IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kaustubh Kulkarni

05-08-2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- The following methodologies were used to analyze data:
  - Data Collection using web scraping and SpaceX AP.
  - Exploratory Data Analysis (EDA), including data wrangling, data visualization and interactive visual analytics.
  - Machine Learning algorithms and Prediction techniques.
- Summary of all results:
  - It was possible to collect valuable data from public sources.
  - EDA allowed to identify which features are the best to predict success of launchings.
  - Machine Learning Prediction showed the best model to predict which characteristics are important to drive this opportunity by the best way, using all collected data.

Introduction

- Project background and context:
  - SpaceX a rocket company launches satellites at low price like 70% less than their competitor since they land their satellites for reusing them to launch.
- Problems you want to find answers:
  - We use the previous data of launches of Falcon 9 rocket to predict the probability of the booster landing back to the pad influenced/corelated with the space launch site, the payload, orbit, mass, landing pad location and the version of the booster.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data from Space X was obtained from 2 sources:
    - Space X API - https://api.spacexdata.com/v4/rockets/
    - Web Scraping – https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy
- Perform data wrangling
  - Collected data was enriched by creating a landing outcome label based on outcome data after summarizing and analyzing features.
- Perform exploratory data analysis (EDA) using visualization and SQL

Methodology

Executive Summary

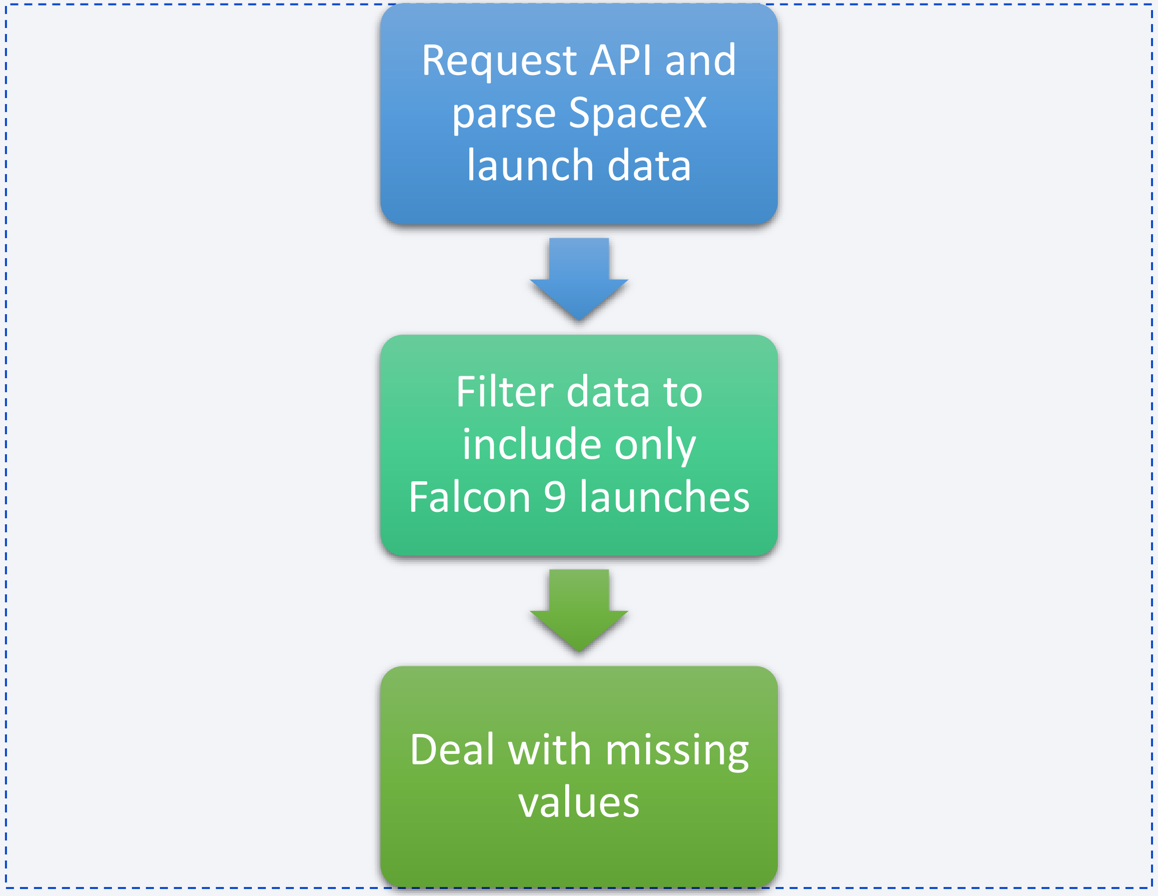
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Data that was collected until this step were normalized, divided in training and test data sets and evaluated by four different classification models, being the accuracy of each model evaluated using different combinations of parameters.

Data Collection

- Data sets were collected from
  - Data Collection was done using get request to the Space X API (https://api.spacexdata.com/v4/rockets/)
  - 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 from (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy) using 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.

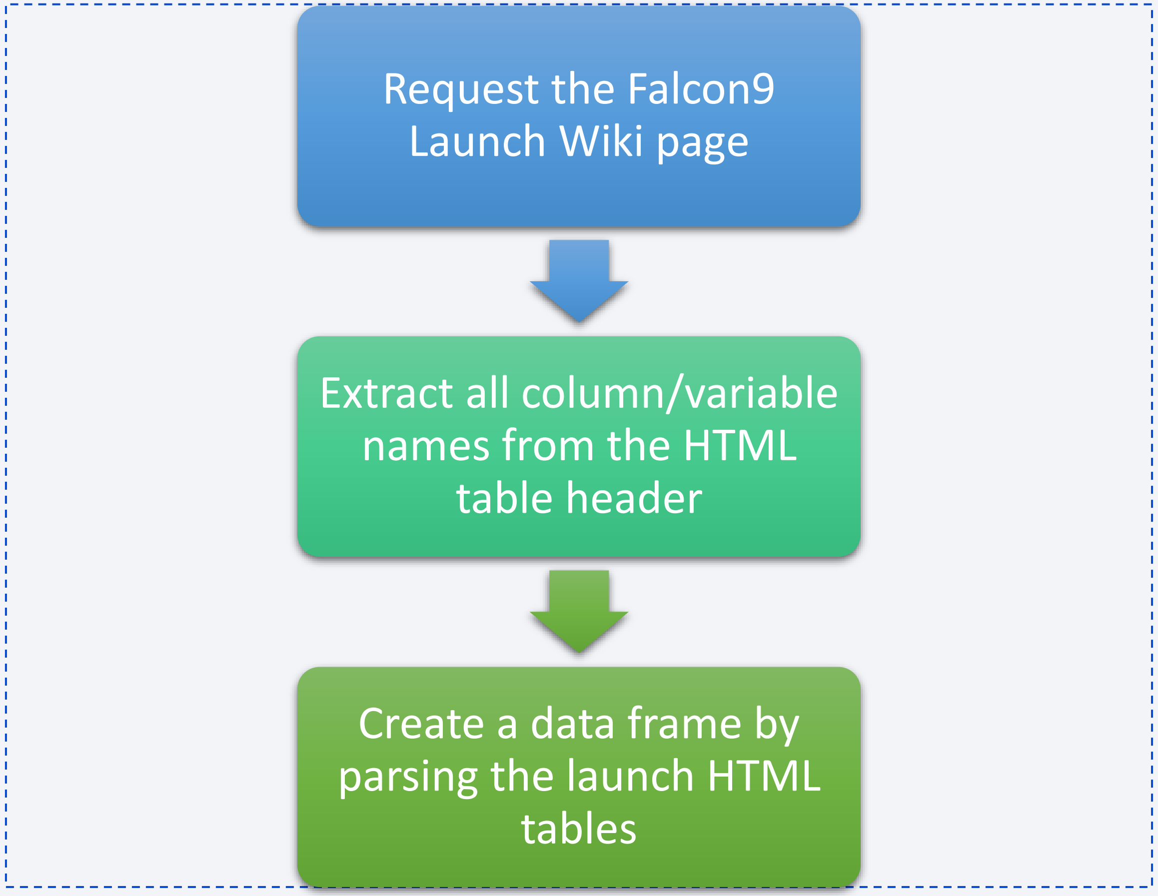
Data Collection – SpaceX API

- Source code:
  - https://github.com/kkulkarni2199/applied_data_science_capstone/blob/main/Data%20Collection%20API.ipynb



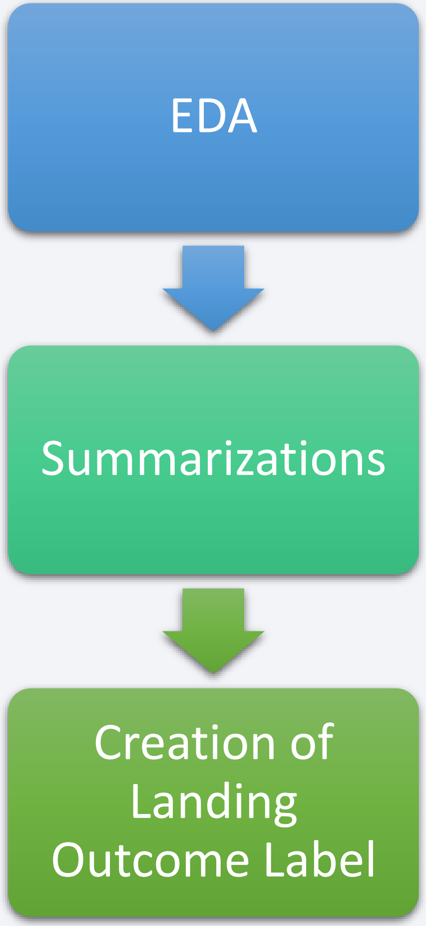
Data Collection - Scraping

- Source Code:
  - https://github.com/kkulkarni2199/applied_data_science_capstone/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb



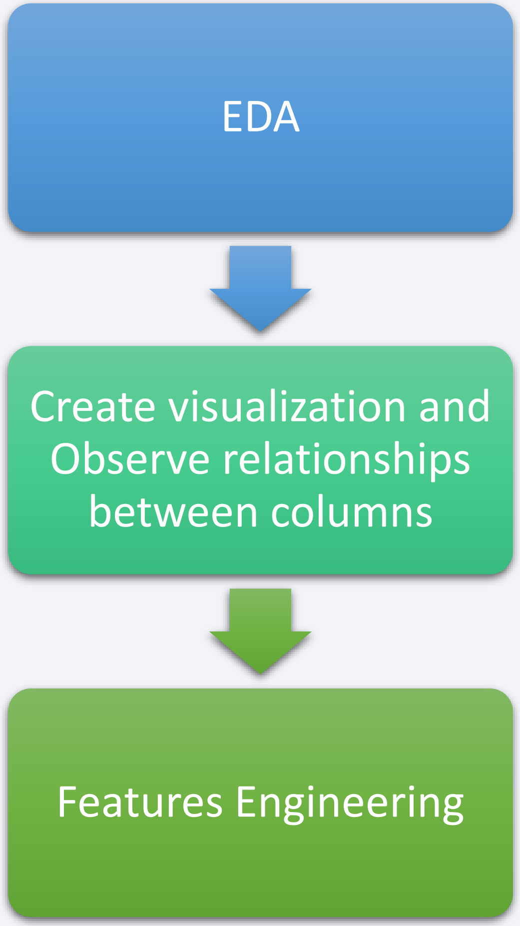
Data Wrangling

- Initially 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 to csv.
- Source code : https://github.com/kkulkarni2199/applied_data_science_capstone/blob/main/Data%20Wrangling.ipynb



EDA with Data Visualization

- We explored the data by visualization the relationship between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type, launch success yearly trend
- Do features engineering on columns that may affect the future success rate.
- Source Code : https://github.com/kkulkarni2199/applied_data_science_capstone/blob/main/EDA%20with%20Data%20Visualization.ipynb



EDA with SQL

- We loaded the dataset into database and performed the following SQL queries:
  - Names of the unique launch sites.
  - Top 5 launch sites whose name begin with string ‘CCA’.
  - Total payload mass carried by boosters launched by NASA (CRS).
  - Average payload mass carried by booster version F9 v1.1.
  - Date when the first successful landing outcome in ground pad was achieved.
  - Names of the boosters which have success and payload mass between 4000 and 6000 kg.
  - Total number of successful and failure mission outcomes.
  - Names of the booster versions which have carried the maximum payload mass.
  - Failed landing outcomes, their booster versions and launch site names for in year 2015.
  - Rank of count of landing outcomes between date 2010-06-04 and 2017-03-20.

Source Code: https://github.com/kkulkarni2199/applied_data_science_capstone/blob/main/EDA.ipynb

Build an Interactive Map with Folium

- Markers, circles, lines and marker clusters were used with Folium Maps:
  - Markers indicate points like launch sites.
  - Circles indicate highlighted areas around specific coordinates, like NASA Johnson Space Center.
  - Marker clusters indicates groups of events in each coordinates, like launches in launch site.
  - Lines are used to indicate distances between two coordinates.

- Source Code : https://github.com/kkulkarni2199/applied_data_science_capstone/blob/main/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

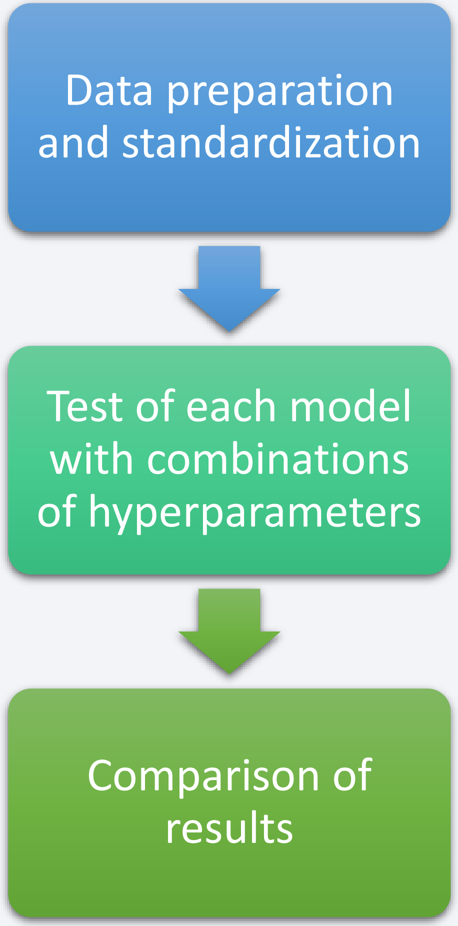
- The following graphs and plots were used to visualize data:
  - Percentage of launches by site
  - Payload range
- This combination allowed to quickly analyze the relation between payloads and launch sites, helping to identify where is the best place to launch according to payloads.

- Source Code : https://github.com/kkulkarni2199/applied_data_science_capstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Four classifications models were compared: logistic regression, support vector machine, decision tree and K nearest neighbors.

- Source Code : https://github.com/kkulkarni2199/applied_data_science_capstone/blob/main/Machine%20Learning%20Prediction.ipynb

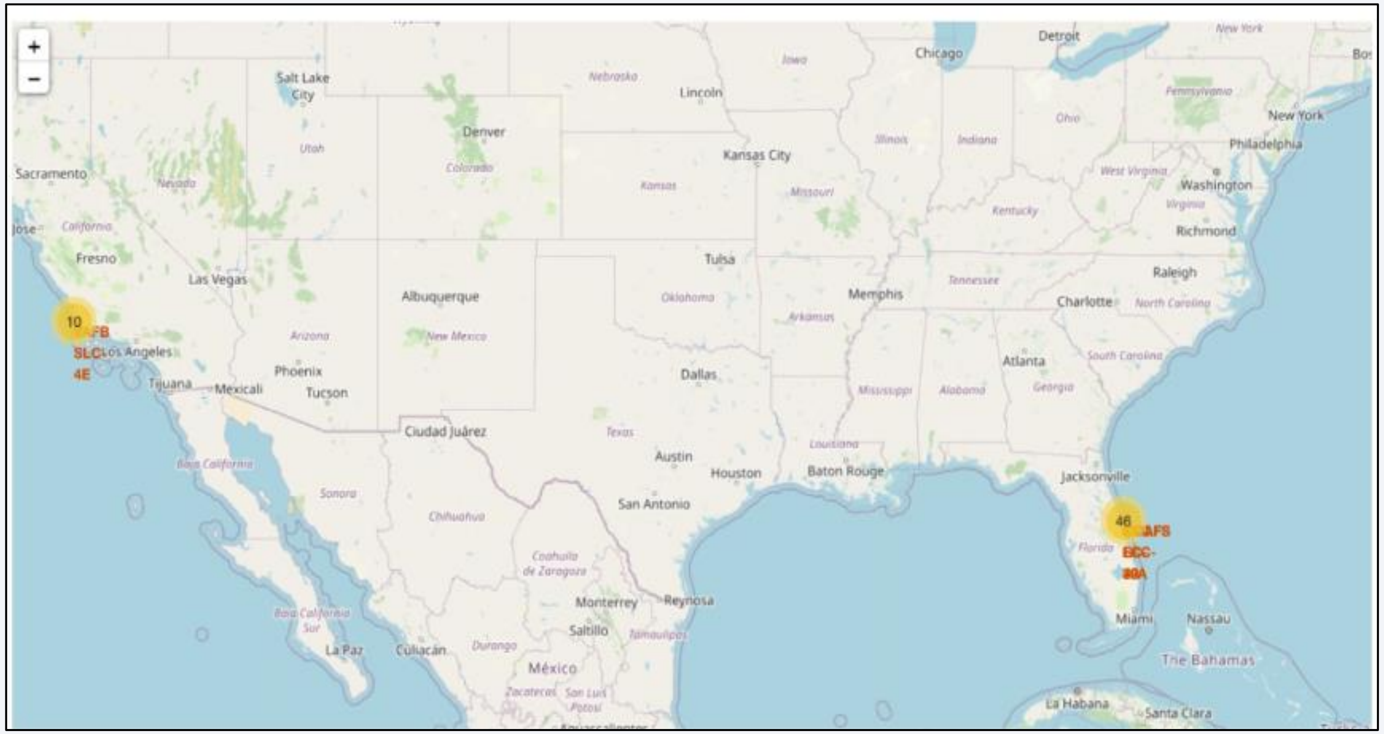


Results

- Exploratory data analysis results
  - Space X uses 4 different launch sites.
  - The first launches were done to Space X itself and NASA.
  - The average payload of F9 v.1.1 booster is 2,928 kg.
  - The first success landing outcome happened in 2015 five years after the first launch.
  - Many Falcon 9 booster versions were successful at landing in drone ships having payload above the average.
  - Almost 100% of mission outcomes were successful.
  - Two booster versions failed at landing in 2015: F9 v1.1 B1012 and F9 v1.1 B1015.
  - The number of landing outcomes became as better as years passed.

Results

- Interactive analytics demo:
  - Using interactive analysis to identify that launch sites use to be in safety places.
  - Most launches happen at east coast launch sites.



Results

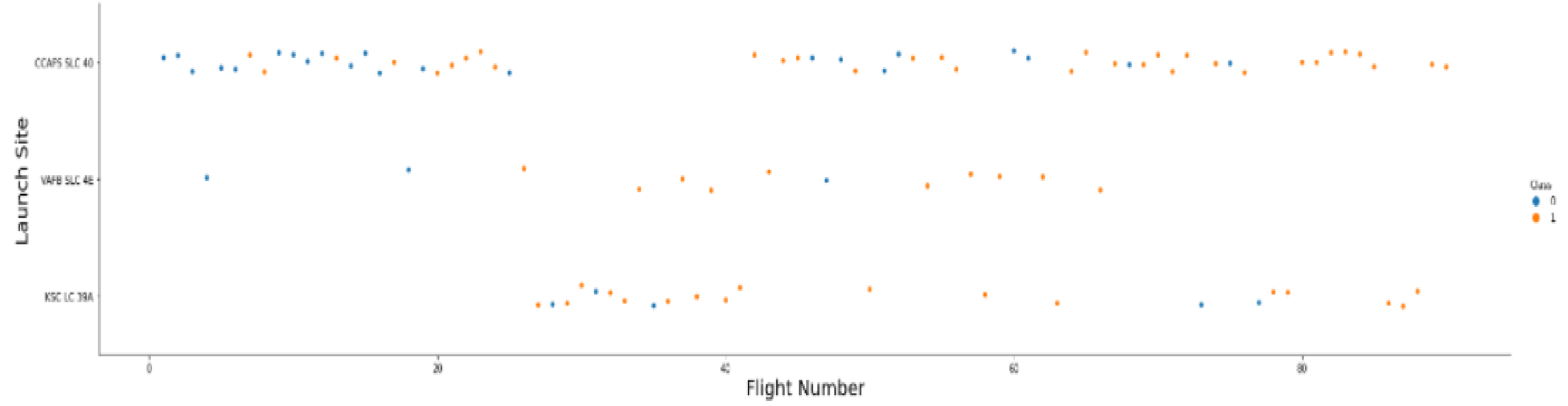
- Predictive analysis results:
  - Predictive Analysis showed that Decision Tree Classifier is the best model to predict successful landings, having over 87% accuracy for test data over 94%.

Section 2

Insights drawn from EDA

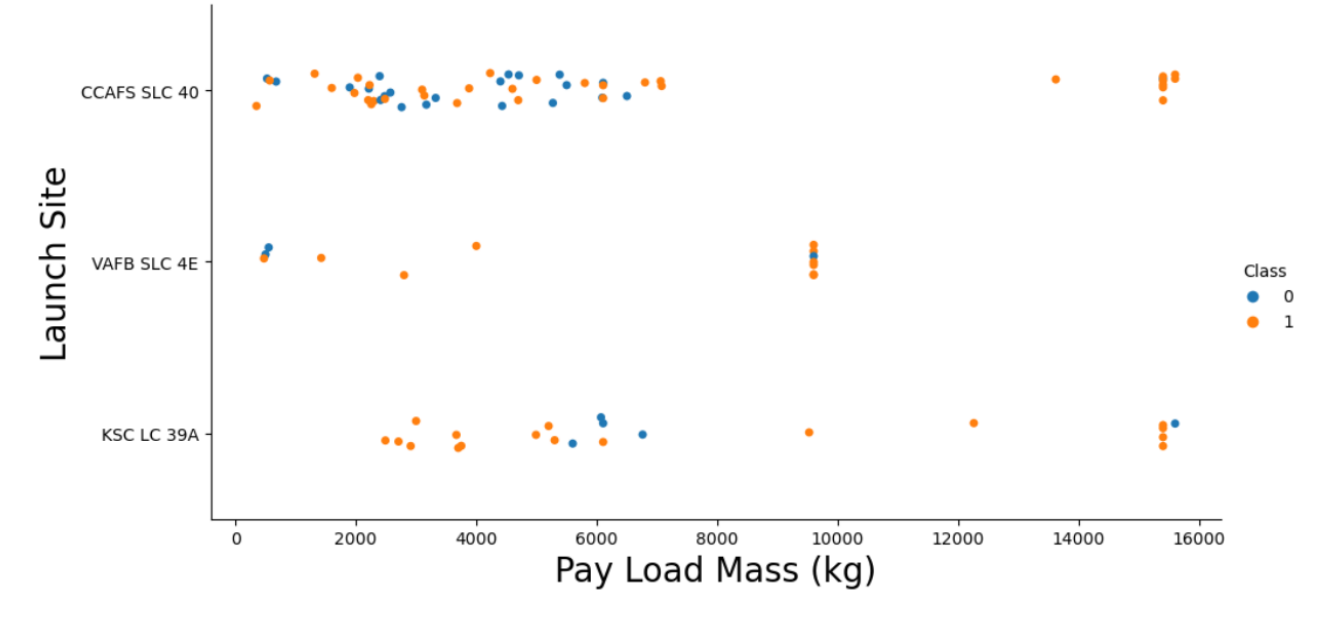
Flight Number vs. Launch Site

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



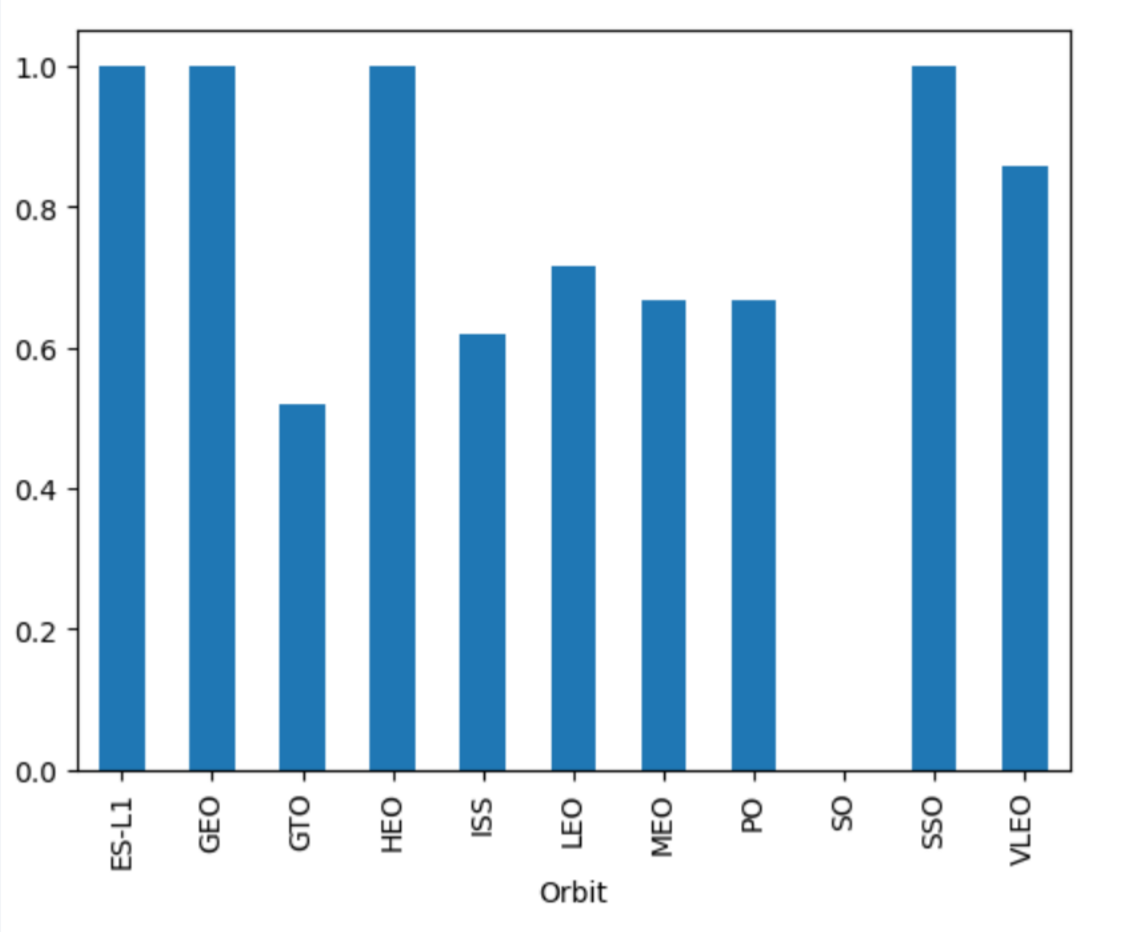
Payload vs. Launch Site

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



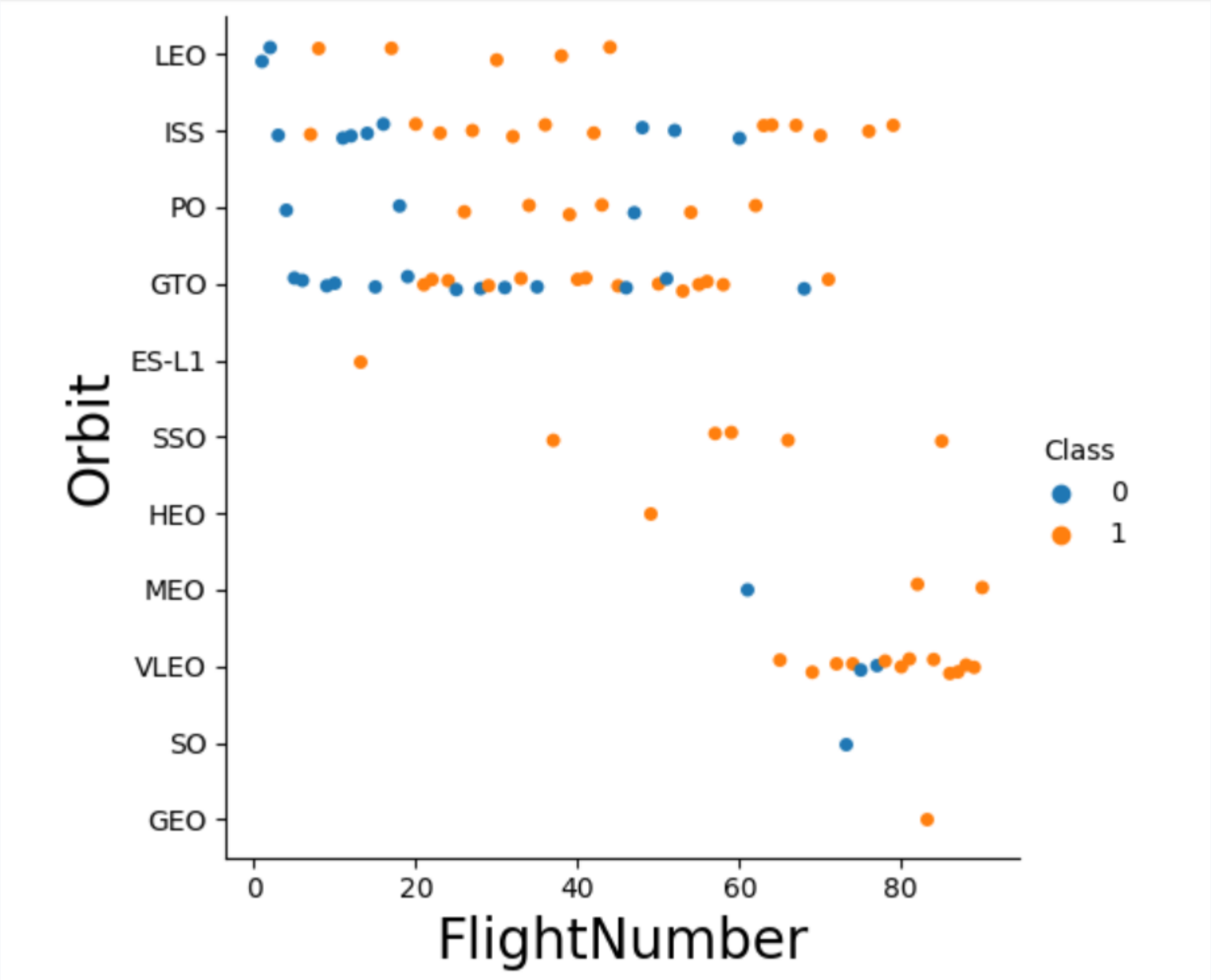
Success Rate vs. Orbit Type

- The biggest success rate happens to orbits:
  - ES-L1
  - GEO
  - HEO
  - SSO



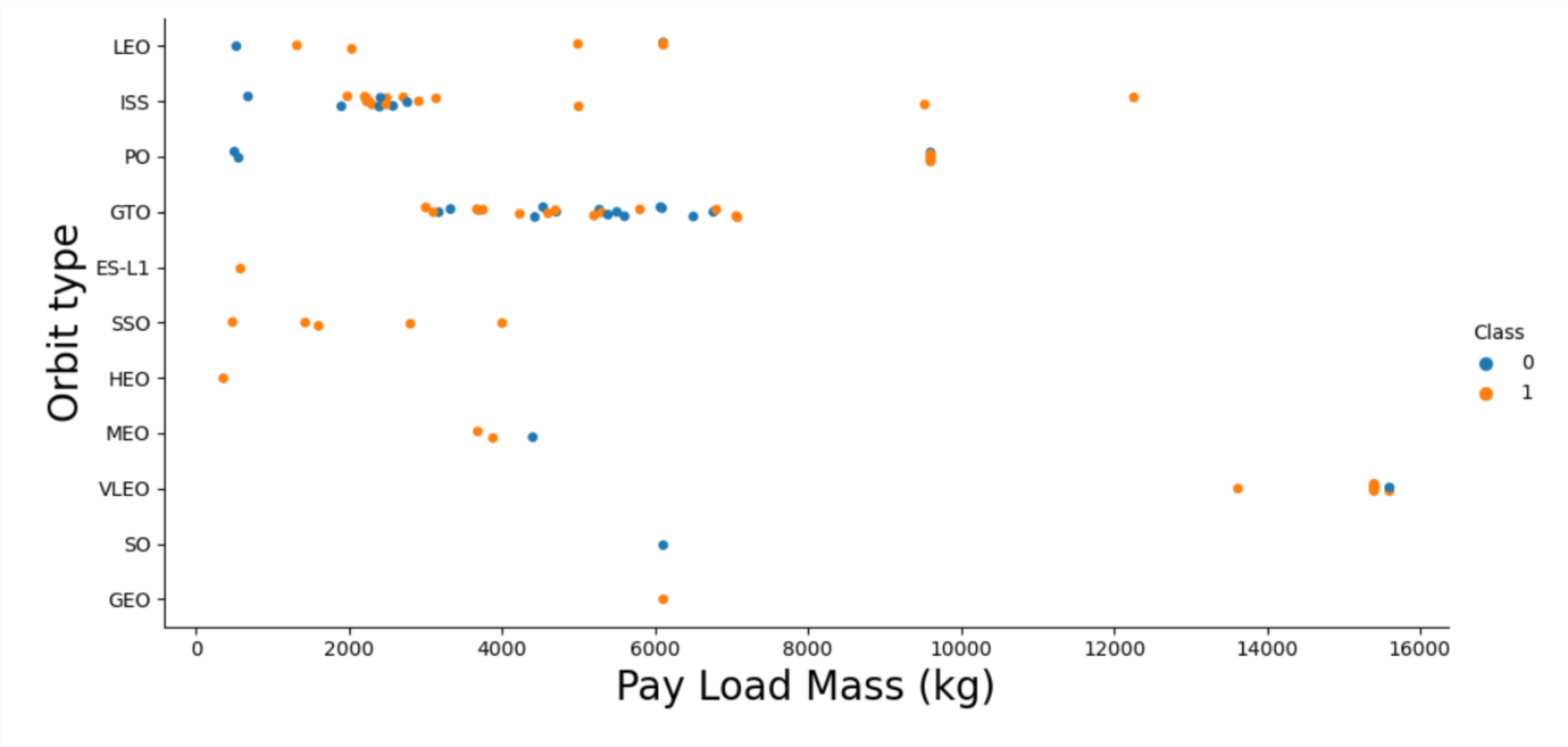
Flight Number vs. Orbit Type

We observed that 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.



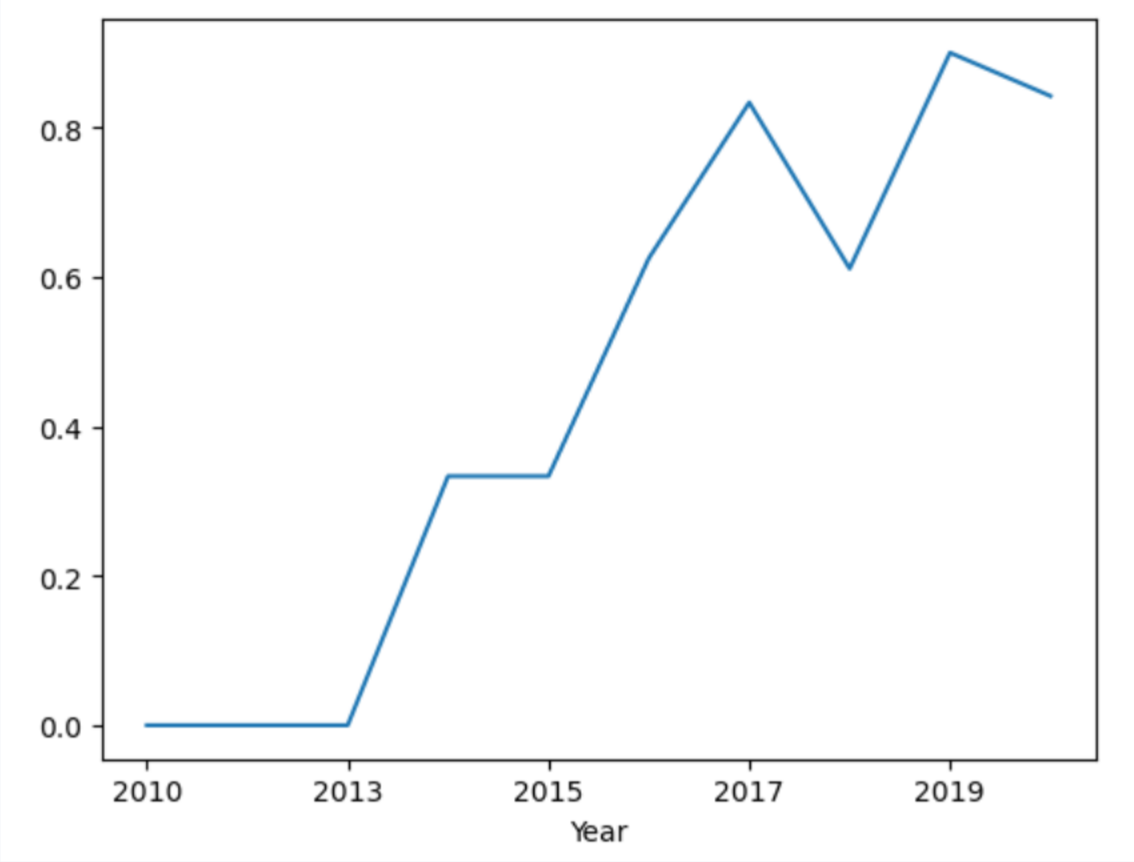
Payload vs. Orbit Type

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



Launch Success Yearly Trend

- Success rate started increasing in 2013 till 2020.



All Launch Site Names

- According to data, there are four launch sites.

Launch Sites
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC – 4E

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

Launch Site Names Begin with 'CCA'

- We used the **LIKE** operator to show the names of launch sites that begin with 'CCA':

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

- 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)

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

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

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

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

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.

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)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

```
The total number of failed mission outcome is:
```

Out[16]:

	failureoutcome
0	1

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

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.

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)
```

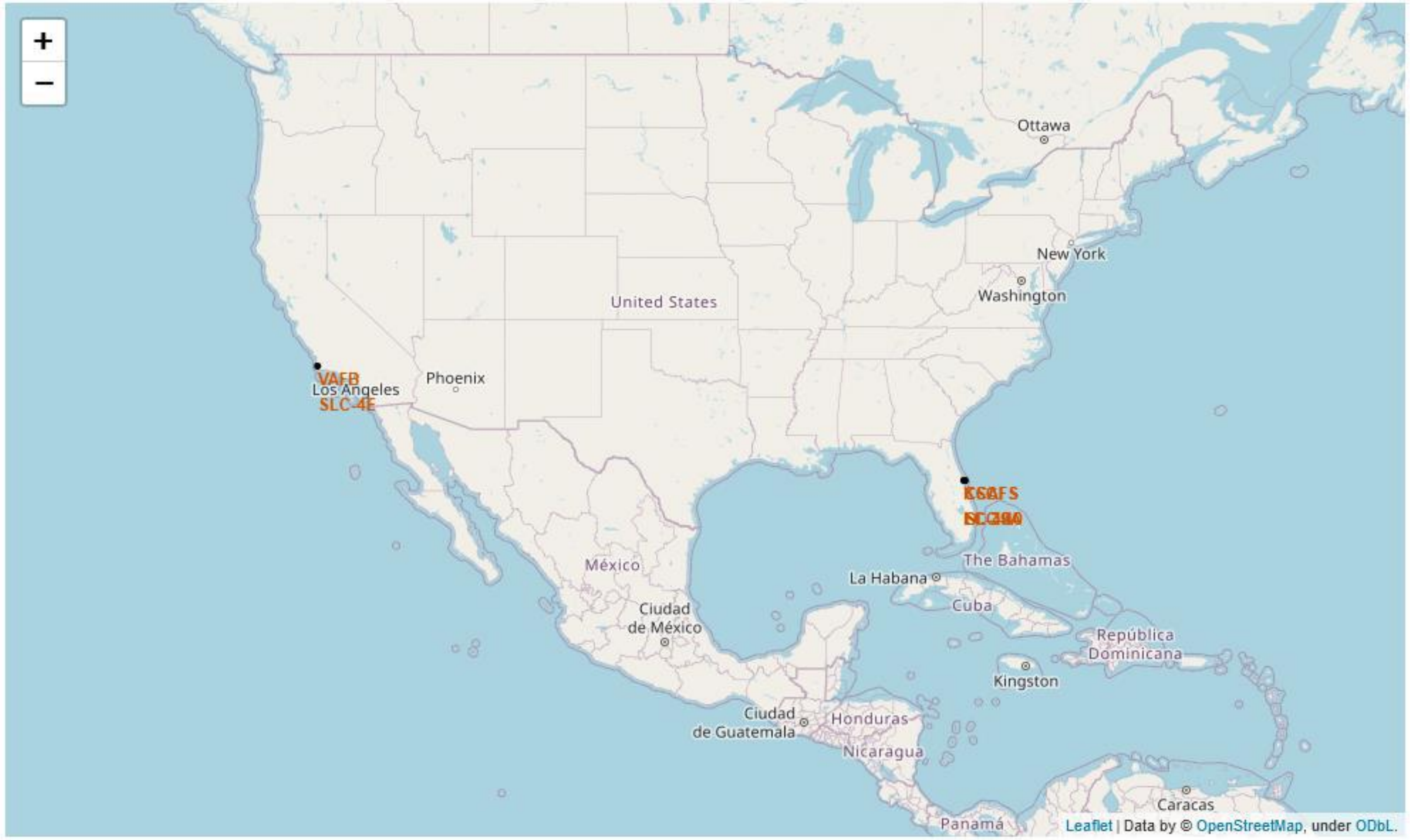
Out[19]:

	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

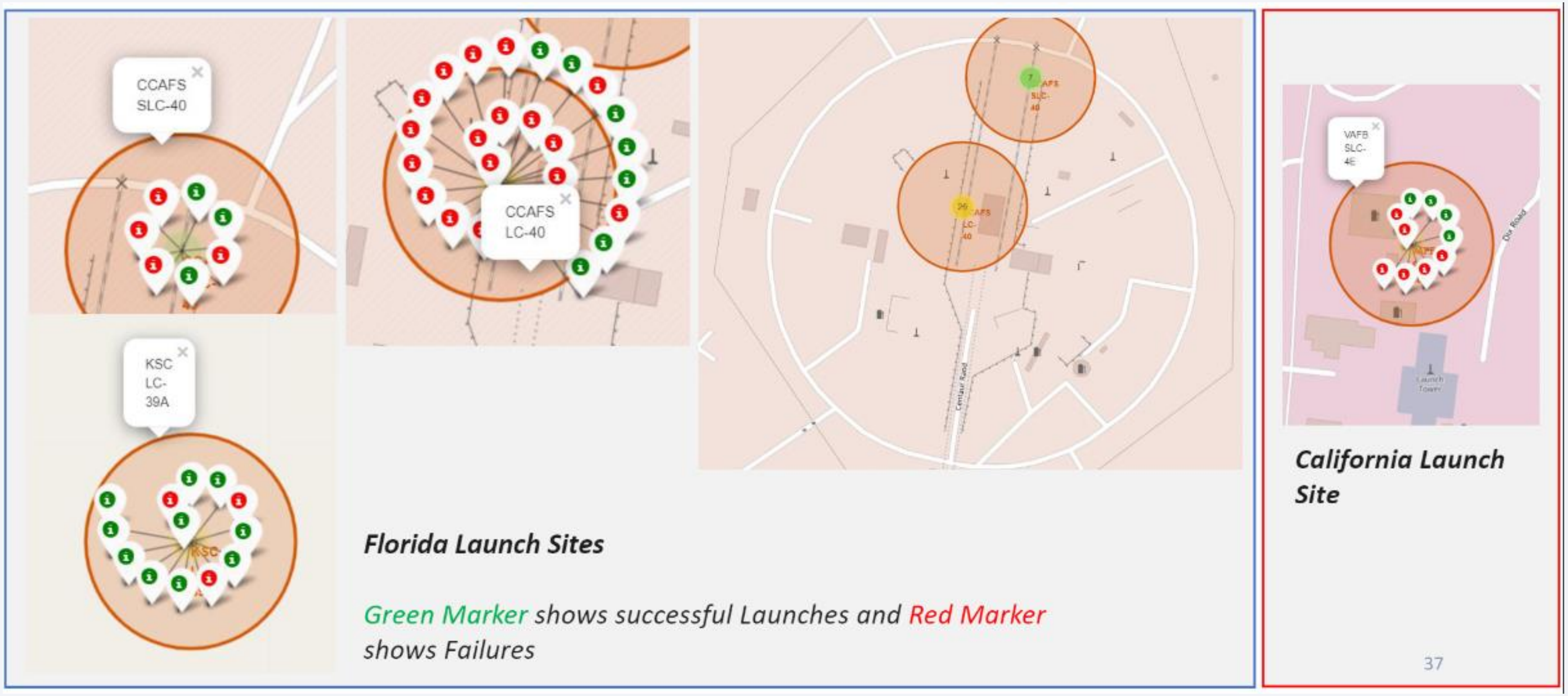
Launch Sites Proximities Analysis

All launch sites

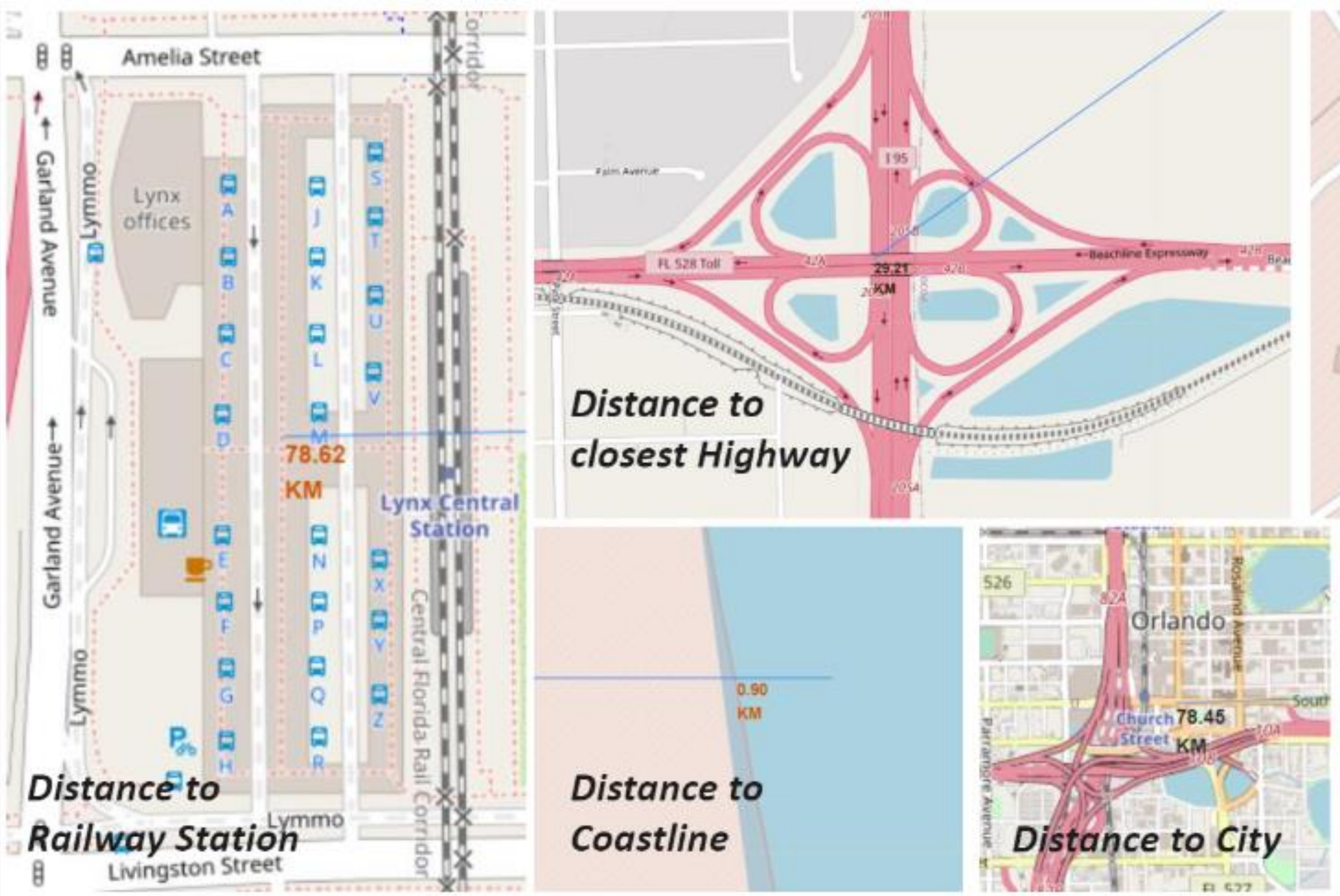


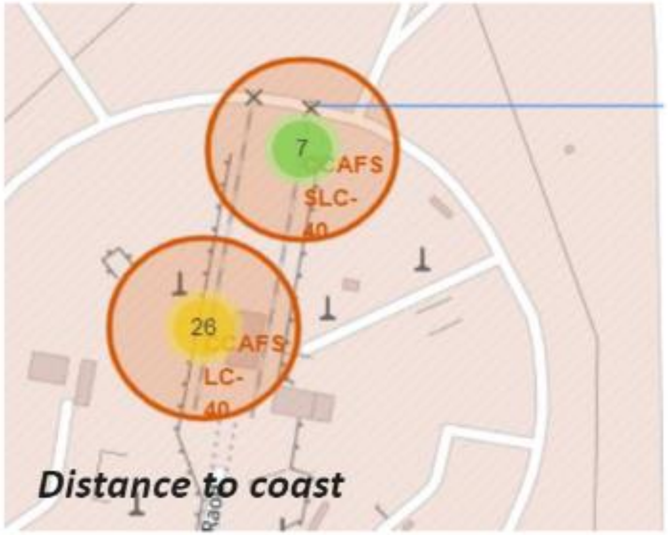
- Launch sites are near sea, probably by safety, but not too far from roads and railroads.

Markers showing launch sites with color labels



Launch Site distance to landmarks



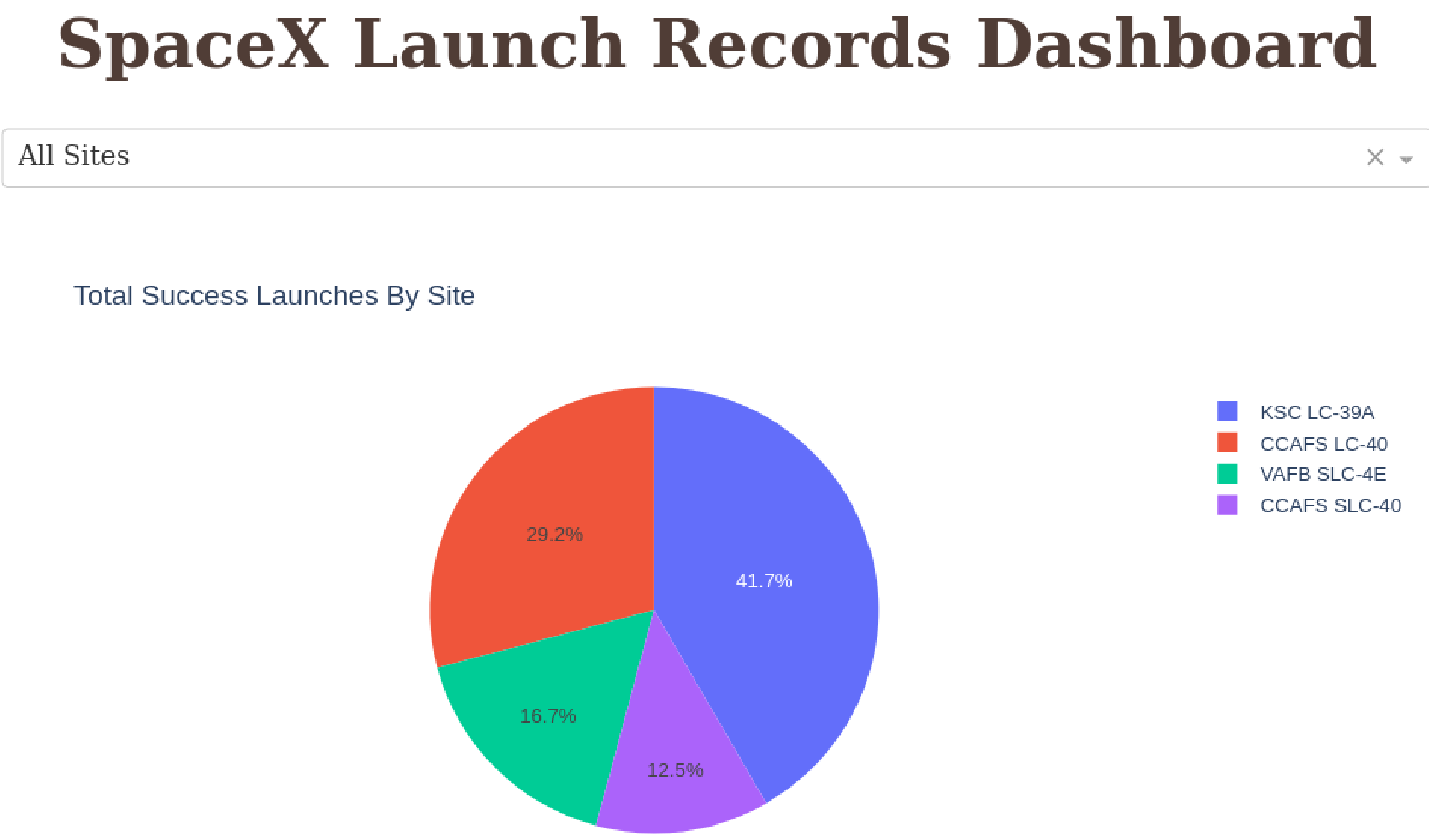


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

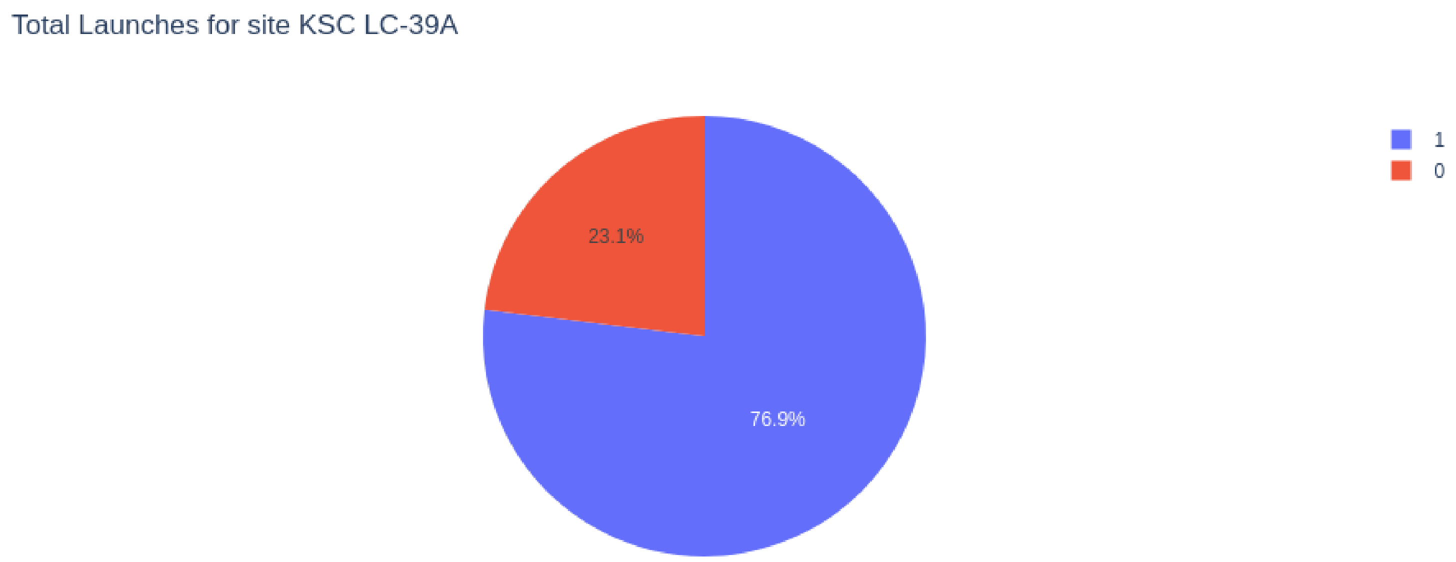
Section 4

Build a Dashboard with Plotly Dash

Successful Launches by Site

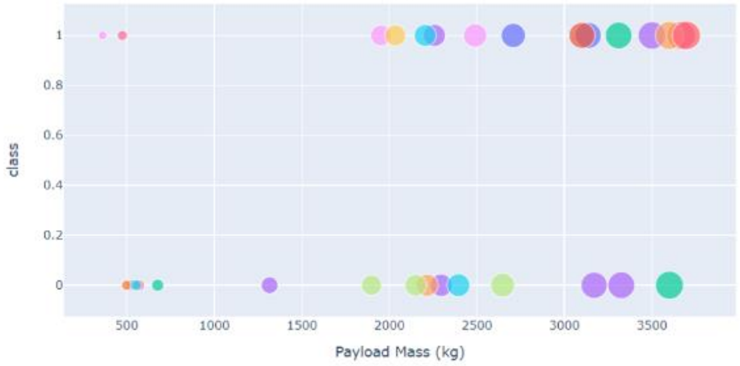


Launch Success Ratio for KSC LC-39A

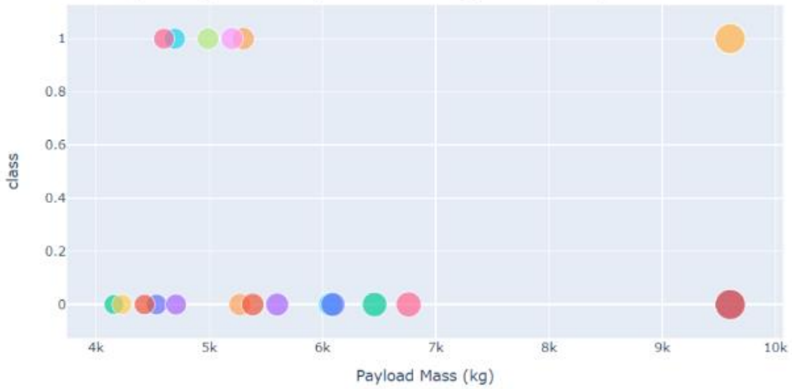


Payload vs Launch Outcomes

Low Weighted Payload 0kg – 4000kg



Heavy Weighted Payload 4000kg – 10000kg



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

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Four classification models were tested, and the model with the highest classification accuracy is Decision Tree Classifier, which has accuracies over than 87%.

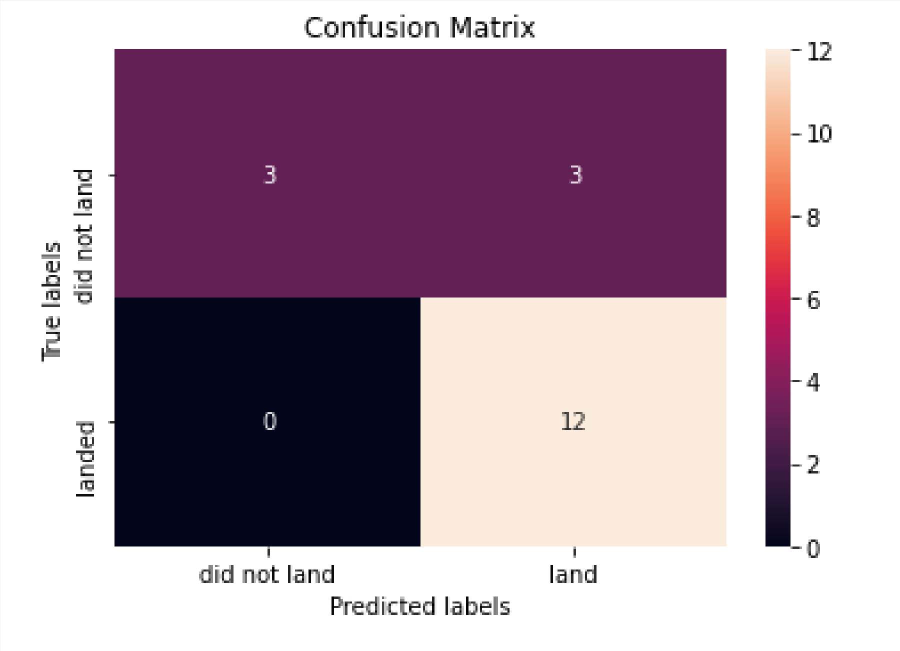
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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

- Different data sources were analyzed, refining conclusions along the process.
- The best launch site is KSC LC-39A.
- Launches above 7,000kg are less risky.
- Although most of mission outcomes are successful, successful landing outcomes seem to improve over time, according the evolution of processes and rockets.
- Decision Tree Classifier can be used to predict successful landings and increase profits.

Thank you!