IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mohammed Abubaker
26.10.2024

Outline

EXECUTIVE SUMMARY

INTRODUCTION

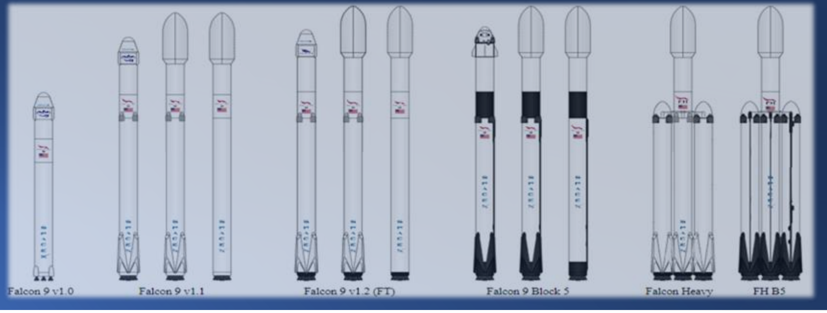
METHODOLOGY

RESULTS

CONCLUSION

APPENDIX

Executive Summary



Summary of methodologies

- Data Collection through Request to the SpaceX API.
- Web Scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia.
- Execute SQL Queries to understand the Spacex Dataset using SQLite Database.
- Exploratory Data Analysis and Feature Engineering using Pandas, Seaborn, and Matplotlib.
- Interactive Visual Analytics using Folium.
- Interactive Dashboard with Ploty Dash.
- Machine Learning Prediction of Successful Landing using Logistic Regression, Support Vector Machine, Decision Tree, and k-Nearest Neighbors.

Summary of all results

- Success landing rate since 2013 kept increasing till 2020.
- There is a relation between success rate and orbit type; ES-L1, GEO, HEO, and SSO have the highest success rates, while GTO has the lowest success rates. However, most of the launches were from GTO orbit.
- All launch sites are in proximity to the Equator line. As well as they are in very close proximity to the coast.
- The launch site KSC LC-39A has the largest successful launches.
- F9 Booster version B5 and FT has the highest launch success rate.
- Payload mass has an effect of the launch success rate, for example, the range 6K-8K kg has the lowest rate.
- Machine Learning achieved a test accuracy of 83.33% in predicting success landing.

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

- Can we predict if the Falcon 9 first stage will land successfully?
- What are the factors that affect the landing success rate?
- Is there a relation between launch site locations and success rate?
- What are the relationships between different features that affect the landing success rate?

Section 1

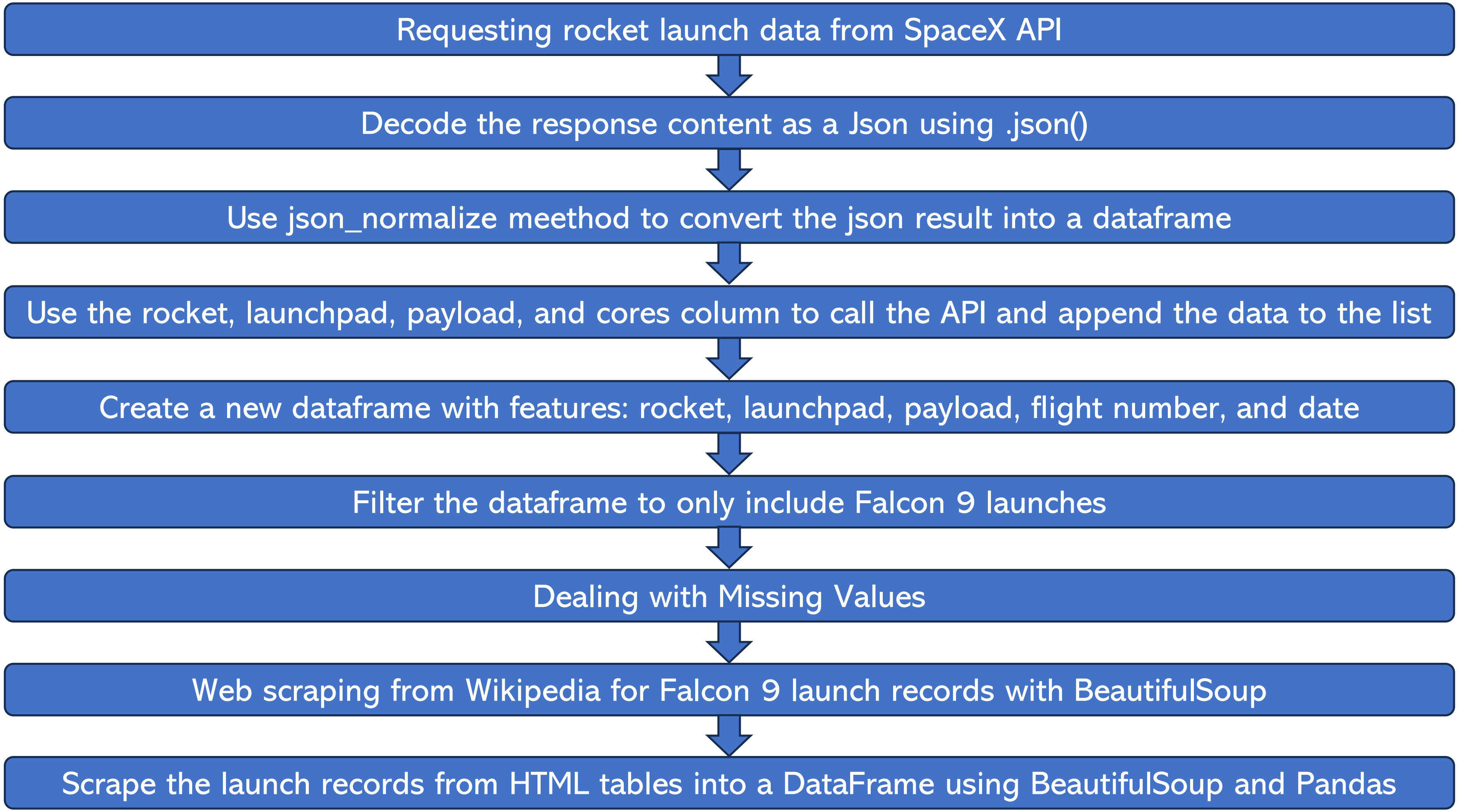
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - Removing NaN column, Standarizing the features, and Applying one-hot encoder to categorical features.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The classifier object was built by fitting on the training data, using GridSearchCV to find best hyperparameters for each model, then the model was evaluating on the test data using accuracy score and confusion matrix.

Data Collection



Requesting rocket launch data from SpaceX API

Decode the response content as a Json using .json()

Use json_normalize meethod to convert the json result into a dataframe

Use the rocket, launchpad, payload, and cores column to call the API and append the data to the list

Create a new dataframe with features: rocket, launchpad, payload, flight number, and date

Filter the dataframe to only include Falcon 9 launches

Dealing with Missing Values

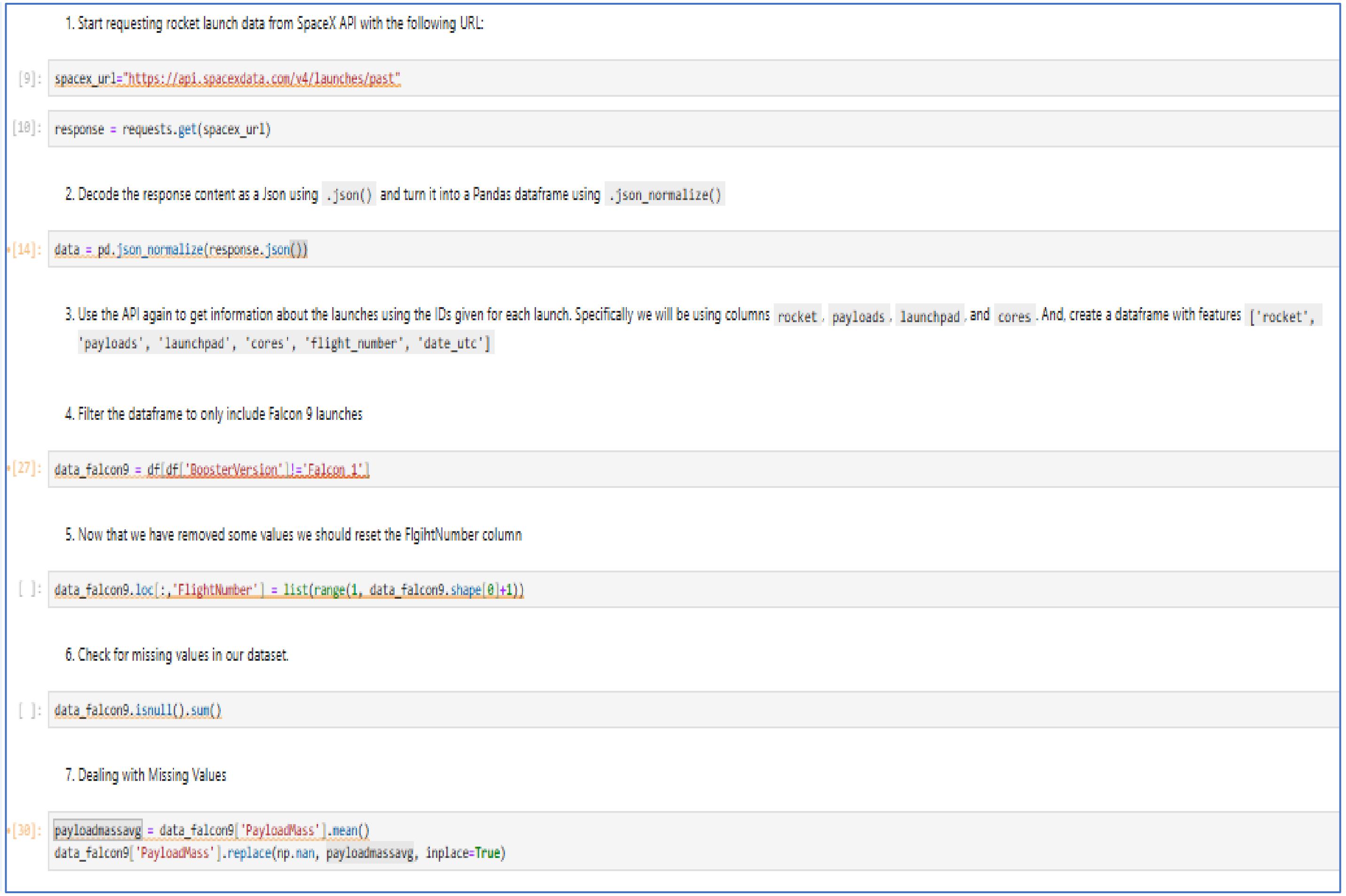
Web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup

Scrape the launch records from HTML tables into a DataFrame using BeautifulSoup and Pandas

Data Collection – SpaceX API

- Performing a get request to the SpaceX API and some basic data wrangling and formatting.

- The GitHub URL of the completed SpaceX API calls notebook:
  - https://github.com/mba-github32/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Lab1-spacex-data-collection-api.ipynb



1. Start requesting rocket launch data from SpaceX API with the following URL:

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

```
[10]: response = requests.get(spacex_url)
```

2. Decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
[14]: data = pd.json_normalize(response.json())
```

3. Use the API again to get information about the launches using the IDs given for each launch. Specifically we will be using columns `rocket`, `payloads`, `launchpad`, and `cores`. And, create a dataframe with features `['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']`

4. Filter the dataframe to only include Falcon 9 launches

```
[27]: data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

5. Now that we have removed some values we should reset the FlgihtNumber column

```
[ ]: data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
```

6. Check for missing values in our dataset.

```
[ ]: data_falcon9.isnull().sum()
```

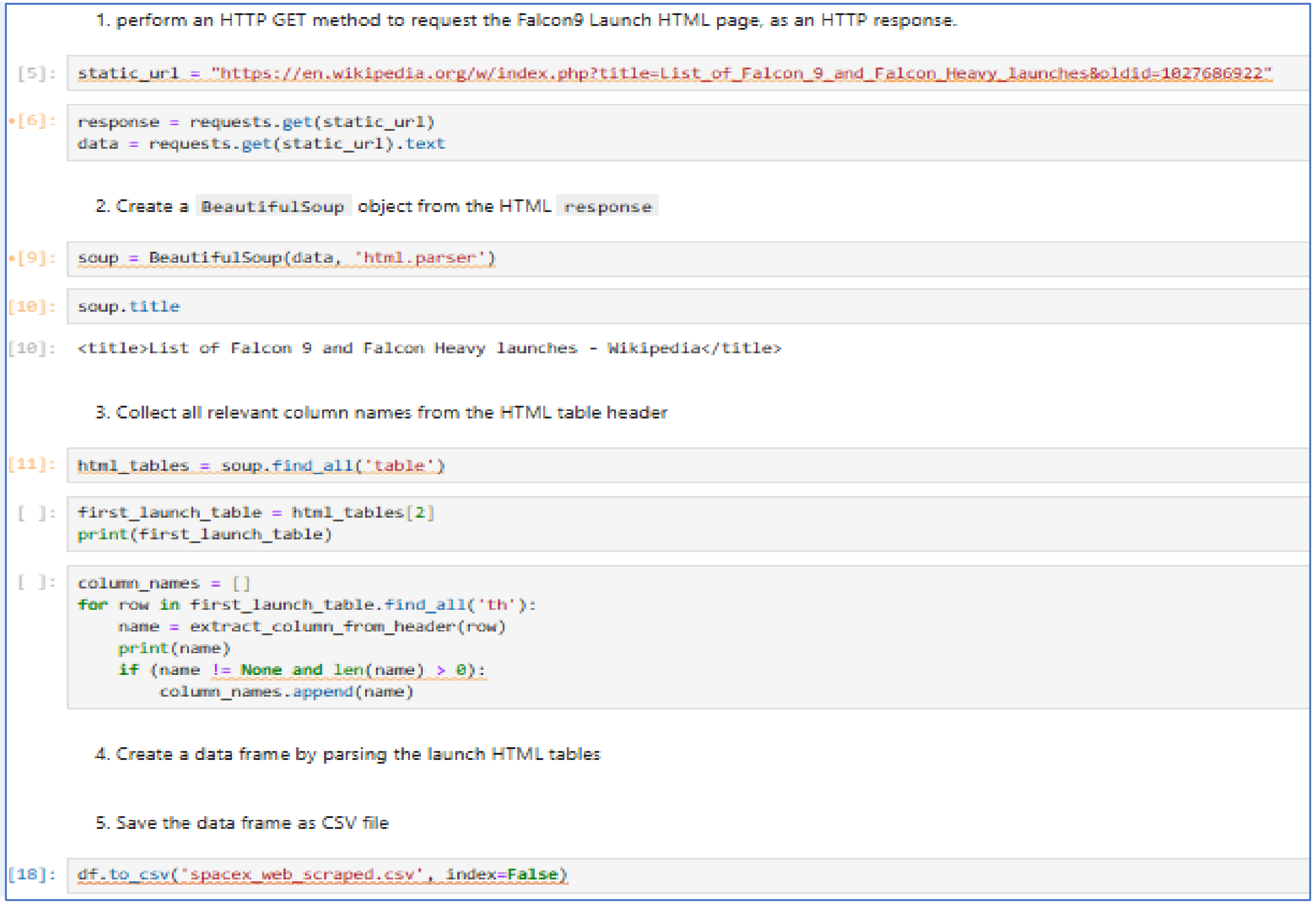
7. Dealing with Missing Values

```
[30]: payloadmassavg = data_falcon9['PayloadMass'].mean()
      data_falcon9['PayloadMass'].replace(np.nan, payloadmassavg, inplace=True)
```

Data Collection - Scraping

- Applying web scrapping to webscrap Falcon 9 launch records with BeautifulSoup, then parsing the HTML table and converting it into a pandas data frame.

- The GitHub URL of the completed scraping notebook:
  - https://github.com/mba-github32/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Lab1-spacex-webscraping.ipynb



1. perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

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

[6]: response = requests.get(static_url)
     data = requests.get(static_url).text
```

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

```
[9]: soup = BeautifulSoup(data, 'html.parser')

[10]: soup.title

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

3. Collect all relevant column names from the HTML table header

```
[11]: html_tables = soup.find_all('table')

[ ]: first_launch_table = html_tables[2]
     print(first_launch_table)

[ ]: column_names = []
     for row in first_launch_table.find_all('th'):
         name = extract_column_from_header(row)
         print(name)
         if (name != None and len(name) > 0):
             column_names.append(name)
```

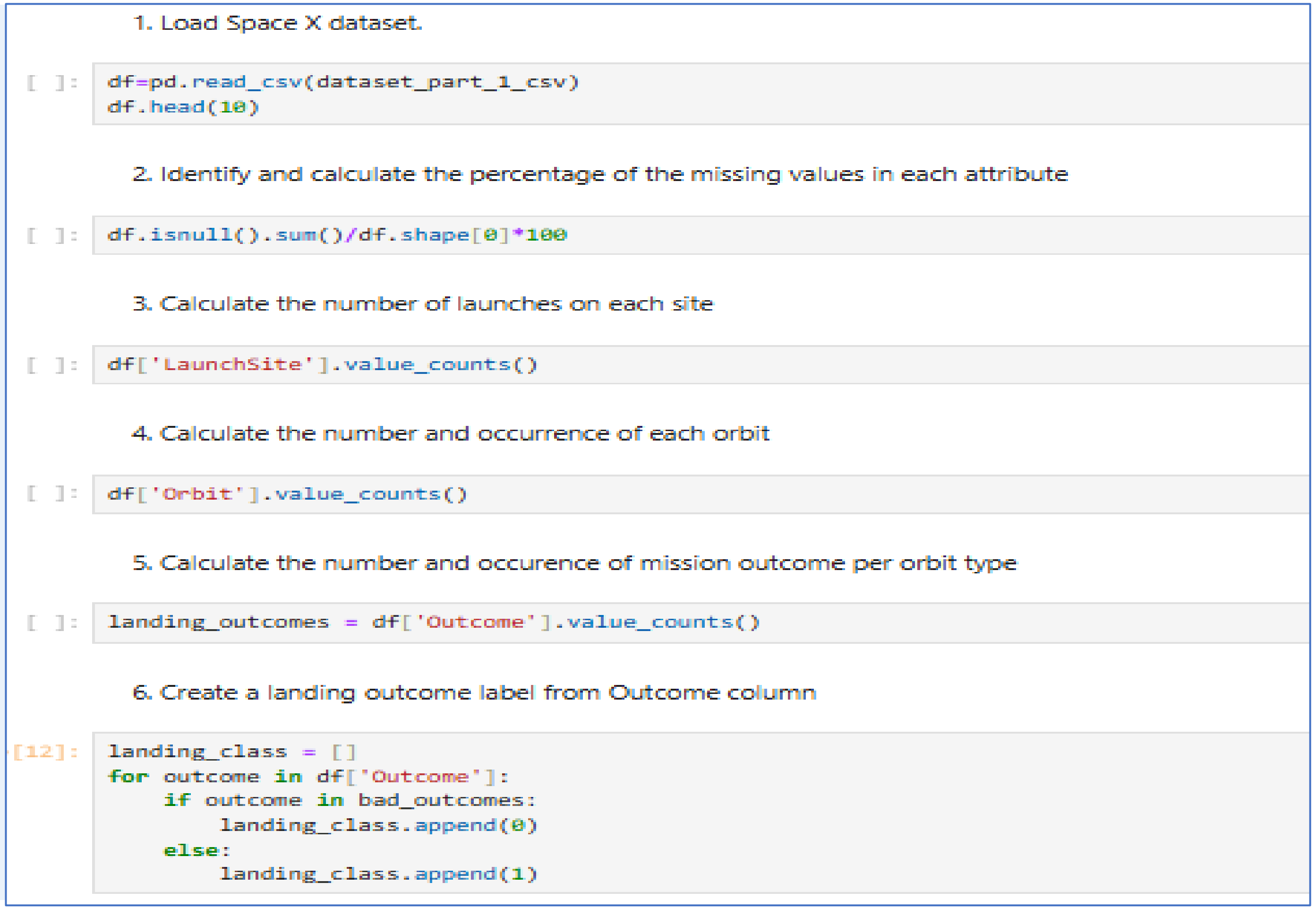
4. Create a data frame by parsing the launch HTML tables

5. Save the data frame as CSV file

```
[18]: df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

- Performing exploratory Data Analysis and determining Training Labels.
- Calculating the number of launches on each site, the number and occurrence of each orbit, and the number and occurrence of mission outcome per orbit type.
- Creating a landing outcome label from Outcome column.
- The GitHub URL of the completed data wrangling notebook:
  - https://github.com/mba-github32/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Lab2-spacex-data%20wrangling.ipynb



1. Load Space X dataset.

```
[ ]: df=pd.read_csv(dataset_part_1_csv)
     df.head(10)
```

2. Identify and calculate the percentage of the missing values in each attribute

```
[ ]: df.isnull().sum()/df.shape[0]*100
```

3. Calculate the number of launches on each site

```
[ ]: df['LaunchSite'].value_counts()
```

4. Calculate the number and occurrence of each orbit

```
[ ]: df['Orbit'].value_counts()
```

5. Calculate the number and occurence of mission outcome per orbit type

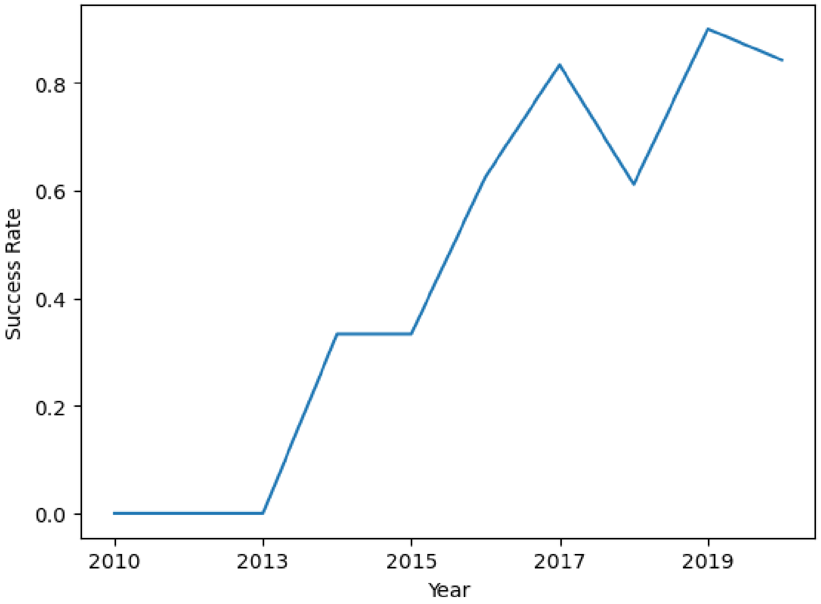
```
[ ]: landing_outcomes = df['Outcome'].value_counts()
```

6. Create a landing outcome label from Outcome column

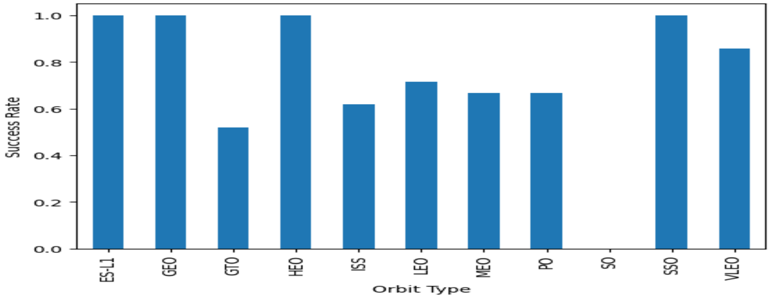
```
[12]: landing_class = []
      for outcome in df['Outcome']:
          if outcome in bad_outcomes:
              landing_class.append(0)
          else:
              landing_class.append(1)
```

EDA with Data Visualization

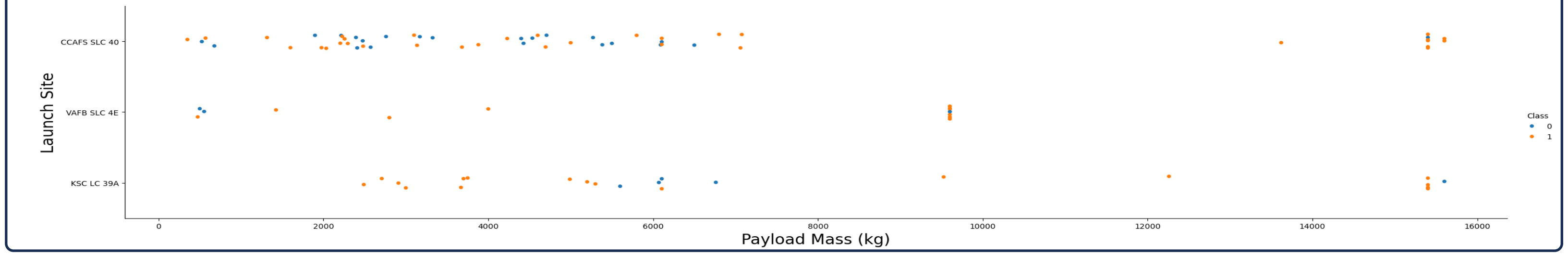
- A line chart with x axis to be Year and y axis to be average success rate, to get the average launch success trend. (used with time series data)



- A bar chart to show success rate of each orbit type. (used to show count for each category)



- A scatter chart to visualize the relationship between Payload Mass and Launch Site. (used to show relationship between two variables)



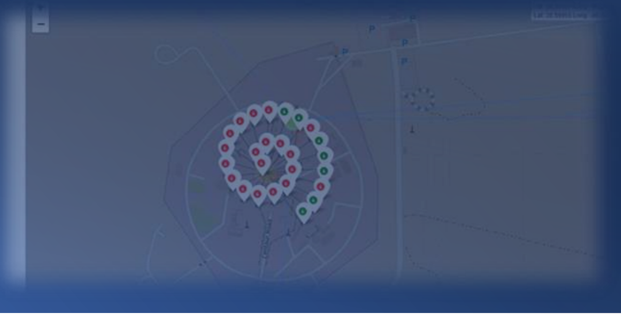
- The GitHub URL of the completed data visualization notebook:
  - https://github.com/mba-github32/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Lab4-edadataviz.ipynb

EDA with SQL

- Load the SQL extension into Jupyter notebook and establish a connection with the SQLite3 database.
- Load the SpaceX dataset (CSV file) into the corresponding table (SPACEXTBL).
- Create table SPACEXTABLE to remove blank rows from SPACEXTBL table.
- %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE ➔ to display the names of the unique launch sites in the space mission.
- Other SQL to perform different data analysis such as:
  - Display 5 records where launch sites begin with the string 'KSC'
  - Display average payload mass carried by booster version F9 v1.1
  - List the date where the successful landing outcome in drone ship was achieved.
  - List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000.
  - And others can be found in the provided github link below.

- The GitHub URL of the completed EDA with SQL notebook:
  - https://github.com/mba-github32/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Lab3-eda-sqlite.ipynb

Build an Interactive Map with Folium



- We first need to create a folium Map object, with an initial center location to be NASA Johnson Space Center at Houston, Texas.
- We used folium.Circle object to add a highlighted circle area with a text label for each launch site in data frame.
- We used folium.Marker for each launch site on the site map to customized the style font and color.
- We enhanced the map by adding the launch outcomes for each site, by creating MarkerCluster object:
  - For each launch result in the data frame, we added a folium.Marker to marker_cluster
- We explores and analyzed the proximities of launch sites:
  - We added a MousePosition on the map to get coordinate for a mouse over a point on the map.
  - We Created a marker with distance to a closest city, railway, highway from CCAFS SLC-40 launch site.
  - We found that the launch sites are in close proximity to coastline and keep certain distance away from cities.

- The GitHub URL of the completed Interactive Map with Folium notebook:
  - https://github.com/mba-github32/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Lab5_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We created an interactive web-based Dashboard with Plotly Dash with a title ‘SpaceX Launch Records Dashboard’.
- We added a dropdown list to enable Launch Site selection:
  - Upon selecting a certain site or all sites, the pie chart appears to show Total Success Launches.
- We added a payload slider to select Payload Mass (Kg) dynamically:
  - Upon choosing a certain payload mass, a scatter chart interactively changed to reflect the ‘Correlation between Payload and Success for Site’ for the different booster version.

- The GitHub URL of the completed Dashboard with Plotly Dash notebook:
  - https://github.com/mba-github32/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Lab6_spacex_dash_app.py

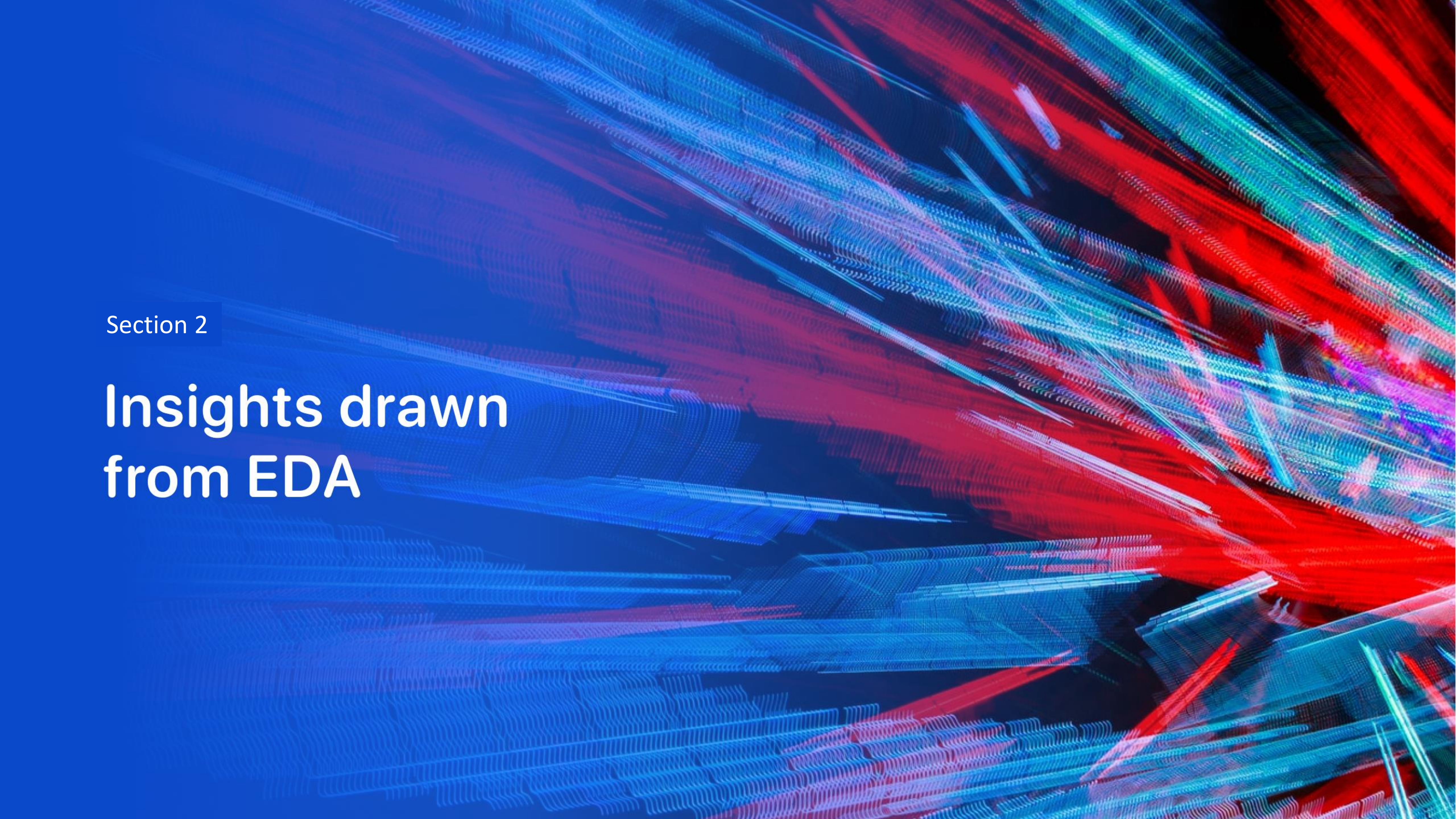
Predictive Analysis (Classification)

- Perform exploratory Data Analysis and determine Training Labels:
  - create a column for the class
  - Standardize the data
  - Split into training data and test data
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression using Grid SearchCV, then train the classifier with these hyperparameter on training set.
- Evaluate the model on test set by looking at its confusion matrix and Calculating its accuracy.

- The GitHub URL of the completed Predictive Analysis (Classification) notebook:
  - https://github.com/mba-github32/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Lab7_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

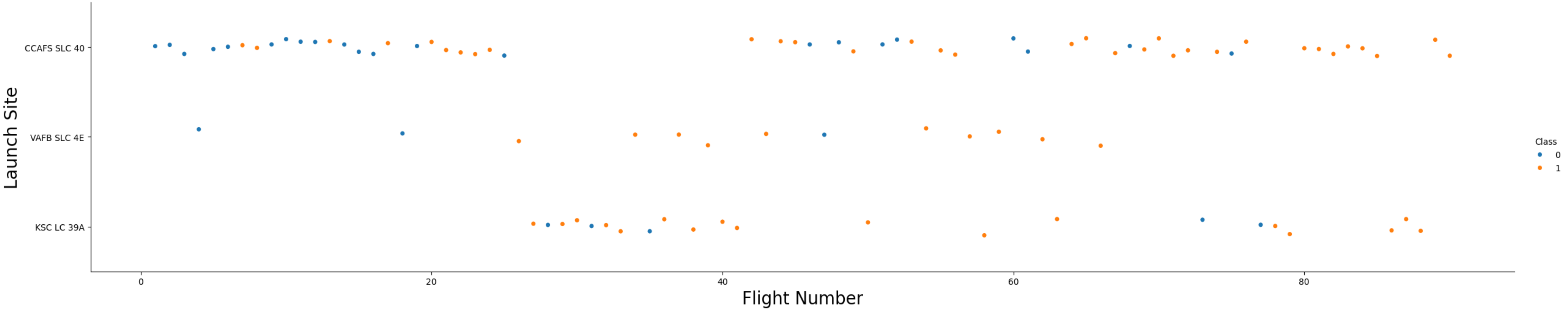


Section 2

Insights drawn from EDA

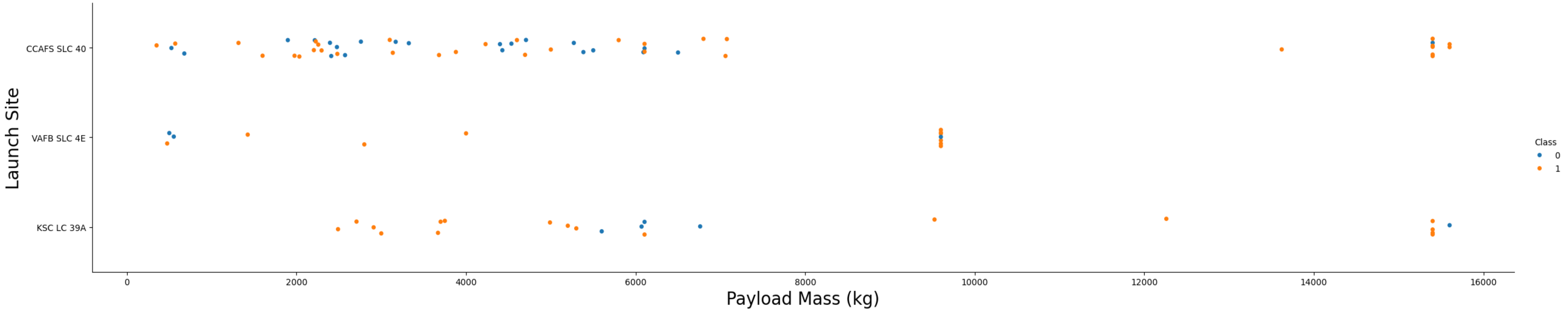
Flight Number vs. Launch Site

- Form the scatter plot, as the flight number increases, the higher success rate achieved at launch sites.



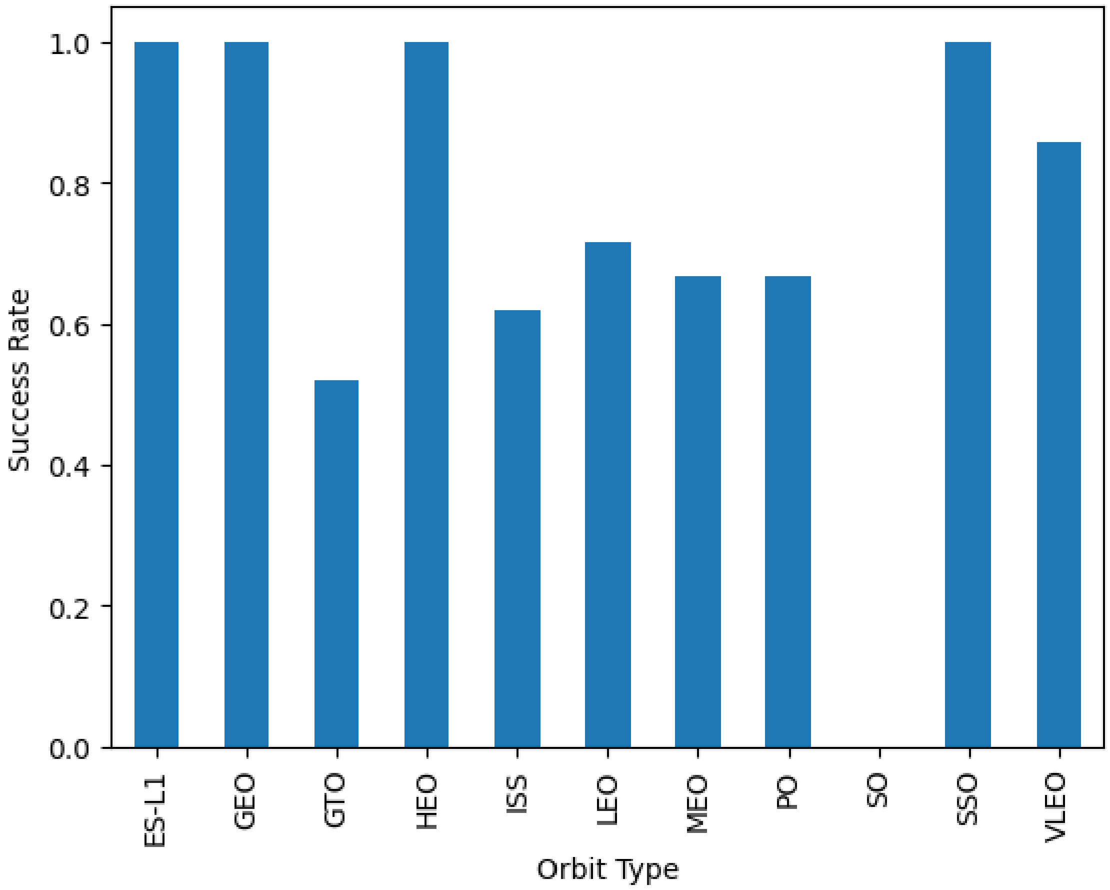
Payload vs. Launch Site

- For the VAFB SLC 4E launch site there are no rockets launched for heavy payload mass (greater than 10k Kg).
- For KSC LC 39A, the most successful rate occurs when the payload between 2k and 6k Kg.
- For CCAFS SLC 40, the success rate increases as the payload is greater than 6k Kg.



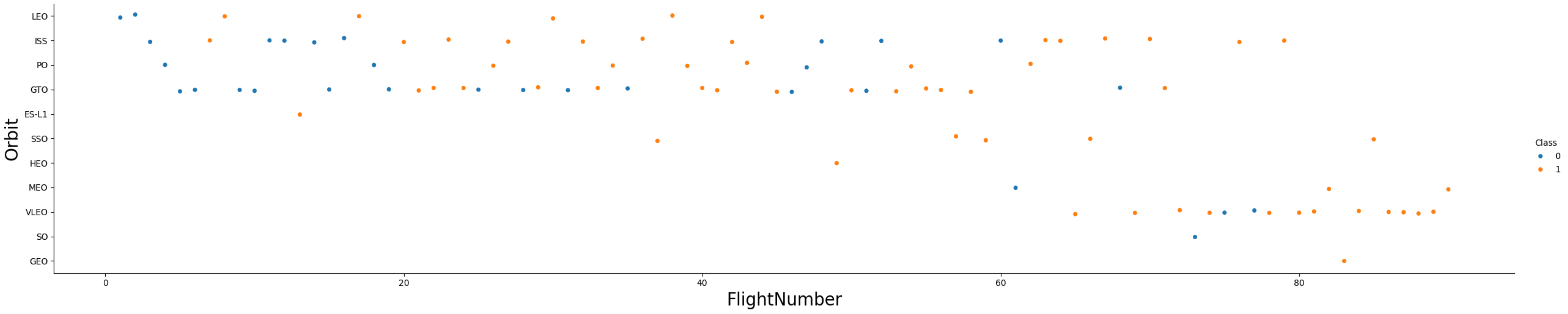
Success Rate vs. Orbit Type

- From the bar chart, ES-L1, GEO, HEO, and SSO have the highest success rates, while GTO has the lowest success rates.



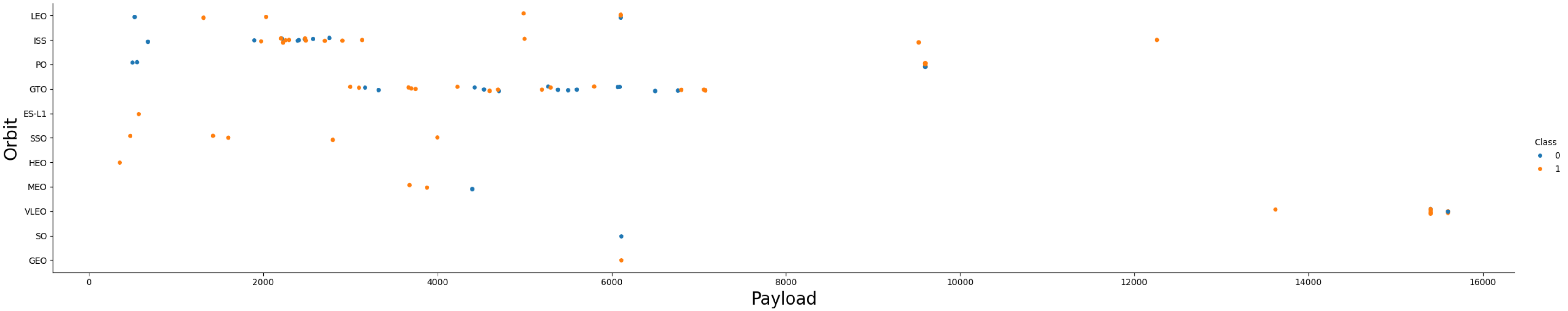
Flight Number vs. Orbit Type

- From the scatter plot, the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.



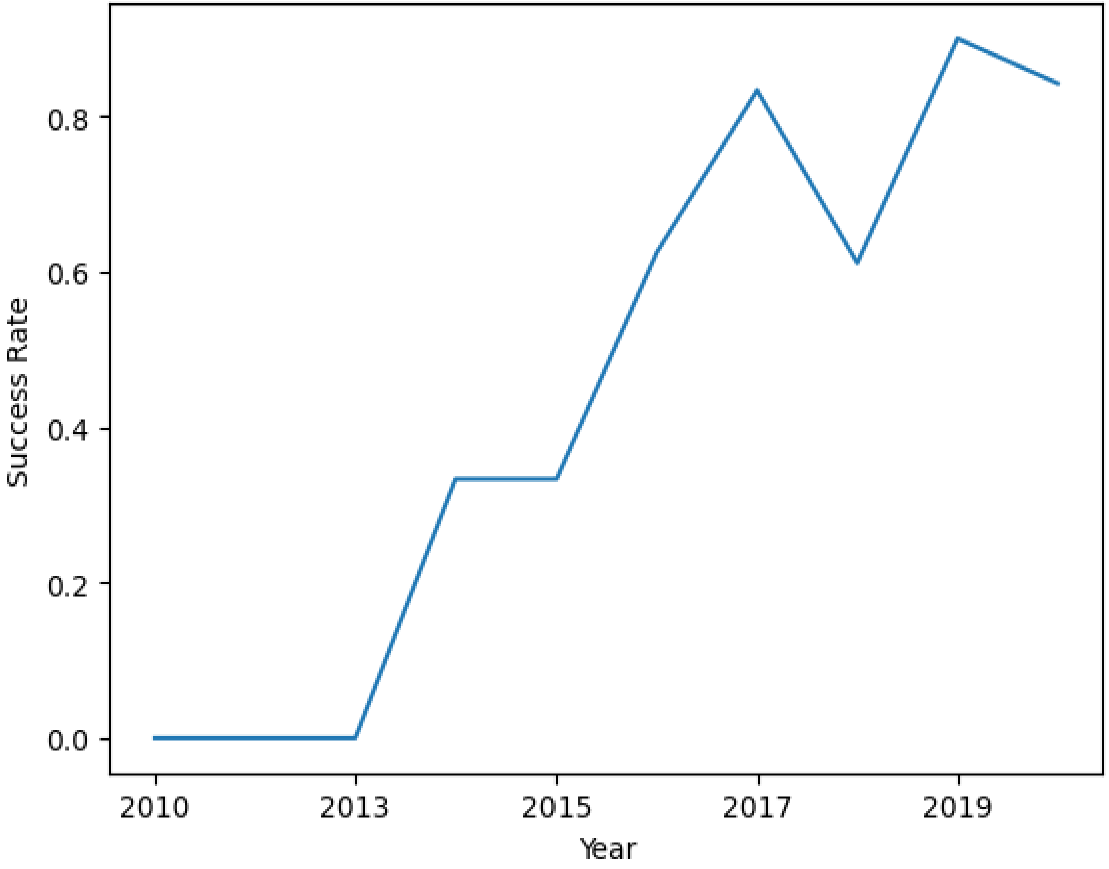
Payload vs. Orbit Type

- From the scatter plot, with heavy payloads the successful landing rate are more for Polar, LEO and ISS. However, for GTO, it is difficult to distinguish between successful and unsuccessful landings as both outcomes are present.



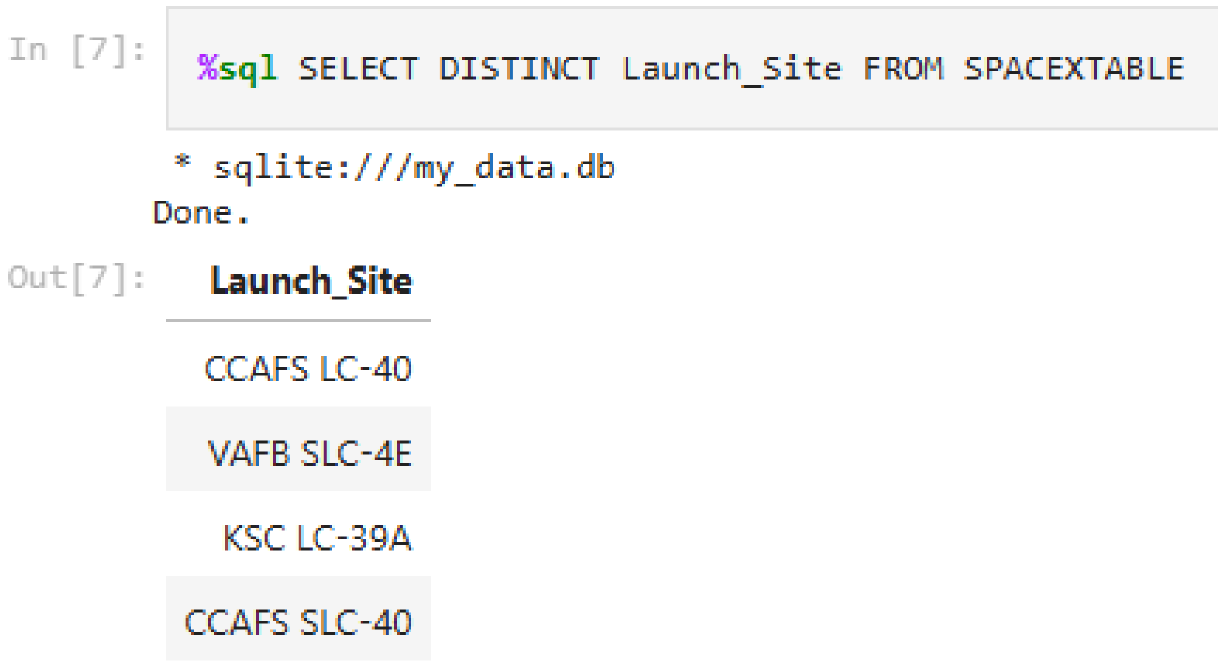
Launch Success Yearly Trend

- From the line chart, we can observe that the success rate since 2013 kept increasing till 2020.



All Launch Site Names

- We used the magic sql to execute the SELECT query within the jupyter notebook.
- The keyword DISTINCT helps to retrieved the unique Launch Sites stored in the dataset.



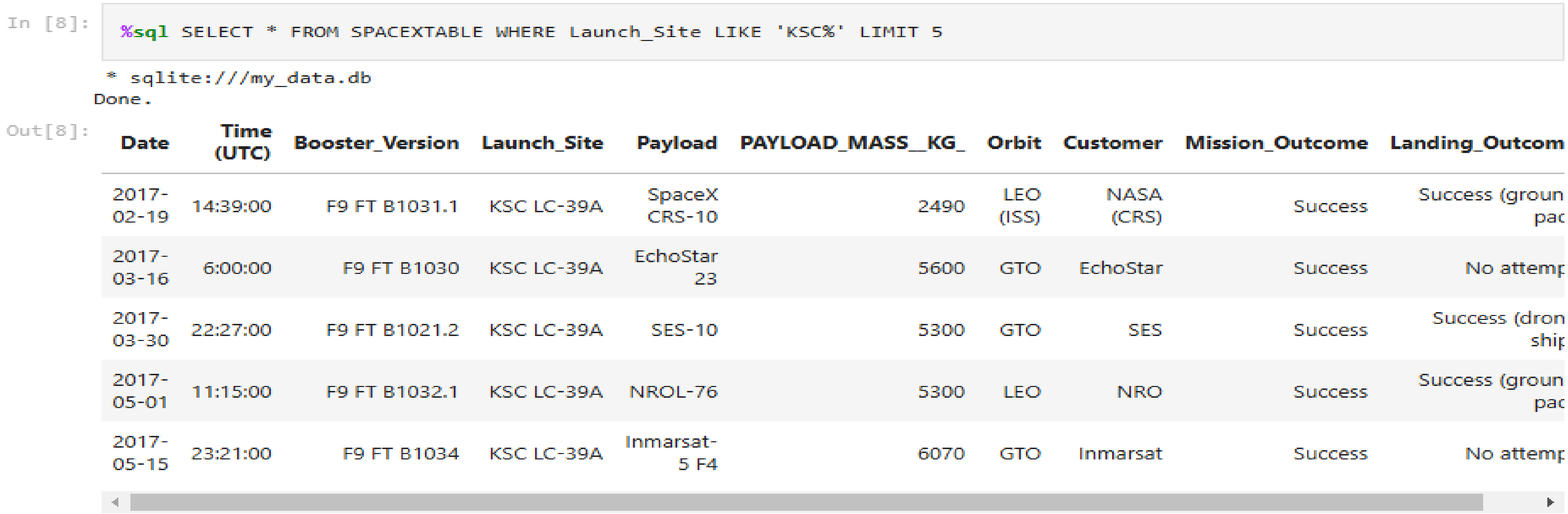
```
In [7]:  %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

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

```
Out[7]:   Launch_Site
         CCAFS LC-40
         VAFB SLC-4E
         KSC LC-39A
        CCAFS SLC-40
```

Launch Site Names Begin with 'KSC'

- The LIKE operator is used in a WHERE clause to search for the pattern ‘KSC%’, where % sign wrote at the end to indicate that we are searching for the site that begins with ‘KSC’. The LIMIT 5 is used to show the first 5 records retrieved form the SELECT query.



```
In [8]: %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'KSC%' LIMIT 5
```

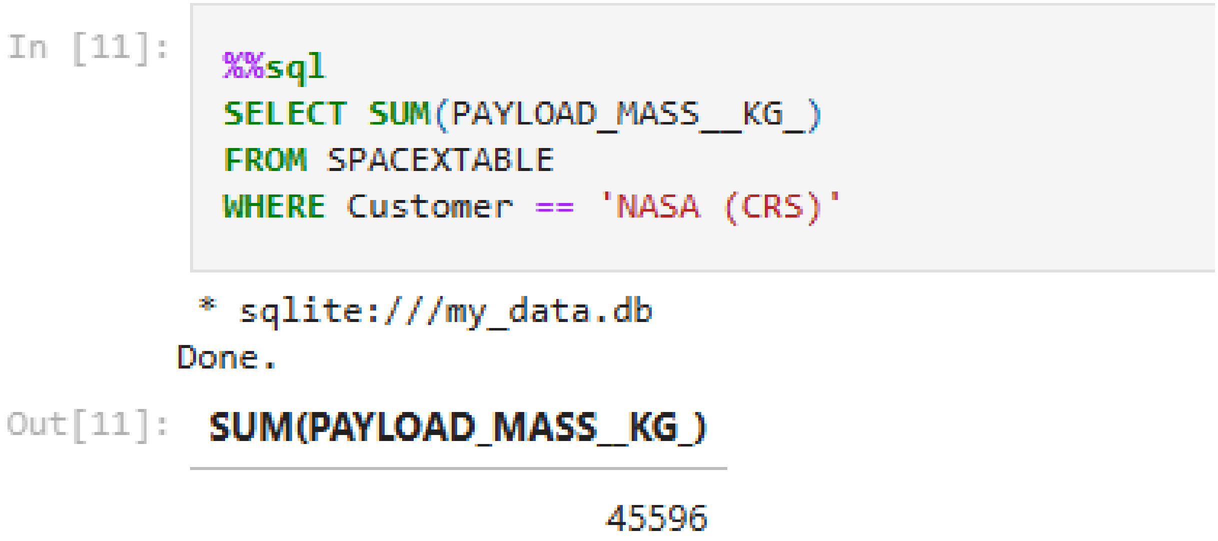
```
 * sqlite:///my_data.db
Done.
```

Out[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcom
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (groun pac
2017-03-16	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attemp
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (dron ship
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (groun pac
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attemp

Total Payload Mass

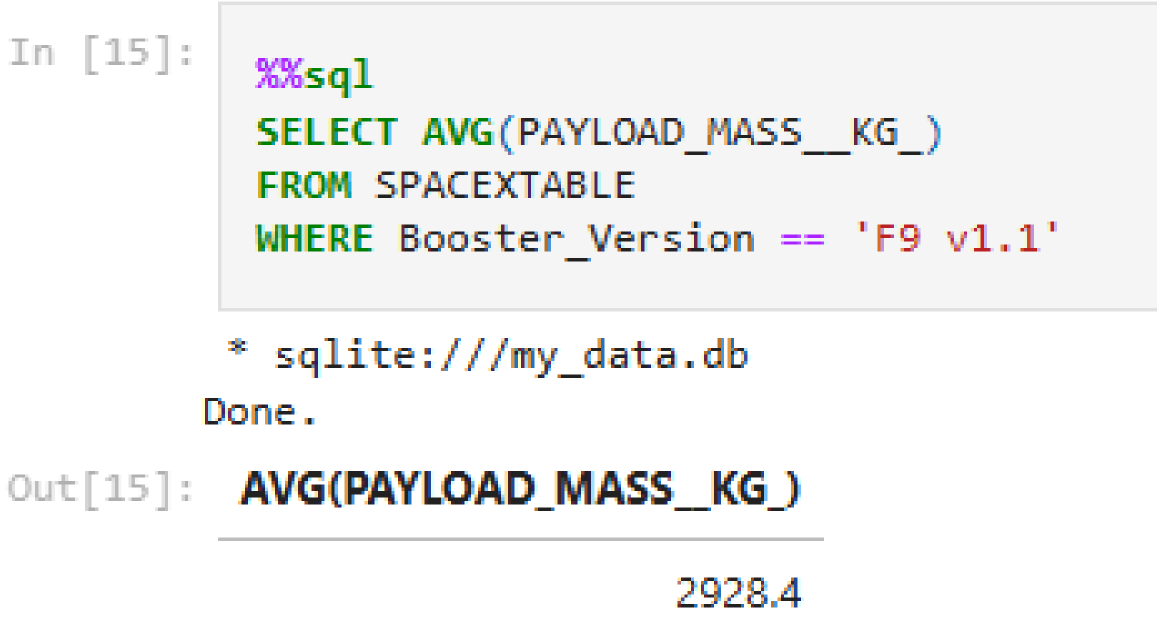
- We used the SUM aggregate function to calculate the total payload mass of ‘PAYLOAD_MASS__KG_’ column.
- The results tells that the total payload mass of NASA (CRS) is 45,596 Kg.



```
In [11]: %%sql
         SELECT SUM(PAYLOAD_MASS__KG_)
         FROM SPACEXTABLE
         WHERE Customer == 'NASA (CRS)'

          * sqlite:///my_data.db
         Done.

Out[11]: SUM(PAYLOAD_MASS__KG_)
                          45596
```

Average Payload Mass by F9 v1.1

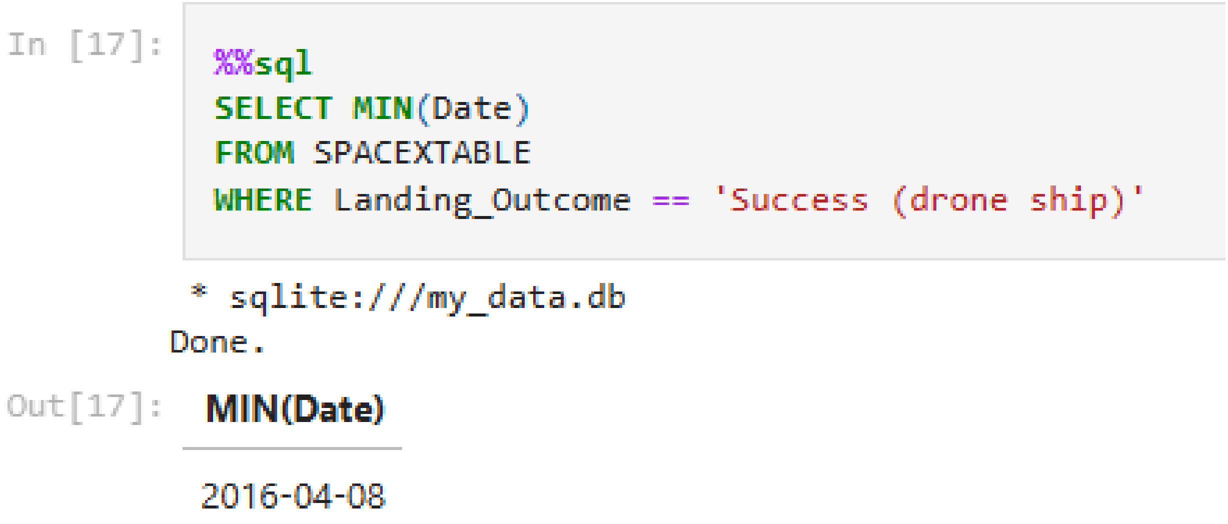
- We used the AVG aggregate function to calculate the average payload mass carried by booster version F9 v1.1.
- The results tells that this average is 2,928.4 Kg.



```
In [15]:  %%sql
          SELECT AVG(PAYLOAD_MASS__KG_)
          FROM SPACEXTABLE
          WHERE Booster_Version == 'F9 v1.1'

           * sqlite:///my_data.db
          Done.

Out[15]:  AVG(PAYLOAD_MASS__KG_)
                          2928.4
```

First Successful Ground Landing Date

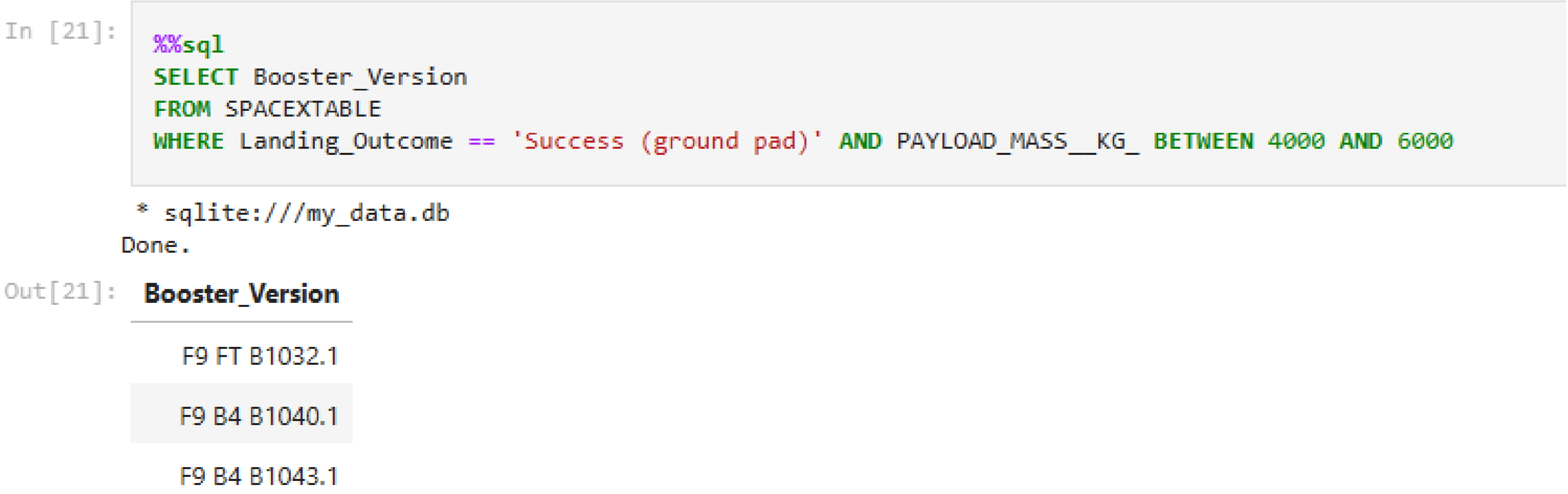
- We used the MIN aggregate function on ‘Date’ column to find the date of the first successful landing outcome on drone ship.
- The results tells that this date is 2016-04-08.



```
In [17]:  %%sql
          SELECT MIN(Date)
          FROM SPACEXTABLE
          WHERE Landing_Outcome == 'Success (drone ship)'

           * sqlite:///my_data.db
          Done.

Out[17]:  MIN(Date)
          2016-04-08
```

Successful Ground Pad Landing with Payload between 4000 and 6000

- In the WHERE clause, we filtered the Booster to have a Landing Outcome to be ‘Success (ground pad)’ at the same time we used BETWEEN to consider payload mass to be greater than 4000 but less than 6000.



In [21]:

```
%%sql
SELECT Booster_Version
FROM SPACEXTABLE
WHERE Landing_Outcome == 'Success (ground pad)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

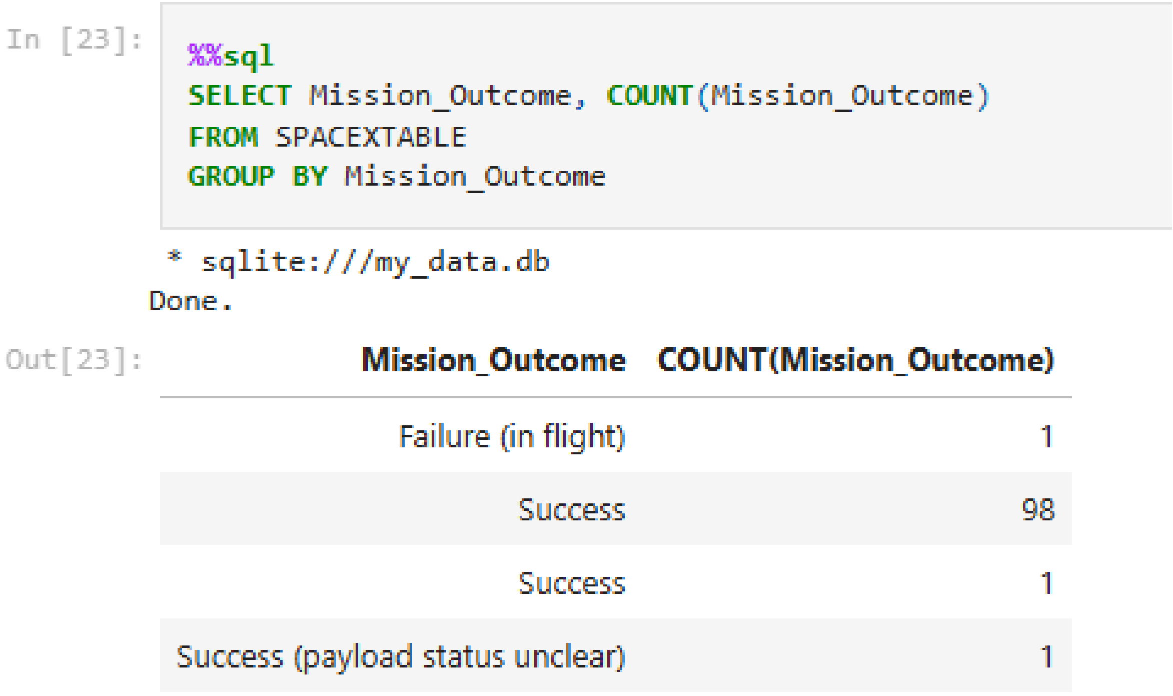
```
 * sqlite:///my_data.db
Done.
```

Out[21]:

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

- We used the COUNT aggregate function on ‘Mission_Outcome’ column to calculate the total number of successful and failure mission outcomes.
- It is important to use the GROUP BY clause such that the count will be calculated based on grouping similar Mission_Outcome.



In [23]:

```
%%sql
SELECT Mission_Outcome, COUNT(Mission_Outcome)
FROM SPACEXTABLE
GROUP BY Mission_Outcome
```

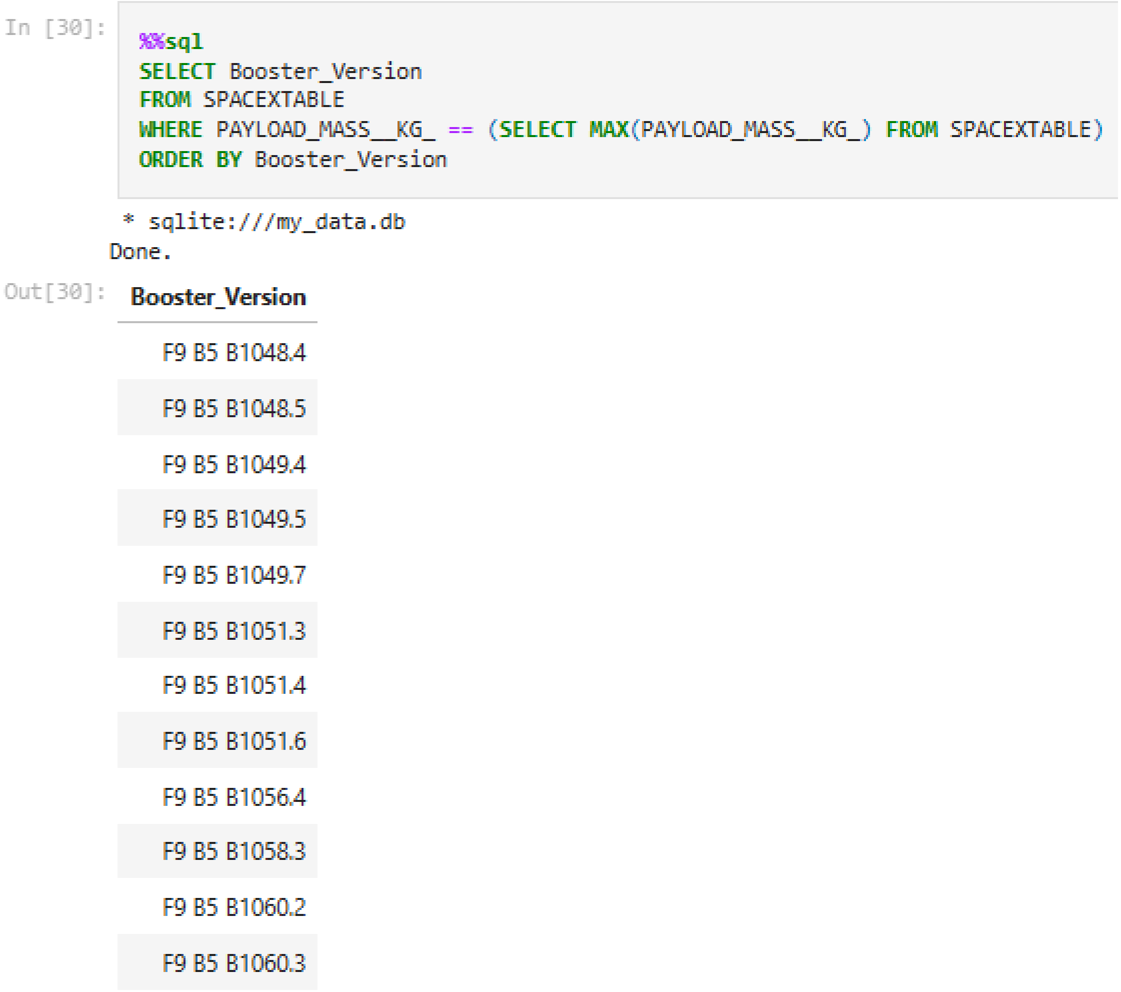
```
 * sqlite:///my_data.db
Done.
```

Out[23]:

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

Boosters Carried Maximum Payload

- We used a subquery to retrieve the maximum payload carried by boosters.
- Then this maximum number is compared with ‘PAYLOAD_MASS__KG’ of the boosters in the WHERE clause to list the names of the booster which have carried the maximum payload mass.
- The ORDER BY is used to arrange the result alphabetically in in ascending order.



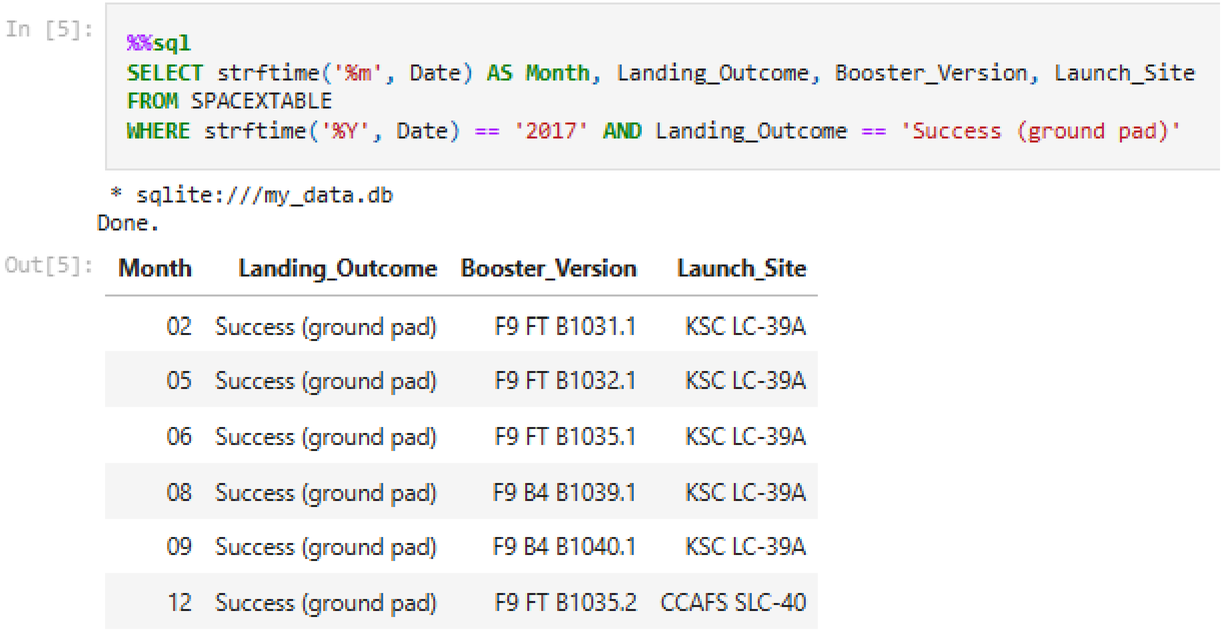
```
In [30]:
%%sql
SELECT Booster_Version
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ == (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
ORDER BY Booster_Version

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

Out[30]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2017 Launch Records

- We used the strftime function with %m to extract month from the Date, and %Y to extract year from the Date, such that WHERE clause considered the year 2017 only.
- The shown query was executed to list the records which displayed the month, successful landing outcomes in ground pad, booster versions, and launch site for the months in year 2017



In [5]:

```
%%sql
SELECT strftime('%m', Date) AS Month, Landing_Outcome, Booster_Version, Launch_Site
FROM SPACEXTABLE
WHERE strftime('%Y', Date) == '2017' AND Landing_Outcome == 'Success (ground pad)'
```

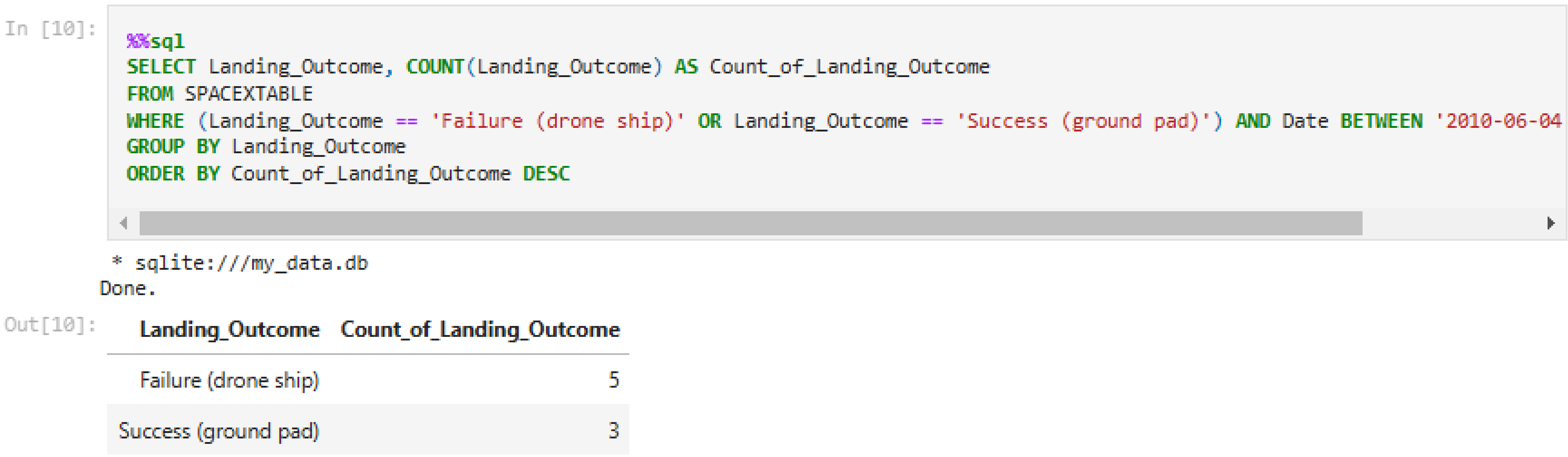
```
 * sqlite:///my_data.db
Done.
```

Out[5]:

Month	Landing_Outcome	Booster_Version	Launch_Site
02	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
05	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
06	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
08	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
09	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
12	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

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

- In the WHERE clause, we filtered the Landing Outcome to be ‘Failure (drone ship)’ OR ‘Success (ground pad)’ at the same time the Date should be BETWEEN 2010-06-04 AND 2017-03-20.
- The GROUP BY clause is used to COUNT similar Landing Outcome which obey the results of the WHERE clause as a group, while the ORDER BY clause with DESC to order the results based on the count in descending order.



In [10]:

```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome) AS Count_of_Landing_Outcome
FROM SPACEXTABLE
WHERE (Landing_Outcome == 'Failure (drone ship)' OR Landing_Outcome == 'Success (ground pad)') AND Date BETWEEN '2010-06-04
GROUP BY Landing_Outcome
ORDER BY Count_of_Landing_Outcome DESC
```

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

Out[10]:

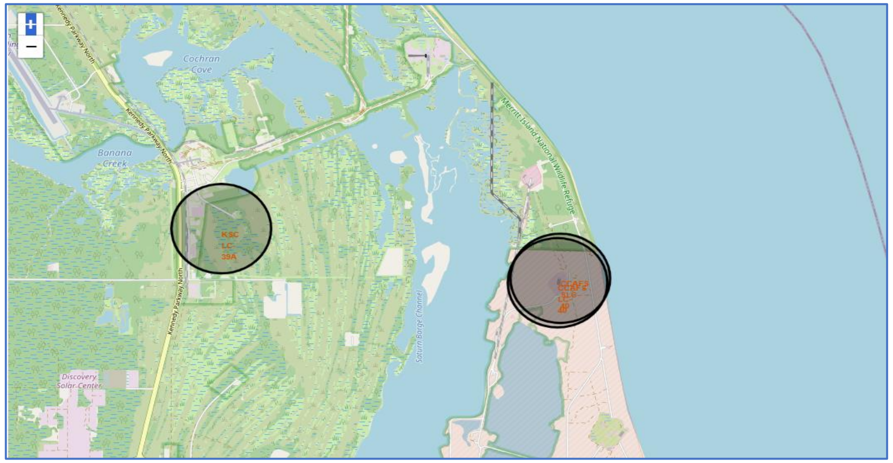
Landing_Outcome	Count_of_Landing_Outcome
Failure (drone ship)	5
Success (ground pad)	3

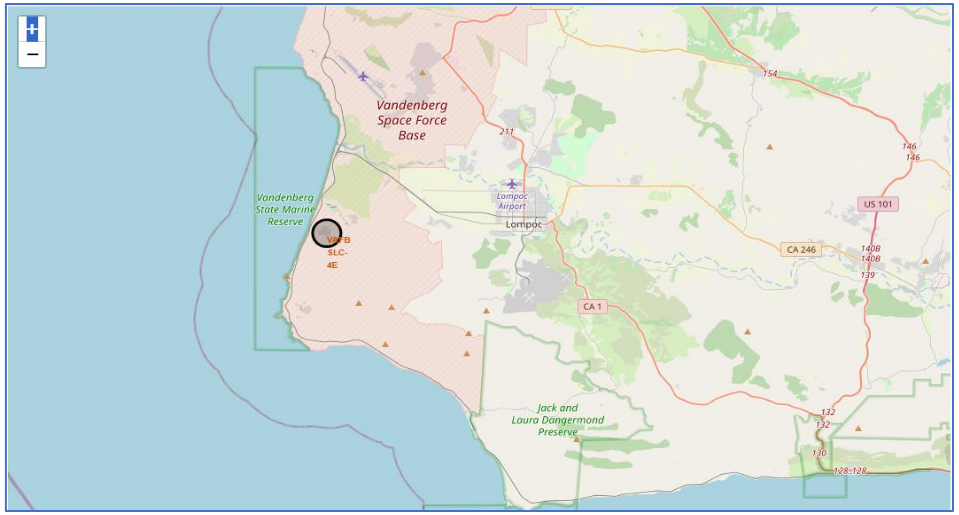
Section 3

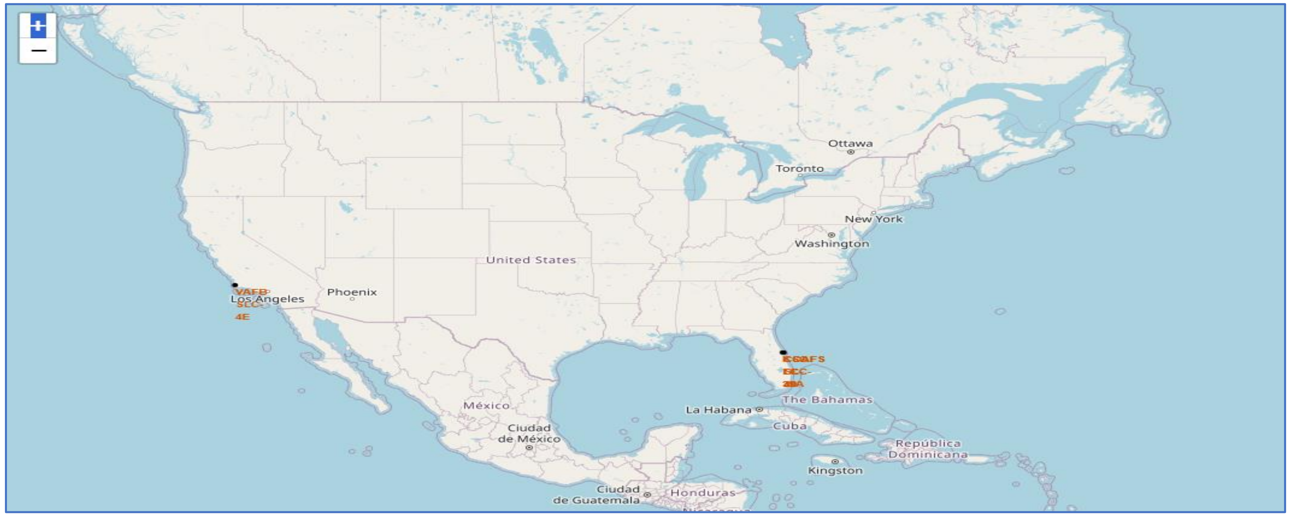
Launch Sites Proximities Analysis

Mark all launch sites on a map

- There are four launch sites: CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, and VAFB SLC-4E.
- All launch sites are in proximity to the Equator line.
- All launch sites are in very close proximity to the coast.

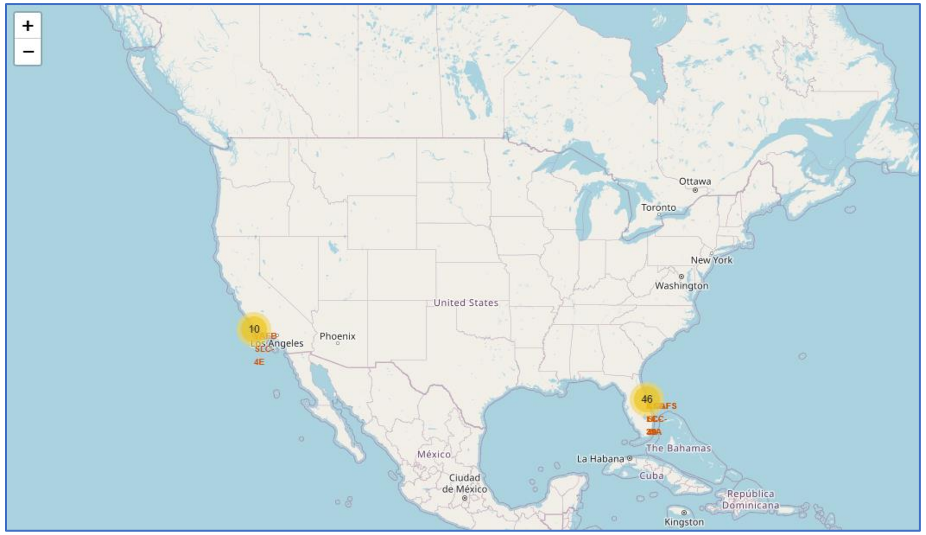


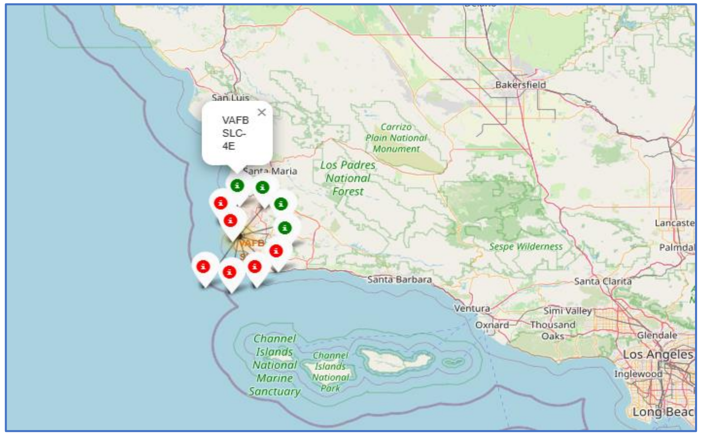

+
−
Vandenberg Space Force Base
Vandenberg State Marine Reserve
VAFB SLC-4E
Lompoc Airport
Lompoc
154
211
146
US 101
CA 246
CA 1
Jack and Laura Dangermond Preserve

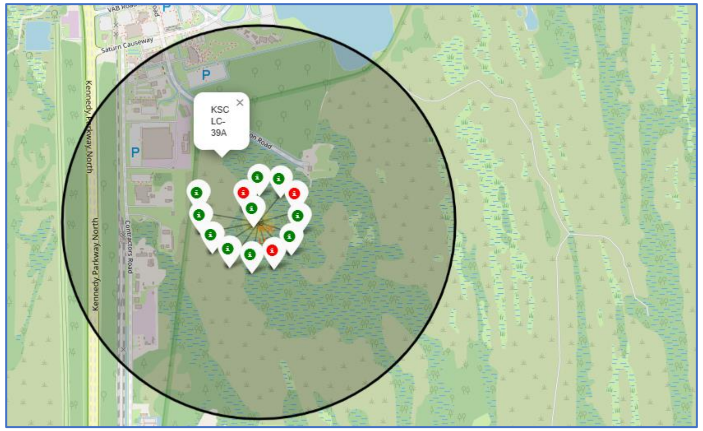


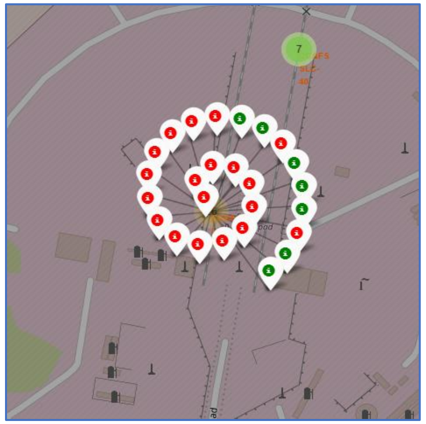
Mark the success/failed launches for each site on the map

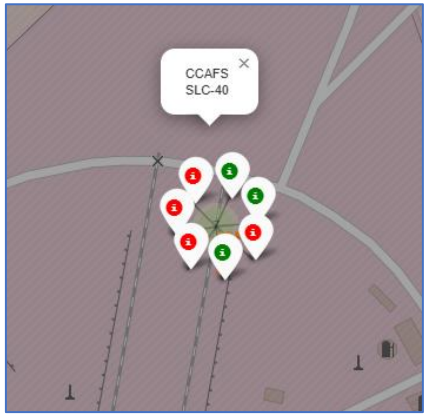
- Since launches only occurred at one of the four sites, many launch records share the same coordinates. To simplify the map, marker clusters were used.
- Green markers represent successful launches, while red markers indicate failed launches.





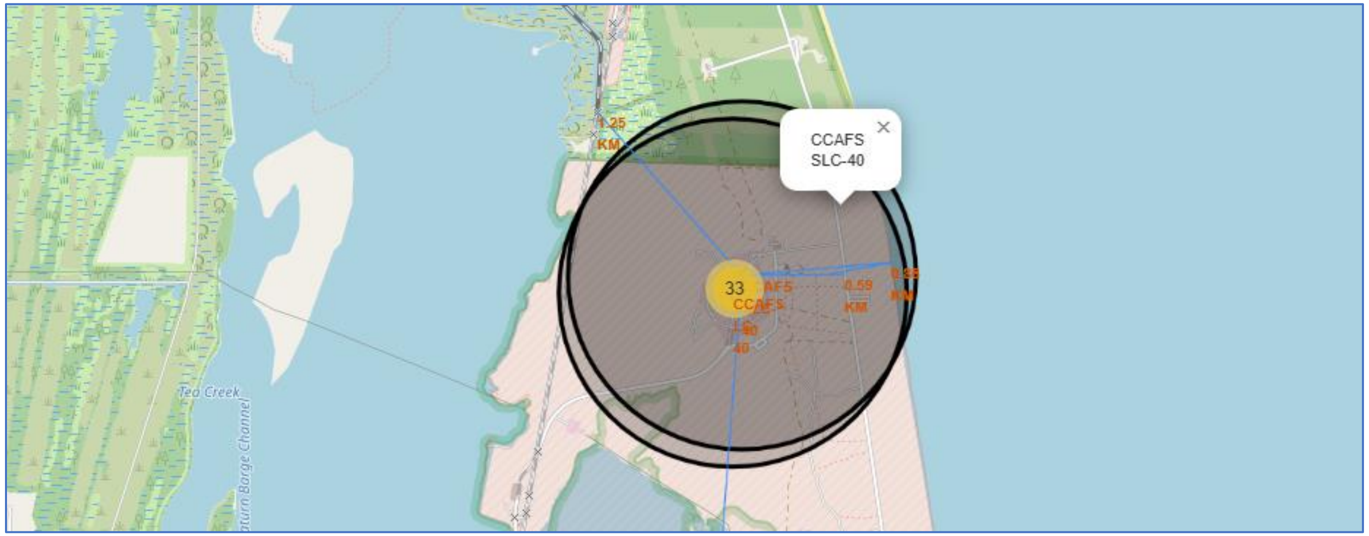


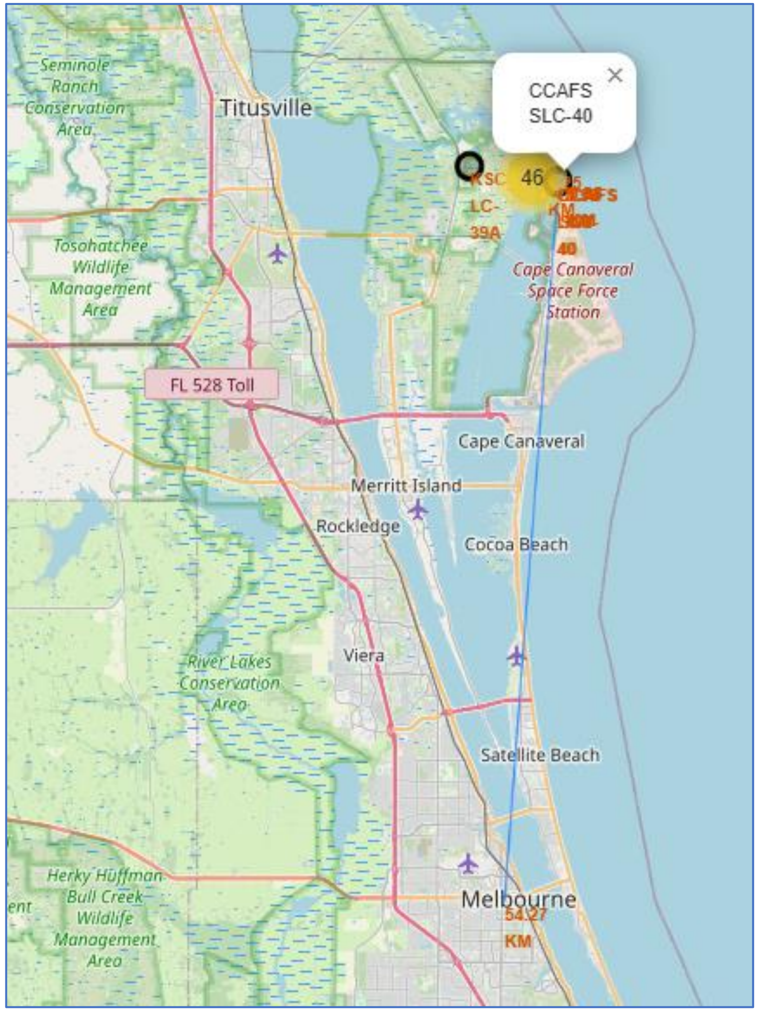


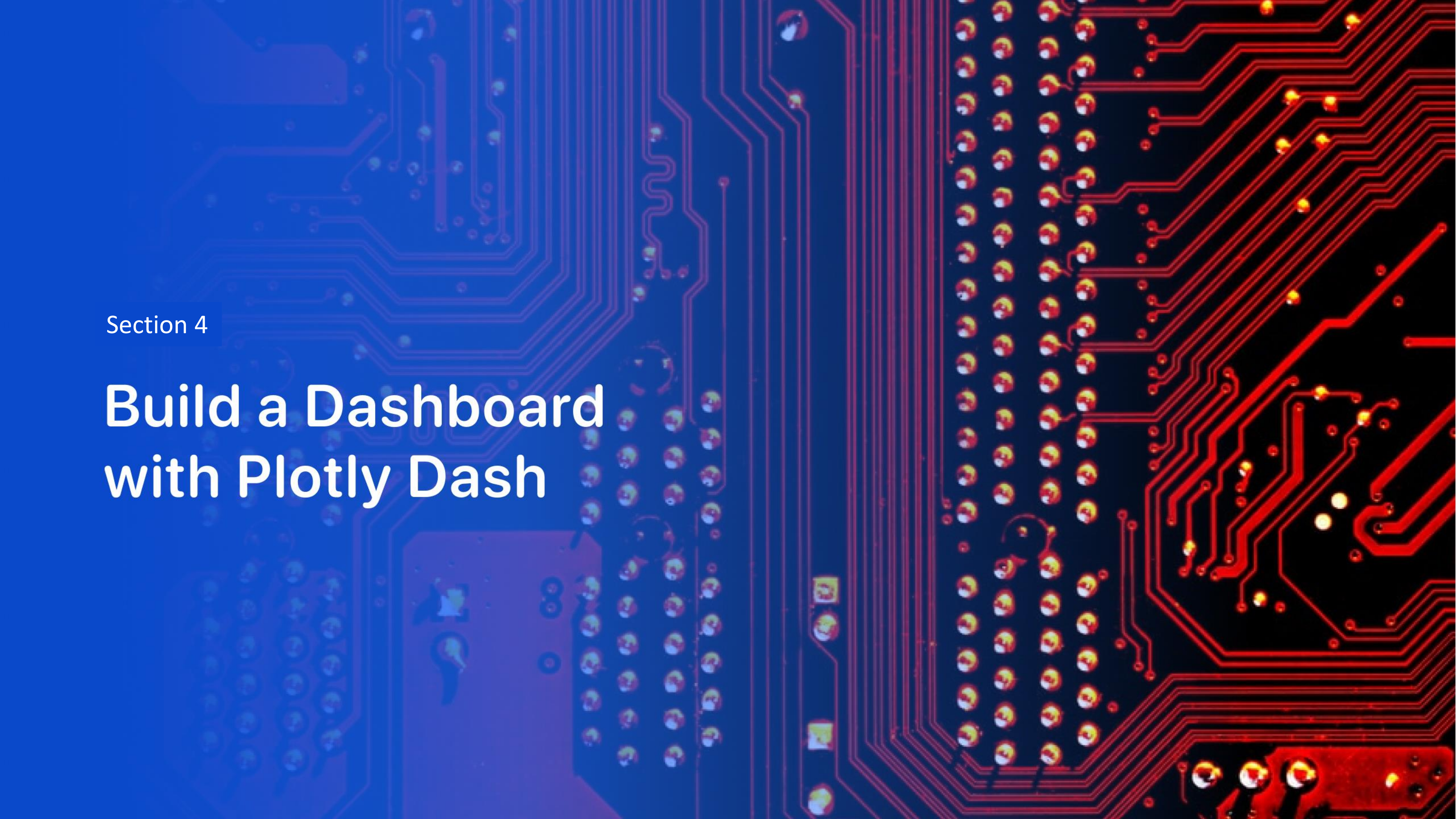


Calculate the distances between a launch site to its proximities

- We drew a line between a launch site to its closest city, railway, highway, and coastline. You need to use MousePosition object to find their coordinates on the map.
- The distance from the CCAFS SLC-40 launch site is in close proximity to the coastline about 0.86 km. However, it is away from the nearest city (Melbourne) about 54 km.




CCAFS
SLC-40
Seminole Ranch Conservation Area
Titusville
KSC LC-39A
46
CCAFS
KM
40
Cape Canaveral Space Force Station
Tosohatchee Wildlife Management Area
FL 528 Toll
Cape Canaveral
Merritt Island
Rockledge
Cocoa Beach
Viera
River Lakes Conservation Area
Satellite Beach
Herky Huffman Bull Creek Wildlife Management Area
Melbourne
54.27
KM

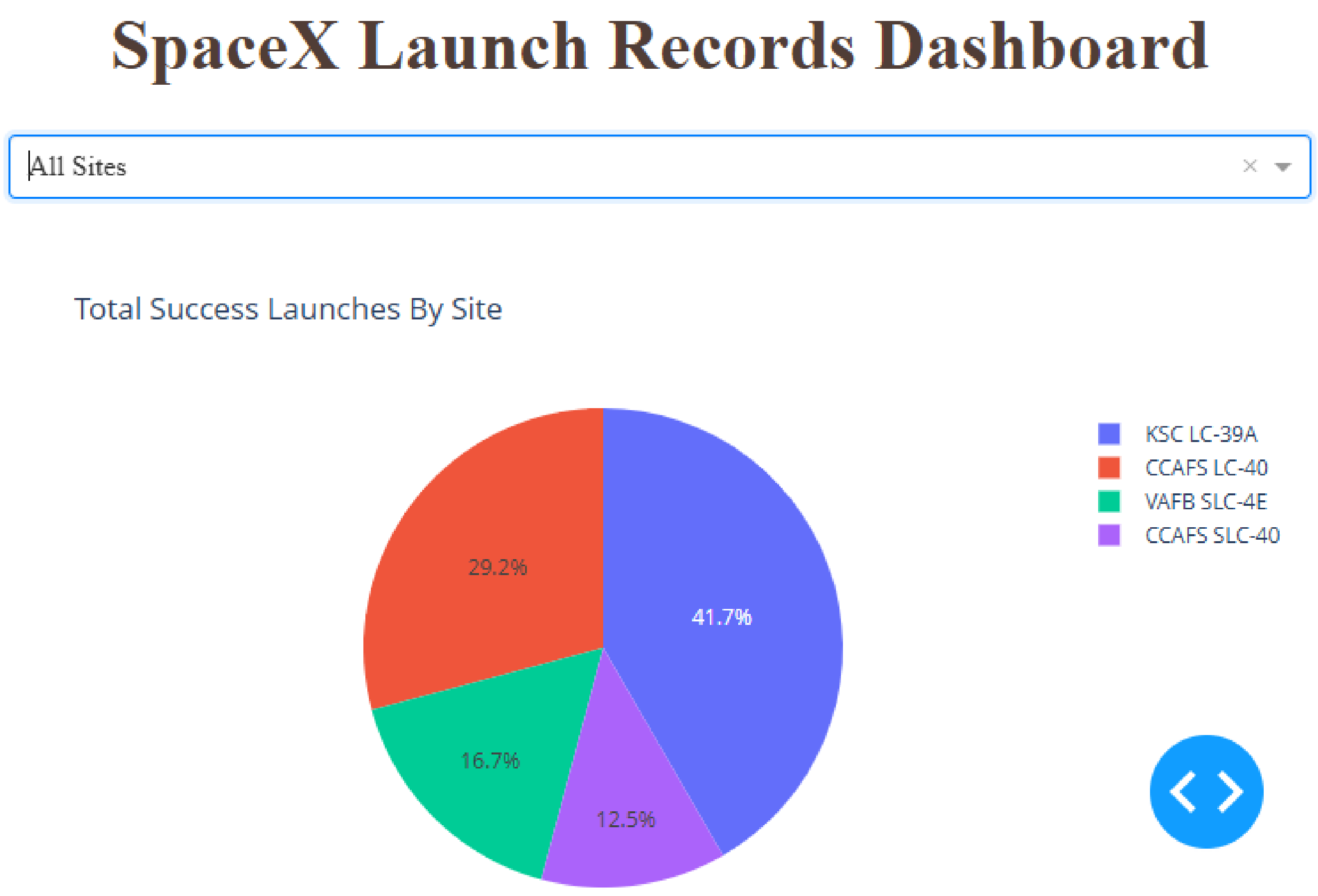


Section 4

Build a Dashboard with Plotly Dash

Launch success count for all sites

- A pie chart shows the total success launches by site, It is obvious that the KSC LC-39A had the most success rate at 41.7% among other sites.



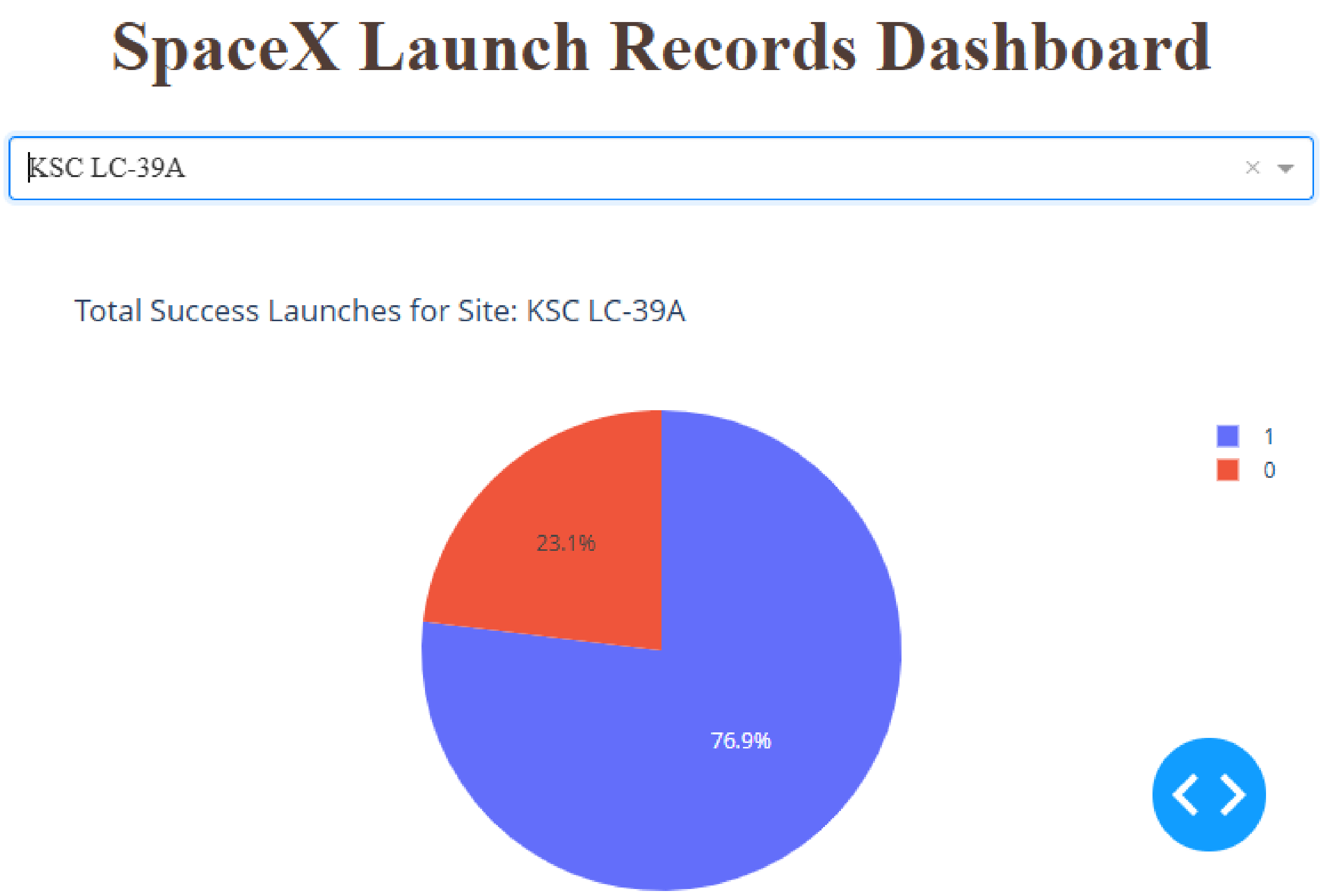
SpaceX Launch Records Dashboard

All Sites

Total Success Launches By Site

Launch site with highest launch success ratio

- A pie chart shows the total success launches for the launch site with the highest success rate.
- The KSC LC-39A launch site had a success rate of 76.9%.



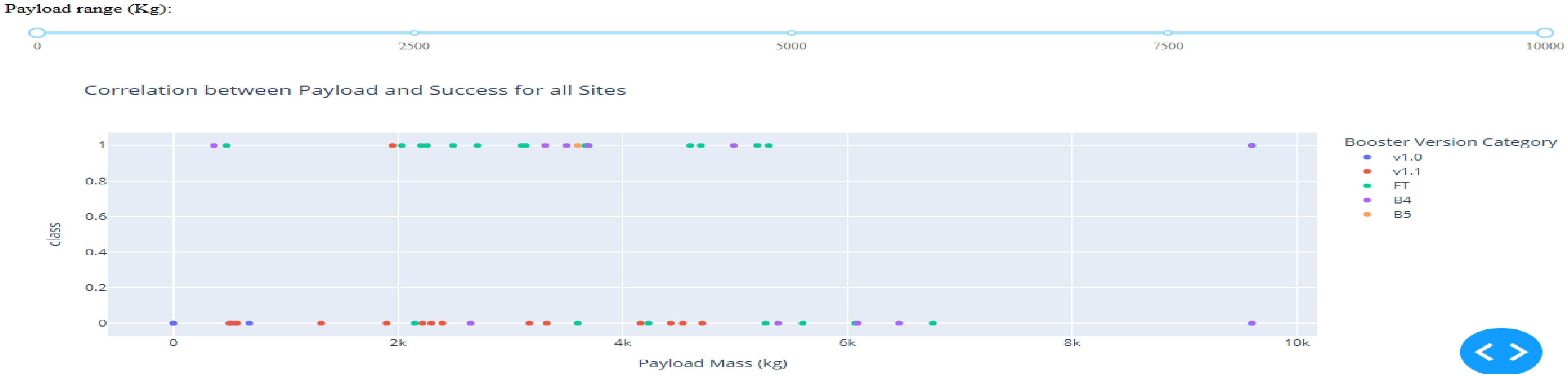
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success Launches for Site: KSC LC-39A

Payload vs. Launch Outcome for all sites

- A scatter plot shows the correlation between payload and success for all sites with different booster versions. The class of 1 indicates the successful launch, while 0 indicates the failure in launch.
- A range slider for payload is used to be able to easily select different payload range and interactively shows the effect on the scatter plot results.
- The payload ranges that had the highest launch success rate are 4.6K - 5.2K => ~80%, 3.5K – 4K => ~80%, 2K - 4K => ~60%.
- The F9 Booster versions B5 and FT had the highest launch success rate of 100% and ~66%, respectively. However, FT had more launches count.

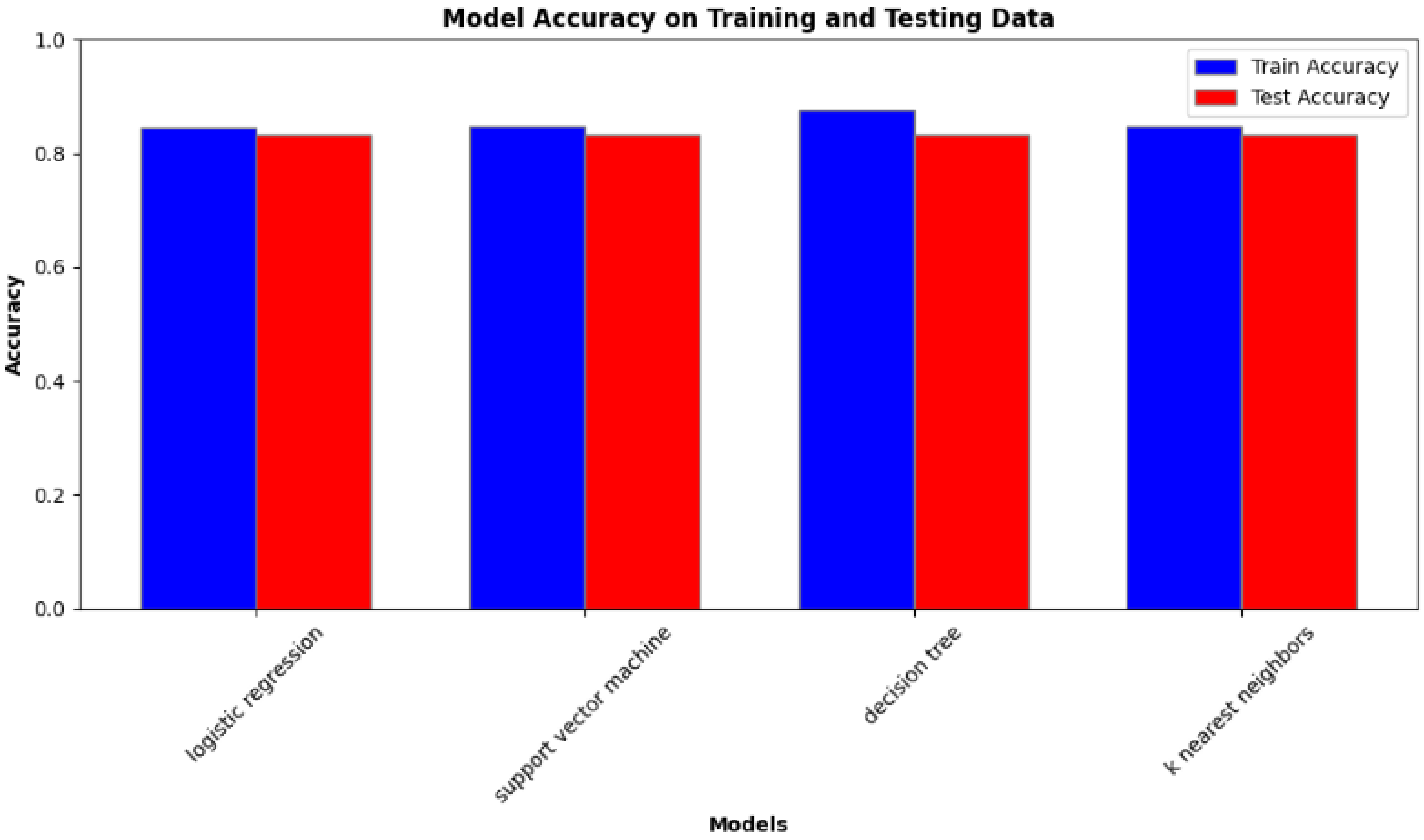


Section 5

Predictive Analysis (Classification)

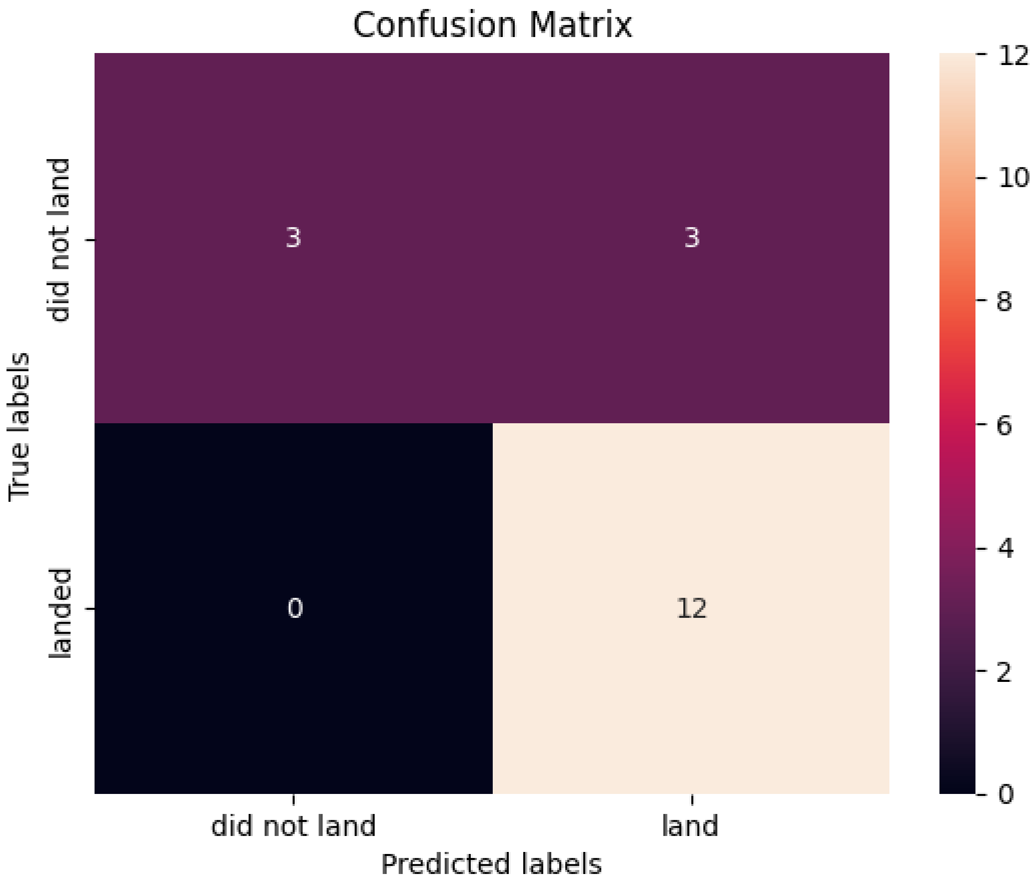
Classification Accuracy

- A bar chart shows the Decision Tree algorithm achieved the best training accuracy. However, all the algorithms performed equally based on the Test accuracy.



Confusion Matrix

- All the models performed equally on the test prediction.
- The false positives (FP) total 3, meaning these instances were classified as successful landings, but they actually were not.
- All the actual successful landings (12) were correctly predicted.



Conclusions

- Increasing success rate over time.
- Orbit type influence on success rates.
- Geographical advantage of launch sites. All SpaceX launch sites are located in proximity to the equator and coastlines.
- The launch site KSC LC-39A has the largest successful launches.
- Among the various launch sites, KSC LC-39A boasts the highest number of successful launches. Additionally, Falcon 9 booster versions B5 and FT have the highest success rates.
- The mass of the payload influences the success rate of launches.
- Machine learning models can be employed to predict the success of rocket landings, achieving a test accuracy of 83.33%.

Appendix

- The main github link for the whole project can be viewed from:
  - https://github.com/mba-github32/Data-Science-and-Machine-Learning-Capstone-Project/tree/main
- The interactive maps of the launch sites' locations analysis with folium can be viewed from:
  - https://nbviewer.org/github/mba-github32/Data-Science-and-Machine-Learning-Capstone-Project/blob/main/Lab5_launch_site_location.ipynb

Thank you!