IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Paraskevi (Vivian) SYNTETA
(Synteta@gmail.com)
21/12/2023

Outline

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

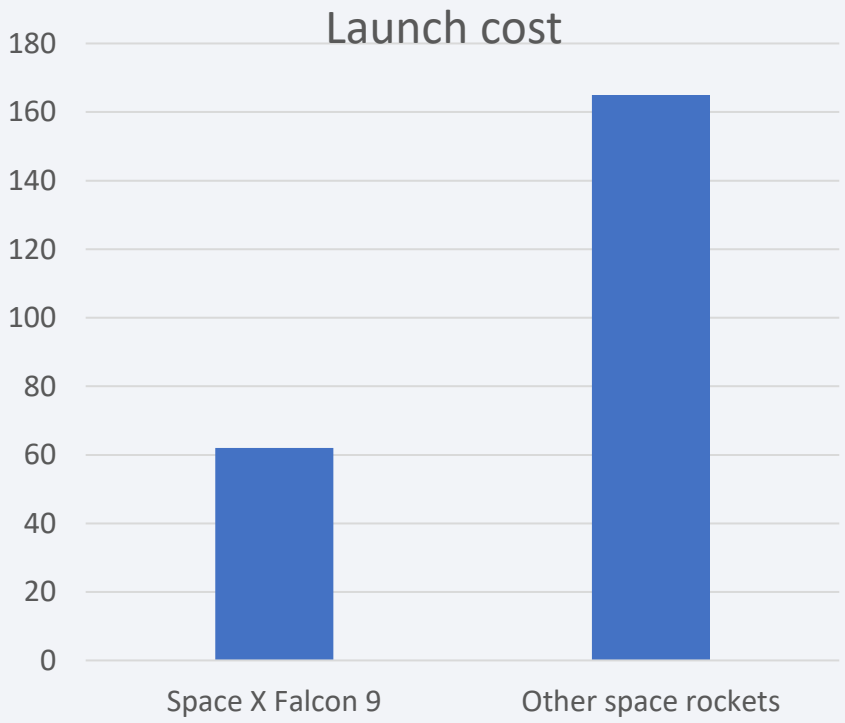
Executive Summary

- In the space race, Space X made a difference lowering the launch cost of their Falcon 9 rocket by saving as much as possible the first stage that is the costliest and reuse it.
- Thus, being able to predict the launch success seems critical.
- This work, using data science methodology started by collecting SpaceX data using it's API and web scraping a Wikipedia page, cleaning and wrangling it. To find the factors that influence the outcome and in which way, we have used Exploratory Data Analysis with visualization and SQL, interactive visual analytics through Folium maps and Plotly Dashboards. And to make predictions on the outcome we managed to built, tune and evaluate a ML model for launch outcome prediction of the Falcon 9 with high accuracy (94.4%).

Results

- Factors from EDA: Flight no, launch site location and proximity, year, payload mass, orbit type, booster
- Best site KSC LC-39A (close to coastline and far from inhabited areas), worst CCAPS SLC-40
- Best Payload 2K-4K and FT booster version, worst Payload 0-2K, 4K-10K and booster v1.1 (mass in Kg)
- Best orbits ES-L1, GEO, HEO, SSO (100% success), worst SO
  - Best orbits with heavy payload are LEO, ISS and Polar (still with medium success)
- Best classification ML model for outcome prediction is the Decision tree coupled with specific hyperparameters.

Introduction



- Space X Falcon 9 rocket launch cost ($62M)
- Savings thanks to reuse of the first stage.
- First Stage Landing Prediction can determine the cost of a launch.
  - can be used if an alternate company wants to bid against space X for a rocket launch.

Project

- Create a machine learning pipeline to predict if the first stage will land based on various data.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Gather launches data (SpaceX API & web scrapping of Wikipedia page Falcon 9)
- Perform data wrangling
  - Handle missing values, convert outcomes to training labels and add to a new column “Class”
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Find correlations between various factors through visualization techniques
  - Get insights from data through SQL queries
- Perform interactive visual analytics using Folium and Plotly Dash
  - Visualize launch sites on map and get insights from location and proximities
  - Interact with data and query them on launch site, success and payload range to get best combinations
- Perform predictive analysis using classification models
  - Build, tune, evaluate 4 classification models to get the best for prediction

Data Collection

- Request rocket launch data from SpaceX API:
  - https://api.spacexdata.com/v4/launches/past
  - Actually used a static json object to make request results more consistent: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/data
- Clean the data

Data Collection – SpaceX API

0. • Request rocket launch data from SpaceX API
1. • Decode response as a json
2. • Turn it to a pandas DataFrame
3. • Keep only a subset of the DF with 6 cols ('rocket','payloads','launchpad','cores','flight_number','date_utc')
4. • Keep only datetime and Restrict dates from 'date_utc'
5. • Create lists from filtered API data based on IDs (e.g, from rocket find the Booster version, from payload the mass and the orbit, etc)
6. • Combine into a dictionary
7. • Create a new DF out of the above dict
8. • Filter to include only launches of Falcon 9

GitHub

- https://github.com/VivianSynteta/IBMDataScienceProfCertifCapstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Links

- Wikipedia page "List of Falcon 9 and Falcon Heavy launches"
  - Actually, snapshot updated on 9th June 2021
- GitHub

Collect data through web scrapping a Wikipedia page	• Request the page with HTTP GET
Extract Falcon 9 launch records from an HTML table	• Create a beautifulsoup object of the HTTP response
Parse the table and convert it into a Pandas data frame	• Extract the column names • Create a dict from table row data • Turn dict to DF
Export to CSV	

Data Wrangling

1. Deal with missing values (PayloadMass, LaunchingPad)
2. Replace nan with mean
3. convert outcomes to Training Labels 1/0 = booster successfully/unsuccessfully landed
4. Create a new column Class in the DF with those outcomes
5. Export the dataset to a CSV

GitHub:

- https://github.com/VivianSynteta/IBMDataScienceProfCertifCapstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb
- https://github.com/VivianSynteta/IBMDataScienceProfCertifCapstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Exploratory Data Analysis

- preliminary insights about relationships; how each important variable would affect success rate
- select the features used in success prediction

Charts/Graphs

Five (5) scatter point charts

- Flight number vs Payload Mass
- Flight number vs Launch Site
- Payload Mass vs Launch Site
- Flight number vs Orbit
- Payload Mass vs Orbit

A bar chart : Success rate per Orbit

A line plot: Launch success per Year

Feature Engineering

- select the features used in success prediction
- One-hot encoding
  - Create dummy variables to categorical columns – ended up with 80 cols
- Cast all numeric columns to `float64`

GitHub

- https://github.com/VivianSynteta/IBMDataScienceProfCertifCapstone/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

Load dataset into a Db2 table & execute the following SQL queries to get insights on data

- select DISTINCT "Launch_Site" from SPACEXTABLE;
- select * from SPACEXTABLE where "Launch_Site" like 'CCA%' limit 5;
- SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload_mass FROM SPACEXTABLE;
- SELECT AVG("PAYLOAD_MASS__KG_") AS Average_payload_mass_BoosterF9v11 FROM SPACEXTABLE where "Booster_Version"='F9 v1.1';
- SELECT MIN(DATE) FROM SPACEXTABLE where "Landing_Outcome" LIKE 'Success%';
- SELECT Booster_Version FROM SPACEXTABLE where "Landing_Outcome"='Success (drone ship)' AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;
- SELECT "Landing_Outcome", Count(*) FROM SPACEXTABLE AS Outcome_count where "Landing_Outcome" IN ('Success', 'Failure') GROUP BY "Landing_Outcome";
- SELECT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_"=(select max(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
- select substr(Date,6,2), "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTABLE where "Landing_Outcome"='Failure (drone ship)' AND substr(Date,0,5)='2015';
- SELECT "Landing_Outcome", Count(*) FROM SPACEXTABLE where "Landing_Outcome" IN (select distinct "Landing_Outcome" FROM SPACEXTABLE) GROUP BY "Landing_Outcome";

GitHub: https://github.com/VivianSynteta/IBMDataScienceProfCertifCapstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Launch success rate may depend on **many factors** (payload mass, orbit type etc)

EDA with Visualization: discovered preliminary correlations between launch site and success rates

Folium - more interactive visual analytics

- Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations.

Created a Folium map object initiated at NASA with

- circles and markers on each launch site,
- marker clusters for launch outcomes on each site (green for success, red for failure)
- Polyline and marker with the distance of a selected coastline and a railway

GitHub:
https://github.com/VivianSynteta/IBMDataScienceProfCertifCapstone/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

Check launches success or failure per site or in ALL sites

- **Input**: Dropdown for Launch Site selection (default value ALL sites)
- **Output**: Pie chart with Success vs. Failed counts launches for dropdown selection

Check launches success or failure per site with various payloads

- **Input**:
  - Dropdown for Launch Site selection (default value ALL sites)
  - Slider to select payload range
- **Output**: Scatter chart to show the correlation between payload and launch success

GitHub:

- https://github.com/VivianSynteta/IBMDataScienceProfCertifCapstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

Created ML pipeline to predict if the first stage will land given the collected data

- Load data
- Standardize wrangled data using a transform function (X DF)
- Filter column “Class” from data (Y Pandas series)

Split into training data and test data (80-20%)

Classification Models trained: Logistic Regression, SVM, Classification Tree and KNN

- Best hyperparameters are selected using the function GridSearchCV
- Find the method performs best using test data (check accuracy and use of confusion matrix)

GitHub:
https://github.com/VivianSynteta/IBMDataScienceProfCertifCapstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

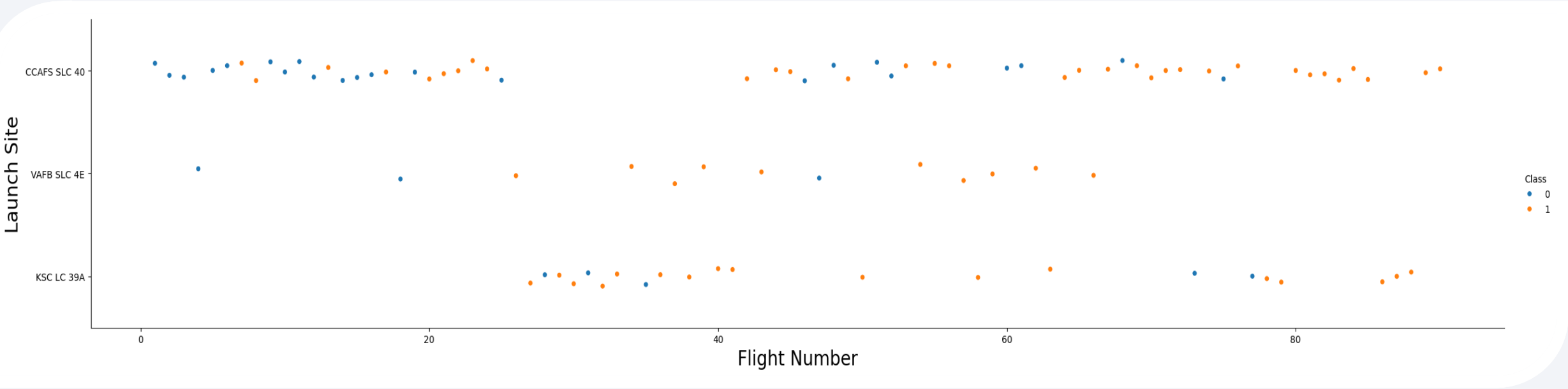
Results

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

Section 2

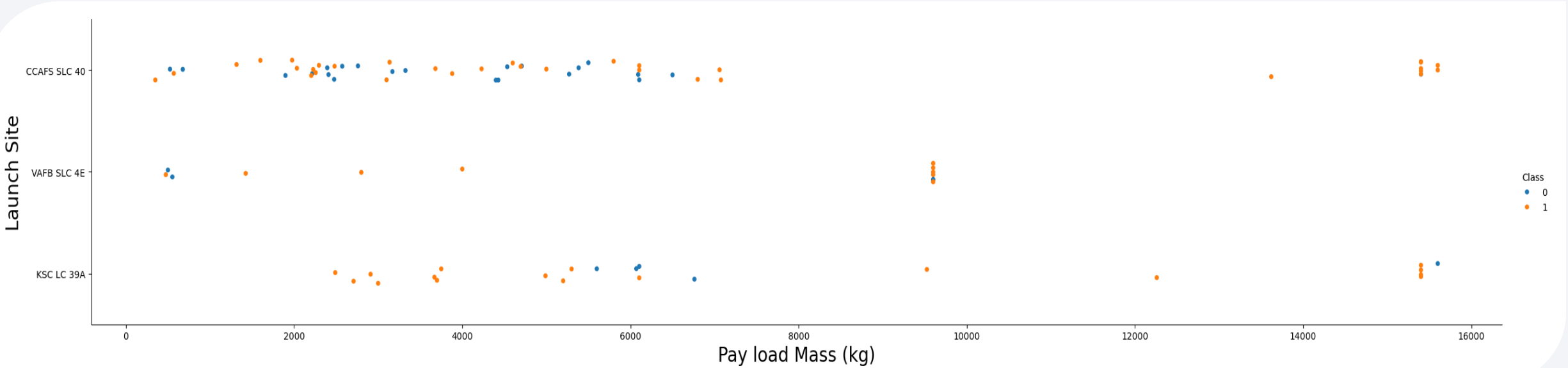
Insights drawn from EDA

Flight Number vs. Launch Site



- We clearly see that for all sites, the more flights, the more successes in landing
- Launch site VAFB SLC 4E has very few launches
- Launch site CCAFS SLC 40 has the most launches but the lowest success rate
- Launce sites VAFB SLC 4E & KSC LC 39A have the highest success rate

Payload vs. Launch Site



- One can see that various payloads can have successful landing in all launch sites
- Only Launch site VAFB-SLC does not have launches more than 10000Kg Payload Mass
- Very few launches take place with heavy payload mass.

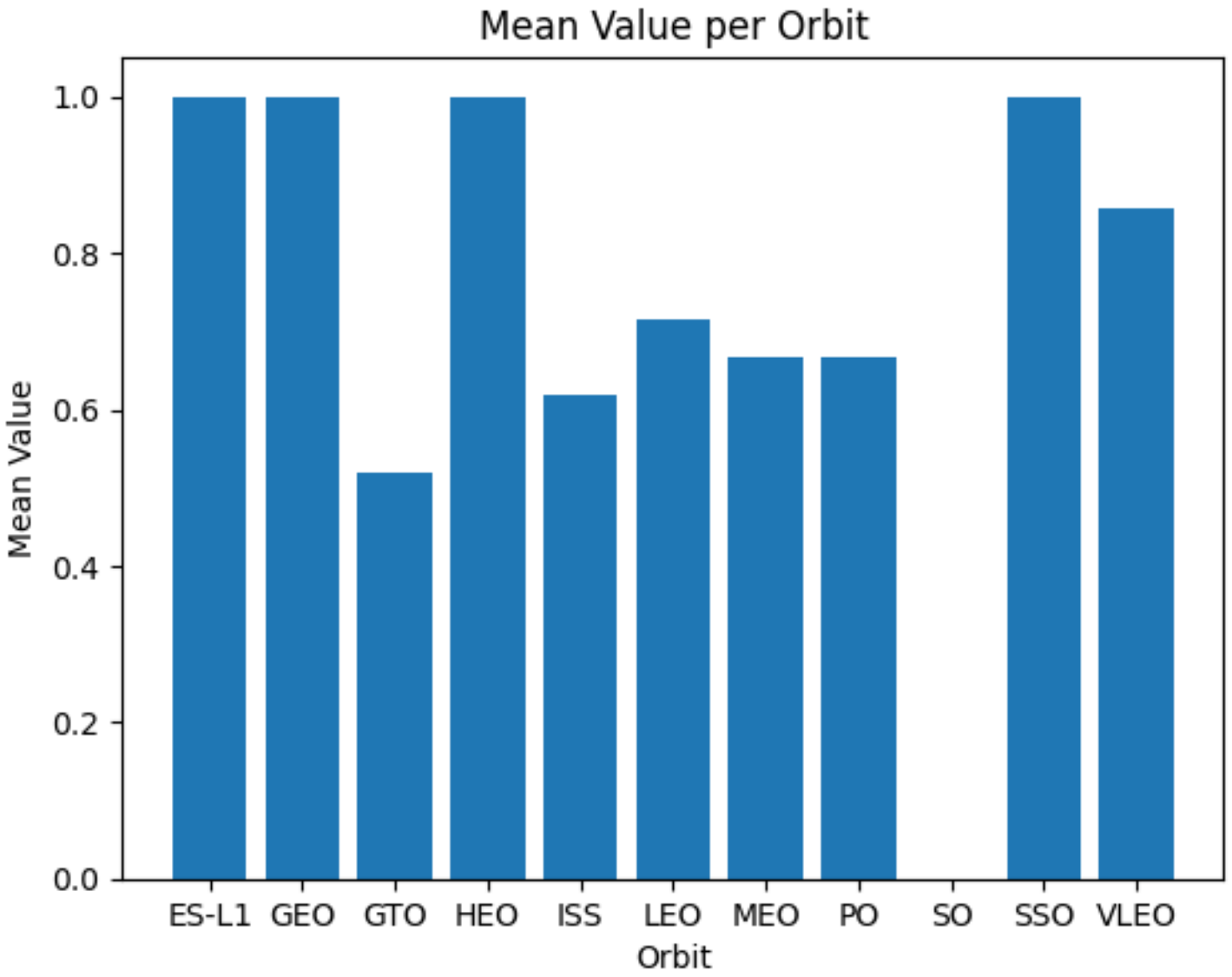
Success Rate vs. Orbit Type

Orbits with high success rate (100%):

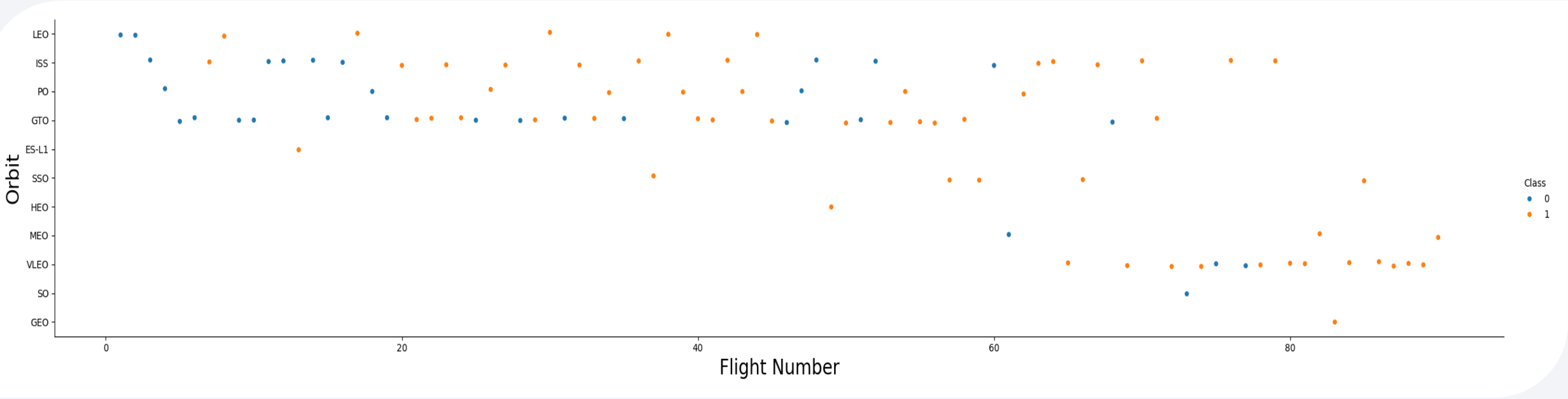
- ES-L1
- GEO
- HEO
- SSO

Lowest rate (0%):

- SO

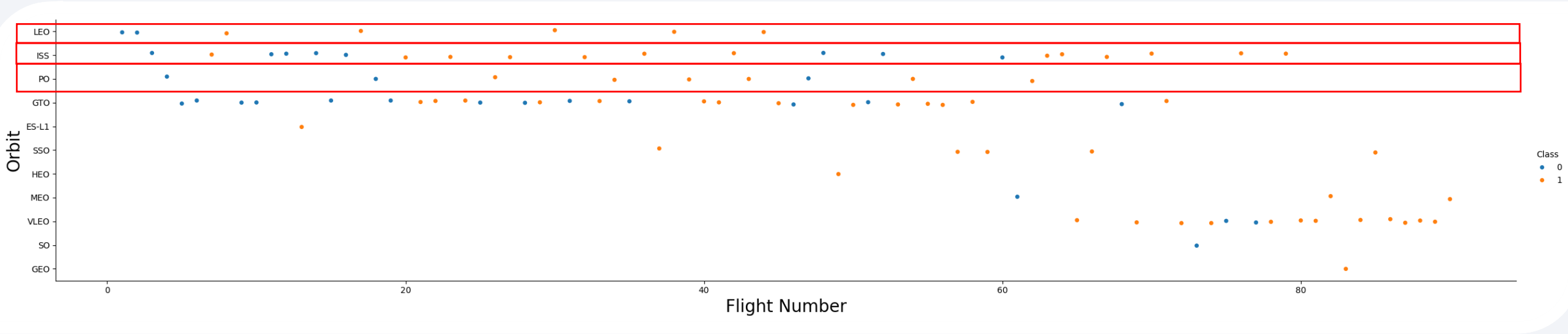


Flight Number vs. Orbit Type



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

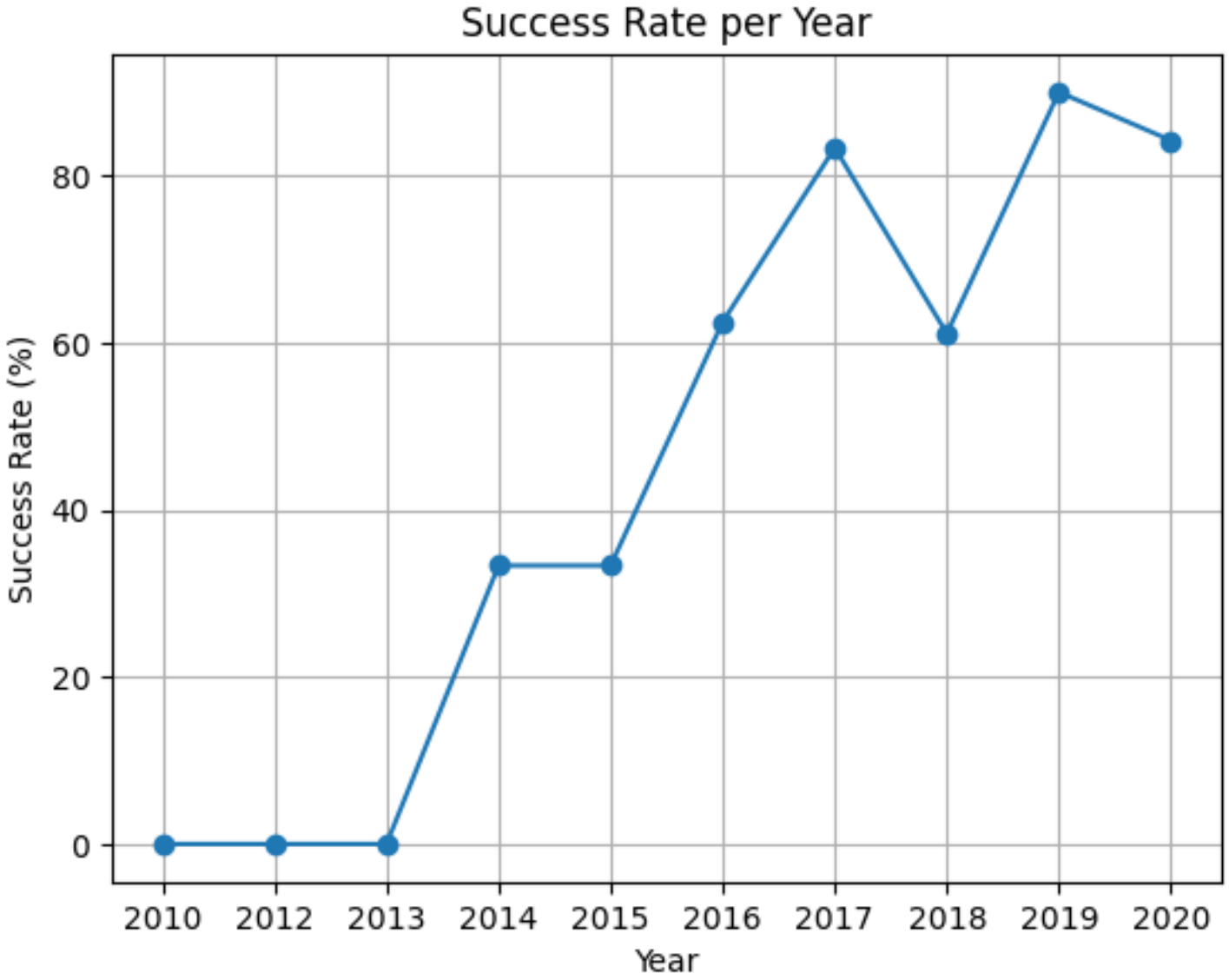
Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for LEO, ISS and Polar.
- However, for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both here.

Launch Success Yearly Trend

- we observe that the success rate since 2013 kept increasing till 2020 with a small drop during 2017-2018
- Best lately (2019-2020)



All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

```
%sql select DISTINCT "Launch_Site" from SPACEXTABLE;
```

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

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

Four (4) launch sites

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTABLE where "Launch_Site" like 'CCA%' limit 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload_mass FROM SPACEXTABLE;
```

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

total_payload_mass
619967

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") AS Average_payload_mass_BoosterF9v11 FROM SPACEXTABLE where "Booster_Ve
```

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

Average_payload_mass_BoosterF9v11
2928.4

First Successful Ground Landing Date

```
%sql SELECT MIN(DATE) FROM SPACEXTABLE where "Landing_Outcome" LIKE 'Success%';
```

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

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT Booster_Version FROM SPACEXTABLE where "Landing_Outcome"='Success (drone ship)' AND "PAYLOAD_MAS
```

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

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

```
%sql SELECT Booster_Version FROM SPACEXTABLE
where "Landing_Outcome"='Success (drone ship)'
AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;
```

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT "Landing_Outcome", Count(*) FROM SPACEXTABLE AS Outcome_count where "Landing_Outcome" IN ('Succe
```

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

Landing_Outcome	Count(*)
Failure	3
Success	38

```
%sql SELECT "Landing_Outcome", Count(*)
FROM SPACEXTABLE
AS Outcome_count
where "Landing_Outcome" IN ('Success', 'Failure')
GROUP BY "Landing_Outcome";
```

Boosters Carried Maximum Payload

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_"=(select max(PAYLOAD_MASS__K
```

```
 * 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

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE
WHERE "PAYLOAD_MASS__KG_“
=(select max(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

Query in a query

2015 Launch Records

```
%sql select substr(Date,6,2), "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEX
```

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

substr(Date,6,2)	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

```
%sql select substr(Date,6,2), "Landing_Outcome", "Booster_Version", "Launch_Site"
from SPACEXTABLE
where "Landing_Outcome"='Failure (drone ship)' AND substr(Date,0,5)='2015';
```

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

```
%sql SELECT "Landing_Outcome", Count(*) FROM SPACEXTABLE where "Landing_Outcome" IN (select distinct "Landi
```

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

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

```
%sql SELECT "Landing_Outcome", Count(*) FROM SPACEXTABLE

where "Landing_Outcome"

IN (select distinct "Landing_Outcome" FROM SPACEXTABLE)

GROUP BY "Landing_Outcome";
```

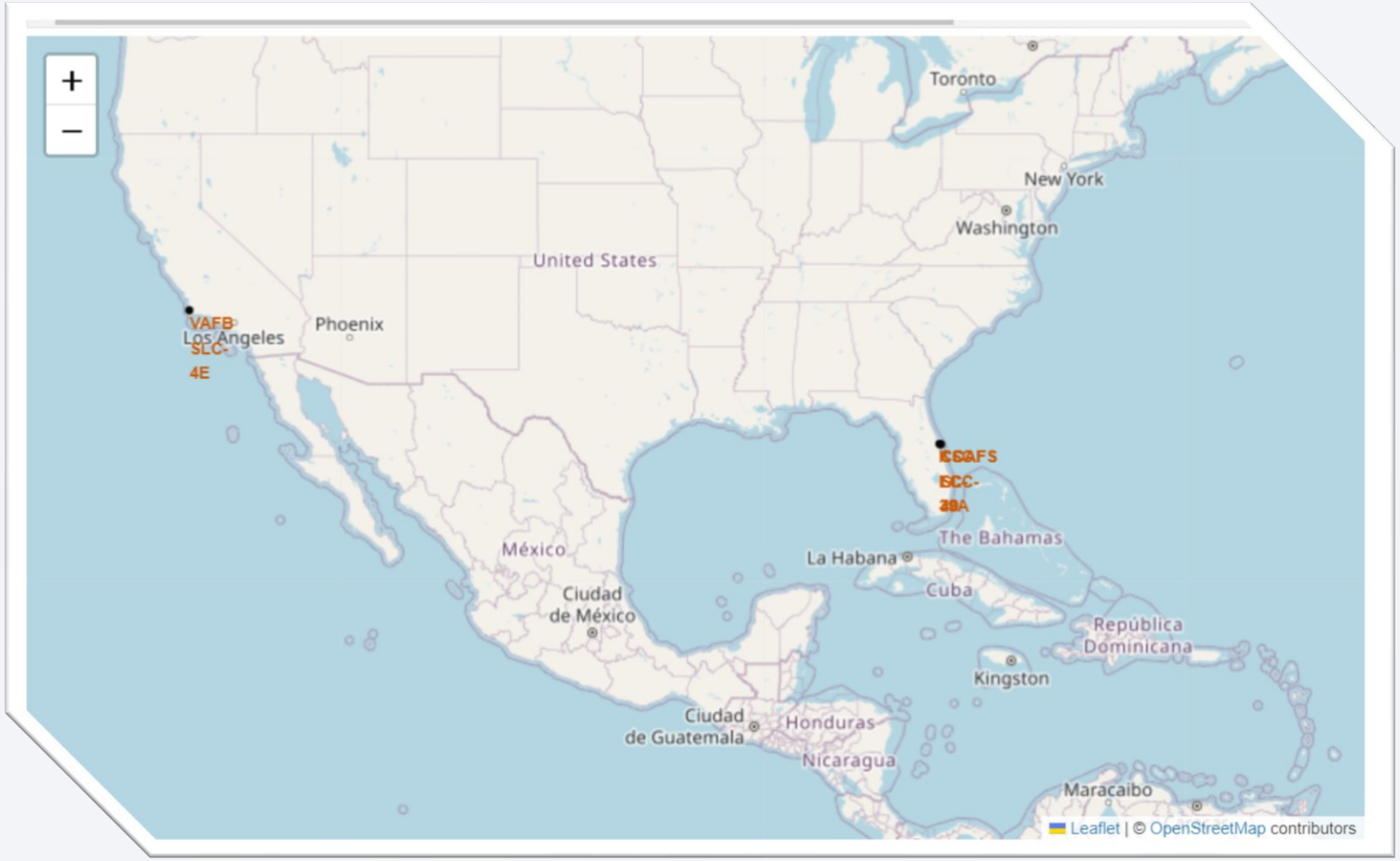
Section 3

Launch Sites Proximities Analysis

Launch sites' location

Findings:

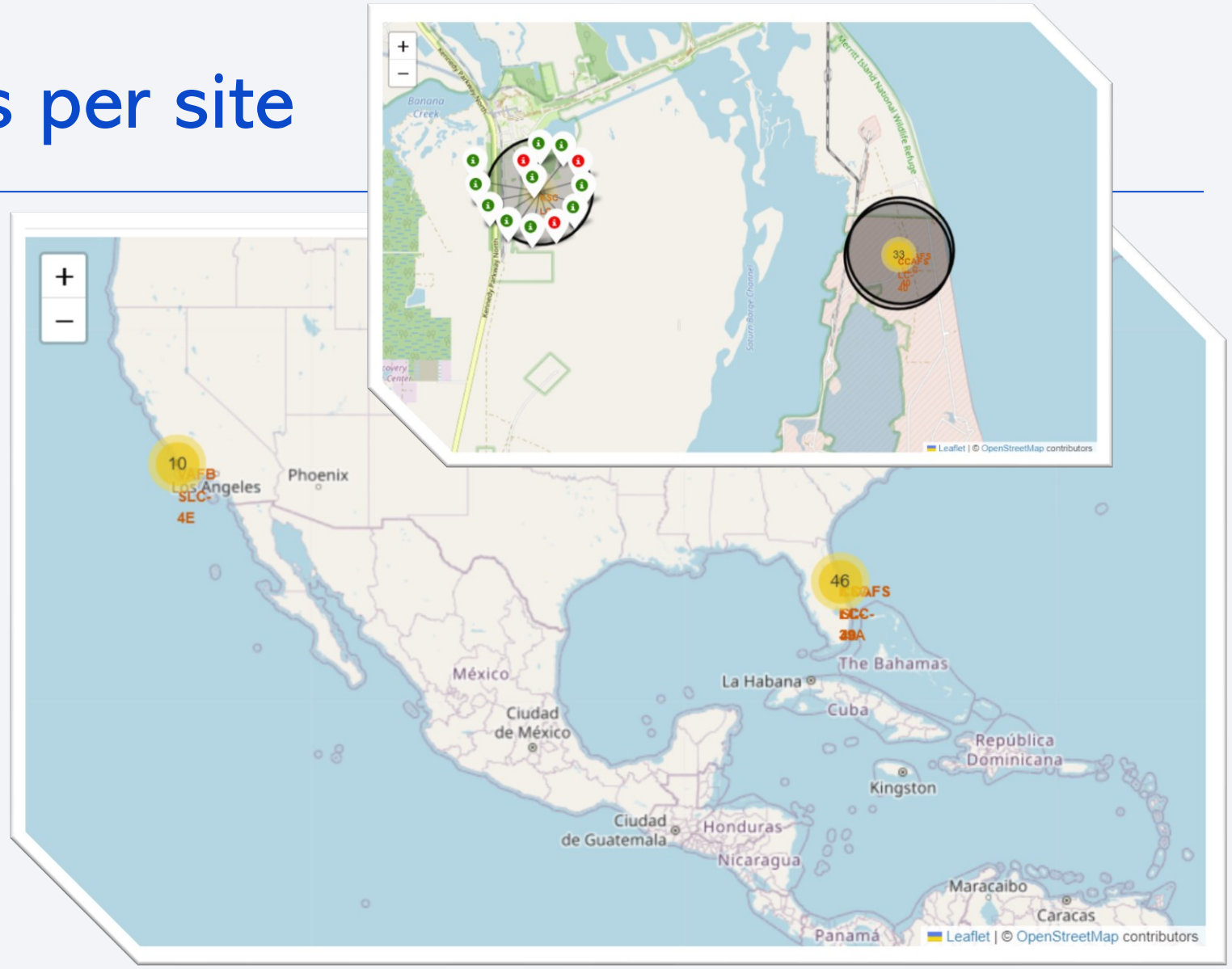
- Four (4) launch sites
- All launch sites in proximity to the Equator line
- All launch sites in very close proximity to the coast



Launch outcomes per site

Findings:

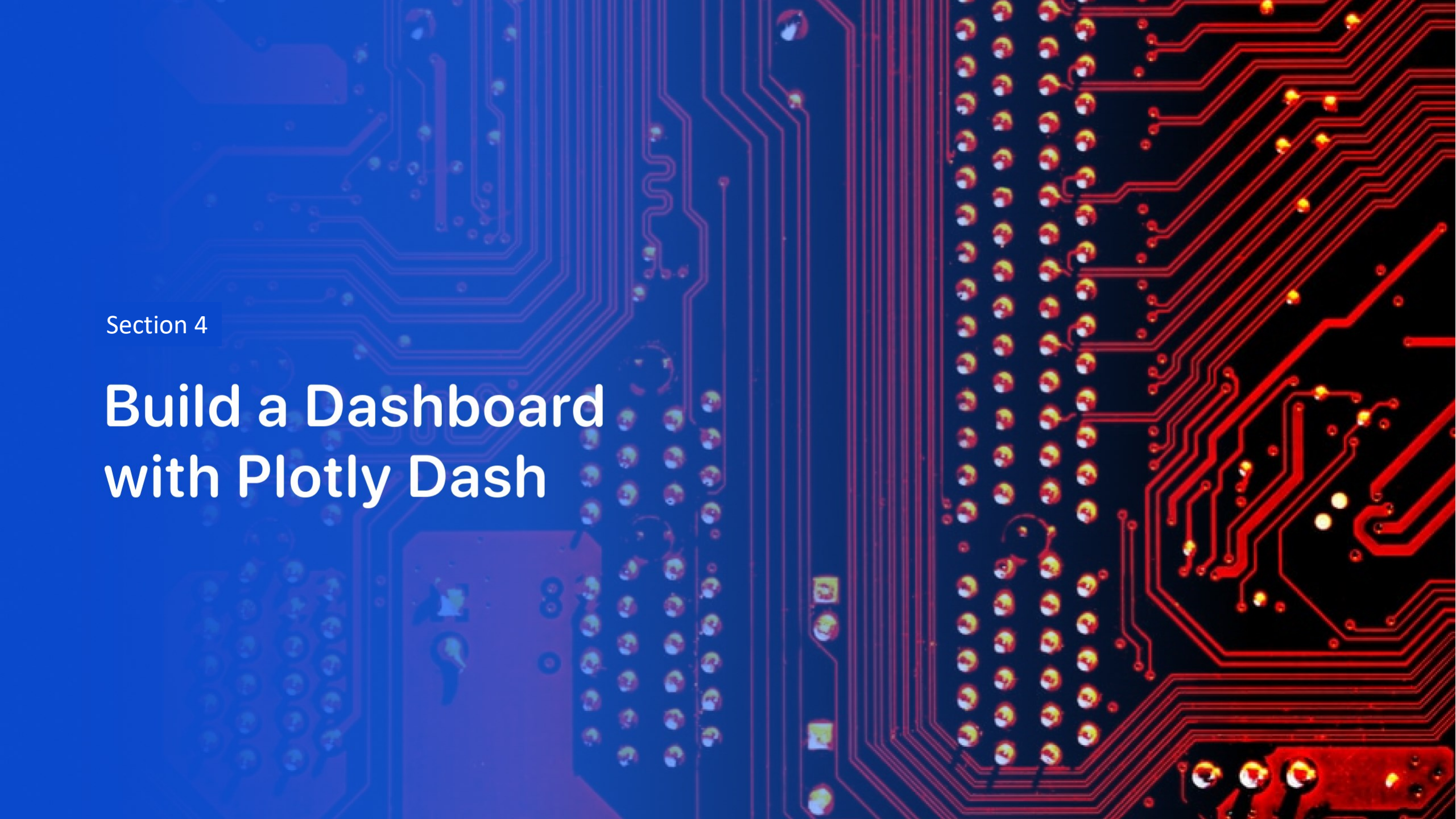
- Most launches from site CCAFS SLC 40
- Site with highest success rates KSC LC-39A



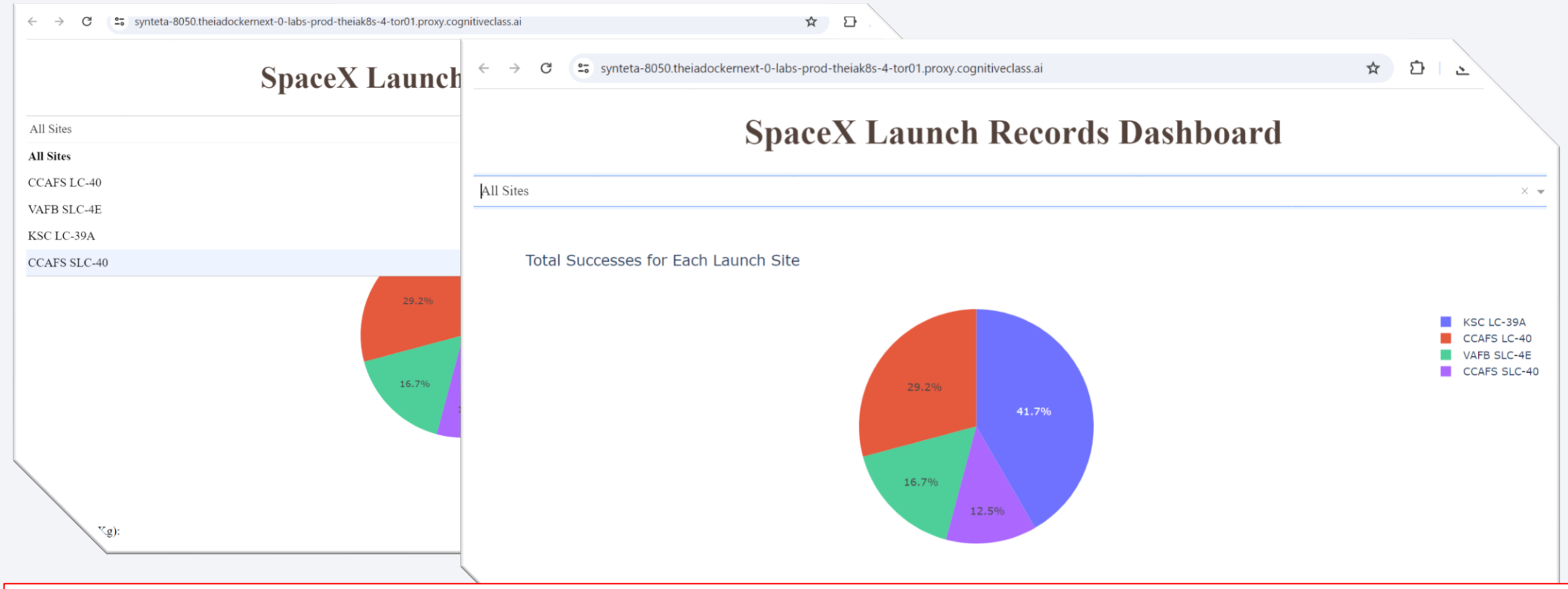
Launch site and proximity to coastline

Findings:

- Launch sites are close to coastline, railways (<1Km)
- Launch sites are far from cities, highways


Section 4
Build a Dashboard with Plotly Dash

Launch success count for ALL sites



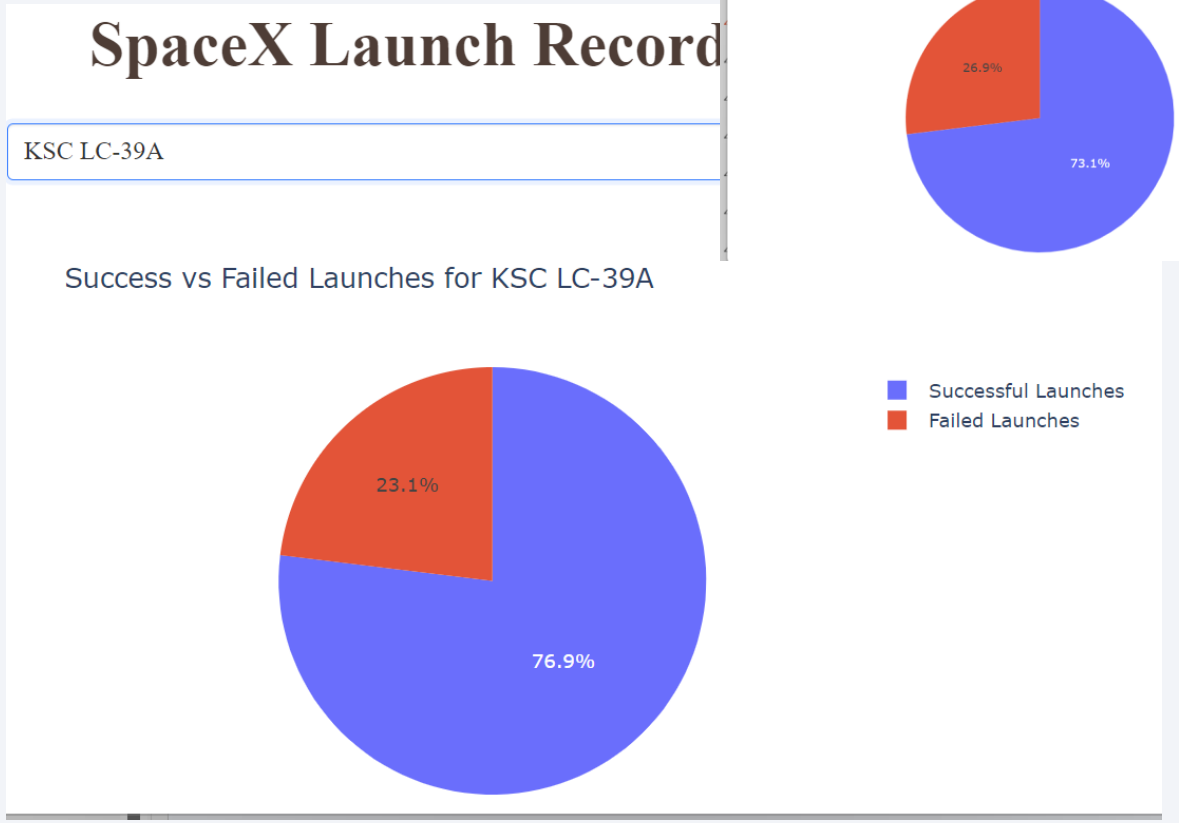
Findings:

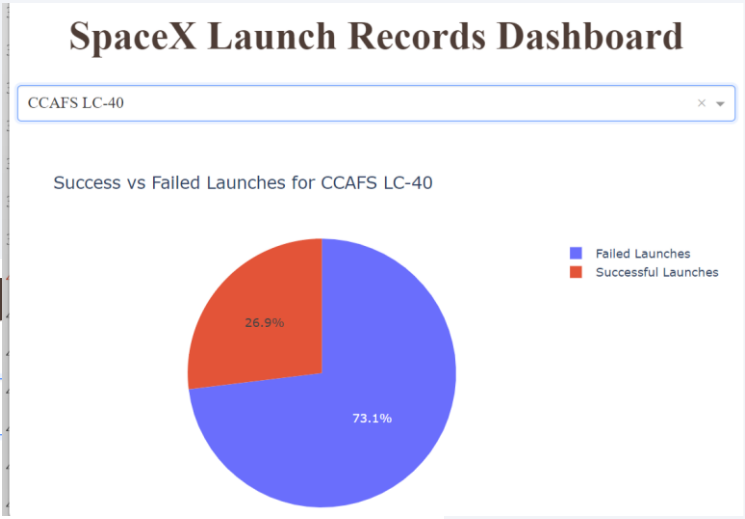
- Launch sites KSC LC-39A and CCAPS SLC-40 have the largest and the smallest successful launches respectively

Launch site with highest success ratio

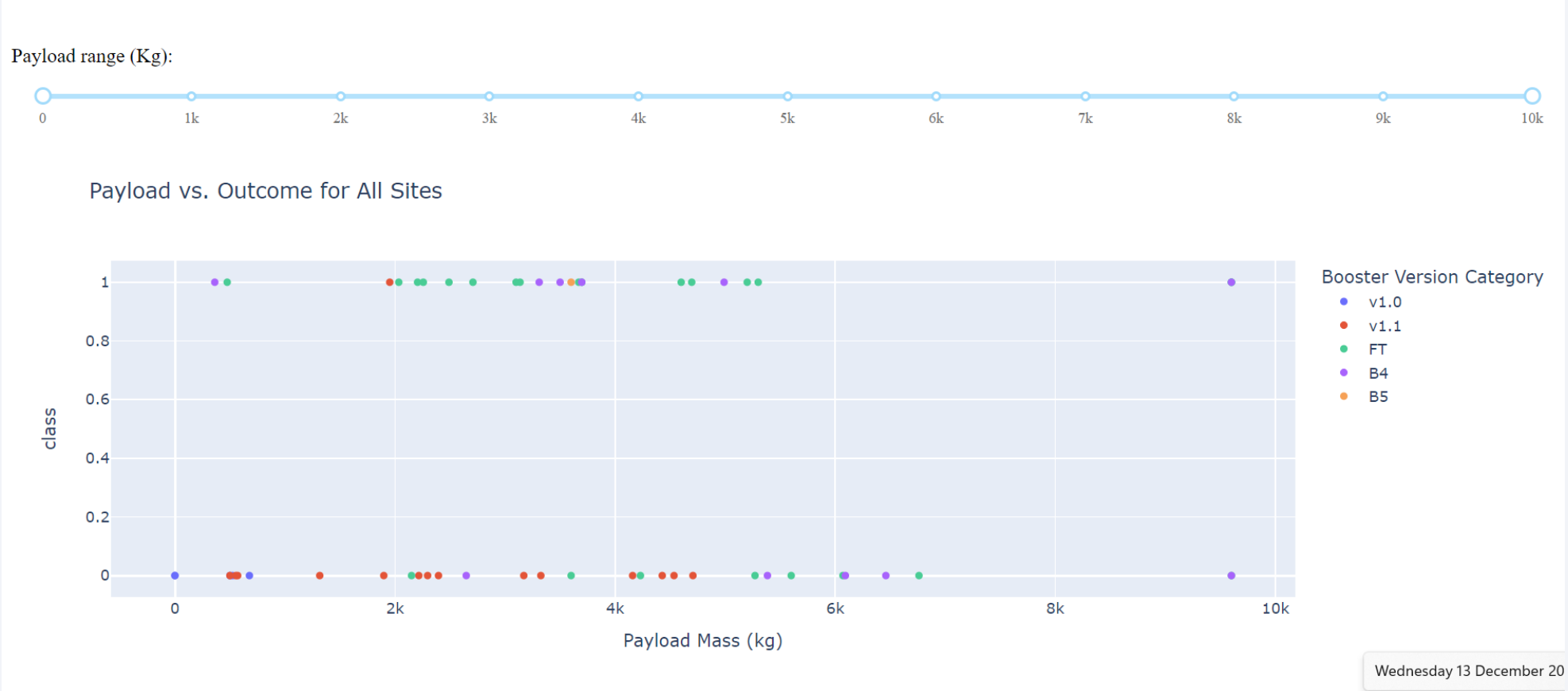
Findings:

- The site KSC LC-39A has the highest launch success rate
- Second best is CCAFS LC-40





Payload vs. Launch Outcome I (all payloads)



Findings:

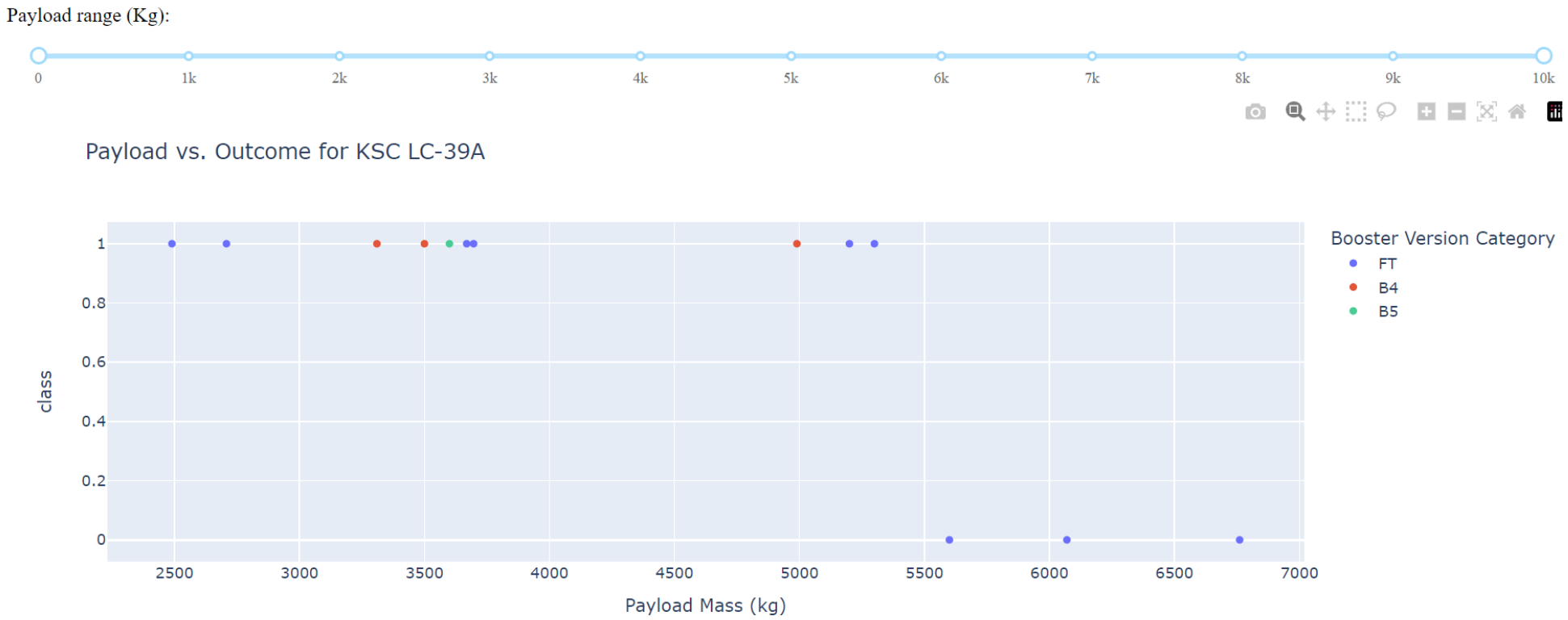
- Payload 2K-4K and FT booster version have the largest success rate.
- Payload 0-2K, 4K-10K and booster v1.1 have the lowest success rate

Payload vs. Launch Outcome II (range of payload)

Findings:

- Payload more than 4K has mostly failures apart from the window 4.5K-5.5K that has a few succeses

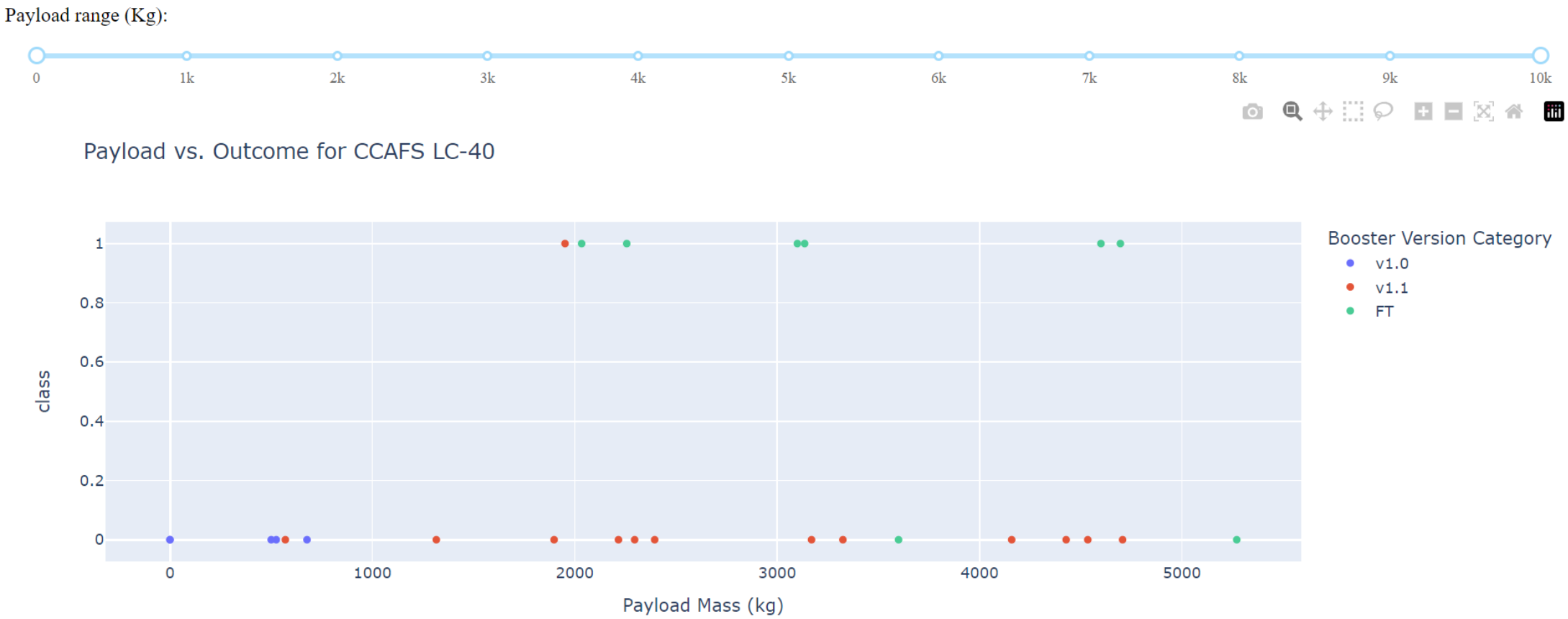
Payload vs. Launch Outcome (KSC LC-39A)



Findings:

- Successes: payload<5.5K, FT, B4, B5 boosters - Failures: payload>5.5K, FT booster
- No launches >7K

Payload vs. Launch Outcome (CCAFS LC-40)



Findings:

- Successes: payload in 2K-5K (few) mostly FT booster - Failures: mostly v1.1 booster
- No launches >5.5K payload

Section 5

Predictive Analysis (Classification)

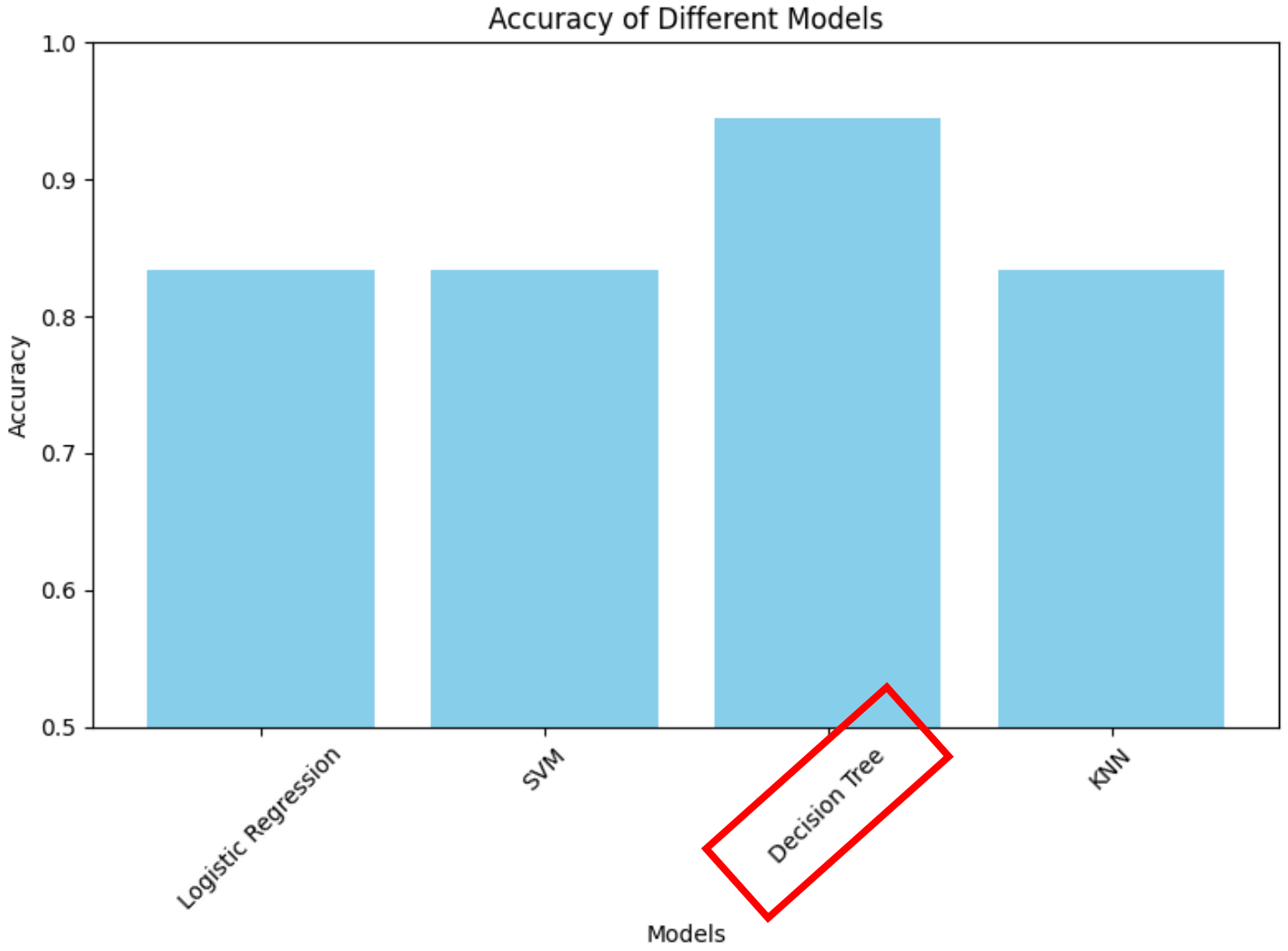
Classification Accuracy

Best model:

Decision Tree (accuracy 94.4%)

Tuned hyperparameters (best parameters):

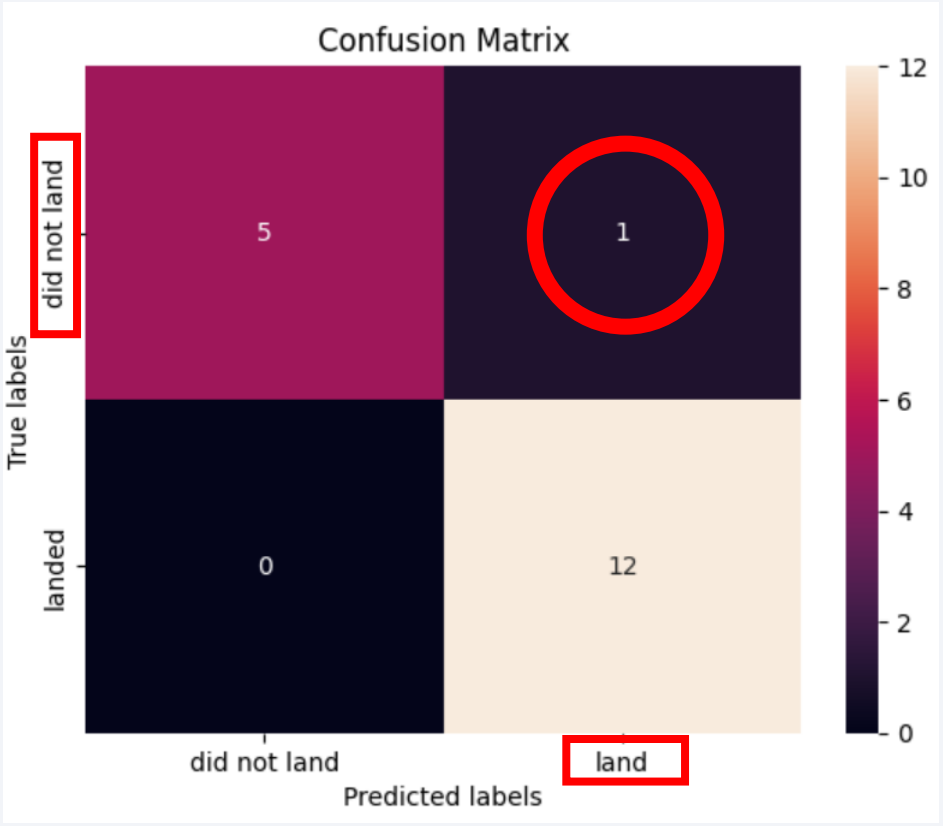
{'criterion': 'entropy', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'best'}

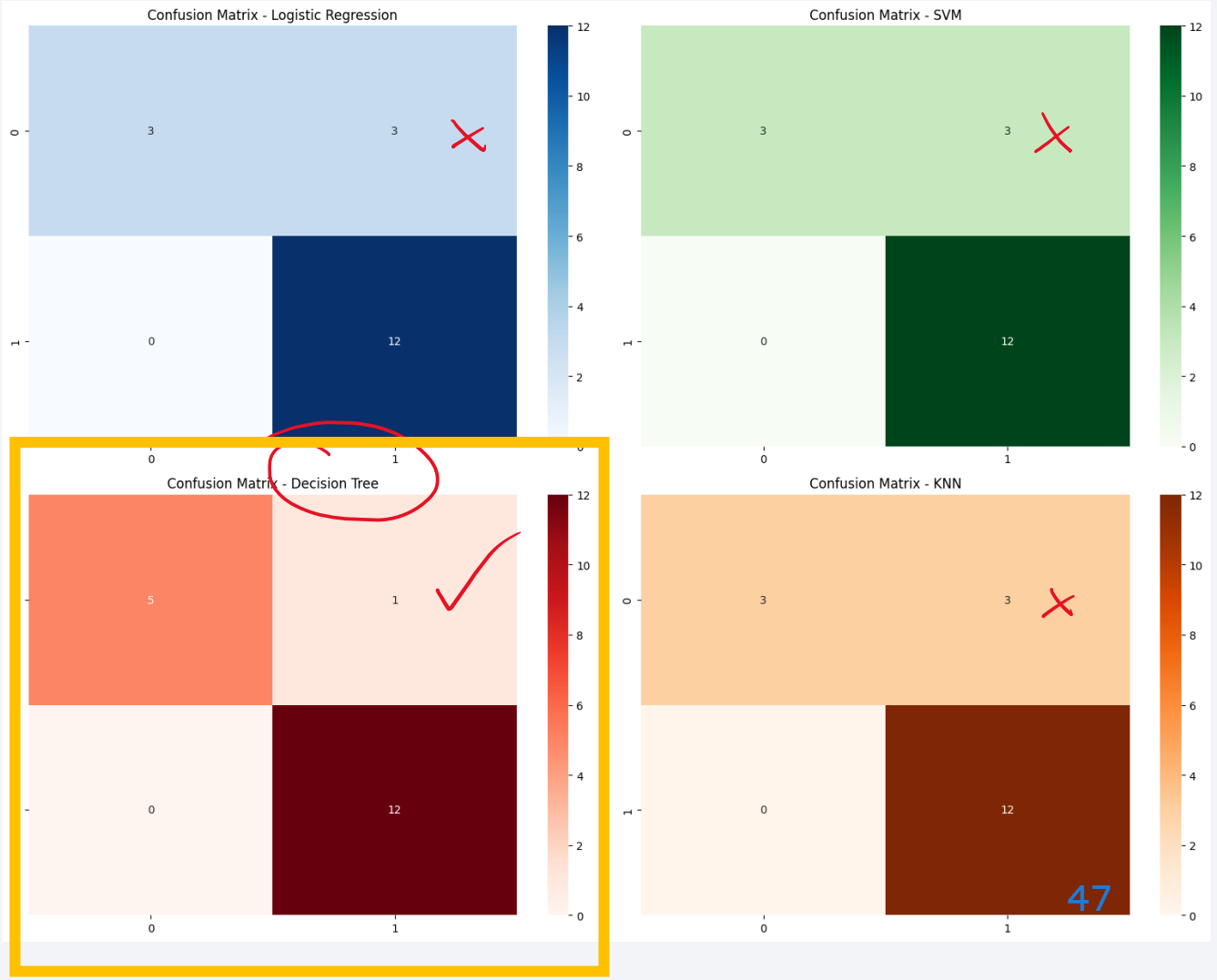


Confusion Matrix

Best performing model:

Decision Tree (least false positives)





Conclusions

General

- Launch site location and proximity, year, payload mass, orbit and booster are important factors for launch success
- No of launches does not affect much per site but it does totally

Details

- Location closest to Equator, coastline and railways but as far as possible from cities and highways
- Launch site highest success rate is not because of many launches
- Best site KSC LC-39A, worst CCAPS SLC-40
- Best Payload 2K-4K and FT booster version, worst Payload 0-2K, 4K-10K and booster v1.1
- Best orbits ES-L1, GEO, HEO, SSO (100% success), worst SO
- Best orbits with heavy payload are LEO, ISS and Polar (still with medium success)

Best outcome prediction

- Decision Tree Model (see best parameters)
- Best Machine learning model (highest accuracy and least false positives)

Appendix

- Project GitHub: https://github.com/VivianSynteta/IBMDataScienceProfCertifCapstone
- Dataset (CSV): https://github.com/VivianSynteta/IBMDataScienceProfCertifCapstone/blob/main/Spacex.csv
- Dataset for Dash (CSV): https://github.com/VivianSynteta/IBMDataScienceProfCertifCapstone/blob/main/spacex_launch_dash.csv
- Dashboard in Python: https://github.com/VivianSynteta/IBMDataScienceProfCertifCapstone/blob/main/spacex_dash_app.py

Thank you!