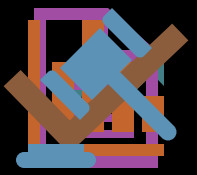
IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tim Kong
30 October, 2021



Outline

Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

Executive Summary

Summary of methodologies

Data collection through web scrapping and API

Data wrangling

EDA with data visualization and SQL

Interactive map with Folium and Ploty Dash

Predictive analysis – classification

Summary of all results

Data analysis results with interavtive visulaizations

Predictive anaylsis results

Introduction

Project background and context

- This project is to predict if the first stage of the SpaceX Falcon 9 rocket will land successfully. SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- if the first stage rocket landing successfully can be predicted accurartely, the cost of a launch can be determined. With the help of all the Data Science findings and models, the associated company can make more informed bids against SpaceX for a rocket launch.

Problems that lead to finding answers

- What factors affect the successs rate of a land?
- Does the relationship between the variables that impact the outcome?

Section 1

Methodology

Methodology

Data collection methodology

- Web scrapping with BS4 from Wikipedia page titled - "List of Falcon 9 and Falcon Heavy launches"
- SpaceX REST API

Perform data wrangling

- One hot encoding techniques with pandas

Perform exploratory data analysis (EDA)

- Data visualization and SQL

Perform interactive visual analytics

- Folium and Plotly Dash

Perform predictive analysis using classification models

- Build, tune, evaluate fourclassification models

Data Collection – SpaceX REST API

- Get response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

- Convert Response to .json file

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

- Apply auxiliary functions clean data Store data in a list of dictionary then turn into a data fame

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
               'Date': list(data['date']),...}
df = pd.DataFrame(launch_dict)
```

GitHub URL to notebook

Data Collection – Web Scraping

1
```
response = requests.get(static_url)
```

2
```
soup = BeautifulSoup(response.text)
```

3
```
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
```

4
```
column_names = []

for th in first_launch_table.find_all('th'):
    th = th.getText().strip()
    if th.isnumeric() == False:
        if th[-1] == ']':
            column_names.append(th.split('[')[0].strip())
        else:
            column_names.append(th)
```

5
```
launch_dict= dict.fromkeys(column_names)
```

6
```
df=pd.DataFrame(launch_dict)
df.to_csv('spacex_web_scraped.csv', index=False)
```

- Get response from html
- Get BeautifulSoup object
- Locate interested table
- Extract column names
- Append data to keys of dictionary
- Export data frame as .CSV file

GitHub URL to notebook

Data Wrangling

01

Calculate the number of launches on each site

02

Calculate the number and occurrence of each orbit

03

Calculate the number and occurence of mission outcome per orbit type

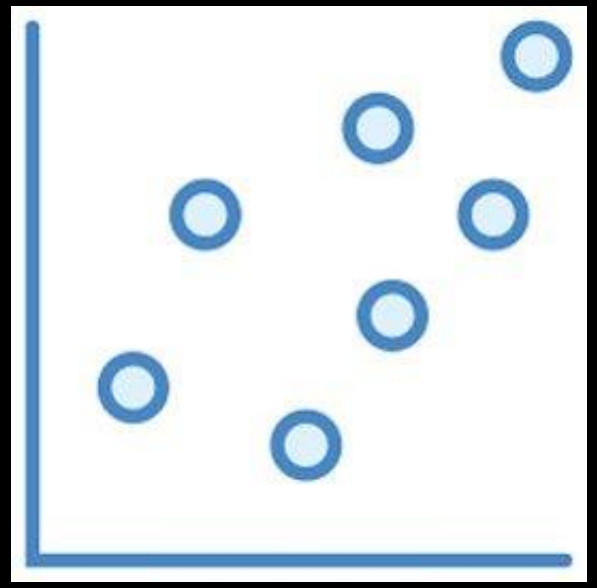
04

Create a landing outcome label from Outcome column

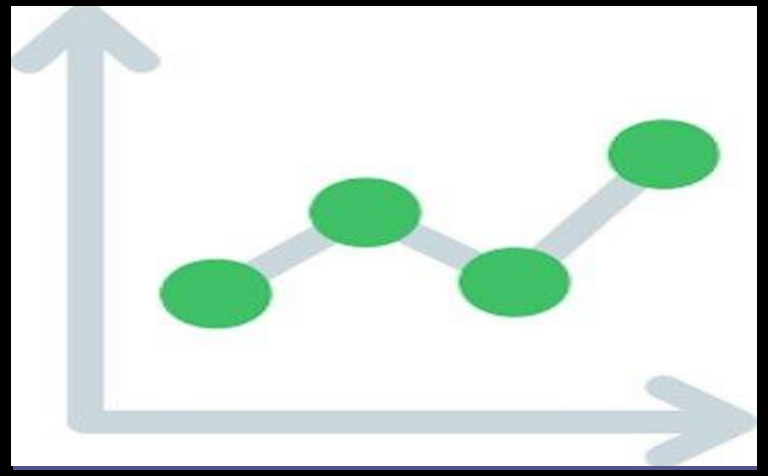
05

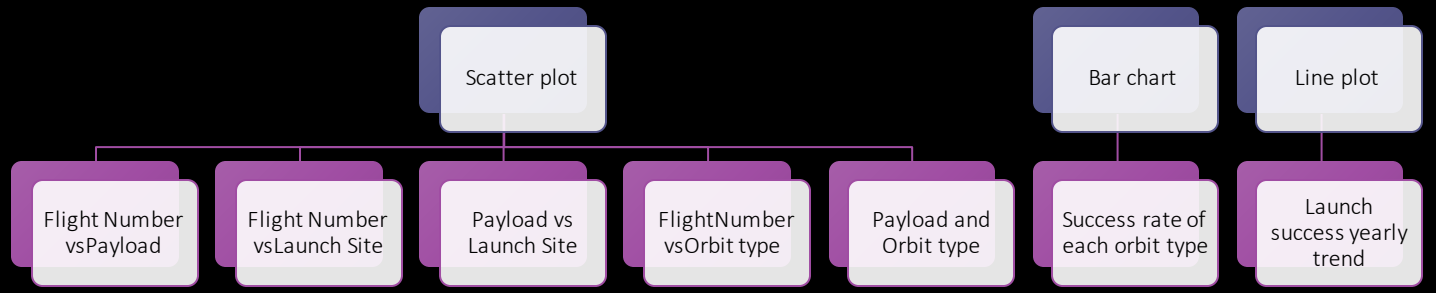
Export dataframe as .CSV file

GitHub URL to notebook



EDA with Data Visualization





GitHub URL to notebook

EDA with SQL

- 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 achieved
- 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 URL to notebook

Build an Interactive Map with Folium

1

Start creating an interactive map by passing the location longitude and latitude.

2

Mark all launch sites on the map with circle markers.

3

Mark the success/ failed launches for each site on the map with marker cluster.

4

Calculate the distances between a launch site to its proximities with Haversine's formula.

GitHub URL to notebook

Pie chart is included to allow user to select to show total success of all sites or a specific site

Scatter chart shows the relationship between the pay load mass and the success rate of all sites or a specific site

The dashboard is hosted by PythonAnywhere on web. A link to the website will be timkong.pythonanywhere.com

Build a Dashboard with Plotly Dash

GitHub URL to notebook

Predictive Analysis (Classification)

Build	Load data as X and Y variables with Y being converted to NumPy array while X being standardized. Split data into train and test data sets Create classifier object then create a GridSearchCV object. Fit the object to find the best parameters from the input parameters.
Evaluate	Check accuracy for each classifier Visualize on Confusion Matrix
Improve	Feature engineering Different tuning algorithm
Find	Find classifier with best score and parameters

GitHub URL to notebook

EXPLORATORY DATA ANALYSIS RESULTS

INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

PREDICTIVE ANALYSIS RESULTS

Results

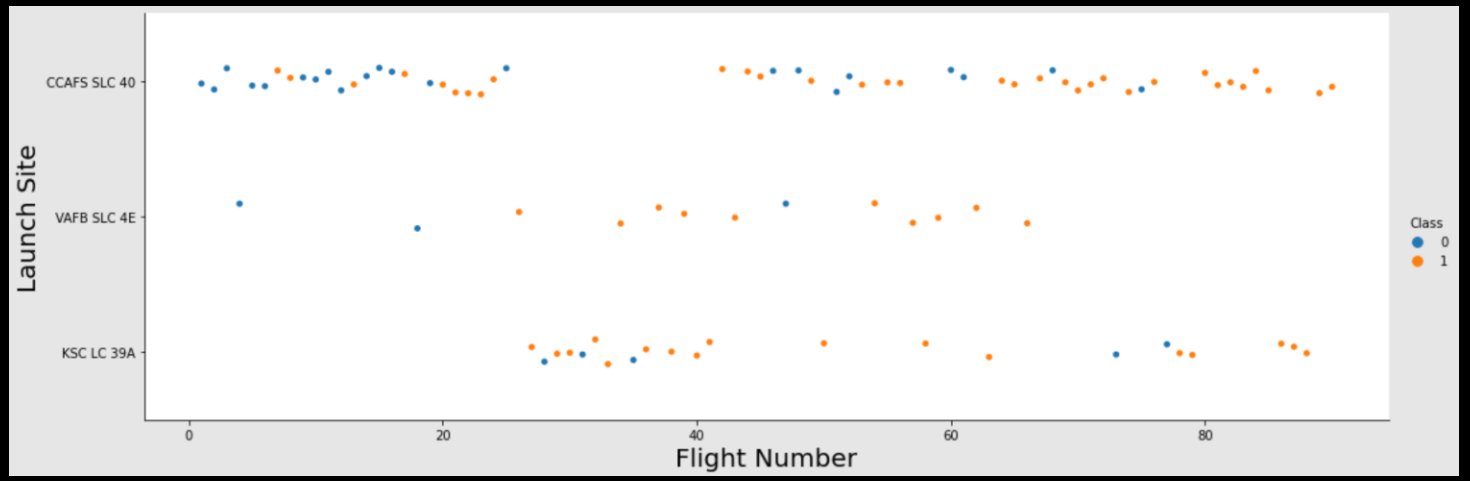
Section 2

Insights drawn from EDA

Click to add text

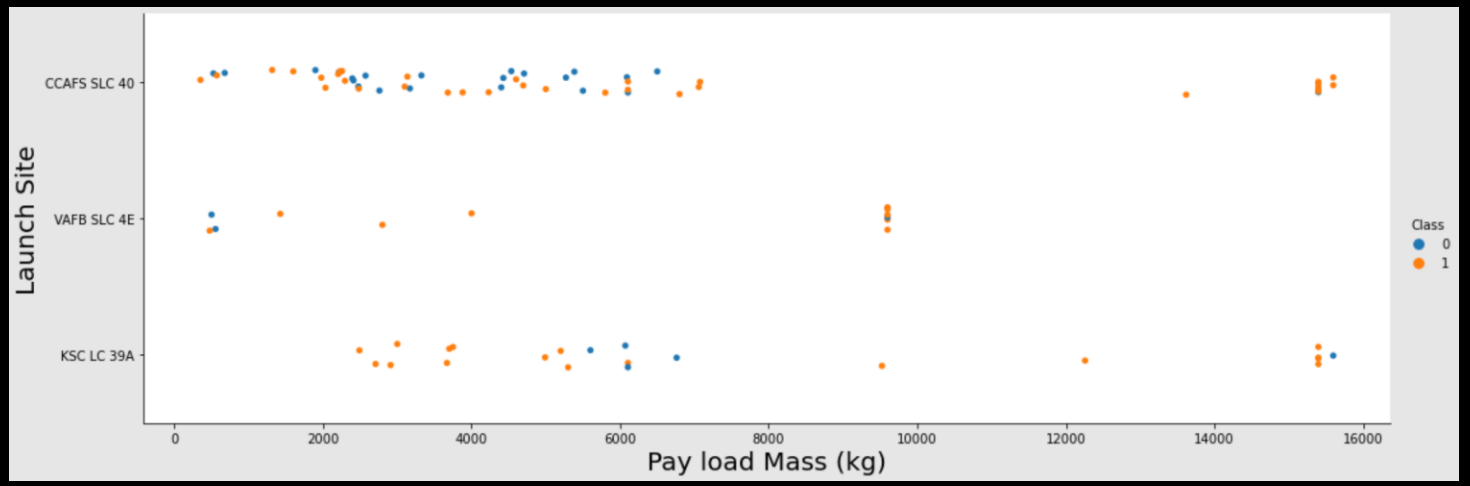
EDA with visualization

Flight Number vs. Launch Site



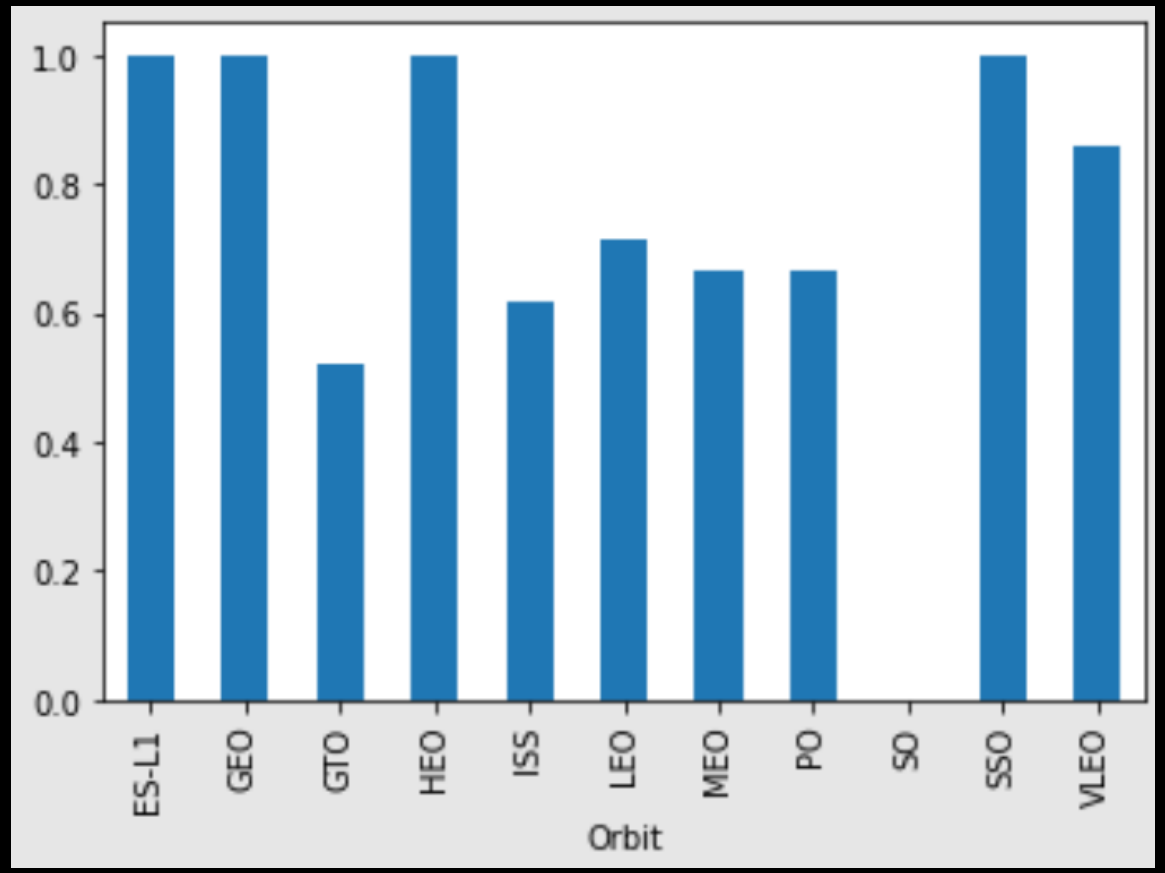
There's generally more success as the flight number increases

Payload vs. Launch Site



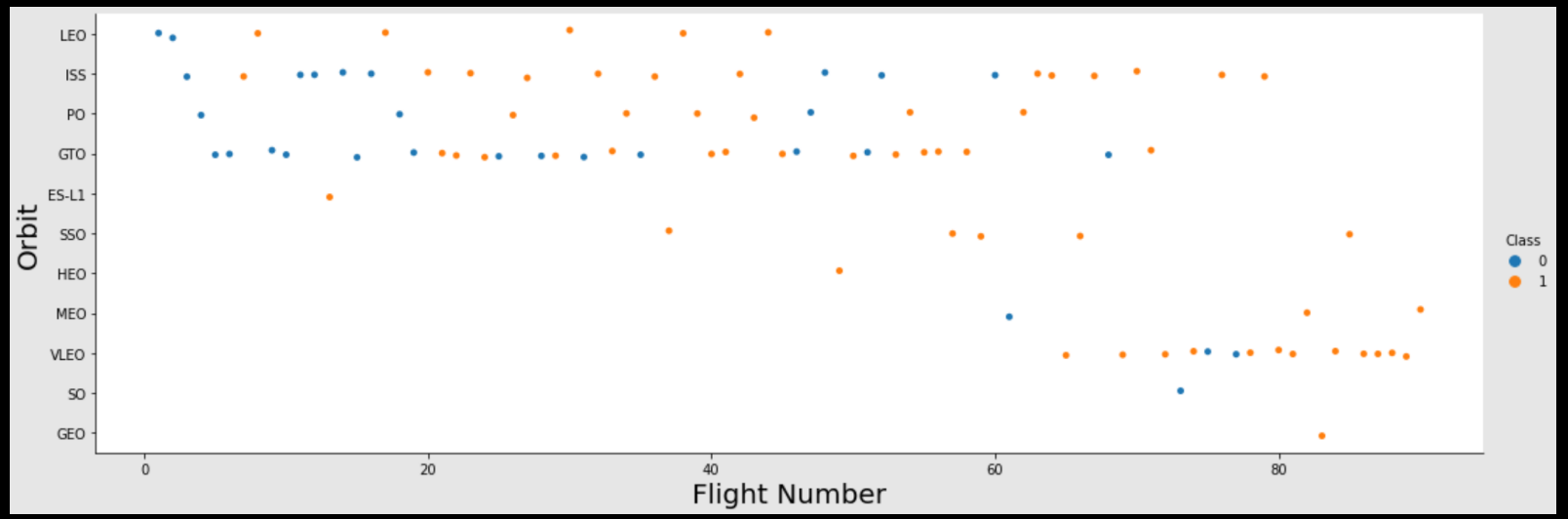
High success rate when payload is greater than 8000 kg

Success Rate vs. Orbit Type



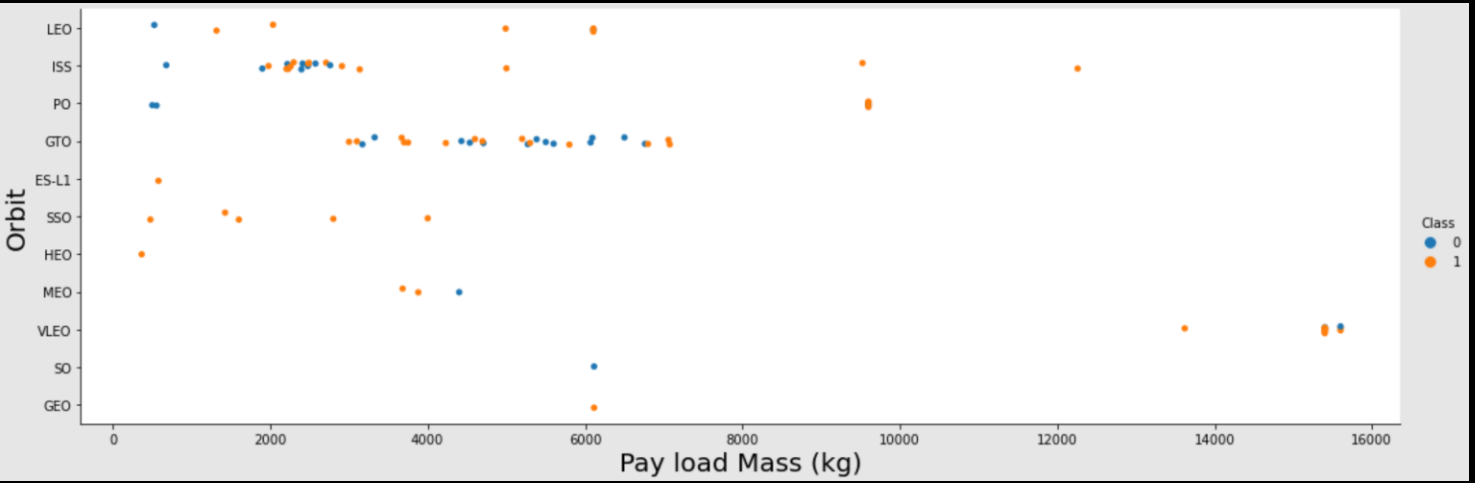
Orbit ES-L1, GEO, HEO, SSO have the highest success rate

Flight Number vs. Orbit Type

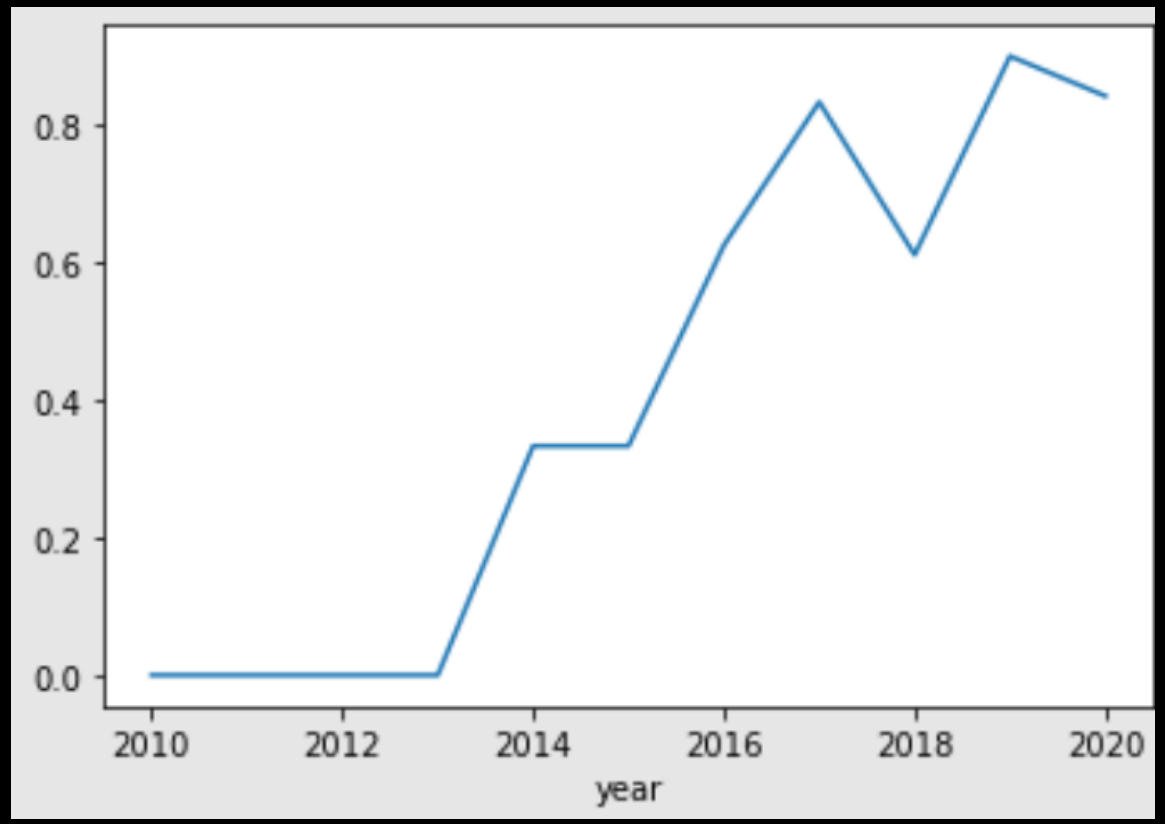


Leo has increasing success rate as the flight number increases. There seems to be no relationship between GTO success rate and the flight number

Payload vs. Orbit Type



Heavy payload has negative impact on the GTO orbit and positive impact on the LEO and ISS orbits



Launch Success Yearly Trend

Succes rate keeps increasing since year 2013

EDA with SQL

All Launch Site Names

SQL Query

```
%%sql
SELECT DISTINCT(LAUNCH_SITE) FROM SPACEX
```

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

Description

With function *DISTINCT* we pull unique values of column *LAUNCH_SITE* from table *SPACEX*

Launch Site Names Begin with 'CCA'

SQL Query

```
%%sql
SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

Description

Use clause **LIMIT 5** and **LIKE** operator with **%** wild card, we fetch 5 records from table **SPACEX** that **LAUNCH_SITE** starts with '**CCA**'

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

Total Payload Mass

SQL Query

```
%%sql
select sum(PAYLOAD_MASS__KG_) as "Total payload mass by NASA (CRS)" from SPACEX
where customer = 'NASA (CRS)'
```

Description

Use function **SUM** and **WHERE** clause to calculate the total value of column **PAYLOD_MASS_KG** from table **SPACEX**

Total payload mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

SQL Query

```
%%sql
select avg(PAYLOAD_MASS__KG_) as "Average payload mass by NASA (CRS)" from SPACEX
where BOOSTER_VERSION = 'F9 v1.1'
```

Description

Use function **AVG** and **WHERE** clause to calculate the average value of column **PAYLOAD_MASS_KG** from table **SPACEX**

Average payload mass by NASA (CRS)
2928

First Successful Ground Landing Date

SQL Query

```
%%sql
select min(DATE) as " Date of first succesful landing outcome in ground pad " from SPACEX
where LANDING__OUTCOME = 'Success (ground pad)'
```

Description

Use function **MIN** and **WHERE** clause to calculate the minimum value of column **DATE** from table **SPACEX** with column **LANDING_OUTCOME** as condition.

Date of first succesful landing outcome in ground pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

```
%%sql
select BOOSTER_VERSION,PAYLOAD_MASS__KG_,LANDING__OUTCOME from SPACEX
where PAYLOAD_MASS__KG_ between 4000 and 6000
and LANDING__OUTCOME = 'Success (drone ship)'
```

booster_version	payload_mass__kg_	landing__outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Description

select *BOOSTER_VERSION* with condition as *PAYLOAD_MASS__KG_ between 4000 and 6000* and *LANDING__OUTCOME = 'Success (drone ship)'*

Total Number of Successful and Failure Mission Outcomes

SQL Query

```
%%sql
select MISSION_OUTCOME, count(*) as "count" from SPACEX
group by MISSION_OUTCOME
```

mission_outcome	count
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Description

Group data by *MISSION_OUTCOME* and get the count of each group item

Booster Carried Maximum Payload

SQL Query

```
%%sql
select BOOSTER_VERSION,PAYLOAD_MASS__KG_ from SPACEX
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEX)
```

Description

By using sub quires, select *BOOSTER_VERSION* with the max *PAYLOD_MASS_KG*

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Record

SQL Query

```
%%sql
select LANDING__OUTCOME as "Failed droneship landing in 2015", BOOSTER_VERSION, LAUNCH_SITE from SPACEX
where LANDING__OUTCOME = 'Failure (drone ship)' and year(DATE) = '2015'
```

Failed droneship landing in 2015	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Description

Select *LANDING_OUTCOME* with condition as *LANDING__OUTCOME = 'Failure (drone ship)'* and *year(DATE) = '2015'*

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

SQL Query

```
%%sql
select LANDING__OUTCOME,count(*) as count from SPACEX
where DATE between '2010-06-04'and '2017-03-20'
group by LANDING__OUTCOME
order by count desc
```

Description

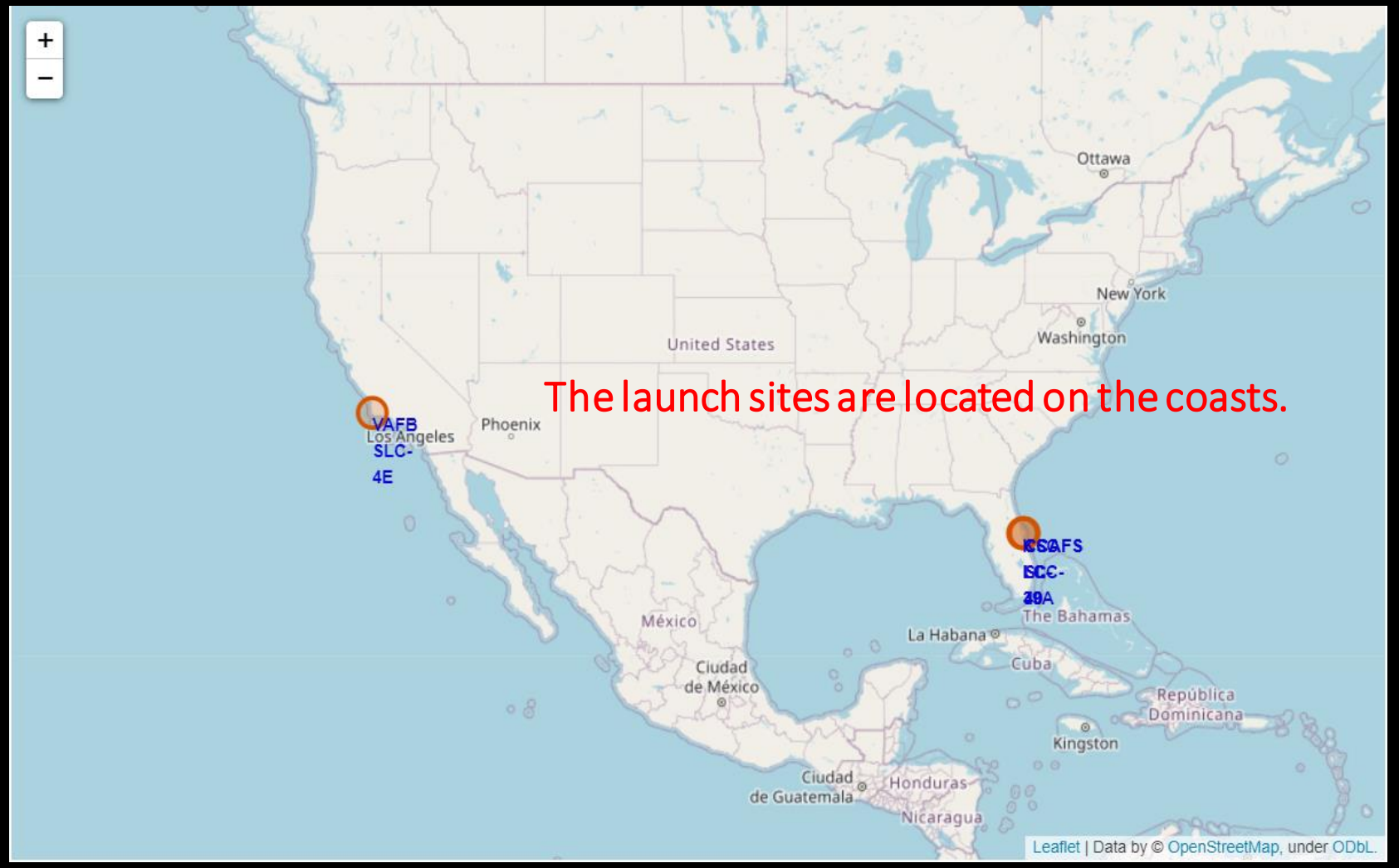
Group by and select *LANDING_OUTCOME* where condition being *DATE between '2010-06-04'and '2017-03-20'*.
Display the data in descending order with *COUNT DEC*

landing__outcome	COUNT
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

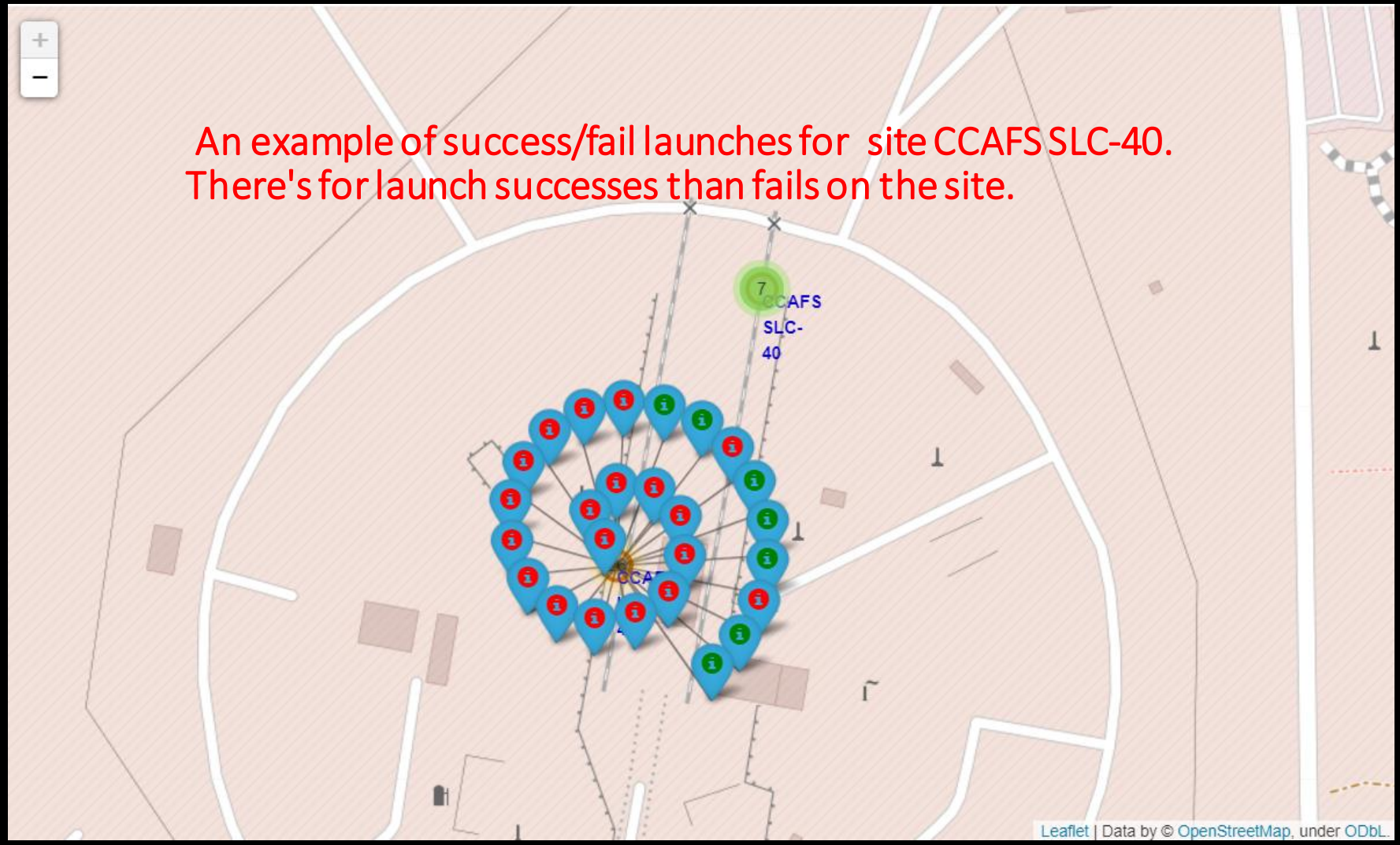
Section 4

Launch Sites Proximities Analysis

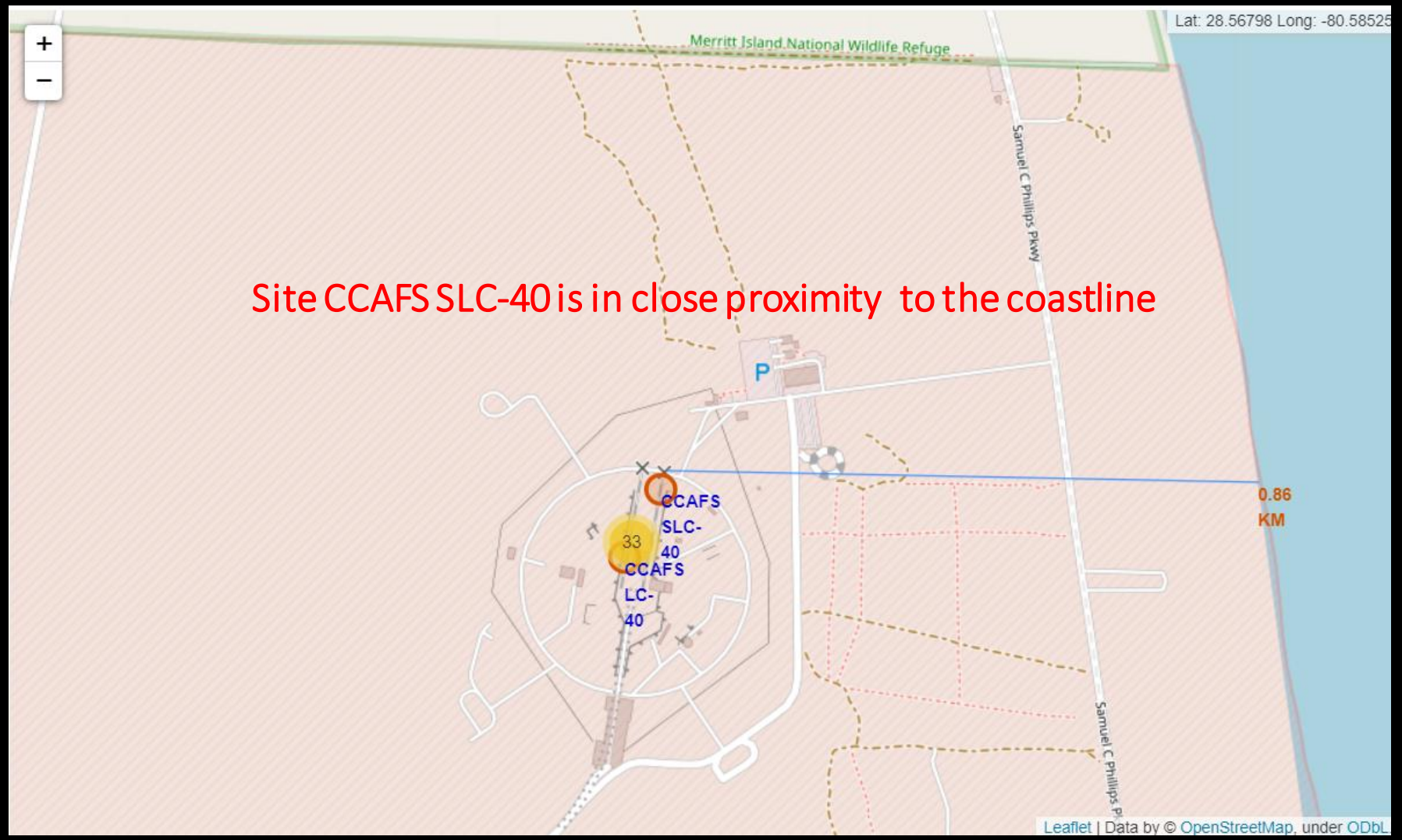
Mark all Launch Sites

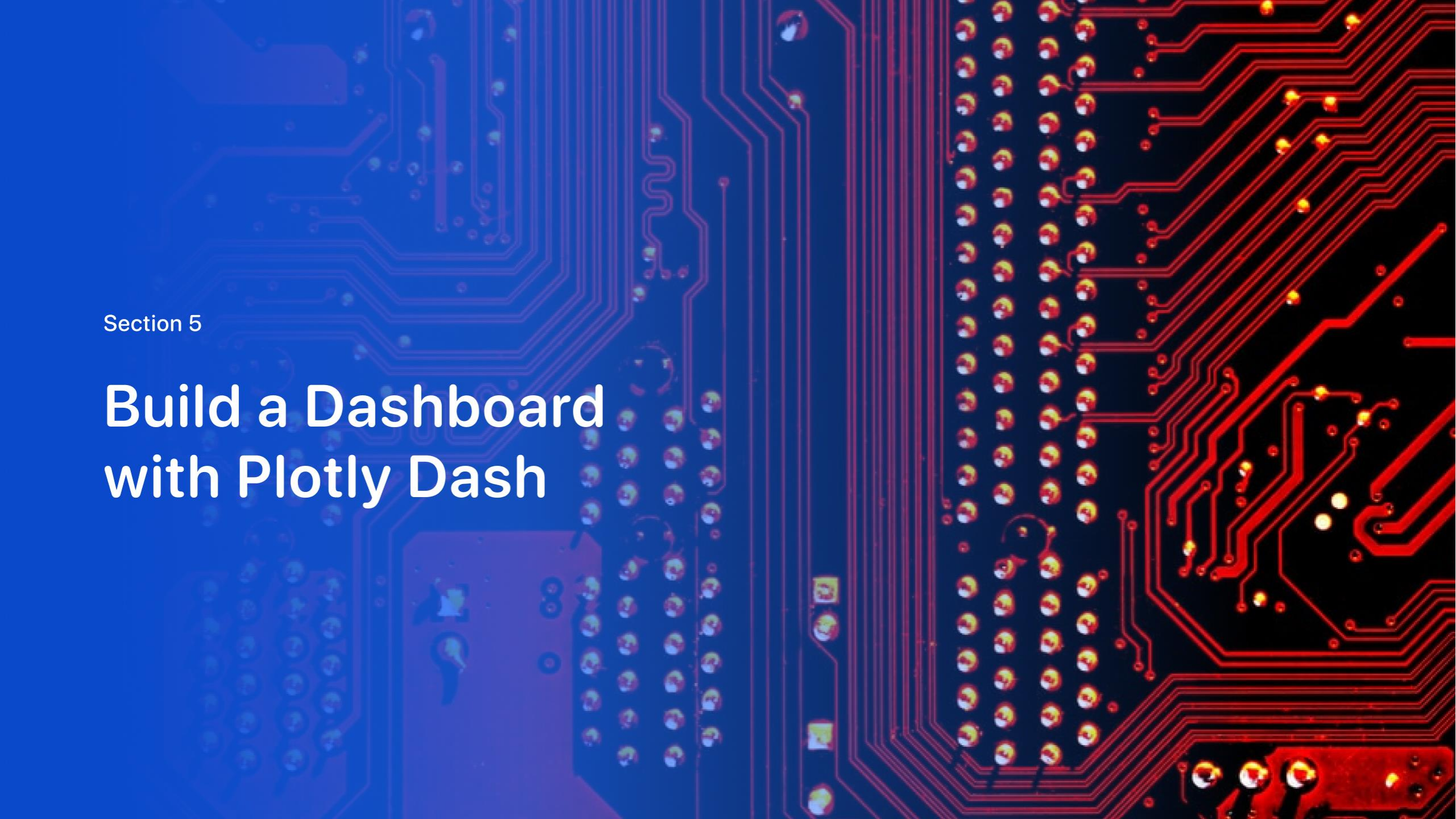


Mark Success/Fail Launches for Each Site



Calculate Distances between A Launch Site to Its Proximities

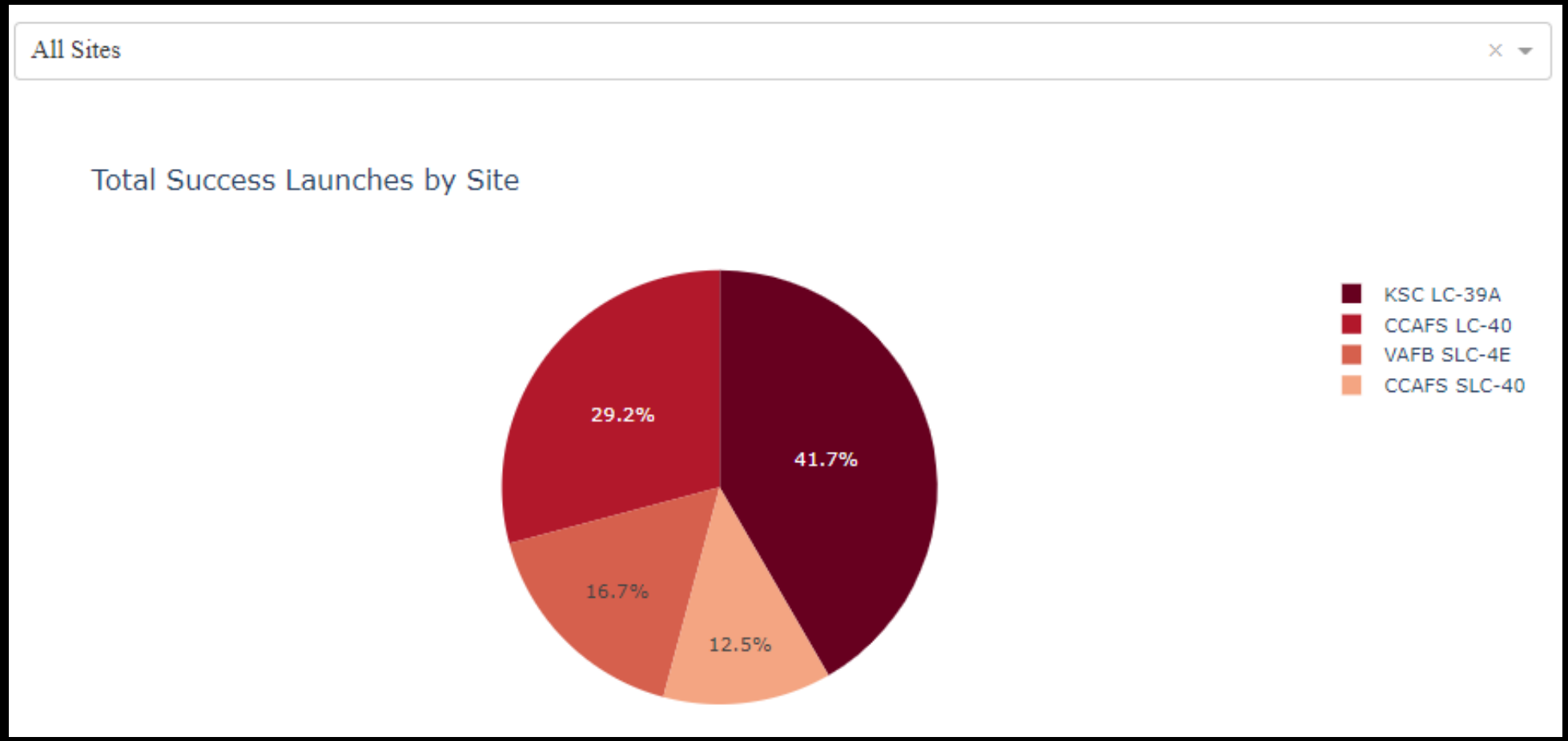




Section 5

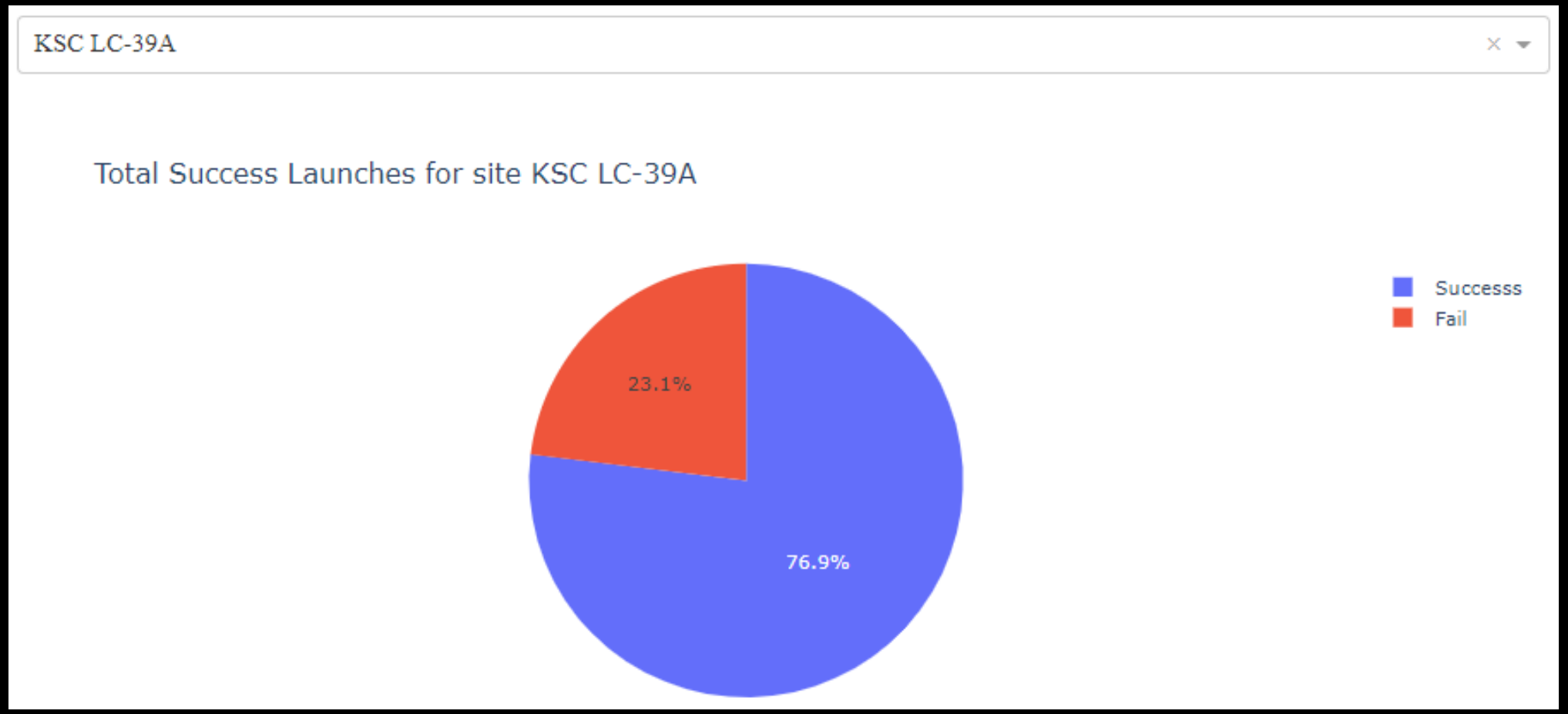
Build a Dashboard with Plotly Dash

Total Success Launches by All Sites



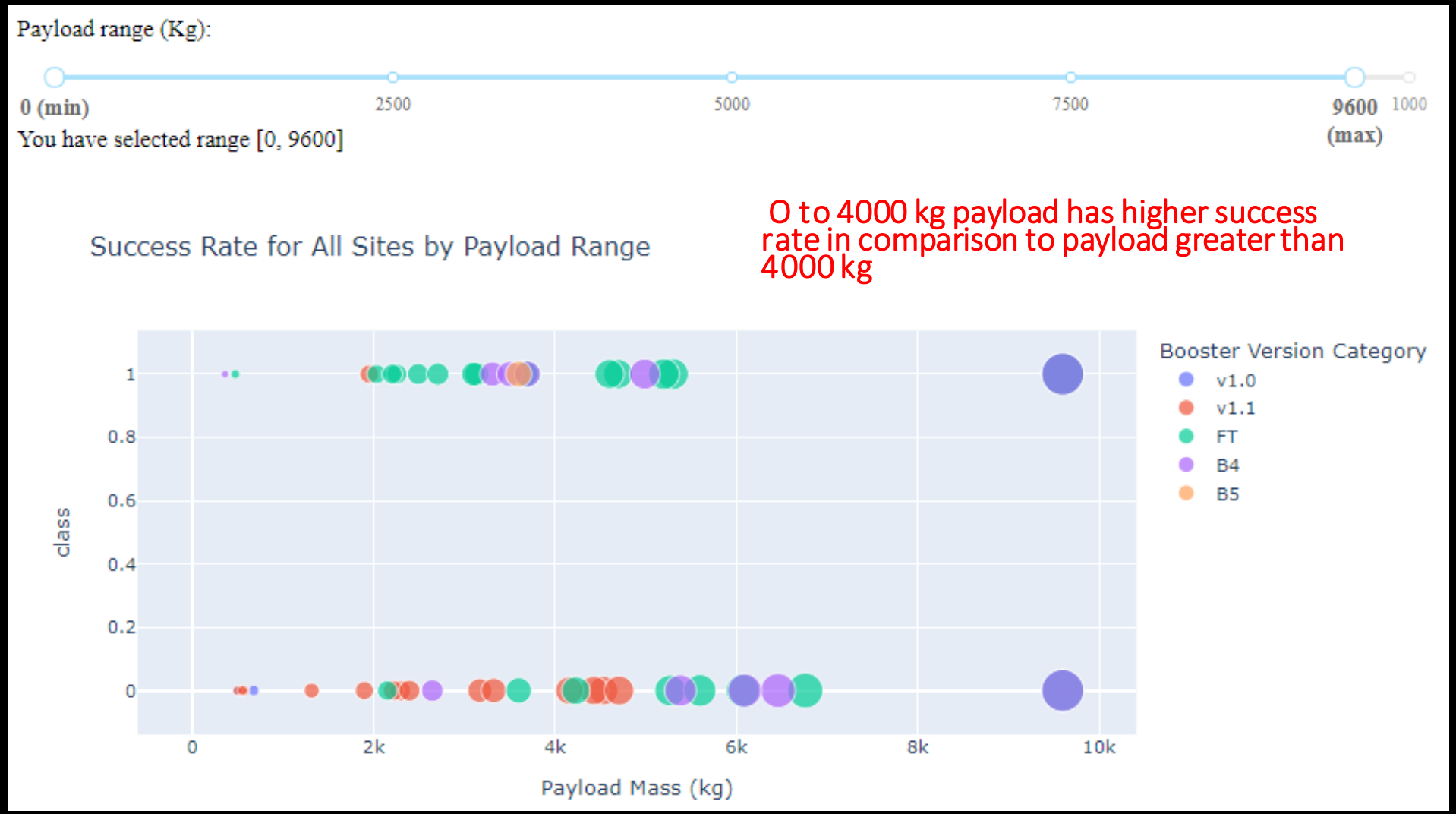
Site KSC LC-39A has the highest number of success launches

Launch Site with Highest Succes Rate



Site KSC LC-39A has 76.9% of launch success rate

Effect of Payload on Succes Rate



Section 6

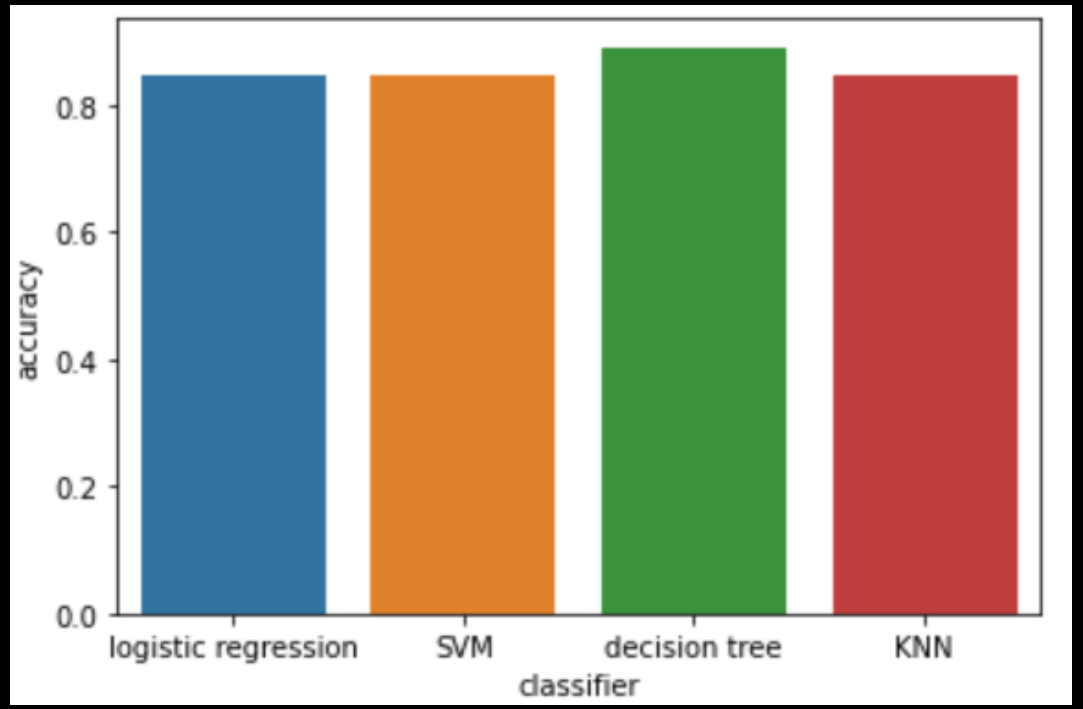
Predictive Analysis (Classification)

Classification Accuracy

All four models output very similar accuracies with decision tree classifier being the best classier as it achieves 0.89% of accuracy.

After the model is trained on test data, all four models achieved a 0.833% accuracy

	classifier	best parameter	accuracy	accuracy on test data
0	logistic regression	{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}	0.846429	0.833333
1	SVM	{'C': 1.0, 'gamma': 0.03162277660168379, 'kern...	0.848214	0.833333
2	decision tree	{'criterion': 'entropy', 'max_depth': 8, 'max_...	0.891071	0.833333
3	KNN	{'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}	0.848214	0.833333



Confusion Matrix

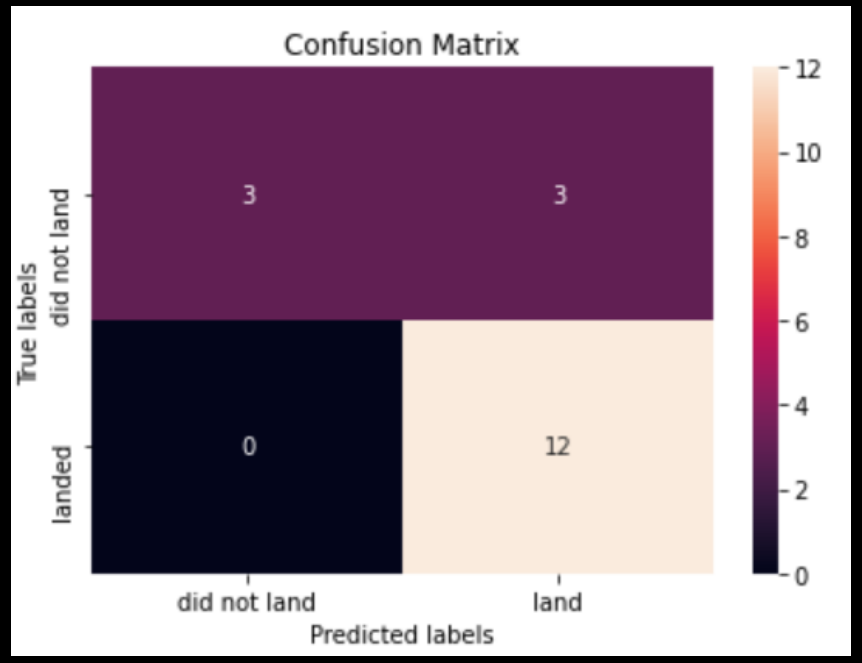
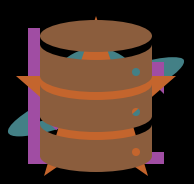


Figure on the left shows the confusion matrix of the best performing classifier - Decision Tree.

The matrix shows that the model correctly predicts 12 true successful landing and 3 true unsuccessful landing. The model also incorrectly predicts 3 false unsuccessful landing.



Conclusions

Orbit ES-L1, GEO, HEO, SSO have the highest success rate

Succes rate of SpaceX launches has been increasing with time since 2013

Site KSC LC-39A has the best launch success rate

Decision Tree classifier is the best predictive model for the provided dataset

Appendix

INTRACTIVE DASHBOARD APP: PLOTLY AND DASH

INTERACTIVE MAP VISUALIZATION: FOLIUM

ONLINE WEB HOSTING SERVICE: **PYTHONANYWHERE**

Thank you!