IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sujeet Kumar Singh
August 22, 2025

Outline

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

Executive Summary

This project explores SpaceX Falcon 9 rocket launch data to assess first-stage landing outcomes and estimate launch costs. Conducted from the perspective of a competing aerospace firm—SpaceY—the analysis aims to extract actionable insights that could inform strategic decisions and technological improvements.

Below is an overview of the methodologies and key findings presented in this report.

□ **Methodology Overview**

- **Data Acquisition**: Compiled relevant launch data from publicly available sources
- **Data Preparation**: Cleaned and structured datasets for analysis
- **Exploratory Analysis**: Leveraged SQL and visual tools to uncover trends and patterns
- **Geospatial Mapping**: Developed an interactive map using Folium to visualize landing sites
- **Dashboard Development**: Created dynamic dashboards with Plotly Dash for real-time insights
- **Predictive Modeling**: Applied classification techniques to forecast landing success

📊 **Summary of Findings**

- **Exploratory Insights**: Identified correlations between launch parameters and landing outcomes
- **Interactive Demonstrations**: Included screenshots showcasing dashboard functionality and map interactivity
- **Predictive Results**: Presented model performance and key features influencing landing success

Introduction

This capstone project is a culmination of the IBM Data Science Professional Certificate program, designed to showcase practical expertise in data science and machine learning through the analysis of real-world data. The objective is to apply advanced analytical techniques and present the findings in a comprehensive report.

In this scenario, a fictional competitor to SpaceX—referred to as SpaceY—leverages publicly available data on SpaceX’s Falcon 9 rocket launches. The analysis focuses on evaluating the success rates of first-stage landings and estimating the cost per launch. SpaceY uses these insights to strategically position itself in competitive bidding for future launch contracts. While SpaceX promotes a launch cost of $62 million for Falcon 9, other providers typically exceed $165 million per launch, highlighting a significant cost advantage.

The project is executed using Python within Jupyter Notebooks, which serve as the primary environment for data collection, processing, and analysis. Both the notebooks and the final report (in PDF format) are hosted on my GitHub repository.

Key components of this report include:

- Data acquisition and preprocessing
- Exploratory data analysis (EDA) using visualization and SQL
- Interactive mapping with Folium
- Dashboard creation using Plotly Dash
- Development and evaluation of machine learning classification models

The report concludes with a comparative analysis of various ML algorithms in predicting the likelihood of successful Falcon 9 first-stage landings.

Section 1

Methodology

Methodology

The dataset for this project was sourced from the SpaceX REST API and the Falcon 9 launch records available on Wikipedia. The data underwent a thorough wrangling process, which included cleaning, formatting for visualization, and extracting relevant features for machine learning applications. These features were used to train classification models such as Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN).

Exploratory Data Analysis (EDA) was conducted using a combination of SQL queries and data visualization techniques to uncover patterns and insights. For geospatial and interactive analytics, Python libraries such as **Folium** and **Plotly Dash** were employed to create dynamic visual representations of the data.

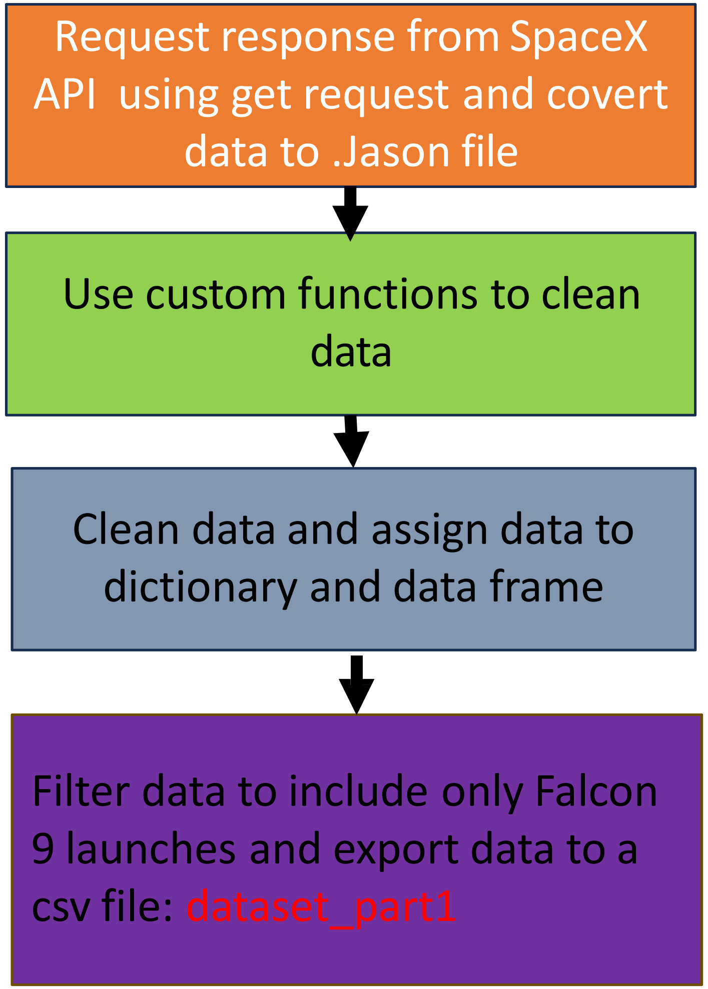
Finally, predictive modeling was carried out using classification algorithms implemented with **Scikit-learn**, aiming to forecast the success of Falcon 9 first-stage landings. Model performance was evaluated to determine the accuracy and reliability of each approach.

Data Collection

🔍 Key Steps in Data Collection and Visualization

1. **Data Acquisition:** Retrieved launch data from the SpaceX API and converted it into a structured .json format.
2. **Data Filtering and Preparation:** Scraped and refined the dataset to focus exclusively on Falcon 9 launches. The cleaned data was organized into a DataFrame and dictionary structure, then exported to a .csv file for further analysis.
3. **Data Visualization:** Generated plots and visual representations to explore trends and insights within the dataset.

Data Collection: SpaceX API

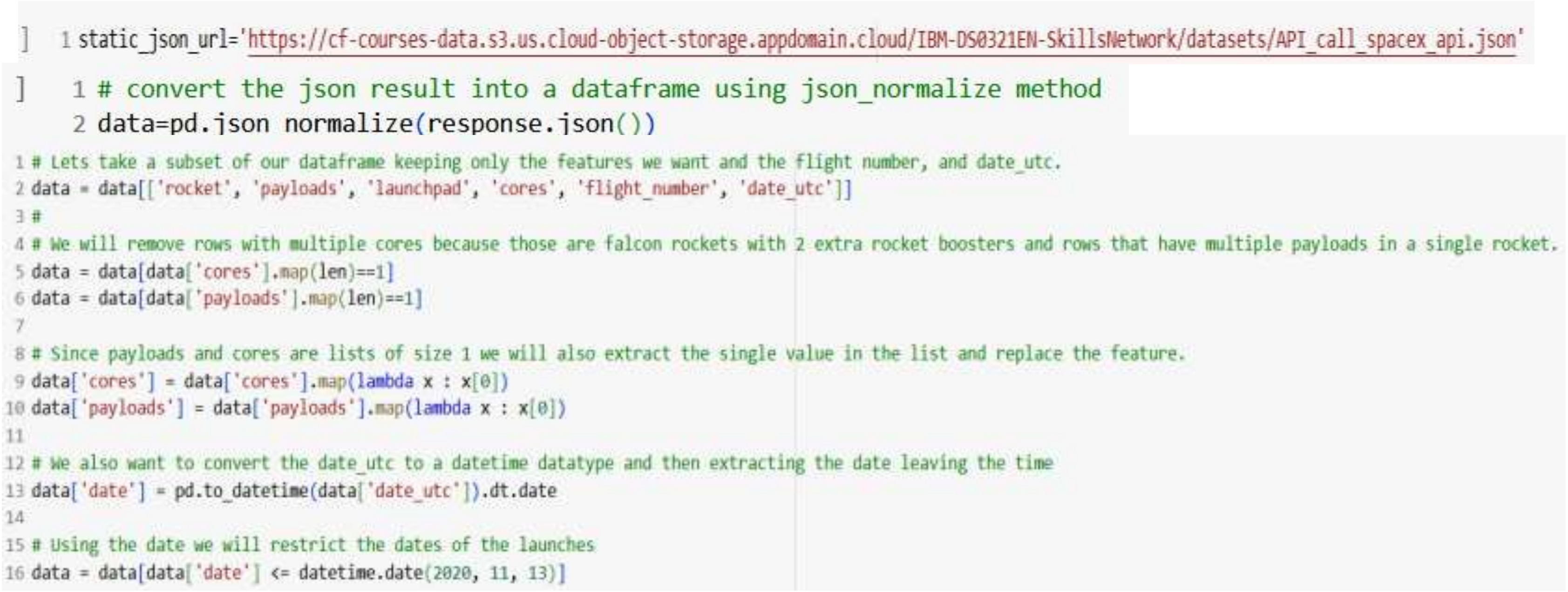


Request response from SpaceX API using get request and covert data to .Jason file

Use custom functions to clean data

Clean data and assign data to dictionary and data frame

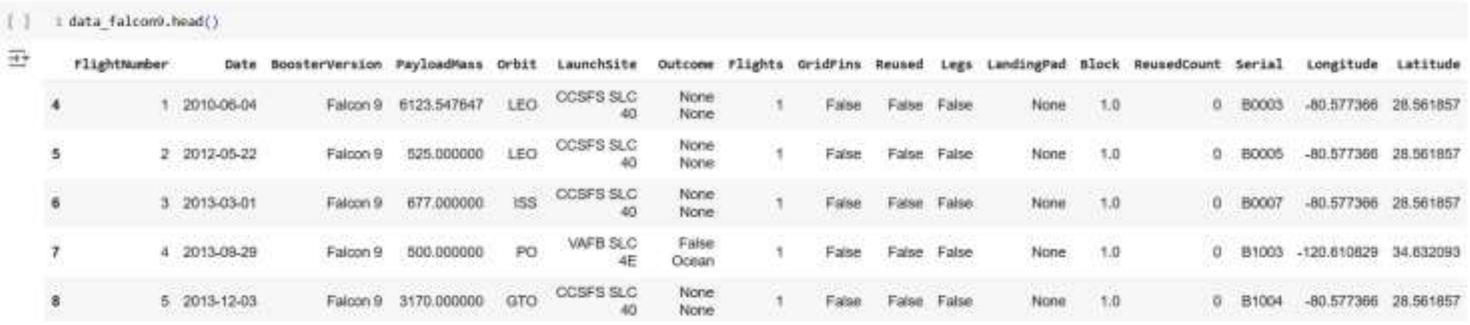
Filter data to include only Falcon 9 launches and export data to a csv file: dataset_part1



```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

```
# convert the json result into a dataframe using json_normalize method
data=pd.json_normalize(response.json())
```

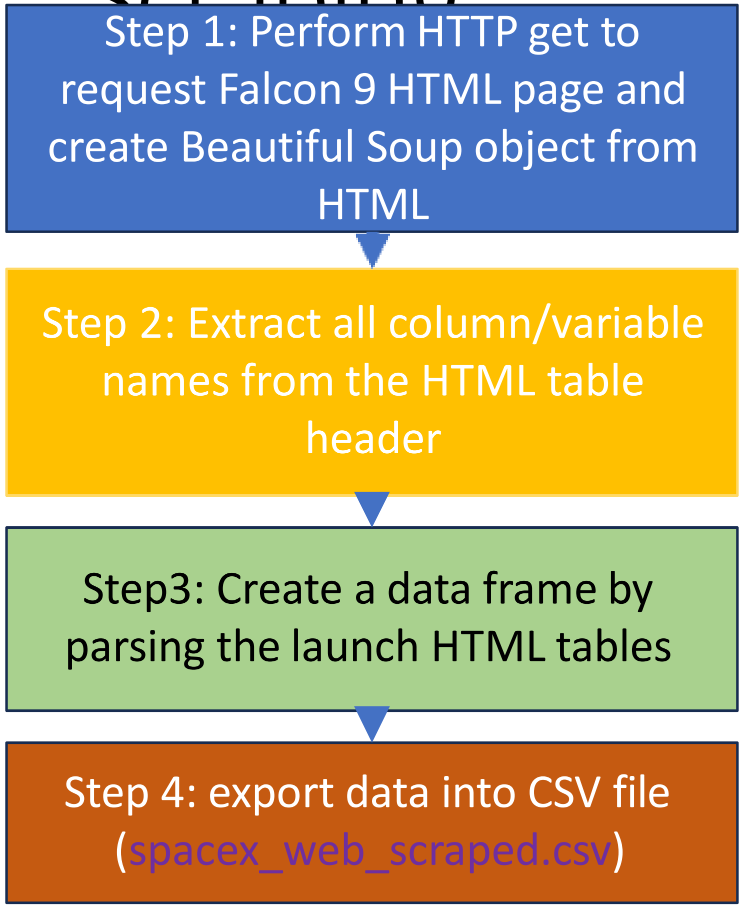
```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]
#
# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```



```
data_falcon9.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857

Data Collection: Scraping

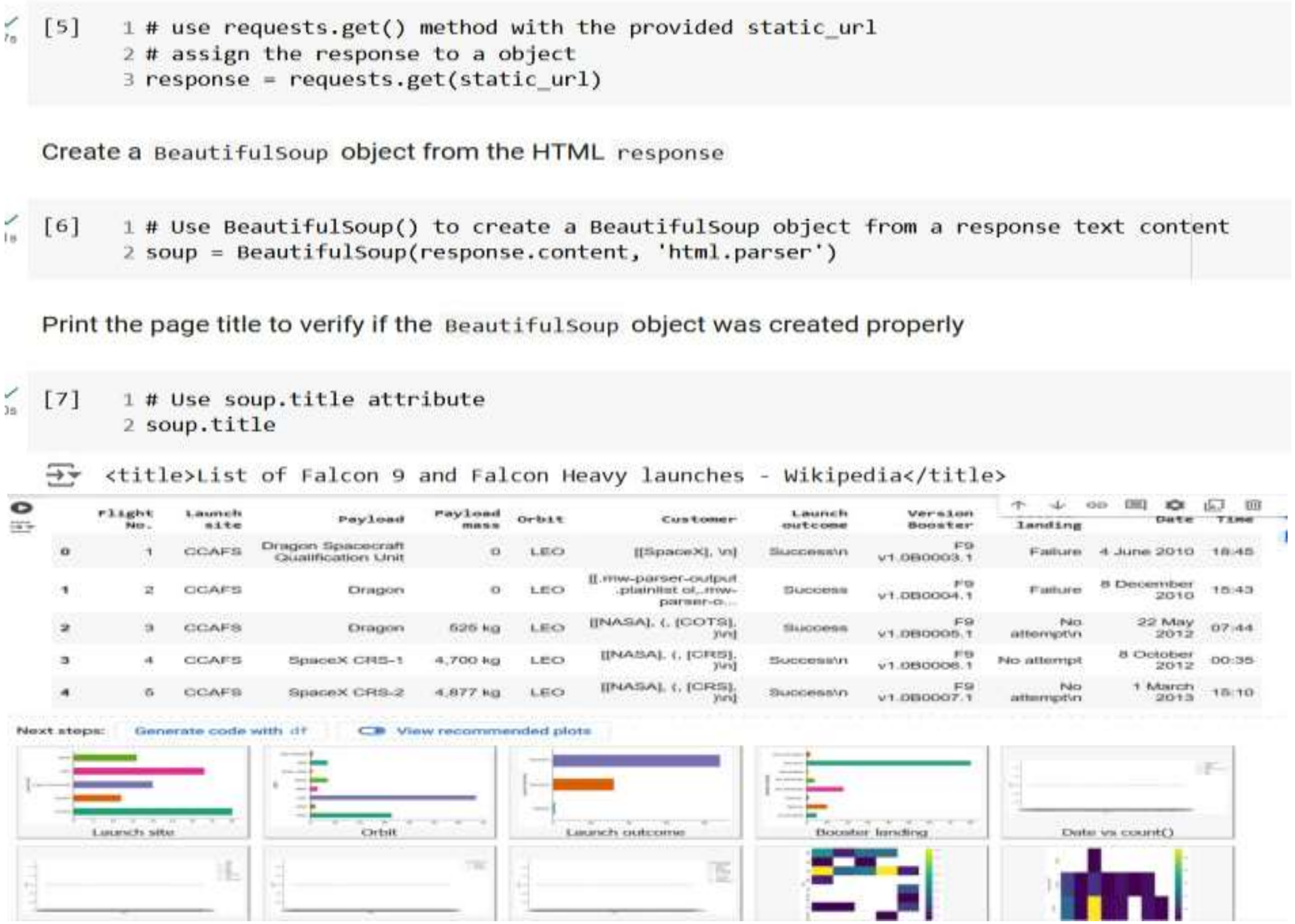


Step 1: Perform HTTP get to request Falcon 9 HTML page and create Beautiful Soup object from HTML

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

Step3: Create a data frame by parsing the launch HTML tables

Step 4: export data into CSV file (spacex_web_scraped.csv)



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

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

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

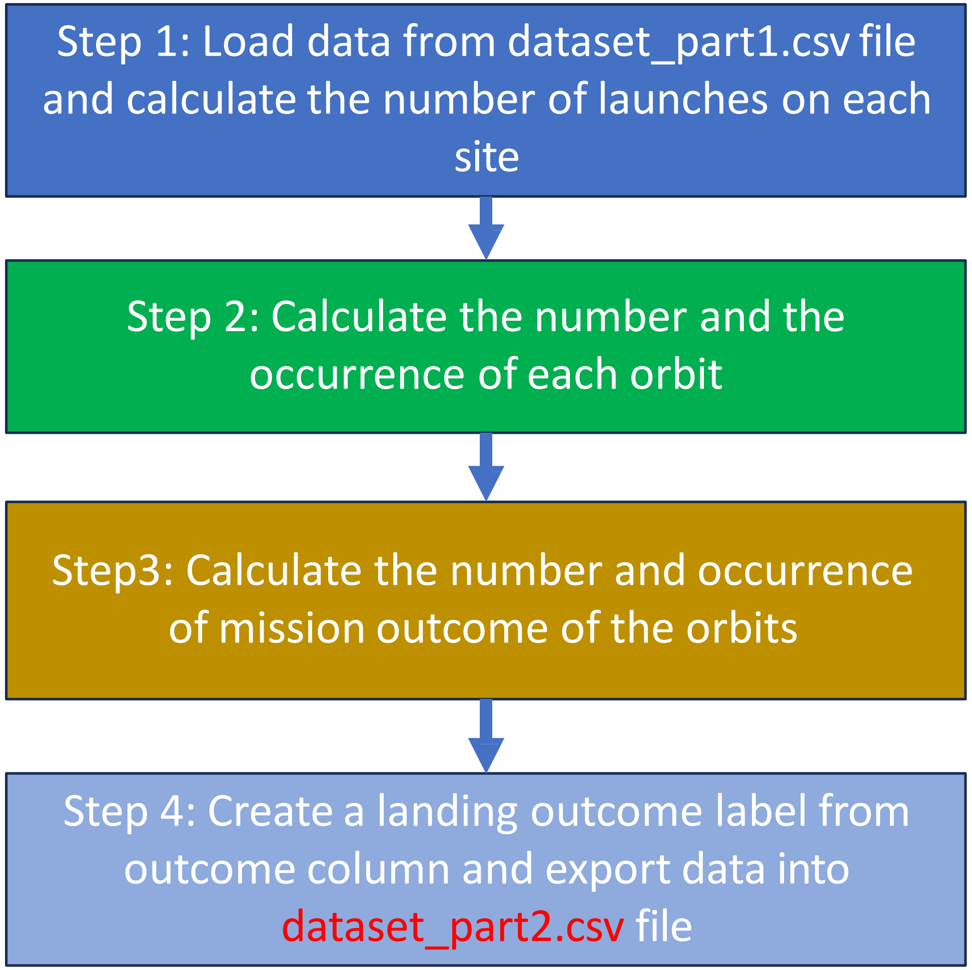
Print the page title to verify if the `BeautifulSoup` object was created properly

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

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	[[SpaceX], \n]	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	[[.mw-parser-output .plainlist ol,.mw-parser-o...	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	[[NASA], (, [COTS],)\n]	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	[[NASA], (, [CRS],)\n]	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	[[NASA], (, [CRS],)\n]	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

Data Collection: Data Wrangling

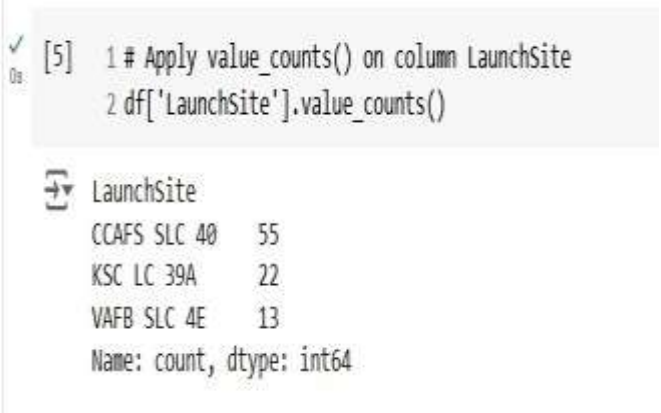


Step 1: Load data from dataset_part1.csv file and calculate the number of launches on each site

Step 2: Calculate the number and the occurrence of each orbit

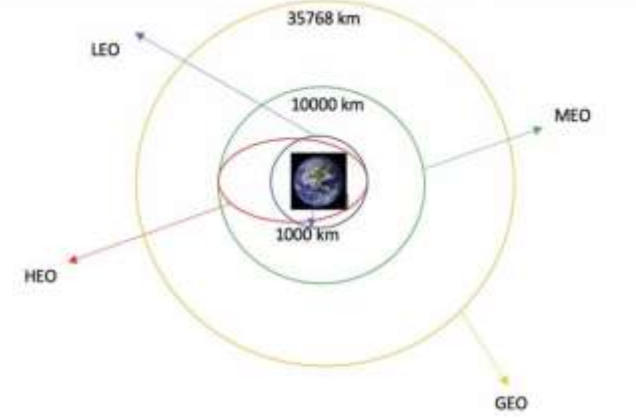
Step3: Calculate the number and occurrence of mission outcome of the orbits

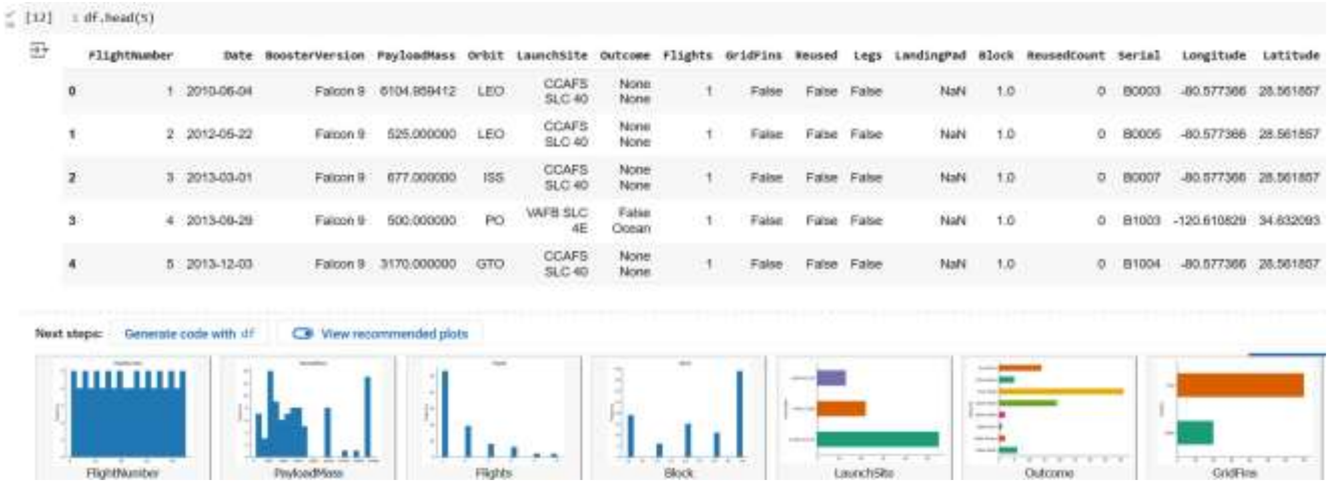
Step 4: Create a landing outcome label from outcome column and export data into dataset_part2.csv file



```
[5]  # Apply value_counts() on column LaunchSite
     df['LaunchSite'].value_counts()
```

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



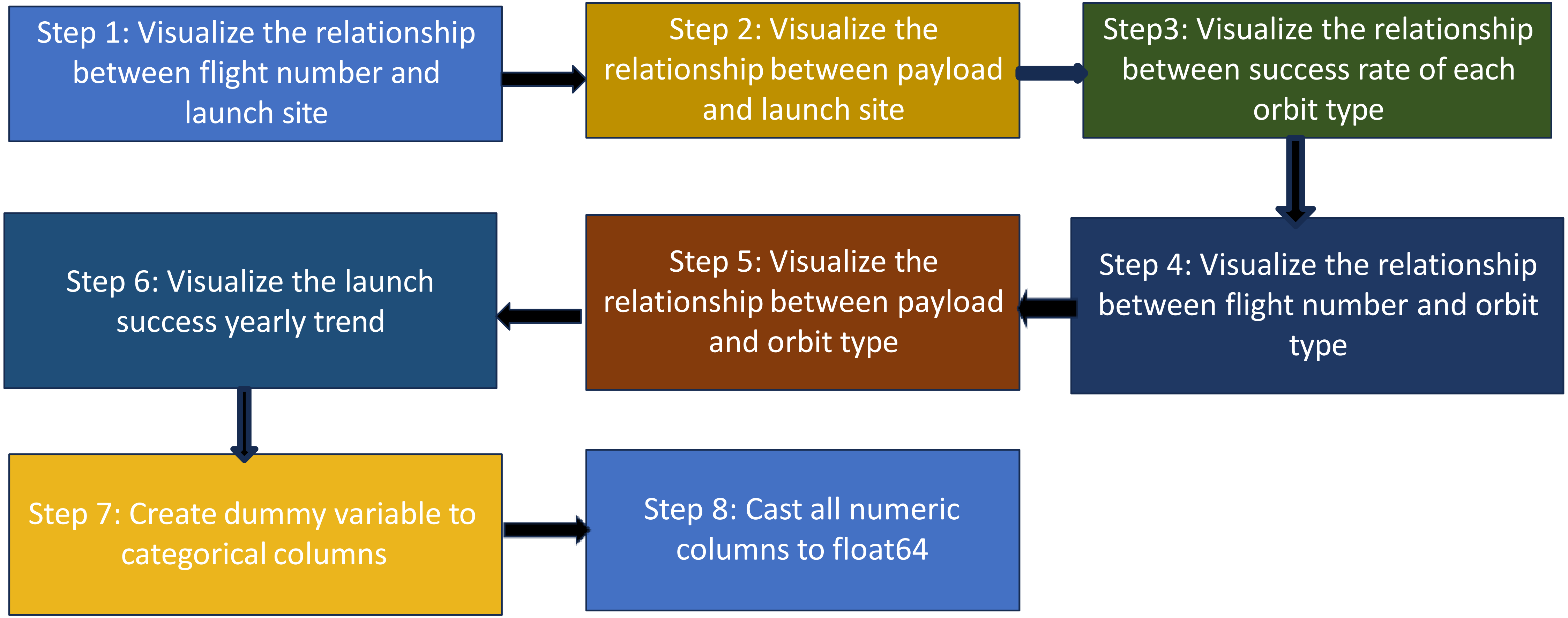


```
[12]  df.head(5)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857

EDA with Data Visualization

Use Matplotlib and Seaborn for data visualization



Step 1: Visualize the relationship between flight number and launch site

Step 2: Visualize the relationship between payload and launch site

Step3: Visualize the relationship between success rate of each orbit type

Step 4: Visualize the relationship between flight number and orbit type

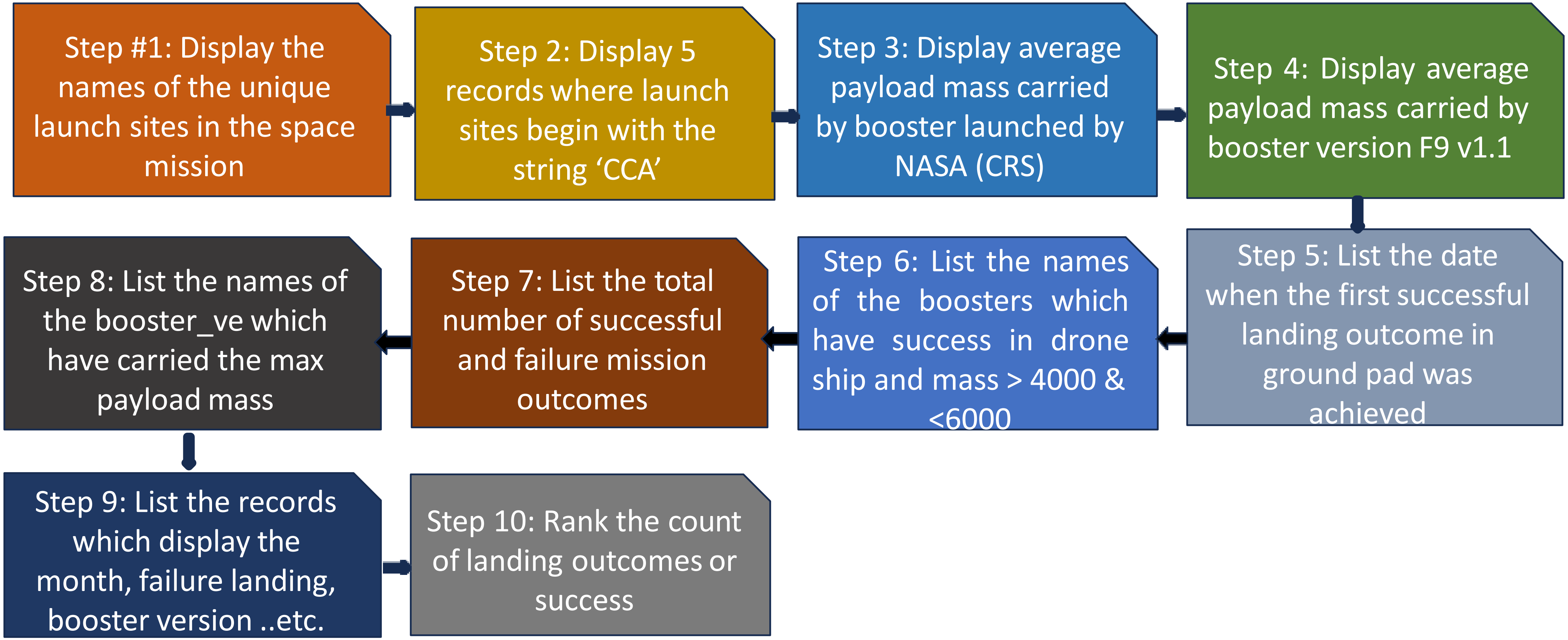
Step 5: Visualize the relationship between payload and orbit type

Step 6: Visualize the launch success yearly trend

Step 7: Create dummy variable to categorical columns

Step 8: Cast all numeric columns to float64

EDA with SQL



Step #1: Display the names of the unique launch sites in the space mission

Step 2: Display 5 records where launch sites begin with the string ‘CCA’

Step 3: Display average payload mass carried by booster launched by NASA (CRS)

Step 4: Display average payload mass carried by booster version F9 v1.1

Step 5: List the date when the first successful landing outcome in ground pad was achieved

Step 6: List the names of the boosters which have success in drone ship and mass > 4000 & <6000

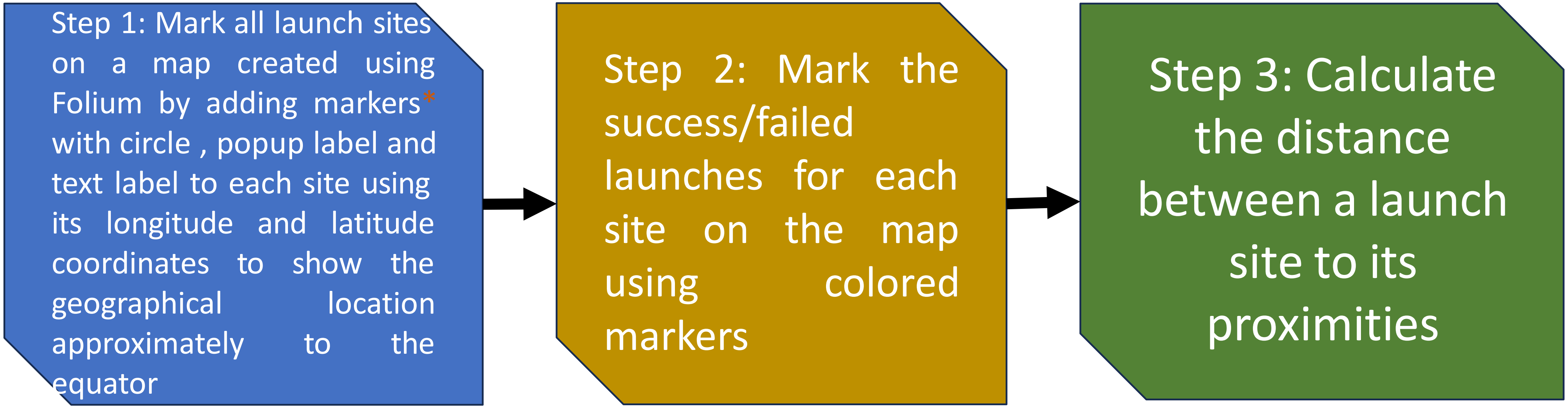
Step 7: List the total number of successful and failure mission outcomes

Step 8: List the names of the booster_ve which have carried the max payload mass

Step 9: List the records which display the month, failure landing, booster version ..etc.

Step 10: Rank the count of landing outcomes or success

Build an Interactive Map with Folium



Step 1: Mark all launch sites on a map created using Folium by adding markers* with circle , popup label and text label to each site using its longitude and latitude coordinates to show the geographical location approximately to the equator

Step 2: Mark the success/failed launches for each site on the map using colored markers

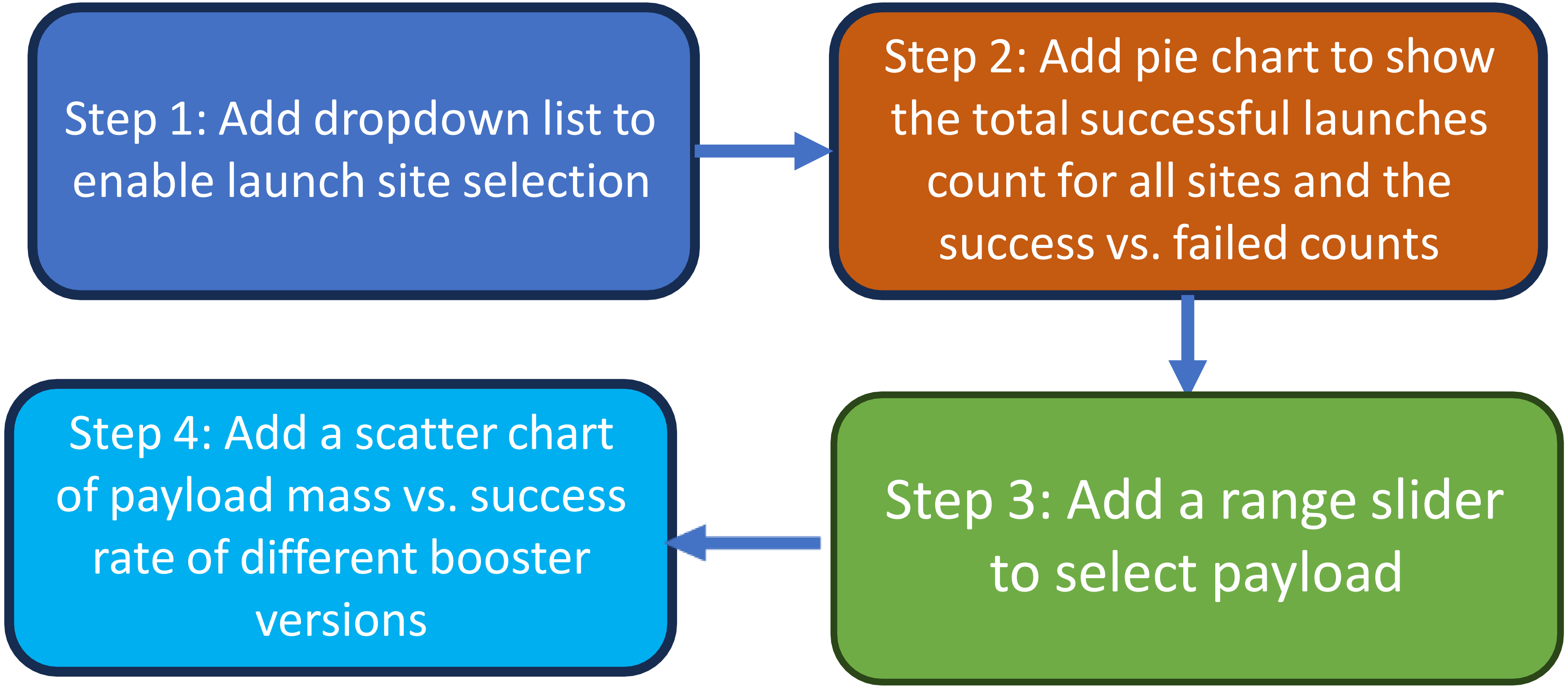
Step 3: Calculate the distance between a launch site to its proximities

*** Explanation:**

From the visual analysis of the launch site KSC LC-39A we can clearly see that it is:

- **relative close to railway (15.23 km)**
- **relative close to highway (20.28 km)**
- **relative close to coastline (14.99 km)**
- **Also the launch site KSC LC-39A is relative close to its closest city Titusville (16.32 km).**
- **Failed rocket with its high speed can cover distances like 15-20 km in few seconds. It could be potentially dangerous to populated areas.**

Build a Dashboard with Plotly Dash



Step 1: Add dropdown list to enable launch site selection

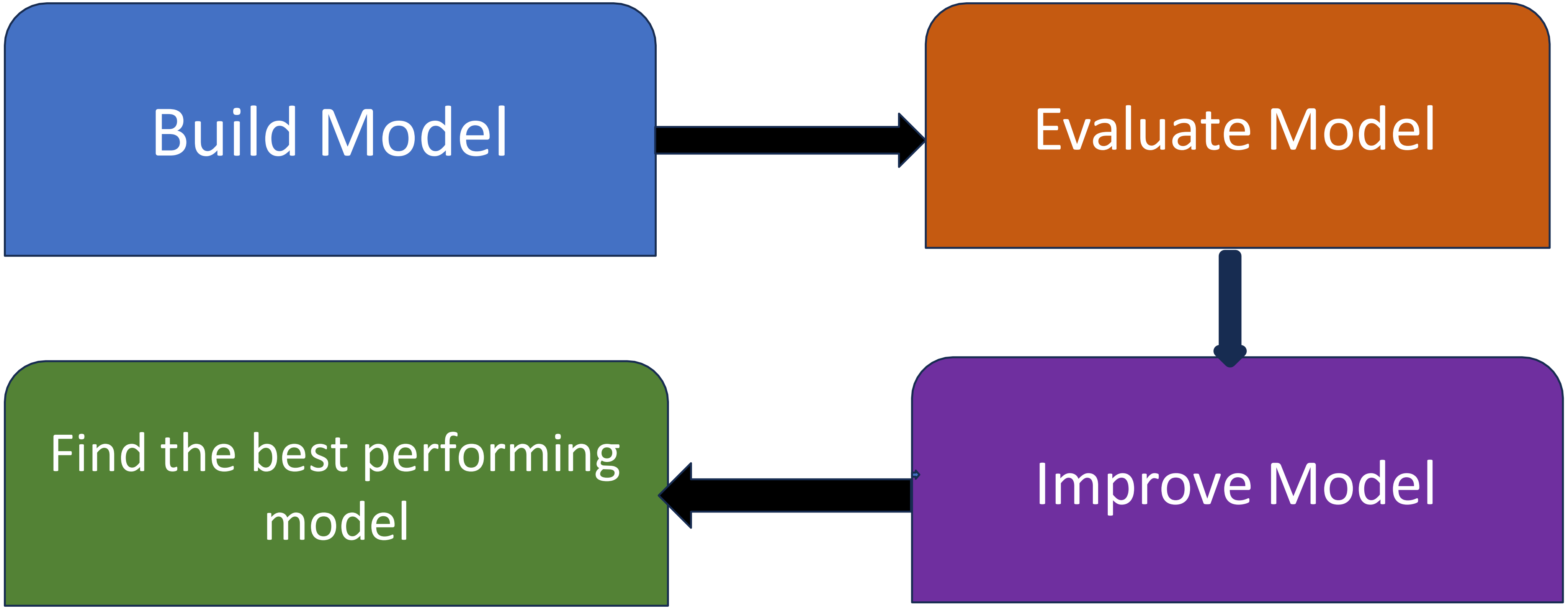
Step 2: Add pie chart to show the total successful launches count for all sites and the success vs. failed counts

Step 3: Add a range slider to select payload

Step 4: Add a scatter chart of payload mass vs. success rate of different booster versions

The dashboard is built using Dash web

Predictive Analysis (Classification): Overview



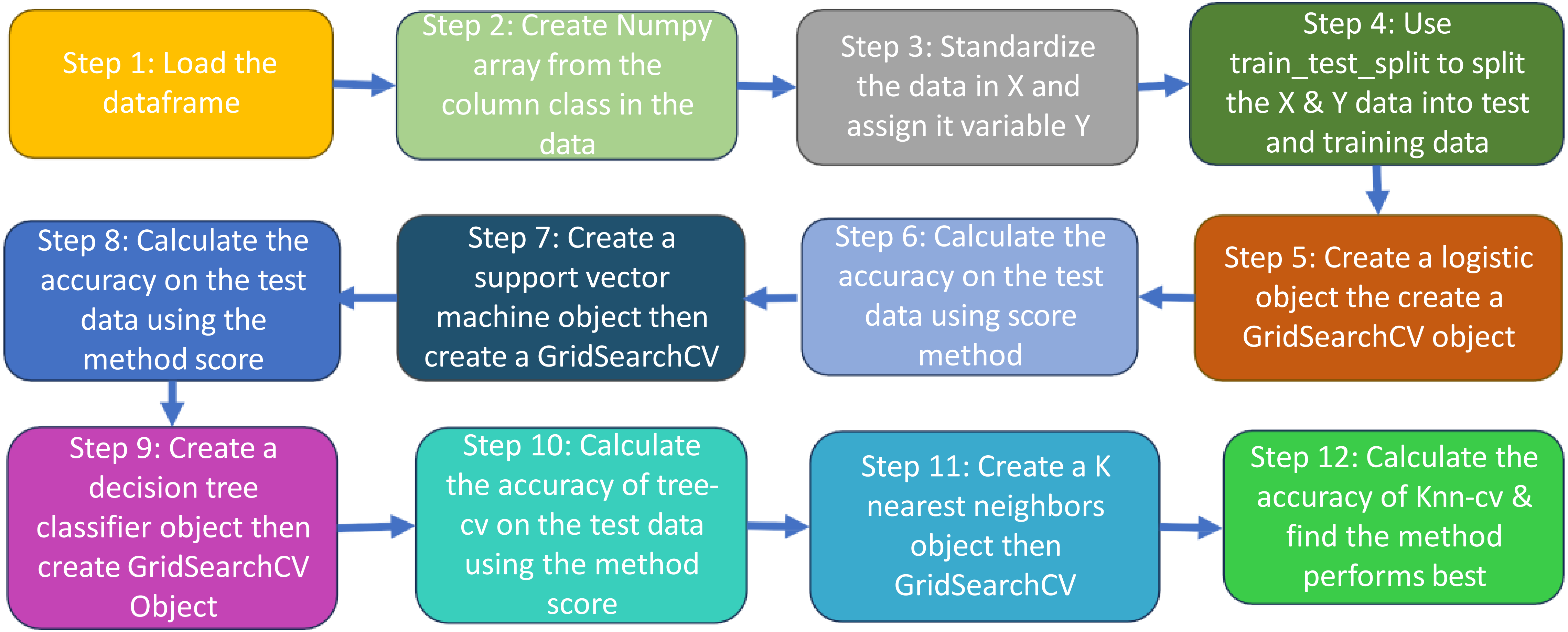
Build Model

Evaluate Model

Improve Model

Find the best performing model

Predictive Analysis (Classification) Steps



Step 1: Load the dataframe

Step 2: Create Numpy array from the column class in the data

Step 3: Standardize the data in X and assign it variable Y

Step 4: Use train_test_split to split the X & Y data into test and training data

Step 5: Create a logistic object the create a GridSearchCV object

Step 6: Calculate the accuracy on the test data using score method

Step 7: Create a support vector machine object then create a GridSearchCV

Step 8: Calculate the accuracy on the test data using the method score

Step 9: Create a decision tree classifier object then create GridSearchCV Object

Step 10: Calculate the accuracy of tree-cv on the test data using the method score

Step 11: Create a K nearest neighbors object then GridSearchCV

Step 12: Calculate the accuracy of Knn-cv & find the method performs best

Results

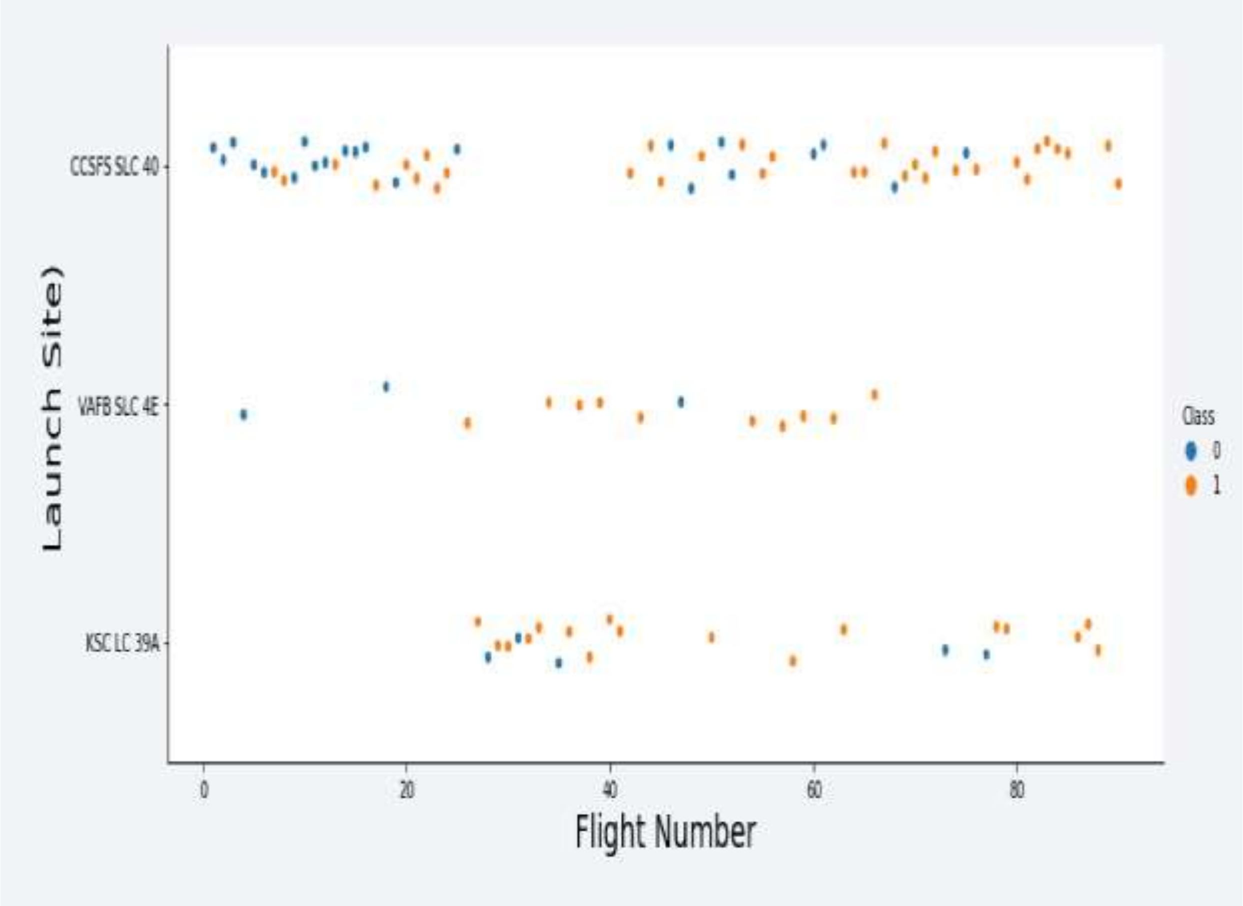
- Exploratory data analysis results.
- Interactive analytics demo in screenshots.
- Predictive analysis results.

Section 2

Insights drawn from EDA

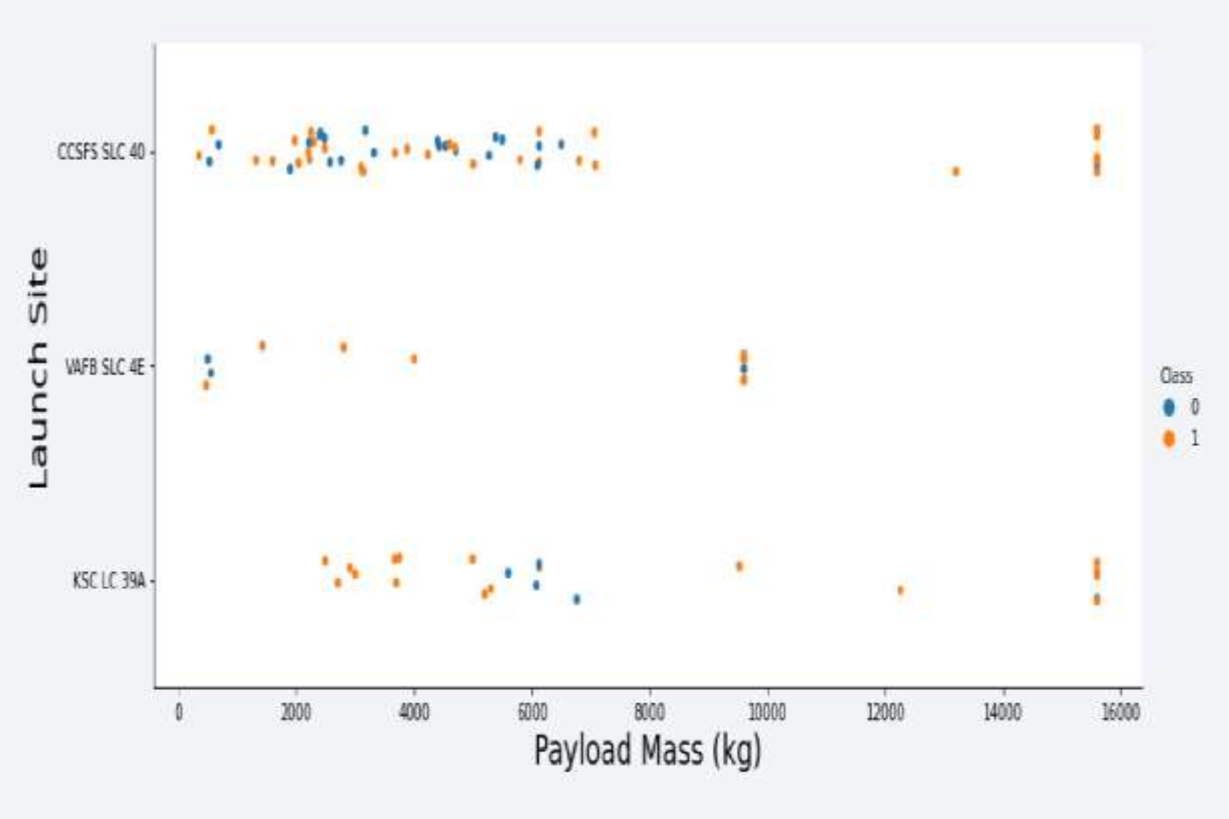
Flight Number vs. Launch Site

- **The majority of the flights were launched from the CCAFS SLC 40 sites.**
- **The VAFB SLC 4E and KSC LC 39A sites have higher success rates than other sites.**
- **Newer flights have higher success rates than older flights.**



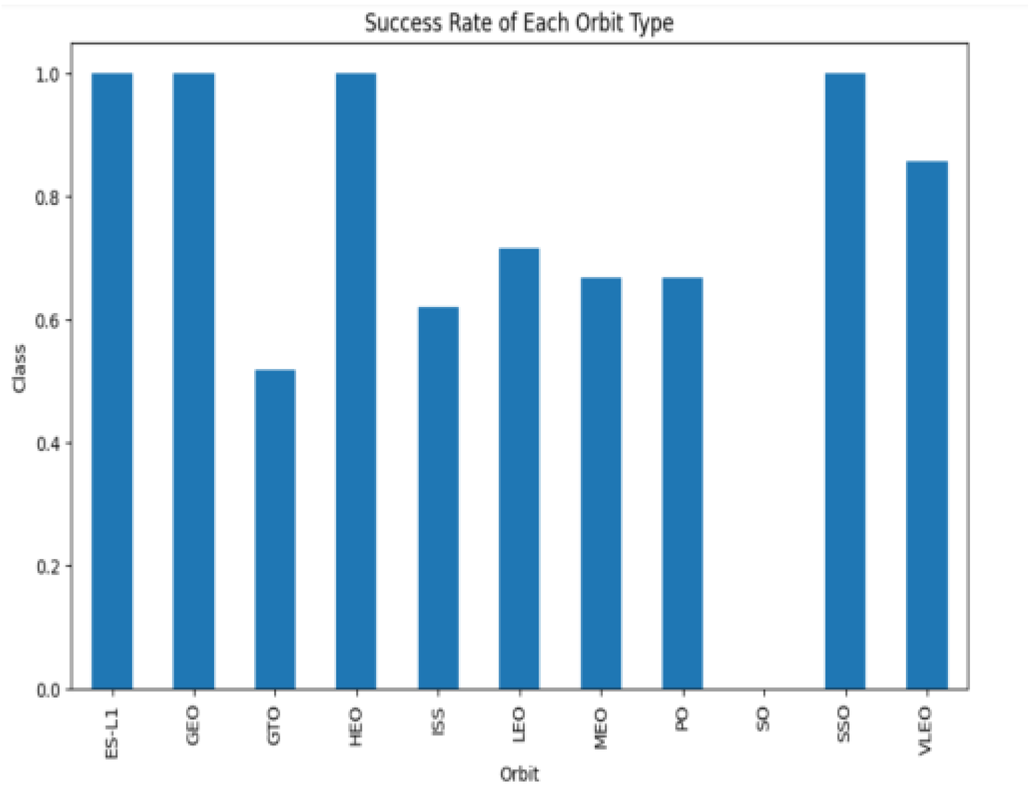
Payload Mass vs. Launch Site

- **The majority of the flights with payload mass above 7000 Kg were successful.**
- **KSC LC 39A success rate for payload mass under 5500 kg is 100%.**
- **For all launch sites the success rate is proportional to the payload mass.**



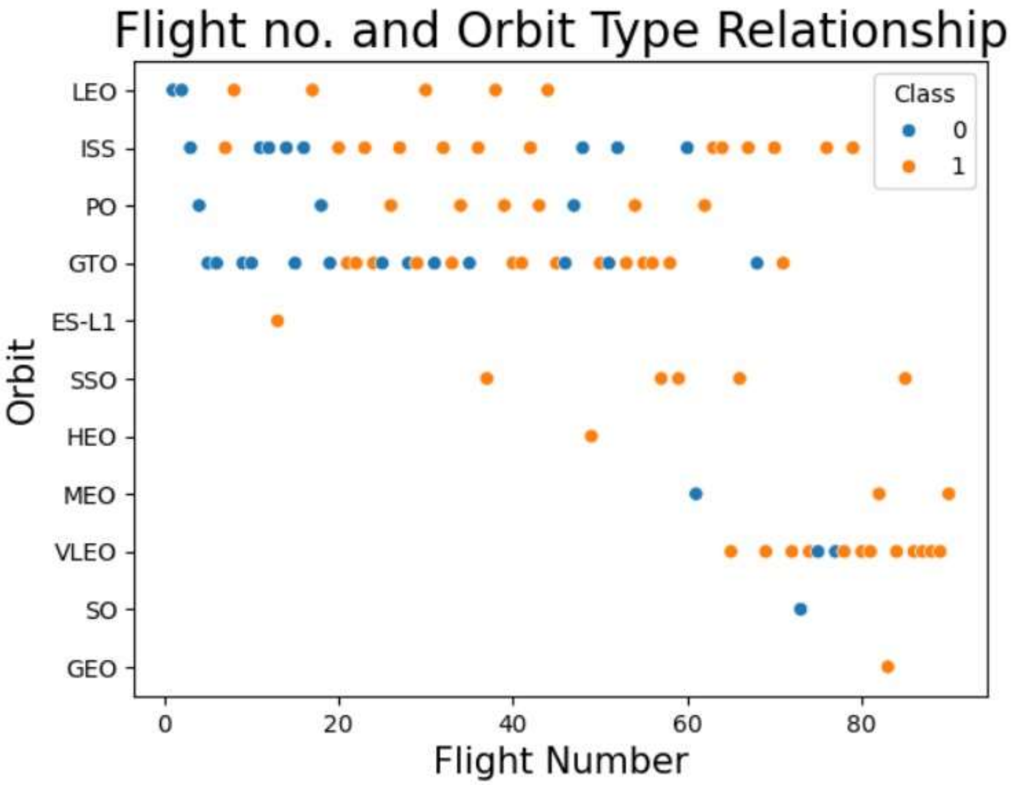
Success Rate vs. Orbit Type

- **The OS orbit has 0% success rate.**
- **The ELS-1, GEO, HEO and SSO orbits have 100% success rate.**
- **Orbits GTO, ISS, LEO, MEO and PO success rate is higher than 50% and less than 75%.**



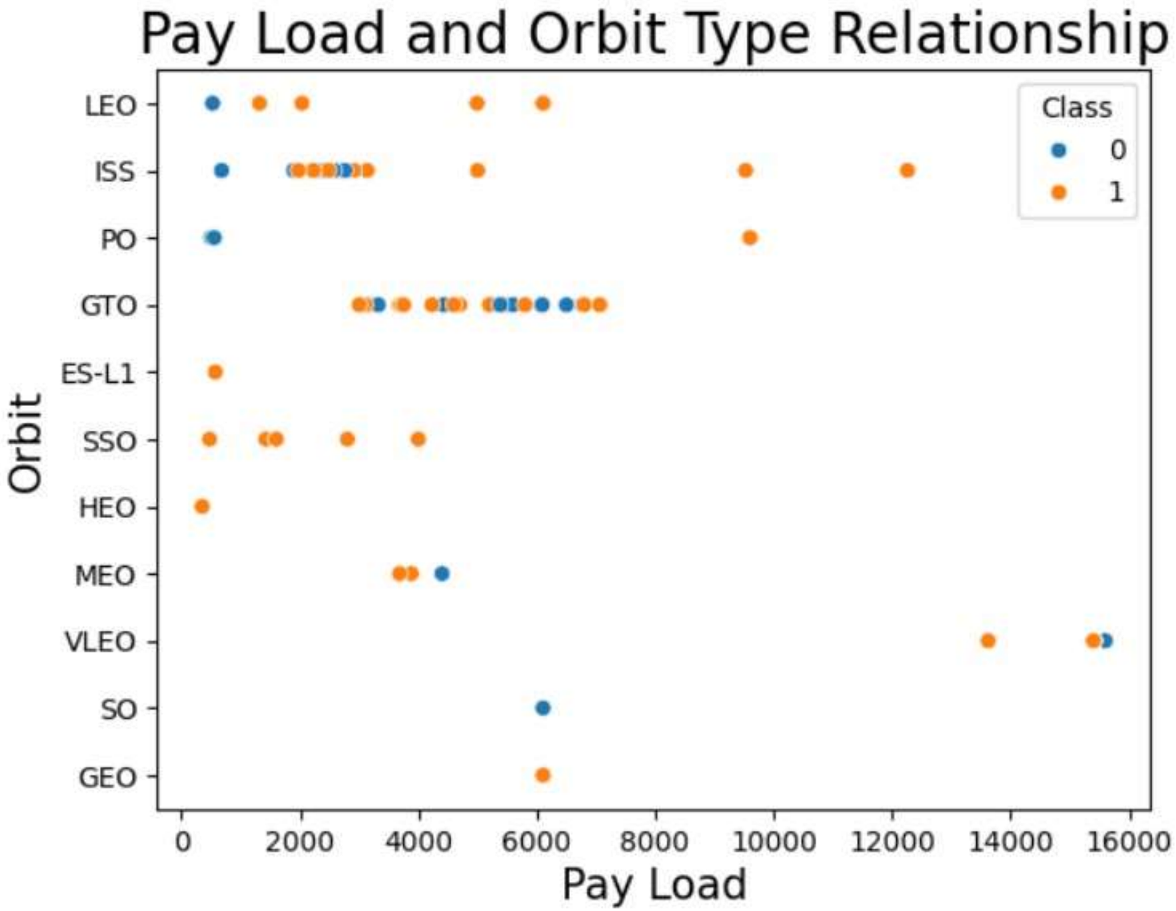
Flight Number vs. Orbit Type

- **The majority of the flights were launches to the ISS and GTO orbits.**
- **The data suggests that there is no relationship between the flight number and the orbit type.**



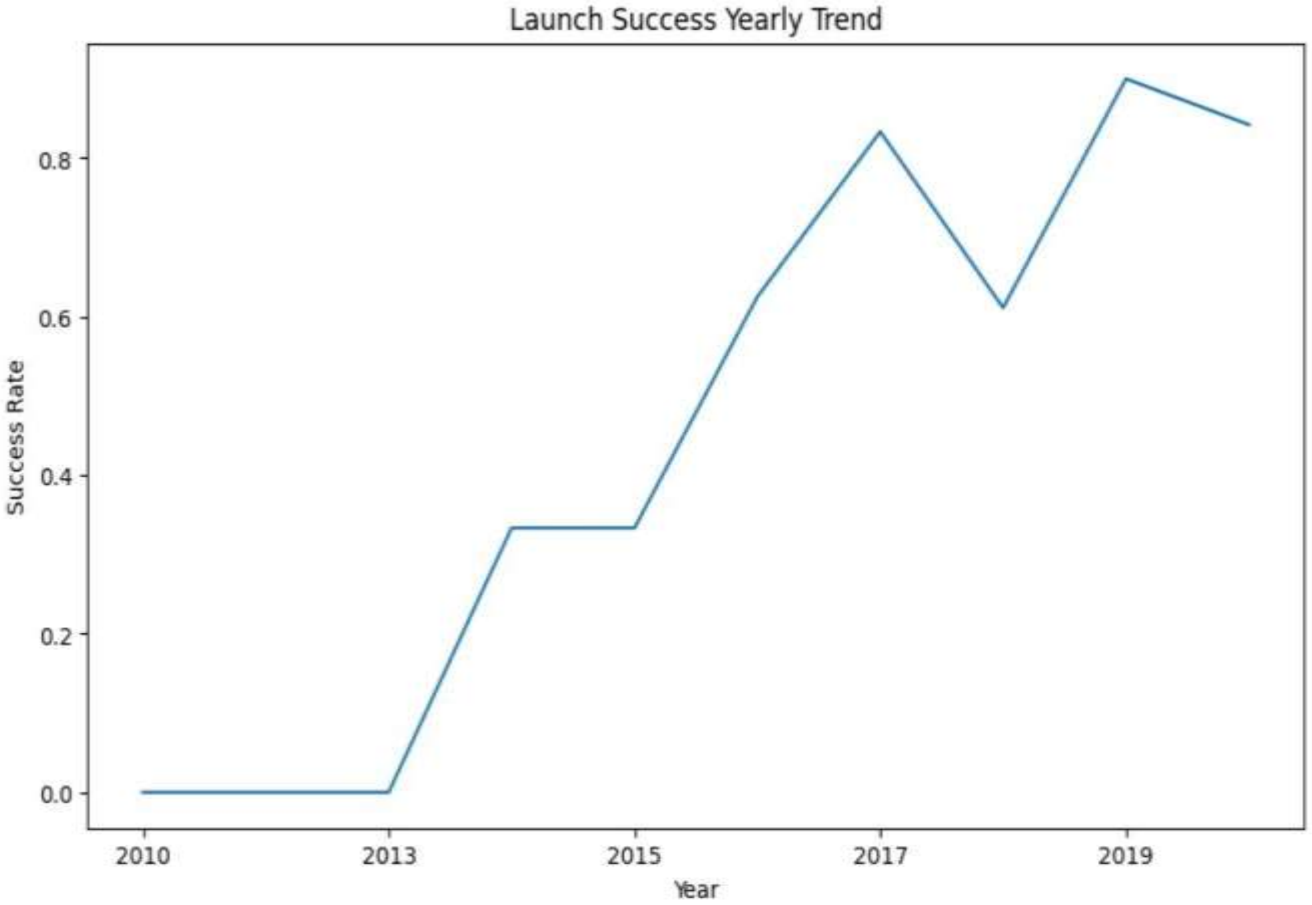
Payload Mass vs. Orbit Type

- **Payload masses above 10000 Kg were placed in PO, ISS and LEO orbits.**
- **Payload masses above 4000 and less than 8000 Kg were placed in the GTO orbit.**



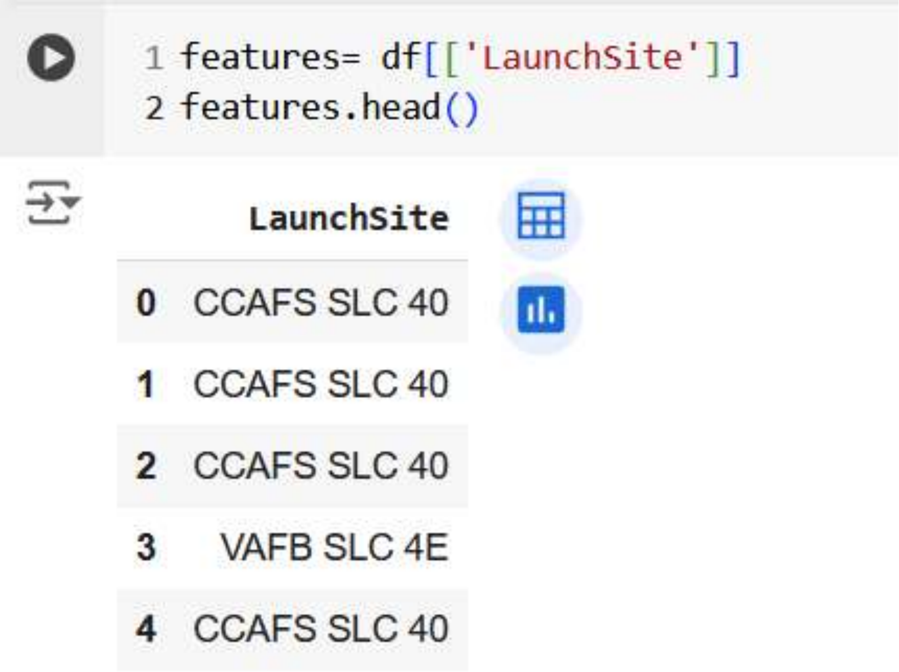
Launch Success Yearly Trend

- **The launches success rate increased steadily since 2013.**
- **The increase in the success rate between 2013 and 2017 was linear.**
- **During 2018 there was a drop in the launches success rate.**



All Launch Site Names

The names of the unique launch sites and the query structure for obtaining these sites is shown below.

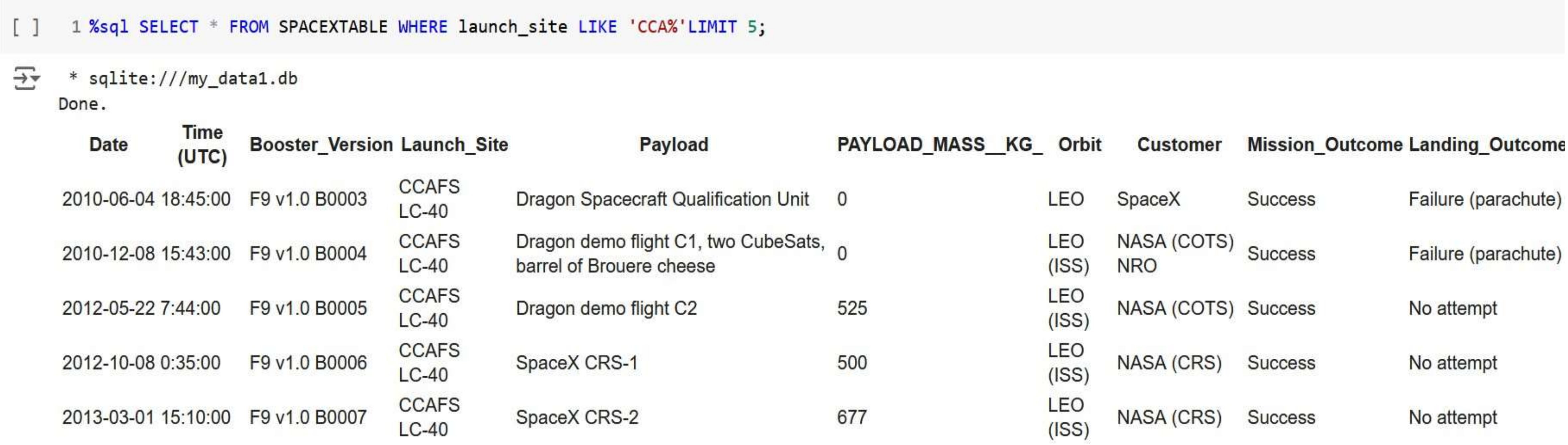


```
features= df[['LaunchSite']]
features.head()
```

```
     LaunchSite
0  CCAFS SLC 40
1  CCAFS SLC 40
2  CCAFS SLC 40
3   VAFB SLC 4E
4  CCAFS SLC 40
```

Launch Site Names Begin with 'CCA'

5 records for launch sites begin with the string 'CCA’ and the query used for obtaining the information is shown below.



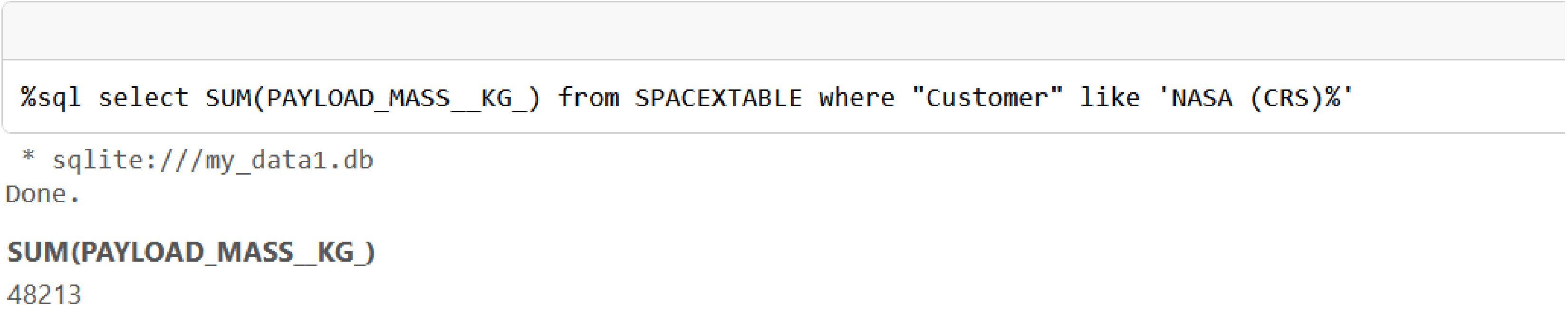
```
%sql SELECT * FROM SPACEXTABLE WHERE launch_site LIKE 'CCA%'LIMIT 5;
```

```
 * 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

Total Payload Mass

- **The calculated total payload mass carried by boosters from NASA site =45596 Kg.**
- **The query for obtaining the total payload mass is shown below.**



```
%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Customer" like 'NASA (CRS)%'
```

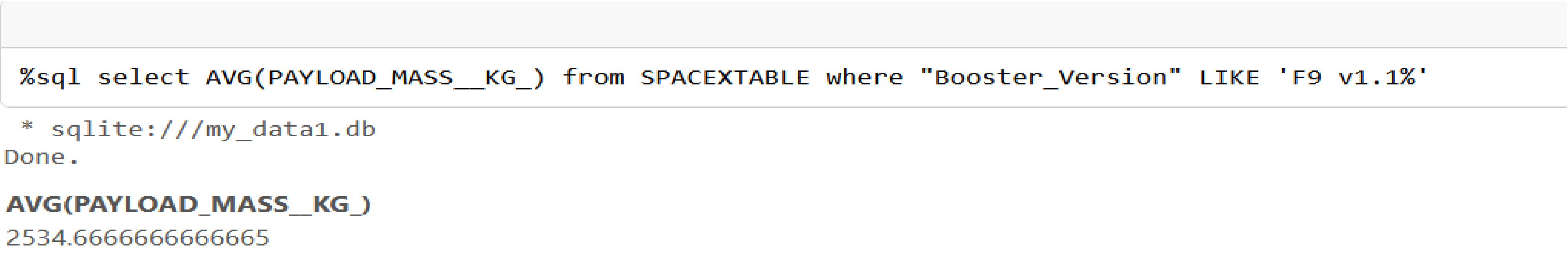
```
 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)

48213

Average Payload Mass by F9 v1.1

- **The average payload mass carried by booster version F9 v1.1=2534.7 Kg.**
- **Furthermore, the query used to calculate the average payload mass carried by booster F9 v1.1 is shown below.**



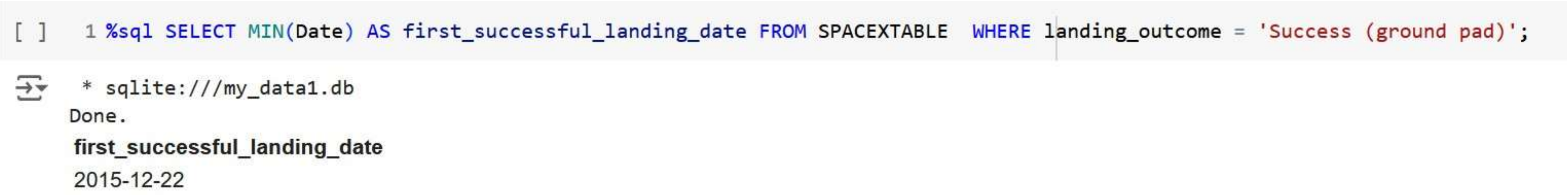
```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Booster_Version" LIKE 'F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- **The first successful landing outcome on a ground pad was in 2015-12-22.**
- **The query for obtaining this result is shown below.**

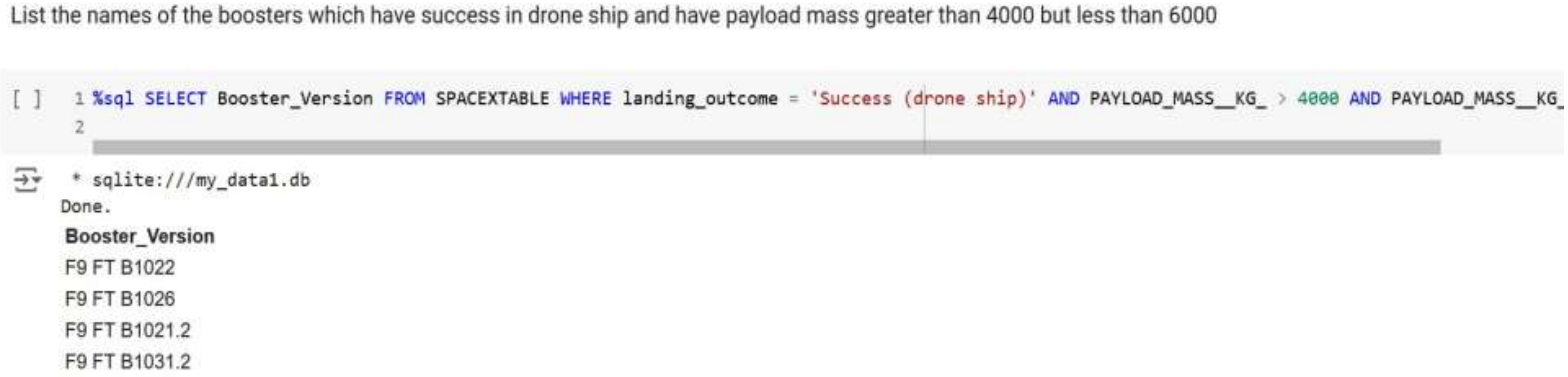


```
%sql SELECT MIN(Date) AS first_successful_landing_date FROM SPACEXTABLE  WHERE landing_outcome = 'Success (ground pad)';
```

```
 * sqlite:///my_data1.db
Done.
first_successful_landing_date
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- **List of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 is shown below.**
- **The query used in obtaining this information is shown below.**



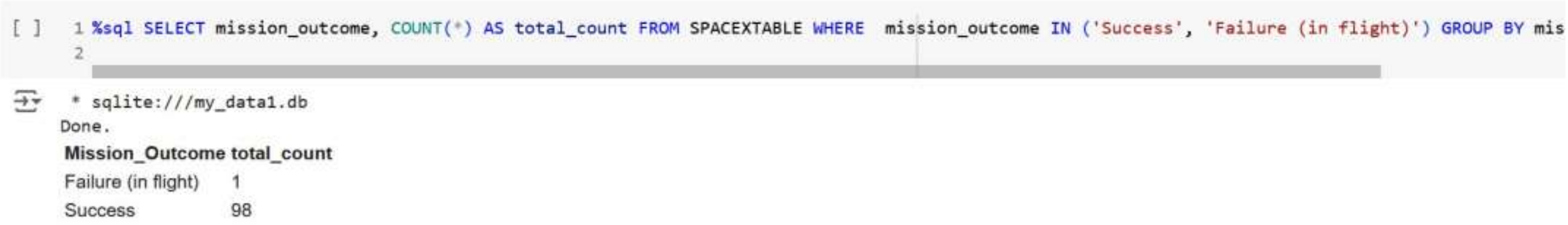
List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE landing_outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_
```

```
 * 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

- **The total number of successful and failed missions is as follows:**
  - **Failure (in flight)= 1**
  - **Successful number of flights= 98**
- **The query result is shown below.**

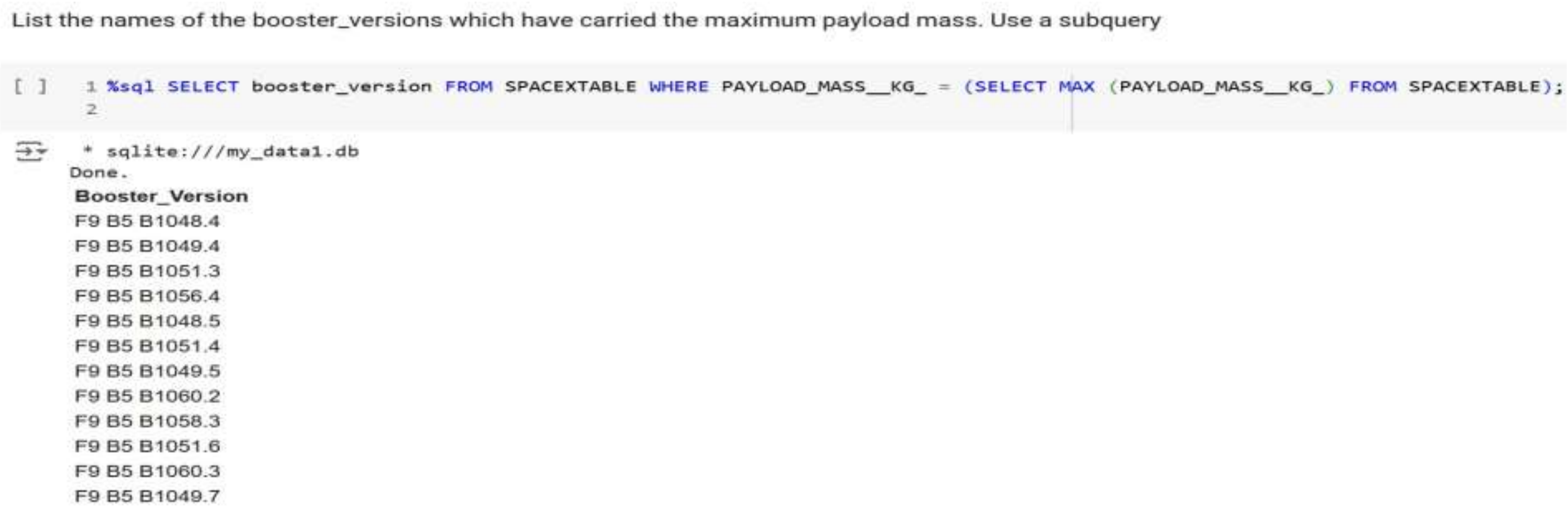


```
%sql SELECT mission_outcome, COUNT(*) AS total_count FROM SPACEXTABLE WHERE  mission_outcome IN ('Success', 'Failure (in flight)') GROUP BY mis
```

```
* sqlite:///my_data1.db
Done.
Mission_Outcome total_count
Failure (in flight)    1
Success                98
```

Boosters Carried Maximum Payload

- **List of the boosters which have carried the maximum payload mass are shown below.**
- **The query used in obtaining the booster names is shown below.**



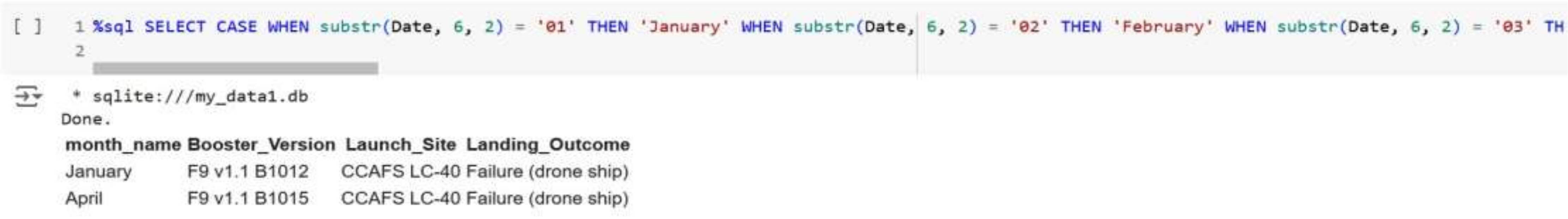
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql SELECT booster_version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX (PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

```
 * 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

- **List of the failed ”landing_outcomes” in drone ship, their booster version, and the launch site name during year 2015 is shown below.**
- **The query used in obtaining the information is shown below.**



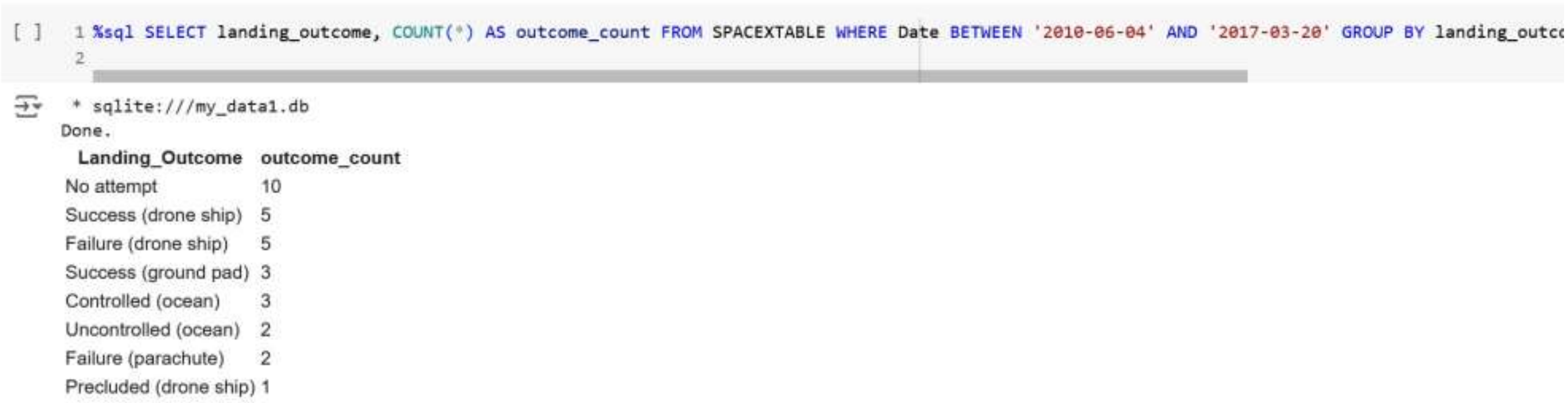
```
%sql SELECT CASE WHEN substr(Date, 6, 2) = '01' THEN 'January' WHEN substr(Date, 6, 2) = '02' THEN 'February' WHEN substr(Date, 6, 2) = '03' TH
```

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

month_name	Booster_Version	Launch_Site	Landing_Outcome
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- **A rank of the count of landing outcomes (such as Failure (drone ship) or success (ground pad)) between the dates 2010-06-04 and 2017-03-20, in descending order is shown below.**
- **The query used to obtain the results is shown below.**



```
%sql SELECT landing_outcome, COUNT(*) AS outcome_count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing_outc
```

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

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

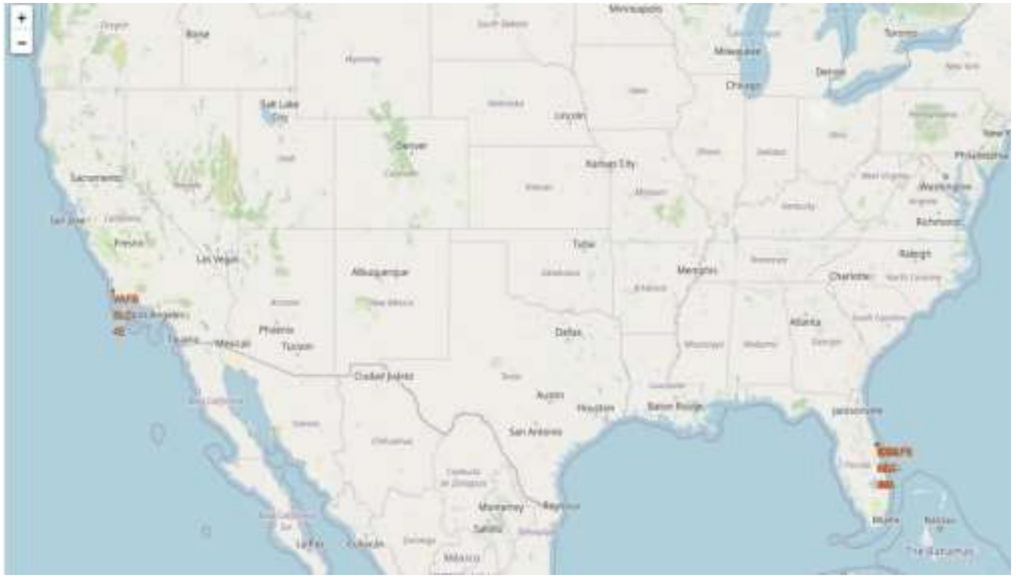
Section 3

Launch Sites Proximities Analysis

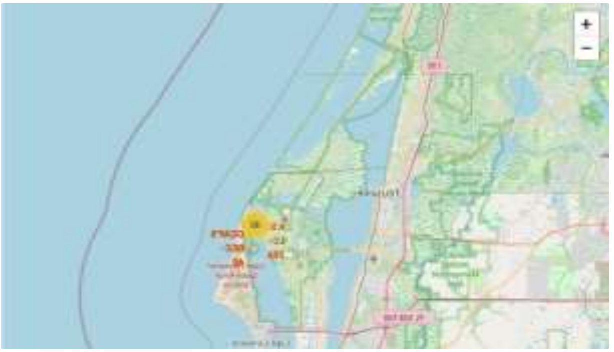
USA Launch Sites in California and Florida

- Most of Launch sites considered in this project are in proximity to the Equator line. Launch sites are made at the closest point possible to Equator line, because anything on the surface of the Earth at the equator is already moving at the maximum speed (1670 kilometers per hour). For example launching from the equator makes the spacecraft move almost 500 km/hour faster once it is launched compared half way to north pole.

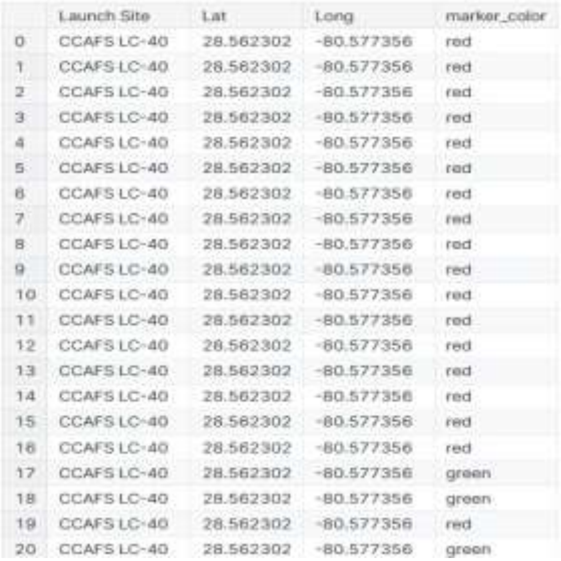
- All launch sites considered in this project are in very close proximity to the coast While starting rockets towards the ocean we minimize the risk of having any debris dropping or exploding near people.



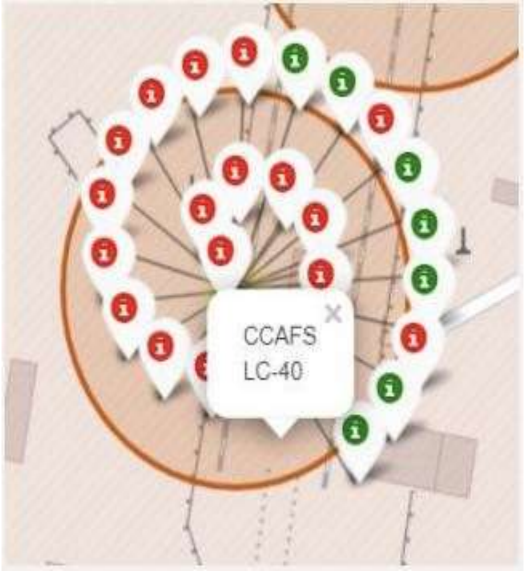
San Luis Obispo
Bakersfield
Santa Maria
10
FB
SLC-
4E
Lancaster
Victorville
Santa Barbara
Palmdale
Santa Clarita
Oxnard
Burbank
Fontana
Los Angeles
Riverside
Long Beach
San Clemente
Oceanside

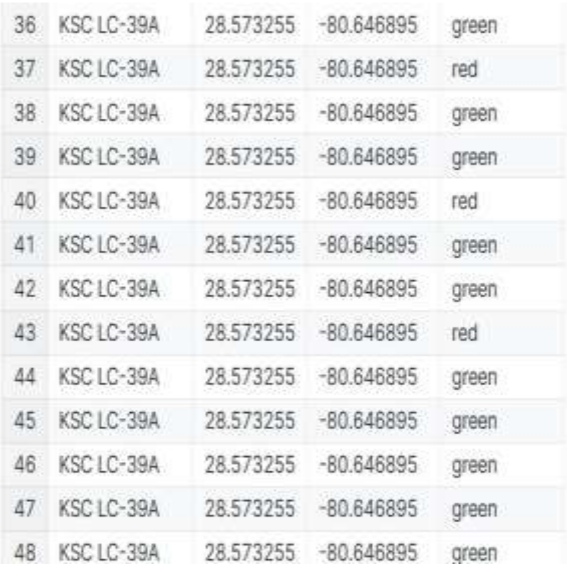


Color Labels Showing the Launch Sites on a Map

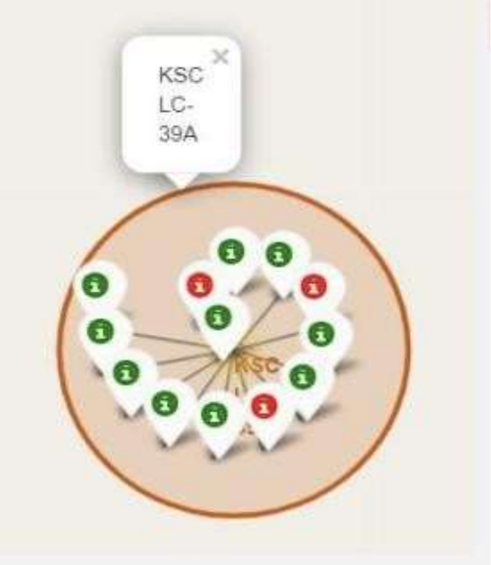


	Launch Site	Lat	Long	marker_color
0	CCAFS LC-40	28.562302	-80.577356	red
1	CCAFS LC-40	28.562302	-80.577356	red
2	CCAFS LC-40	28.562302	-80.577356	red
3	CCAFS LC-40	28.562302	-80.577356	red
4	CCAFS LC-40	28.562302	-80.577356	red
5	CCAFS LC-40	28.562302	-80.577356	red
6	CCAFS LC-40	28.562302	-80.577356	red
7	CCAFS LC-40	28.562302	-80.577356	red
8	CCAFS LC-40	28.562302	-80.577356	red
9	CCAFS LC-40	28.562302	-80.577356	red
10	CCAFS LC-40	28.562302	-80.577356	red
11	CCAFS LC-40	28.562302	-80.577356	red
12	CCAFS LC-40	28.562302	-80.577356	red
13	CCAFS LC-40	28.562302	-80.577356	red
14	CCAFS LC-40	28.562302	-80.577356	red
15	CCAFS LC-40	28.562302	-80.577356	red
16	CCAFS LC-40	28.562302	-80.577356	red
17	CCAFS LC-40	28.562302	-80.577356	green
18	CCAFS LC-40	28.562302	-80.577356	green
19	CCAFS LC-40	28.562302	-80.577356	red
20	CCAFS LC-40	28.562302	-80.577356	green





36	KSC LC-39A	28.573255	-80.646895	green
37	KSC LC-39A	28.573255	-80.646895	red
38	KSC LC-39A	28.573255	-80.646895	green
39	KSC LC-39A	28.573255	-80.646895	green
40	KSC LC-39A	28.573255	-80.646895	red
41	KSC LC-39A	28.573255	-80.646895	green
42	KSC LC-39A	28.573255	-80.646895	green
43	KSC LC-39A	28.573255	-80.646895	red
44	KSC LC-39A	28.573255	-80.646895	green
45	KSC LC-39A	28.573255	-80.646895	green
46	KSC LC-39A	28.573255	-80.646895	green
47	KSC LC-39A	28.573255	-80.646895	green
48	KSC LC-39A	28.573255	-80.646895	green

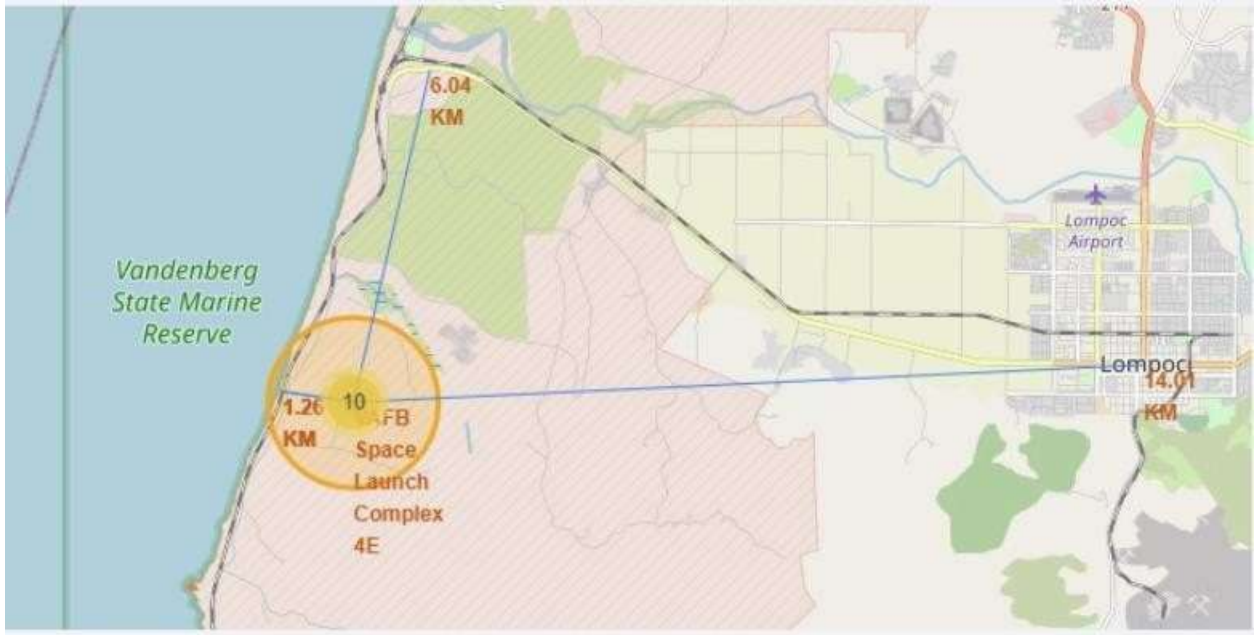


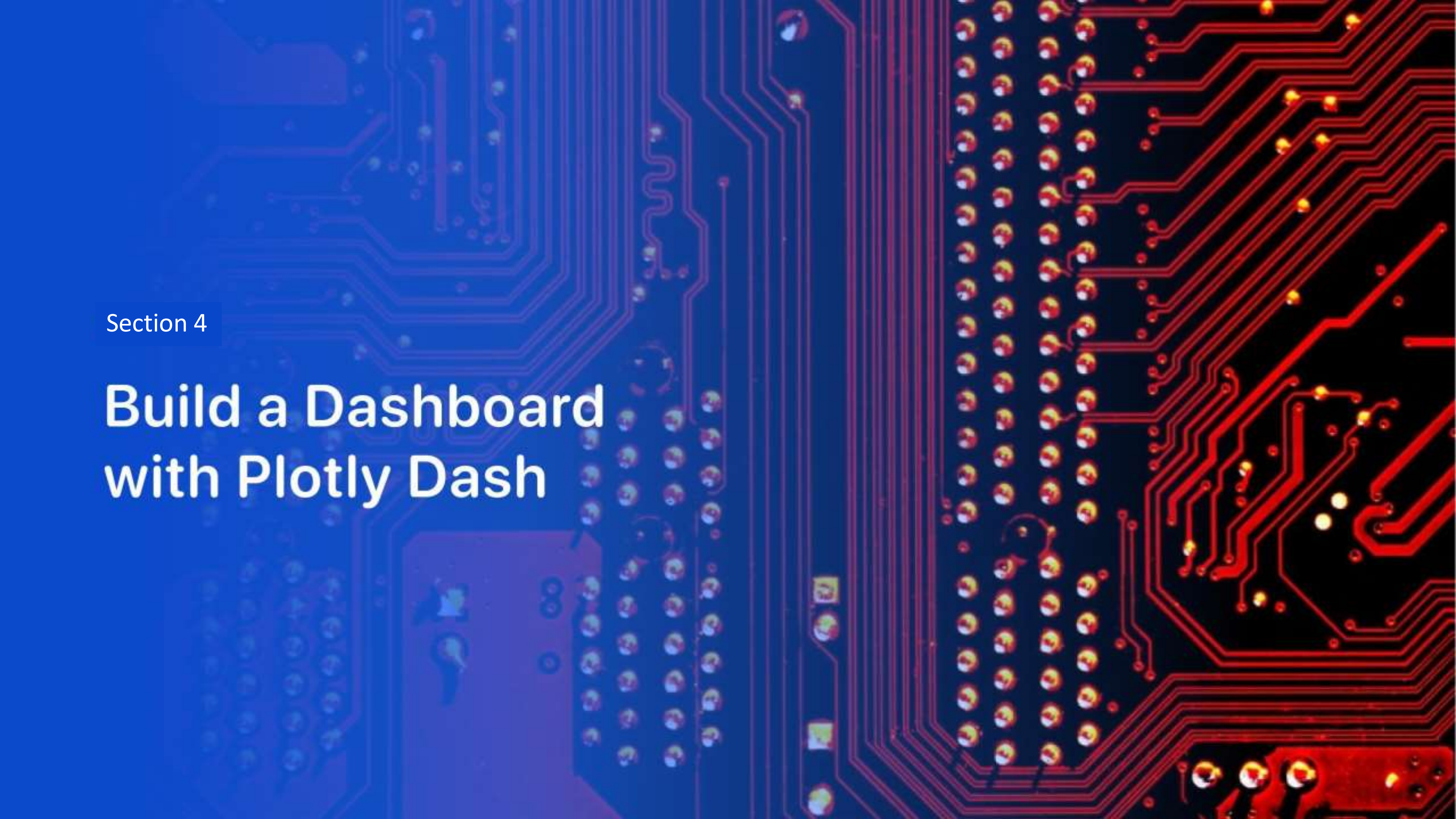
Green= Successful Launch

Red= Failed Launch

Safe Distance to Launch Site

The obtained results indicate that all launch sites are at safe distance from railway lines and cities.

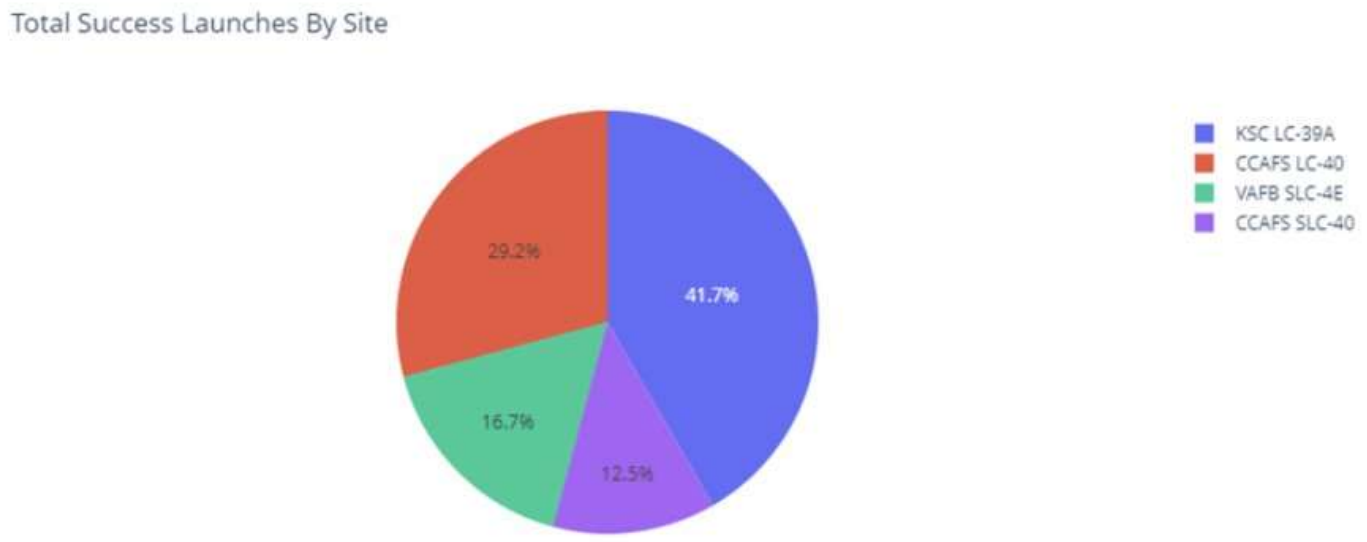




Section 4

Build a Dashboard with Plotly Dash

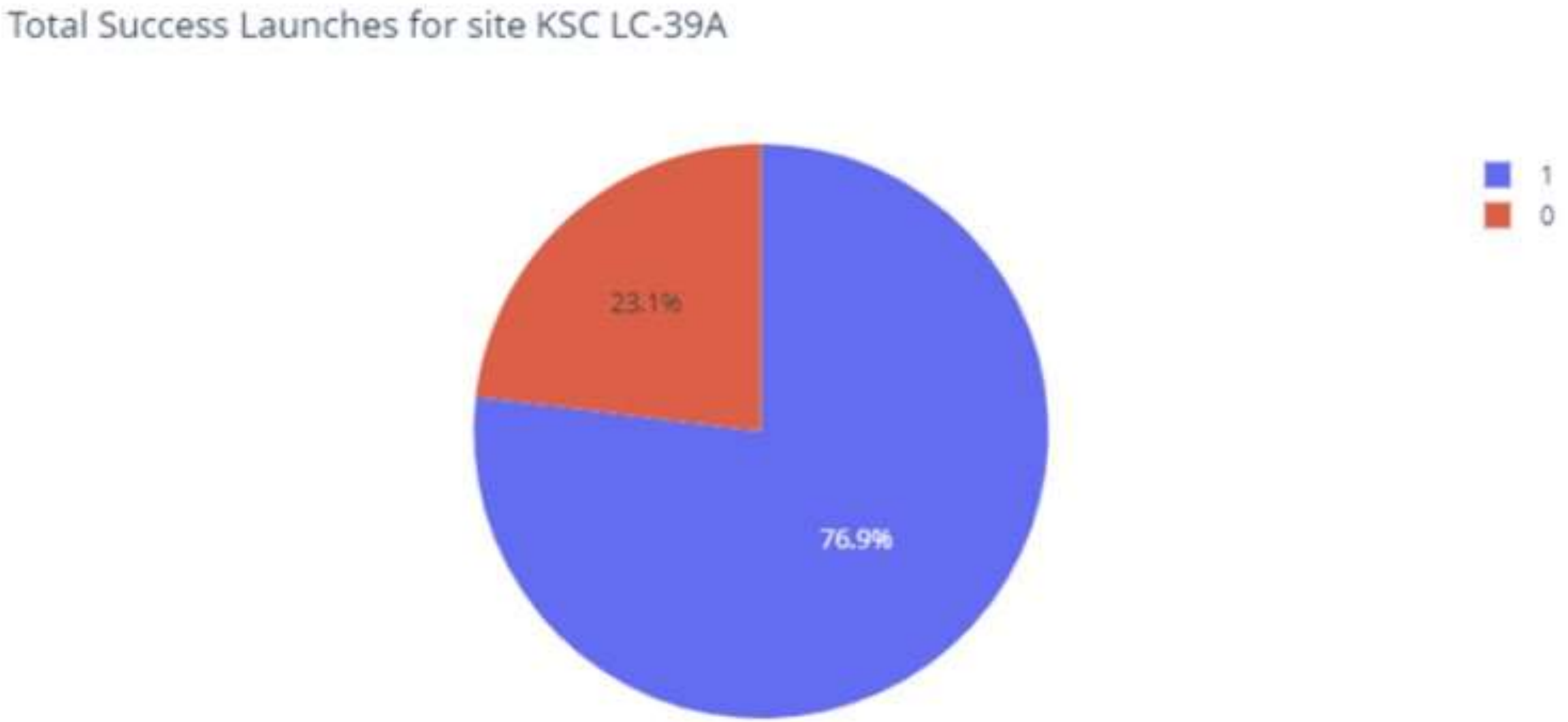
Total Launch Success for All Sites



The highest success launch rates were recorded at these sites :

1. KSC LC-39A (41.7%)
2. CCAFS LC-40 (29.2%)

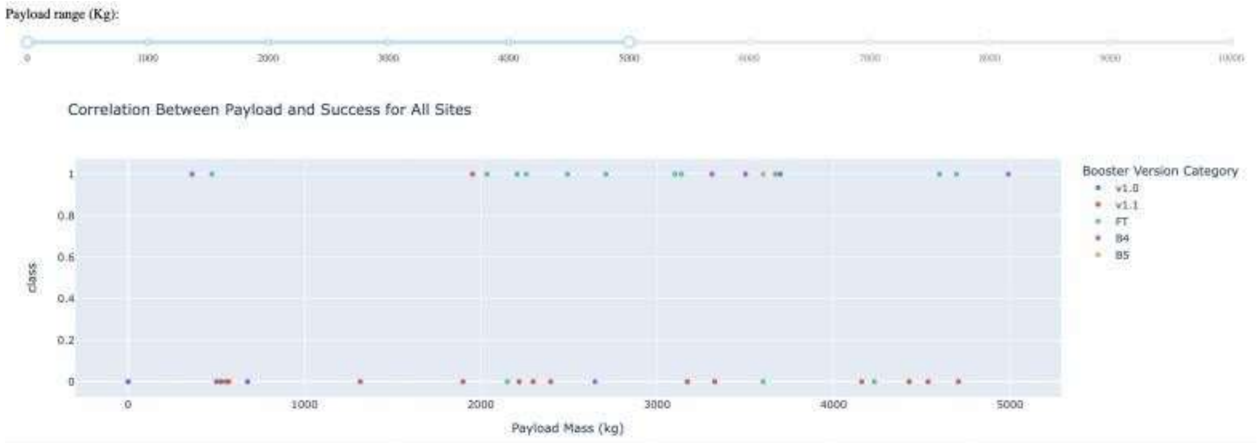
KSC LC-39 Launch Site Success Rate

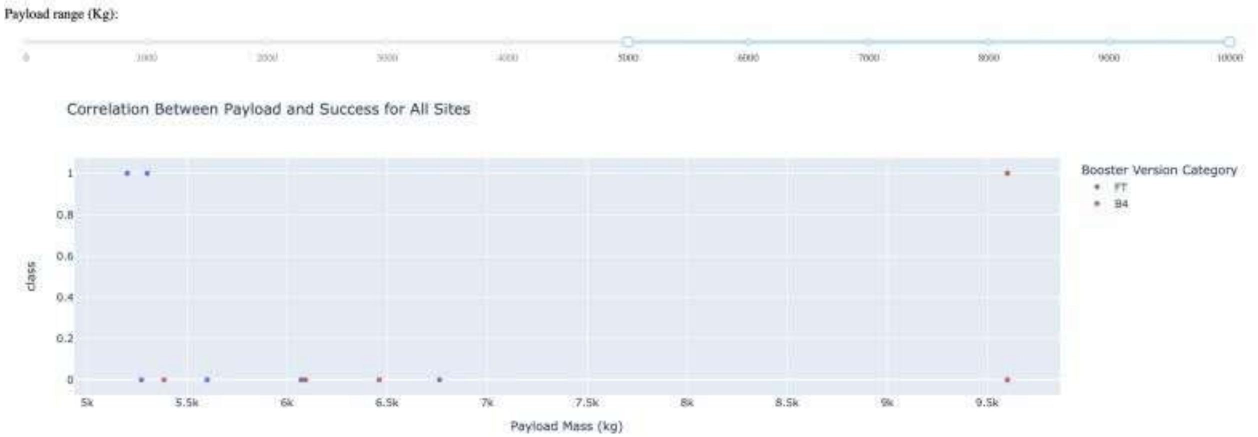


Site KSC LC-39 success rate is 76.9%

Payload vs. Launch Outcome for All Sites

Highest success rate for payloads is between 2000 and 5500 Kgs

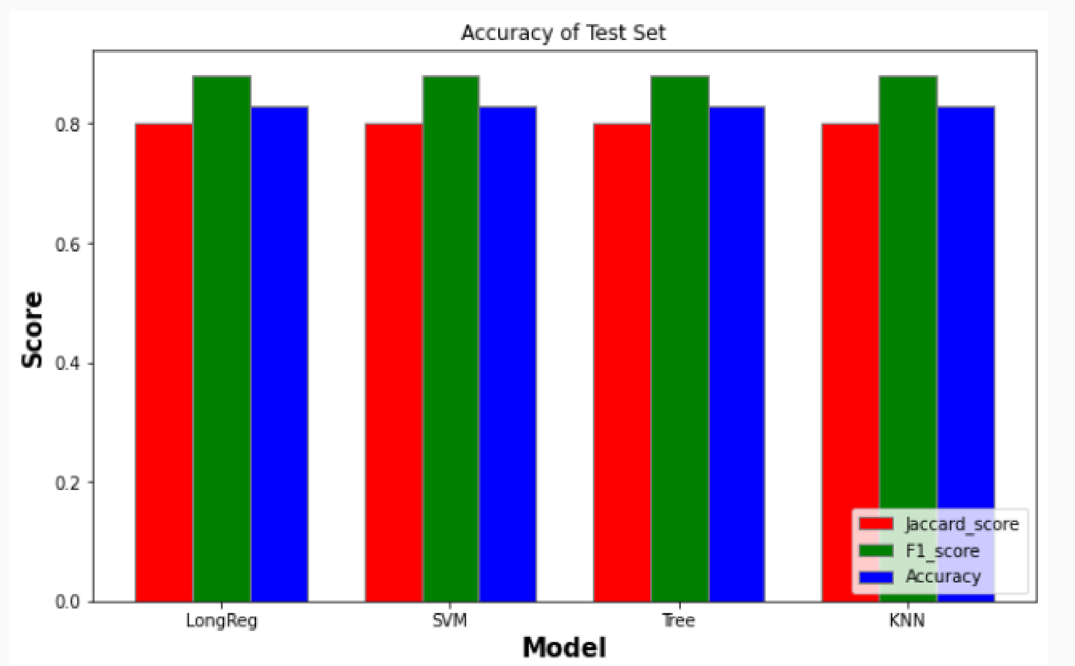


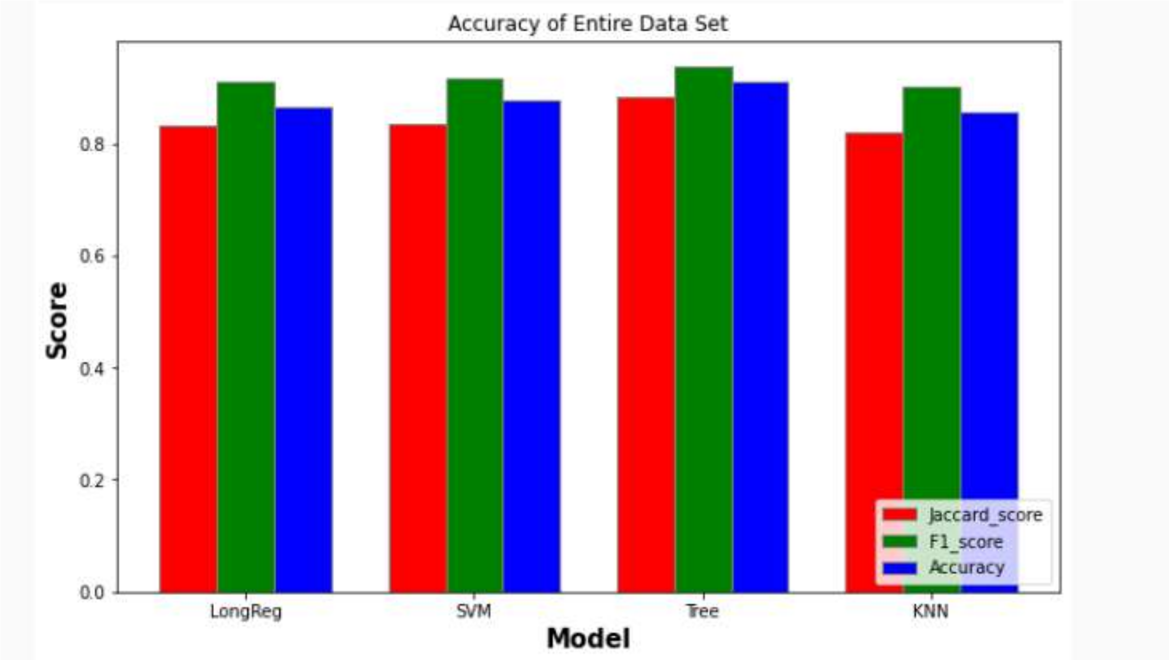


Section 5

Predictive Analysis (Classification)

Classification Accuracy

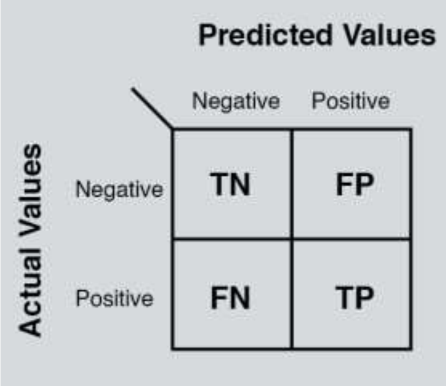




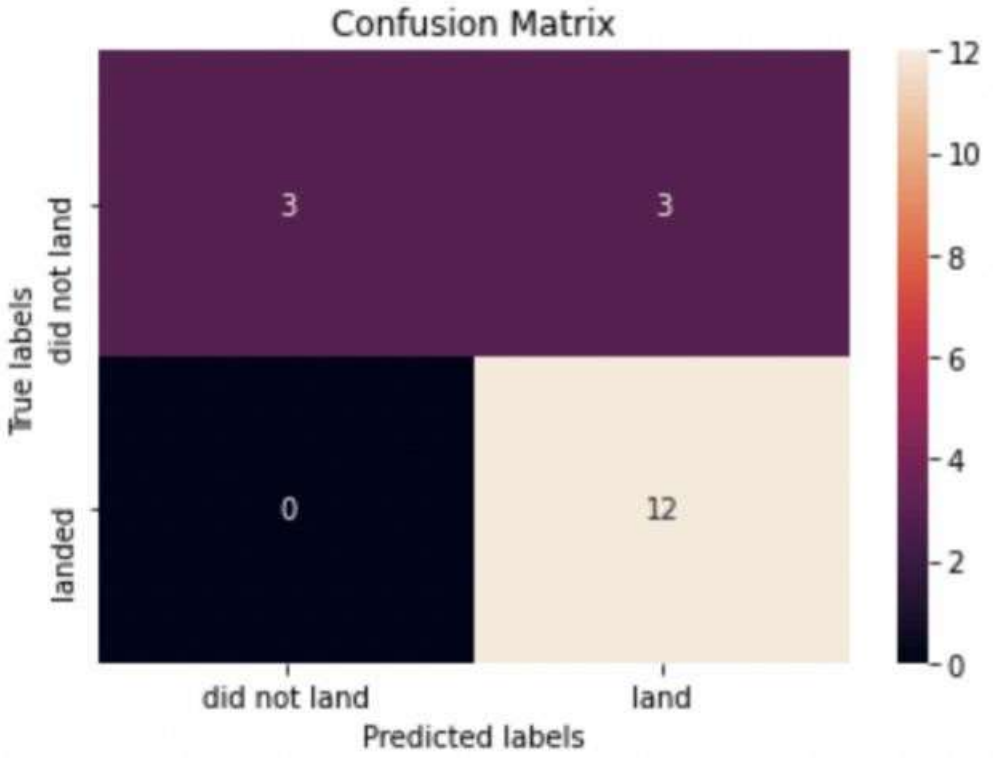
- **Using the test set the same accuracy results were obtained from the four models.**
- **The Tree Model provided the best accuracy results for the entire data set.**

Confusion Matrix

- **The confusion matrix analysis suggests that the best performing model is the Logistic Regression model.**
- **The confusion matrix predicts 13 true positives, 3 false positives, 3 true positive, and 0 false negative.**



		Predicted Values	
		Negative	Positive
Actual Values	Negative	TN	FP
	Positive	FN	TP



Conclusions

- The success rate for the rocket launches increased after 2013.
- Orbits GEO, HEO, ES-L1 and SSO have 100%

  launch success rate.
- Launch site KSC LC-39A has the highest success rate.
- The Decision Tree model is the best ML algorithm for analyzing the SpaceX data set and provided the best accuracy results.

Appendix

Thank you!