IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Michail Milioritsas

21/09/2022

Outline

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

Executive Summary

- The project scope is to predict the successful landing of SpaceX Falcon 9 first stage rockets.
- To this end data from past SpaceX launches are collected, pre-processed, visualized and explored. The most influential parameters are recognized and used for building and evaluating predictive models.
- Four Classifiers are evaluated regarding their ability to predict the landing outcome: the k-Nearest Neighbors, the Support Vector Machine, the Decision Tree and the Logistic Regression models, all available from the sci-kit learn python package.
- All classifiers are found to predict the landing outcomes with a same accuracy score of about 83%, probably due to small dataset size.

Introduction

- **General information**

  - SpaceX advertises Falcon 9 rocket launch cost to $65 million.

  - Competitors estimate cost to be $165 million.

  - SpaceX cost is lower due to the reuse of the first stage partition of the rockets.

- **Research question**

  - Can the cost of a launch be determined?

  - Need to answer whether the first stage rocket will land successfully (unharmed).

  - Development of predictive models is required, trained on past launch data.

- **Goal**

  - Provide other companies with insight for competing in the sector.

Section 1

Methodology

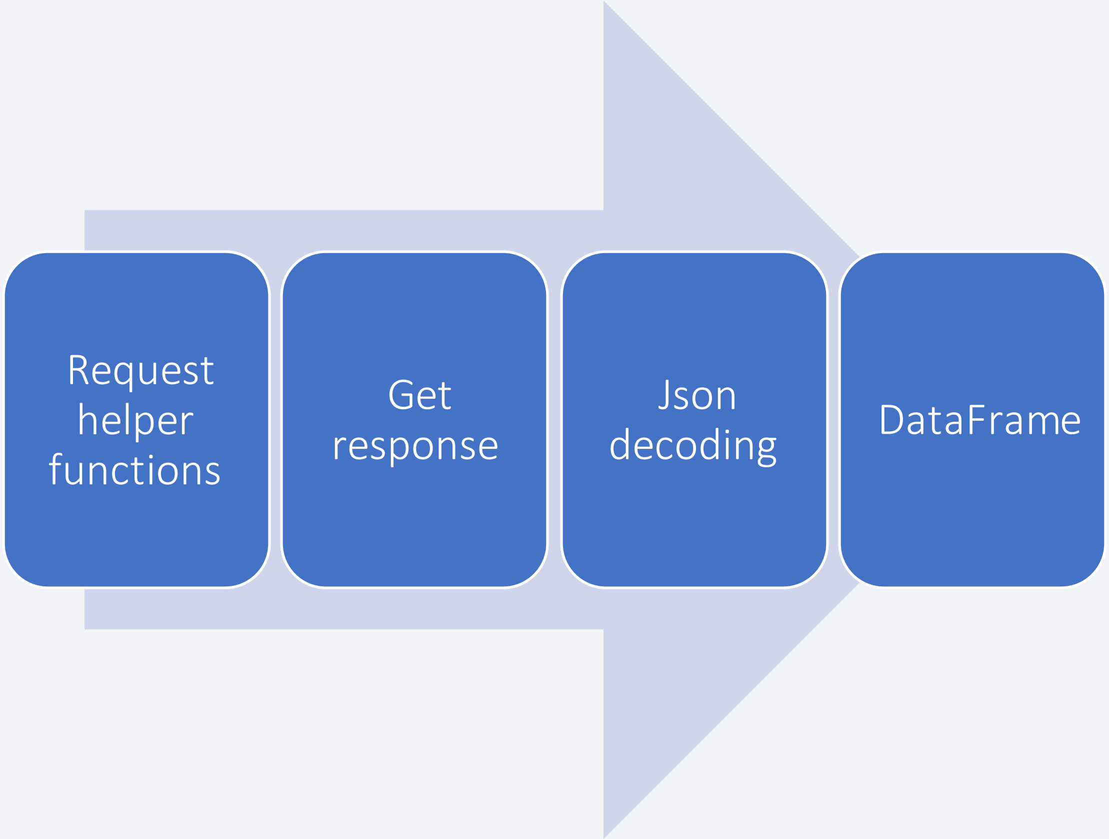
Methodology

- Data collection methodology:
  - Data was collected from several nodes of the SpaceX API and from wikipedia tables.
- Perform data wrangling:
  - Data was fetched and pre-processed with the python Pandas, the requests and the BeautifulSoup libraries.
- Exploratory data analysis (EDA) performed using visualization tools and SQL.
  - Visualization: python Matplotlib, Seaborn.
- Interactive visual analytics performed using Folium and Plotly Dash.
- Predictive analysis performed using classification models
  - Predictor data normalized with StandardScaler and all data split with train_test_split.
  - Classifiers used: k-NN, SVC, Decision Tree, Logistic Regression.
  - Best estimators selected with GridSearchCV and evaluated with score() method.

Data Collection – SpaceX API

- Calling several SpaceX nodes with the requests python library.
- Use of helper functions to extract data from different nodes.
- Response in json format needs to be decoded and converted to dataframe with methods json() and json.normalize().

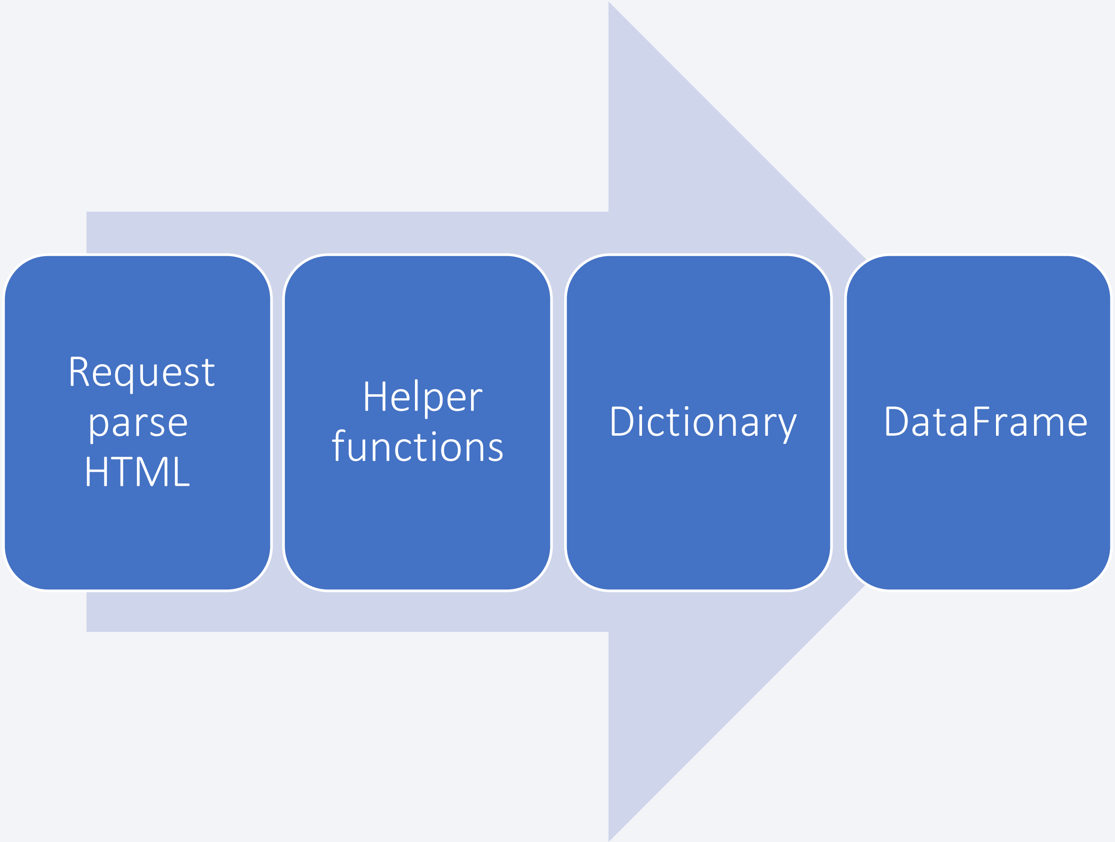
- GitHub URL of the SpaceX API calls notebook



Data Collection - Scraping

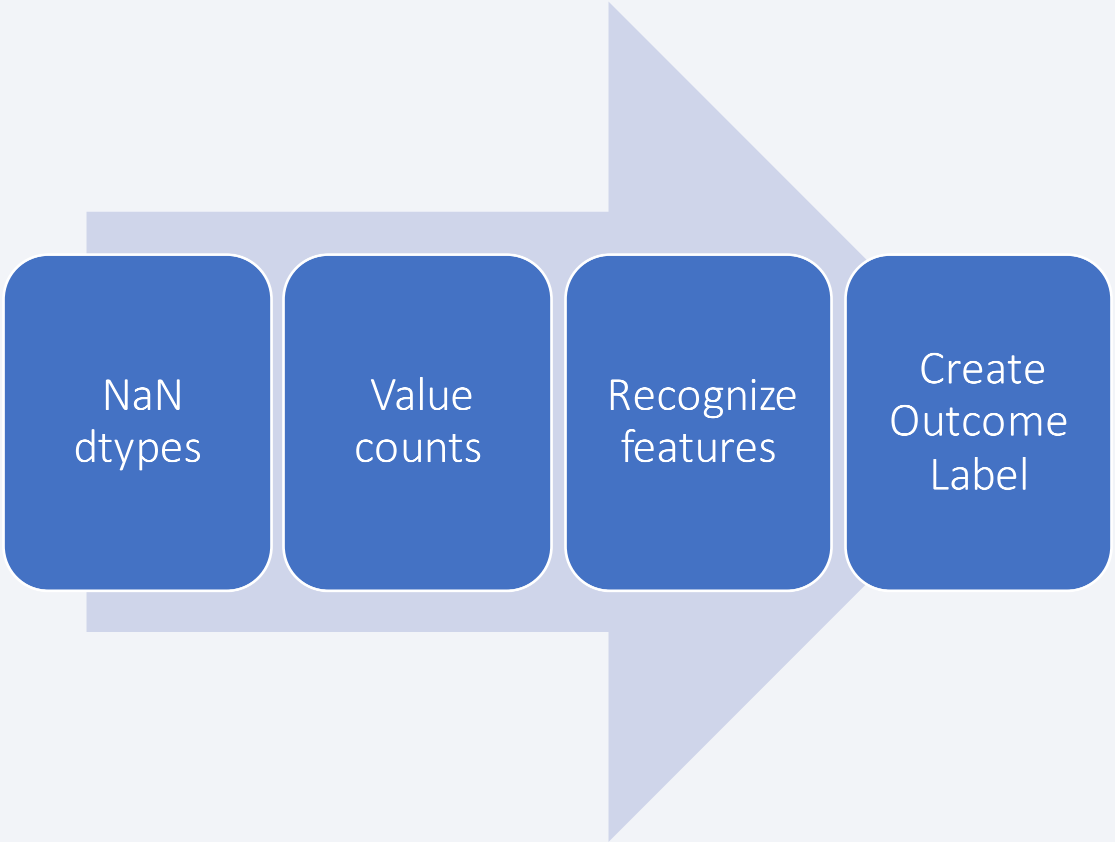
- Request and parse HTML file with python BeautifulSoup.
- Helper functions to extract info from scraped tables columns.
- Create python DataFrame from dictionary.

- GitHub URL web scraping notebook



Data Wrangling

- DataFrames - identify null values and data types.
- Count value appearances.
- Recognize launch sites and launch orbits.
- One-hot-encoding of landing outcomes (success/failure) and label creation.
- GitHub URL data wrangling notebook



EDA with Data Visualization

- Produced scatter plots in order to recognize the effect of the features on landing outcome and to explore whether they can be used for prediction.
  - Flight No. vs Payload, Flight No. vs Launch site, Payload vs. Launch Site etc., with respect to landing outcome.
- Bar chart of Orbits vs Landing success rate, in order to show orbit influence on outcome.
- Line plot of Time vs success rate - effect of time.
- Predictors are selected Based on conclusions drawn from plots .

- GitHub URL EDA with data visualization notebook.

EDA with SQL

- Performed SQL queries:
  1. Displaying the names of the unique launch sites in the space mission.
  2. Showing 5 records where launch sites begin with the string 'CCA'.
  3. Displaying the total payload mass carried by boosters launched by NASA (CRS).
  4. Showing average payload mass carried by booster version F9 v1.1.
  5. Listing the date when the first successful landing outcome in ground pad was achieved.
  6. Listing the boosters names with success in drone ship and with payload mass greater than 4000 but less than 6000.
  7. Listing the total number of successful and failure mission outcomes.
  8. Listing the names of the booster versions which have carried the maximum payload mass.
  9. Listing the records for year 2015 which displaying several features.
  10. Ranking the count of successful landing outcomes between two dates.
- GitHub URL of EDA with SQL notebook

Build an Interactive Map with Folium

- Objects added to Folium Map:
  - Circles and markers showing position of and information on launch sites.
  - Marker Clusters showing the successful/failed launches for each launch site.
  - Lines and markers showing the distances of launch sites from existing infrastructure.
- Folium objects contribute to understanding the effect of launch site location and launch site closeness to facilities (roads, railroads, cities, etc.) on landing outcome.

- GitHub URLs of Folium map notebook and produced images.

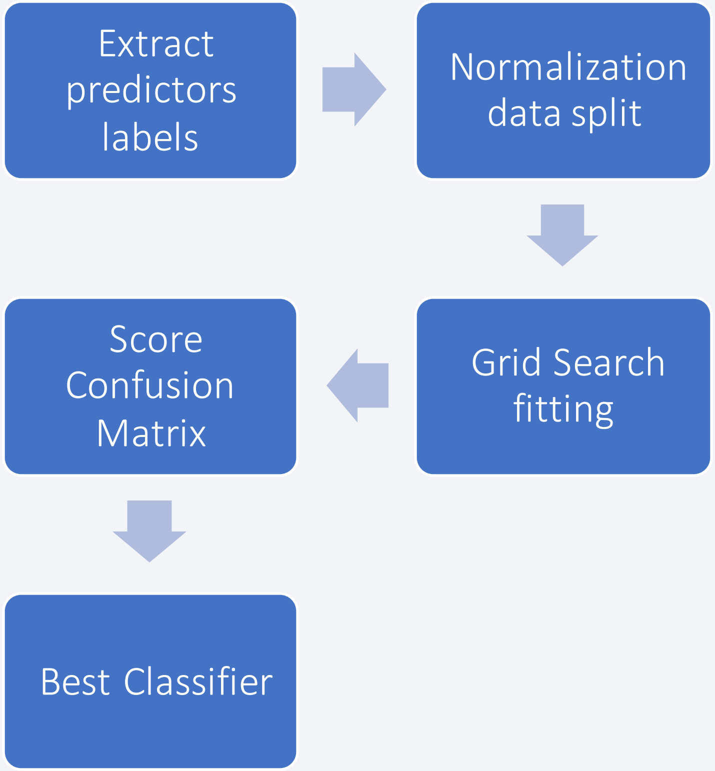
Build a Dashboard with Plotly Dash

- Plots and interactions added to Dashboard:
  - Pie charts showing success rates for all sites and success/failure rates for each site.
  - Scatter plot demonstrating successful/failed launches based on Payload mass and Booster version.
  - Dropdown menu for showing pie charts, scatterplots for one site or for all sites.
  - Range slider for choosing Payload mass range in scatterplots.
- Plots and interactions help recognizing which launch site locations, payload mass and booster versions have the highest/lowest launch success rates.

- GitHub URL of the Plotly Dash .py file.

Predictive Analysis (Classification)

- Extracting predictors and target variable.
- Normalize predictor data with StandardScaler().
- Split the dataset into train and test data with train_test_split()
- Finding best estimators with GridSearchCV and fit them with the training data.
- Calculate accuracy score on test data and plot confusion matrices.
- Select the best classifier.
- GitHub URL of predictive analysis lab.



Results

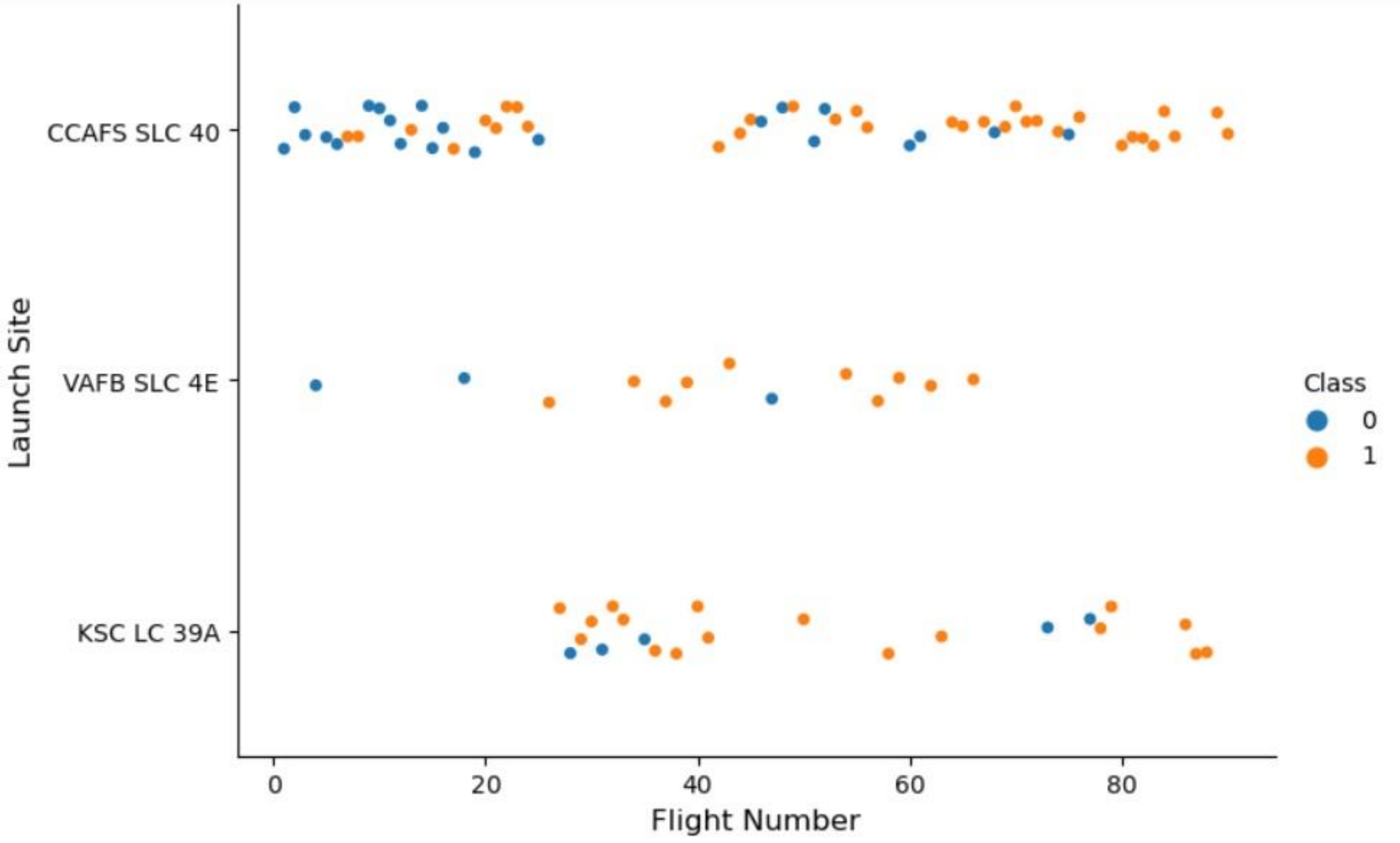
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

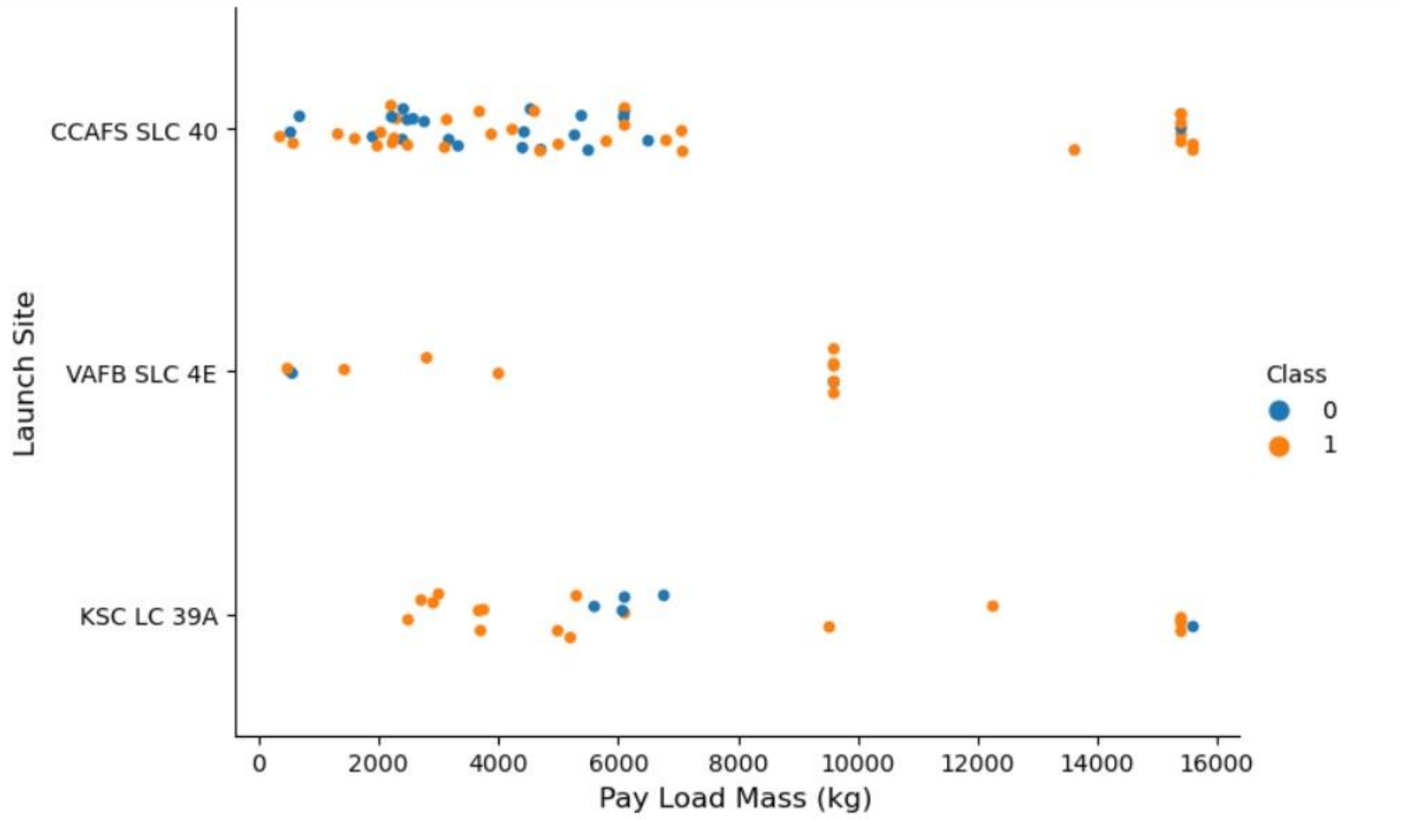
Flight Number vs. Launch Site

- Number of successful launches increases with larger Flight No.
- Larger Fight No. means a later flight, therefore success increases with time.
- All flights with number > 80 are successful.



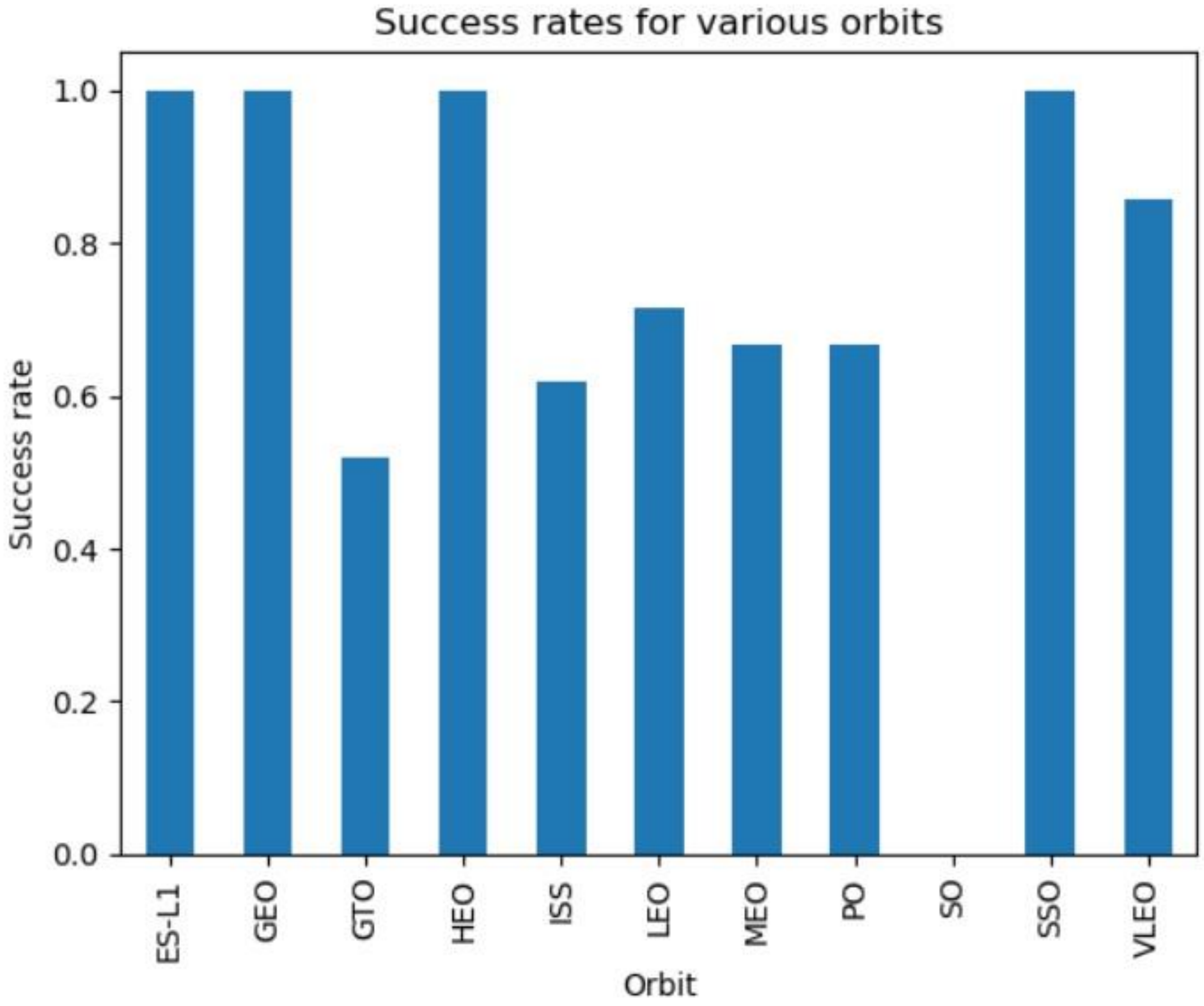
Payload vs. Launch Site

- No launches from VAFB SLC 4E for payload > 10000kg.
- Almost all launches from VAFB SLC 4E are successful.
- Higher success for payload larger than 10000kg.



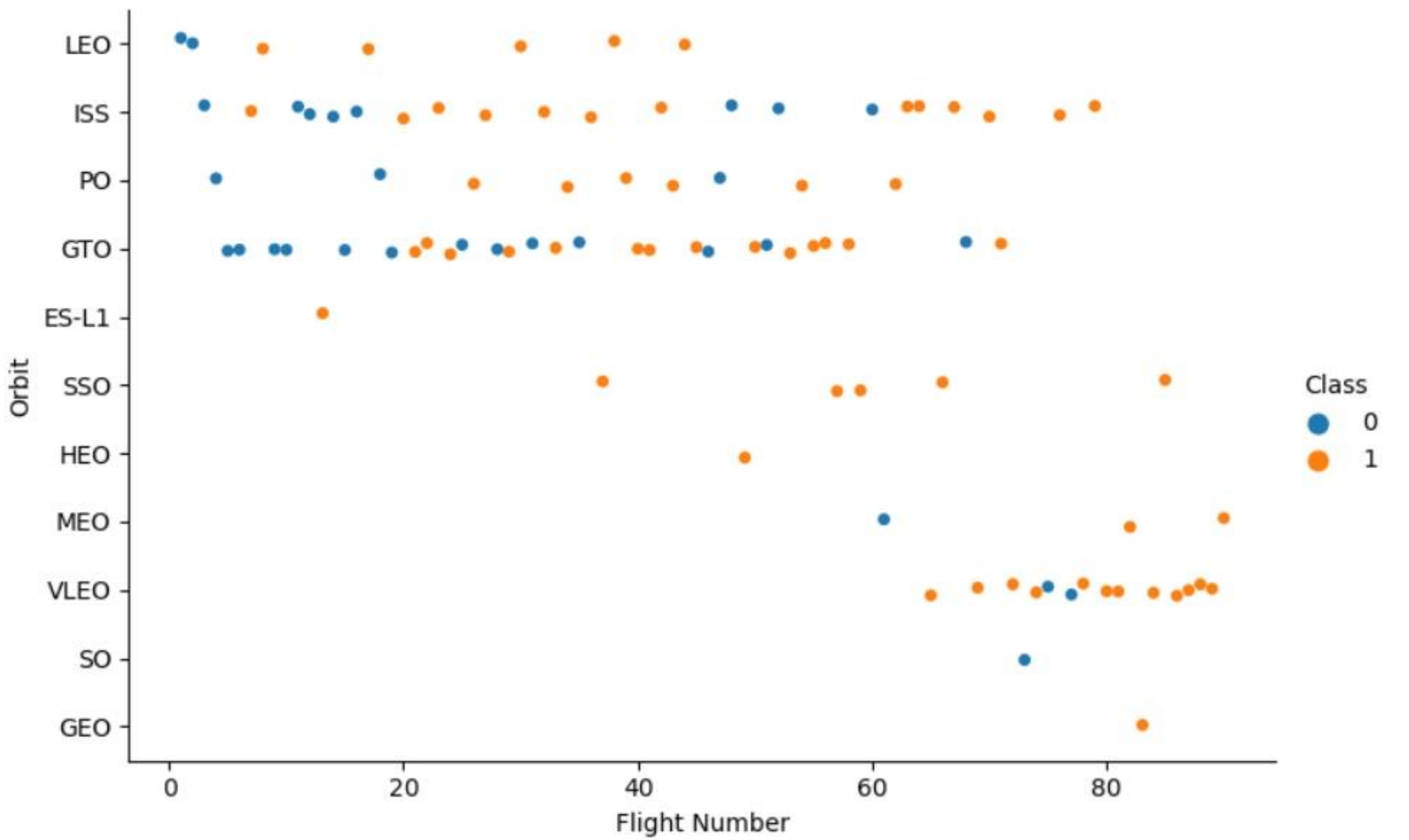
Success rate vs. Orbit Type

- Orbits ES-L1, GEO, HEO, and SSO have100% success rate.
- Orbits ISS, LEO, MEO, PO have similar rates, around 60%.
- Lowest success rate for orbit GTO.
- No data available for SO orbit.



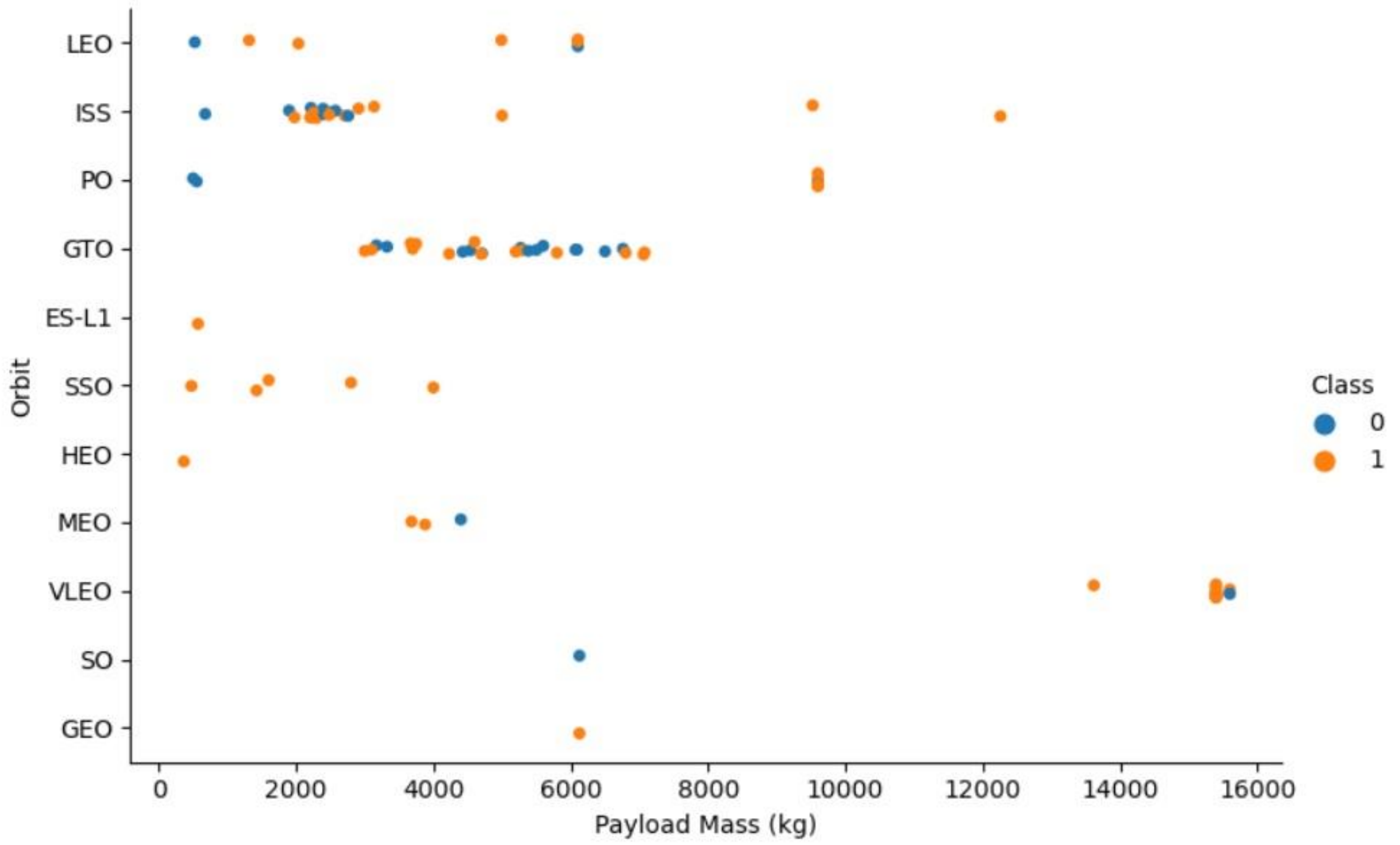
Flight number vs. Orbit Type

- Probably positive relationship between number of flights and success rate in LEO orbit.
- Less clear positive relationship for PO and ISS orbits.
- No clear relationship for orbit GTO.



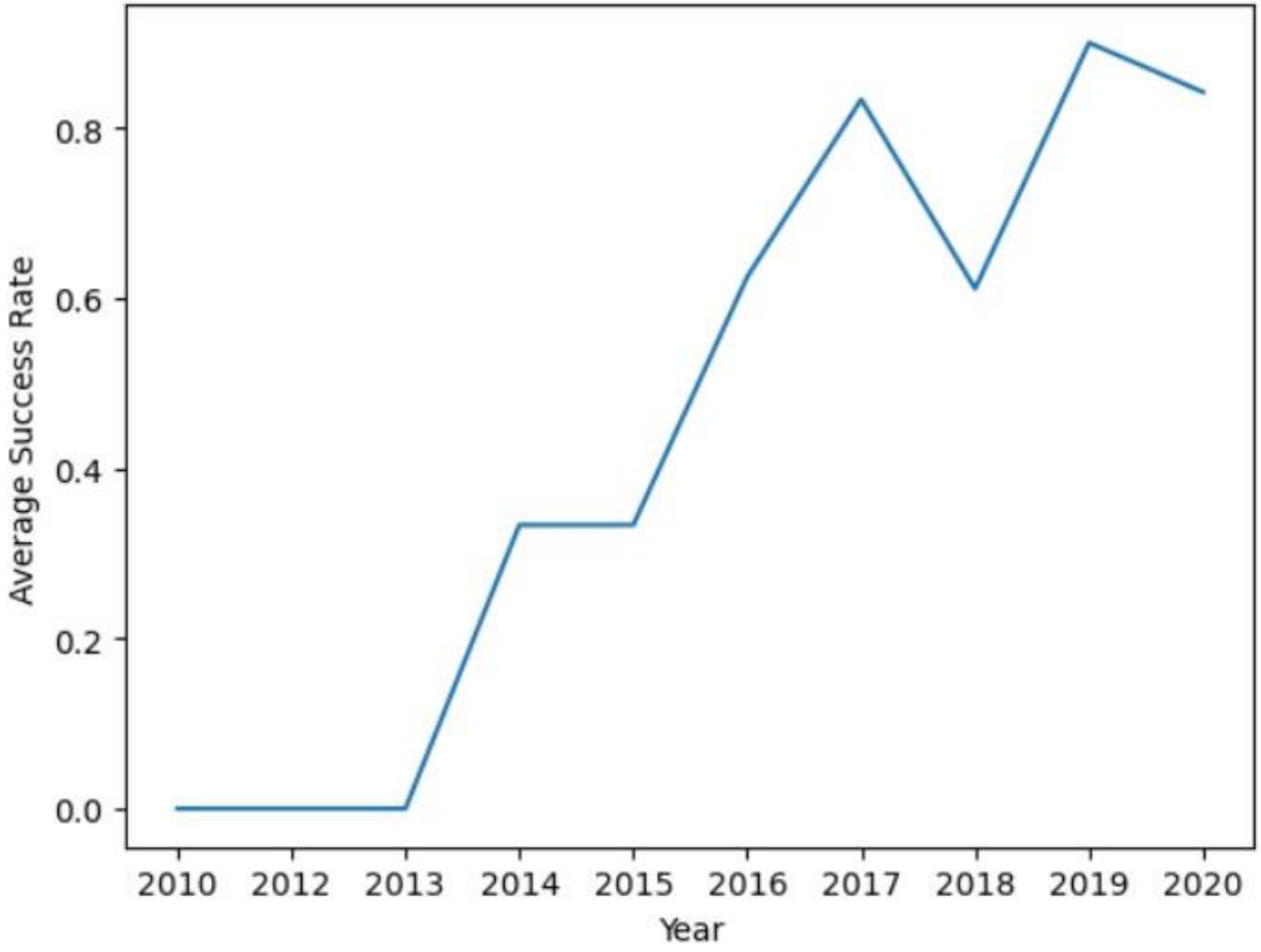
Payload vs. Orbit Type

- Positive relationship between payload mass and success rate in LEO, PO and ISS orbits.
- No clear relationship for GTO orbit .
- Not enough data for other orbits.



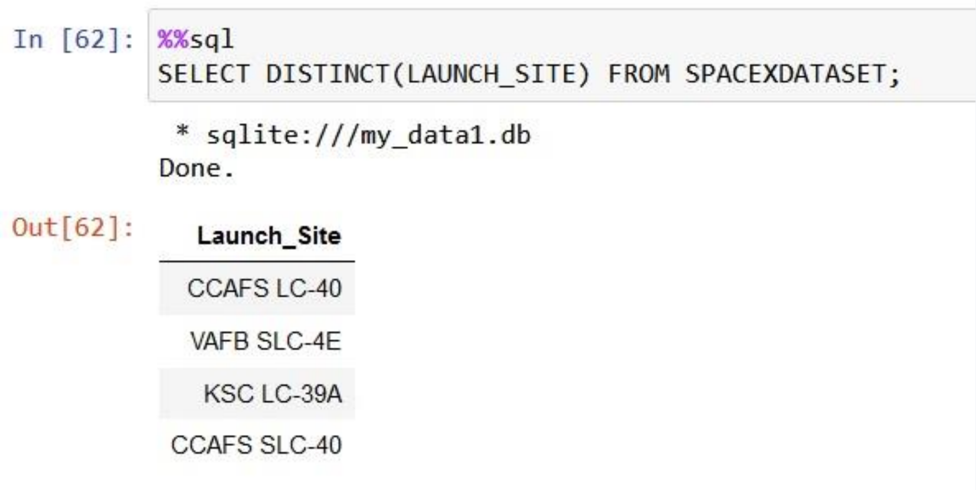
Launch Succes Yearly Trend

- Average success rate increases significantly from 2013 and on.
- Increasing trend temporarily interrupted in 2018 and 2020.



All Launch Site Names

- The names of the unique launch sites:



```
In [62]: %%sql
         SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXDATASET;

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

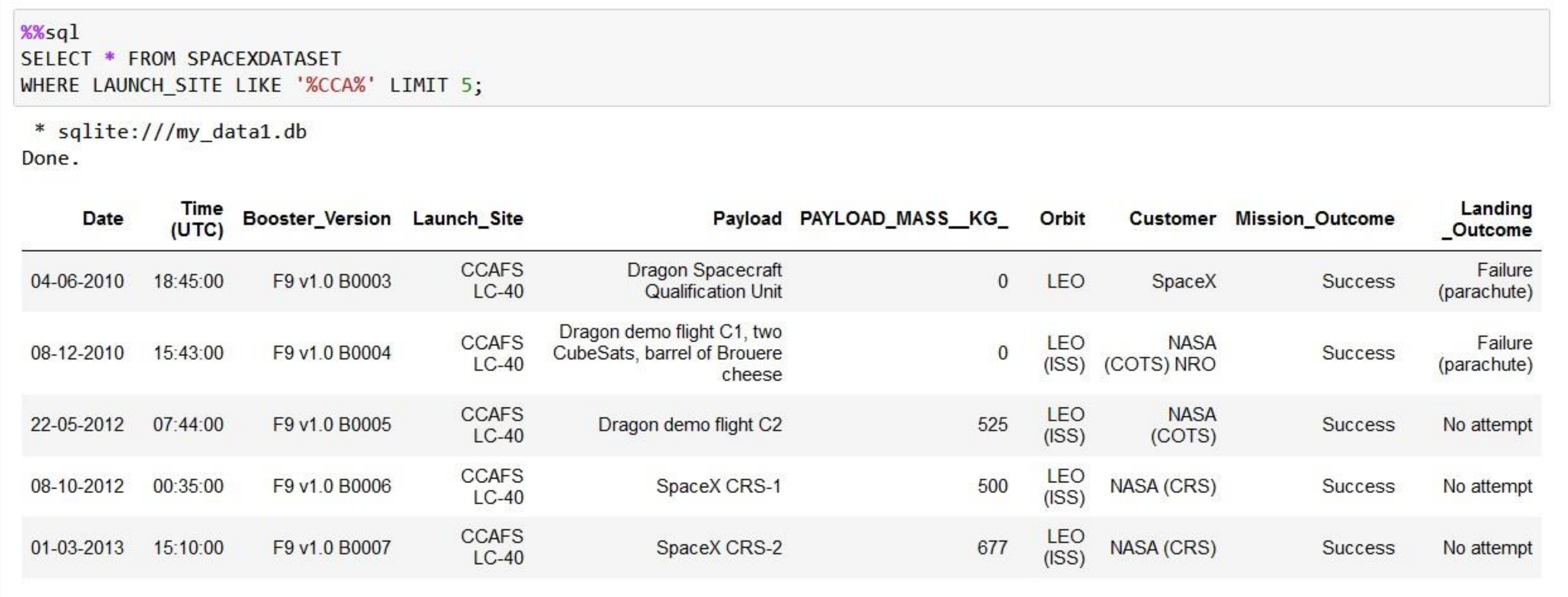
Out[62]:

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

- DISTINCT function used to present unique items from table SPACEXDATASET.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`:



```
%%sql
SELECT * FROM SPACEXDATASET
WHERE LAUNCH_SITE LIKE '%CCA%' LIMIT 5;
```

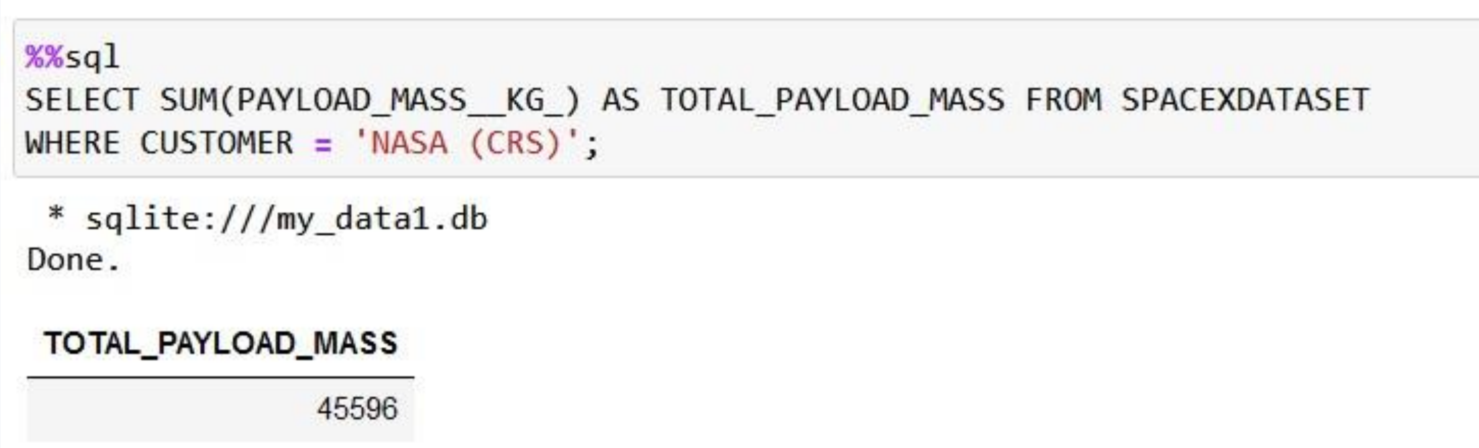
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- WHERE, LIKE and LIMIT queries used to extract data from table.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA (CRS)



```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)';
```

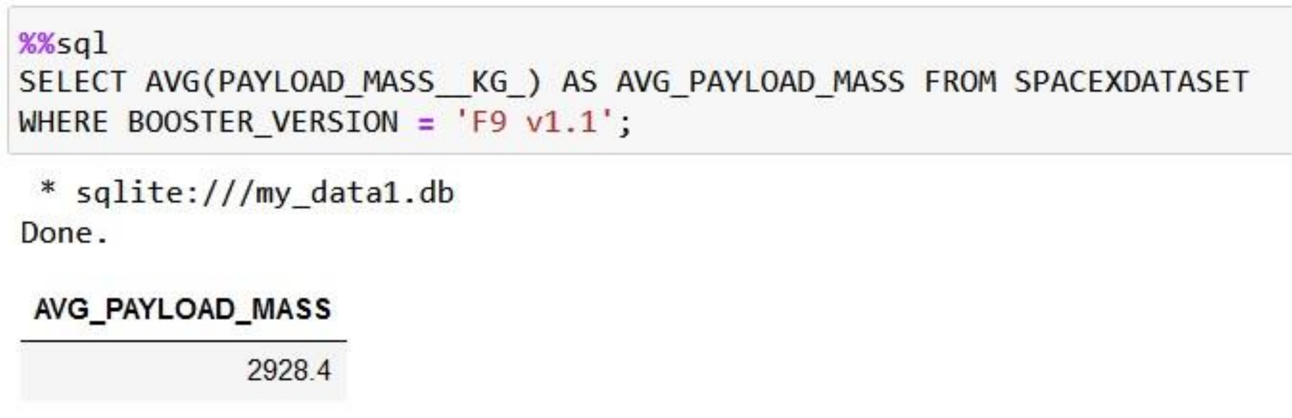
```
 * sqlite:///my_data1.db
Done.
```

TOTAL_PAYLOAD_MASS
45596

- SUM for calculating the total mass and WHERE queries were used.

Average Payload Mass by F9 v1.1

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



```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS FROM SPACEXDATASET
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

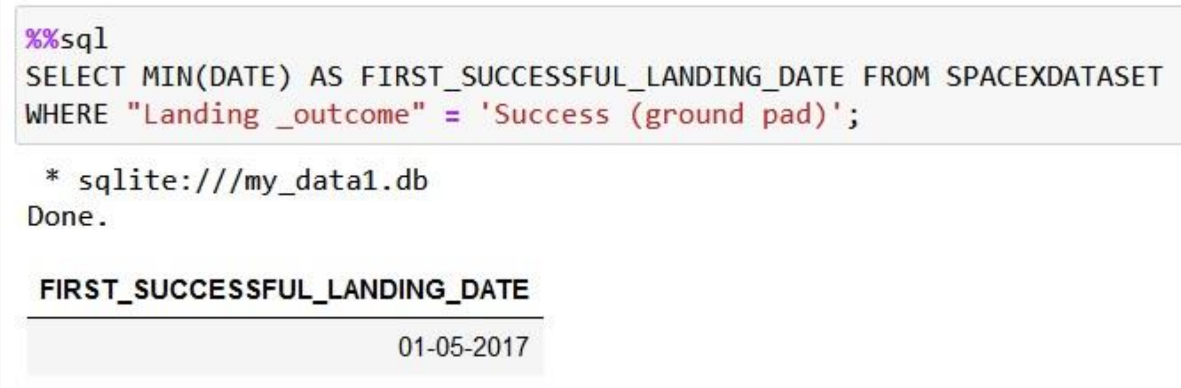
```
 * sqlite:///my_data1.db
Done.
```

AVG_PAYLOAD_MASS
2928.4

- AVG for computing mean mass and WHERE queries used.

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad



```
%%sql
SELECT MIN(DATE) AS FIRST_SUCCESSFUL_LANDING_DATE FROM SPACEXDATASET
WHERE "Landing _outcome" = 'Success (ground pad)';
```

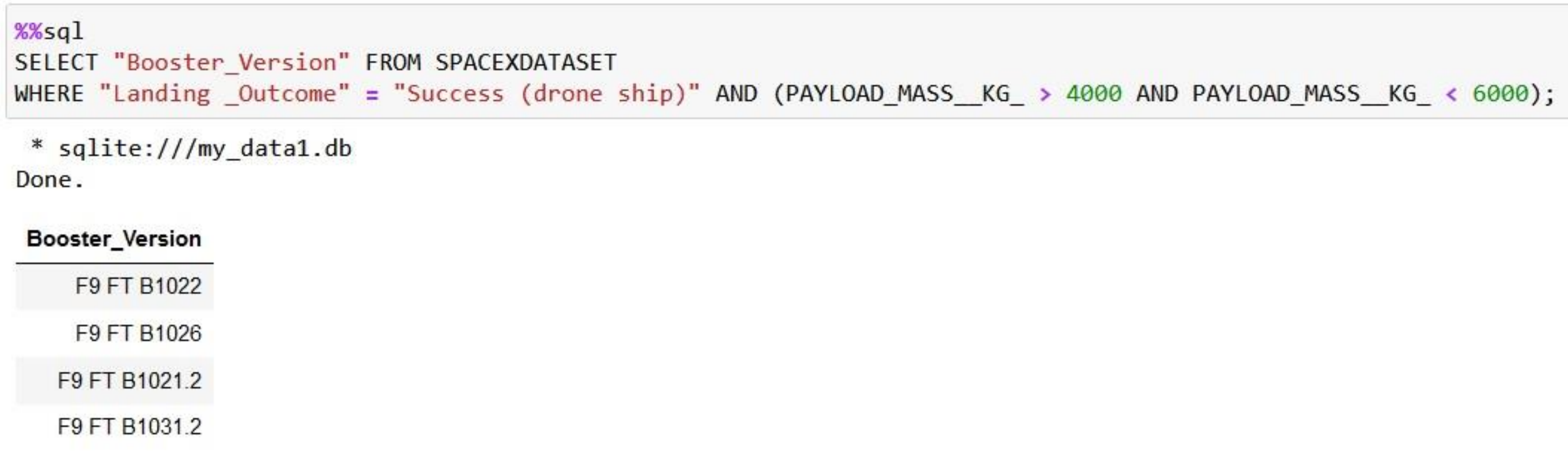
```
 * sqlite:///my_data1.db
Done.
```

FIRST_SUCCESSFUL_LANDING_DATE
01-05-2017

- MIN function used to find smallest date. WHERE query used to locate successful outcomes on ground pads.

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



```
%%sql
SELECT "Booster_Version" FROM SPACEXDATASET
WHERE "Landing _Outcome" = "Success (drone ship)" AND (PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000);
```

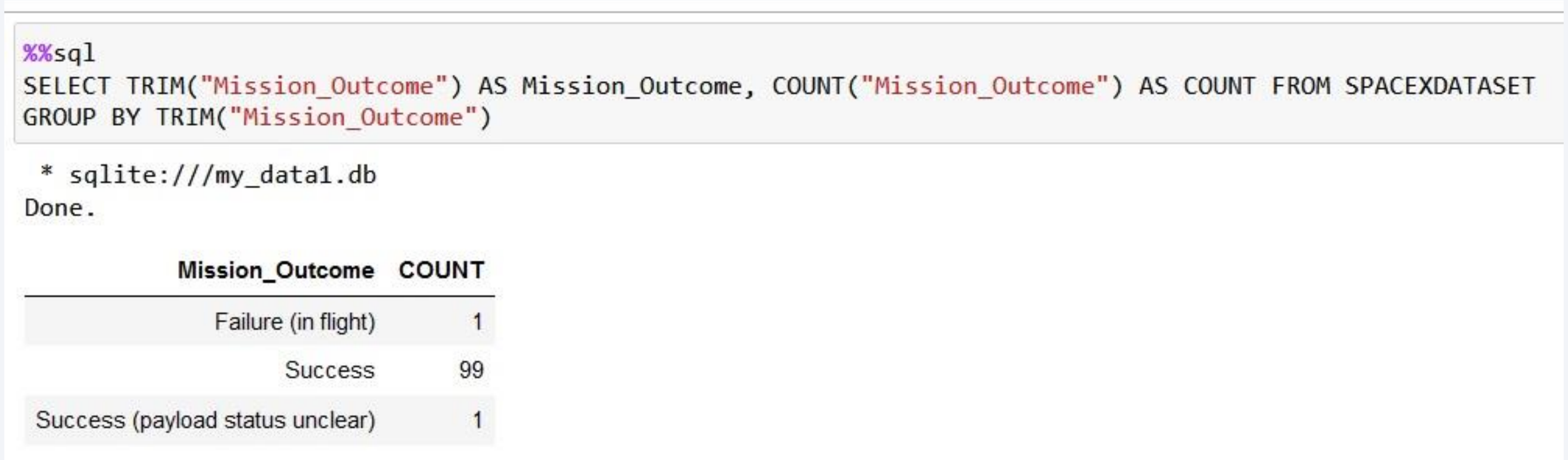
```
 * sqlite:///my_data1.db
Done.
```

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

- WHERE query used to locate successful outcome on drone ship, AND query used to specify payload mass range.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes



```
%%sql
SELECT TRIM("Mission_Outcome") AS Mission_Outcome, COUNT("Mission_Outcome") AS COUNT FROM SPACEXDATASET
GROUP BY TRIM("Mission_Outcome")
```

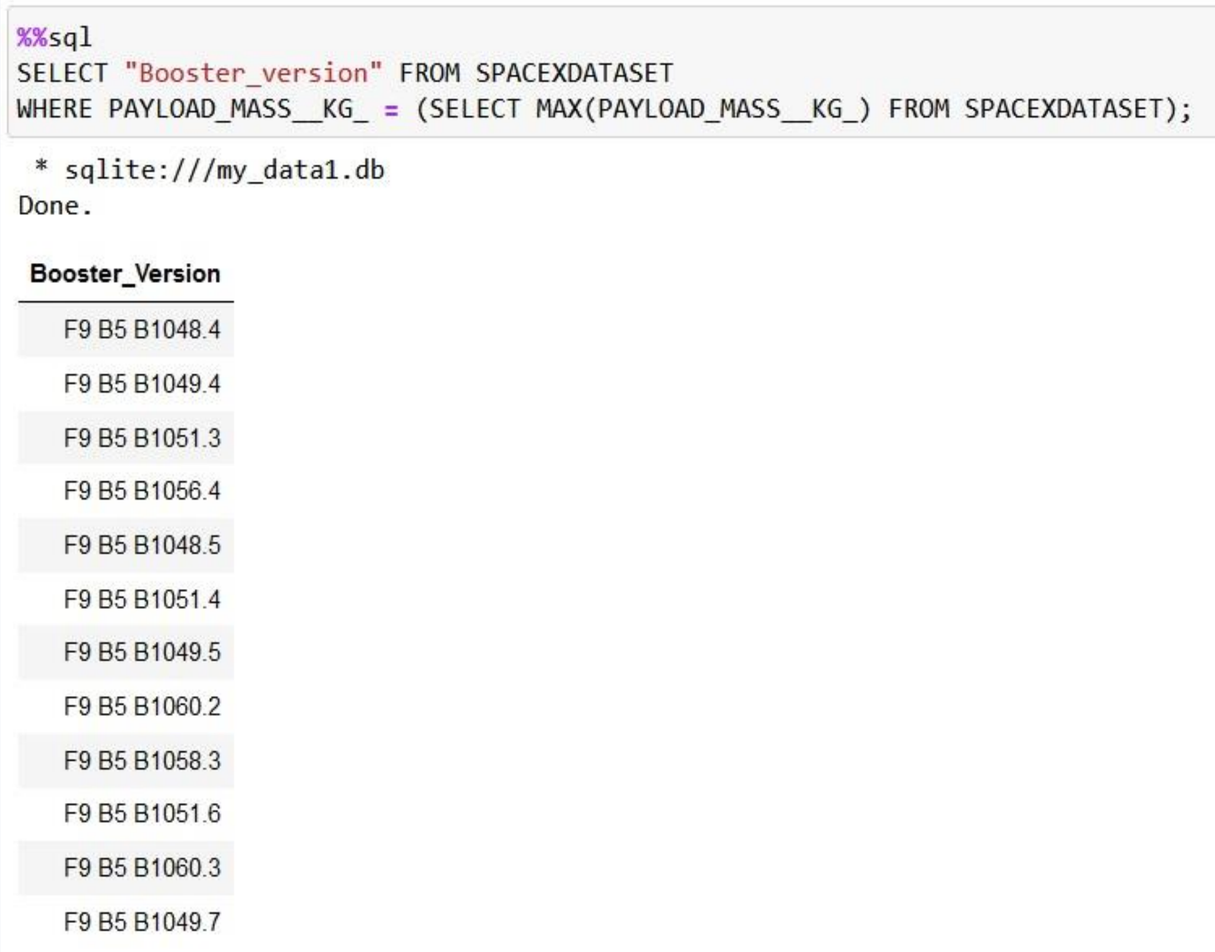
```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- Used TRIM function to remove blank spaces from strings. A record with Mission outcome "Success " (with blank) shows as a separate row.
- COUNT used to count outcomes where outcomes are grouped with GROUP BY.

Boosters Carried Maximum Payload

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



```
%%sql
SELECT "Booster_version" FROM SPACEXDATASET
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET);
```

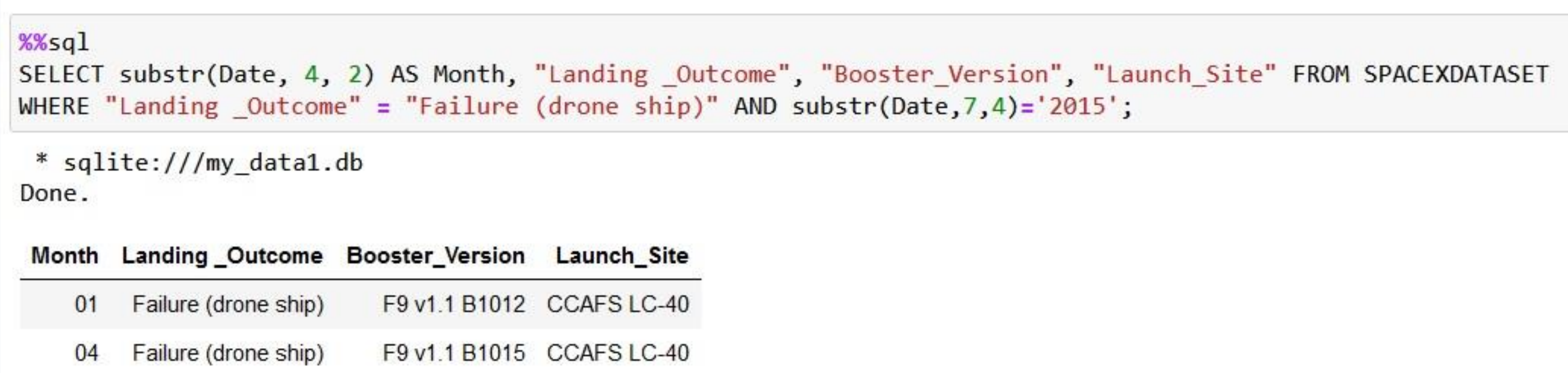
```
 * sqlite:///my_data1.db
Done.
```

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

- Used subquery in order to apply MAX function to find maximum payload mass.
- Payload mass located by WHERE clause

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.



```
%%sql
SELECT substr(Date, 4, 2) AS Month, "Landing _Outcome", "Booster_Version", "Launch_Site" FROM SPACEXDATASET
WHERE "Landing _Outcome" = "Failure (drone ship)" AND substr(Date,7,4)='2015';
```

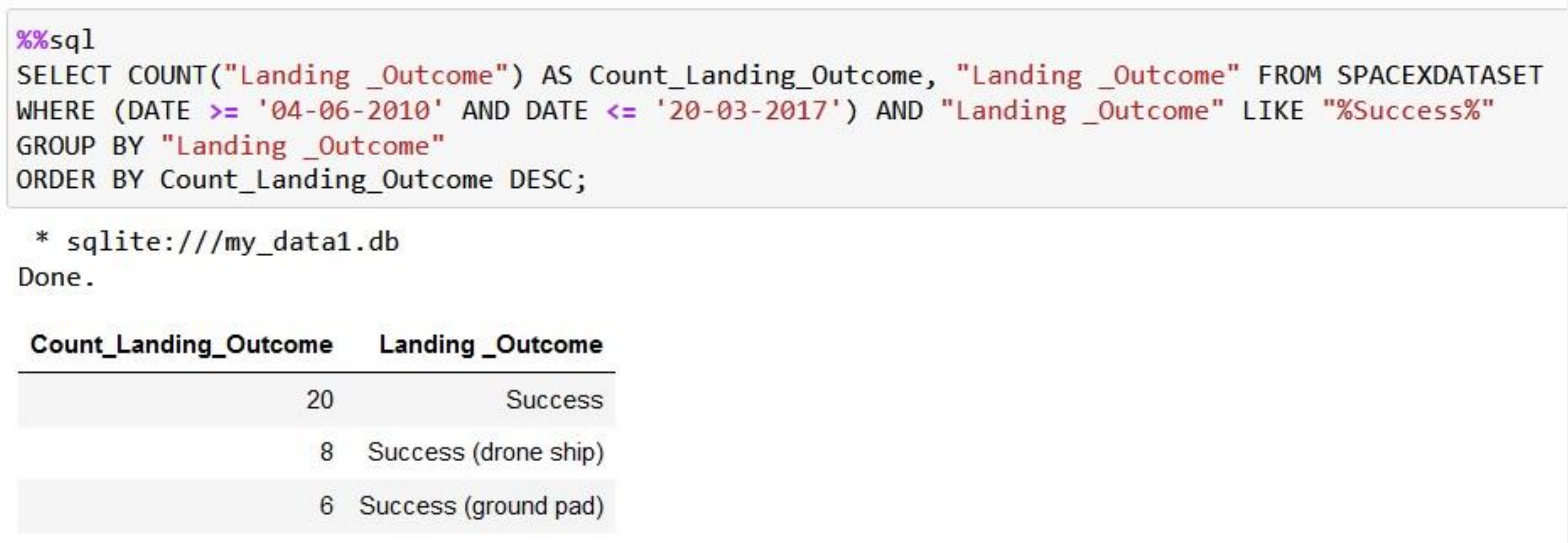
```
 * sqlite:///my_data1.db
Done.
```

Month	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Used SUBSTR function to extract months and year because SQLite does not have DAY and YEAR functions.

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

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



```
%%sql
SELECT COUNT("Landing _Outcome") AS Count_Landing_Outcome, "Landing _Outcome" FROM SPACEXDATASET
WHERE (DATE >= '04-06-2010' AND DATE <= '20-03-2017') AND "Landing _Outcome" LIKE "%Success%"
GROUP BY "Landing _Outcome"
ORDER BY Count_Landing_Outcome DESC;
```

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

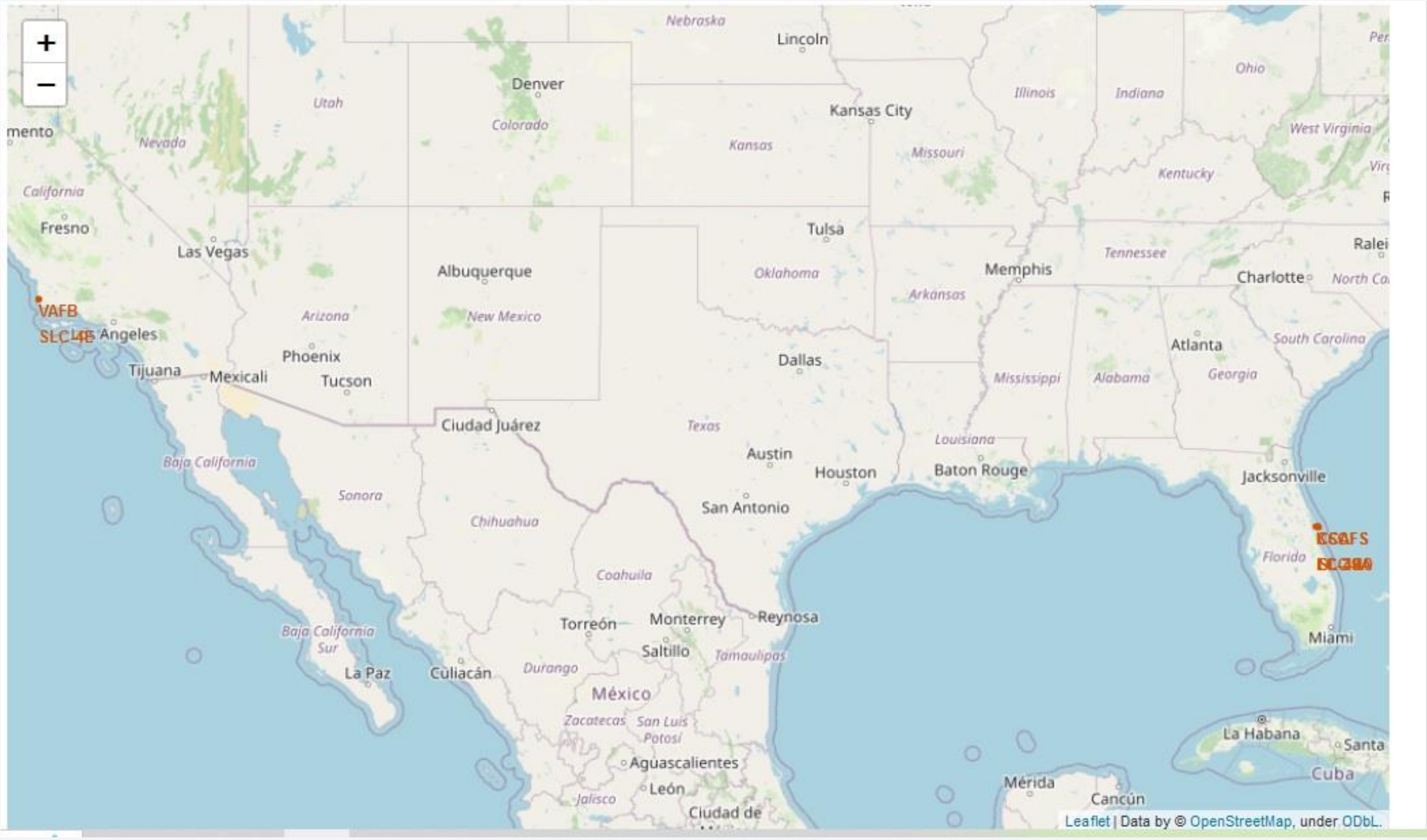
Count_Landing_Outcome	Landing _Outcome
20	Success
8	Success (drone ship)
6	Success (ground pad)

- COUNT used to count number of grouped outcomes (grouped with GROUP BY). Results ordered with ORDER BY, from larger to smaller count with DESC.

Section 3

Launch Sites Proximities Analysis

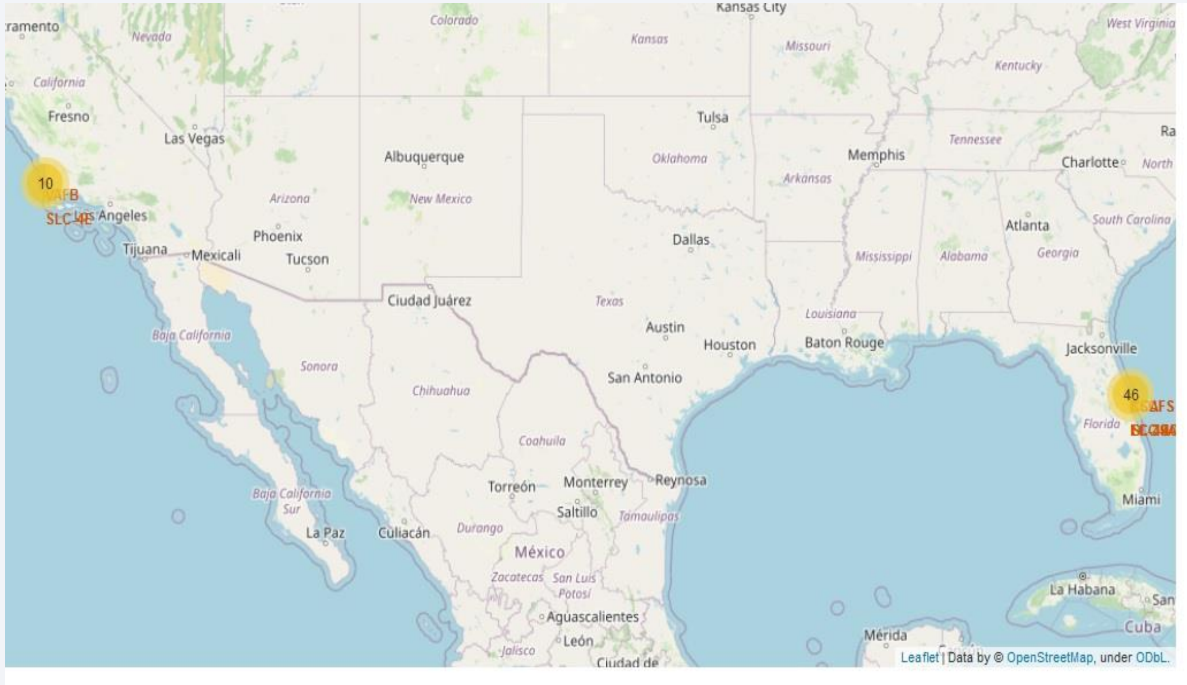
Launch Site locations

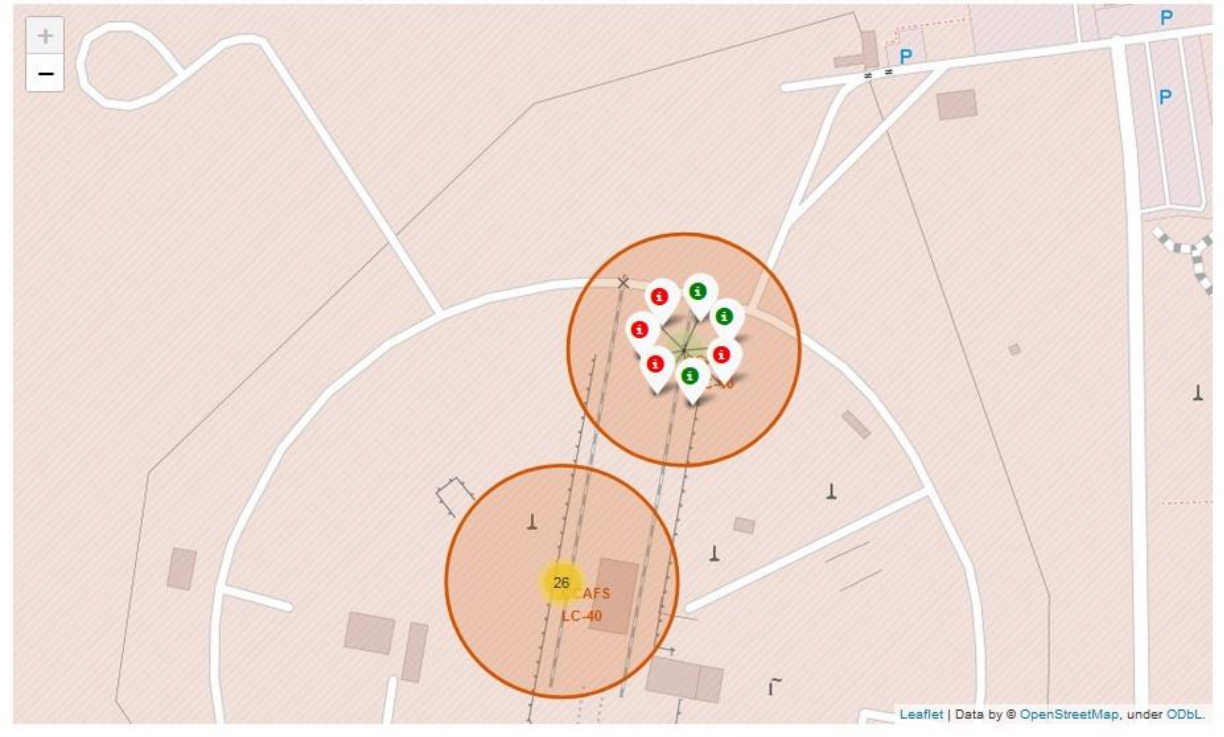


- Launch sites are located close to the sea in order to perform sea-based launches, which increase payload capacity and reduce cost.

- Locations are close to the equator, because the earth speed is larger there. Therefore rockets have larger speed once they are launched.

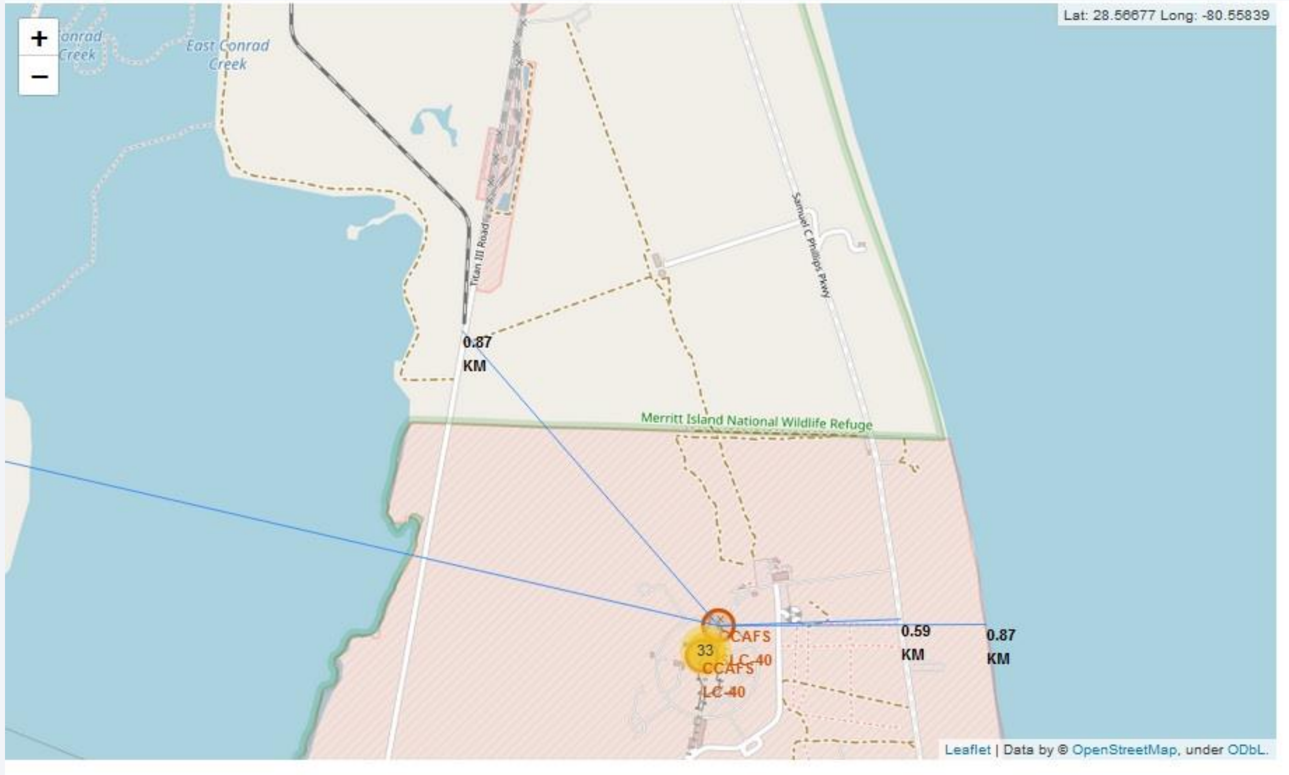
Marking launches' outcomes on map



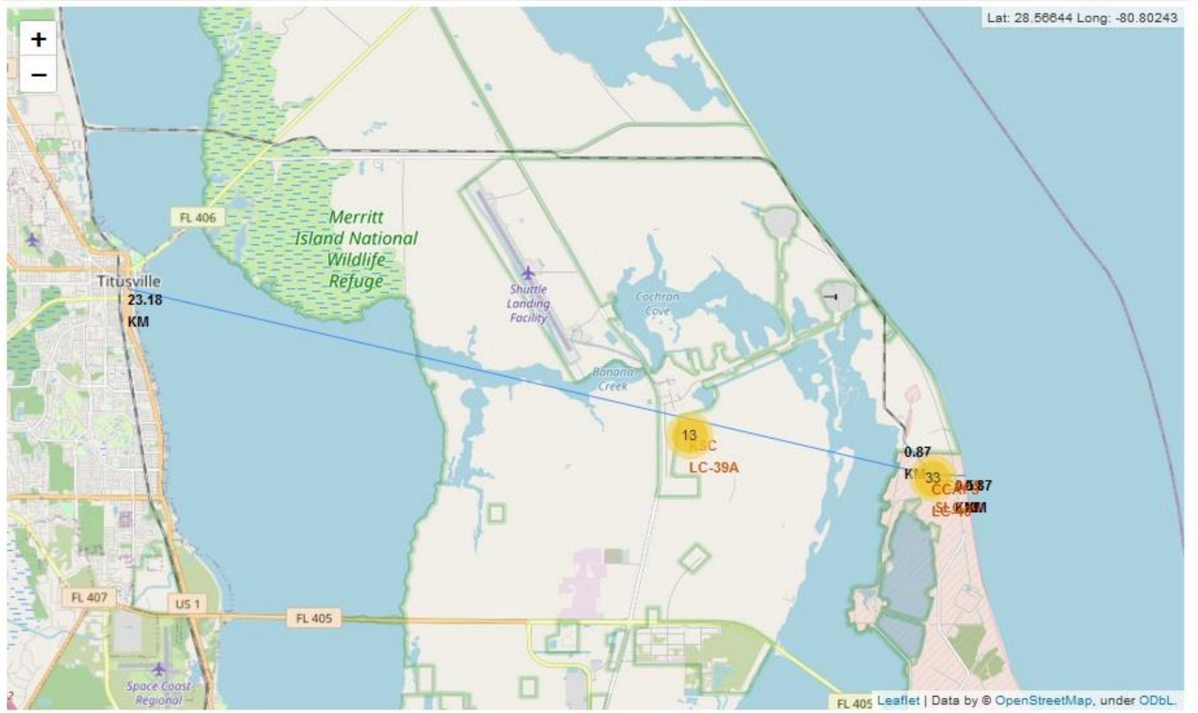


- Marking launching outcomes for each site helps identifying the launch sites with higher sucess rate.

Launch Site proximity to infrastructure



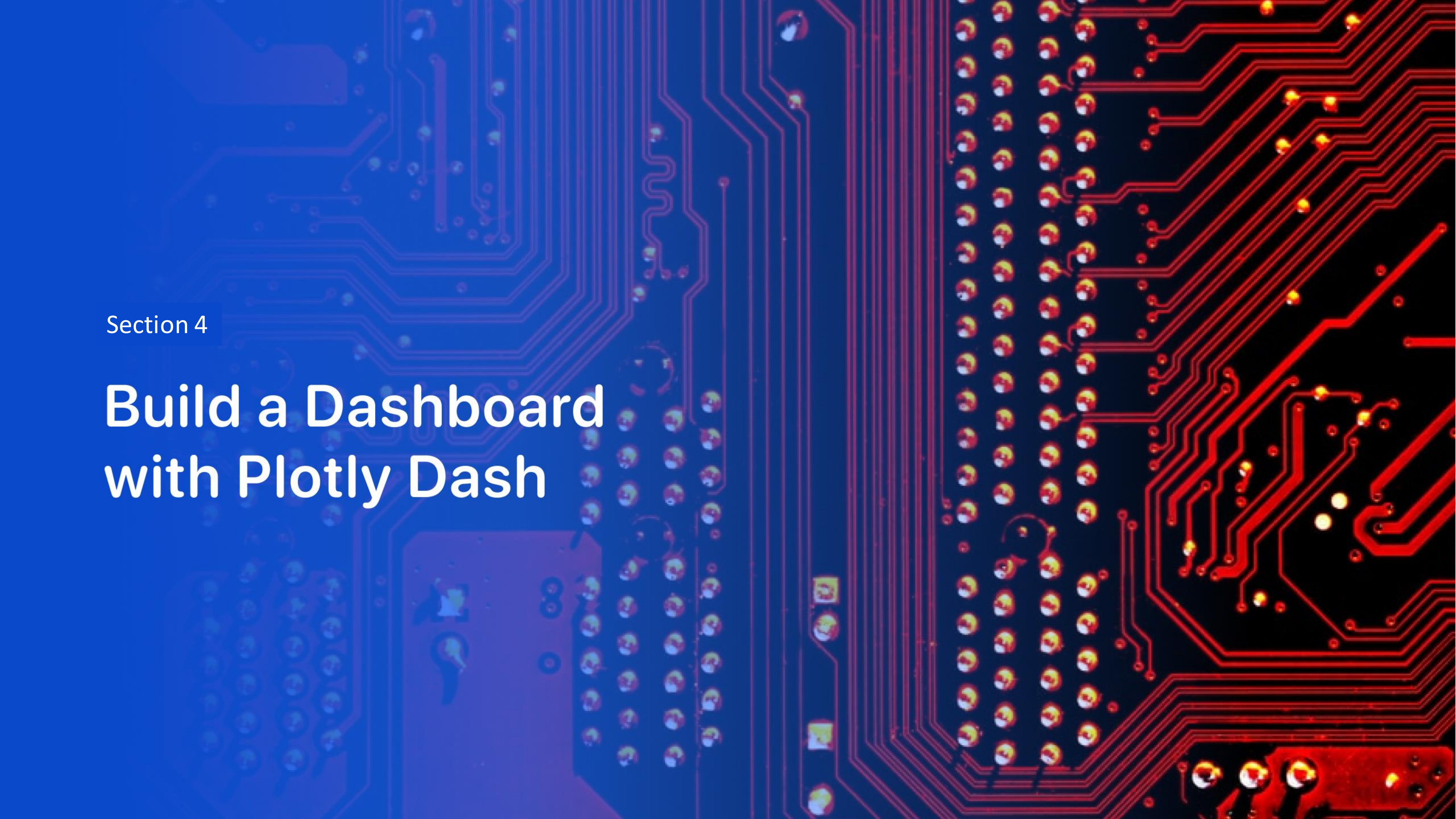
Lat: 28.56677 Long: -80.55839
+
−
East Conrad Creek
Titan III Road
Samuel C Phillips Pkwy
0.87 KM
Merritt Island National Wildlife Refuge
CCAFS LC-40
33
0.59 KM
0.87 KM
Leaflet | Data by © OpenStreetMap, under ODbL.

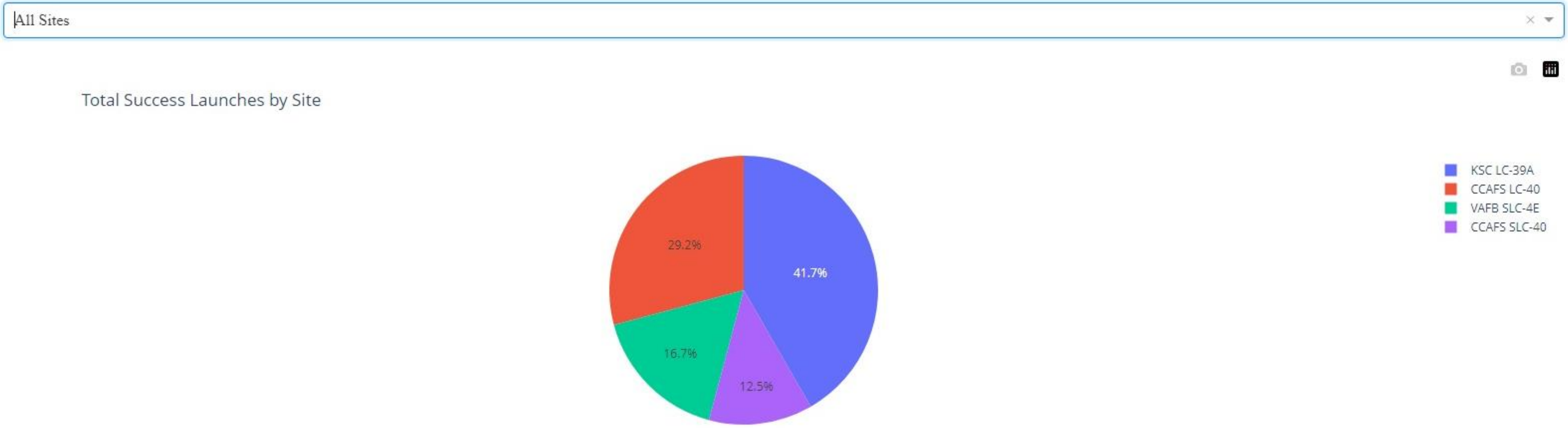


- Launch sites are situated close to infrastructure such as roads, railways and cities, which are crucial for providing easy and fast transport of equipment, supplies procurement etc.

Section 4

Build a Dashboard with Plotly Dash



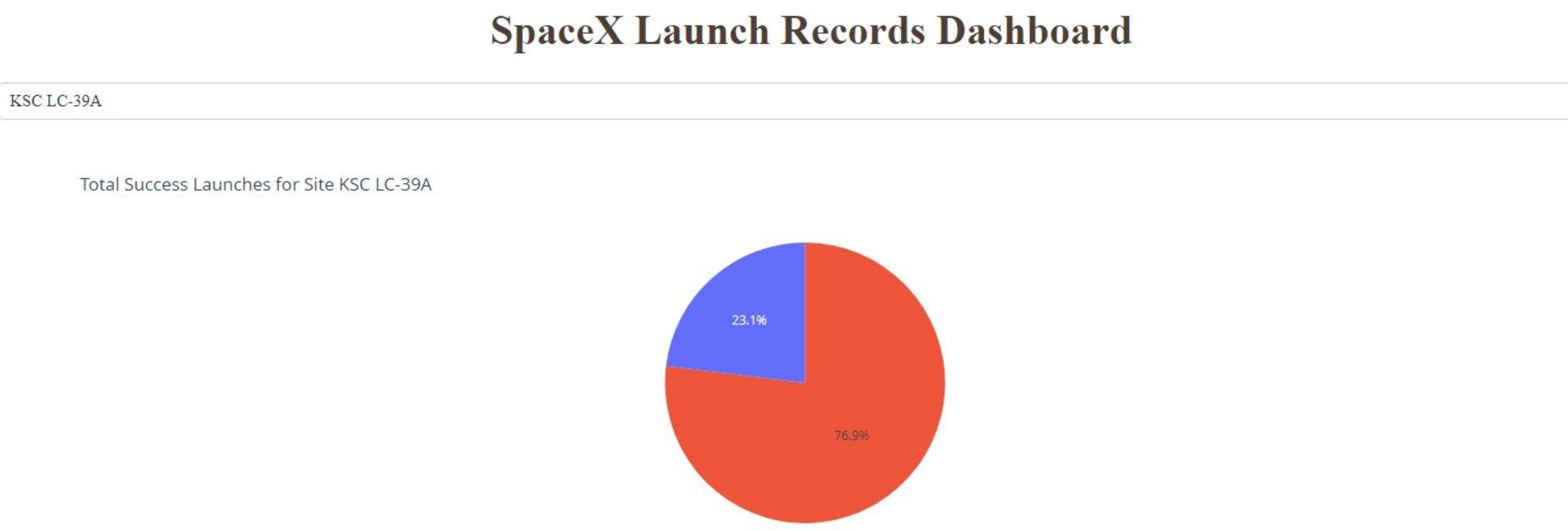


All Sites

Total Success Launches by Site

Success rate for Launch Sites

- Site KSC LC-369A has the highest launch success rate equal to 41.7%.
- The lowest success rate is observed for site CCAFS SLC-40 at 12.5%.



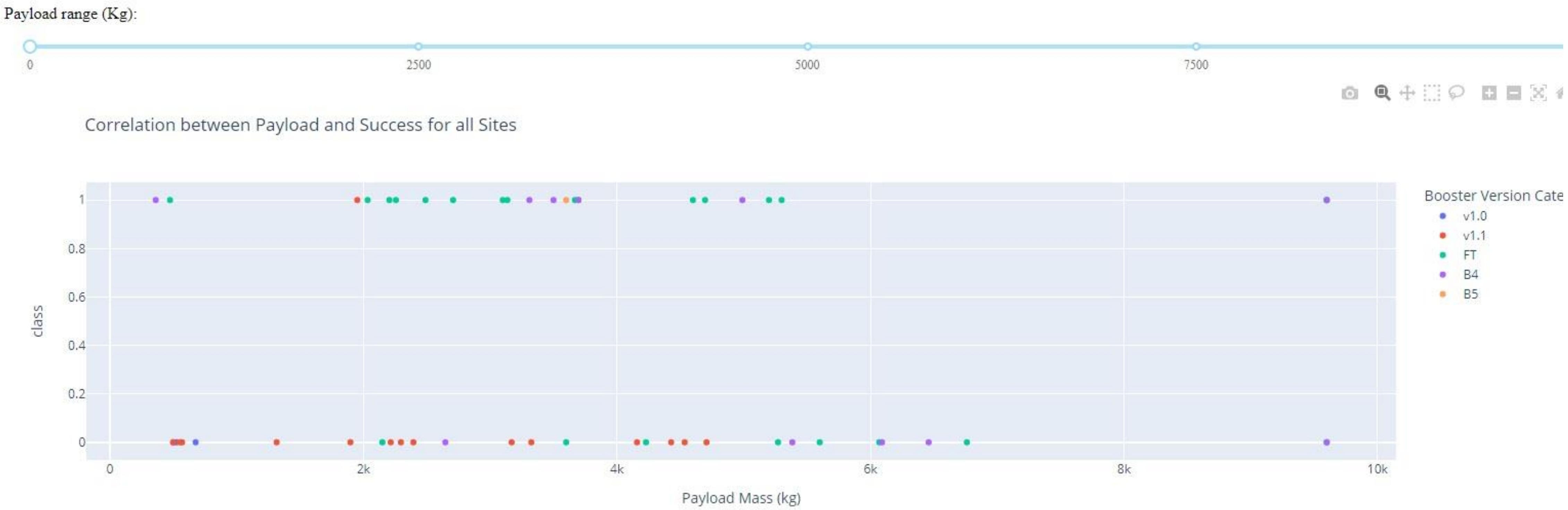
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success Launches for Site KSC LC-39A

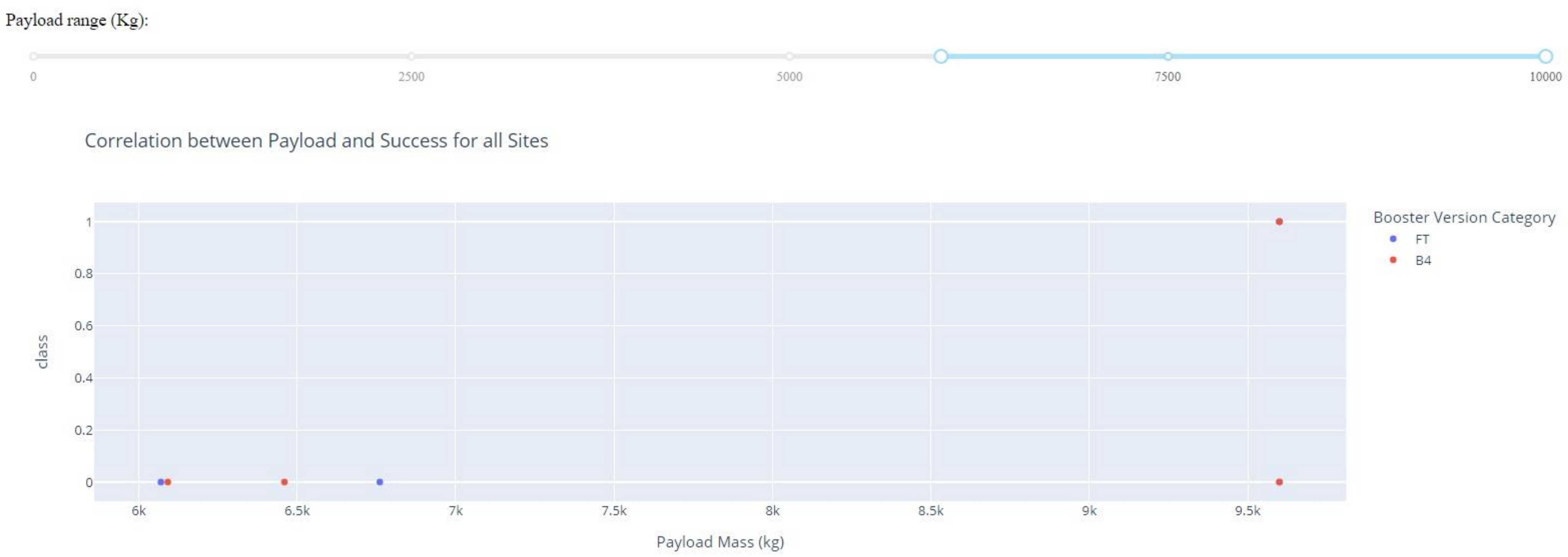
Site with the highest launch success ratio

- At site KSC LC-39A the successful launches account for 76.9% of the total.
- Failed launches are 23.1% of total.


Payload range (Kg):

Payload vs Launch Outcome for all Sites

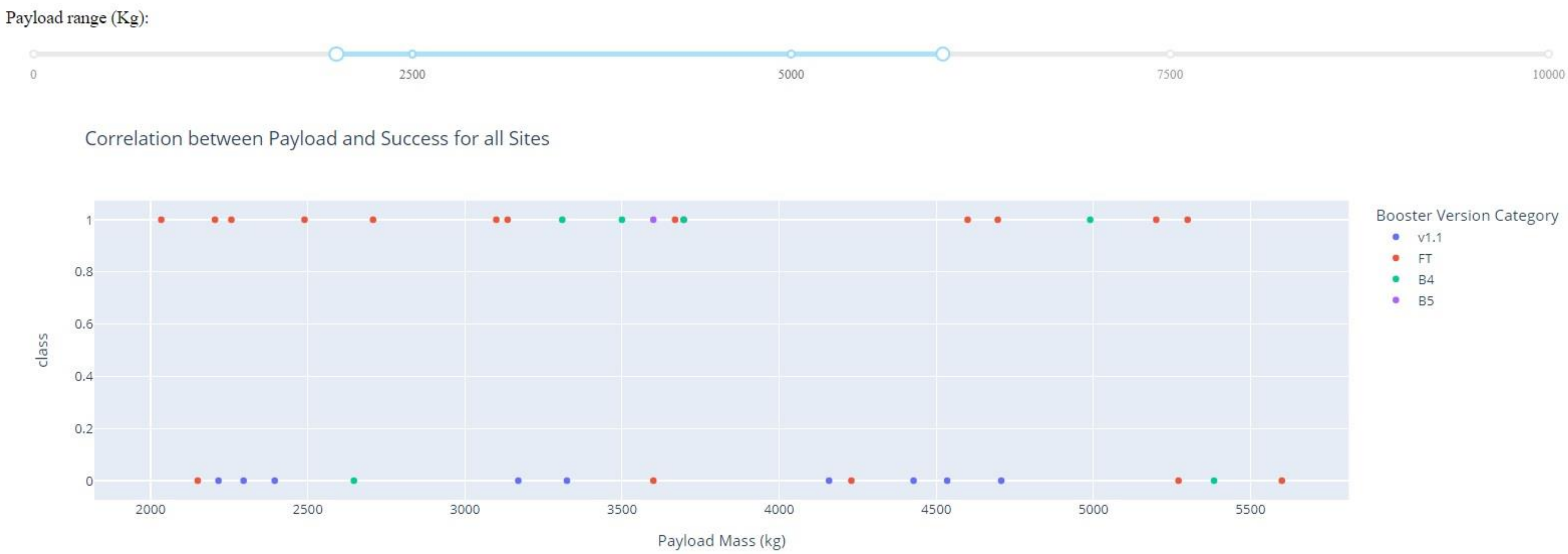
- Highest success rate observed for a payload range of 2000 to 4000kg.
- Most successful launches have been achieved with FT booster version.
- Most unsuccessful launches have occurred with v1.1 booster version.



Payload range (Kg):

Payload vs Launch Outcome - Range 6k-10k kg

- The lowest success rate is observed for payload mass larger than 6000kg.



Payload range (Kg):

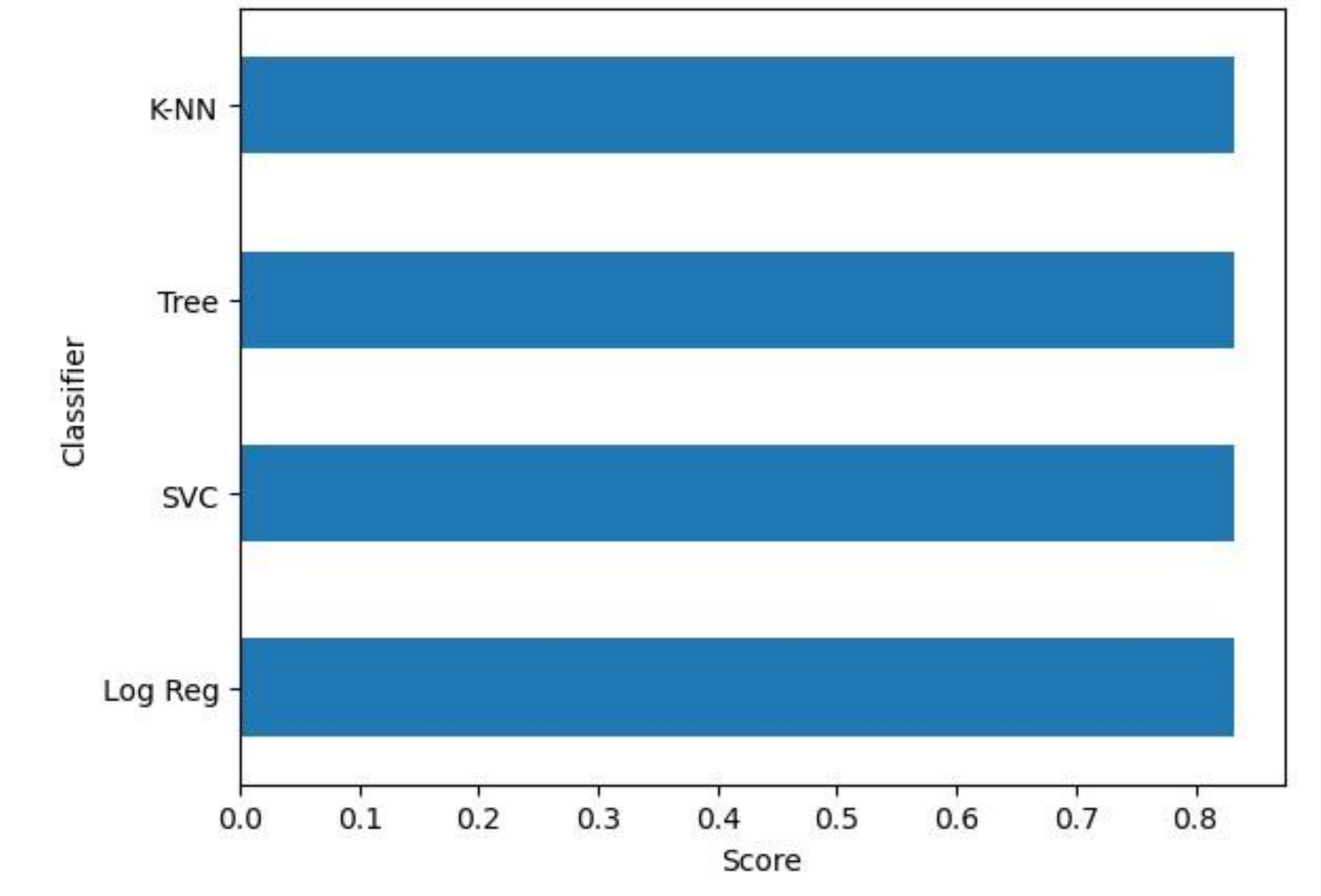
Payload vs Launch Outcome - Range 2k-6k kg

- The majority of launches that take place have a payload in the range 2000kg - 6000kg.

Section 5

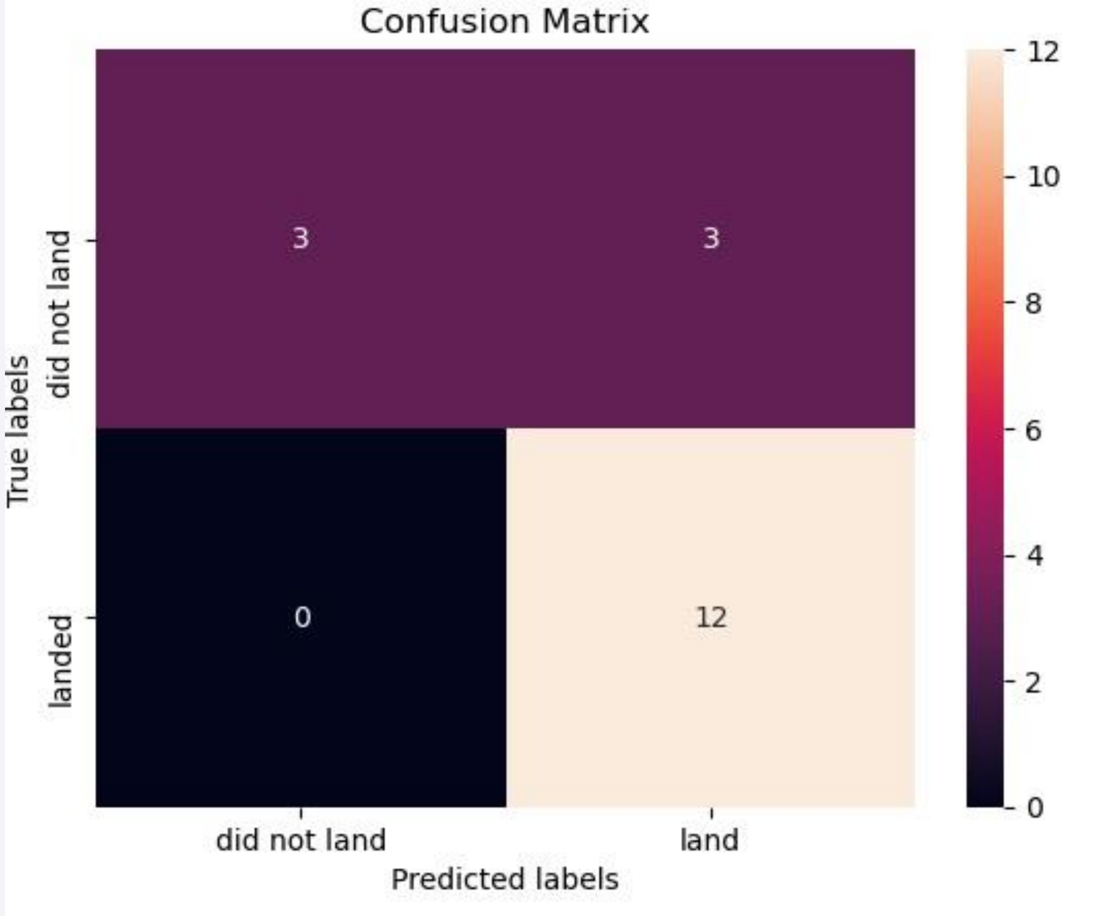
Predictive Analysis (Classification)

Classification accuracy



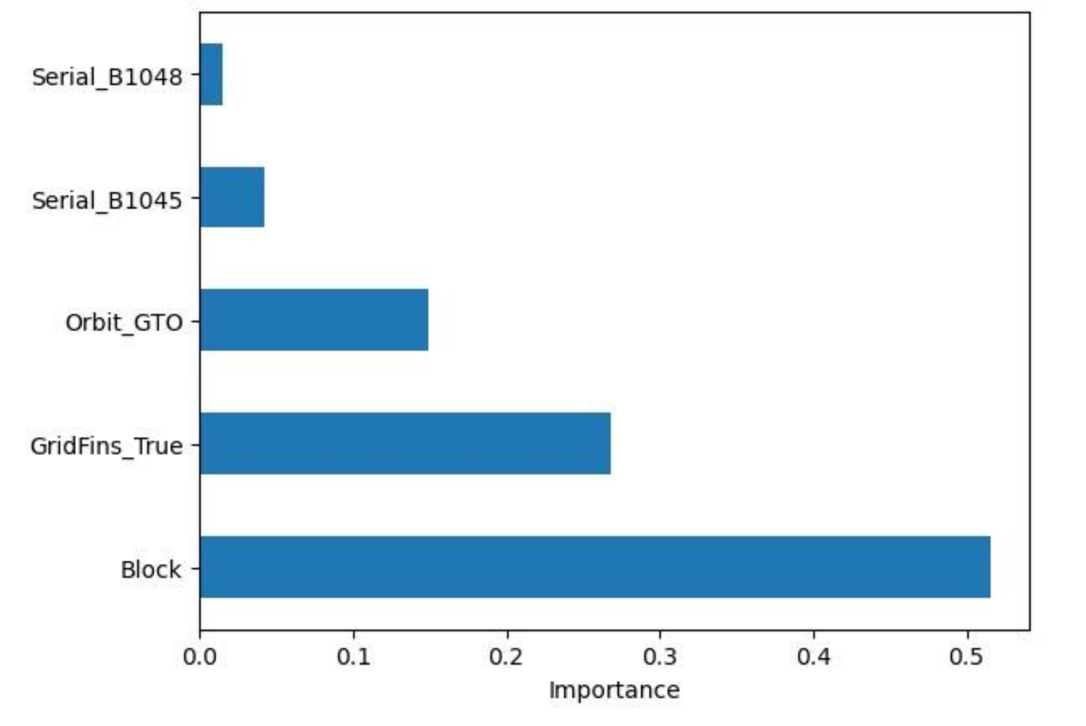
- All classifiers have the same accuracy.
- Probably attributed to the small size of the dataset.
- Accuracy score of 83%.

Confusion Matrix



- Since all models have the same accuracy, the confusion matrix is also identical.
- 3 unsuccessful and 12 successful landings were correctly predicted.
- 3 unsuccessful landings were wrongly predicted as successful, showing that the models might be problematic regarding false positives.

Feature importance for Tree Classifier



- Feature importance shows how important each feature is for model prediction.
  5 most influential features are shown.
- Acquired by fitting the best tree classifier and by calling the feature_importances_ attribute.
- Block and GridFins_True seem to be most important for prediction.
- Features with nearly zero importance may be discarded in order to reduce computational cost.
- The dataset is too small to draw safe conclusions.

Conclusions

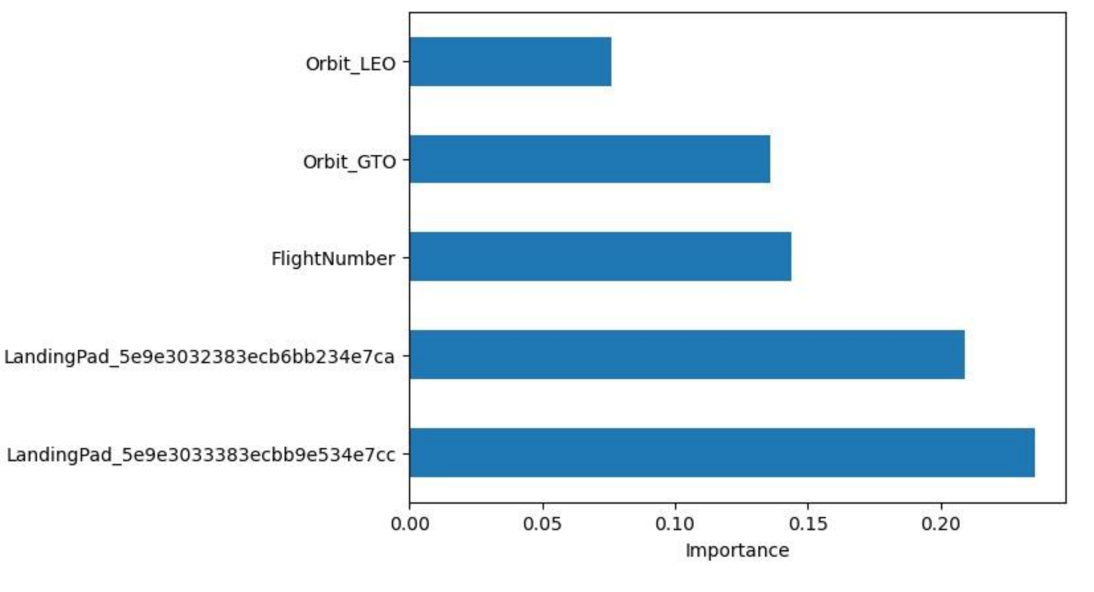
- Classifiers were found to predict the launch outcome fairly well, with an accuracy score of 83%.
- All models used performed almost identically, due to the small size of the dataset.
- The models need to be improved regarding false positive predictions, i.e. successful landing.
- Factors that may influence the launch outcome, such as flight number, payload, launch, site, orbit type, time and proximity to infrastructure were recognized.
- Highest success rates are achieved:
  - at the KSC LC-369A site,
  - with the FT booster version,
  - with a payload range from 2000kg to 4000kg.
- Feature importance from tree classifier showed that Block and GridFins_True were the most important for model prediction.

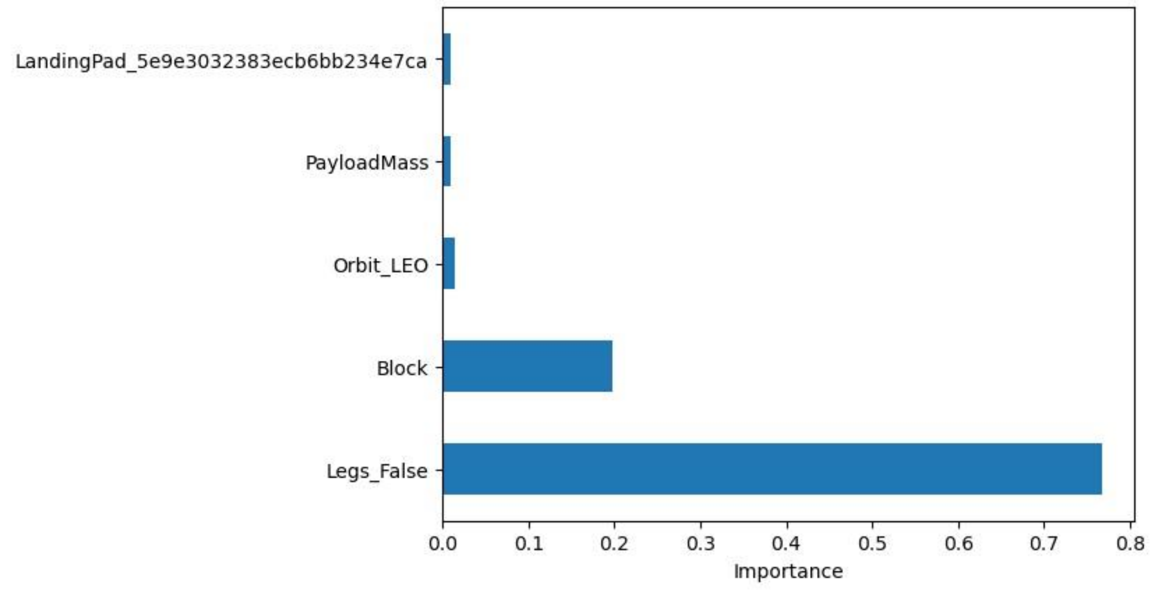
Limitations and recommendations

- Limitations
  - The dataset used for predictions was too small leading to identical performance for all models and to varying feature importance results every time the tree classifier was trained (see Appendix).
  - A limited number of classifiers was evaluated against the test data.
- Recommendations
  - The use of a larger dataset is required in order to draw safer conclusions.
  - More predicting algorithms need to be trained and evaluated, such as Random Forests, Gradient Boosting classifiers and neural networks.
  - Importance of features needs to be investigated in order to recognize and remove these who do not affect the predictions significantly. This can lead to reducing computational cost.

Appendix

Different feature importance diagrams after two consecutive runs with the best tree classifier.





Thank you!