IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Roger Webber
February, 2023

Outline

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

Executive Summary

- **Summary of methodologies**

I. Collecting the data using API and Web Scraping

II. Data Wrangling and Exploratory Data Analysis (EDA)

III. Data Visualization

IV. Interactive Visual Analytics (Folium) and Dashboard (Plotly Dash)

V. Predictive Analysis (Classification) using ML

- **Summary of all results**

➢ EDA results

➢ Interactive Analytics with screenshots

➢ Best model to predict the success of landing

Introduction

- Project background and context

- 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.
- Determining if the first stage will land, we can determine the cost of a launch.
- The main goal of the project is to evaluate the viability of the new company Space Y to compete with Space X.

- Problems we want to find answers:

1. Factors that influence the landing outcome.
2. How the relationship between rocket variables impact the landing success.
3. Find the method that performs best to predict landing success.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One-hot encoding converting outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building classification models to ensure the best results evaluation

Data Collection

The data colletion process involves the combination of API requests from SpaceX REST API and Web Scraping data from SpaceX's Wikipedia

Data Collection – SpaceX API

Github: https://github.com/rogerwcn/Applied-Data-Science-Capstone/blob/main/Data%20Collection%20API.ipynb

Requesting data from SpaceX API using URL:

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

response = requests.get(spacex_url)
```

Converting the response into a json file

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

Use json_normalize meethod to convert the json result into a dataframe

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

Data Cleaning and Treatment

```
# 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 that have multiple payloads in a sing
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 feature.
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)]
```

Filter the dataframe to only include Falcon 9 launches

```
data_falcon9 = data2[data2['BoosterVersion']!='Falcon 1']

data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

Dealing with Missing Values

```
data_falcon9.isnull().sum()

# Calculate the mean value of PayloadMass column
payload_mean = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, payload_mean)
```

Export **CSV** file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

Github: https://github.com/rogerwcn/Applied-Data-Science-Capstone/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb

Request the Falcon9 Launch Wiki page from its URL

```
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)
```

Create a BeautifulSoup object from the HTML

```
soup = BeautifulSoup(response.text, 'html')
```

Extract all column/variable names from the HTML table header

```
html_tables = soup.find_all('table')
```

Extract column from header

```
tc = first_launch_table.find_all('th')
for th in tc:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
            column_names.append(name)
```

Creating a dictionary from the extracted column names

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

Creating a dataframe from the dictionary

```
df=pd.DataFrame(launch_dict)
```

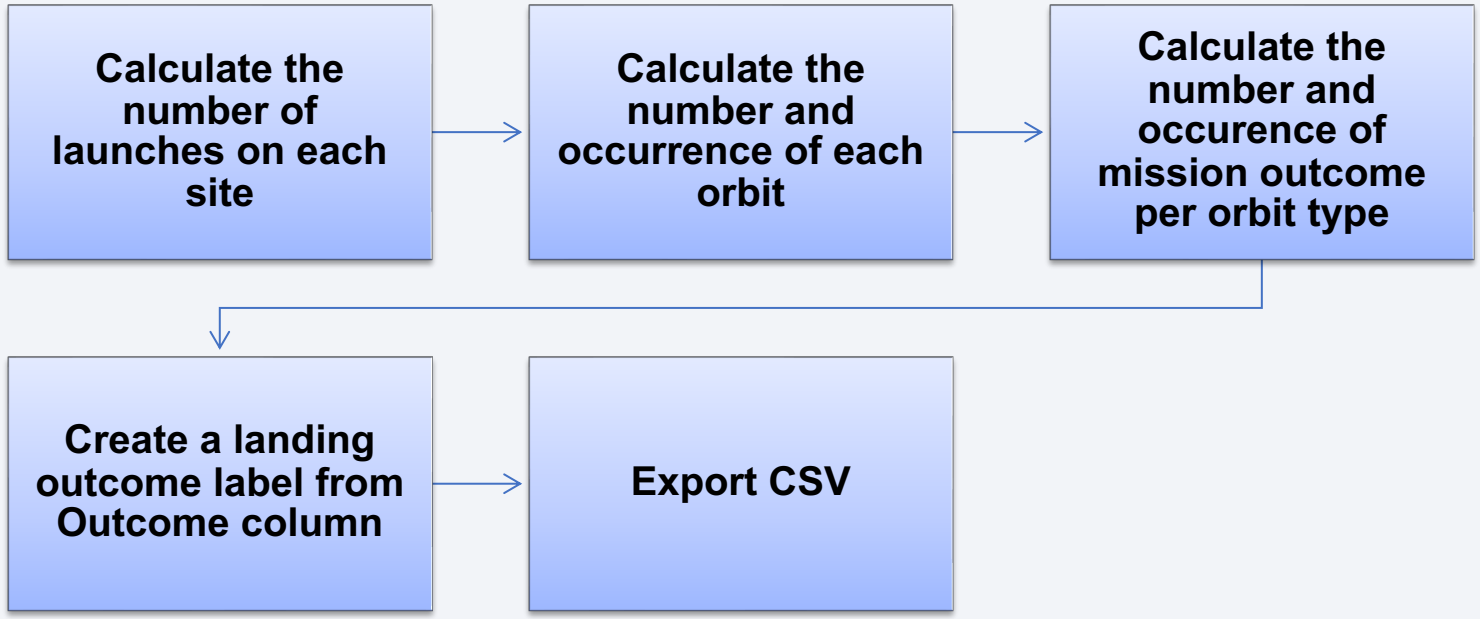
Export **CSV** file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

Github: https://github.com/rogerwcn/Applied-Data-Science-Capstone/blob/main/Data%20Wrangling_Spacex.ipynb

- There were several different cases where the booster did not land successfully.
- The data wrangling consisted to convert the outcomes into Training Labels, which 1 means the booster successfully landed and 0 means it was unsuccessful.



EDA with Data Visualization

Github: https://github.com/rogerwcn/Applied-Data-Science-Capstone/blob/main/EDA_with_Vizualization.ipynb

The following chats were plotted

- Payload Mass & Flight Number
- Flight Number & Launch Site
- Payload Mass & Launch Site
- Flight Number & Orbit Type
- Payload & Orbit

- Scatter plots are an essential type of data visualization that shows relationships between variables.
- Vertical bar charts are useful to compare different categorical or discrete variables.
- Line charts presents sequential values to help you identify trends and make predictions.

EDA with SQL

Github: https://github.com/rogerwcn/Applied-Data-Science-Capstone/blob/main/EDA%20with%20SQL.ipynb

The following SQL queries were performed:

1. Displaying the names of the unique launch sites in the space mission
2. Displaying 5 records where launch sites begin with the string 'CCA'
3. Displaying the total payload mass carried by boosters launched by NASA (CRS)
4. Displaying average payload mass carried by booster version F9 v1.1
5. Listing the date when the first successful landing outcome in ground pad was acheived
6. Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. Listing the total number of successful and failure mission outcomes
8. Listing the names of the booster versions which have carried the maximum payload mass
9. Listing the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
10. 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

Build an Interactive Map with Folium

Github: https://github.com/rogerwcn/Applied-Data-Science-Capstone/blob/main/Interactive%20Map%20with%20Folium.ipynb

1. Markers were created for all launch records. If a launch was successful (class=1), then a green marker and if a launch was failed, we use a red marker (class=0).

2. Circles indicate highlighted areas around specific coordinates, like NASA Johnson Space Center.

3. Lines represent the distance between a launch site to its proximities such as highway, railways, coastlines, etc.

Build a Dashboard with Plotly Dash

Github: https://github.com/rogerwcn/Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py

An interactive dashboard with Plotly dash was built

Launch Site Dropdown list

- Select Launch Site.

Pie charts

- Show the total successful launches count for all sites and the success vs. Failed counts for the site, if a specific Launch Site was selected.

Slider

- Select Payload range.

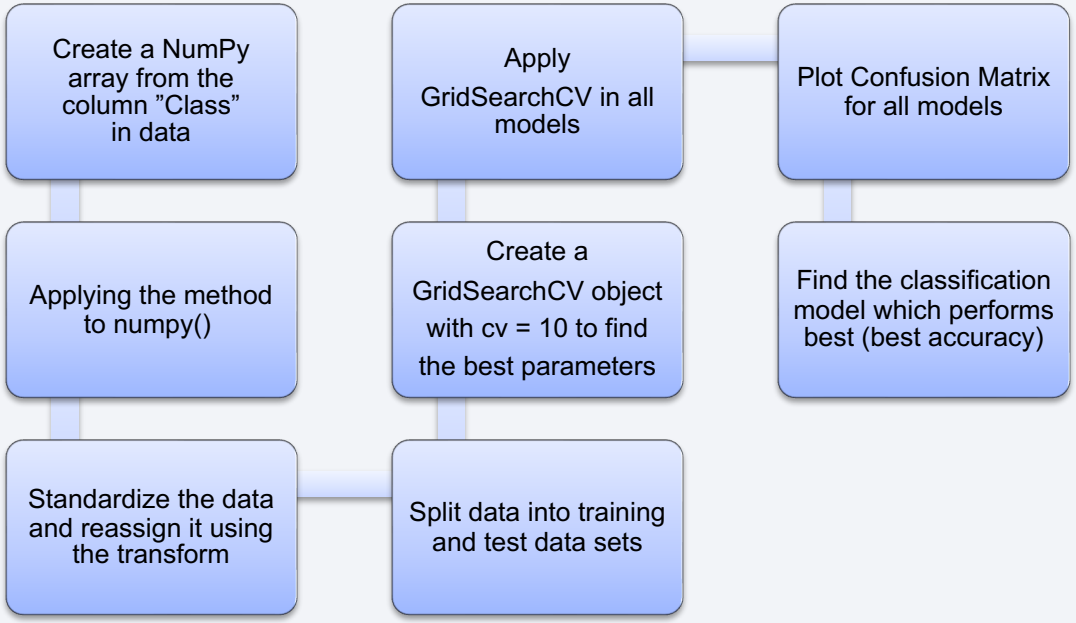
Scatter Chart

- Show the correlation between Payload and Launch Success.

Predictive Analysis (Classification)

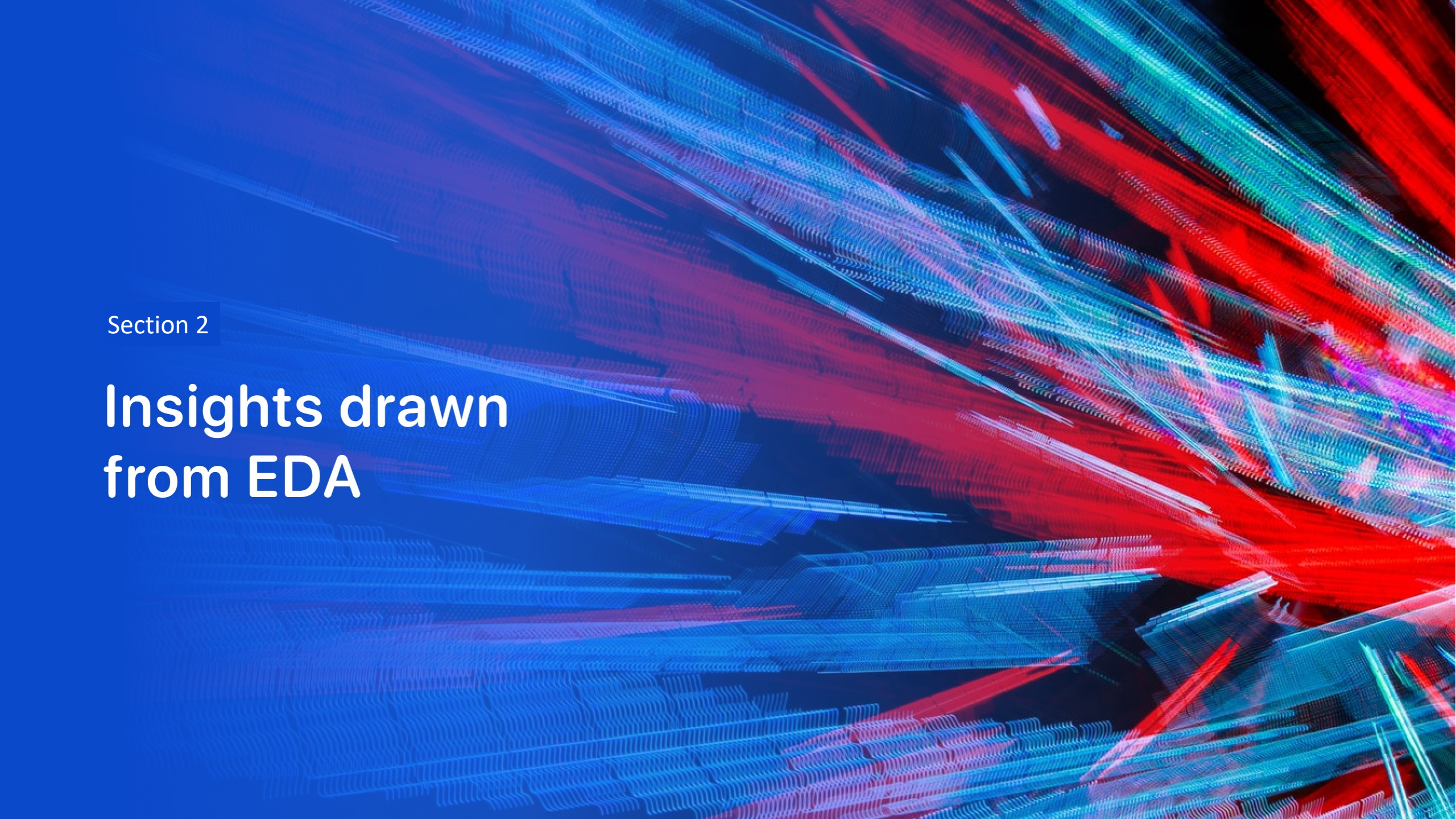
Github: https://github.com/rogerwcn/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine_Learning_Prediction.jupyterlite.ipynb

- The predictive analysis was performed comparing four classifications models: (i) logistic regression; (ii) support vector machine (SVC); (iii) Decision Tree; K nearest neighbors (KNN)
- To find the best model, the following steps were accomplished:

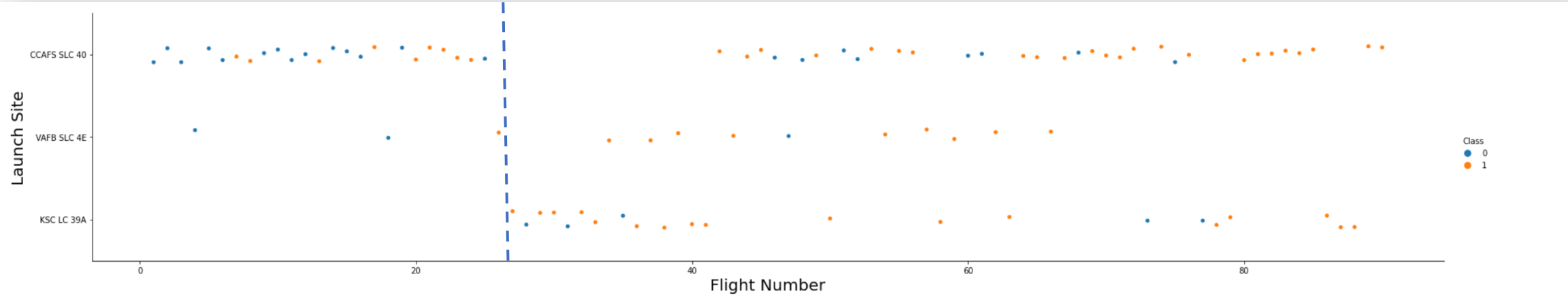


Results

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

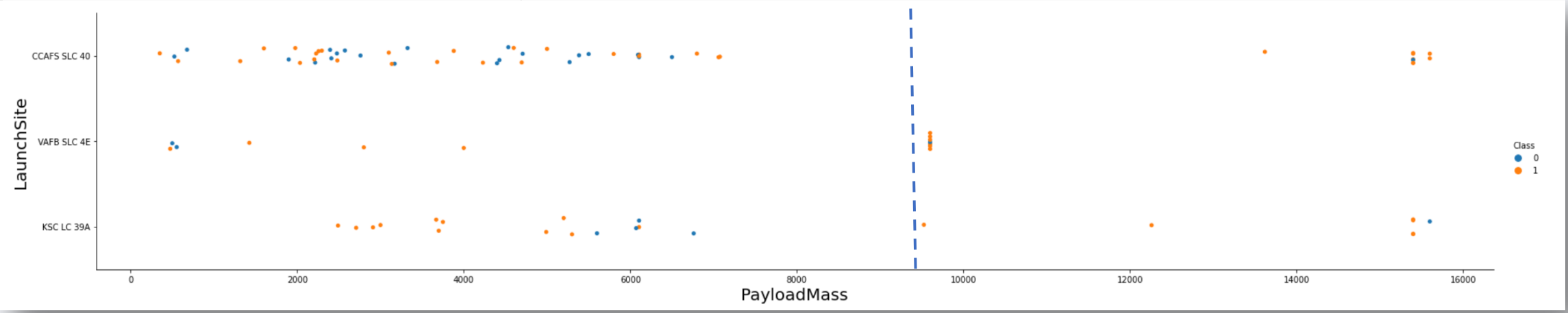

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



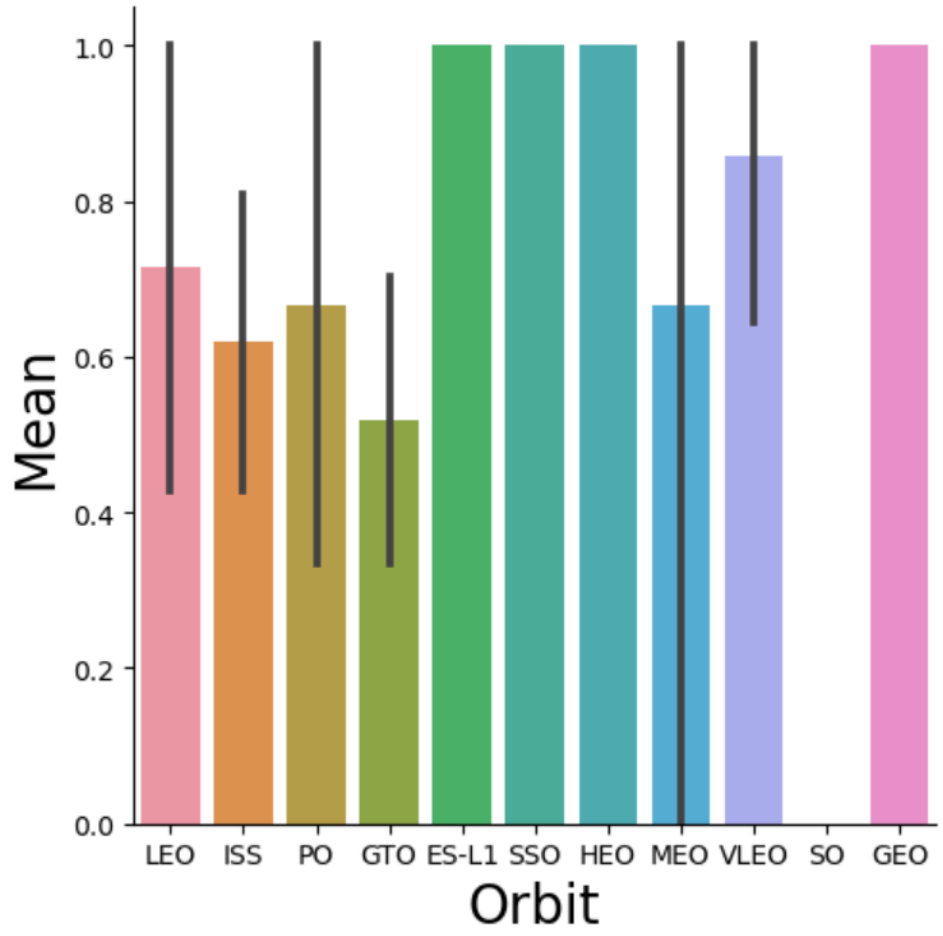
- The scatter plot shows Launch success increases as flight number increases. This means that more flights the more experienced Space X got and success rate got better.
- CCAFS seems to be the best launch site

Payload vs. Launch Site



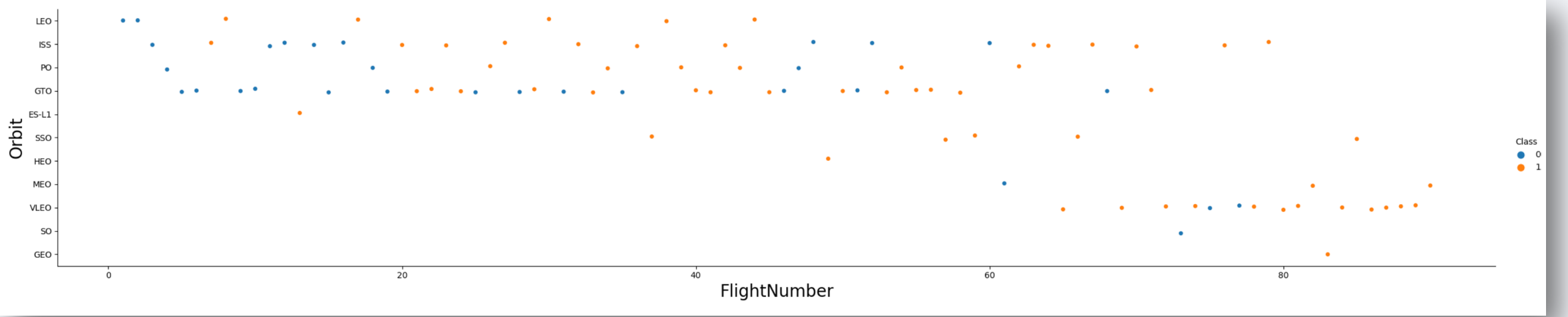
- There is a high unsuccessful rate of landings when Payloads are between 2000-7000 kg.
- From 9000 payload (dashed line) and on there is good success of rate
- VAFV SLC 4E seems not support very high (over 10000) Payloads

Success Rate vs. Orbit Type



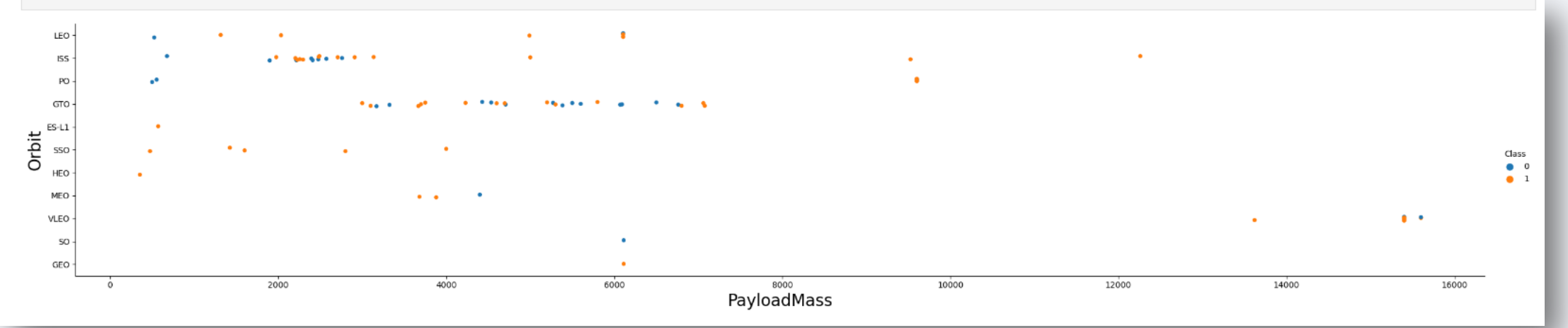
- Orbits ES-L1, SSO, HEO and GEO have 100% success rate.
- Orbits LEO, ISS, PO, GTO, MEO and VLEO have success rate between 85-50% .
- SO is the only one with 0% success rate.

Flight Number vs. Orbit Type



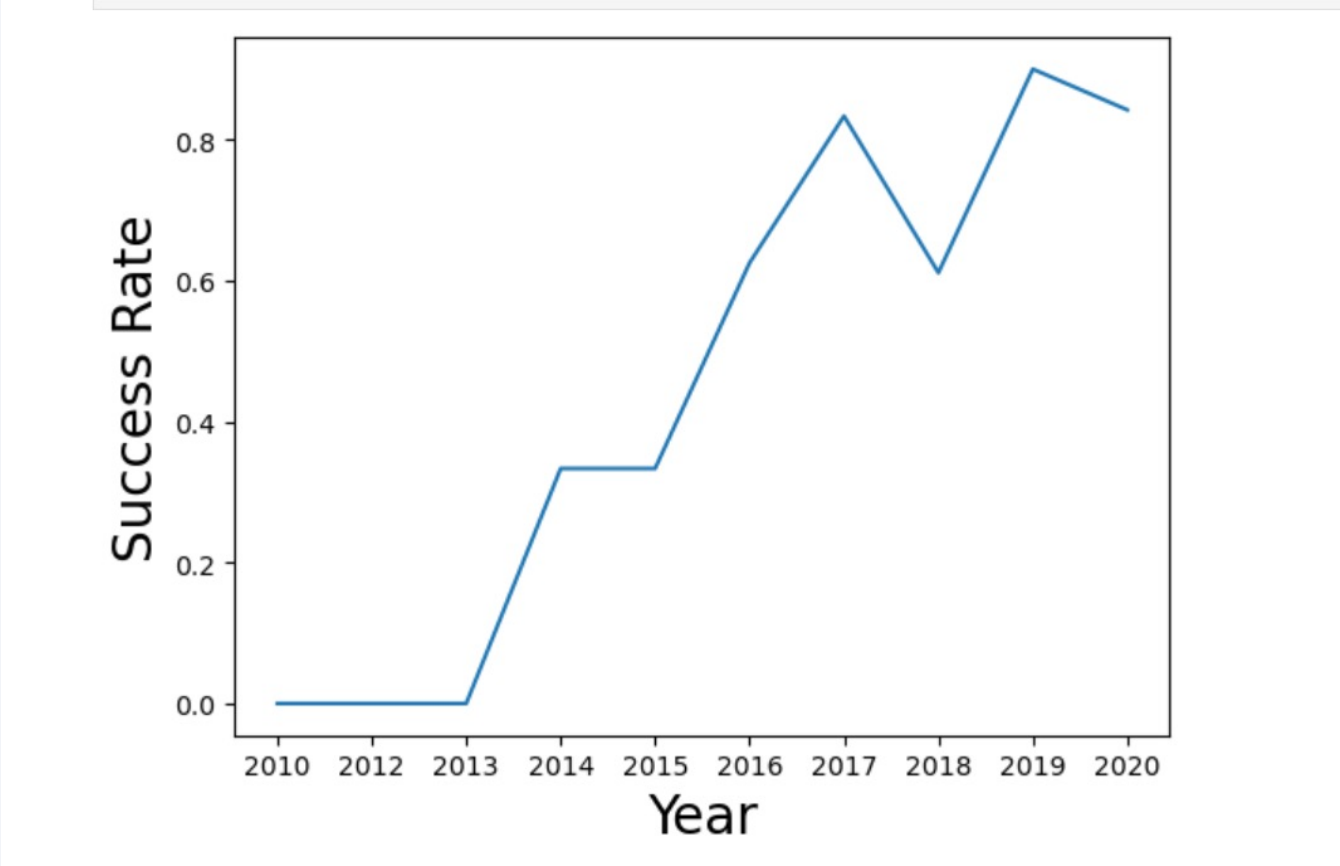
- There is a high unsuccessful rate of landings when Payloads are between 2000-7000 kg.
- From 9000 payload (dashed line) and on there is good success of rate
- VAFV SLC 4E seems not support very high (over 10000) Payloads

Payload vs. Orbit Type



- Higher payloads are assigned to VLEO
- GTO has all launches concentrated between 3000-7000 kg Payloads. However, there is no clear relationship between rate of success and Payloads.
- SO and GO have only 1 launch each.

Launch Success Yearly Trend



- There is positive trend of success rate since 2013 til 2020, even though 2017 and 2019 showed a slight dip.

All Launch Site Names

- There are four unique launch sites.
- DISTINCT statement was used to return the name as follows:

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

```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
```

```
 * ibm_db_sa://kcz11642:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5a
Done.
```

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

Launch Site Names Begin with 'CCA'

- Query to find 5 records where launch sites begin with `CCA`
- Used condition LIKE and wildcard ‘CCA%’

Display 5 records where launch sites begin with the string 'CCA'

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

```
 * ibm_db_sa://kcz11642:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/BLUDB
Done.
```

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

- Total payload carried was calculated using the function SUM and WHERE clause.

```
%sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXTBL where customer = 'NASA (CRS)';
```

```
 * ibm_db_sa://kcz11642:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/BLUDB
Done.
```

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1
- Used AVG function addressed to the PAYLOAD_MASS_KG_ column and WHERE clause to filter booster version F9 v1.1

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

```
sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://kcz11642:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/BLUDB
Done.
```

avg_payload
2928

First Successful Ground Landing Date

- Data of first successful landing outcome on ground pad
- Used MIN function and WHERE clause to filter LANDING_OUTCOME with Success (ground pad)

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

```
sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://kcz11642:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/BLUDB
Done.
```

first_success_gp
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Used DISTINCT statement and WHERE clause with PAYLOAD_MASS_KG and LANDING_OUTCOME filters

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME = 'Success (drone ship)';
```

```
 * ibm_db_sa://kcz11642:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/BLUDB
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Used GROUP BY for MISSION_OUTCOME and COUNT to find the quantity for each group.

List the total number of successful and failure mission outcomes

```
sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://kcz11642:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/BLUDB
Done.
```

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

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass

```
sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL) ORDER BY BOOSTER_VERSION;
```

```
 * ibm_db_sa://kcz11642:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/BLUDB
Done.
```

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

2015 Launch Records

- List the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE_PART('YEAR', DATE) = 2015;
```

```
 * ibm_db_sa://kcz11642:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/BLUDB
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

- 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

```
sql SELECT LANDING__OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY QTY DE
```

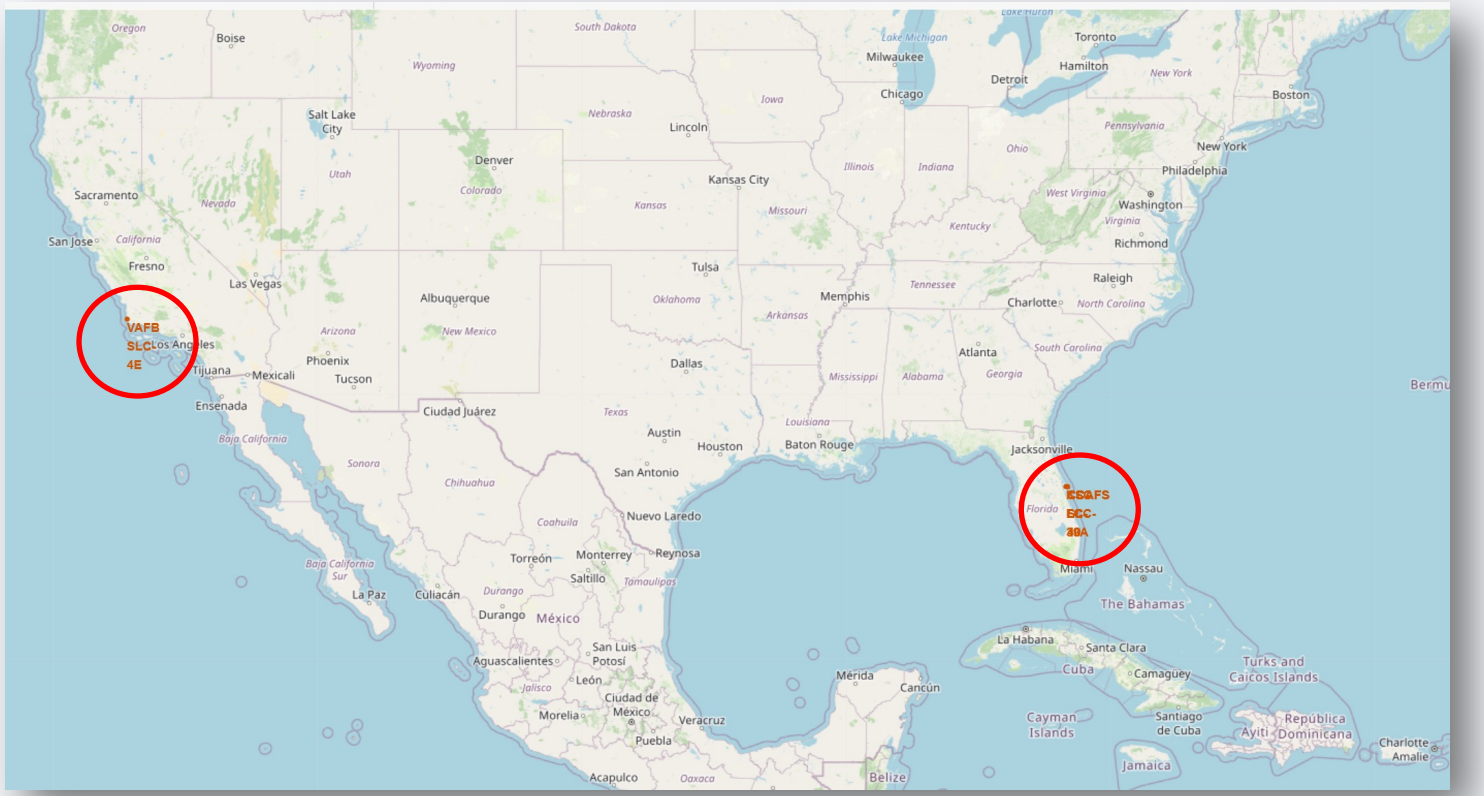
```
 * ibm_db_sa://kcz11642:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/BLUDB
Done.
```

landing__outcome	qty
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 3

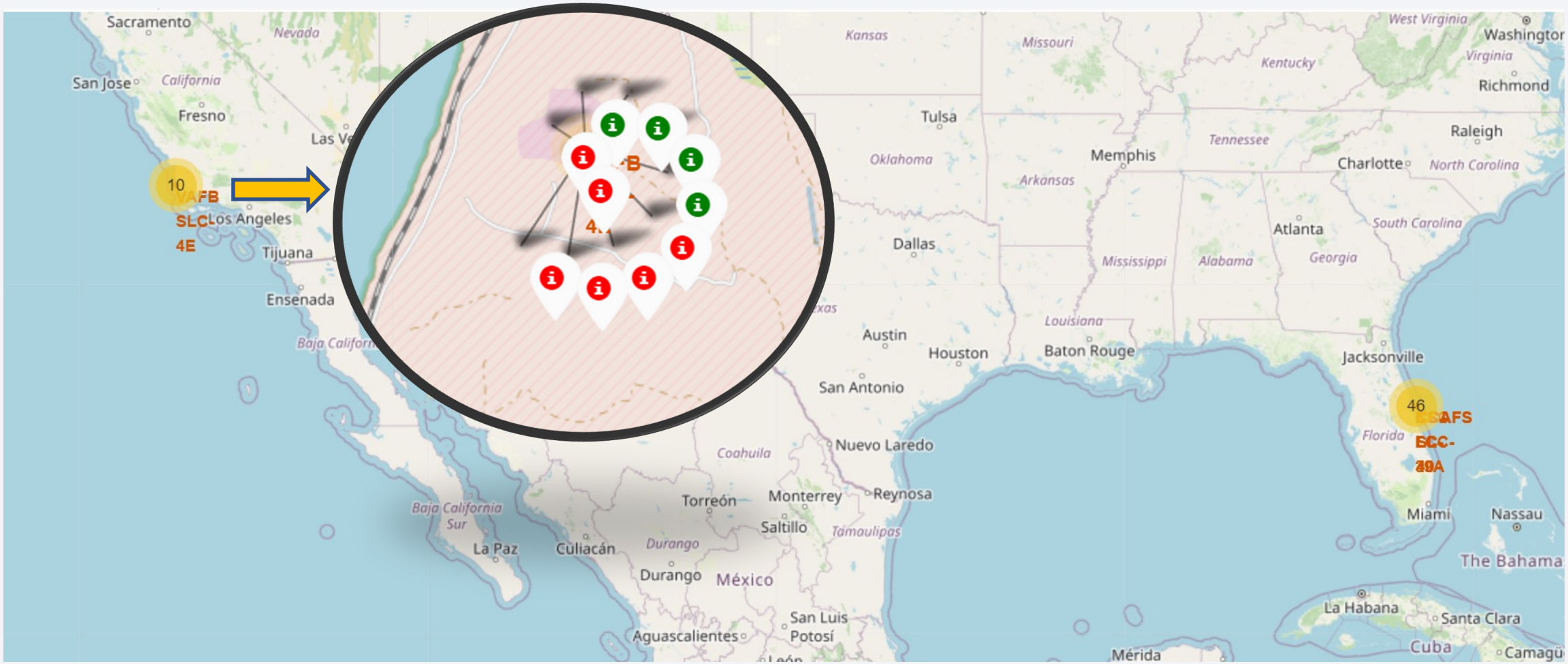
Launch Sites Proximities Analysis

Location of all Launch Sites on Folium Map

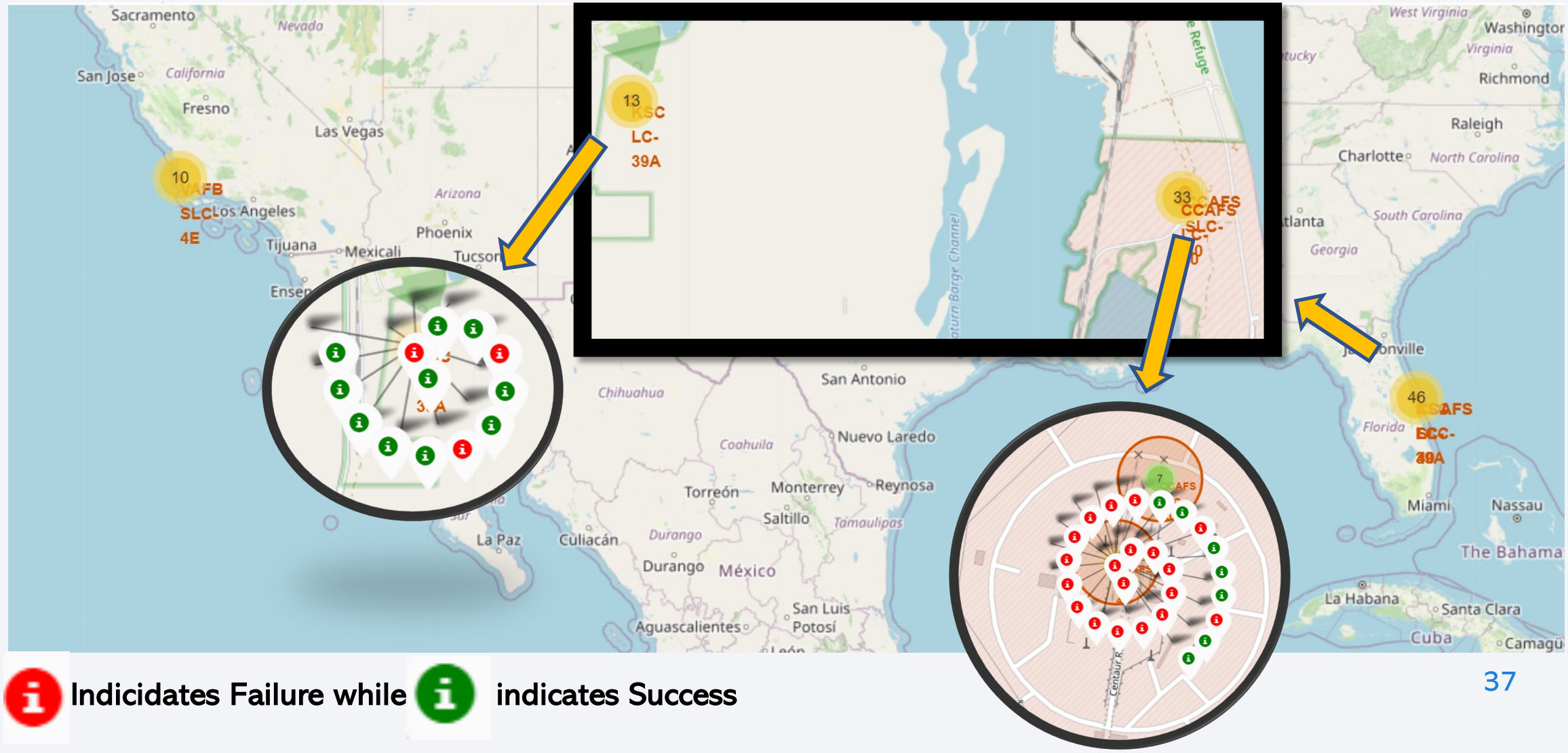


From the screenshot, it can be infer that SpaceX launch site are all located near the coast.

Launch Records per Site

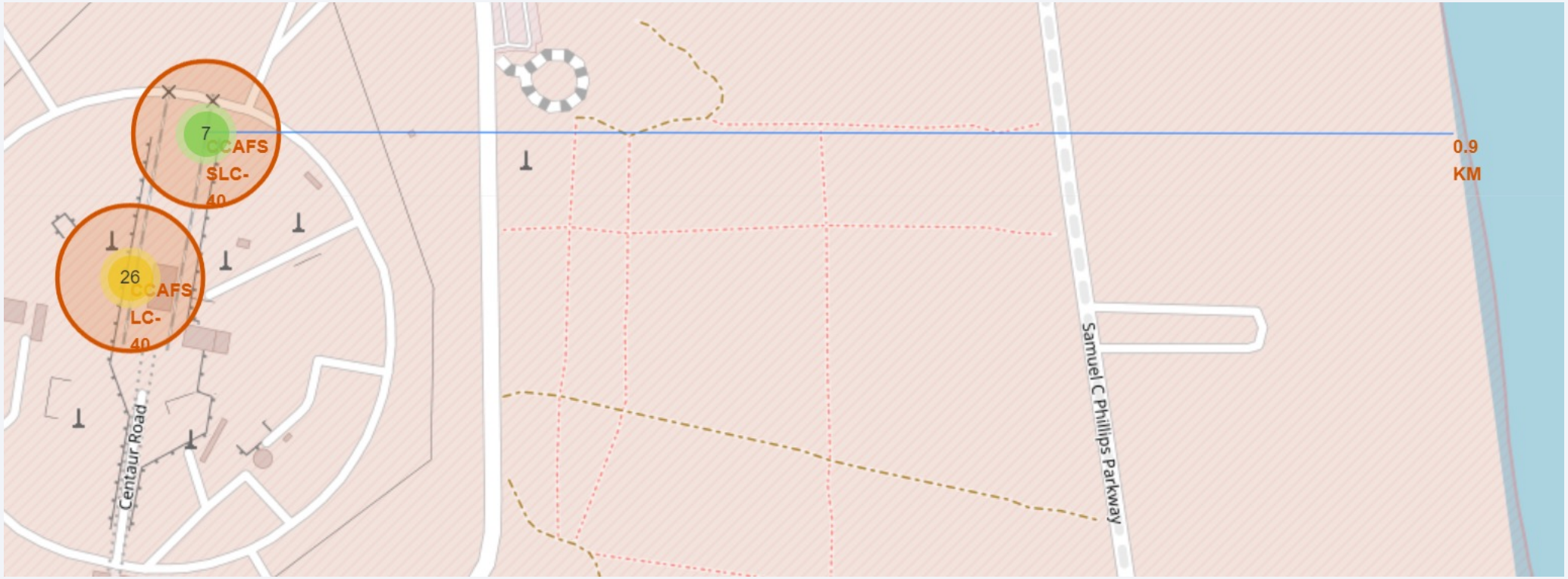


Launch Records per Site

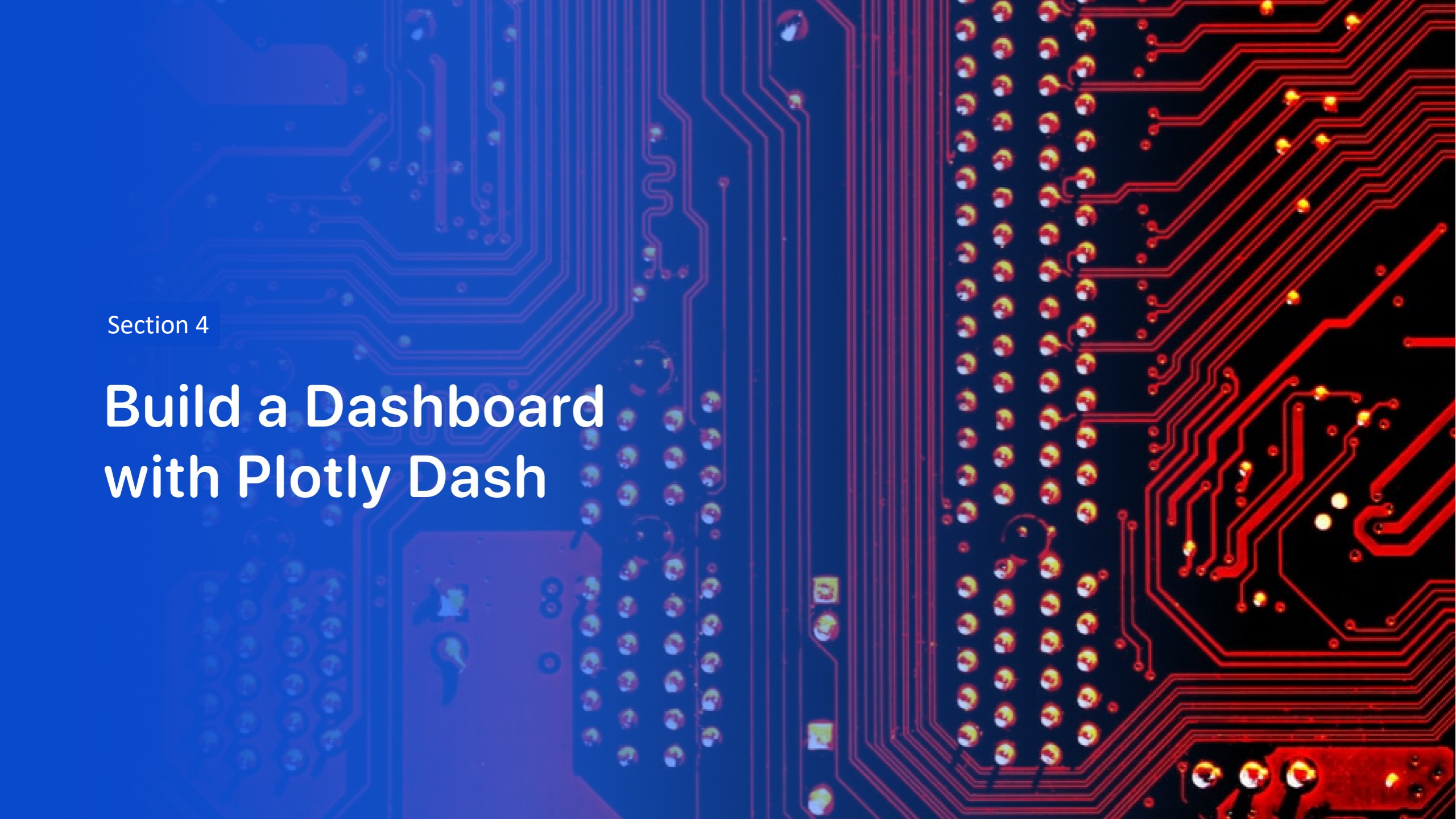


Indicidates Failure while indicates Success

Launch Sites Neighborhood



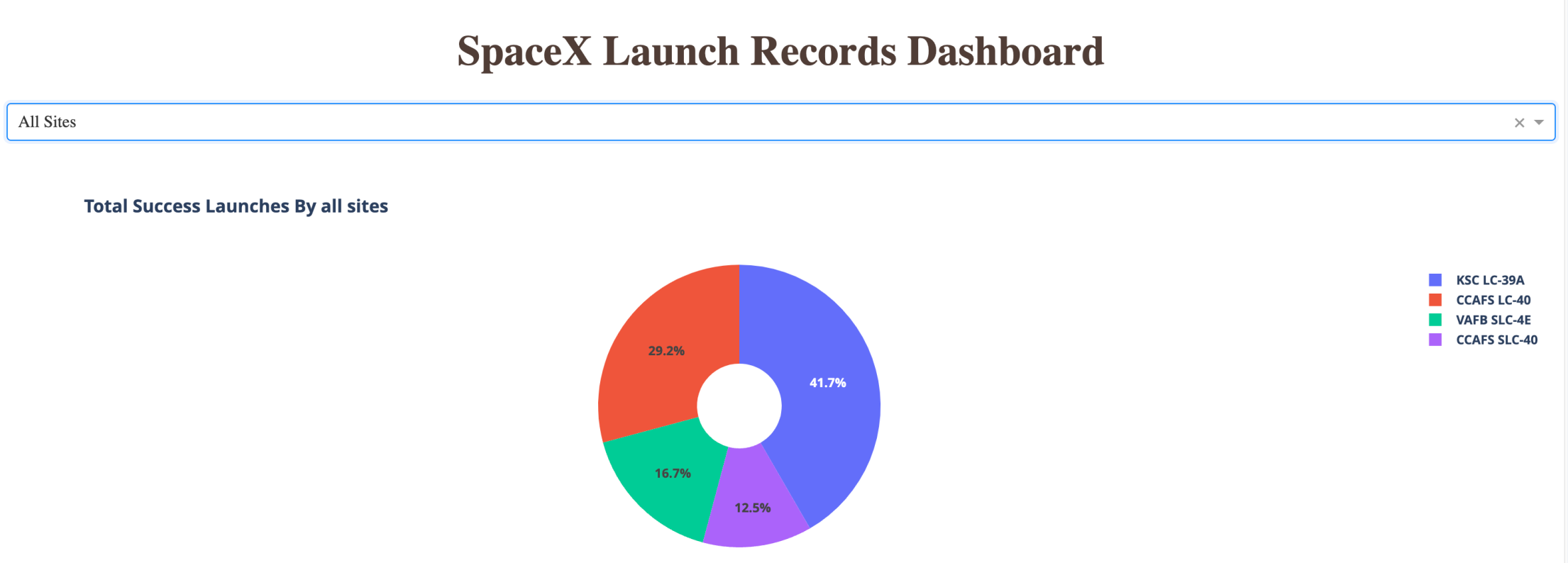
- CCAFAS is almost 1 km from the east coastline.



Section 4

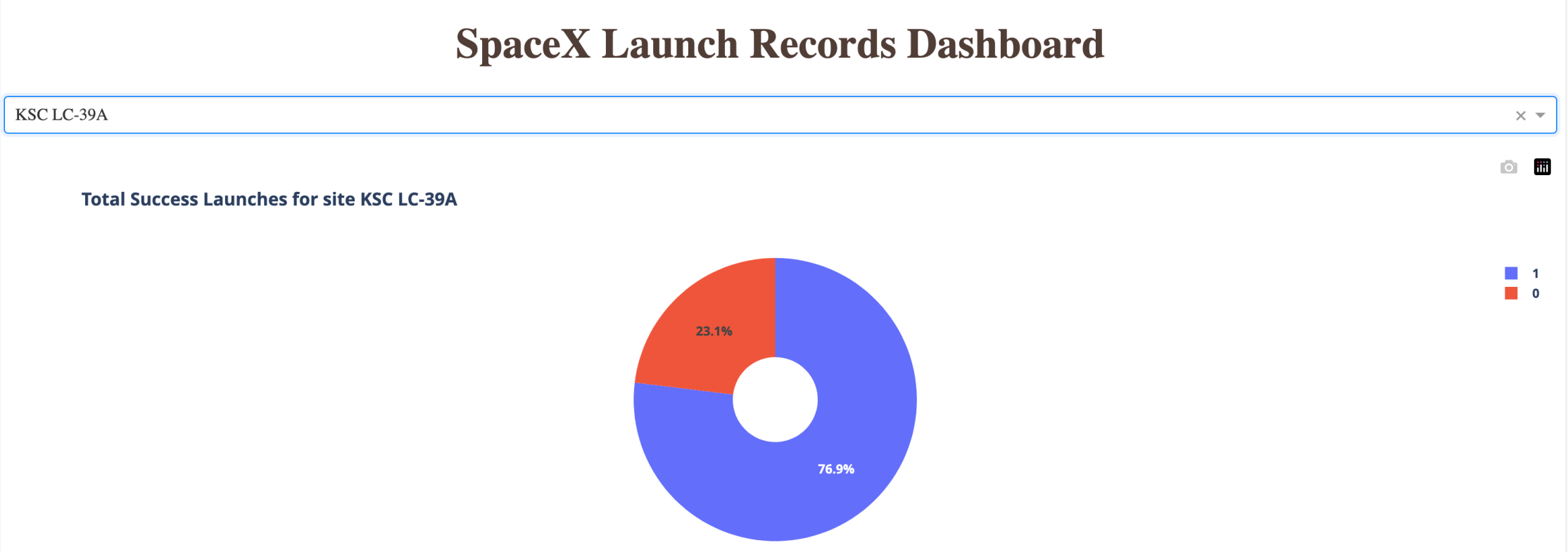
Build a Dashboard with Plotly Dash

Success Launches by all sites (piechart)



- The piechart providing the success launches is a good way to see the best location.
- Its clear that KSC LC-39A had the best outcome.

KSC LC-39A Launch Success Ratio



- KSC LC-39A has the highest success launch (76.8%).

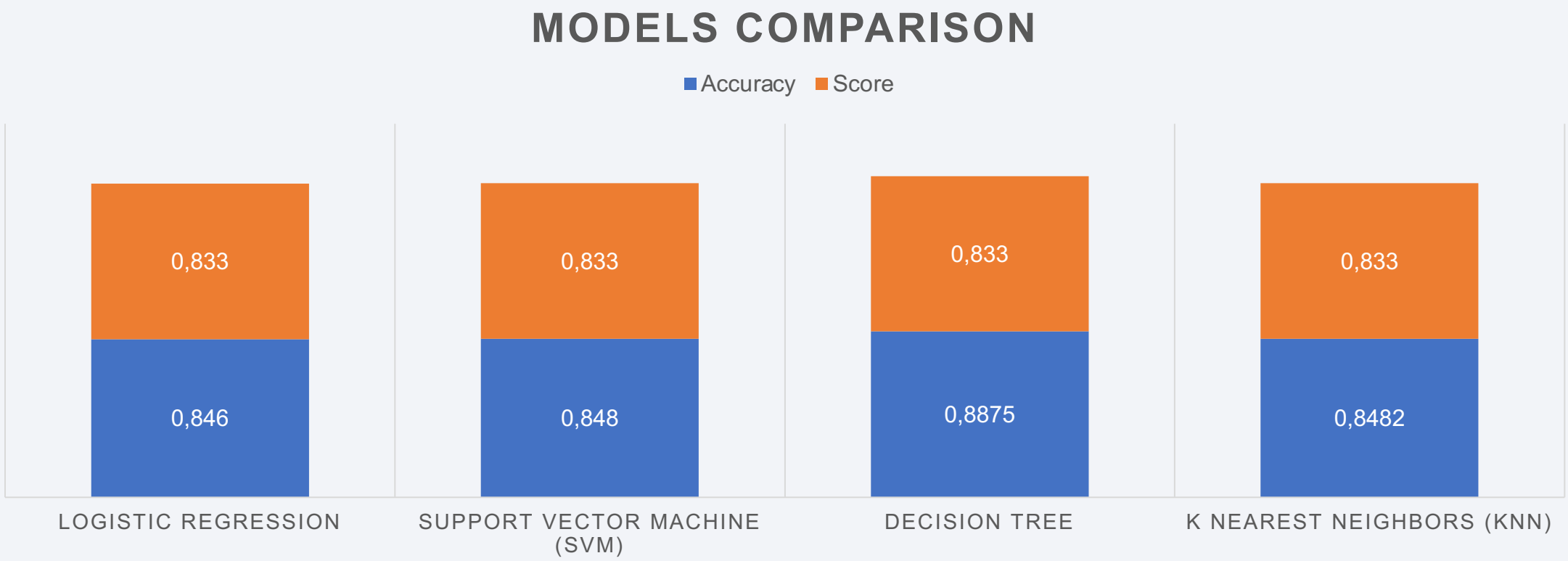
Payload vs. Launch Outcome Scatter Plot

- Comparing the 2 plots, its possible to conclude that lower Payloads (0-4000 kg) have a better success launch for all sites than higher Payloads (4000-8000 kg).

Section 5

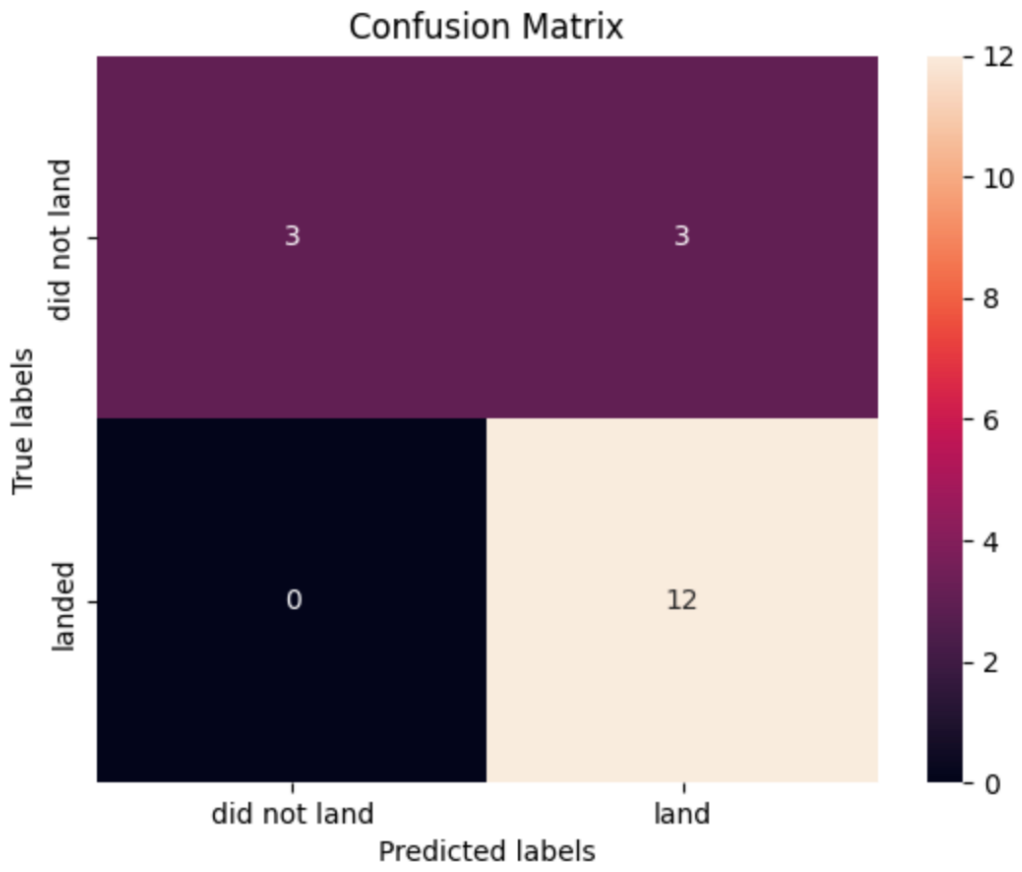
Predictive Analysis (Classification)

Classification Accuracy



- Decision Tree has the best accuracy (88,75%).

Best Model Confusion Matrix



- Confusion Matrix for Decision Tree is shown.
- All models provided the same Confusion Matrix.
- The model has 12 TP (true positive) and 3 TN (true negative), which can be considered a good classification.
- 3 FP (false positive) indicates 3 unsuccessful landings classified as successful, which is not good.

Conclusions

- The success rate for SpaceX launches has been increasing over time.
- KSC LC-39A had the most successful launches (76.9%) of any sites.
- The Payload has a significant impact under launch success. Low weighted Payload had better outcomes compared to heavy weighted Payloads.
- Decision Tree Classifier provided the best results for the presented dataset.

Appendix

- Presentation, python codes and screenshots can all be found in GITHUB.

Thank you!