IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chaiyasit Suebmak
21.03.2022

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:
  - SpaceX API + Scraping from Wikipedia thru BS4 package
- Perform data wrangling
  - One-hot encoding for categorial features.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Perform exploratory Data Analysis and determine Training Labels
  - Find best Hyperparameter for SVM, Classification Trees and Logistic Regression

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

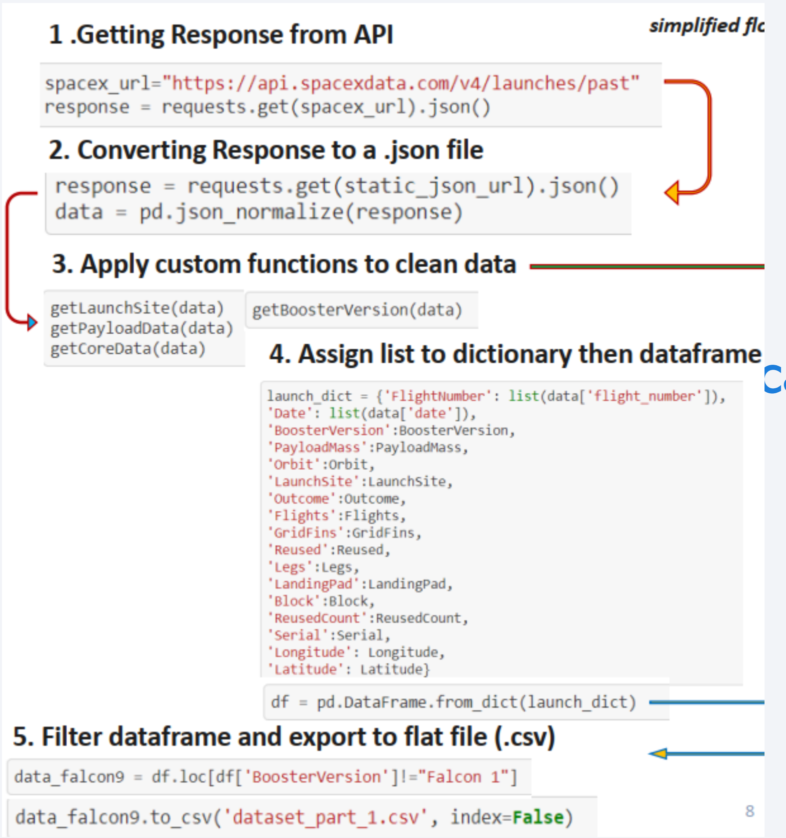
Data collection was done using get request to the SpaceX API.
Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
We then cleaned the data, checked for missing values and fill in missing values where necessary.
In addition, 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

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook
- https://github.com/chaiysue/ibm_data_science_capstone_spacex/blob/c176eb5b874208456ddf1ce3aed365da585494bc/01%20Data%20Collection%20API.ipynb

Place here



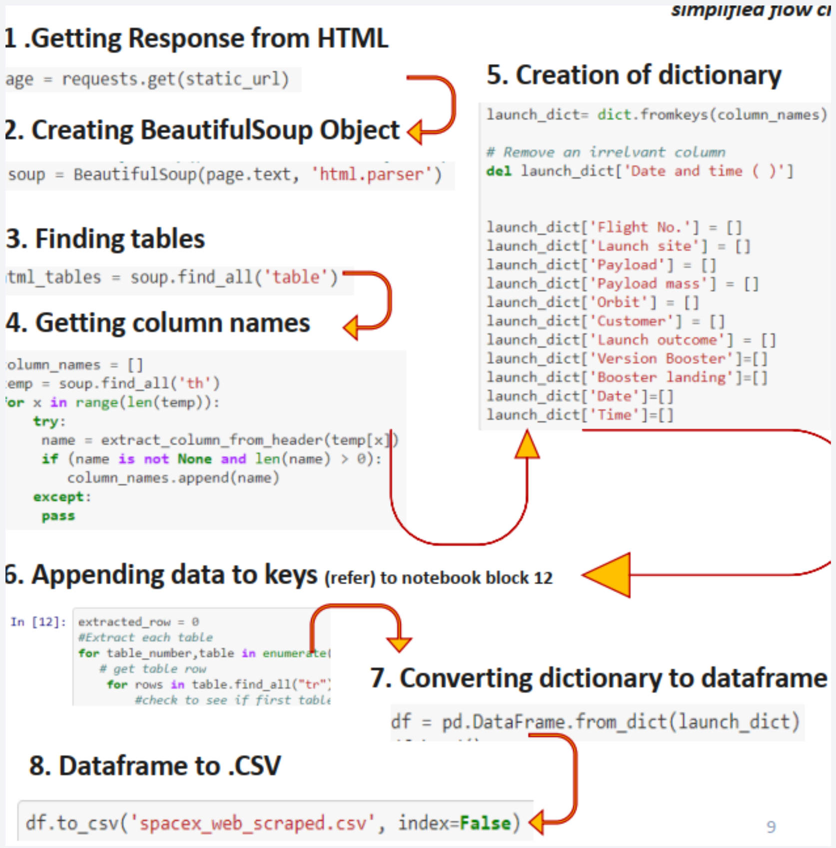
calls

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook https://github.com/chaiysue/ibm_data_science_capstone_spacex/blob/c176eb5b874208456ddf1ce3aed365da585494bc/02%20Data%20Collection%20with%20Web%20Scraping.ipynb

Place

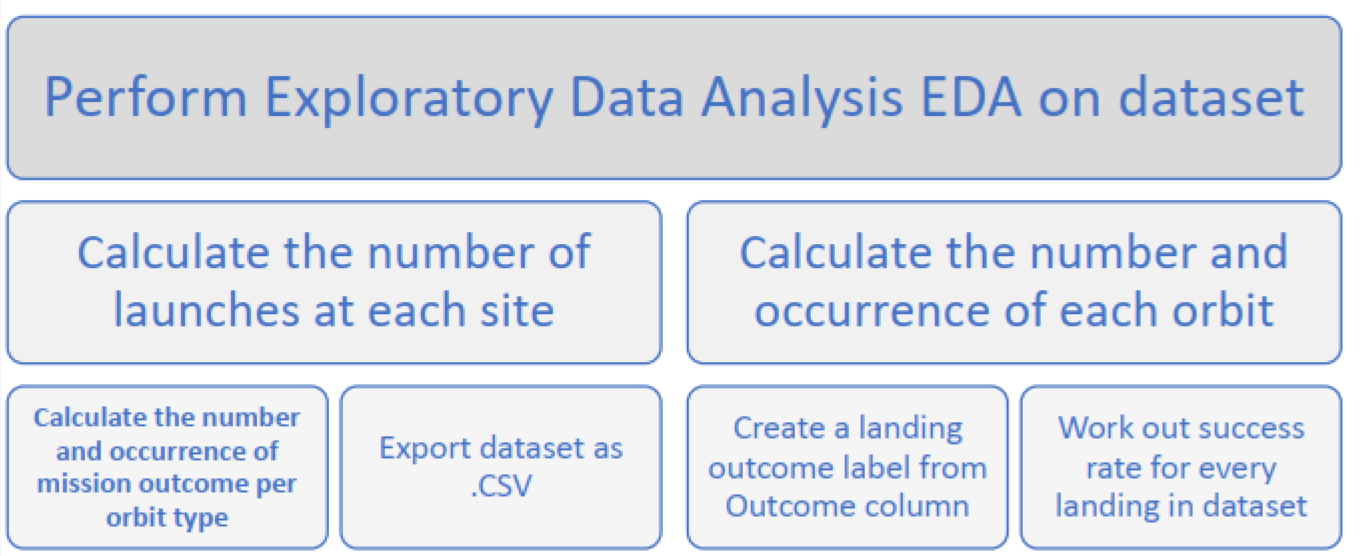


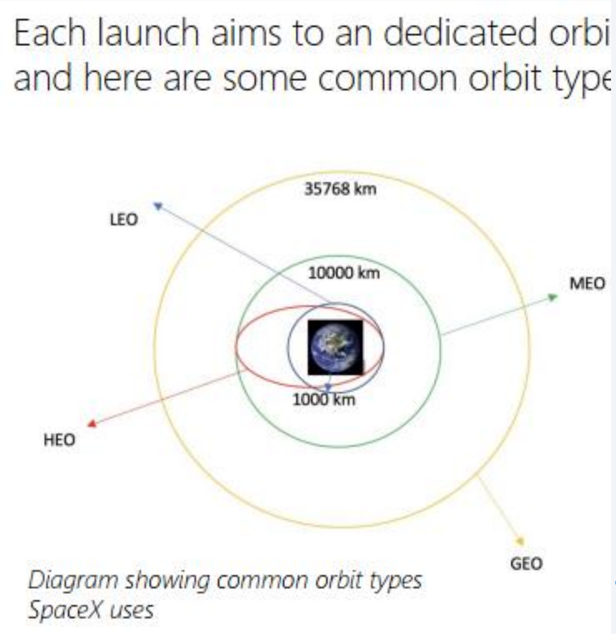
ere

Data Wrangling

https://github.com/chaiysue/ibm_data_science_capstone_spacex/blob/c176eb5b874208456ddf1ce3aed365da585494bc/03%20Data%20Wrangling.ipynb

- Performed exploratory data analysis and determined the training labels.
- Calculated the number of launches at each site, and the number and occurrence of each orbits
- Created landing outcome label from outcome column and exported the results to csv.



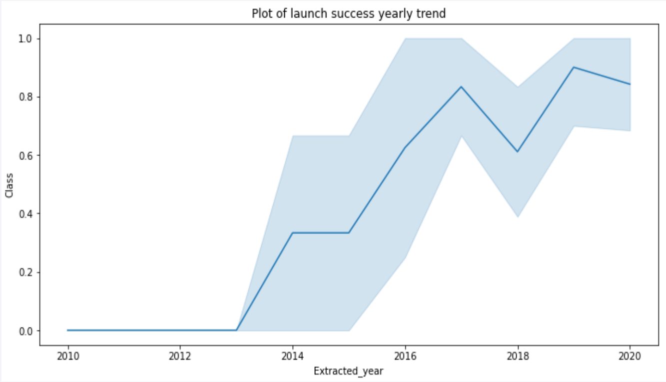


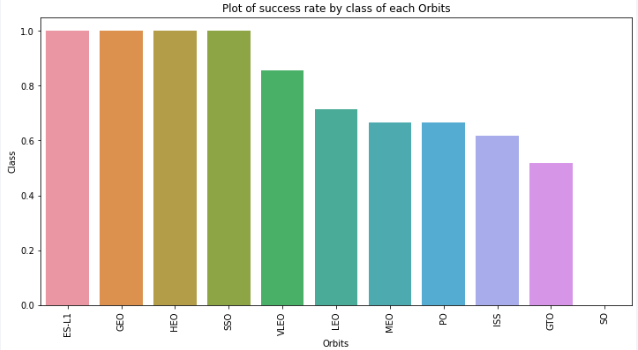
EDA with Data Visualization

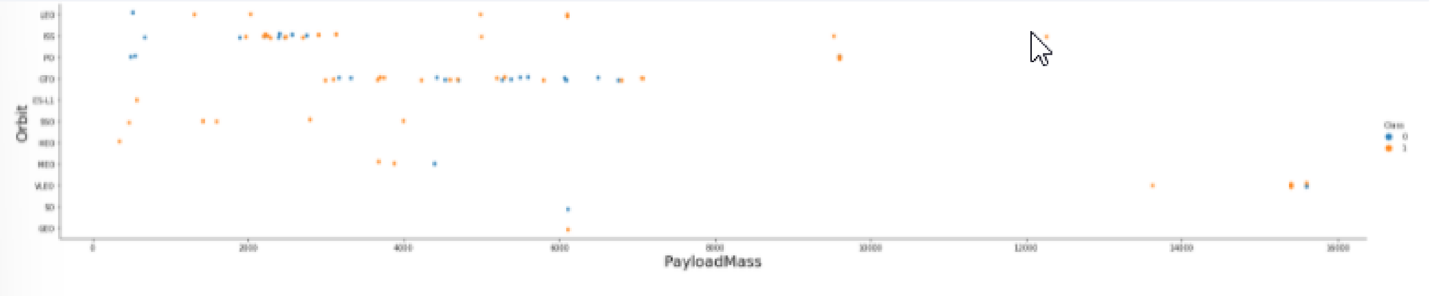
- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.

GitHub URL

https://github.com/chaiysue/ibm_data_science_capstone_spacex/blob/c176eb5b874208456ddf1ce3aed365da585494bc/04%20EDA%20with%20Data%20Visualization.ipynb







Visualize the relationship between Flight Number and Launch Site

EDA with SQL

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

Load CSV to SQLite with Create New Table and Perform Analysis on Table Name SpaceX

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'KSC'
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date where the successful landing outcome in drone ship was achieved.
- Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000
- but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster_versions which have carried the maximum payload mass.
- Listing the records which will display the month names, successful landing_outcomes in ground pad ,booster

Github LINK

https://github.com/chaiysue/ibm_data_science_capstone_spacex/blob/c176eb5b874208456ddf1ce3aed365da585494bc/05%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

To visualize the Launch Data into an interactive map. We took the Latitude and Longitude Coordinates at each launch site and added a Circle Marker around each launch site with a label of the name of the launch site.
We assigned the dataframe launch_outcomes(failures, successes) to classes 0 and 1 with Green and Red markers on the map in a MarkerCluster()
Using Haversine's formula we calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks

Example of some trends in which the Launch Site is situated in.

- 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

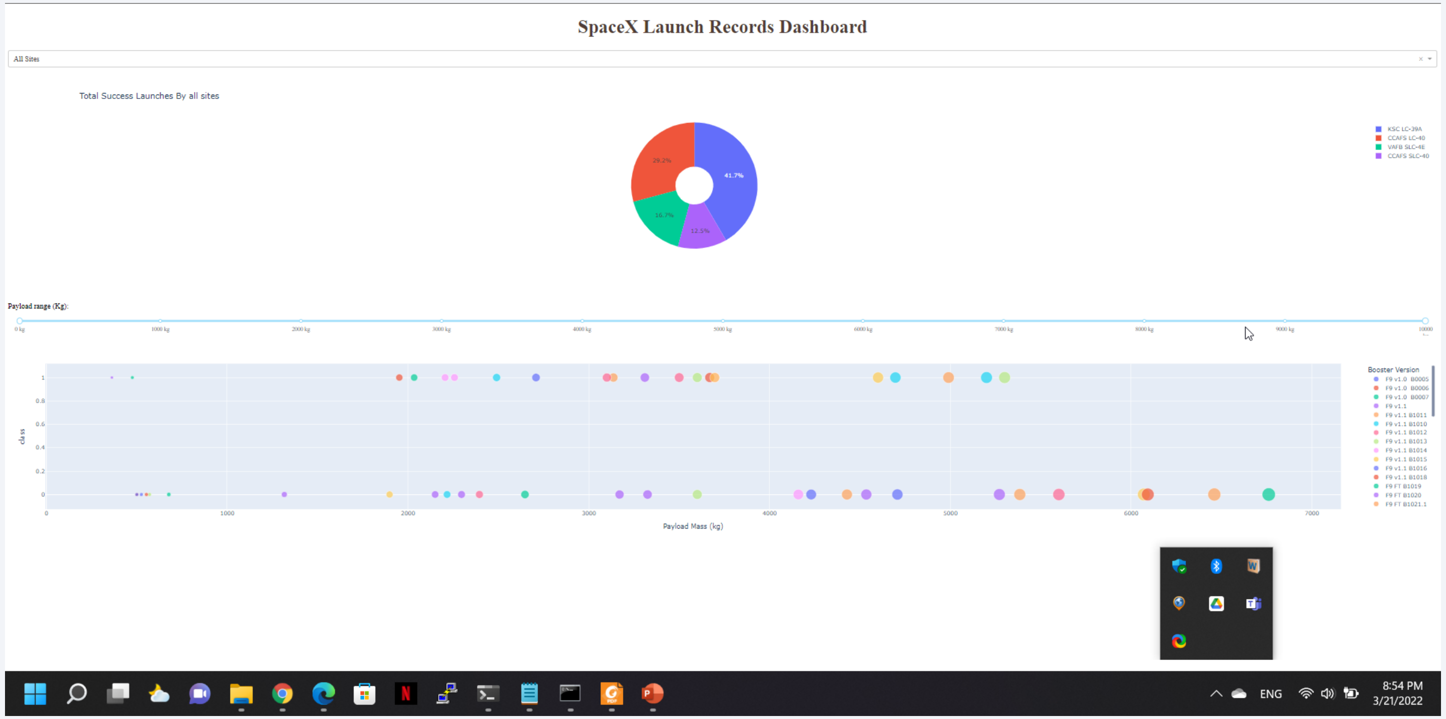
Do launch sites keep certain distance away from cities? Yes

Github link

https://github.com/chaiysue/ibm_data_science_capstone_spacex/blob/c176eb5b874208456ddf1ce3aed365da585494bc/06%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb

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.



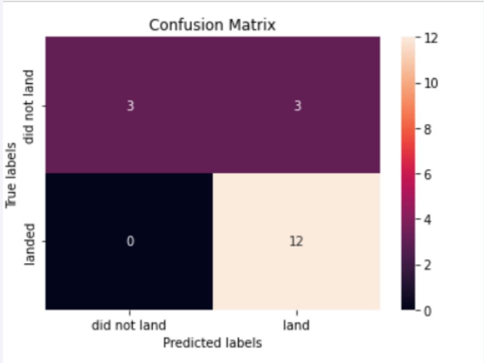
https://github.com/chaiysue/ibm_data_science_capstone_spacex/blob/c176eb5b874208456ddf1ce3aed365da585494bc/app.py

Predictive Analysis (Classification)

- Classification Problem on SpaceX DataSet
- 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.

github

https://github.com/chaiysue/ibm_data_science_capstone_spacex/blob/c176eb5b874208456ddf1ce3aed365da585494bc/07%20Machine%20Learning%20Prediction.ipynb



Best model is DecisionTree with a score of 0.8785714285714284

Results

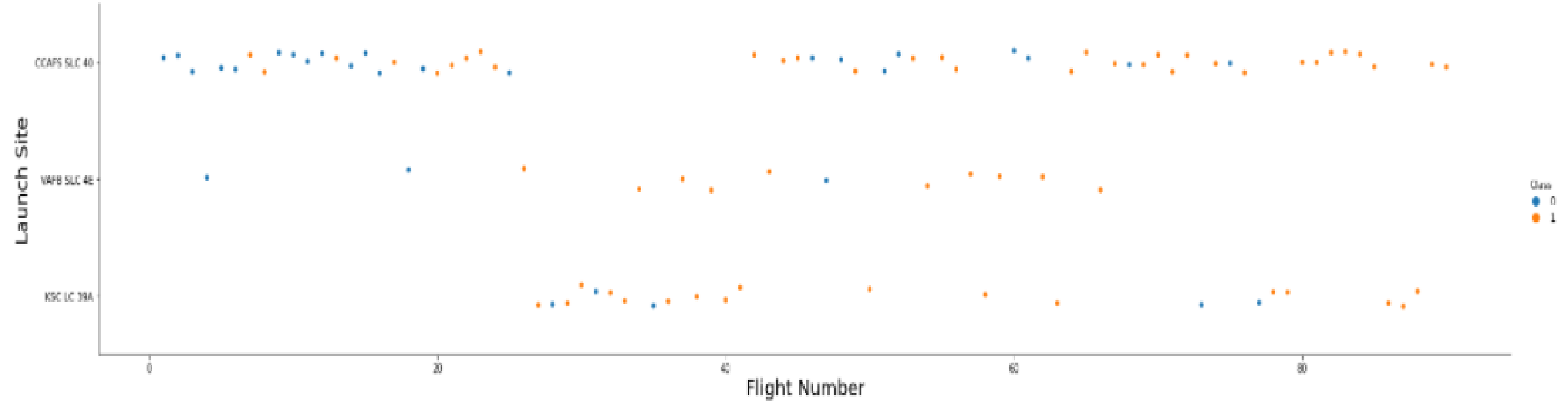
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

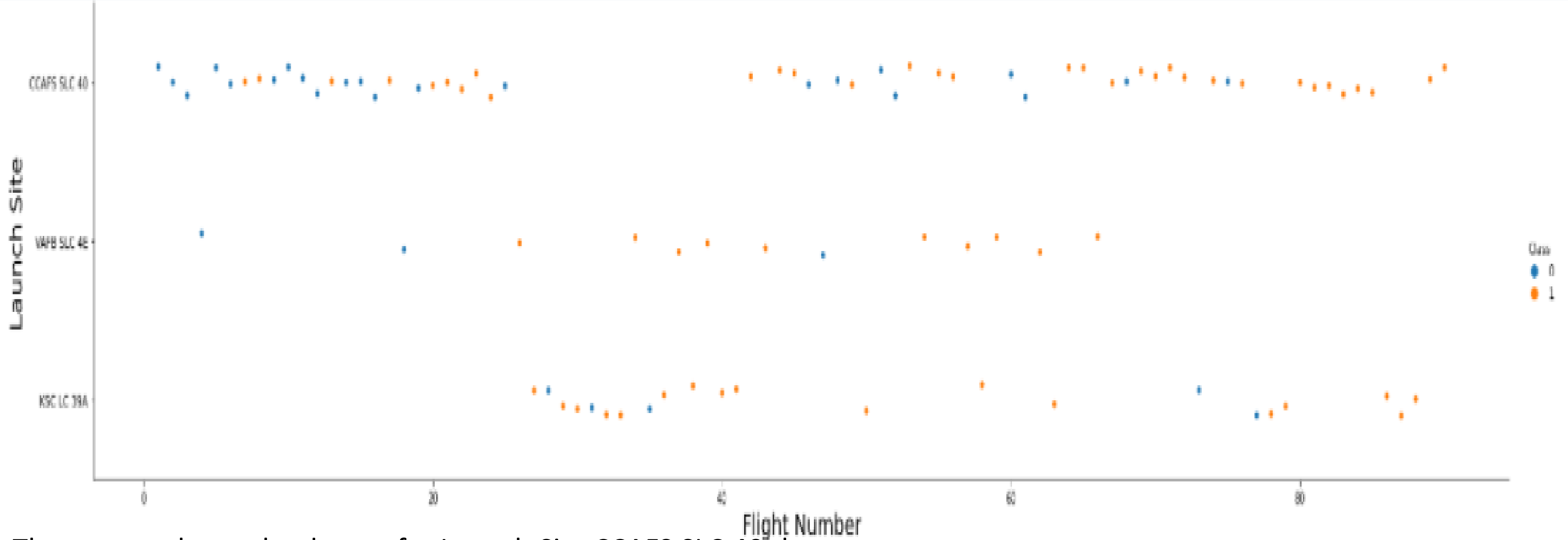
Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations



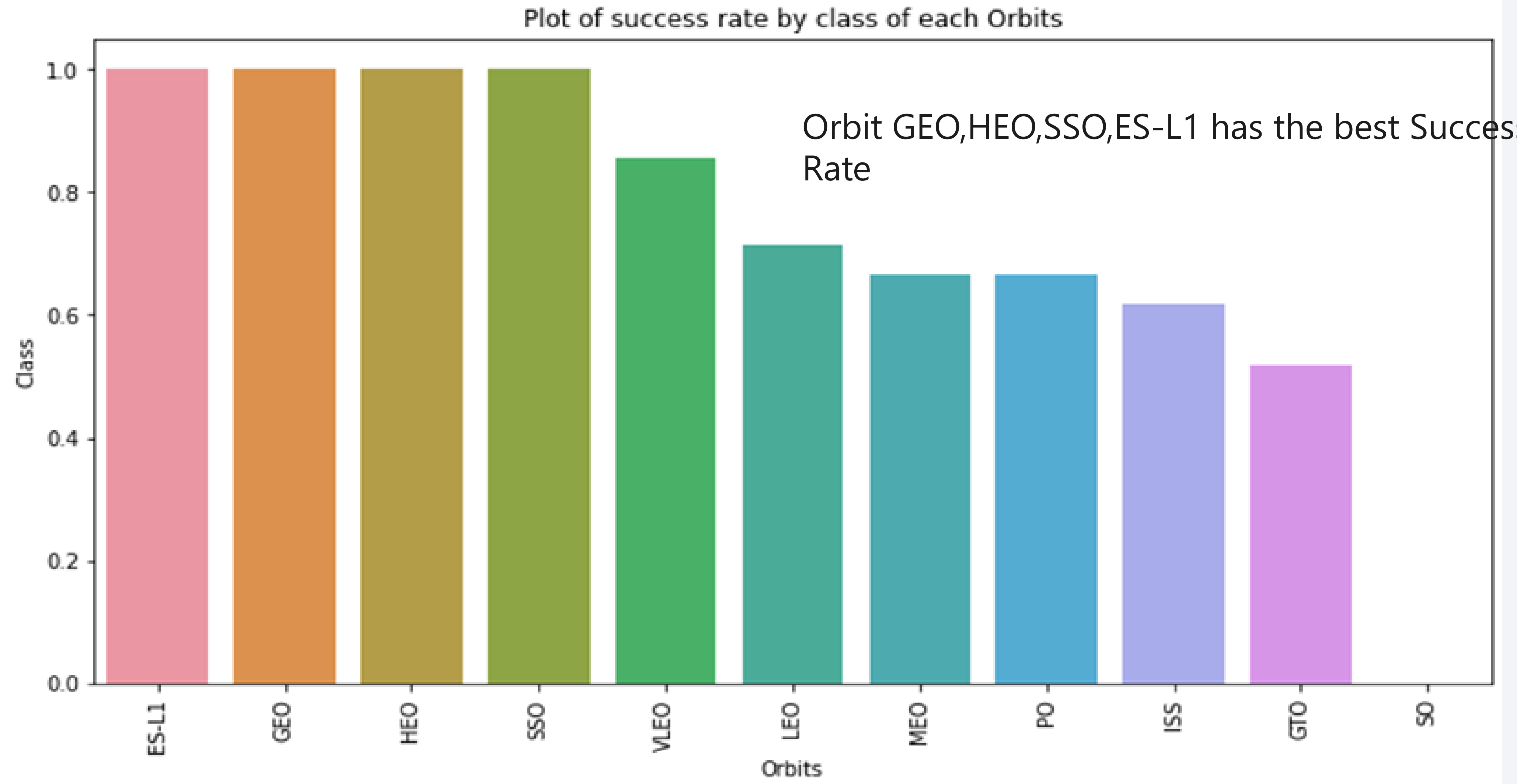
The more amount of flights at a launch site the greater the success rate at a launch site.

Payload vs. Launch Site



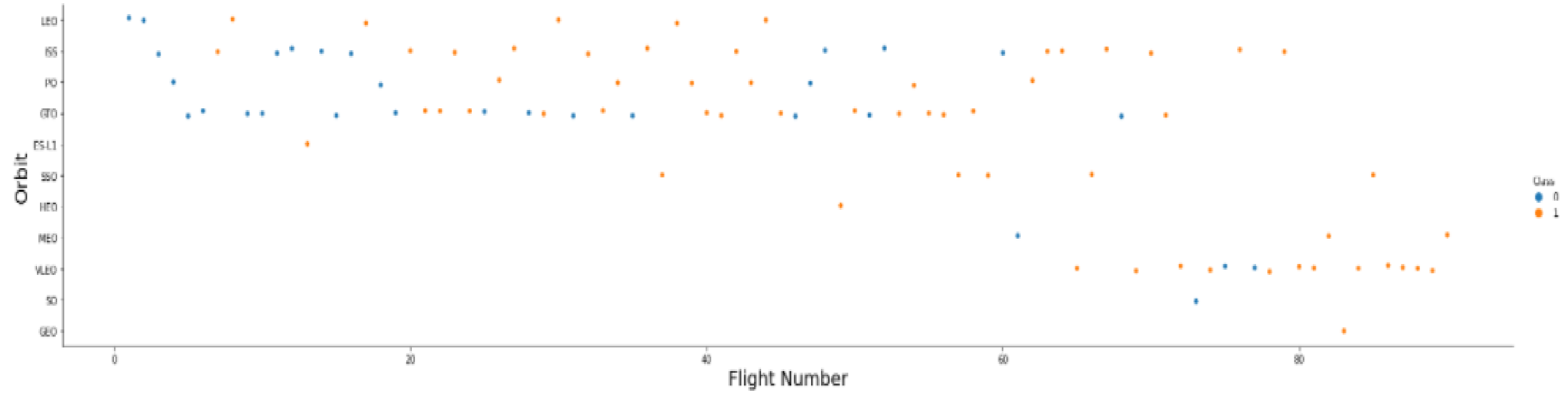
The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket.
There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependant on Pay Load Mass for a success launch.

Success Rate vs. Orbit Type



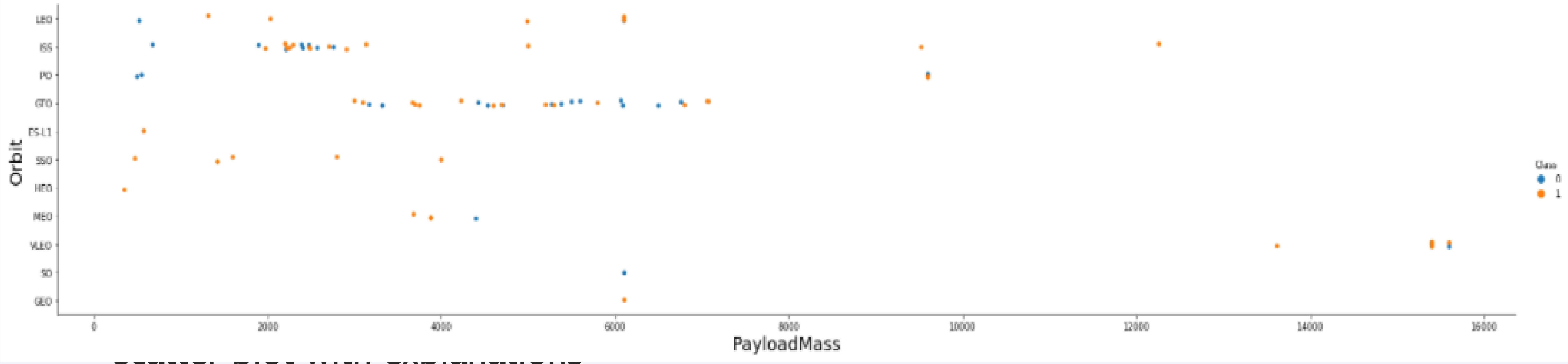
Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Flight Number vs. Orbit Type



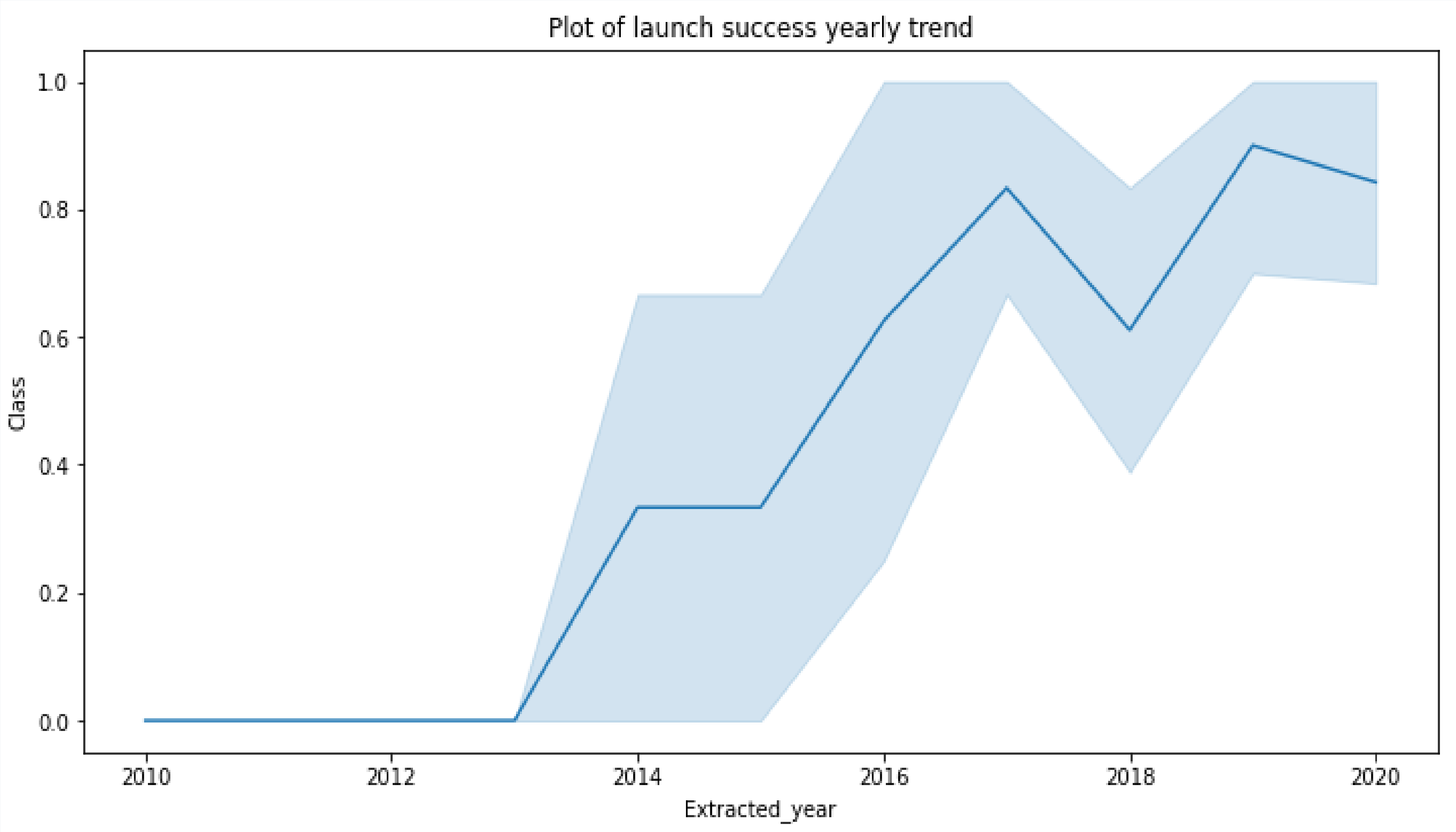
You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



Heavy
payloads have a negative influence
on GTO orbits and positive on GTO
and Polar LEO (ISS) orbits.

Launch Success Yearly Trend



the success rate since 2013 kept increasing till 2020

All Launch Site Names

Task 1

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

on here

In [13]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[13]:

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

QUERY EXPLAINATION
Using the word DISTINCT in the query means that it will only show Unique values in the LaunchSite column from SpaceX

Launch Site Names Begin with 'CCA'

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

In [14]:

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

Out[14]:

	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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	2012-08-10	0:35:00	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

Using the word TOP 5 in the query means that it will only show 5 records from SpaceX 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.

Total Payload Mass

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

In [15]:
```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[15]:

	Total_PayloadMass
0	45596

Using the function SUM summates the total in the column PayloadMassKG
The WHERE clause filters the dataset to only perform calculations on Customer NASA (CRS)

Average Payload Mass by F9 v1.1

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

In [16]:
```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[16]:

	Avg_PayloadMass
0	2928.4

Using the function AVG works out the average in the column PayloadMassKG
The WHERE clause filters the dataset to only perform calculations on Booster_version F9 v1.1

First Successful Ground Landing Date

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

Hint:Use min function

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

	FirstSuccessfull_landing_date
0	2015-12-22

Using the function MIN works out the minimum date in the column Date

The WHERE clause filters the dataset to only perform calculations on LandingOutcome Success (drone ship)

Successful Drone Ship Landing with Payload between 4000 and 6000

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

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

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

Selecting only Booster_Version
The WHERE clause filters the dataset to LandingOutcome = Success (drone ship)
The AND clause specifies additional filter conditions PayloadMassKG > 4000 AND PayloadMassKG < 6000

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
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

used wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.

Boosters Carried Maximum Payload

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

```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

	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

Using the word DISTINCT in the query means that it will only show Unique values in the BoosterVersion column from tblSpaceX

GROUP BY puts the list in order set to a certain condition.

DESC means its arranging the dataset into descending order

2015 Launch Records

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

```
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)
```

	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)

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

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

```
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)
```

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

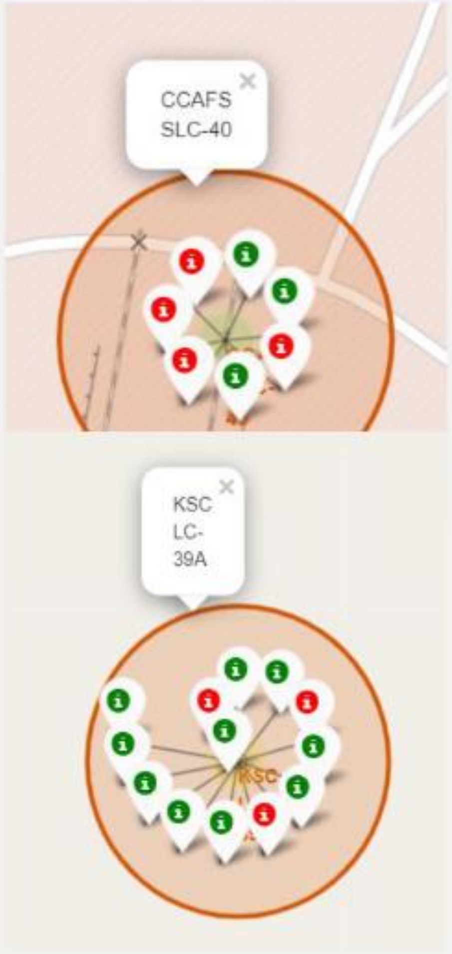
We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20. We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

Section 3

Launch Sites Proximities Analysis

All launch sites global map markers

Markers showing launch sites with color labels



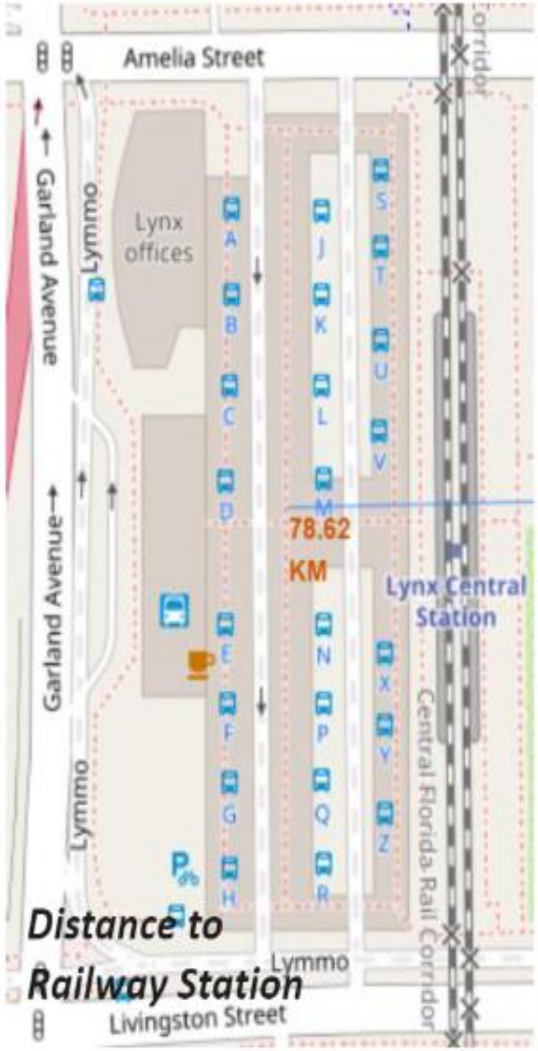
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

Launch Site distance to landmarks

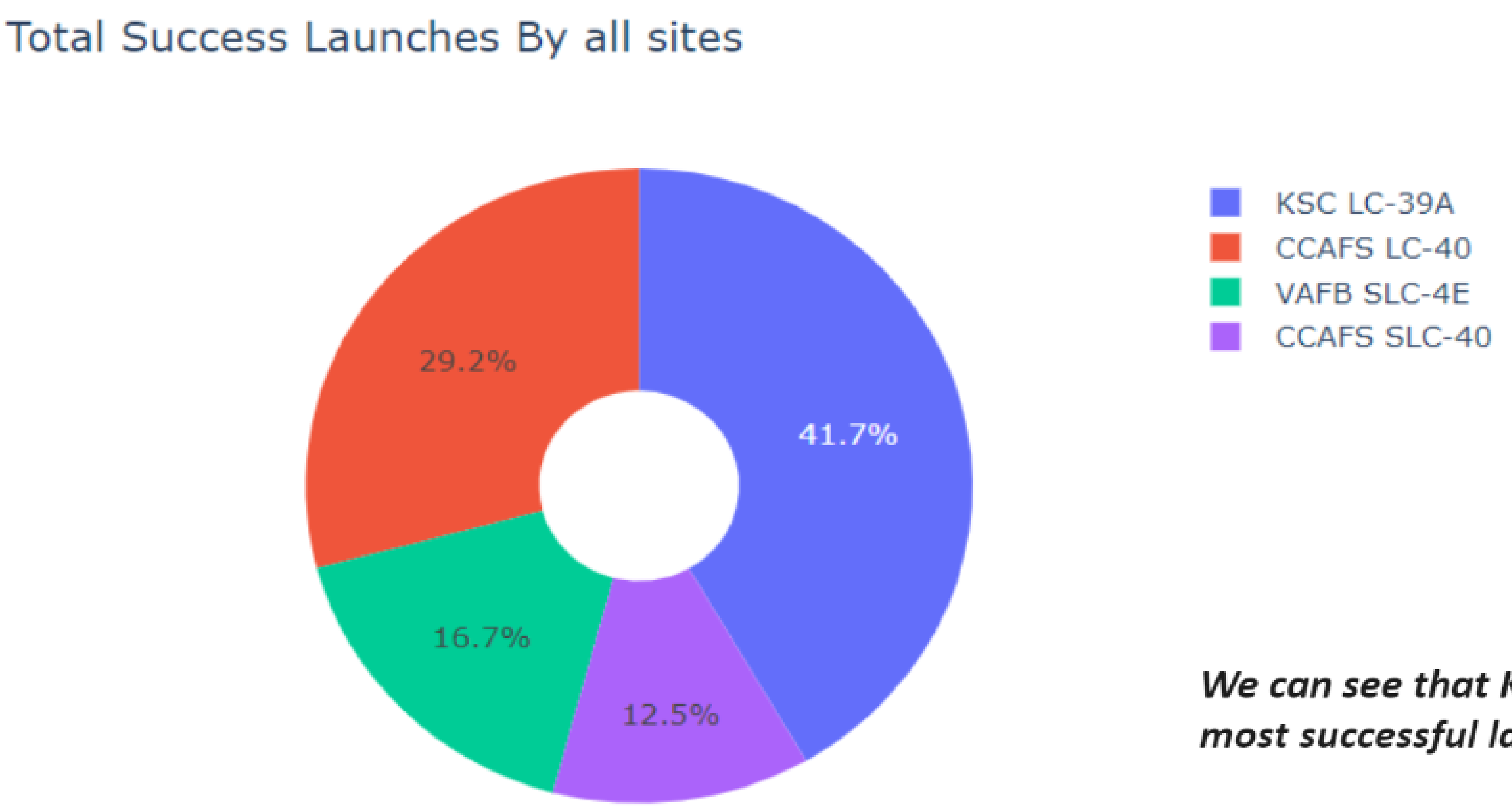


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

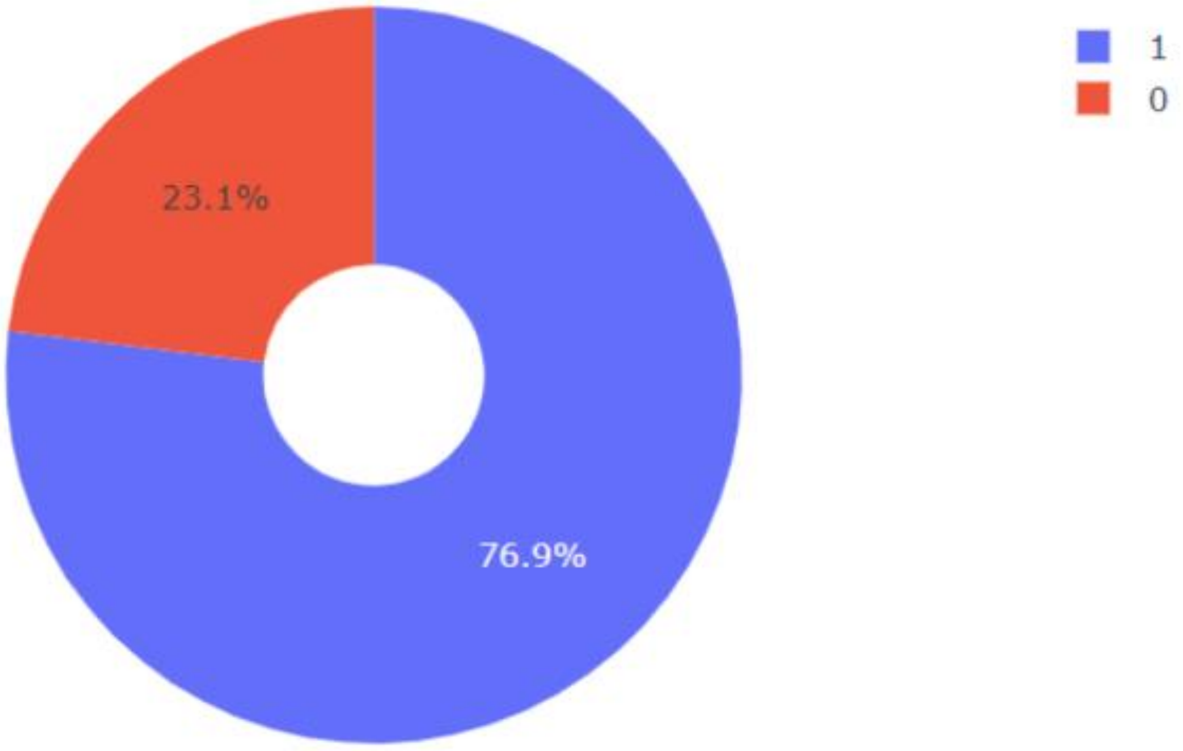
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



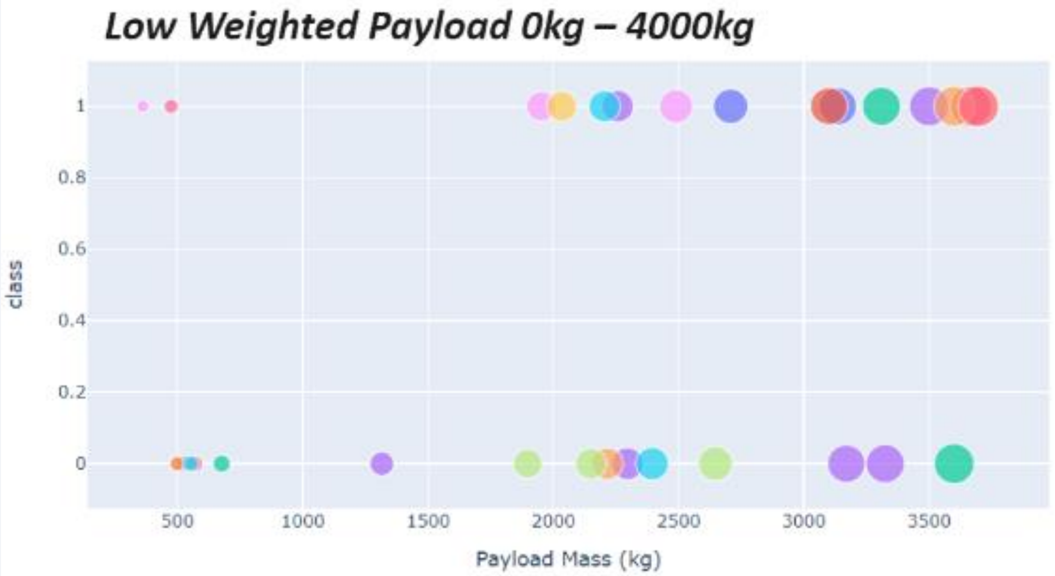
We can see that KSC LC-39A had the most successful launches from all the sites

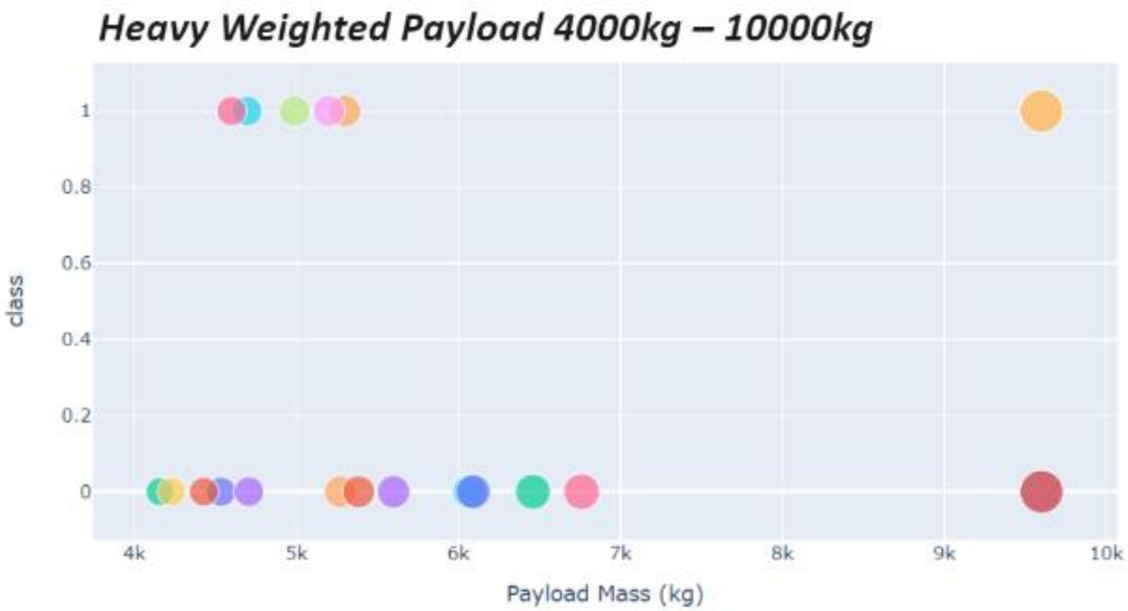
Pie chart showing the Launch site with the highest launch success ratio



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





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

Section 5

Predictive Analysis (Classification)

Classification Accuracy

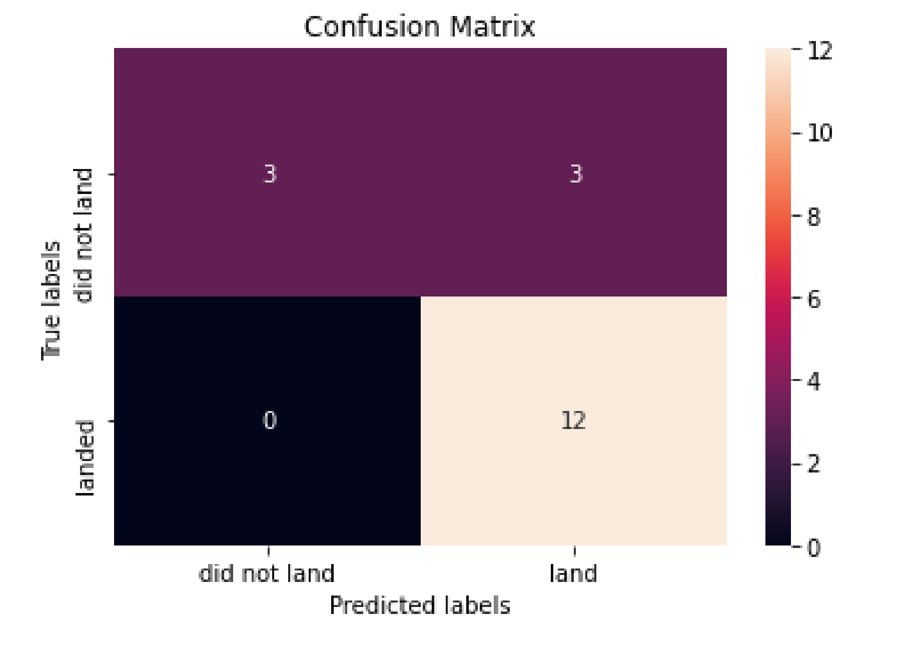
```
In [25]: models = {'KNeighbors':knn_cv.best_score_,
                       'DecisionTree':tree_cv.best_score_,
                       'LogisticRegression':logreg_cv.best_score_,
                       'SupportVector': svm_cv.best_score_}

         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 0.8785714285714284
         Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 2,
         'splitter': 'random'}
```

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

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.

Conclusions

- The Tree Classifier Algorithm is the best for Machine Learning for this dataset
- Low weighted payloads perform better than the heavier payloads
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches
- We can see that KSC LC-39A had the most successful launches from all the sites Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

GitHub - chaiysue/ibm_data_science_capstone_spacex

Thank you!