Winning Space Race with Data Science

Ryan Scott
June 5th, 2022

Outline

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

Executive Summary

- In this research study, we used the following methodologies:
  - SpaceX API & web scraped the launch data from Wiki
  - Performed an exploratory data analysis using SQL and Python
  - Created visualizations using Folium & Plotly
  - Classification Predictive Analysis
- In research study, we have determined the following results:
  - Price of each launch:
  - Will we be able to reuse the first stage:

Introduction

- SpaceY is a space exploration company with the mission of reducing costs and promoting space tourism

- For this study, we wanted to determine the following questions:
    - What is the price for each launch?
    - Will we be able to use the first phase from the launch?

Section 1

Methodology

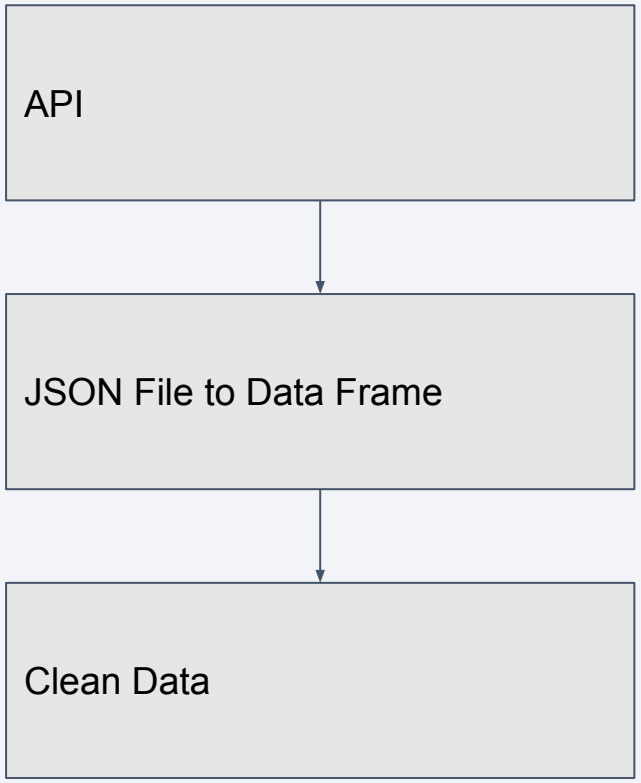
Methodology

Executive Summary

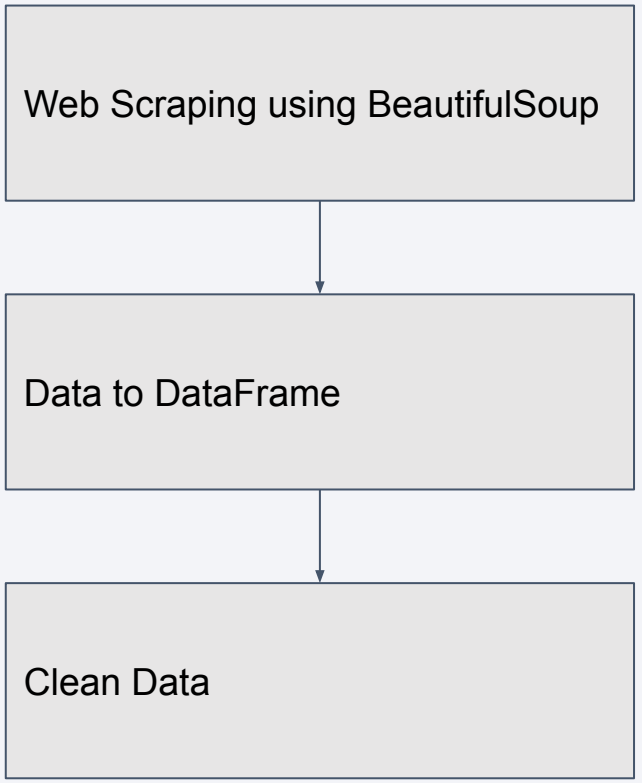
- Data collection methodology:
  - Rest API from SpaceX
  - Web scraping data from Wiki
- Perform data wrangling
  - The data was placed into data frames and cleaned using the Pandas library
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

Used the Rest API from SpaceX to collect data



Scraped launch data from Wikipedia



Data Collection – SpaceX API

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```



```
response.status_code
```



```
response = requests.get(static_json_url)

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

Github

Data Collection - Scraping

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(static_url).text
```



```
soup = BeautifulSoup(response, 'html.parser')
html_tables = soup.find_all('table')
```

Github

Data Wrangling

- We used the pandas library to view and count the data in our columns and loops to transform some data into floats.

```
landing_class = []
for i in df['Outcome']:
    if i in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```



```
df['Class']=landing_class
df[['Class']].head(8)
```

GitHub

EDA with Data Visualization

- To visualize our data, we used the following graphs:
  - Scatterplot of PayloadMass and Flight number
  - Scatterplot of Flight number and Launch site
  - Bar chart of Orbit and Class
  - Line plot of Class By Year

GitHub

EDA with SQL

Following EDA with SQL tasks were performed:

- 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 successful 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. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for 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

GitHub

Build an Interactive Map with Folium

We created a map using the Folium library and noted the following key points, marked with an orange circle:

- NASA Johnson Space Center
- Launch Sites
- Markers presenting successful (Green) and unsuccessful (red) launches
- Markers presenting distance from launch site to key locations

Github

Build a Dashboard with Plotly Dash

For better understanding and visualizing data investigated in the project dashboard with dropdown was implemented.

Its conencted to pie chart representing success/fail ratio of a chosen launch site.

It is also possible to navigate through payload mass with use of a rangeslider

Scatter plot was used to prsent relationship between success and payload mass.

GitHub

Predictive Analysis (Classification)

- The best model was found and implemented out of four tests: KNN, Logistic Regression, SVM, and Decision Tree

- GridSearch CV with determined parameters were used for each model

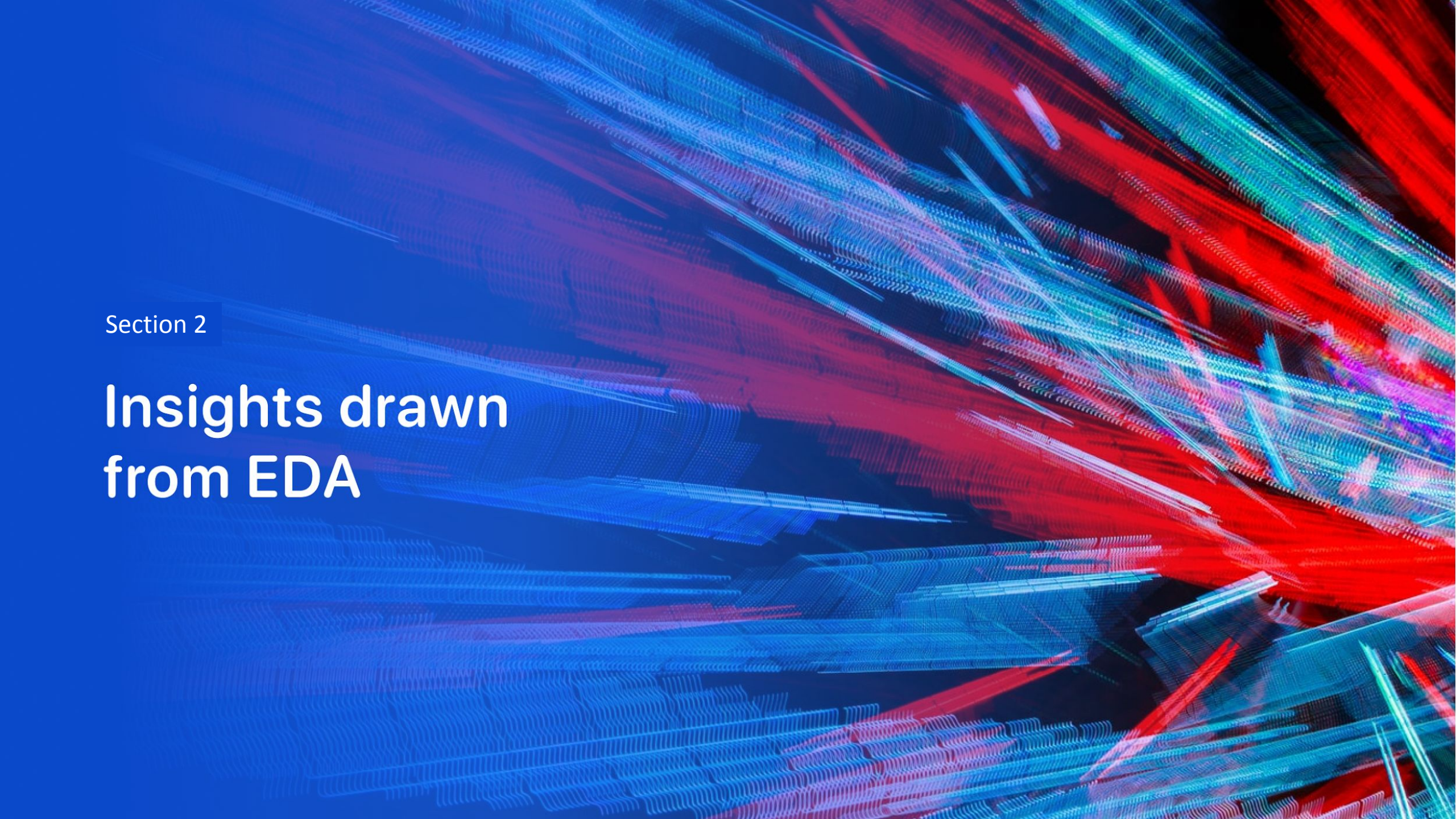
GitHub

Results

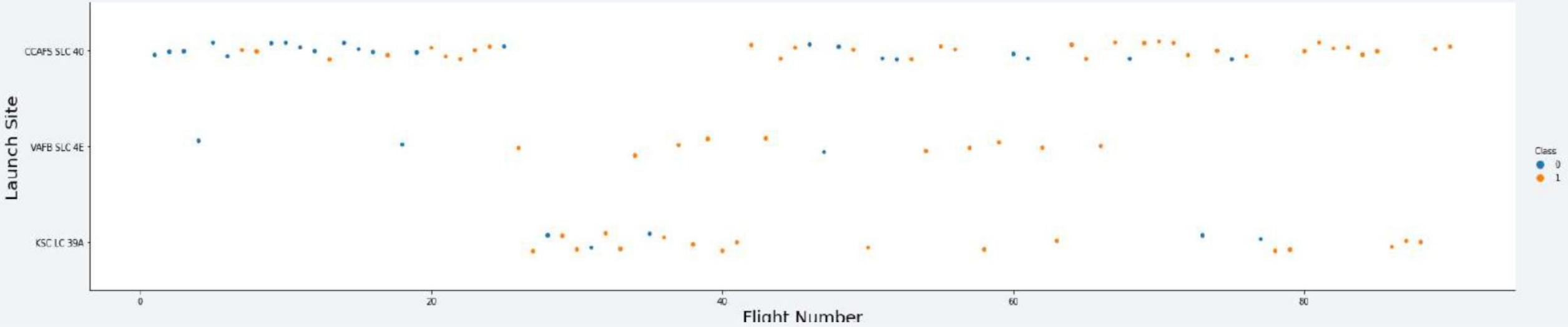
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

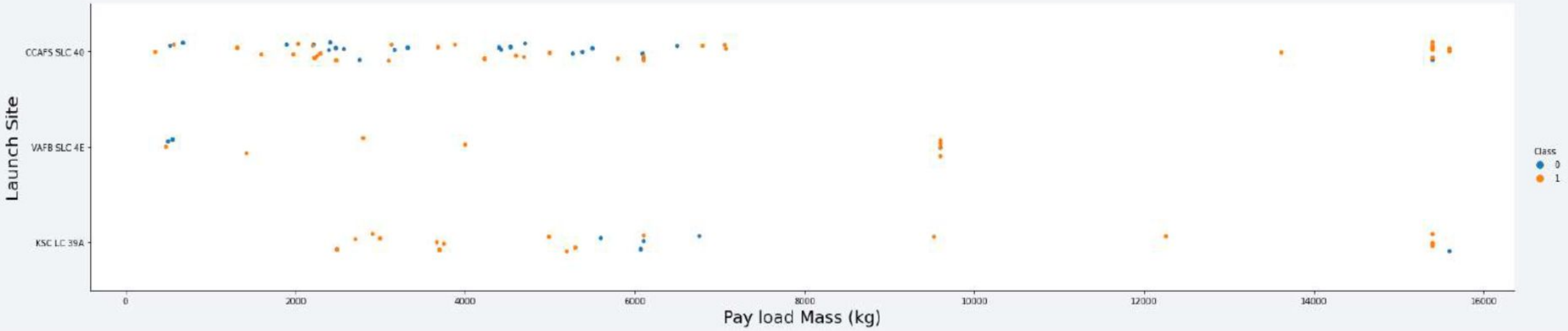


Flight Number vs. Launch Site



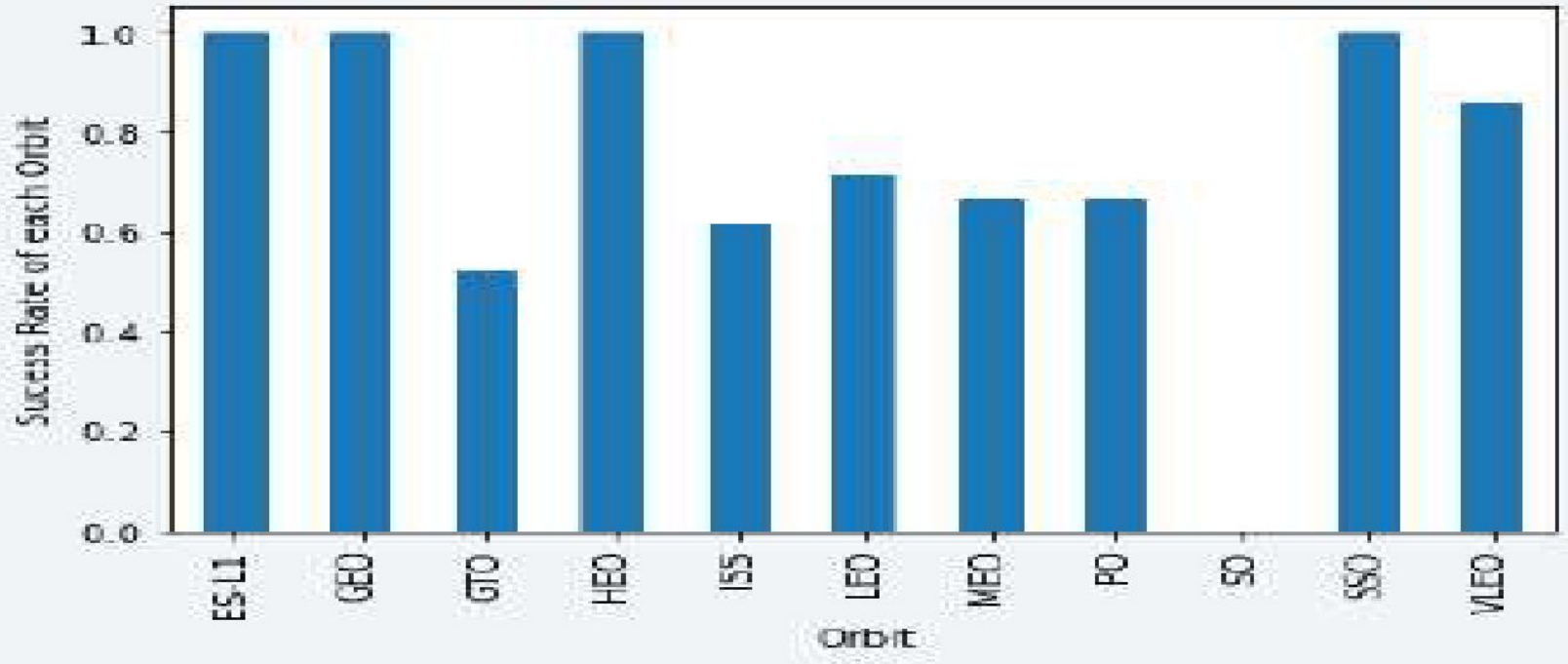
The more trials the better result

Payload vs. Launch Site



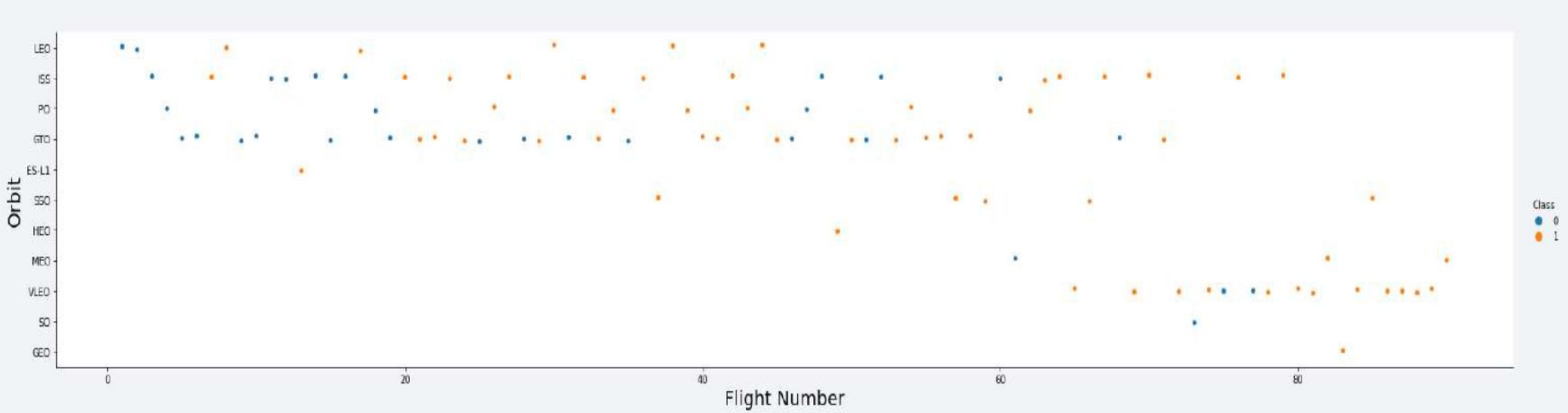
Two launch sites are able to service a heavy payload.

Success Rate vs. Orbit Type



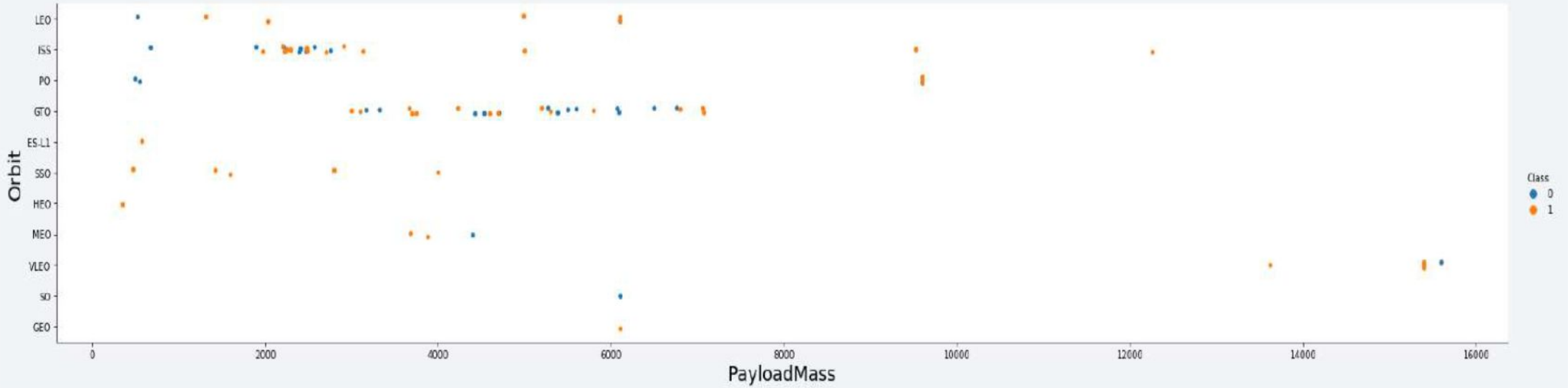
Four out of all orbit types considered in the project have success rate of 1 (100% success)

Flight Number vs. Orbit Type



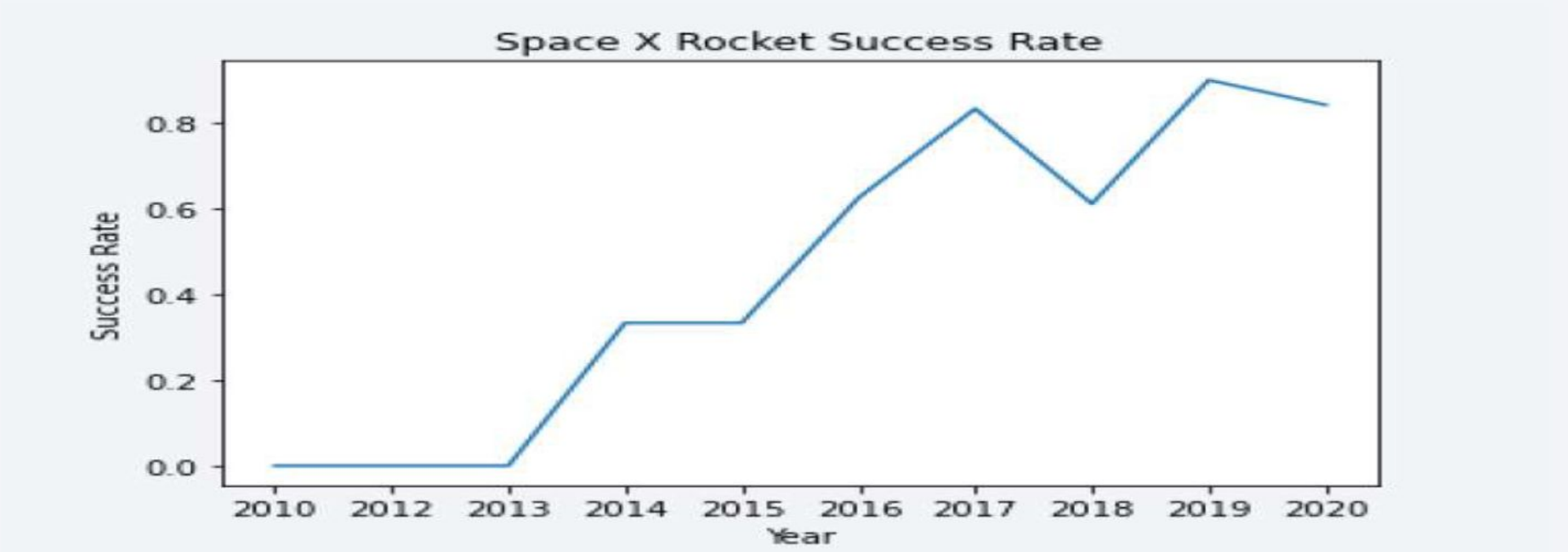
First flights were not satisfactorily successful. With increase of flight numbers we observe less blue points (unsuccessfull launnches). Also increase of flight numbers was accompanied by diversification of targeted orbits.

Payload vs. Orbit Type



It is visible that large payload launches were more succesful and the largest were sent to only one orbit. Small payloads were launched rarely. Worth mentioning is a fact that some orbits have only successful launches

Launch Success Yearly Trend



A clear trend is visible. SpaceX seems to gain experience year by year which result in increasing success rate.

All Launch Site Names

```
SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

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

Launch Site Names Begin with 'CCA'

```
SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
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
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

Result was reachad by filtering by WHERE + LIKE and it was limited to 5 by LIMIT.

Total Payload Mass

```
SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

SUM("PAYLOAD_MASS__KG_")
45596

The above formula calculates total payload mass ordered by NASA

Average Payload Mass by F9 v1.1

```
SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
```

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

- The above SQL formulas return average payload mass carried by a F9 1.1 rocke version

First Successful Ground Landing Date

```
SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'
```

MIN("DATE")
01-05-2017

The above formula retuns minimum value from a DATE under contition that landing was successful . That is how we receive a date of the first successful landing.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (drone ship)' \
AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

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

The above SQL formula returns number of drones with had payloads between 4000 and 6000 that landed successfuly.

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, \
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

SUCCESS	FAILURE
100	1

Outcomes are filtered, than added and displayed in relevant columns - SUCCESS and FAILURE.

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 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

The above formula filters payloads and returns only heaviest. Finaly names of heaviest payloads are returned.

2015 Launch Records

```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING  OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

The above code returns failed landings on a drone ship that took place in 2015.

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

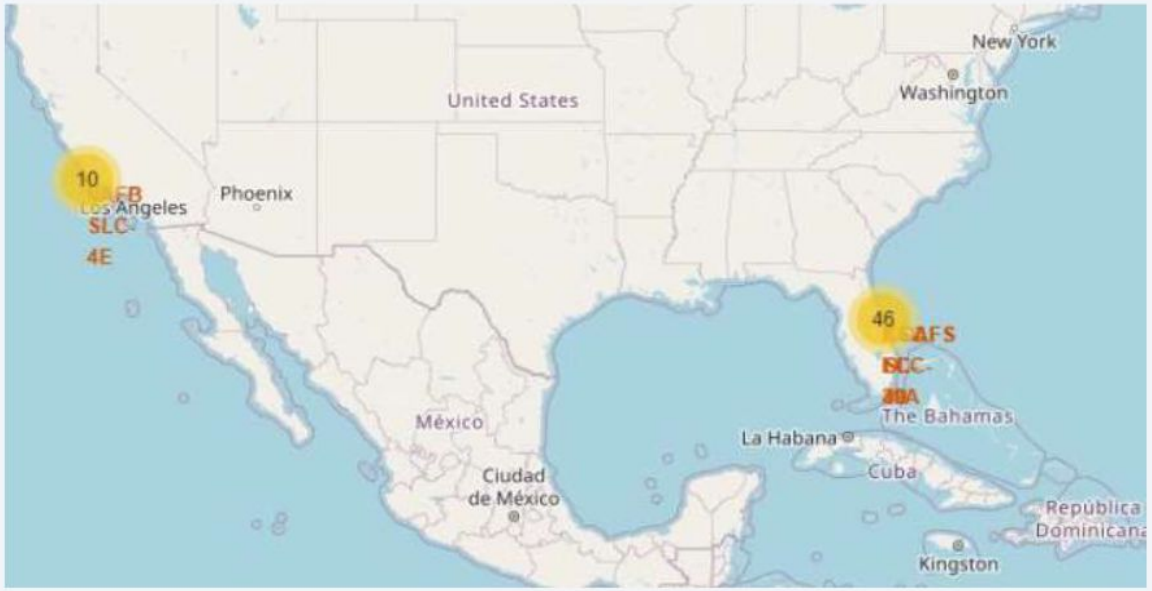
```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING  OUTCOME") DESC ;
```

Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

The above query groups entries by landing outcome. Data is arranged in descending order. Count of landing orders of successful launches and landing is returned.


Section 3
Launch Sites
Proximities Analysis

Launch Sites Location

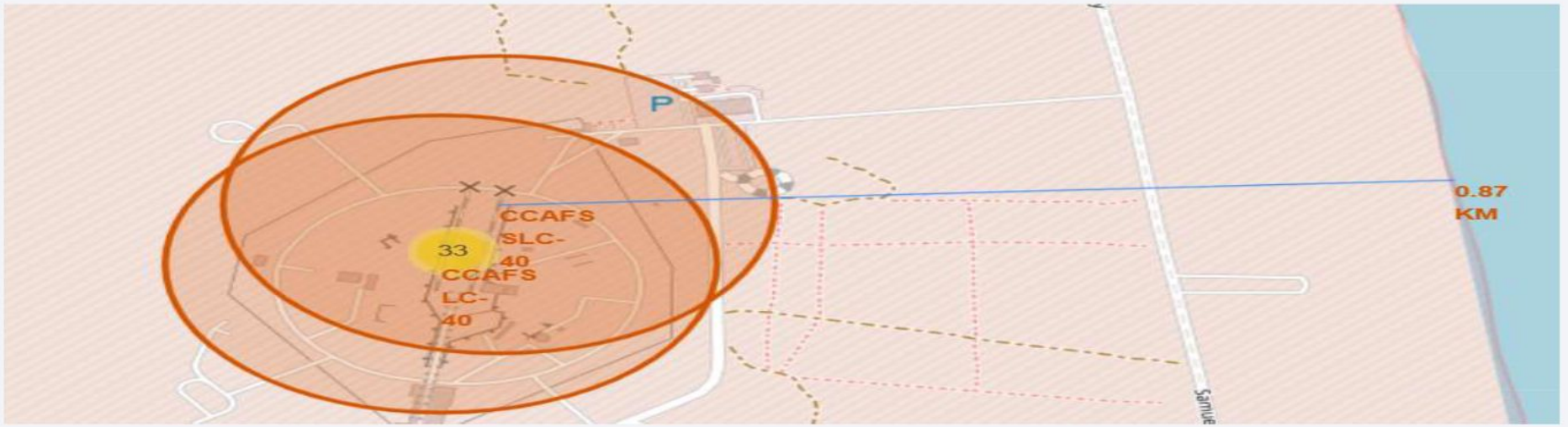


SpaceX are located at ocean shores.

Succes/Fail per launch site

In each launch site sucessful mission was pointed by a green marker. Mission that was a failure received a red marker.

Launch Sites proximities

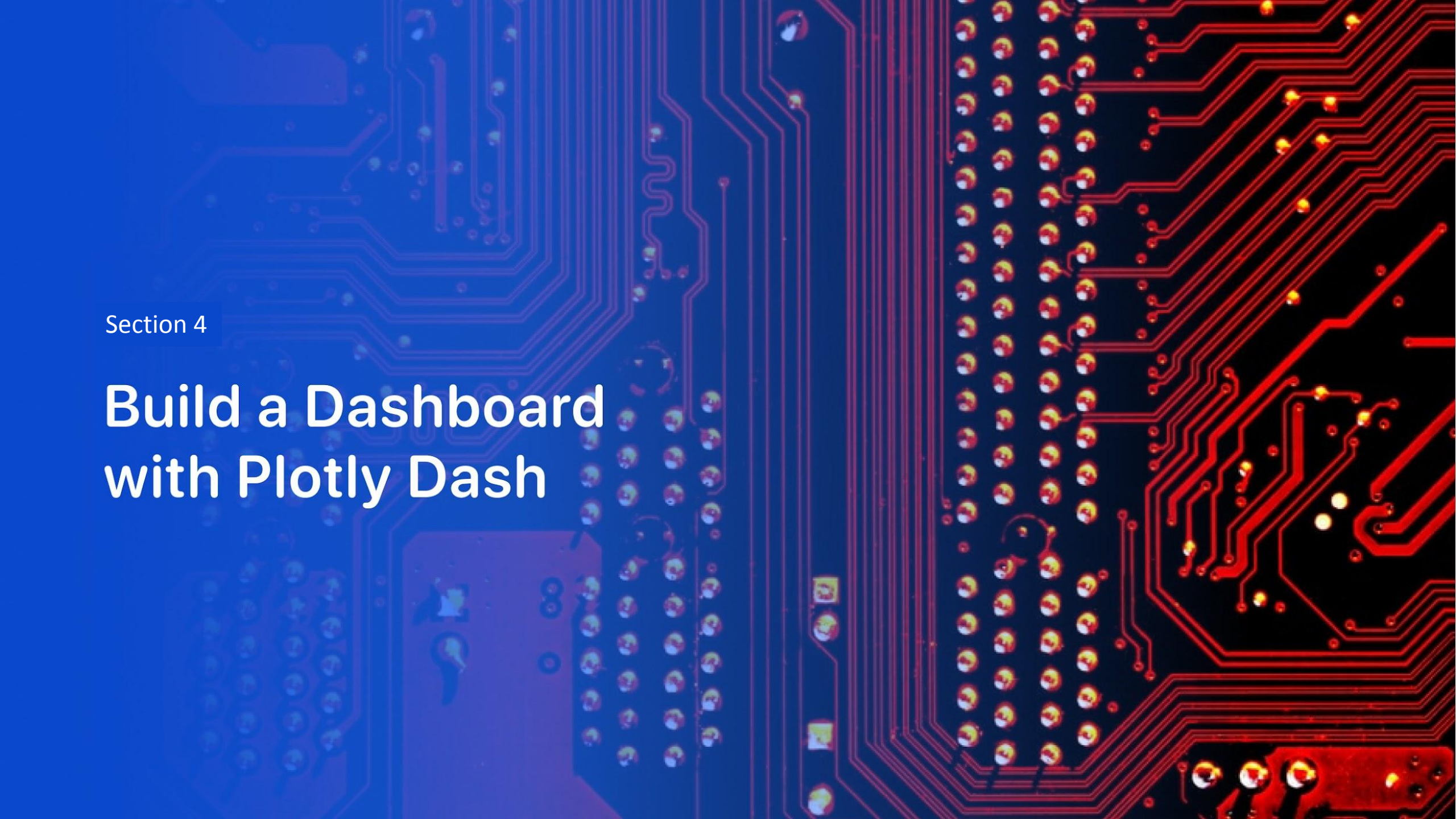


Each launch site needs to be close to important proximities, in particular to:

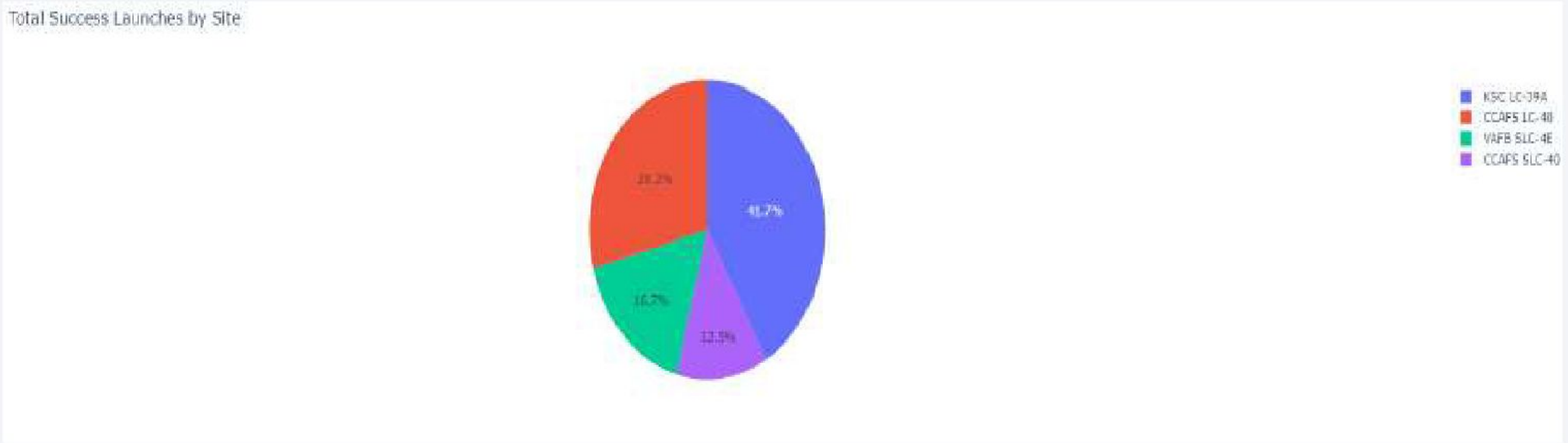
railway

highway

coastline

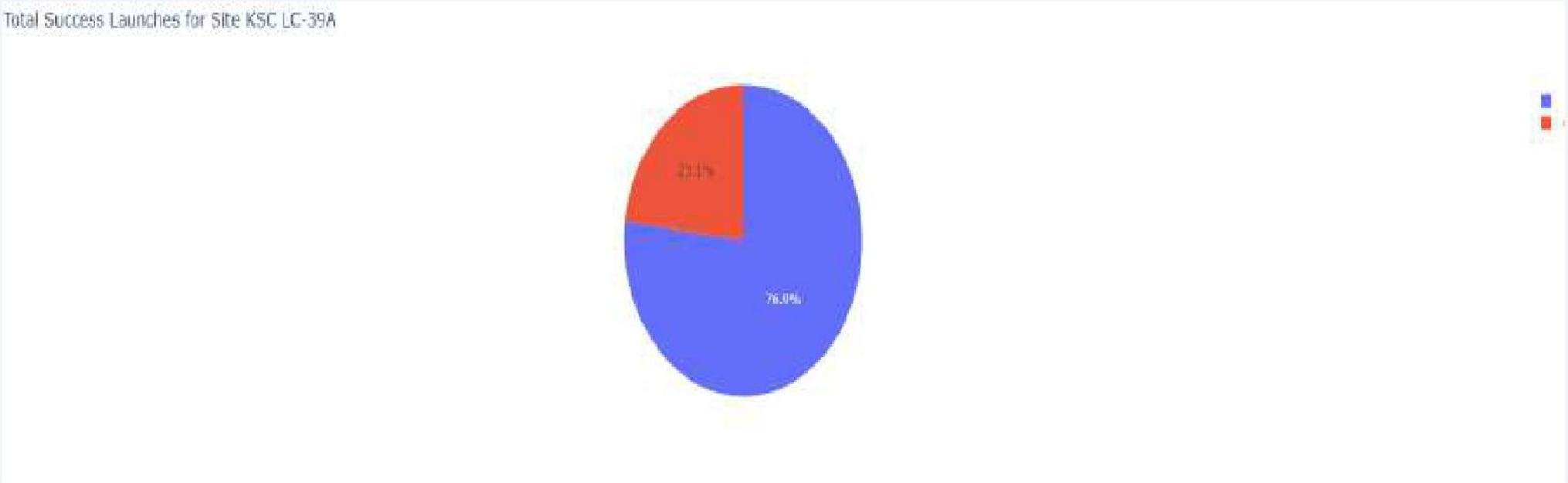

Section 4
Build a Dashboard with Plotly Dash

Site effectiveness



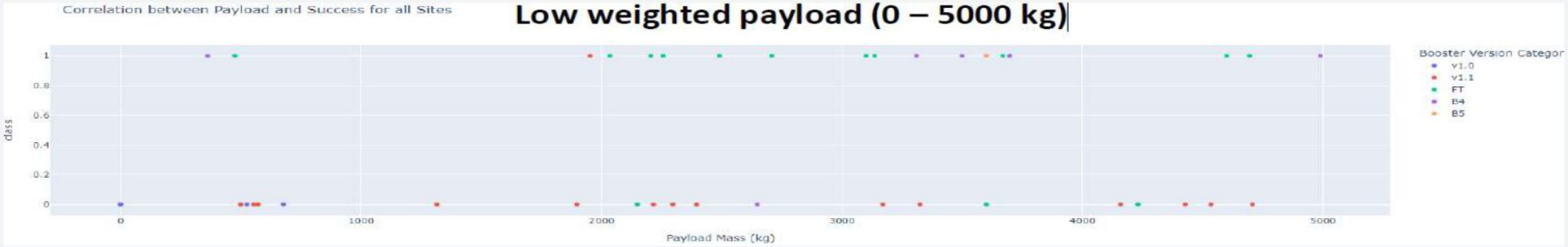
It is clear that KSC LC-39A launch site is most effective.

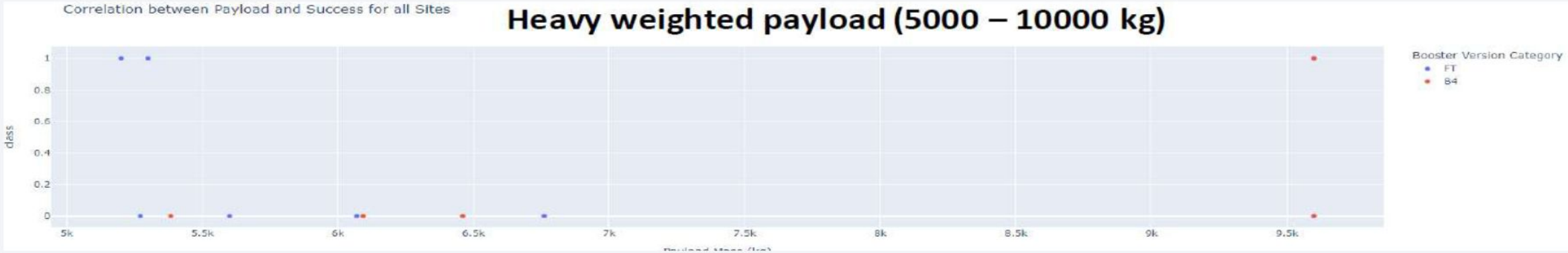
KSC LC-39A launch site effectiveness



Almost 77% missions were succesfull.

Low vs high payload effectiveness



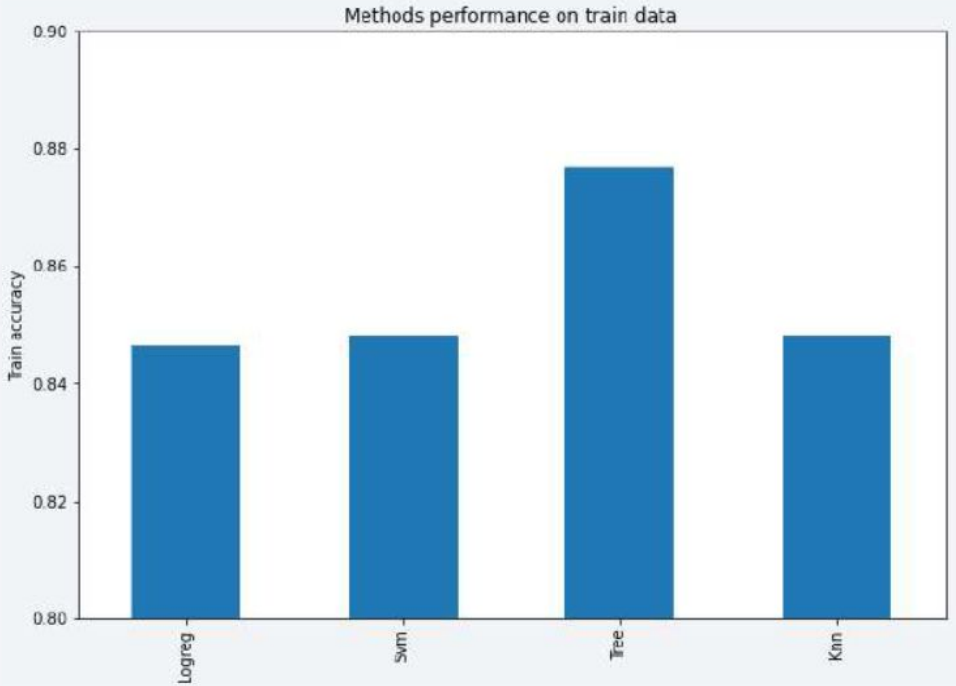


Only two launch sites were used for heaviest loads. Launching lighter payloads has more succeses.

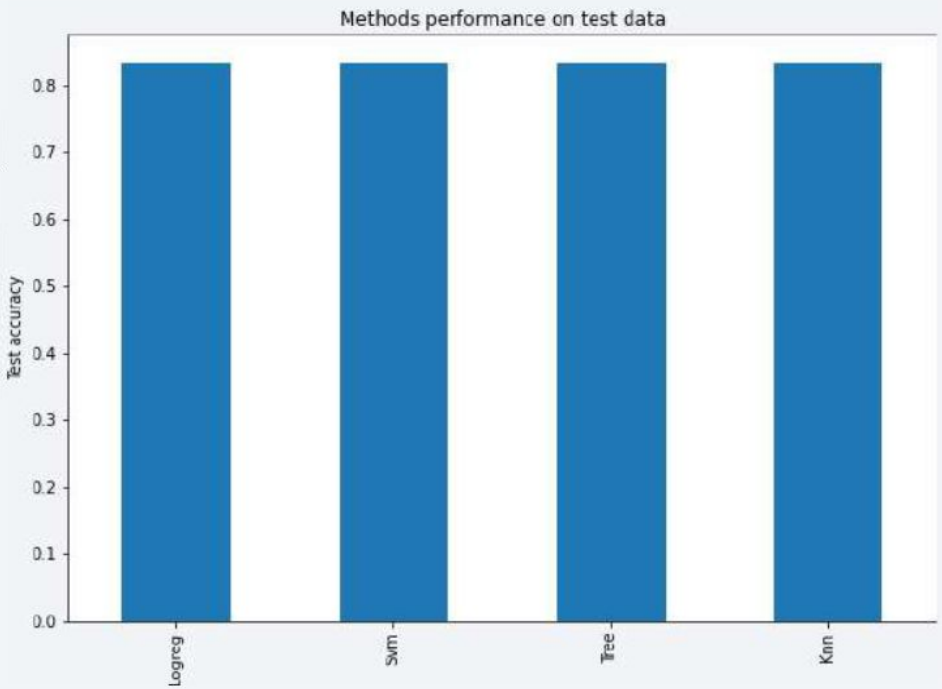
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Accuracy Train	Accuracy Test
0.876786	0.833333
0.848214	0.833333
0.848214	0.833333
0.846429	0.833333



The best choice of a model for our project would be Decission Tree. It performed best on train data. Accuracy test were the same for all implemented models.

Confusion Matrix

False positives are most problematic for each model.

Conclusions

- It is clearly visible that with time success/fail ratio of SpaceX launches increased
- With data we were provided it is impossible to justify superiority of KSC LC-39A launch site.
- It is best to launch a rocket directing GEO, SSO, ES-L1 or HEO orbit.
- Some orbits seem to be more suitable for heavier payloads.
- Due to best train accuracy we recomend to use a Deciision Tree clasifier.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!