IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

DIVYA PRAKASH
27-04-2023

Outline

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

Executive Summary

Summary of methodologies

- Data collection
- Data wrangling
- EDA with data visualization
- Building an interactive map with Folium
- Summary of all results
- EDA results
- Interactive analytics
- Predictive analytics

Introduction

- In this capstone, we will predict if the Falcon 9 first stage 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. 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 SpaceX for a rocket launch.
- Our responsibility is to establish the cost of each launch. We'll accomplish this by compiling data on Space X and building dashboards for your team. Whether SpaceX will reuse the first stage will also be decided. You will train a machine learning model and utilise open data to forecast whether SpaceX will reuse the first stage rather than utilising rocket science to evaluate whether the first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - We are going to be working with data from the SpaceX REST API, which is an API that collects information about SpaceX launches. We will receive data on launches using this API, including details about the rocket used, the payload delivered, the launch specs, the landing specifications, and the outcome of the landing.
- Perform data wrangling
  - The data is then pre-processed by scaling, normalizing, and transforming it into a format that can be used for analysis.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
- KNN, DT, LR, SVM models have been built and evaluated for the best classifier.

Data Collection

- We are going to be working with data from the SpaceX REST API, which is an API that collects information about SpaceX launches.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

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.
- https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/00563da9-35c3-4cf0-b6a2-c91256608168/view?access_token=5ce472b7ecc1f5d829efef268647a90e60aa6dd9bab86b746ac30ae7ce7a6683

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/fba46625-14b6-413c-8629-17cd6d516282/view?access_token=2d1d8ade0025a65cf364ca8f02136d97002e80fee73fca97d0efeb6b39891f67

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

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

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

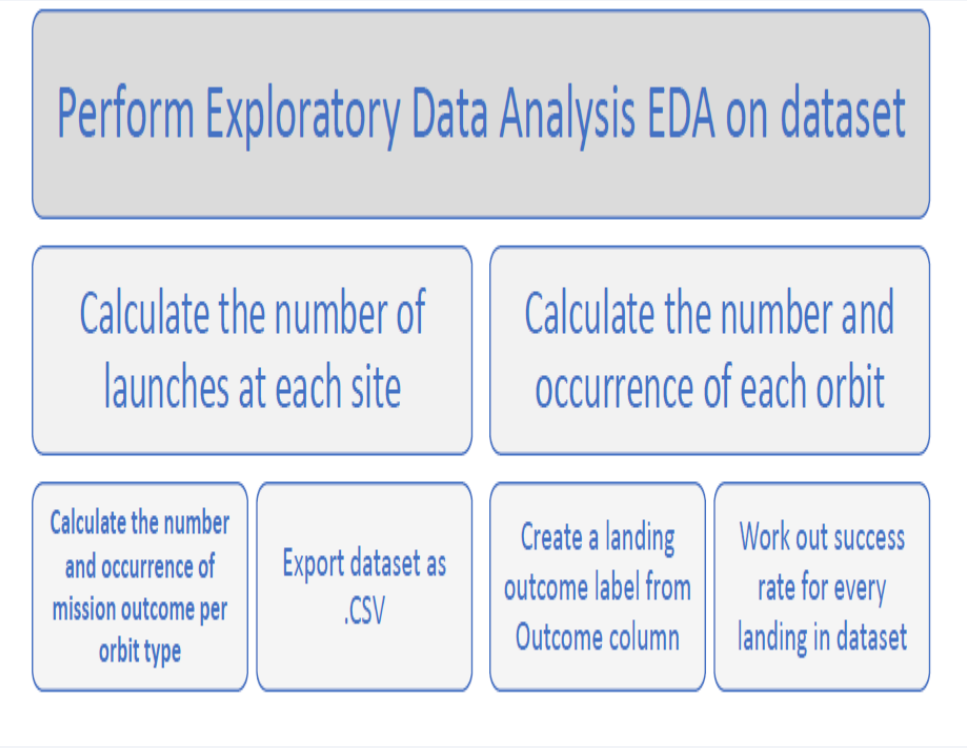
In [7]:
```
# Use soup.title attribute
soup.title
```

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

Data Wrangling

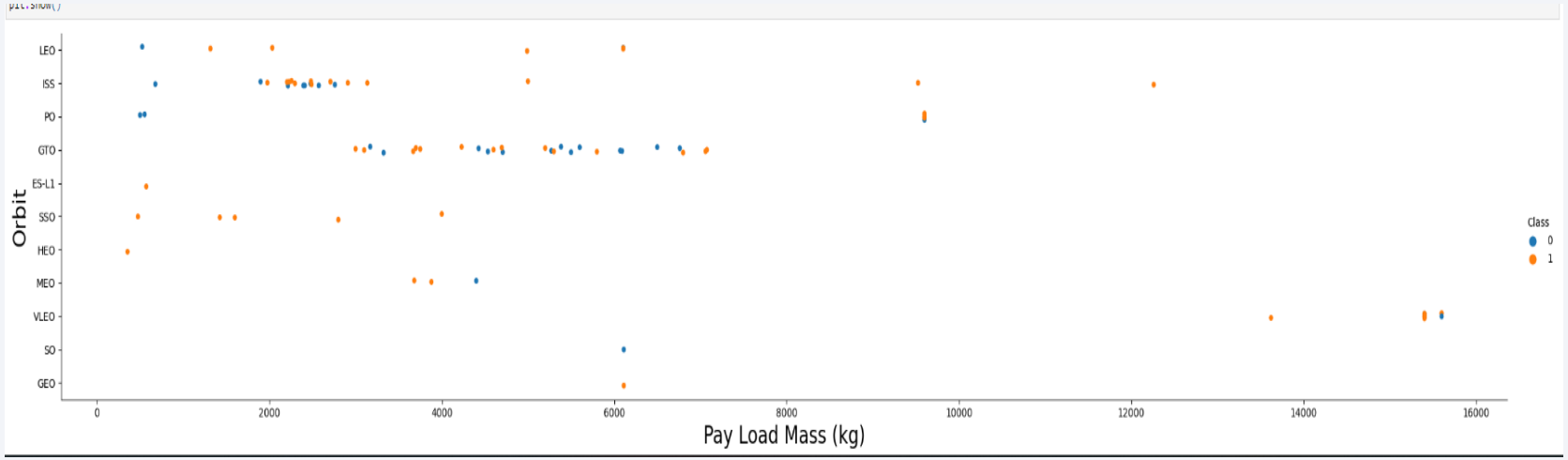
- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results.

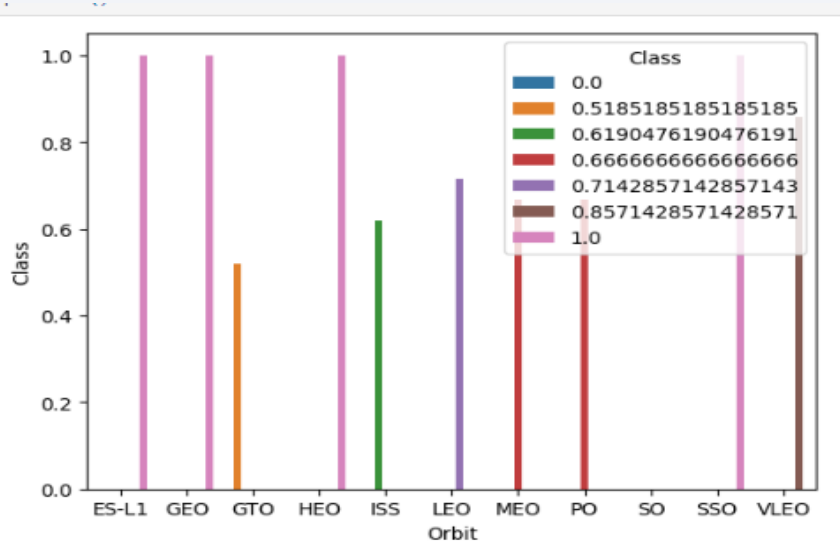
https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/ede839e7-1ce4-4b01-b41a-79fe1e225b46/view?access_token=112c882d099d188fe13e049c1e55c6cbccae41bff2fae2f88812f8e5a9adedfe

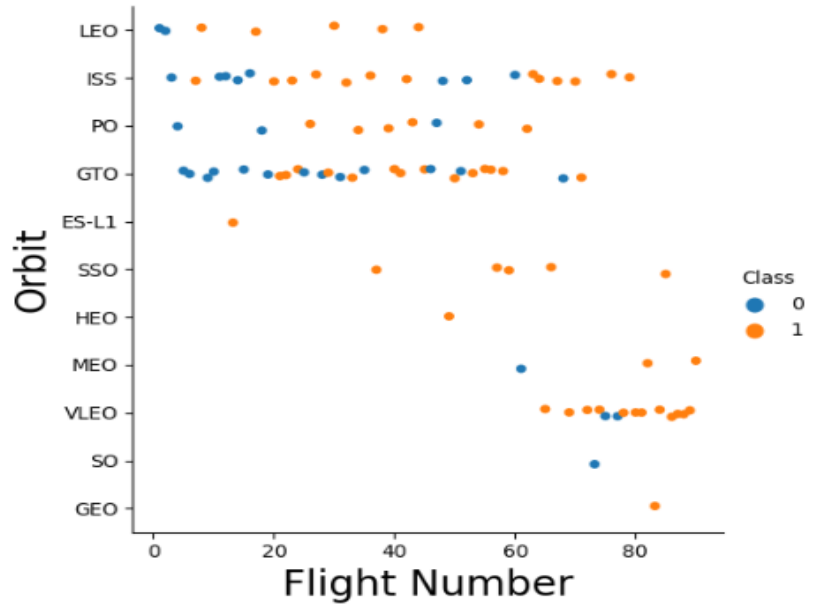


EDA with Data Visualization

- 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.
- https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/a3a105b9-75c3-45d3-b312-1d61d2bc01a1/view?access_token=8e096c215f9cc2a40b848515fe8232e104f0bbdf5e2905baaad2160e05092dca







EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL 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.
- https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/8c177694-6ce8-4c8c-8c6a-02de4d699631/view?access_token=61bcfb4cd29e1539e19d9843539e85ea61120a4f9c6a501166bf9c4b1bcb0bac

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.
  - https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/3b22fb42-d772-4088-a637-f8d51b14ee85/view?access_token=b16f3936bca26fd866ec44aaa2202fc92366293fe622780123b9be31d79c6492

Build a Dashboard with Plotly Dash

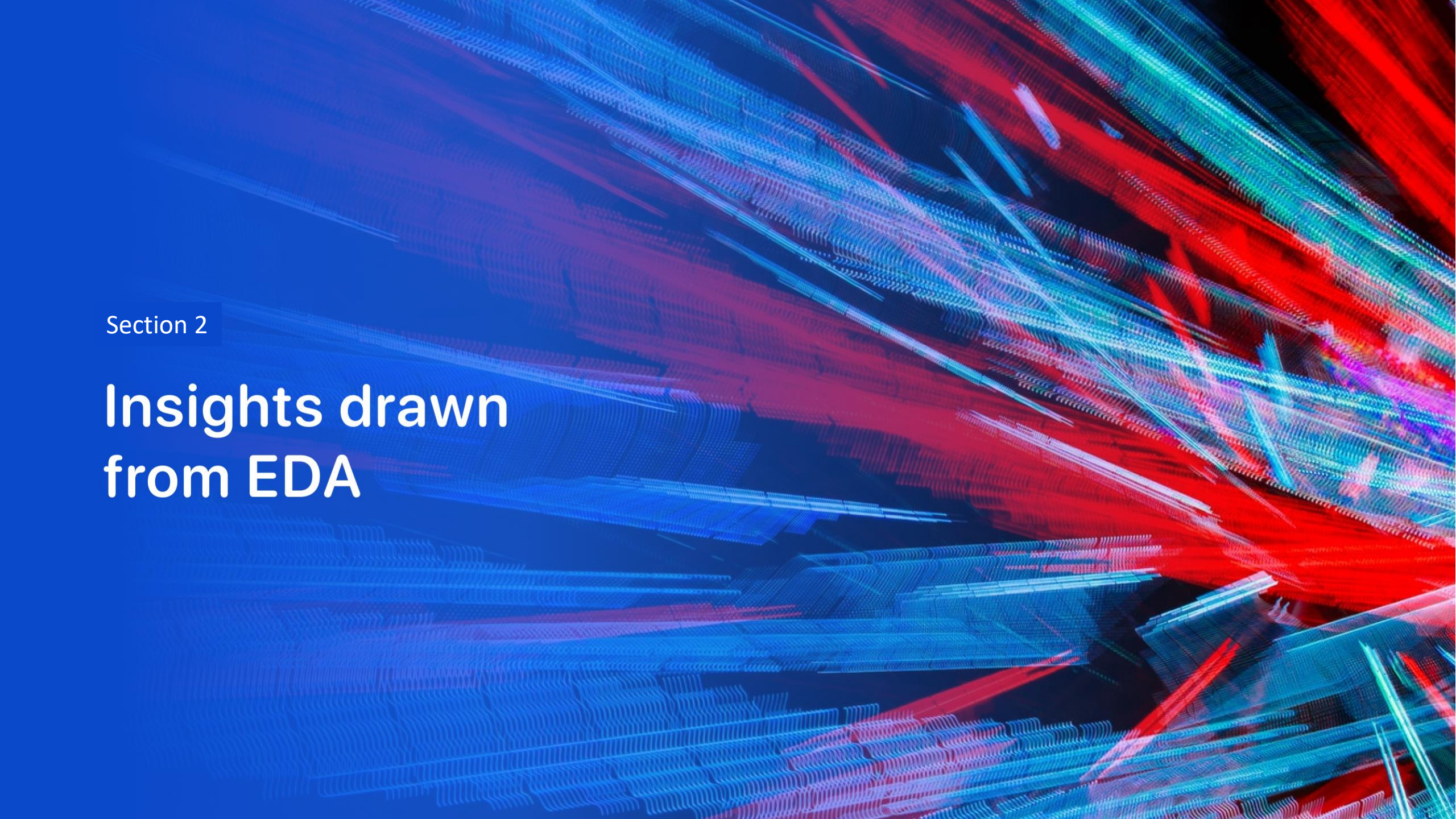
- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

Predictive Analysis (Classification)

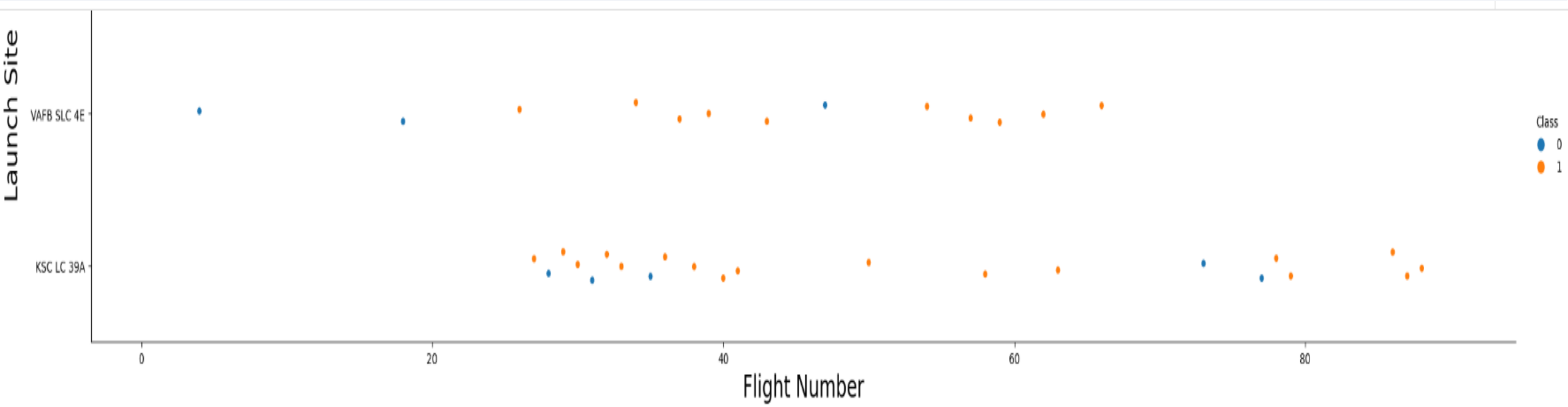
- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/d8048a2e-4476-4dc6-988e-082ac6aa01f7/view?access_token=7dfa4179a85cd1b86153c38bbda905cd333e87913159b525e00850ba116d81f1

Results

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


Section 2
Insights drawn from EDA

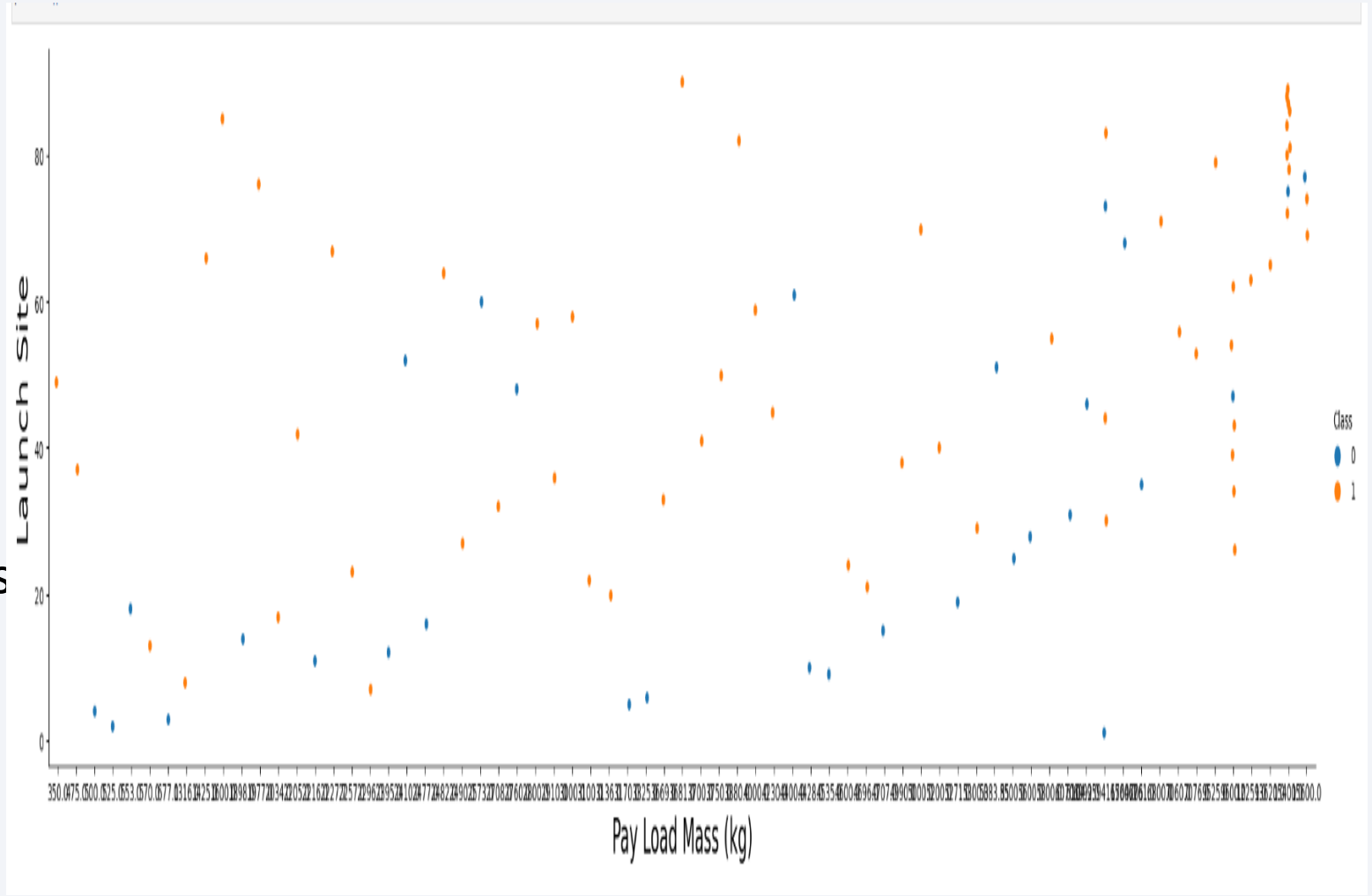
Flight Number vs. Launch Site



The more amount of flights at a launch site the greater the success rate at a launch site.

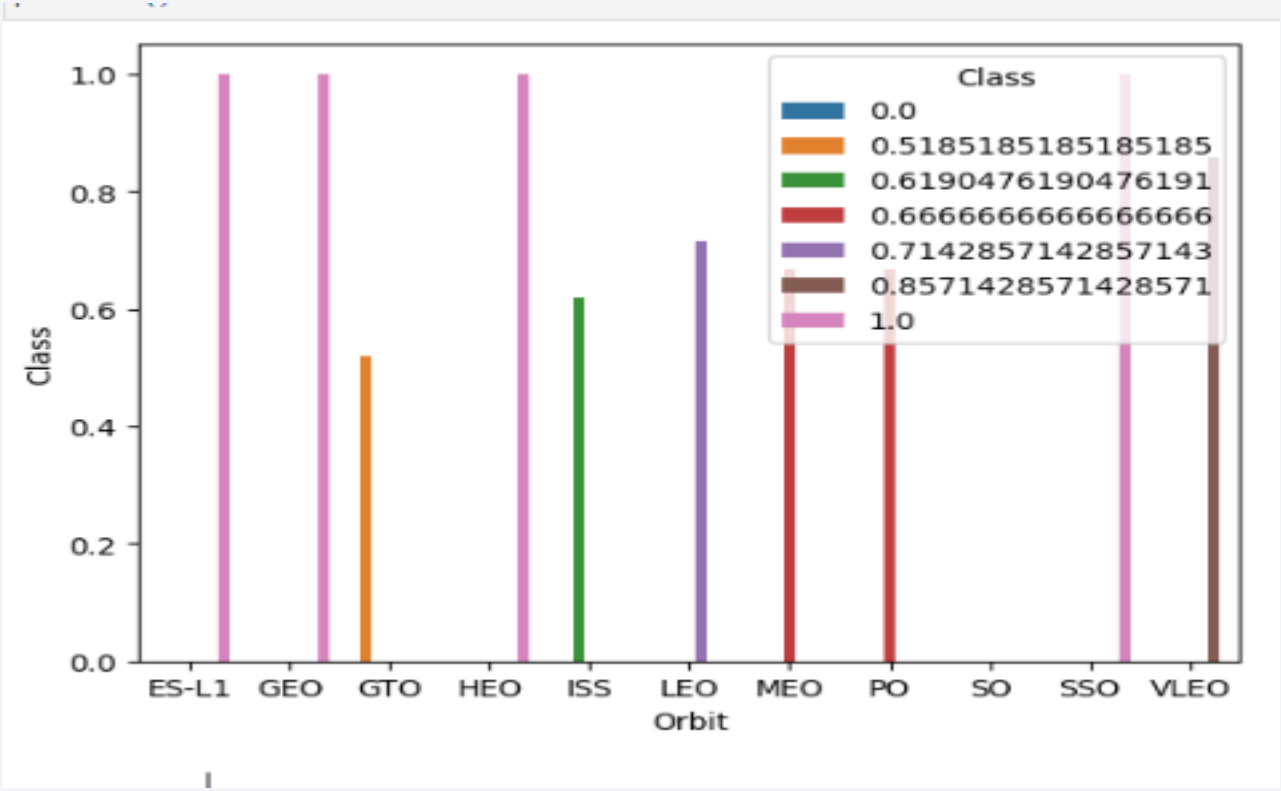
Payload vs. Launch Site

- The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket.
- There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependant on Pay Load Mass for a success launch.



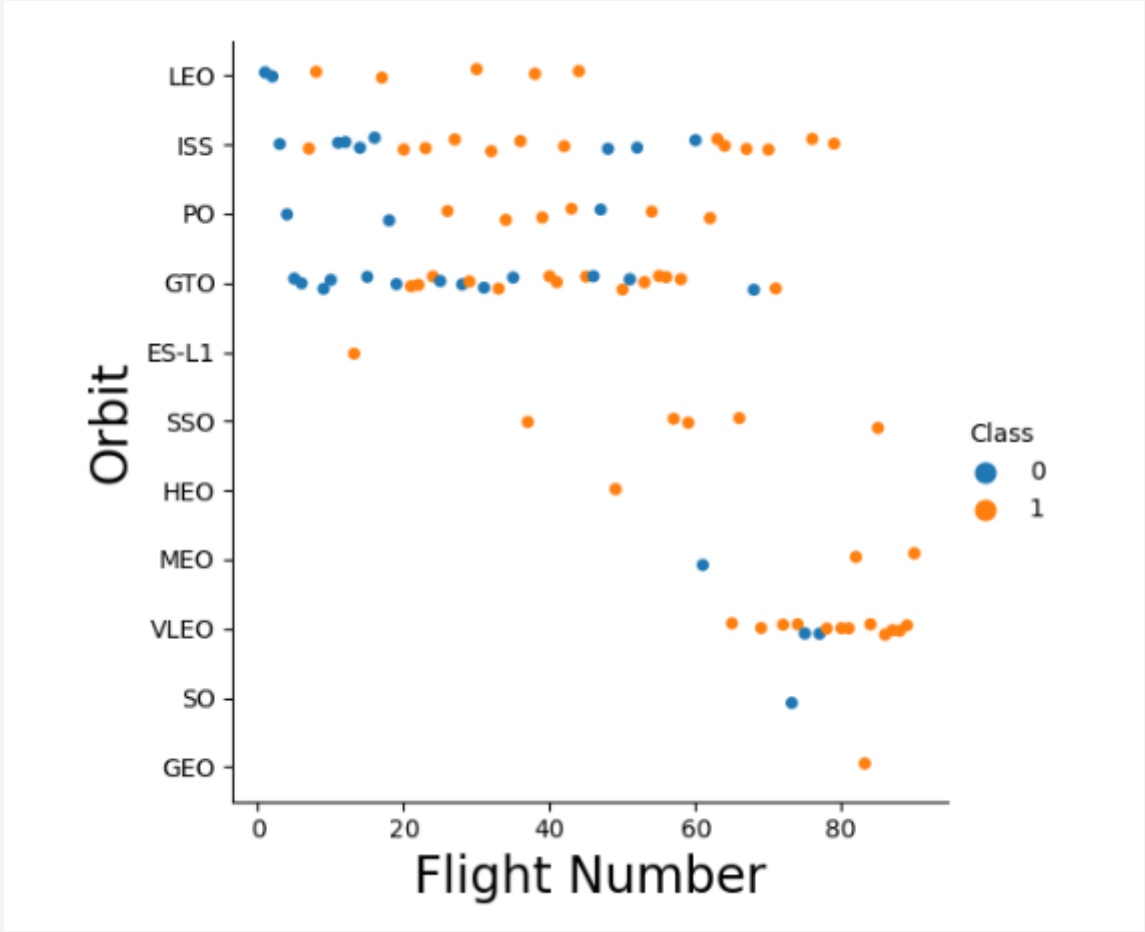
Success Rate vs. Orbit Type

- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate



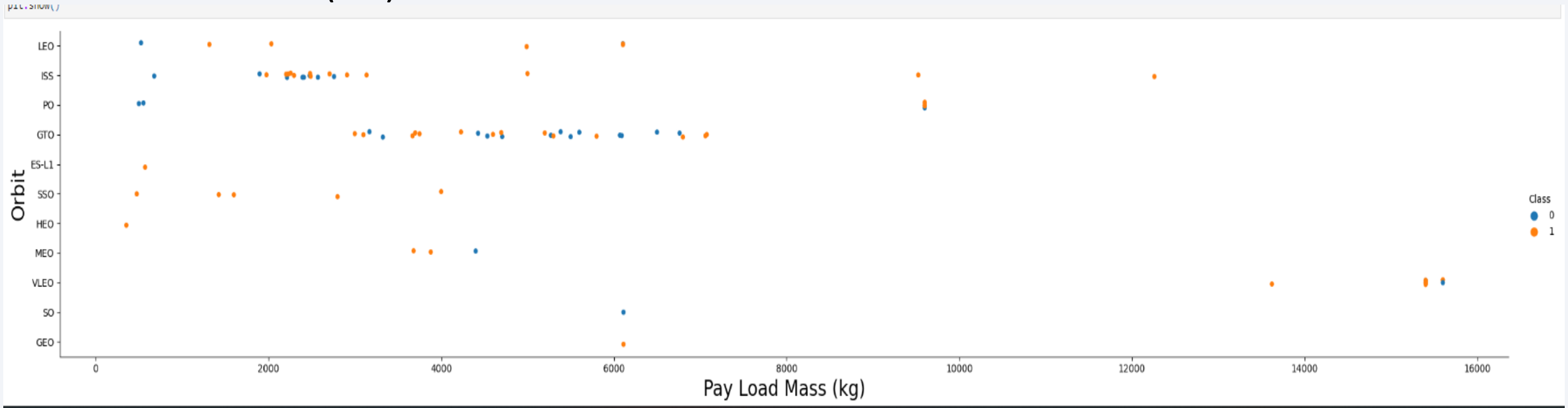
Flight Number vs. Orbit Type

- You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



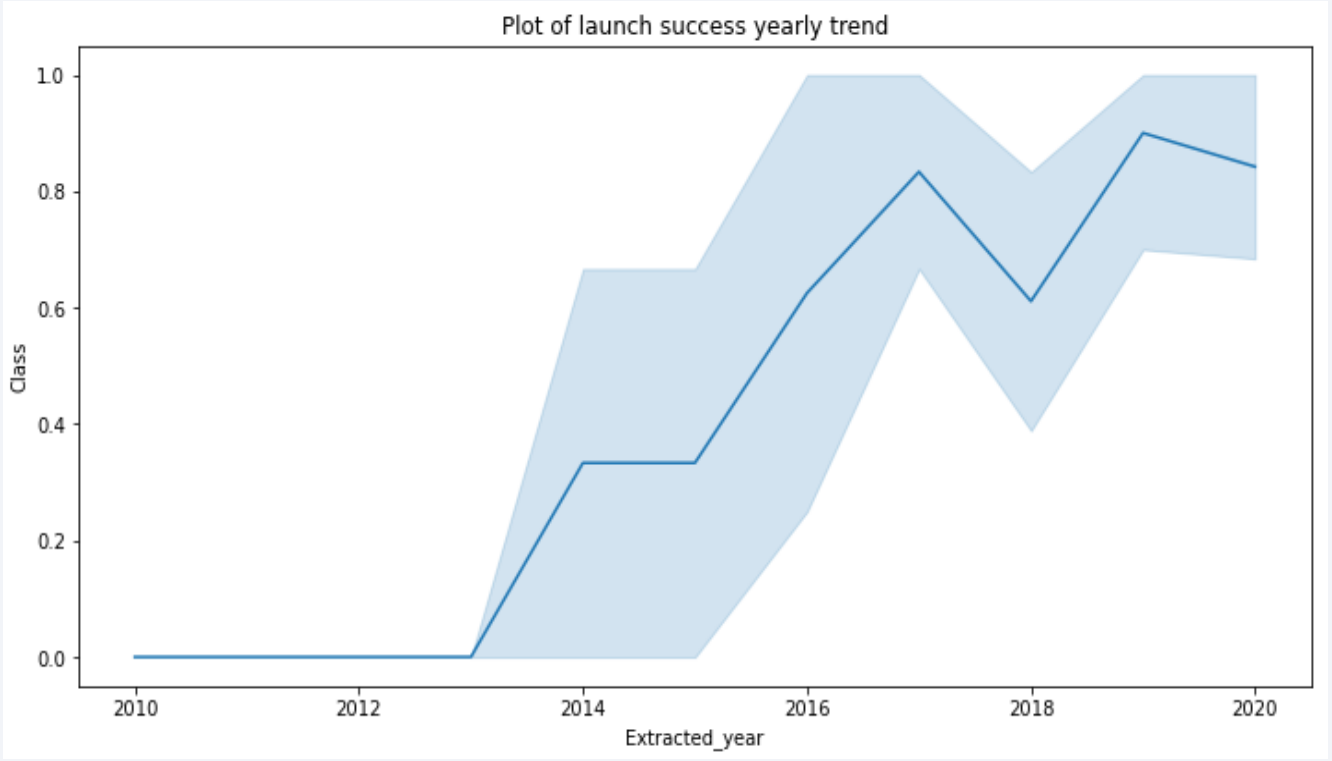
Payload vs. Orbit Type

- You should observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.



Launch Success Yearly Trend

- We can observe that the success rate since 2013 kept 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

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- Using the word ***TOP 5 in the query means that it will only show 5 records from tblSpaceX and LIKE keyword has a wild card*** with the words ***'KSC%'the percentage in the end suggests that the Launch_Site name must start with KSC.***

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

In [11]:
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- selectSUM(PAYLOAD_MASS_KG_)TotalPayloadMassfromtblSpaceXwhereCustomer='NASA(CRS)'",'TotalPayloadMass

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

In [12]:
```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

- Using the function ***AVG works out the average in the column PAYLOAD_MASS_KG_***
- The ***WHEREclause filters the dataset to only perform calculations on Booster_version F9 v1.1***

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

In [13]:
```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

- Using the function ***MIN works out the minimum date in the column Date***
- The ***WHEREclause filters the dataset to only perform calculations on Landing_Outcome Success (drone ship)***

In [14]:
```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

Out[14]:

	firstsuccessfull_landing_date
0	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

In [15]:
```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

Out[15]:

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- We used wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.
- a much harder query I must say, we used subqueries here to produce the results. The ***LIKE '%foo%' wildcard shows that in the record the foo phrase is in any part of the string in the records for example.***

List the total number of successful and failure mission outcomes

In [16]:
```
task_7a = '''
          SELECT COUNT(MissionOutcome) AS SuccessOutcome
          FROM SpaceX
          WHERE MissionOutcome LIKE 'Success%'
          '''

task_7b = '''
          SELECT COUNT(MissionOutcome) AS FailureOutcome
          FROM SpaceX
          WHERE MissionOutcome LIKE 'Failure%'
          '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

Out[16]:

The total number of successful mission outcome is:

	successoutcome
0	100

The total number of failed mission outcome is:

	failureoutcome
0	1

Boosters Carried Maximum Payload

- Using the word ***DISTINCT in the query means that it will only show Unique values in the Booster_Version column from tblSpaceX***
- ***GROUP BY puts the list in order set to certain condition.***
- ***DESC means its arranging the dataset into descending order.***

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

In [17]:
```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

Out[17]:

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

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

In [18]:
```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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. here

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

In [19]:
```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

Out[19]:

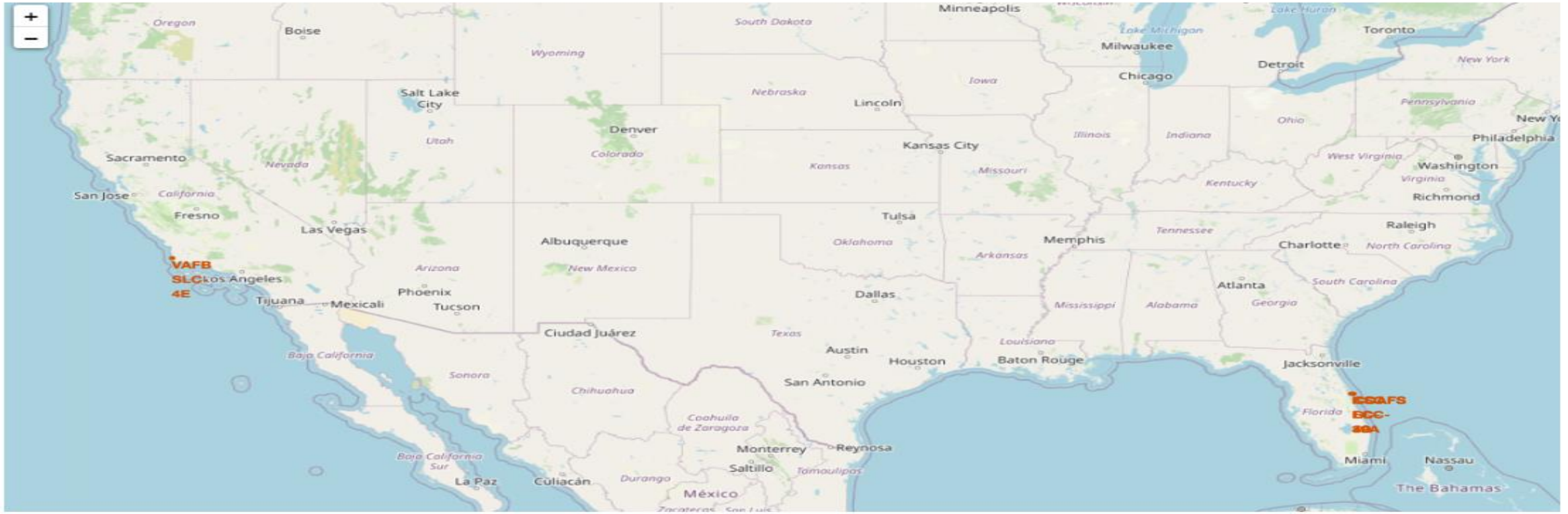
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

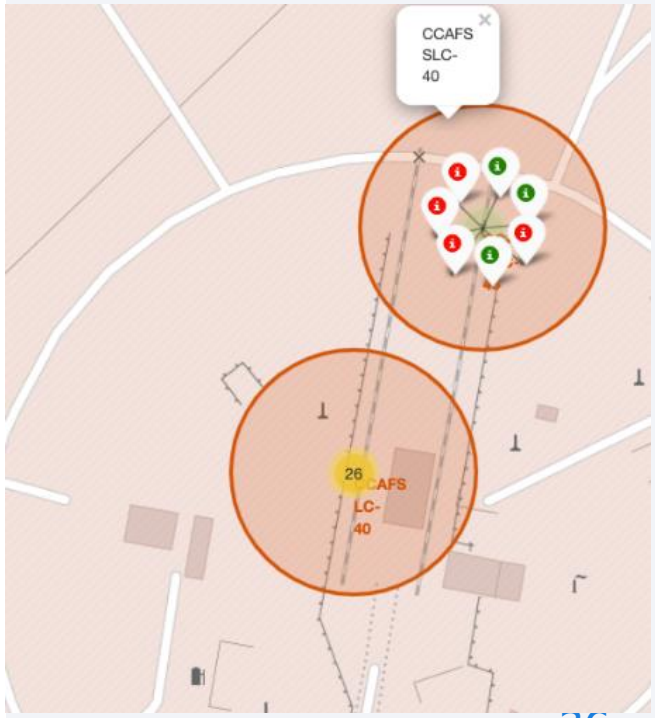
Global Map Markers

- ***We can see that the SpaceX launch sites are in the United States of America coasts.Florida and California***



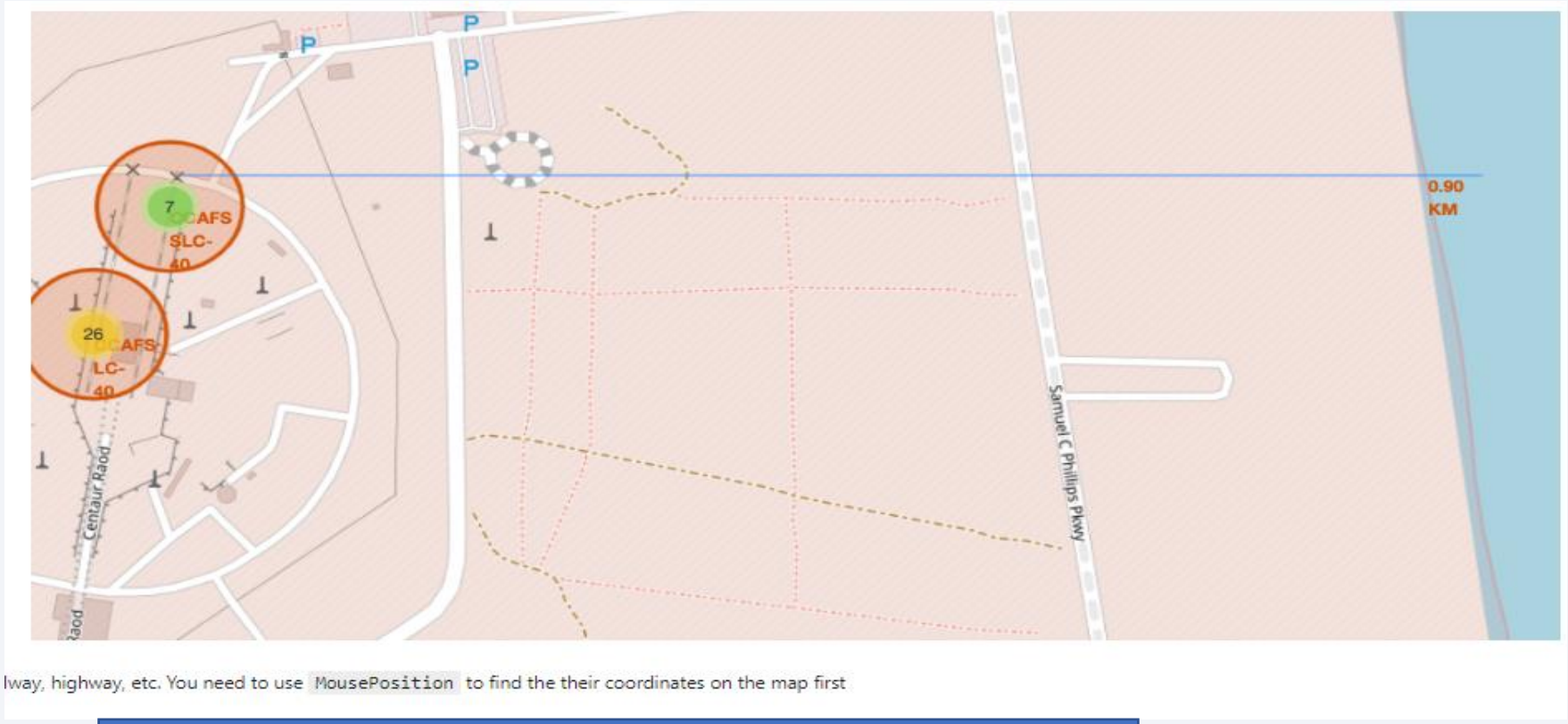
Colour Labelled Markers

- *Green Marker shows successful Launches andRed Markershows Failures* findings on the screenshot

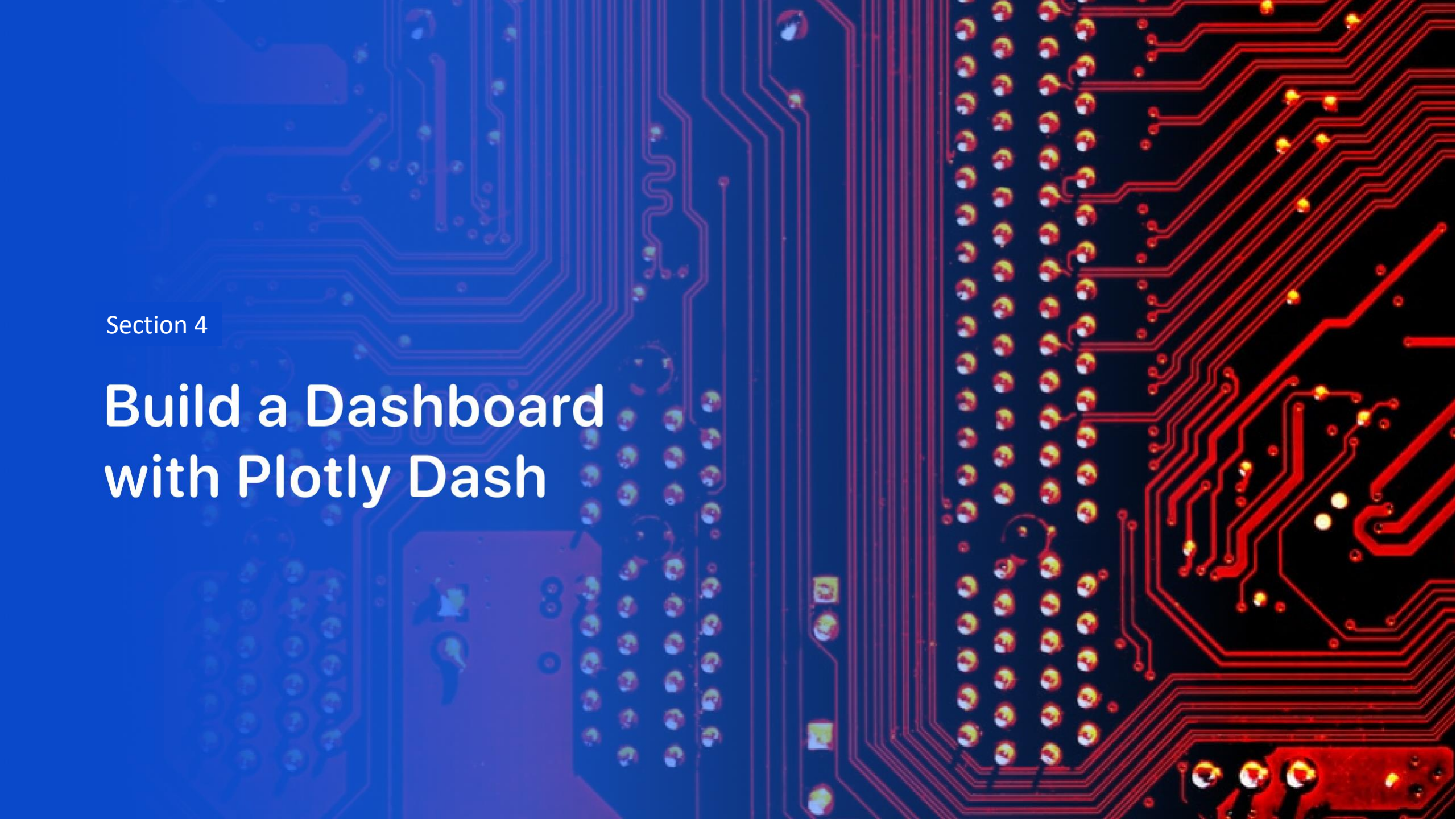


I be able to easily identify which launch sites have relatively high success rates.

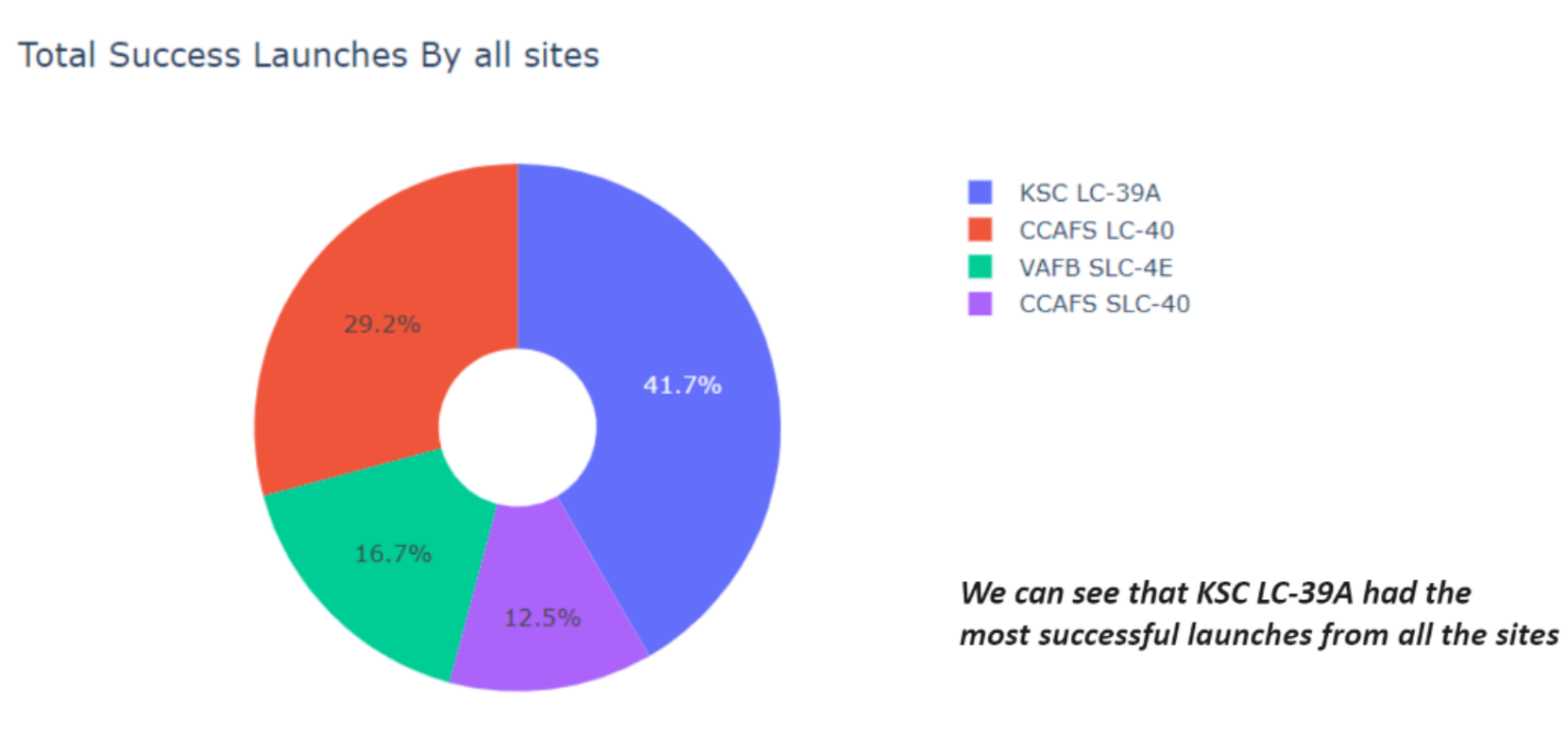
Launch Site to Distance to Landmarks



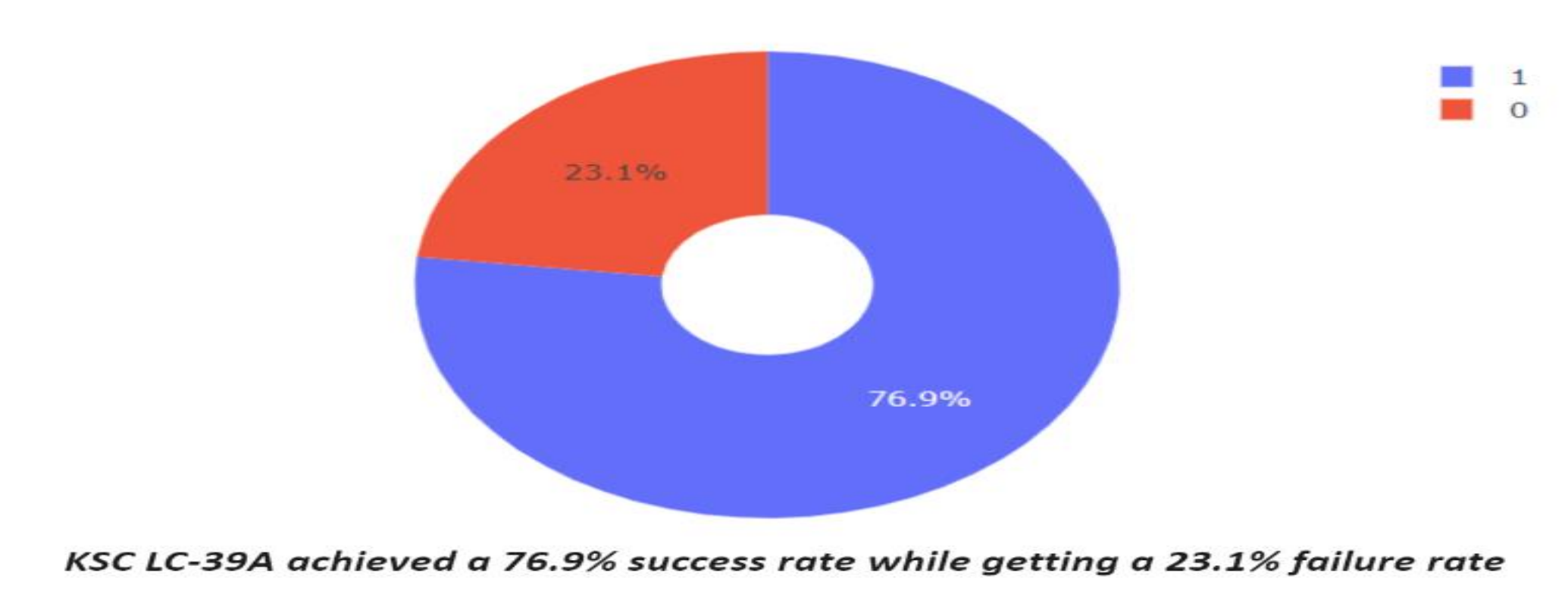
Distance to coast


Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site

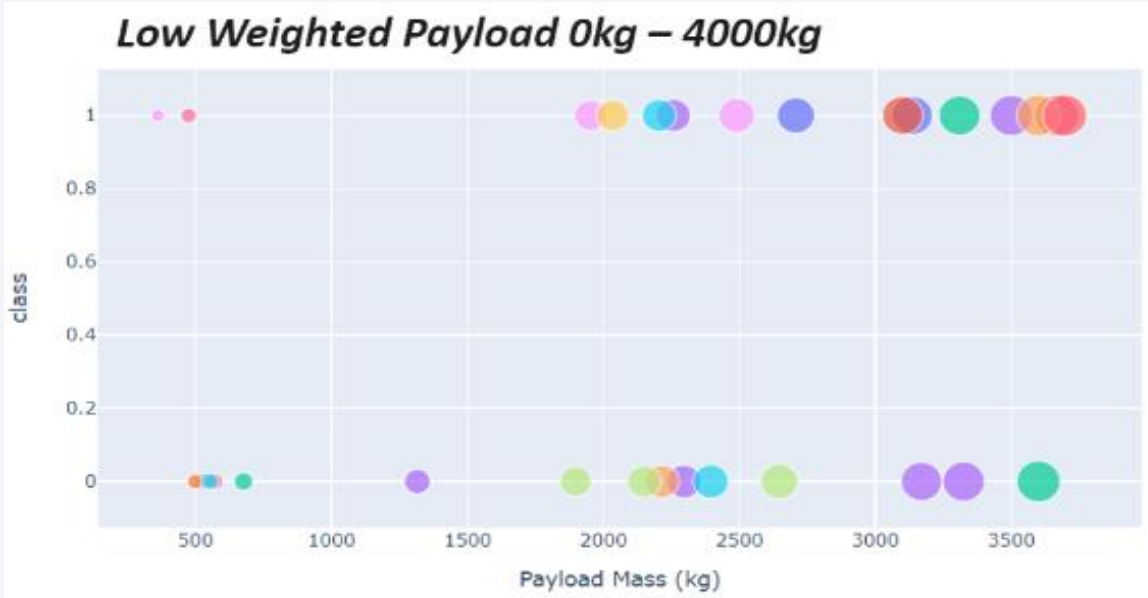


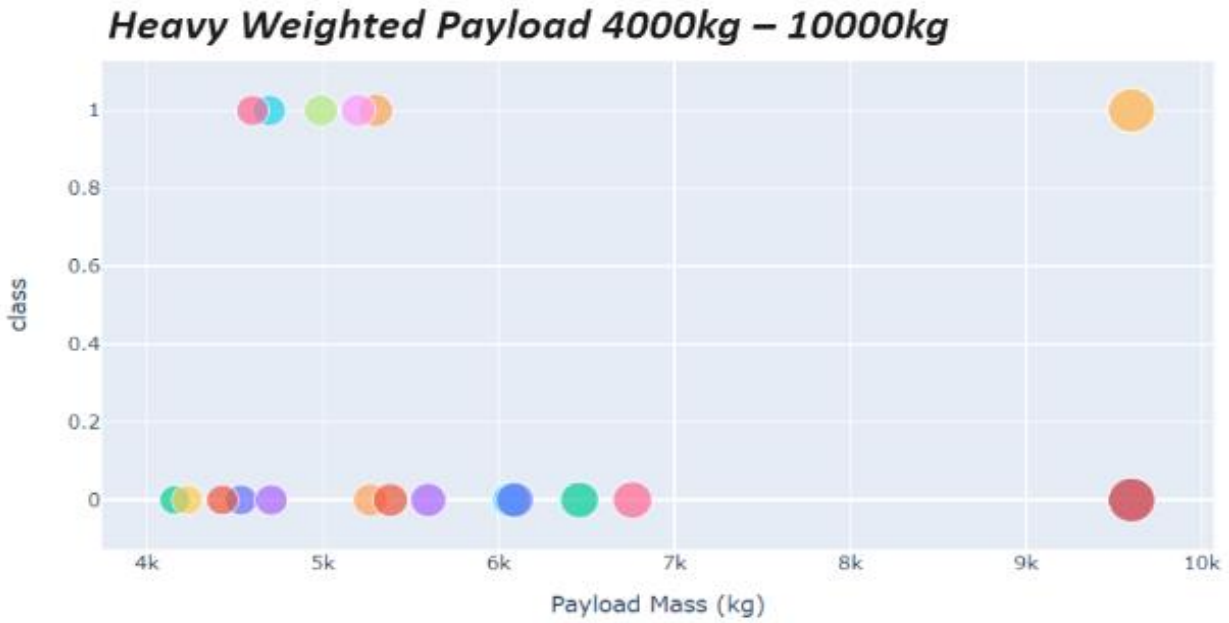
Pie chart for the launch site with highest launch success ratio



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

Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy.

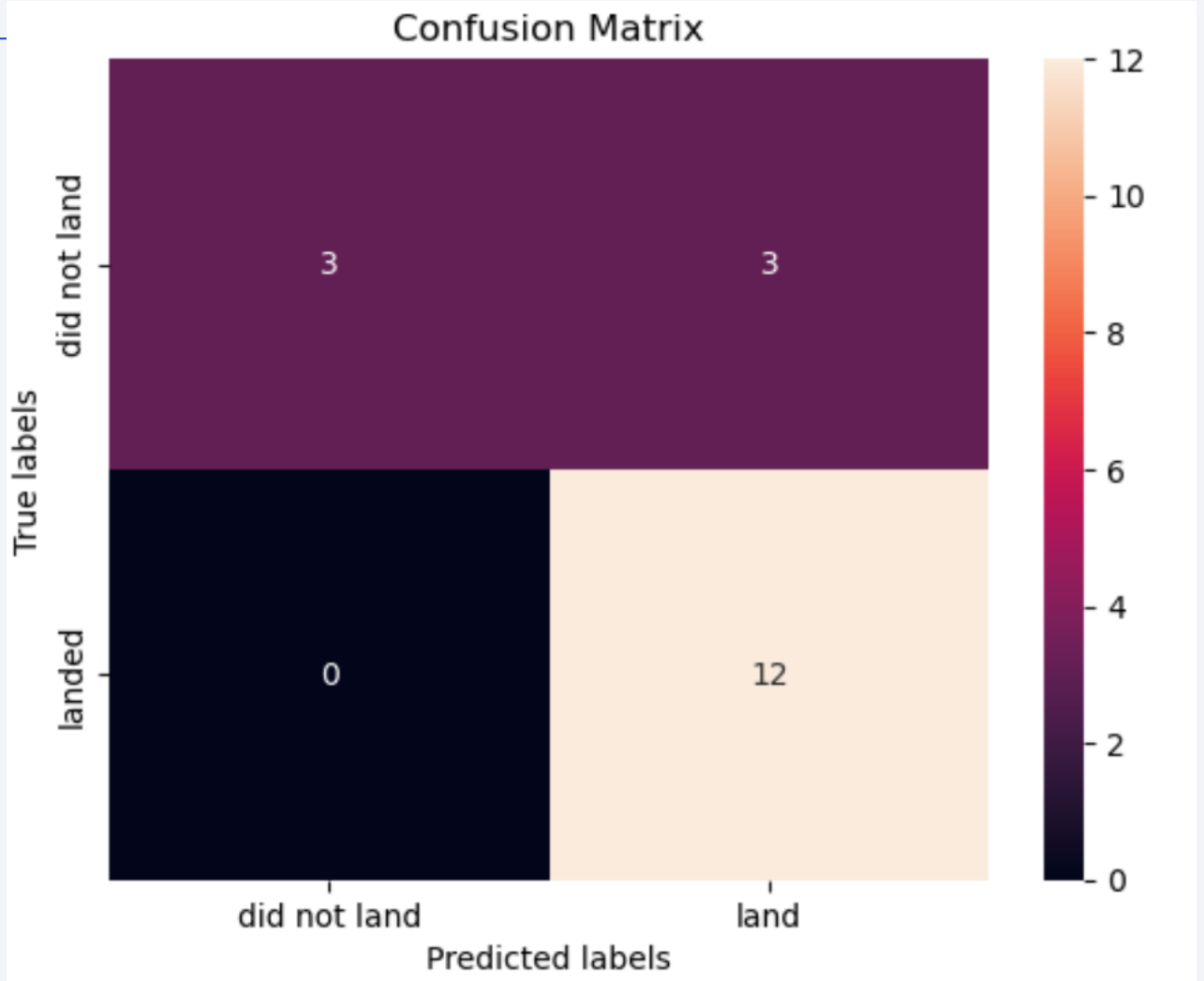
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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

- Examining the confusion matrix, we see that Tree can distinguish between the different classes. We see that the major problem is false positives.



Examining the confusion matrix, we see that logistic regression can distinguish between

Conclusions

- For this dataset, the Tree Classifier Algorithm offers the best machine learning results.
- The performance of lighter payloads is superior to that of heavy payloads.The success rate of SpaceX launches is closely correlated with the number of years it will take them to perfect the launches;
- From all the sites, we can see that KSC LC 39A had the most prosperous launches.
- GEO,HEO,SSO,ES L1 Orbit has the highest Success Rate.

Thank you!