IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Martin Luo
April 7, 2023

Outline

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

Executive Summary

Summary of methodologies

- Data Collection via API, Web Scraping
- Exploratory Data Analysis (EDA) with Data Visualization
- EDA with SQL
- Interactive Map with Folium
- Dashboards with Plotly Dash
- Predictive Analysis

Summary of all results

- Exploratory Data Analysis results
- Interactive maps and dashboard
- Predictive results

Introduction

- ## Project background and context

- The objective of this project is to predict whether the first stage of the Falcon 9 rocket will land successfully. According to SpaceX's website, the cost of a Falcon 9 rocket launch is $62 million, significantly lower than the $165 million price tag of other launch providers. The key reason for this substantial price difference lies in SpaceX's ability to reuse the first stage of the rocket. By accurately predicting the success of the first stage landing, we can better estimate the cost of a launch. This information is valuable for other companies considering competing with SpaceX in the rocket launch market, as it helps them make informed decisions about pricing and resource allocation.

- ## Problems you want to find answers

1. What are the main characteristics of a successful or failed landing ?
2. What are the effects of each relationship of the rocket variables on the success or failure of a landing ?
3. What are the conditions which will allow SpaceX to achieve the best landing success rate ?

Section 1

Methodology

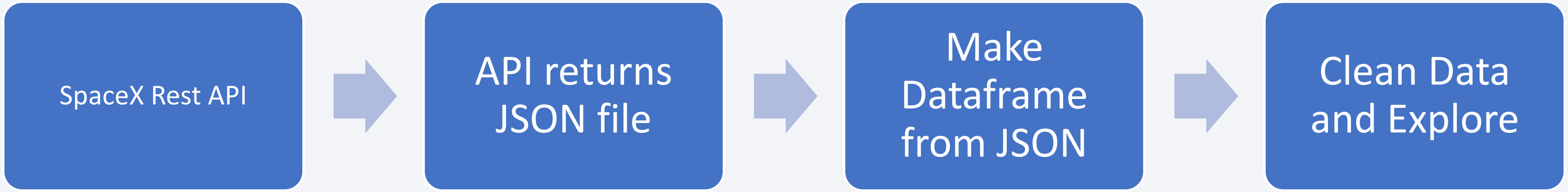
Methodology

Executive Summary

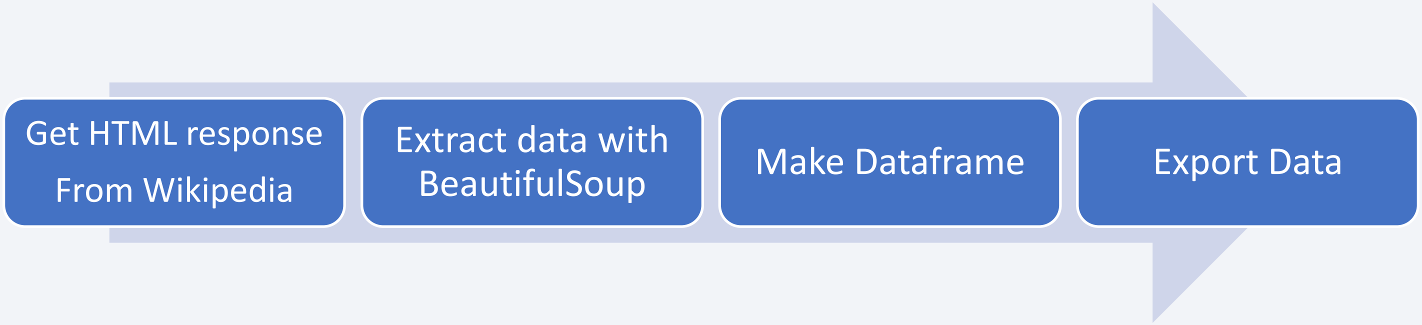
- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Describe how data was processed
  - One Hot Encoding for classification models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

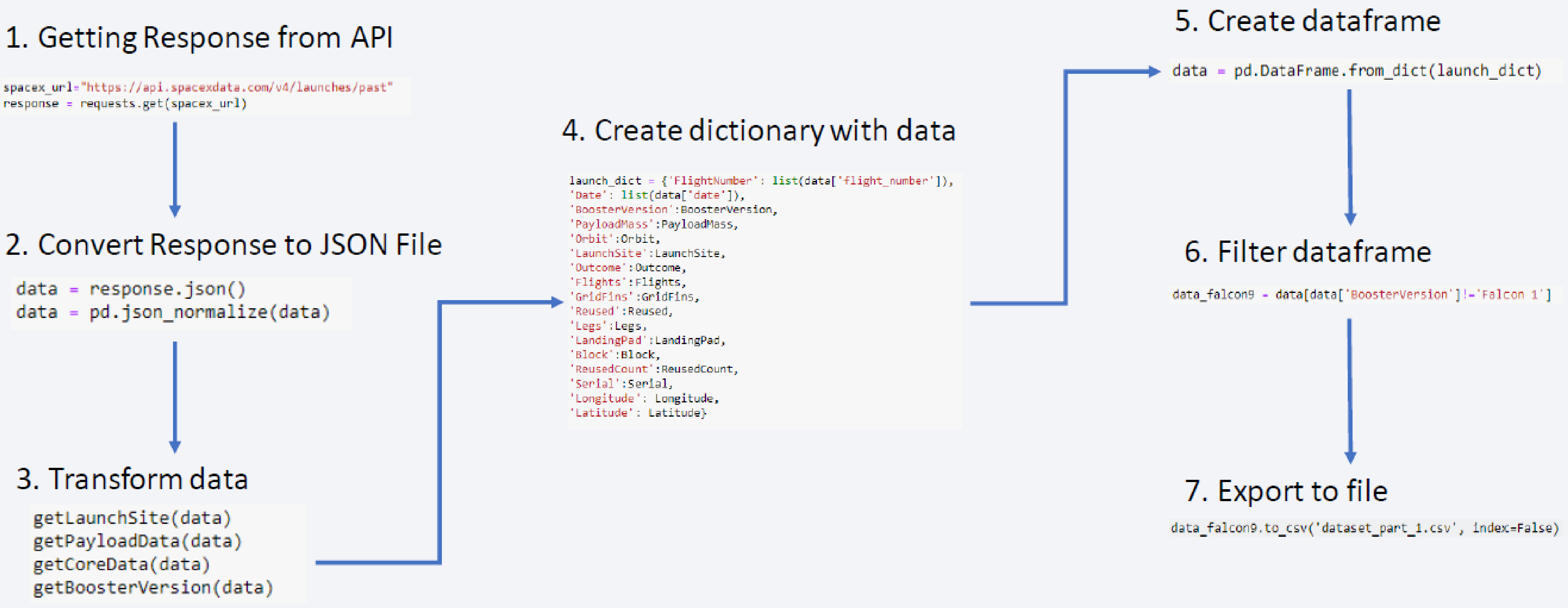
- Datasets are collected from Rest SpaceX API and web scrapping Wikipedia
- The information obtained by the API are rockets, launches, and payload information.
- The Space XREST API URL is api.spacexdata.com/v4/



- The information obtained by the web scrapping of Wikipedia are launches, landing, and payload information.
- URL: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922


Get HTML response From Wikipedia
Extract data with BeautifulSoup
Make Dataframe
Export Data

Data Collection – SpaceX API



1. Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Convert Response to JSON File

```
data = response.json()
data = pd.json_normalize(data)
```

3. Transform data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
getBoosterVersion(data)
```

4. Create dictionary with data

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Create dataframe

```
data = pd.DataFrame.from_dict(launch_dict)
```

6. Filter dataframe

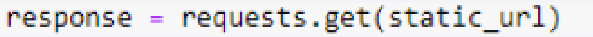
```
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
```

7. Export to file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

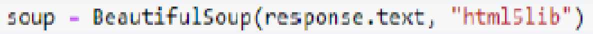
Data Collection - Scraping

1. Getting Response from HTML



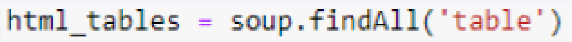
```
response = requests.get(static_url)
```

2. Create BeautifulSoup Object



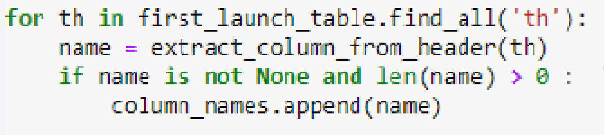
```
soup = BeautifulSoup(response.text, "html5lib")
```

3. Find all tables



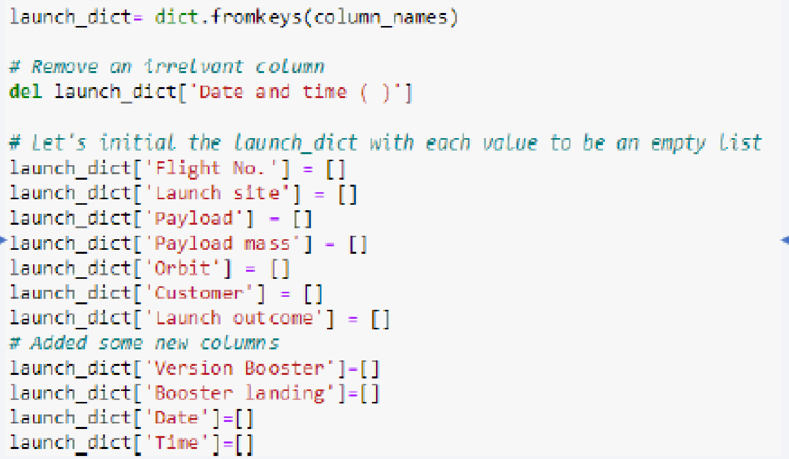
```
html_tables = soup.findAll('table')
```

4. Get column names



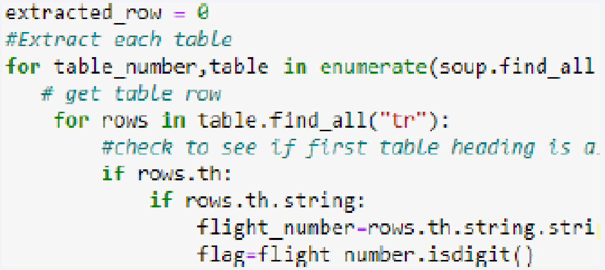
```
for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0 :
        column_names.append(name)
```

5. Create dictionary



```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

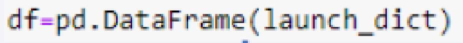
6. Add data to keys



```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is a
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.stri
                flag=flight_number.isdigit()
```

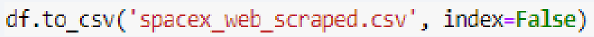
See notebook for the rest of code

7. Create dataframe from dictionary



```
df=pd.DataFrame(launch_dict)
```

8. Export to file

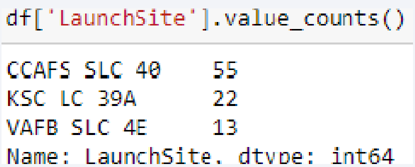


```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

- In the dataset, there are several cases where the booster did not land successully.
- True Ocean, True RTLS, True ASDS means the mission has been successful.
- False Ocean, False RTLS, False ASDS means the mission was a failure.
- We need totransformstring variables intocategoricalvariables where 1 meansthe mission has been successful and 0 means the mission was afailure.

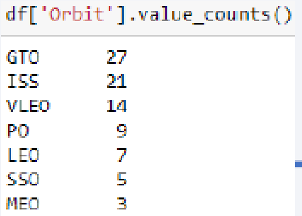
1. Calculate launches number for each site



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

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

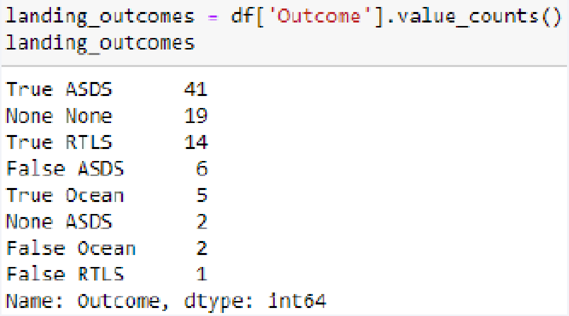
2. Calculate the number and occurence of each orbit



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

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
```

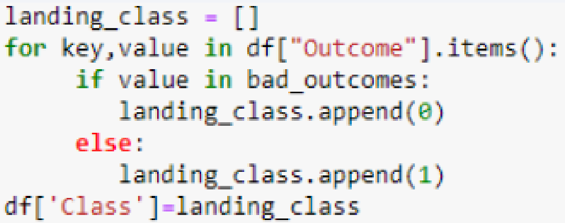
3. Calculate number and occurrence of mission outcome per orbit type



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

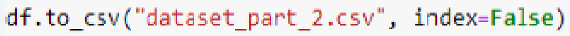
```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

4. Create landing outcome label from Outcome column



```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

5. Export to file



```
df.to_csv("dataset_part_2.csv", index=False)
```

EDA with Data Visualization

- Scatter Graphs
- 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

Scatter plots show relationship between variables.

This relationship is called the correlation.

- Bar Graph
- Success rate vs. Orbit

Bar graphs show the relationship between numeric and categoric variables.

- Line Graph
- Success rate vs. Year

Line graphs show data variables and their trends. Line graphs can help to show global behavior and make prediction for unseen data.

EDA with SQL

We performed SQL queries to gather and understand data from dataset:

- Displaying the names of the unique lauunch sites in the space mission.
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS).
- Display average payload mass carried by booster version F9 v1.1.
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- List the total number of successful and failure mission outcomes.
- List the names of the booster_versions which have carried the maximum payload mass.
- List the records which will display the month names, faiilure landing_ouutcomes in drone ship, booster versions, launch_site forthe months in year 2015.
- Rank the count of successful landiing_outcomes between the date04-06-2010 and 20-03-2017in descending order.

Build an Interactive Map with Folium

Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas

- Red circle at NASA Johnson Space Center's coordinate with label showing its name(folium.Circle, folium.map.Marker).
- Red circles at each launch site coordinates with label showing launch site name (folium.Circle, folium.map.Marker, folium.features.DivIcon).
- The grouping of points in a cluster to display multiple and different information for the same coordinates (folium.plugins.MarkerCluster).
- Markers to show successful and unsuccessful landings. Greenfor successful landing and Redfor unsuccessful landing. (folium.map.Marker, folium.Icon).
- Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them. (folium.map.Marker, folium.PolyLine, folium.features.DivIcon*)*

These objects are created in order to understand better the problem and the data.We can show easily all launch sites, their surroundings and thenumber of successful and unsuccessful landings.

Build a Dashboard with Plotly Dash

Dashboard has dropdown,pie chart,rangeslider andscatter plot components

- Dropdownallows a user to choose the launch site or all launch sites (dash_core_components.Dropdown).
- Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component(plotly.express.pie).
- Rangeslider allows a user to select a payload mass in a fixed range (dash_core_components.RangeSlider).
- Scatter chart showsthe relationship between two variables, in particular Success vs Payload Mass (plotly.express.scatter).

Predictive Analysis (Classification)

Data preparation

- Load dataset
- Normalize data
- Split data into training and test sets.

Model preparation

- Selection of machine learning algorithms
- Set parameters for each algorithm to GridSearchCV
- Training GridSearchModel models with the training dataset

Model evaluation

- Get the best hyperparameters for each type of model
- Compute accuracy for each model with the test dataset
- Plot Confusion Matrix

Model comparison

- Comparison of models according to their accuracy
- The model with the best accuracy will be chosen (see Notebook for result)

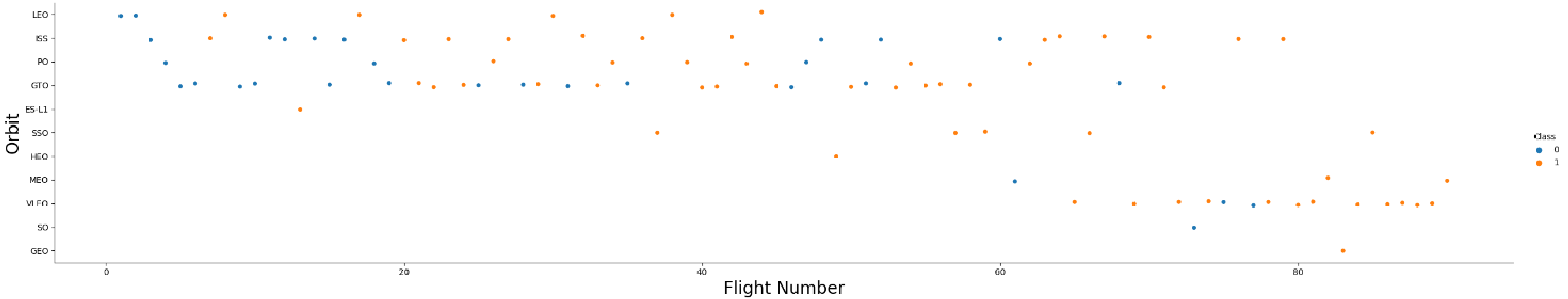
Results

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

Section 2

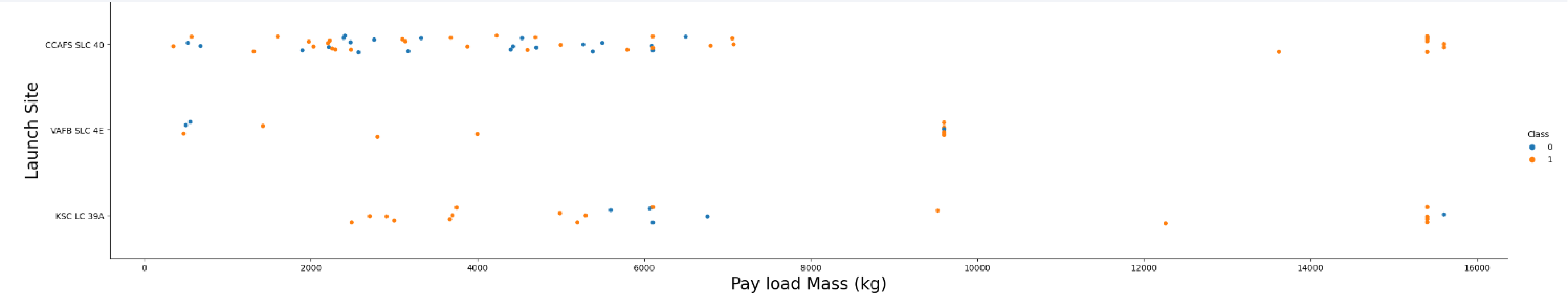
Insights drawn from EDA

Flight Number vs. Launch Site



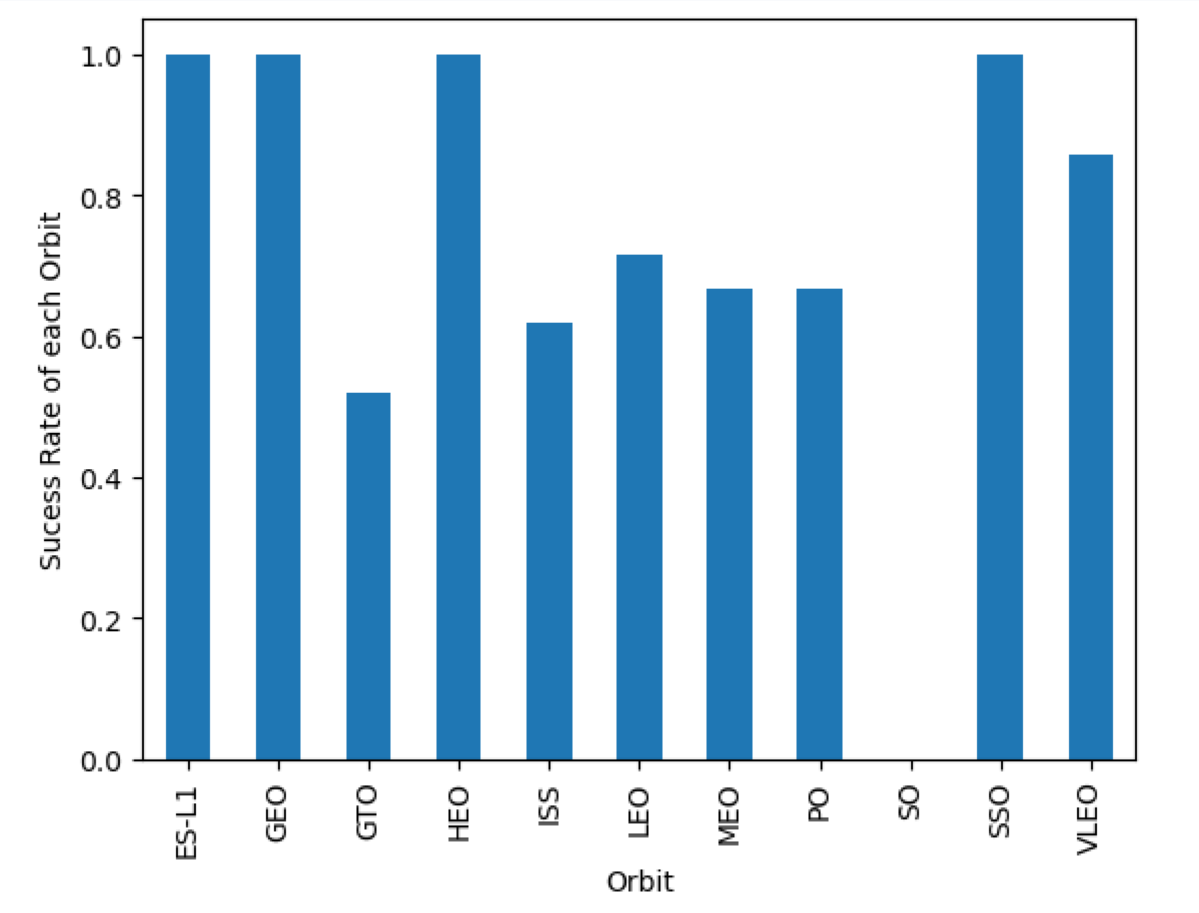
It is evident that the success rate for each site is on the rise.

Payload vs. Launch Site



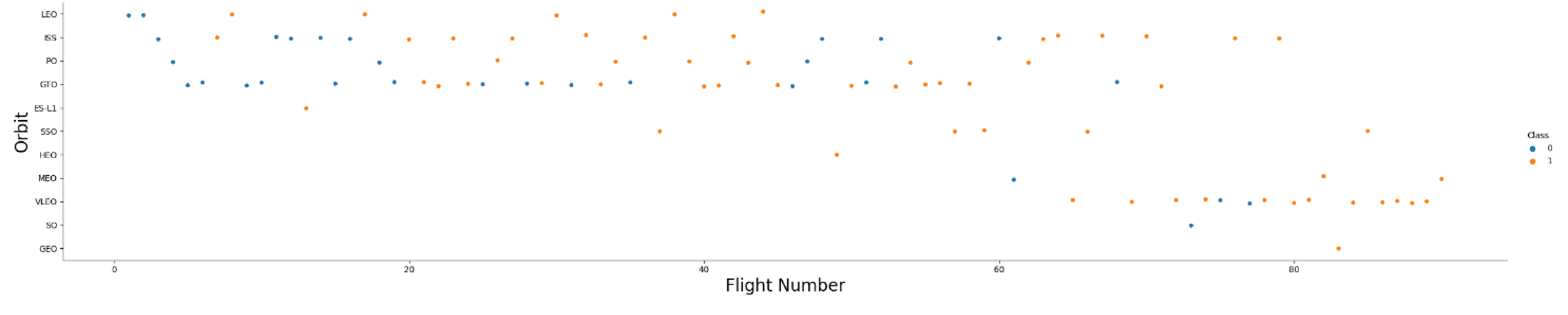
The choice of a suitable payload weight is crucial for a successful landing, as it depends on the launch site; while a heavier payload may be necessary in some cases, an excessively heavy one can result in a failed landing.

Success Rate vs. Orbit Type



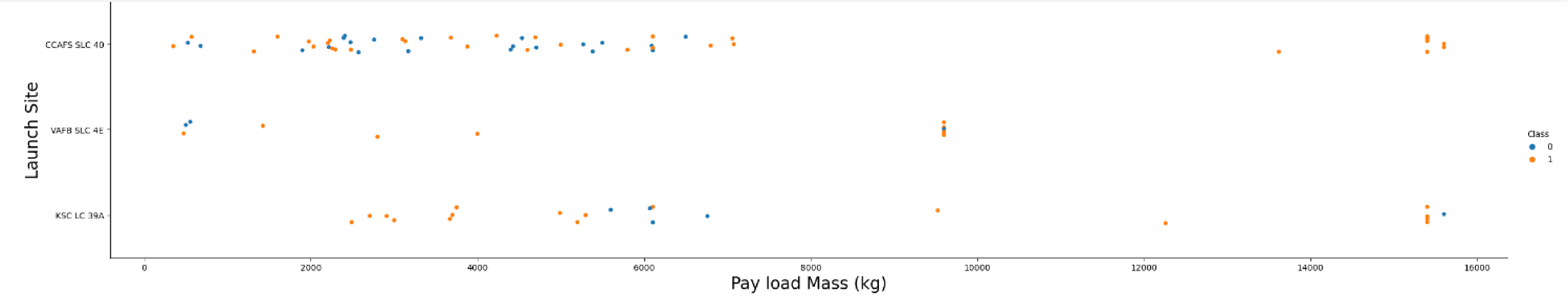
The plot illustrates the success rates of various orbit types, and it is evident that ES L1, GEO, HEO, and SSO exhibit the highest success rates

Flight Number vs. Orbit Type



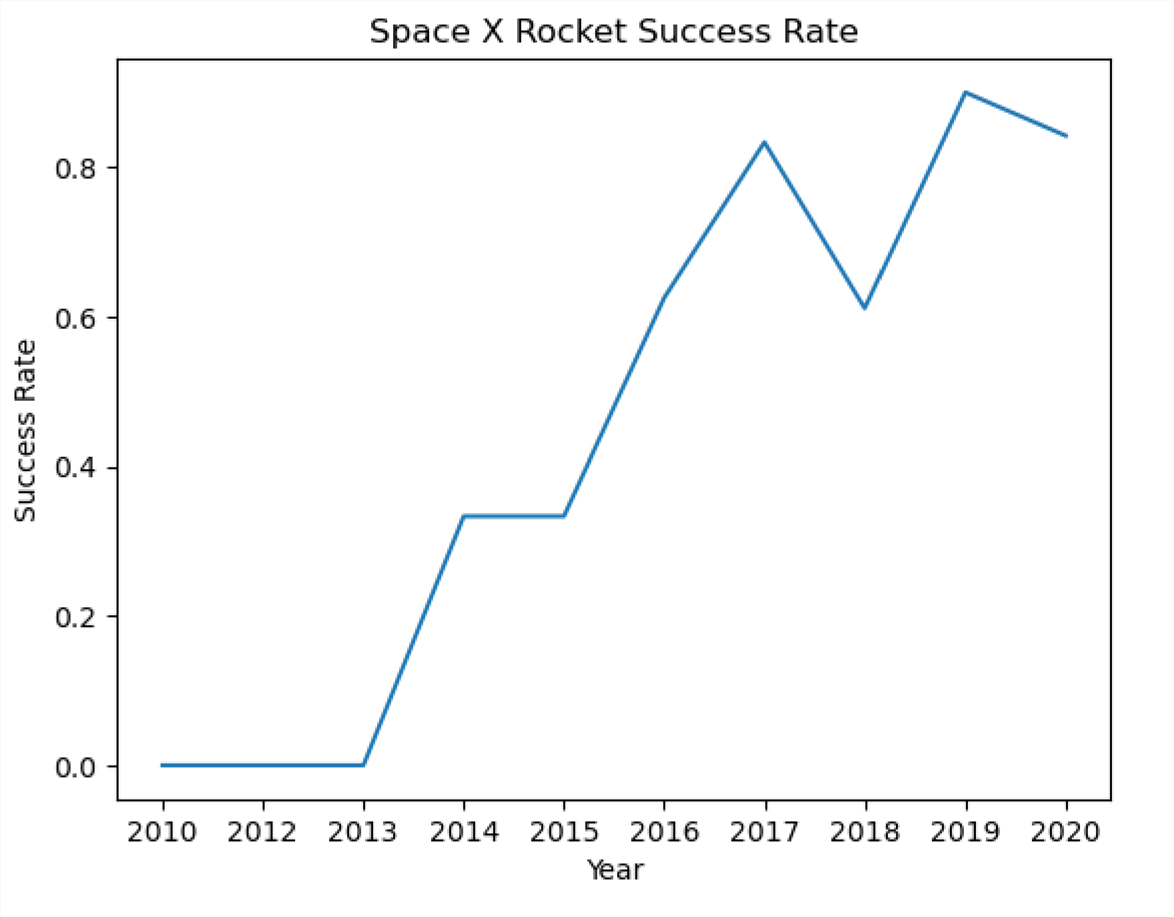
While we observe an increase in success rate with the number of flights for the LEO orbit, there seems to be no such correlation for the GTO orbit. However, we can hypothesize that the high success rates of orbits such as SSO or HEO could be attributed to the experience gained from previous launches of other orbits.

Payload vs. Orbit Type



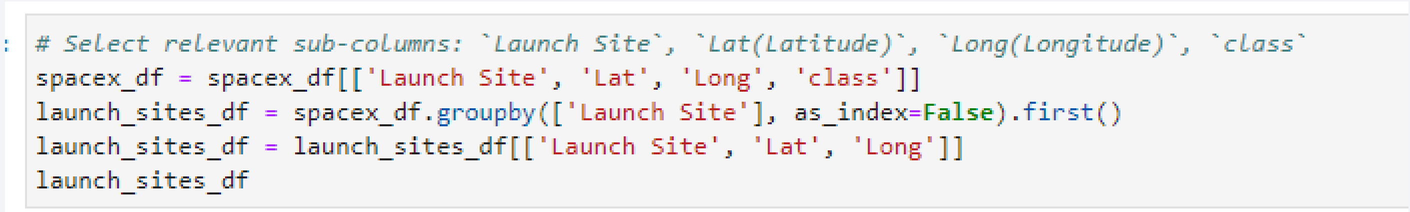
The weight of payloads can significantly impact the success rate of launches in certain orbits. Specifically, heavier payloads increase the success rate for the LEO orbit, whereas reducing the payload weight improves the success rate for a GTO orbit.

Launch Success Yearly Trend

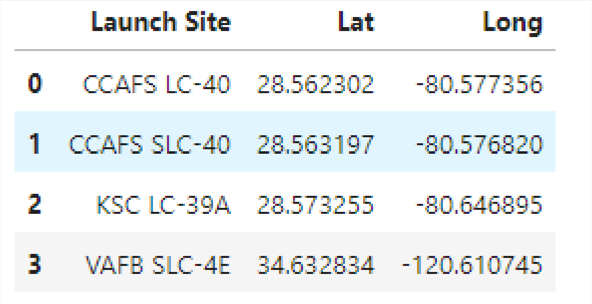


From 2013 onwards, there has been a noticeable upward trend in the success rate of Space X Rockets.

All Launch Site Names

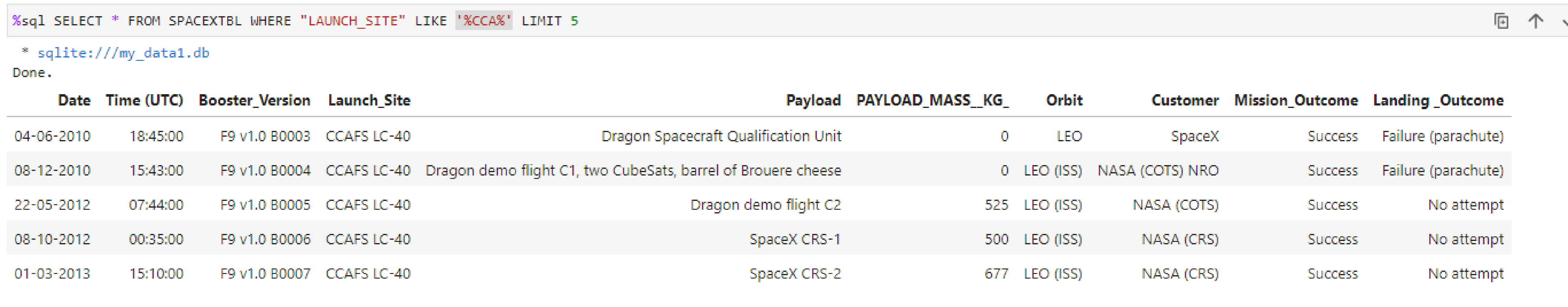


```
# Select relevant sub-columns: `Launch Site`, `Lat(Latitude)`, `Long(Longitude)`, `class`
spacex_df = spacex_df[['Launch Site', 'Lat', 'Long', 'class']]
launch_sites_df = spacex_df.groupby(['Launch Site'], as_index=False).first()
launch_sites_df = launch_sites_df[['Launch Site', 'Lat', 'Long']]
launch_sites_df
```



	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

Launch Site Names Begin with 'CCA'



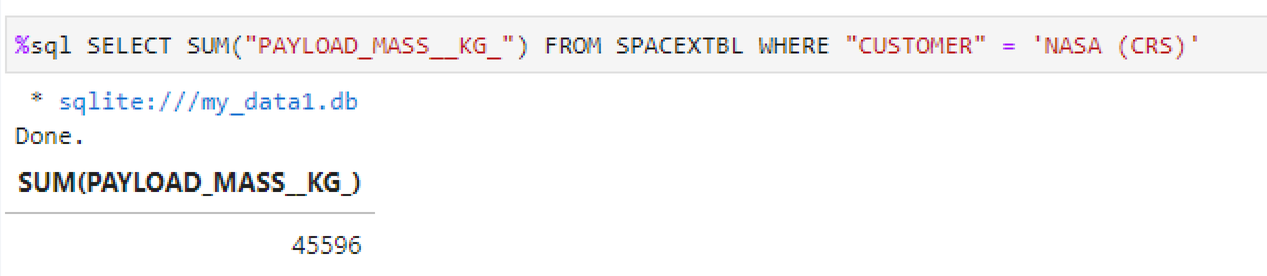
```
%sql SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5
```

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

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

When using the WHERE clause with a LIKE clause, the launch sites that include the substring 'CCA' are filtered. Adding LIMIT 5 to the query retrieves the first 5 records from the filtered results.

Total Payload Mass



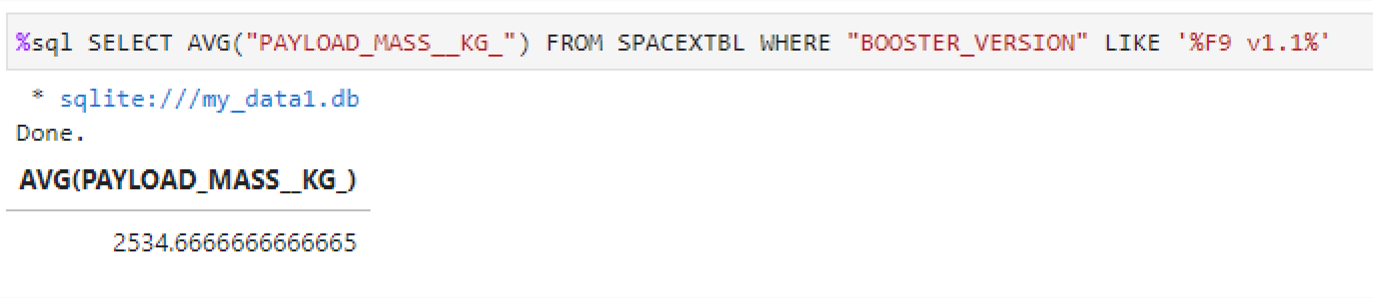
```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

This query calculates the total payload mass for all missions where the customer is NASA (CRS).

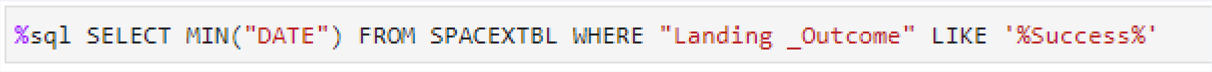
Average Payload Mass by F9 v1.1



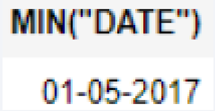
```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'

 * sqlite:///my_data1.db
Done.
AVG(PAYLOAD_MASS__KG_)
      2534.6666666666665
```

This query calculates the average payload mass for all missions where the booster version includes the substring 'F9 v1.1'.

First Successful Ground Landing Date



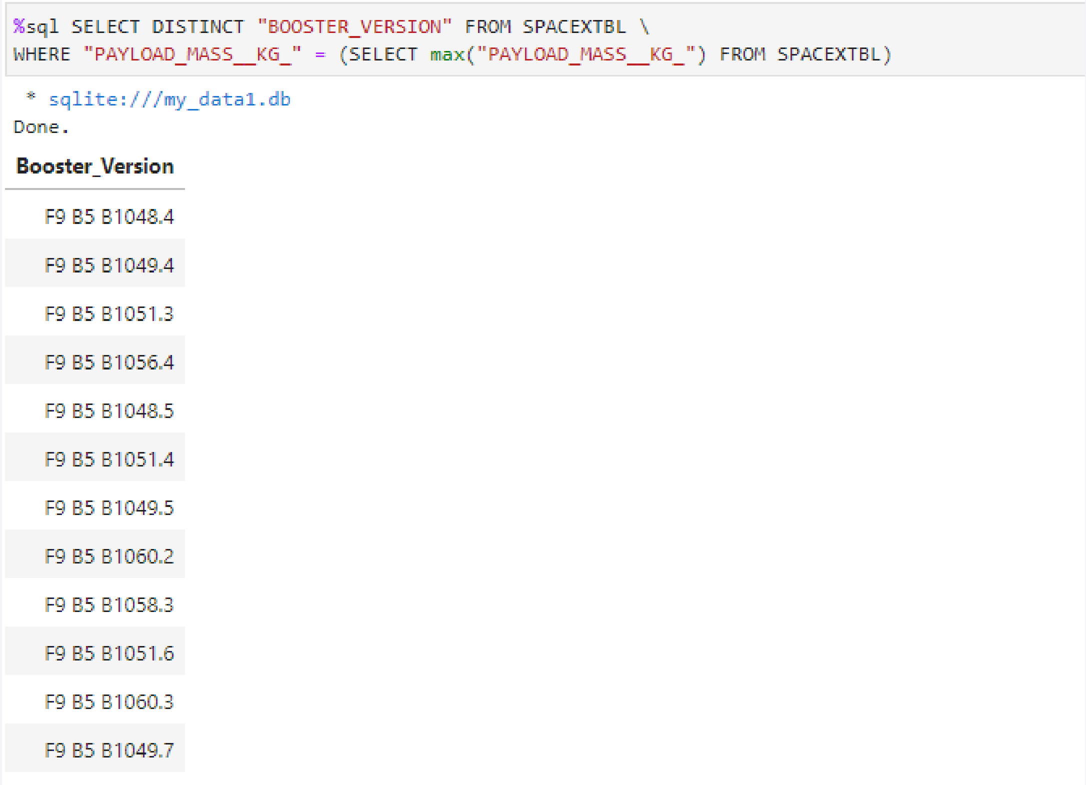
```
%sql SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'
```



MIN("DATE")
01-05-2017

This query retrieves the record of the oldest successful landing by using the WHERE clause to filter the dataset and keep only the records where the landing was successful. The MIN function is then applied to select the record with the earliest date.

Successful Drone Ship Landing with Payload between 4000 and 6000



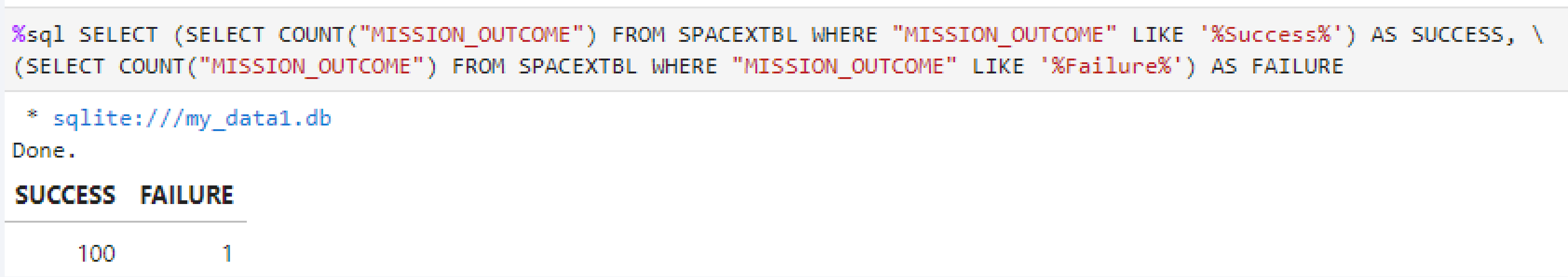
```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

```
 * 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

This query retrieves the booster version for all missions where the landing was successful and the payload mass is between 4000 and 6000 kg. The WHERE and AND clauses are used to filter the dataset based on these criteria.

Total Number of Successful and Failure Mission Outcomes



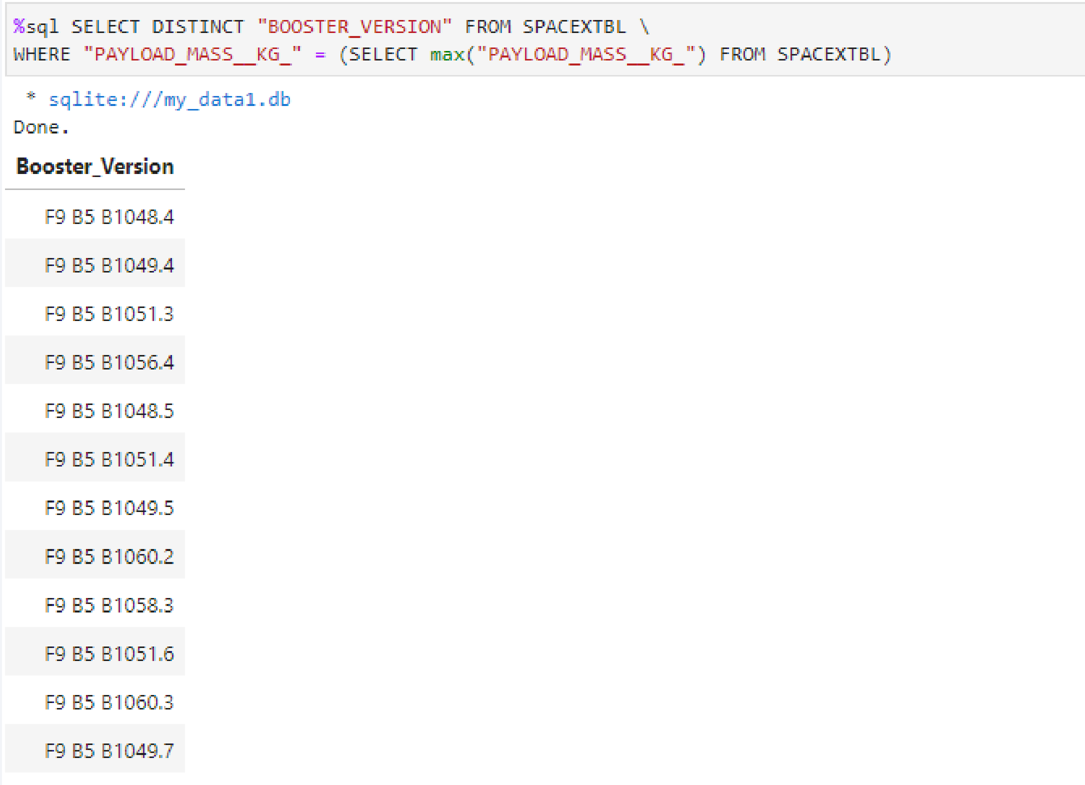
```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, \
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

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

SUCCESS	FAILURE
100	1

The first SELECT statement displays the subqueries that generate the results. The first subquery counts the number of successful missions, while the second subquery counts the number of unsuccessful missions. The WHERE clause, combined with the LIKE operator, filters the records based on the mission outcome. The COUNT function is then used to count the number of records that match the filtering criteria.

Boosters Carried Maximum Payload



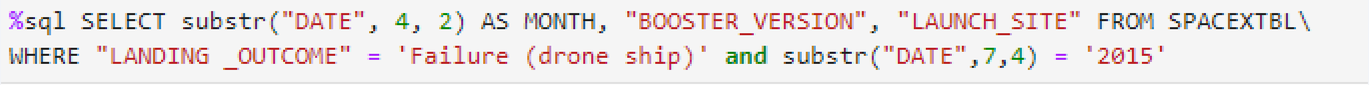
```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

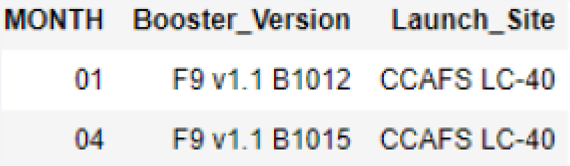
Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

This query retrieves the booster version for all missions where the landing was successful and the payload mass is between 4000 and 6000 kg. The WHERE and AND clauses are used to filter the dataset based on these criteria.

2015 Launch Records



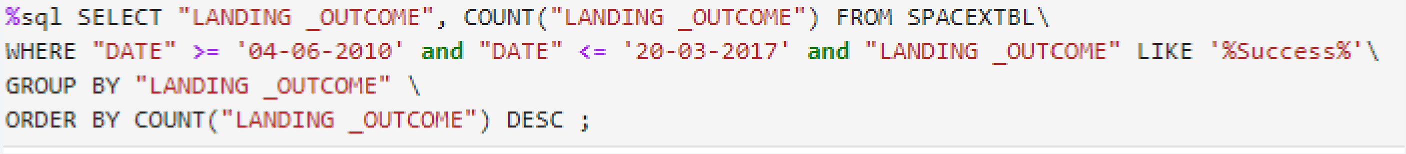
```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```



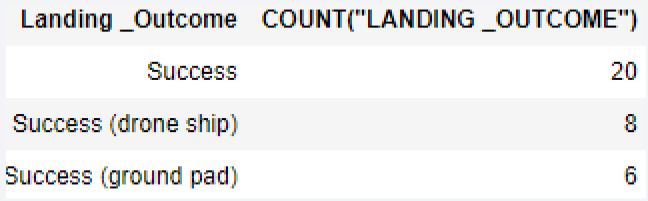
MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

This query retrieves the month, booster version, and launch site for all missions where the landing was unsuccessful and the landing date was in 2015. The Substr function is used to extract the month and year from the DATE field: Substr(DATE, 4, 2) shows the month, and Substr(DATE, 7, 4) shows the year.

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



```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```



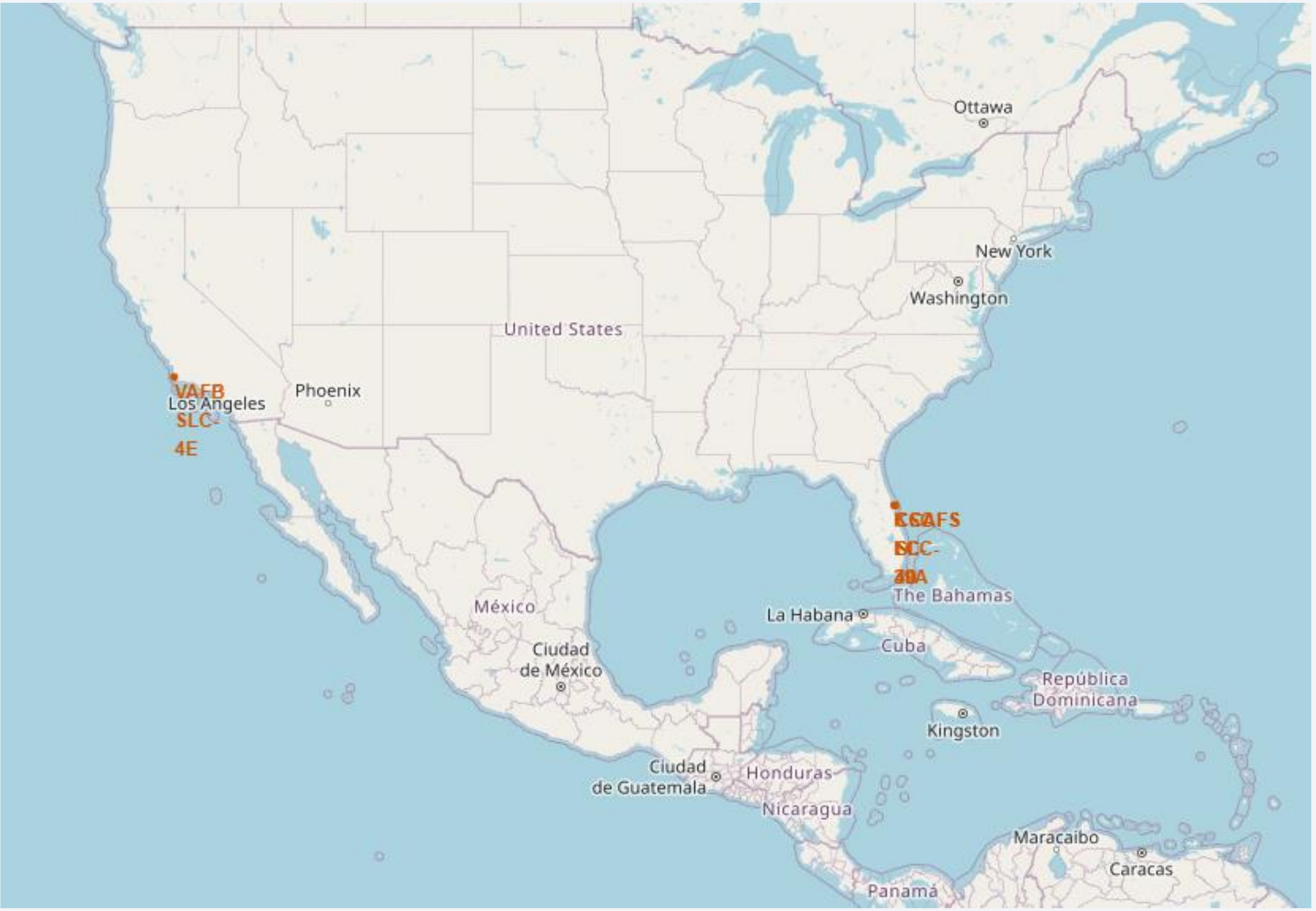
Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

This query retrieves the landing outcomes and their respective counts for all missions where the landing was successful and the date was between 04/06/2010 and 20/03/2017. The GROUP BY clause is used to group the results by landing outcome, while the ORDER BY COUNT DESC clause sorts the results in decreasing order of the count.

Section 3

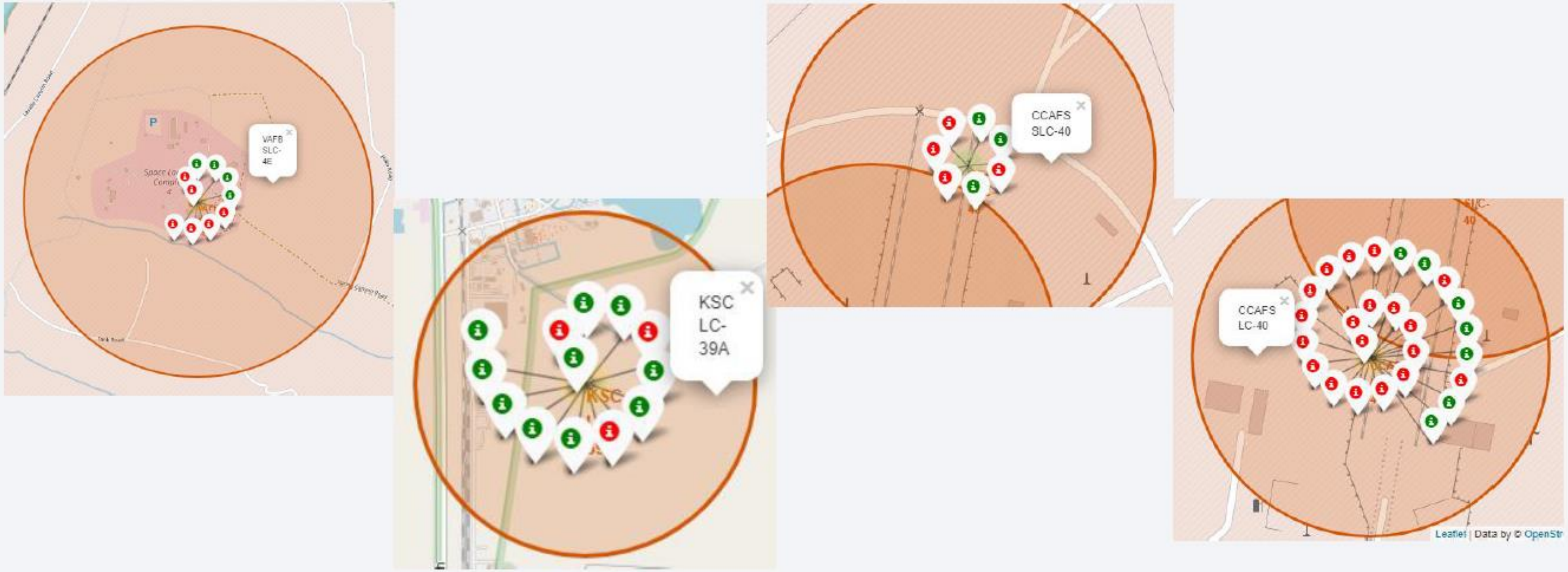
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>



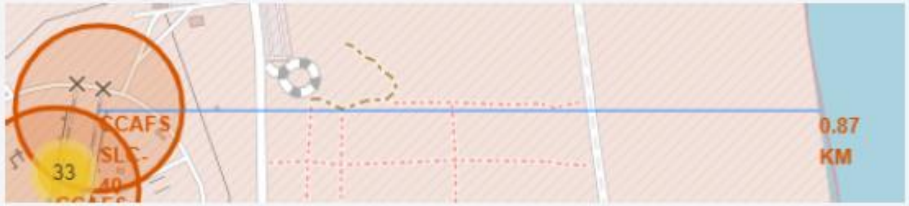
It is evident that the launch sites of Space X are situated along the coastline of the United States.

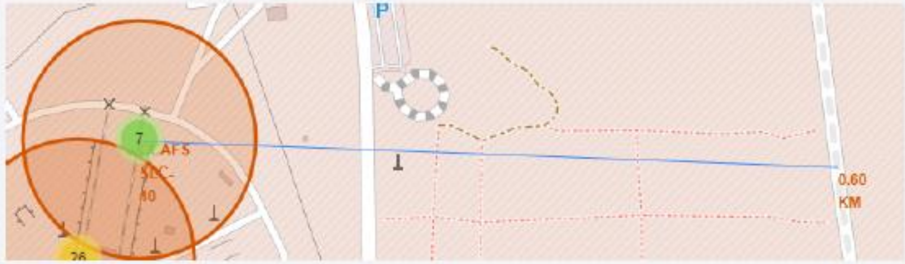
<Folium Map Screenshot 2>

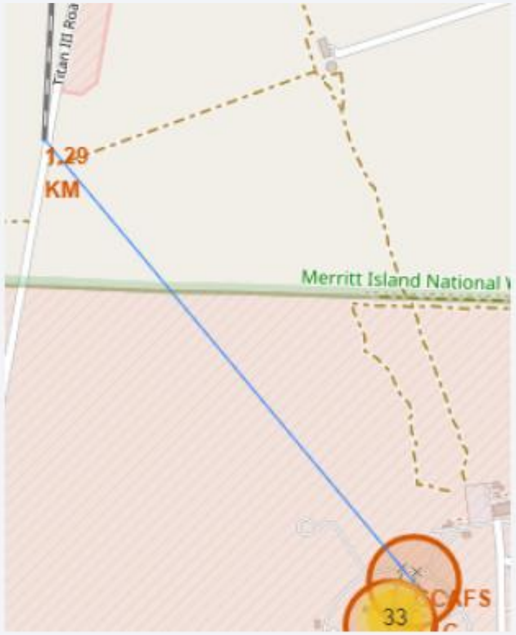


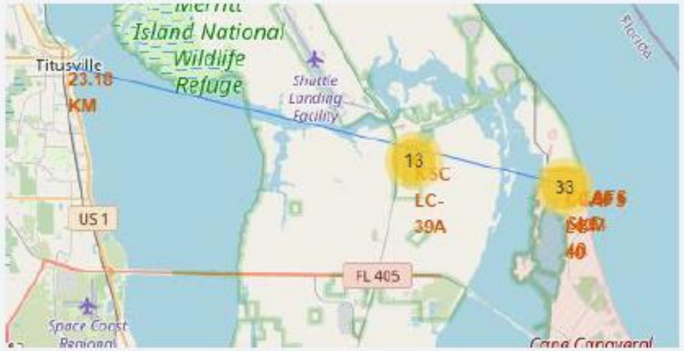
Greenmarker represents successful launches. Redmarker represents unsuccessful launches. We note that KSC LC-39A has a higher launch success rate.

<Folium Map Screenshot 3>

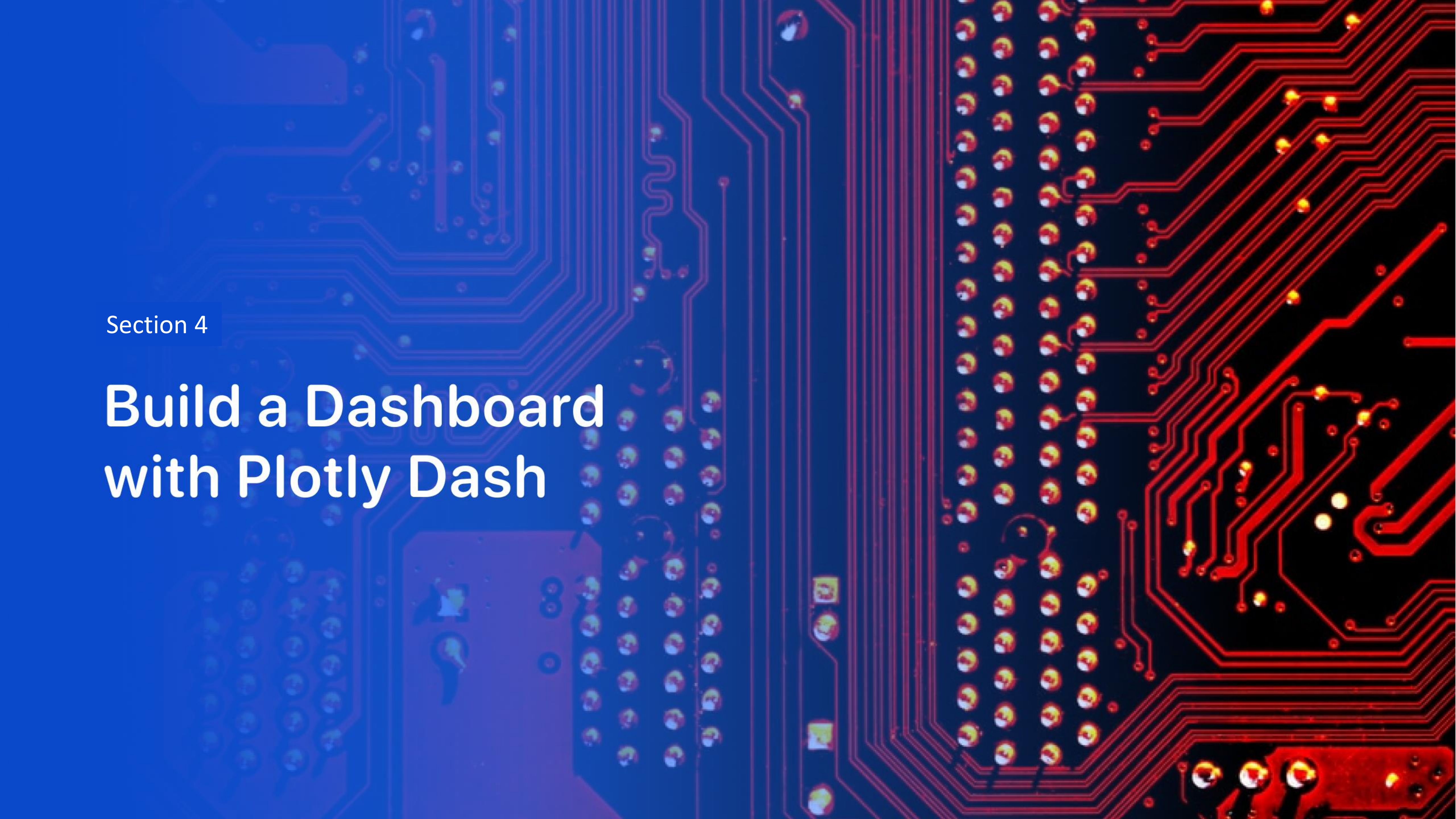








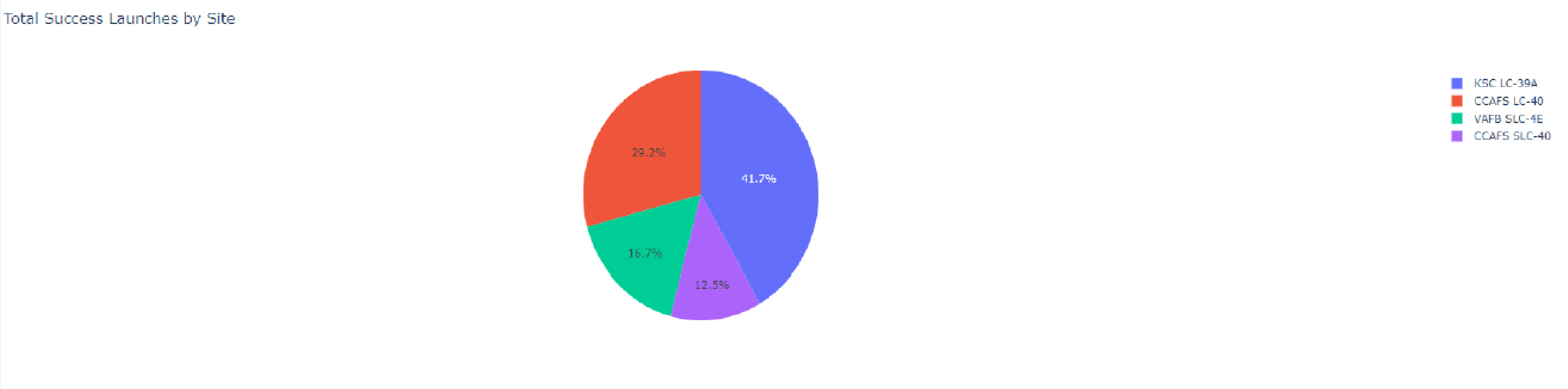
Is CCAFS SLC-40 in close proximity to railways ? Yes
Is CCAFS SLC-40in close proximity to highways ? Yes
Is CCAFS SLC-40in close proximity to coastline ? Yes
DoCCAFS SLC-40keeps certain distance away from cities ? No



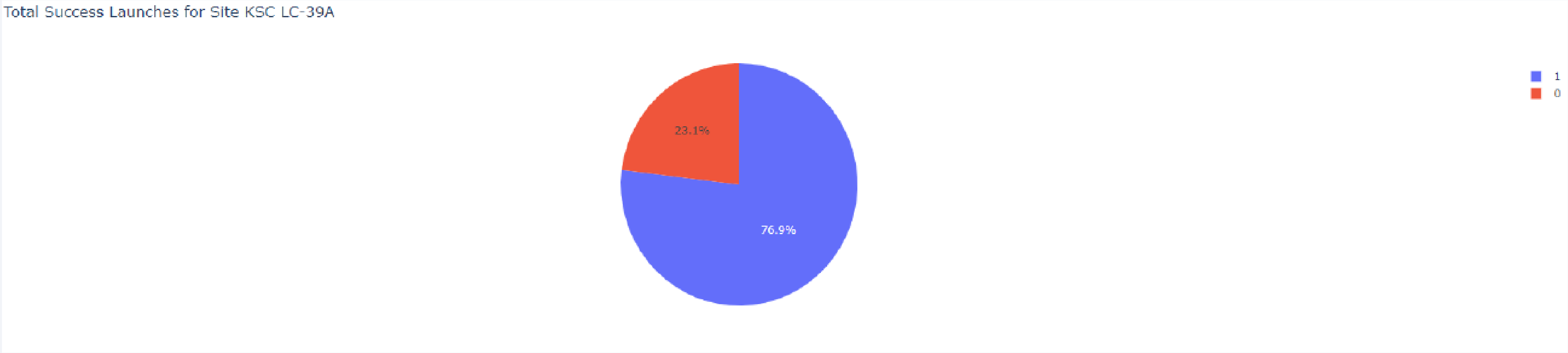
Section 4

Build a Dashboard with Plotly Dash

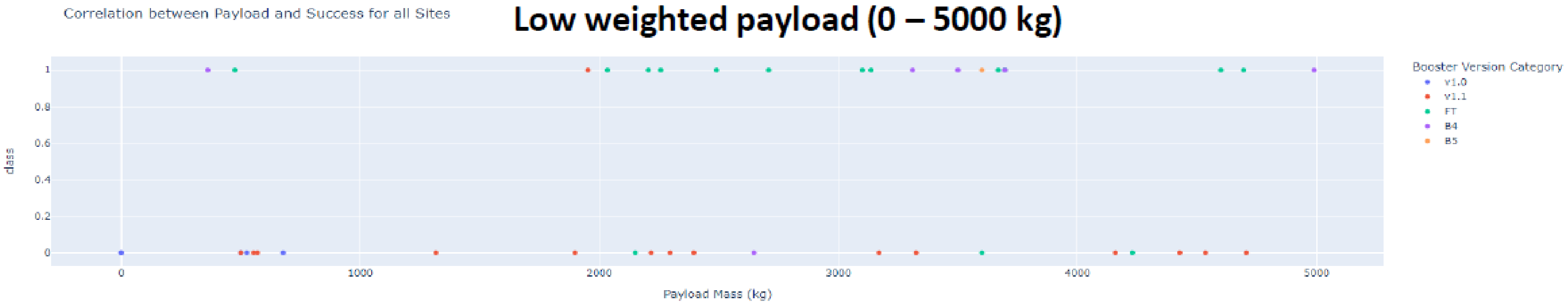
<Dashboard Screenshot 1>


Total Success Launches by Site
41.7%
29.2%
16.7%
12.5%
🟦 KSC LC-39A
🟥 CCAFS LC-40
🟩 VAFB SLC-4E
🟪 CCAFS SLC-40

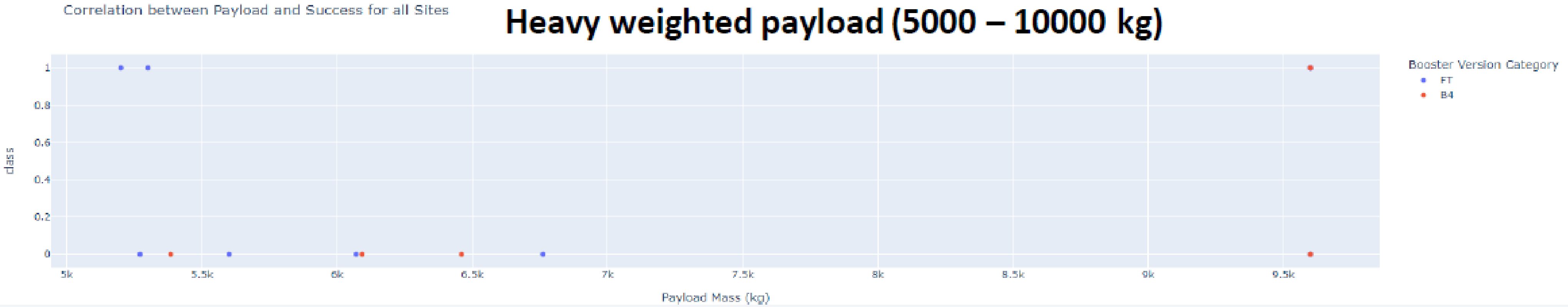
<Dashboard Screenshot 2>


Total Success Launches for Site KSC LC-39A
23.1%
76.9%
🟦 1
🟥 0

<Dashboard Screenshot 3>


Correlation between Payload and Success for all Sites

Low weighted payload (0 – 5000 kg)


Correlation between Payload and Success for all Sites

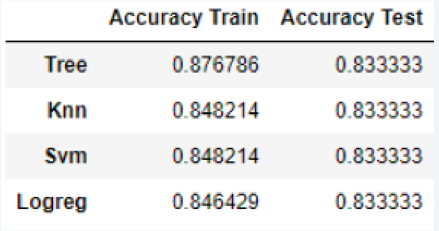
Heavy weighted payload (5000 – 10000 kg)

Low-weighted payloads have a better success rate than heavy-weighted payloads.

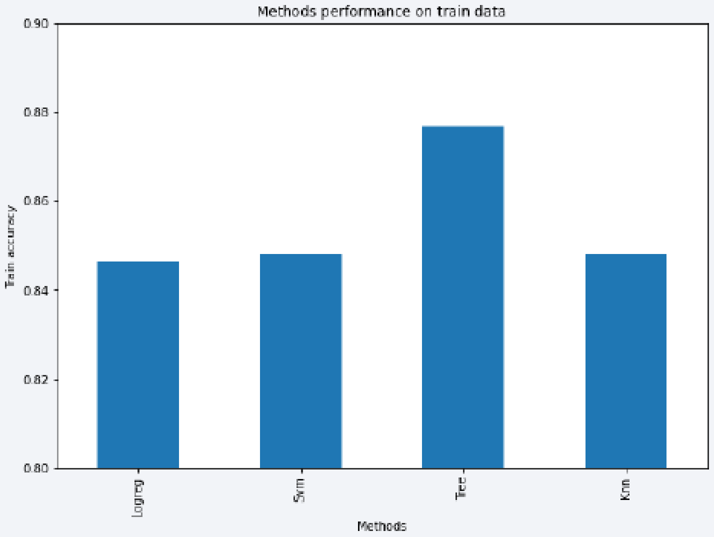
Section 5

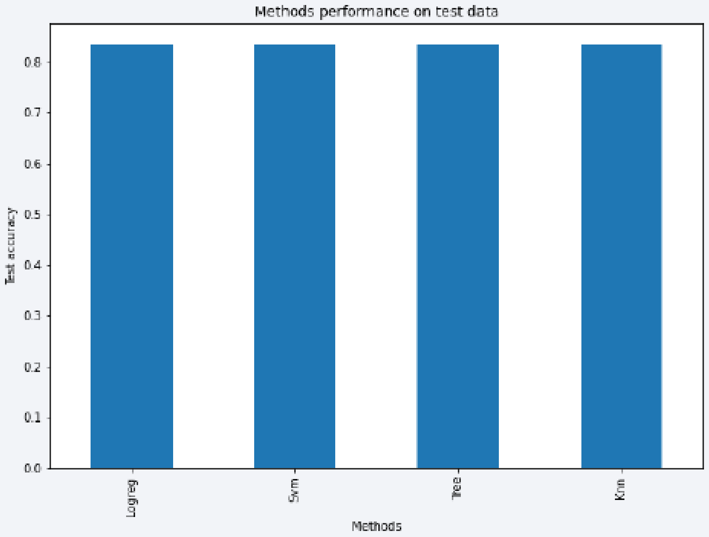
Predictive Analysis (Classification)

Classification Accuracy



	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333

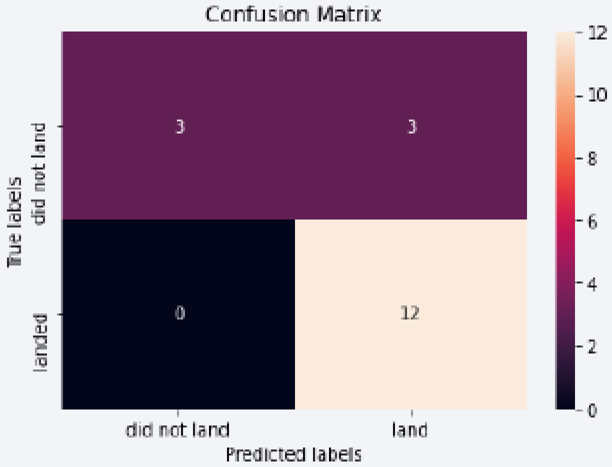




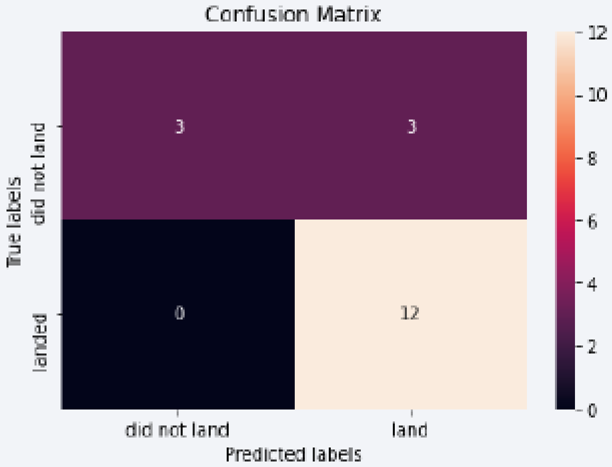
Based on the results of the accuracy test, all the methods exhibited similar performance. To make a more definitive decision, we may consider obtaining additional test data. However, if we had to choose a method right now, we would opt for the decision tree.

Confusion Matrix

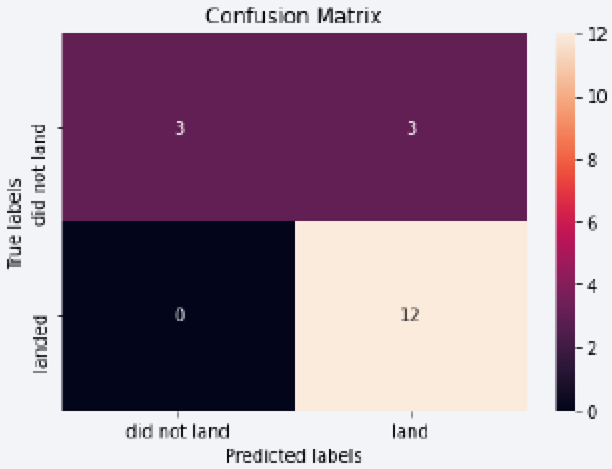
Logistic regression



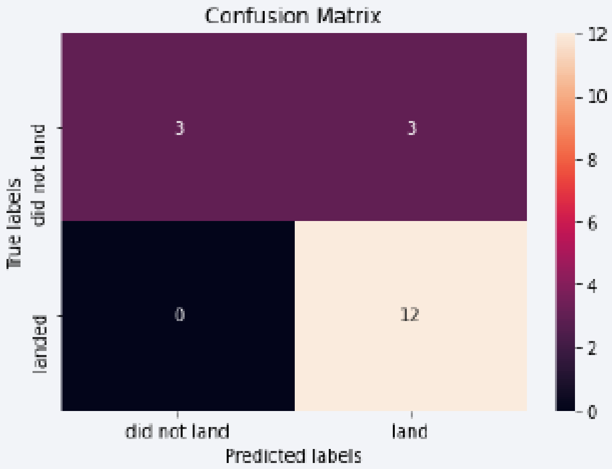
Decision Tree



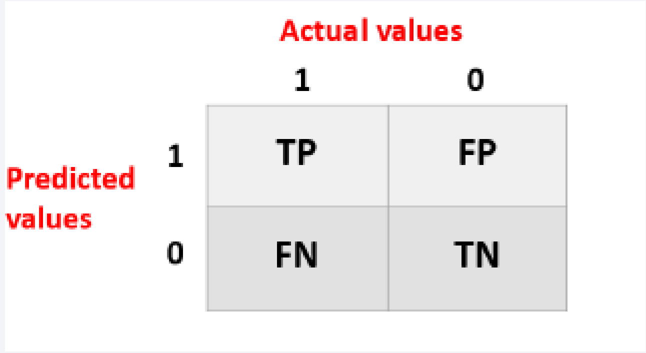
kNN



SVM



Given that the test accuracies are equivalent, the confusion matrices for these models are also identical. It is noteworthy that all the models have a tendency to produce false positives, which is the main issue that needs to be addressed.



		Actual values	
		1	0
Predicted values	1	TP	FP
	0	FN	TN

Conclusions

- Several factors contribute to the success of a space mission, including the launch site, the orbit, and the number of previous launches. The accumulation of knowledge gained from previous launches has been a significant factor in the progression from launch failures to successes.
- According to the data, GEO, HEO, SSO, and ES L1 are the orbits with the highest success rates. Payload mass is also an important consideration for the success of a mission, with different orbits requiring different payload masses. Generally, lighter payloads perform better than heavier ones.
- Despite KSC LC 39A being identified as the best launch site, the data currently available does not provide an explanation for this observation. Obtaining additional atmospheric or other relevant data could be useful in addressing this issue.
- For this dataset, we have determined that the Decision Tree Algorithm is the most suitable model, despite the test accuracies being identical for all the models used. This decision was based on the better train accuracy of the Decision Tree Algorithm."

Thank you!