IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kamalesh K B
19-06-2023

Outline

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

Executive Summary

- Summary of methodologies
    - Data collection through API & with Web Scraping
    - Data Wrangling
    - EDA with SQL
    - EDA with Visualization
    - Interactive Map with Folium
    - Dashboard with Plotly Dash
    - Predictive Analysis (Classification)

Summary of all results

- EDA Results
- Interactive Analysis
- Predictive Analysis

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
  - The project task is to predicting if the first stage of the SpaceX Falcon 9 rocket will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scraping from Wikipedia
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning and data cleaning of null values and irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, SVM, DT models have been built and evaluated for the best classifier

Data Collection

- The source of datasets are :
  - Using get request to the SpaceX Rest API
  - And using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - In addition Wikipedia of falcon 9 launch records with BeatifulSoup
  - And extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- Data collection with SpaceX REST API calls.

https://github.com/Kamalesh-02/SpaceX-Falcon9-Landing-Prediction/blob/main/1.%20The%20Data%20Collection%20With%20API.ipynb

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 [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

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

In [30]:
```
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.

https://github.com/Kamalesh-02/SpaceX-Falcon9-Landing-Prediction/blob/main/2.%20The%20Data%20Collection%20With%20Web%20Scraping.ipynb

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

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

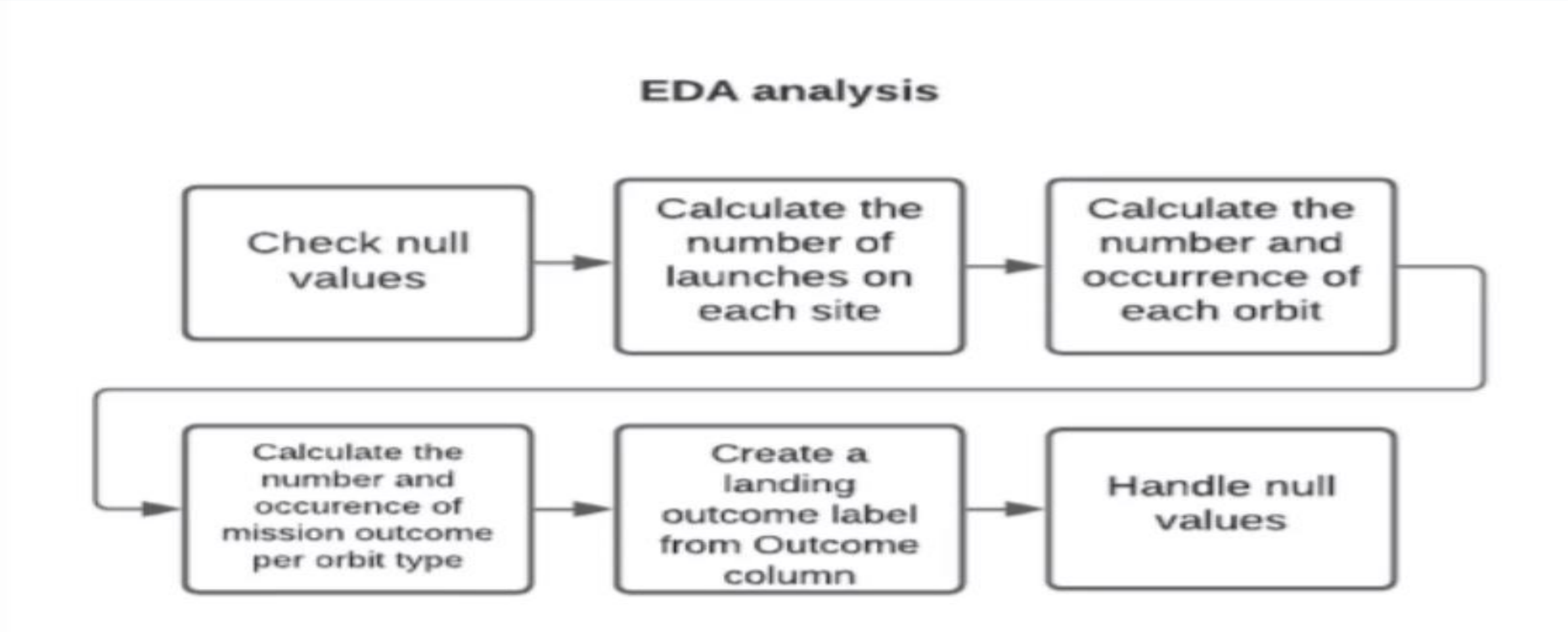
In [7]:
```
# Use soup.title attribute
soup.title
```

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
```

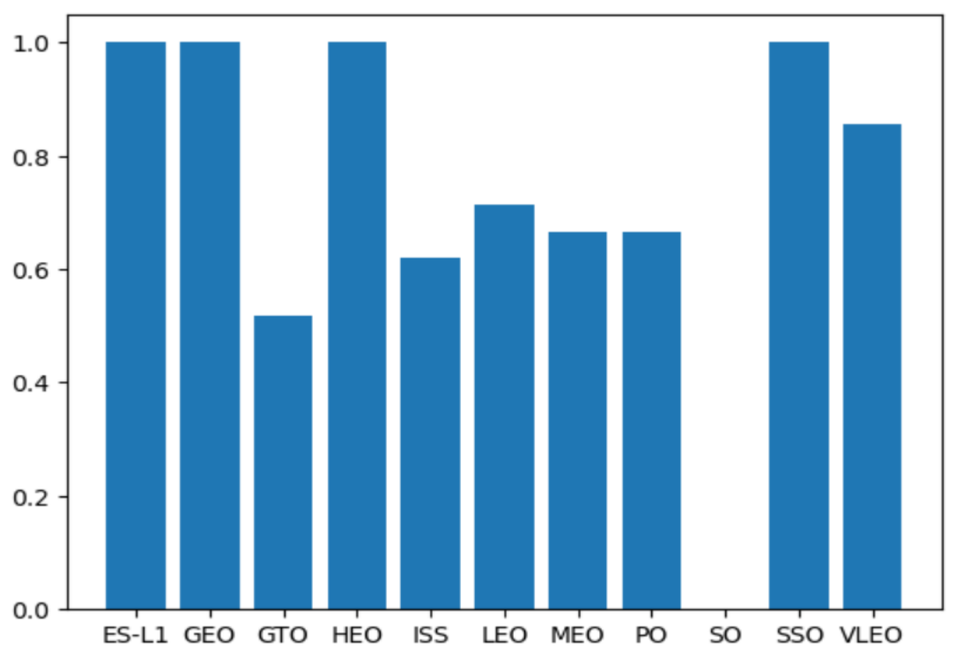
Data Wrangling

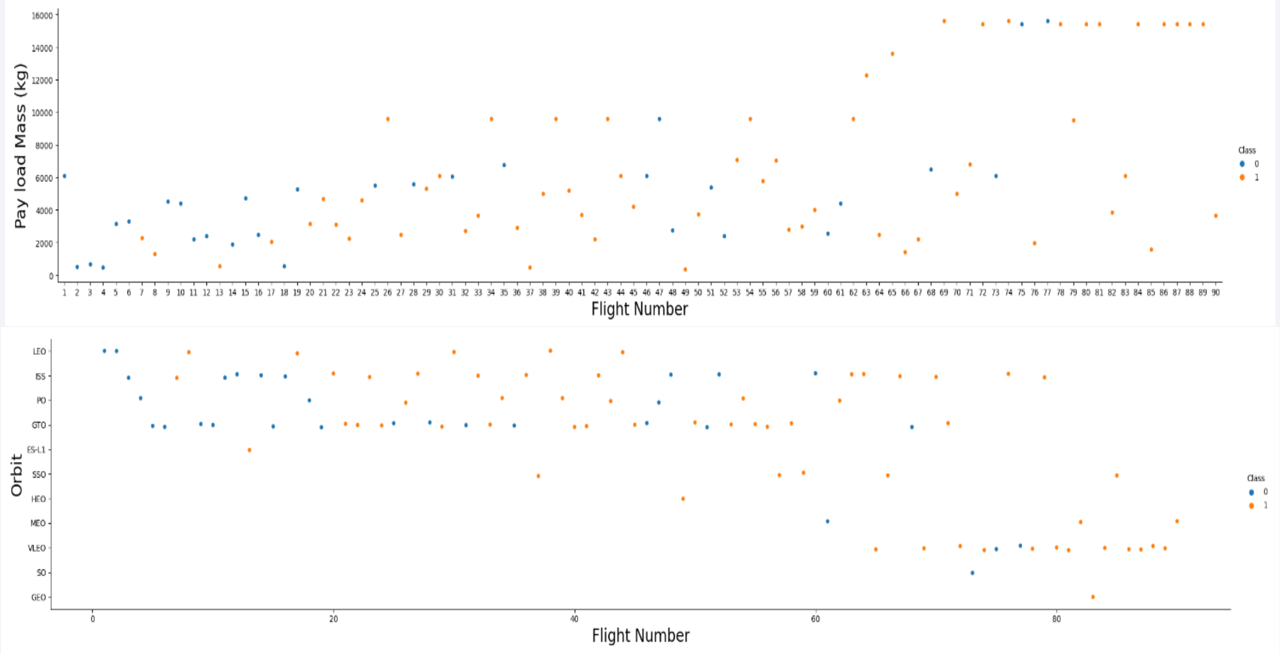


https://github.com/Kamalesh-02/SpaceX-Falcon9-Landing-Prediction/blob/main/3.%20Data%20Wrangling.ipynb

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.





https://github.com/Kamalesh-02/SpaceX-Falcon9-Landing-Prediction/blob/main/5.%20EDA%20with%20Visualization.ipynb

EDA with SQL

SQL queries performed includes:

- ***Display the names of the unique launch sites in the space mission***
- ***Display 5 records where launch sites begin with the string 'CCA'***
- ***Display the total payload mass carried by boosters launched by NASA (CRS)***
- ***Display average payload mass carried by booster version F9 v1.1***
- ***List the date when the first succesful landing outcome in ground pad was acheived.***
- ***List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000***
- ***List the total number of successful and failure mission outcomes***
- ***List the names of the booster_versions which have carried the maximum payload mass. Use a subquery***
- ***List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.***
- ***Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.***

https://github.com/Kamalesh-02/SpaceX-Falcon9-Landing-Prediction/blob/main/4.%20EDA%20with%20SQL.ipynb

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://github.com/Kamalesh-02/SpaceX-Falcon9-Landing-Prediction/blob/main/6.%20Interactive%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a 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/Kamalesh-02/SpaceX-Falcon9-Landing-Prediction/blob/main/7.%20Interactive%20Dashboard%20with%20Plotly%20Dash.ipynb

Predictive Analysis (Classification)

- Loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built SVM, KNN & Logistic Regresion 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.
- The SVM, KNN and Logistic Regression model achieved the highest accuracy at 83.3%, while the SVM performs the best in terms of Area under the curve at 0.958.

https://github.com/Kamalesh-02/SpaceX-Falcon9-Landing-Prediction/blob/main/8.%20The%20Machine%20Learning%20Prediction.ipynb

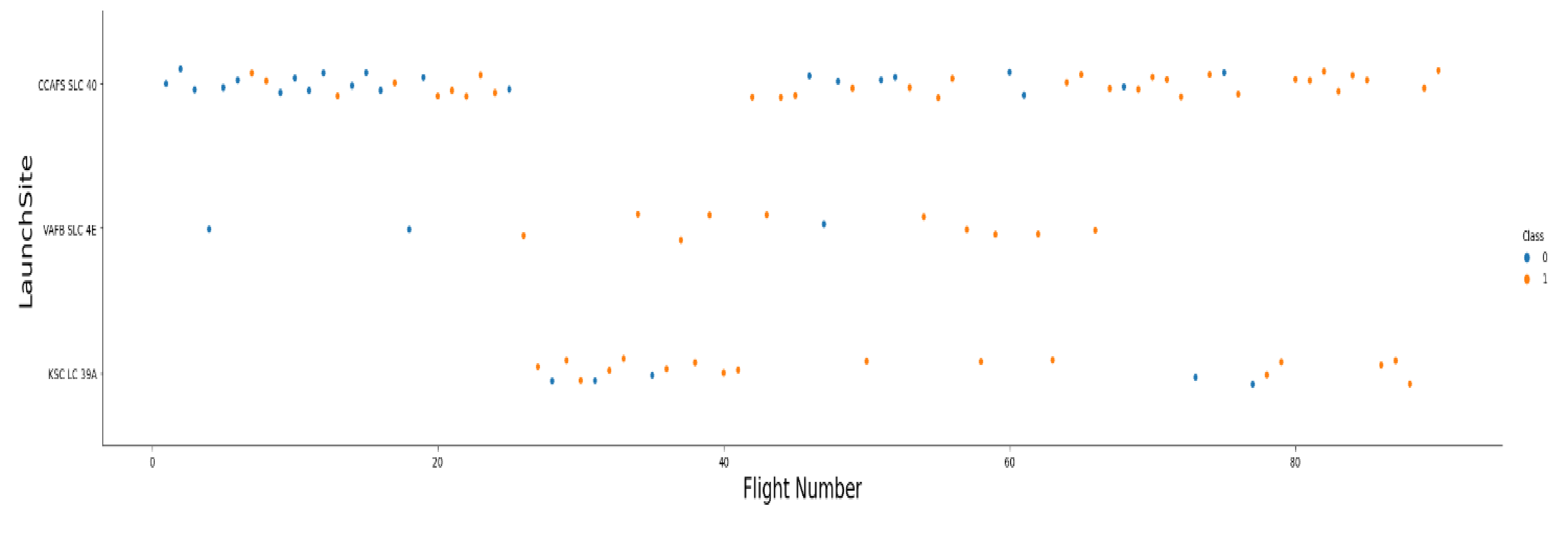
Results

- The SVM, KNN and Logistic Regression models are the best in terms of prediction accuracy for this dataset.
- Low weighted payloads perform better than the heavier payloads
- The success rates for SpaceX launches is directly propositional time in years they will eventually perfect the launches.
- KSC LC 39A had the most successful launches from all the sites.
- Orbit GEO, HEO, SSO, ES L1 has the best Success Rate.

Section 2

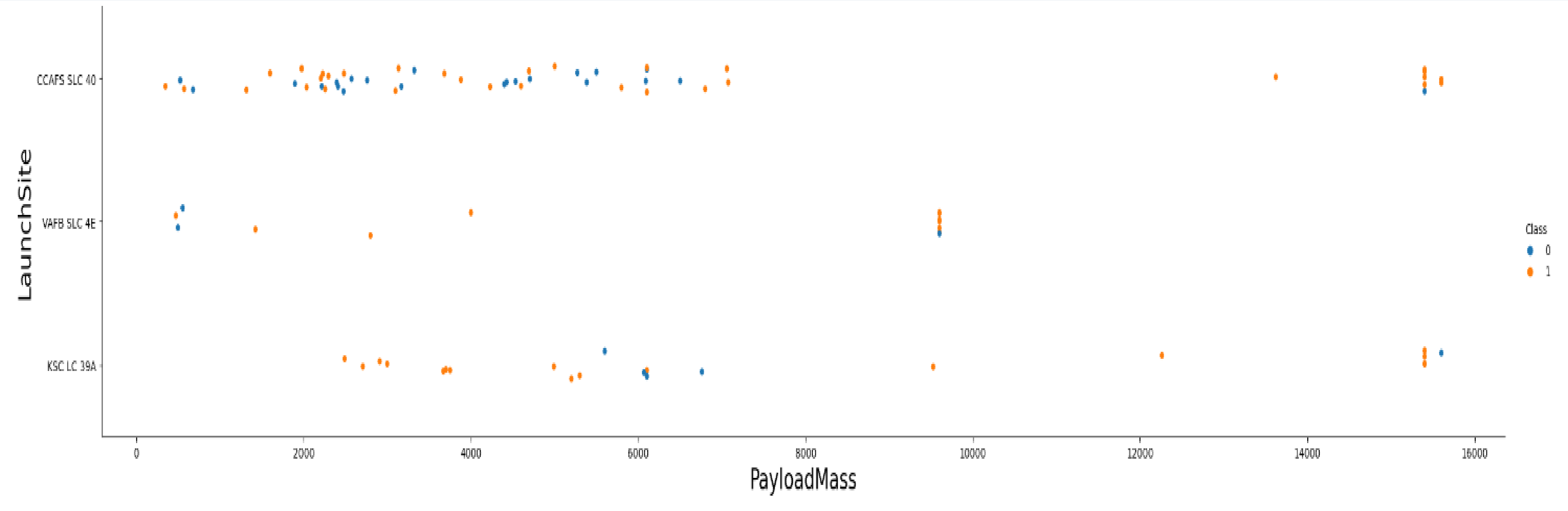
Insights drawn from EDA

Flight Number vs. Launch Site



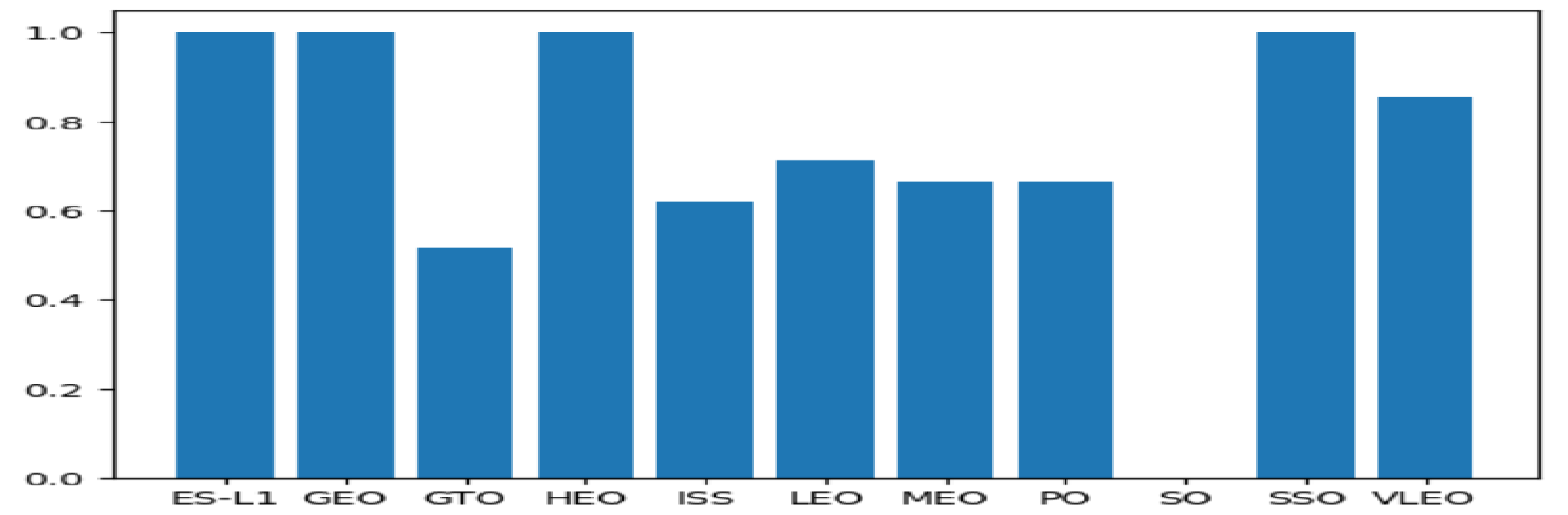
- Launch from the site of CCAFS SLC 40 are significantly higher than launches from other sites.

Payload vs. Launch Site



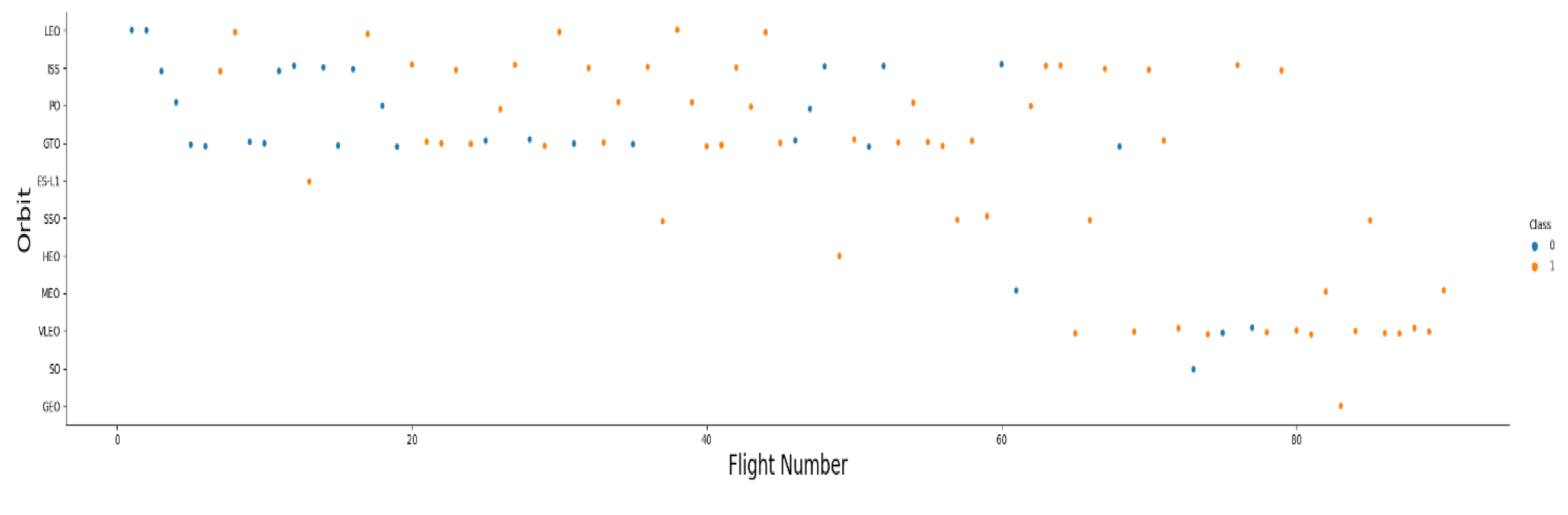
- The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.

Success Rate vs. Orbit Type



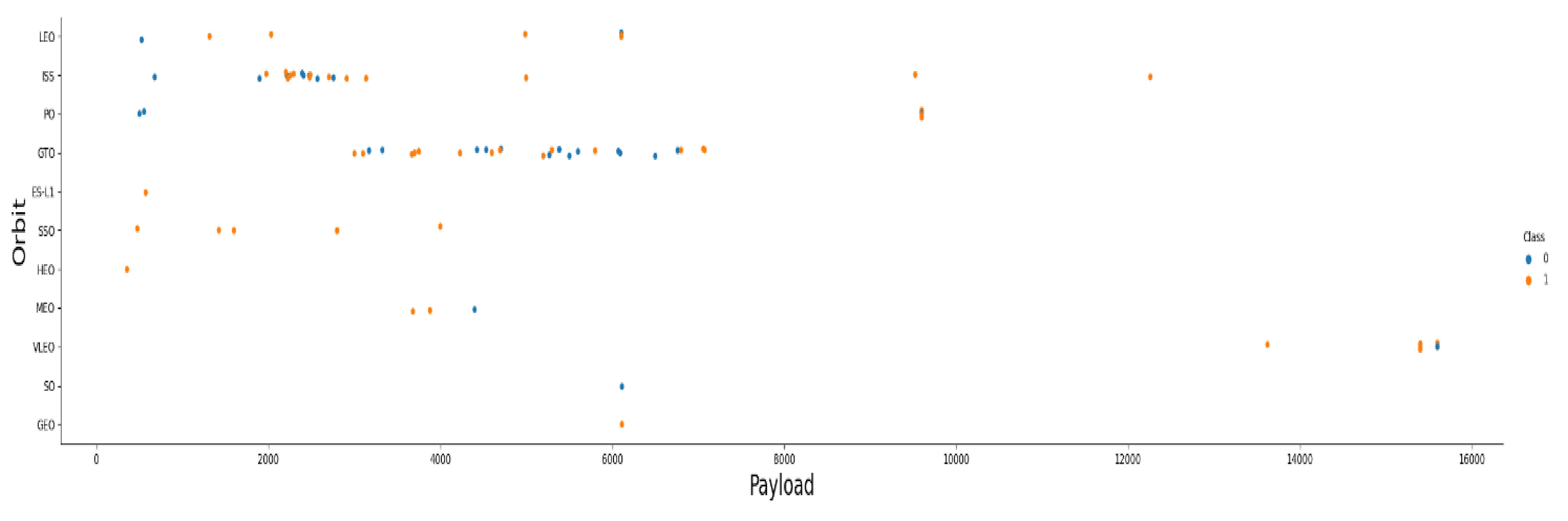
- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

Flight Number vs. Orbit Type



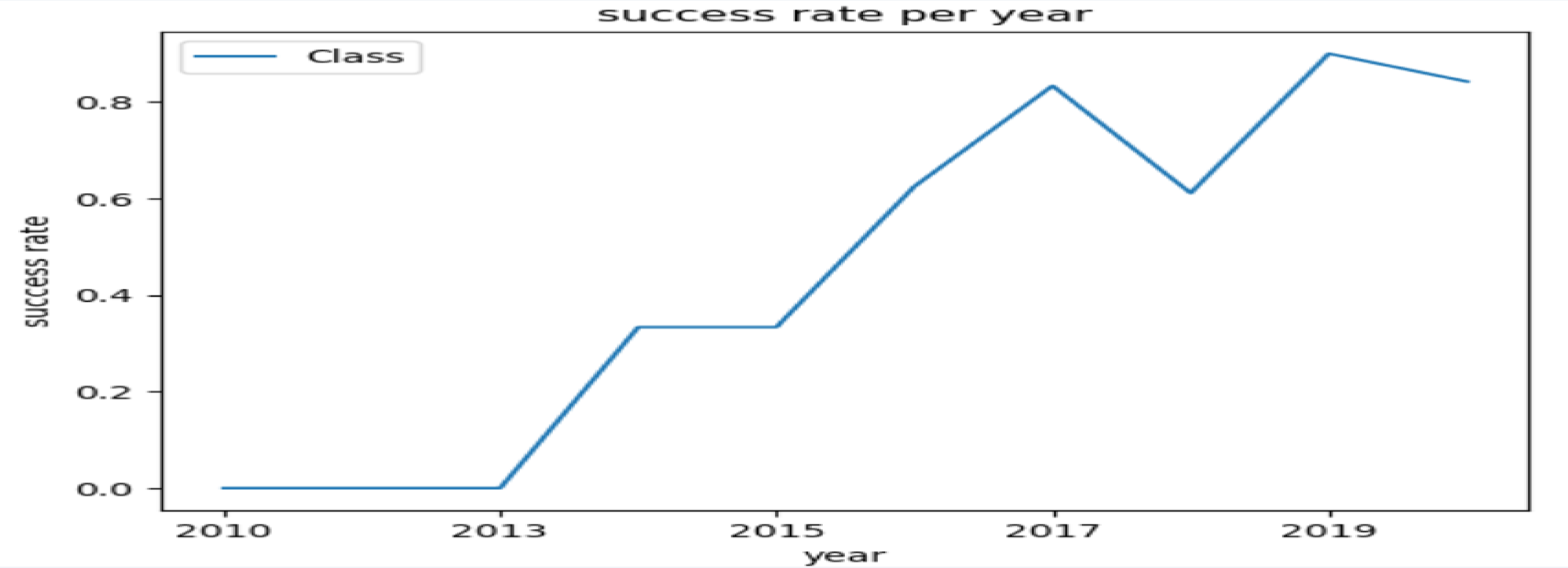
- A trend can be observed of shifting to VLEO launches in recent years
- We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.

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. Due to potentially advance in technology.

All Launch Site Names

```
%sql select distinct Launch_Site from SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5;
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) as Total_Payload_By_NASA_CRS from SPACEXTBL where Customer == "NASA (CRS)";
```

```
 * sqlite:///my_data1.db
Done.
```

Total_Payload_By_NASA_CRS
45596.0

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) as Average_Payload_By_F9_v1 from SPACEXTBL where Booster_Version == "F9 v1.1";
```

```
 * sqlite:///my_data1.db
Done.
```

Average_Payload_By_F9_v1
2928.4

First Successful Ground Landing Date

```
%sql SELECT MIN(Date) AS FirstSuccessfull_landing_date FROM SpaceXTBL WHERE Landing_Outcome LIKE 'Success (ground pad)';
```

```
 * sqlite:///my_data1.db
Done.
```

FirstSuccessfull_landing_date
01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 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

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

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

	failureoutcome
0	1

Boosters Carried Maximum Payload

%sql select Booster_Version,PAYLOAD_MASS__KG_ from SPACEXTBL where PAYLOAD_MASS__KG_ == (select Max(PAYLOAD_MASS__KG_) from SPACEXTBL);

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600.0
F9 B5 B1049.4	15600.0
F9 B5 B1051.3	15600.0
F9 B5 B1056.4	15600.0
F9 B5 B1048.5	15600.0
F9 B5 B1051.4	15600.0
F9 B5 B1049.5	15600.0
F9 B5 B1060.2	15600.0
F9 B5 B1058.3	15600.0
F9 B5 B1051.6	15600.0
F9 B5 B1060.3	15600.0
F9 B5 B1049.7	15600.0

2015 Launch Records

```python
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)

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

```
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)
```

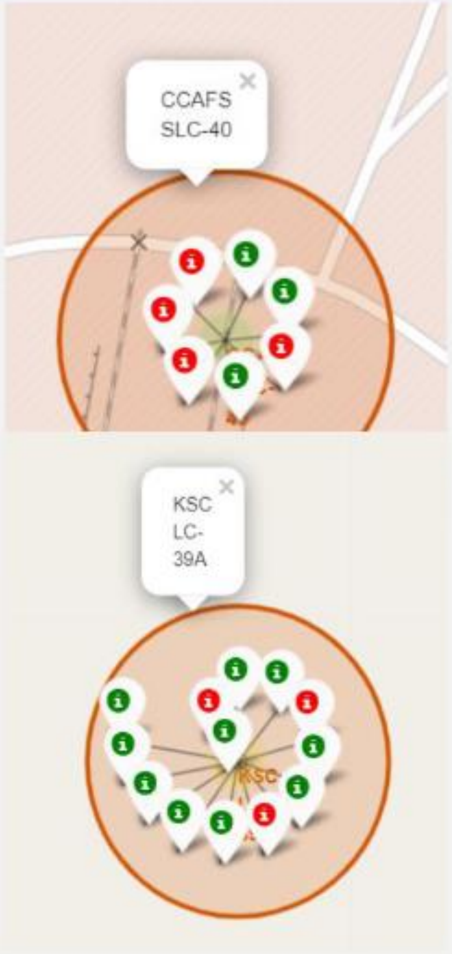
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

All launch sites global map markers

Markers showing launch sites with color labels



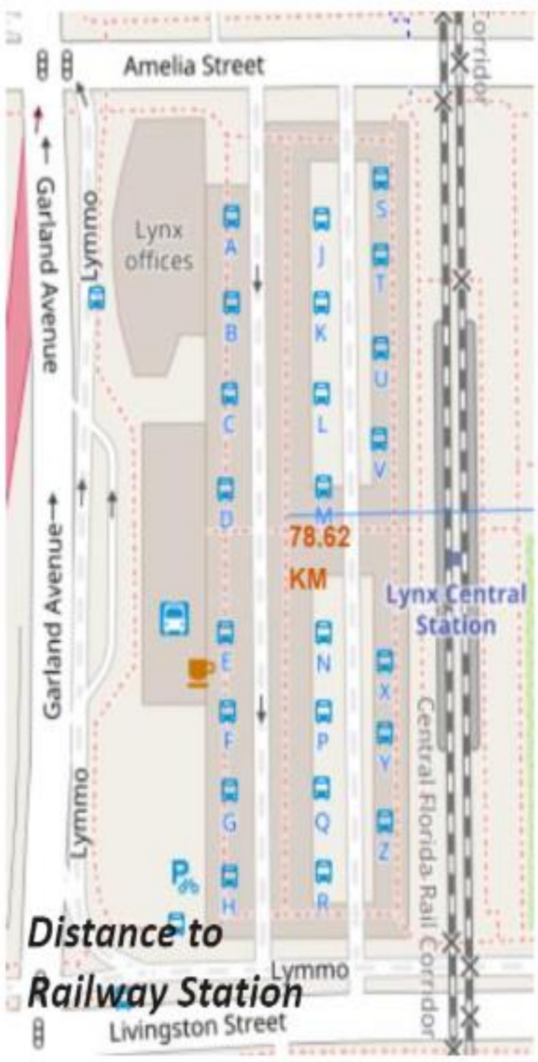
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

Launch Site distance to landmarks

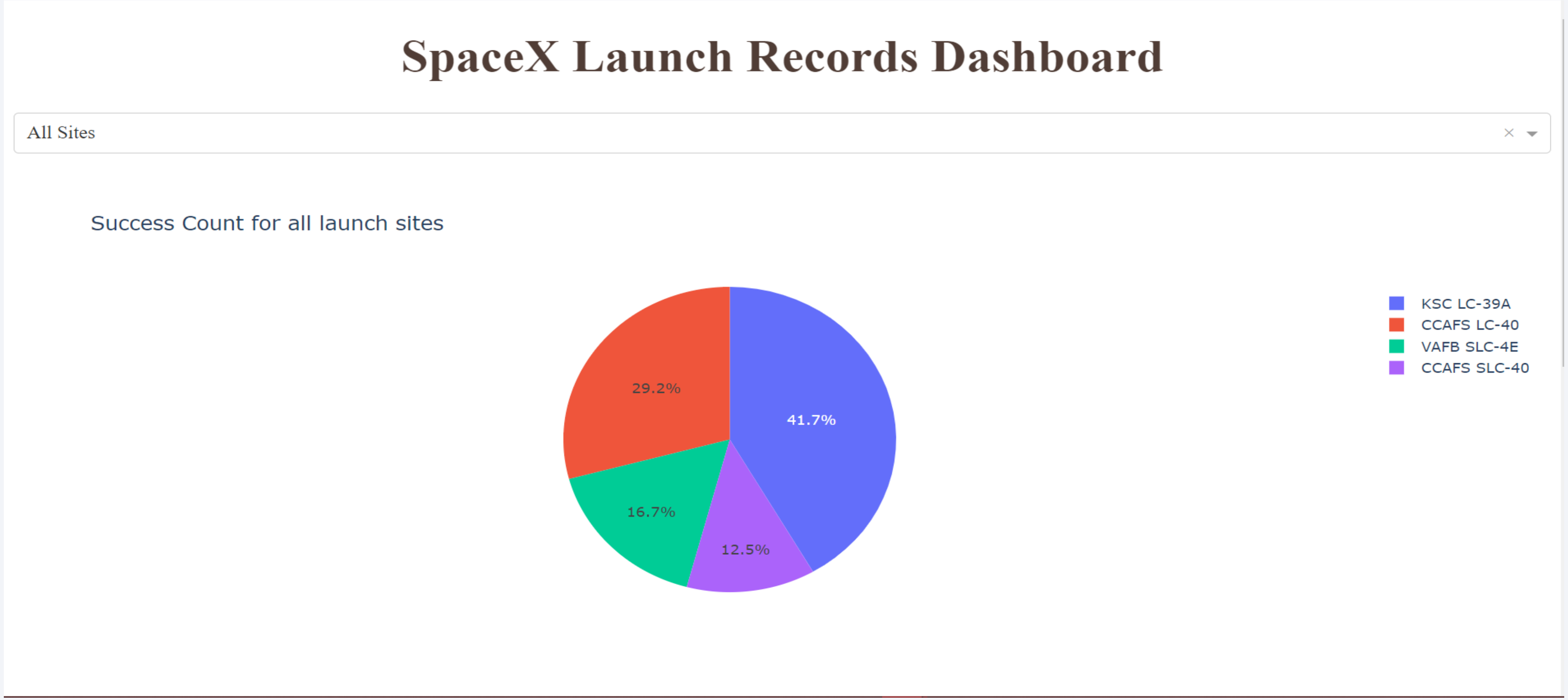


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

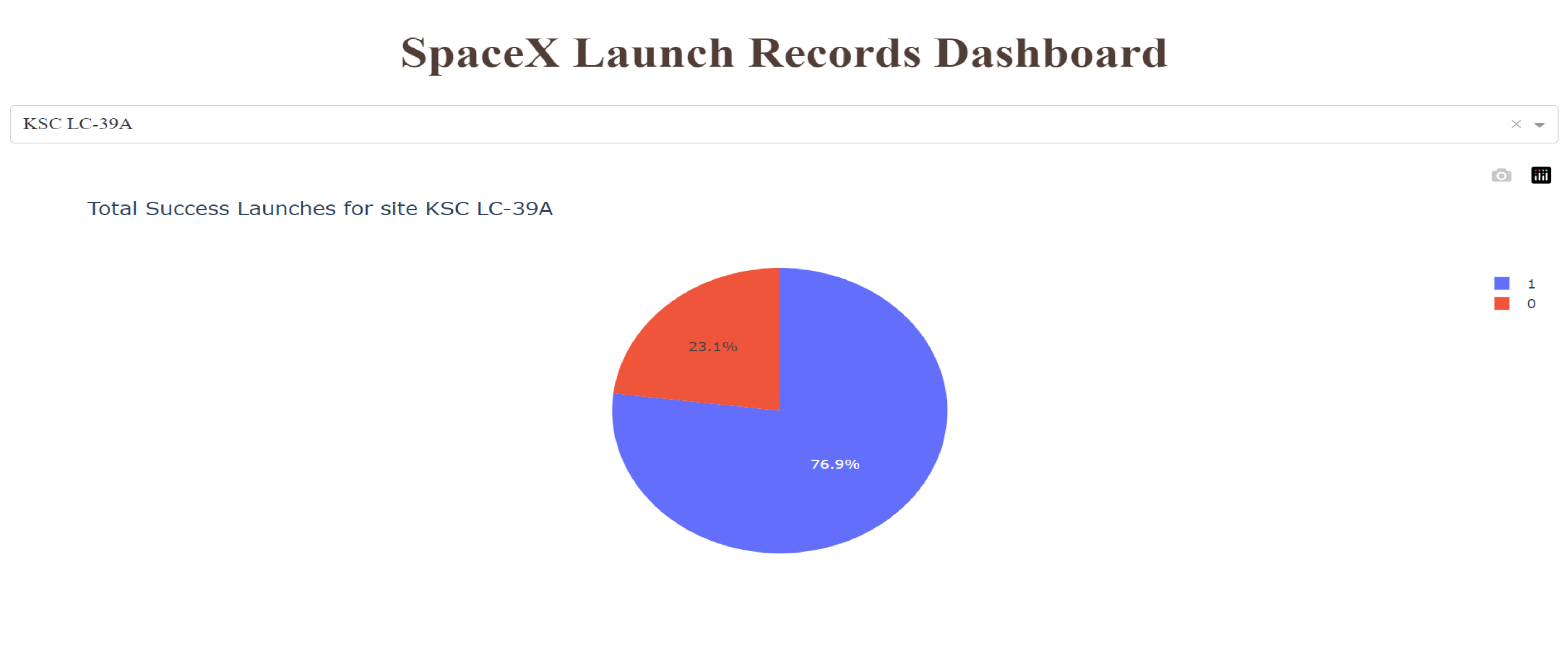
Section 4

Build a Dashboard with Plotly Dash

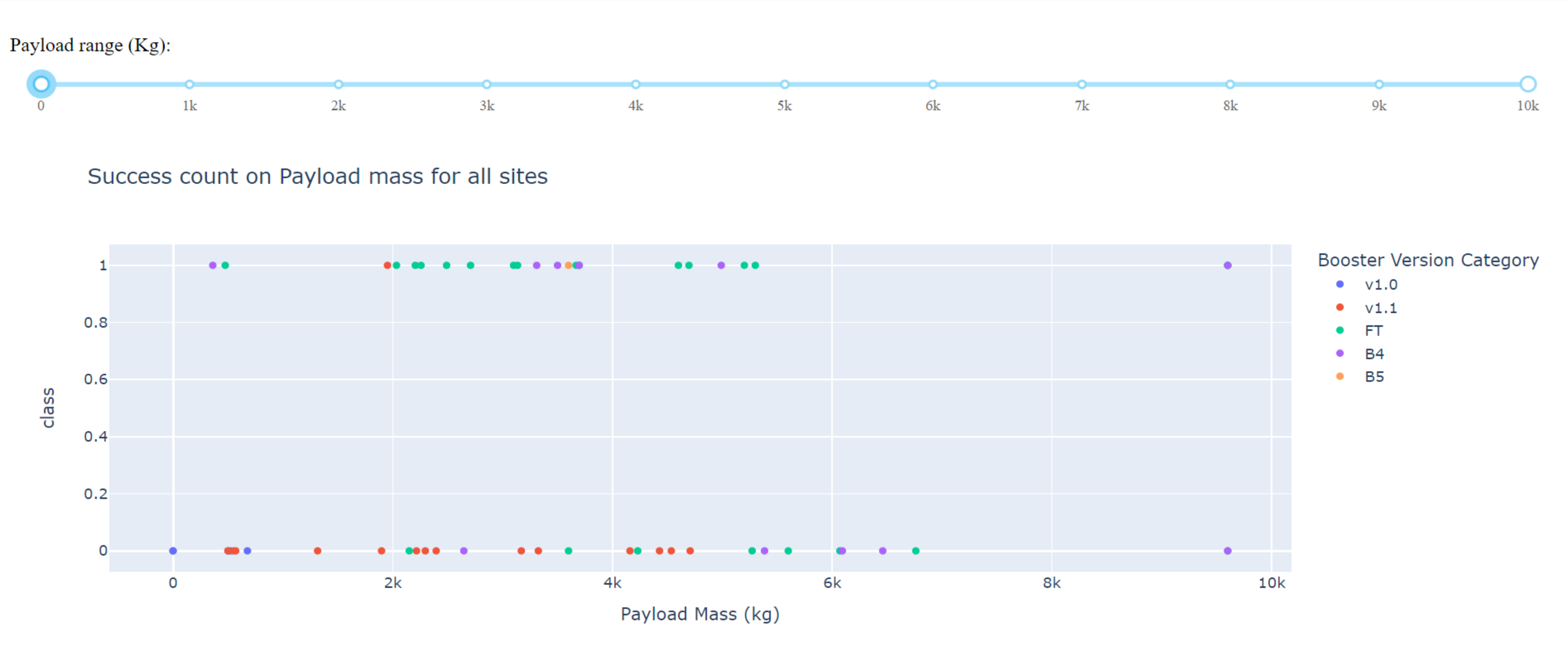
Pie chart to show the total successful launches count for all sites.



Pie chart show the launch site with highest launch success ratio



Payload vs. Launch Outcome scatter plot for all sites



Section 5

Predictive Analysis (Classification)

Classification Accuracy

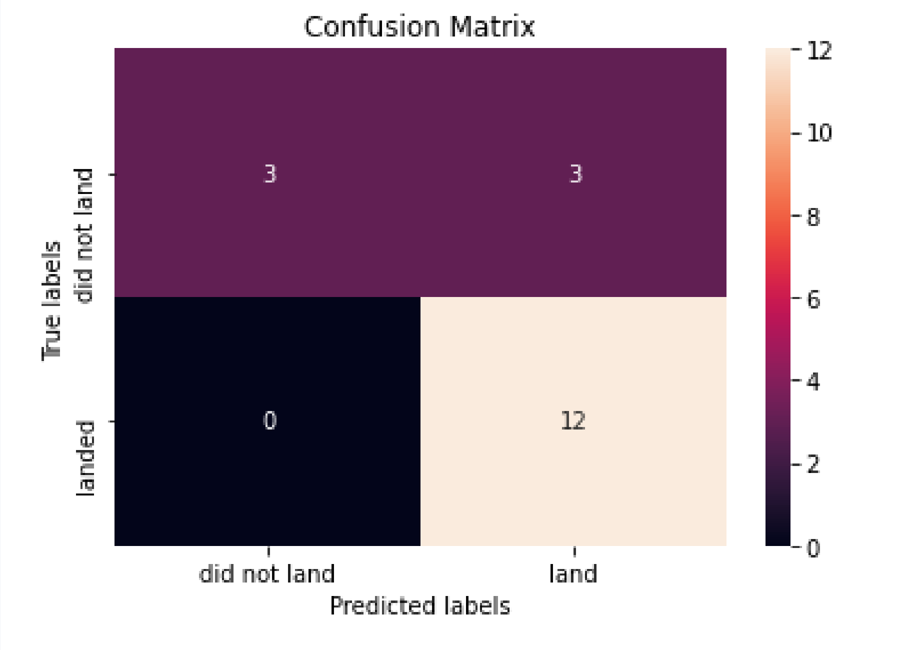
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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 0.8928571428571427
Best params is : {'criterion': 'entropy', 'max_depth': 14, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split':
10, 'splitter': 'best'}
```

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

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

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Low weighted payloads perform better than the Heavier payloads.
- KSC LC 39A had the most successful launches from all the sites.
- Orbit GEO, HEO, SSO, ES L1 has the best success rate.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!