IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Limeng Jiang>

<11/11/2023>

Outline

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

Executive Summary

- The methodologies included in this project are data collection, data wrangling, exploratory data analysis (EDA), interactive data analytics, and predicative analysis (with machine learning algorithms).
- The results from this project include data analysis results, data visualization results, and predicative analysis results.

Introduction

- Project background and context

  The goal of this project is to predict if the Falcon 9 first stage will land successfully. 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 also be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems to be answered
  - What attributes and training labels need to be used for the predicative model.
  - The effect of each feature on the outcome of the launch.

Section 1

Methodology

Methodology

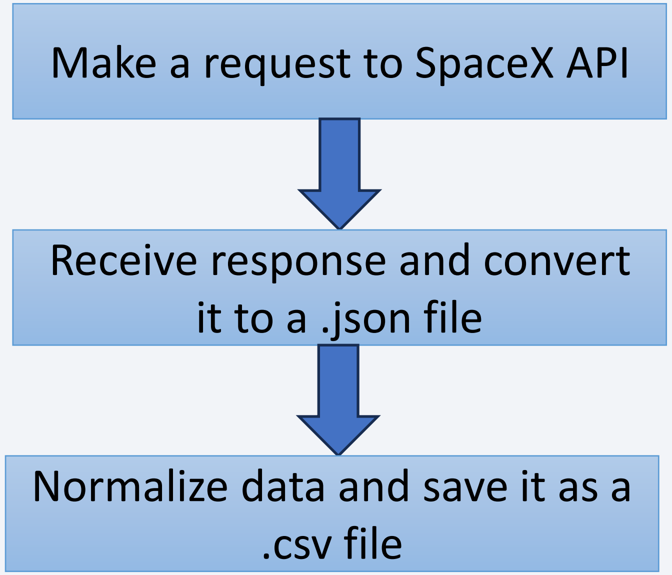
Executive Summary

- Data collection methodology:
  - Make a request to the SpaceX API
  - Web scraping from the Wikipedia page (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)
- Perform data wrangling
  - Calculate the number of launches on each site, number and occurrence of each orbit, number and occurrence of mission outcome of the orbits, and create labels for outcomes.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Initialize different classification models, use Grid Search to look for best model parameters, and compare their training and test accuracy scores to select the best performing model.

Data Collection

- Data collection process
  - For SpaceX API, we make a request to the SpaceX API and then go on to clean the requested data.
  - For web scraping, we extract a Falcon 9 launch records HTML table from Wikipedia and then parse the table and convert it into a Pandas data frame

SpaceX API

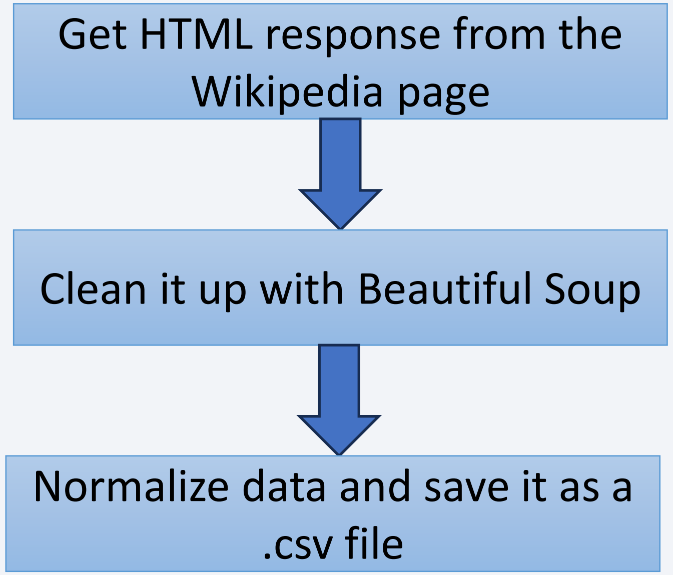


Make a request to SpaceX API

Receive response and convert it to a .json file

Normalize data and save it as a .csv file

Web scraping



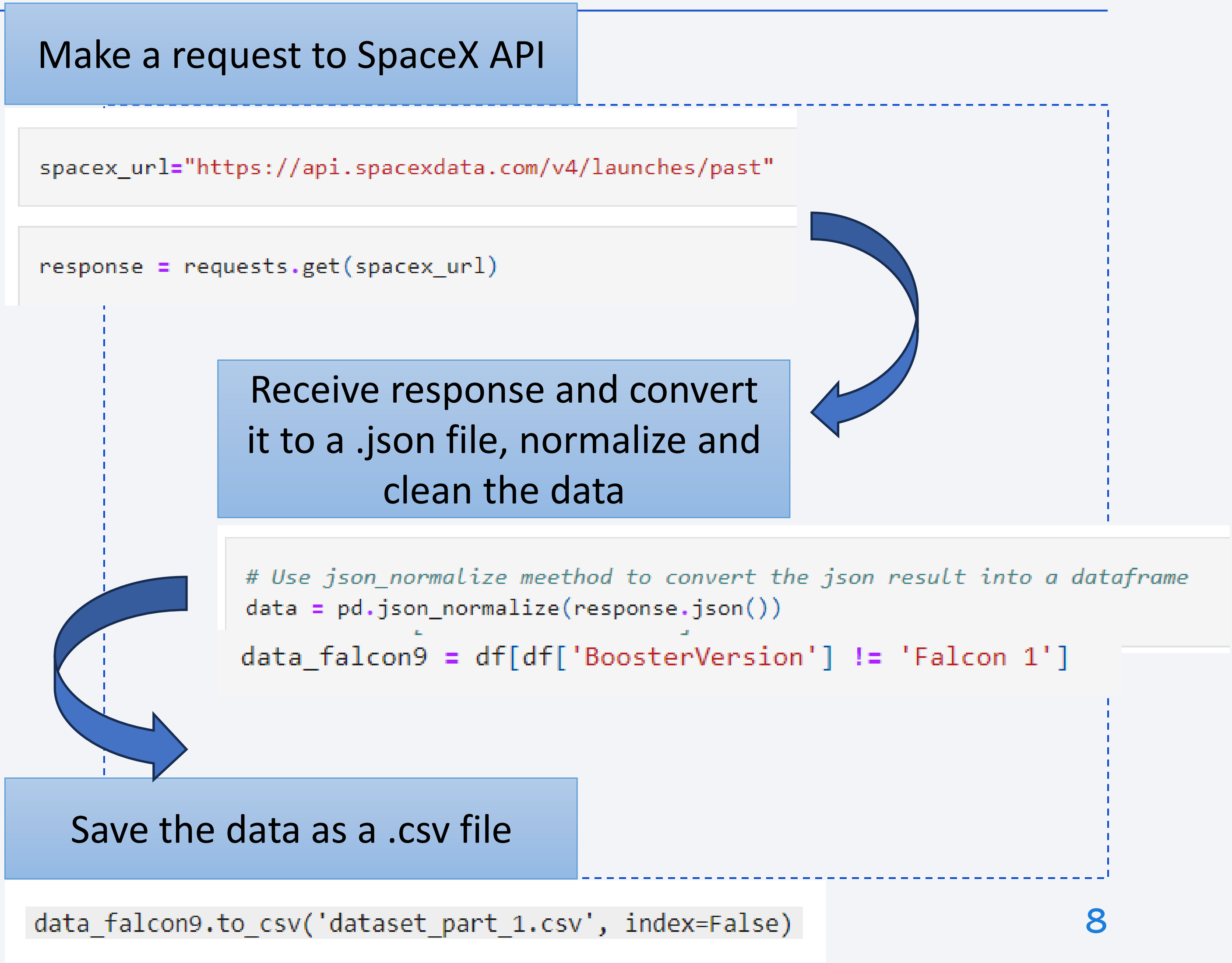
Get HTML response from the Wikipedia page

Clean it up with Beautiful Soup

Normalize data and save it as a .csv file

Data Collection – SpaceX API

- Reference to the SpaceX API calls notebook: https://github.com/jlmaurora233/IBM_DS/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Make a request to SpaceX API

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

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

Receive response and convert it to a .json file, normalize and clean the data

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

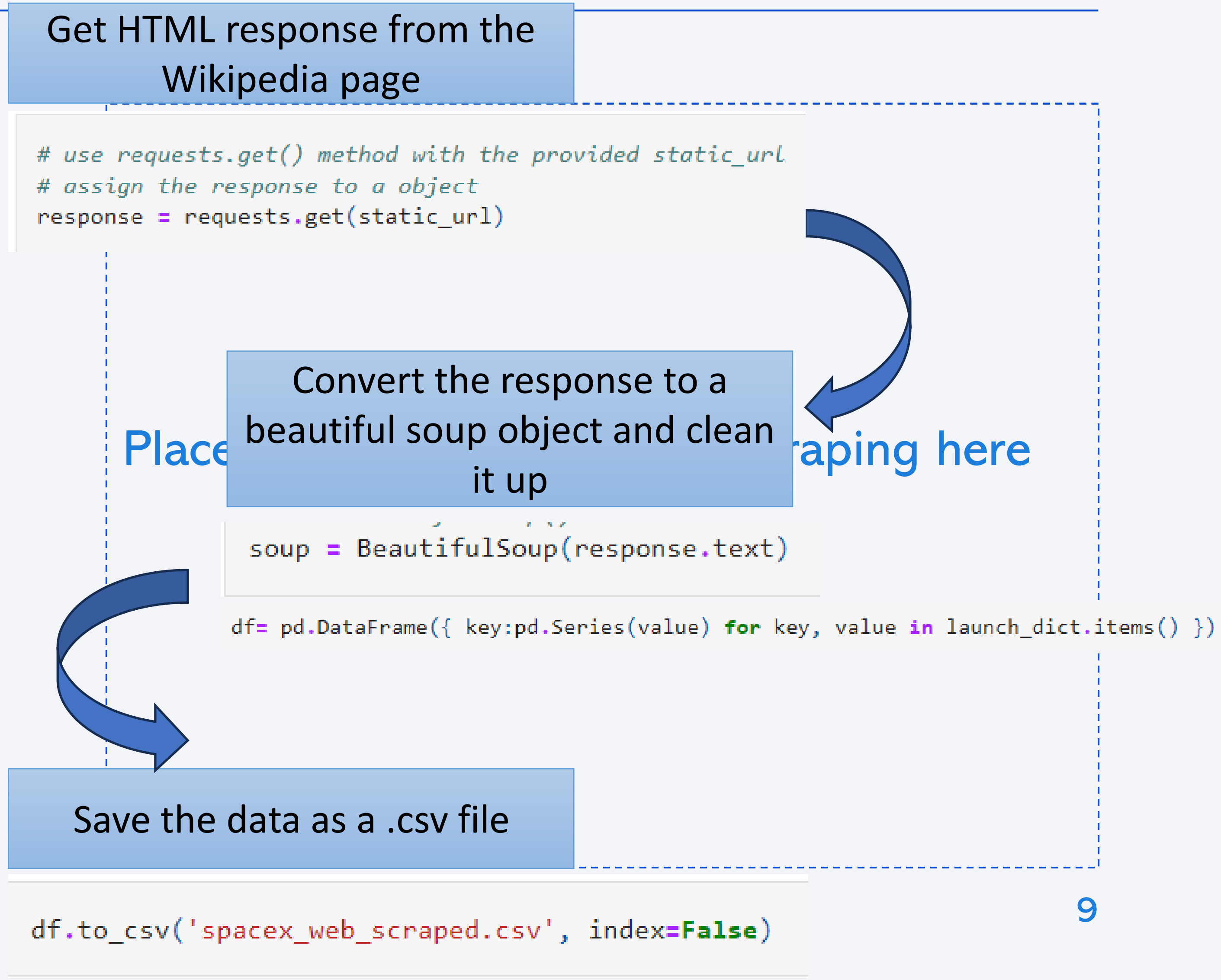
```
data_falcon9 = df[df['BoosterVersion'] != 'Falcon 1']
```

Save the data as a .csv file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

- Reference to the web scraping notebook: https://github.com/jlmaurora233/IBM_DS/blob/main/jupyter-labs-webscraping.ipynb



Get HTML response from the Wikipedia page

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

Place

Convert the response to a beautiful soup object and clean it up

raping here

```
soup = BeautifulSoup(response.text)
```

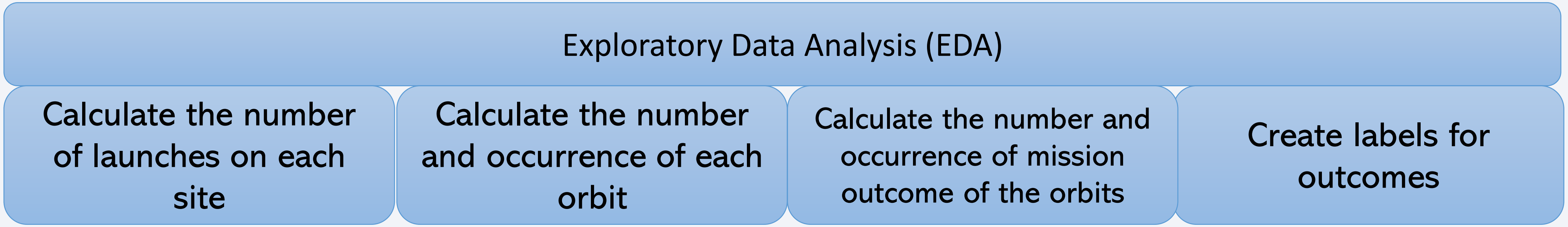
```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

Save the data as a .csv file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

- Data processing:
  - In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.
  - In this lab we will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.



Exploratory Data Analysis (EDA)

Calculate the number of launches on each site

Calculate the number and occurrence of each orbit

Calculate the number and occurrence of mission outcome of the orbits

Create labels for outcomes

Reference to the data wrangling notebook: https://github.com/jlmaurora233/IBM_DS/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Scatter plots:
  - Flight Number VS Payload Mass
  - Flight Number VS Launch Site
  - Flight Number VS Orbit Type
- Bar plots:
  - Success Rate of Each Orbit Type
- Line graphs:
  - Yearly Success Rate VS Years
- Reference to the EDA with data visualization notebook: https://github.com/jlmaurora233/IBM_DS/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

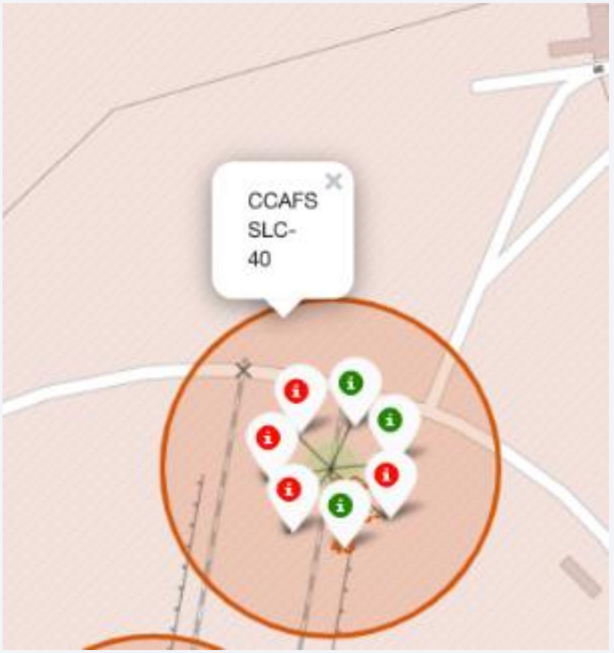
EDA with SQL

- List of SQL query tasks performed for EDA
  - **Display the names of the unique launch sites in the space mission**
  - **Display 5 records where launch sites begin with the string 'CCA’**
  - **Display the total payload mass carried by boosters launched by NASA (CRS)**
  - **Display average payload mass carried by booster version F9 v1.1**
  - **List the date when the first succesful landing outcome in ground pad was acheived.**
  - **List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000**
  - **List the total number of successful and failure mission outcomes**
  - **List the names of the booster_versions which have carried the maximum payload mass.**
  - **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.**
  - **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, in descending order.**
- Reference to the EDA with SQL notebook: https://github.com/jlmaurora233/IBM_DS/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- The Folium map includes…
  - Circle markers to represent different launch sites. The color of the markers represent the launch outcomes, with green=success and red=failure
  - Lines to represent the distance between launch sites and other landmarks

- Reference to the interactive map with Folium map: https://github.com/jlmaurora233/IBM_DS/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

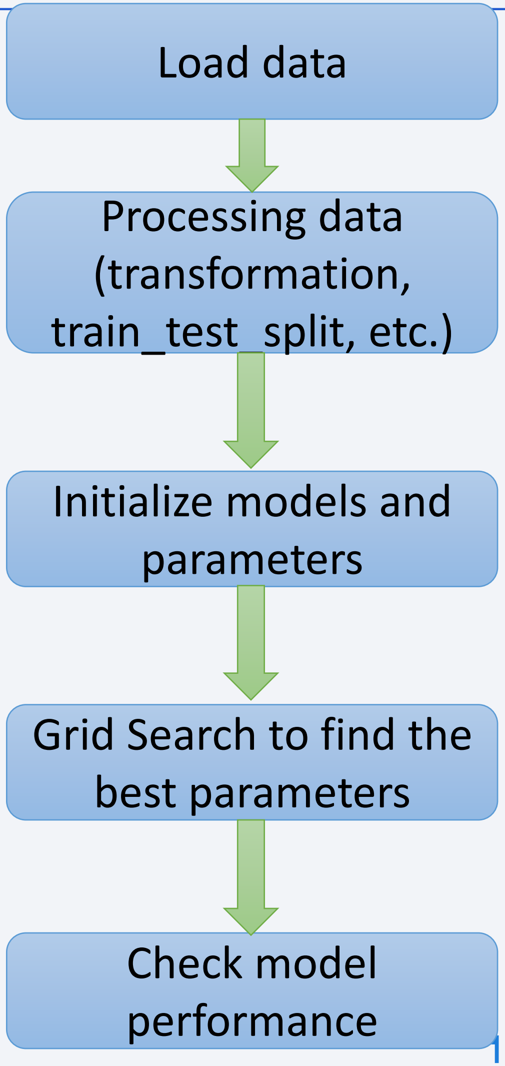


Build a Dashboard with Plotly Dash

- The Dashboard includes…
  - A dropdown section which lists individual launch sites and an option for “All Sites”. That can be used to change the plots to show the exact launch site’s statistics you would like to see.
  - Pie charts showing the total launches of all sites/ successful and failed launches for individual site.
  - A range slider which allows you to slide to different Payload Mass ranges. That will limit the range of payload mass for the scatter plots.
  - Scatter plots showing the relationship between Outcome and Payload Mass for different Booster Versions.
- Reference to the Plotly Dash lab: https://github.com/jlmaurora233/IBM_DS/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Model Build-up
  - Load the dataset into a pandas dataframe
  - Data transformation (e.g. convert data into arrays)
  - Split the data into training and test sets
  - Initialize machine learning algorithms
  - Set the parameter set and fit them to GridSearchCV to find the best parameters
- Model Evaluation
  - Check the training accuracy and test accuracy scores
  - Plot out the confusion matrix
- Find the best performing classification model
  - Choose the model with the highest accuracy scores
- Reference to the predictive analysis lab: https://github.com/jlmaurora233/IBM_DS/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb



Load data

Processing data (transformation, train_test_split, etc.)

Initialize models and parameters

Grid Search to find the best parameters

Check model performance

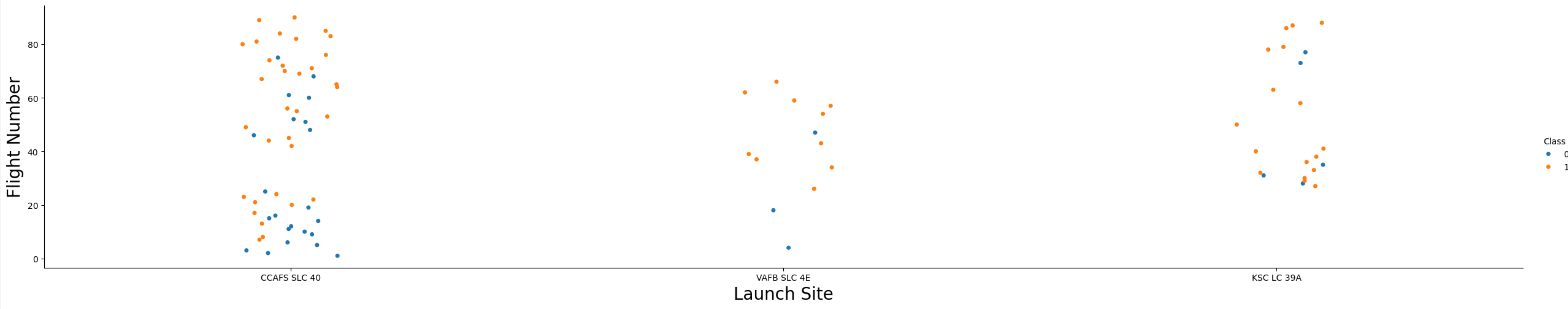
Results

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

Section 2

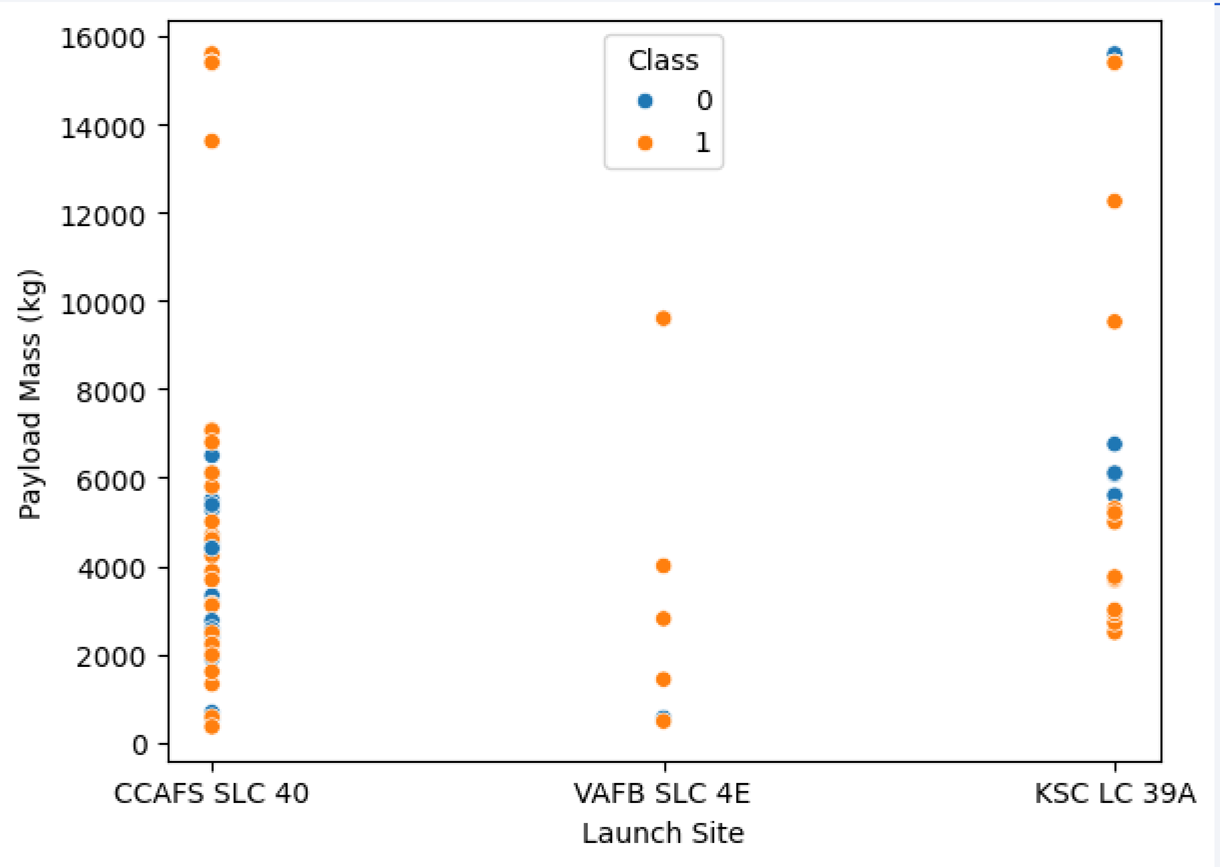
Insights drawn from EDA

Flight Number vs. Launch Site



- As the flight number increases, the launch outcome tends to be successful.

Payload vs. Launch Site

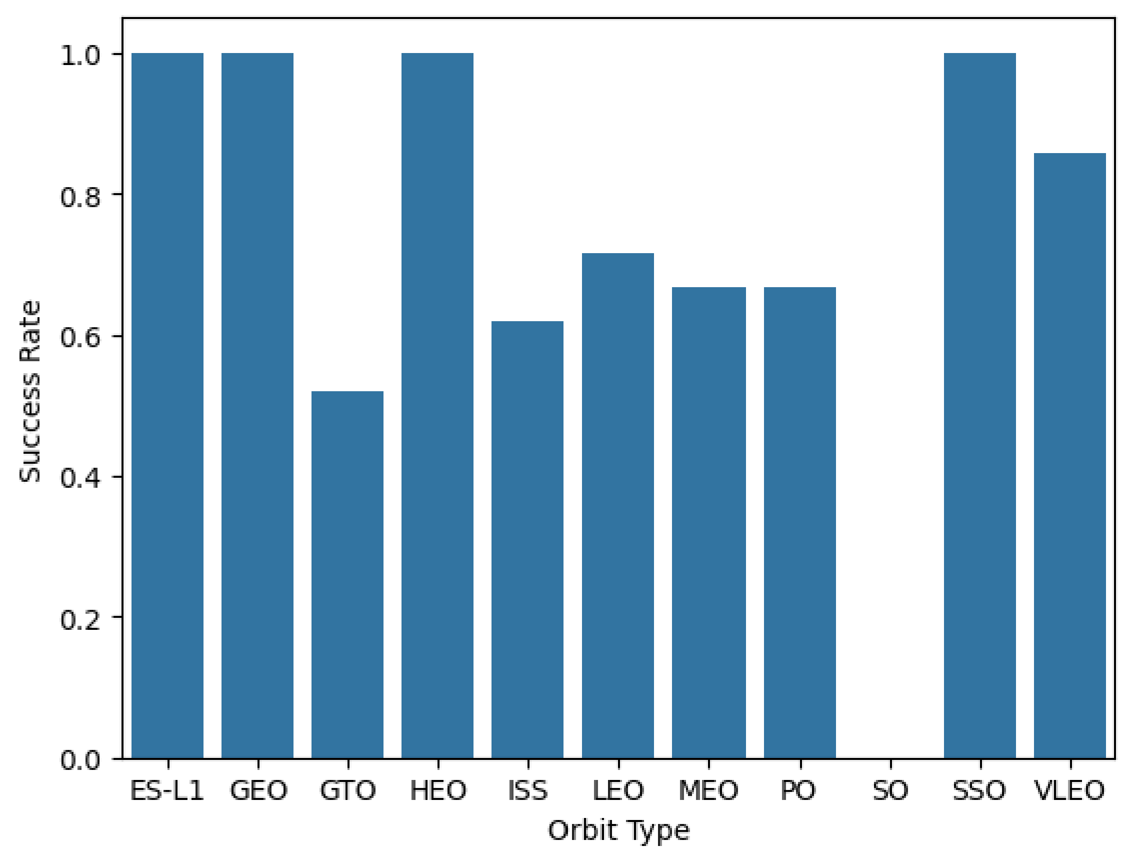


- For CCAFS SLC-40, the launch outcome does not depend on the payload mass;

For VAFB SLC-4E, as payload mass increases, the launch outcome tends to be successful;

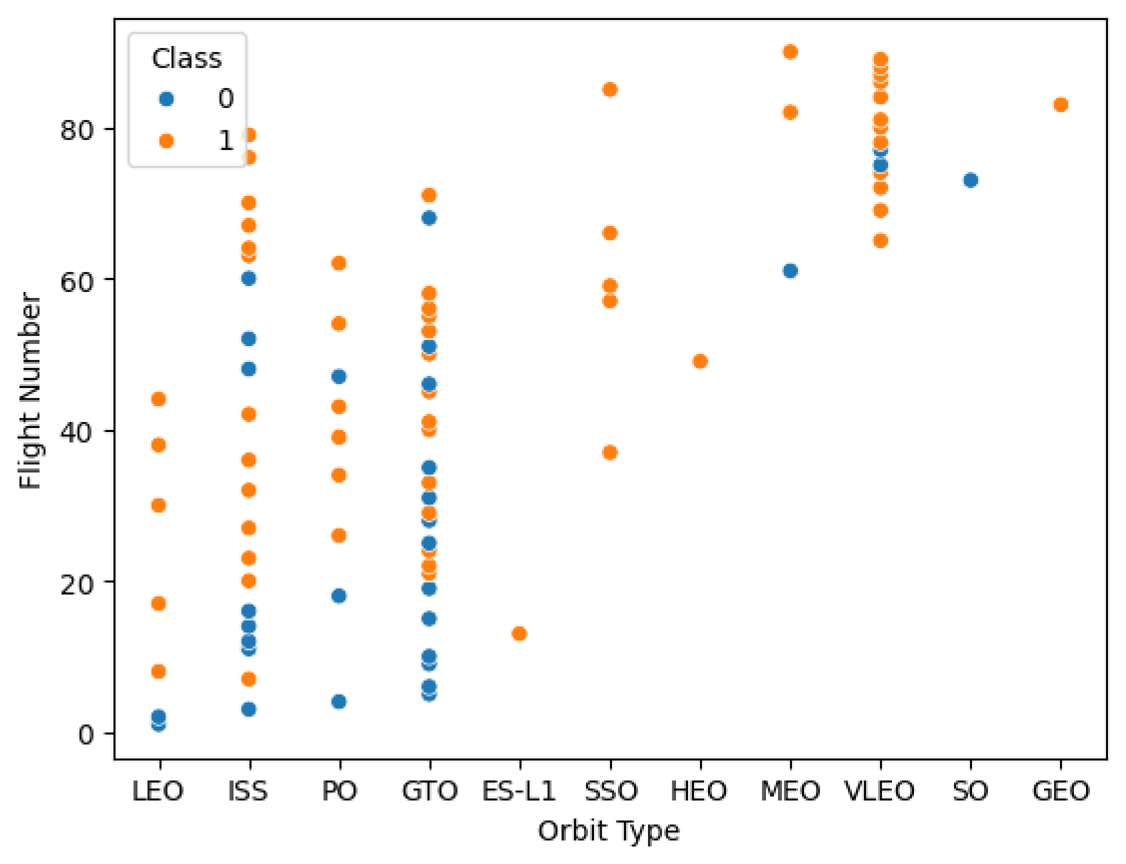
For KSC LC-39A, when the payload mass is between 6000 and 8000 kg, the launch outcome tends to be failed.

Success Rate vs. Orbit Type



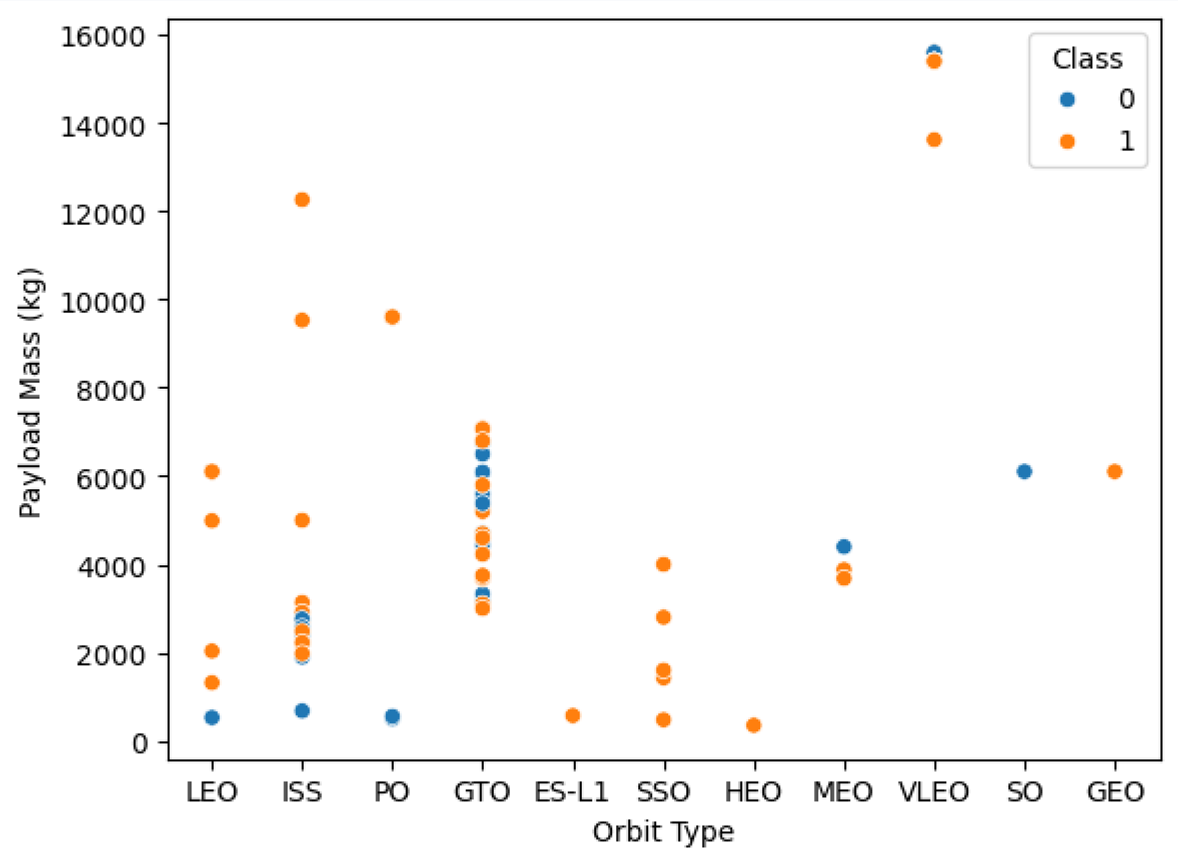
- ES-L1, GEO, HEO and SSO have the highest success rate, while SO has no successful launch according to the available data.

Flight Number vs. Orbit Type



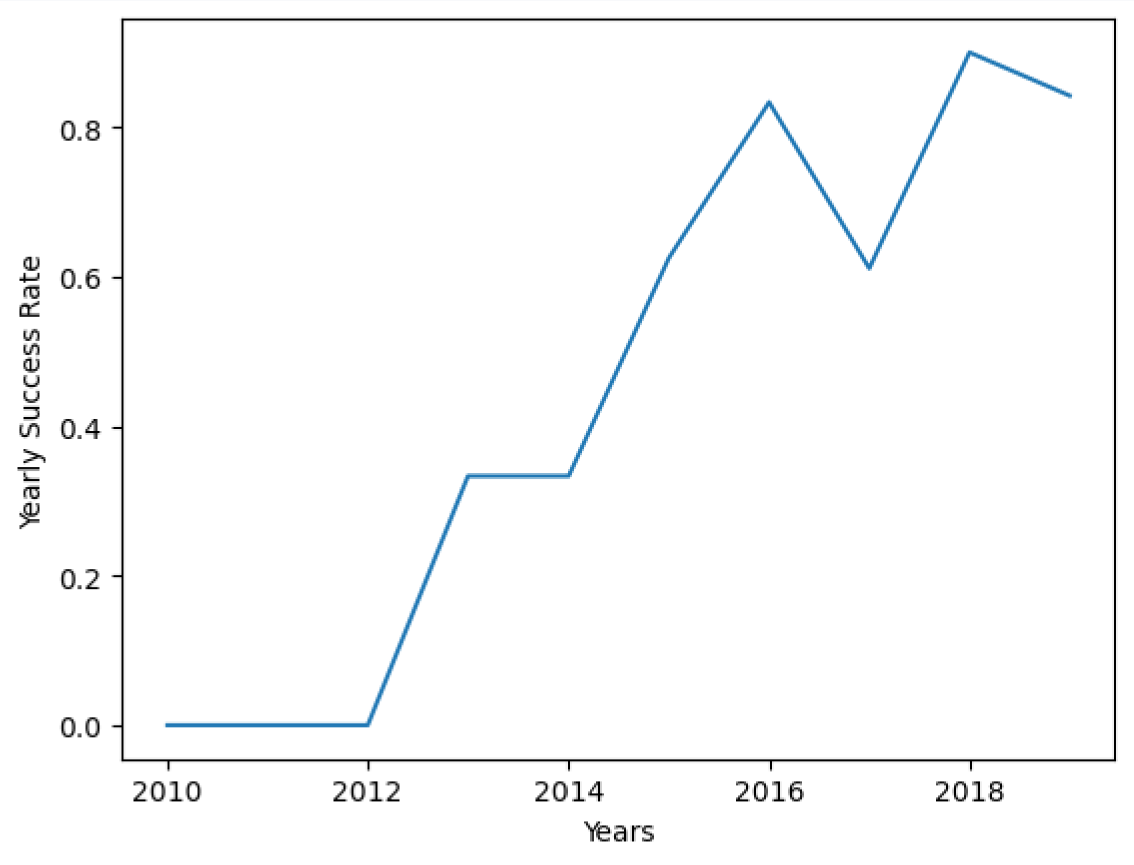
- In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

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.

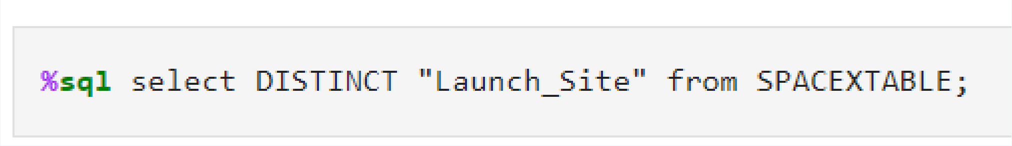
Launch Success Yearly Trend



- Starting 2013, the success rate keeps increasing.

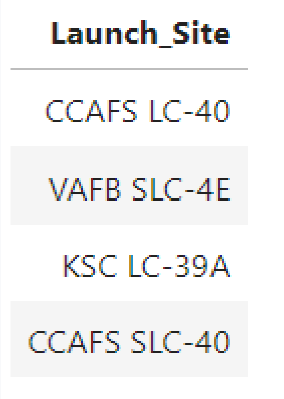
All Launch Site Names

- Query



```
%sql select DISTINCT "Launch_Site" from SPACEXTABLE;
```

- Result



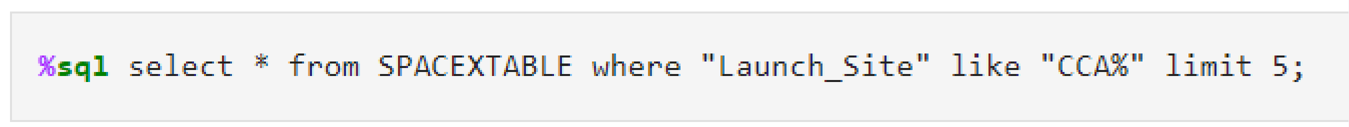
Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

- Explanation

Use DISTINCT to select the unique values in the column

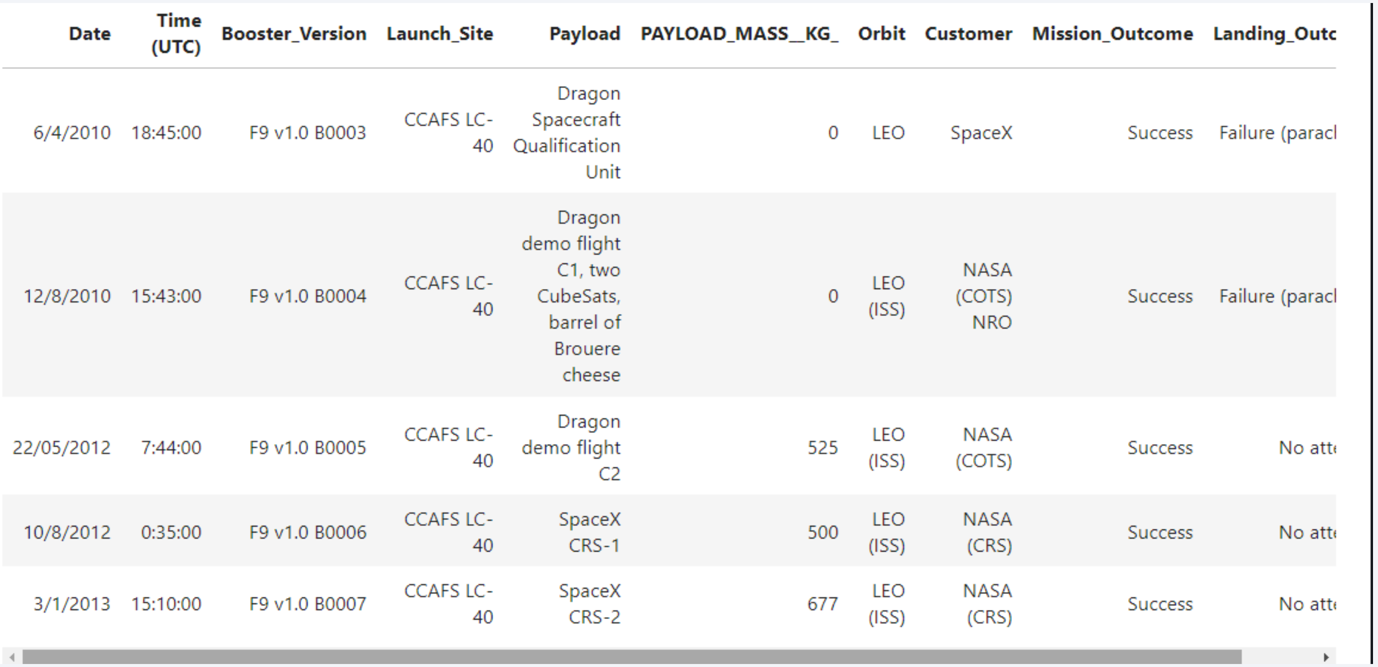
Launch Site Names Begin with 'CCA'

- Query



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

- Result



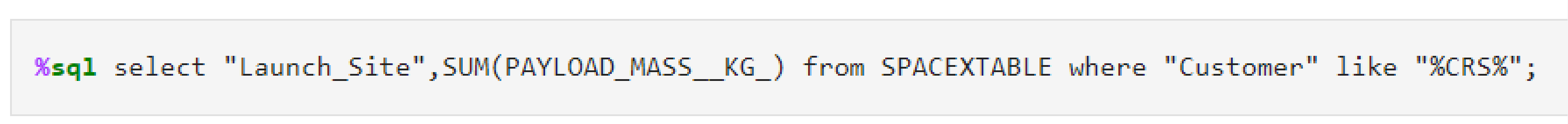
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
6/4/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (paracl
12/8/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (paracl
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No att
10/8/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No att
3/1/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No att

- Explanation

Use LIMIT to make sure that only 5 sites are shown; use LIKE “CCA%” to select names beginning with “CCA”

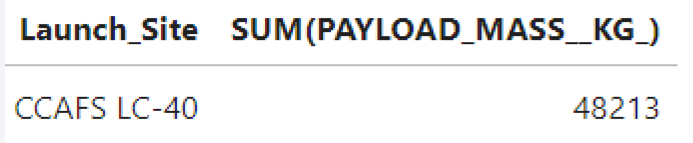
Total Payload Mass

- Query



```
%sql select "Launch_Site",SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Customer" like "%CRS%";
```

- Result



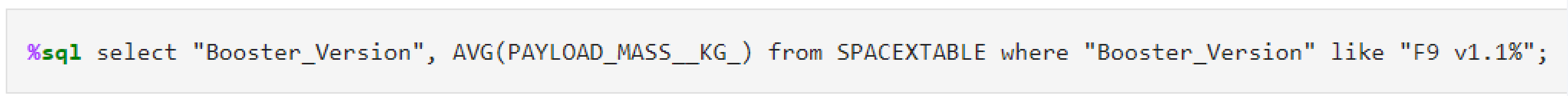
Launch_Site	SUM(PAYLOAD_MASS__KG_)
CCAFS LC-40	48213

- Explanation

Use SUM to calculate the total value; use “Customer” LIKE “%CRS%” to get boosters launched by NASA (CRS)

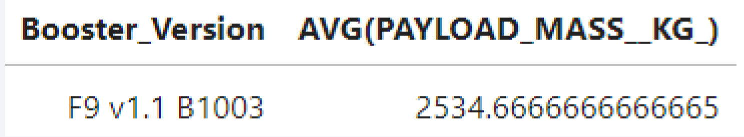
Average Payload Mass by F9 v1.1

- Query



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

- Result



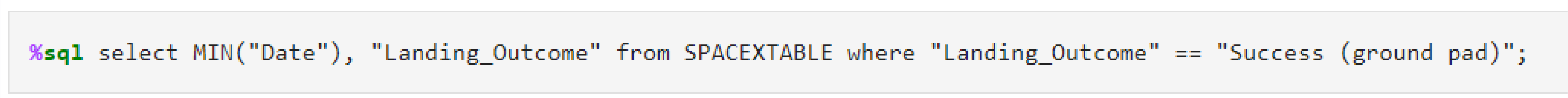
Booster_Version	AVG(PAYLOAD_MASS__KG_)
F9 v1.1 B1003	2534.6666666666665

- Explanation

Use AVG to calculate the average value; use “Booster_Version” LIKE “F9 v1.1%”to specify the booster version we want

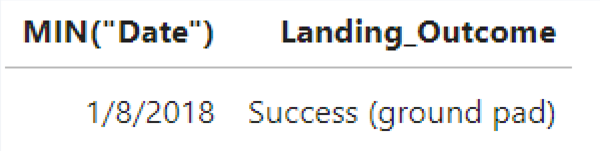
First Successful Ground Landing Date

- Query



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

- Result



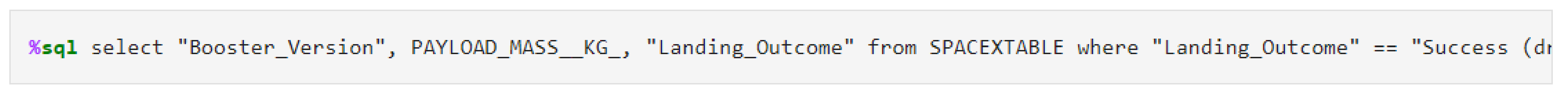
MIN("Date")	Landing_Outcome
1/8/2018	Success (ground pad)

- Explanation

Use MIN to find the earliest date; use the WHERE clause to specify that the landing outcome is in ground pad

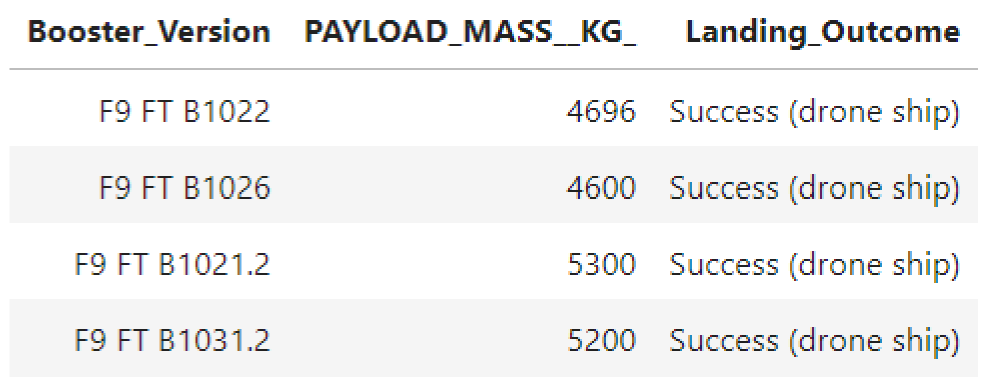
Successful Drone Ship Landing with Payload between 4000 and 6000

- Query



```
%sql select "Booster_Version", PAYLOAD_MASS__KG_, "Landing_Outcome" from SPACEXTABLE where "Landing_Outcome" == "Success (dr
```

- Result



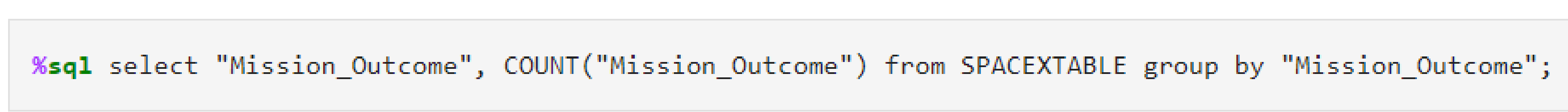
Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

- Explanation

Use the WHERE clause to limit the “Landing_Outcome” and PAYLOAD_MASS_kg_

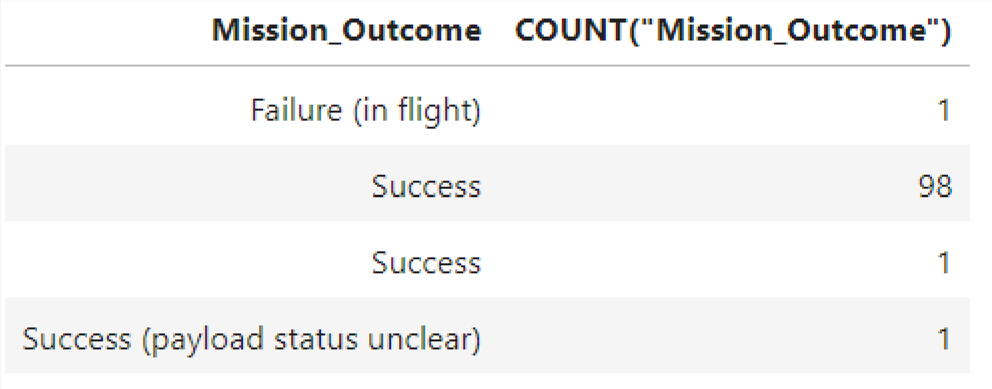
Total Number of Successful and Failure Mission Outcomes

- Query



```
%sql select "Mission_Outcome", COUNT("Mission_Outcome") from SPACEXTABLE group by "Mission_Outcome";
```

- Result



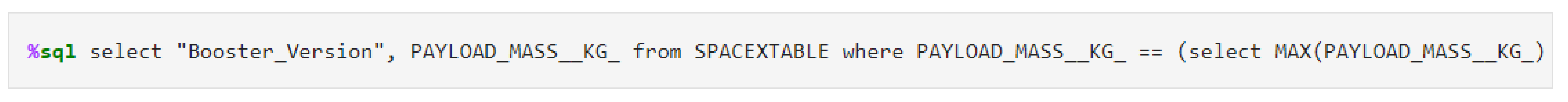
Mission_Outcome	COUNT("Mission_Outcome")
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- Explanation

Use the COUNT to get the total number of mission outcomes; use GROUP BY to have the results counted based on the mission outcomes

Boosters Carried Maximum Payload

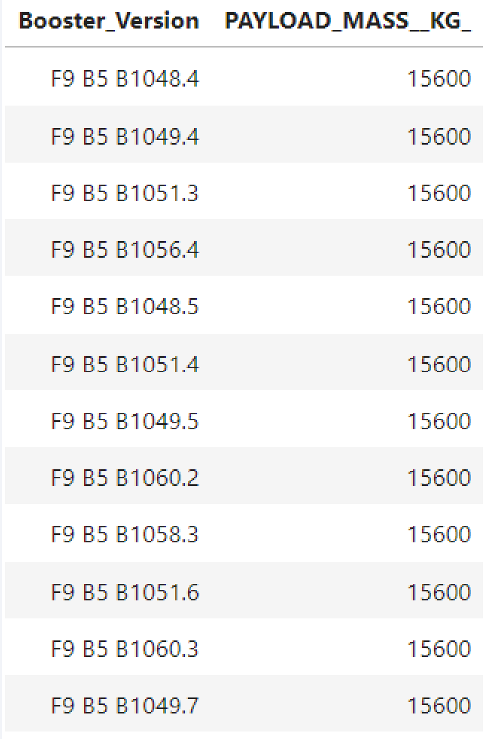
- Query



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

  - Full query: %sql select "Booster_Version", PAYLOAD_MASS__KG_ from SPACEXTABLE where PAYLOAD_MASS__KG_ == (select MAX(PAYLOAD_MASS__KG_) from SPACEXT

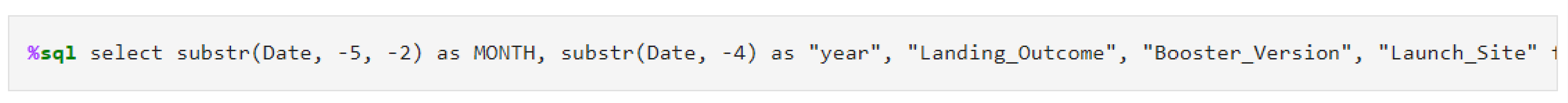
- Result



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

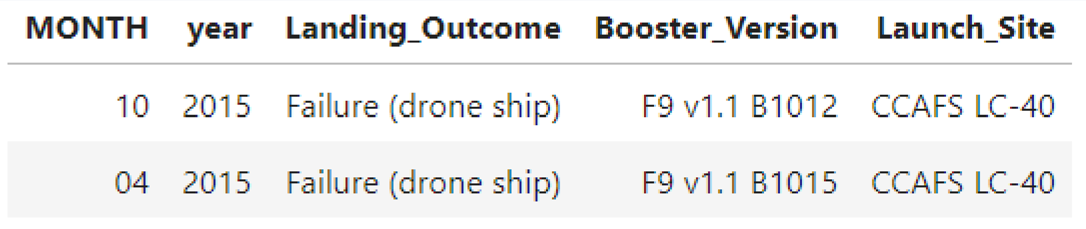
2015 Launch Records

- Query



```
%sql select substr(Date, -5, -2) as MONTH, substr(Date, -4) as "year", "Landing_Outcome", "Booster_Version", "Launch_Site" f
```

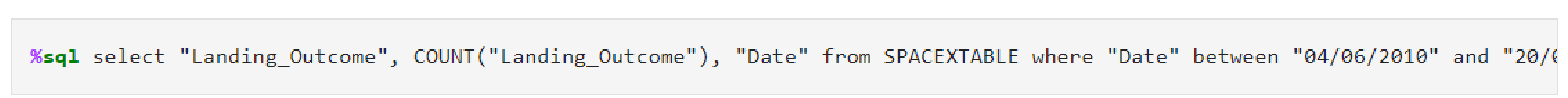
- Result



MONTH	year	Landing_Outcome	Booster_Version	Launch_Site
10	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

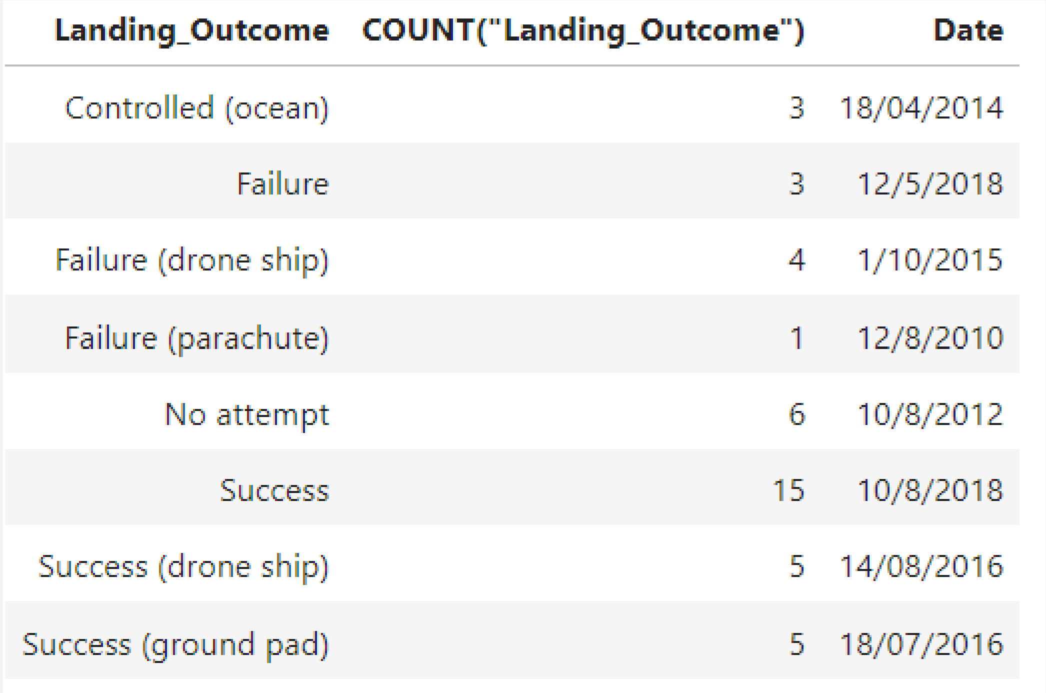
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- Query



```
%sql select "Landing_Outcome", COUNT("Landing_Outcome"), "Date" from SPACEXTABLE where "Date" between "04/06/2010" and "20/
```

- Result

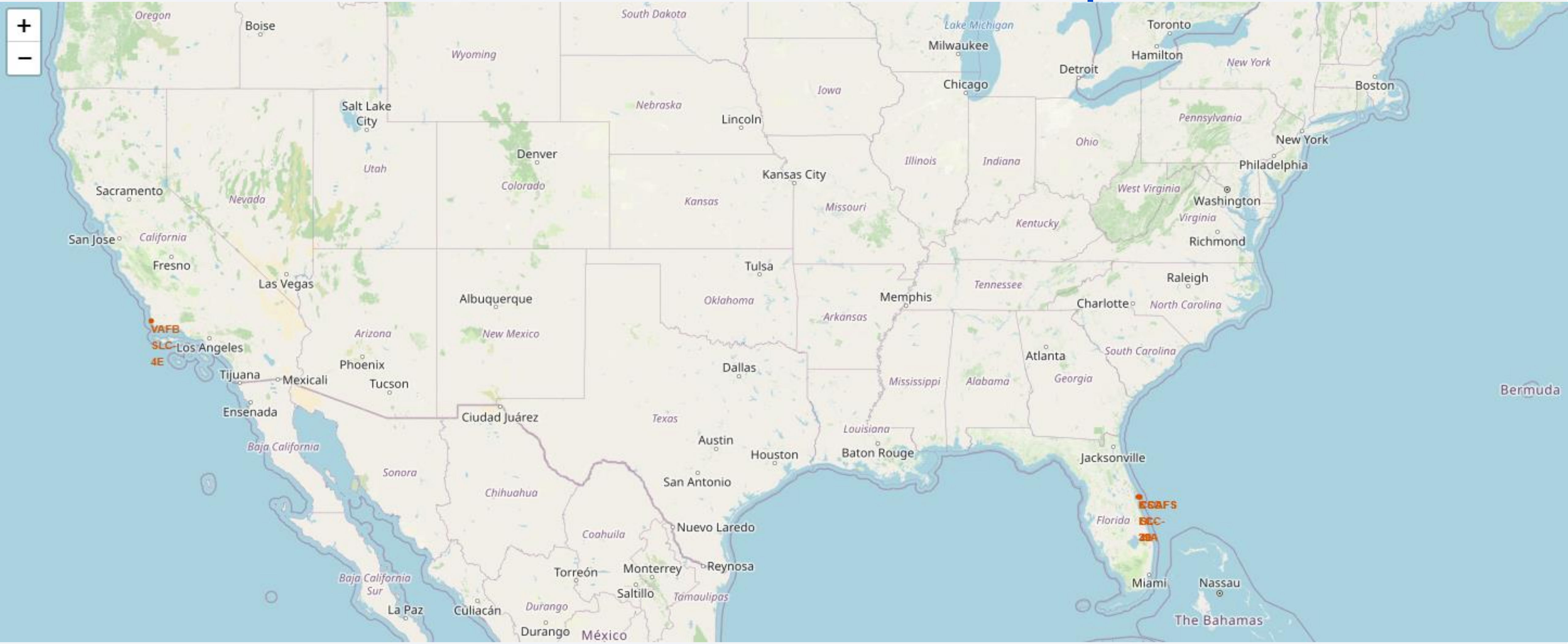


Landing_Outcome	COUNT("Landing_Outcome")	Date
Controlled (ocean)	3	18/04/2014
Failure	3	12/5/2018
Failure (drone ship)	4	1/10/2015
Failure (parachute)	1	12/8/2010
No attempt	6	10/8/2012
Success	15	10/8/2018
Success (drone ship)	5	14/08/2016
Success (ground pad)	5	18/07/2016

Section 3

Launch Sites Proximities Analysis

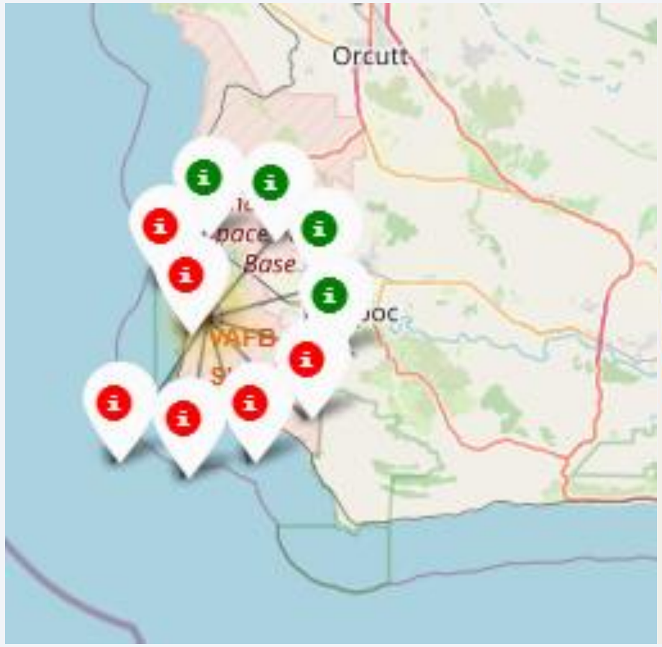
Mark All Launch Sites on a Folium Map



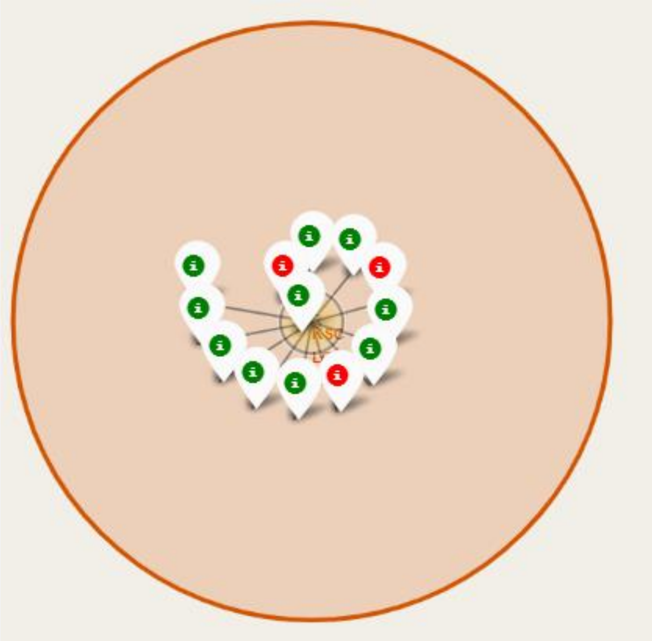
- Are all launch sites in proximity to the Equator line? No
- Are all launch sites in very close proximity to the coast? Yes

Mark Success/Failed Launches For Each Site

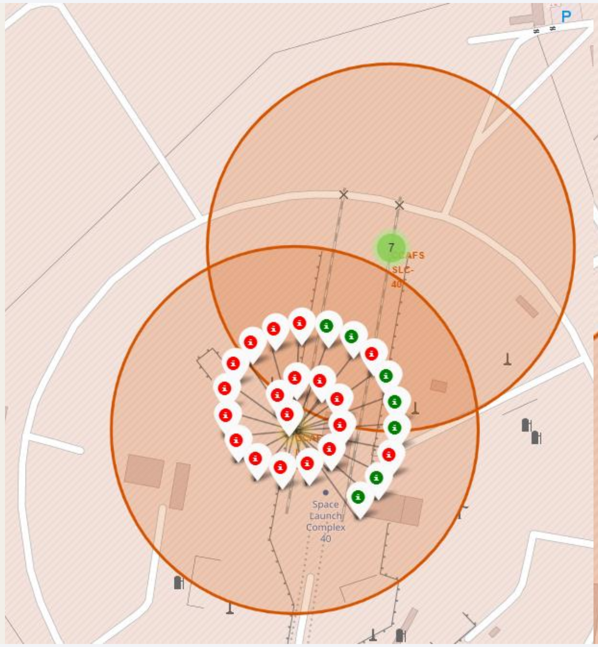
VAFB SLC-4E



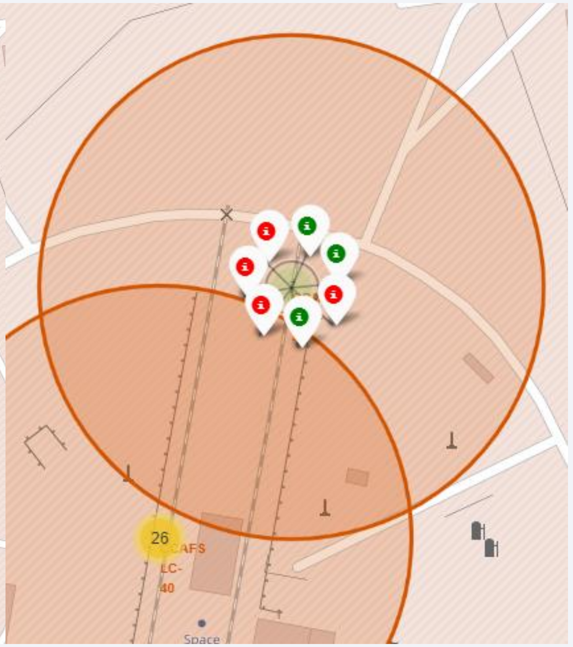
KSC LC-39A



CCAFS LC-40

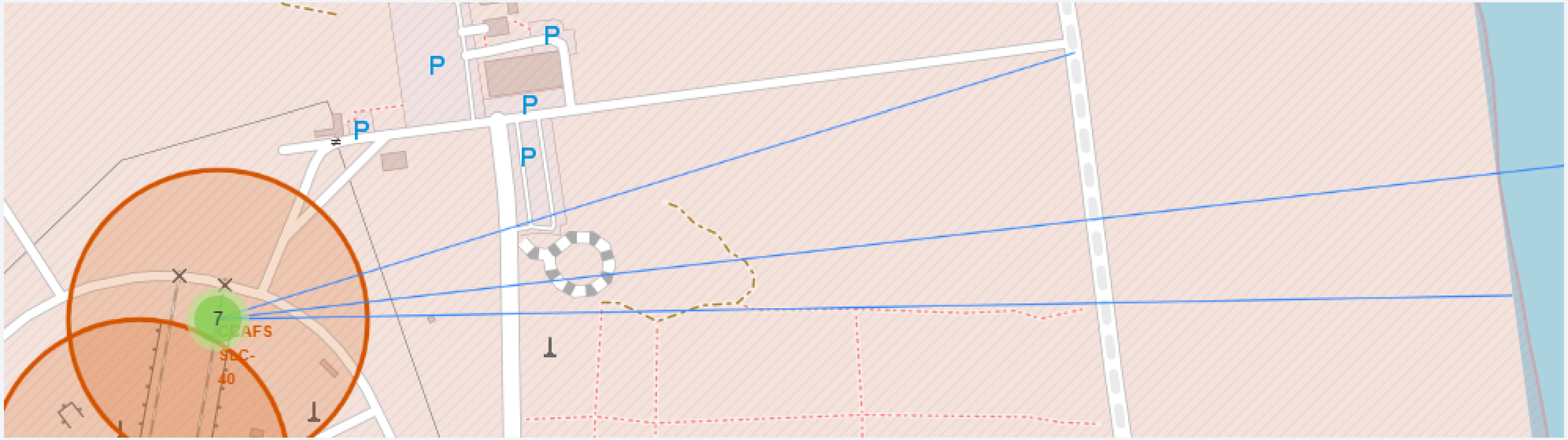


CCAFS SLC-40

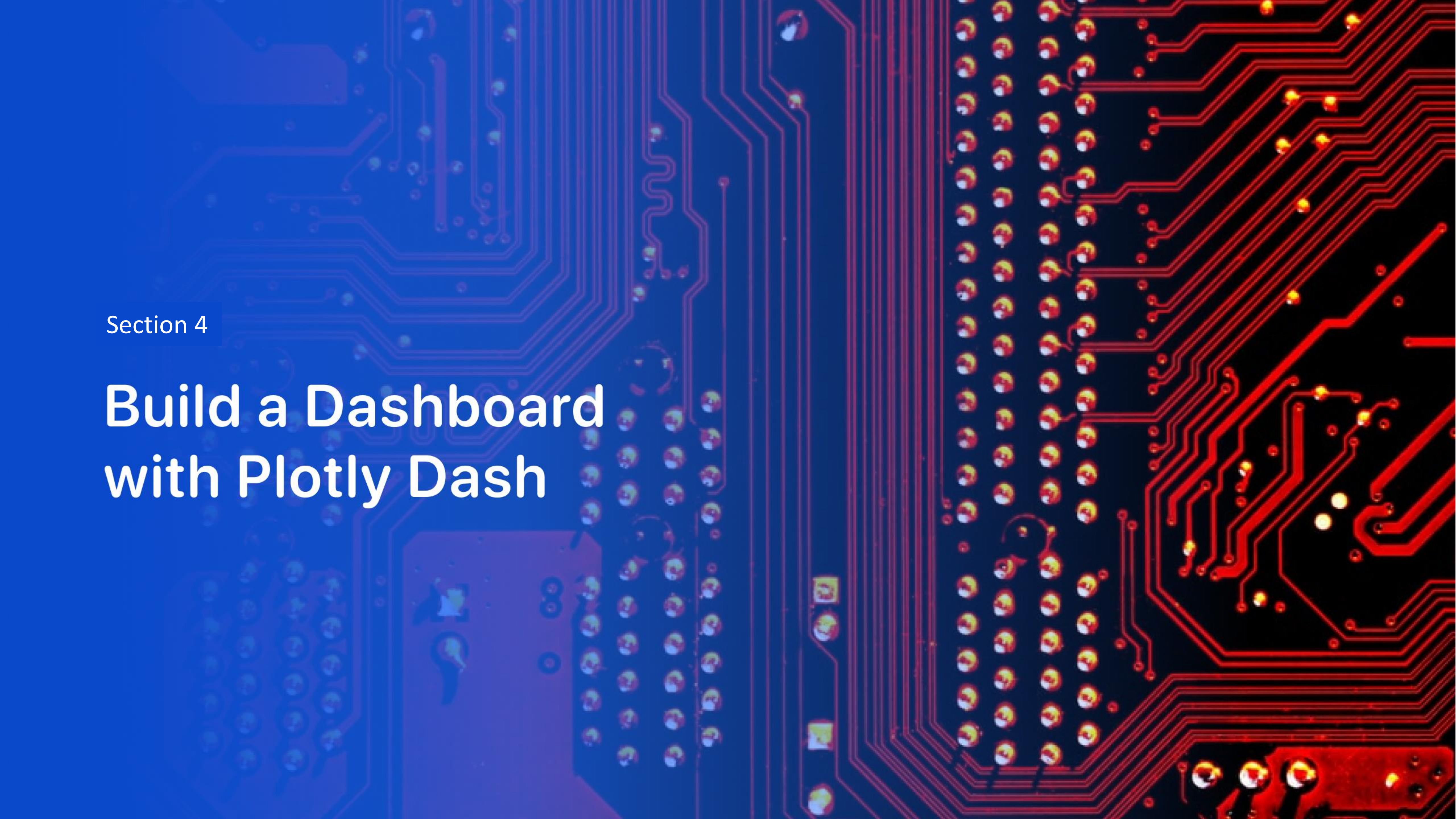


KSC LC-39A and CCAFS LC-40 have relatively high success rate.

Distances Between A Launch Site to Its Proximities



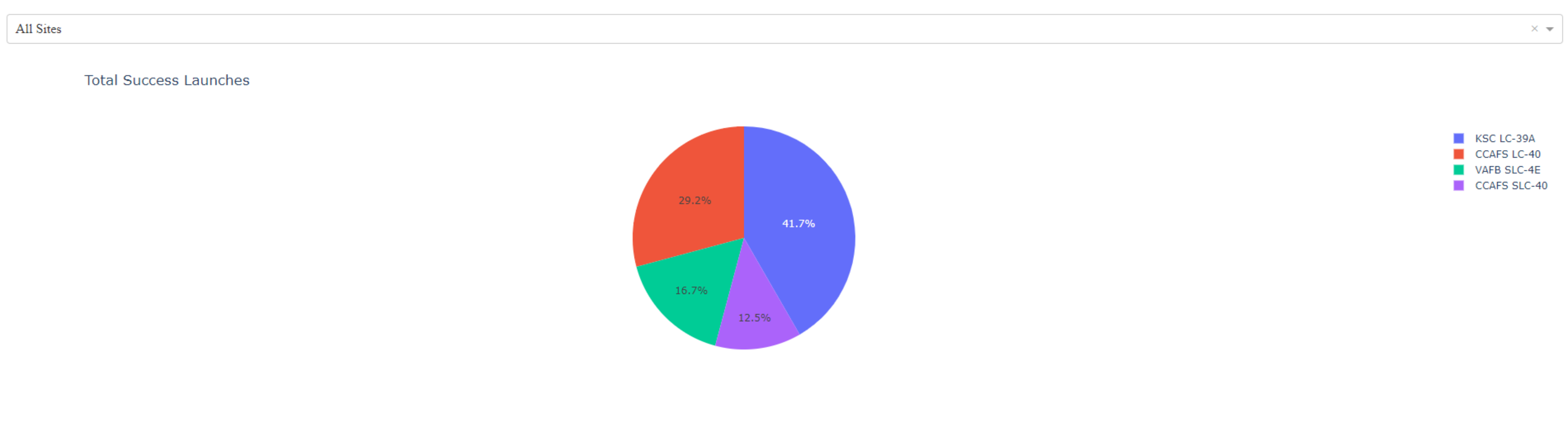
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes



Section 4

Build a Dashboard with Plotly Dash

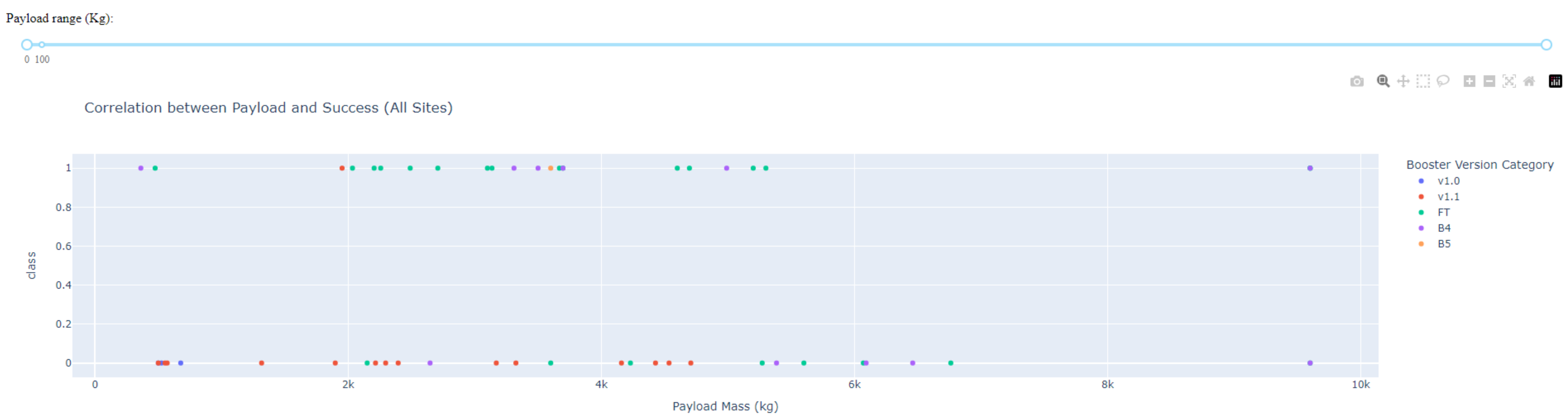
Total Success Launches For All Sites



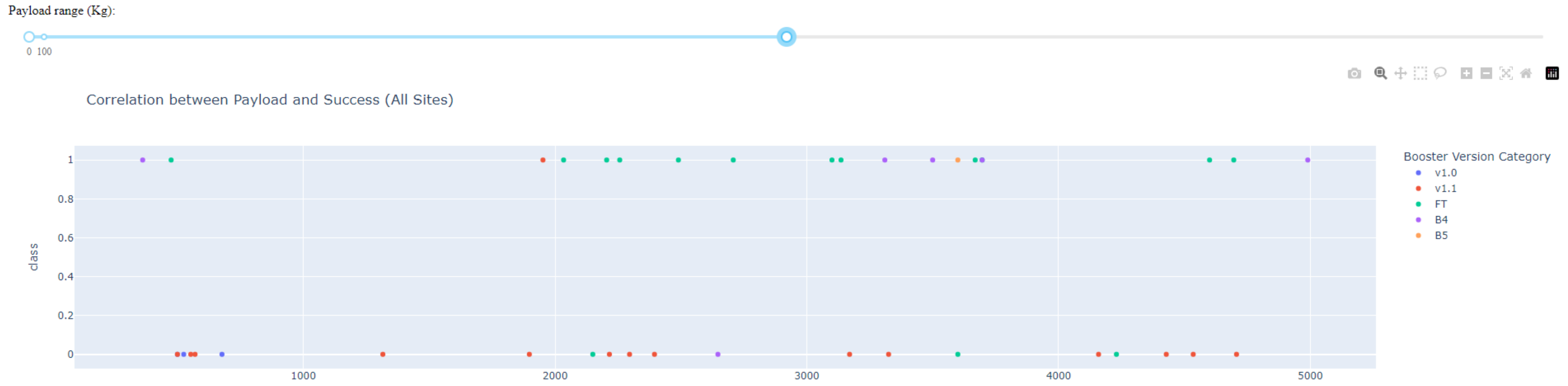
All Sites

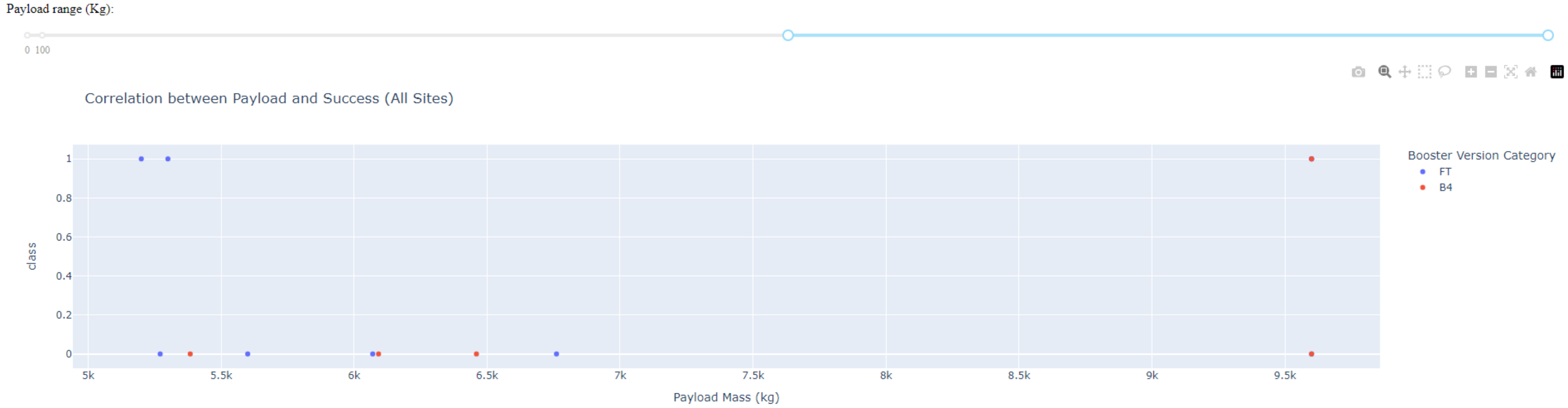
KSC LC-39A has the most successful launches.

Correlation Between Payload Mass and Success



Correlation Between Payload Mass and Success



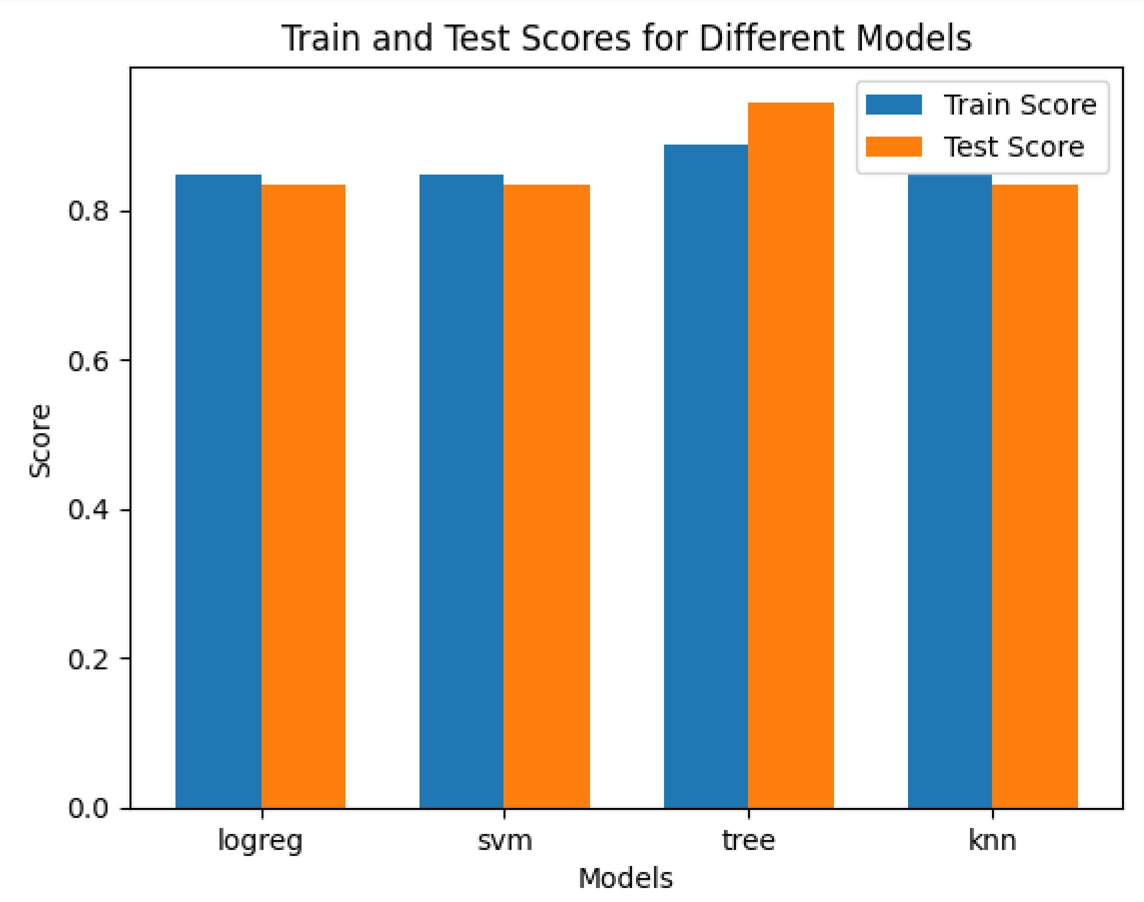


Overall, the success rate for lower payload range is higher than that for the higher payload range.

Section 5

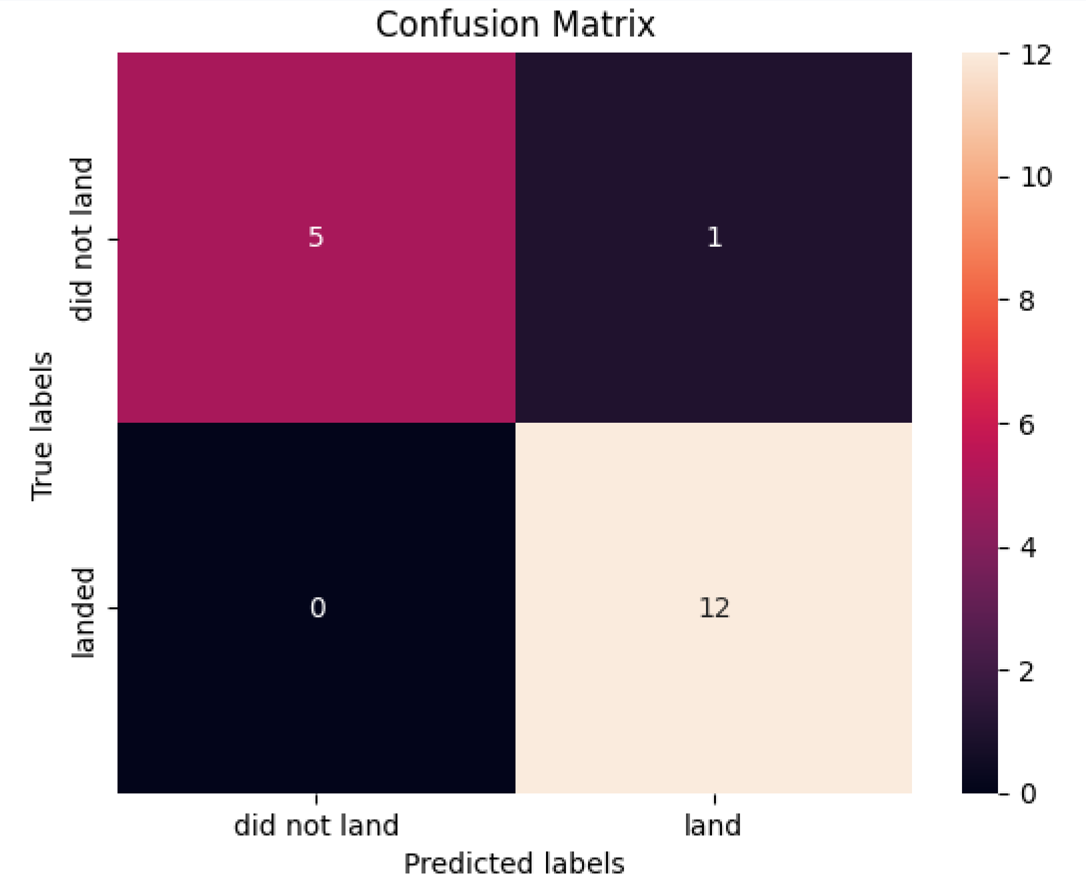
Predictive Analysis (Classification)

Classification Accuracy



- The tree model has the highest accuracy.

Confusion Matrix



- There is 1 FP and 0 FN, which also results in a high recall rate (recall=TP/(TP+FN)=1.0)

Conclusions

- The yearly success rate for SpaceX launches increases as years pass by.
- ES-L1, GEO, HEO and SSO have the highest success rate
- Low payload mass result in a better outcome than the heavier payload mass.
- KSC LC-39A has the highest successful launches.
- The tree classifier makes the best predication on the launch outcome.

Appendix

- Full list to the datasets used for this SpaceX project and the corresponding notebooks: https://github.com/jlmaurora233/IBM_DS

Thank you!