IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alexandre Nassabay

2023-11-14

Outline

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

Executive Summary

- Methodologies
  - Data Collection
  - Data Wrangling
  - Data Analysis
  - Data Visualization
- Results

Introduction

Private space exploration market has reached USD 447 billion on 2023 and is expected to reach USD 1 trillion in 2030 (McKinsey and WEF). Several companies can be highlighted, such as Virgin Galactic, providing suborbital spaceflights, Rocket Lab as a small satellite provider, Blue Origin with sub-orbital and orbital reusable rockets, and SpaceX with reusable rockets, satellite deploying and Satellite Internet Provider.

Each SpaceX launch is announced with a cost of 62 million dollars, while other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Thus, the reusability of the first stage is crucial to company performance.

Studying the SapceX Launch operations we will seek to answer the following questions as benchmark to establish competition with them:

- How much each launch will cost?
- The first stage will be reused?
- The first stage will land successfully?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API
  - Web Scraping Wikipedia
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

SpaceX have an API, open to public use, with information about their launches;

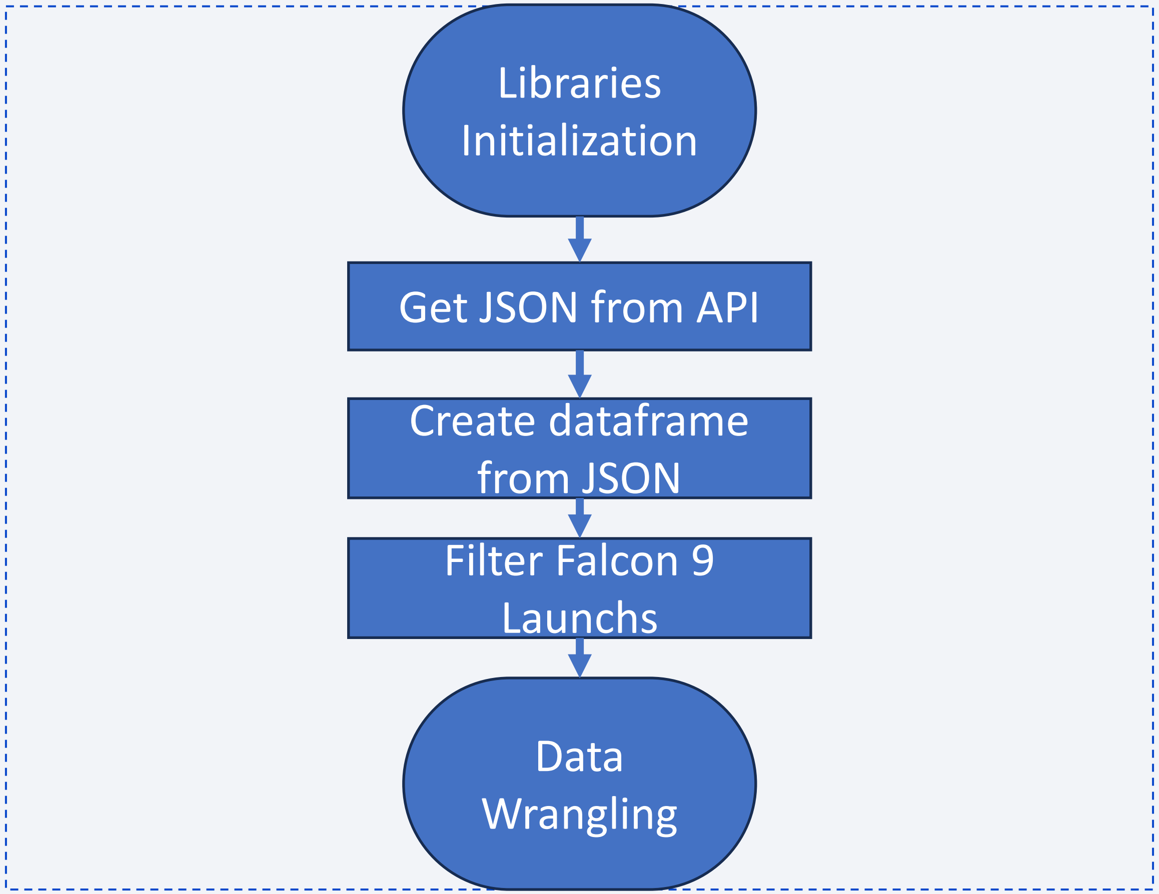
This is not enough data for this analysis, so it was complemented with Web Scraping from Wikipedia's article about the Falcon 9 spaceship.

Data Collection – SpaceX API

The SpaceX REST API is:

https://api.spacexdata.com/v4/

The script used to retrieve data from SpaceX API is on GitHub.

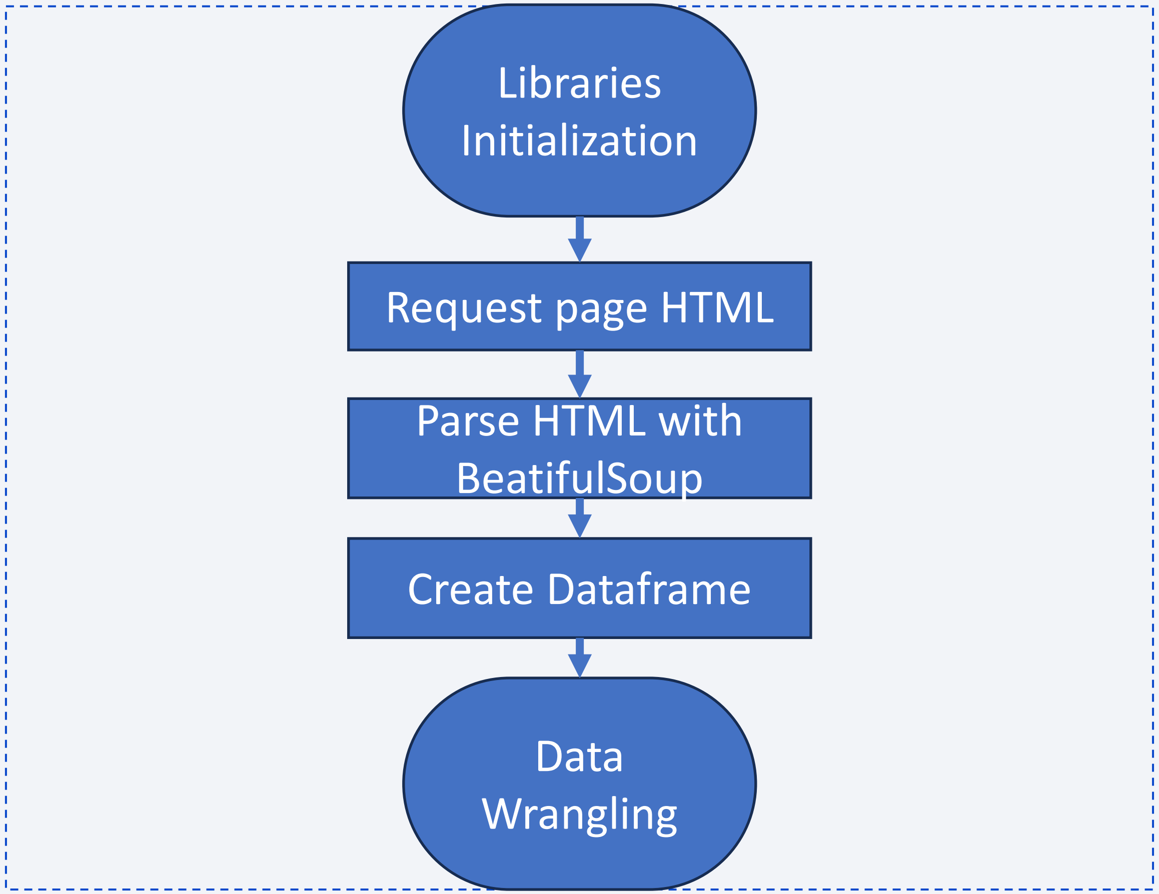


Data Collection - Scraping

The Wikipedia's article scraped is:

https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

The script used to scrape data from Wikipedia is on GitHub.



Data Wrangling

The acquired data is not ready for analysis due missing values.

First, we identify the fields thar have incomplete data.

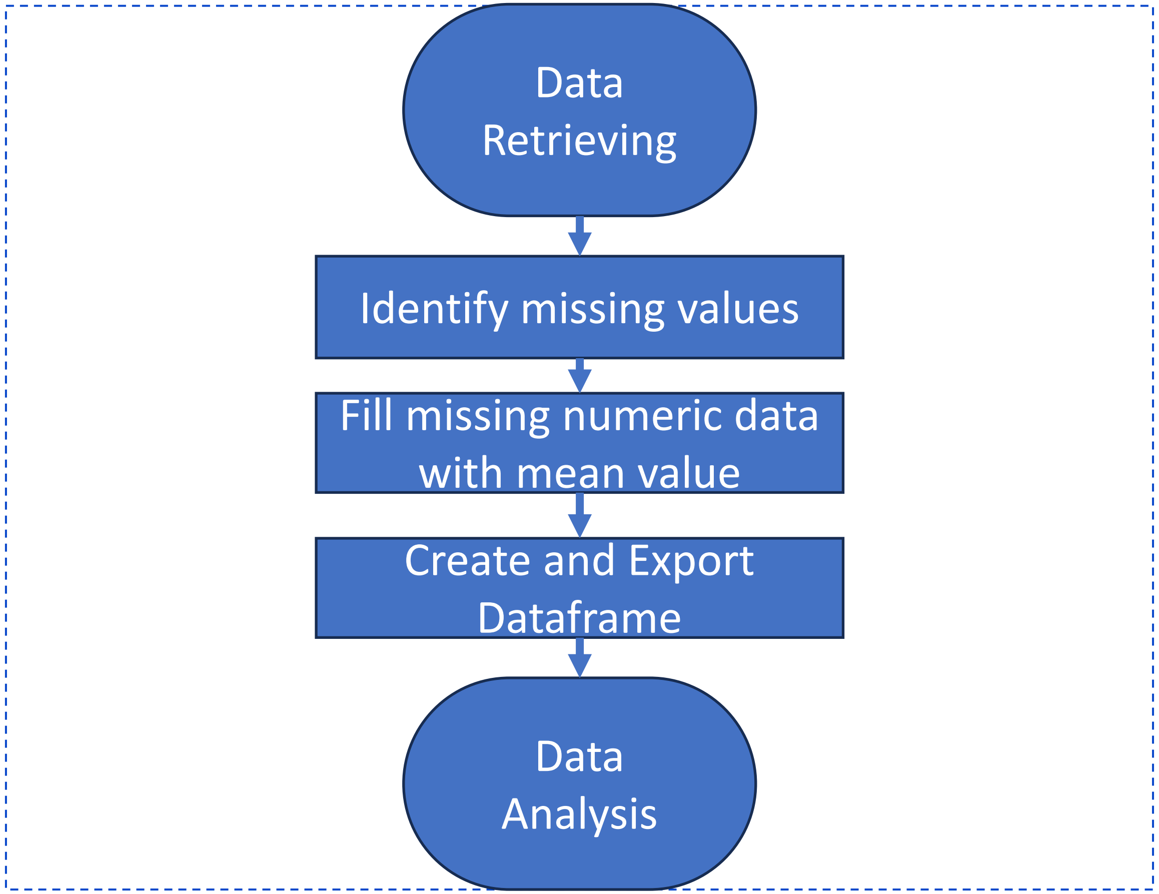
Then we identify if the incomplete fields are relevant for our analysis. If so, it will be treated, else it can be ignored or discarded.

On this dataset Payload Mass and Launchpad have missing data. The Launchpad is not critical, given that the launch sites and their coordinates are known. However, Payload Mass is directed related to the amount of fuel required to mission and the possible reuse of the first stage. The missing values will have to be estimated.

Data Wrangling

The chosen estimate for missing values is the mean value of Payload mass of known launches.

The script used for data wrangling is on GitHub.



Data Retrieving
Identify missing values
Fill missing numeric data with mean value
Create and Export Dataframe
Data Analysis

EDA with SQL

The following queries were performed:

- Create table (loading data from the CSV);
- 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 achieved;

EDA with SQL

- List the names of the boosters which have success in drone ship and have payload mass greater than 4000kg but less than 6000kg;
- List the total number of successful and failure mission outcomes;
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery;
- 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;

The script used for exploratory data analysis with SQL is on GitHub.

EDA with Data Visualization

The following charts types were used:

- Scatterplots – They are a good tool to identify correlation between variables.
    - Payload Mass vs Flight Number;
    - Launch Site vs Flight Number;
    - Launch Site vs Payload Mass;
    - Mission Success (Class) vs Mission Orbit;
    - Mission Orbit vs Flight Number;
    - Mission Orbit vs Payload Mass;

EDA with Data Visualization

- Bar graphs – The are good to represent relationships envolving cathegorical variables.
  - Mission Success (Class) vs Mission Orbit;

- Line graphs – They are used to visualize trends on numerical data.
  - Mission Success (Class) vs Launch Date;

The script used for exploratory data analysis with data visualization is on GitHub.

Build an Interactive Map with Folium

Folium is a library for geospatial data analysis. Its main object is a map where your data will be projected with help of another objects:

- Marker: this object pinpoints places of interest. In this analysis each mark represents a launch; Markers on a green color indicates a successful mission, while markers on red represent a failure;
- Mark Cluster: this object groups markers that are associates with the same place. The launch markers were clustered to their respective launch sites;
- Circles: Multifunctional visual indicators. If scaled with map, can represent an action radius or area. On this case they point where the launch sites and were mission command center are in place of markers;
- Lines: Multifunctional visual indicators. If scaled with map, can represent distance or scale. On this case they were use as a distance indicator between two anchor markers;
- Explain why you added those objects

The script used for representing geospatial data with Folium is on GitHub.

Build a Dashboard with Plotly Dash

A dashboard summarizing success data for the launches were made, containing the following plots:

- Pie chart: Show the percentage of success and failure of the site selected on dropdown menu; If the option “All sites” is selected, the pie will show the contribution of each site to the total of successful missions;
- Scatter plot: Shows the correlation among Success Rate of missions (Class) vs the Payload Mass, color-coded to the booster used as first stage. The data can be filtered for payload mass range between 0kg and 10000kg, in intervals of 1000kg and by the launch site dropdown as well.

Build a Dashboard with Plotly Dash

This plots allow to explore the following questions:

- Which site has the largest fraction of successful launches? (KSC LC-39A)
- Which site has the highest launch success rate? (KSC LC-39A)
- Which payload range(s) has the highest launch success rate? (3000kg to 4000kg with 70% of success)
- Which payload range(s) has the lowest launch success rate? (6000kg to 7000kg with no successful mission so far)
- Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate? (B5 with 100%, but only one flight so it can be taken as outlier. FT has 65% success rate and 20 missions attempted in total)

The script used for representing geospatial data with Folium is on GitHub.

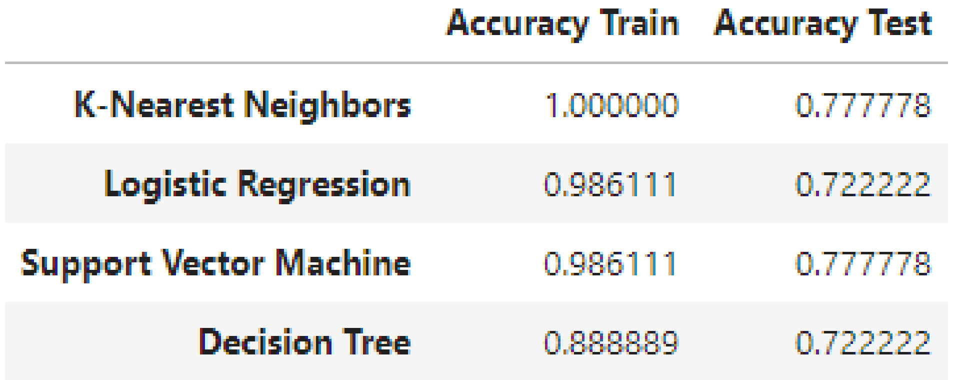
Predictive Analysis (Classification)

One of our goals is to predict if the first stage of rocket will land successfully. The following models will be used to tack this question:

- Logistic Regression;
- Support Vector Machine;
- Decision Tree;
- K-Nearest Neighbors;

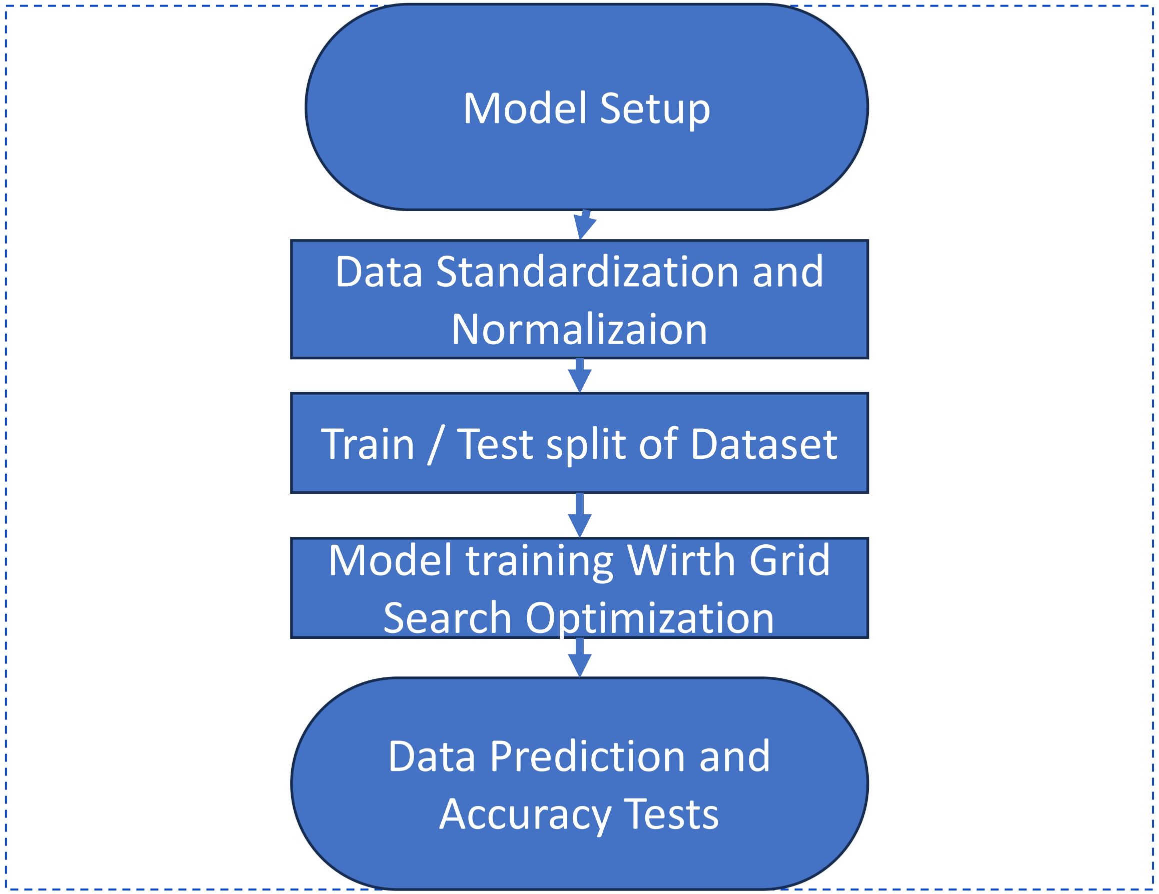
Predictive Analysis (Classification)

All the models were submitted to the workflow depicted on right.



	Accuracy Train	Accuracy Test
K-Nearest Neighbors	1.000000	0.777778
Logistic Regression	0.986111	0.722222
Support Vector Machine	0.986111	0.777778
Decision Tree	0.888889	0.722222

The script used for data classification is on GitHub.



Model Setup

Data Standardization and Normalizaion

Train / Test split of Dataset

Model training Wirth Grid Search Optimization

Data Prediction and Accuracy Tests

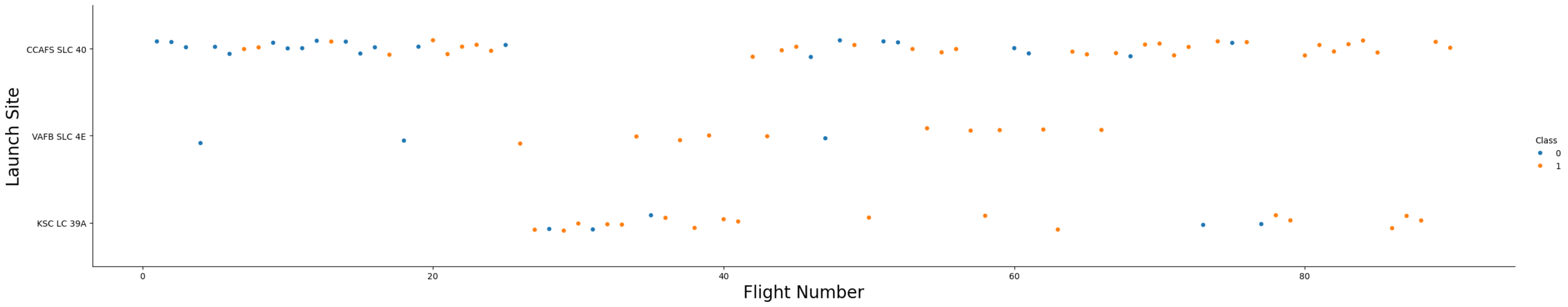
Results

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

Section 2

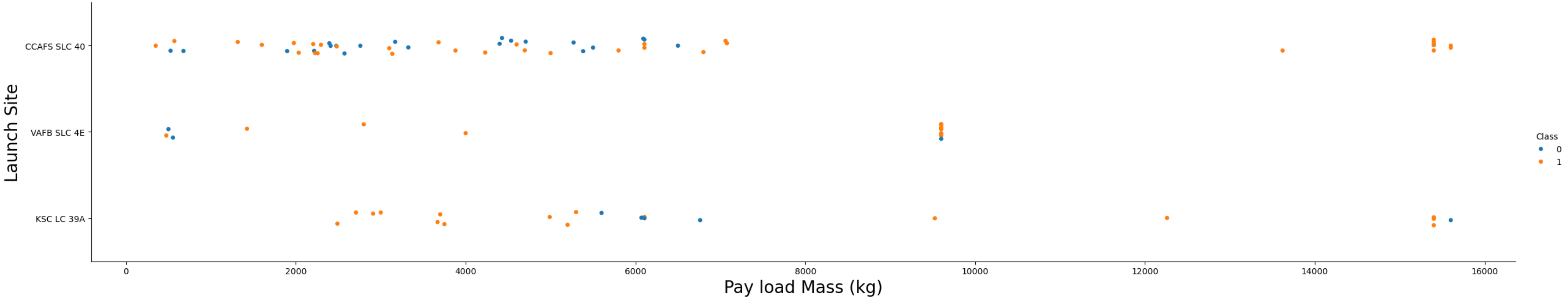
Insights drawn from EDA

Flight Number vs. Launch Site



- There is no special correlation between launch site and flight number;
- Success rate increased over time;

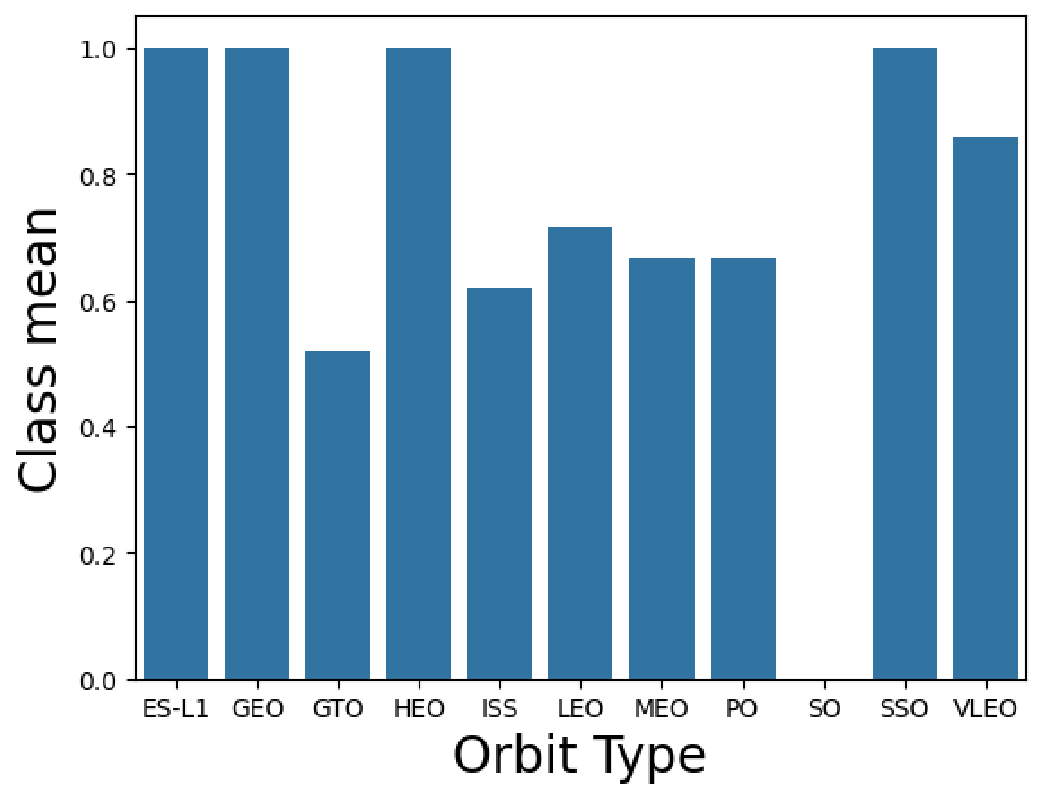
Payload vs. Launch Site



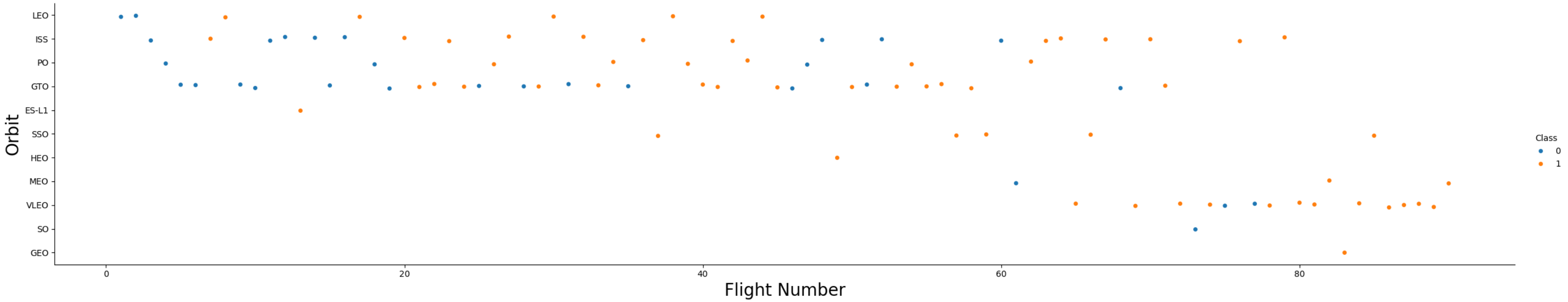
- Launch site WAFB SLC 4E seems to be limited to payloads below 10000kg;
- KSC LC 39A seems to handle better smaller payloads;

Success Rate vs. Orbit Type

- The orbits with higher altitude are usually associate with more expensive payloads and, consequently, tighter procedures, which may exclude booster reuse. This might be a fator in their better success rate.

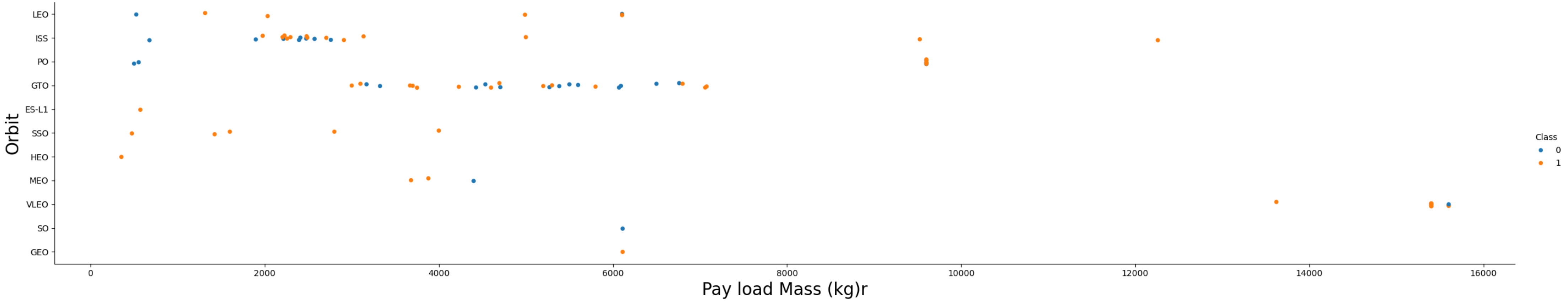


Flight Number vs. Orbit Type



- Overall success rate increase for all mission orbits;
- Similar conditions may allow transfer knowledge from one type of mission to another;

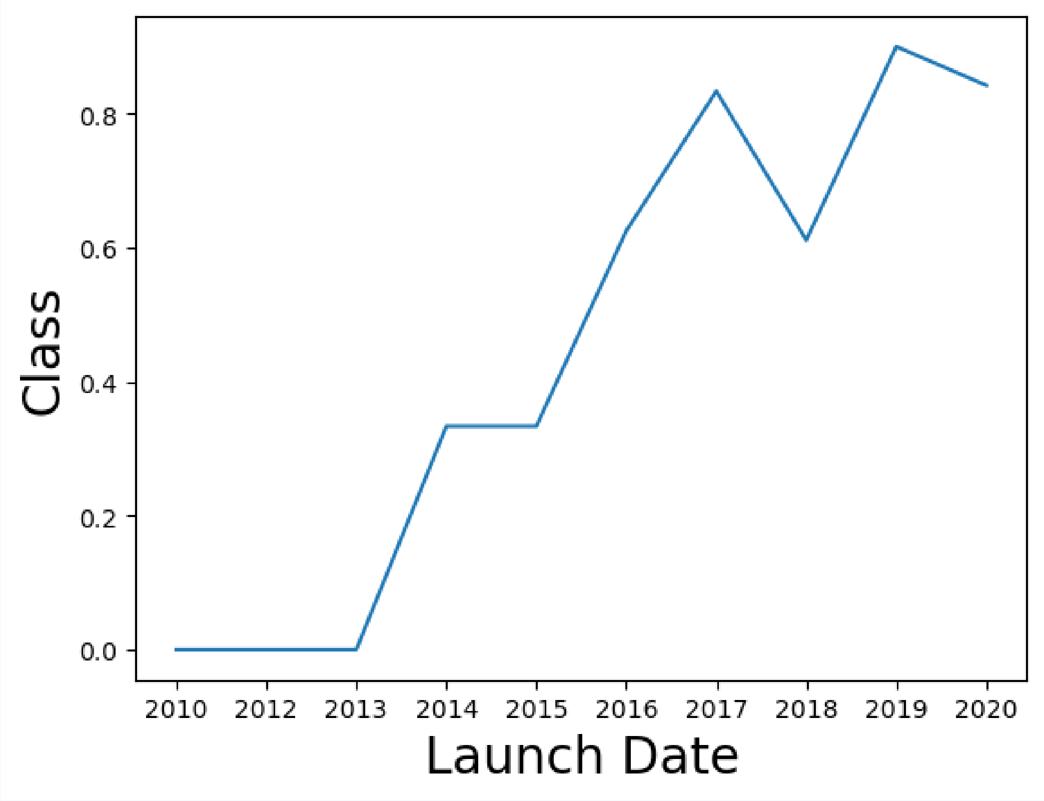
Payload vs. Orbit Type



- Heavier payloads have in general higher success rate, in special LEO, ISS and PO;

Launch Success Yearly Trend

- We can see clearly the result of the knowledge accumulated with all launches as success rate increases with time;
- It's also visible the diminishing returns on success rate increase;



All Launch Site Names

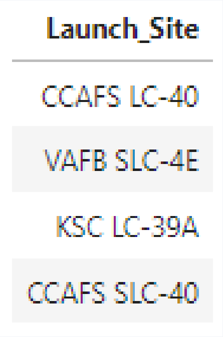
SQL Query:

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
```

Explanation:

The DISTINCT keyword forces the query to return results with no repetitions.

Query Result:



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

Launch Site Names Begin with 'CCA'

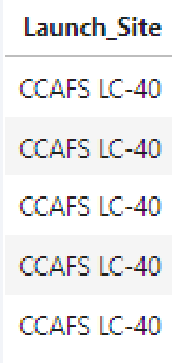
SQL Query:

%sql SELECT "Launch_Site" FROM SPACEXTABLE WHERE "Launch_Site" LIKE "CCA%" LIMIT 5

Explanation:

The LIKE keyword limits the query to return results with "CCA%" charcaters, with “%” acting as a wildcard.

Query Result:



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

Total Payload Mass

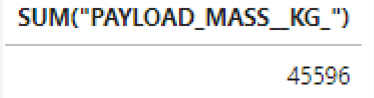
SQL Query:

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Customer" == "NASA (CRS)"
```

Explanation:

The SUM keyword totalizes the content of the PAYLOAD_MASS__KG_ column and the WHERE "Customer" == "NASA (CRS)“ clause limits the query to the desired Costumer.

Query Result:



SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

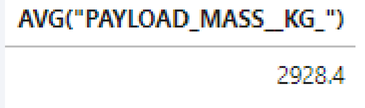
SQL Query:

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Booster_Version" == "F9 v1.1"
```

Explanation:

The AVG keyword averages the content of the PAYLOAD_MASS__KG_ column and the WHERE "Booster_Version" == "F9 v1.1" clause limits the query to the desired Booster.

Query Result:



AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

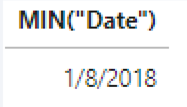
SQL Query:

```
%sql SELECT MIN("Date") FROM SPACEXTABLE WHERE "Landing_Outcome" == "Success (ground pad)"
```

Explanation:

The MIN keyword picks the smallest, i.e., the earliest value of Date column and the WHERE "Landing_Outcome" == "Success (ground pad)" clause limits the query to the successful missions on ground pad.

Query Result:



MIN("Date")
1/8/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

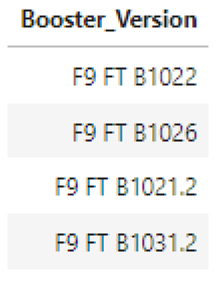
SQL Query:

```
%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" == "Success (drone
ship)" AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000
```

Explanation:

The DISTINCT keyword forces the query to return results with no repetitions. The WHERE "Landing_Outcome" == "Success (drone ship)" AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000 clause limits the query to Palyload mass range for the successful missions on drone ship.

Query Result:



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

Total Number of Successful and Failure Mission Outcomes

SQL Query:

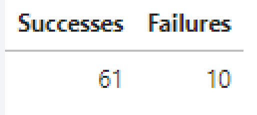
```
%%sql SELECT (SELECT COUNT("Landing_Outcome") FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE "Success%")
AS Successes,\

(SELECT COUNT("Landing_Outcome") FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE "Failure%") AS Failures
```

Explanation:

We have a subquery to count every landing success and another subquery to count every landing failure.

Query Result:



Successes	Failures
61	10

Boosters Carried Maximum Payload

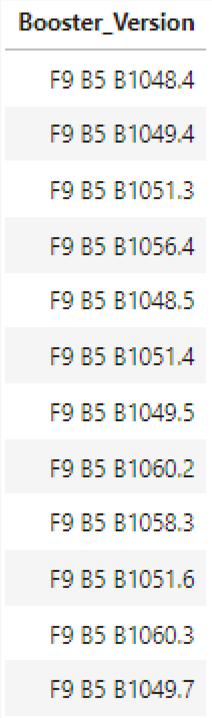
SQL Query:

%sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_“ = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)

Explanation:

We have a subquery to find the maximum payload carried, while main query picks each distinct booster that reached tis limit.

Query Result:



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

SQL Query:

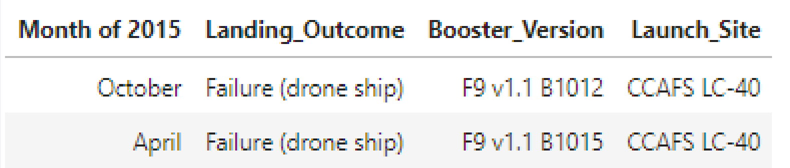
```
%sql SELECT \
CASE \
WHEN (substr("Date", -7, 2) = '01' OR substr("Date", -7, 2) = '/1') THEN 'January' \
WHEN (substr("Date", -7, 2) = '02' OR substr("Date", -7, 2) = '/2') THEN 'February' \
WHEN (substr("Date", -7, 2) = '03' OR substr("Date", -7, 2) = '/3') THEN 'March' \
WHEN (substr("Date", -7, 2) = '04' OR substr("Date", -7, 2) = '/4') THEN 'April' \
WHEN (substr("Date", -7, 2) = '05' OR substr("Date", -7, 2) = '/5') THEN 'May' \
WHEN (substr("Date", -7, 2) = '06' OR substr("Date", -7, 2) = '/6') THEN 'June' \
WHEN (substr("Date", -7, 2) = '07' OR substr("Date", -7, 2) = '/7') THEN 'July' \
WHEN (substr("Date", -7, 2) = '08' OR substr("Date", -7, 2) = '/8') THEN 'August' \
WHEN (substr("Date", -7, 2) = '09' OR substr("Date", -7, 2) = '/9') THEN 'September' \
WHEN (substr("Date", -7, 2) = '10') THEN 'October' \
WHEN (substr("Date", -7, 2) = '11') THEN 'November' \
WHEN (substr("Date", -7, 2) = '12') THEN 'December' \
ELSE substr("Date", -7, 2) \
END AS "Month of 2015", \
"Landing_Outcome", "Booster_Version" , "Launch_Site" \
FROM SPACEXTABLE \
WHERE substr("Date", -4, 4) = "2015" AND "Landing_Outcome" = "Failure (drone ship)"  )
```

2015 Launch Records

Explanation:

The CASE clause act together with substr() function to parse the data column and properly assign the month.

Query Result:



Month of 2015	Landing_Outcome	Booster_Version	Launch_Site
October	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

SQL Query:

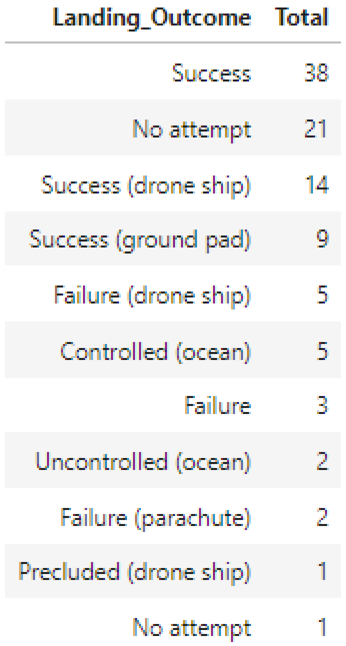
```
%sql SELECT "Landing_Outcome", COUNT("Landing_Outcome") AS "Total" \
FROM \
(SELECT *, \
CASE \
WHEN ("Year" < '2017' AND "Year" > '2010') THEN '1' \
WHEN ("Year" = '2017' AND "Month" < '03') THEN '1' \
WHEN ("Year" = '2017' AND "Month" > '03') THEN '0' \
WHEN ("Year" = '2010' AND "Month" > '04') THEN '1' \
WHEN ("Year" = '2010' AND "Month" < '04') THEN '0' \
WHEN ("Year" = '2017' AND "Month" = '03' AND "Day" <= '30') THEN '1' \
WHEN ("Year" = '2017' AND "Month" = '03' AND "Day" > '30') THEN '0' \
WHEN ("Year" = '2010' AND "Month" = '04' AND "Day" >= '06') THEN '1' \
WHEN ("Year" = '2010' AND "Month" = '04' AND "Day" <6) THEN '0' \
ELSE '0' \
END \
AS "Result" \
FROM \
(SELECT \
"Landing_Outcome", "Date", substr("Date", -4, 4) AS "Year", \
CASE \
WHEN (substr("Date", -7, 2) = '01' OR substr("Date", -7, 2) = '/1') THEN '01' \
WHEN (substr("Date", -7, 2) = '02' OR substr("Date", -7, 2) = '/2') THEN '01' \
WHEN (substr("Date", -7, 2) = '03' OR substr("Date", -7, 2) = '/3') THEN '03' \
WHEN (substr("Date", -7, 2) = '04' OR substr("Date", -7, 2) = '/4') THEN '04' \
WHEN (substr("Date", -7, 2) = '05' OR substr("Date", -7, 2) = '/5') THEN '05' \
WHEN (substr("Date", -7, 2) = '06' OR substr("Date", -7, 2) = '/6') THEN '06' \
WHEN (substr("Date", -7, 2) = '07' OR substr("Date", -7, 2) = '/7') THEN '07' \
WHEN (substr("Date", -7, 2) = '08' OR substr("Date", -7, 2) = '/8') THEN '08' \
WHEN (substr("Date", -7, 2) = '09' OR substr("Date", -7, 2) = '/9') THEN '09' \
ELSE substr("Date", -7, 2) \
END AS "Month", \
CASE \
WHEN (substr("Date", 0, 3) = '01' OR substr("Date", 0, 3) = '1/') THEN '01' \
WHEN (substr("Date", 0, 3) = '02' OR substr("Date", 0, 3) = '2/') THEN '02' \
WHEN (substr("Date", 0, 3) = '03' OR substr("Date", 0, 3) = '3/') THEN '03' \
WHEN (substr("Date", 0, 3) = '04' OR substr("Date", 0, 3) = '4/') THEN '04' \
WHEN (substr("Date", 0, 3) = '05' OR substr("Date", 0, 3) = '5/') THEN '05' \
WHEN (substr("Date", 0, 3) = '06' OR substr("Date", 0, 3) = '6/') THEN '06' \
WHEN (substr("Date", 0, 3) = '07' OR substr("Date", 0, 3) = '7/') THEN '07' \
WHEN (substr("Date", 0, 3) = '08' OR substr("Date", 0, 3) = '8/') THEN '08' \
WHEN (substr("Date", 0, 3) = '09' OR substr("Date", 0, 3) = '9/') THEN '09' \
ELSE substr("Date", 0, 3) \
END AS "Day" \
FROM SPACEXTABLE)) \
GROUP BY "Landing_Outcome" ORDER BY "Total" DESC
```

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

Explanation:

Three subqueries were made. The first limits the year, the second limits the month and the third limits the day. This solution was used because direct date comparison was not working.

Query Result:



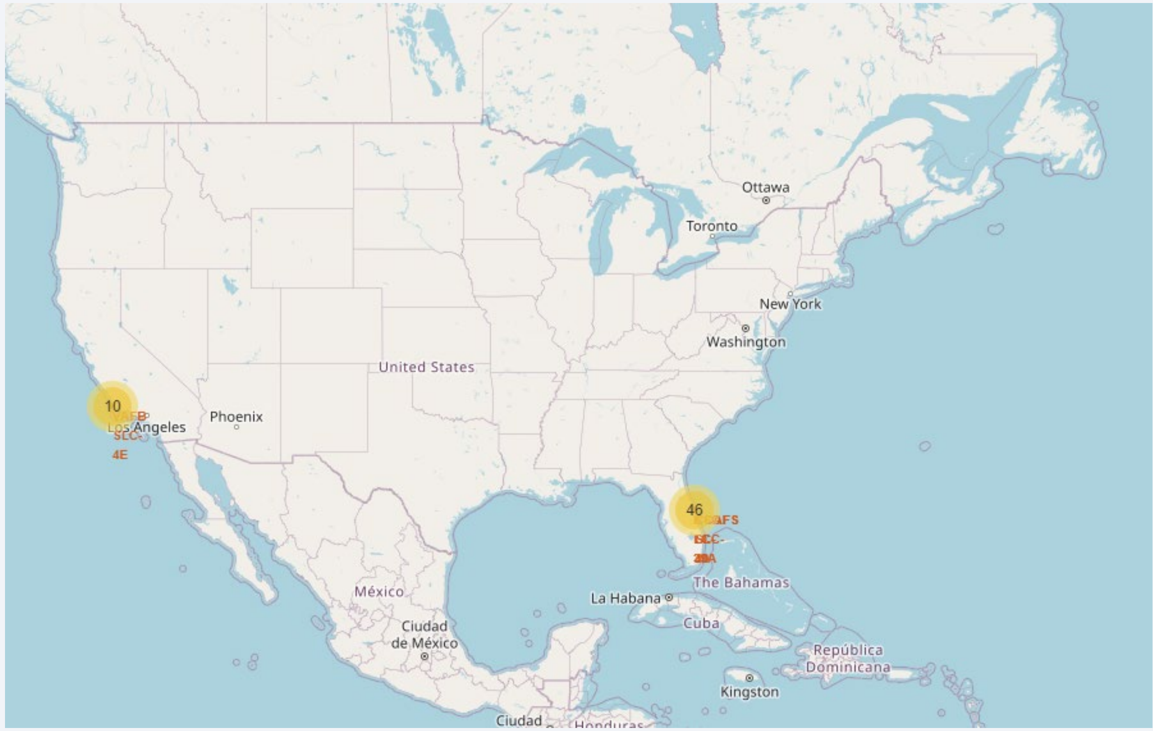
Landing_Outcome	Total
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1

Section 3

Launch Sites Proximities Analysis

Launch Stations

This maps shows that SpaceX uses launch sites in each US coast.



This location reduces risk of debris hitting inhabited areas in case of failures.

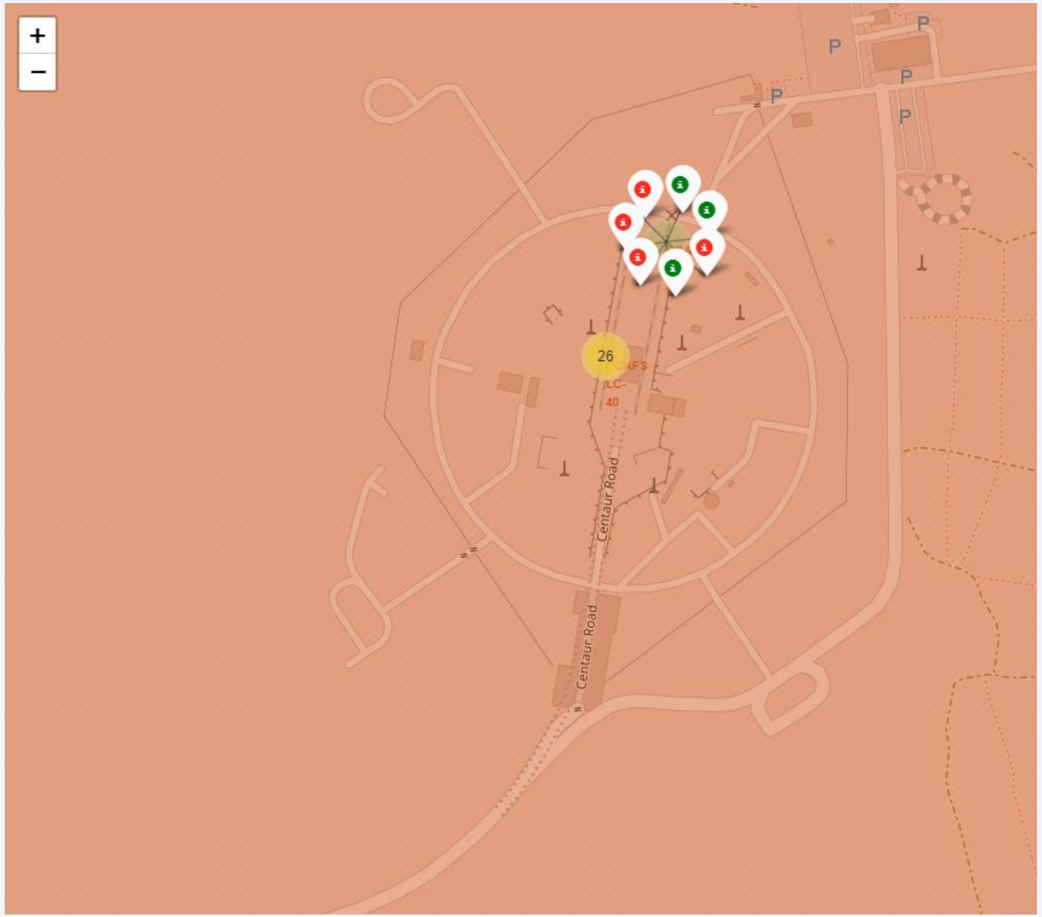
The need of this two locations arise from some missions requiring a retrograde orbit, so they must be launch in opposite direction from prograde orbits. To do this safely you need locations facing ocean from opposite angles.

Launch Count Mark Clusters

The map on left show CCAFS LC-40 site and its launch pads, each one have a marker cluster.

When not clicked, the cluster show the number of launches made from there.

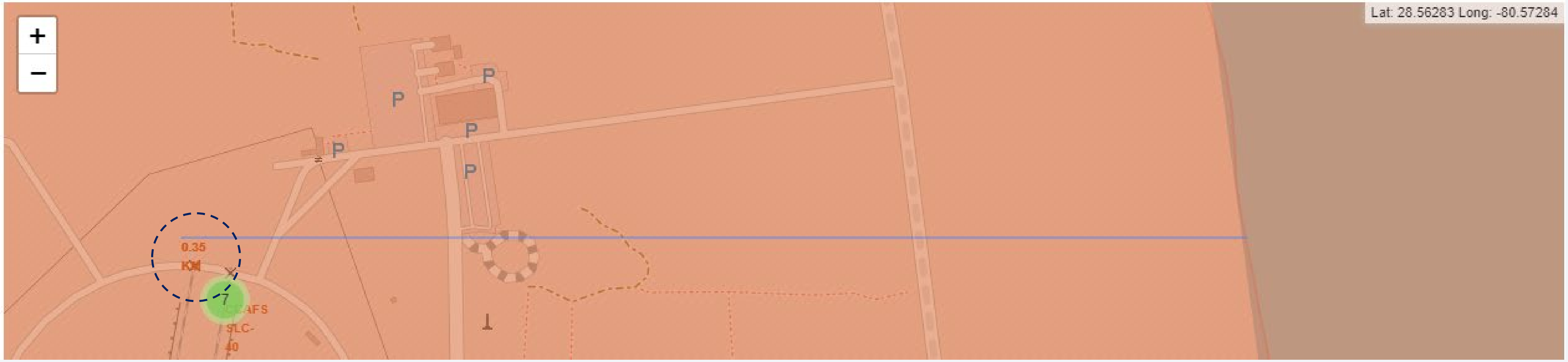
When clicked, the cluster explodes the markers indicating success and failures.

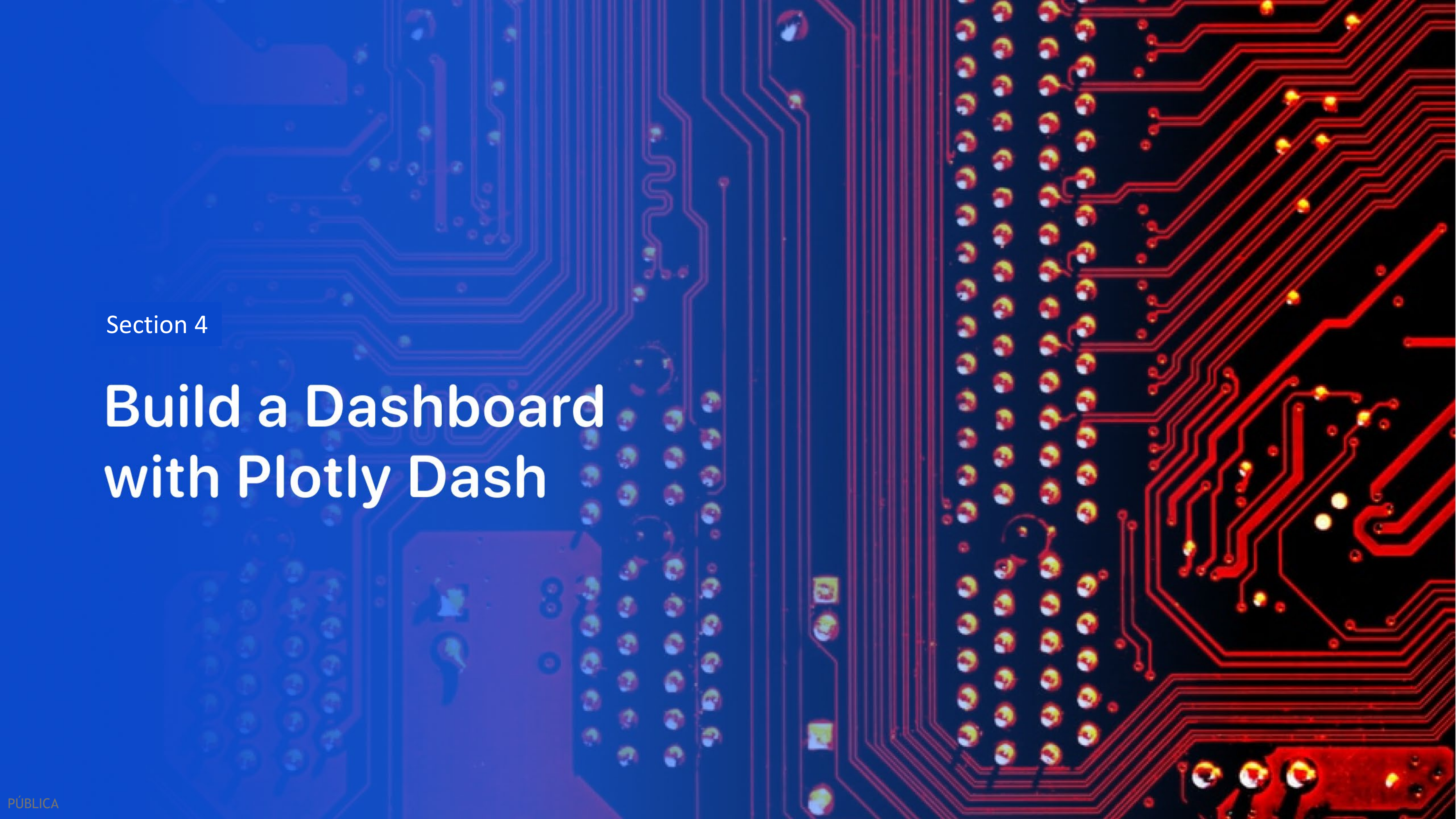


Distance from Points of Interest

The map below shows distance from coastline – about 350m (105 ft)

Besides the pointed importance of open ocean for the launch, other element close to launch sit may be relevant, like large roads and railroad for logistic of heavy loads (rocket themselves not built or serviced on launch sites)

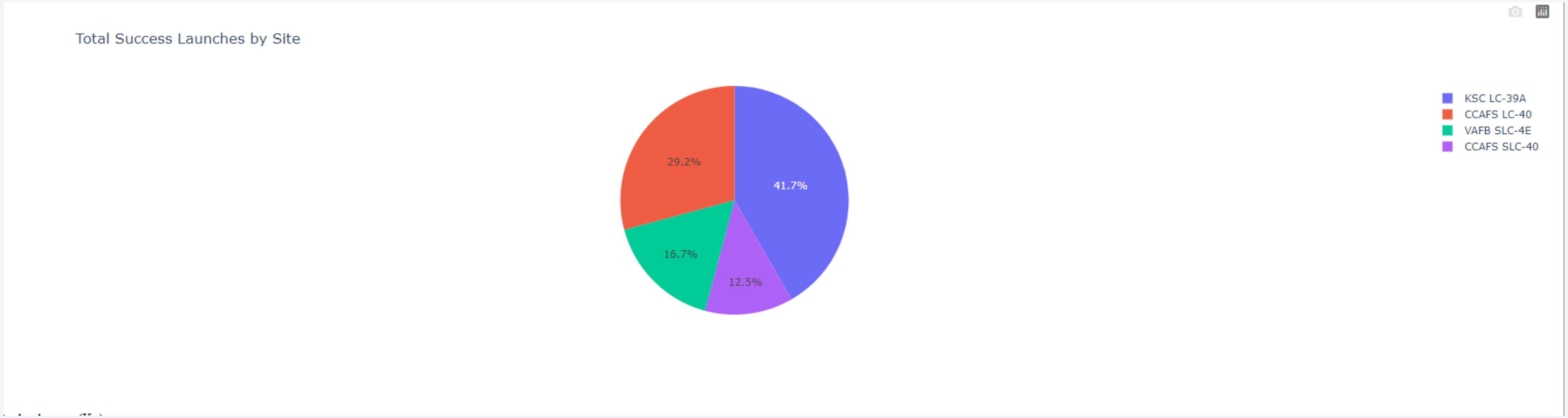




Section 4

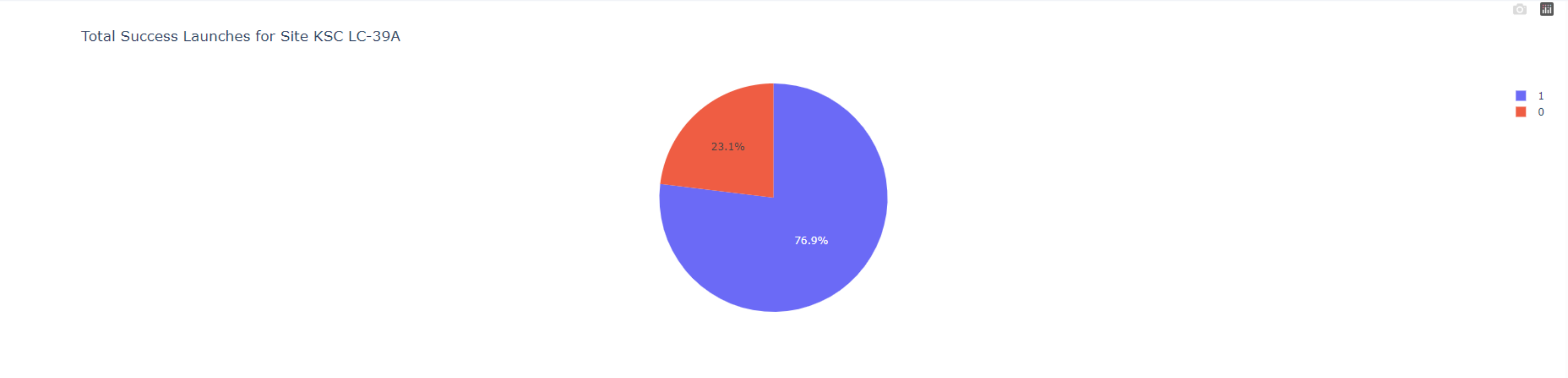
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



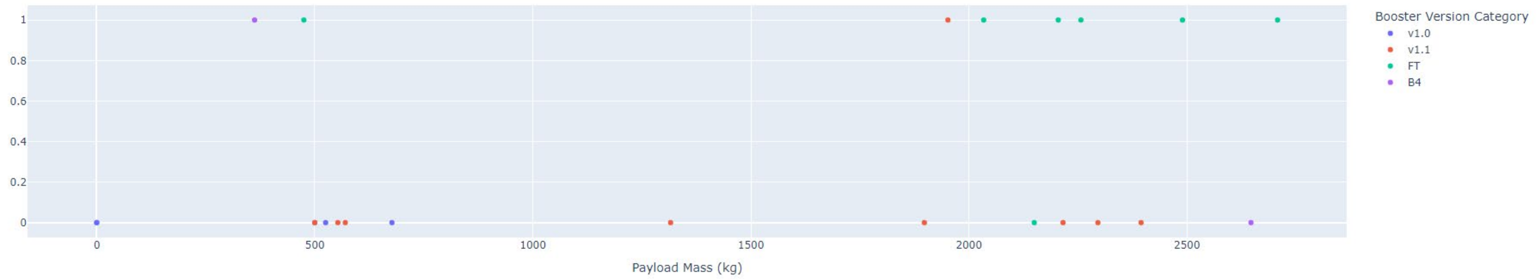
The KSC LC-39A booster have the highest mission success rate.

<Dashboard Screenshot 2>

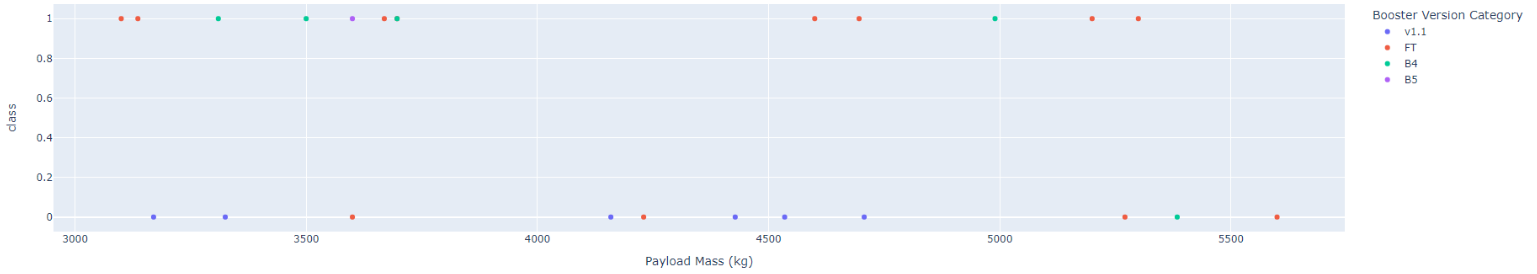


The KSC LC-39A has a 76.9% success rate.

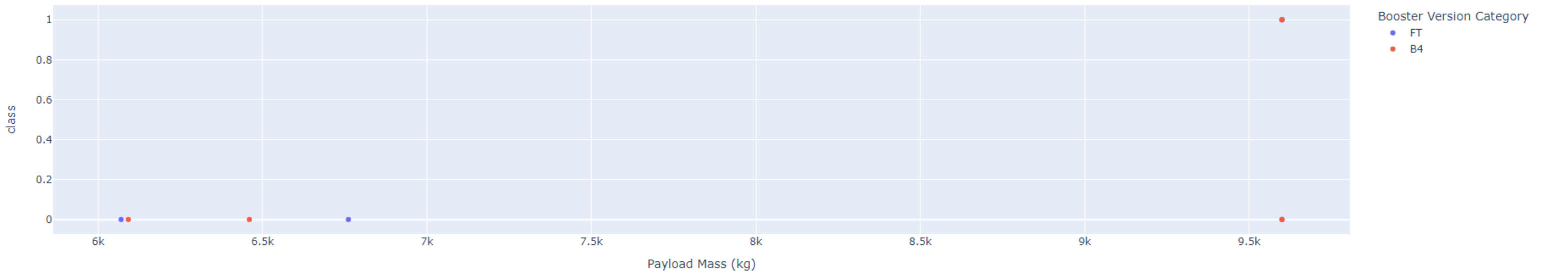
Paylod Correlations



Correlation between Payload and Success for all Sites



Correlation between Payload and Success for all Sites



Correlation between Payload and Success for all Sites

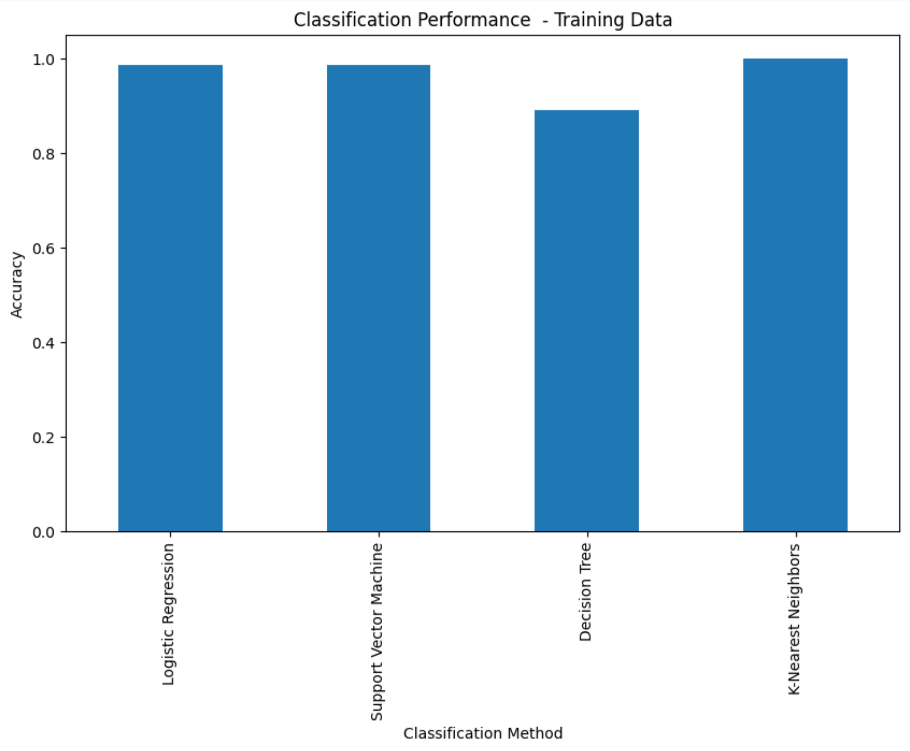
Payload Correlations

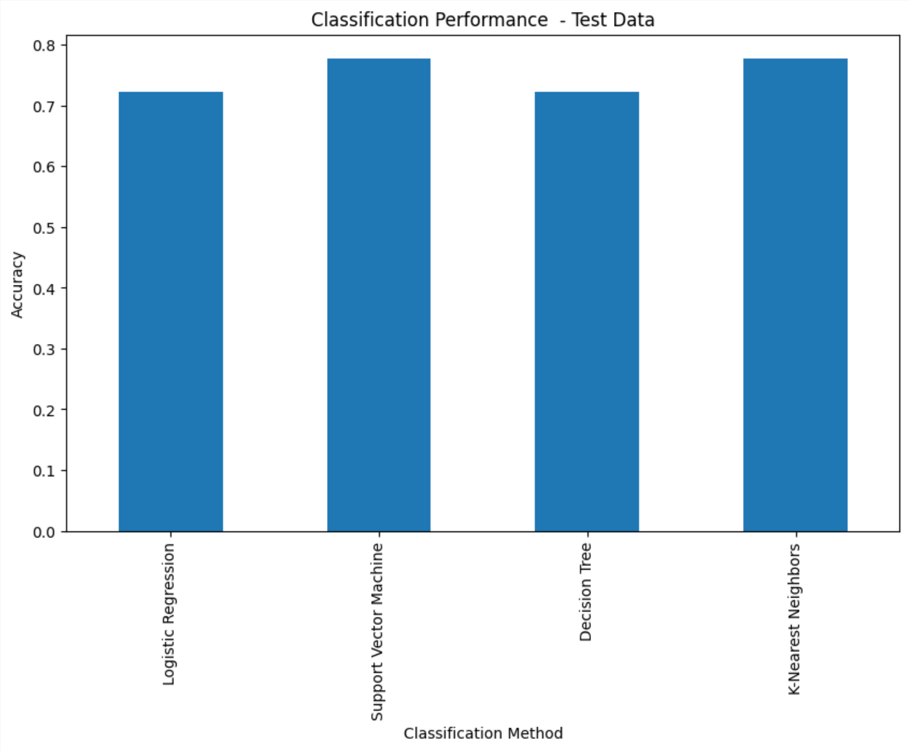
Successful mission are concentrated on 2000kg-6000kg, meaning mission on limit of capacity of boosters are more prone to failure.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



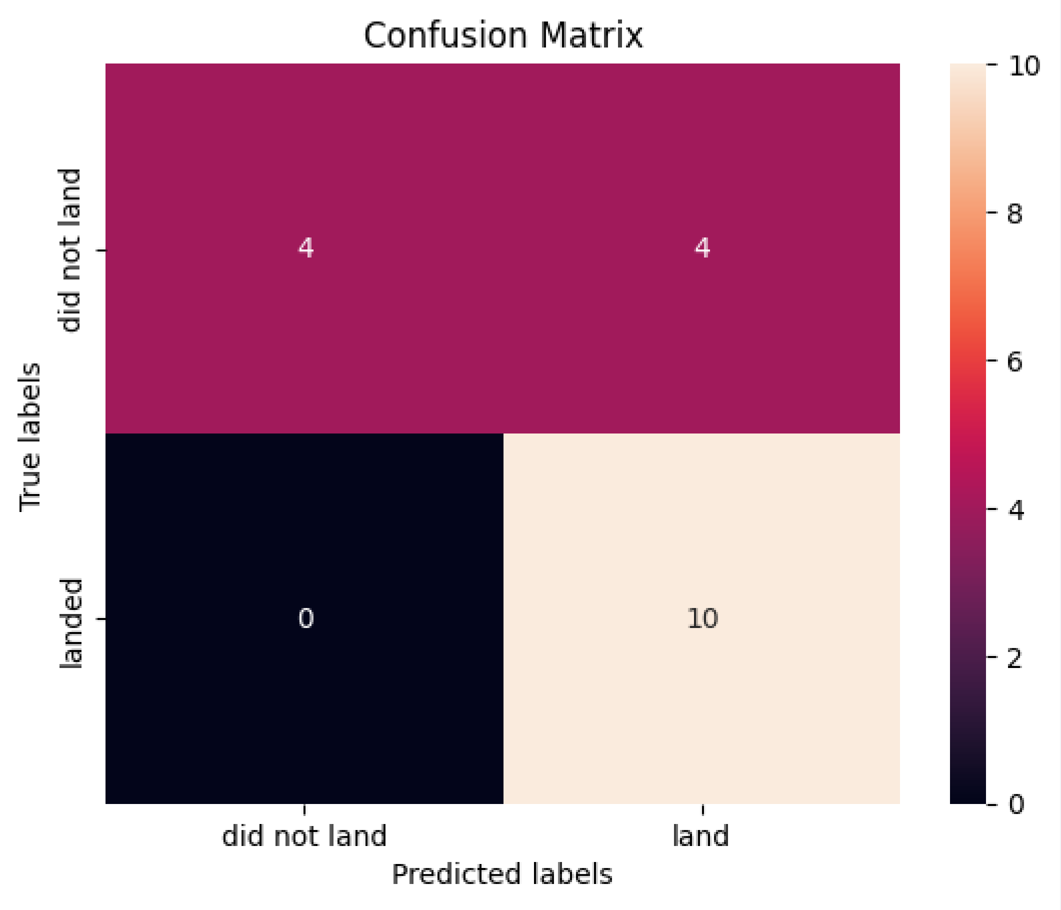


The model with best accuracy is KNN, with train accuracy of 1 and test accuracy with 0,778

Confusion Matrix

The confusion matrix compares the prediction of a model with the measured data

The matrix points no False Positives and 4 False Negatives, so the model is pessimistic with respect reality



Conclusions

- A key factor for mission success is experience in running missions (number of previous attempts);
- A higher rate of mission success can be reached if the first stage is not pushed into its lower or higher payload limits;
- The gathered data do not explain success rate differences between Launch Sites.
- The most performant classification model was KNN with SVM in second, with both having same test accuracy. KNN training accuracy of 1 can be a sign of overfitting, but more test data would be required to discard this model;
- This developments allow us to properly predict the success of a booster stage landing;

Appendix

About a prediction of cost launch:

This figures were given previously

SpaceX Launch Cost: USD 62 million

Others Launch Cost: USD 165 million

SpaceX Success Rate (from SQL section): s = 86%

SpaceX Failure Rate (from SQL section): f = 14%

Appendix

About a prediction of cost launch:

Assumptions:

Cost prediction for a Mission that will reuse the Booster;

SpaceX cost reflect always reusing the boosters;

Other companies do not reuse r boosters;

A launch cost estimate can be made:

$$Cost = NonReuse\ Cost\ \times Failure\ Rate + Reuse\ Cost\ \times Success\ Rate$$
$$Cost = 165\ \times 0.14\ + 62\ \times 0.86$$
$$Cost = USD\ 76.42\ million$$

Thank you!