IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

I Kadek Anom Sukawirasa Putra
Jun 16 2022

Outline

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

Executive Summary

- First, data was gathered from the public SpaceX API and SpaceX Wikipedia page. The ‘class’ label column was created to categorize successful landings. SQL,visualization library such as sns, folium maps, and dash were used to investigate data. List of relevant columns are compiled as features. Using a single hot encoding, all category variables were converted to binary. GridSearchCV was used to determine the best parameters for machine learning models after standardizing data. Finally all of the models’ accuracy scores visualized to choose which best model.

- Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors were the four machine learning models developed. With an accuracy percentage of around 83.33 percent, they all gave similar outcomes. All of the models overestimated the likelihood of successful landings. For better model determination and accuracy, more data is required.

Introduction

Background

- Commercial space age is here
- SpaceX has best pricing ($62 million vs. $165 million USD).Mostly due to the ability to recover a portion of the rocket (Stage 1)
- SpaceY wants to compete with SpaceX

Problems

We've been tasked by Space Y with developing a machine learning model that can predict whether or not a Stage 1 recovery would be successful.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data Collected from SpaceX public API and SpaceX Wikipedia page combined
- Perform data wrangling
  - Classifying true data landings as successful and unsuccessful otherwise
- 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 are produced, and tuned using GridSearchCV

Data Collection

The data was gathered using a combination of API requests from the SpaceX public API and web scraping data from a table in the Wikipedia entry for SpaceX.

SpaceX API Data Columns:

FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins,

Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

Wikipedia Webscrape Data Columns:

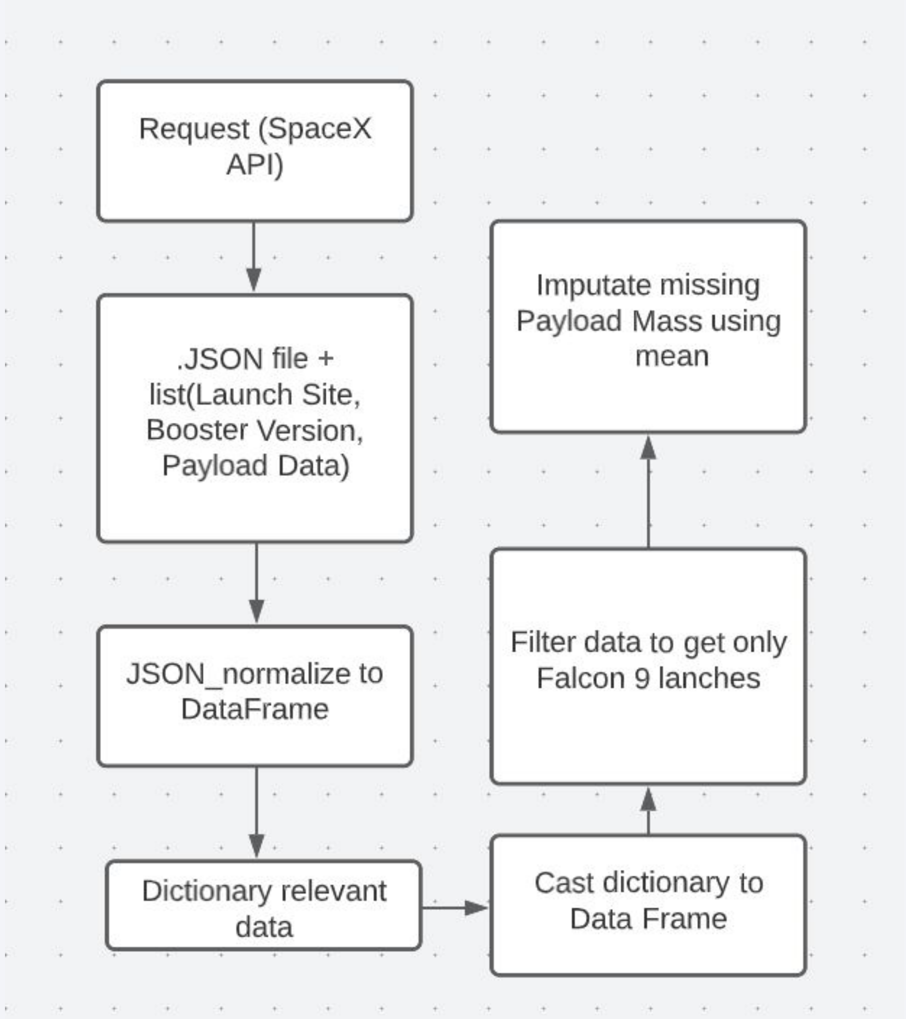
Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection – SpaceX API

The flowchart show data collection from SpaceX API.

- First, we request to SpaceX API and get JSON file data and list consisting of Launch Site, Booster Version, Payload Data
- Then, we normalize data and cast it to Data Frame
- Later, We make dictionary consisting relevant data and cast it to Data Frame
- Finally, filter data to get only falcon 9 launches and imputate missing Payload Mass using mean

Github url: https://github.com/NewbeeTryToCode/IBM-Data-Science-Capstone/blob/main/Week%201%20Introduction/SpaceY.ipynb

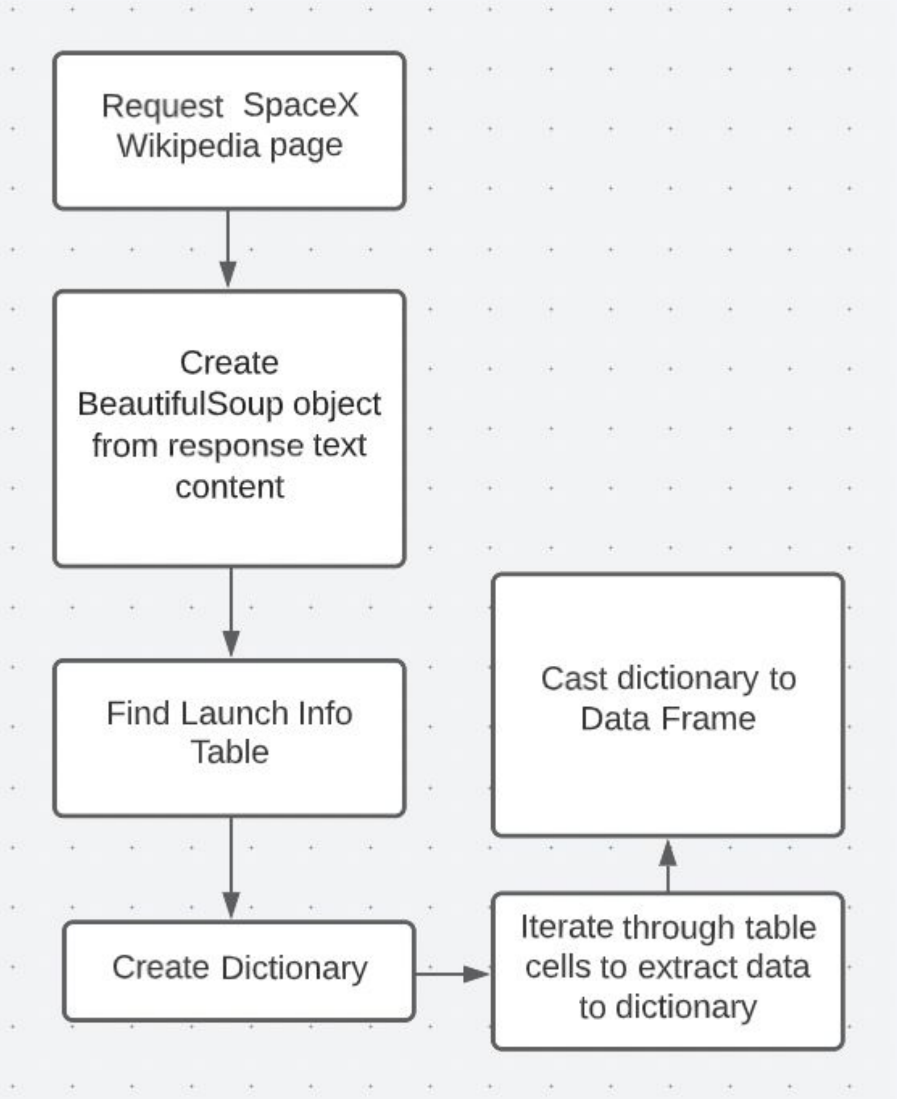

Request (SpaceX API)
.JSON file + list(Launch Site, Booster Version, Payload Data)
JSON_normalize to DataFrame
Dictionary relevant data
Cast dictionary to Data Frame
Filter data to get only Falcon 9 lanches
Imputate missing Payload Mass using mean

Data Collection - Scraping

The flowchart show data collection from Web Scraping.

- First, we request to SpaceX Wikipedia page
- Then, create BeautifulSoup object from response text content, and find launch info table
- Later, We make dictionary and iterate through table cells to extract data to dictionary
- Finally, cast dictionary to Data Frame

Github url: https://github.com/NewbeeTryToCode/IBM-Data-Science-Capstone/blob/main/Week%201%20Introduction/jupyter-labs-webscraping%20(1).ipynb


Request SpaceX Wikipedia page
Create BeautifulSoup object from response text content
Find Launch Info Table
Create Dictionary
Iterate through table cells to extract data to dictionary
Cast dictionary to Data Frame

Data Wrangling

Perform exploratory data analysis and determine training labels

Outcome column has two components: ‘Mission Outcome’ ‘Landing Location’

New training label column ‘class’ with a value of 1 if ‘Mission Outcome’ is True and 0 otherwise.

Value Mapping:

True ASDS, True RTLS, & True Ocean – set to -> 1

None None, False ASDS, None ASDS, False Ocean, False RTLS – set to -> 0

Github url:
https://github.com/NewbeeTryToCode/IBM-Data-Science-Capstone/blob/main/Week%201%20Introduction/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb

EDA with Data Visualization

Exploratory data analysis performed on Flight Number, Launch Site, Payload Mass, Orbit, Year, and Class.

Plot Used:
Flight Number vs Launch Site,Payload vs Launch Site, Success rate for each orbit type,Flight Number vs Orbit type, Payload vs Orbit type, Yearly trend of success rate

Scatter plots, line charts, and bar plots were used to compare associations between variables in order to determine if there was a relationship so that the machine learning model could be trained using them

Github url :
https://github.com/NewbeeTryToCode/IBM-Data-Science-Capstone/blob/main/Week%202%20EDA/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Loaded dataset into IBM DB2 Database
- Queried using SQL Python integration
- Queries were made to get a better understanding of the dataset
- Queried information consist launch site name, mission outcome, various payload size of customers and booster version, and landing outcome

- Github url : https://github.com/NewbeeTryToCode/IBM-Data-Science-Capstone/blob/main/Week%202%20EDA/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Launch Sites, successful and unsuccessful landings, and proximity examples to major locations: Railway, Highway, Coast, and City are all marked on Folium maps.
- This enables us to comprehend why launch sites are situated in the locations that they are. It also shows successful landings in relation to their location.

Github url :
https://github.com/NewbeeTryToCode/IBM-Data-Science-Capstone/blob/main/Week%202%20EDA/lab_jupyter_launch_site_location%20(1).ipynb

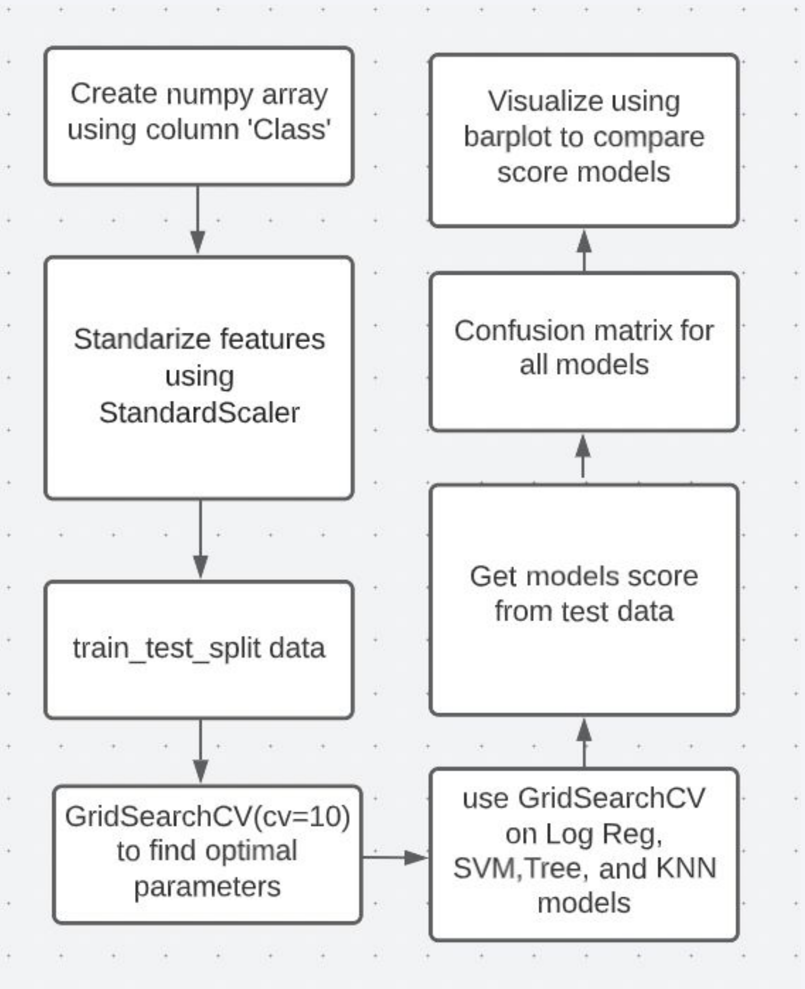
Build a Dashboard with Plotly Dash

- Dashboard includes a pie chart and a scatter plot
- Pie chart can be selected to show distribution of successful landings across all launch sites or selected individual launch site
- Scatter plot takes 2 inputs : Landing Sites and Payload Mass on a slider between 0 and 10000 kg
- The pie chart is used to visualize Launch Site success rate
- And the scatter plot can help us see how success varies across launch sites, payload mass, and booster version category

Github url :
https://github.com/NewbeeTryToCode/IBM-Data-Science-Capstone/blob/main/Week%202%20EDA/lab_jupyter_launch_site_location%20(1).ipynb

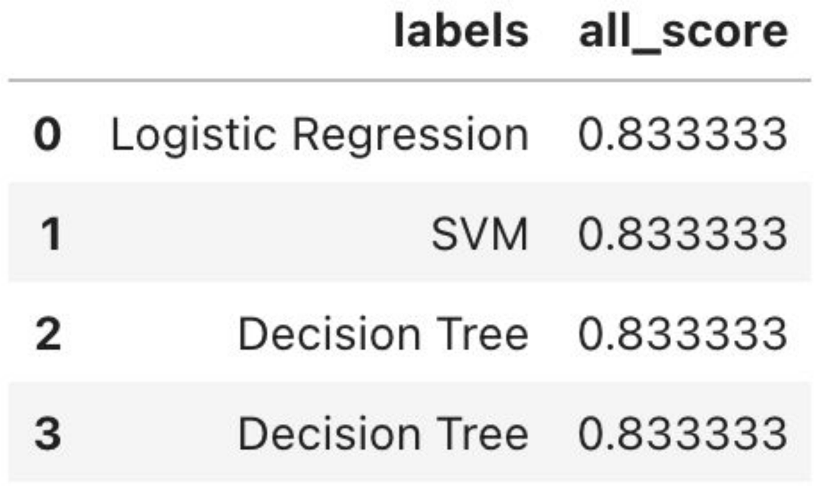
Predictive Analysis (Classification)

The flowchart show Predictive Analysis using Logistic Regression, SVM, Decision Tree, and KNN models


Create numpy array using column 'Class'
Standarize features using StandardScaler
train_test_split data
GridSearchCV(cv=10) to find optimal parameters
use GridSearchCV on Log Reg, SVM,Tree, and KNN models
Get models score from test data
Confusion matrix for all models
Visualize using barplot to compare score models

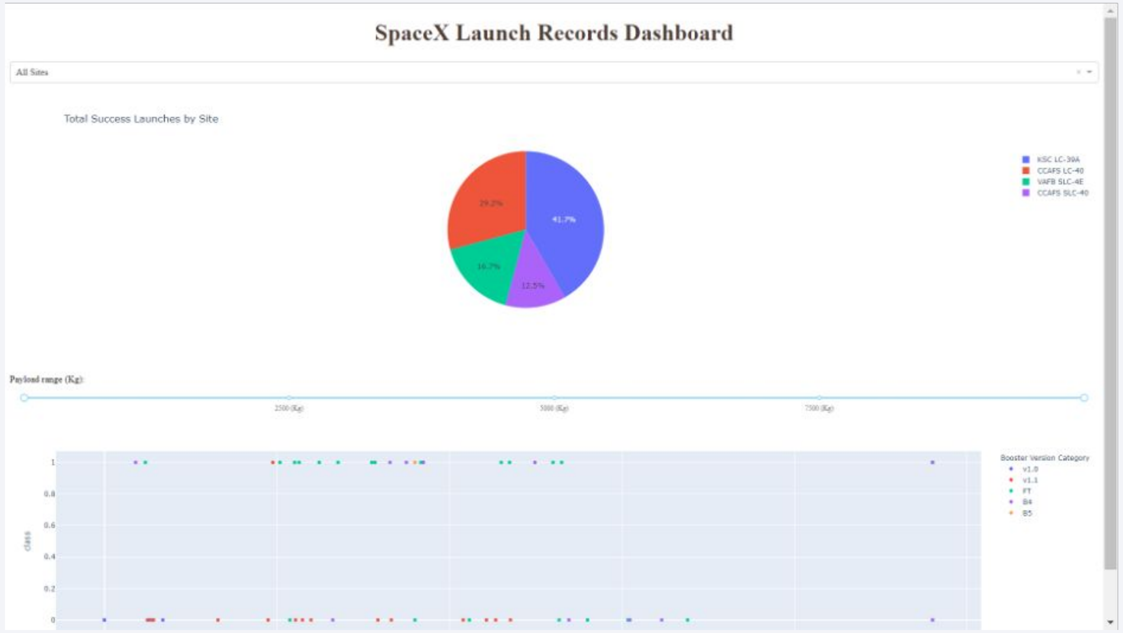
Github url:
https://github.com/NewbeeTryToCode/IBM-Data-Science-Capstone/blob/main/Week%204%20Predictive%20Analysis%20(Classification)/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results



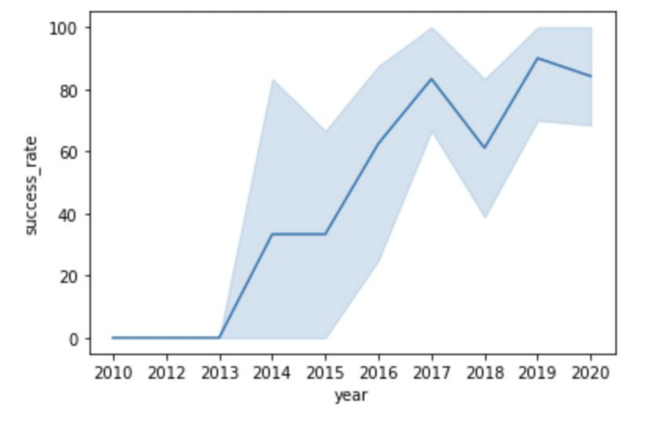
	labels	all_score
0	Logistic Regression	0.833333
1	SVM	0.833333
2	Decision Tree	0.833333
3	Decision Tree	0.833333

Predictive Analysis



SpaceX Launch Records Dashboard

Dashboard

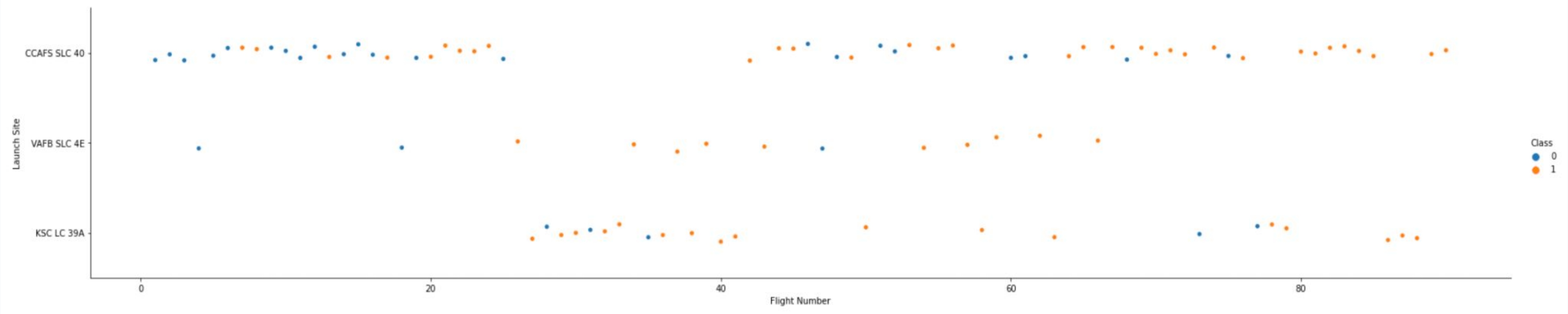


EDA Analysis

Section 2

Insights drawn from EDA

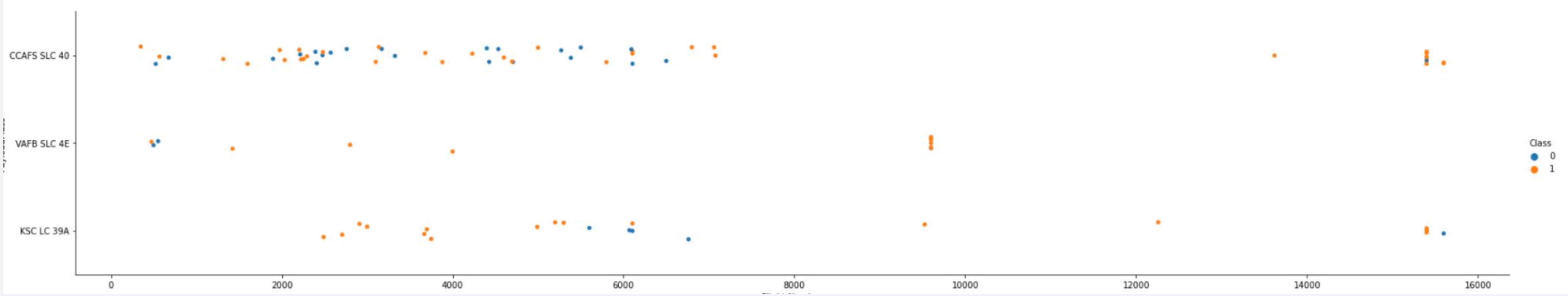
Flight Number vs. Launch Site



Orange indicates successful launch; Blue indicates unsuccessful launch.

The graph depicts a rising success rate over time (indicated in Flight Number). There was most likely a major breakthrough around flight 20 that greatly raised the success rate. Because it has the most traffic, CCAFS appears to be the primary launch point.

Payload vs. Launch Site



Orange indicates successful launch; Blue indicates unsuccessful launch.

The graph depicts a rising success rate over time (indicated in Payload). Payload mass appears fall between 0-6000 kg. Different launch site also seem use different payload mass. There are no rockets launch for VAFB SLC 4E with heavy payload

Success Rate vs. Orbit Type

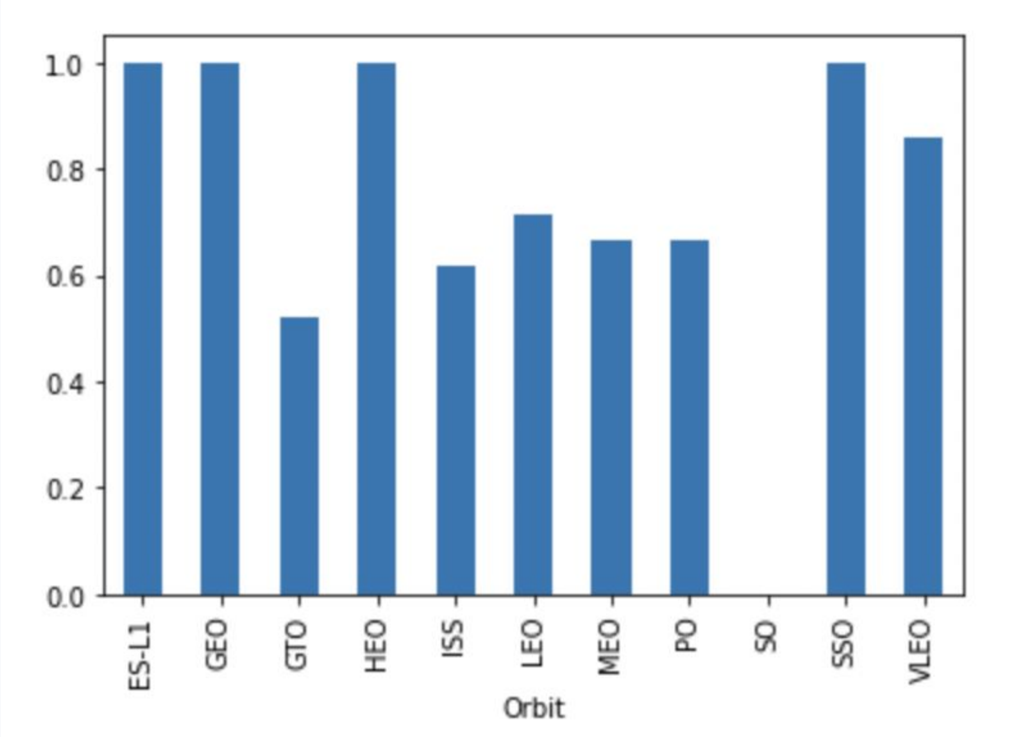
ES-L1 (1), GEO (1), HEO (1) have 100% success rate (sample sizes in parenthesis)

SSO (5) has 100% success rate

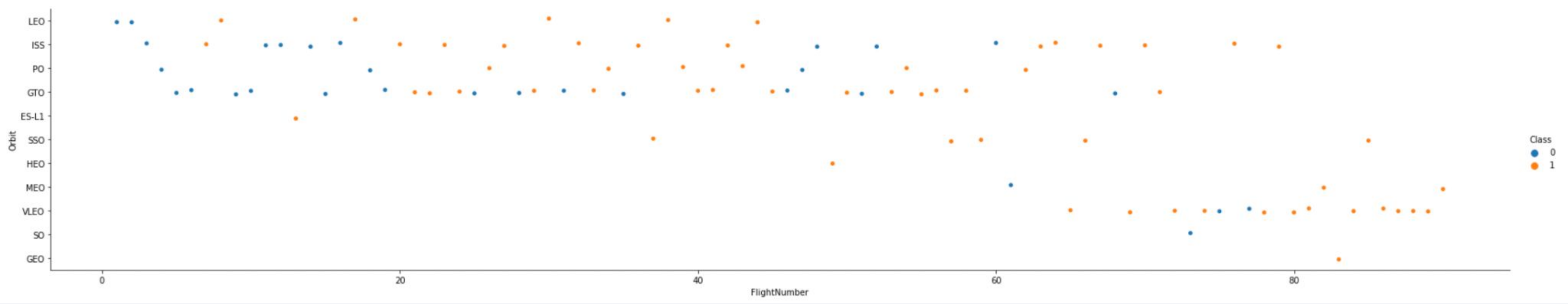
VLEO (14) has decent success rate and attempts

SO (1) has 0% success rate

GTO (27) has the around 50% success rate but largest sample



Flight Number vs. Orbit Type



Orange indicates successful launch; Blue indicates unsuccessful launch.

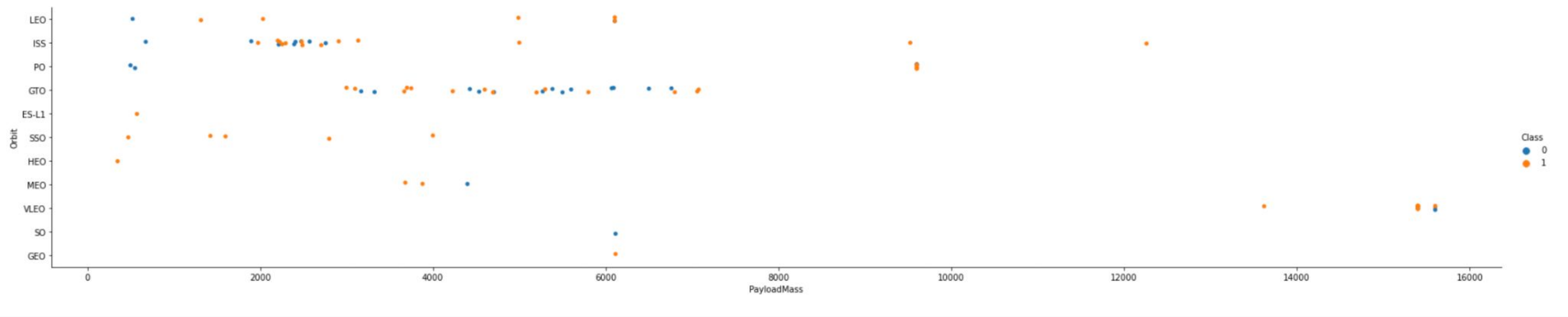
Launch Orbit preferences changed over Flight Number.

Launch Outcome seems to correlate with this preference.

SpaceX started with LEO orbits which saw moderate success LEO and returned to VLEO in recent launches

SpaceX appears to perform better in lower orbits or Sun-synchronous orbits

Payload vs. Orbit Type



Orange indicates successful launch; Blue indicates unsuccessful launch.

Payload mass seems to correlate with orbit

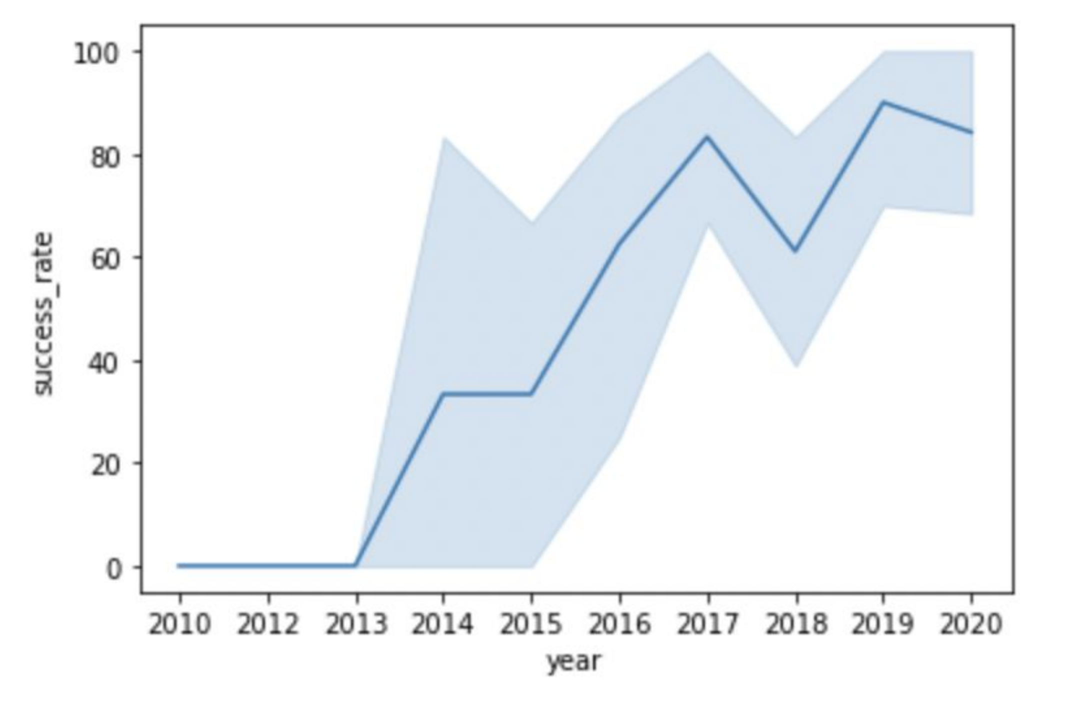
LEO and SSO seem to have relatively low payload mass

The other most successful orbit VLEO only has payload mass values in the higher end of the range

Launch Success Yearly Trend

Success generally increases over time since 2013 with a slight dip in 2018

success in recent year is around 80%

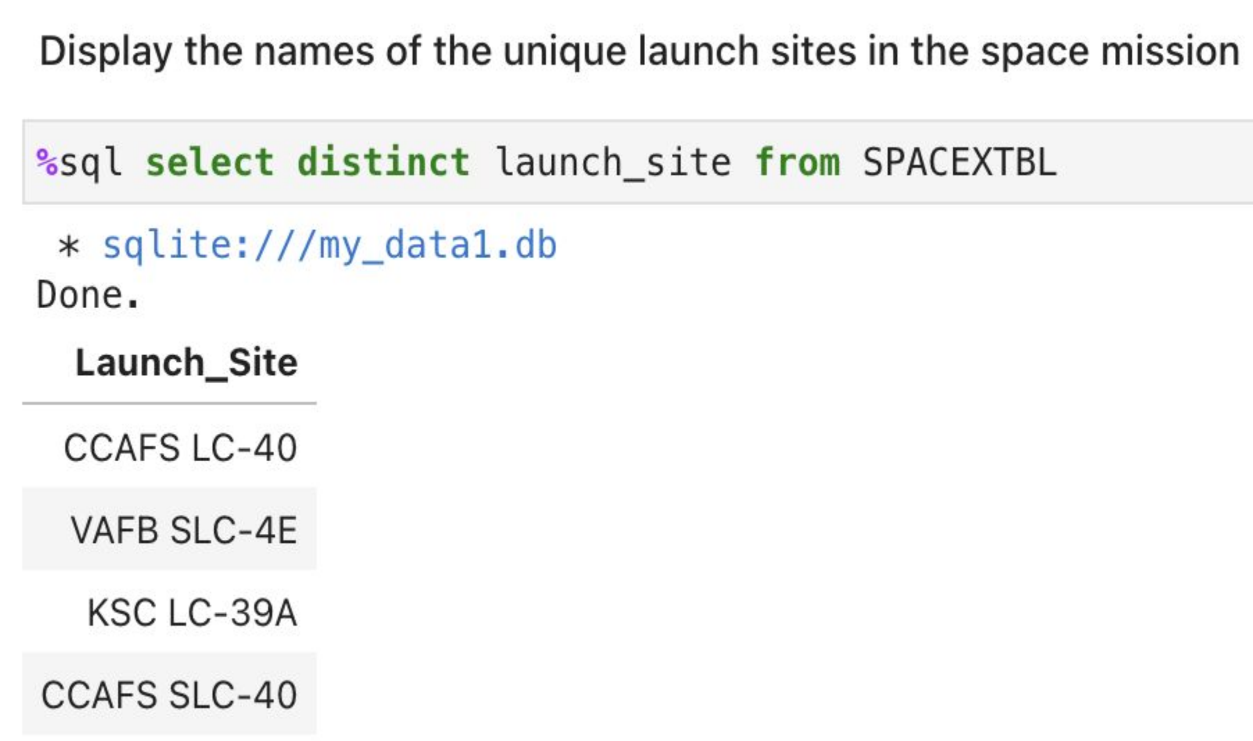


All Launch Site Names

Querying unique launch site from spaceX database

CCAFS LC-40, and CCAFS SLC-40 likely represent same launch site with data entry errors.

CCAFS LC-40 was the previous name. Likely only 3 unique launch_site values: CCAFS SLC-40, KSC LC-39A, SLC-4E



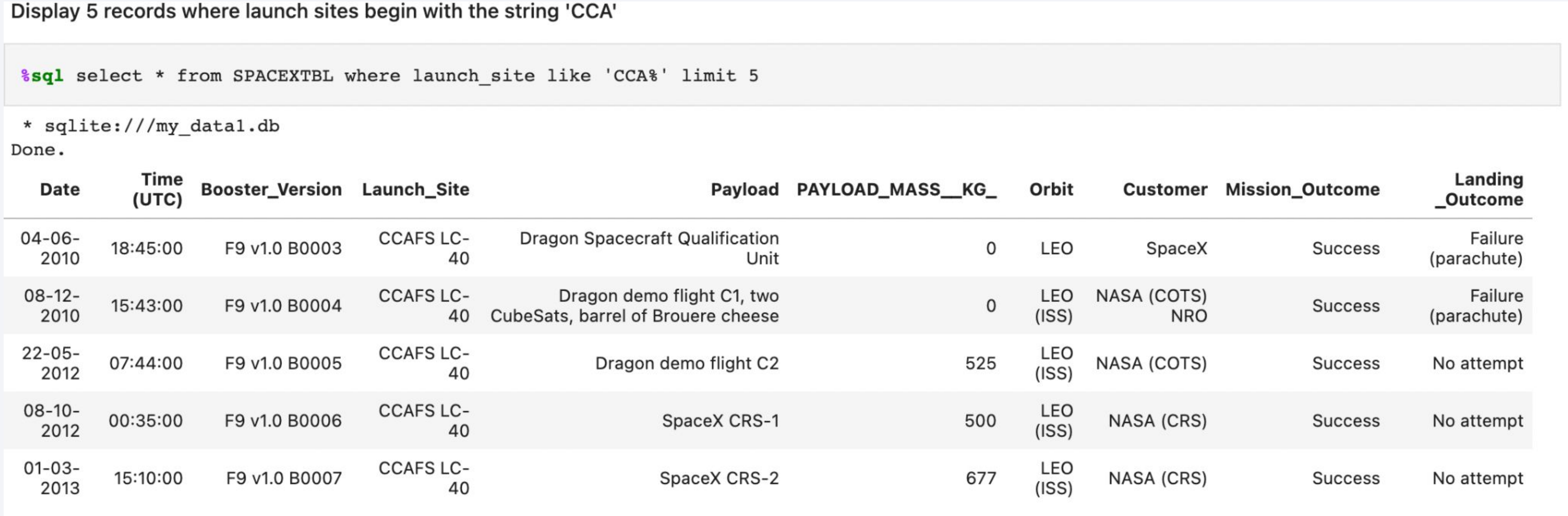
Display the names of the unique launch sites in the space mission

```
%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

Launch Site Names Begin with 'CCA'



Display 5 records where launch sites begin with the string 'CCA'

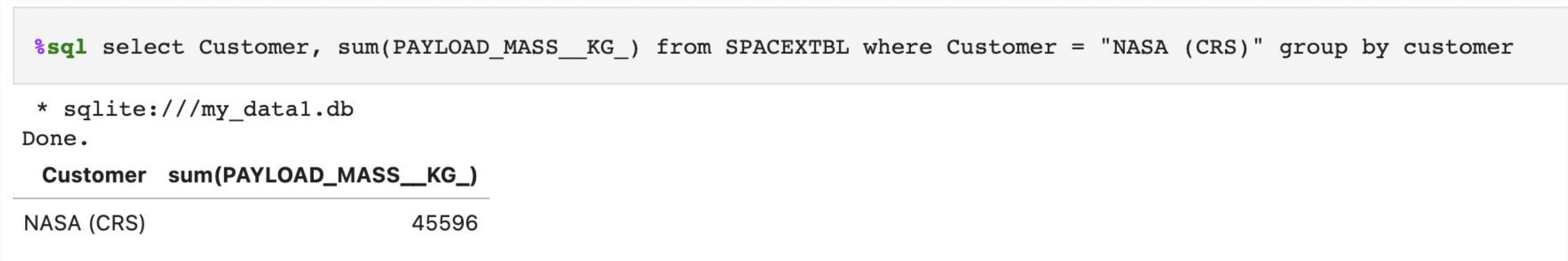
```
%sql select * from SPACEXTBL where launch_site like 'CCA%' limit 5
```

```
 * 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

First five entries with launch_site name begin with CCA

Total Payload Mass

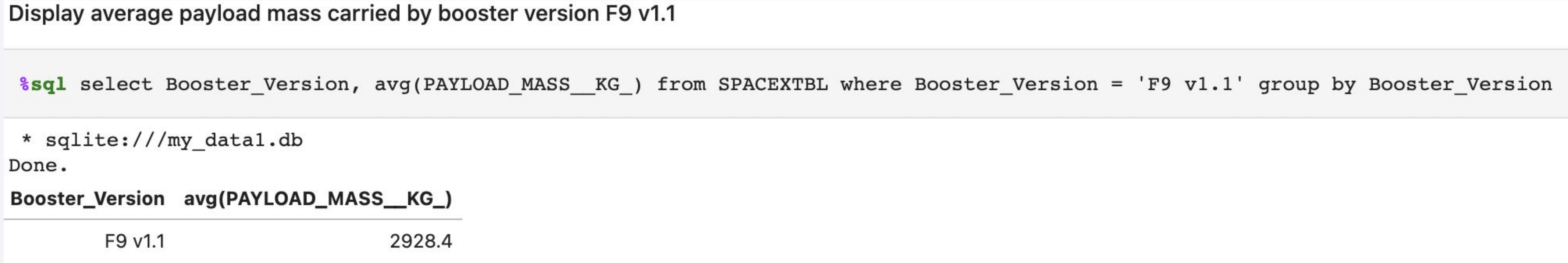


```
%sql select Customer, sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = "NASA (CRS)" group by customer
```

```
 * sqlite:///my_data1.db
Done.
 Customer    sum(PAYLOAD_MASS__KG_)
NASA (CRS)                    45596
```

This query sums the total payload mass in kg where NASA was the customer.
CRS stands for Commercial Resupply Services which indicates that these payloads were sent to the International Space Station (ISS).

Average Payload Mass by F9 v1.1



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

```
%sql select Booster_Version, avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1' group by Booster_Version
```

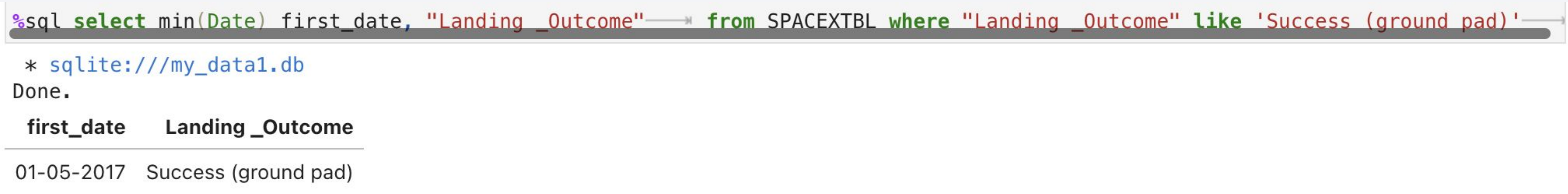
```
 * sqlite:///my_data1.db
Done.
```

Booster_Version	avg(PAYLOAD_MASS__KG_)
F9 v1.1	2928.4

This query calculates the average payload mass or launches which used booster version F9 v1.1

Average payload mass of F9 1.1 is on the low end of our payload mass range

First Successful Ground Landing Date

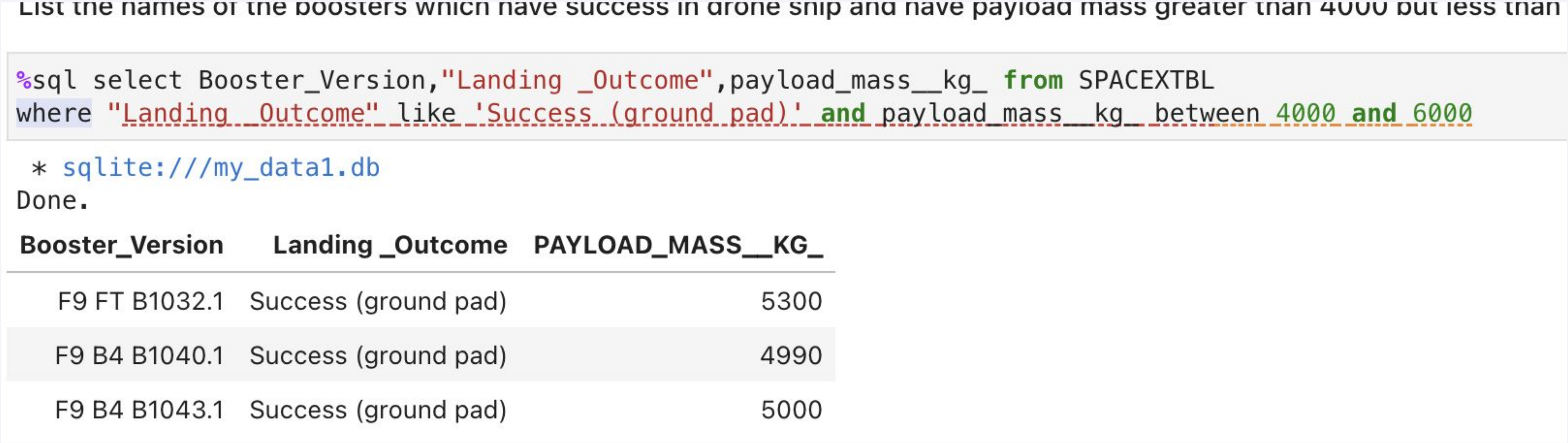


```
%sql select min(Date) first_date, "Landing _Outcome"    from SPACEXTBL where "Landing _Outcome" like 'Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
 first_date    Landing _Outcome
01-05-2017   Success (ground pad)
```

This query returns the first successful ground pad landing date.
First ground pad landing wasn’t until mid of 2017.Successful landings in general appear in 2017.

Successful Drone Ship Landing with Payload between 4000 and 6000



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

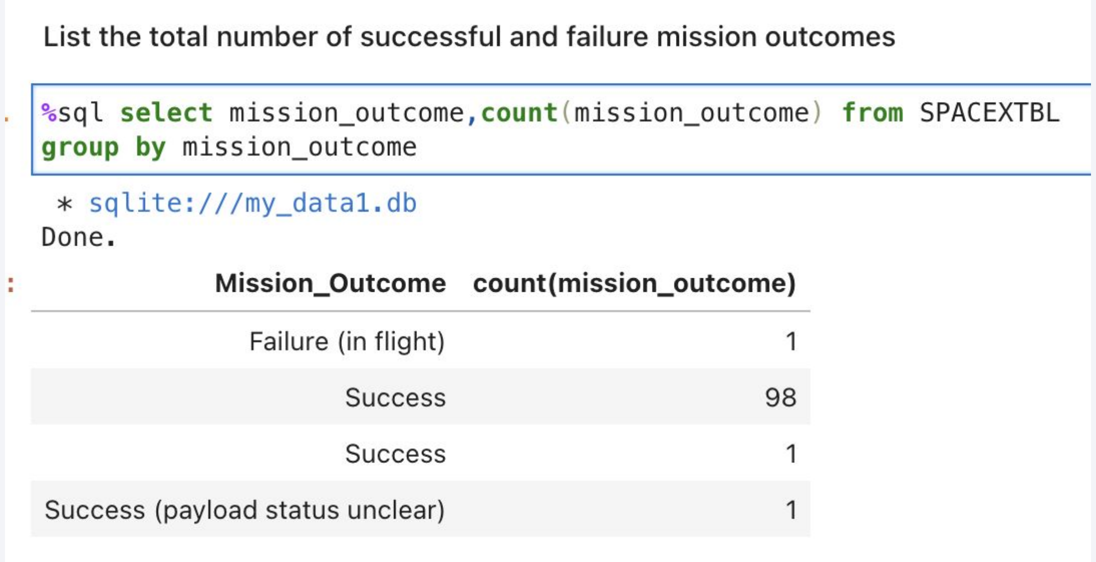
```
%sql select Booster_Version,"Landing _Outcome",payload_mass__kg_ from SPACEXTBL
where "Landing _Outcome" like 'Success (ground pad)' and payload_mass__kg_ between 4000 and 6000
```

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

Booster_Version	Landing _Outcome	PAYLOAD_MASS__KG_
F9 FT B1032.1	Success (ground pad)	5300
F9 B4 B1040.1	Success (ground pad)	4990
F9 B4 B1043.1	Success (ground pad)	5000

This query returns the four booster versions that had successful drone ship landings and a payload mass between 4000 and 6000 non inclusively.

Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

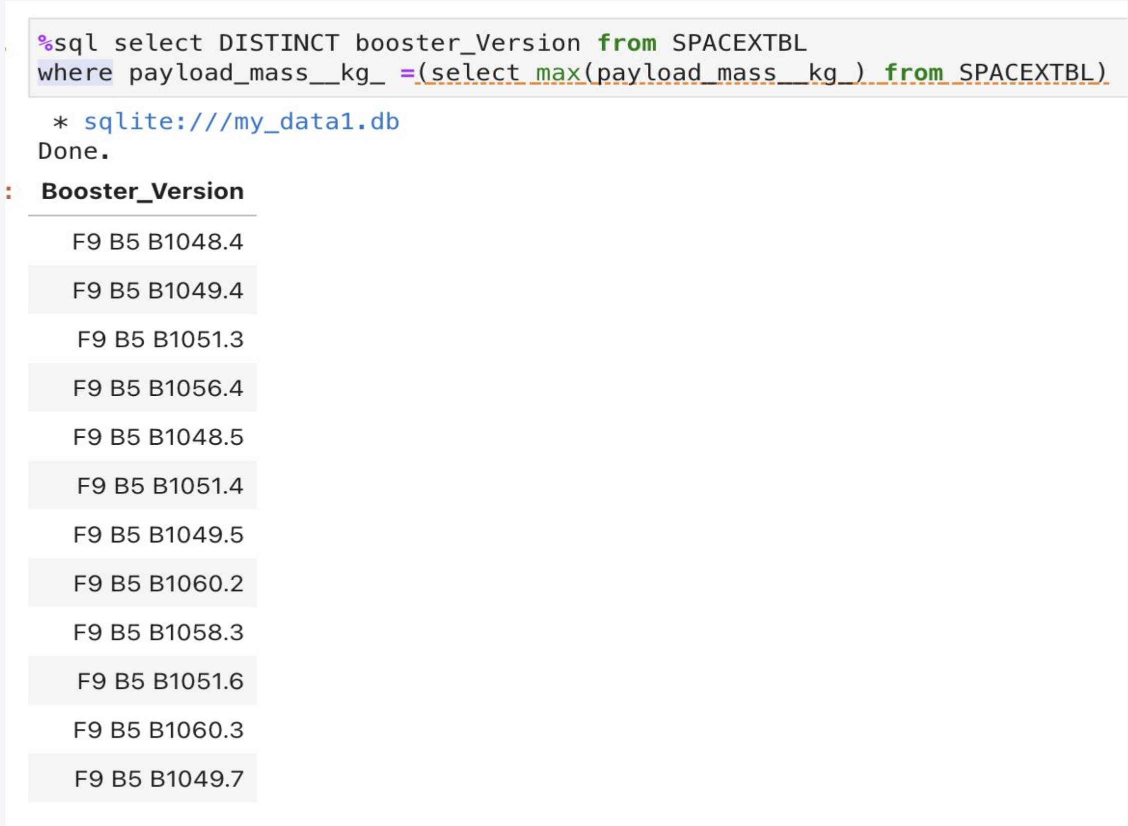
```
%sql select mission_outcome,count(mission_outcome) from SPACEXTBL
group by mission_outcome
```

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

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

This query returns a count of each mission outcome.
SpaceX appears to achieve its mission outcome nearly 99% of the time.
This means that most of the landing failures are intended.
Interestingly, one launch has an unclear payload status and unfortunately one failed in flight.

Boosters Carried Maximum Payload



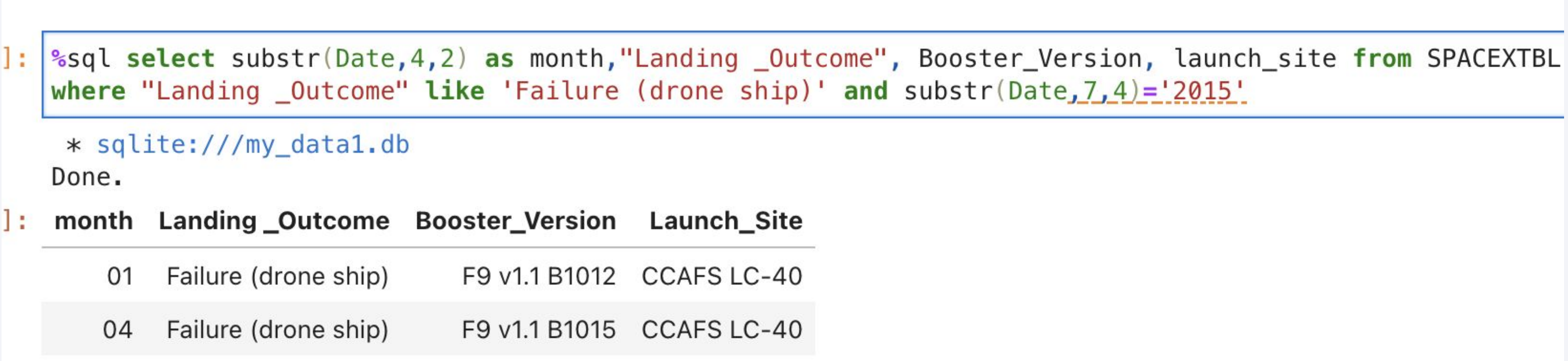
```
%sql select DISTINCT booster_Version from SPACEXTBL
where payload_mass__kg_ =(select max(payload_mass__kg_) from SPACEXTBL)
```

```
 * 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 returns the booster versions that carried the highest payload mass of 15600 kg.
These booster versions are very similar and all are of the F9 B5 B10xx.x variety.
This likely indicates payload mass correlates with the booster version that is used.

2015 Launch Records



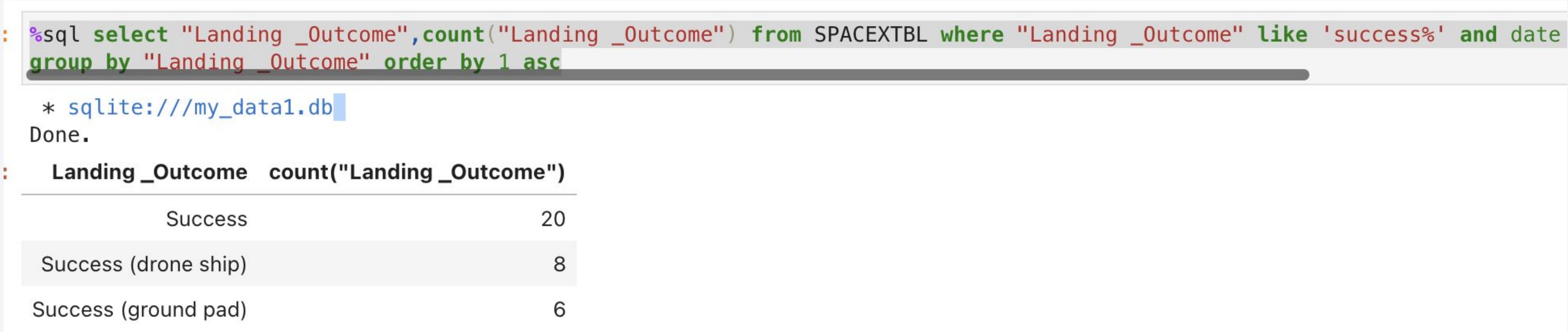
```
]: %sql select substr(Date,4,2) as month,"Landing _Outcome", Booster_Version, launch_site from SPACEXTBL
   where "Landing _Outcome" like 'Failure (drone ship)' and substr(Date,7,4)='2015'

    * 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

This query returns the Month, Landing Outcome, Booster Version, Payload Mass (kg), and Launch site of 2015 launches where stage 1 failed to land on a drone ship. There were two such occurrences.

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



```
%sql select "Landing _Outcome",count("Landing _Outcome") from SPACEXTBL where "Landing _Outcome" like 'success%' and date
group by "Landing _Outcome" order by 1 asc
```

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

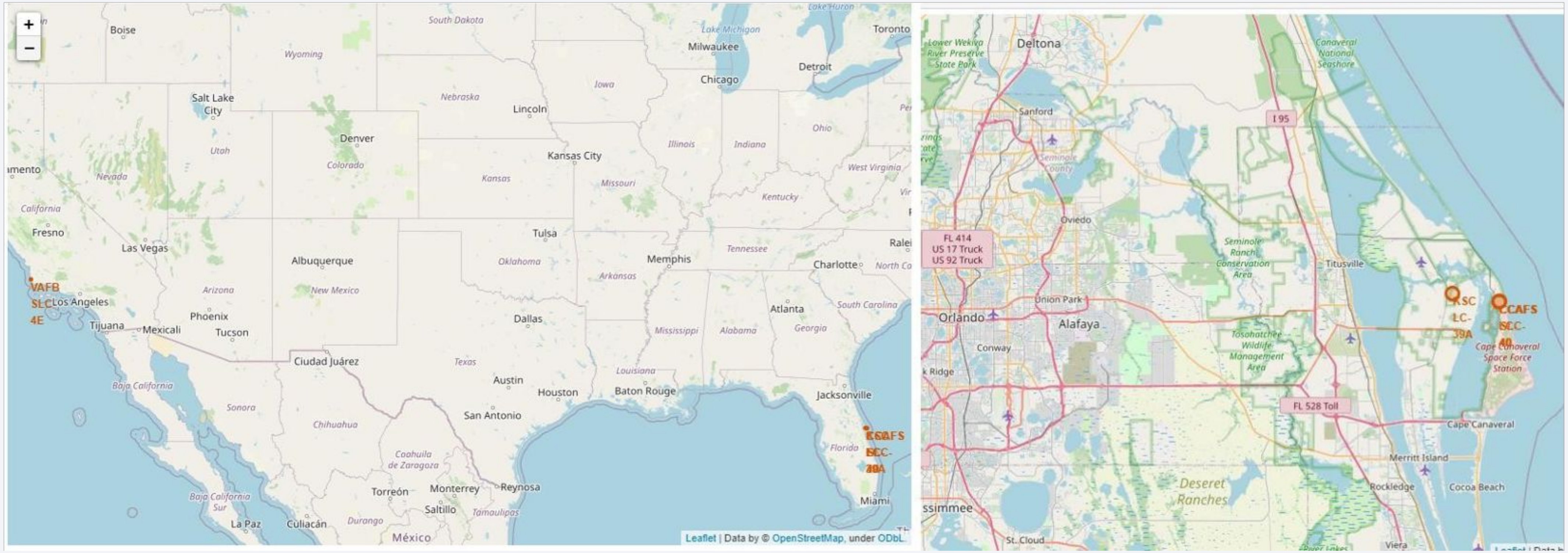
Landing _Outcome	count("Landing _Outcome")
Success	20
Success (drone ship)	8
Success (ground pad)	6

This query returns a list of successful landings and between 2010-06-04 and 2017-03-20 inclusively.
There are three types of successful landing outcomes: success (without landing description),drone ship and ground pad landings.
There were 34 successful landings in total during this time period

Section 3

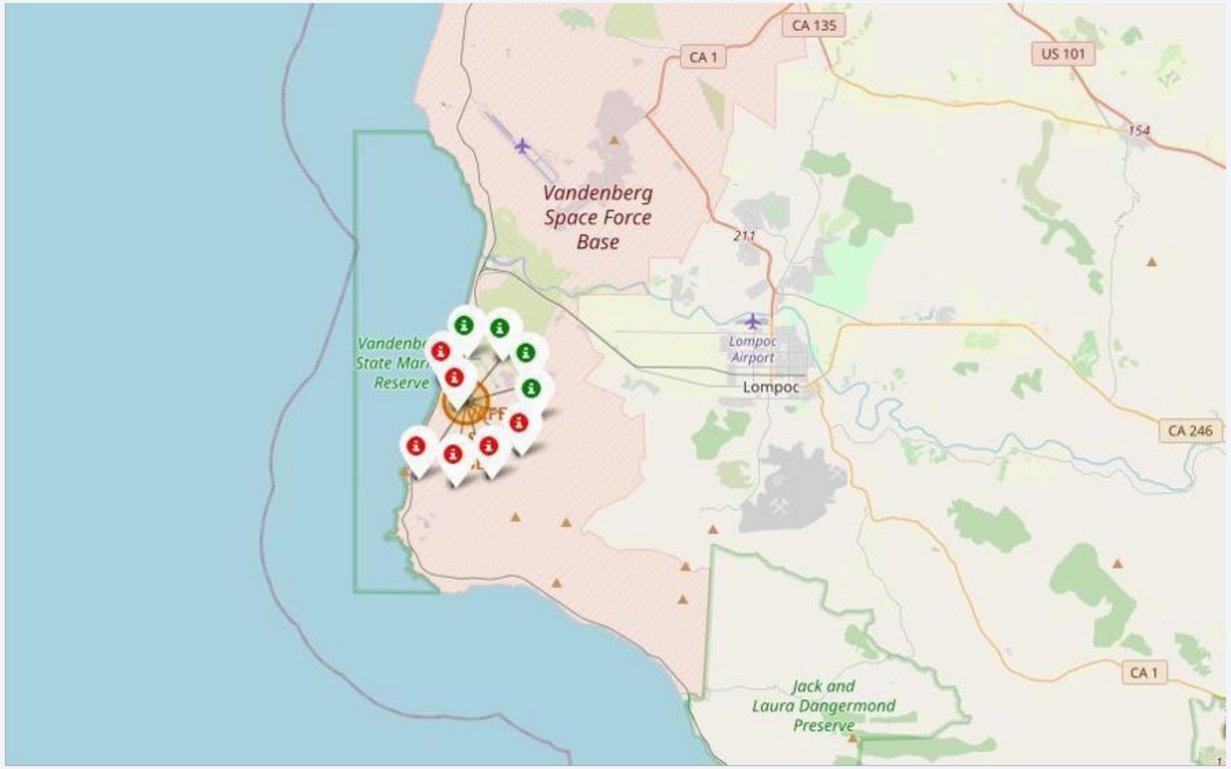
Launch Sites Proximities Analysis

Launch Site Locations



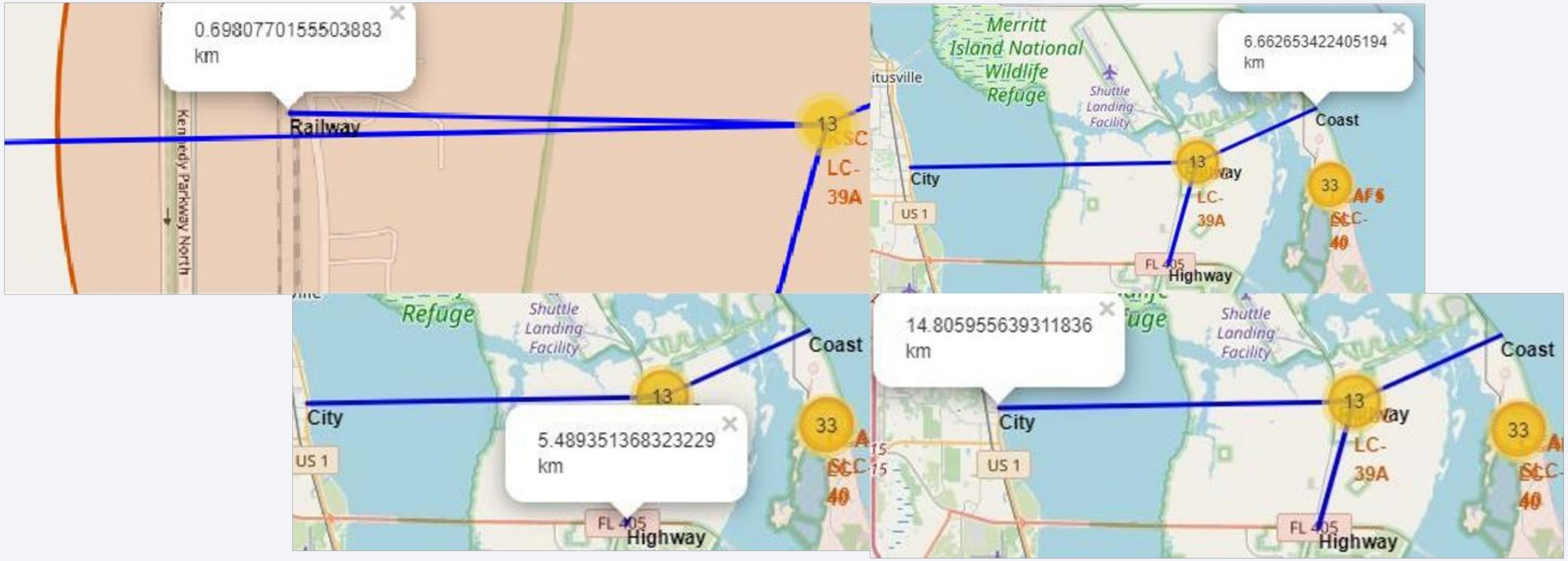
The left map shows all launch sites relative US map. The right map shows the two Florida launch sites since they are very close to each other. All launch sites are near the ocean.

Color-Coded Launch Markers

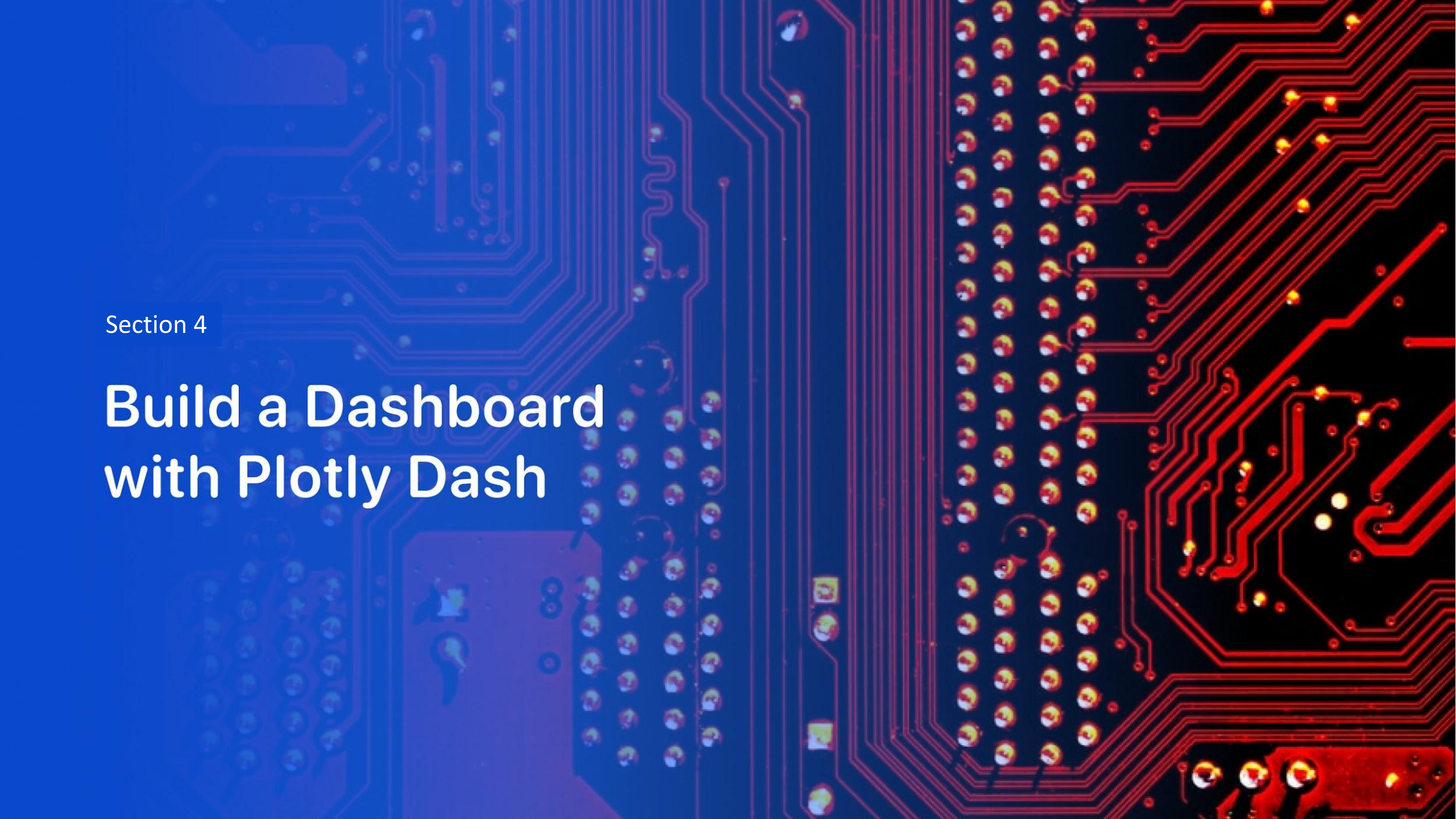

CA 135
CA 1
US 101
154
Vandenberg Space Force Base
211
Lompoc Airport
Lompoc
CA 246
Vandenberg State Marine Reserve
CA 1
Jack and Laura Dangermond Preserve

Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed landing (red icon). In this example VAFB SLC-4E shows 4 successful landings and 6 failed landings.

Key Location Proximities



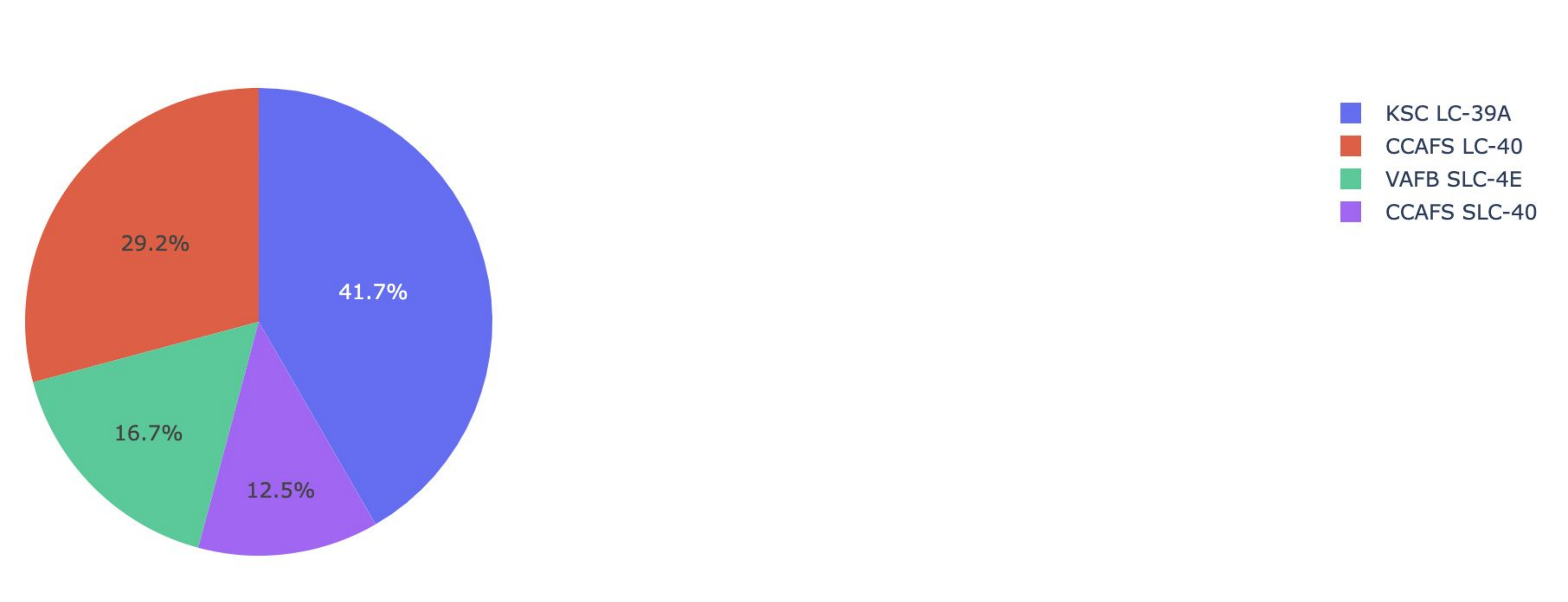
Using KSC LC-39A as an example, launch sites are very close to railways for large part and supply transportation. Launch sites are close to highways for human and supply transport. Launch sites are also close to coasts and relatively far from cities so that launch failures can land in the sea to avoid rockets falling on densely populated areas.



Section 4

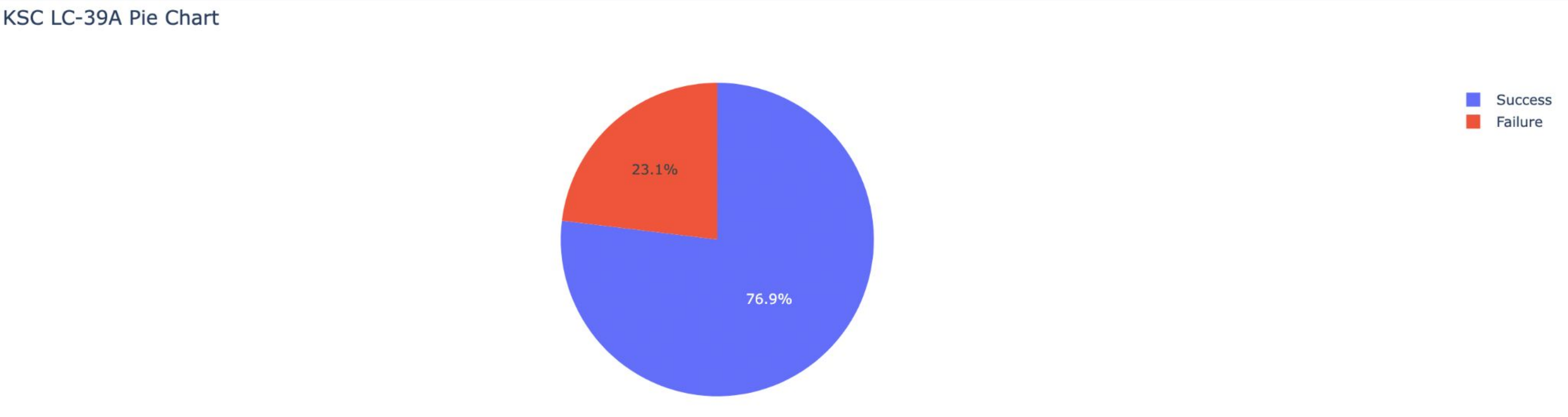
Build a Dashboard with Plotly Dash

Successful Launches across Launch Sites



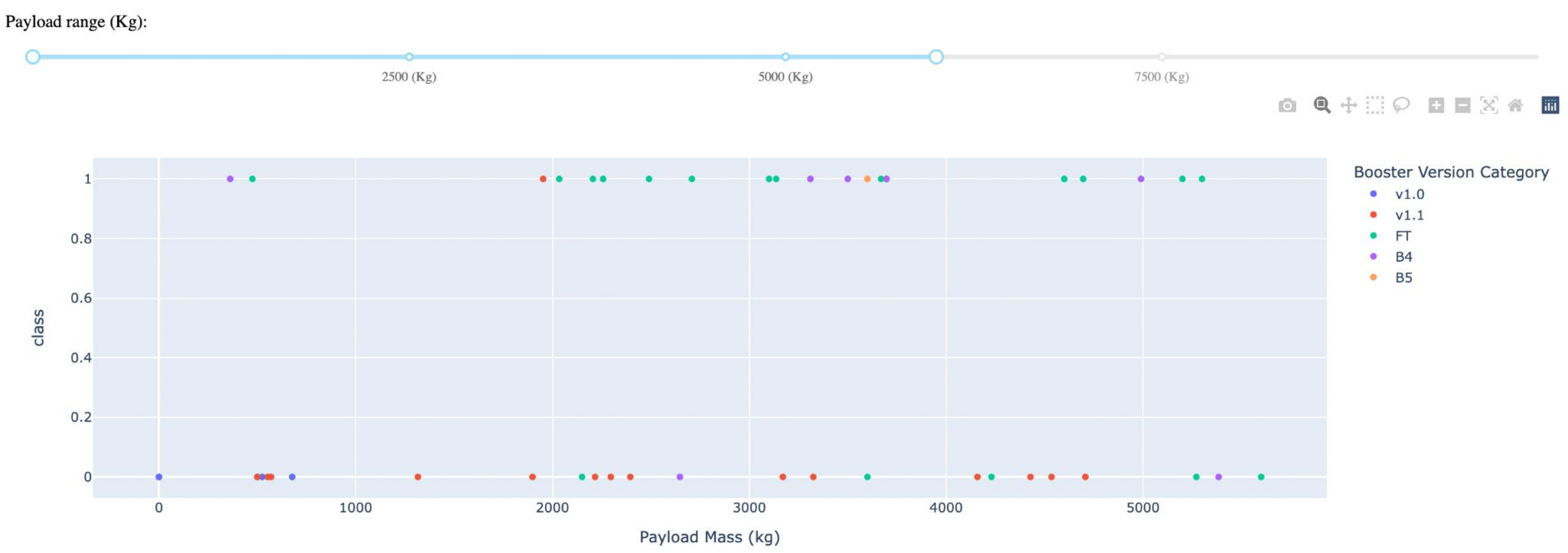
This is the distribution of successful landings across all launch sites. CCAFS LC-40 is the old name of CCAFS SLC-40 so CCAFS and KSC have the same amount of successful landings, but a majority of the successful landings performed before the name change. VAFB has the smallest share of successful landings. This may be due to smaller sample and increase in difficulty of launching in the west coast.

Highest Success Rate Launch Site



KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.

Payload Mass vs Success vs Booster Version Category



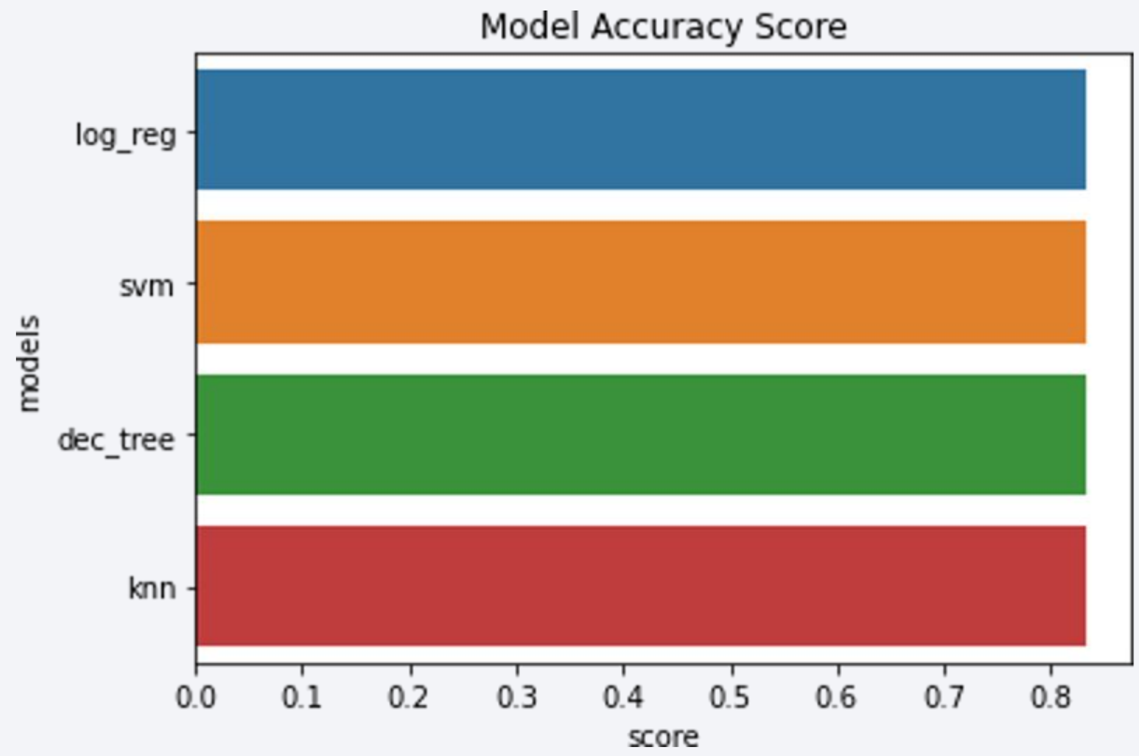
Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600. Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size. In this particular range of 0-6000, interestingly there are two failed landings with payloads of zero kg.

Section 5

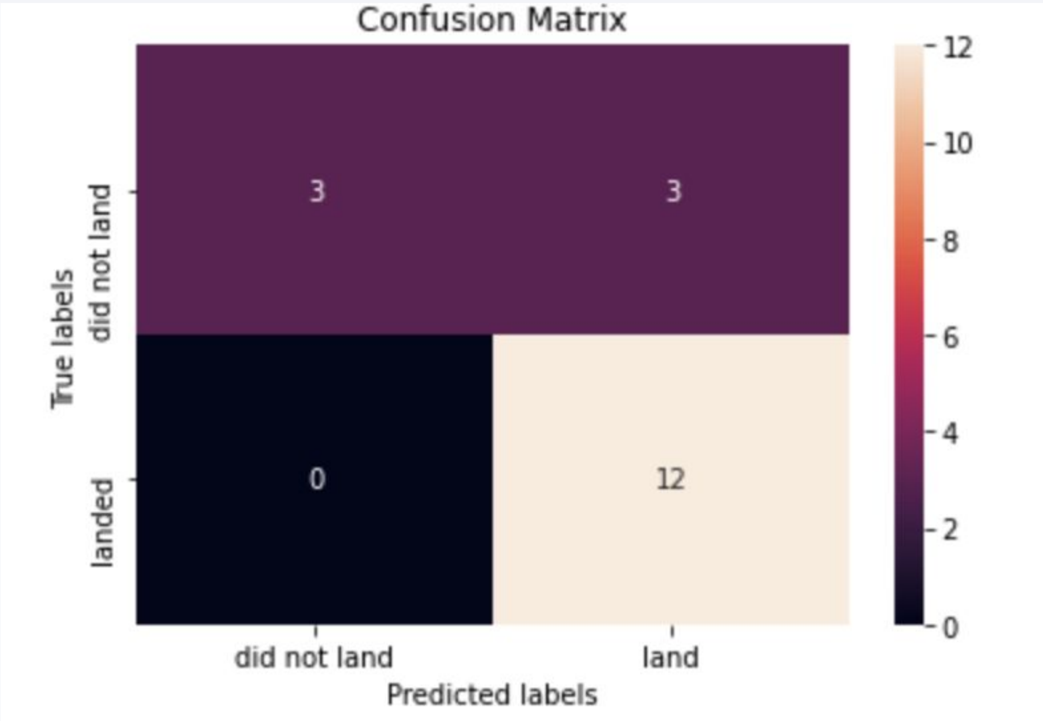
Predictive Analysis (Classification)

Classification Accuracy

All models had virtually the same accuracy on the test set at 83.33% accuracy. It should be noted that test size is small at only sample size of 18. This can cause large variance in accuracy results, such as those in Decision Tree Classifier model in repeated runs. We likely need more data to determine the best model.



Confusion Matrix



Since all models performed the same for the test set, the confusion matrix is the same across all models. The models predicted 12 successful landings when the true label was successful landing.
The models predicted 3 unsuccessful landings when the true label was unsuccessful landing. The models predicted 3 successful landings when the true label was unsuccessful landings (false positives). Our models over predict successful landings.

Conclusions

- Our task: to develop a machine learning model for Space Y who wants to bid against SpaceX
- The goal of model is to predict when Stage 1 will successfully land to save ~$100 million USD
- Used data from a public SpaceX API and web scraping SpaceX Wikipedia page
- Created data labels and stored data into a DB2 SQL database
- Created a dashboard for visualization
- We created a machine learning model with an accuracy of 83%
- Allon Mask of SpaceY can use this model to predict with relatively high accuracy whether a launch will have a successful Stage 1 landing before launch to determine whether the launch should be made or not
- If possible more data should be collected to better determine the best machine learning model and improve accuracy

Appendix

GitHub repository url:

https://github.com/NewbeeTryToCode/IBM-Data-Science-Capstone

Instructors:

Instructors: Rav Ahuja, Alex Aklson, Aije Egwaikhide, Svetlana Levitan, Romeo Kienzler, Polong Lin, Joseph Santarcangelo, Azim Hirjani, Hima Vasudevan, Saishruthi Swaminathan, Saeed Aghabozorgi, Yan Luo

Special Thanks to All Instructors:

https://www.coursera.org/professional-certificates/ibm-data-science?#instructors

Thank you!