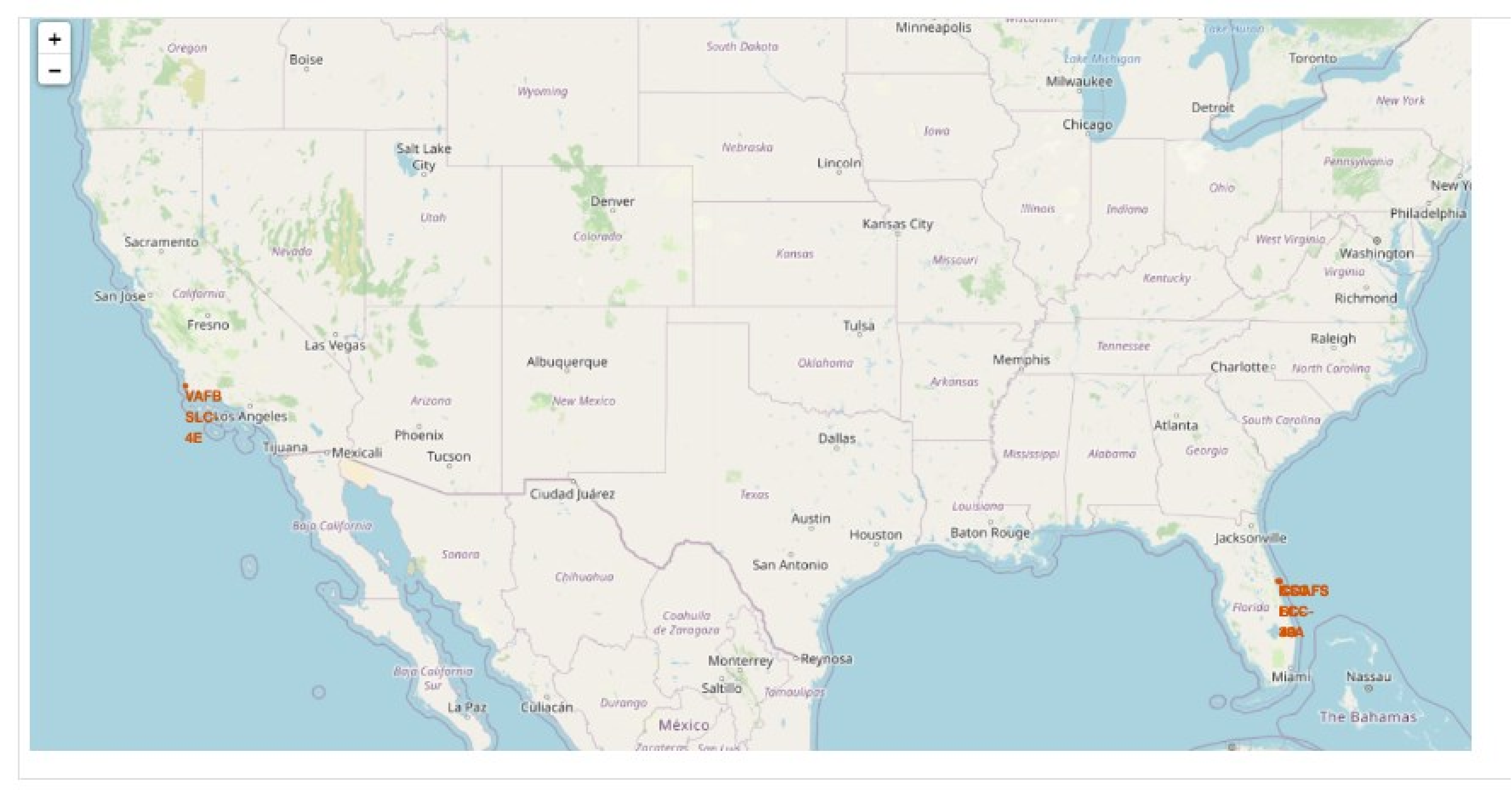
```
[40]: %sql select Landing_Outcome, count(*) from SPACEXTBL where Date between '2011-06-04' and '2017-03-20' group by Landing_Outcome order by 2 desc

 * sqlite:///my_data1.db
Done.
[40]:
```

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

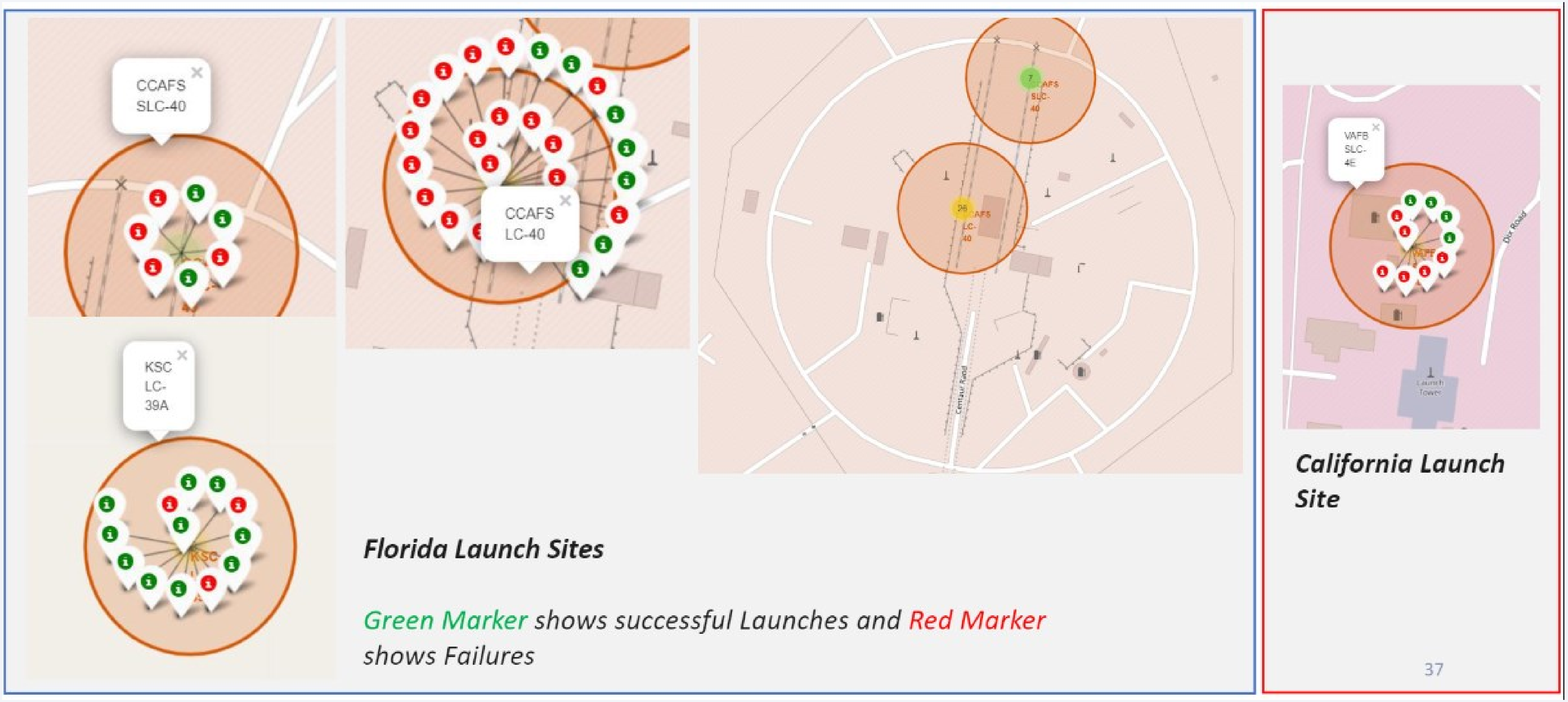
Section 3

Launch Sites Proximities Analysis

All launch sites global map markers



Markers showing launch sites with color labels

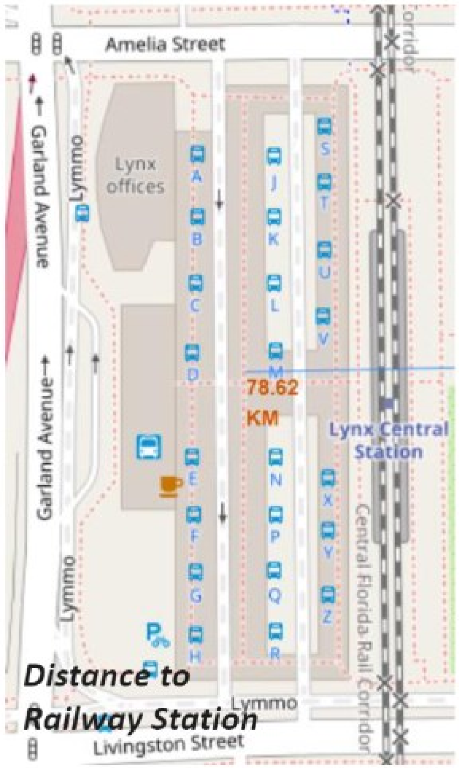


Florida Launch Sites

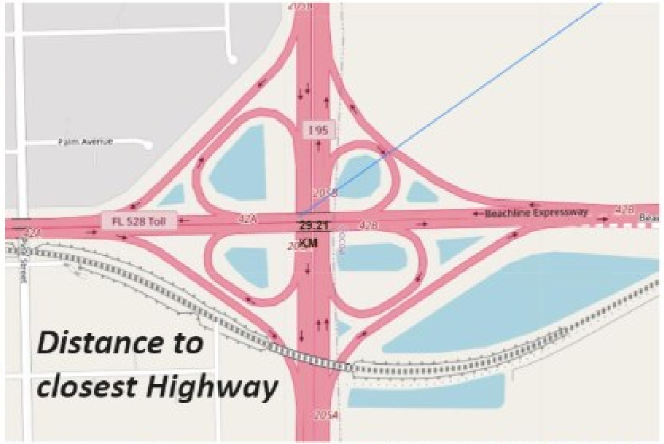
Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

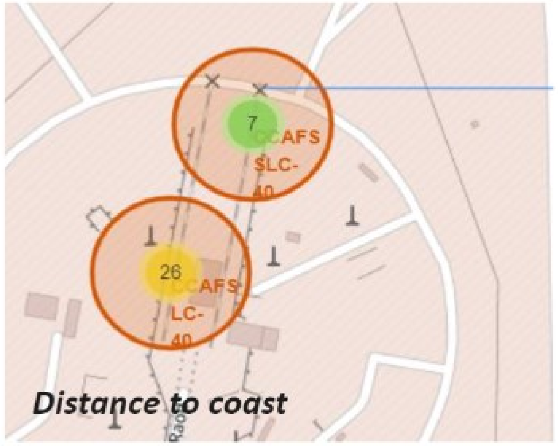
Launch Site distance to landmarks



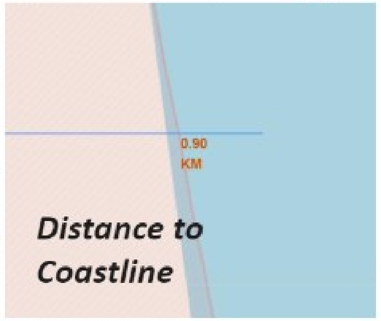
Distance to Railway Station



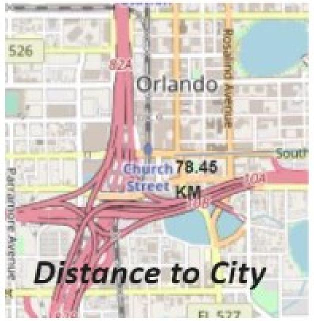
Distance to closest Highway



Distance to coast



Distance to Coastline

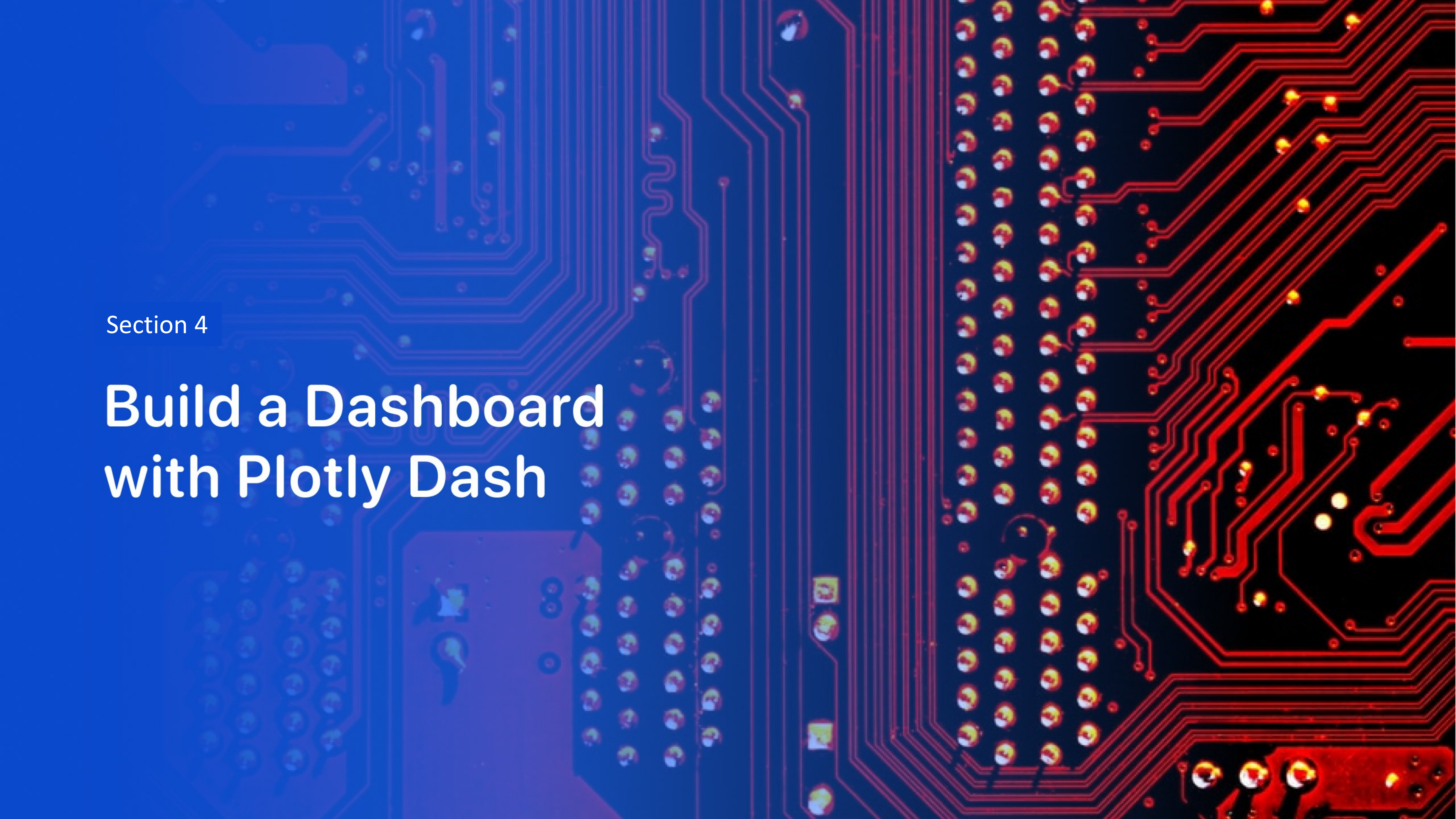


Distance to City

- •Are launch sites in close proximity to railways? No
- •Are launch sites in close proximity to highways? No
- •Are launch sites in close proximity to coastline? Yes
- •Do launch sites keep certain distance away from cities? Yes

Section 4

Build a Dashboard with Plotly Dash



<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title

- Show the screenshot of launch success count for all sites, in a piechart

- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title

- Show the screenshot of the piechart for the launch site with highest launch success ratio

- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title

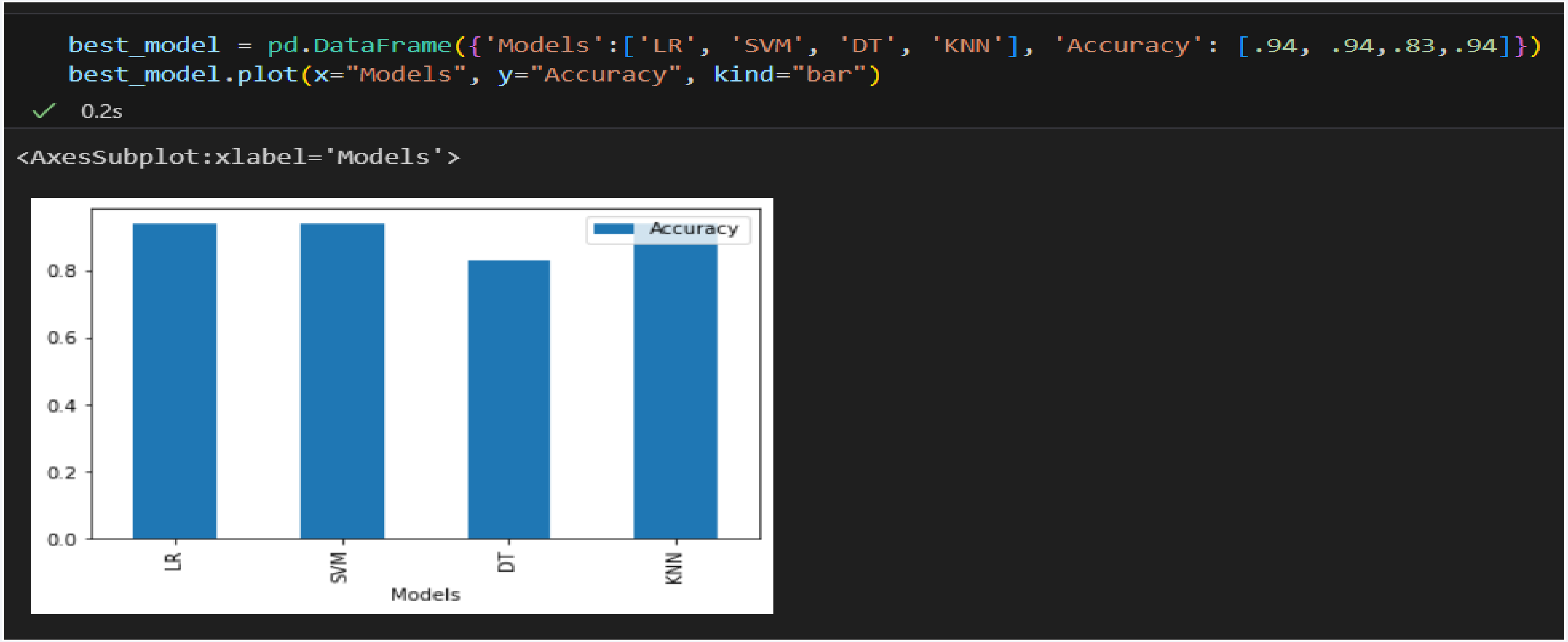
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



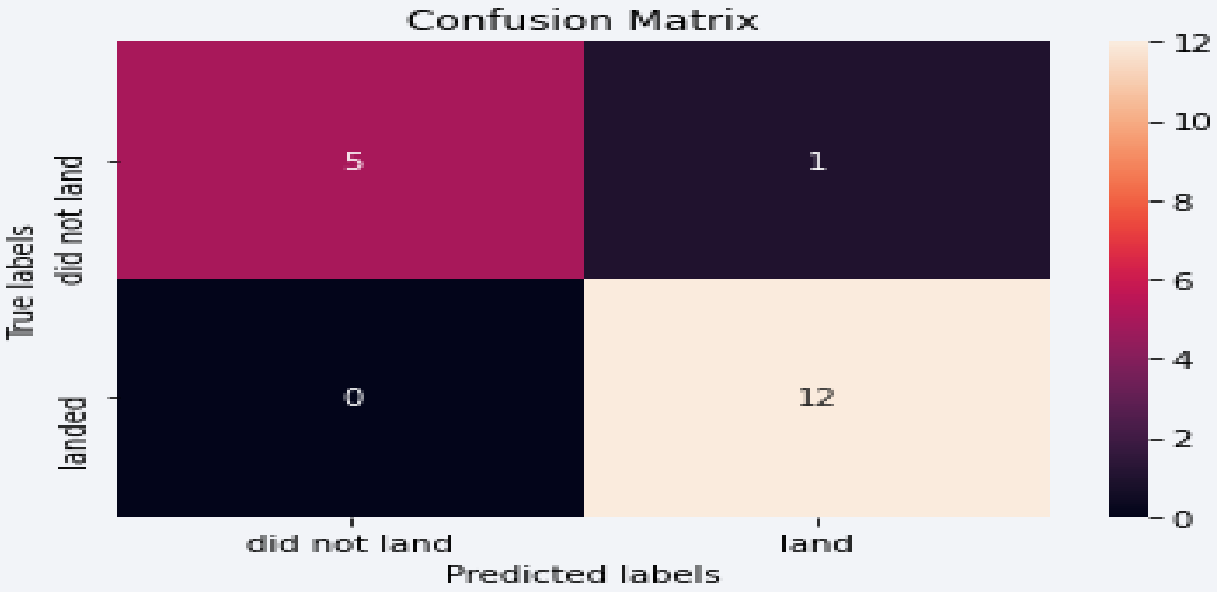
```
best_model = pd.DataFrame({'Models':['LR', 'SVM', 'DT', 'KNN'], 'Accuracy': [.94, .94,.83,.94]})
best_model.plot(x="Models", y="Accuracy", kind="bar")
```

```
✓ 0.2s
```

```
<AxesSubplot:xlabel='Models'>
```

According to the bar chart, LR, SVM, and KNN performed best, followed by DT.

Confusion Matrix



The above confusion matrix is from SVM algorithm. The model predicted all successful landings (no misclassification for landing).

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.
- Based on accuracy, LR, SVM and KNN are the best machine learning algorithm for this task.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!