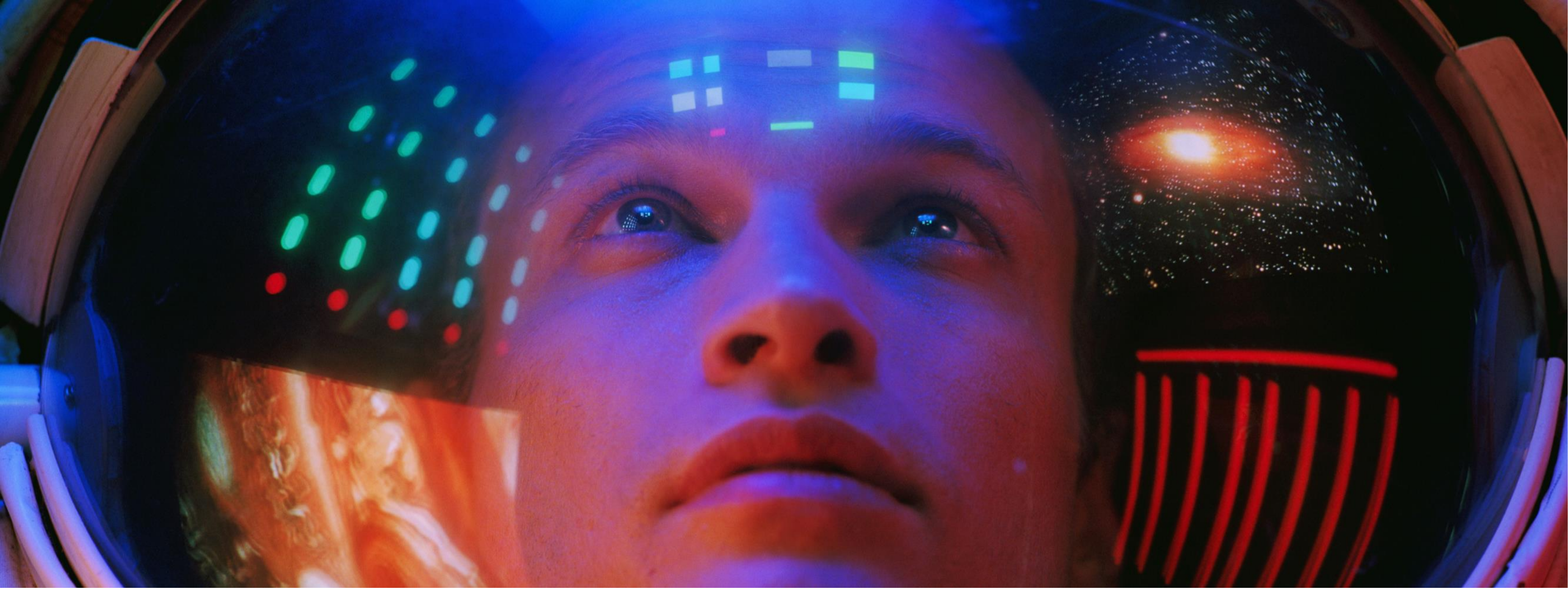
WINNING THE SPACE RACE WITH DATA SCIENCE

Gareth Sims
December 2024

CONTENTS



1. EXECUTIVE SUMMARY

EXECUTIVE SUMMARY

The purpose of this project was to predict, using various data science techniques, the likelihood of SpaceX's Falcon 9 rockets landing successfully.

Publicly available data was collected, prepared and analysed before evaluating the accuracy of four machine learning models to predict the outcome of launches.

The main conclusions were:

1. All models perform equally with a test accuracy of 83.33%
2. As the Payload increases beyond 5,000 kg the success rate falls dramatically
3. The launch site with the highest success rate of 76.9% is KSC LC-39A
4. Despite a fall in 2018, the average success rate for landings has increased every year from 2013 to 2019
5. Orbit types ES-L1, GEO, HEO & SSO all had a mean success rates of 100%.

2. INTRODUCTION

INTRODUCTION

I work as a data scientist for SpaceY, a rocket company which aims to compete against SpaceX.

SpaceX (Space Exploration Technologies Corp.) is a private aerospace company founded by Elon Musk in 2002. Its goal is to reduce the cost of space travel and make human life multiplanetary, particularly by enabling the colonization of Mars. SpaceX has developed several rockets, including the Falcon 1, Falcon 9, and Falcon Heavy, with the Starship currently under development for deep space missions.

A key differentiator for Space X is that it claims to be cheaper than its rivals. For example, on its website, Space X advertises that the cost of launching its Falcon 9 is $62 million. This price is significantly lower than competitors whose costs can be in excess of $165 million. One of the main reasons SpaceX is able to keep costs low is that can reuse the first stage of its rockets by landing them successfully.

Therefore, by determining whether the first stage will land, it is possible to estimate the cost of a launch. This critical information can then be used by SpaceY to decide whether we should bid against SpaceX for a rocket launch.

The aim of this project is to use publicly available data and machine learning techniques to predict the landing success of the first stage of Falcon 9 rockets.

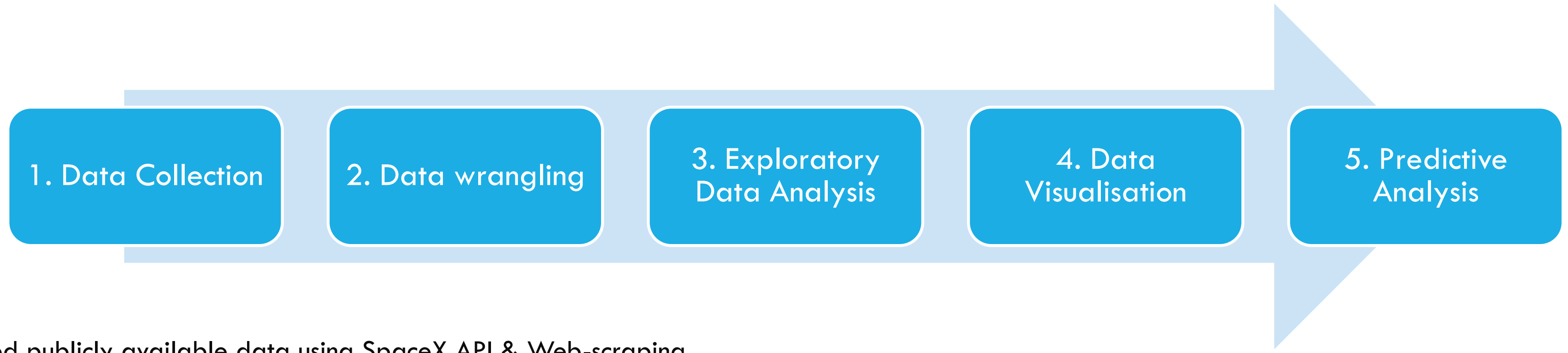
3. METHODOLOGY

METHODOLOGY

The methodology used in this project is outlined in the flow chart below.

It followed a sequential process from data collection through to predictive analytics. I used several different Python libraries, tools and techniques to collect, wrangle, visualize and analyse the data.

Details of each phase of the methodology will be explained in subsequent slides.



1. Collected publicly available data using SpaceX API & Web-scraping
2. Performed data wrangling to clean and transform the data so that it could be analysed
3. Undertook Exploratory Data Analysis using SQL, Pandas and Matplotlib
4. Created an interactive dashboard using Folium & Plotly Dash
5. Performed predictive analysis using a variety of machine learning models

1. DATA COLLECTION

Publicly available data was collected from TWO sources:

1. SpaceX REST API using the following URL: https://api.spacexdata.com/v4/launches/past
2. Wikipedia website: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

The next two slides outline the steps taken to collect the data from these sources

1.1 DATA COLLECTION: SPACE X API

The following tasks were performed to extract and clean the data using the SpaceX API. I have also included some relevant code snippets used in each task. For the complete code please go to Github URL for Data Collection - API

Task 1: To request and parse the SpaceX launch data

- I used a GET request to retrieve the data
- I then converted the json format to a data frame

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

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

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

Task 2: To filter the data frame to only include Falcon 9 launches

- I removed the Falcon 1 launches from the BoosterVersion column

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

Task 3: Remove missing values

- The PayloadMass column had 5 null values which were removed by allocating the mean value for the non-null values in that column

```
payloadmassmean = data_falcon9['PayloadMass'].mean()
```

```
data_falcon9['PayloadMass'].replace(np.nan, payloadmassmean, inplace=True)
```

1.2 DATA COLLECTION: WEB SCRAPING

The following tasks were performed to extract and clean the data from the Wikipedia page. I have also included some relevant code snippets used in each task. For the complete code please go to [Github URL for Data Collection – Web Scraping]()

TASK 1: To request the Falcon9 Launch Wiki page from its URL

- I performed an HTTP GET method to request the relevant HTML page as an HTTP response
- I created a BeautifulSoup object from the HTML response

```
data  = requests.get(static_url).text
```

```
soup = BeautifulSoup(data, 'html.parser')
```

Task 2: To extract all column/variable names from HTML table header

- I found all the tables on the webpage and iterated through the <th> elements to extract column names

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

```
first_launch_table = html_tables[2]
```

Task 3: To create a data frame by parsing the launch HTML tables

- I created an empty dictionary with keys from the extracted column names and then filled up the dictionary with launch records
- I finally parsed the launch records into *launch_dict* and created a dataframe

```
launch_dict= dict.fromkeys(column_names)
```

```
payload_mass = get_mass(row[4])
launch_dict['Payload mass'].append(payload_mass)
print(payload)
```

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

2. DATA WRANGLING

The following tasks were performed to initially explore the data and to determine what labels would be used for training the machine learning models. I have also included some relevant code snippets used in each task. For the complete code please go to Github URL for Data Wrangling

TASK 1: To calculate the number of launches on each site

- I used the method .value_counts on the column LaunchSite to determine the number of launches at each site

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

TASK 2: To calculate the number and occurrence of each orbit

- I used the method .value_counts() to determine the number and occurrence of each orbit in the column Orbit

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

TASK 3: To calculate the number and occurrence of landing outcomes by orbit type

- I used the method .value_counts() on the column Outcome to determine the number of landing_outcomes.
- I created a set of outcomes (bad_outcomes) where the second stage did not land successfully

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

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

TASK 4: To create a landing outcome label from Outcome column

- Using the Outcome column, I created a list where the element is zero if the corresponding row in Outcome is in the set bad_outcome; otherwise, it's one. I then assigned it to the variable landing_class:

```
landing_class = [0 if x in bad_outcomes else 1 for x in df['Outcome']]
```

3. EXPLORATORY DATA ANALYSIS (EDA)

This section is divided into two parts:

1. EDA with Visualisation
2. EDA using SQL

3.1 EDA WITH DATA VISUALISATION

I used a variety of charts from the **Matplotlib** and **Seaborn** libraries to help identify any patterns and trends within the data. The charts and associated code can be found at GitHub URL for EDA

To help visualize relationships between variables I used a variety or different charts: .

A **Scatter Point Chart**, which is good at showing the relationship between two numeric variables, was used to explore the relationship between:

1. Launch site and Flight Number
2. Payload Mass and Launch Site
3. Flight Number and Orbit Type
4. Payload Mass and Orbit Type

A **Bar Chart**, which is useful to compare different values across subgroups, was used to show how the Success Rate of each Orbit Type compared against one another

A **Line Chart**, which is good at showing changes over time, was used to visualize the yearly trend for Launch Successes.

3.2 EDA WITH SQL

SQL was used to perform the following queries:

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List the date when the first successful landing outcome in ground pad was achieved.
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster versions which have carried the maximum payload mass. Use a subquery
9. List the records which will display the month names, failure landing outcomes in drone ship, booster versions, launch site for the months in year 2015.
10. 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.

The SQL code can be found at GitHub URL for EDA with SQL

4. DATA VISUALISATION

This section is divided into two parts:

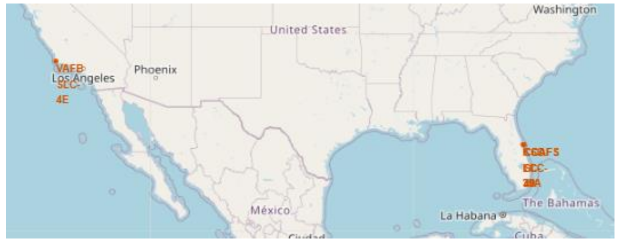
1. Building an interactive map using Folium
2. Building a dashboard with Plotly Dash

4.1 AN INTERACTIVE MAP USING FOLIUM

Folium is a Python library that helps create several types of maps. It was used to create an interactive map of the launch sites used by SpaceX. The relevant code can be found at [GitHub URL for Interactive Map with Folium]

There were 3 main tasks performed by Folium:

1. Using Markers, all Launch sites were made easily visible on the map



2. Using a MarketCluster object, the launch successes at each site were clearly shown. Successful launches were marked in green whereas failures were marked in red

3. Finally, a PolyLine was used to plot the distances between a launch site and its proximity to roads, railways and coastlines

4.2 DASHBOARD USING PLOTLY DASH

Plotly Dash was used to create an interactive application to enable users to easily visualize SpaceX launch data in real-time. The relevant code to create the dashboard can be found at GitHub URL for Dash App

The dashboard contains a number of features including:

1. A Drop-down component to allow users to select the launch site they want to analyse
2. A callback function to render a Pie-chart based on the selected site dropdown choice
3. A Range Slider to select a range of different Payloads
4. A Callback function to render a Scatter Plot showing the relationship between launch outcome and the selected Payload range

5. PREDICTIVE ANALYSIS (CLASSIFICATION)

A machine learning pipeline was developed to predict if the first stage will land. Details of the code can be found at [GitHub URL for Machine Learning prediction]

Scikit was used to access a variety of ML classification models.

The main steps in the process were:

1. Loading the data into a data-frame and standardising it
2. Splitting the data into **training** and **testing** data using the function *train_test_split*
3. The following classification models were evaluated: **Logistic Regression, Support Vector Machine (SVM), Decision Tree,** and **K Nearest Neighbours** (KNN)
4. To fit the models to the training data, *GridSearchCV* is used to find the best parameters
5. Each trained model is then tested using the testing data set
6. An *accuracy score* is then calculated

4. RESULTS

RESULTS

This section outlines the key results from the exploratory data analysis and is divided into four main areas:

1. Insights from the EDA
2. Proximity analysis of the launch sites
3. Interactive dashboard using Plotly Dash
4. Predictive Analysis

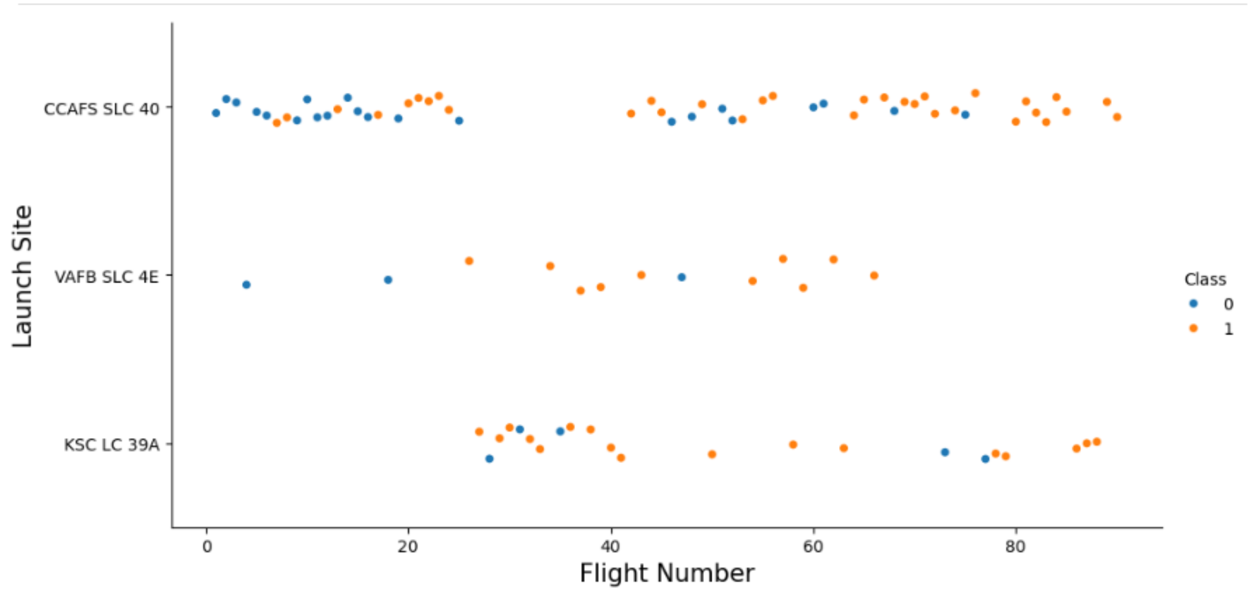
1. INSIGHTS FROM EDA

The following slides provide details of the results from the EDA. Charts, tables and code snippets have been presented to help display the findings.

FLIGHT NUMBER VS. LAUNCH SITE

The scatter chart shows the relationship between flight number and launch site. Some key observations include:

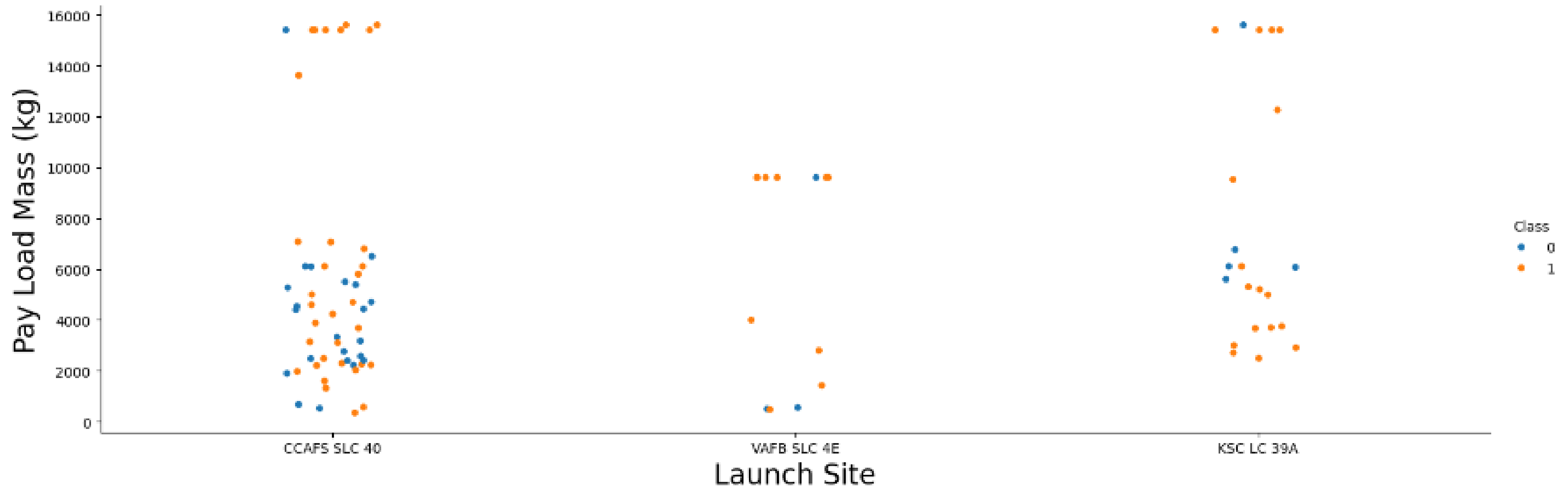
1. All flights after 80 were successful (Class 1)
2. The first 20 flights suffered a high proportion of failures (15/20 or 75%)
3. VAFB launch site was discontinued after launch 70



PAYLOAD VS. LAUNCH SITE

The scatter chart shows the relationship between payload and launch site. Some key observations include:

1. At VAFB-SLC, there were no flights where the payload was above 10,000kg
2. Landings at KSC-LC where Payloads were below 5000kg, were all successful

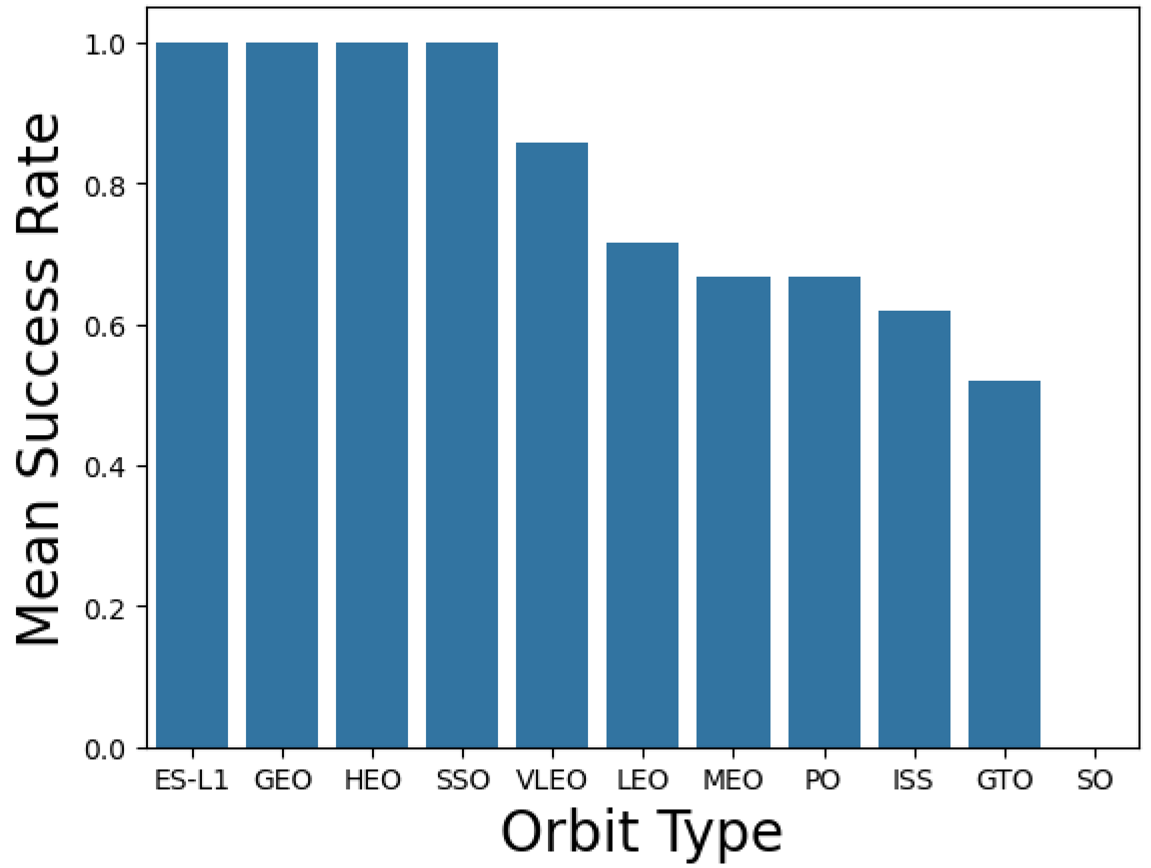


SUCCESS RATE VS. ORBIT TYPE

The bar chart opposite shows the relative success rate by Orbit type.

It's clear to see that the most successful Orbit types are ES-L1, GEO, HEO & SSO.

They all had a mean success rates of 100%. At the opposite end of the scale, Orbit type SO had a 0% success rate

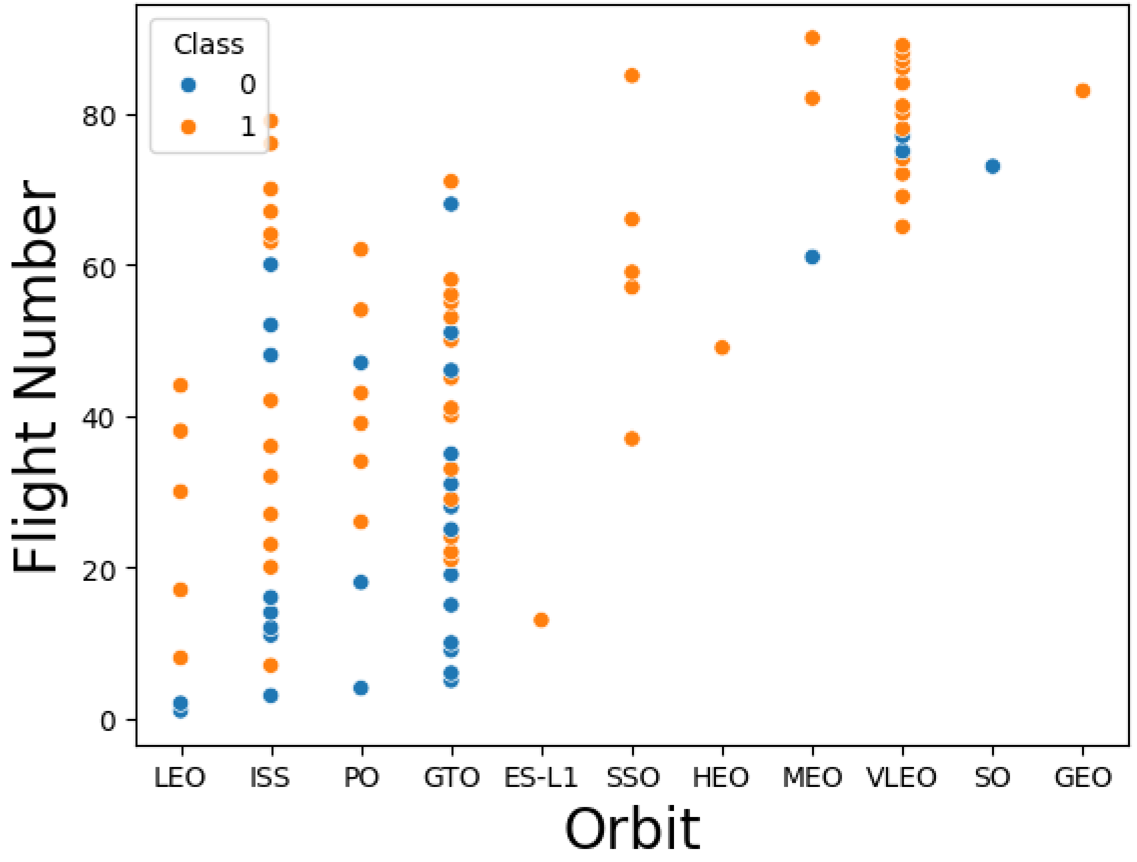


FLIGHT NUMBER VS. ORBIT TYPE

The scatter plot of flight number vs orbit type shows some interesting patterns.

At LEO, success seems to be related to higher flight numbers, whereas at GTO, there is no clear relationship.

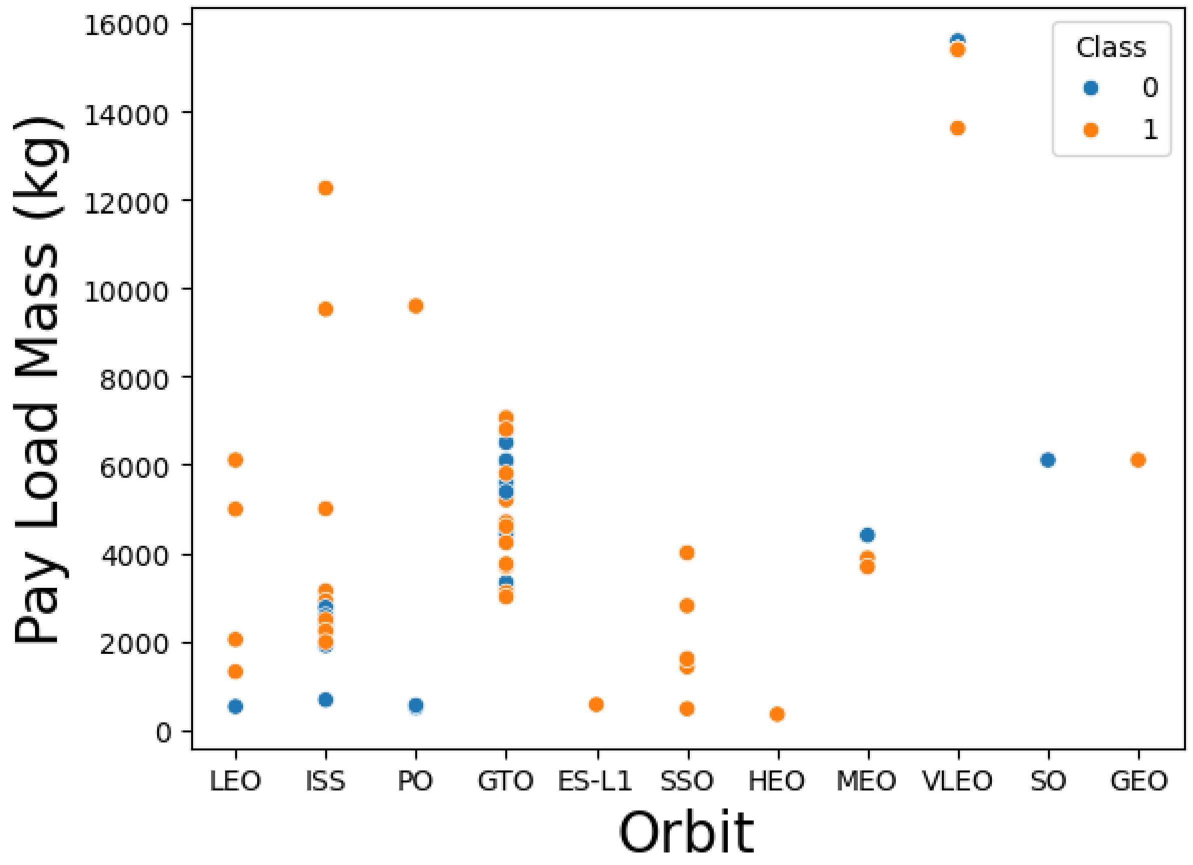
VLEO became the most popular orbit for later flight numbers (>60)



PAYLOAD VS. ORBIT TYPE

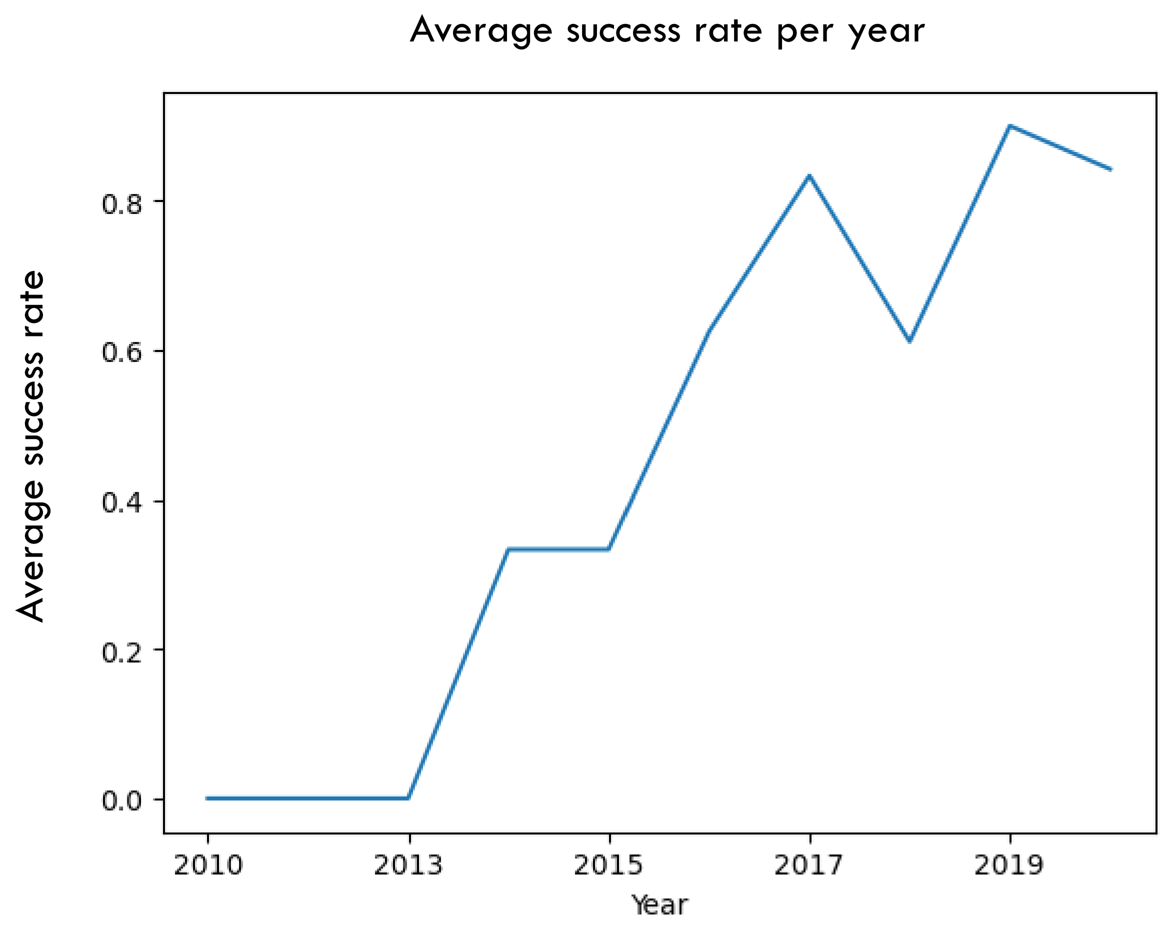
The chart illustrates that for LEO & ISS, heavier payloads are more successful

Again, it is difficult to identify any clear pattern for successful landings at GTO



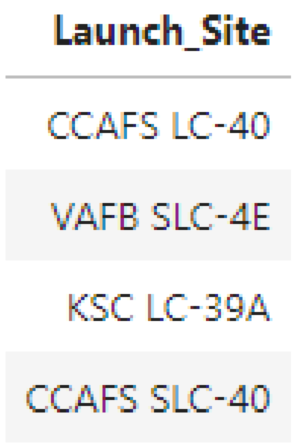
LAUNCH SUCCESS YEARLY TREND

Despite a fall in 2018, the average success rate for landings has increased every year from 2013 to 2019



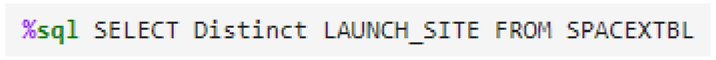
ALL LAUNCH SITE NAMES

The following names are unique launch sites:



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

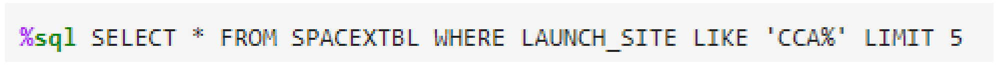
The following SQL query was used to extract this information from the SPACEXTBL table:



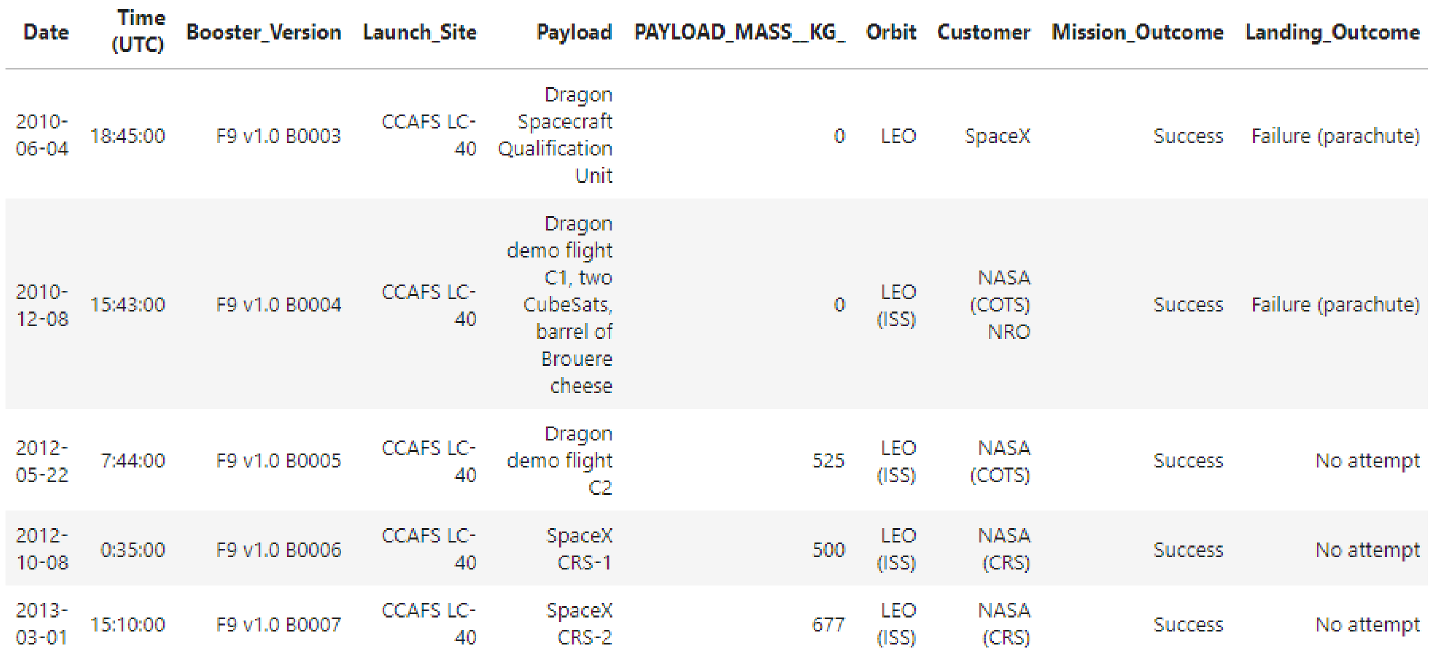
```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL
```

LAUNCH SITE NAMES BEGIN WITH 'CCA'

Using the following SQL query, I was able to extract 5 records where launch site names began with ‘CCA’



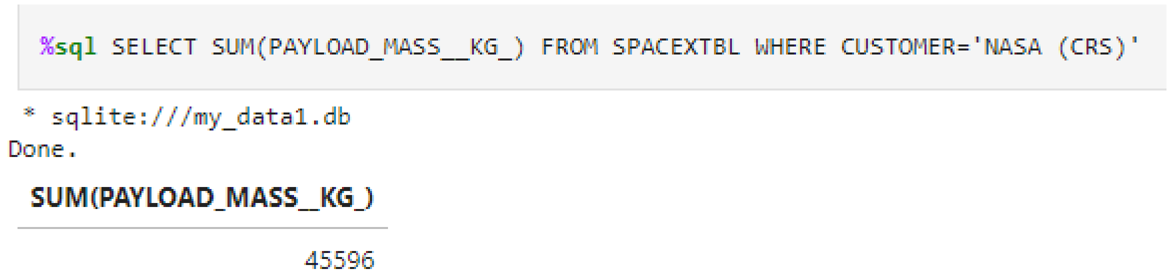
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```



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

Using the following SQL query, I was able to calculate the total payload mass carried by boosters launched by NASA (CRS) was 45,596 kg:



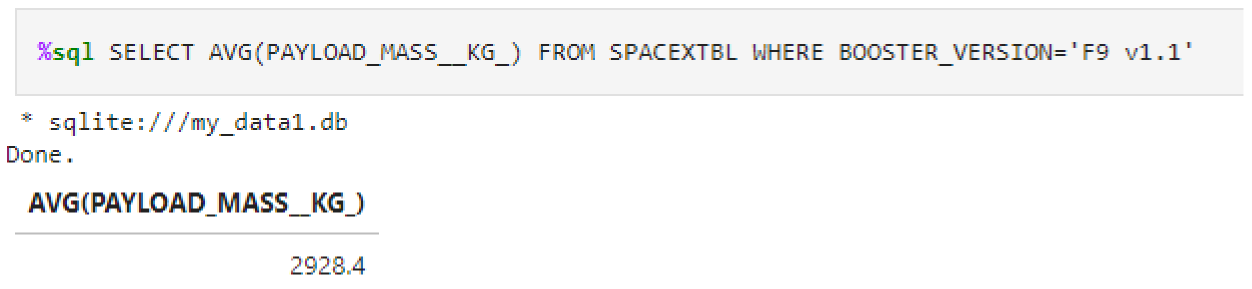
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

AVERAGE PAYLOAD MASS BY F9 V1.1

Using the following SQL query, I was able to calculate the average (mean) payload mass carried by the booster version F9 v1.1 was 2,928 kg:



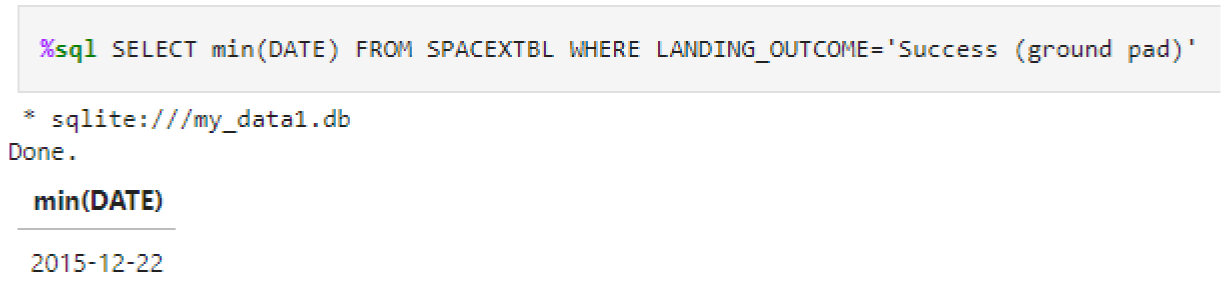
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

FIRST SUCCESSFUL GROUND LANDING DATE

Using the following SQL query, I was able to find that first successful landing on ground pad was 22nd December 2015:



```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING_OUTCOME='Success (ground pad)'
```

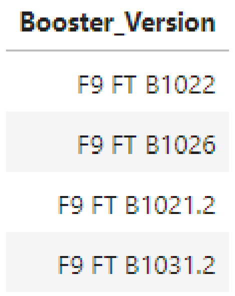
```
 * sqlite:///my_data1.db
Done.
```

min(DATE)
2015-12-22

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

Using the following SQL query, I was able to list the names of the boosters of successful drone ship landings with payloads between 4,000 and 6,000 kg:

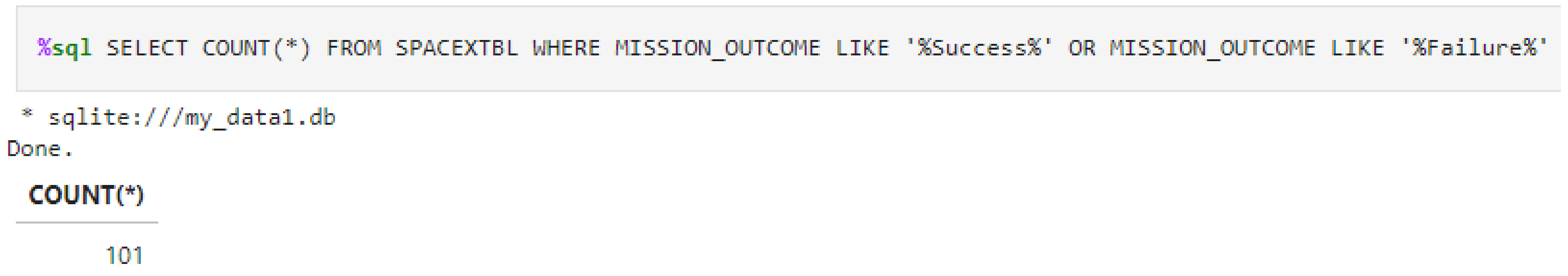
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING_OUTCOME='Success (drone
ship)'
```



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

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

Using the following SQL query, I was able to calculate that the total number (count) of successful and failed mission outcomes was 101:

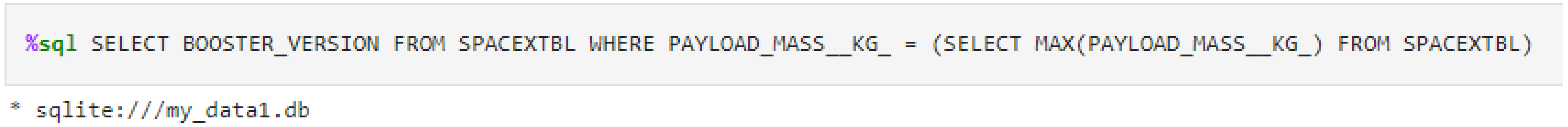


```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
```

```
 * sqlite:///my_data1.db
Done.
COUNT(*)
     101
```

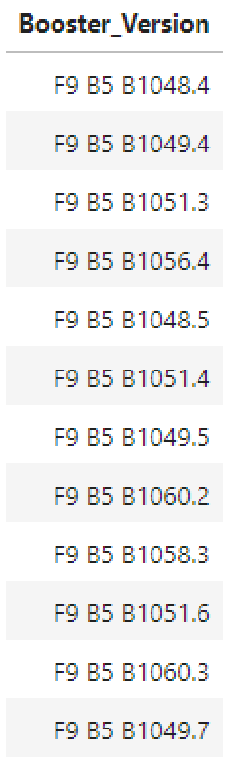
BOOSTERS CARRIED MAXIMUM PAYLOAD

Using the following SQL query, I was able to find the names of the booster versions which carried the maximum payload:



```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

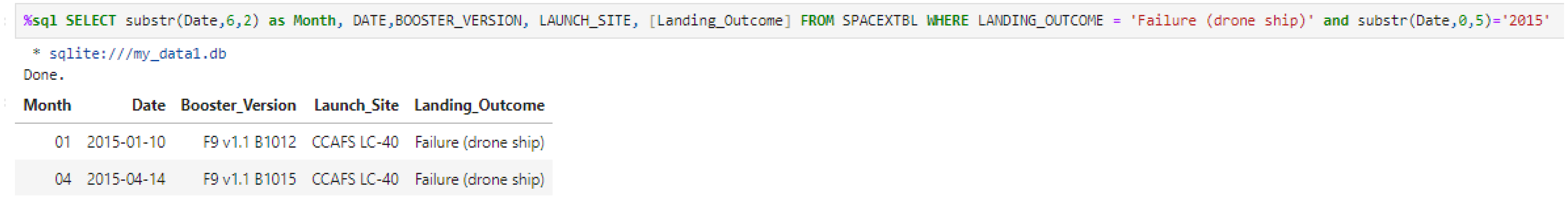
```
 * sqlite:///my_data1.db
```



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

Using the following SQL query, I was able to find the failed launch records for the drone ship and booster versions in 2015:



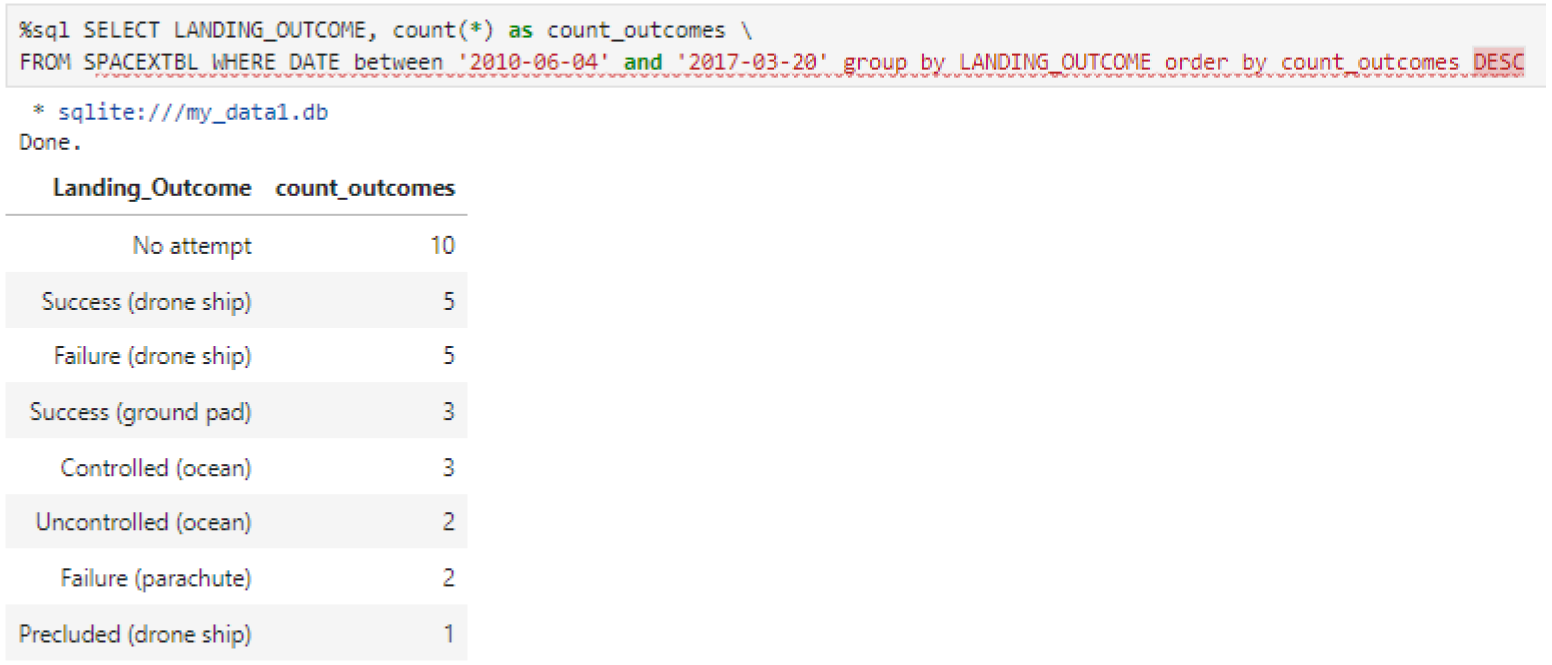
```
%sql SELECT substr(Date,6,2) as Month, DATE,BOOSTER_VERSION, LAUNCH_SITE, [Landing_Outcome] FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Failure (drone ship)' and substr(Date,0,5)='2015'
```

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

Month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

RANK LANDING OUTCOMES BETWEEN 2010-06-04 AND 2017-03-20

Using the following SQL query, I was able to rank, in descending order, the landing outcomes between 4th June 2010 and 20th March 2017:



```
%sql SELECT LANDING_OUTCOME, count(*) as count_outcomes \
FROM SPACEXTBL WHERE DATE between '2010-06-04' and '2017-03-20' group by LANDING_OUTCOME order by count_outcomes DESC
```

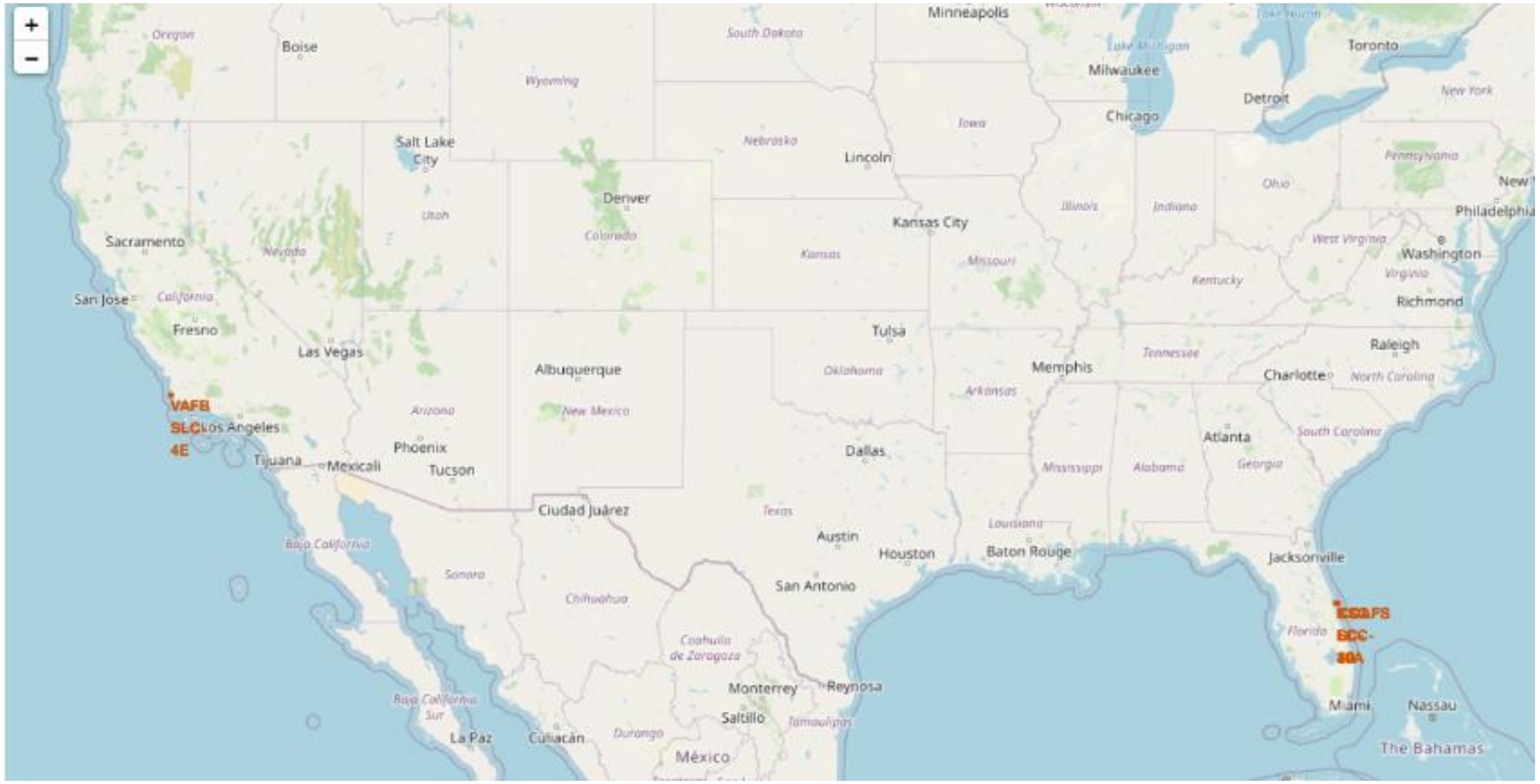
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	count_outcomes
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

2. PROXIMITY ANALYSIS OF THE LAUNCH SITES

The following slides provide details of the results from the proximity analysis using Folium.

GLOBAL MAP SHOWING THE LOCATION OF ALL THE LAUNCH SITES

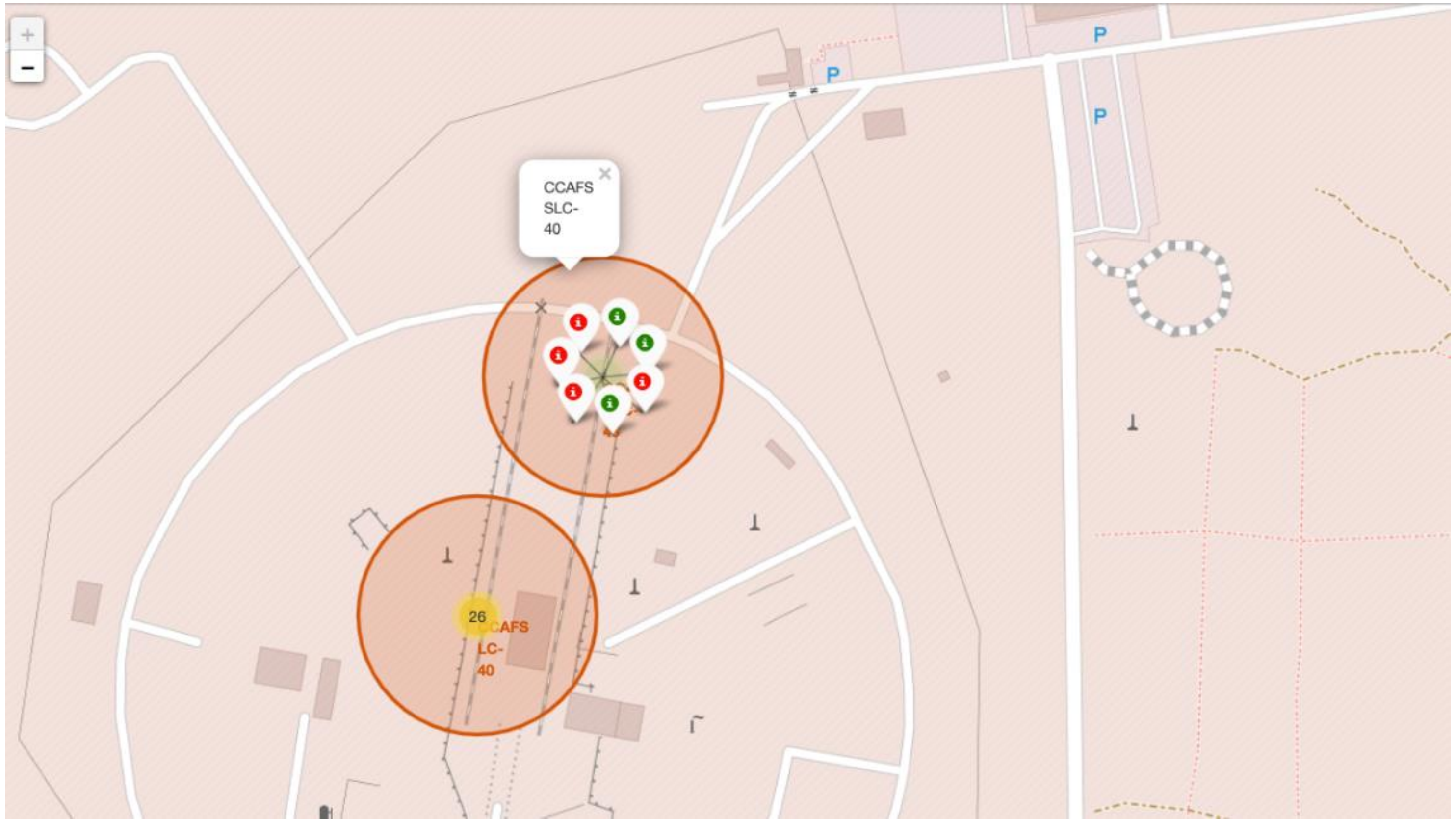


From the map we can see that all the launch sites are located near the ocean and close to the equator.

Being located near the ocean helps reduce the risk of launching near populated areas

Being closer to the equator allows the rocket to take optimum advantage of the Earth's rotational speed, giving the rockets an extra boost and reduce the amount of fuel needed.

USING MARKERS TO DISPLAY LAUNCH RECORDS

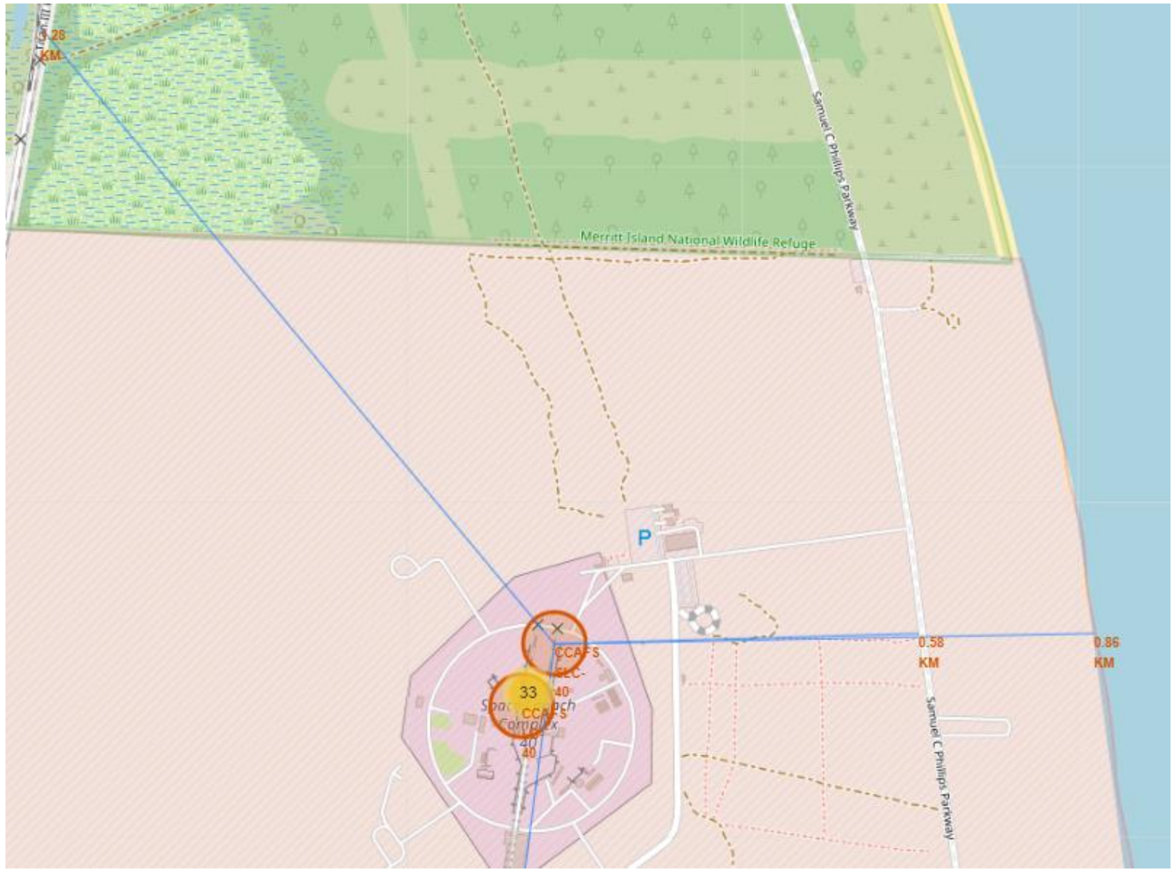


From the screenshot, the launch site CCAFS SLC-40 has been highlighted.

Red markers have been used to depict failed launches and green markers to highlight successful ones.

The image clearly shows that there were 4 failed launches and 3 successful ones at this site

LAUNCH SITE DISTANCES TO ITS PROXIMITIES


KM
Samuel C Phillips Parkway
Merritt Island National Wildlife Refuge
P
0.58 KM
0.86 KM
CCAFS SLC-40
33
CCAFS SLC-40
Samuel C Phillips Parkway

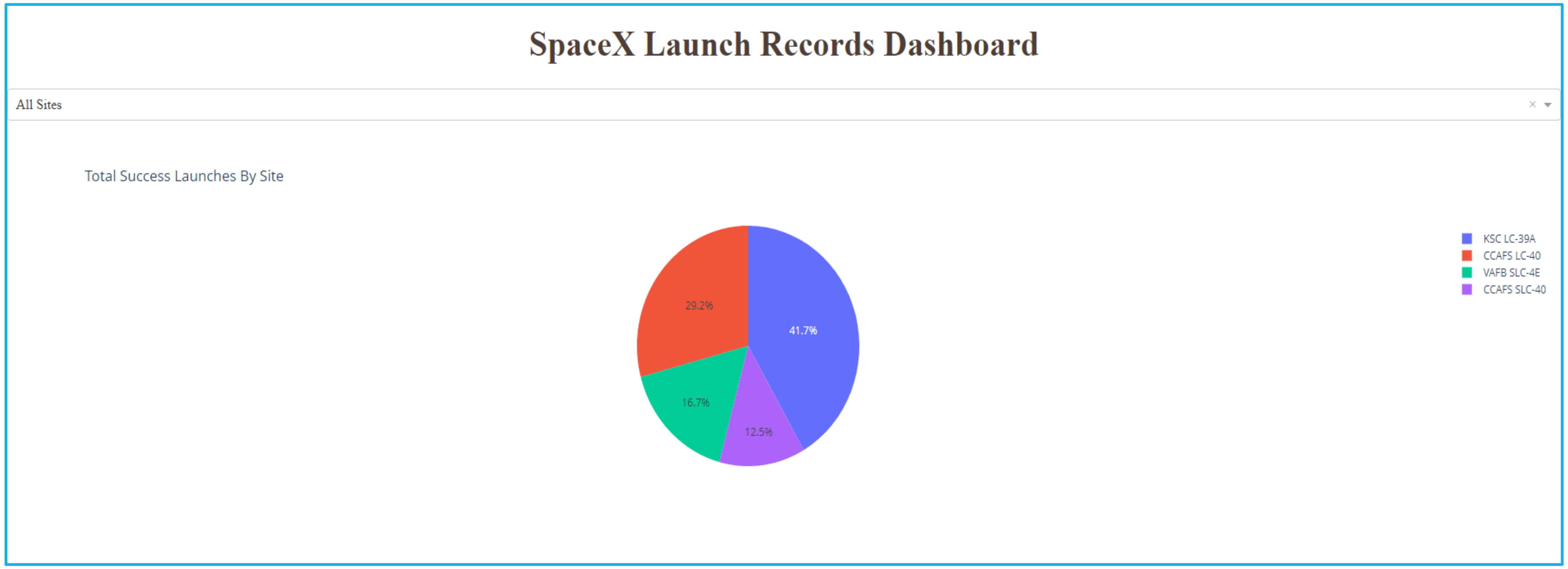
Using blue lines, the distances from sites to nearby geographical features such as coastlines, railways and roads were plotted and calculated.

For example, the distance to the coast from CCAFS SLC-40 is approximately 0.86km

3. INTERACTIVE DASHBOARD USING PLOTLY DASH

The following slides provide screenshots and explanations of the interactive dashboard created in Plotly Dash

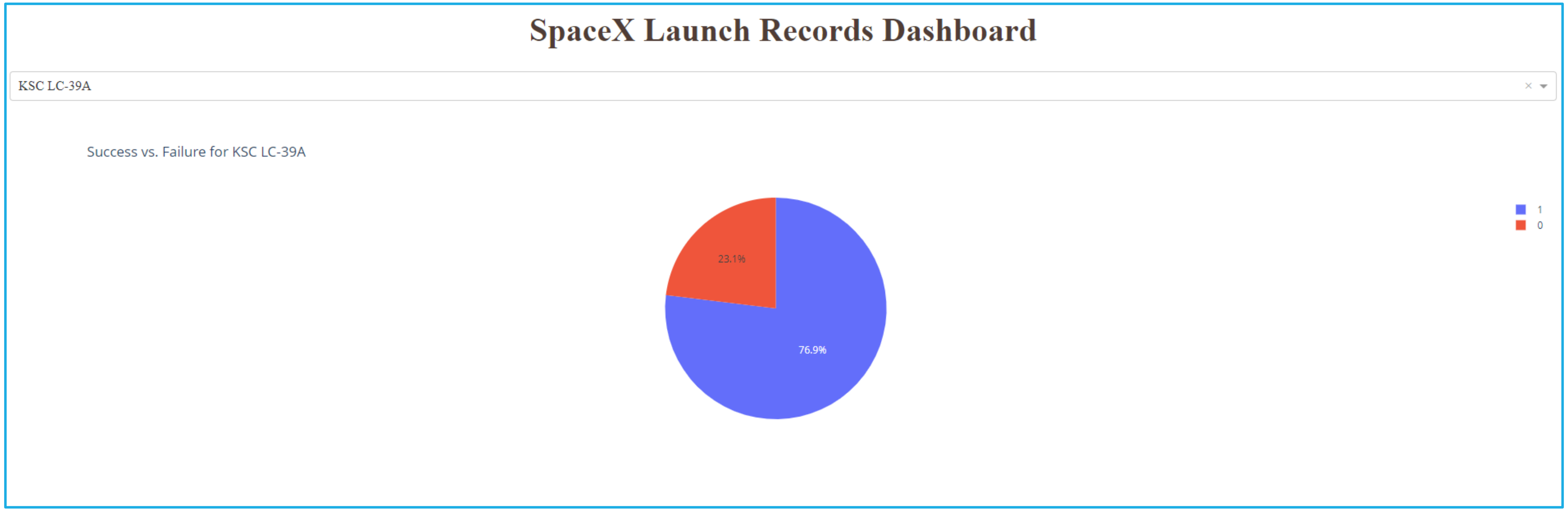
TOTAL SUCCESSFUL LAUNCHES ACROSS ALL SITES



The screenshot above from the Dash application shows the success count for all sites as a percentage of total successful launches

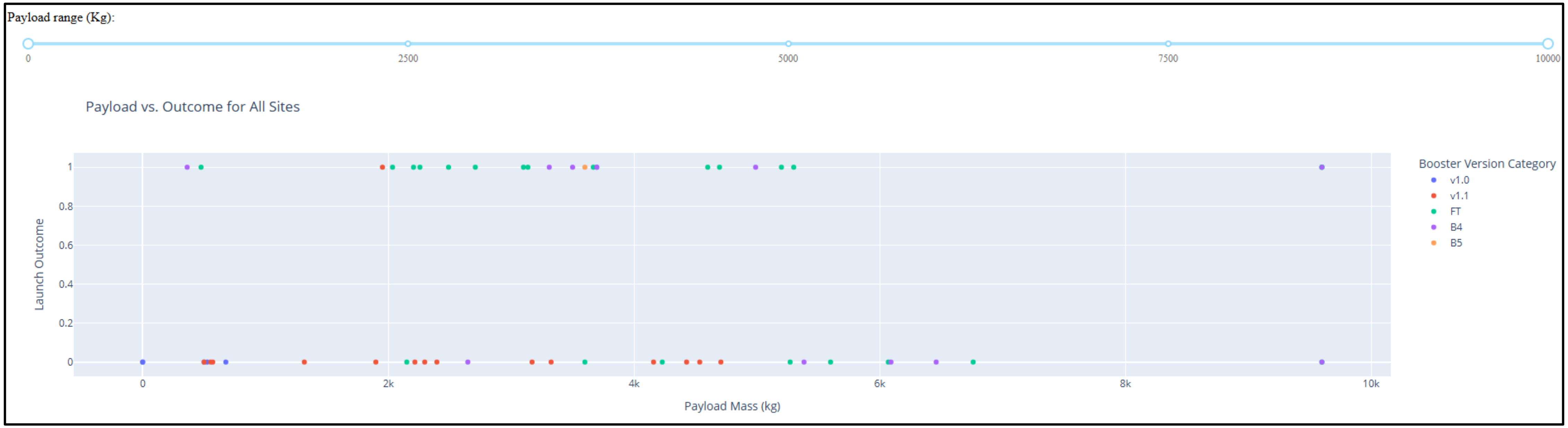
KSC LC 39A contributed the highest percentage, accounting for 41.7% of all successful launches, whereas CCAFS SLC-40 had the lowest success rate at 12.5%

LAUNCH SITE WITH THE HIGHEST SUCCESS RATIO



Drilling down into the launch site with the highest success rate, KSC LC-39A, it is clear from the screenshot above that this site achieved a launch success rate of 76.9% versus a failure rate of 23.1%

PAYLOAD VERSUS LAUNCH OUTCOME FOR ALL SITES



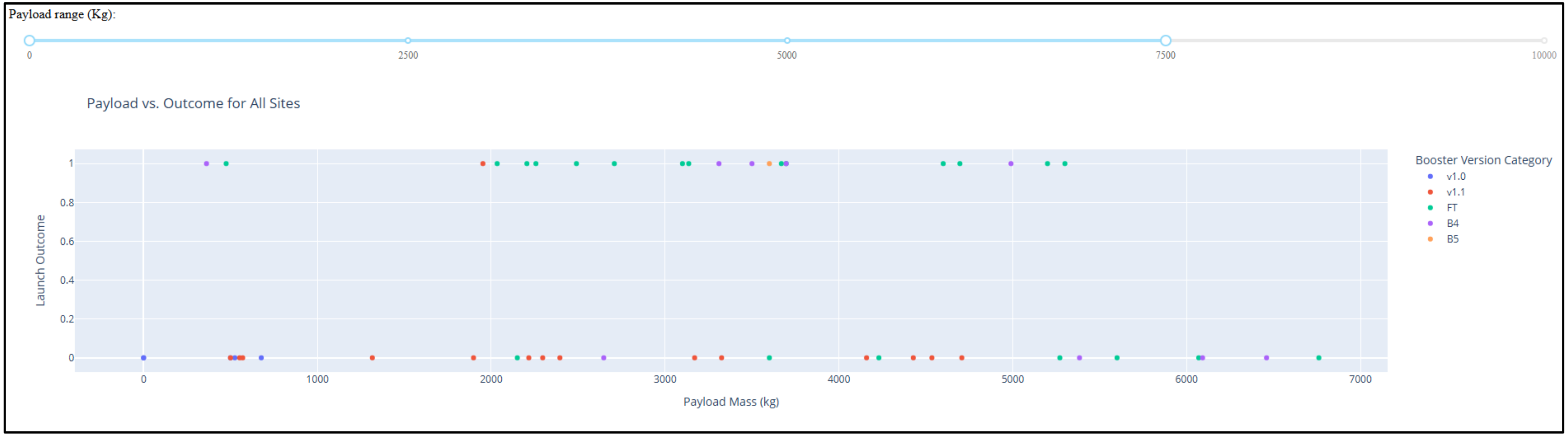
This screenshot shows the launch outcome (success =1, failure = 0) for all sites

The range slider for Payload has been set to 10,000 kg. From this, we can clearly see that the Booster version FT has the highest number of successful outcomes (green dots)

We can also see that the most successful Payload (kg) is between 2k to 6k, however, this range also contains the most failures

The next two slides show results for different Payload ranges by adjusting the Payload slider

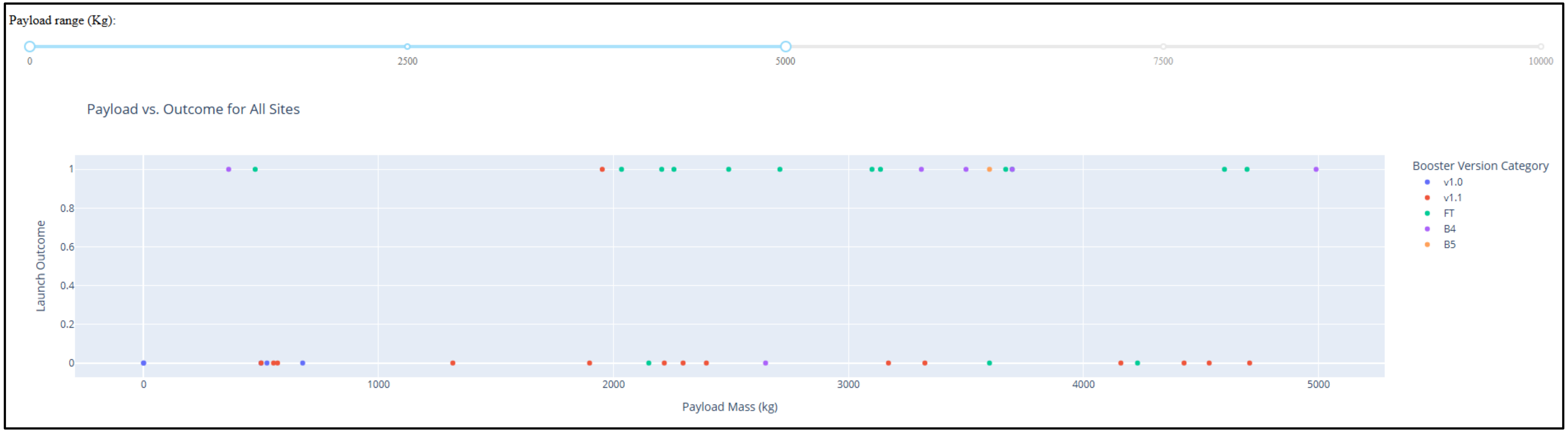
PAYLOAD RANGE UP TO 7,500 KG



The Payload slider has been set to a maximum of 7,500kg. This enables us to zoom into the results for this range.

It's clear to see that there were no successful outcomes between 6,000 and 7,000 kg

PAYLOAD RANGE UP TO 5,000 KG



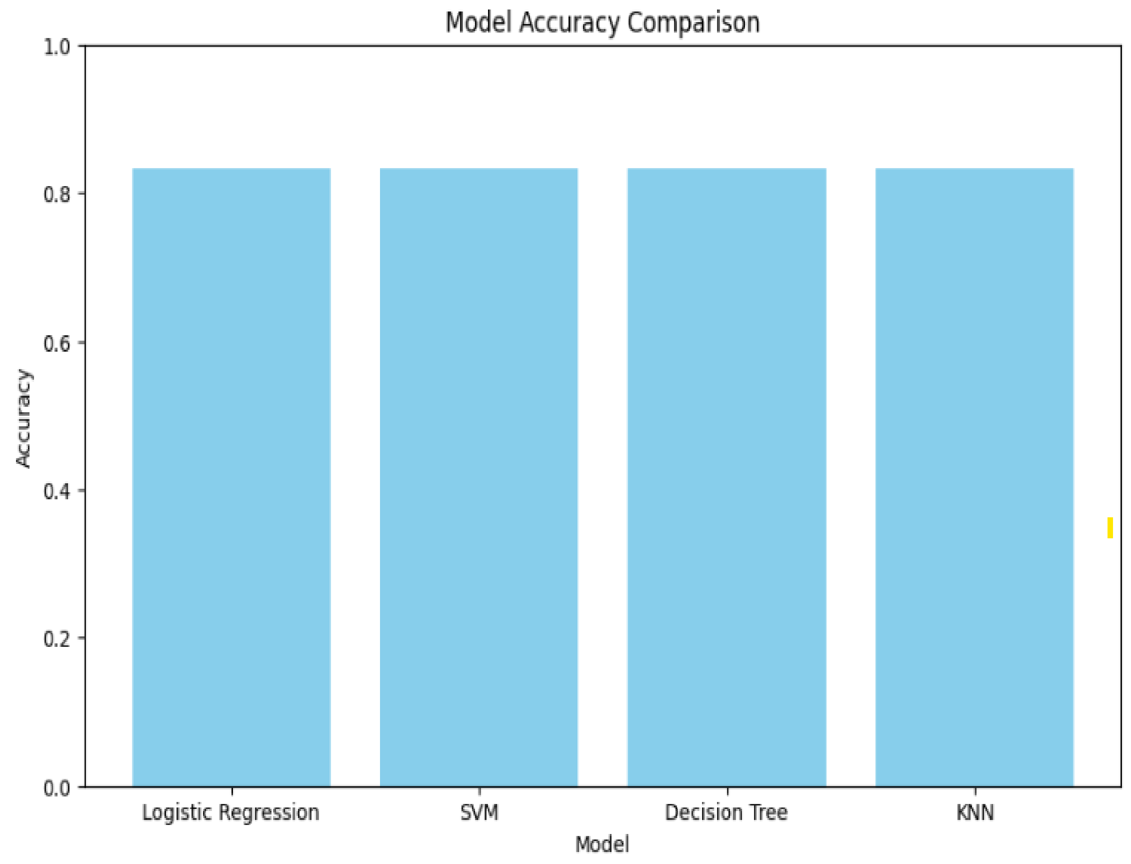
The Payload slider has been set to a maximum of 5,000kg. This enables us to zoom into the results for this range.

It's clear to see that between 1,000 and 2,000kg there was only 1 successful outcome using Booster version v1.1

4. PREDICTIVE ANALYSIS

The following slides shows a comparison of how each of the predictive models performed based on their accuracy score. A confusion matrix, highlighting the number of true positives and false negatives, is also displayed.

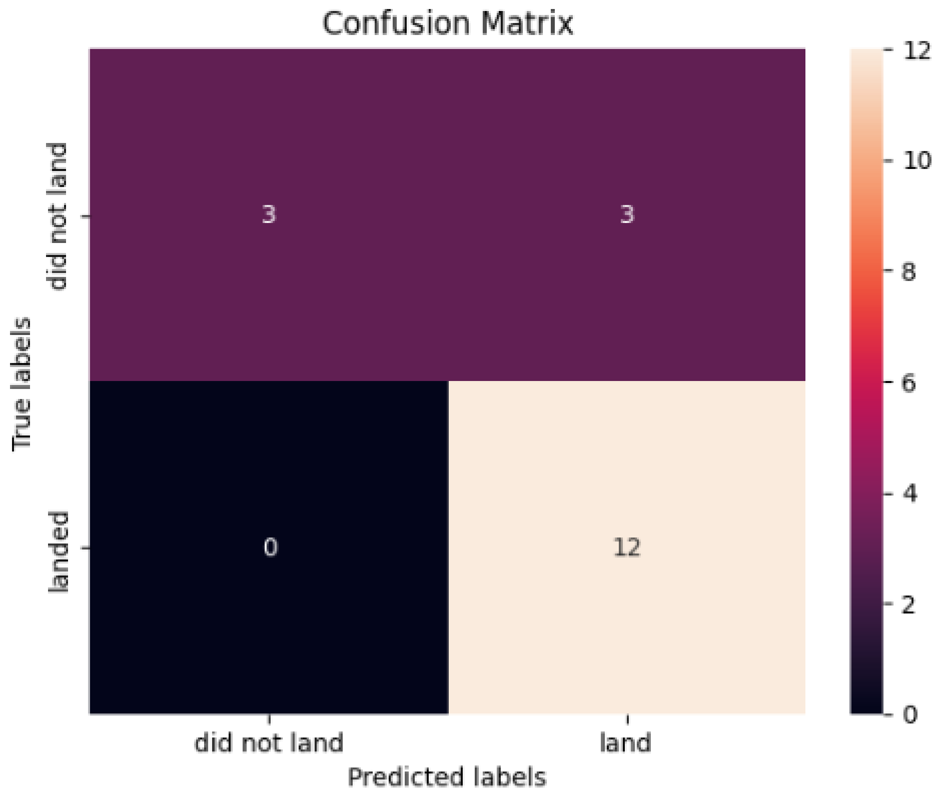
CLASSIFICATION ACCURACY



The bar chart shows that models have the same accuracy score of 0.833

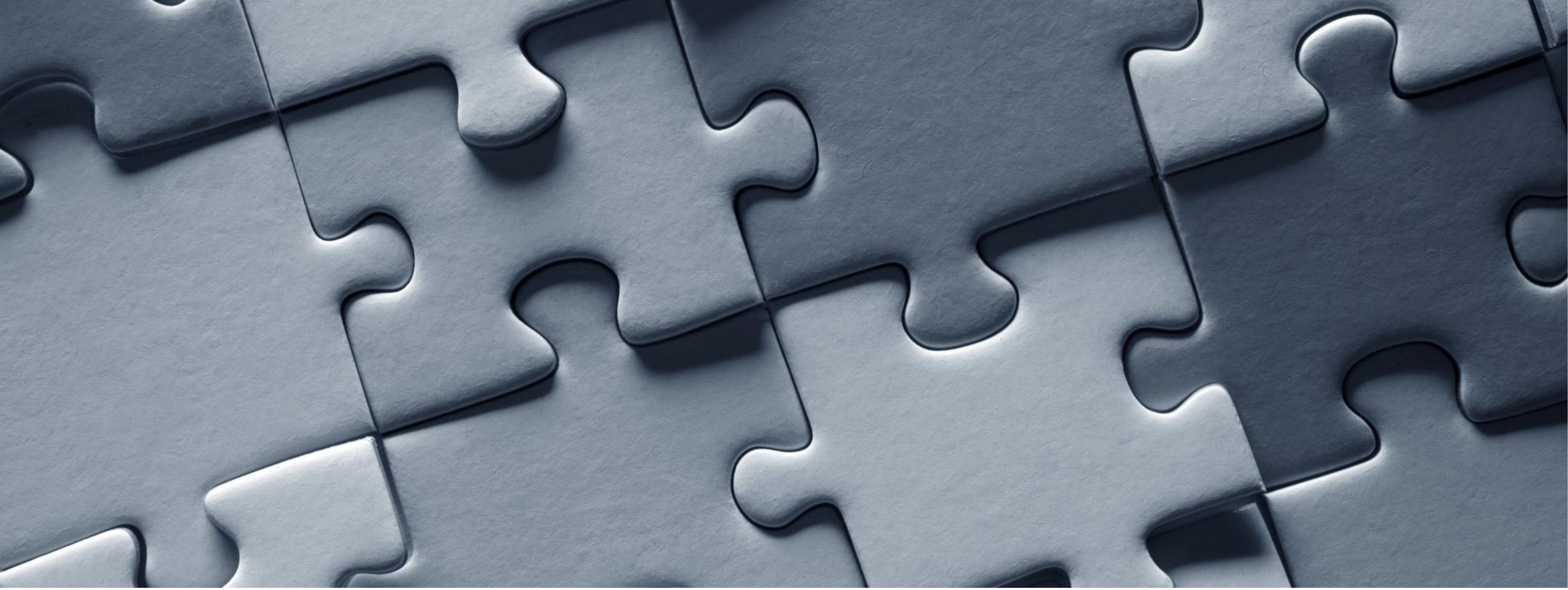
This is likely because the size of the test data is relatively small

CONFUSION MATRIX



This confusion matrix is the same for each classification model

All models have 12 True Positives & 3 False Negatives



5. CONCLUSION

CONCLUSION

The following main conclusions can be drawn from this analysis:

1. All models perform equally with a test accuracy of 83.33%
2. As the Payload increases beyond 5,000 kg the success rate falls dramatically
3. The launch site with the highest success rate of 76.9% is KSC LC-39A
4. Despite a fall in 2018, the average success rate for landings has increased every year from 2013 to 2019
5. Orbit types ES-L1, GEO, HEO & SSO all had a mean success rates of 100%.

THANK YOU