IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<k.kubo>

<2023/12/21>

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Use SpaceX Rest API
- Data Wraging
- Get Data from SQL
- Visualize Data
- Build Map
- Build DashBoard

Introduction

- SpaceX advertises the launch cost of its Falcon 9 rocket at $62 million, which is considerably cheaper than the $165 million plus costs of other providers. The cost efficiency of SpaceX is mainly attributed to the reuse of the first stage. Predicting the success of landings can aid in estimating launch costs. This information is valuable for competitors aiming to secure rocket launch contracts in bidding wars against SpaceX.
- Can we determine whether the first stage of Falcon 9 will successfully land using the provided data?

Section 1

Methodology

Methodology

Executive Summary

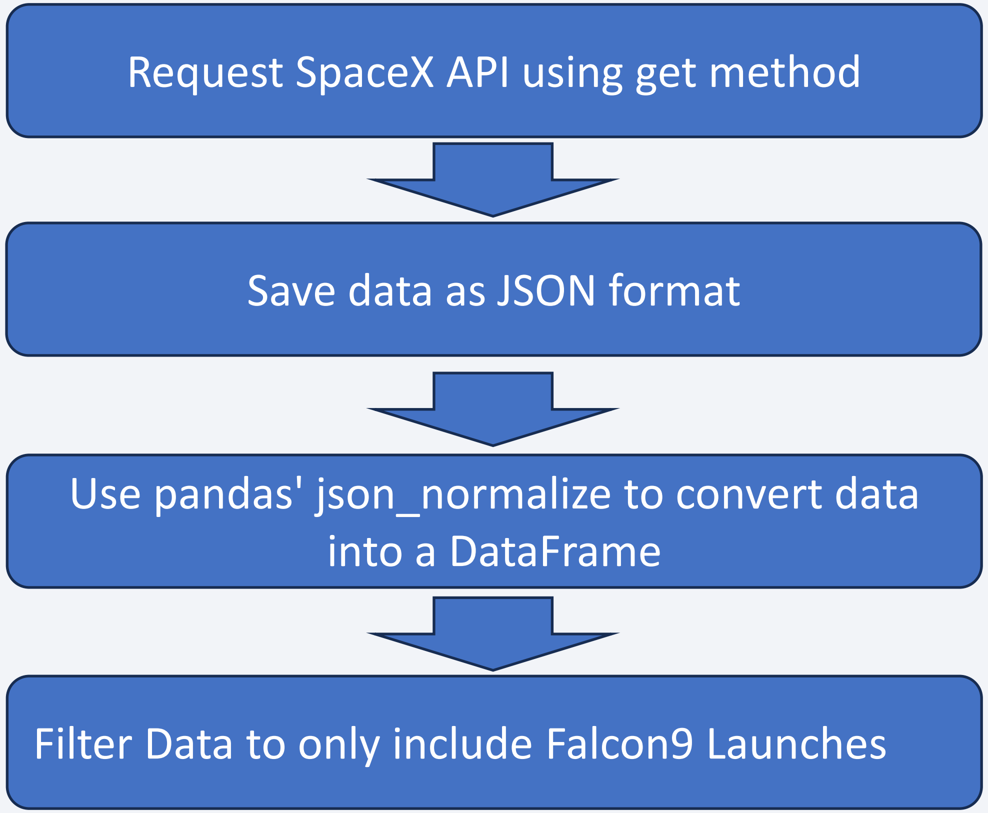
- Data collection methodology:
  - Using SpaceX Rest API
- Perform data wrangling
  - Using API to get the data from json.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Using Decision Tree Clasification.

Data Collection

- In the data collection process, a combination of API requests from the SpaceX REST API and web scraping data from the tables on SpaceX's Wikipedia page was used.
- Scraping of dates from the tables on SpaceX's Wikipedia page.
- To obtain complete information about the launches for detailed analysis, both methods needed to be used.

Data Collection – SpaceX API

URL: https://github.com/kubooooooo/testrepo/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Request SpaceX API using get method

Save data as JSON format

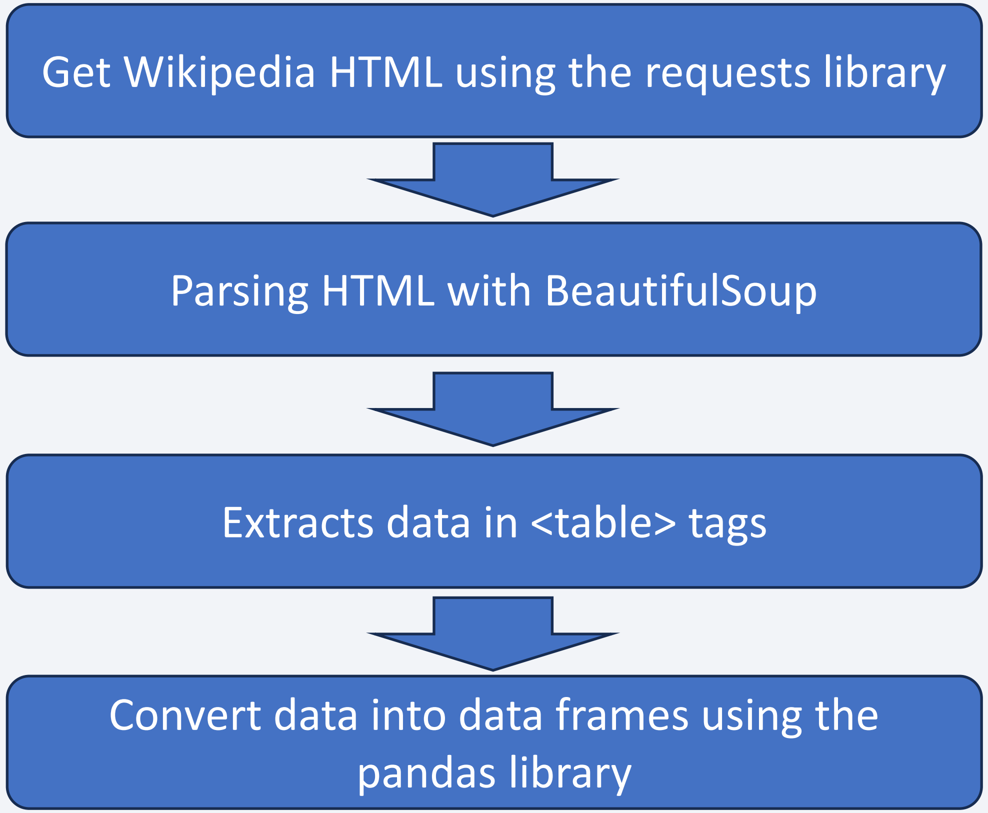
Use pandas' json_normalize to convert data into a DataFrame

Filter Data to only include Falcon9 Launches

Data Collection - Scraping

URL:

https://github.com/kubooooooo/testrepo/blob/main/jupyter-labs-webscraping.ipynb



Get Wikipedia HTML using the requests library

Parsing HTML with BeautifulSoup

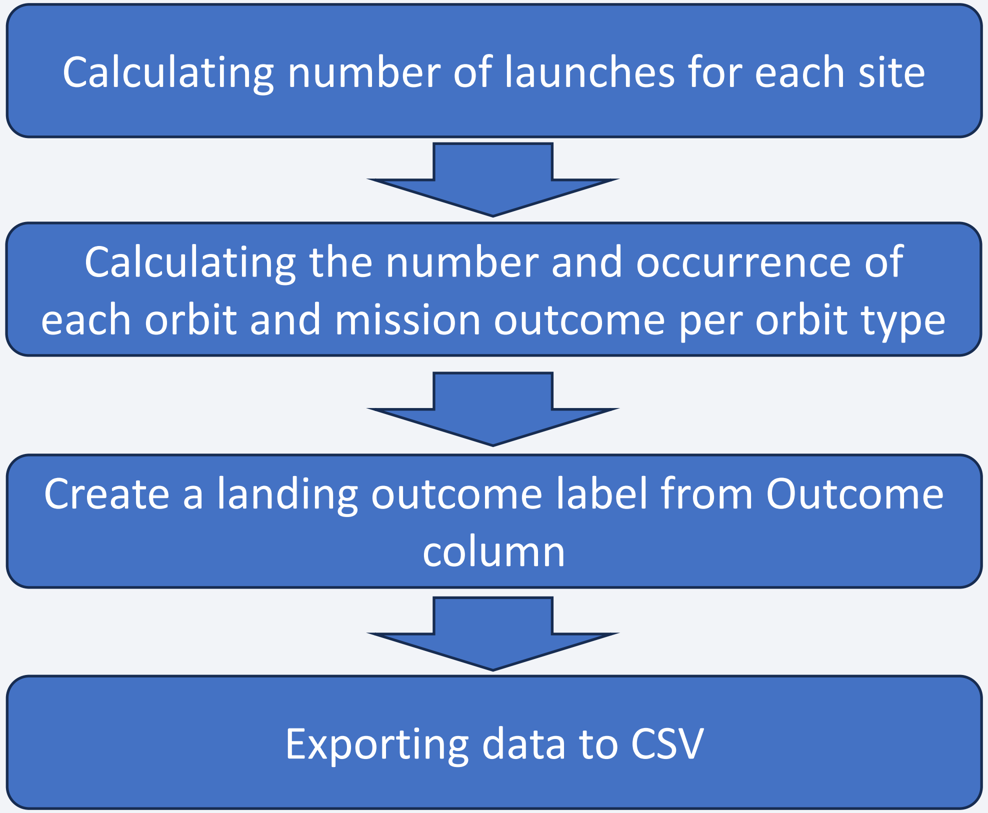
Extracts data in <table> tags

Convert data into data frames using the pandas library

Data Wrangling

URL:

https://github.com/kubooooooo/testrepo/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



Calculating number of launches for each site

Calculating the number and occurrence of each orbit and mission outcome per orbit type

Create a landing outcome label from Outcome column

Exporting data to CSV

EDA with Data Visualization

- Exploratory Data Analysis was conducted on variables such as flight number, payload mass, launch site, orbit, class, and year.
- Scatter plots, line graphs, and bar charts were used to evaluate the relationships between variables and to ascertain their suitability for training machine learning models.
- These visualizations were instrumental in determining the presence or absence of relationships between variables before incorporating them into the model training process. Analysis included the number of flights versus payload mass, number of flights versus launch location, payload mass versus launch location, orbit versus success rate, number of flights versus orbit, payload versus orbit, and the annual trend of success rate.

URL:

https://github.com/kuboooooo/testrepo/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Collected data from IBM's cloud storage platform into a DataFrame and created SQL tables.
- Utilized SQLite integrated with Python to perform EDA (Exploratory Data Analysis) for gaining valuable insights from the data.
- Discovered information such as clear launch sites, specific entries related to particular launch sites, aggregations of different payloads, information about successful landings, and details regarding failed landings.
- URL:

https://github.com/kuboooooo/testrepo/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Folium maps are effectively used for pinpoint visualization of key data such as launch and landing sites, distinguishing between successful and failed landings, and proximity to critical locations like railways, highways, coasts, and cities.
- These maps serve as comprehensive visual tools to delve deeper into the rationale behind specific chosen launch locations.
- By overlaying these significant locations with landing success spots, a clear correlation between these factors and successful landings can be discerned.
- This visualization not only reveals geographical considerations affecting launch site placements but also provides valuable insights into strategic positioning relative to key infrastructure, aiding in the analysis of landing successes in relation to spatial context and proximity to these crucial features.

Build a Dashboard with Plotly Dash

- Dashboard:
  Pie Chart
  Scatter Plot.

- Choose to display a pie chart:
  Distribution of landing success rates at all rocket launch sites Success rates at individual launch sites.
  The scatter plot has two inputs: All sites or individual sites, and Payload mass on a slider ranging from 0 to 10000 kg.
  The pie chart is used to visualize the success rates of launch sites.

Predictive Analysis (Classification)

- Splitting the Class Column
- Using StandardScaler to standardize the data
- Splitting the data into Train/Test variables
- Using GridSearchCV for parameter tuning
- Applying tuned parameters to LogReg, SVM, Decision Tree, KNN models
- Scoring the models on the test set
- Confusion matrices for all models
- Comparing the scores of the models

URL:
https://github.com/kubooooooo/testrepo/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

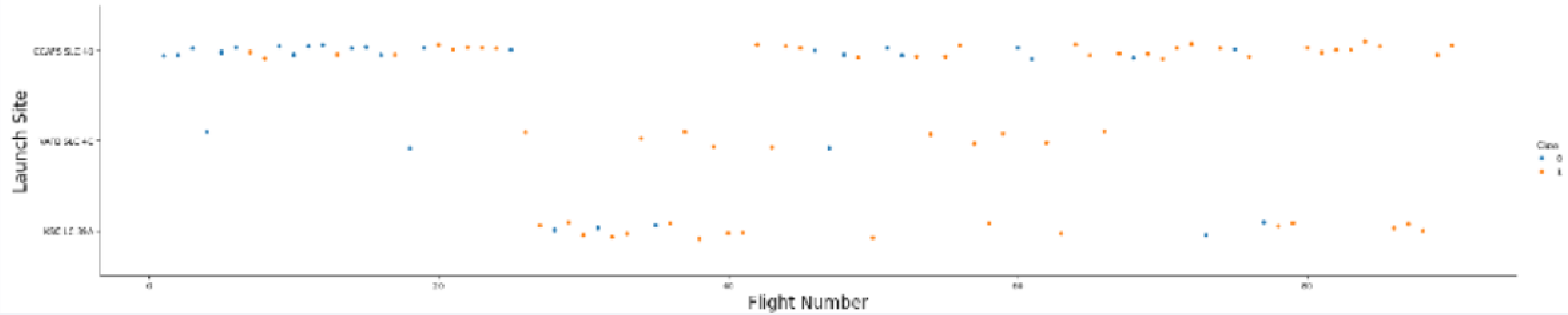
Results

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

Section 2

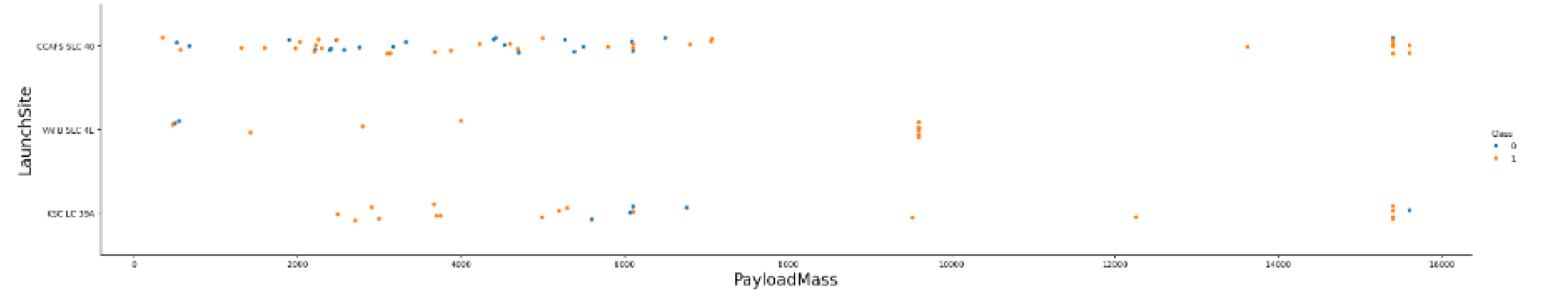
Insights drawn from EDA

Flight Number vs. Launch Site



- CCAFS SLC 40 has the highest number of Flight Numbers.

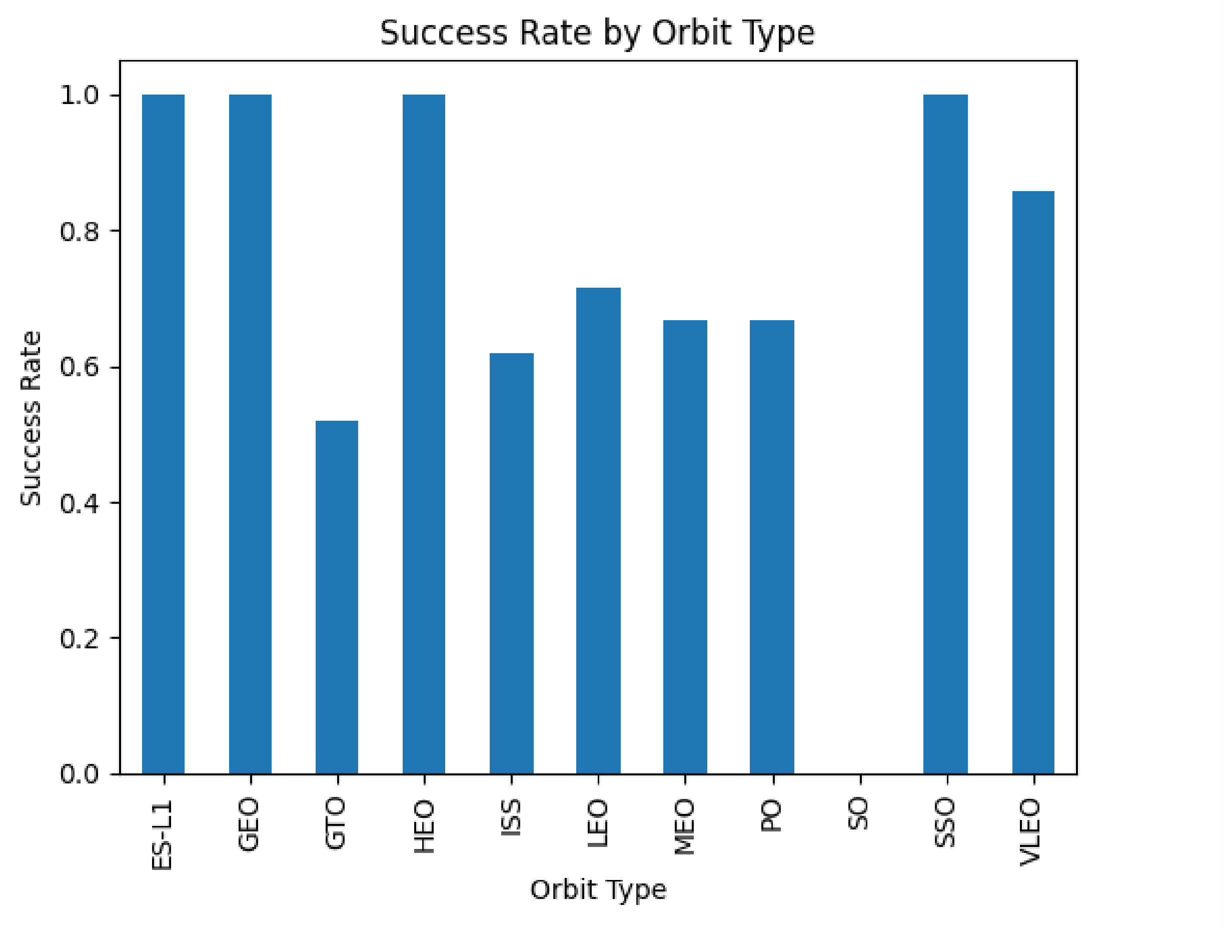
Payload vs. Launch Site



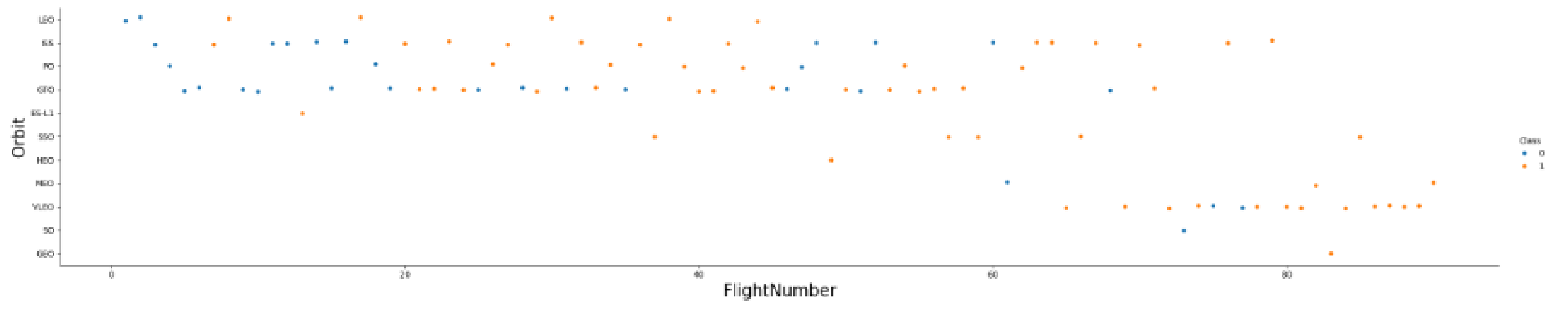
- Looking at the scatter plot of payload versus launch site, it can be seen that at the VAFB-SLC launch site, rockets with a heavy payload mass (over 10,000) have not been launched.

Success Rate vs. Orbit Type

- Analyze the ploted bar chart try to find which orbits have high sucess rate.

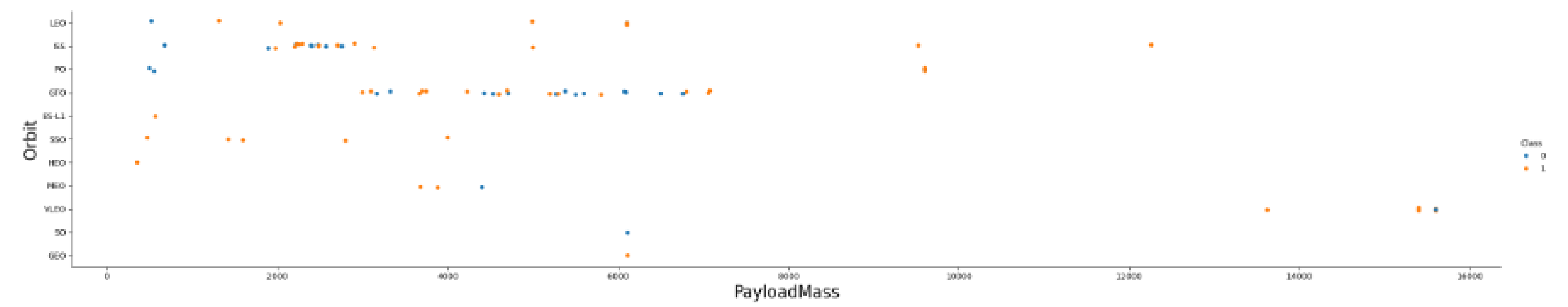


Flight Number vs. Orbit Type



- In LEO orbit, success seems to be related to the number of flights, but in GTO orbit, there appears to be no relation to the number of flights.

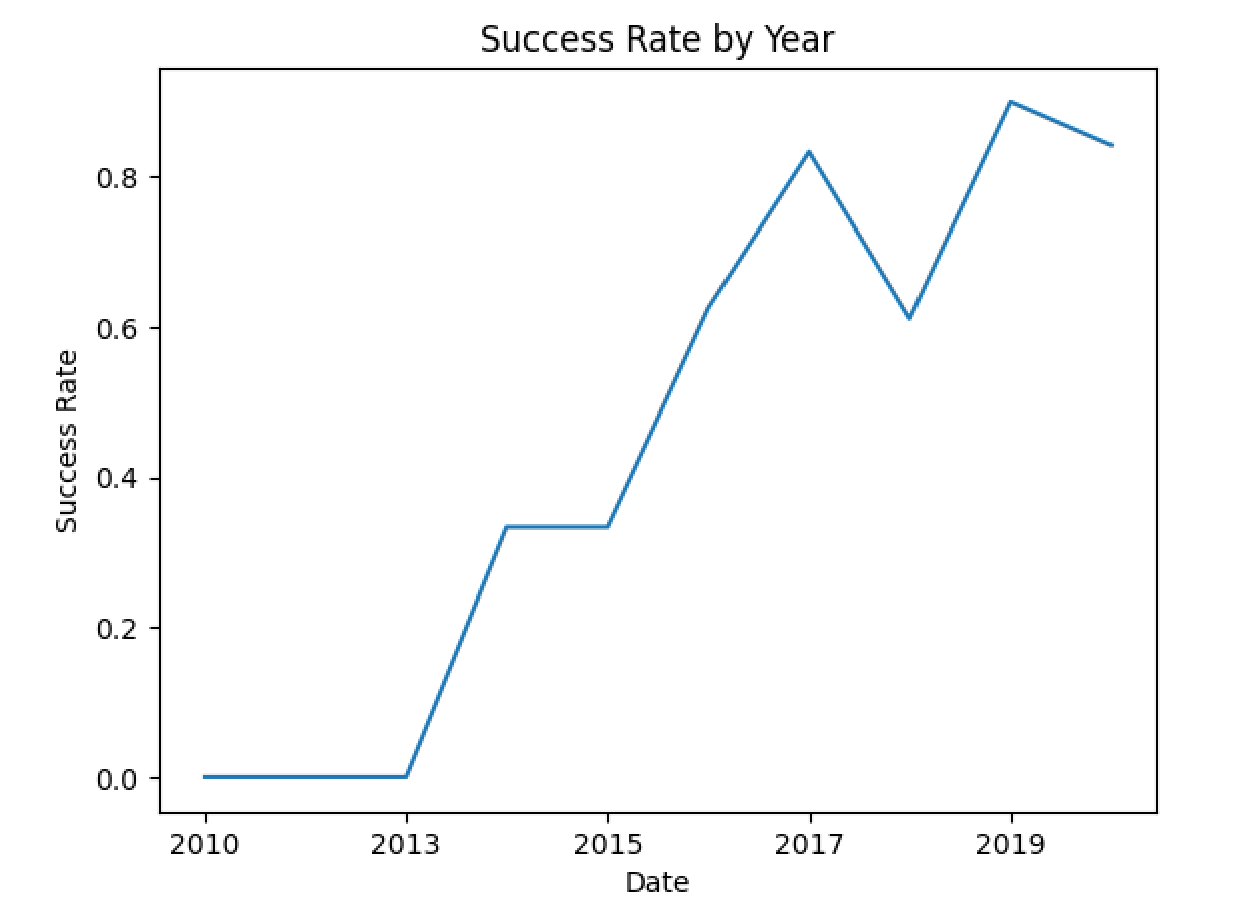
Payload vs. Orbit Type



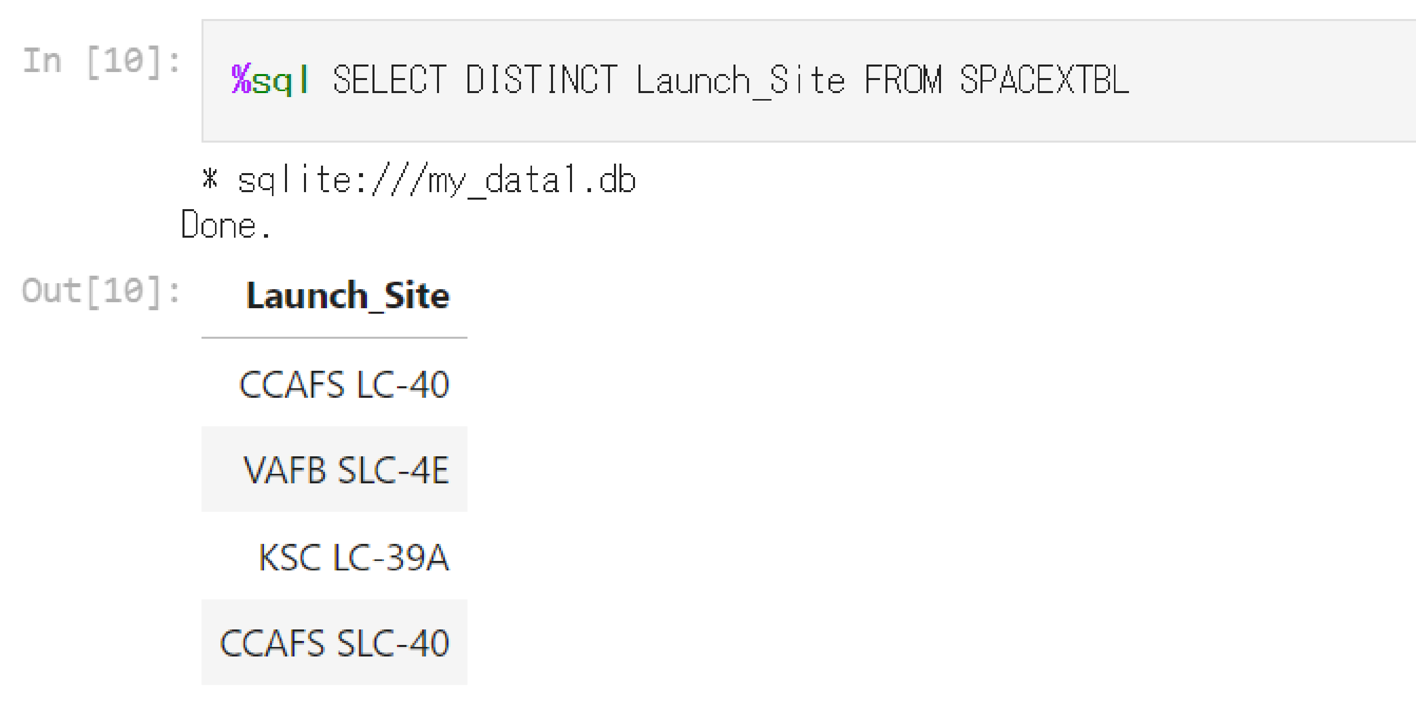
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

Launch Success Yearly Trend

- From 2013 to 2020, the success rate has been steadily increasing.



All Launch Site Names



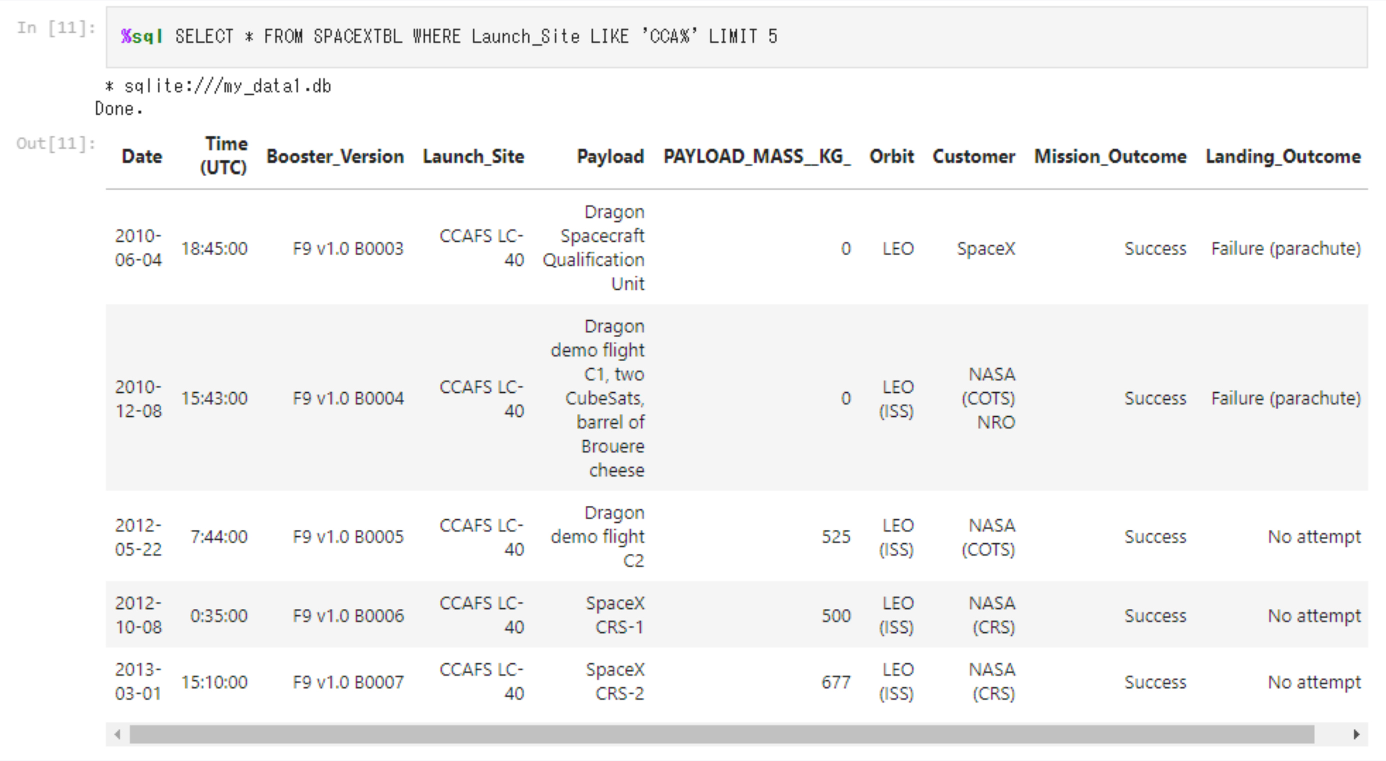
```
In [10]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTBL

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

Out[10]:

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

- Extract unique values from the 'Launch_Site' column using the DISTINCT method

Launch Site Names Begin with 'CCA'

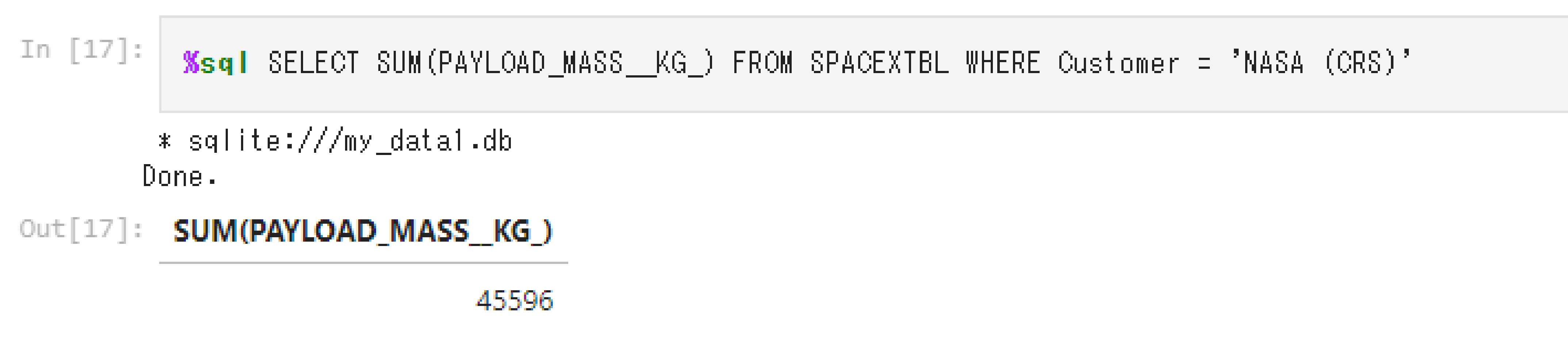


```
In [11]: %sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE ’CCA%’ LIMIT 5

 * sqlite:///my_data1.db
Done.
Out[11]:
```

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

- Select and display the first 5 rows from the SPACEXTBL table where the Launch_Site column starts with 'CCA’.

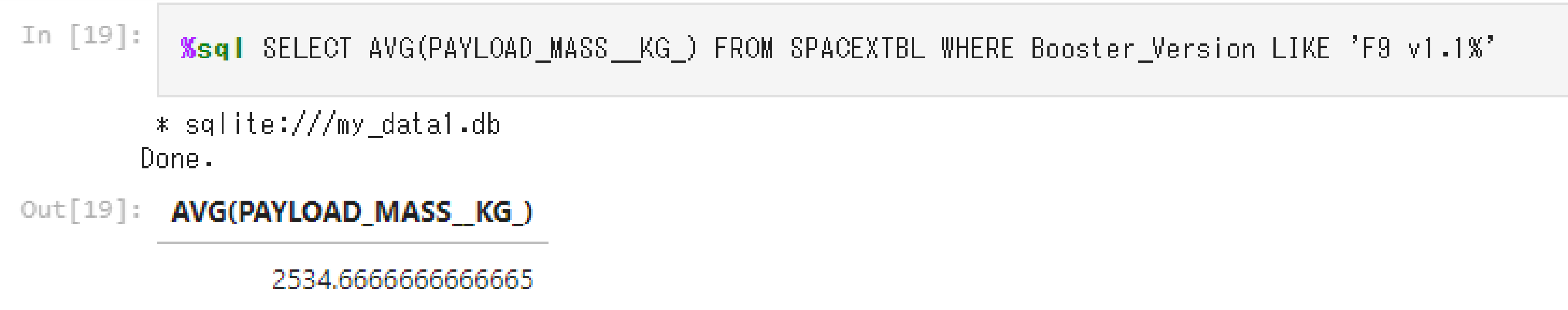
Total Payload Mass



```
In [17]:  %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'

 * sqlite:///my_data1.db
Done.

Out[17]:  SUM(PAYLOAD_MASS__KG_)
                         45596
```

- Calculate the total sum (total payload mass) of the PAYLOAD_MASS__KG_ column values for all rows in the SPACEXTBL table where the value of the Customer column is 'NASA (CRS)'.

Average Payload Mass by F9 v1.1



In [19]:

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE ’F9 v1.1%’
```

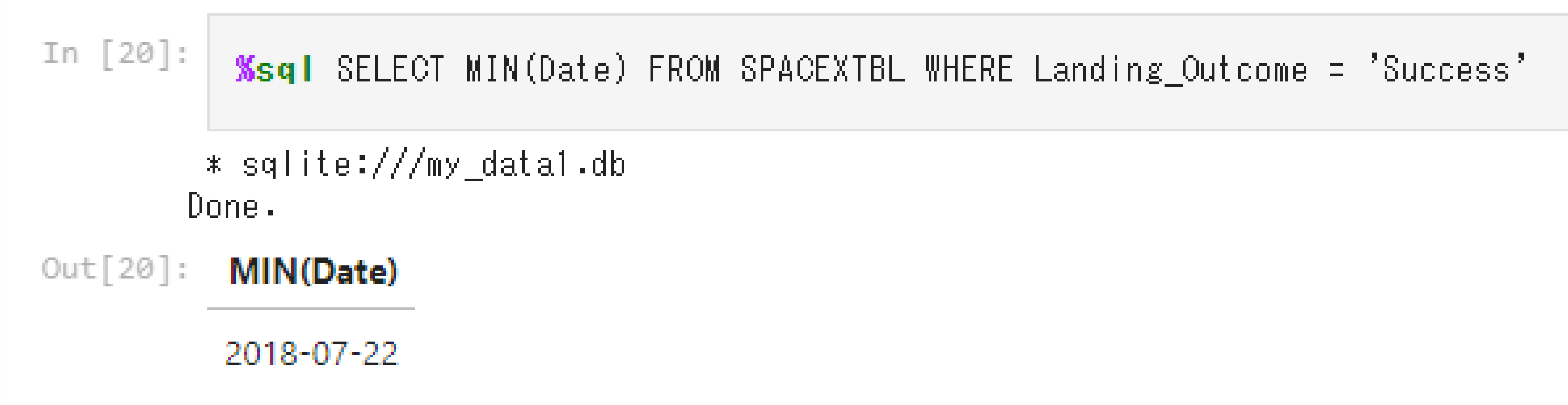
```
 * sqlite:///my_data1.db
Done.
```

Out[19]:

AVG(PAYLOAD_MASS__KG_)
2534.666666666665

- Calculate the average (mean payload mass) of the values in the PAYLOAD_MASS__KG_ column for all rows in the SPACEXTBL table whose value in the Booster_Version column begins with 'F9 v1.1’.

First Successful Ground Landing Date



```
In [20]: %sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success'
```

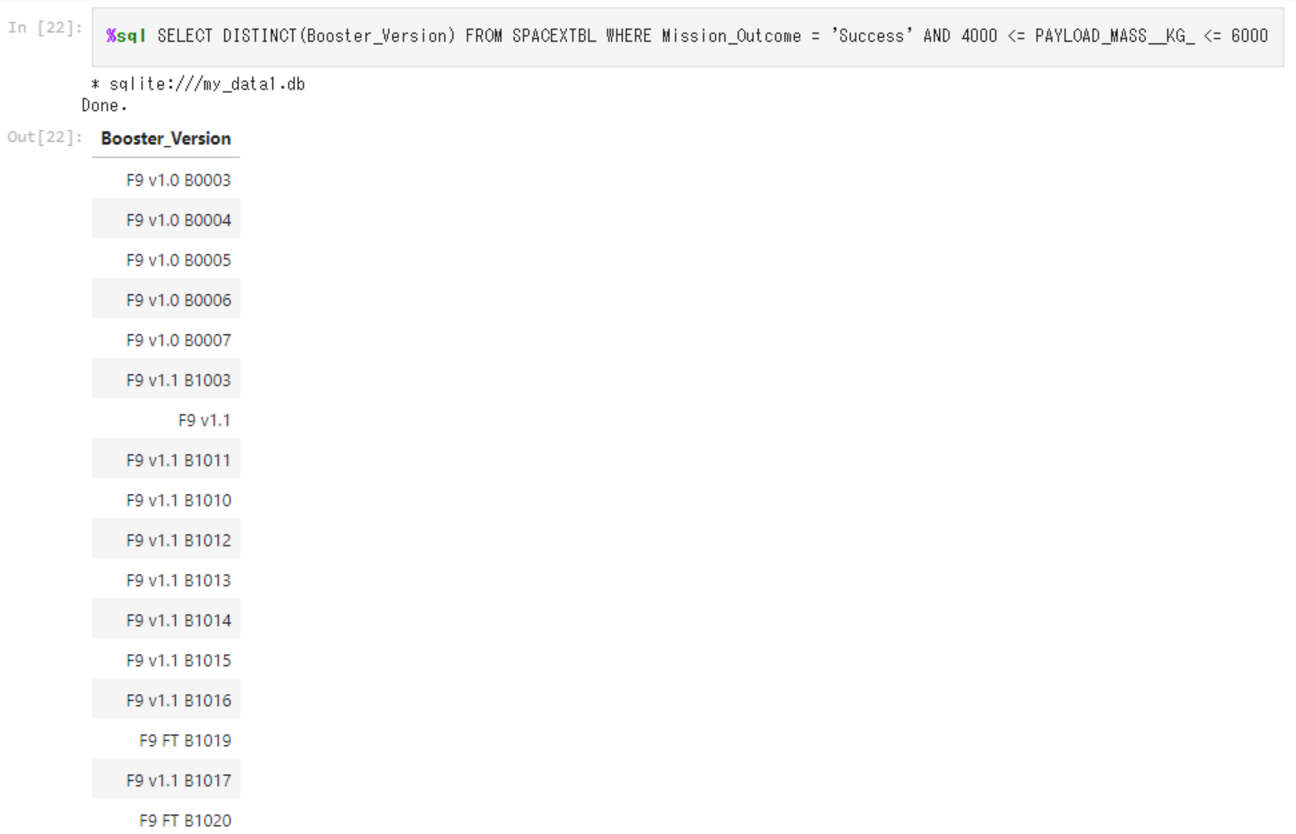
```
 * sqlite:///my_data1.db
Done.
```

Out[20]:

MIN(Date)
2018-07-22

- Find the earliest date (the date of the first successful landing) among all rows in the SPACEXTBL table where the value of the Landing_Outcome column is 'Success' (successful landing).

Successful Drone Ship Landing with Payload between 4000 and 6000



In [22]:

```
%sql SELECT DISTINCT(Booster_Version) FROM SPACEXTBL WHERE Mission_Outcome = 'Success' AND 4000 <= PAYLOAD_MASS__KG_ <= 6000
```

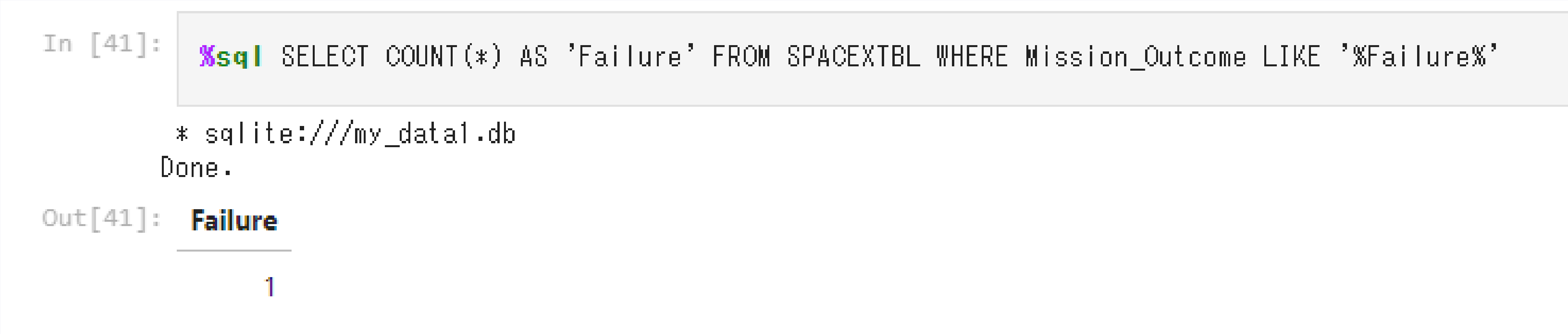
```
 * sqlite:///my_data1.db
Done.
```

Out[22]:

Booster_Version
F9 v1.0 B0003
F9 v1.0 B0004
F9 v1.0 B0005
F9 v1.0 B0006
F9 v1.0 B0007
F9 v1.1 B1003
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1010
F9 v1.1 B1012
F9 v1.1 B1013
F9 v1.1 B1014
F9 v1.1 B1015
F9 v1.1 B1016
F9 FT B1019
F9 v1.1 B1017
F9 FT B1020

- List all unique booster versions in the SPACEXTBL table that have a "successful" mission outcome and a payload mass in the range of 4000 kg to 6000 kg.

Total Number of Successful and Failure Mission Outcomes



```
In [41]: %sql SELECT COUNT(*) AS 'Failure' FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Failure%'
```

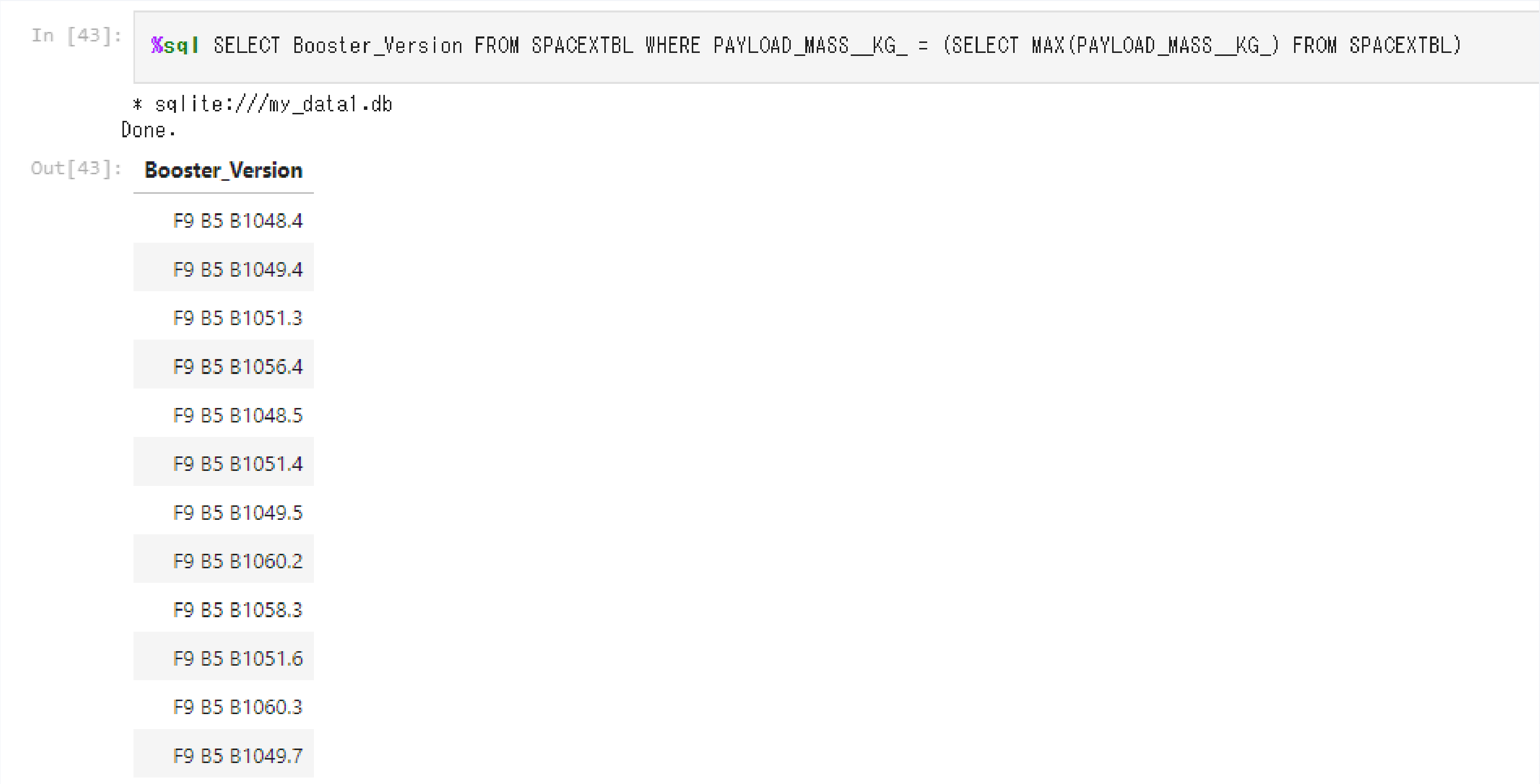
```
 * sqlite:///my_data1.db
Done.
```

Out[41]:

Failure
1

- Count the number of all rows in the SPACEXTBL table that contain 'Failure' in the Mission_Outcome column.

Boosters Carried Maximum Payload



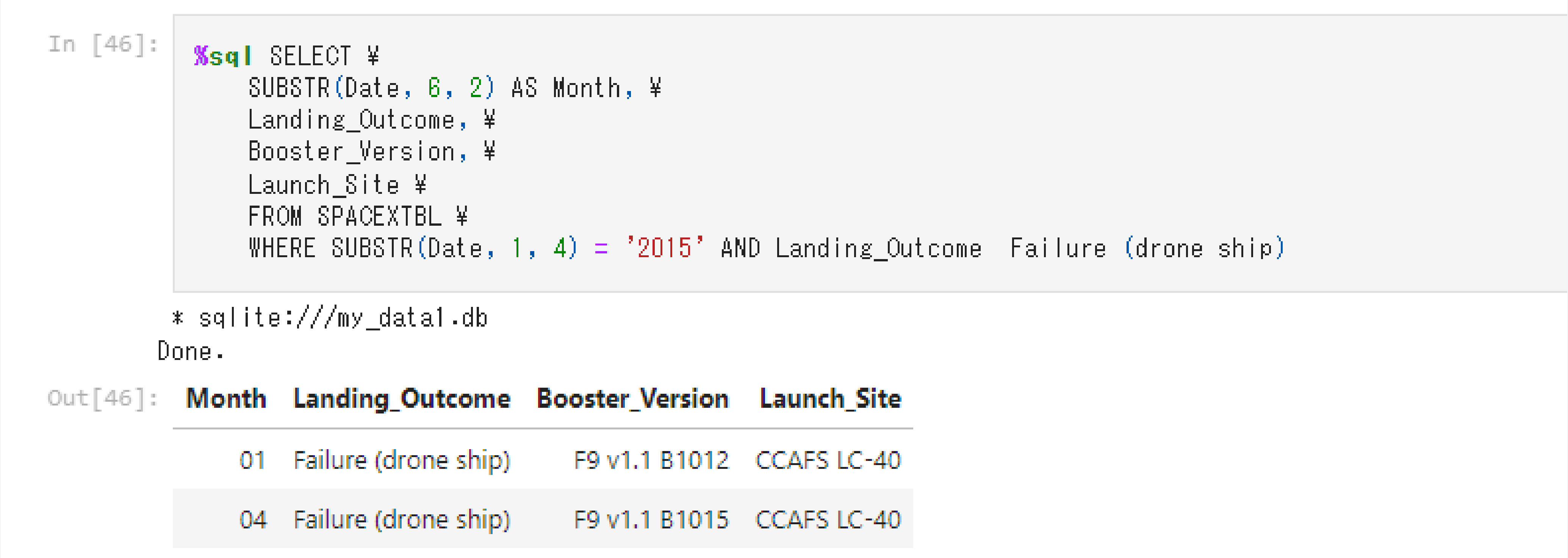
```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * 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

- Identify the booster version of the mission with the heaviest (largest) payload mass in the SPACEXTBL table.

2015 Launch Records



In [46]:

```
%sql SELECT ¥
    SUBSTR(Date, 6, 2) AS Month, ¥
    Landing_Outcome, ¥
    Booster_Version, ¥
    Launch_Site ¥
    FROM SPACEXTBL ¥
    WHERE SUBSTR(Date, 1, 4) = '2015' AND Landing_Outcome  Failure (drone ship)
```

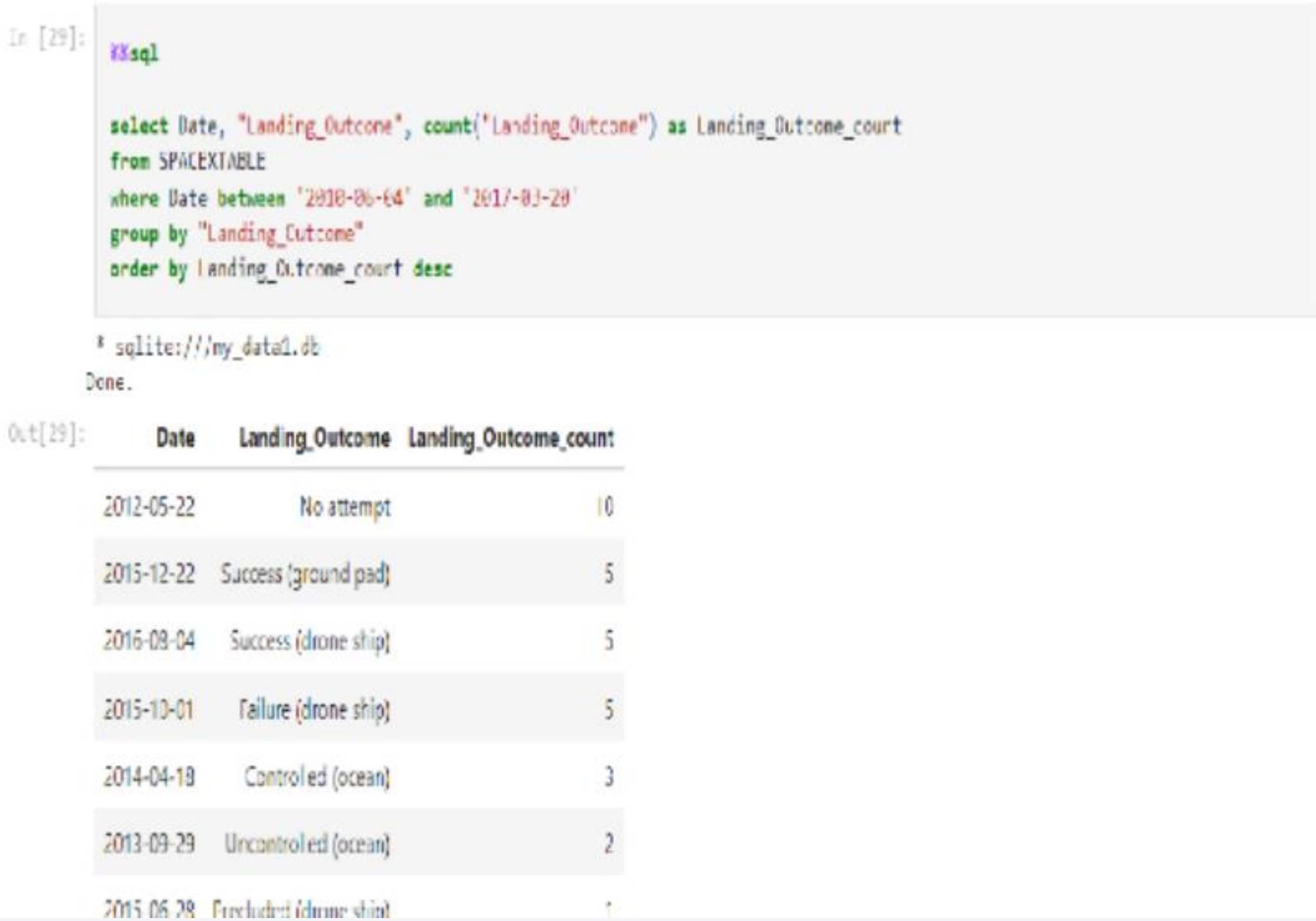
```
 * sqlite:///my_data1.db
Done.
```

Out[46]:

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

- Information on failed drone ship landing missions in 2015 (month, landing outcome, booster version, launch site) selected from the SPACEXTBL table.

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



In [29]:

```
%%sql

select Date, "Landing_Outcome", count("Landing_Outcome") as Landing_Outcome_count
from SPACEXTABLE
where Date between '2010-06-04' and '2017-03-20'
group by "Landing_Outcome"
order by Landing_Outcome_count desc
```

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

Out[29]:

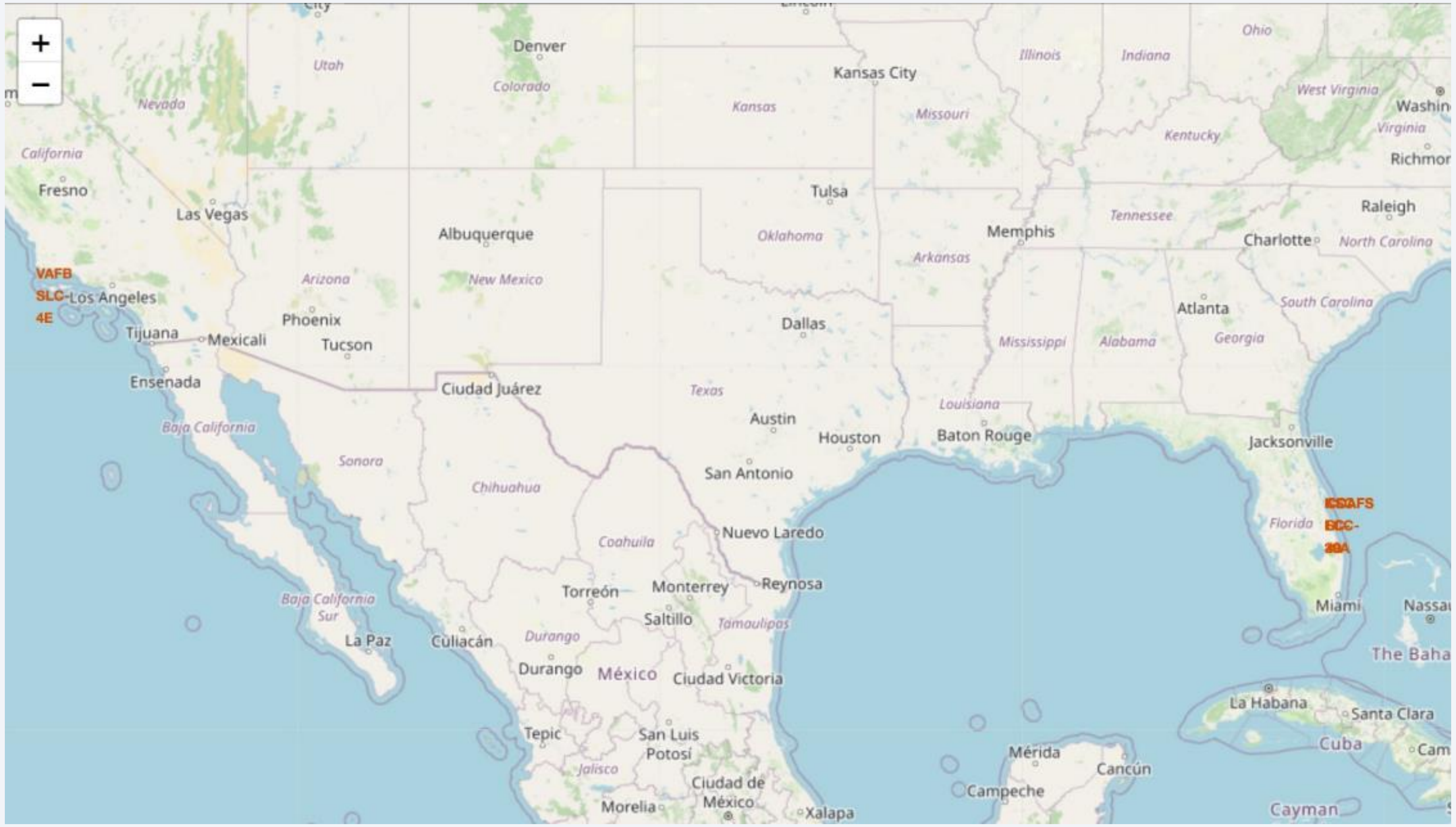
Date	Landing_Outcome	Landing_Outcome_count
2012-05-22	No attempt	10
2015-12-22	Success (ground pad)	5
2016-08-04	Success (drone ship)	5
2015-10-01	Failure (drone ship)	5
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2015-06-28	Precluded (drone ship)	1

- 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,

Section 3

Launch Sites Proximities Analysis

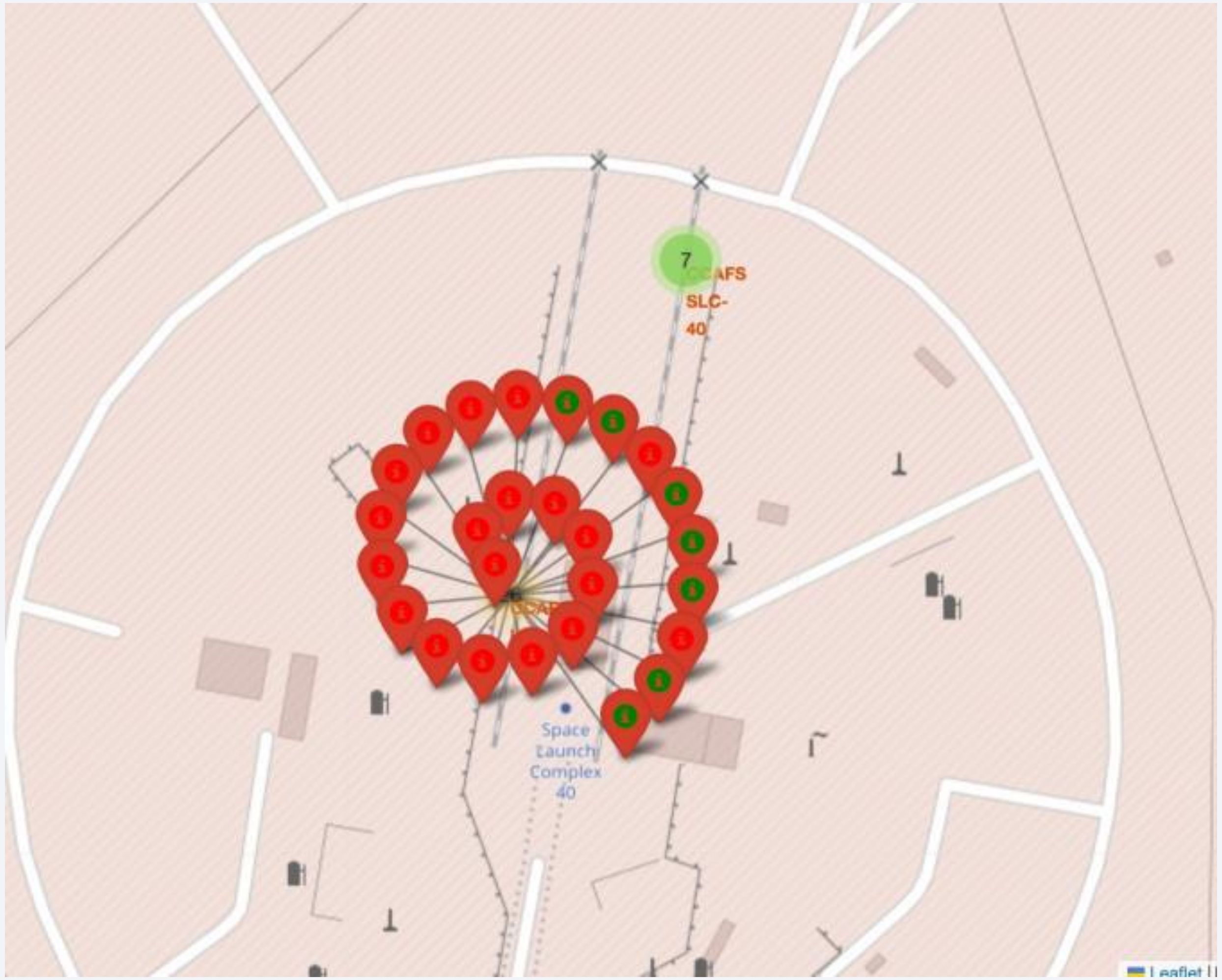
<All launch sites location markers on global map>



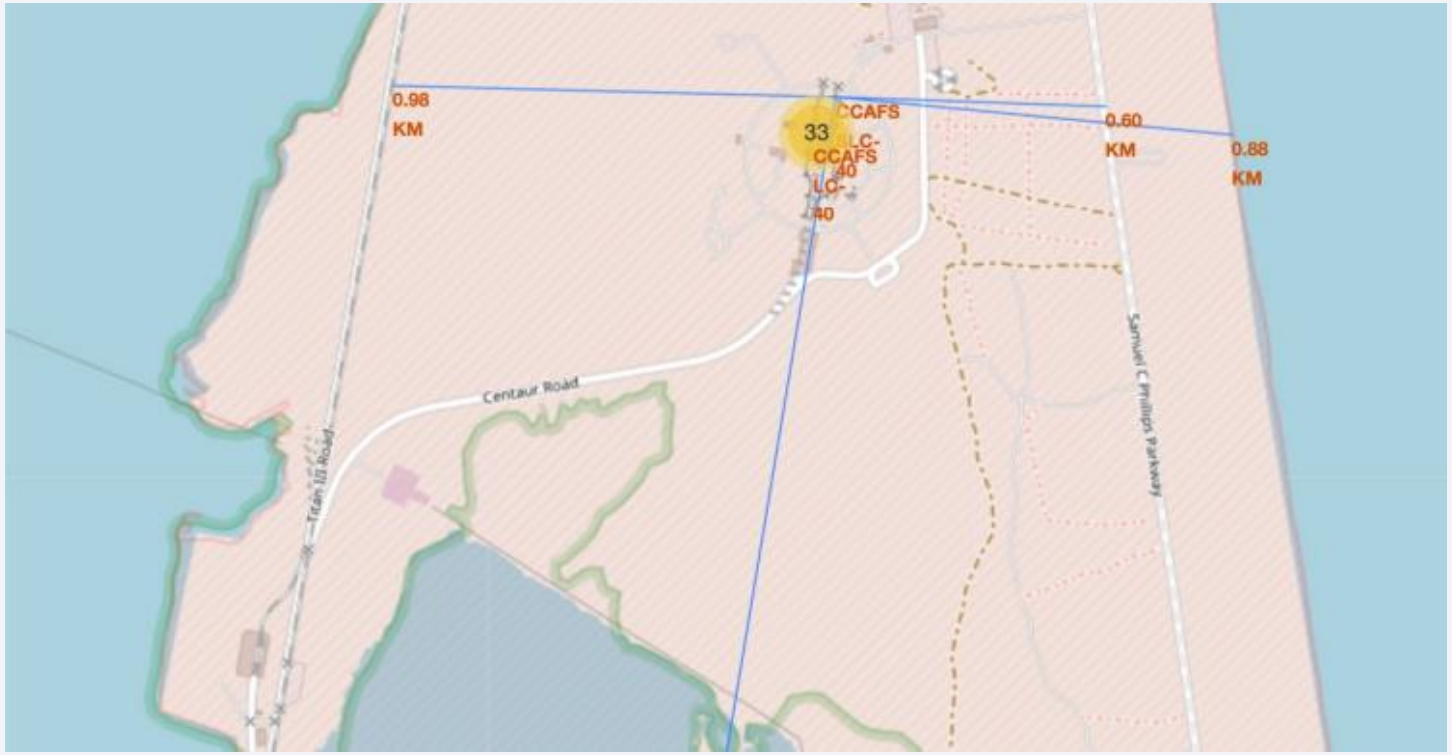
- All launch sites in very close proximity to the coast

<Color-labeled launch outcomes>

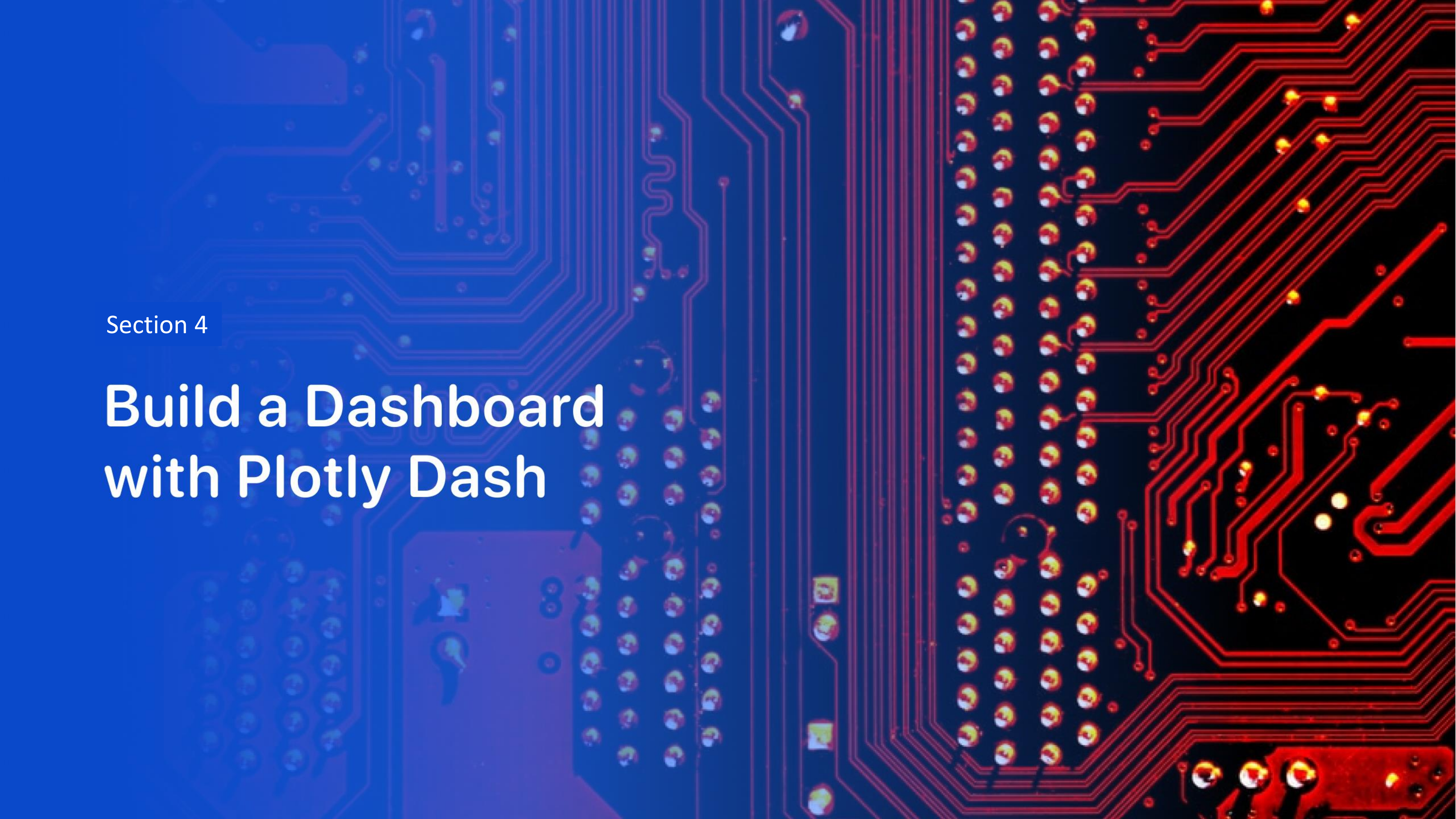
- the number of launches for each launch sites



<Distance calculated to railway, highway, coastline>



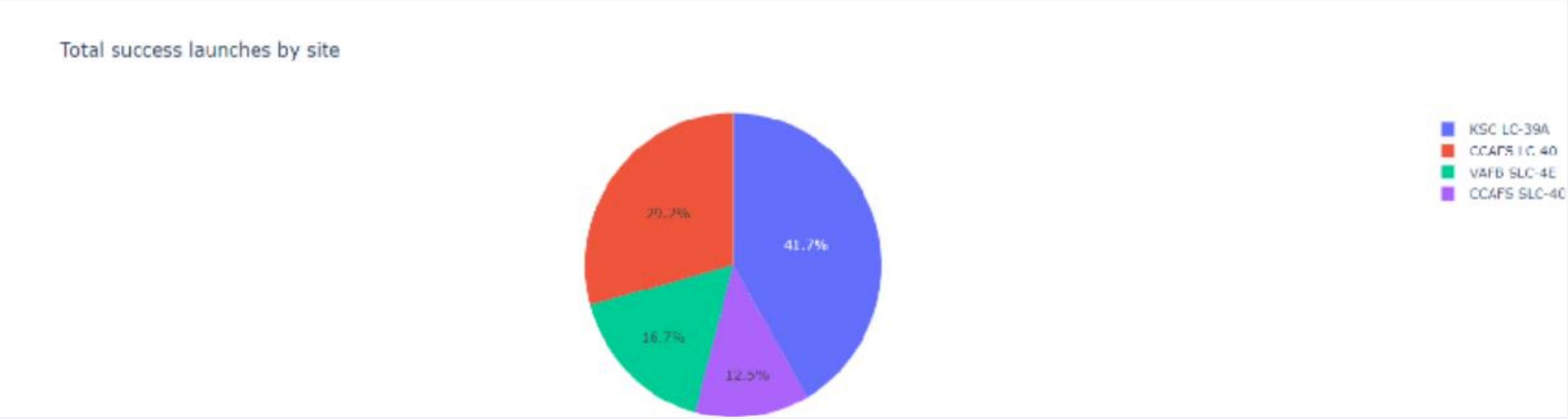
- The launch site is very close to railway, highway, and coastline



Section 4

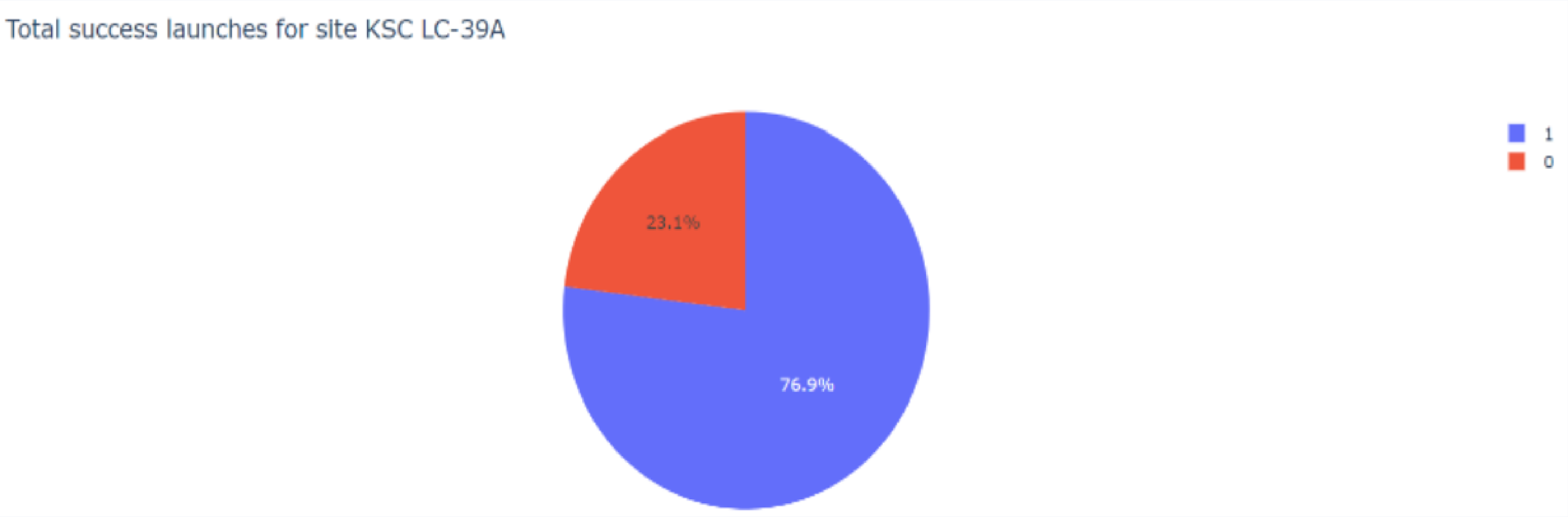
Build a Dashboard with Plotly Dash

<Total Success Launches by Site>



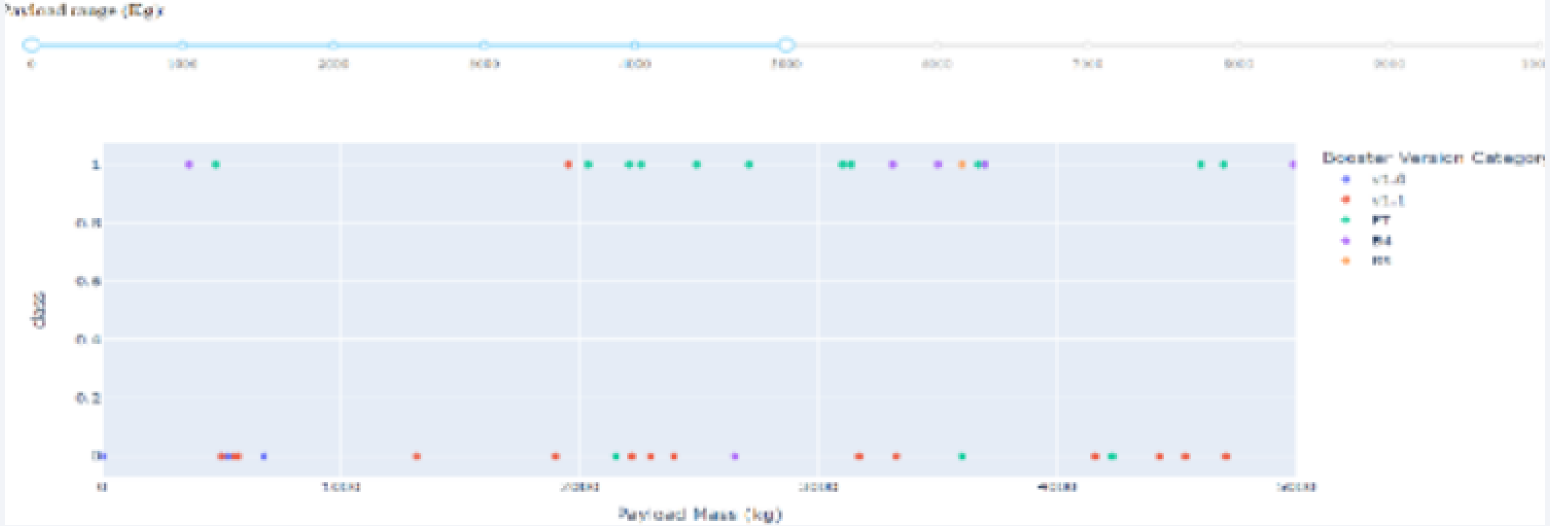
- KSC LC-39A has highest launches.

<Total Success KSC LC-39A>



- KSC LC-39A has high score of total success.

<Payload vs. Launch Outcome>



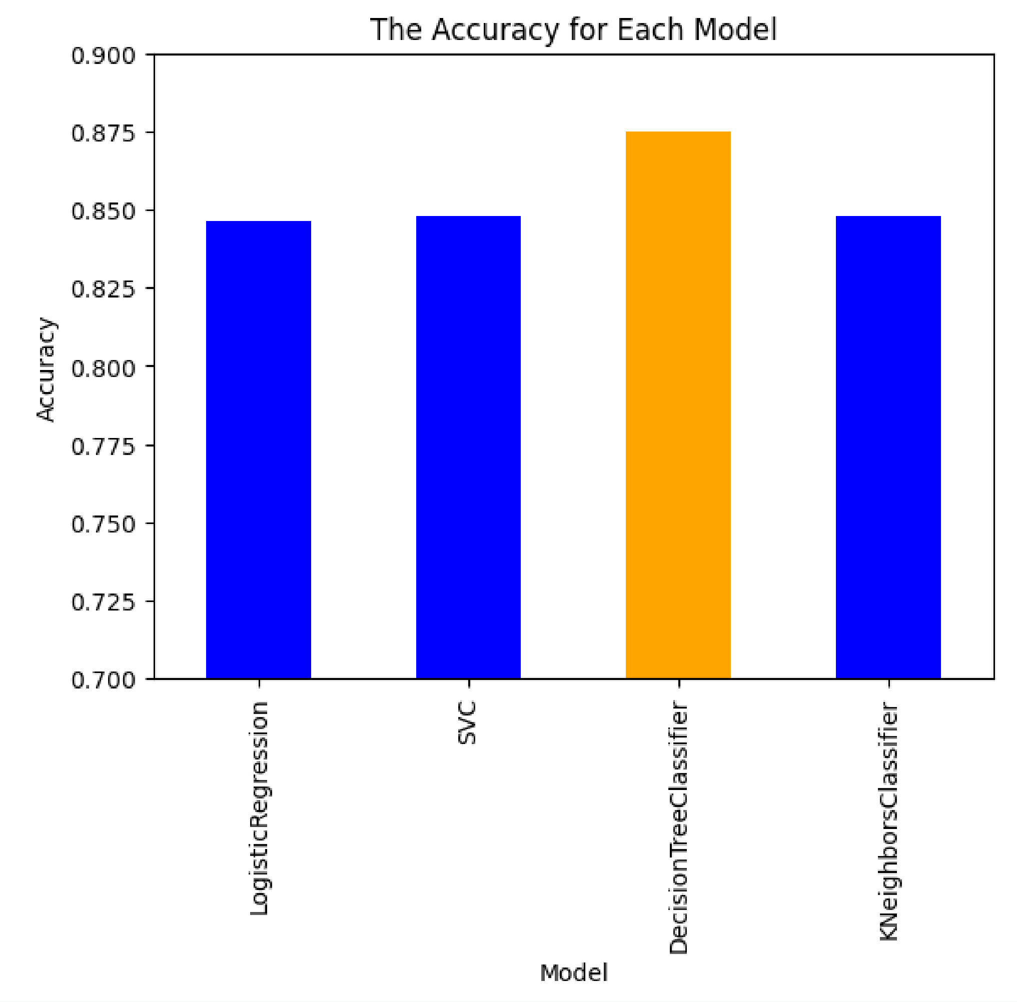
- Payload and Lauch are greate factors from high score of total success.

Section 5

Predictive Analysis (Classification)

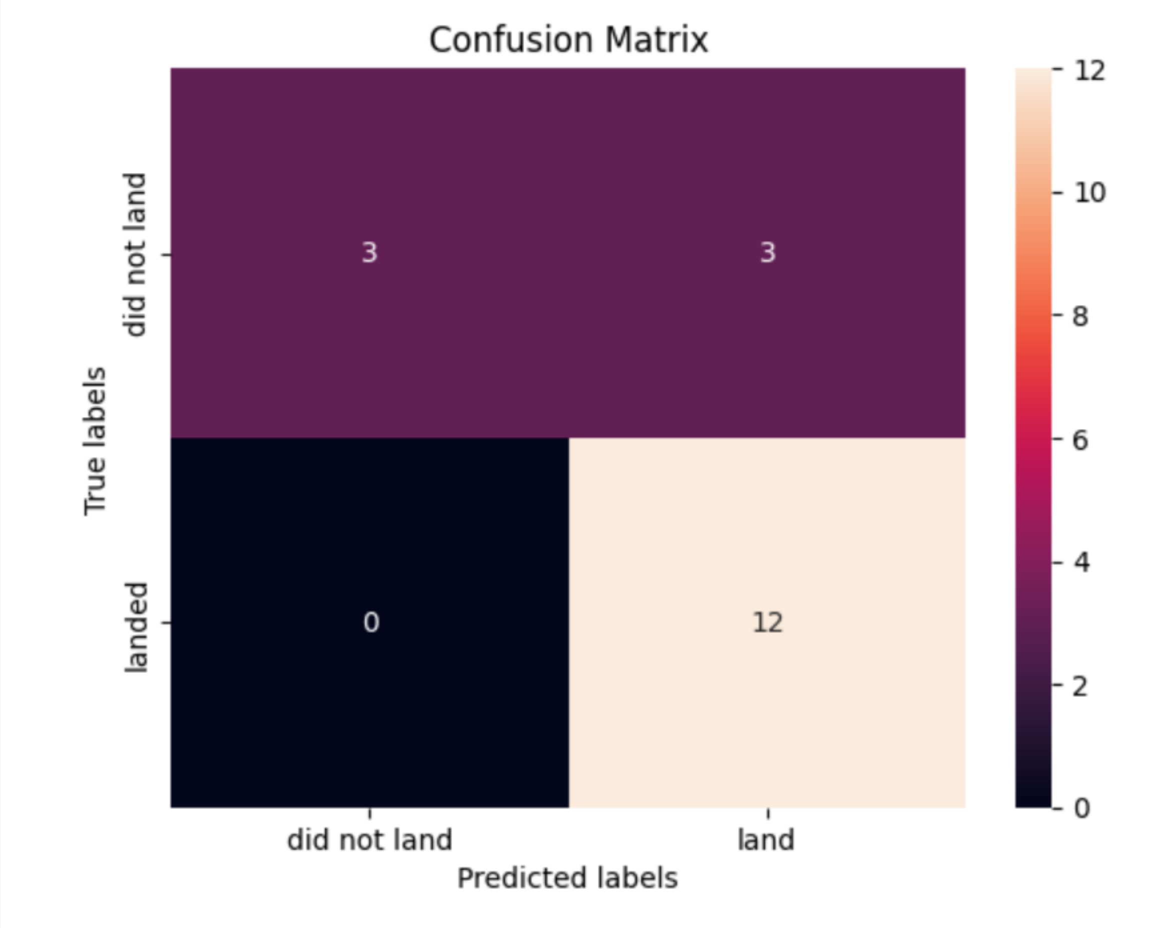
Classification Accuracy

- The best model is the Decision Tree Classifier.



Confusion Matrix

- The confusion matrix indicates the accuracy of the model.



Conclusions

- Develop a machine learning model for SpaceY, who wants to bid for SpaceX.
- To save a significant amount of money by predicting when Stage 1 will be successful.
- Insufficient data was obtained to deem the model useful. Therefore, it is recommended to consider alternative approaches or wait for more data to arrive at a useful conclusion.

Thank you!