IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

VISHWAJEET RAUT

13 January 2023

Outline

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

Executive Summary

❖ Summary of methodologies:-

- Data Collection through API
- Data Collection with 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 all results:-

- Exploratory Data Analysis result
- Interactive analytics in screenshots
- Predictive Analytics result

Introduction

❖ **Project background and context:-**

- Space X 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 Space X can reuse the first stage. Therefore, if we can determine if the first stage will land , we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

❖ **Problems you want to find answers:-**

- What factors determine if the rocket will land successfully?
- The interaction amongst various features that determine the success rate of a successful landing.
- What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

❖Data collection methodology:

- Data was collected using SpaceX API & Web Scraping.

❖Perform data wrangling:

- One-hot encoding applied on categorical features.

❖Perform exploratory data analysis (EDA) using visualization and SQL

❖Perform interactive visual analytics using Folium ,Plotly & Dash

❖Perform predictive analysis using classification models

- How to build & evaluate classification models.

Data Collection

- Describe how data sets were collected.
- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.

• The link to the notebook is

https://github.com/Vishwajeet-Raut/capstone/blob/main/spacex_collection_API.ipynb

OR

https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/0db6152e-f3f2-42e0-af58-cec7befa527b/view?access_token=d2284fd92eadc0041ad31f39982235e4114090d23397695bedf0e61867d8ba81

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook is https://github.com/Vishwajeet-Raut/capstone/blob/main/spacex_collection_API.ipynb

OR

https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/0db6152e-f3f2-42e0-af58-cec7befa527b/view?access_token=d2284fd92eadc0041ad31f39982235e4114090d23397695bedf0e61867d8ba81

1.Get request for rocket launch data using API.

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)
```

2.Use json_normalize method to convert json result to dataframe.

```
In [13]: # Use json_normalize meethod to convert the json result into a dataframe
         static_json_df=response.json()
         data=pd.json_normalize(static_json_df)
```

3.We then performed data cleaning and filling in the missing values.

```
In [32]: # Calculate the mean value of PayloadMass column
         PayloadMass = pd.DataFrame(data_falcon9['PayloadMass'].values.tolist()).mean(1)
         print(PayloadMass)
         # Replace the np.nan values with its mean value
         rows = data_falcon9['PayloadMass'].values.tolist()[0]
         df_rows = pd.DataFrame(rows)
         df_rows = df_rows.replace(np.nan,PayloadMass)
         data_falcon9['PayloadMass'][0] = df_rows.values
         data_falcon9
```

Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is

https://github.com/Vishwajeet-Raut/capstone/blob/main/spacex_webscraping.ipynb

OR

https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/942ba0c1-ef15-4fd1-b6c4-4a6100c2e052/view?access_token=ddeaa5c382a61be34c38332c41b47609f2c5bb3bf00d0953d507433c980d8c19

1.Apply HTTP Get method to request the Falcon 9 rocket launch page.

```
In [4]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
In [5]: # use requests.get() method with the provided static_url
        # assign the response to a object
        html_data = requests.get(static_url)
        html_data.status_code
```

```
Out[5]: 200
```

2.Create a BeautifulSoup object from the HTML response.

Create a `BeautifulSoup` object from the HTML `response`

```
In [6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
        soup = BeautifulSoup(html_data.text,'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
In [7]: # Use soup.title attribute
        soup.title
```

```
Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3.Extract all column names from the HTML table header.

```
In [10]: column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name)>0):
            column_names.append(name)
    except:
        pass
```

4.Create a dataframe by parsing the launch HTML tables.

5.Export data to CSV.

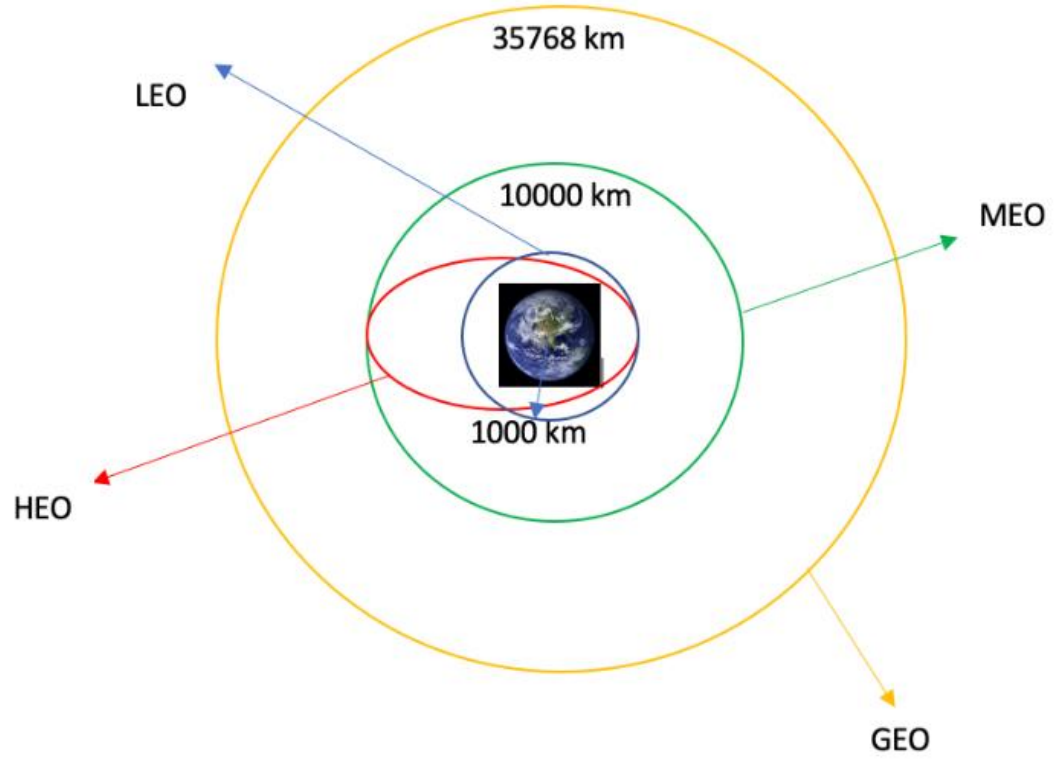
Data Wrangling

- We performed exploratory data analysis.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is

https://github.com/Vishwajeet-Raut/capstone/blob/main/spacex_wrangling.ipynb

OR

https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/418e9658-ff7c-4062-9e72-368947b6b609/view?access_token=86a71015699182af8fb8f1ae29e0cae5436109bc55bb979e81fa84fbf1f7ddf4



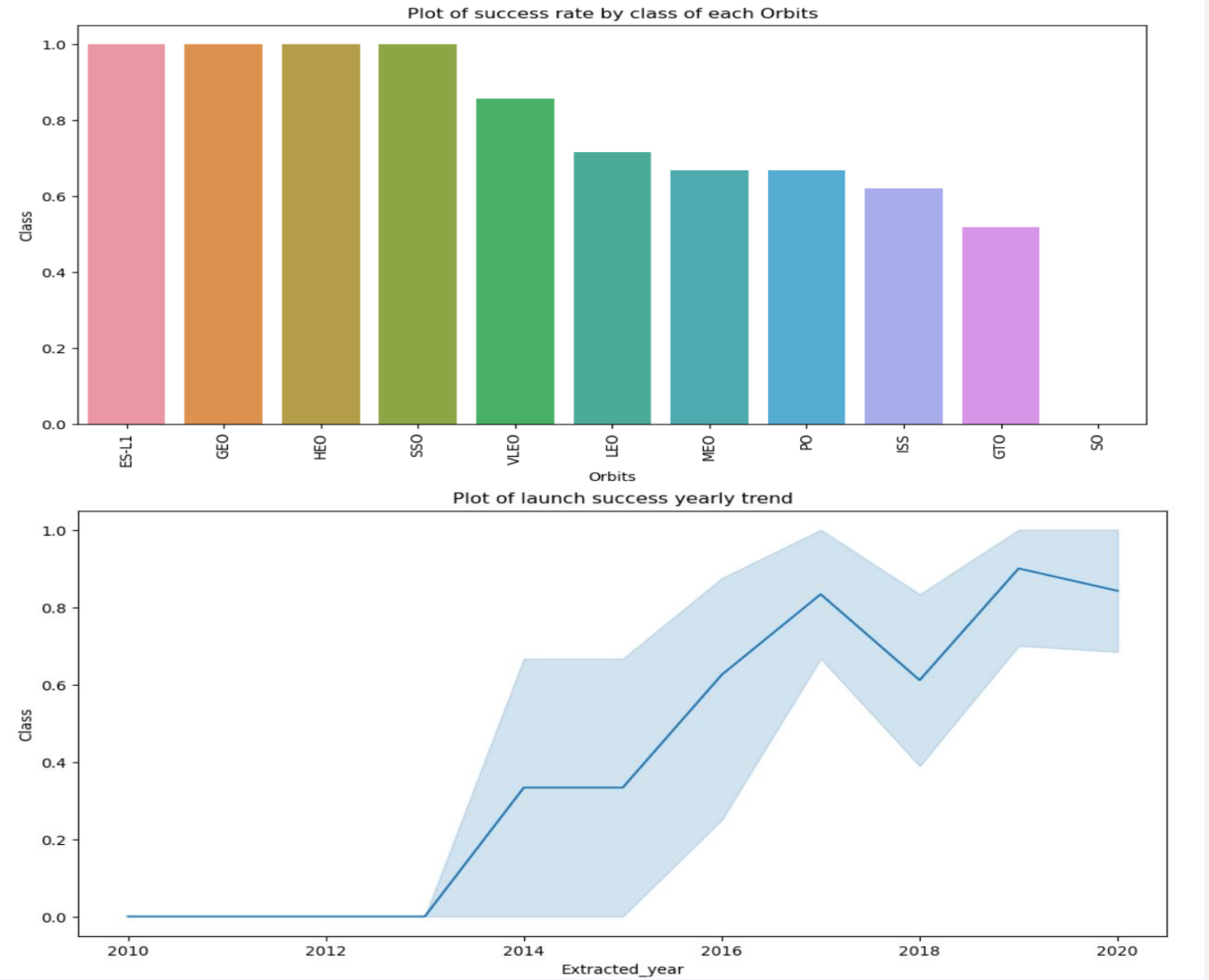
EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- The link to the notebook is

https://github.com/Vishwajeet-Raut/capstone/blob/main/spacex_visualization.ipynb

OR

https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/98c0b5f5-8563-43d9-bbfe-df7f29b8df2c/view?access_token=947da9ca9d640688dcfcaf6bbb3087c1237cf8fb57e2f69ad84178dafb8a46e4



EDA with SQL

❖ We loaded the SpaceX dataset into database without leaving the jupyter notebook.

❖ We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:

- The names of unique launch sites in the space mission.
- The total payload mass carried by boosters launched by NASA (CRS)
- The average payload mass carried by booster version F9 v1.1
- The total number of successful and failure mission outcomes
- The failed landing outcomes in drone ship, their booster version and launch site names.

The link to the notebook is

https://github.com/Vishwajeet-Raut/capstone/blob/main/spacex_sql.ipynb

OR

https://eu-de.dataplatform.cloud.ibm.com/analytics/notebooks/v2/25f77616-5cdb-45b3-8b99-a415efd00a91/view?access_token=59d490ed14768187e8dfeefcdb31eff13096c06f442568cd75c44cf07519973c

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.

  i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:-

- Are launch sites near railways, highways and coastlines.
- Do launch sites keep certain distance away from cities.

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly, dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- The link to the notebook is

https://github.com/Vishwajeet-Raut/capstone/blob/main/spacex_dash_app.py

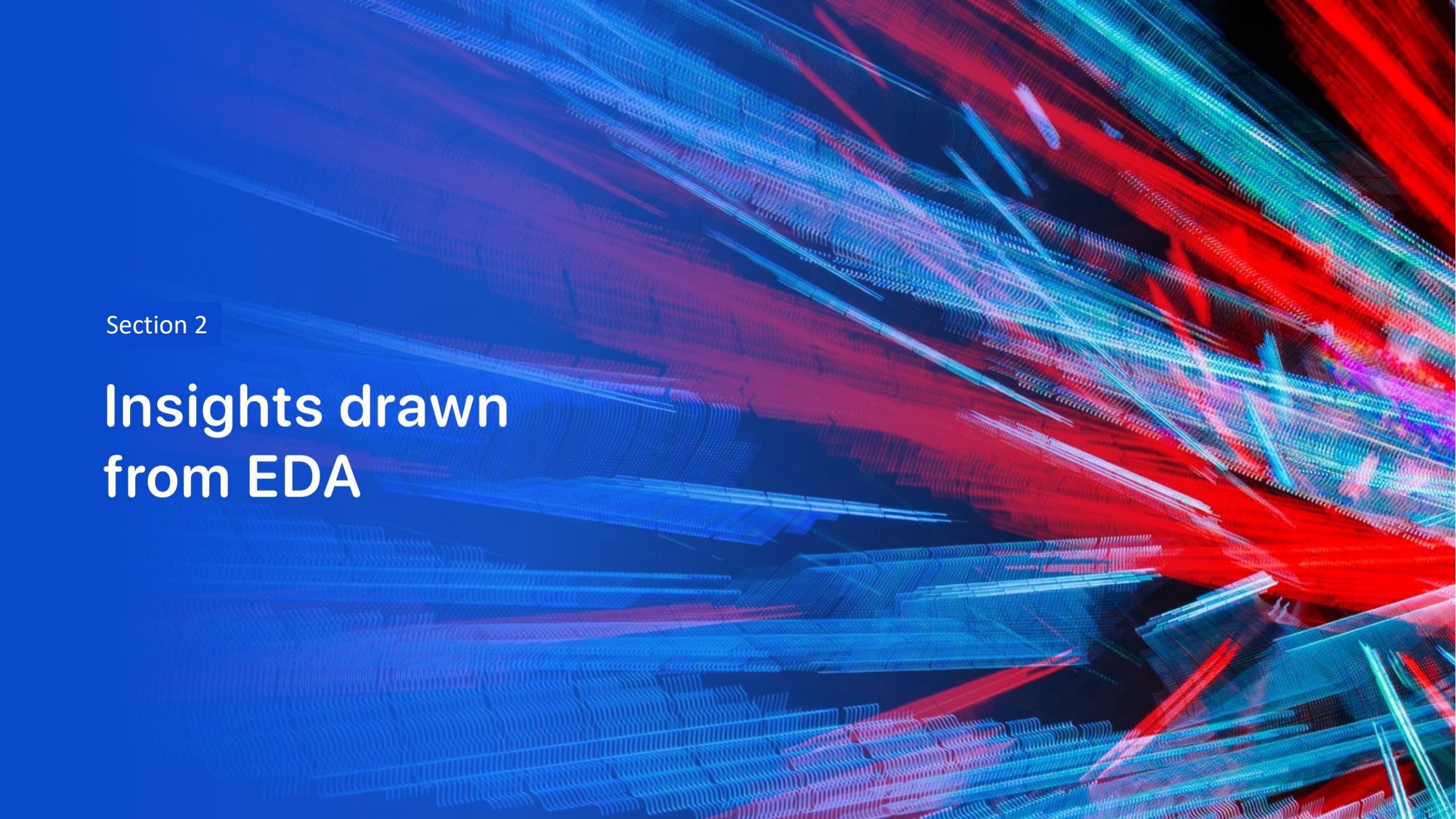
Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- The link to the notebook is

https://github.com/Vishwajeet-Raut/capstone/blob/main/SpaceX_Machine_Learning_Prediction.ipynb

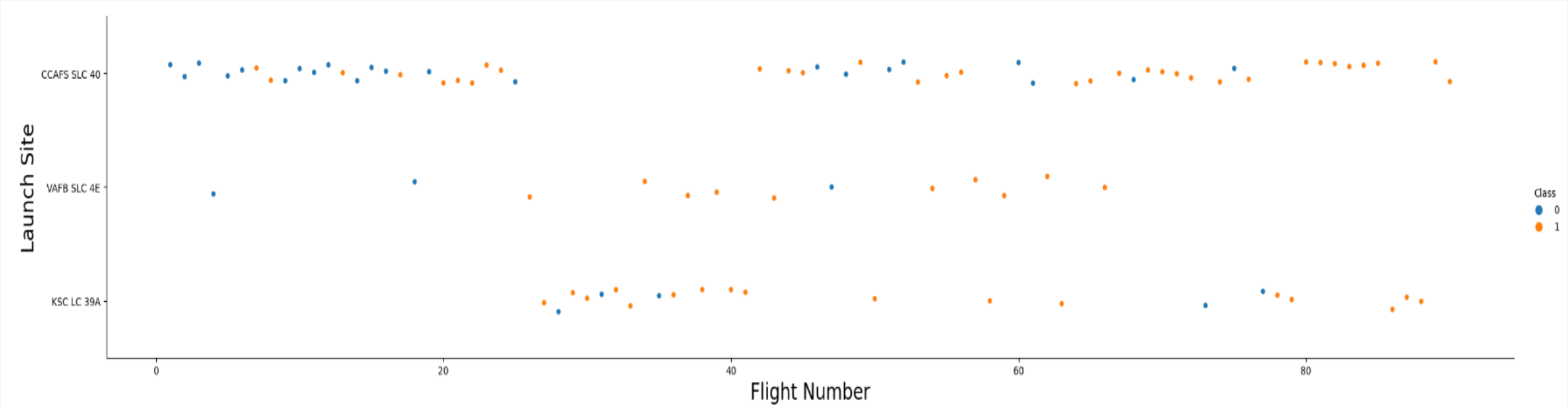
Results

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


Section 2
Insights drawn from EDA

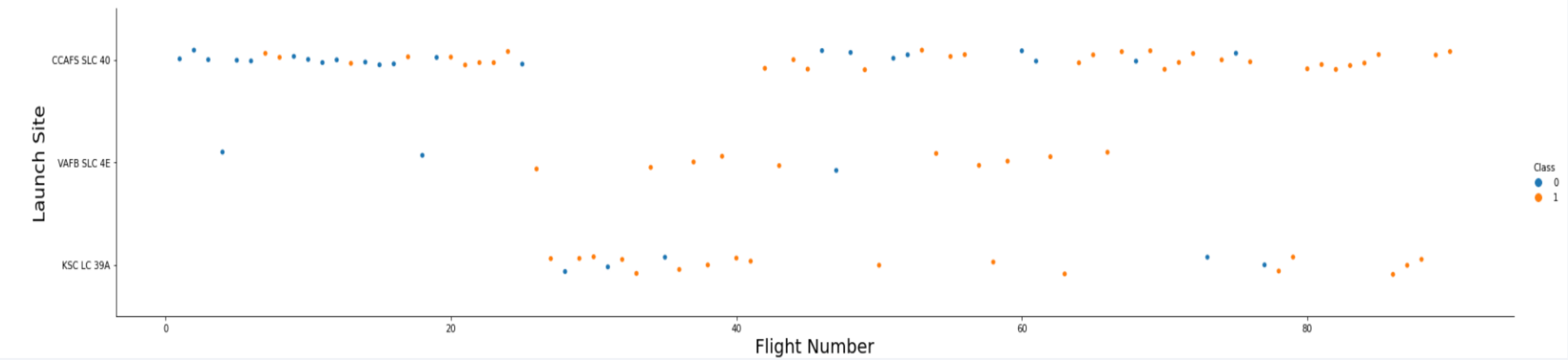
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



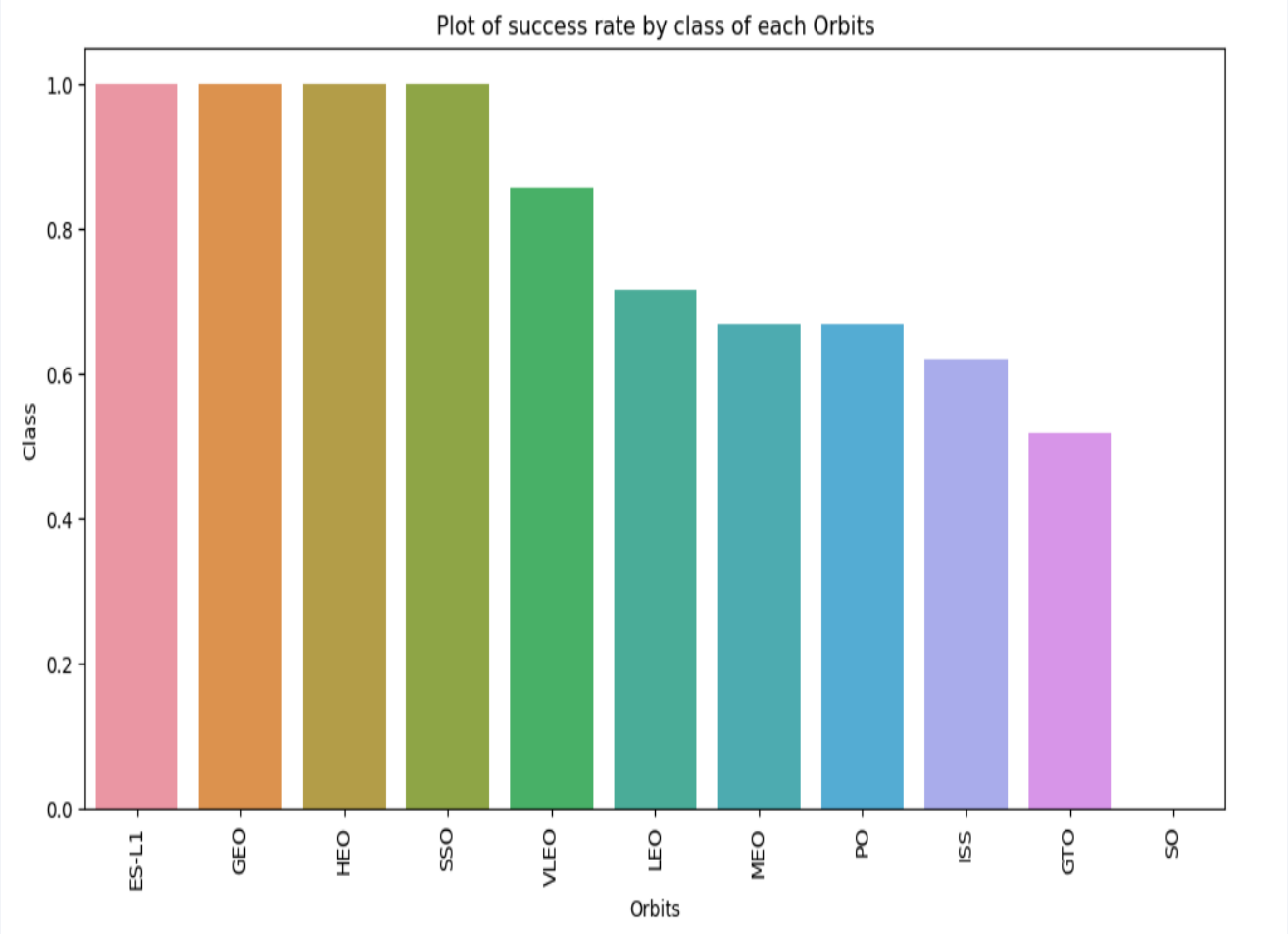
Payload vs. Launch Site

- The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



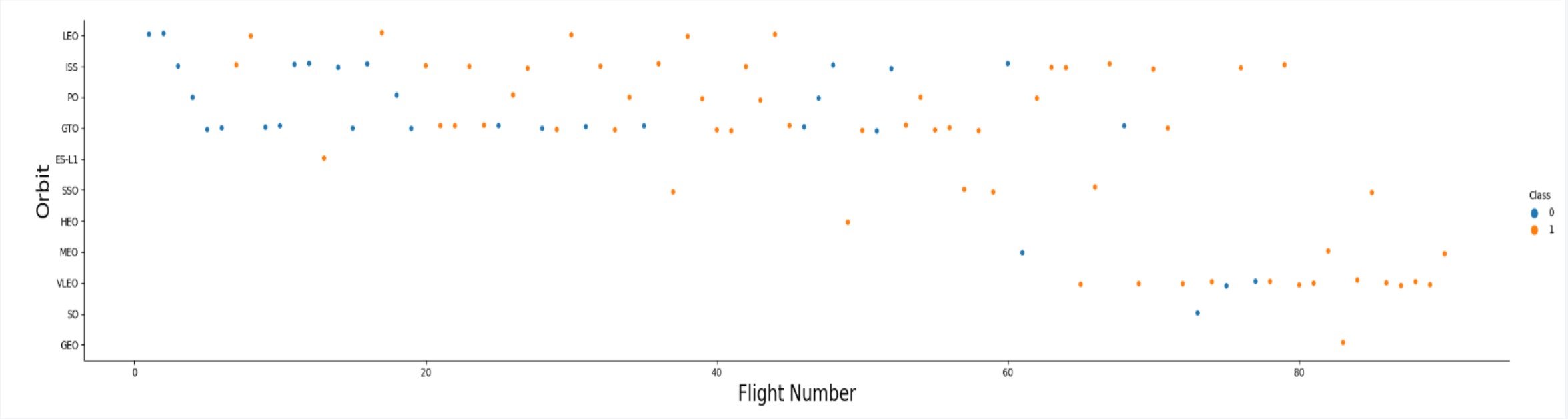
Success Rate vs. Orbit Type

• From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



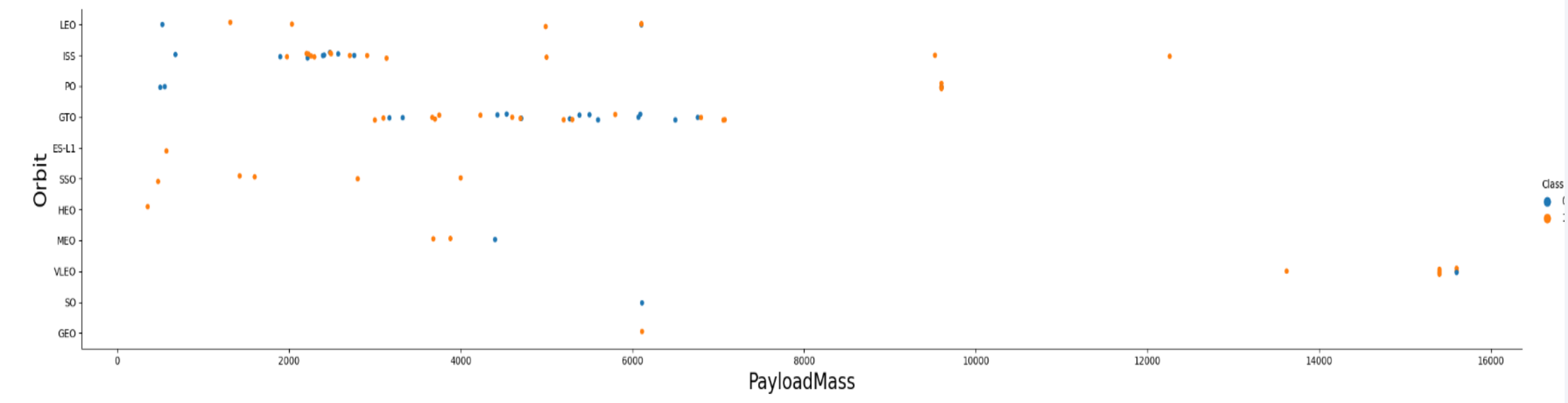
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



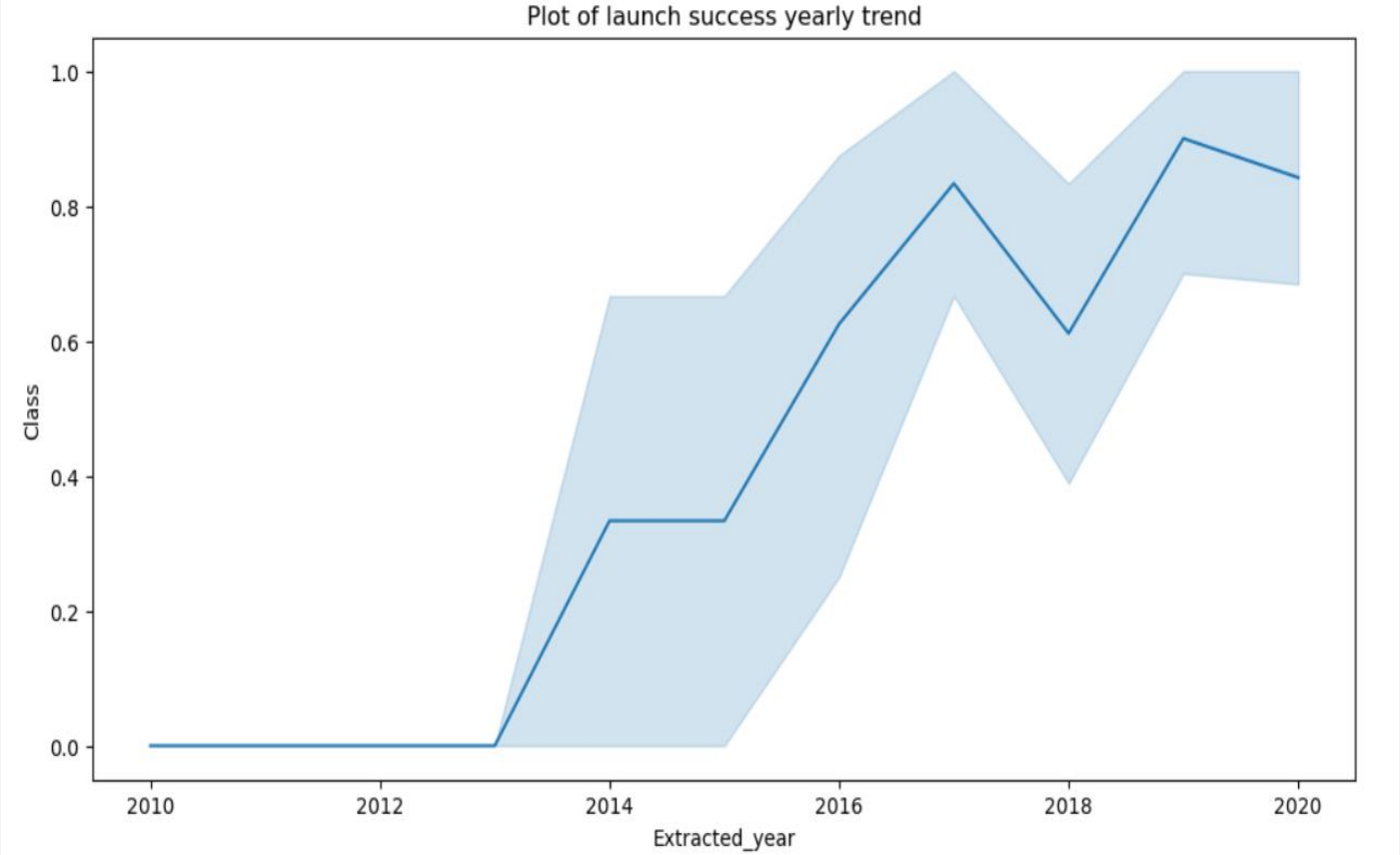
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

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

```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL
```

```
 * ibm_db_sa://twp10921:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

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

Launch Site Names Begin with 'CCA'

- We used the query above to display 5 records where launch sites begin with `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://twp10921:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/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

- We calculated the total payload carried by boosters from NASA as 45596 using the query below

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

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

```
 * ibm_db_sa://twp10921:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

1
45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928

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

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

```
 * ibm_db_sa://twp10921:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

1
2928

First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

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

Hint:Use min function

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

```
 * ibm_db_sa://twp10921:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000.

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 BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)'
```

```
 * ibm_db_sa://twp10921:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- We used like '%' to filter for WHERE Mission Outcome was a success or a failure.

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
```

```
 * ibm_db_sa://twp10921:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

1
101

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://twp10921:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
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

2015 Launch Records

- We used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015.

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

```
%sql SELECT TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME, \
    LANDING__OUTCOME AS LANDING__OUTCOME, \
    BOOSTER_VERSION AS BOOSTER_VERSION, \
    LAUNCH_SITE AS LAUNCH_SITE \
    FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%'
```

```
 * ibm_db_sa://twp10921:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

month_name	landing__outcome	booster_version	launch_site
JANUARY	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
APRIL	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.

- We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order

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 "DATE", COUNT(LANDING__OUTCOME) as COUNT FROM SPACEXTBL \
    WHERE "DATE" BETWEEN '2010-06-04' and '2017-03-20' AND LANDING__OUTCOME LIKE '%Success%' \
    GROUP BY "DATE" \
    ORDER BY COUNT(LANDING__OUTCOME) DESC
```

```
 * ibm_db_sa://twp10921:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

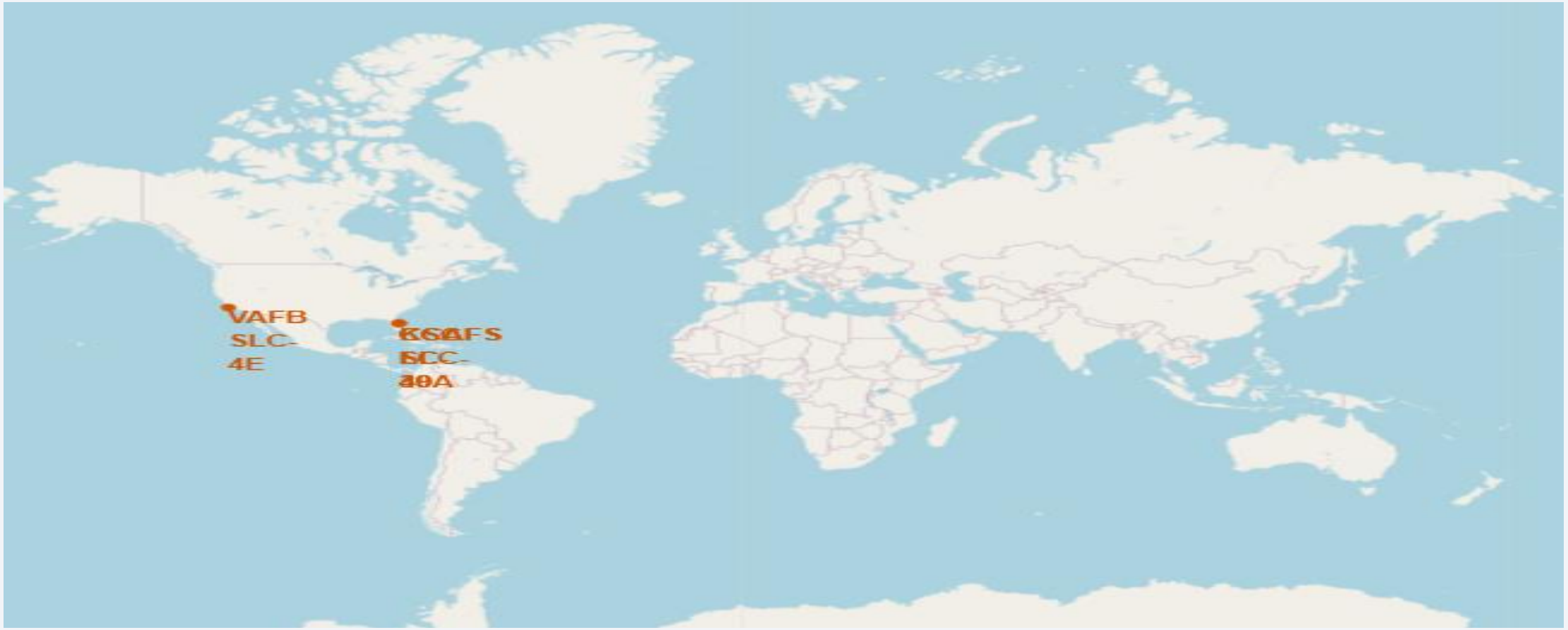
DATE	COUNT
2015-12-22	1
2016-04-08	1
2016-05-06	1
2016-05-27	1
2016-07-18	1
2016-08-14	1
2017-01-14	1
2017-02-19	1

Section 3

Launch Sites Proximities Analysis

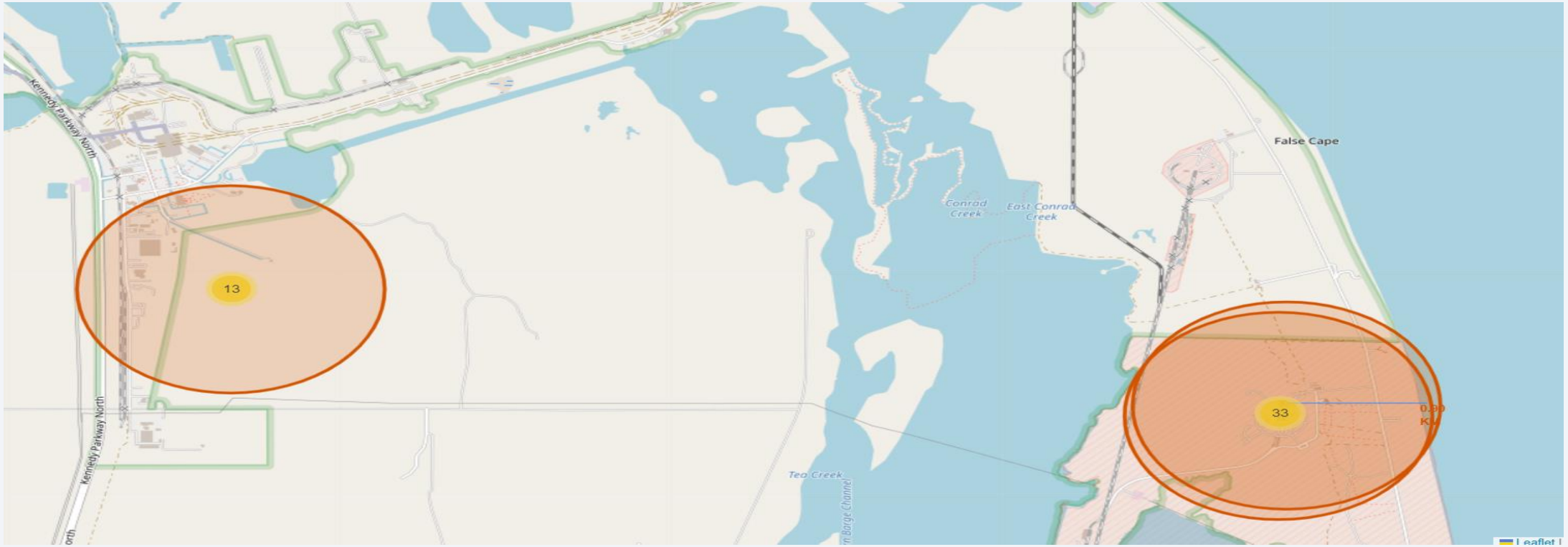
All launch sites global map markers

- We can see that the SpaceX launch sites are in the USA coasts i.e. Florida and California.

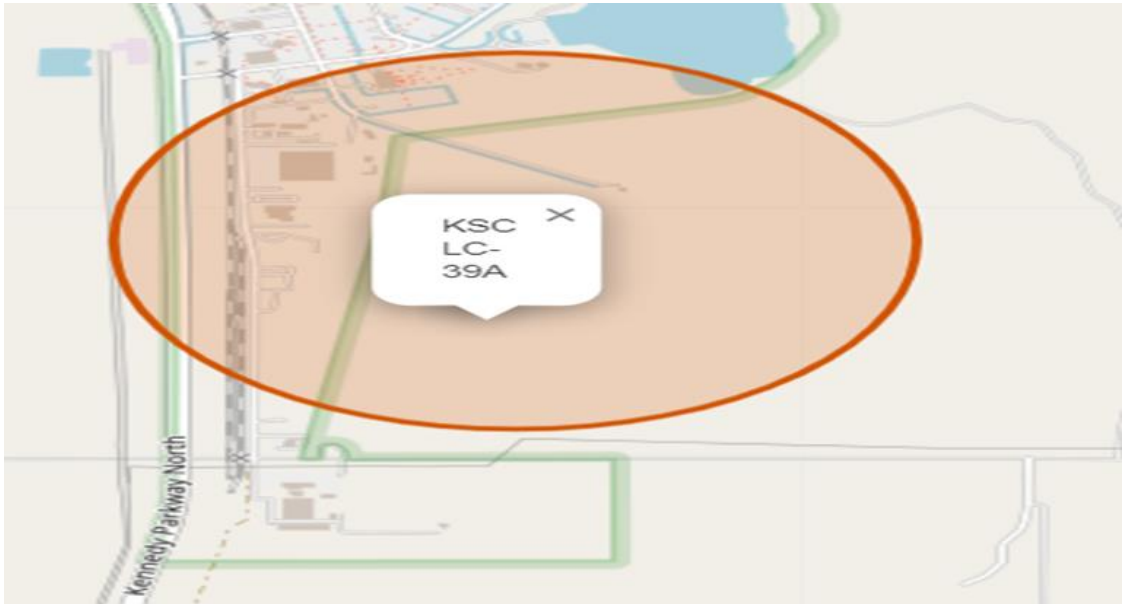


Markers showing launch sites with labels

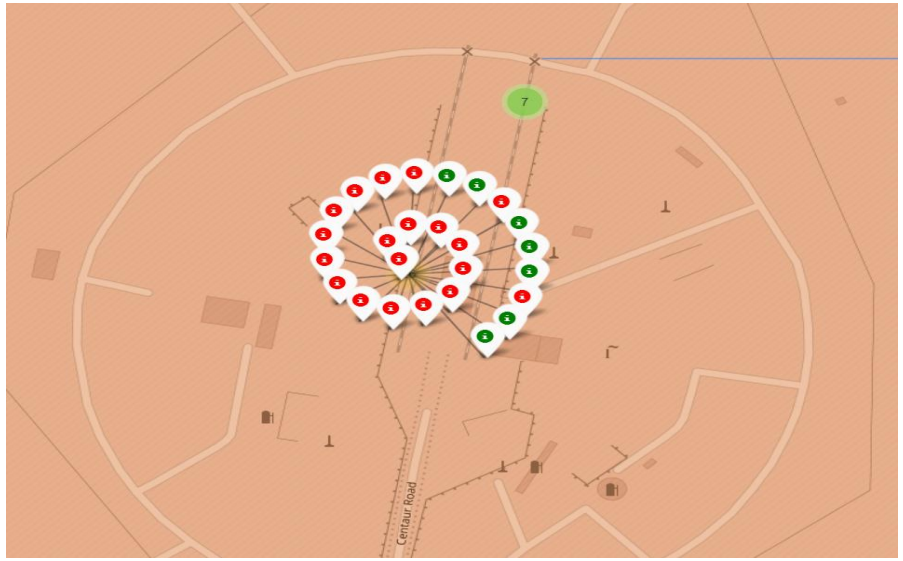
- FLORIDA LAUNCH SITE



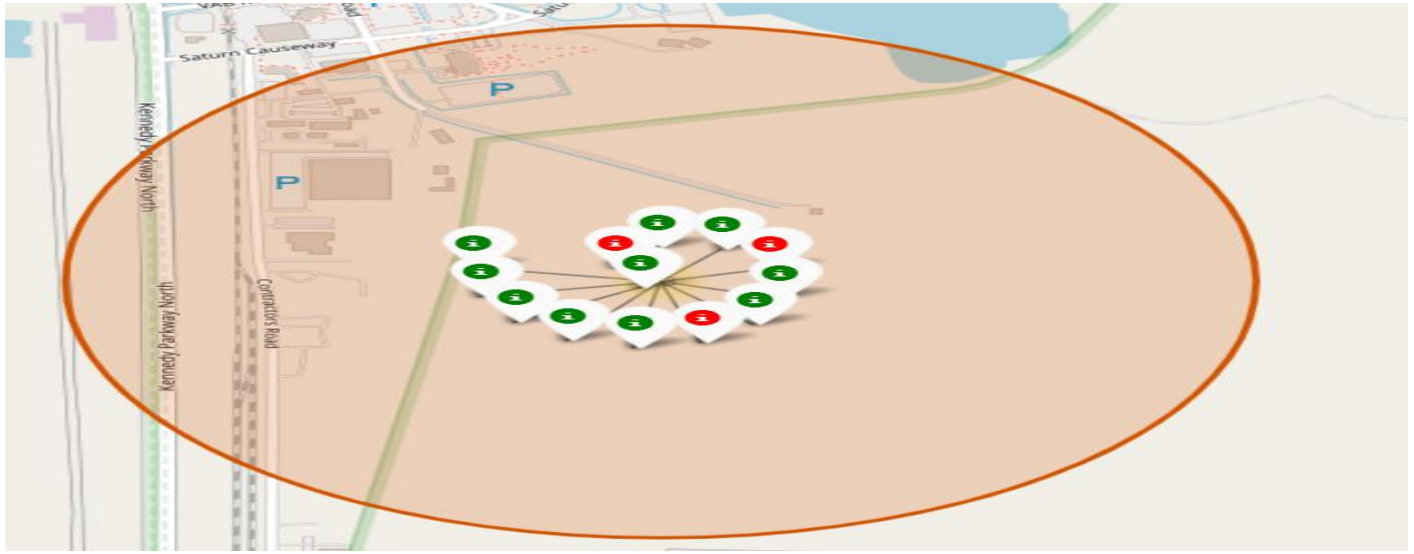

CCAFS
SLC-
40
33


KSC
LC-
39A
Kennedy Parkway North

CCAFS SLC-40

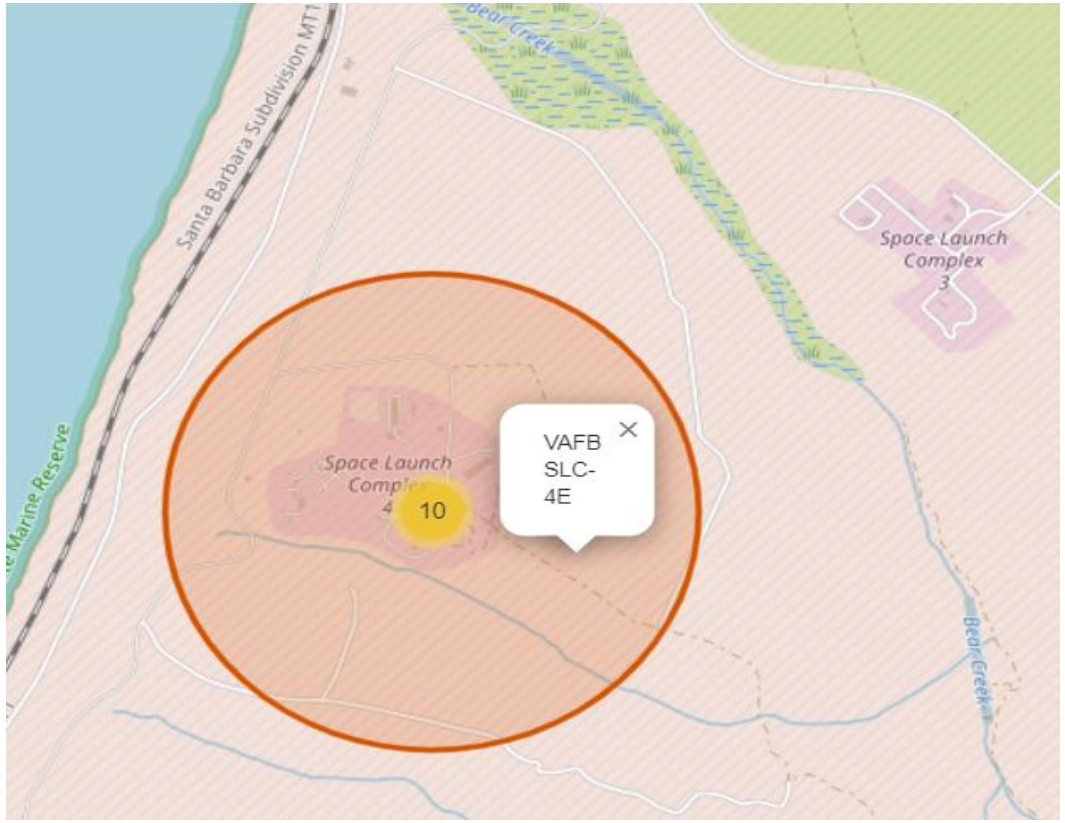


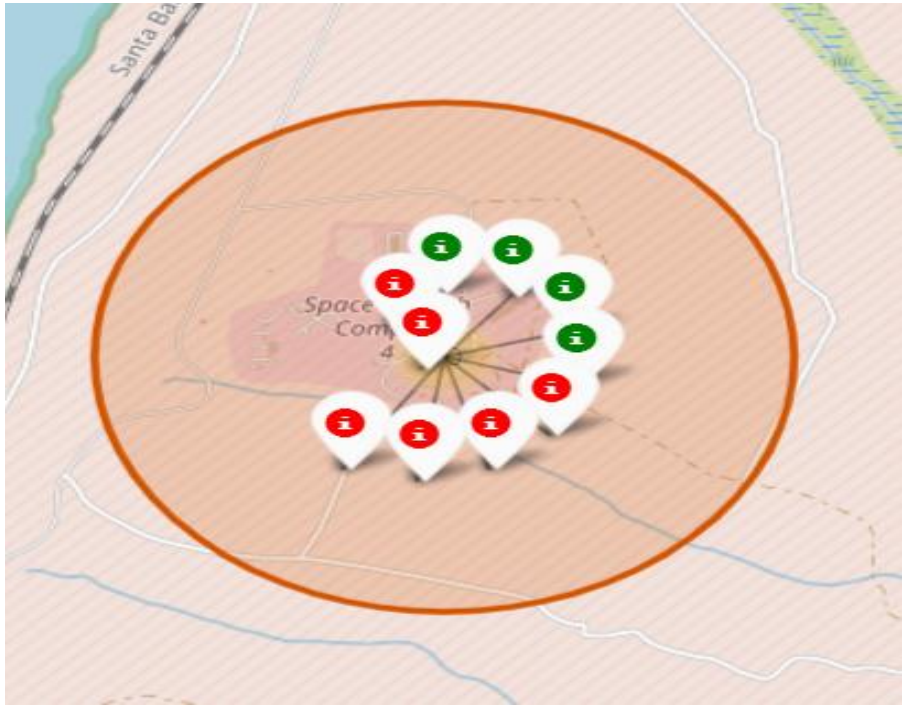
KSCL-39A



Green marker represent successful launch while red marker represent failed launch.

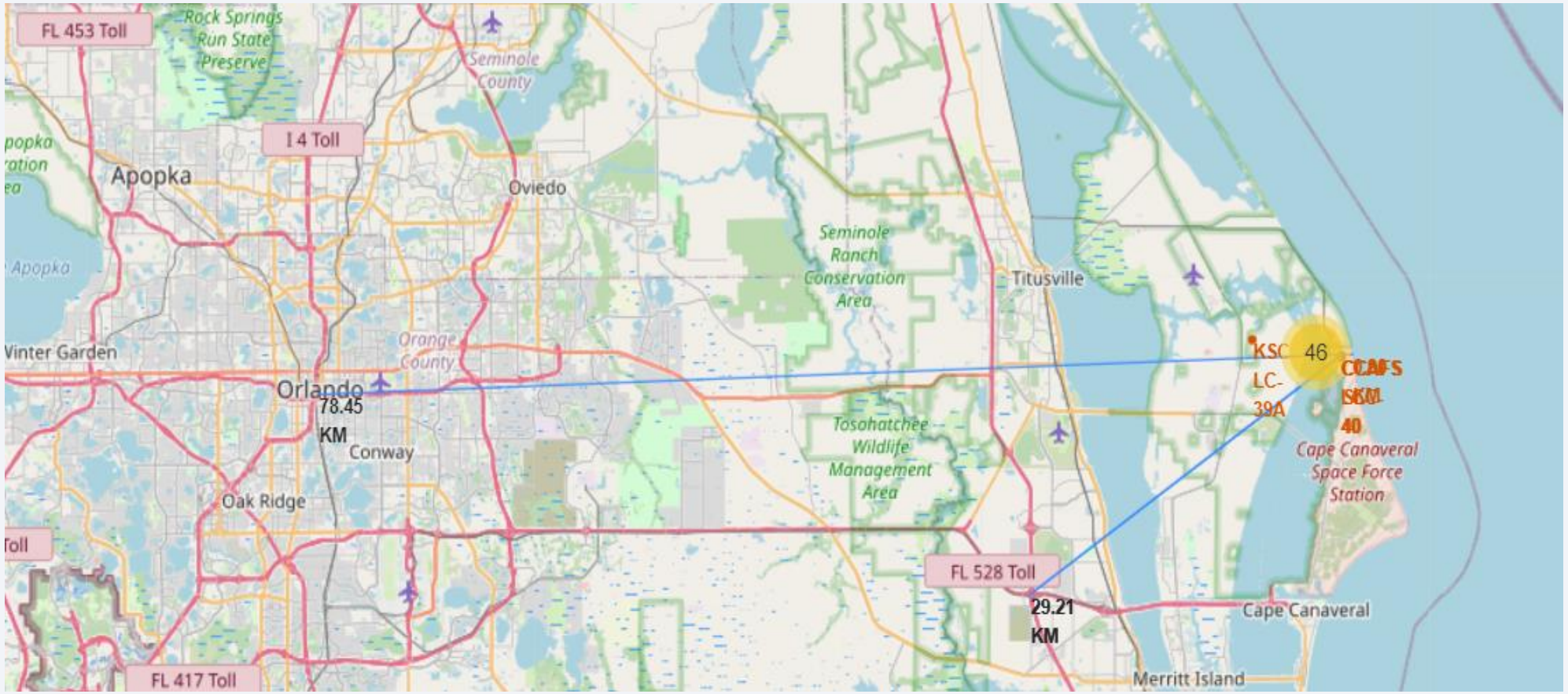
❖ CALIFORNIA LAUNCH SITE



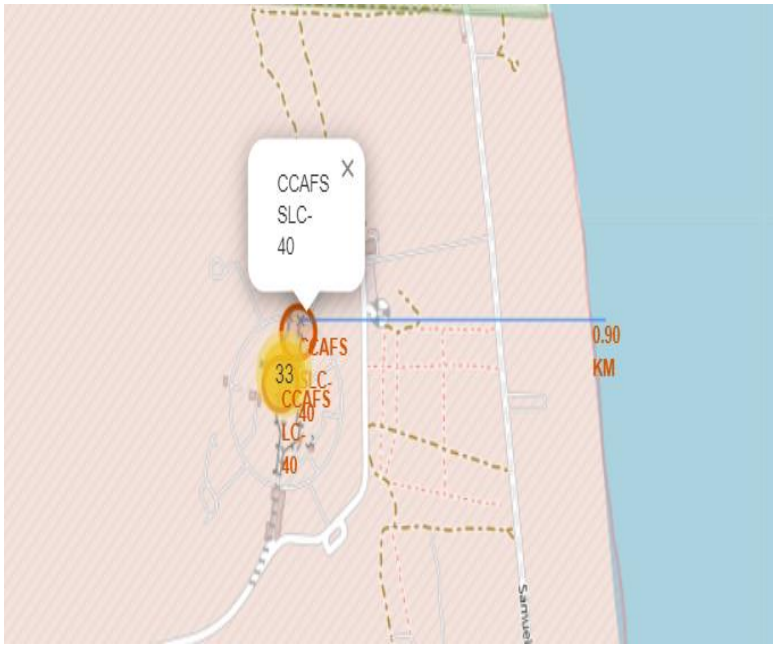


Green marker represent successful launch while red marker represent failed launch.

Launch Site distance to landmarks



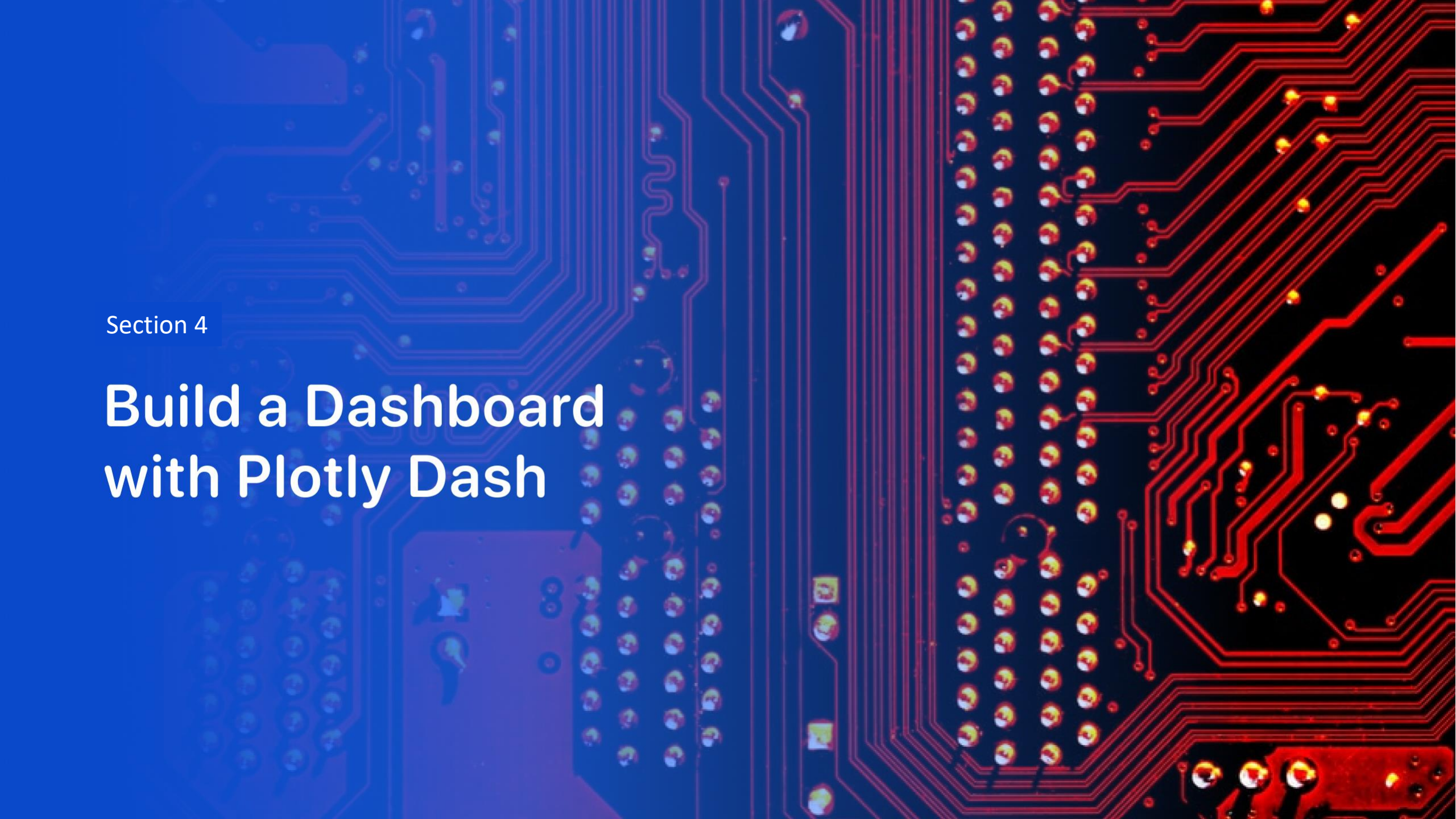
Coastline



Highway

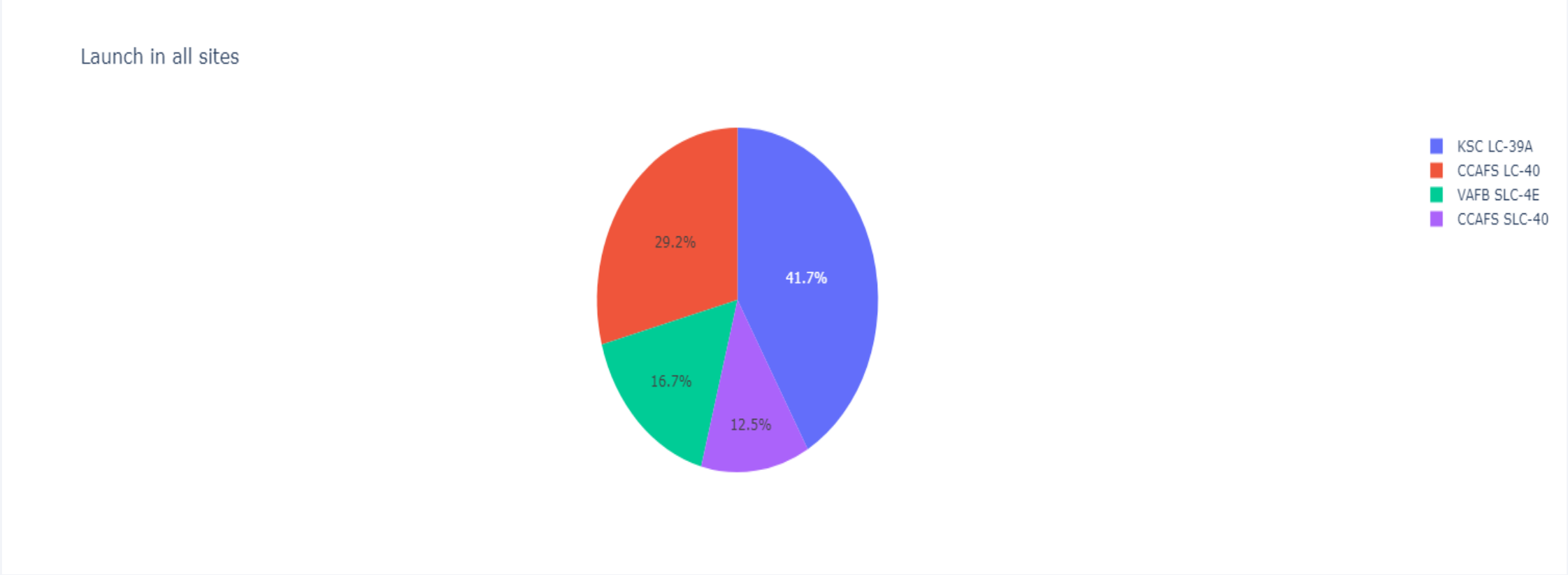
Railway station

- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

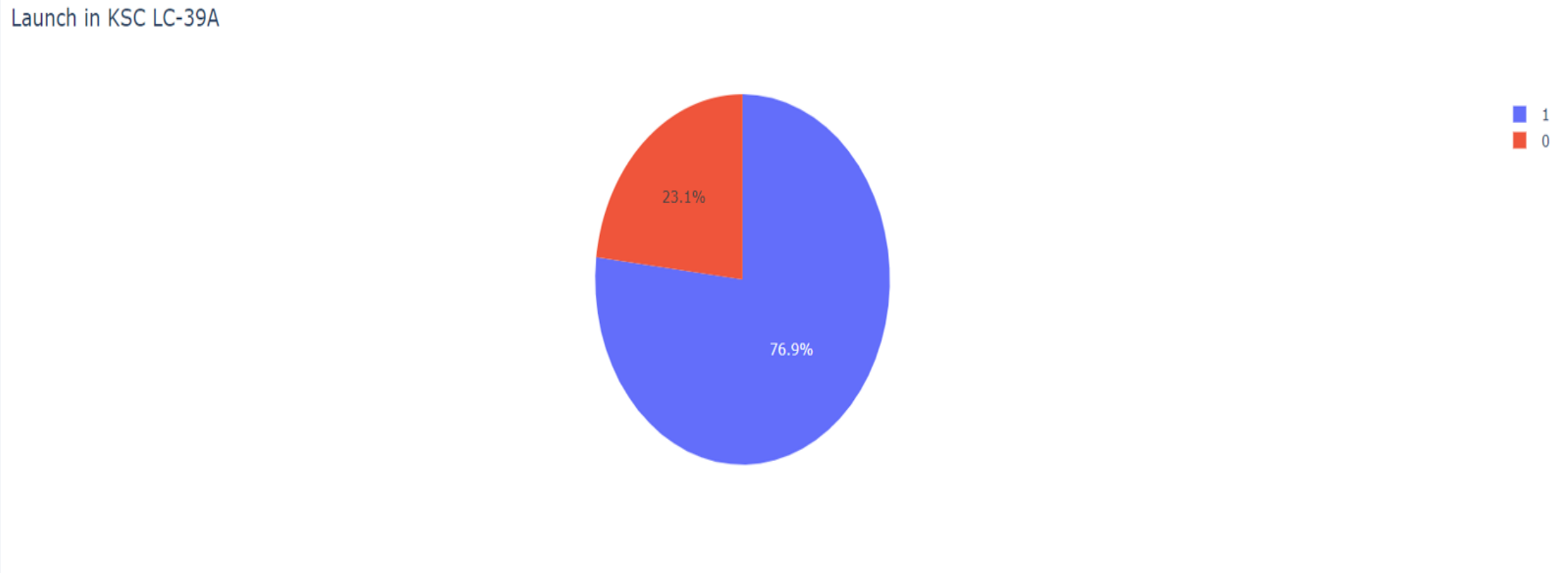
Success Percentage achieved by launch site.

- We can see that KSC LC-39A had the most successful launches from all the sites.



Launch site with the highest launch success ratio

- KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate.



Scatter plot of Payload VS Launch outcome for all sites with different payload selected in the range slider

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Decision Tree Classifier is the model with highest classification accuracy.

Find the method performs best:

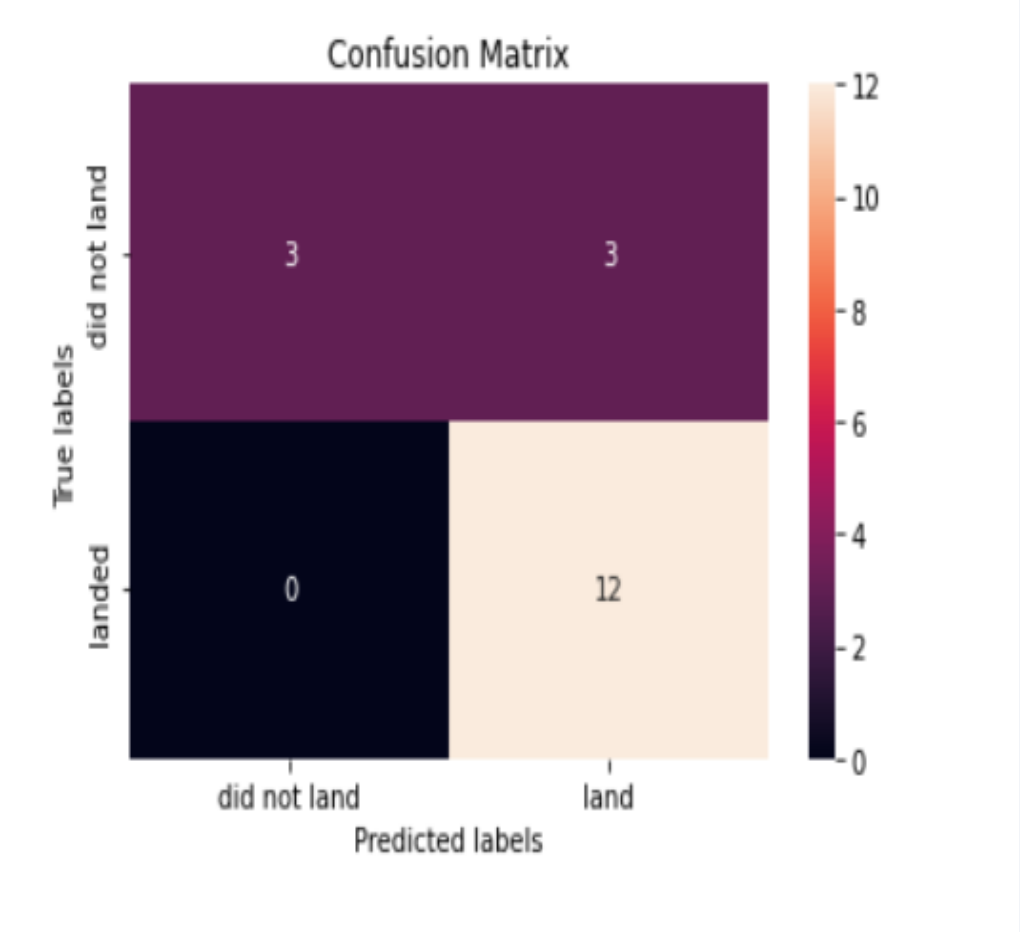
```
[ ]  models = {'KNeighbors':knn_cv.best_score_,
                'DecisionTree':tree_cv.best_score_,
                'LogisticRegression':logreg_cv.best_score_,
                'SupportVector': svm_cv.best_score_}

     bestalgorithm = max(models, key=models.get)
     print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
     if bestalgorithm == 'DecisionTree':
         print('Best params is :', tree_cv.best_params_)
     if bestalgorithm == 'KNeighbors':
         print('Best params is :', knn_cv.best_params_)
     if bestalgorithm == 'LogisticRegression':
         print('Best params is :', logreg_cv.best_params_)
     if bestalgorithm == 'SupportVector':
         print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

From the analysis we can conclude that:

1. The larger the flight amount at a launch site, the greater the success rate at a launch site.
2. Launch success rate started to increase in 2013 till 2020.
3. Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
4. KSC LC-39A had the most successful launches of any sites.
5. The Decision tree classifier is the best machine learning algorithm for this analysis.

Thank you!