Winning Space Race with Data Science

Enis Pinardag
26-09-2021

Outline

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

Executive Summary

- For the capstone assignment, we worked with SpaceX launch data. In order to gather the data we used 2 sources: SpaceX Rest API and Wikipedia webpage
- We improved the quality of the gathered raw data using data analysis techniques for data wrangling.
- After processing the raw data, we started exploring the data using SQL to gain insights.
- Then we applied basic statistical analysis and data visualization to gain further insights into the data.
- We built a dashboard using plotly dash and a map using folium to analyze the data interactively.
- Finally we built, evaluated, and refined predictive models using different algorithms to answer the classification problem at hand.

Findings:

- Based on the results of EDA, payload, orbit type and flight number variables have significant impact on the success rates for SpaceX launches.
- Decision Tree algorithm is the best machine learning algorithm for predicting the outcome of the SpaceX launches with given dataset.

Introduction

- SpaceX is one of the most successful commercial space age pioneers. They sent spacecraft to the International Space Station and manned missions to Space. They provided Starlink, a satellite internet constellation providing satellite Internet access. One reason they can do this is the rocket launches are relatively inexpensive. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards 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.
- For the capstone project we want to;
    - Determine the price of each launch by gathering information about SpaceX and creating dashboards for team.
    - Determine if SpaceX will reuse the first stage by training a machine learning model and use public information to predict if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

Executive Summary

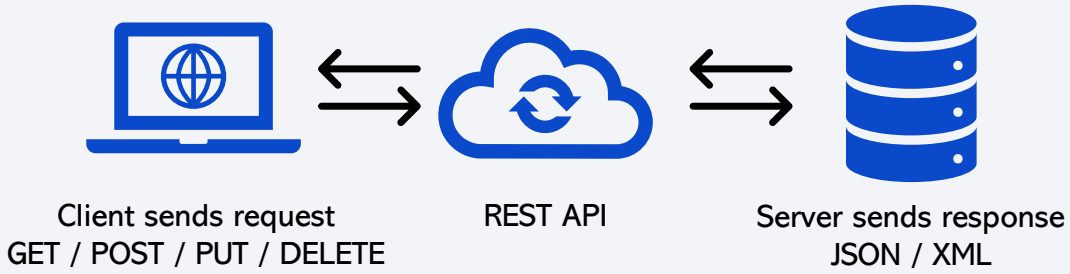
- Data collection methodology:
  - Collecting data with SpaceX REST API
  (https://api.spacexdata.com/v4/launches/past)
  - Collecting data using web scrapping from Falcon Heavy Launches Wikipage
  (https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922)
- Perform data wrangling
  - Processed raw data by handling missing values, applying one-hot encoding, calculating training labels
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Standardize the data, split into training data and test data, find best Hyperparameter for SVM, KNN, Decision Tree and Logistic Regression, calculate the accuracy for all the algorithms, choose the best performing algorithm

Data Collection

- For the capstone assignment, we worked with SpaceX launch data. In order to gather the data, we used the following data sources and methods:
  - **REST API**: We used “api.spacexdata.com/v4/launches/past” to target a specific endpoint of the API to get past launch data. We performed a GET request using the “requests” library to obtain the launch data, which we used to get the data from the API. Our response was in the form of a JSON, specifically a list of JSON objects. To convert this JSON to a “dataframe”, we used the “json_normalize” function. This function allowed us to “normalize” the structured json data into a flat table. Finally exported it into CSV file.
  - **Web Scraping**: As an alternative way to obtain Falcon 9 Launch data, we used the Python BeautifulSoup package to web scrape some HTML tables on the related Wiki pages that contain valuable Falcon 9 launch records. Then we parsed the data from those tables and convert them into a “Pandas” “dataframe” for further visualization and analysis. We transformed this raw data into a clean dataset. Finally exported it into CSV file.

Data Collection – SpaceX API

1) Sending GET request
2) Collecting JSON response
3) Normalizing json into dataframe
4) Cleaning and converting features
5) Creating a dictionary from selected features
6) Transform dictionary to dataframe
7) Filtering dataframe and exporting into a flat file



- GitHub - Data Collection API Notebook

Data Collection – Scraping

1) Sending GET request
2) Creating BeautifulSoup object from response
3) Finding the related HTML tables
4) Extracting feature set
5) Parsing HTML tables to create dictionary
6) Transform dictionary to dataframe
7) Exporting into a flat file

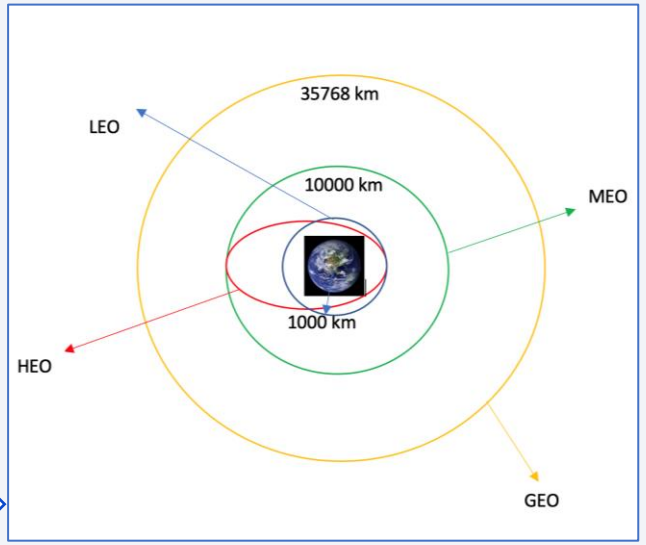
- GitHub - Data Collection Scraping Notebook

Data Wrangling

- 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.
- We converted those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.
- We also processed raw data by handling missing and applying one-hot encoding where necessary.

- GitHub - EDA/Data Wrangling Notebook

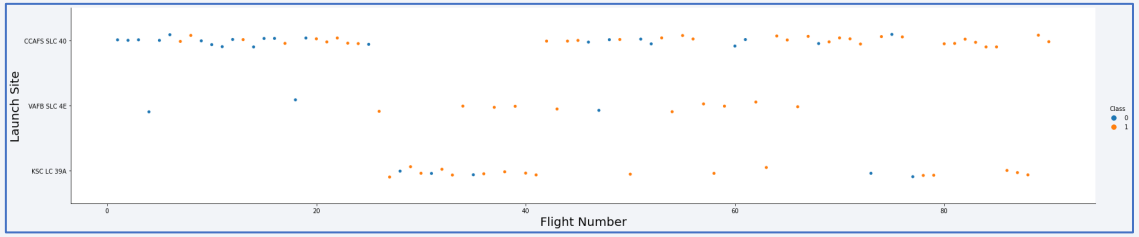
Each launch aims to a dedicated orbit. Here are some common orbit types:



EDA with Data Visualization

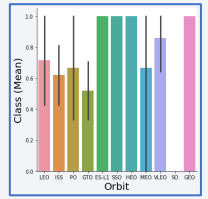
- We applied data visualization to gain further insights into the data. In order to do that, we plotted scatter point charts, bar charts and line charts to visually check if there are any relationship between selected features.
  - We plotted scatter point charts to observe and show relationships between two numeric variables.

    Following charts were plotted:
    - Payload mass vs Flight number
    - Launch site vs Flight number
    - Launch site vs Payload mass
    - Orbit vs Flight number
    - Orbit vs Payload mass



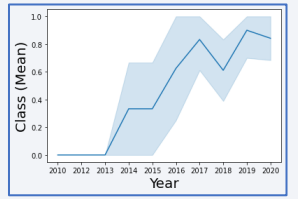
  - We plotted bar charts to perform a comparison of metric values across different subgroups.

    Following chart were plotted:
    - Orbit vs Succes rate



  - We plotted line charts to track changes over short or long periods of time.

    Following chart were plotted:
    - Succes Rate vs Year



- [GitHub - EDA Data Visualization Notebook]

EDA with SQL

- In order to gain a preliminary understanding and get acquainted with the dataset following queries were performed:
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string “CCA”
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date when the first successful landing outcome in groud pad was achieved
  - Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 kg but less than 6000 kg
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster_versions which have carried the maximum payload mass, using subquery
  - Listing the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - Ranking 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

- GitHub - EDA SQL Notebook

Build an Interactive Map with Folium

- We added each site's location on a map using site's latitude and longitude coordinates to gain insights. We used “folium.Circle” to add a highlighted circle area with a text label on these locations.
- We also added markers for each site and created icons showing launch site name.
- In order to visualize multiple launch outcomes for each launch site, we used “MarkerCluster()” and assigned color to launch outcomes, green for successful launches (class = 1) and red unsuccessful ones (class = 0).
- To explore and analyze the proximities of launch sites, we added a “MousePosition()” on the map to get coordinate for a mouse over a point on the map
- By using calculate_distance function, we calculated distances between launch sites and closest railways, coastlines and city centers.
- We drew a “PolyLine()” between CCAFS-SLC40 to nearest railway point and coastline point.

Findings:

- By visualizing all the objects on a map, we discovered that launch sites are in close proximity to coastline and have access to railroads and keep certain distance away from highways and cities.

- GitHub - Interactive Visual Analytics Folium Lab Notebook

Build a Dashboard with Plotly Dash

- In order to perform interactive visual analytics on SpaceX launch data in real-time, we build an interactive dashboard using plotly dash. We added following components to dashboard:
  - Added a launch site drop-down input component
  - Added a callback function to render success-pie-chart based on selected site dropdown
  - Added a range slider to select payload
  - Added a callback function to render the success-payload-scatter-chart scatter plot

Findings:

- By using the dashboard, we can analyze the data in real time and answer many question about SpaceX launches such as;
  - Which site has the largest successful launches?
  - Which site has the highest launch success rate?
  - Which payload range(s) has the highest launch success rate?
  - Which payload range(s) has the lowest launch success rate?
  - Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate?

- GitHub - Interactive Dashboard with Plotly Dash

Predictive Analysis (Classification)

- Building model
  - Separating dependent (Y) and independent (X) variables
  - Standardizing independent variables
  - Splitting data into training and test datasets
  - Performing grid search with different algorithms to optimize hyperparameters
  - Training and testing models using different algorithms with optimized hyperparameters
- Evaluating model
  - Checking accuracy for each model
  - Plotting confusion matrix for each model
- Improving model
  - Feature engineering
  - Dimension reduction
- Finding the best performing classification model
  - The model with the best accuracy score is selected as the champion model

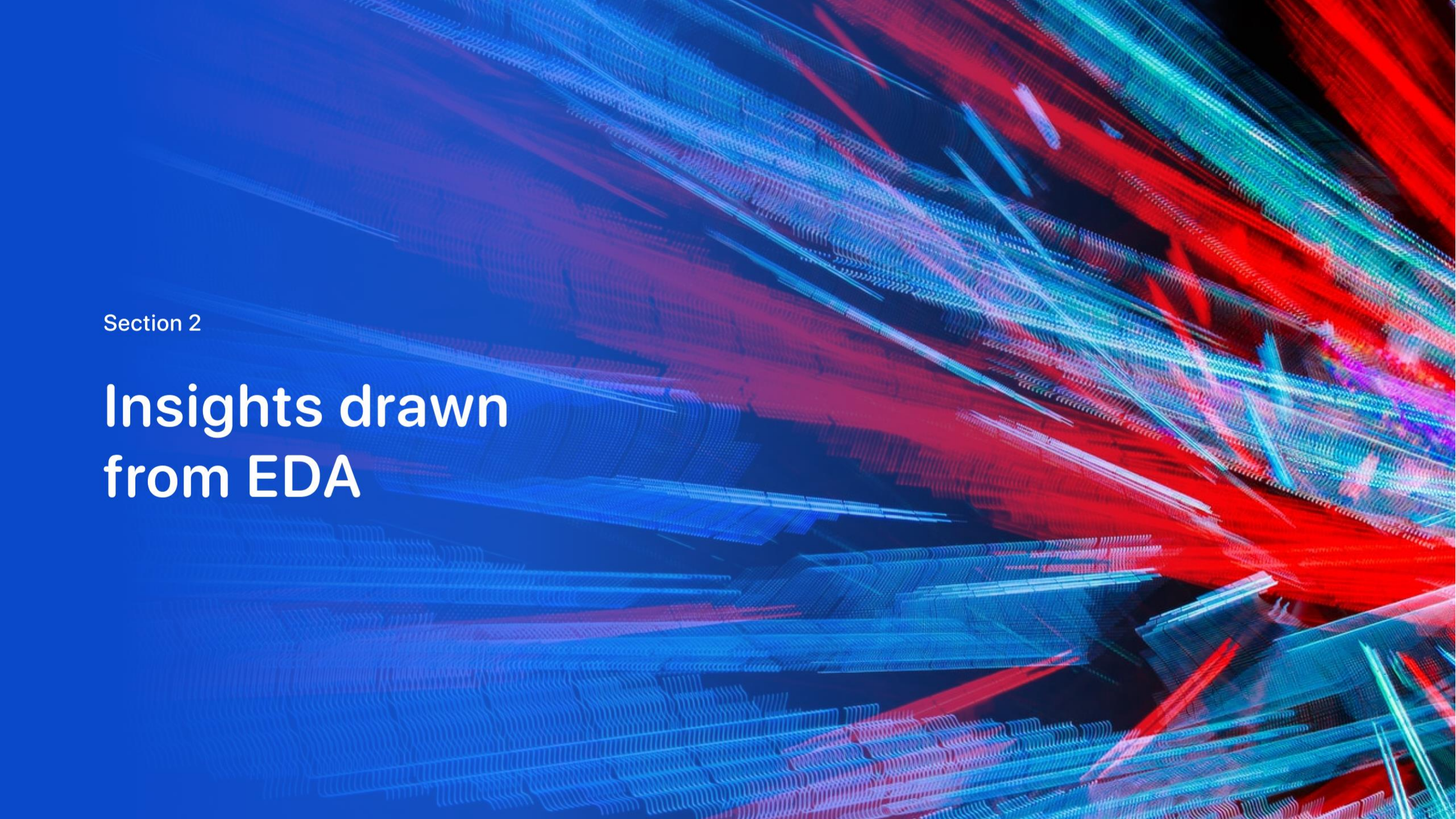
- GitHub - Machine Learning Prediction

Results

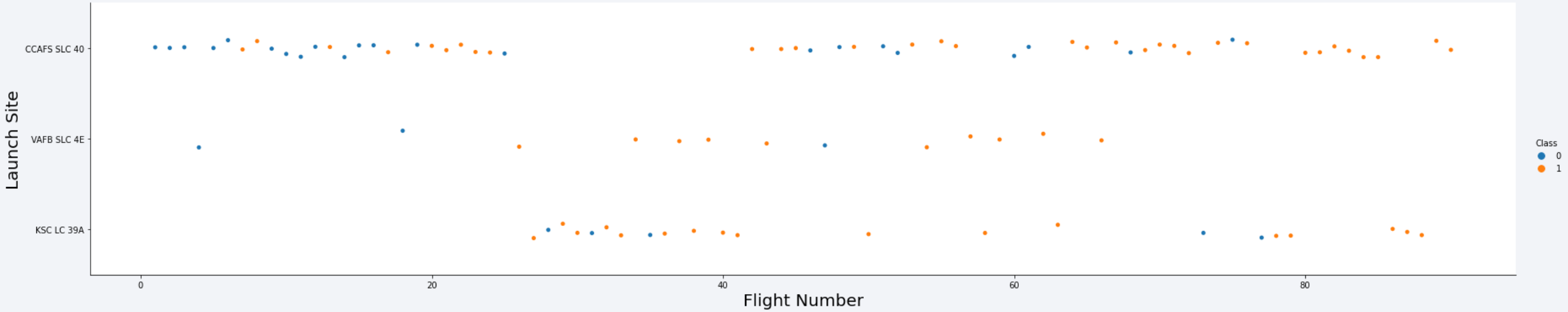
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

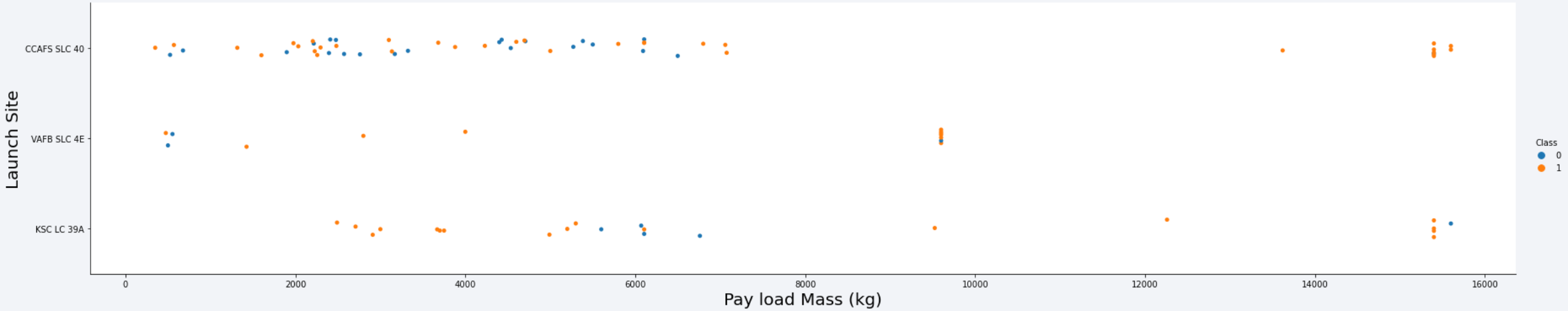


Flight Number vs. Launch Site



- As the number of flights rises, success rate also rises.

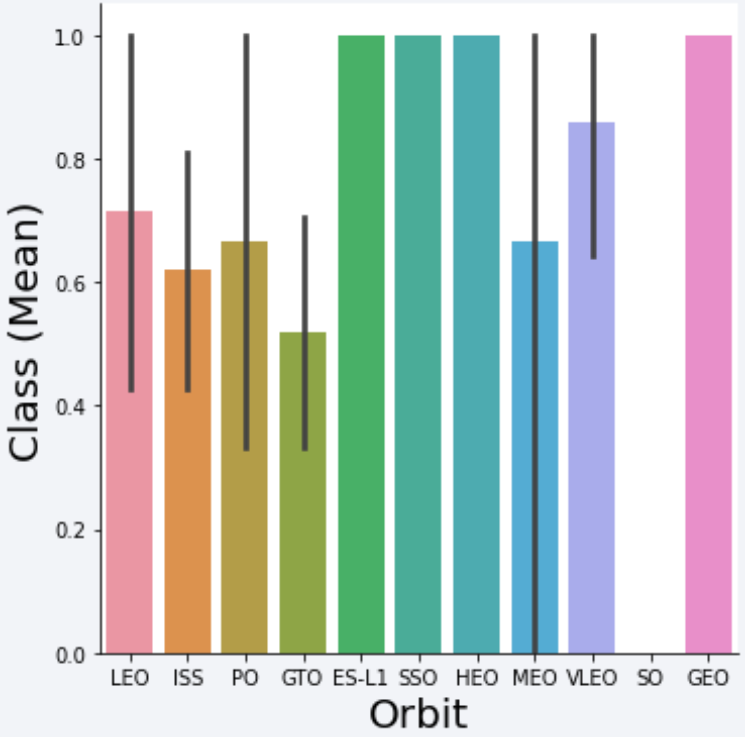
Payload vs. Launch Site



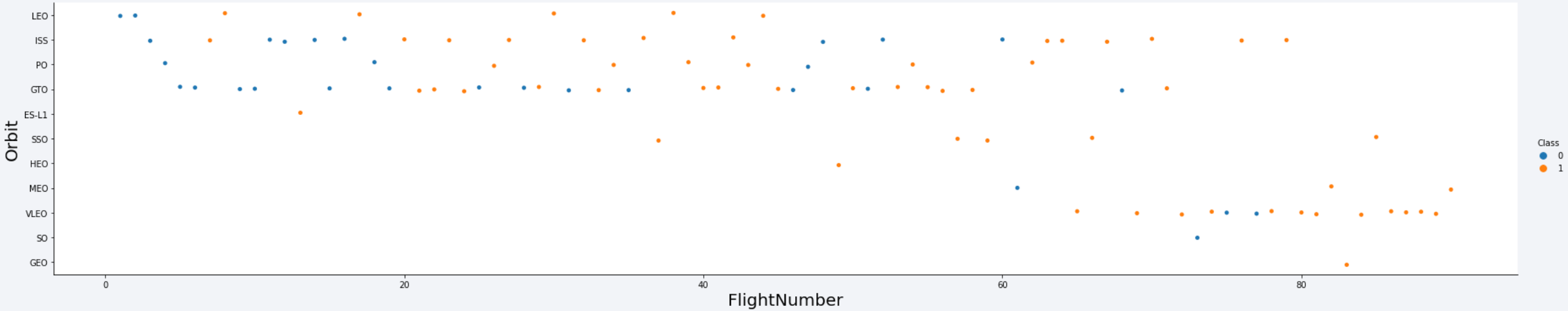
- CCAFS SLC 40 has higher success rate when the payload is greater then ~6.500 kg. KSC LC 39A on the other hand has higher success rate with payloads less than ~5.500 kg.

Success Rate vs. Orbit Type

- ES-L1, SSO, HEO and GEO orbits have 100% success rate.



Flight Number vs. Orbit Type



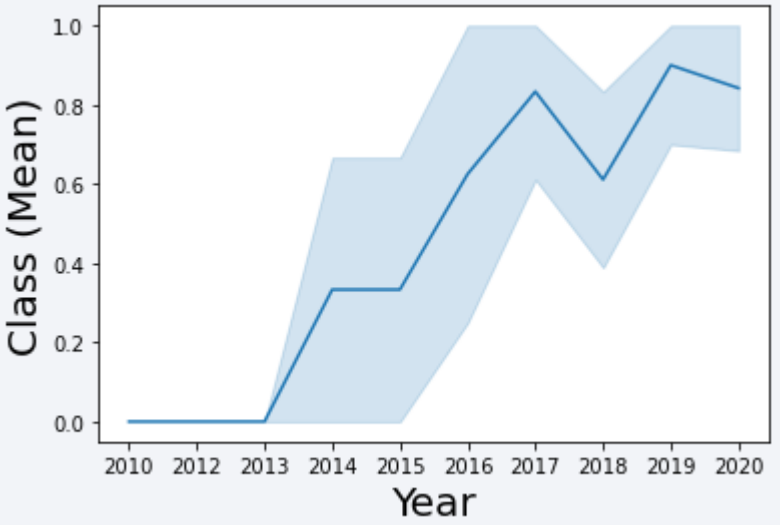
- LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

- Heavy payloads have a negative influence on GTO orbits and positive on Polar LEO (ISS) orbits.

Launch Success Yearly Trend

- We can observe that the success rate since 2013 kept increasing till 2017 and then dropped in 2018.
- In 2019, success rate reached the highest point between 2010 and 2020.



All Launch Site Names

```
%sql select distinct launch_site from spacextbl
```

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

- By using “distinct”, we list each unique launch_site only once.

Launch Site Names Begin with 'CCA'

%sql select * from spacextbl where launch_site like 'CCA%' limit 5

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- By using “limit”, we list only 5 records where launch_site starts with “CCA”.

Total Payload Mass

```
%sql select customer, sum(PAYLOAD_MASS__KG_) as total_payload_mass from spacextbl where customer = 'NASA (CRS)' group by customer
```

customer	total_payload_mass
NASA (CRS)	45596

- By using the function SUM, we summate the total payload for customer NSA (CRS).

Average Payload Mass by F9 v1.1

```
%sql select Booster_Version, avg(PAYLOAD_MASS__KG_) as average_payload_mass from spacextbl where Booster_Version = 'F9 v1.1' group by Booster_Version
```

booster_version	average_payload_mass
F9 v1.1	2928

- By using the function AVG, we calculate the average payload for booster version F9v1.1.

First Successful Ground Landing Date

```
%sql select Landing__Outcome, min(Date) as first_date from spacextbl where Landing__Outcome = 'Success (ground pad)' group by Landing__Outcome
```

landing__outcome	first_date
Success (ground pad)	2015-12-22

- By using the function MIN, we calculate the minimum date for successful landing outcome in ground pad.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select distinct Booster_Version from spacextbl where Landing__Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

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

- By using “distinct”, we list each unique Booster_Version ,which have success in drone ship and have payload mass greater than 4000 but less than 6000, only once.

Total Number of Successful and Failure Mission Outcomes

```
%sql select (case when Mission_Outcome like 'Success%' then 'Success' else 'Failure' end) as outcome, count(Mission_Outcome) as total_number from spacextbl group by (case when Mission_Outcome like 'Success%' then 'Success' else 'Failure' end)
```

outcome	total_number
Failure	1
Success	100

- By using conditional variable (outcome), we list successful and failure mission outcomes.

Boosters Carried Maximum Payload

```
%sql select distinct Booster_Version from spacextbl where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextbl)
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

- By using a subquery to find the max payload, we list booster versions which have carried the maximum payload mass.

2015 Launch Records

```
%sql select distinct Landing__Outcome, Booster_Version, Launch_Site, Date from spacextbl where Landing__Outcome = 'Failure (drone ship)' and year(Date) = 2015
```

landing__outcome	booster_version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

- By using year function, we list failed landing_outcomes in drone ship in year 2015.

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

```
%sql select Landing__Outcome, count(Landing__Outcome) as total from spacextbl where Date between '2010-06-04' and '2017-03-20' group by Landing__Outcome order by count(Landing__Outcome) desc
```

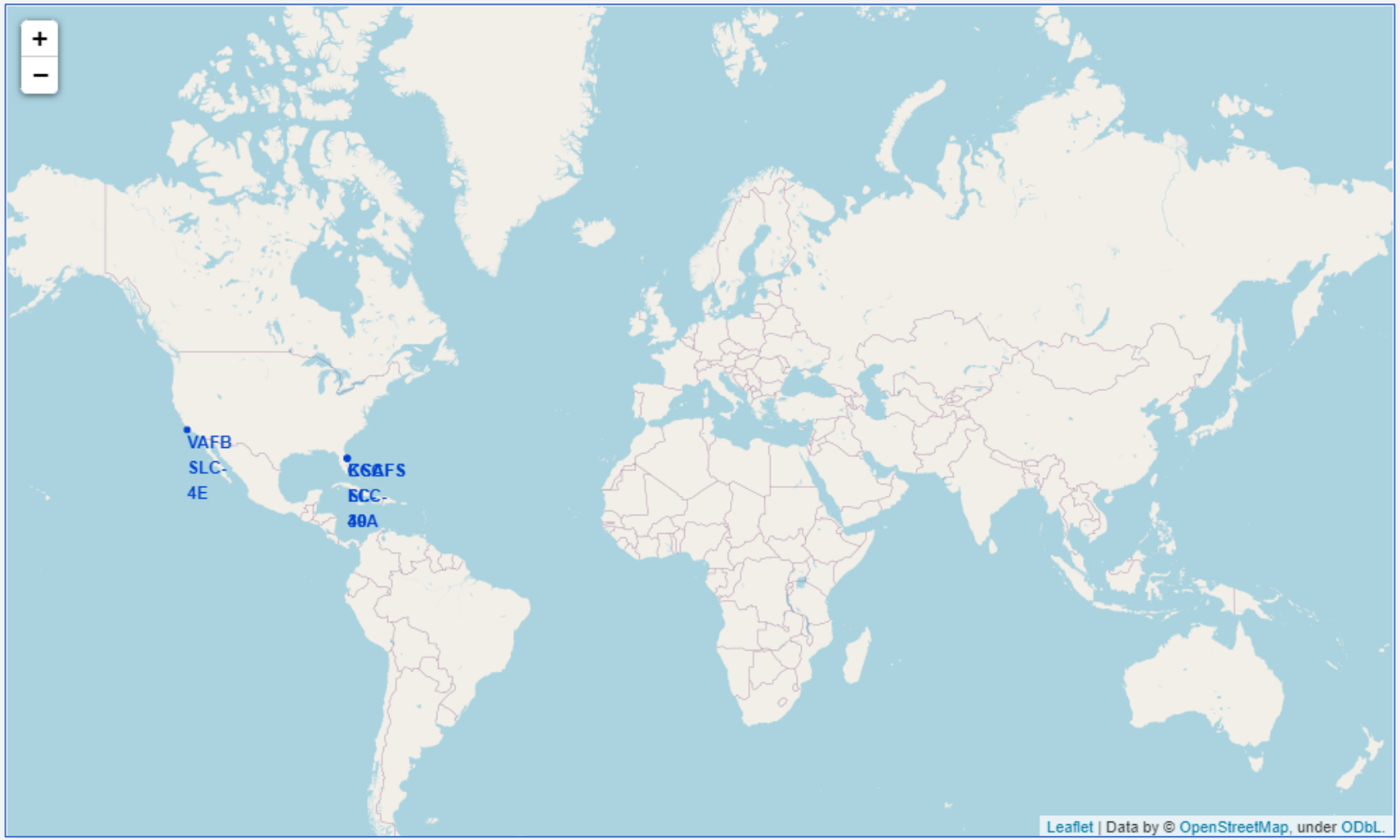
landing__outcome	total
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- By using “order by”, we list frequencies of different landing outcomes in a descending order.

Section 4

Launch Sites Proximities Analysis

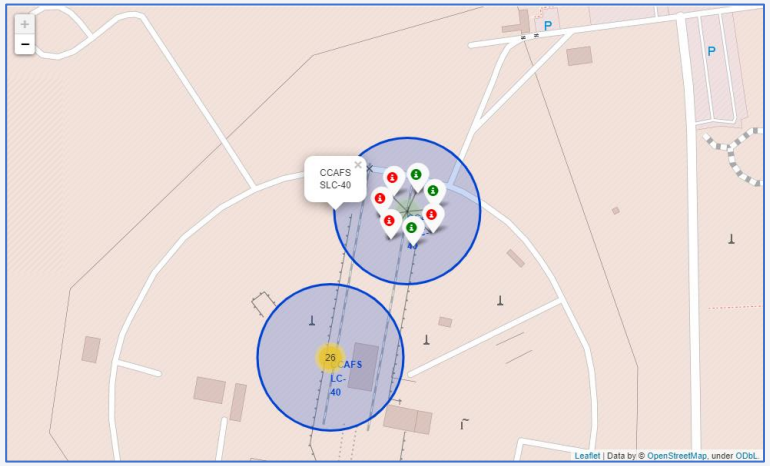
All Launch Sites on Global Map



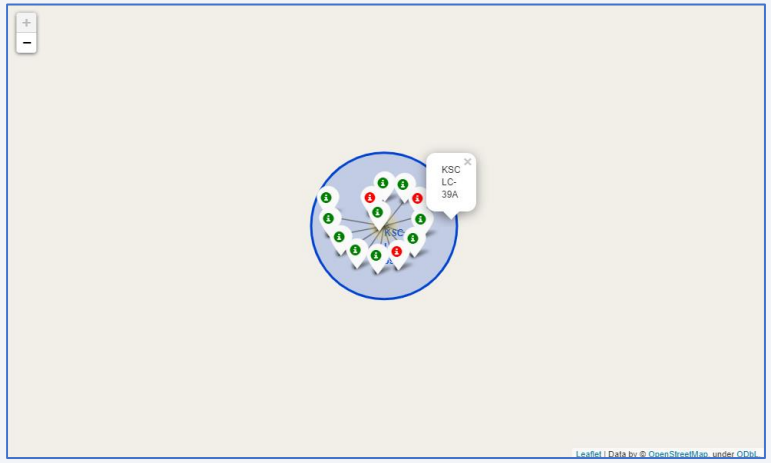
- SpaceX launch sites are located in Florida and California.
- They are located close to equator.

Color-labeled Launch Outcomes on the Map

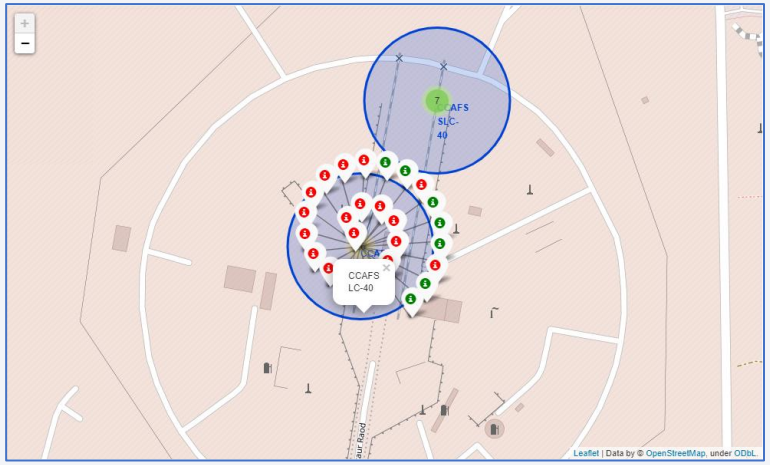
CCAFS SLC-40



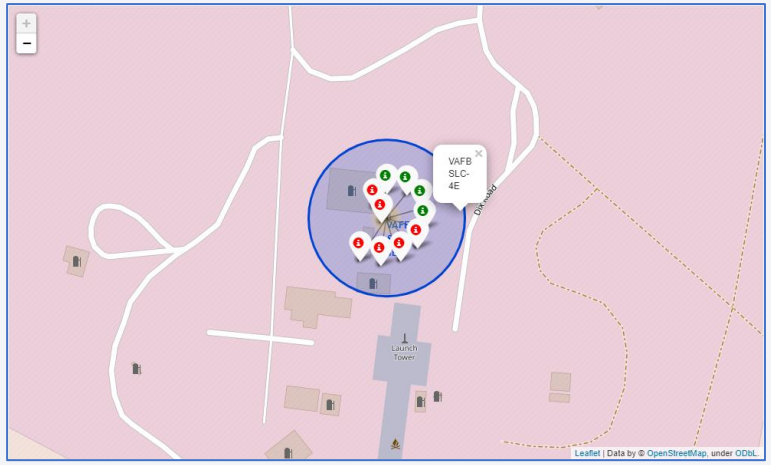
KSC LC-39A



CCAFS LC-40

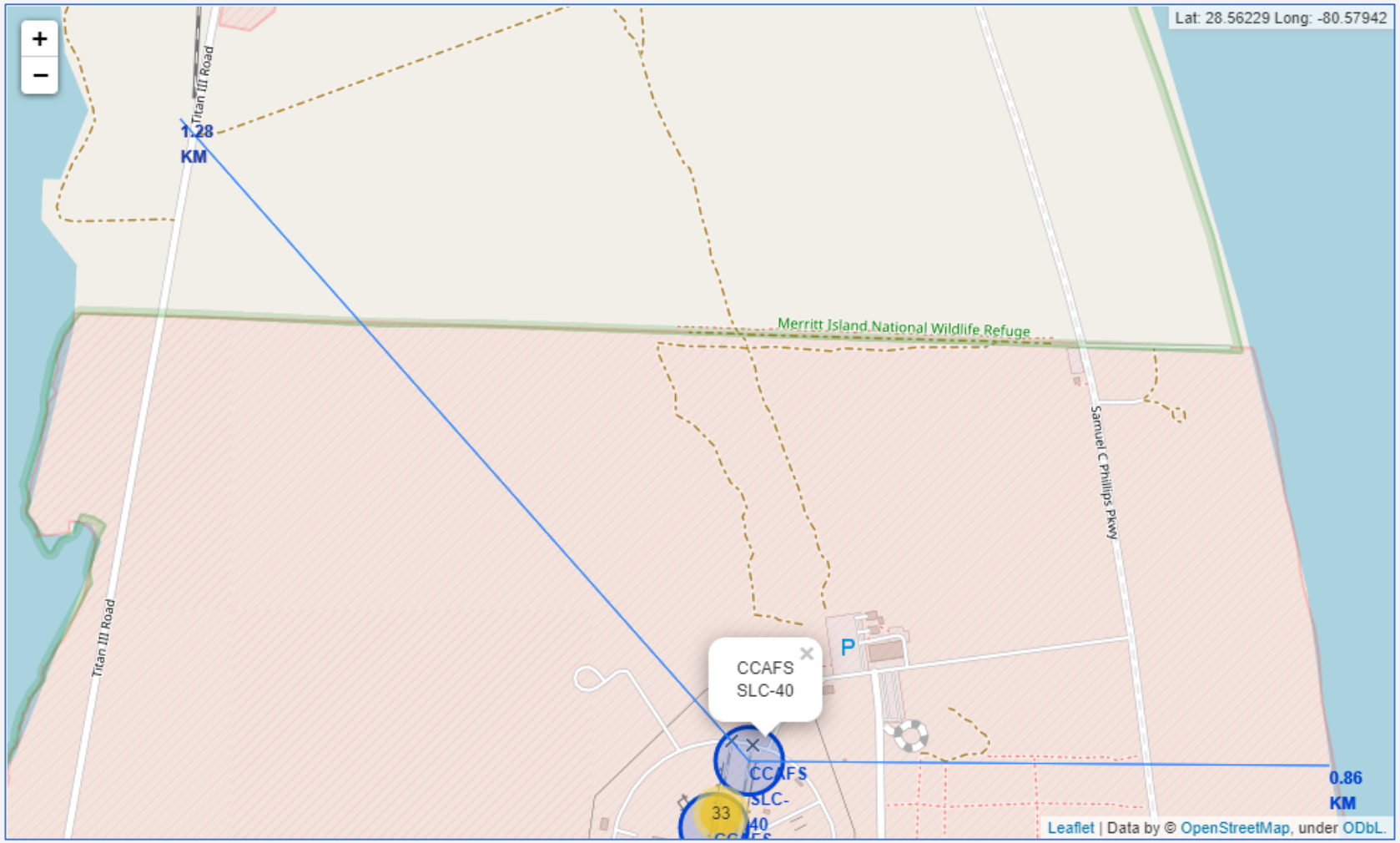


VAFB SLC-4E



- CCAFS LC-40 is the launch site with the most launches, but the success rate is fairly low.
- KSC LC-39 has the second most launches and has a good success rate.

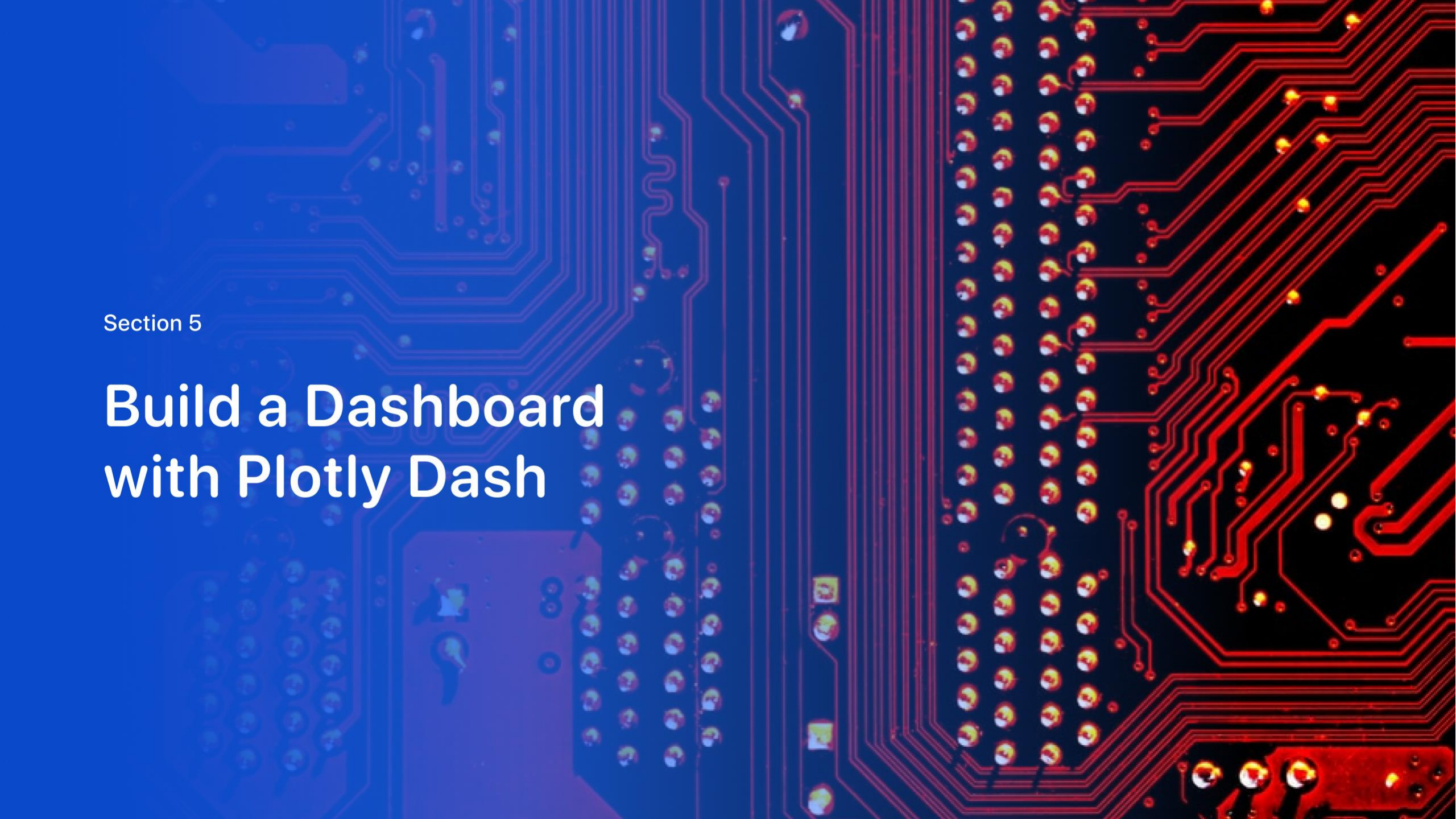
CCAFS SLC-40's Proximities to Railway and Coastline



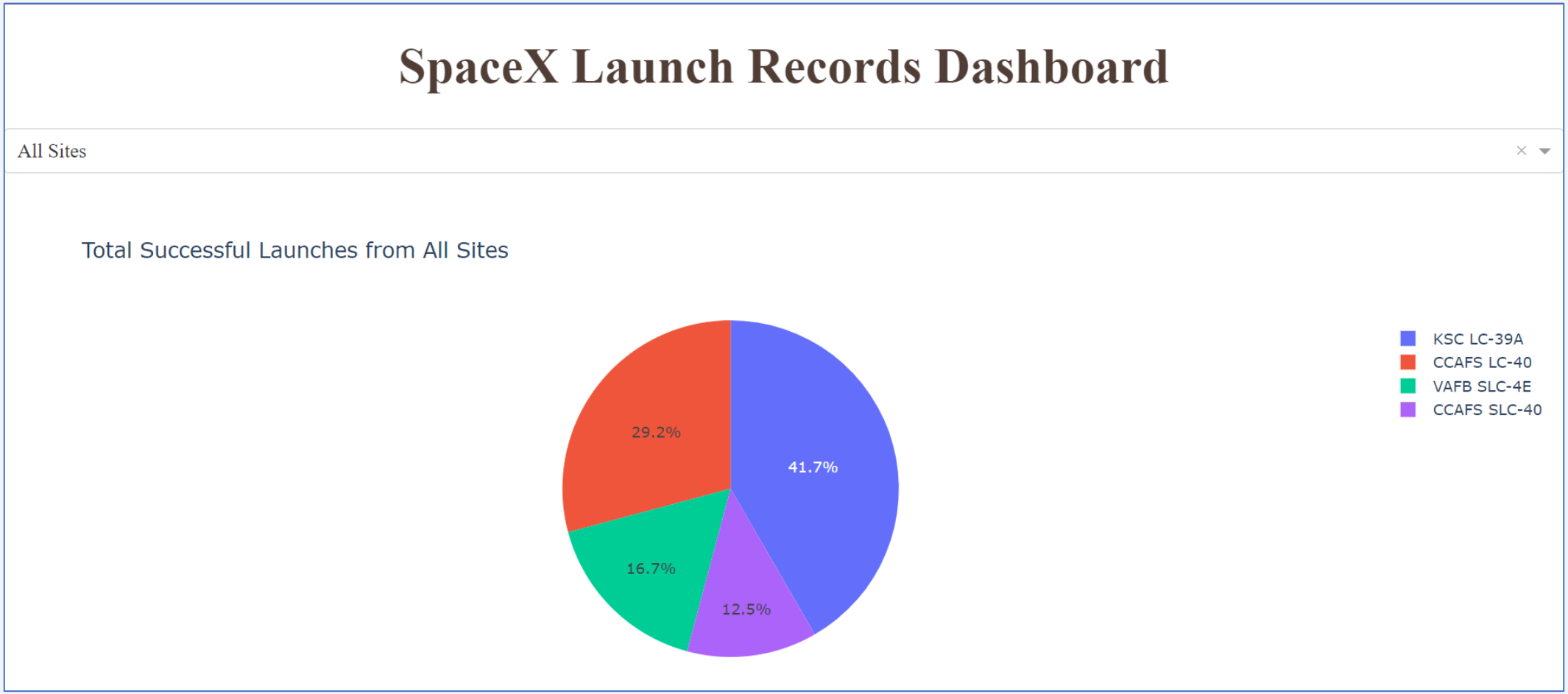
- Launch sites are in close proximity to coastline and have access to railroads and they keep certain distance away from highways and cities.

Section 5

Build a Dashboard with Plotly Dash

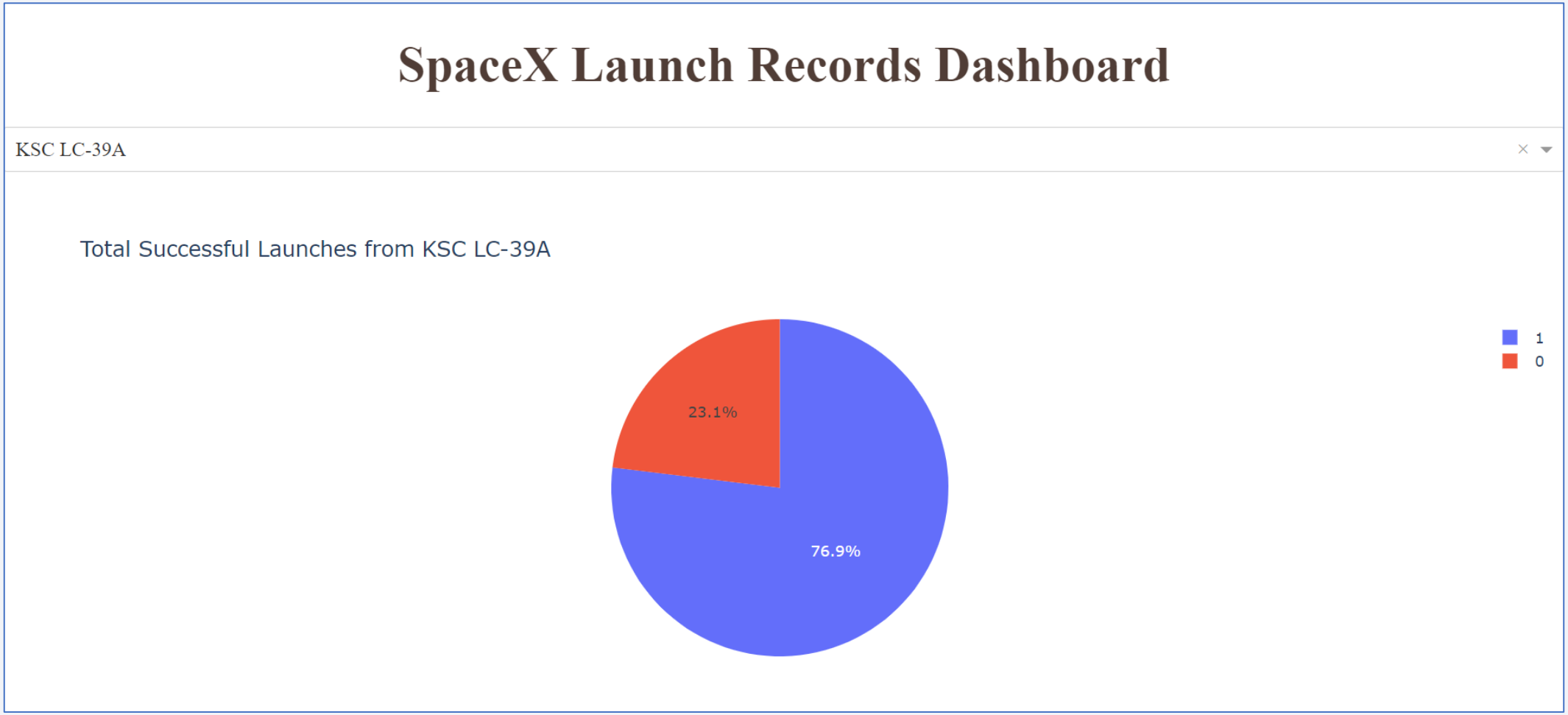


Launch Success Count for All Sites



- KSC LC-39A had the most successful launches from all the sites

KSC LC-39A



- KSC LC-39A has 76.9% success rate on launches.

Payload vs. Launch Outcome Scatter Plot for All Sites

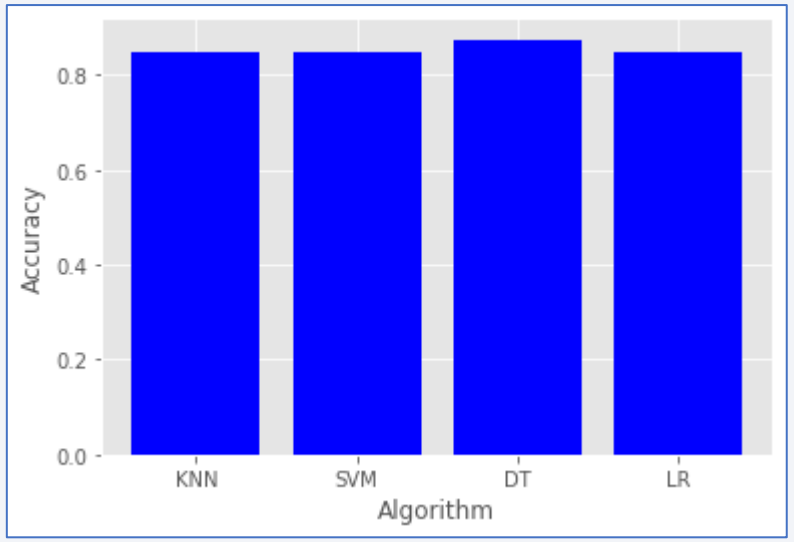
- FT booster had 80% success rate when the payload was between 2.000 and 4.000 kg.
- V1.1 booster failed every launch for the same payloads.

Section 6

Predictive Analysis (Classification)

Classification Accuracy

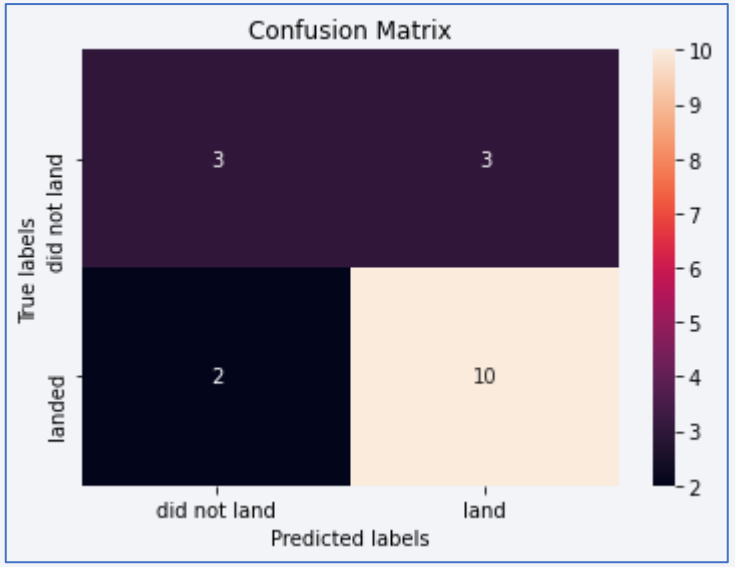
- All 4 algorithms produced accuracy ratios quite close to each other:
  - K-Nearest Neighbors: 0.848,
  - Support Vector Machines: 0.848,
  - Decision Tree: 0.875,
  - Logistic Regression: 0.846
- Decision tree algorithm has the highest classification accuracy.
- Optimized hyperparameters were as follows:
  - Criterion: Gini
  - Max depth: 4
  - Max_features: Sqrt
  - Min samples leaf: 1
  - Min samples split: 2
  - Splitter: Random



- KNN: K-Nearest Neighbors
- SVM: Support Vector Machines
- DT: Decision Tree
- LR: Logistic Regression

Confusion Matrix

- Decision Tree is chosen as the best algorithm based on the high accuracy ratio.
- Decision Tree can distinguish between the different classes.
- The major problem is false positives, where algorithm predicts that the rocket would land but in fact it didn't.



Conclusions

- SpaceX launch sites are in close proximity to coastline and have access to railroads and they have certain distance away from highways and cities.
- Launch sites are located in Florida and California, close to equator.
- CCAFS LC-40 is the launch site with the most launches, but the success rate is fairly low. KSC LC-39 has the second most launches and has a success rate of 76.9%. It's the most successful launch site among 4.
- Low weighted payloads perform better. Payloads between 2.500 and 5.000 kg have the best success rate. FT booster had 80% success rate when the payload was between 2.000 and 4.000 kg.
- ES-L1, SSO, HEO and GEO orbits have 100% success rate.
- As the number of flights rises, success rate also rises. We can observe that the success rate since 2013 kept increasing till 2017 and then dropped in 2018. In 2019, success rate reached the highest point between 2010 and 2020.
- Decision Tree is chosen as the best algorithm based on the high accuracy ratio. But model incorrectly labels significant number of failed landings as successful landings.

Thank you!