IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tianjiao Yang
04/18/2023

Outline

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

Executive Summary

- Data was first collected through API and web scraping, and then analyzed through exploratory analysis using SQL and data visualization. Interactive folium map and dash were used to visualize the data frame. Multiple prediction model were discussed to select the best one to predict outcome.
- In the result, we got bar charts, pie charts, scatter plots, interactive maps, and plotly dash to summarize the data and machine learning algorithm to predict the outcome

Introduction

- Project background

  SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems to be solved
  - What parameters are included to predict outcome?
  - What kind of data cleaning are used to preprocess the data?
  - Which model is suitable for prediction launch result?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected through SpaceX API and Web Scraping
- Perform data wrangling
  - Data was processed through exploratory data analysis
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistic regression, super vector machine, decision tree and k nearest neighbor models are discussed

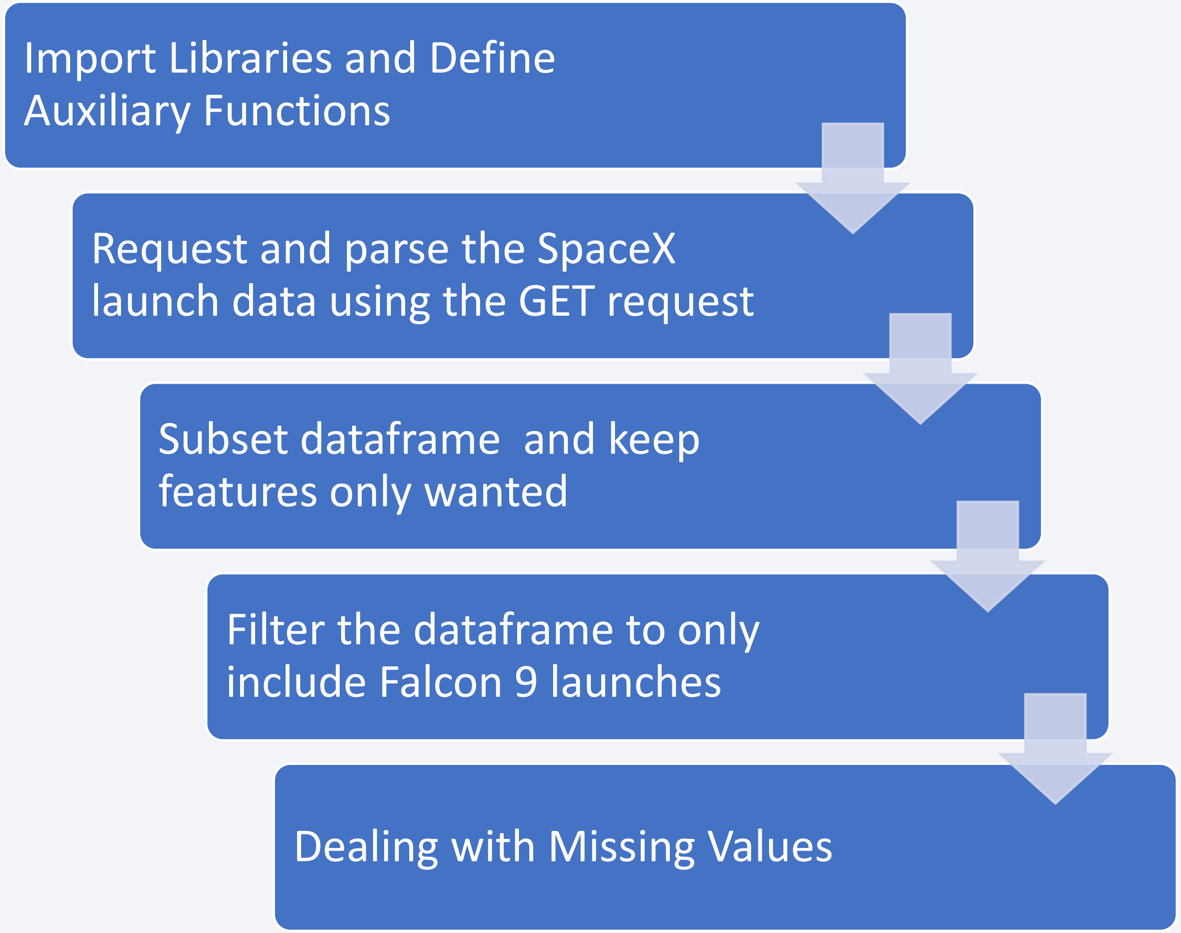
Data Collection

- Data was collected through tow methods: SpaceX API and Web Scraping

- SpaceX API:
    - request rocket launch data from SpaceX API through URL
    - decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()
    - get information about the launches using the IDs given for each launch using columns rocket, payloads, launchpad, and cores
    - Filter the dataframe to only include Falcon 9 launches and replace missing values

- Web Scraping:
    - perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response
    - Create a BeautifulSoup object from the HTML response
    - collect all relevant column names from the HTML table header
    - create an empty dictionary with keys from the extracted column names in the previous task
    - fill up the launch_dict with launch records extracted from table rows and create a dataframe from it

Data Collection – SpaceX API



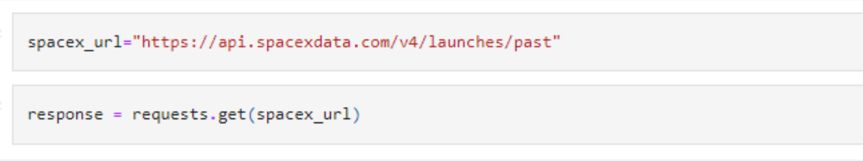
Import Libraries and Define Auxiliary Functions

Request and parse the SpaceX launch data using the GET request

Subset dataframe and keep features only wanted

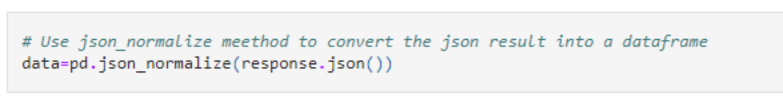
Filter the dataframe to only include Falcon 9 launches

Dealing with Missing Values

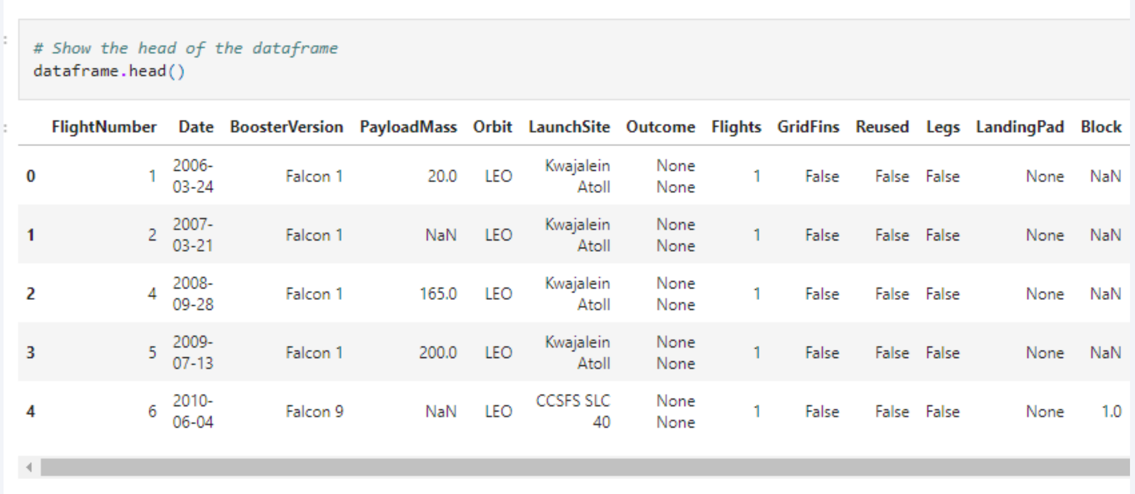


```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

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

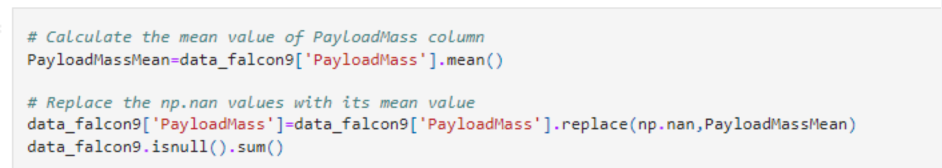


```
# Use json_normalize meethod to convert the json result into a dataframe
data=pd.json_normalize(response.json())
```



```
# Show the head of the dataframe
dataframe.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0

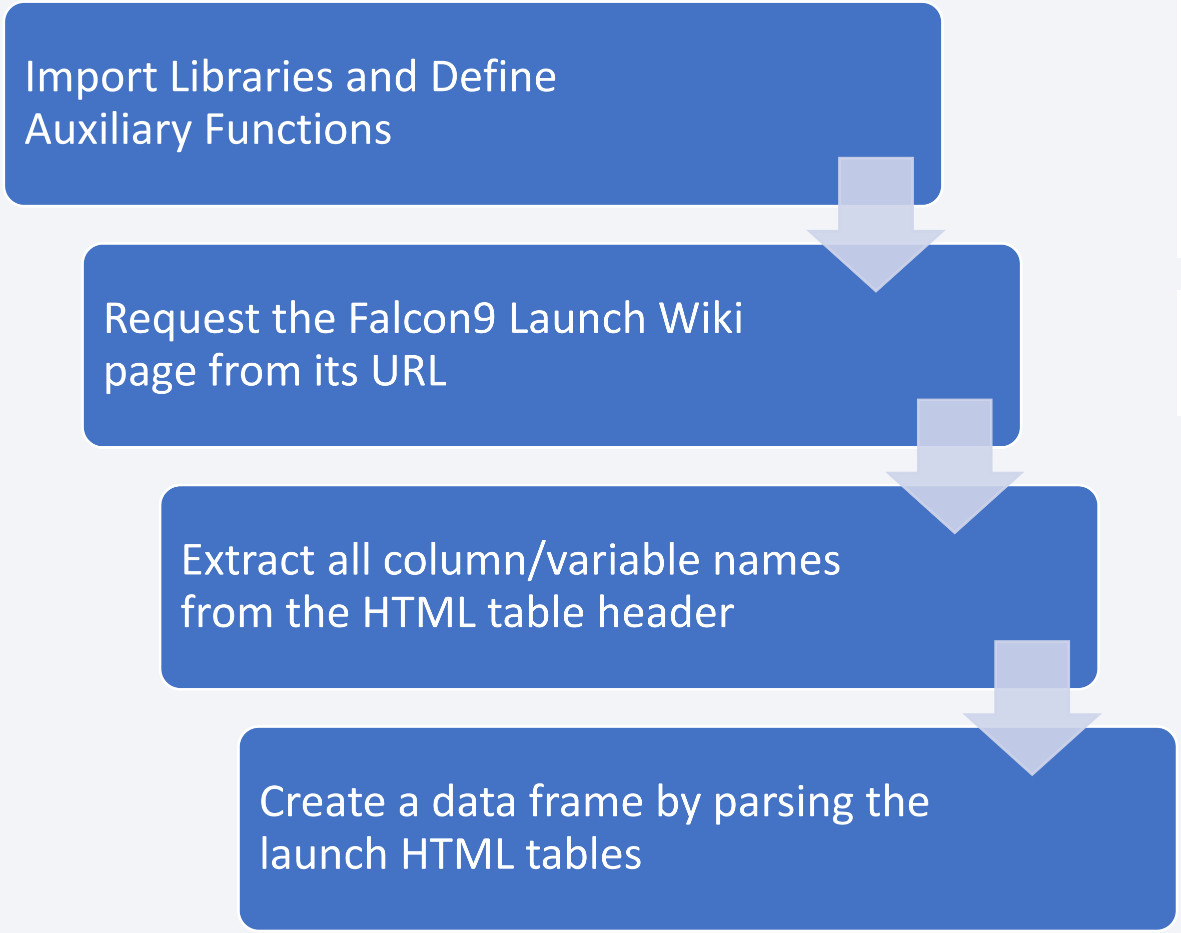


```
# Calculate the mean value of PayloadMass column
PayloadMassMean=data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass']=data_falcon9['PayloadMass'].replace(np.nan,PayloadMassMean)
data_falcon9.isnull().sum()
```

Reference:https://github.com/SophiaQY/IBM-Data-Science/blob/main/Data%20collection%20lab.ipynb

Data Collection - Scraping

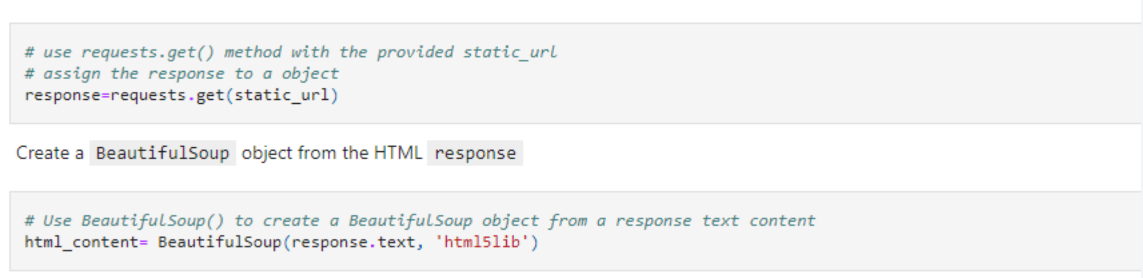


Import Libraries and Define Auxiliary Functions

Request the Falcon9 Launch Wiki page from its URL

Extract all column/variable names from the HTML table header

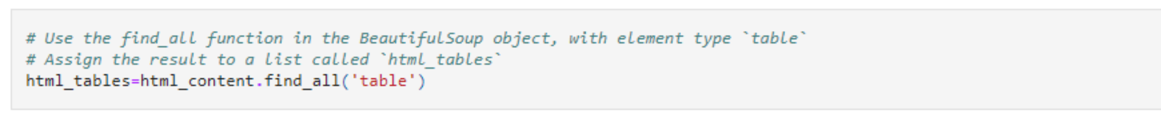
Create a data frame by parsing the launch HTML tables



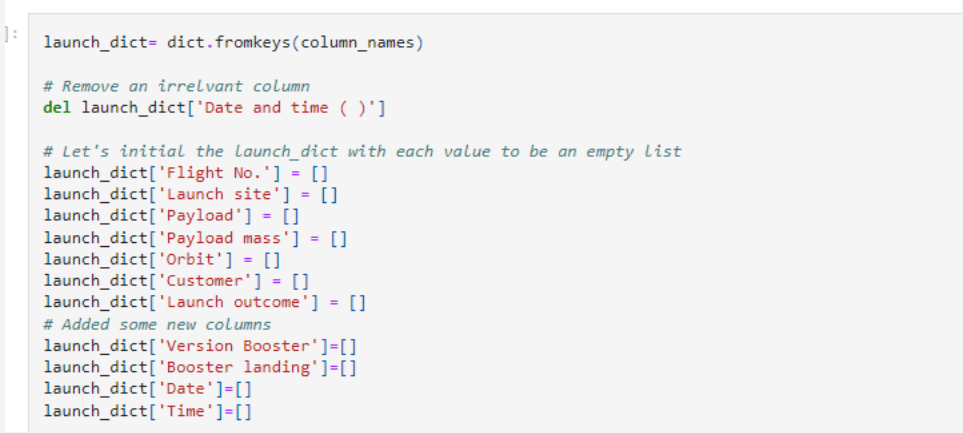
```
# use requests.get() method with the provided static_url
# assign the response to a object
response=requests.get(static_url)
```

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
html_content= BeautifulSoup(response.text, 'html5lib')
```



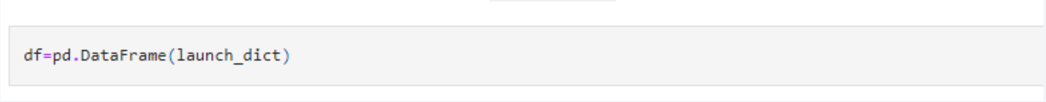
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables=html_content.find_all('table')
```



```
launch_dict= dict.fromkeys(column_names)

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

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



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

Reference:https://github.com/SophiaQY/IBM-Data-Science/blob/main/Data%20collection%20with%20web%20scraping.ipynb

Data Wrangling

Import Libraries:

- Pandas and Numpy

Data analysis

- Discover missing value percentage and column type
- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurence of mission outcome per orbit type
- Create a landing outcome label from Outcome column

Reference:https://github.com/SophiaQY/IBM-Data-Science/blob/main/Data%20wrangling.ipynb

EDA with Data Visualization

Charts plot in the project:

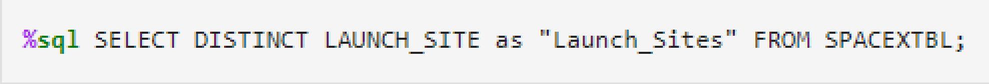
- FlightNumber vs. PayloadMass and overlay the outcome of the launch
- Visualize the relationship between Flight Number and Launch Site
- Visualize the relationship between Payload and Launch Site
- Visualize the relationship between success rate of each orbit type
- Visualize the relationship between FlightNumber and Orbit type
- Visualize the relationship between Payload and Orbit type
- Visualize the launch success yearly trend

Reference:https://github.com/SophiaQY/IBM-Data-Science/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite_submit.ipynb

EDA with SQL

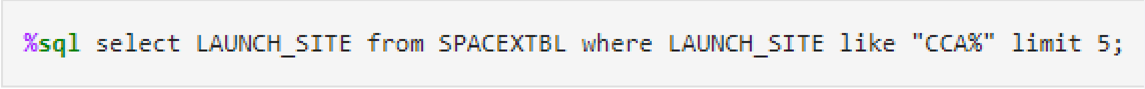
Queries performed:

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



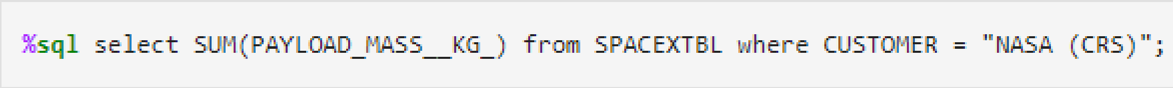
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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



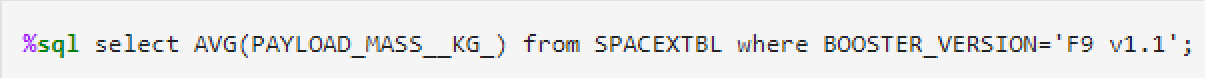
```
%sql select LAUNCH_SITE from SPACEXTBL where LAUNCH_SITE like "CCA%" limit 5;
```

- Display the total payload mass carried by boosters launched by NASA (CRS)



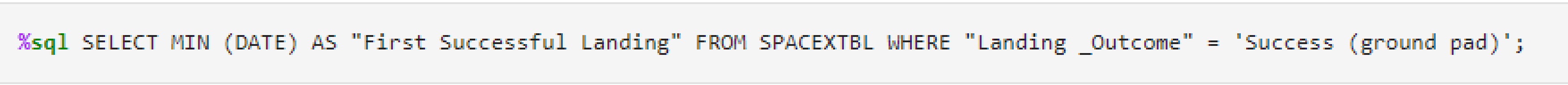
```
%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = "NASA (CRS)";
```

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



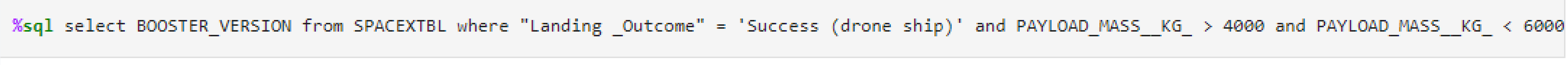
```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION='F9 v1.1';
```

- List the date when the first successful landing outcome in ground pad was achieved



```
%sql SELECT MIN (DATE) AS "First Successful Landing" FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (ground pad)';
```

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

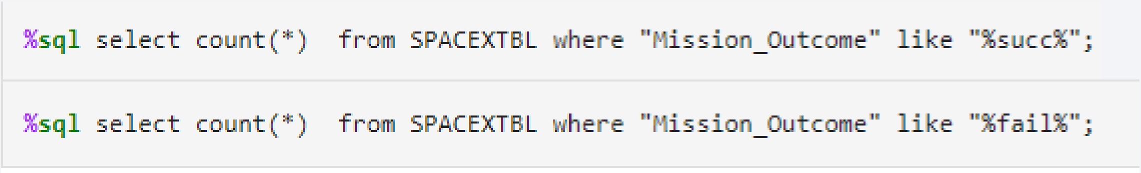


```
%sql select BOOSTER_VERSION from SPACEXTBL where "Landing _Outcome" = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

EDA with SQL

Queries performed(continued):

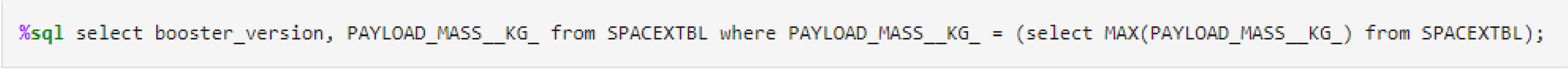
- List the total number of successful and failure mission outcomes



```
%sql select count(*)  from SPACEXTBL where "Mission_Outcome" like "%succ%";
```

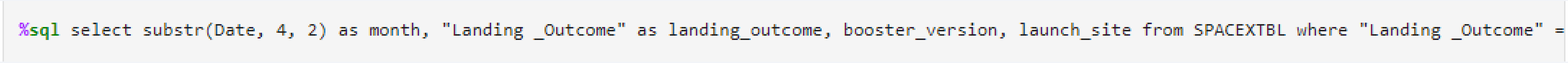
```
%sql select count(*)  from SPACEXTBL where "Mission_Outcome" like "%fail%";
```

- List the names of the booster_versions which have carried the maximum payload mass



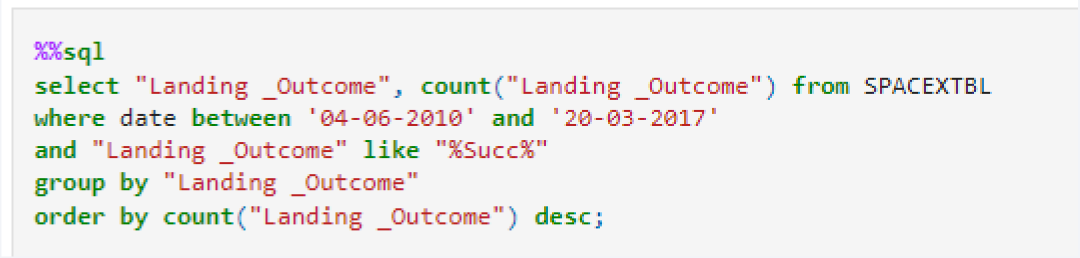
```
%sql select booster_version, PAYLOAD_MASS__KG_ from SPACEXTBL where PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

- 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



```
%sql select substr(Date, 4, 2) as month, "Landing _Outcome" as landing_outcome, booster_version, launch_site from SPACEXTBL where "Landing _Outcome" =
```

- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.



```
%%sql
select "Landing _Outcome", count("Landing _Outcome") from SPACEXTBL
where date between '04-06-2010' and '20-03-2017'
and "Landing _Outcome" like "%Succ%"
group by "Landing _Outcome"
order by count("Landing _Outcome") desc;
```

Reference:https://github.com/SophiaQY/IBM-Data-Science/blob/main/jupyter-labs-eda-sql-coursera_sqllite_submit.ipynb

Build an Interactive Map with Folium

- TASK 1: Mark all launch sites on a map
  - ✓ Create and add folium.Circle and folium.Marker for each launch site on the site map
- TASK 2: Mark the success/failed launches for each site on the map
  - ✓ For each launch result in spacex_df data frame, add a folium.Marker to marker_cluster
- TASK 3: Calculate the distances between a launch site to its proximities
  - ✓ Mark down a point on the closest coastline using MousePosition and calculate the distance between the coastline point and the launch site
  - ✓ Draw a PolyLine between a launch site to the selected coastline point
  - ✓ Create a marker with distance to a closest city, railway, highway, etc

Reference:https://github.com/SophiaQY/IBM-Data-Science/blob/main/lab_jupyter_launch_site_location.jupyterlite_submit.ipynb

Build a Dashboard with Plotly Dash

- Launch result by sites (pie chart)
  - You can choose different launch site to see the success rate
- Launch result vs payload mass by different booster version
  - You can choose different mass ranges of payload mass to see launch result
  - You can also know the booster version of each launch at the same time

Reference:https://github.com/SophiaQY/IBM-Data-Science/blob/main/Dashbord_app_submit.ipynb

Predictive Analysis (Classification)

- We split the data into training and testing data using the function train_test_split. The training data is divided into validation data, a second set used for training data; then the models are trained and hyperparameters are selected using the function GridSearchCV.
- Models built for prediction
  - Logistic Regression
  - Super vector machine
  - Decision tree
  - K nearest neighbor
- The accuracy of different models were compared to select the best model

Reference:https://github.com/SophiaQY/IBM-Data-Science/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

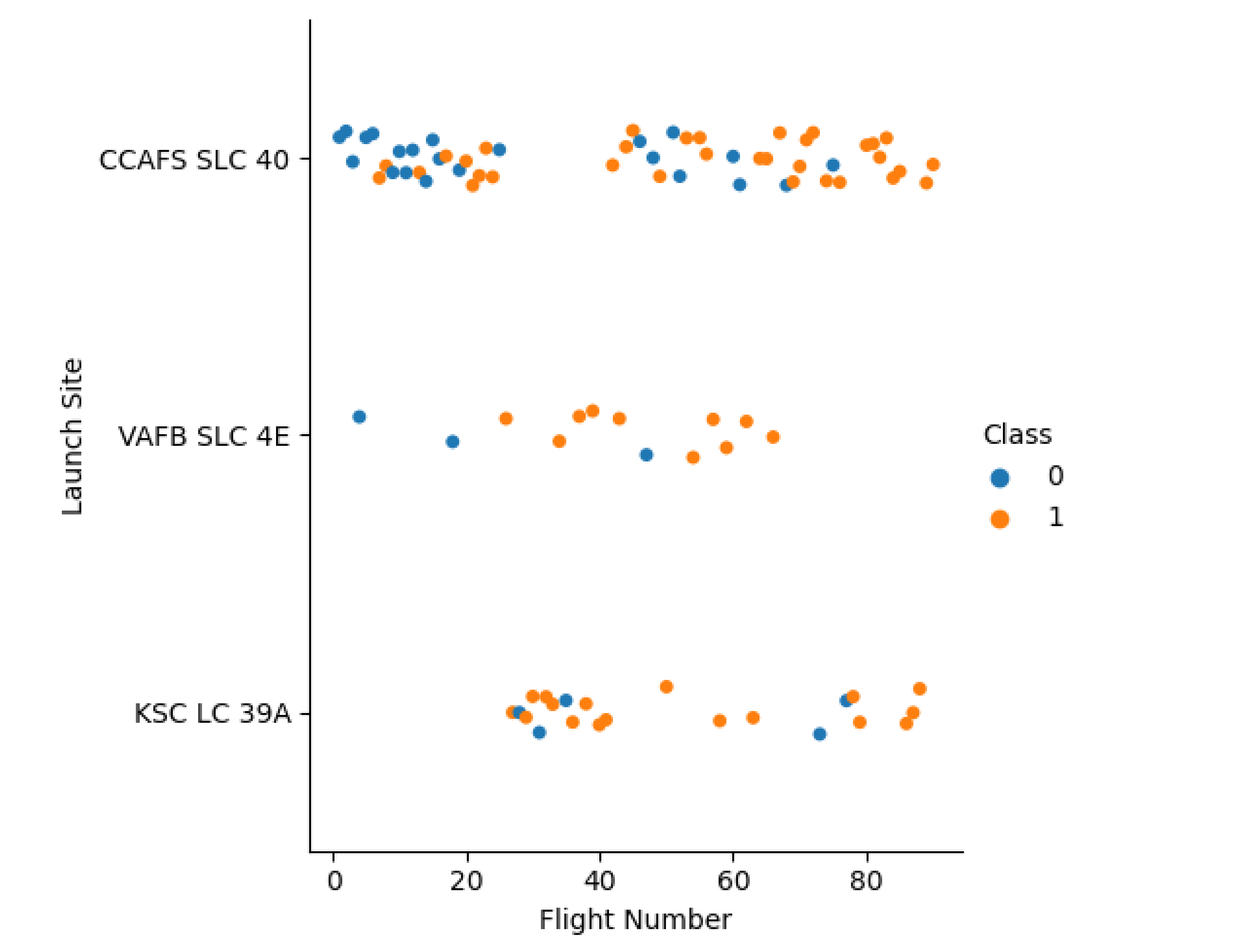
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

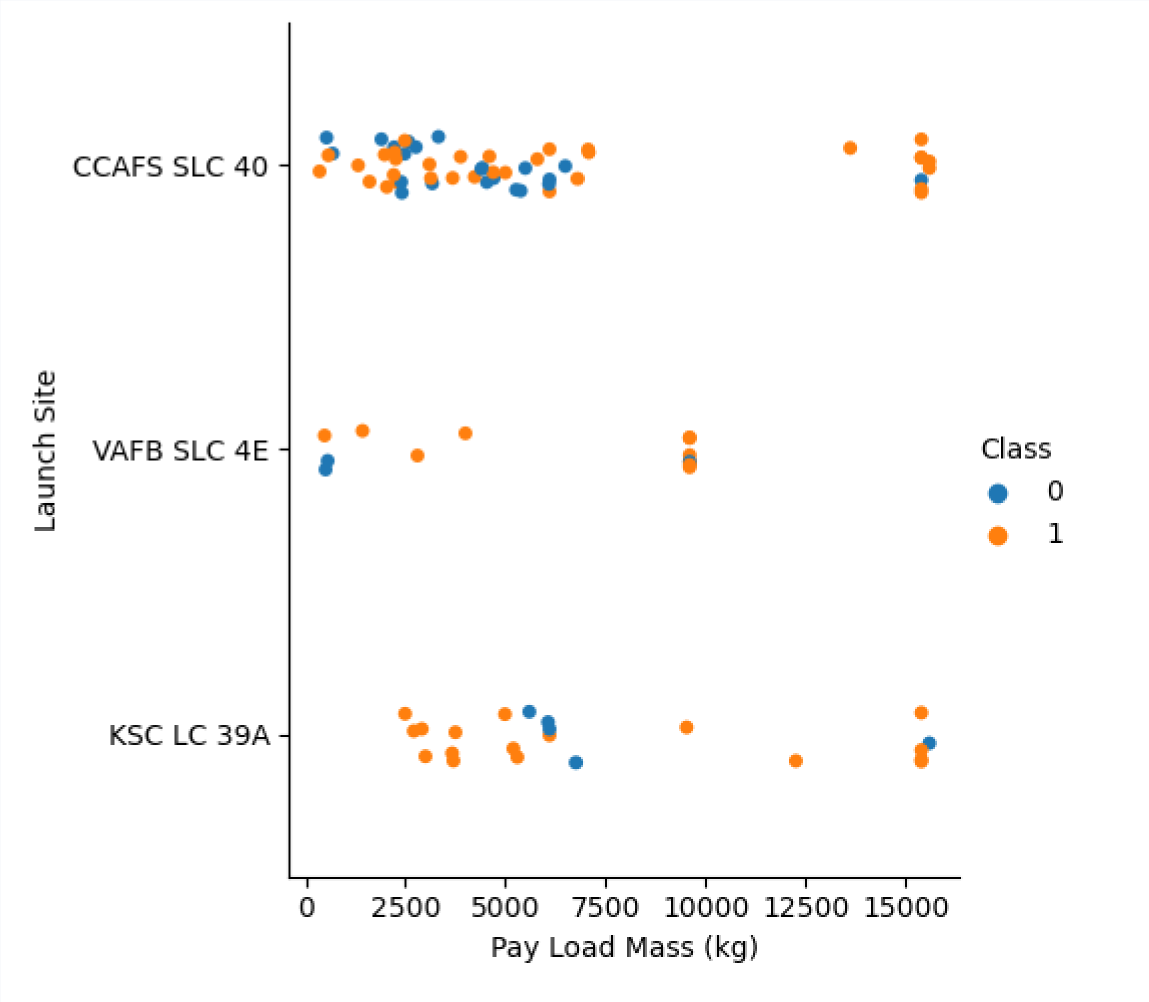
Flight Number vs. Launch Site

- Success rate increases along with flight number among all three launch sites
- CCAFS SLC 40 has most launches
- VAFB SLC 4E doesn’t have launches with flight number higher than 70
- KSC LC 39A doesn’t have flight number lower than 20



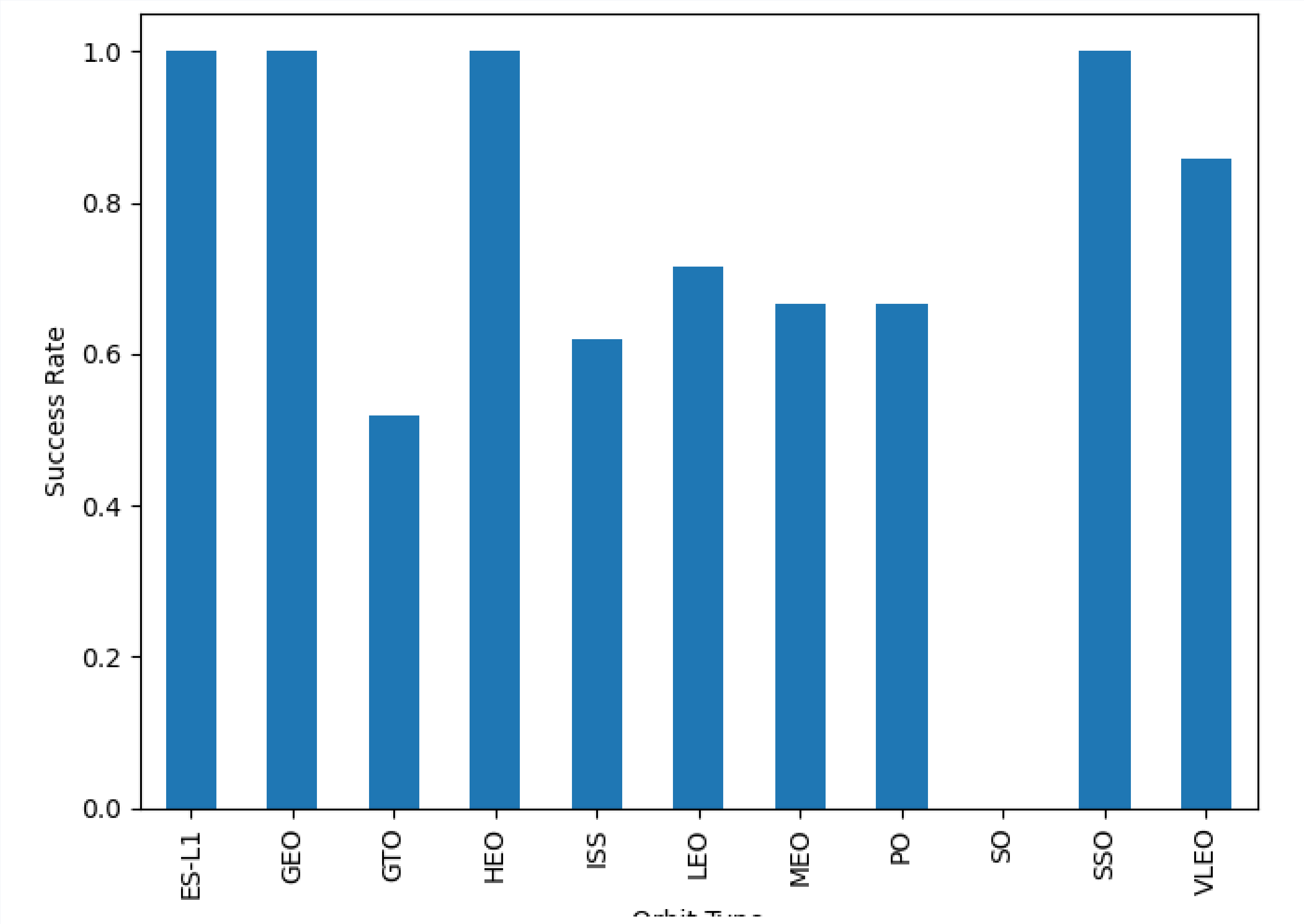
Payload vs. Launch Site

- Higher pay load mass bring higher success rate
- CCAFS SLC 40 launches pay load mass lower than 8000kg and higher than 12500kg
- VAFB SLC 4E launces small to medium play load mass
- KSC LC 39A launches medium to larch play load mass



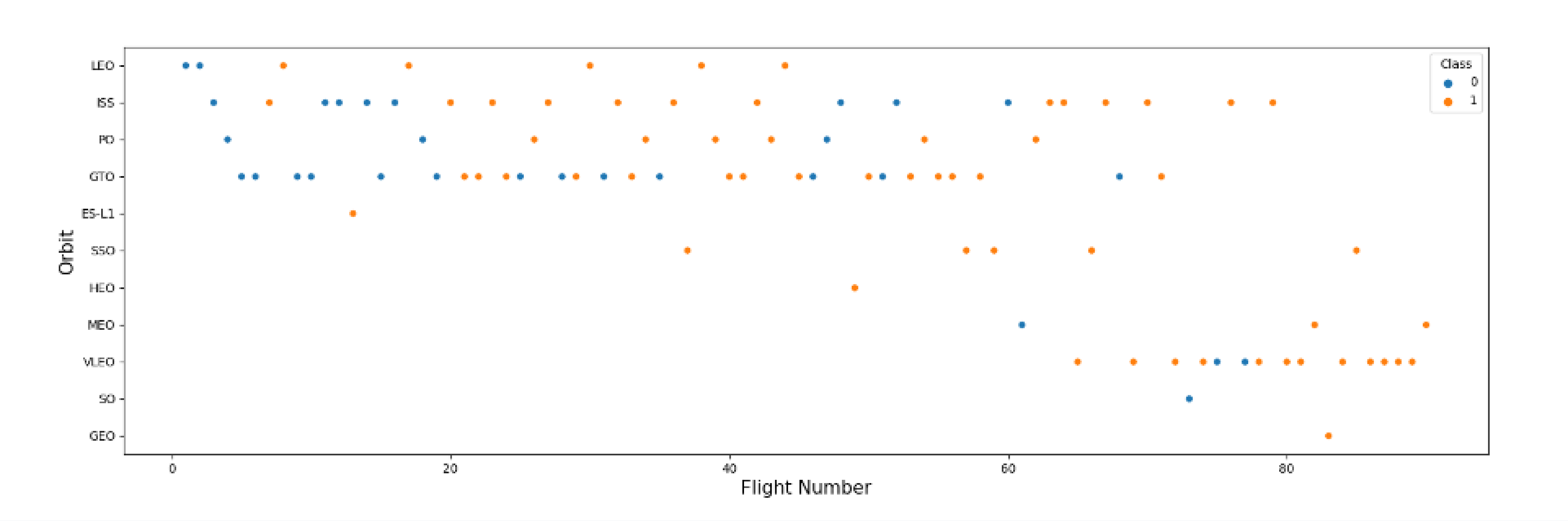
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO and VLEO orbit has high success rate at 0.8 and higher
- GTO, ISS, LEO, MEO, PO has success rate between 0.5 and 0.7
- SO has no success rate



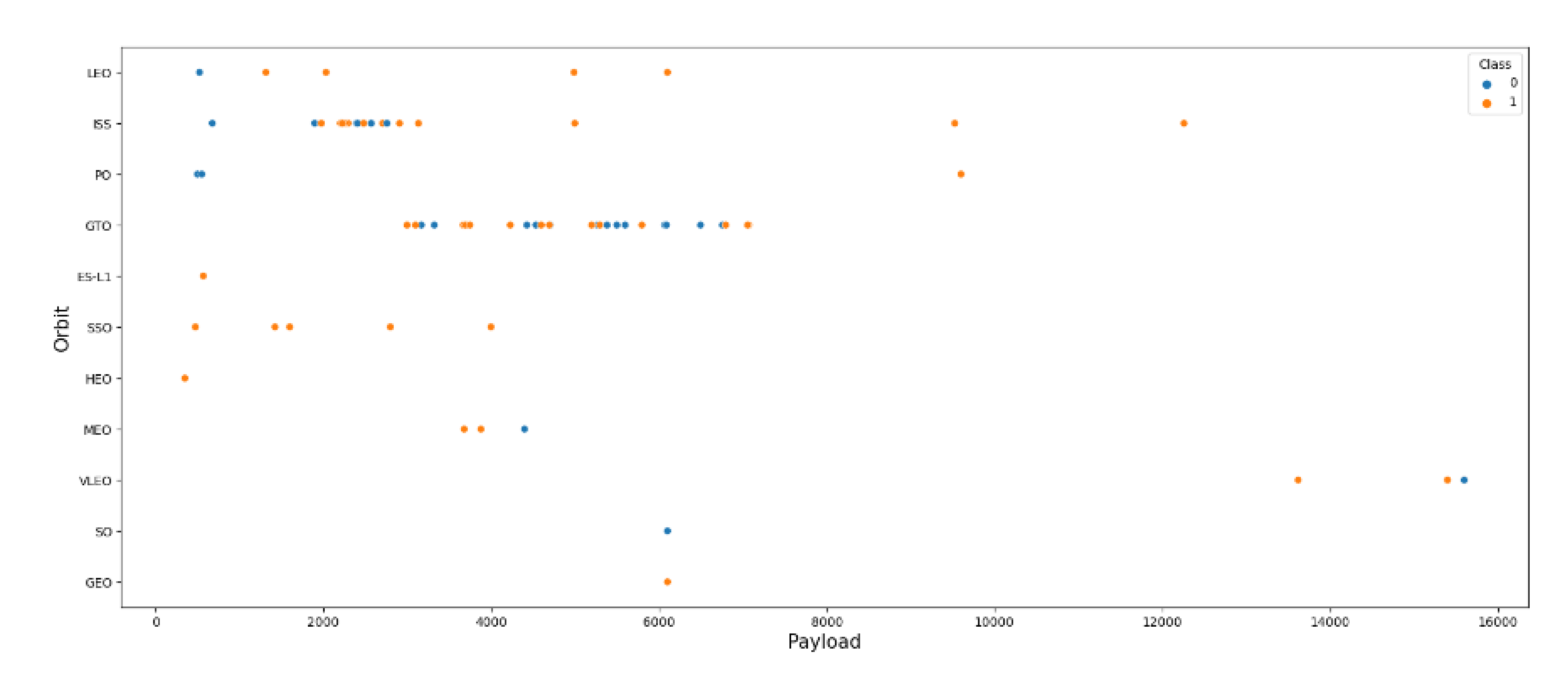
Flight Number vs. Orbit Type

In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



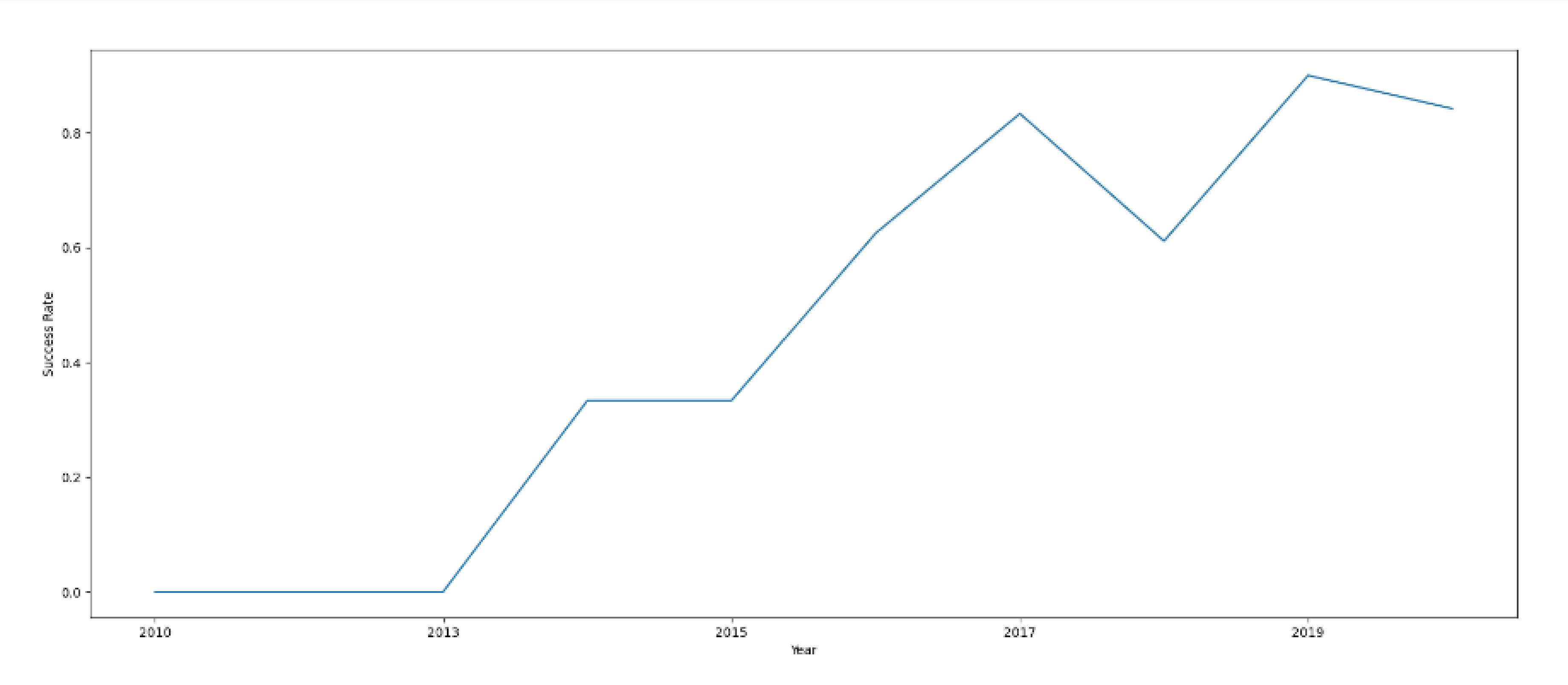
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However, for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



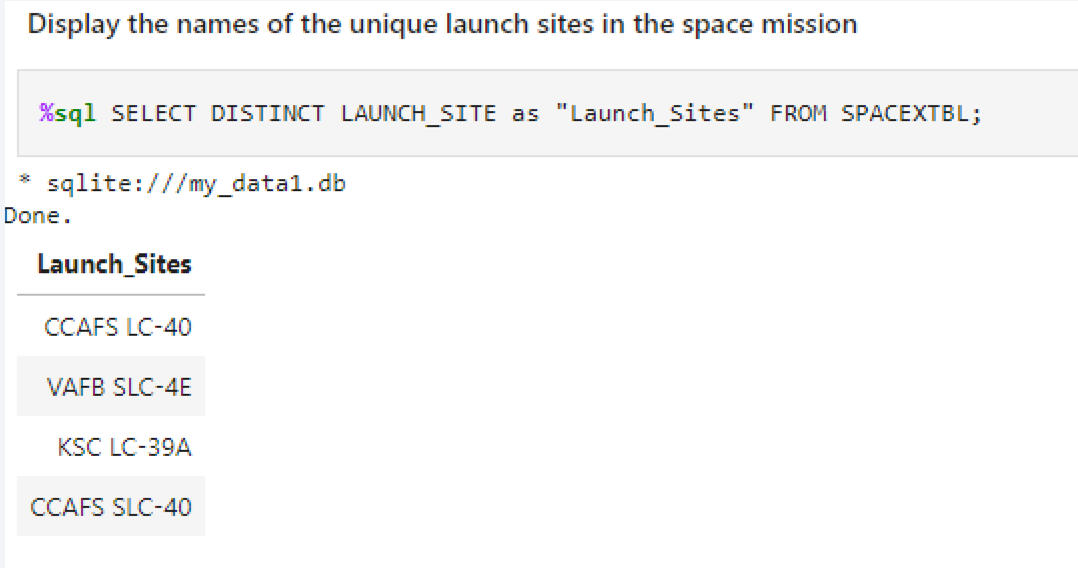
Launch Success Yearly Trend

Success rate since 2013 kept increasing till 2020



All Launch Site Names

Use DISTINCT to find the names of the unique launch sites



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

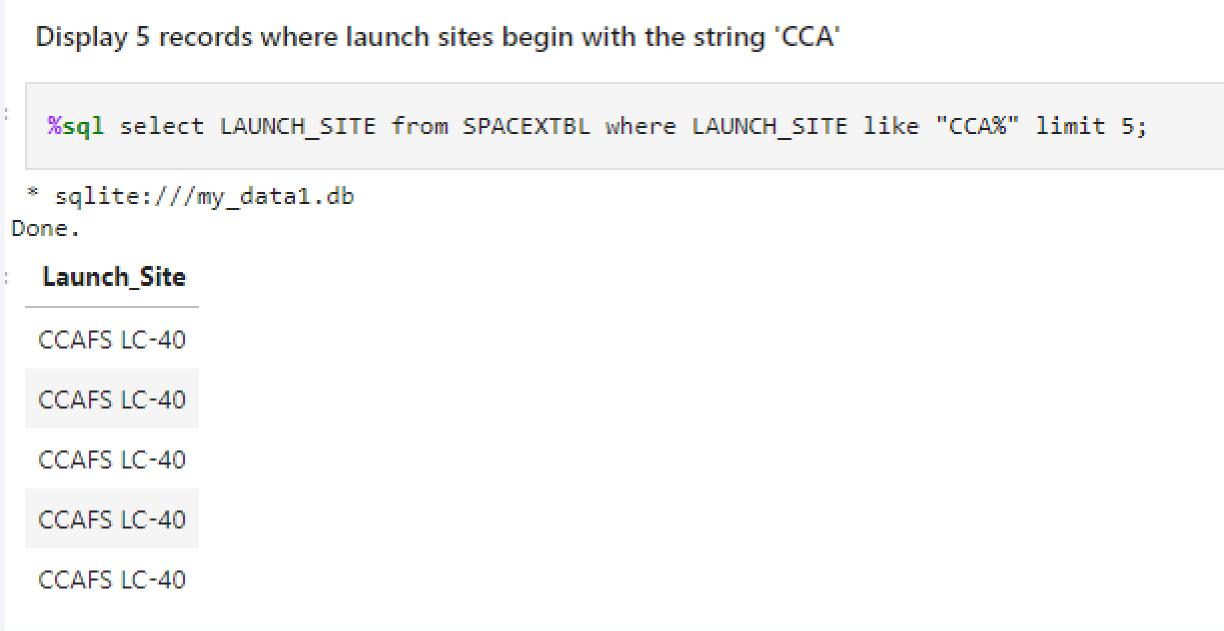
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Use `like “CCA%”` to select launch site begin with `CCA`
- Use `limit 5` to select the first five recods



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

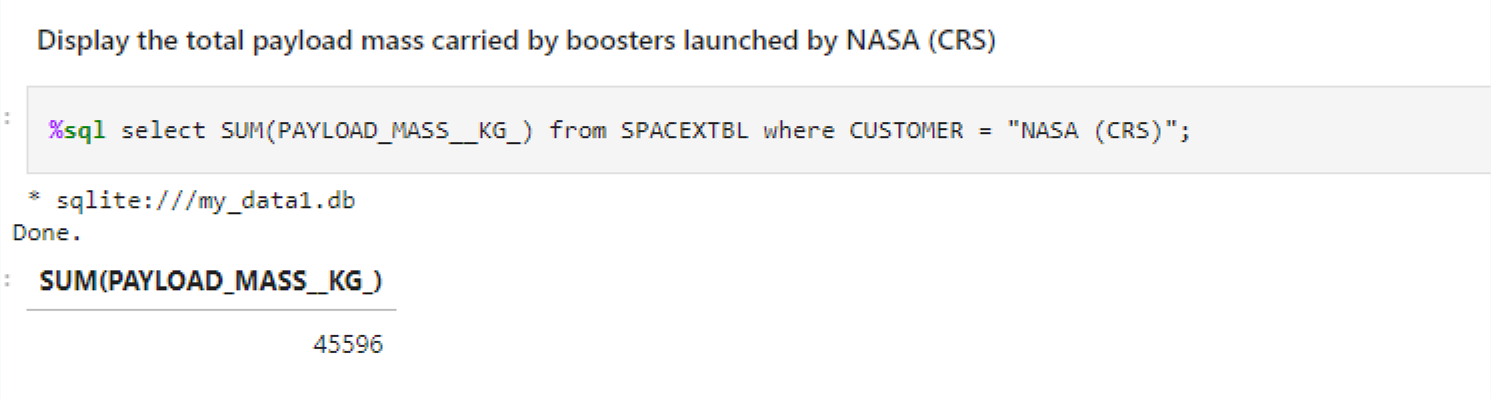
```
%sql select LAUNCH_SITE from SPACEXTBL where LAUNCH_SITE like "CCA%" limit 5;
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Use `sum` to calculate the total payload mass
- Use where clause to select boosters from NASA



Display the total payload mass carried by boosters launched by NASA (CRS)

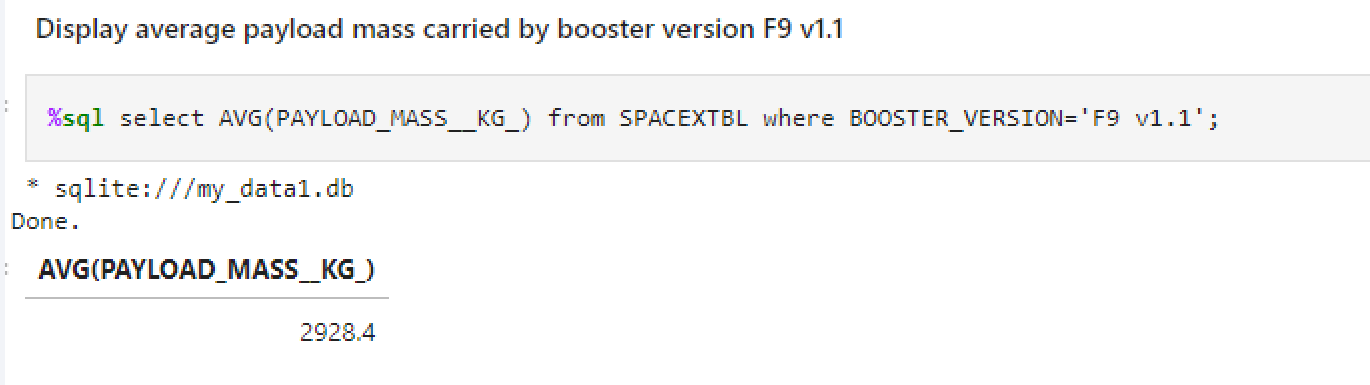
```
%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = "NASA (CRS)";
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Use `AVG` to calculate the average payload mass
- Use where clause to select booster version



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

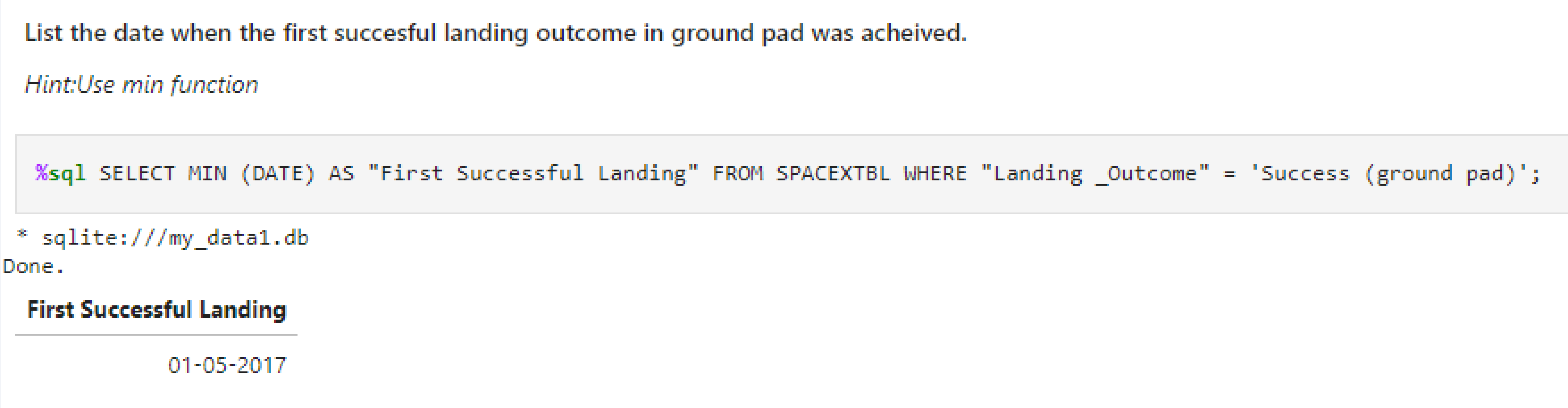
```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION='F9 v1.1';
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Use min() function to choose the date of first successful landing
- Use where clause to select ground pad successful landing



List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

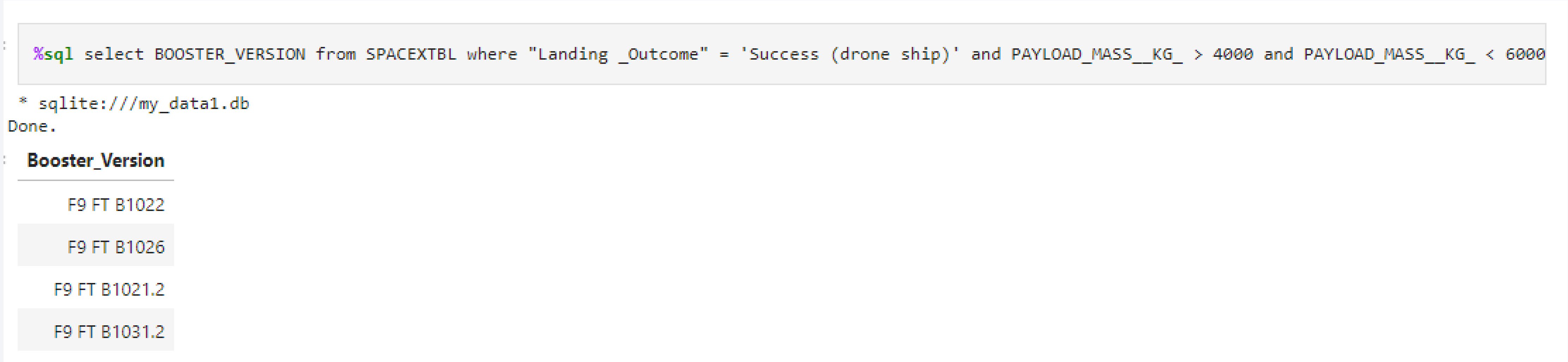
```
%sql SELECT MIN (DATE) AS "First Successful Landing" FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (ground pad)';
```

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

First Successful Landing
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Use and to add two conditions in where clause to select target booster version



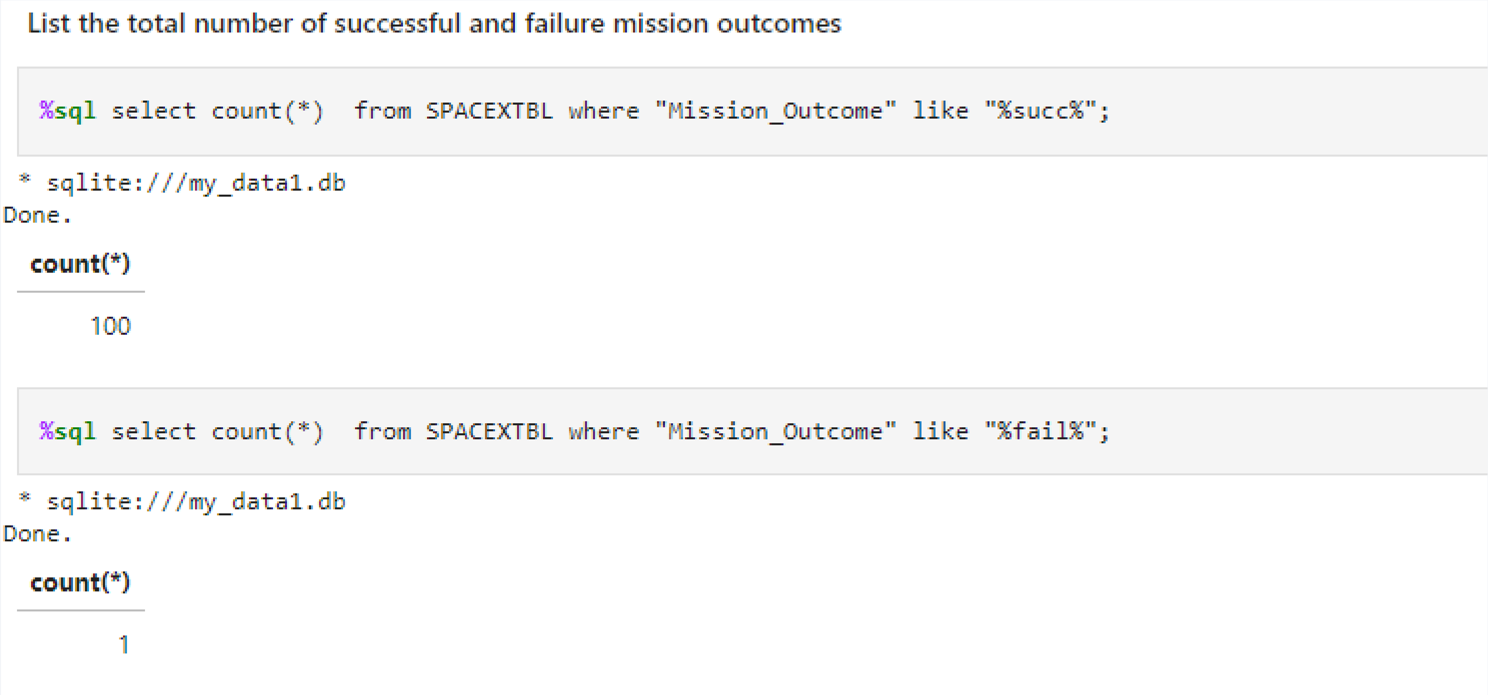
```
%sql select BOOSTER_VERSION from SPACEXTBL where "Landing _Outcome" = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

```
 * 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

- Calculate the total number of successful and failure mission outcomes
- Use count() to calculate the total number of outcomes
- Use `like “%succ%”` and `like “%fail%”` to select successful and failure outcomes



List the total number of successful and failure mission outcomes

```
%sql select count(*)  from SPACEXTBL where "Mission_Outcome" like "%succ%";
```

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

count(*)
100

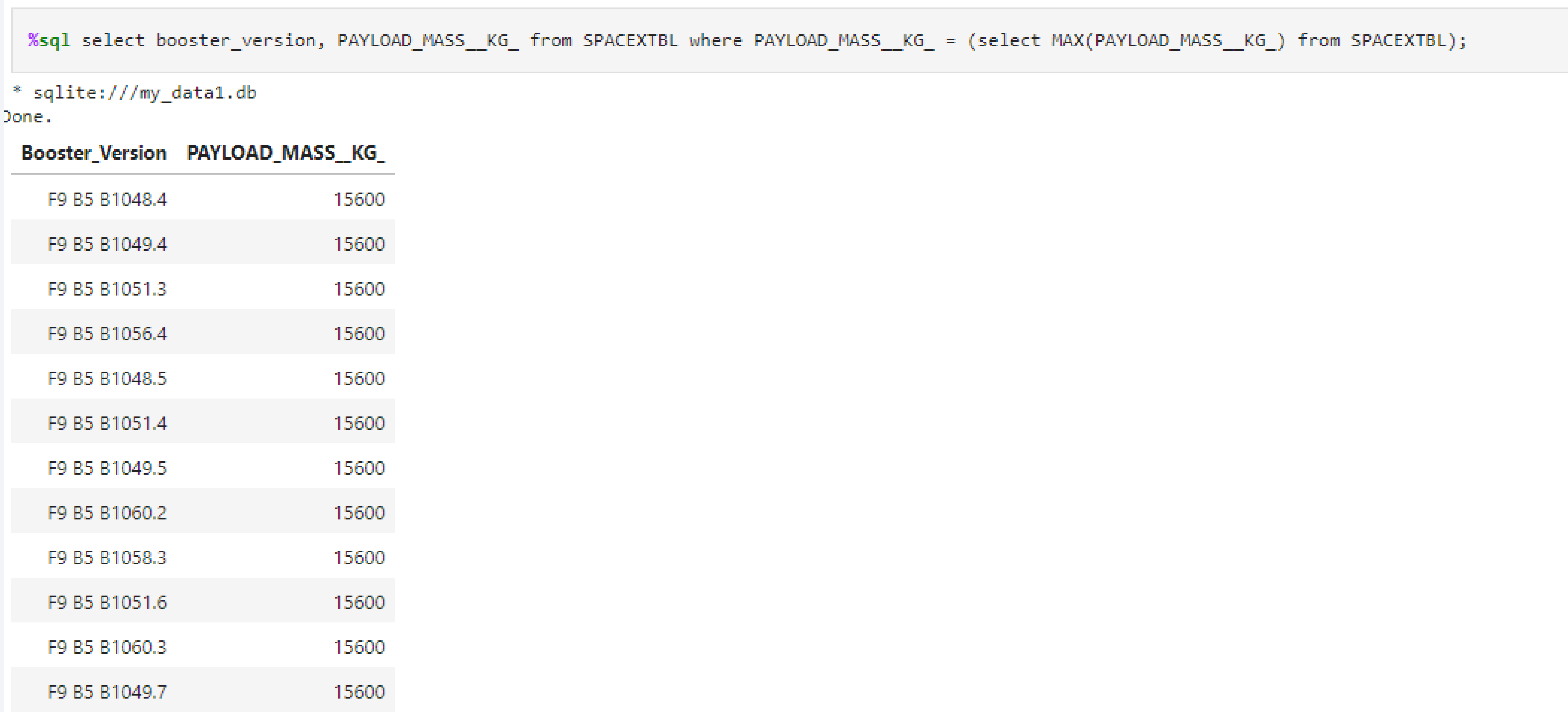
```
%sql select count(*)  from SPACEXTBL where "Mission_Outcome" like "%fail%";
```

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

count(*)
1

Boosters Carried Maximum Payload

List the names of the booster which have carried the maximum payload mass



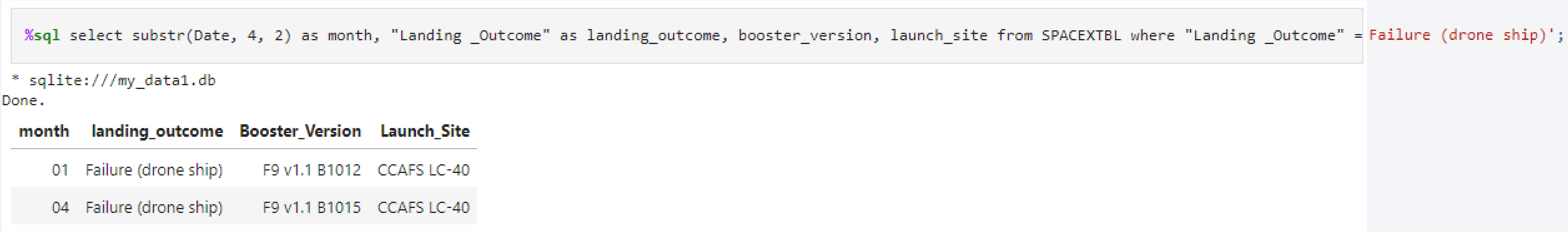
```
%sql select booster_version, PAYLOAD_MASS__KG_ from SPACEXTBL where PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Use substr(Date, 4, 2) to get value for month
- Use where clause to choose drone ship failure type



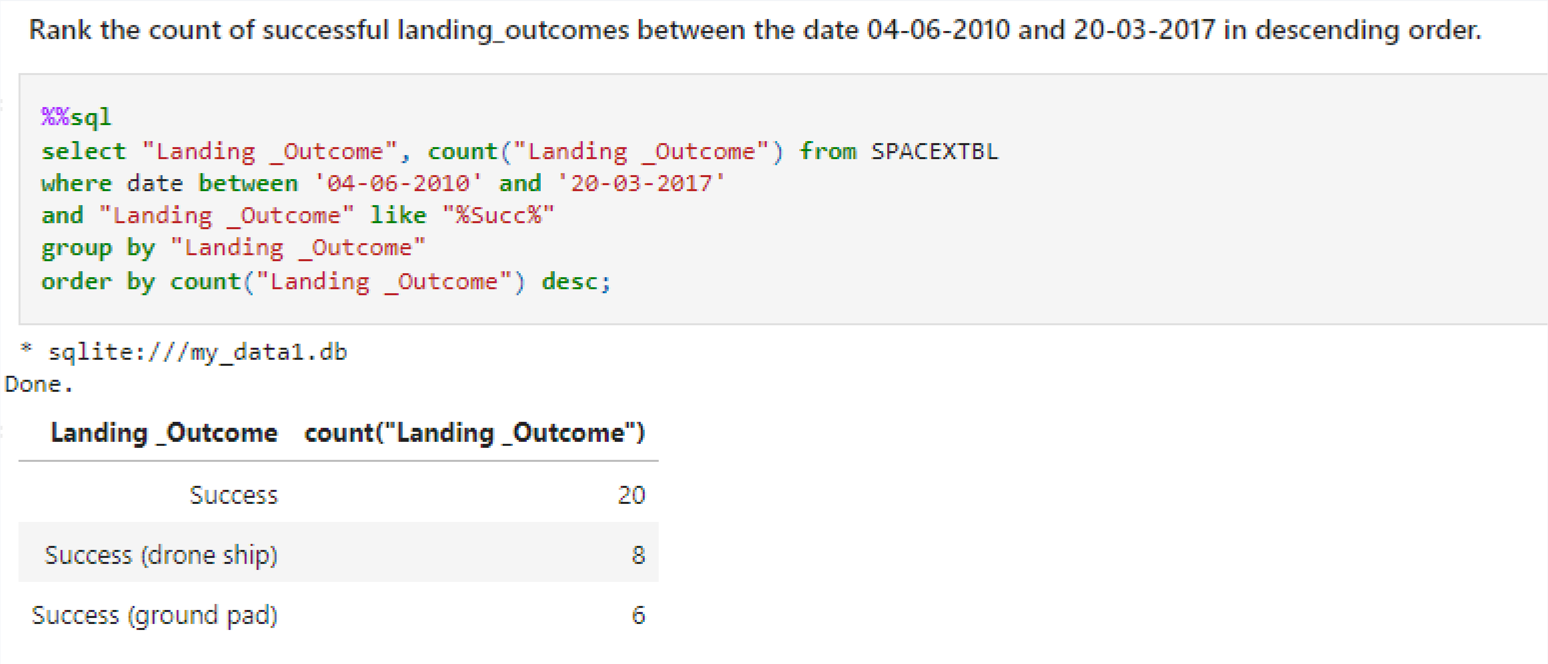
```
%sql select substr(Date, 4, 2) as month, "Landing _Outcome" as landing_outcome, booster_version, launch_site from SPACEXTBL where "Landing _Outcome" = Failure (drone ship)';
```

```
 * 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

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

- Rank the count of success landing between the date 2010-06-04 and 2017-03-20, in descending order
- Use group by to group outcomes by their type



Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%%sql
select "Landing _Outcome", count("Landing _Outcome") from SPACEXTBL
where date between '04-06-2010' and '20-03-2017'
and "Landing _Outcome" like "%Succ%"
group by "Landing _Outcome"
order by count("Landing _Outcome") desc;
```

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

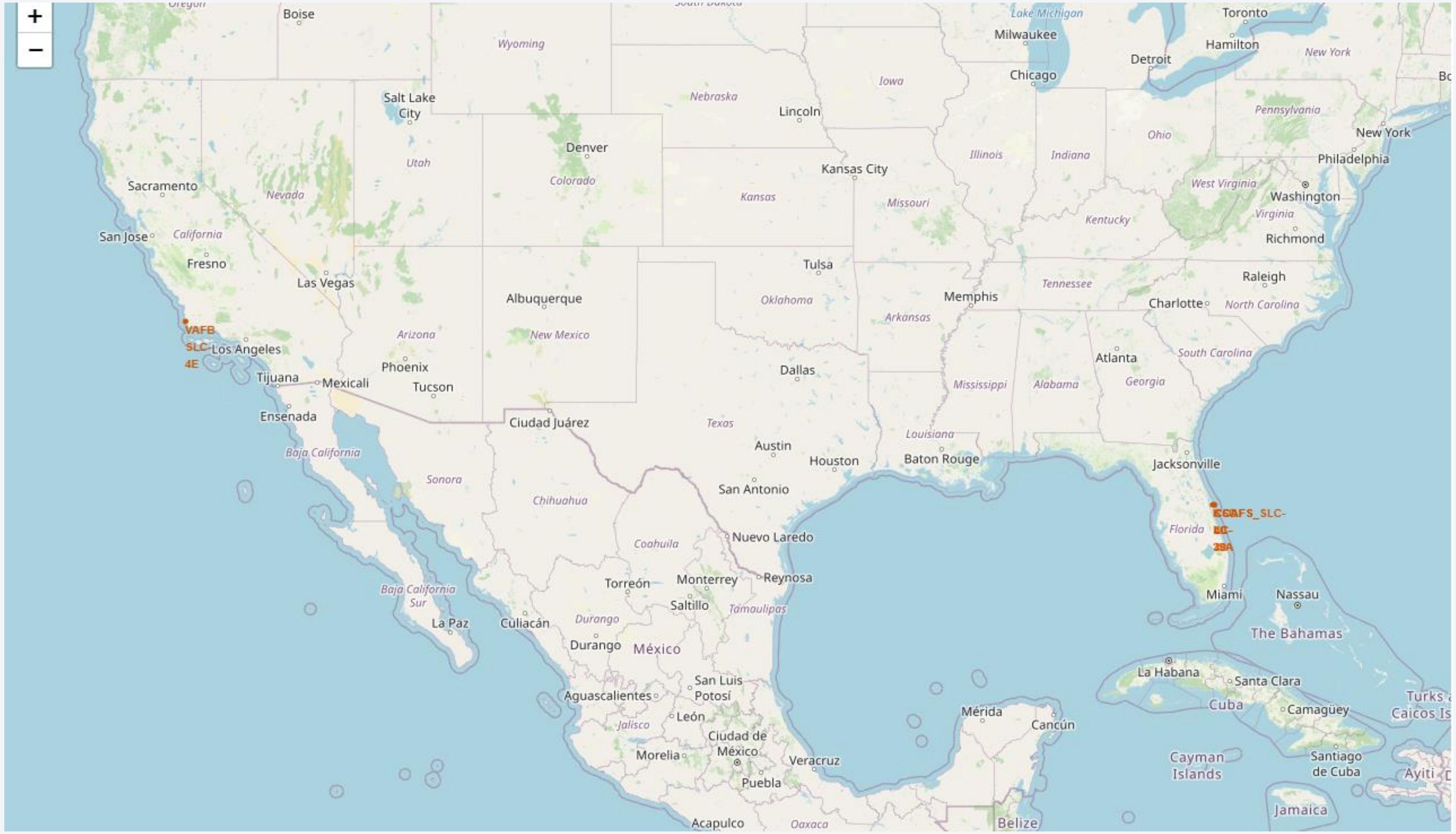
Landing _Outcome	count("Landing _Outcome")
Success	20
Success (drone ship)	8
Success (ground pad)	6

Section 3

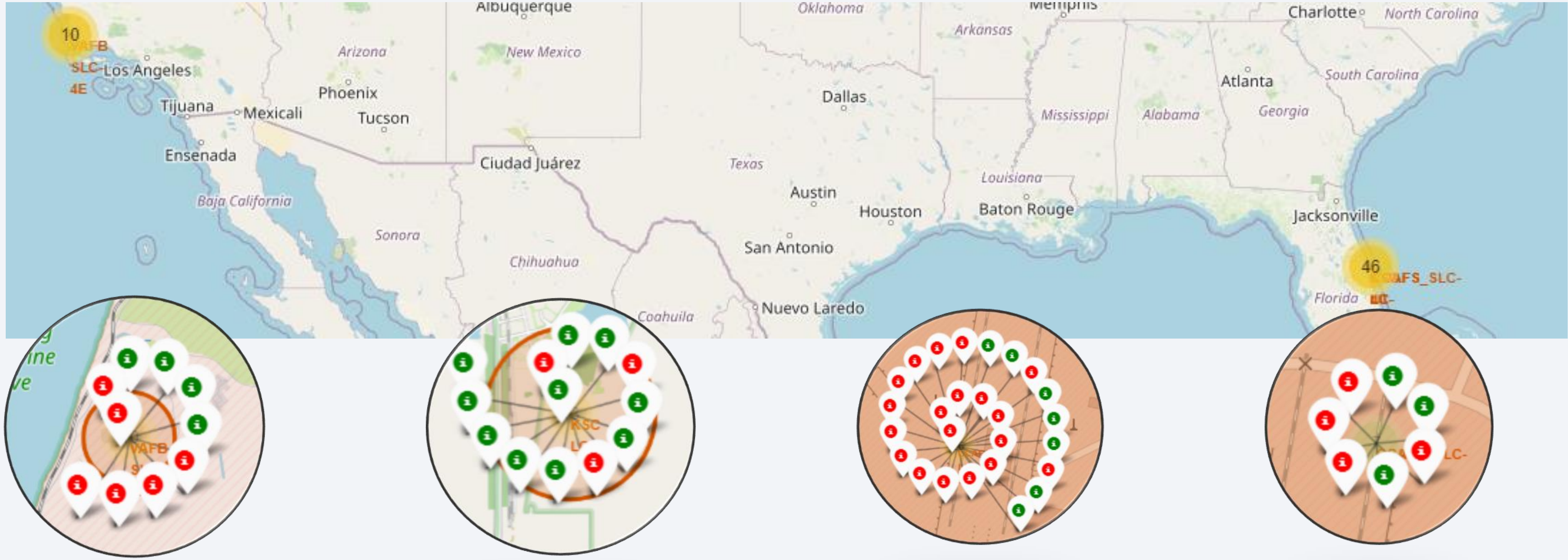
Launch Sites Proximities Analysis

Launch sites location map

There are one launch site in California and three launch sites in Florida

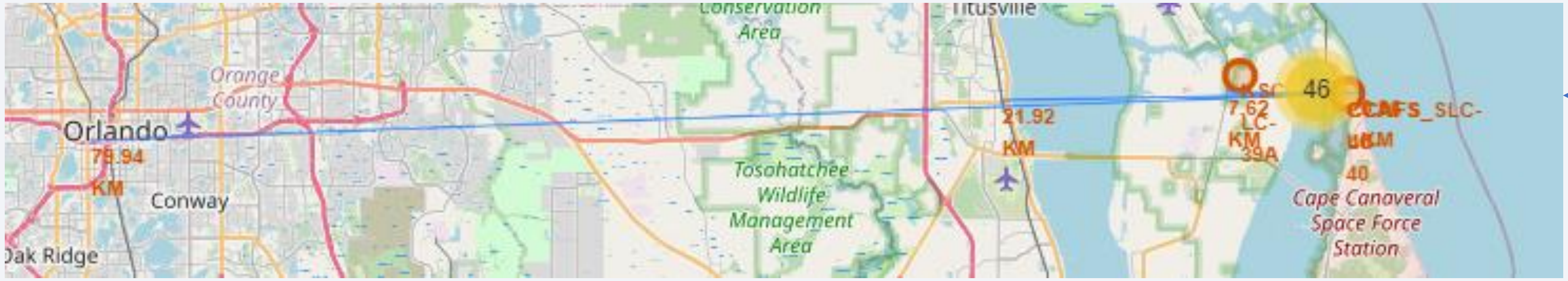


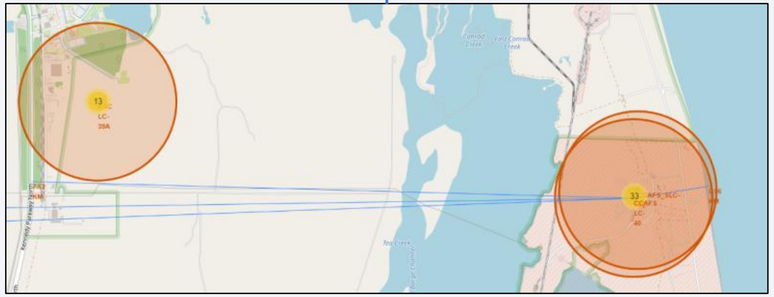
Launch outcomes in all sites



- From left to right: VAFB SLC-4E, KSC LC-39A, CCAFS LC-40, CCAFS_SLC-40
- KSC LC-39A site has highest success rate and CCAFS LC-40 site has most launches with high rate of failures.

Nearest coastline, parkway, highway and large city

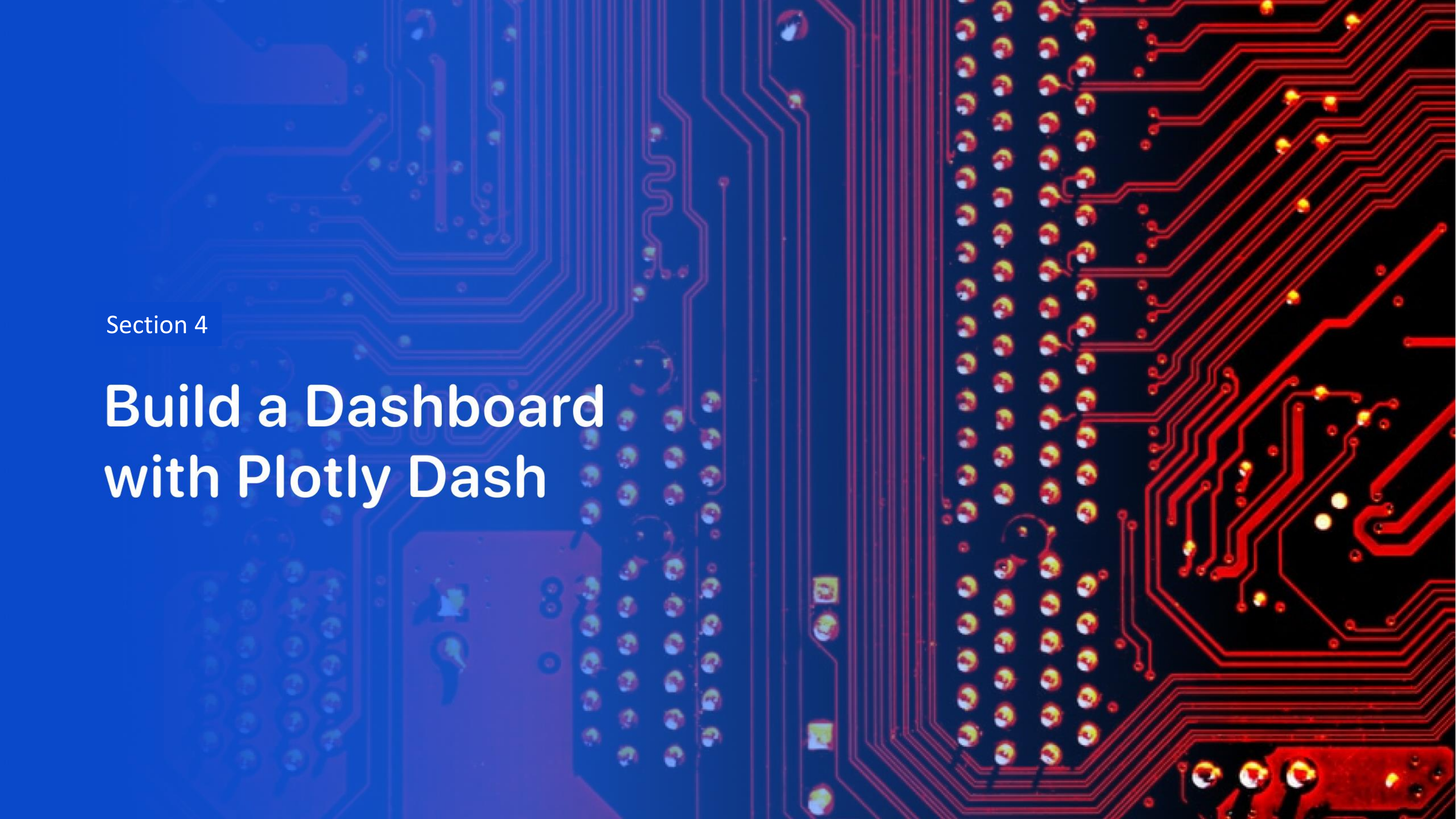




- Nearest coastline to launch site is 0.96 km away
- Nearest parkway to launch site is 7.62 km away
- Nearest highway to launch site is 21.92 away
- Nearest large city Orlando to launch site is 79.94 km away

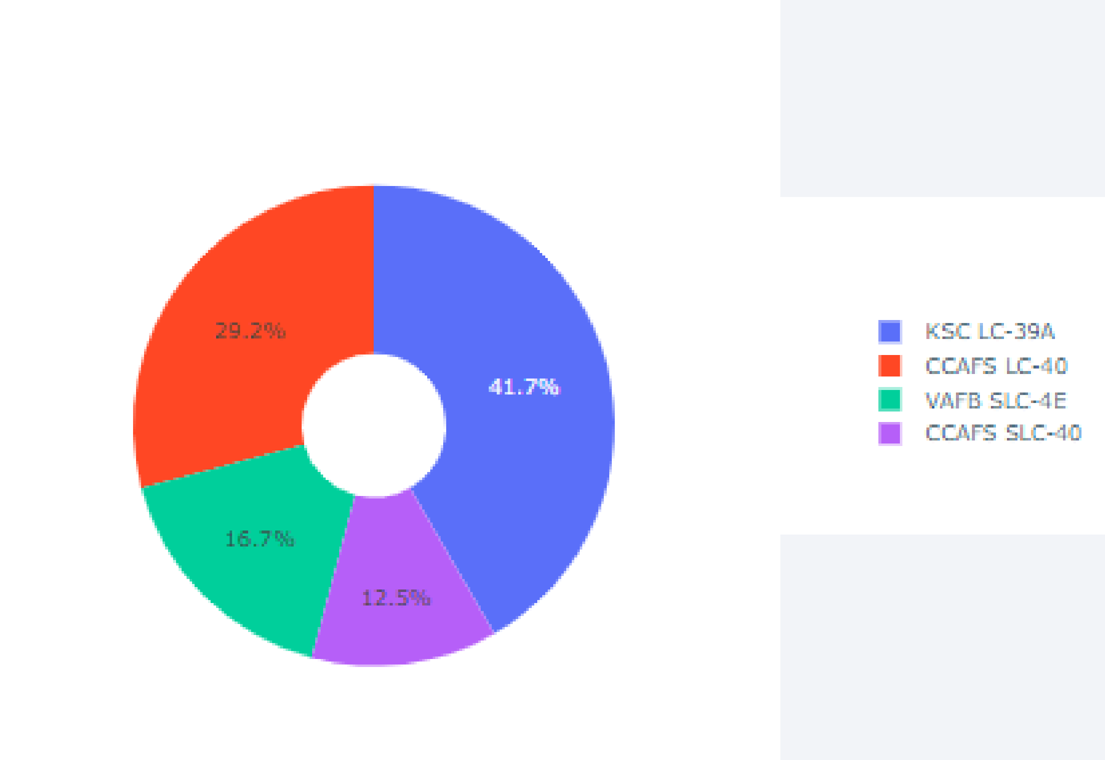
Section 4

Build a Dashboard with Plotly Dash

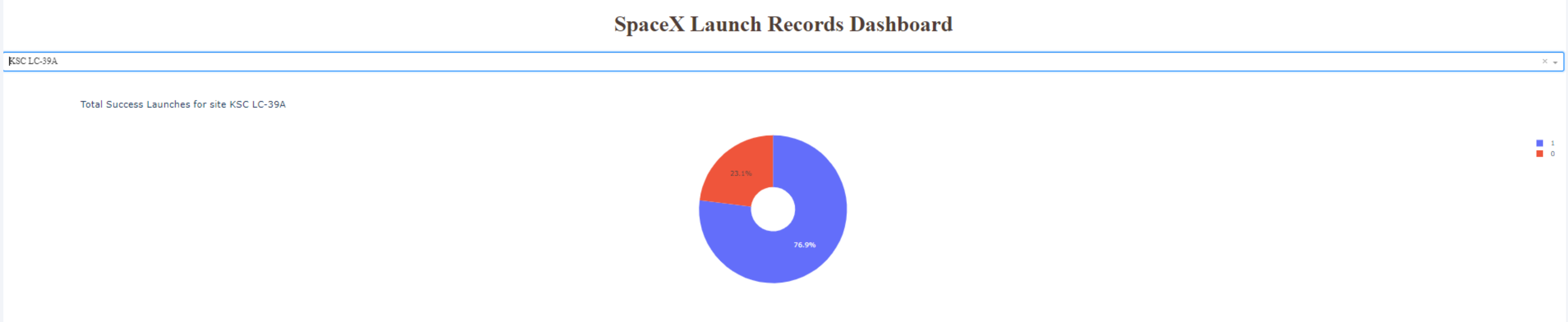


Launch site count for all sites

- This chart shows the percentile of successful launches in each site among total successful launches
- KSC LC-39A has the highest successful rate
- CCAFS SLC-40 has the lowest successful rate



Outcomes by different launchsite



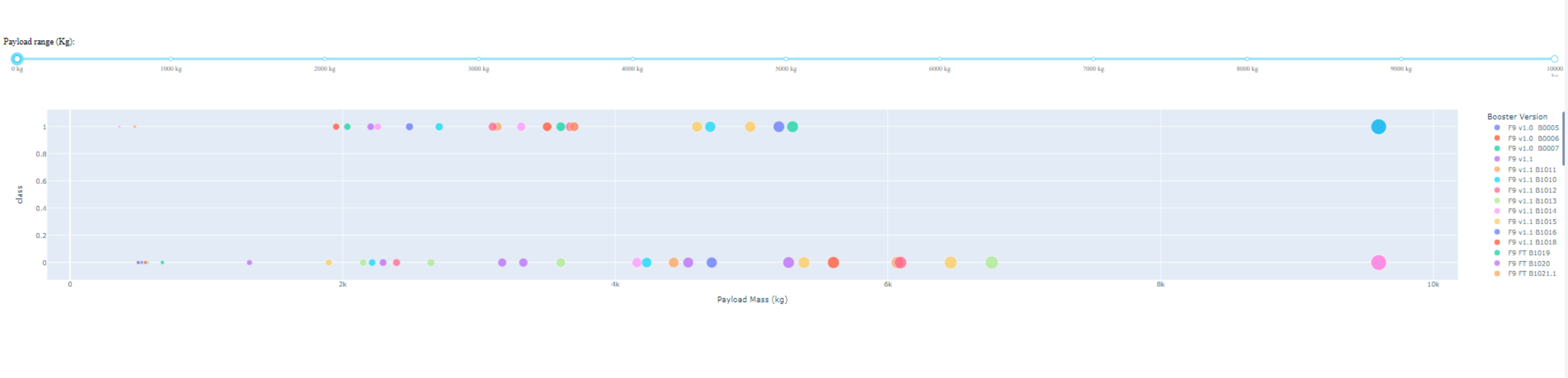
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success Launches for site KSC LC-39A

- KSC LC-39A has highest successful launch rate which is 76.9%

Outcomes by different payload mass

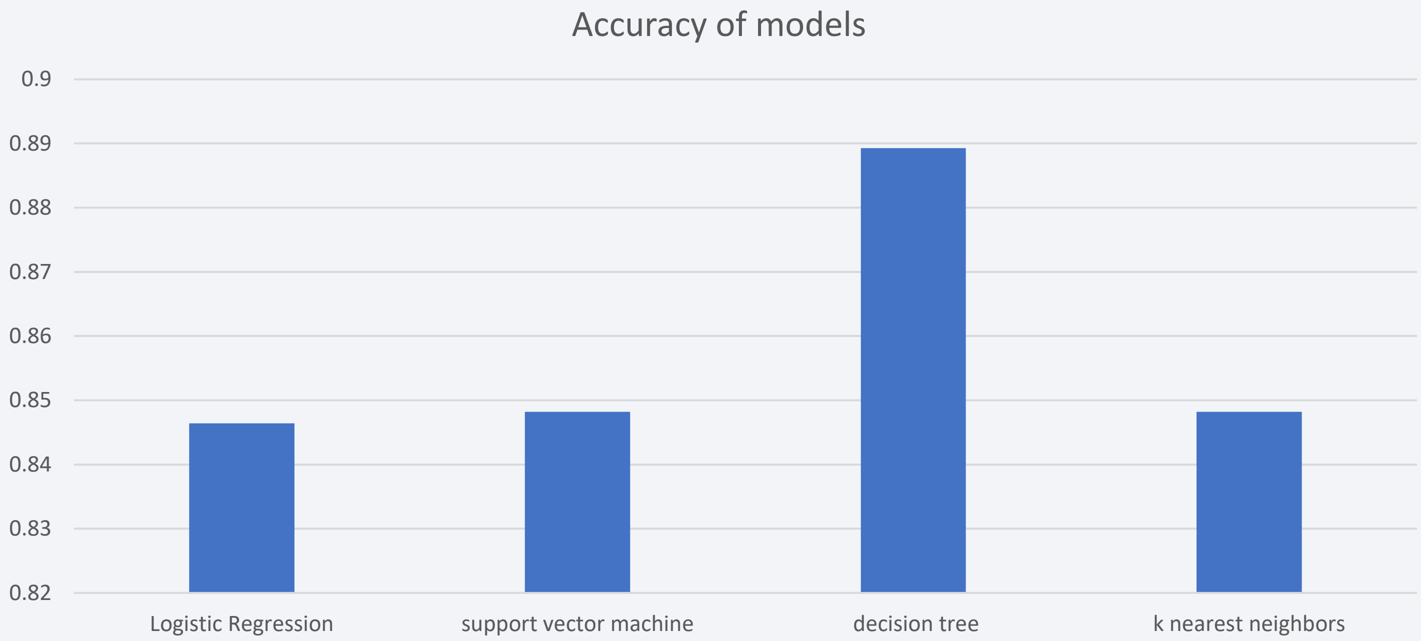


- Launches with lower payload mass have higher successful rate than heavy payload mass

Section 5

Predictive Analysis (Classification)

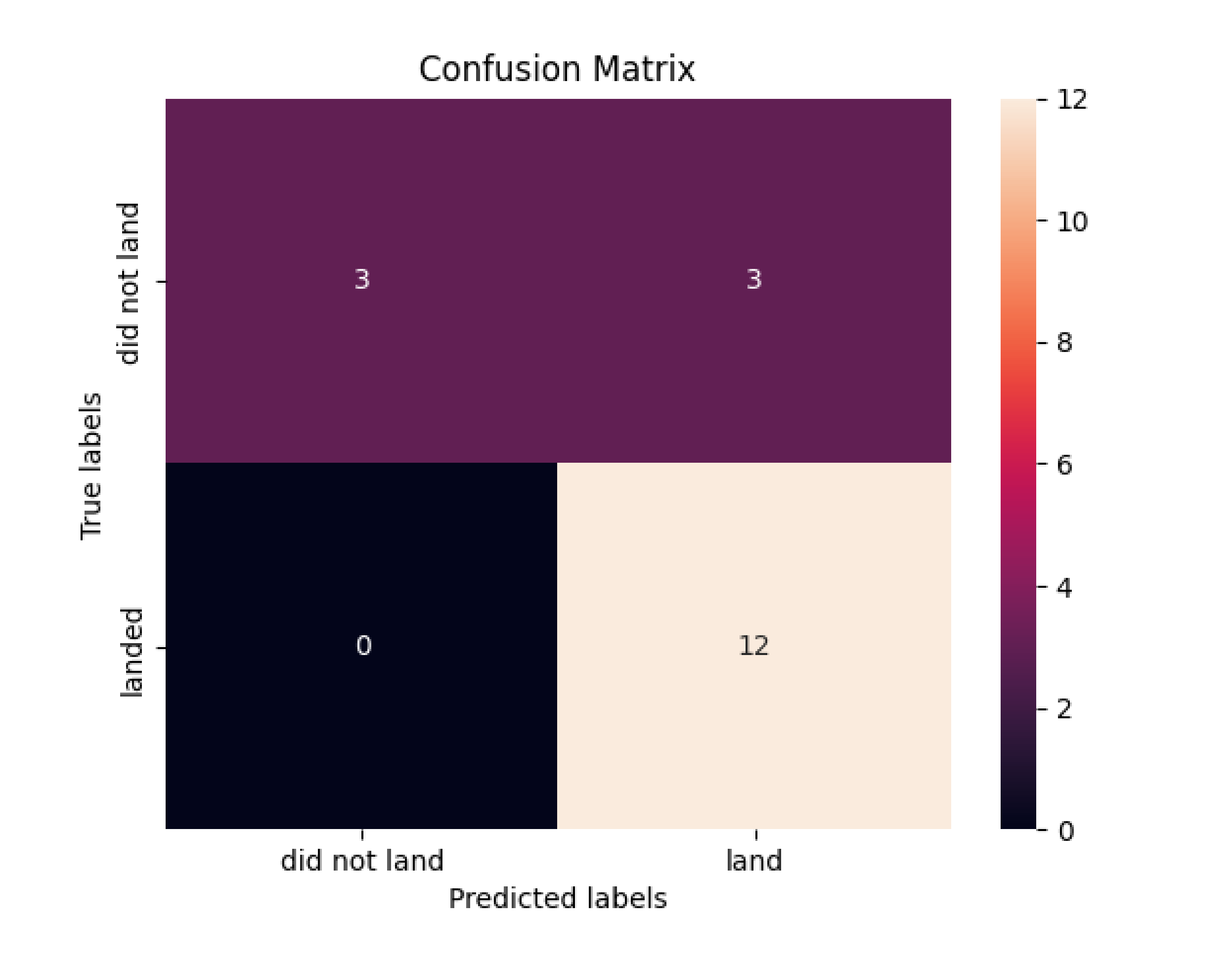
Classification Accuracy



Decision Tree has the best Accuracy with 0.889

Confusion Matrix

- The confusion matrix show that the decision tree algorithm has:
  - very good sensitivity (12/12)
  - bad specificity (3/6)
  - Fair precision (12/15)



Conclusions

- Higher flight number tends to have higher success rate
- Payload mass and orbit have influence in success rate
- Success rate since 2013 kept increasing till 2020
- Launch site in Florida is near to the coastline and highway but away from big cities
- KSC LC-39A launch site has highest successful launch rate
- Decision Tree algorithm gives highest accuracy in predicting landing outcome

Thank you!