Winning Space Race with Data Science

Thomas Johnson

10th June 22

Outline

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

Executive Summary

- **Summary of Methodologies**
  - Data Collection through API
  - Data collection through Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- **Summary of Results**
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predicative Analytics result from Machine Learning Lab

Introduction

SpaceX is a space program which is quickly revolutionizing space travel. It's Falcon 9 rocket launches cost the company around 62 million dollars – roughly 100 million dollars less than other space programs. SpaceX can achieve this due to its re-usable rockets. This huge saving is what allows SpaceX to make extraordinary progress.

The objective of this project is to evaluate the viability of a new company SpaceY competing with SpaceX

Questions to answer:

- What is the best way to estimate the total cost for launches?
- Where is the best place to launch?

Section 1

Methodology

Methodology

- Data collection methodology:
  - Data from Space X was obtained from 2 sources:
    - Space X API (https://api.spacexdata.com/v4/rockets/)
    - WebScraping (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches)
  - Data Wrangling
    - Data was processed using one-hot encoding for categorical features
  - Data Wrangling
    - Data was processed using one-hot encoding for categorical features
  - Exploratory data analysis (EDA) using visualizations and SQL
  - Interactive visual analytics using Folium and Plotly Dash
  - Predictive machine learning models
    - Building, tuning and evaluating classification models

Data Collection

Data collection is the process of collecting, measuring and analysing different types of information using a set of standard validated techniques. The main objective of data collection is to gather information-rich and reliable data and analyse them to aid in business decisions

Techniques used:

- REST API – GET requests were initially used and the response was decoded as JSON and turned into a pandas dataframe using json_normalize(). Data was then cleaned

- Web scraping from Wikipedia – BeautifulSoup was used to extract launch records and information was parsed and converted into a pandas dataframe

Data Collection – SpaceX API

GET request for rocket launch using REST API

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

response = requests.get(spacex_url)
```

json_normalize used to convert json to dataframe

```
data = response.json()
data = pd.json_normalize(data)
```

Data cleaning and missing values

```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

https://github.com/multiyumz/capstone_project/blob/main/data_science_api.ipynb

Data Collection - Scraping

Request the falcon 9 Launch wiki page from url

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

Create BeautifulSoup from the HTML response

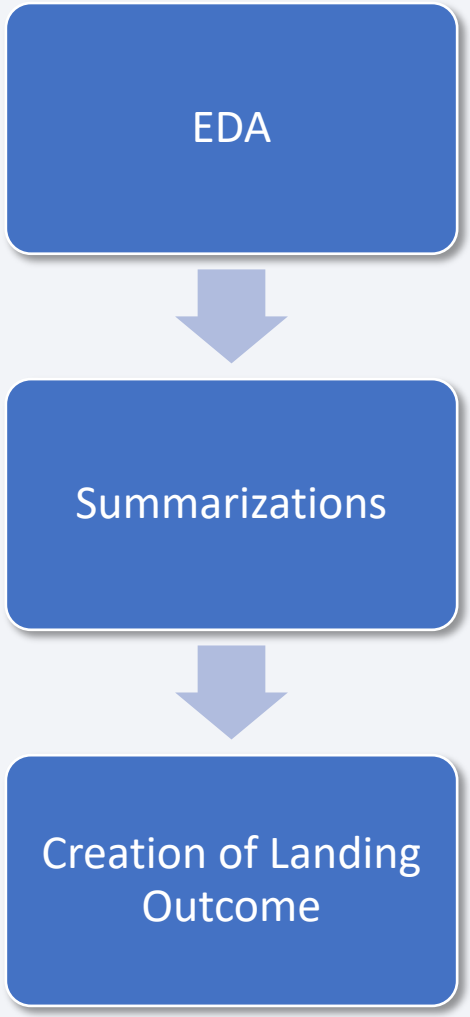
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
bs = BeautifulSoup(response.text, "html.parser")
```

Extract column/variable names from the HTML header

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(bs.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

https://github.com/multiyumz/capstone_project/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling



Initially some Exploratory Data Analysis (EDA) was performed on the data

Summarizations of each launch site, occurrences of each orbit and mission success/failure per orbit type were calculated

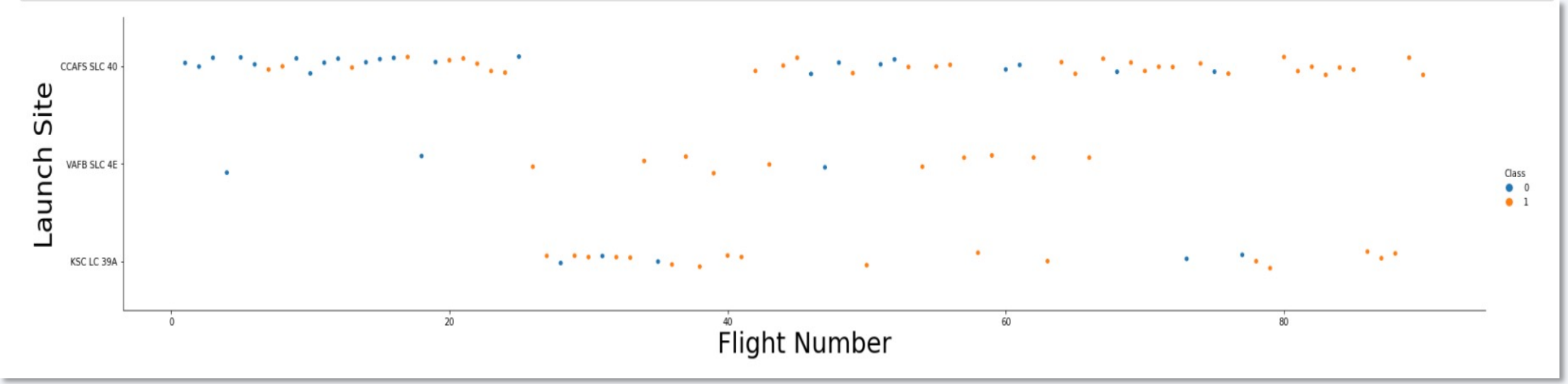
Finally, a landing outcome variable was created from the Outcome column. This will make it easier to perform further analysis, visualizations and machine learning techniques

https://github.com/multiyumz/capstone_project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

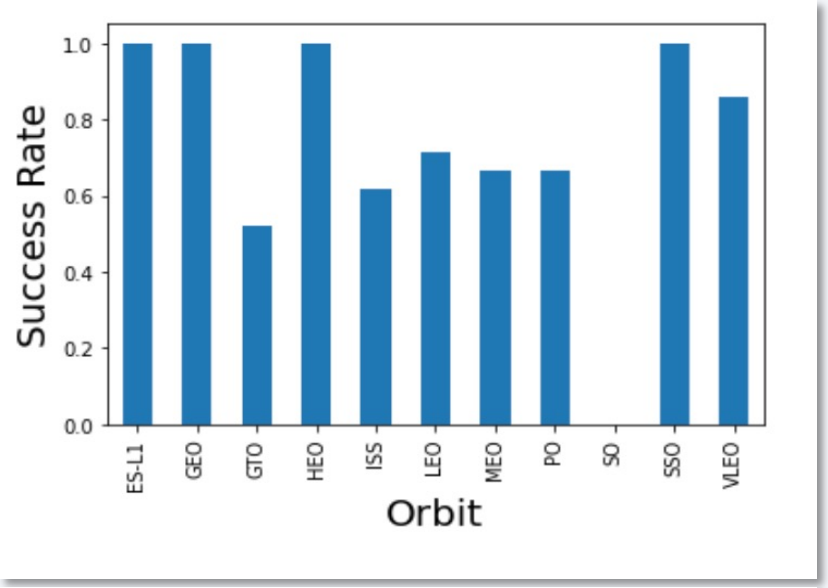
Scatterplots were initially used to visualize the relationships between pairs of features:

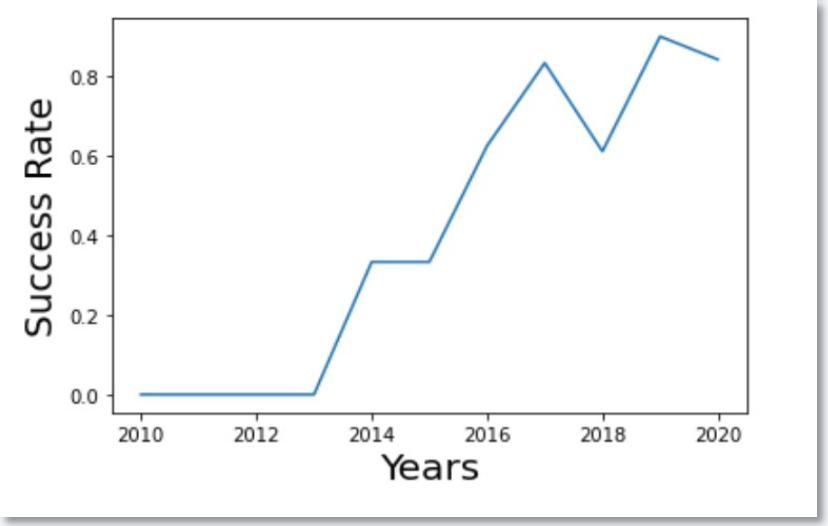
- Payload and Flight Number
- Flight Number and Launch Site
- Payload and Launch Site
- Flight Number and Orbit Type
- Payload and Orbit Type



https://github.com/multiyumz/capstone_project/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with Data Visualization





After getting an idea of relationships between variables via the use of scatterplots we can adopt further visualization tools such as bar and line graphs

Bar graphs were used to determine which orbits have the highest probability success rate

Line graphs were then used to show the trend of successful missions over a ten-year period

https://github.com/multiyumz/capstone_project/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

The following SQL queries were performed:

- Names of the distinct launch sites
- Top 5 launch sites with a name beginning with the string ‘CCA’
- Total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 V1.1
- Date when first successful landing outcome in ground pad was achieved
- Names of the boosters which had a successful in drone ship landing and whose payload mass is between 4000 and 6000 kg
- Total number of successful and failed missions
- Names of the boosters which have carried the maximum payload mass
- Failed drone ship landings, their booster versions, and launch site names in the year 2015
- Successful landing outcomes between 2010-06-04 and 2017-03-20 ranked in descending order

https://github.com/multiyumz/capstone_project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Markers, circles, lines and marker clusters were used with Folium Maps

- Makers indicate points such as launch sites
- Circles indicate highlighted areas around specific coordinates
- Marker clusters indicate groups of events in each coordinate. These were used to simplify the map for more than one point with sharing the same coordinate
- PolyLines were used to show the distance between two coordinates such as launch sites and the coast as well as the distance to the nearest train line, highway and city

https://github.com/multiyumz/capstone_project/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- The following graphs were designed with plotly dash:

  - Pie charts showing percentage of launches per site
  - Scatter graph showing the relationship between mission outcome and Payload mass (kg) for different booster versions

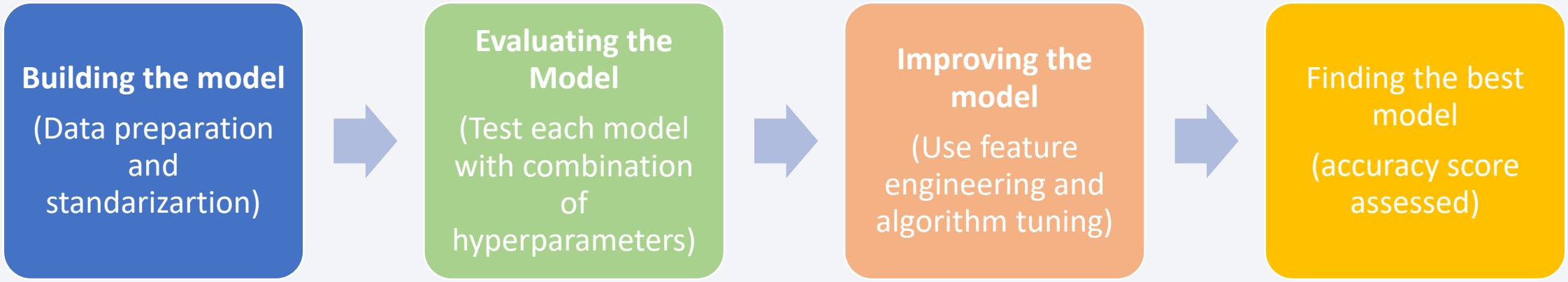
This combination allowed us to quickly analyze the relationship between payload and launch site helps us to quickly identify where is best to launch in future missions

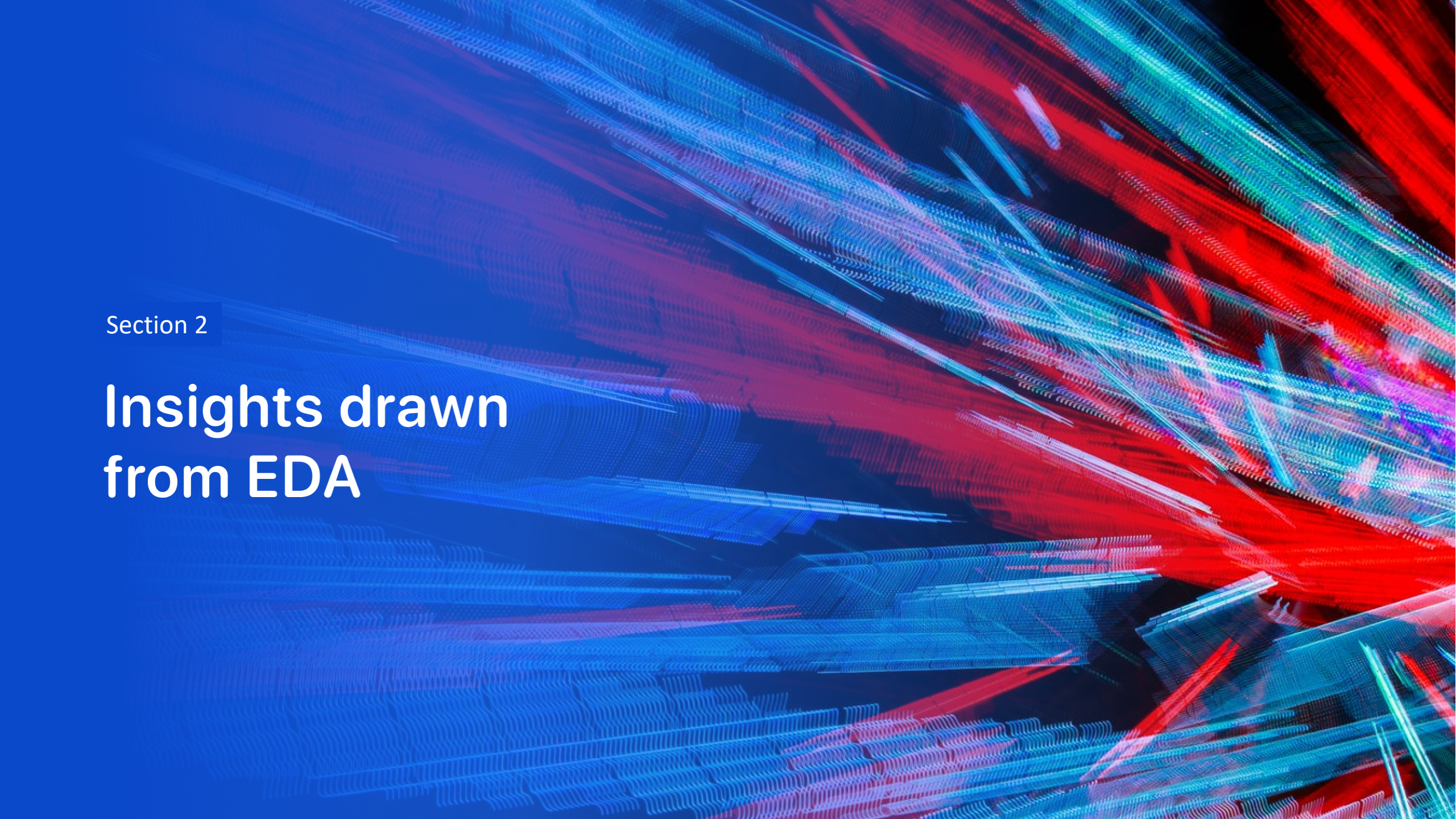
https://github.com/multiyumz/capstone_project/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

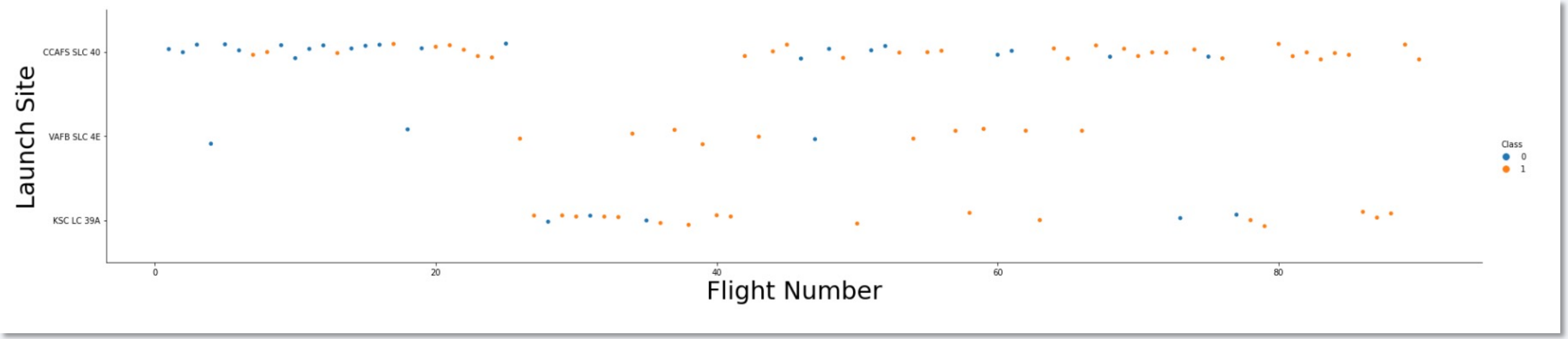
Four classification models were compared:

- Logistic Regression
- Support Vector Machine
- Decision Tree
- K Nearest Neighbours.



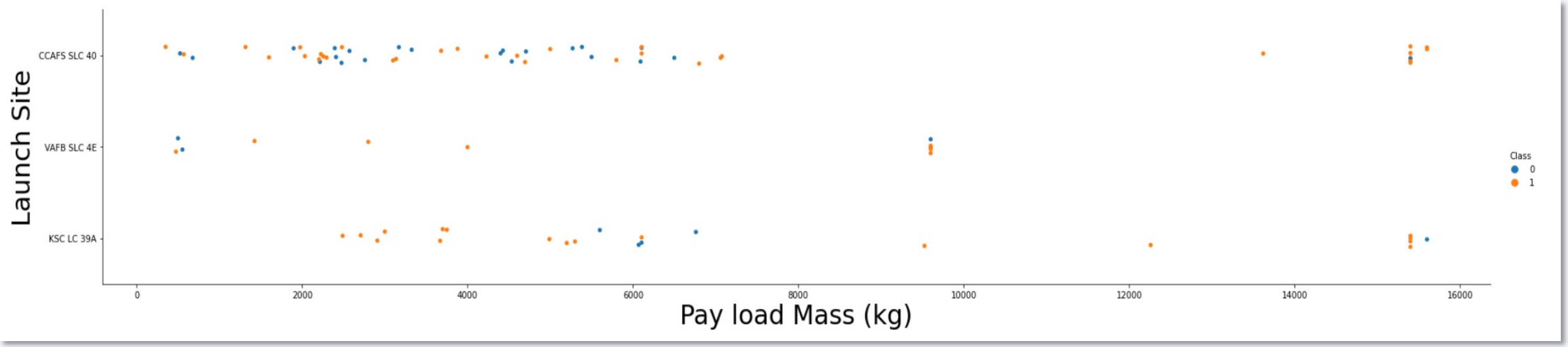

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



- According to the plot we can see that the best launch site is CCAF5 SLC 40, where most of the launches were successful
- VAFB SLC 4E is the next best with the second most successful launches
- Over time we can see the general success rate improved

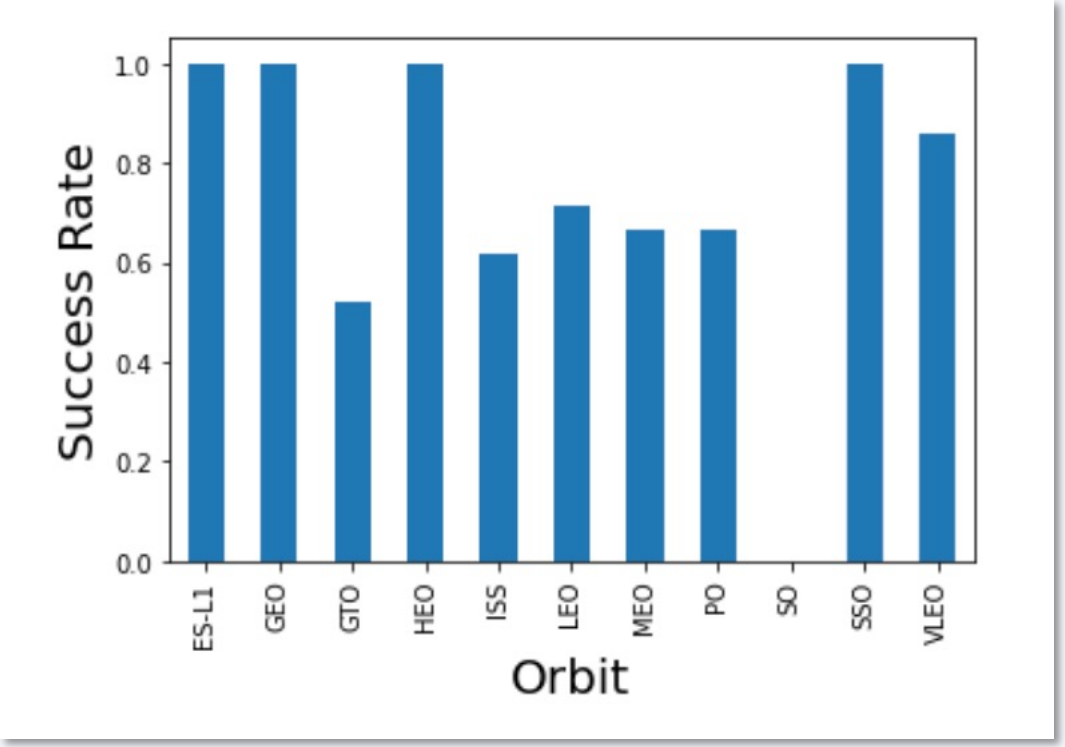
Payload vs. Launch Site



- Scatterplot shows that once payload mass is increased over 9000kg the probability of success is increased dramatically

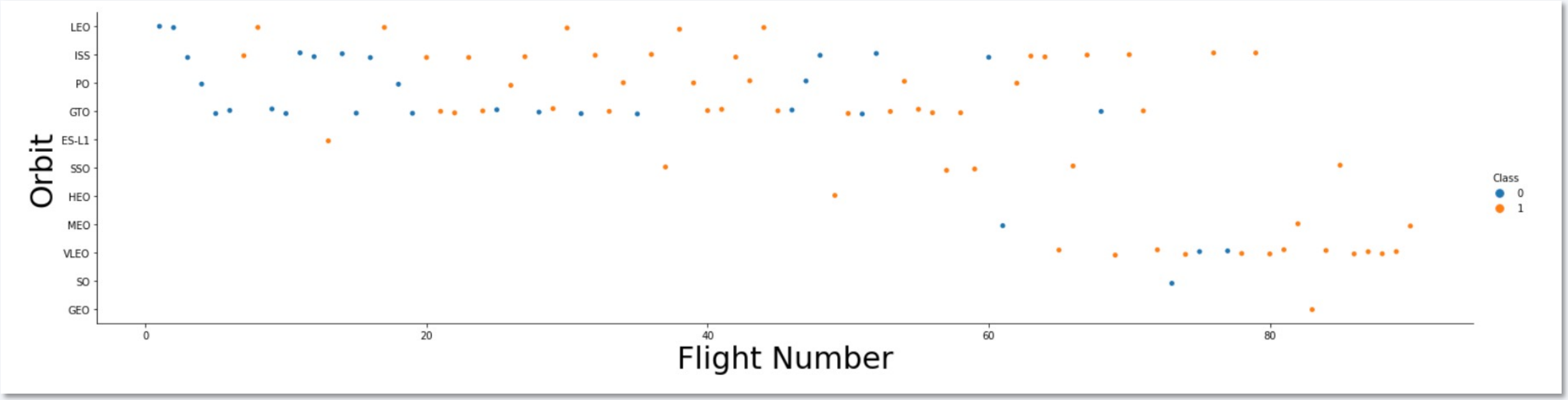
- The plot also shows that payloads over 1200kg only seem possible at CCAFS SLC 40 and KSC LC 39A launch sites

Success Rate vs. Orbit Type



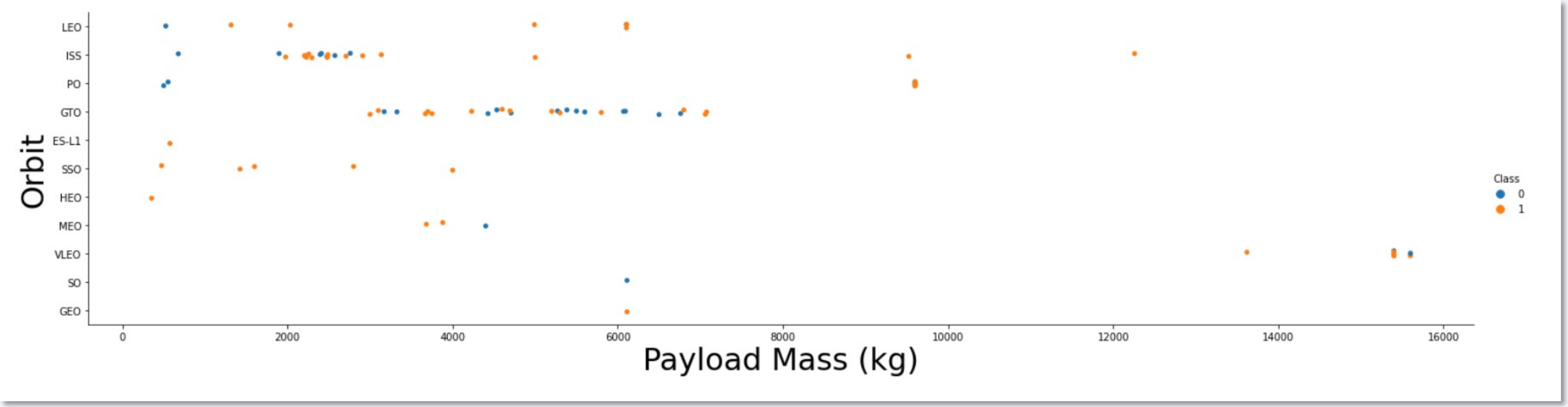
- The bar chart illustrates the potential influence on each orbit has on the landing outcome
- Some orbits (ES-L1, GEO, HEO and SSO) have 100% success rate, while SO has 0% success rate
- However, closer analysis shows that some orbits only had one occurrence. For example, GEO had one successful launch
- Therefore, more analysis is needed before we can draw any conclusions

Flight Number vs. Orbit Type



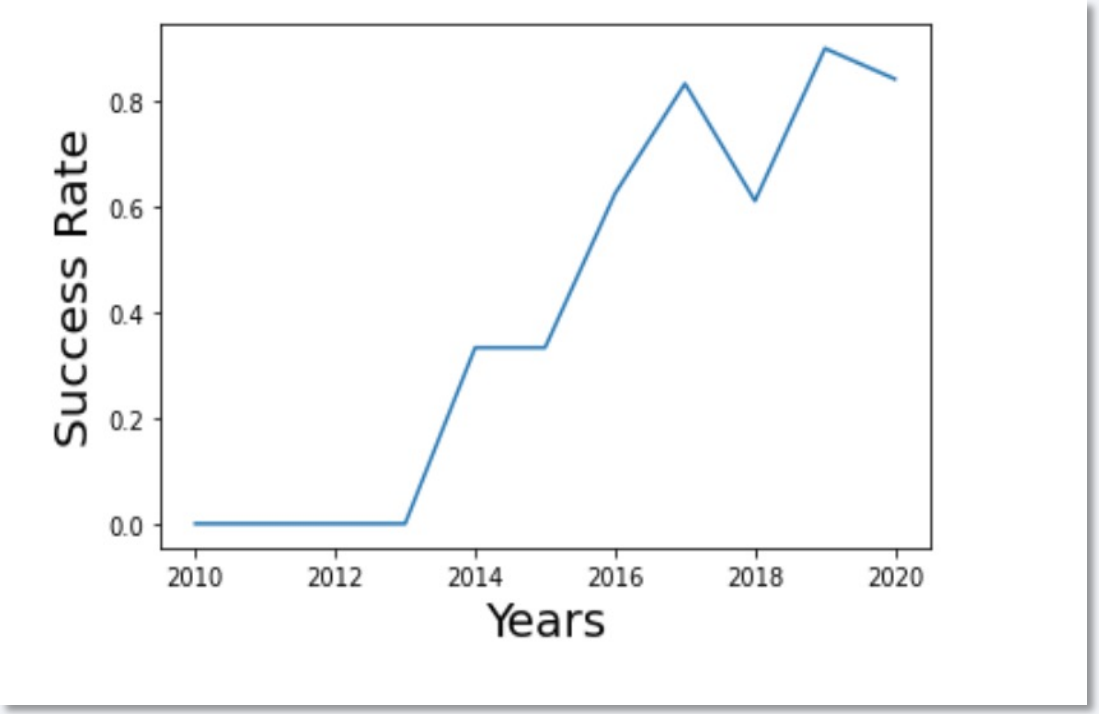
- Scatterplot shows that success rate improves over time for all orbits
- Generally, the success rate is positively correlated with the number of launches
- VLEO should be examined more closely going forward due to the increased number of recent successful flights

Payload vs. Orbit Type



- Heavier payloads have a positive impact on the success of LEO, ISS and PO orbits, with a negative impact for MEP and VLEO orbits
- There is no correlation between payload mass and success rate for GTO orbit
- There are too few launches to SO, GEO and HEO to draw any conclusions

Launch Success Yearly Trend



- There was an increasing trend in success rate from 2013 to 2020
- From 2010 -2013 there was most likely a period of adjustments and improvements accounting for the plateau curve

All Launch Site Names

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

- According to the data there are 4 unique launch sites.
- There were gathered with SQL code using the DISTINCT keyword on the data

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
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

The above SQL query was used to gather the first 5 records with a launch site name beginning with “CAA”

Total Payload Mass

```
%sql SELECT sum(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer='NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

45596 kg was the total sum of the payload mass carried by boosters launched by NASA (CRS)

Average Payload Mass by F9 v1.1

```
%sql SELECT avg(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version='F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

2928.4 kg was the average sum payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

```
%%sql
SELECT min(DATE)
FROM SPACEXTBL
WHERE Landing_Outcome='Success (ground pad)';
```

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

min(DATE)
01/05/2017

The 1st May 2017 was the first successful ground pad landing outcome

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

Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT Mission_Outcome, COUNT(Mission_Outcome) AS TOTAL_NUMBER
FROM SPACEXTBL
GROUP BY Mission_Outcome
```

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

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

Grouping by Mission outcome we were able to count the number of 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)
```

```
 * 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

Using payload mass in a subquery we were able to get the unique booster versions carrying the maximum payload

2015 Launch Records

```
%%sql
SELECT Landing_Outcome, Booster_Version, Launch_Site
FROM SPACEXTBL
WHERE Landing_Outcome = 'Failure (drone ship)'
AND substr(Date,4,2) AND substr(Date,7,4)='2015'
```

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

Landing_Outcome	Booster_Version	Launch_Site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Failed landing outcomes in drone ships, Booster versions and Launch Site records displayed for the year 2015

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

```
%%sql
SELECT Landing_Outcome, count(Landing_Outcome) AS Total_Number
FROM SPACEXTBL
WHERE Mission_Outcome = 'Success' AND DATE BETWEEN '04-06-2010' and '20-03-2017'
GROUP BY Landing_Outcome
ORDER BY Total_Number DESC
```

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

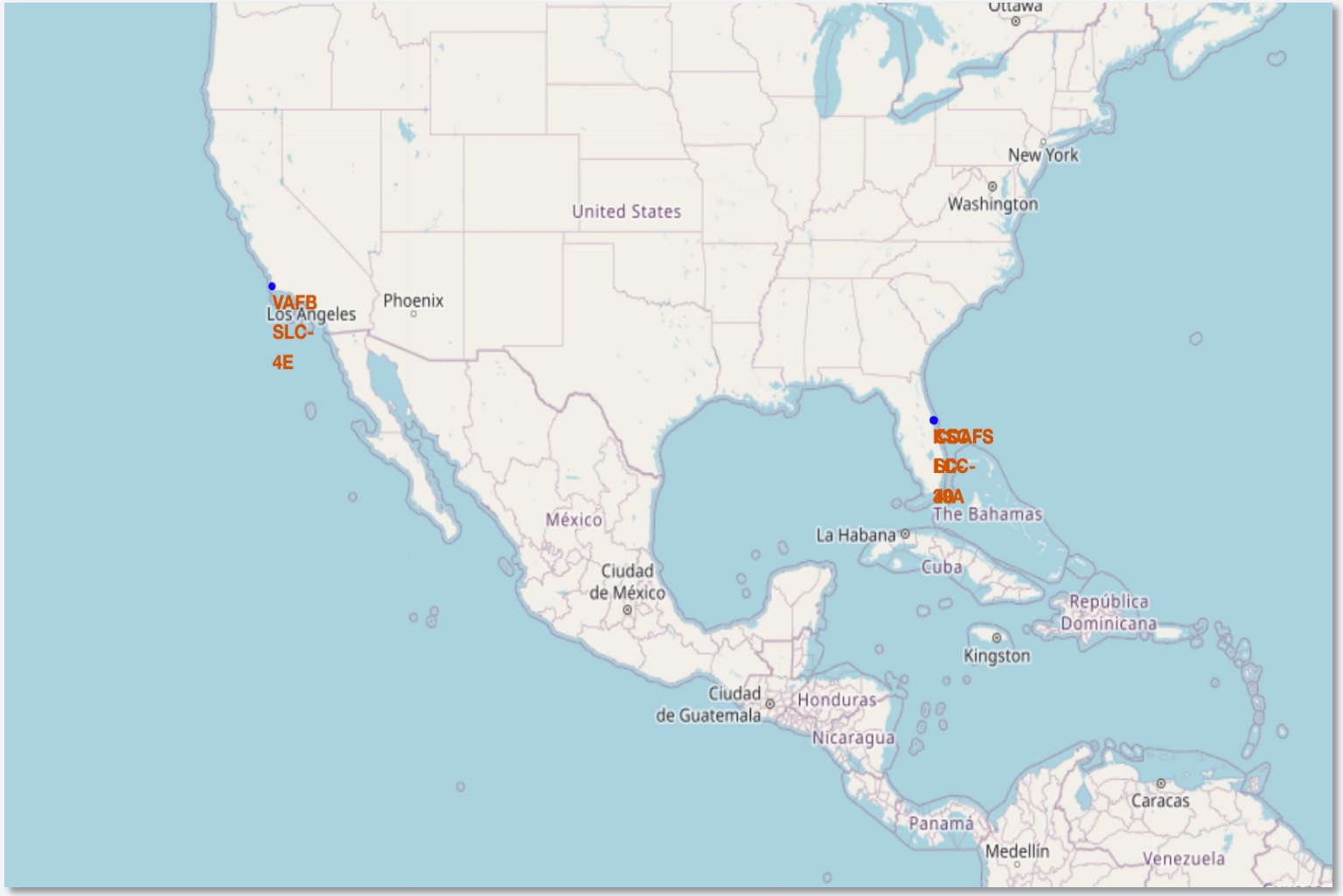
Landing_Outcome	Total_Number
Success	21
No attempt	10
Success (drone ship)	8
Success (ground pad)	5
Failure (drone ship)	5
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

We selected Landing Outcomes and their COUNT depending on WHERE the Mission Outcome wad a success. The date range BETWEEN 04-06-2010 and 20-03-2017 was selected

GROUP BY was applied to Landing Outcome and the results were displayed in descending order using DESC keyword

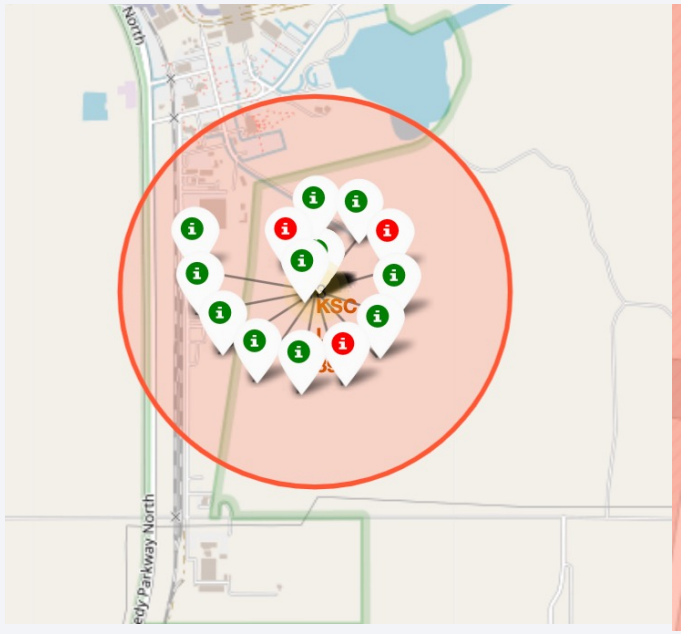

Section 3
Launch Sites
Proximities Analysis

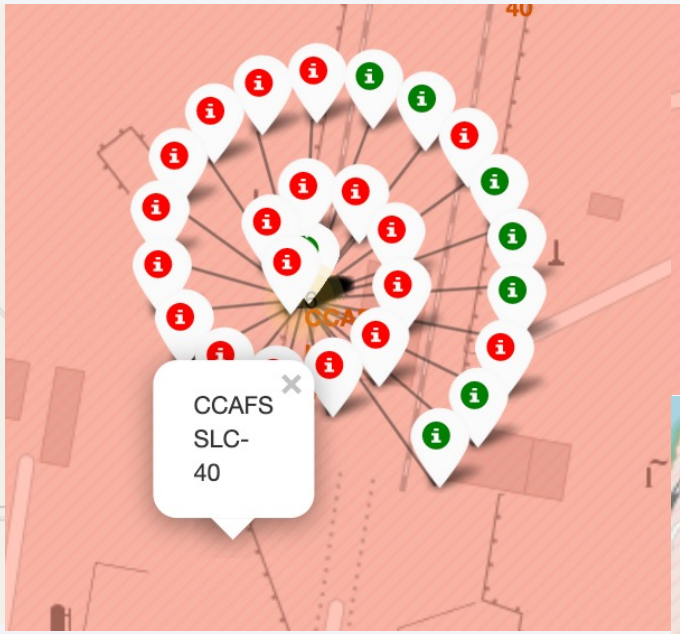
All Launch Sites

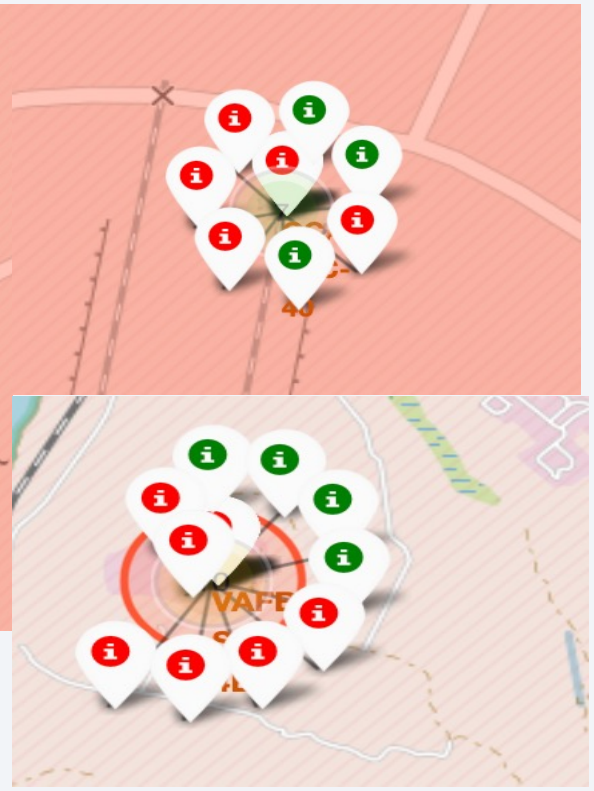


SpaceX launch sites are all located in the United States situated on the east and west coast

Markers showing Launch Outcome by site

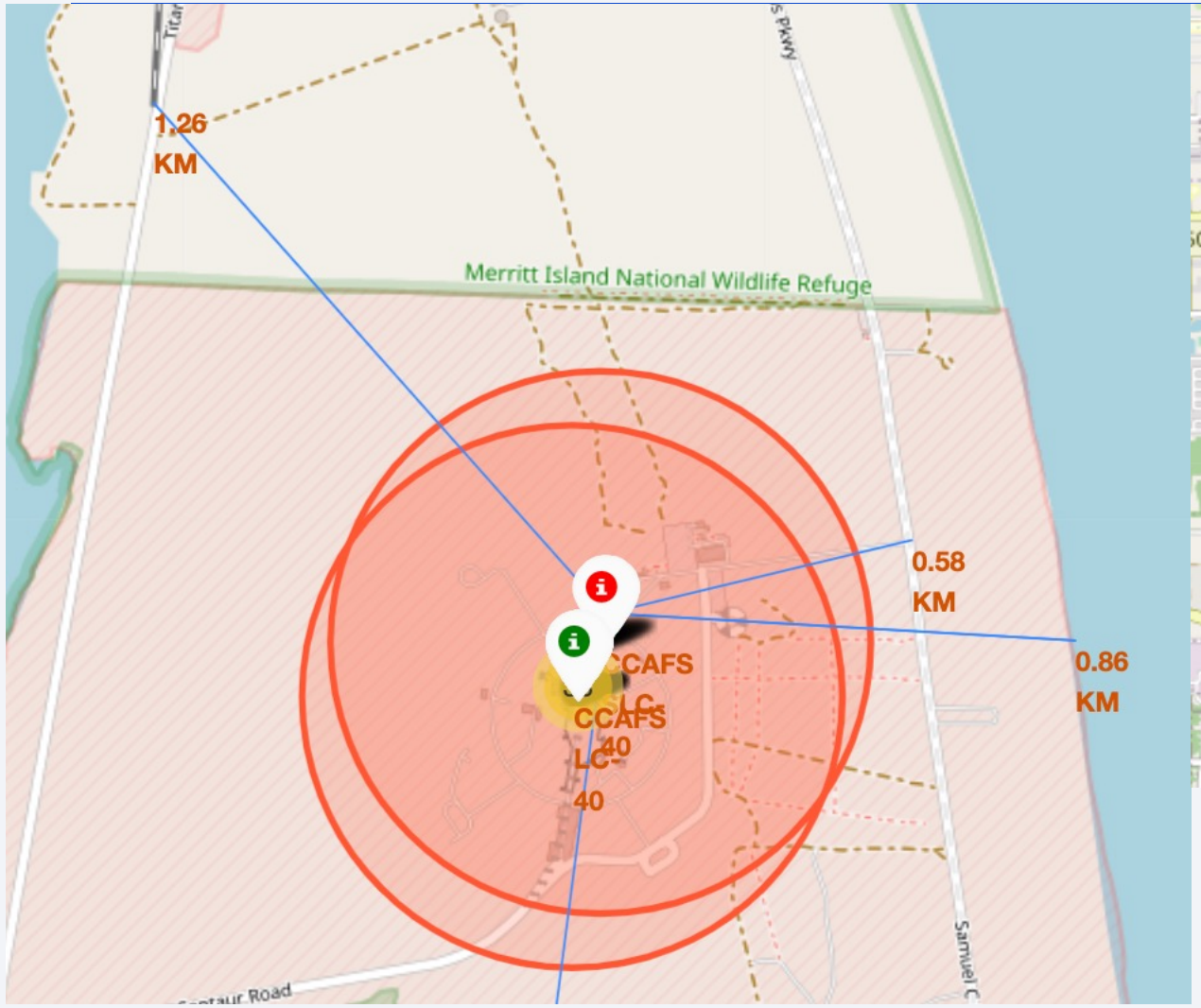


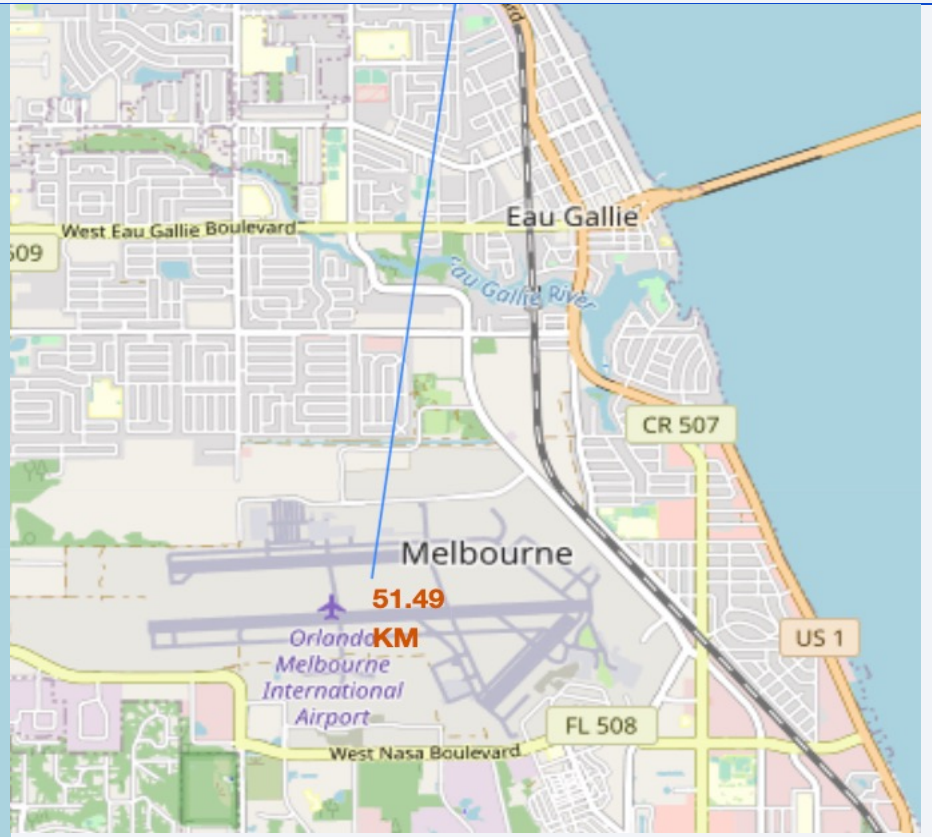




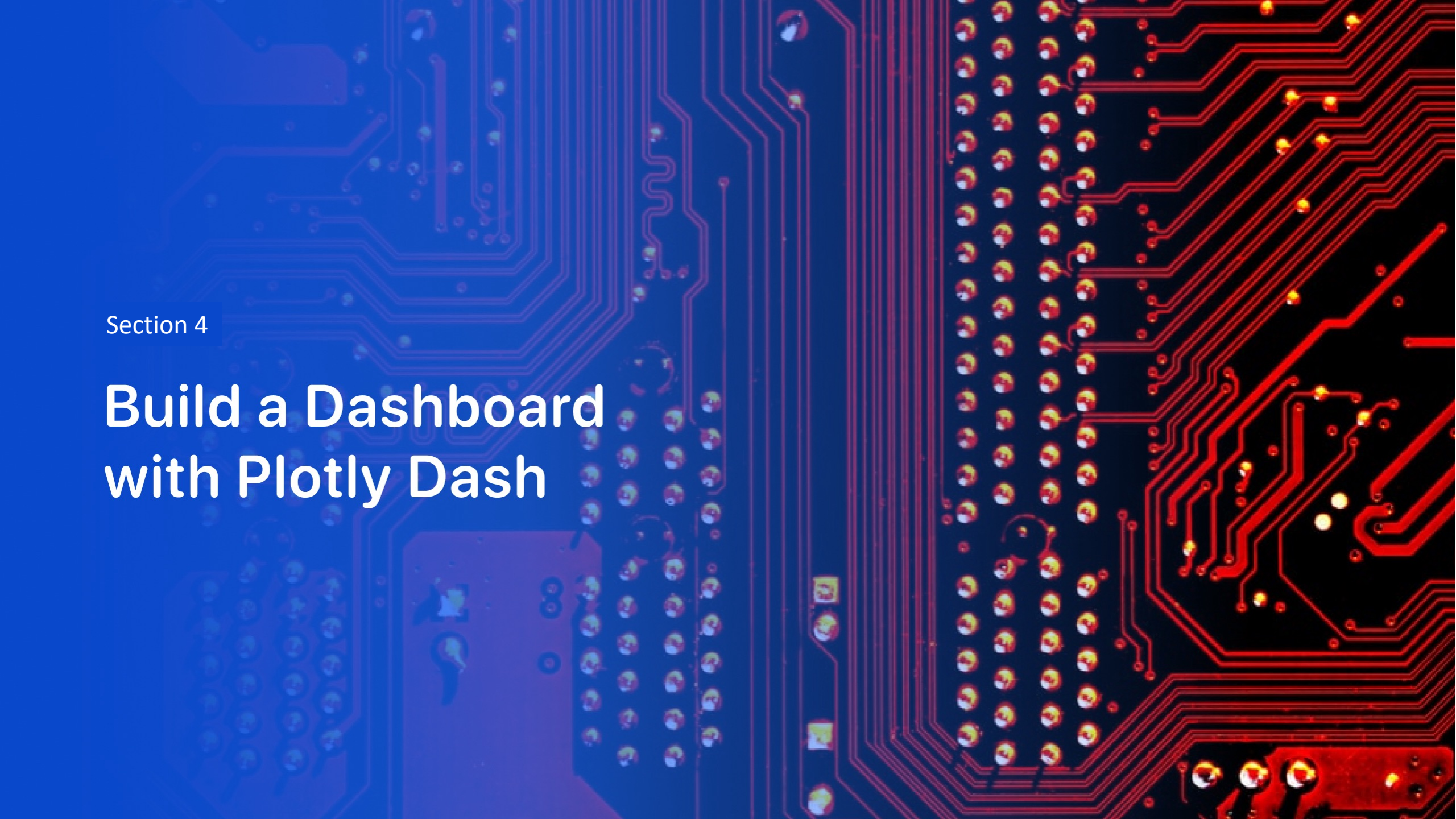
Green markers indicate successful launches while red markers indicate failed launches

Launch Site Distance to Landmarks

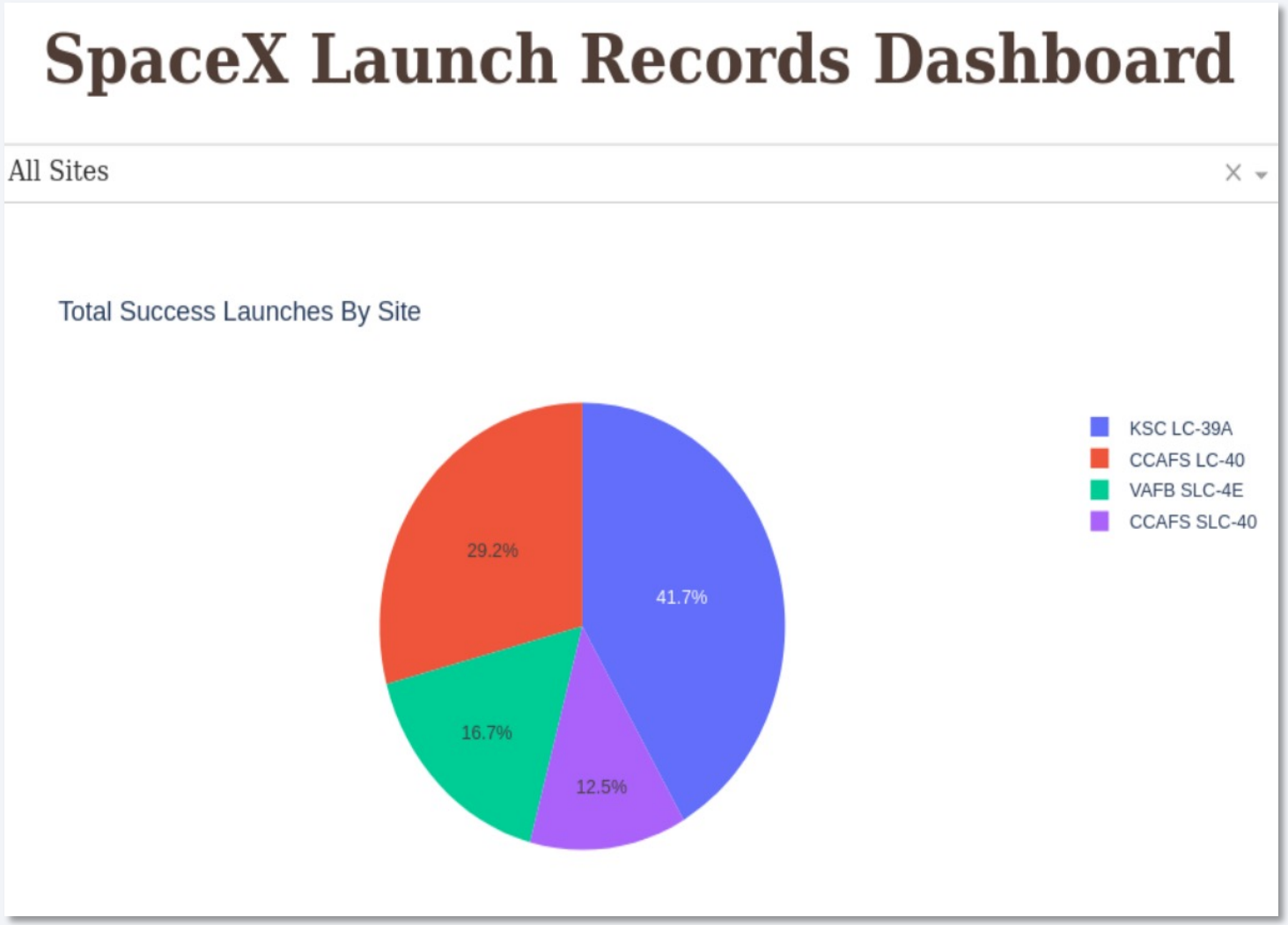




Launch sites are situated near coastlines, railways and highways for logistic purposes but far away from major cities for safety reasons

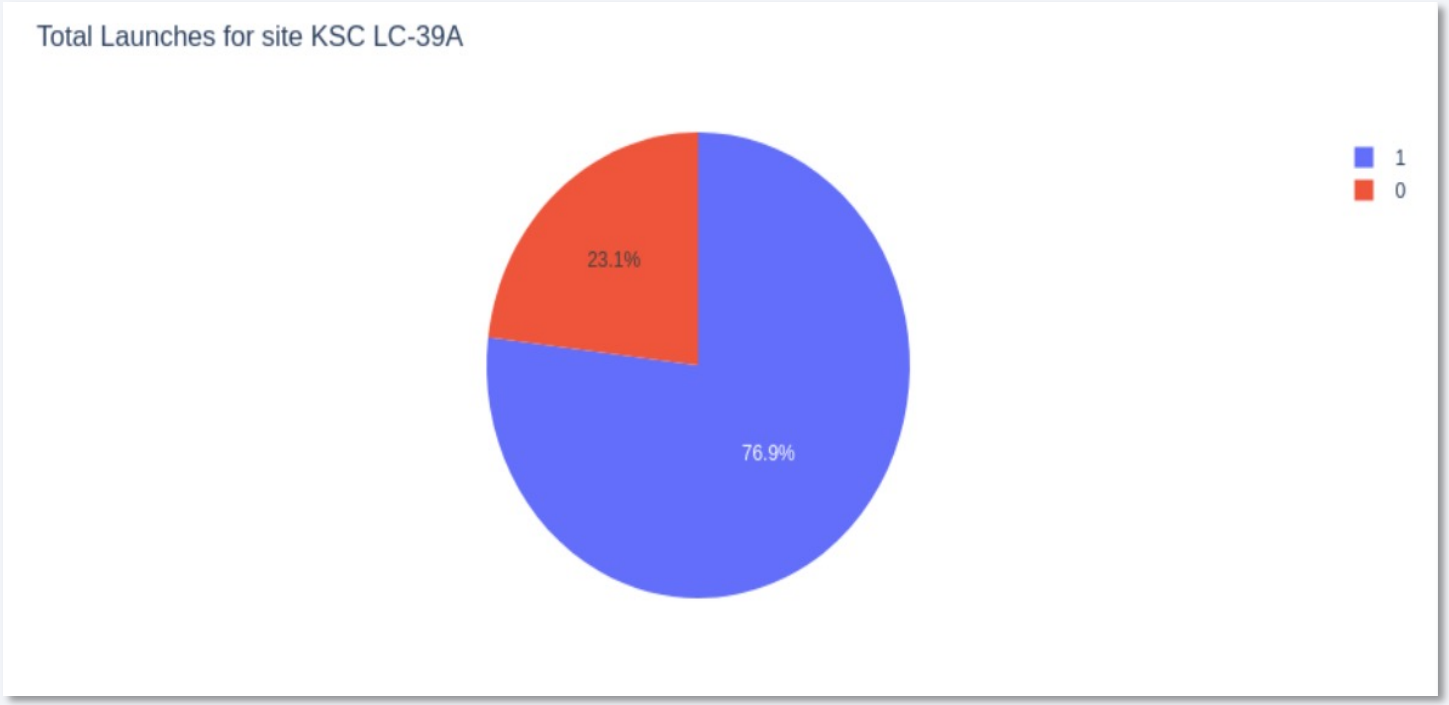

Section 4
Build a Dashboard with Plotly Dash

The success percentage by site



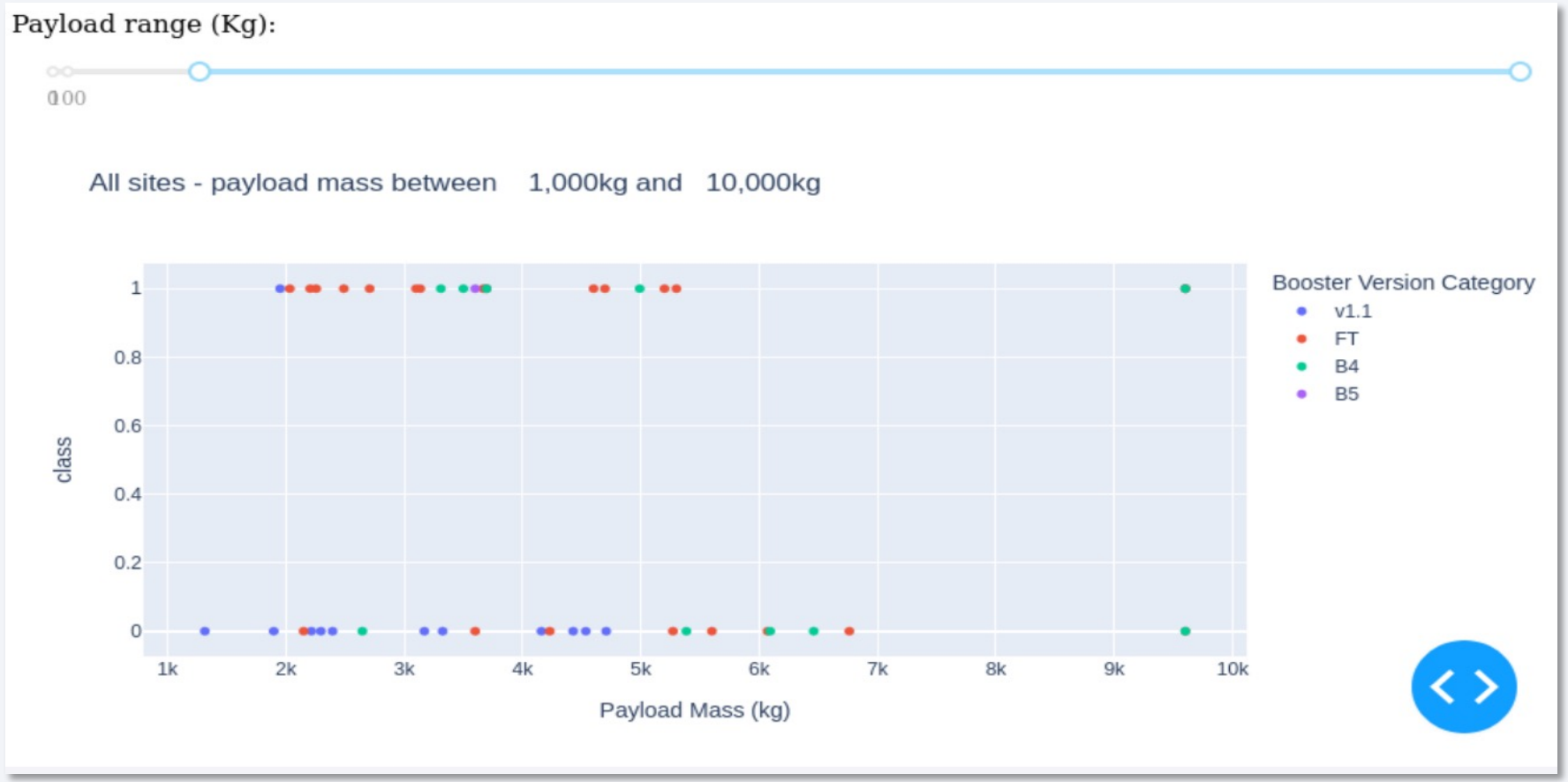
Here we see that KSC LC-39A has the most successful launches from all sites

Highest Launch success ratio: KSC LC-39A



Here we see that KSC LC-39A achieved the best success ration with a 76.9% success rate and a 23.1% failure rate

Payload vs Launch Outcome



We can see that booster version FT with a payload mass < 6000 kg was the most successful over all sites while booster version v1.1 with a payload mass < 6000 kg was the least successful

Section 5

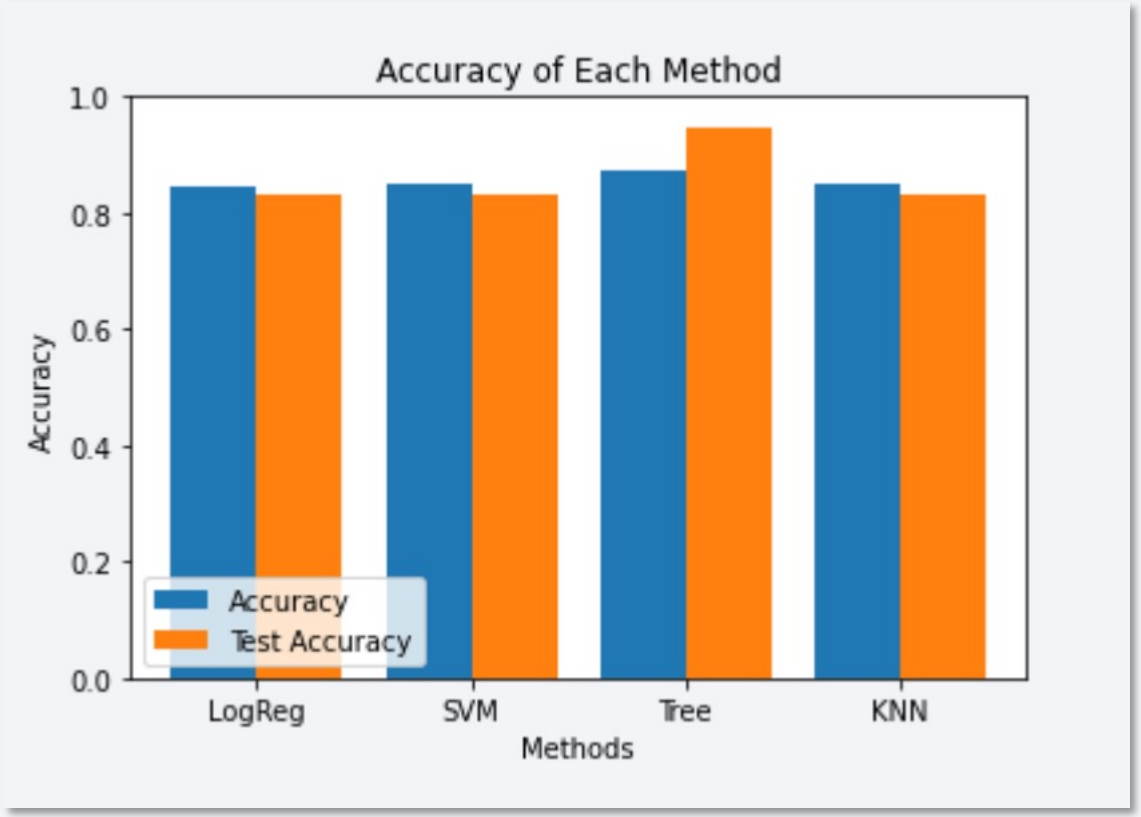
Predictive Analysis (Classification)

Classification Accuracy

- Four classification models were tested: Logistic regression, SVM, Decision Tree Analysis and K Nearest Neighbors
- From the bar chart we can see that Decision Tree Analysis performed the best with a test accuracy of over 88%

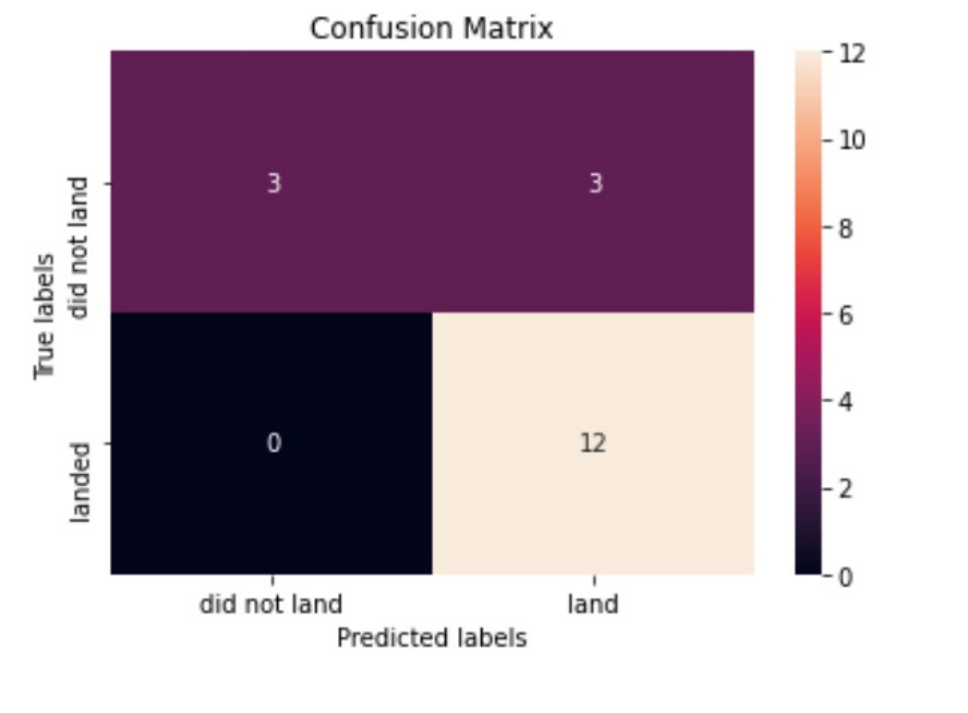
```
print('Accuracy for Logistics Regression method:', logreg_cv.best_score_)
print( 'Accuracy for Support Vector Machine method:', svm_cv.best_score_)
print('Accuracy for Decision tree method:', tree_cv.best_score_)
print('Accuracy for K nearest neighbors method:', knn_cv.best_score_)
```

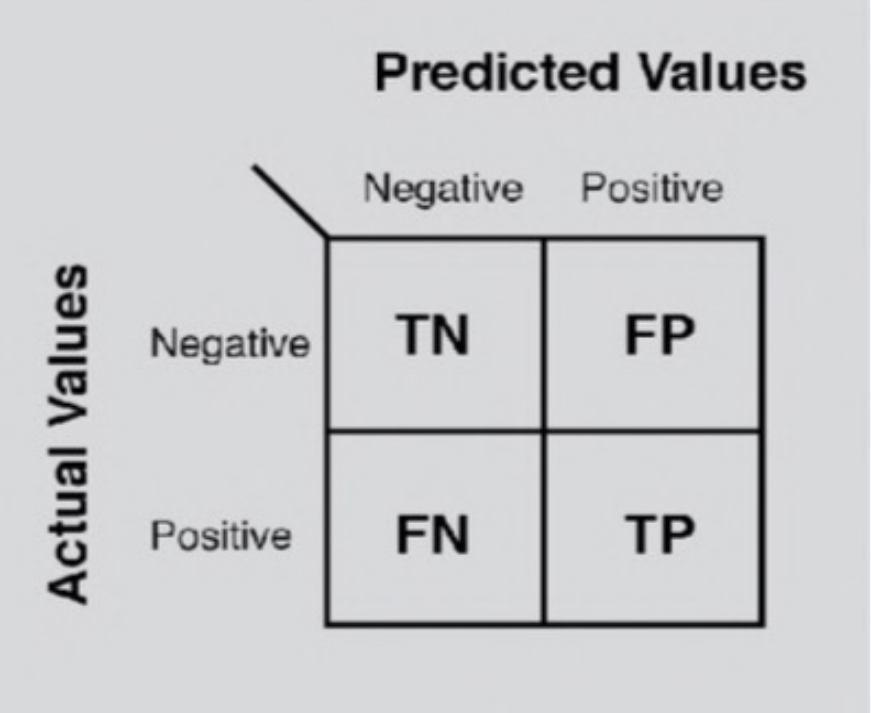
```
Accuracy for Logistics Regression method: 0.8464285714285713
Accuracy for Support Vector Machine method: 0.8482142857142856
Accuracy for Decision tree method: 0.8875
Accuracy for K nearest neighbors method: 0.8482142857142858
```



Decision Tree Confusion Matrix

The decision tree confusion matrix shows that this was the best performing model illustrated by the high number of true positive results





Conclusions

- The most successful launch site was KSC LC-39A with a 76.9% success rate
- The success rate of missions has continued to improve since 2013 most likely due to the improvement of processes
- Launch-sites with a payload mass > 6000 kg appear more successful
- Decision Tree analysis was the best classifier algorithm at predicting mission success from independent variables
- Some orbits had better success than others but due to inconsistent number of launch flights its difficult to draw any solid conclusions from this

Thank you!