IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Abdelrahman Gaber

20th, November 2023

Outline

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

Executive Summary

- In the SpaceY project, our primary objective was to predict the landing status of the most costly part in our rocket launches, facilitating its potential reuse in future endeavors. We utilized a combination of web scraping and open-source APIs to collect diverse data, which was then processed to categorize outcomes as success or failure. Through thorough cleaning and analysis, we identified key factors influencing successful landings, including the launch site, payload mass, and designated orbit.
- To predict success rates in future launches, we trained multiple machine learning models, ultimately determining that the Decision Tree model outperformed others. The decision to favor this model was based on [mention specific criteria or metrics]. This project's success enables the rocket's team to make informed decisions and allocate resources more efficiently. Moving forward, we anticipate continued accuracy in predicting launch outcomes, with potential for ongoing model refinement as new data becomes available.

Introduction

Our primary objective is to leverage data analysis to extract valuable insights concerning rocket launches at SpaceY. Our overarching goal is to optimize resource utilization by determining the likelihood of successfully retrieving and reusing the first stage of each launch. Notably, we have achieved successful landings for certain first stages, prompting us to conduct a comprehensive analysis across numerous launches. The focus of this analysis is to derive data-driven probabilities and insights, enabling us to make informed decisions on whether a particular launch's first stage is likely to land successfully on our platforms for potential reuse.

To achieve this, our approach involves collecting and analyzing data from multiple launches. We will employ robust preprocessing techniques to enhance the quality of our dataset, followed by thorough visualization to discern potential relationships among various features. The ultimate aim is to harness machine learning models that can predict, based on historical data, the probabilities associated with a successful landing for new launches. This predictive capability will significantly contribute to our strategic planning and resource optimization efforts.

Section 1

Methodology

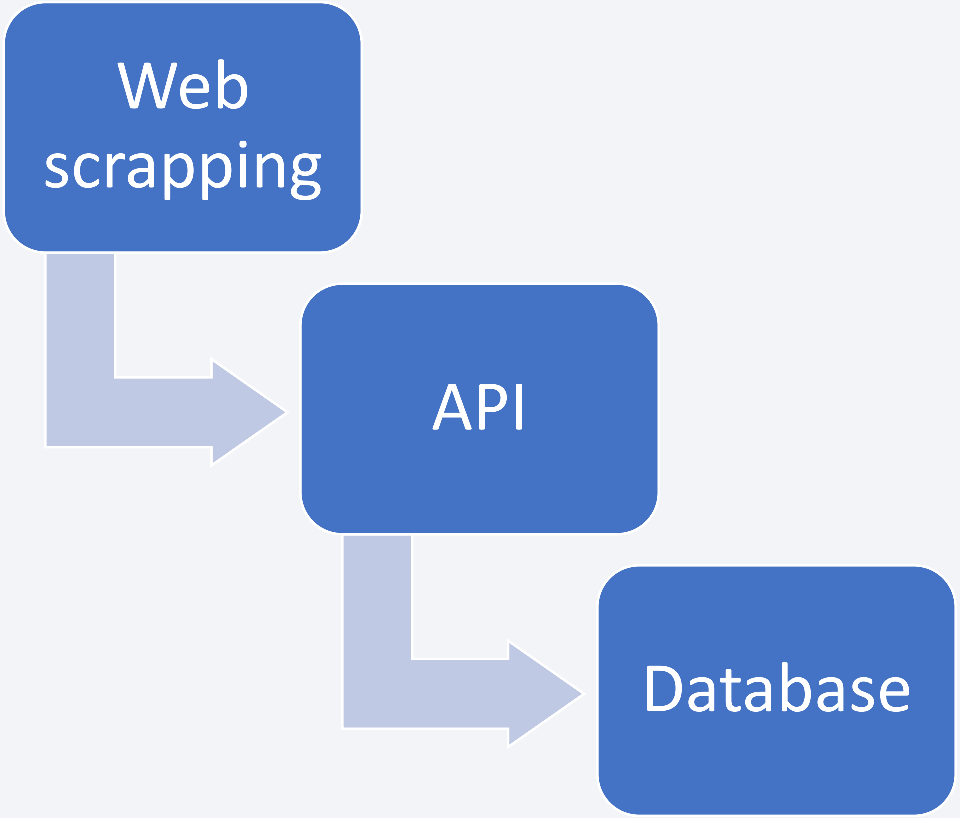
Methodology

Executive Summary

- Data collection methodology:
    - Web scrapping and open source APIs
- Perform data wrangling
    - Removing nulls and one hot encoding
- 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 Gridsearch and multiple models to get best results.

Data Collection

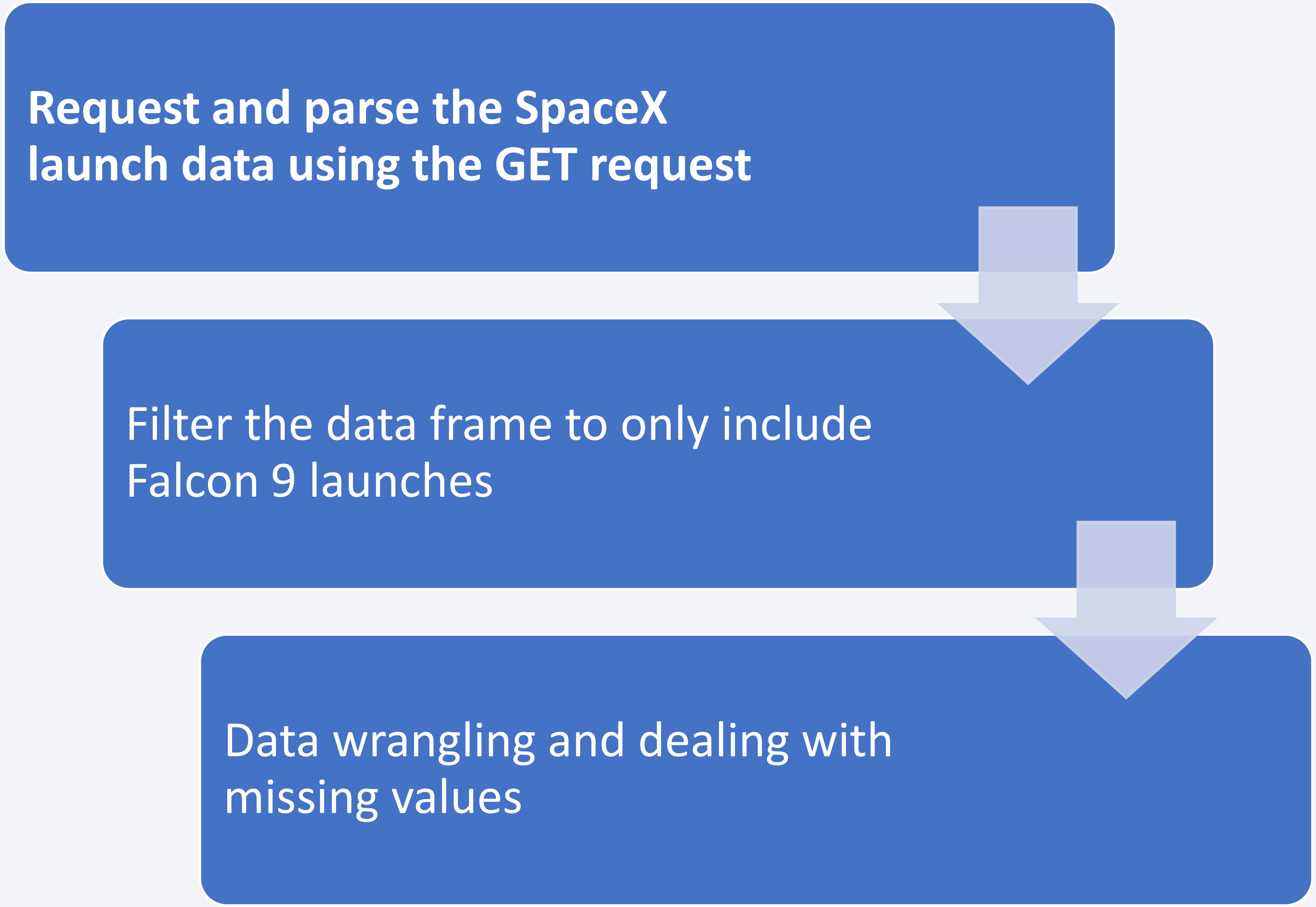
1. Web scrapping
   - We got the initial data about the rocket launches such as rocket IDs.
2. API
   - From the rocket IDs obtained from web scrapping we got the remaining information about those launches.
3. Saving in database
   - Saving the data into our database for further manipulation and cleaning.



Data Collection – SpaceX API

- Presenting the data collection using the spaceX REST API.

- GitHub link for full REST API notebook: https://github.com/gaber16/SpaceX-IBM-Data-Science-Final-project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



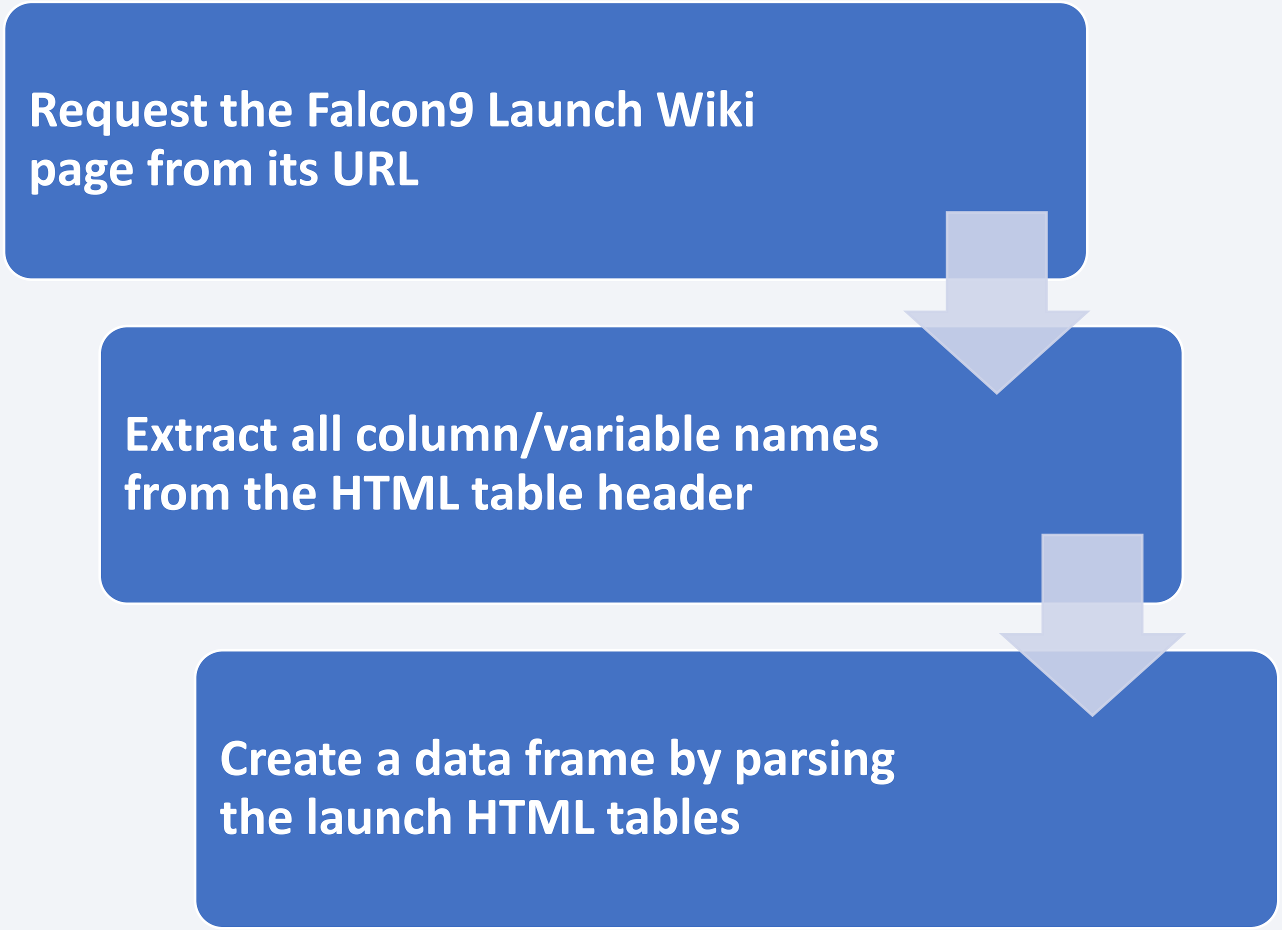
Request and parse the SpaceX launch data using the GET request

Filter the data frame to only include Falcon 9 launches

Data wrangling and dealing with missing values

Data Collection - Scraping

- Using requests and beautifulsoup libraries.
- Web scrapping from Wikipedia and saving the data into a pandas data frame.
- GitHub link: https://github.com/gaber16/SpaceX-IBM-Data-Science-Final-project/blob/main/jupyter-labs-webscraping.ipynb



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

Data Wrangling

- Our data did not need much wrangling and preprocessing as it was saved in the spaceX API and relatively ready for use.
- We checked for duplicates and null data in sensitive columns and replaced the nulls with the mean of the column.
- This part can be found at the end of the data collection using spaceX API.

EDA with Data Visualization

- Scatter plot charts:

    1. Payload mass and orbit vs outcome: conveys the best payload mass for each orbit to get a successful landing.
    2. Payload mass vs outcome: conveys what is the best payload mass range for a successful landing.
    3. Payload mass and launch site vs outcome: conveys information about the different launch site’s best payload mass for a rocket to safely make a land.

- Bar chart for the orbit vs the outcome: conveys the probability of a safe landing for each orbit.

- Line chart for outcome vs year: conveys that the probability of a successful landing increases from 2013 until 2020.

- GitHub link: https://github.com/gaber16/SpaceX-IBM-Data-Science-Final-project/blob/main/module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- SQL commands:
  - Display unique launch sites.
  - Display average payload size of a specific launch
  - Display the date of the first successful landing on a ground pad
  - Display best booster version's for a drone ship with a specific payload mass.
  - Display total number of successful and failed landings.
  - Rank the outcomes in a descending order to check for most common lands.
- GitHub link: https://github.com/gaber16/SpaceX-IBM-Data-Science-Final-project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Map objects:
  1. Markers
  2. Circles
  3. Lines
  4. MarkerClusters
- Those map objects let us view the launch sites of the rockets and how many successful landings for each site this indicating better areas.
- GitHub link: https://github.com/gaber16/SpaceX-IBM-Data-Science-Final-project/blob/main/module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

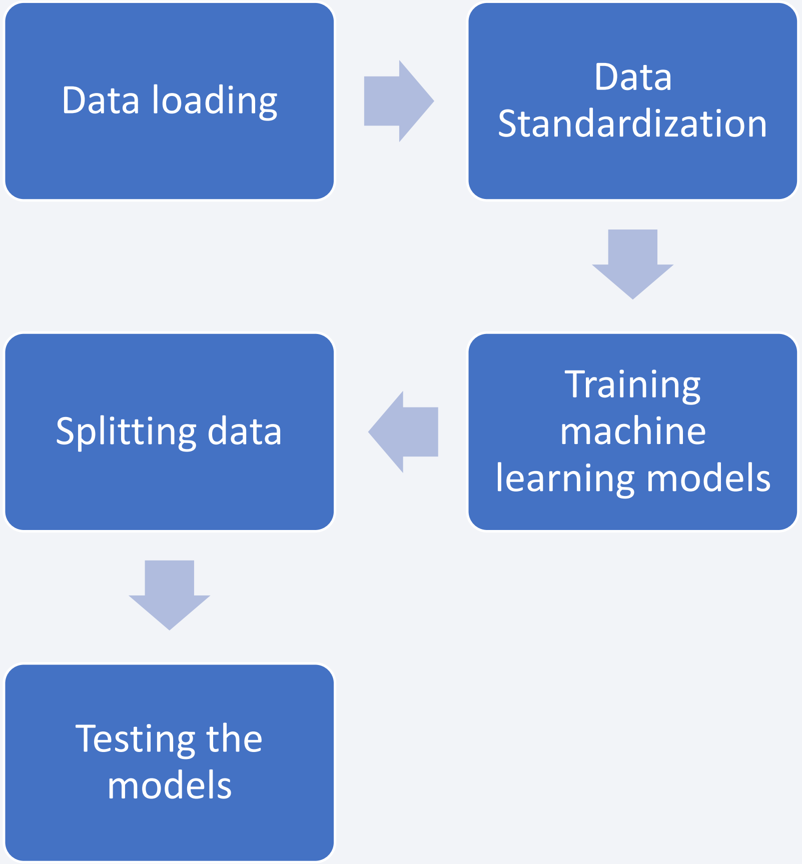
- We added pie charts and scatter plots to the dashboard:
  - Pie chart is for each launch sites with the outcome of the landing and one pie chart comparing all launch sites and their success rates.
  - Scatter plot is for the booster version for every rocket launched from a specified site with the outcome as the label.
- GitHub link: https://github.com/gaber16/SpaceX-IBM-Data-Science-Final-project/blob/main/module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Predictive Analysis (Classification)

- We build several machine learning models such as Logistic regression, Decision trees, Support Vector Machines, and K-nearest-neighbor.

  We split the data 80% for training and 20% for testing and train the models using Gridsearch in sklearn library to get the best hyper parameters for a given model.

  Finally, we evaluate the models using the test samples and the classification metrics such as Accuracy score, Jaccard index, Log-Loss, and F1 score.
- GitHub link: https://github.com/gaber16/SpaceX-IBM-Data-Science-Final-project/blob/main/module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb



Data loading

Data Standardization

Training machine learning models

Splitting data

Testing the models

Results

- Data analysis showed that there are relations between the dependent variable and the independent ones providing insights that applying machine learning models would yield decent results.
- The table below shows the results from our models after testing them on the test data:

Machine Learning Model	Best Score	Best Model parameters
Logistic Regression	0.834	'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'
Decision Trees	0.944	criterion: gini, max_depth: 8, max_features: sqrt, min_samples_leaf: 1, min_samples_split: 10, splitter: best
K Nearest Neighbors	0.834	algorithm: auto, n_neighbors: 10, p: 1
Support Vector Machines	0.834	C: 1.0, gamma: 0.03162277660168379, kernel: sigmoid

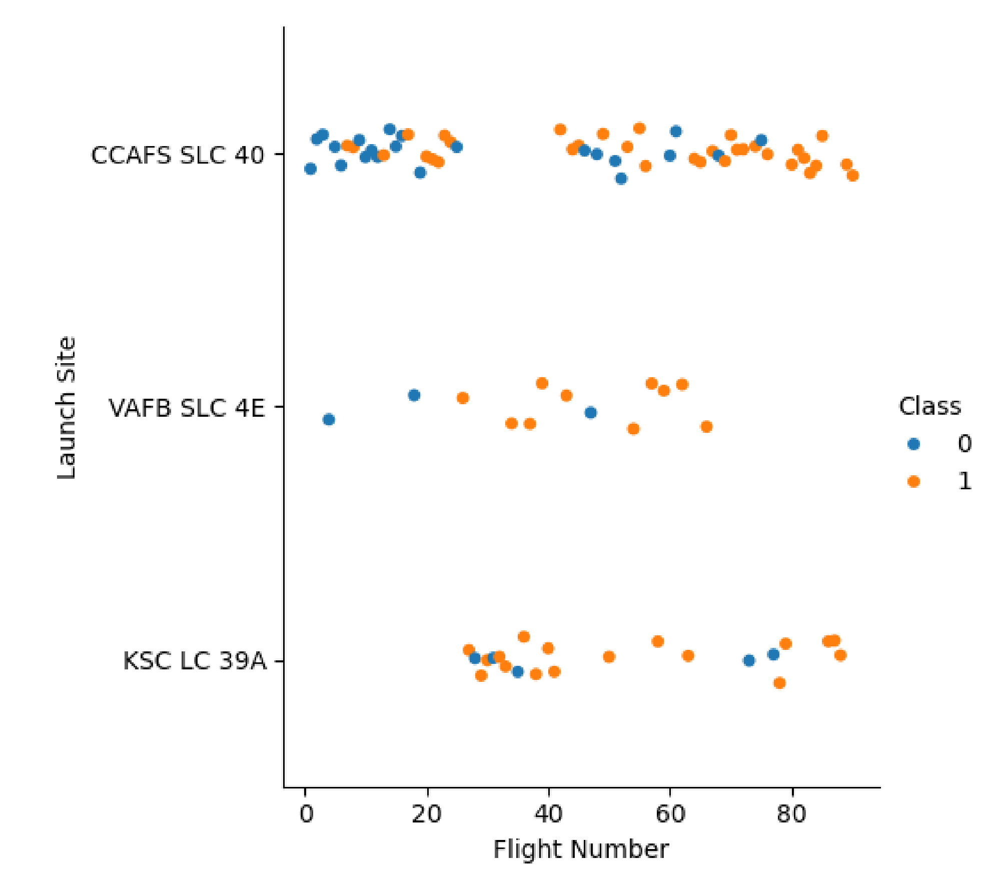
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site

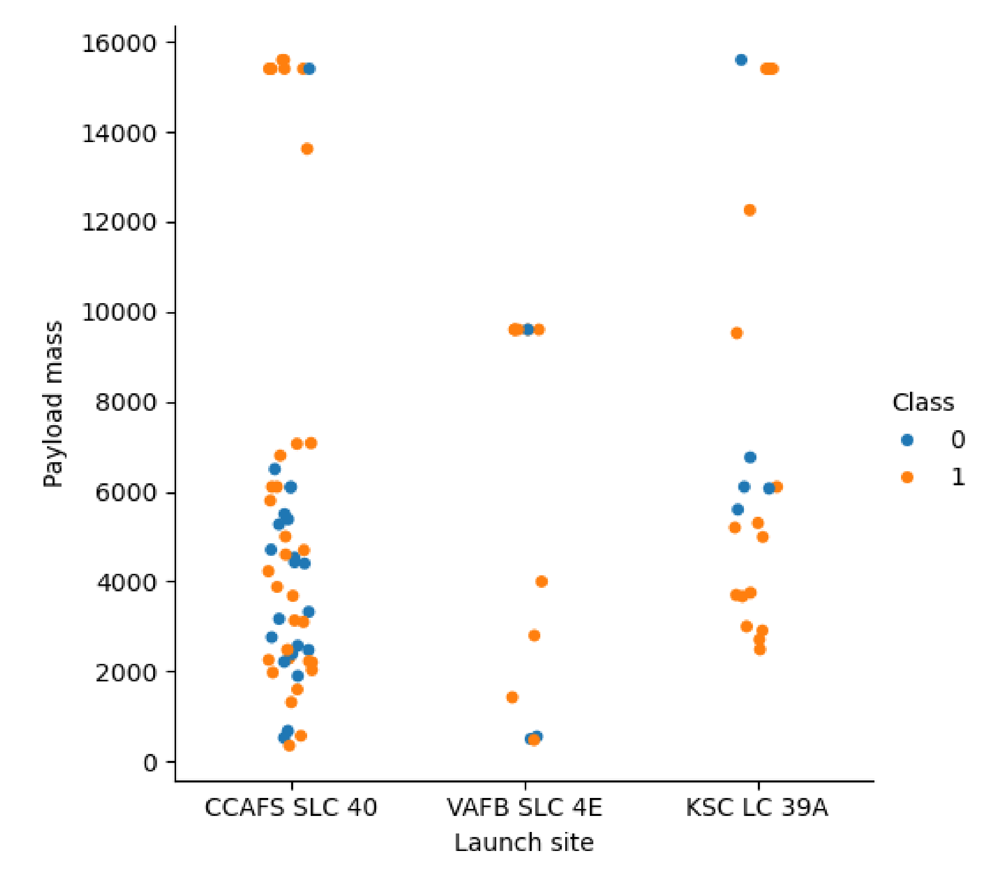
- It shows for the VAFB and KSC sites that the success rate increases beyond a certain number of flights. In this case flight number 40.



Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site

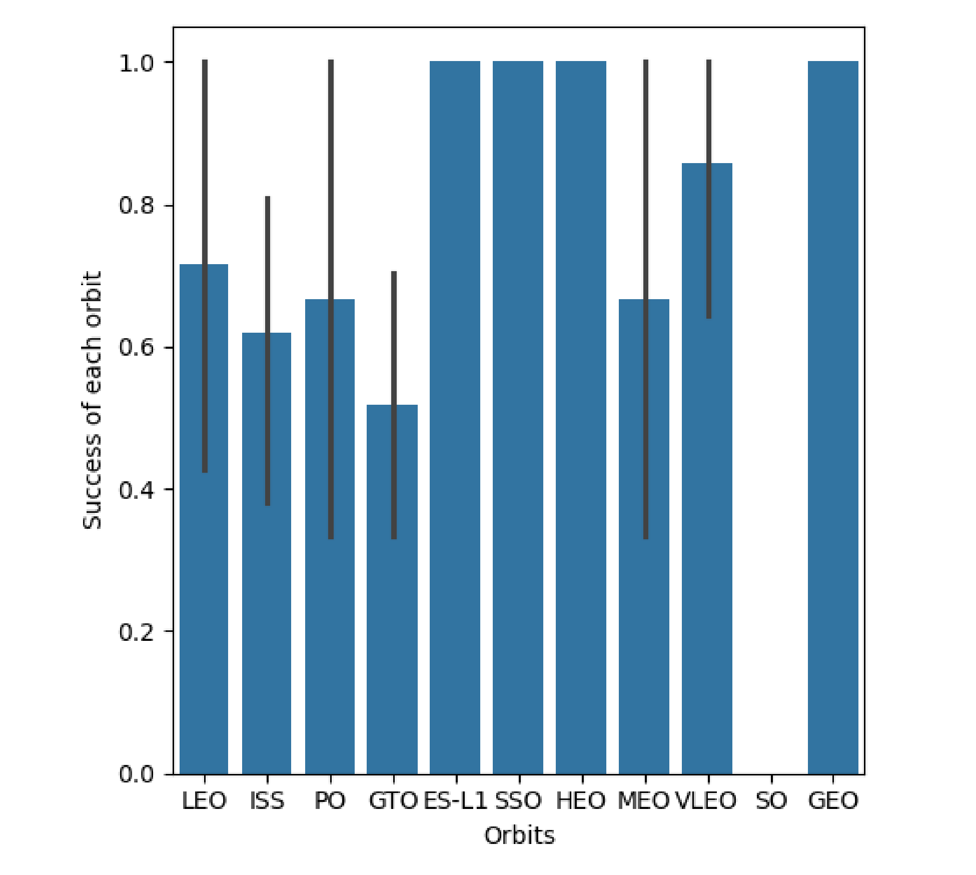
- The VAFB site does not have launches greater than 10k in payload mass
- CCAFS and KSC sites’ success rate is higher for bigger payload in mass than other payloads, beyond the 10k mark.



Success Rate vs. Orbit Type

- Bar chart for the success rate of each orbit type

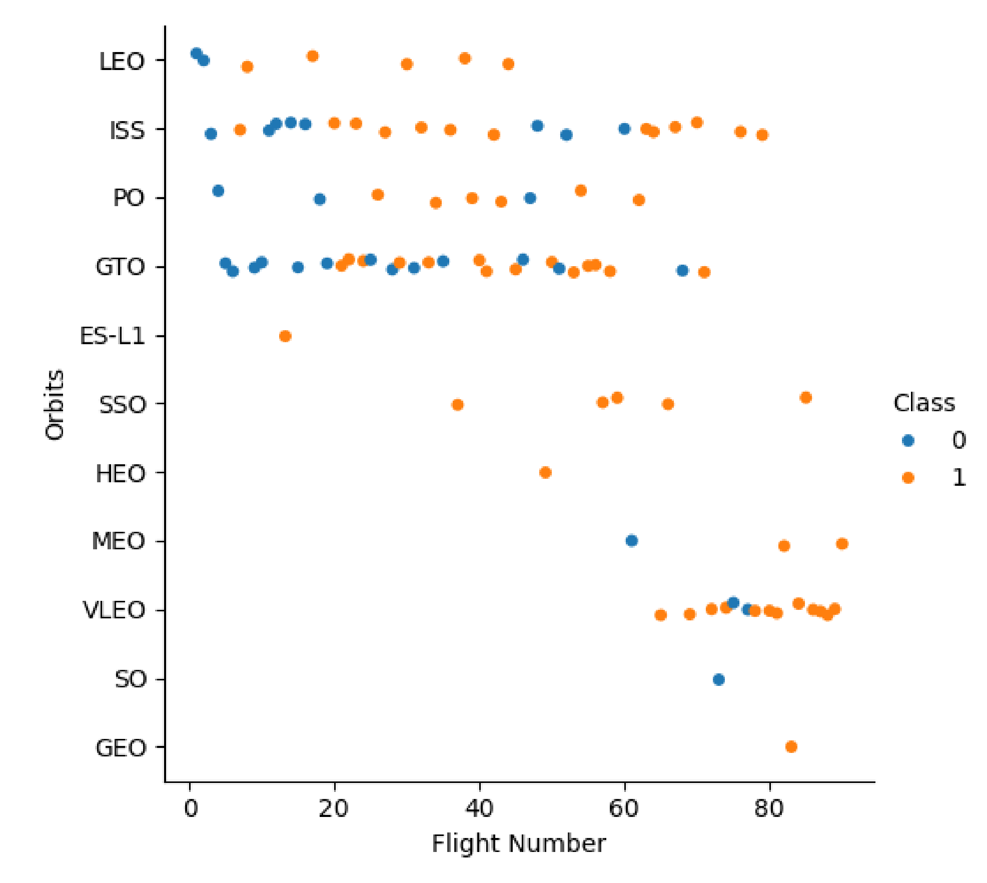
- ES-L1, SSO, HEO, and GEO orbits have a success rate of 100% which makes it an important feature in our prediction.



Flight Number vs. Orbit Type

- Scatter point of Flight number vs. Orbit type

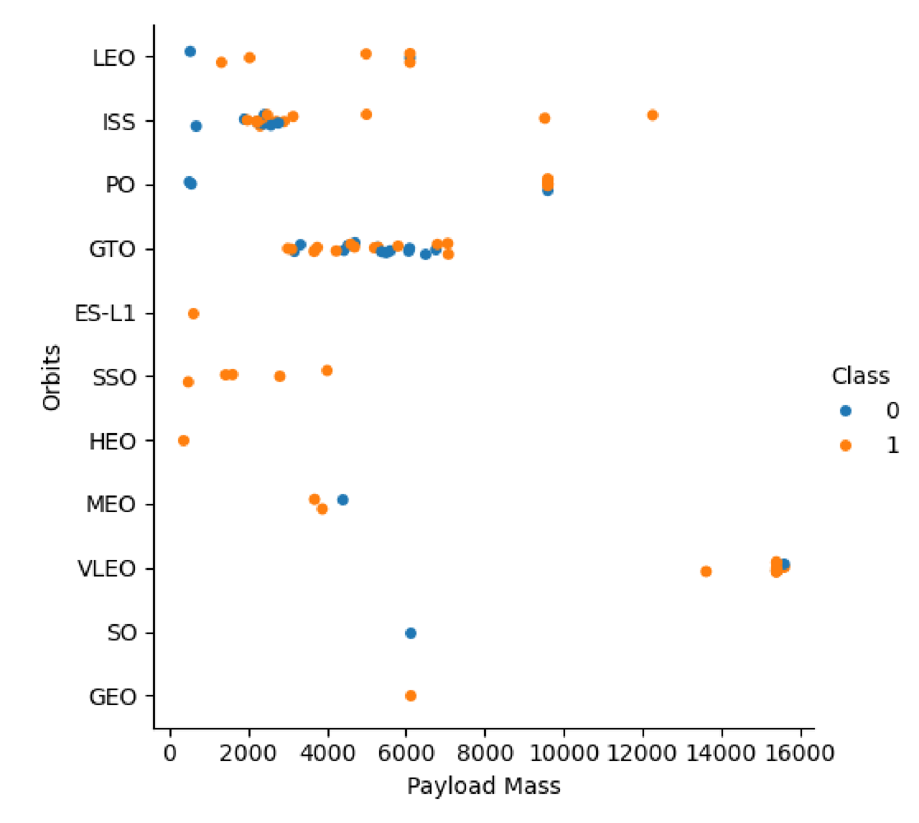
- Some orbits such as SSO, HEO, MEO, VLEO, SO, and GEO all start launches from approximately flight number 40 indicating they are new flight orbits.



Payload vs. Orbit Type

- Scatter point of payload vs. orbit type

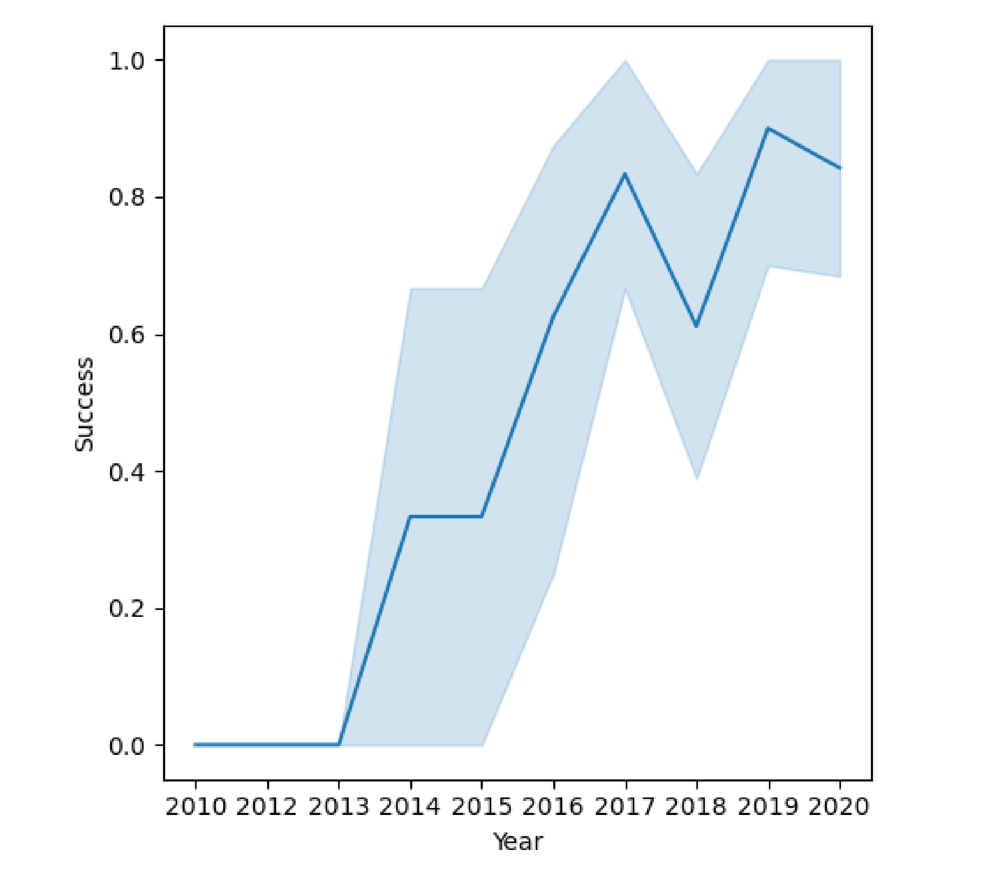
- Orbits with Bigger payload masses' tend to succeed in landing more other than their counter parts.



Launch Success Yearly Trend

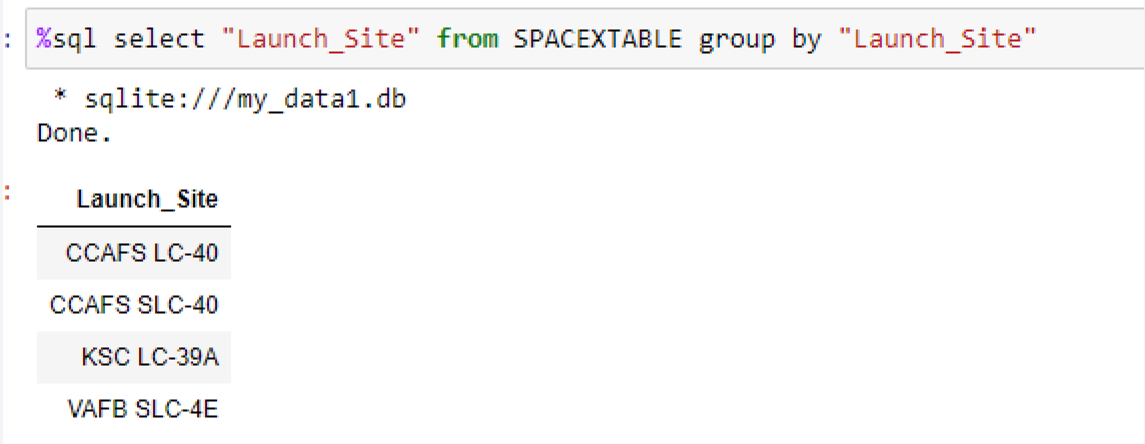
- Line chart of yearly average success rate

- Success rate increased starting from 2013 till 2020 which indicates better probabilities for future landings as technologies improve.



All Launch Site Names

- In the below SQL query we found all unique launch sites by grouping them together.



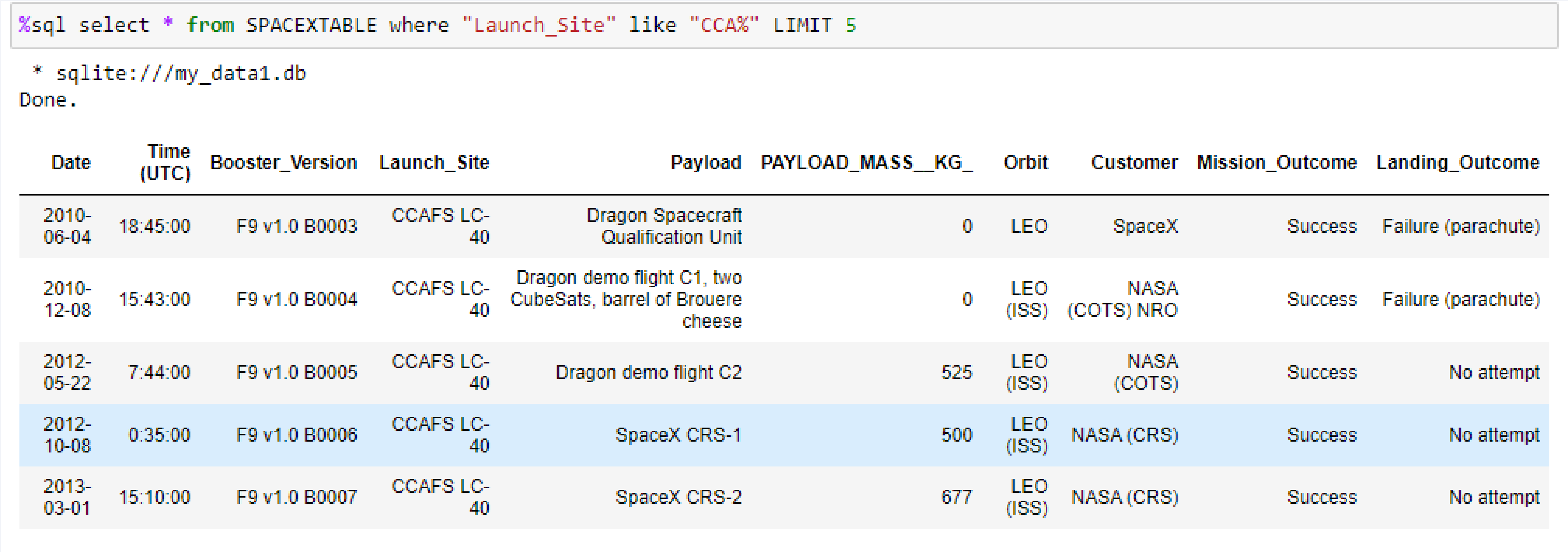
```
%sql select "Launch_Site" from SPACEXTABLE group by "Launch_Site"
```

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

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

Launch Site Names Begin with 'CCA'

- In this SQL query we requested data about launches with the launching site's name starts with "CCA" and only printed 5 of them



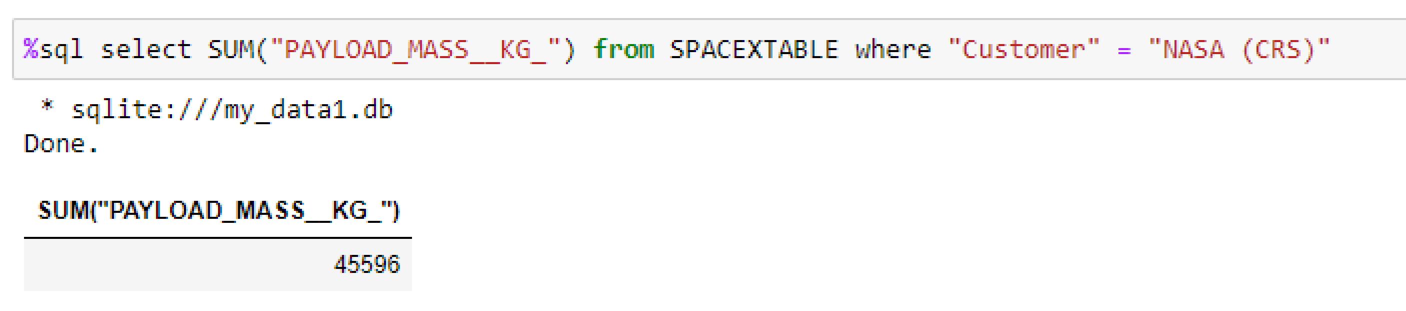
```
%sql select * from SPACEXTABLE where "Launch_Site" like "CCA%" LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- In this SQL query we calculated the sum of the payload masses of rockets launched by NASA (CRS) which turned out to be 45,596



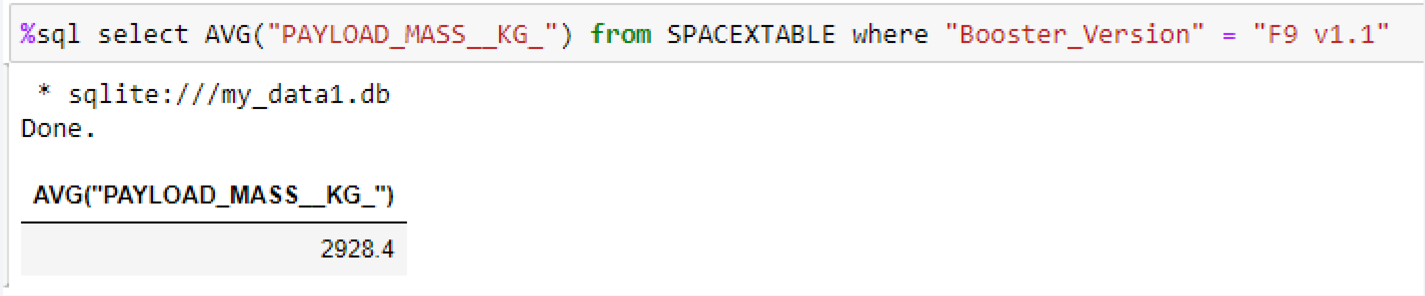
```
%sql select SUM("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Customer" = "NASA (CRS)"
```

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

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- Here we show the average of payload mass carried by rockets with booster version of F9 v1.1



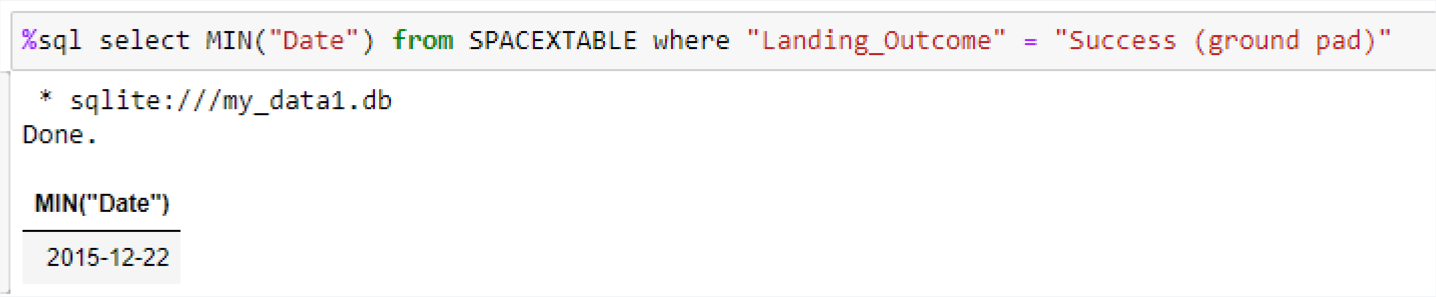
```
%sql select AVG("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Booster_Version" = "F9 v1.1"
```

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

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

- Here we wanted to know the date of the first successful ground pad landing which shows it was in 2015



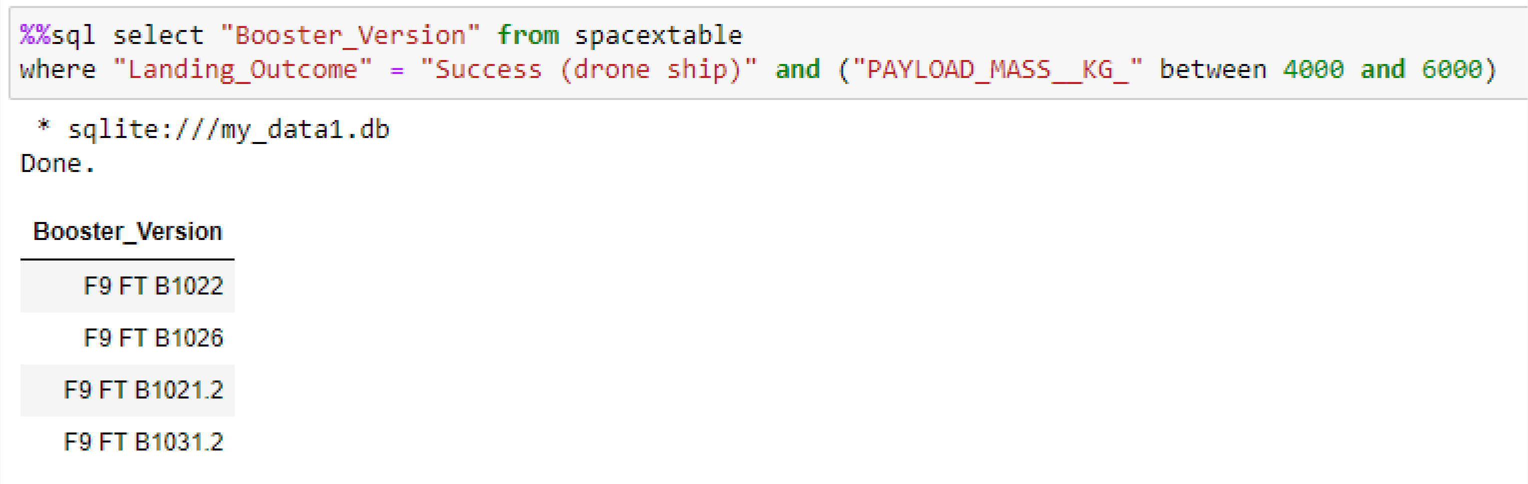
```
%sql select MIN("Date") from SPACEXTABLE where "Landing_Outcome" = "Success (ground pad)"
```

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

MIN("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Here we specified that the landing outcome was on a drone ship then filtered those landings for ones with payload masses between 4000 and 6000



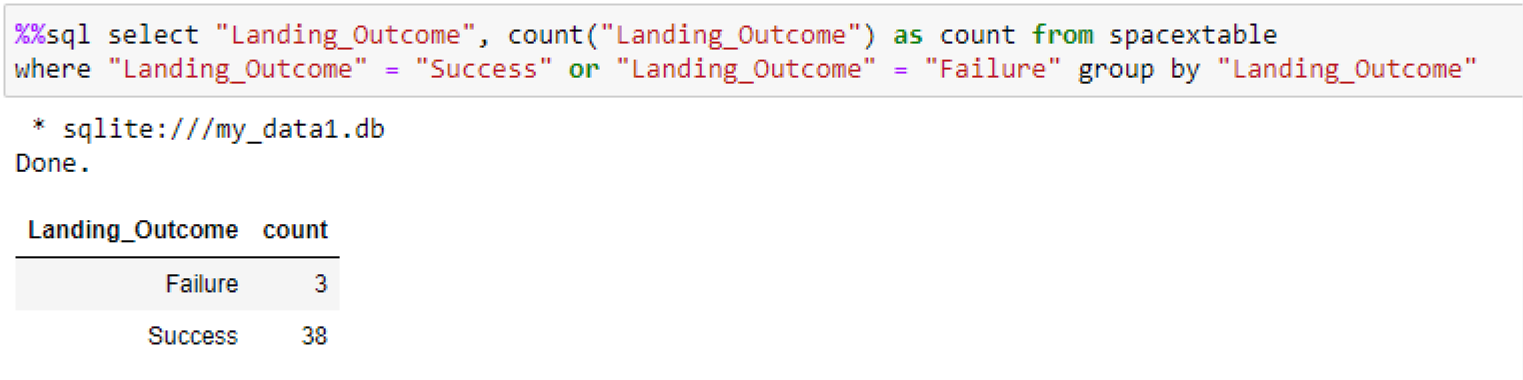
```
%%sql select "Booster_Version" from spacextable
where "Landing_Outcome" = "Success (drone ship)" and ("PAYLOAD_MASS__KG_" between 4000 and 6000)
```

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

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

Total Number of Successful and Failure Mission Outcomes

- This query shows the count of successful landings and failed ones where we specify that only success and failure are the ones we are looking for.



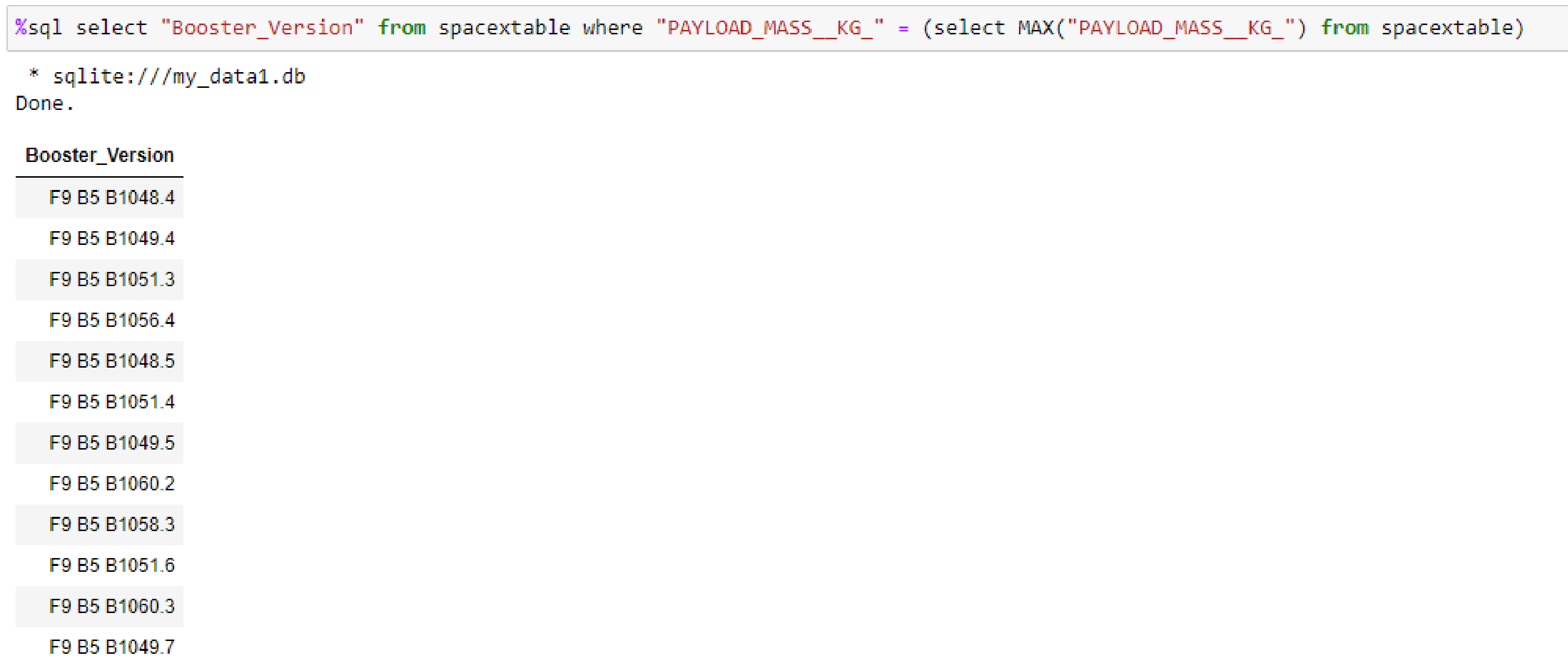
```
%%sql select "Landing_Outcome", count("Landing_Outcome") as count from spacextable
where "Landing_Outcome" = "Success" or "Landing_Outcome" = "Failure" group by "Landing_Outcome"
```

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

Landing_Outcome	count
Failure	3
Success	38

Boosters Carried Maximum Payload

- This query shows the booster versions capable of carrying the maximum payload mass.



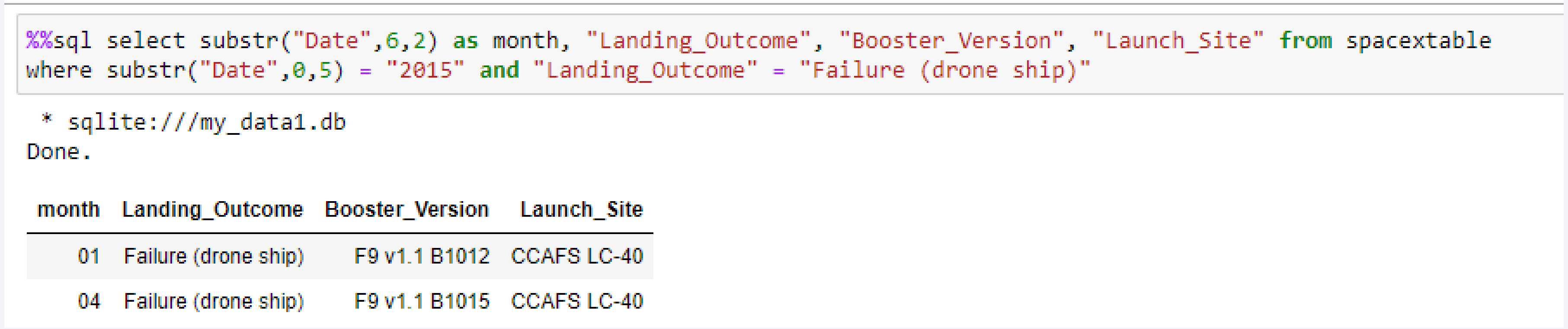
```
%sql select "Booster_Version" from spacextable where "PAYLOAD_MASS__KG_" = (select MAX("PAYLOAD_MASS__KG_") from spacextable)
```

```
 * 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

2015 Launch Records

- Here we specified that the landing outcome to be failure (drone ship) and for the year of 2015 showing the months when the failures happened, the corresponding booster version and launch site.



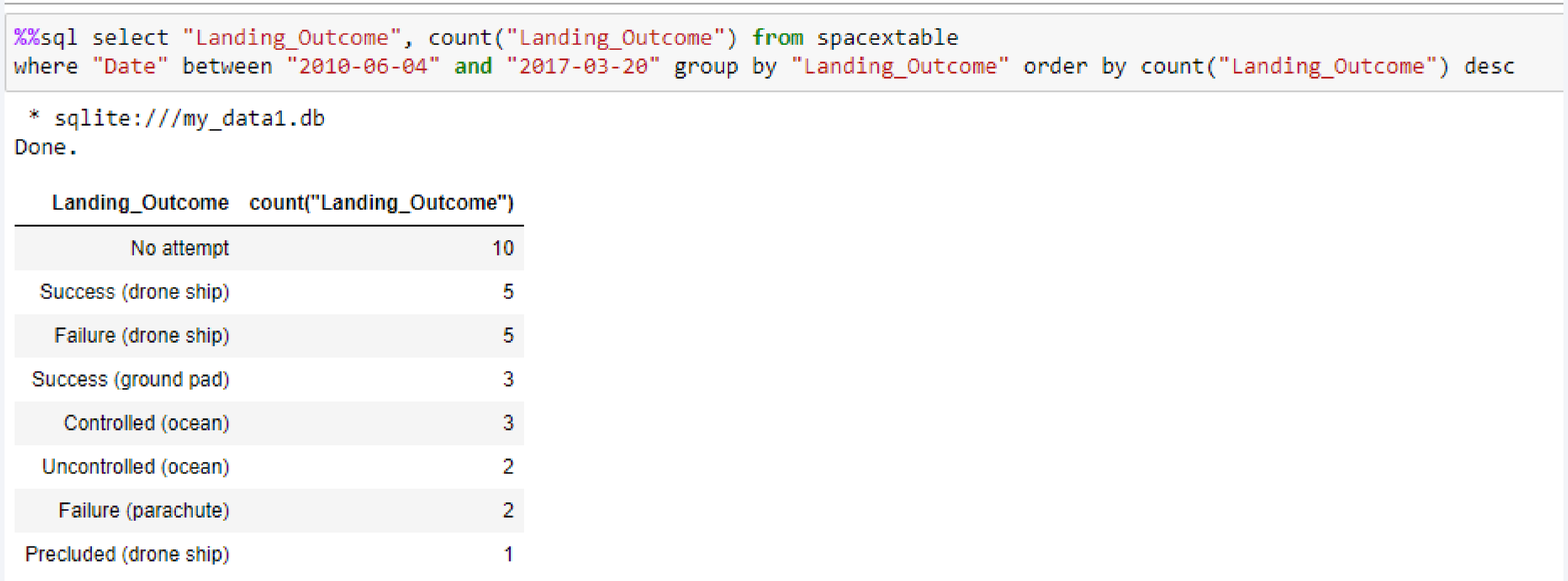
```
%%sql select substr("Date",6,2) as month, "Landing_Outcome", "Booster_Version", "Launch_Site" from spacextable
where substr("Date",0,5) = "2015" and "Landing_Outcome" = "Failure (drone ship)"
```

```
 * 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

- Here we rank the landing outcomes in a descending order describing what was the most common outcome between 2010-06-04 and 2017-03-20



```
%%sql select "Landing_Outcome", count("Landing_Outcome") from spacextable
where "Date" between "2010-06-04" and "2017-03-20" group by "Landing_Outcome" order by count("Landing_Outcome") desc
```

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

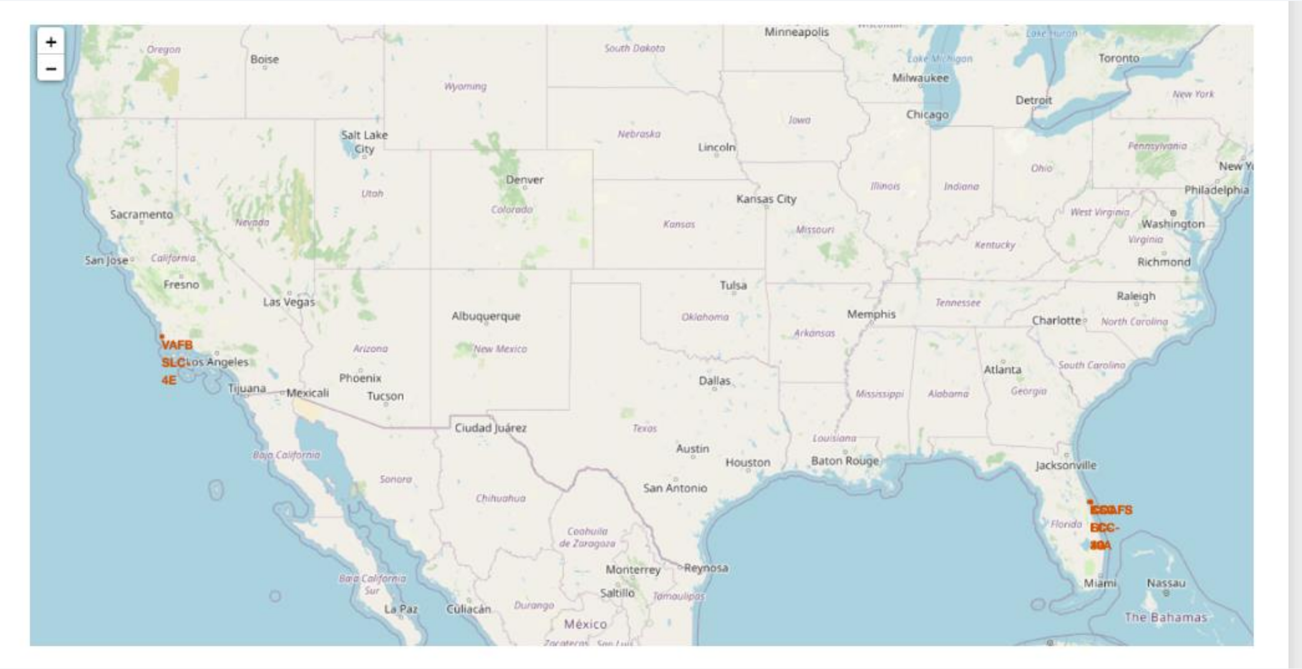
Landing_Outcome	count("Landing_Outcome")
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

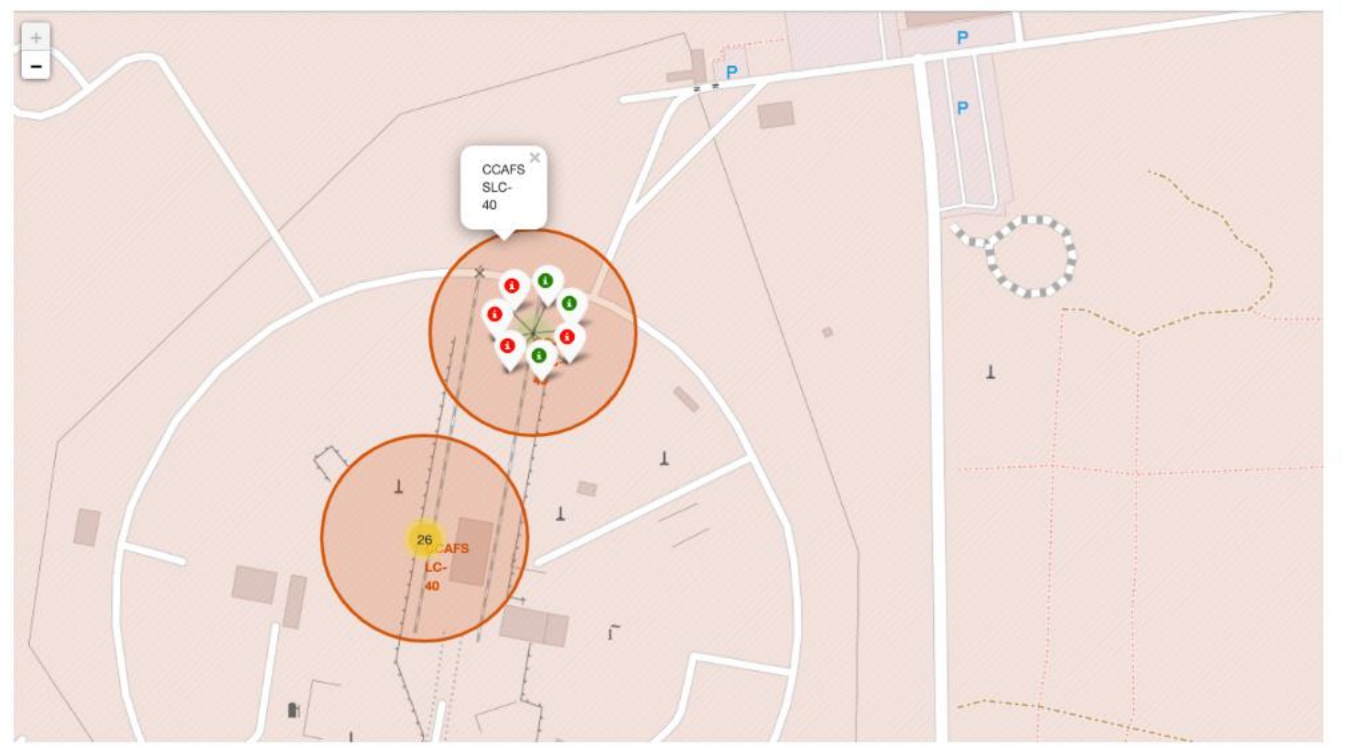
Launch sites locations

- Folium map indicating the locations of all launch sites



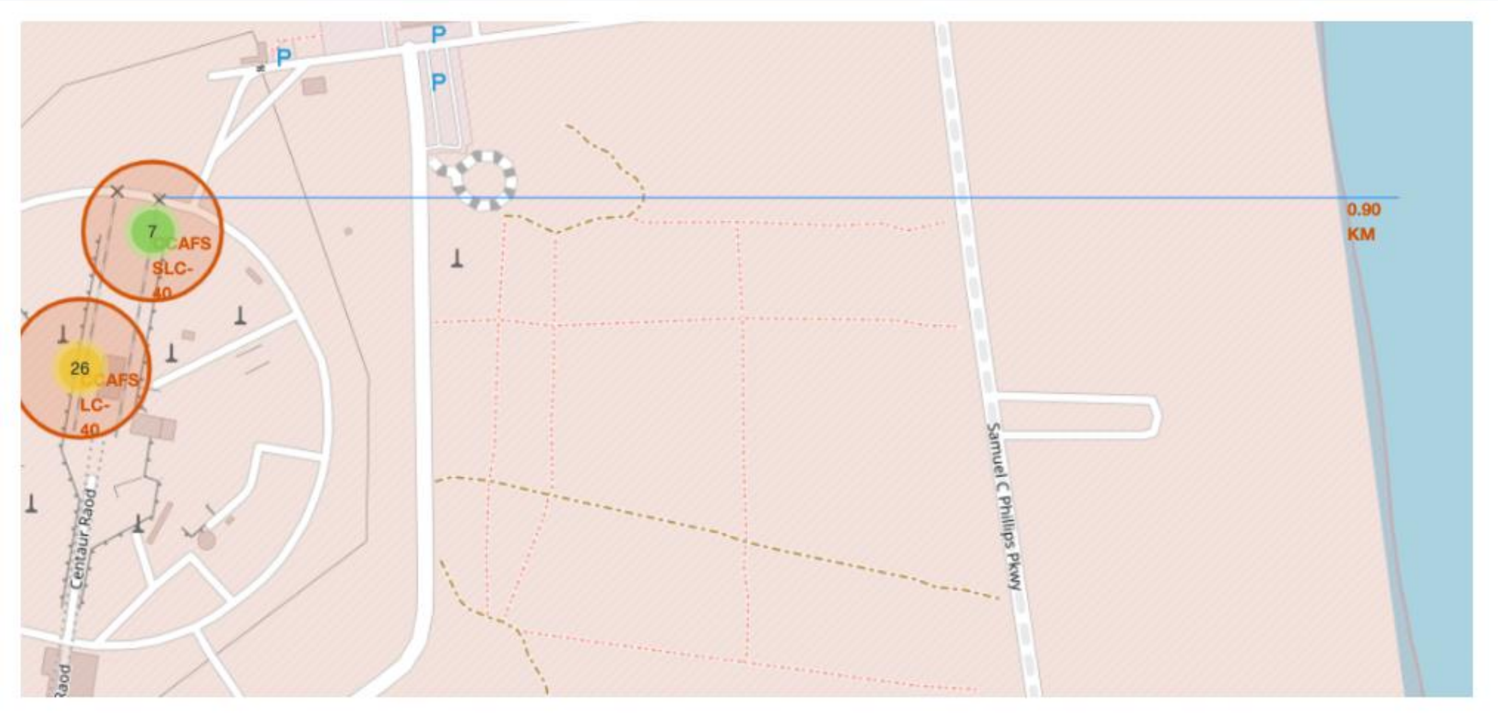
Launch site landing outcome

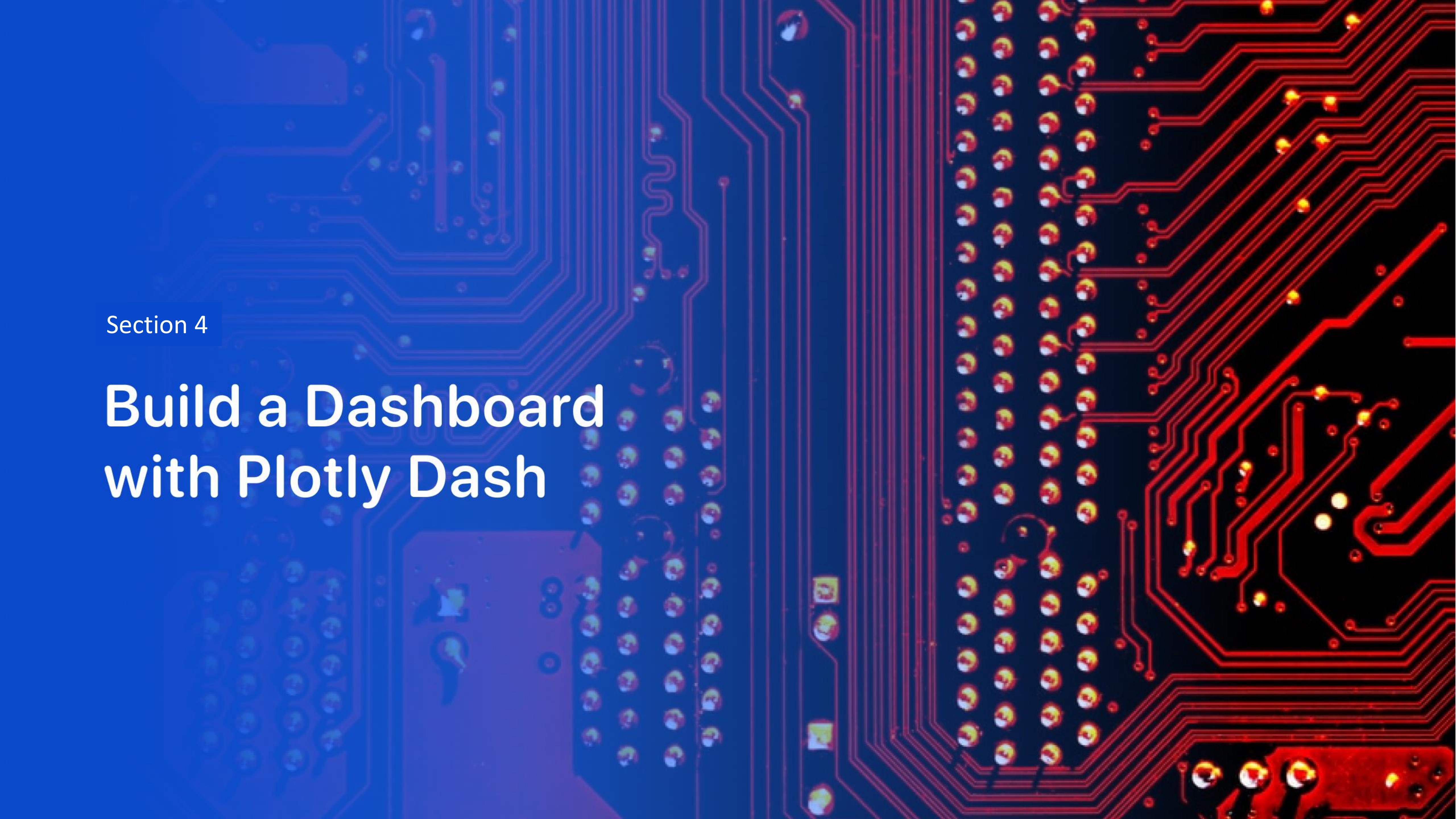
- Here we show the successful and the failed landings on a specific launch site location



Distance from proximities

- Here we show the distance of a launch site to the coastline.



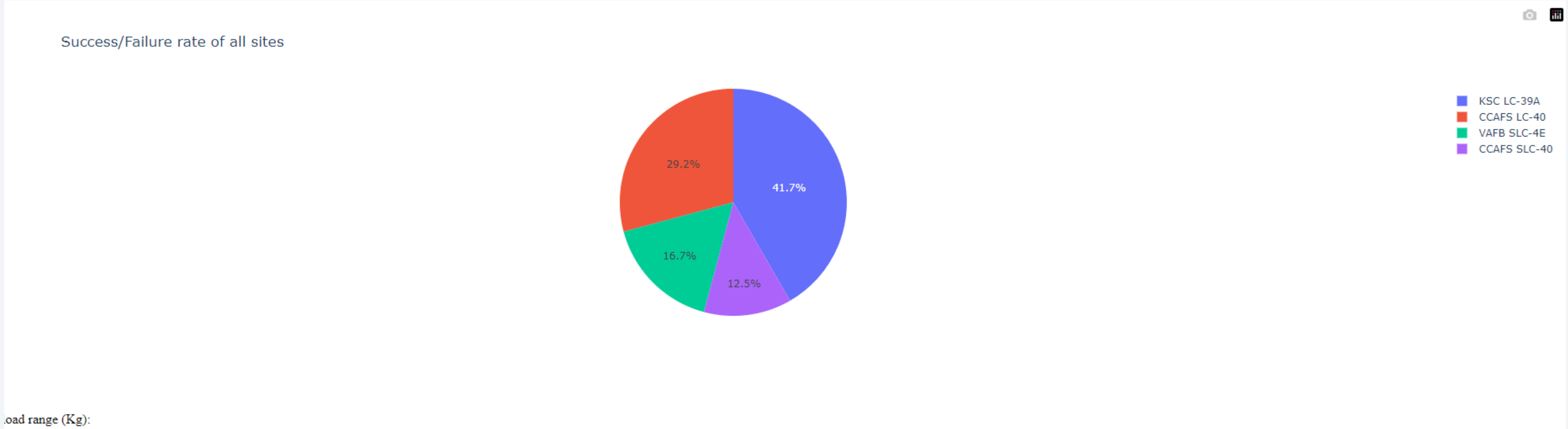


Section 4

Build a Dashboard with Plotly Dash

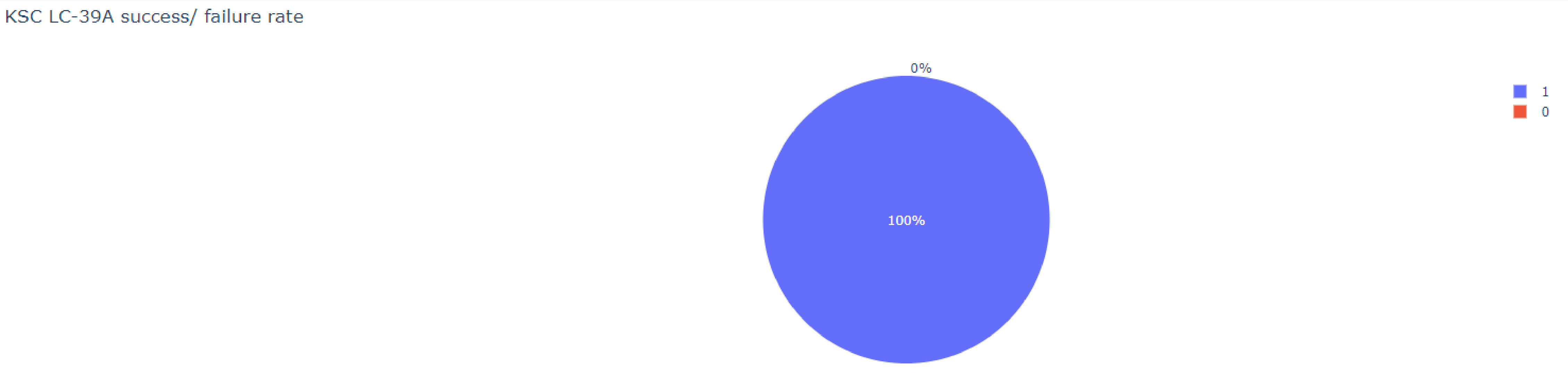
Success rate of Launch sites

- The pie chart shows that KSC launch site has the highest success rate from all sites.



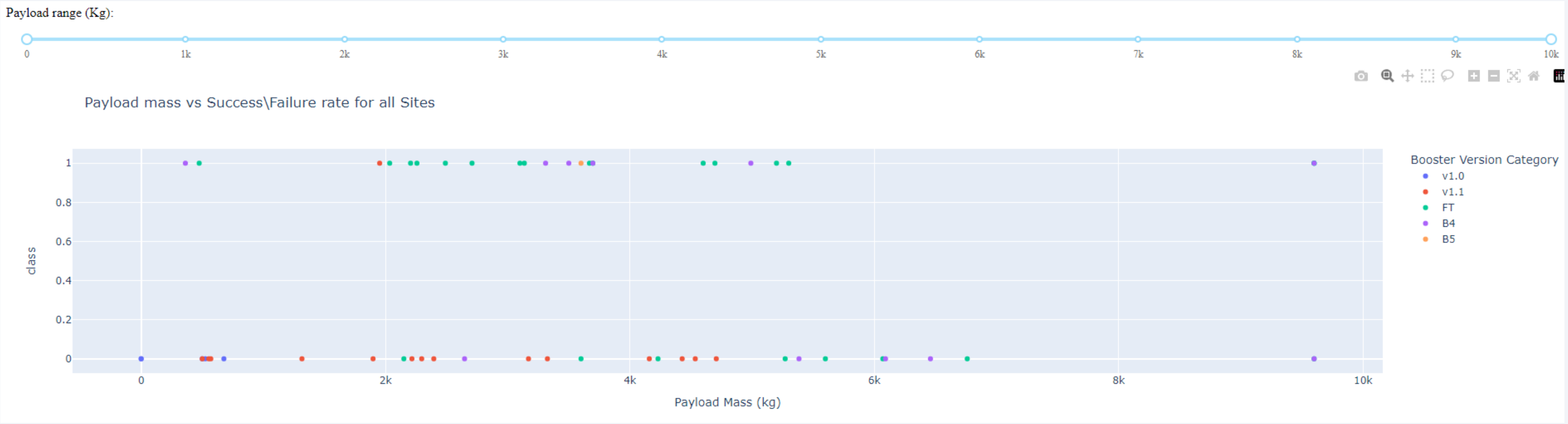
Launch site with highest success rate

- The chart shows the percent of successful landings for the KSC launch which is the highest of all launch sites.



Payload mass vs Launch outcome

- The plot shows that the booster version FT is the highest of successful landings in all sites from other booster versions.

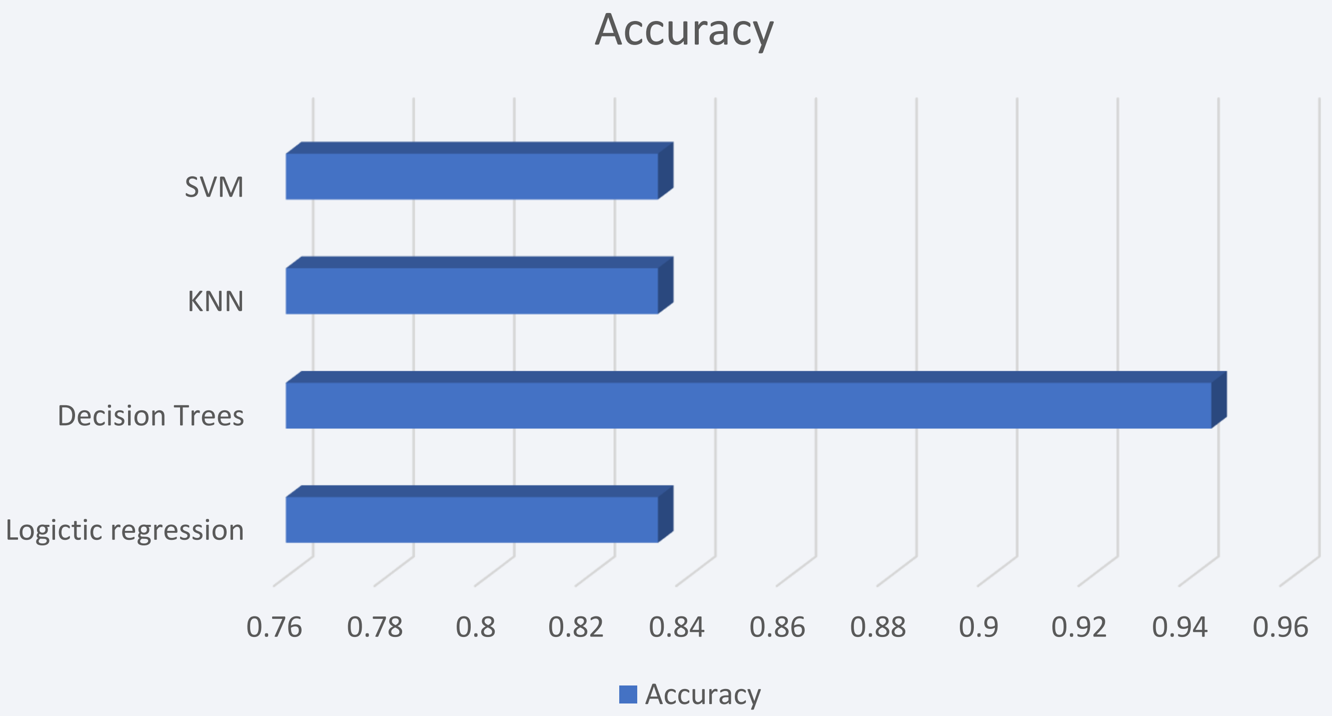


Section 5

Predictive Analysis (Classification)

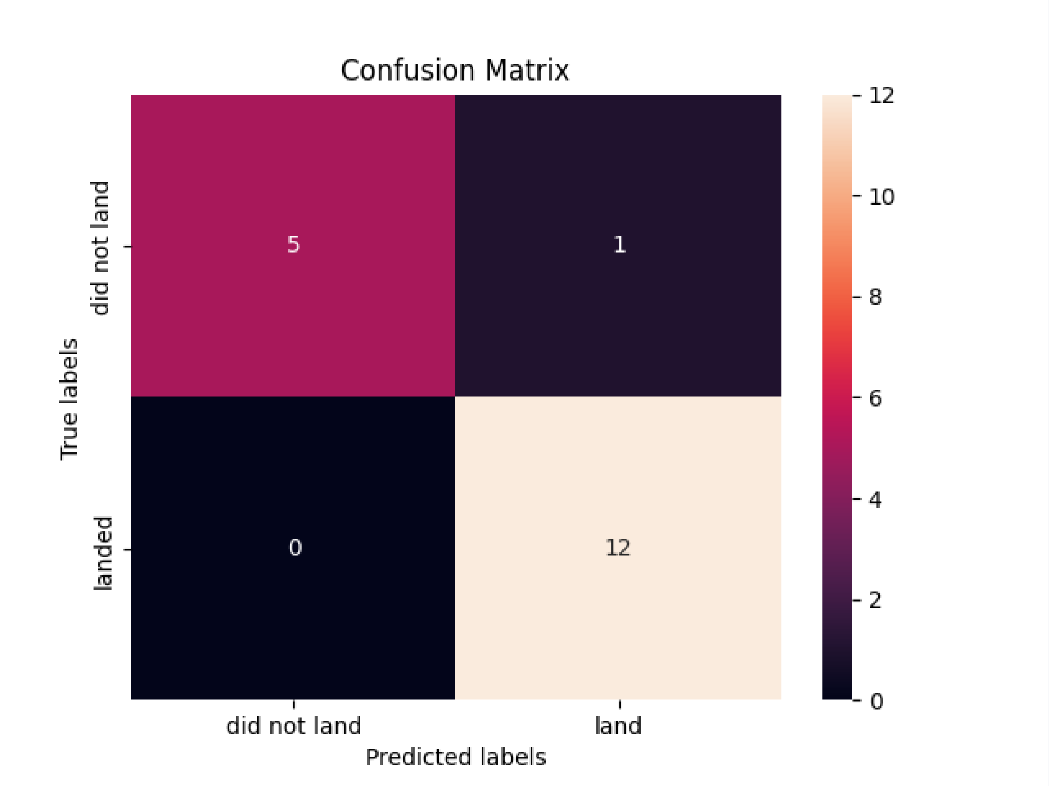
Classification Accuracy

- The model with the highest classification accuracy is the Decision Tree model with classification accuracy of 0.875



Confusion Matrix

- Confusion Matrix for the Decision Tree classifier.



Conclusions

- Data gathered and the features are relevant in solving the problem
- Many features have a relation with the target variable, interactions between features is also helpful.
- Machine learning models can predict the outcome of the landing with a whooping accuracy of 0.94
- The company now can save resources by entering the rocket launch data before launching the rocket to predict whether stage 1 will land successfully or not.

Appendix

- Dataset used in model training: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv
- Dataset used in folium map: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_geo.csv
- Dataset used in visualization: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv

Thank you!