IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alberto Restrepo Lozano
July 4th, 2022

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- SpaceX is able to get more contracts as it is able to reduce costs by landing its main booster. They can more halve the costs of other competitors.

- We need to be able to predict if SpaceX is going to be able or not to land its main booster. If they don't, then the competitors can bid on the project.

Section 1

Methodology

Methodology

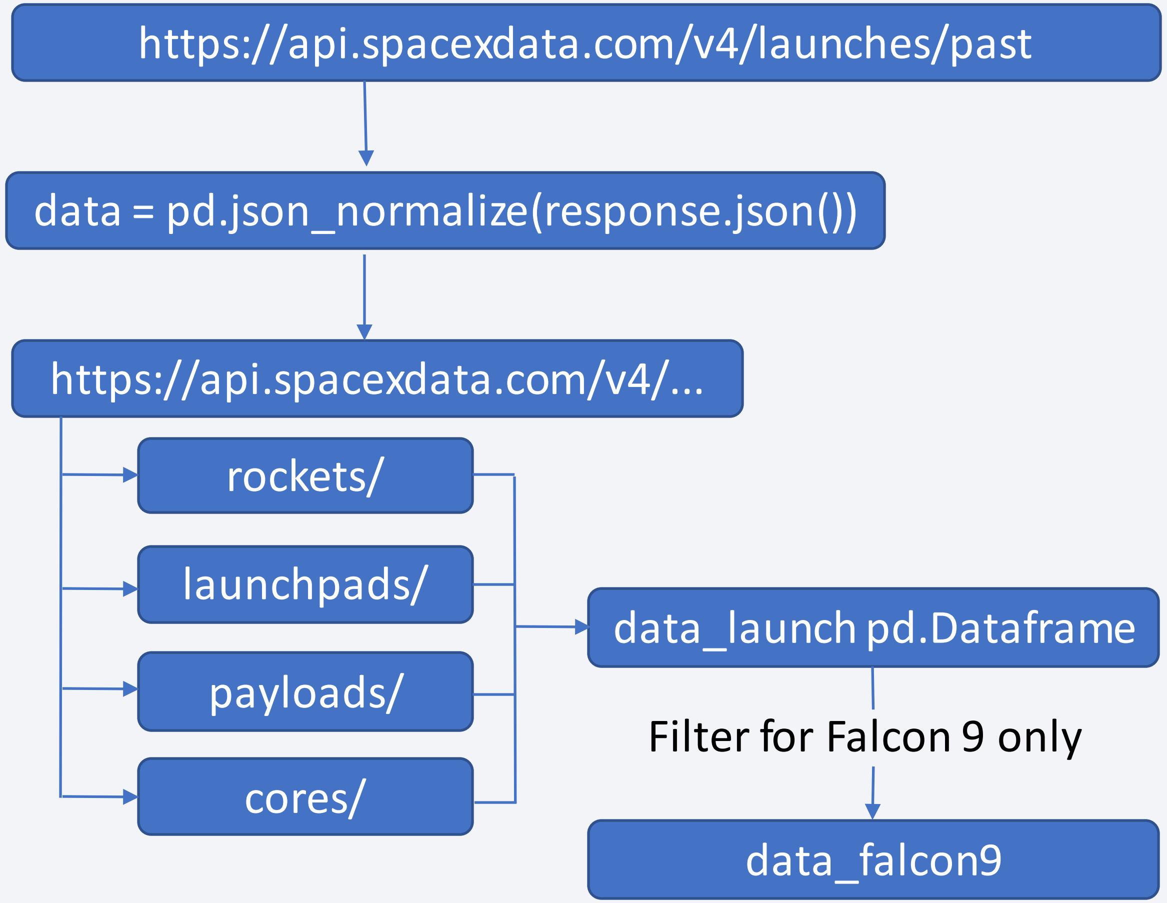
Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX REST API and Beautiful Soup on Wikipedia page
- Perform data wrangling
  - Data was converted to categorical and then one-hot encoded for processing, except for the payload, which was left as an integer.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Four models were built after standardizing the data and they were tested for their accuracy and also a confusion matrix to detect false positives.

Data Collection – SpaceX API

- The dataset is first collected from past launches
- Then precise data from this is collected using various calls
- Finally, the data is arranged into a data frame and filtered to show only the desired launches.

- GitHub link: https://github.com/caiman16/capstone_project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



https://api.spacexdata.com/v4/launches/past

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

https://api.spacexdata.com/v4/...

rockets/

launchpads/

payloads/

cores/

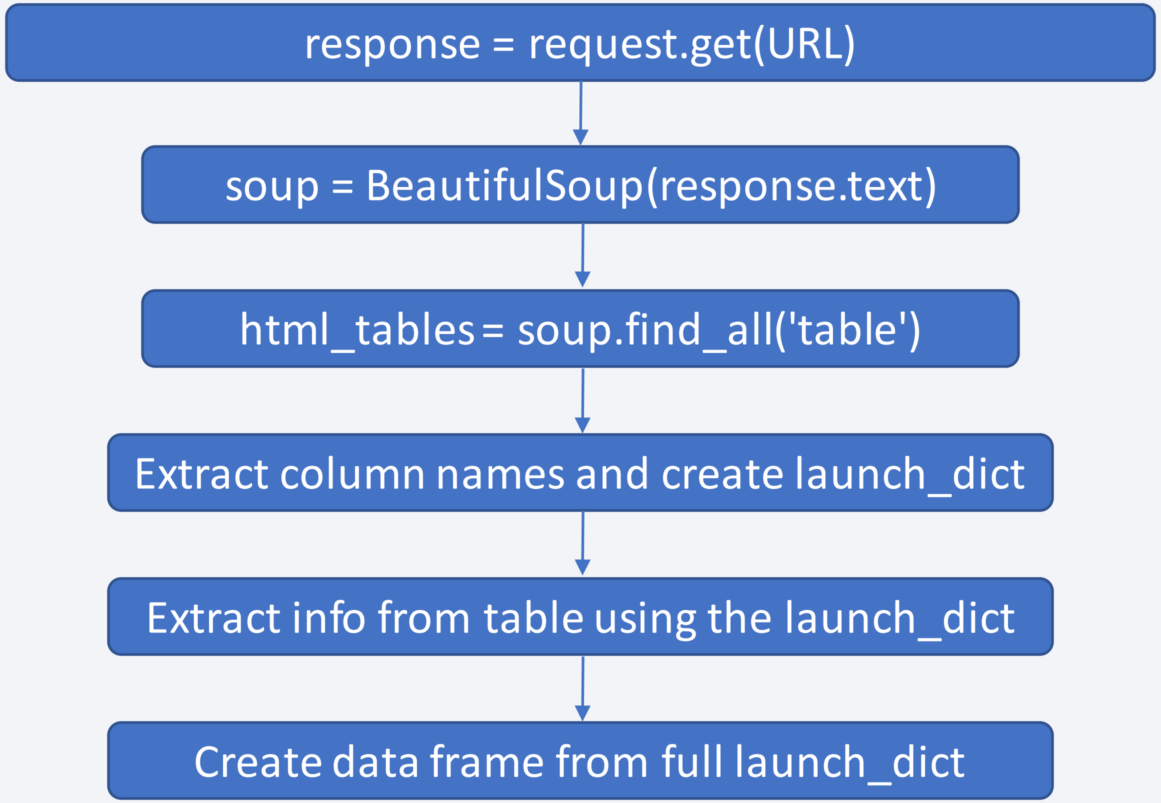
data_launch pd.Dataframe

Filter for Falcon 9 only

data_falcon9

Data Collection - Scraping

- Use requests.get on the URL
- Use BeautifulSoup on the response
- Find the tables and collect the information from all the rows
- GitHub link: https://github.com/caiman16/capstone_project/blob/main/jupyter-labs-webscraping.ipynb



response = request.get(URL)

soup = BeautifulSoup(response.text)

html_tables = soup.find_all('table')

Extract column names and create launch_dict

Extract info from table using the launch_dict

Create data frame from full launch_dict

Data Wrangling

- First, missing values for the payload mass were replaced with the mean
- Then good outcomes and bad outcomes were grouped to get an insight on success rate
- Categorical data was one-hot-encoded to work better with a future model
- Data was filtered to use only the one that was needed
- GitHub link: https://github.com/caiman16/capstone_project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Chart Flight Number vs Payload Mass: To see the evolution of the landings and how payload affects it.
- Chart Flight Number vs Launch Site: To see how the Launch Site affects the probability of landing and its evolution.
- Chart Payload Mass vs Launch Site: To see how the payload affects where the launch can take place.
- Chart Orbit vs Success Rate: To see the relation between the type of orbit and how likely it is succeed.
- Chart Flight Number vs Orbit: To see the evolution of orbits that were tried.

EDA with Data Visualization 2

- Chart Payload Mass vs Orbit: To see how the payload mass affects the type of orbits that are tried.
- Chart Year vs Success Rate: To see the evolution of success across time.
- GitHub link: https://github.com/caiman16/capstone_project/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- SQL Queries performed:
  - Select the distinct launch sites
  - Display the first five records for the "CCA" Launch Site
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display the average payload mass carried by booster version F9 v1.1
  - Select the first date of a successful landing outcome on a ground pad
  - List the names of the boosters with successful drone ship landings and payloads between 4000 and 6000 kg
  - List the total successful and failure mission outcomes
  - List the booster versions which have carried the maximum payload mass

EDA with SQL 2

  - List the records for failure drone ship landings in 2015
  - Rank the successful landing outcomes between 04-06-2010 and 20-03-2017 by type
- GitHub link: https://github.com/caiman16/capstone_project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Started by creating a base map
- Created circles and markers for the launch sites
- Created marker clusters with color codes to know whether or not the landing had been successful
- Created markers and lines to show the distance between launch sites and relevant places (coastline, cities, railway and highway)
- GitHub link: https://github.com/caiman16/capstone_project/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- There are two plots: a pie chart that shows the rate of success of landing; a scatter plot that shows the success of landing by booster type and by payload mass
- There are two interactions: a dropdown menu to select the site or sites to be evaluated; a payload range slider to select the range to study
- These combinations allow for the isolation of characteristics to get different insights as which launch sites has the highest success rate and how payload mass affects the landing success
- GitHub link: https://github.com/caiman16/capstone_project/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Four classification models were chosen:
  - Logistic Regression (LR)
  - Support Vector Machine (SVM)
  - Decision Tree (Tree)
  - K-nearest Neighbors (KNN)
- All data was standardize and normalized using StandardScaler()
- Accuracy was used as the score parameter and a decision matrix was plotted to get the hang about false positives
- GitHub link: https://github.com/caiman16/capstone_project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

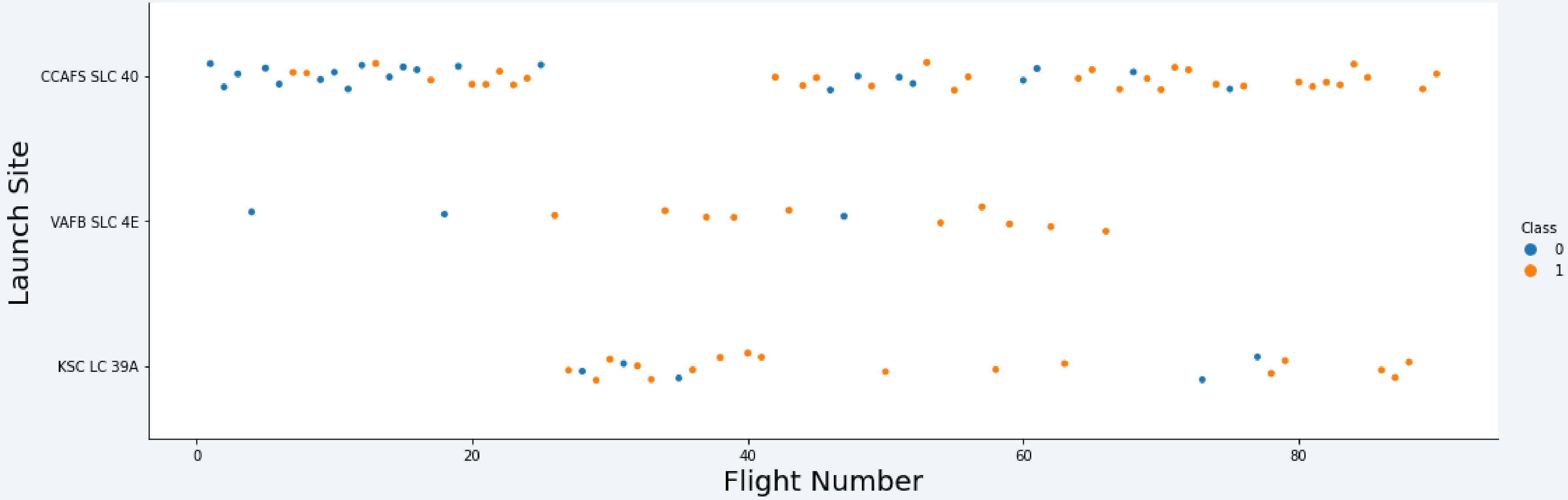
Results

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

Section 2

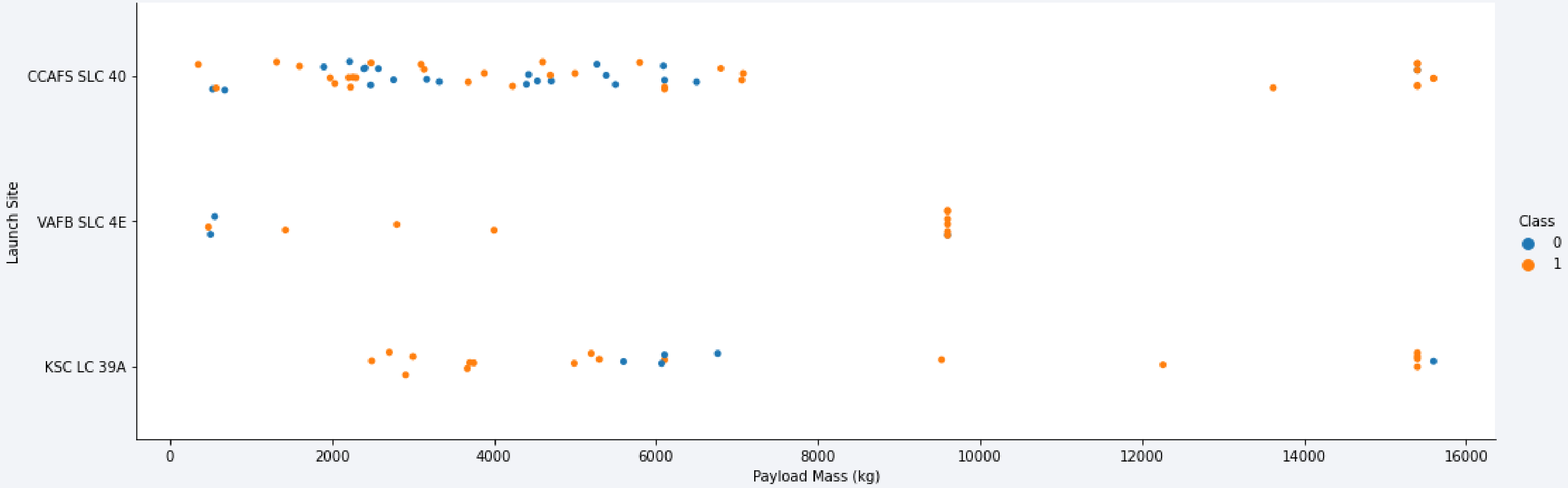
Insights drawn from EDA

Flight Number vs. Launch Site



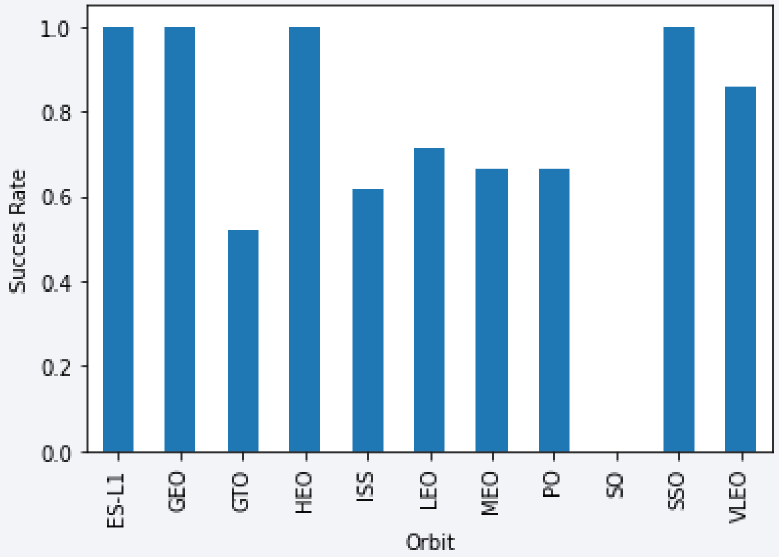
- CCAFS is the most used site by far. There was a gap in site usage after a failure
- KSC seems to be the site with the highest success rate
- As the flight numbers grow, more successes happen

Payload vs. Launch Site



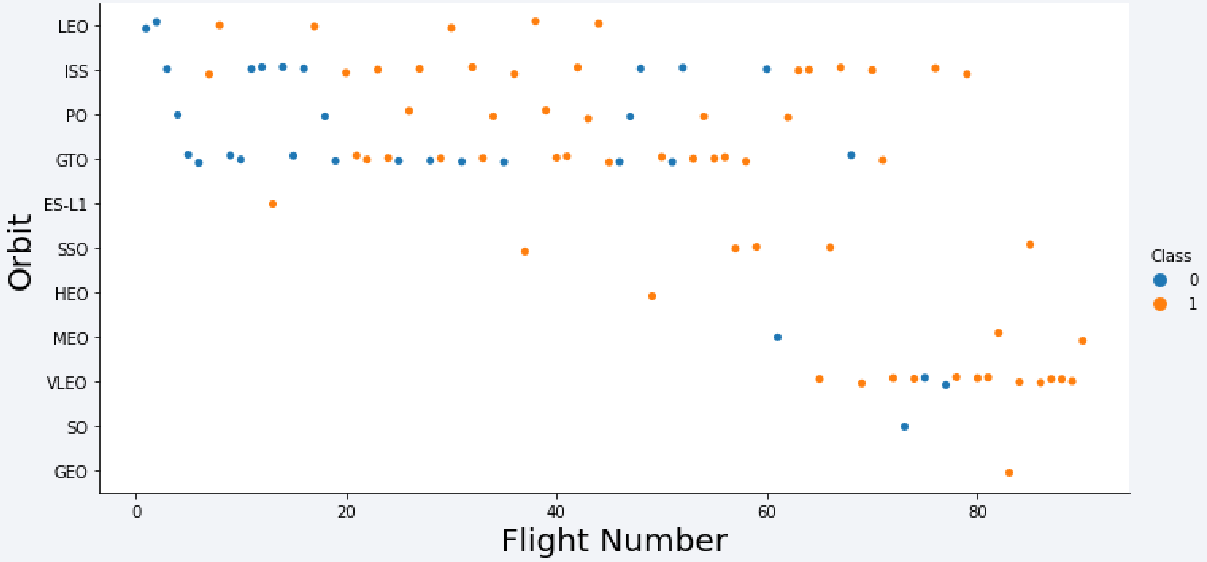
- VAFB seems to have a hard upper limit and only failed on the lower payloads
- Higher payloads seem to have a better chance of success
- CCAFS has the lowest success rate, but has a perfect rate when payload >7000 kg

Success Rate vs. Orbit Type



- ES-L1, GEO, HEO, SSO and VLEO have success rates of over 80%
- SO orbits have the lower success rate (0), followed by GTO orbits(~50%)

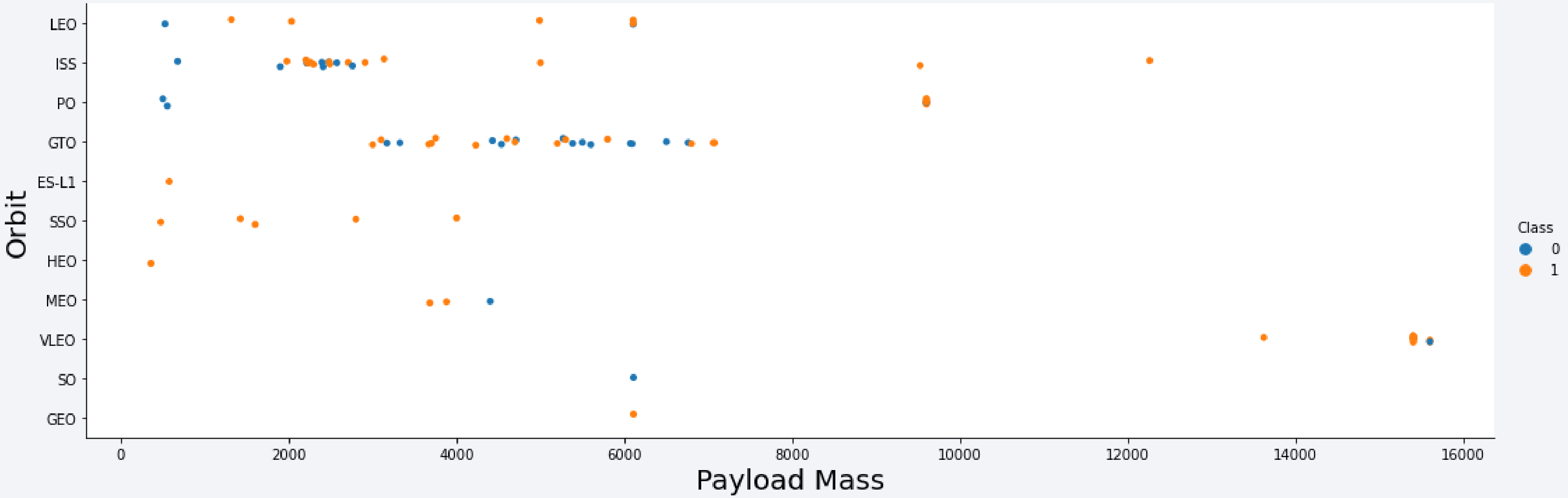
Flight Number vs. Orbit Type



- Apart from the SSO orbit, all the others that had perfect records have only 1 attempt
- At first, 4 types of orbit were tried. Lately, most launches are to the VLEO orbit

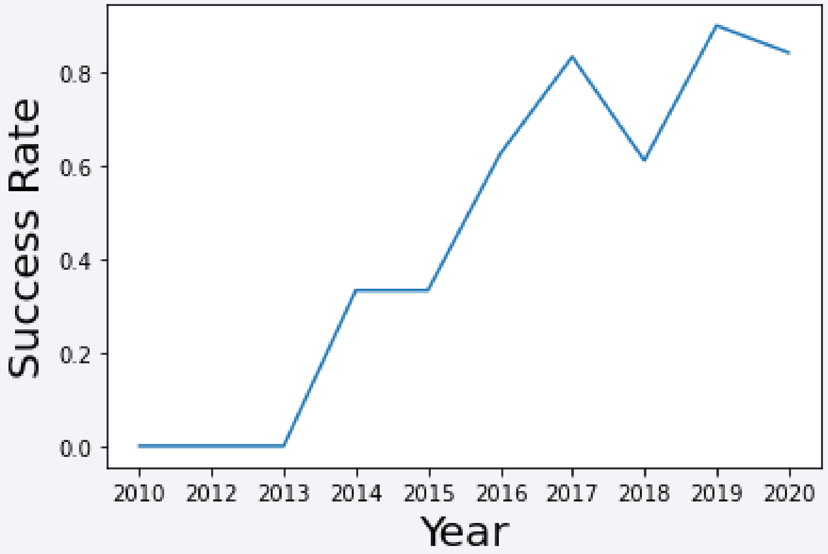
As it was seen before, with time, success rate has increased

Payload vs. Orbit Type



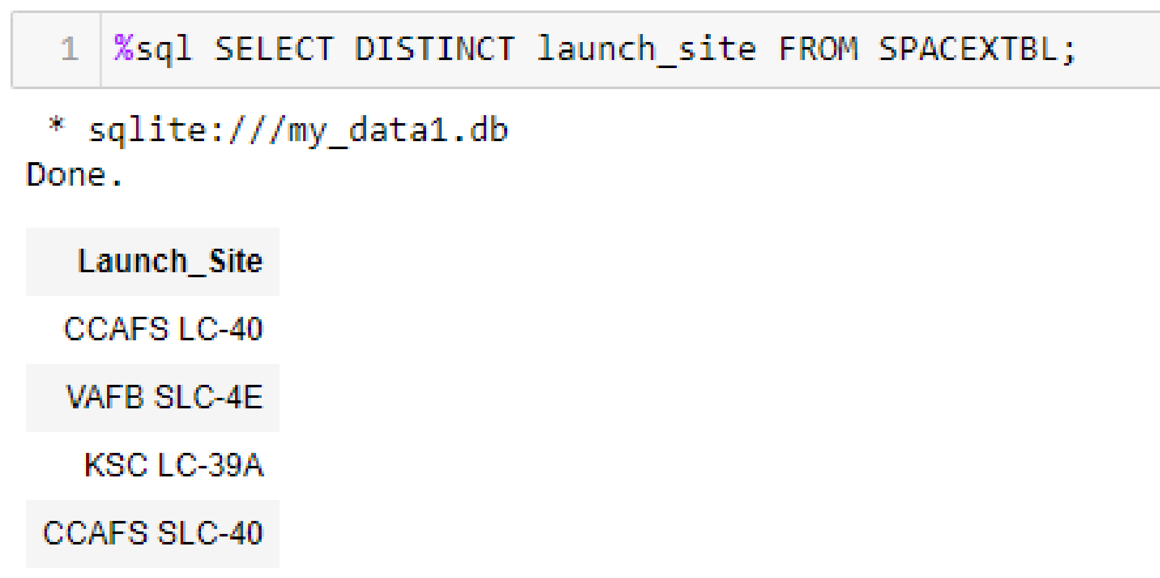
- Payloads to SSO orbit are all >4000 kg
- Higher payloads (>8000 kg) only go to VLEO, PO and ISS orbits
- GTO orbits have a very specific payload range between 2500 and 7500 kg

Launch Success Yearly Trend



- With the years, success has a positive trend generally
- There was a dip in 2018, where success was less than 2017
- At first, success was very low

All Launch Site Names



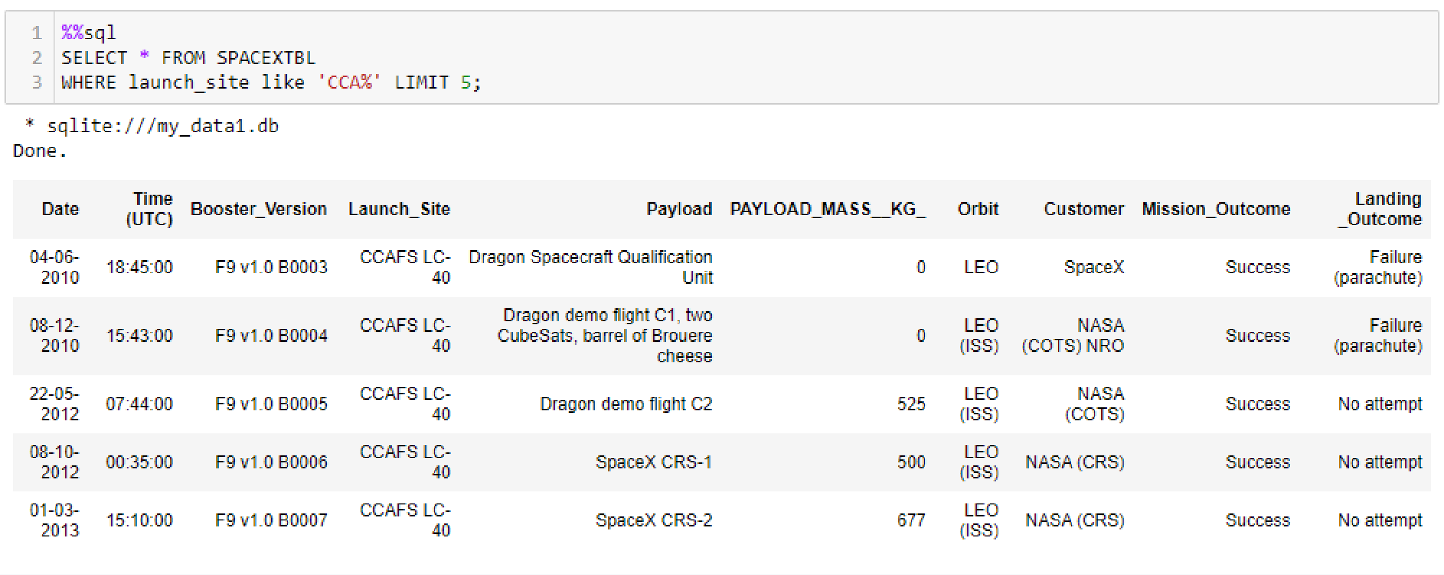
```
%sql SELECT DISTINCT launch_site FROM SPACEXTBL;
```

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

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

- 4 unique launch sites are present in the dataset. By using UNIQUE, there are no duplicates

Launch Site Names Begin with 'CCA'



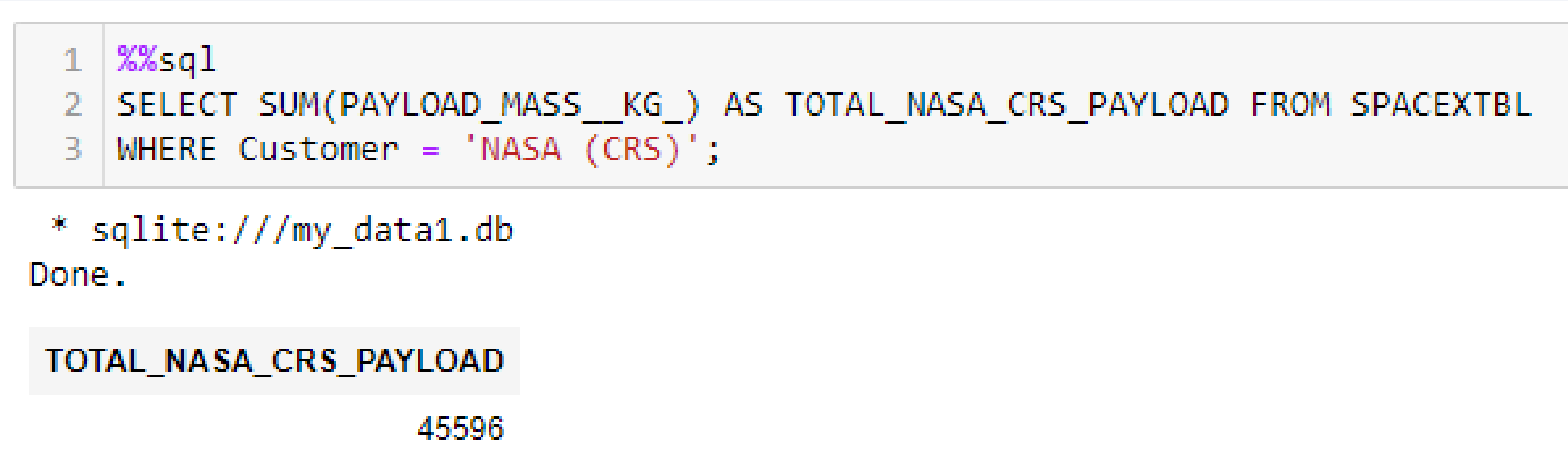
```
%%sql
SELECT * FROM SPACEXTBL
WHERE launch_site like 'CCA%' LIMIT 5;
```

```
 * 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

- By using a combination of LIKE and LIMIT, only the selected number of records from the specified launch site are retrieved

Total Payload Mass



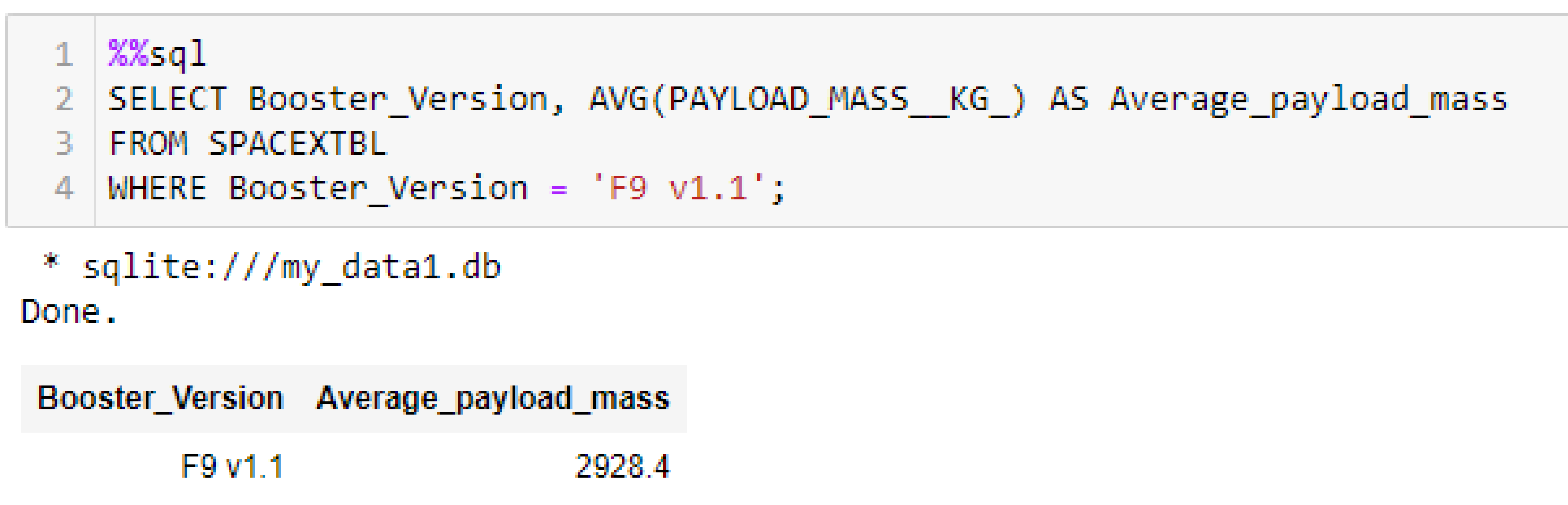
```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_NASA_CRS_PAYLOAD FROM SPACEXTBL
WHERE Customer = 'NASA (CRS)';
```

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

TOTAL_NASA_CRS_PAYLOAD
45596

- By using the aggregate function SUM(), the total amount of payload sent by NASA is retrieved

Average Payload Mass by F9 v1.1



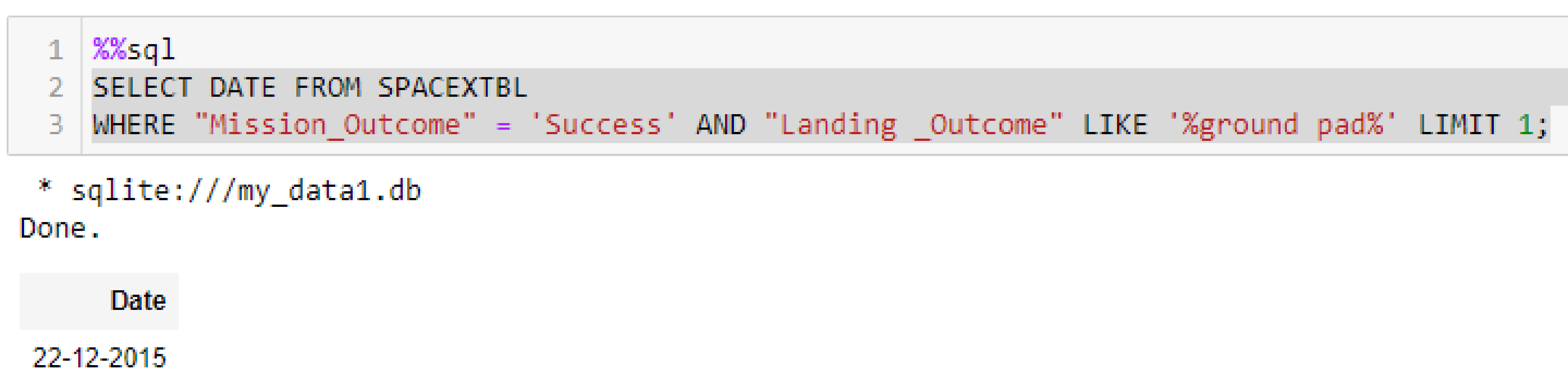
```
%%sql
SELECT Booster_Version, AVG(PAYLOAD_MASS__KG_) AS Average_payload_mass
FROM SPACEXTBL
WHERE Booster_Version = 'F9 v1.1';
```

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

Booster_Version	Average_payload_mass
F9 v1.1	2928.4

- By using the AVG() aggregate function and WHERE clause, only the average payload mass for the specified booster version is retrieved

First Successful Ground Landing Date



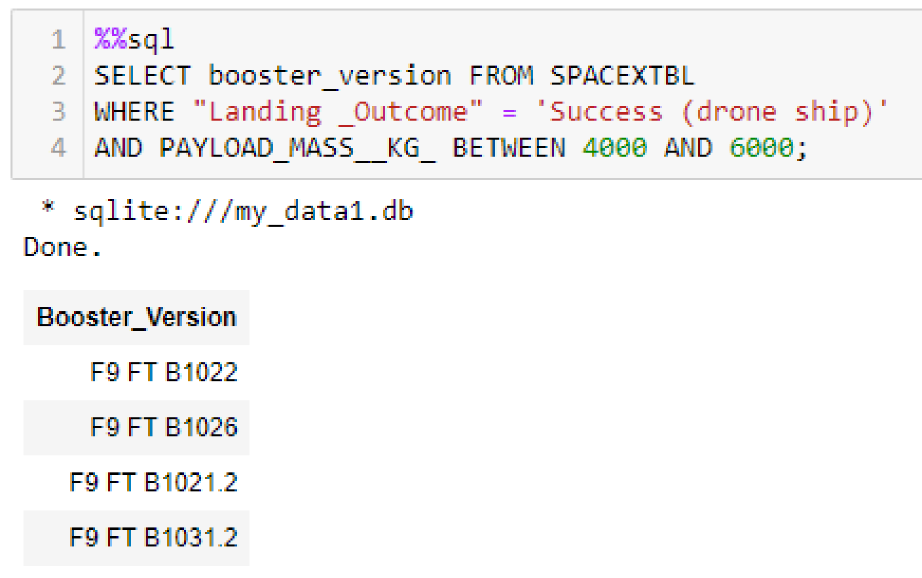
```
%%sql
SELECT DATE FROM SPACEXTBL
WHERE "Mission_Outcome" = 'Success' AND "Landing _Outcome" LIKE '%ground pad%' LIMIT 1;
```

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

Date
22-12-2015

- By combining 2 WHERE clauses and limiting the result to 1, the first occurrence is retrieved

Successful Drone Ship Landing with Payload between 4000 and 6000



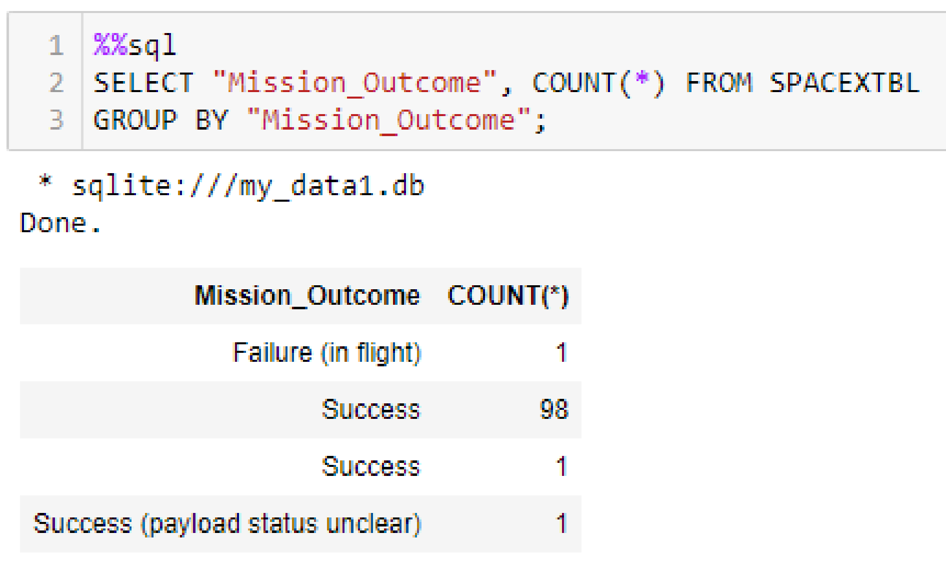
```
%%sql
SELECT booster_version FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

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

- By combining 2 WHERE clauses, in which one of them uses the BETWEEN option, a range of records is retrieved which satisfy the other clause too

Total Number of Successful and Failure Mission Outcomes



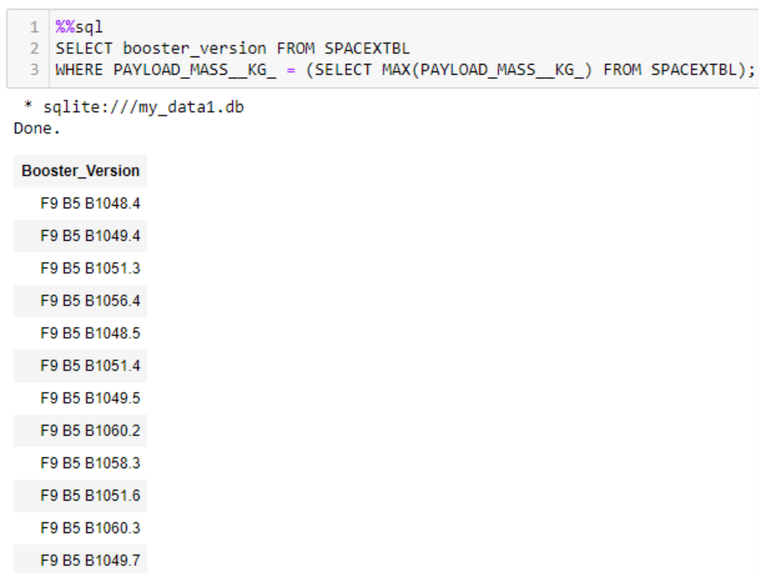
```
%%sql
SELECT "Mission_Outcome", COUNT(*) FROM SPACEXTBL
GROUP BY "Mission_Outcome";
```

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

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

- By using the aggregate function COUNT() and the GROUP BY function, the total count for each unique group can be retrieved

Boosters Carried Maximum Payload



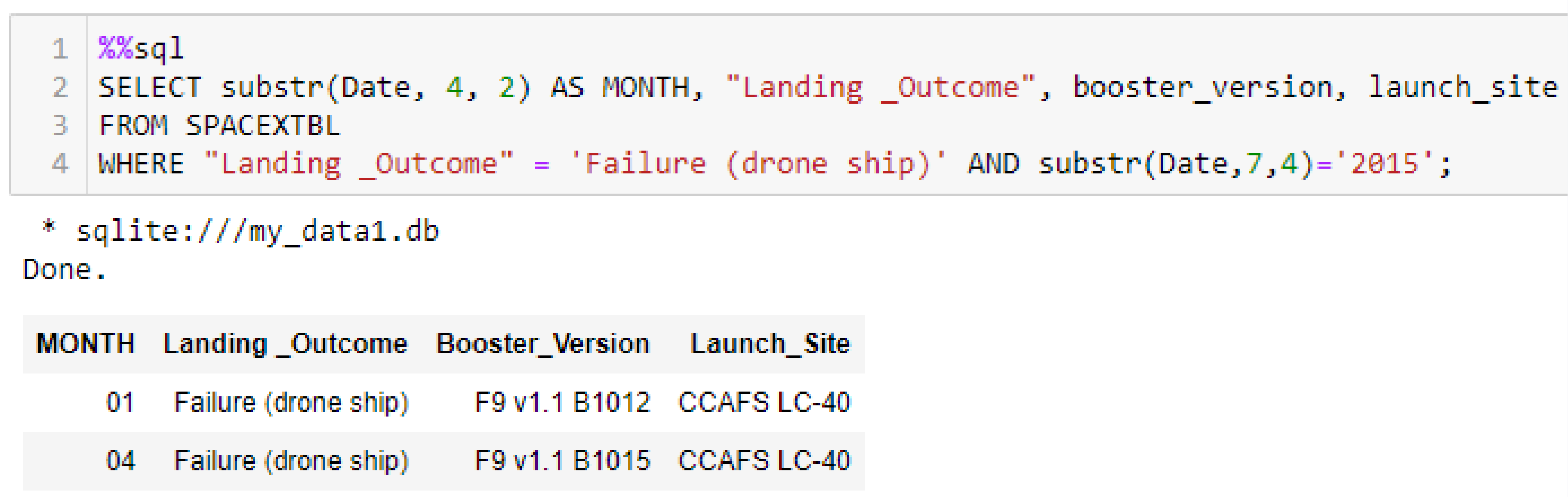
```
%%sql
SELECT booster_version FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * 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

- By using a subquery in the WHERE clause, we can SELECT the MAX() payload as the condition

2015 Launch Records



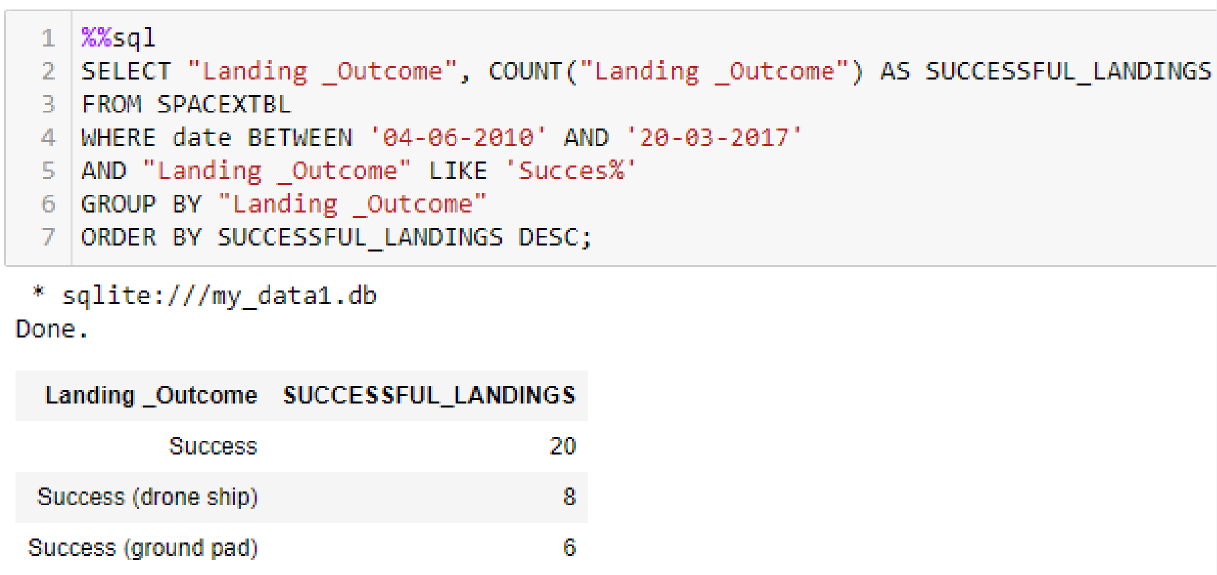
```
%%sql
SELECT substr(Date, 4, 2) AS MONTH, "Landing _Outcome", booster_version, launch_site
FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Failure (drone ship)' AND substr(Date,7,4)='2015';
```

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

MONTH	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- By using substr(Date, #, #), the required month and year can be selected from the Date column. As such, we can search for the months by passing the year

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



```
%%sql
SELECT "Landing _Outcome", COUNT("Landing _Outcome") AS SUCCESSFUL_LANDINGS
FROM SPACEXTBL
WHERE date BETWEEN '04-06-2010' AND '20-03-2017'
AND "Landing _Outcome" LIKE 'Succes%'
GROUP BY "Landing _Outcome"
ORDER BY SUCCESSFUL_LANDINGS DESC;
```

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

Landing _Outcome	SUCCESSFUL_LANDINGS
Success	20
Success (drone ship)	8
Success (ground pad)	6

- By using the aggregator function COUNT(), GROUP BY function on the unique landing outcomes that WHERE also successful and BETWEEN the requested dates, the rank is retrieved

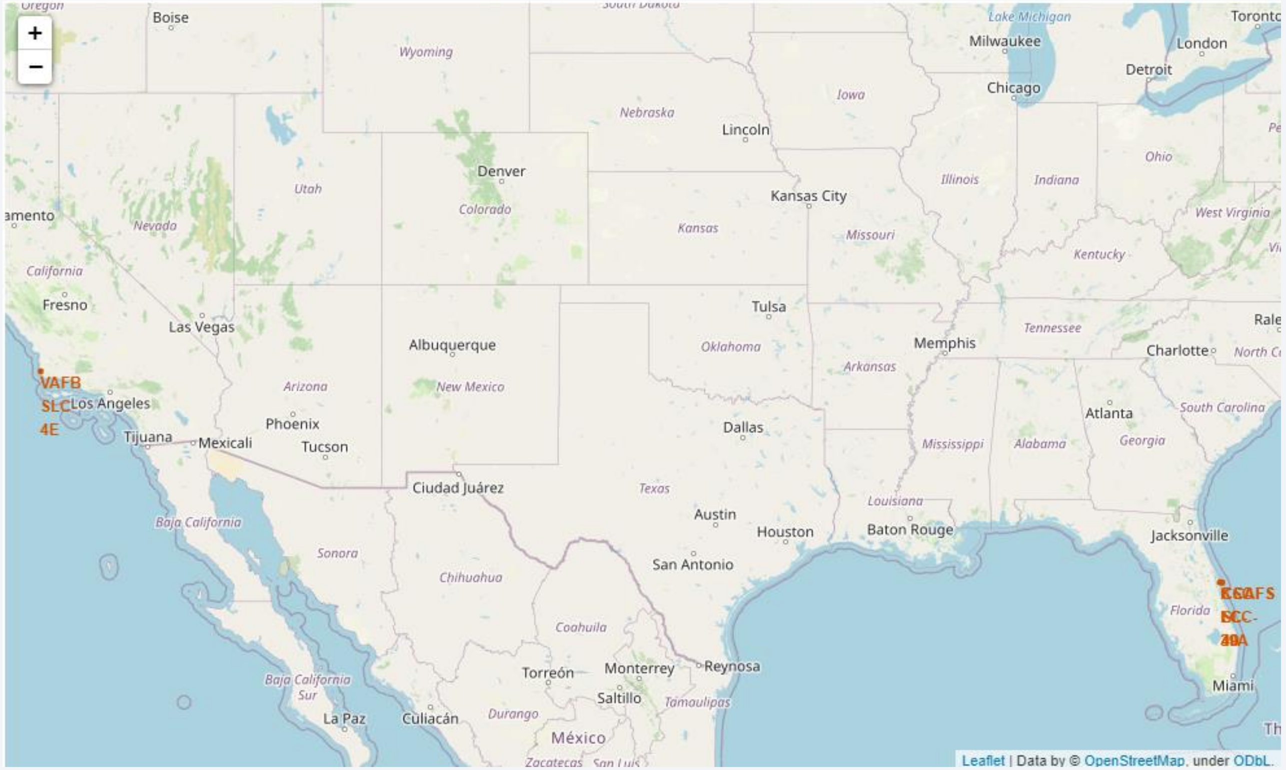
- Also, by using ORDER BY and DESC, the list is ordered in descending order

Section 3

Launch Sites Proximities Analysis

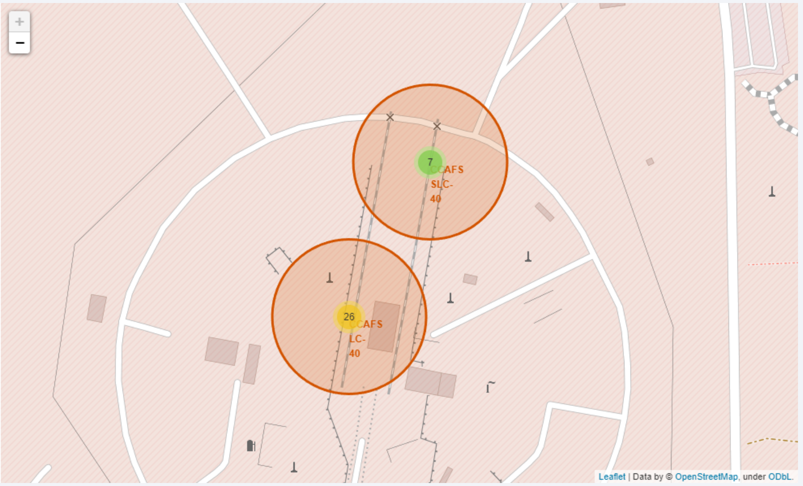
Launch Sites Locations

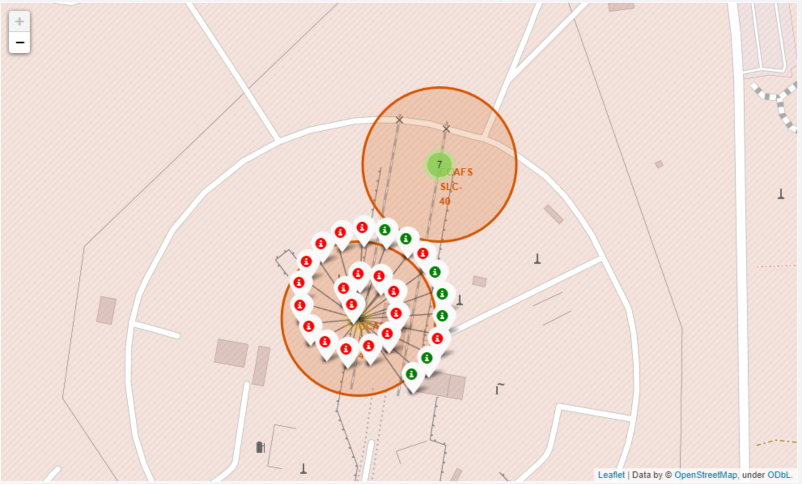
- The map shows the launch sites location on both coasts
- The locations are on southern part of the United States
- They are also very close to the coastline



Landing Outcomes for CAFS LC-40 Launch Site

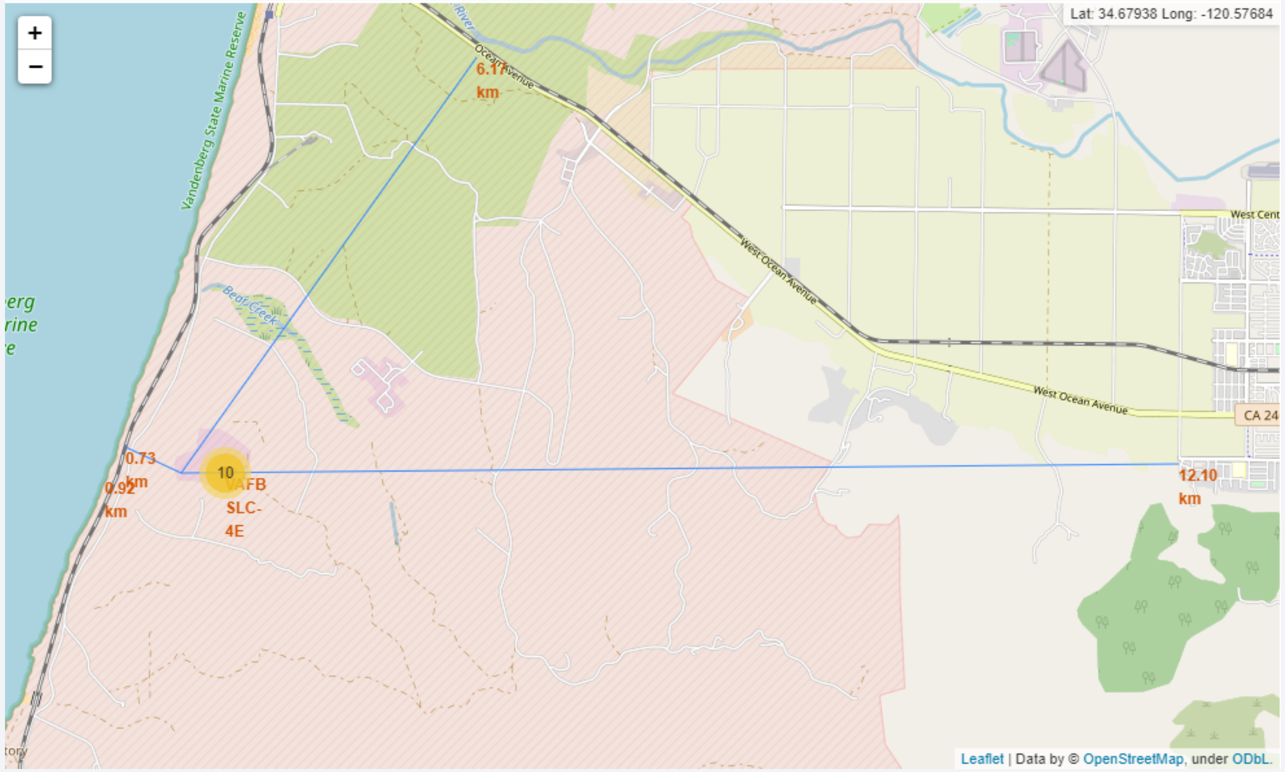
- A cluster of markers show the landing outcomes on the selected site
- Before being developed, all the markers are clustered
- Once developed, details can be obtained





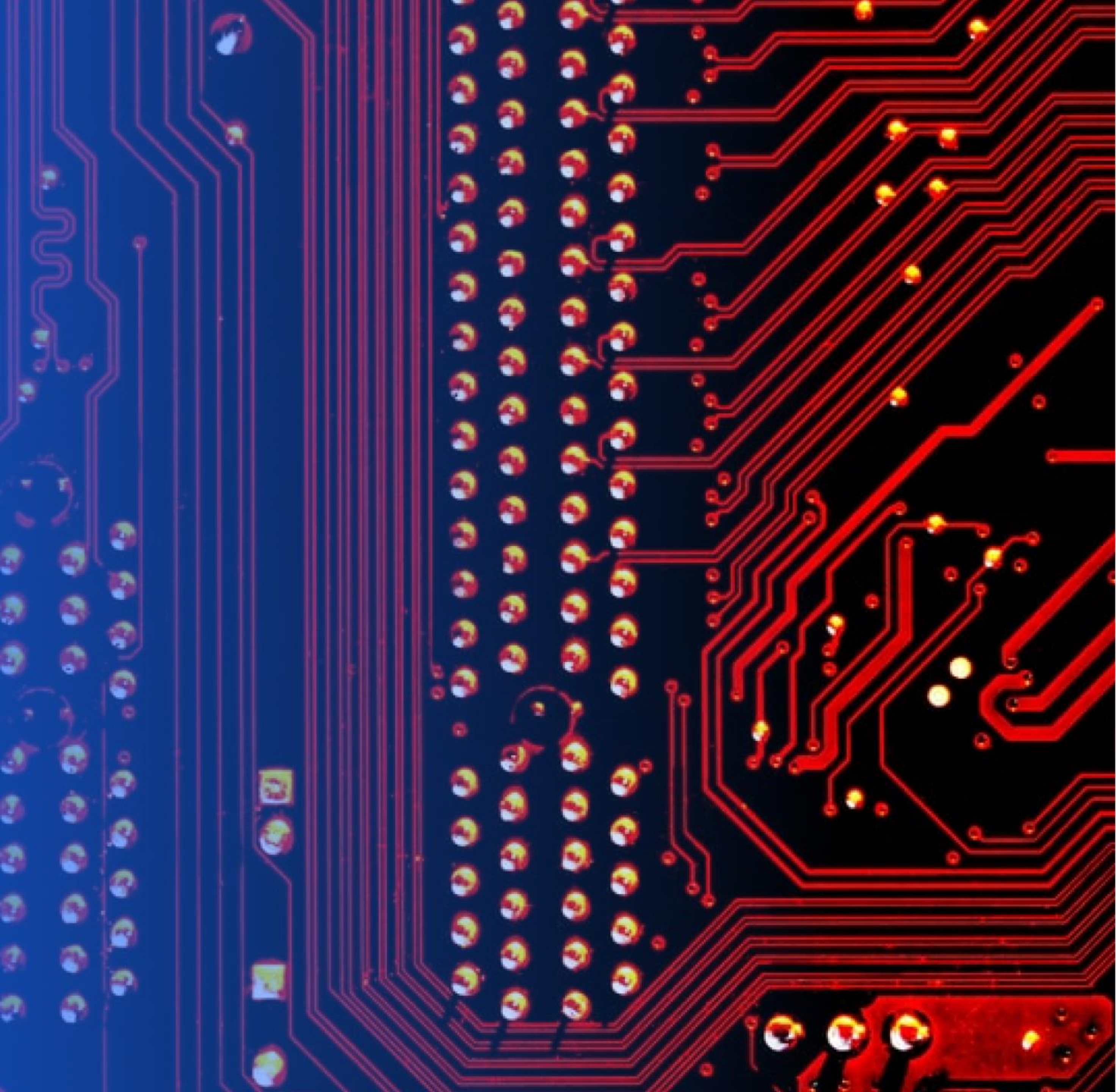
Proximity to Different Objects on VAFB SLC-4E

- The map shows the distance to the coastline, nearest city, nearest railway and nearest highway
- In general, the sites are near the coastline and near a railway
- On the East, the highway is also close by, but on the West is not as close
- The nearest city tends to be further away

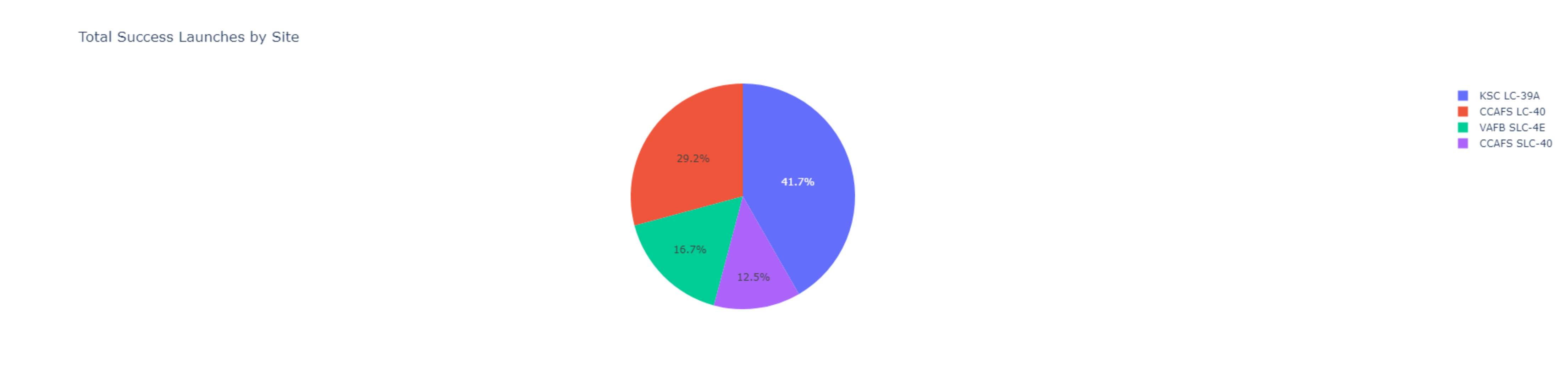


Section 4

Build a Dashboard with Plotly Dash

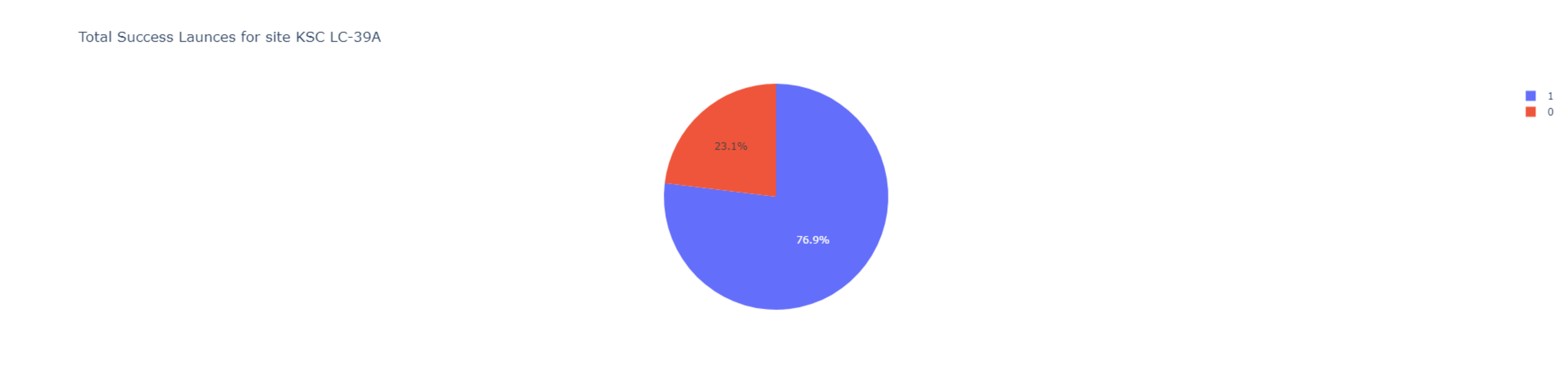


Landing Success Launches by Site



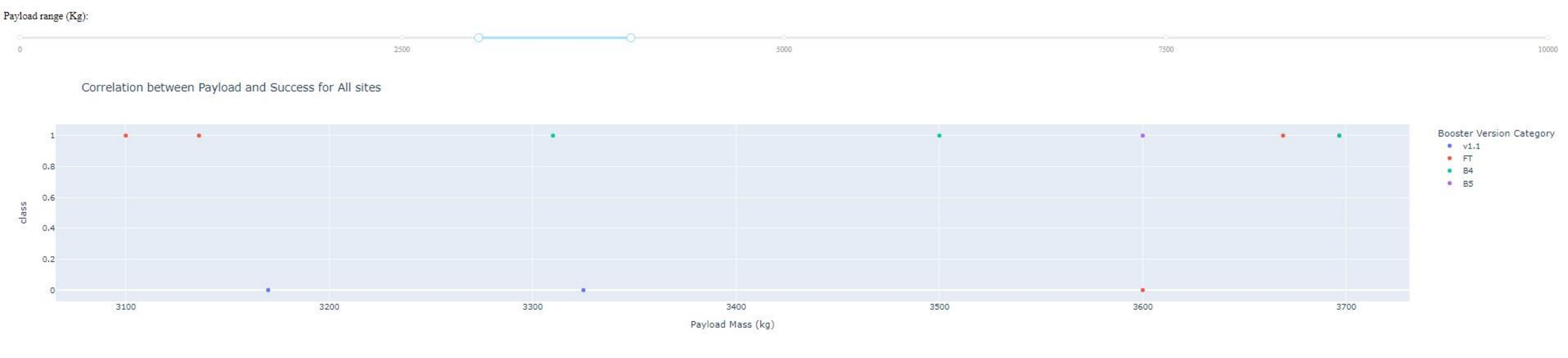
- By using the dropdown menu, ALL SITES are chosen and the result is the different success rates of each site
- KSC LC-39A has the highest proportion of success rate, followed by CCAFS LC-40

KSC LC-39A Total Success Launches Rate



- By selecting KSC LC-39A, its success rate can be shown, where more than 3 of every 4 launches is successful

Payload Range with Highest Success Rate



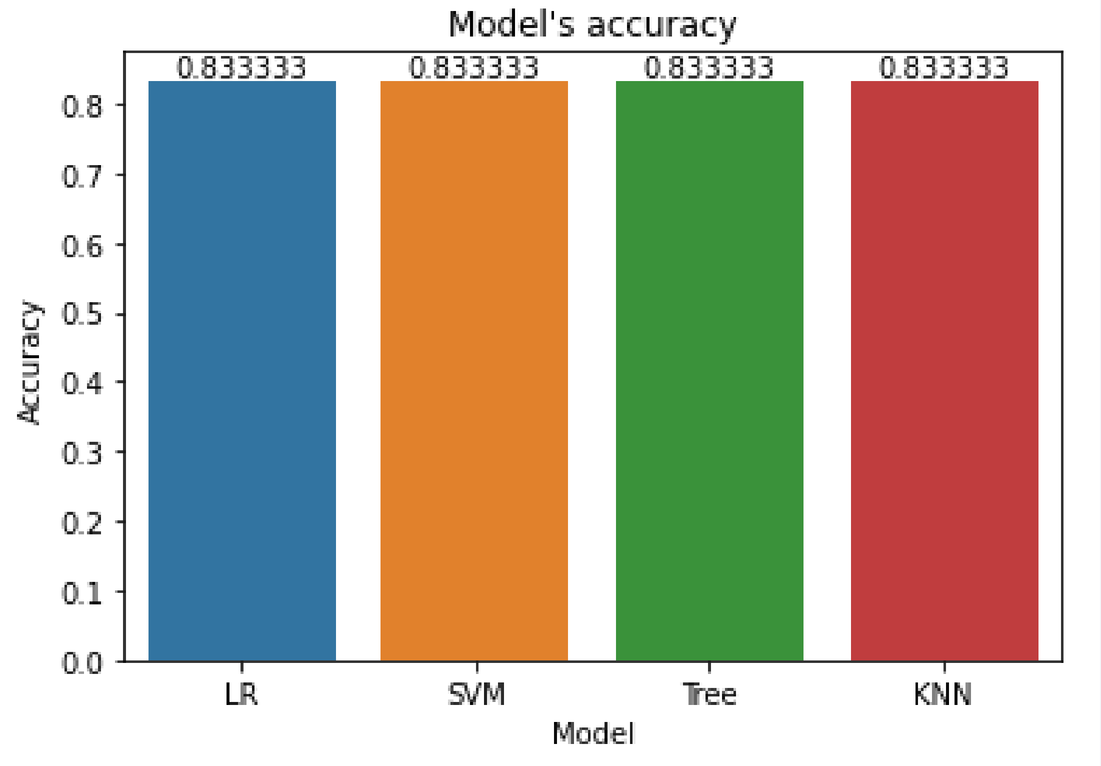
- By using the slider, the payload range with the highest success rate is bewteen 3000 and 4000 kg
- This range has a 70% success rate

Section 5

Predictive Analysis (Classification)

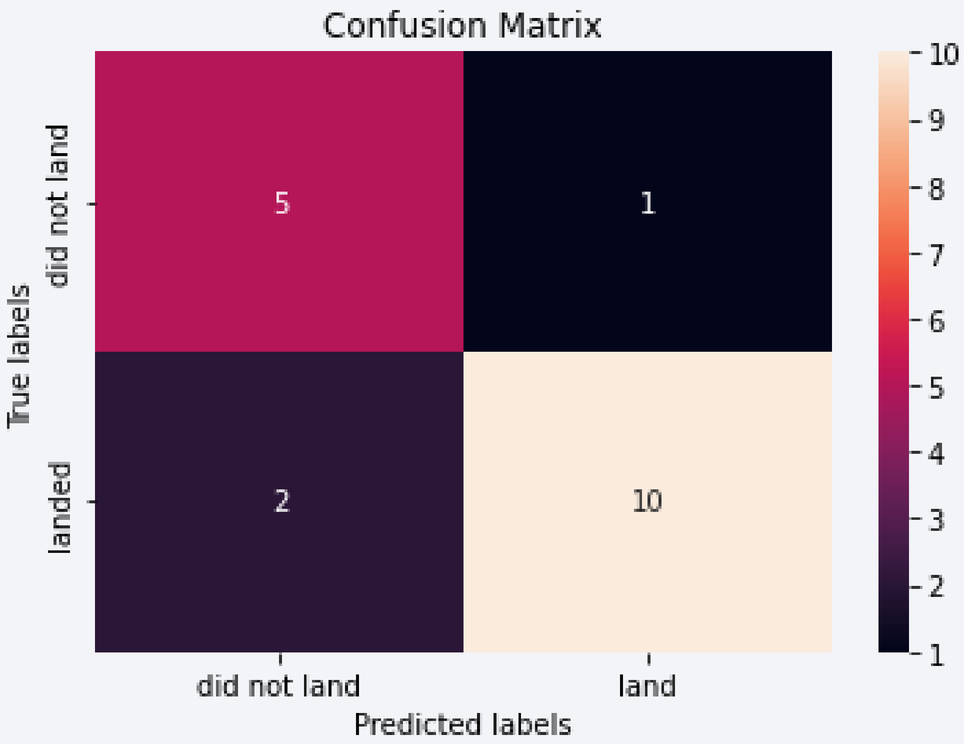
Classification Accuracy

- All the models returned an equal accuracy of 0.83
- As such, another method will need to be used to decide which model to use



Confusion Matrix

- This is the confusion matrix for the Decision Tree Model
- It is chosen above the others as it has just 1 false positive, while the others have 3
- Less false positives mean that the model will make less mistakes when predicting if SpaceX will try to land the booster



Conclusions

- SpaceX is improving their landing successes
- Flights that are launched from KSC have a 75% chance of landing the booster successfully
- With heavier payloads, SpaceX will land the booster with better success
- More data will be needed to properly assess which orbits give a better chance of success
- The Decision Tree model can be used as it predicts less false positives than the other models

Thank you!