Outline

- Executive Summary

- Introduction

- Methodology

- Results

- Conclusion

- Appendix

Executive Summary

- Summary of methodologies:
    1. Data Collection through API
    2. Data Collection with Web Scraping
    3. Data Wrangling
    4. Exploratory Data Analysis with SQL
    5. Exploratory Data Analysis with Data Visualization
    6. Interactive Visual Analytics with Folium
    7. Machine Learning Prediction
- Summary of all results:
    1. Exploratory Data Analysis result
    2. Interactive analytics in screenshots
    3. 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 willland 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:

    - DatafromSpaceXwasobtainedfrom2sources:

    - SpaceXAPI(https://api.spacexdata.com/v4/rockets/)

    - WebScraping (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches)

- Perform data wrangling

    - Collected data was enriched by creating a landing outcome label based on outcome data after summarizing and analyzing features

Executive Summary

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

- 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

- Describe how data sets were collected.

- You need to present your data collection process use key phrases and flowcharts

    • Datasets were collected from SpaceX API (https://api.spacexdata.com/v4/rockets/) and from Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches), using web scraping technics.

Data Collection – SpaceX API

- SpaceX offers a public API from where data can be obtained and then used; •
This API was used according to the flowchart beside and then data is persisted.

• Source code: Data Collection

Data Collection - Scraping

- • Data from SpaceX launches can also be obtained from Wikipedia; • Data are downloaded from Wikipedia according to the flowchart and then persisted.

- Source code: Data Collection with Web Scraping

Data Wrangling

• Initially some Exploratory Data Analysis (EDA) was performed on the dataset.

• Then the summary launches per site, occurrences of each orbit and occurrences of mission outcome per orbit type were calculated.

 • Finally, the landing outcome label was created from Outcome column.
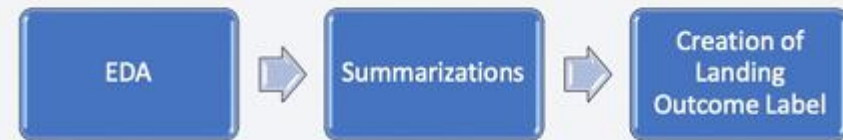
 • Source code: Data Wrangling

EDA with Data Visualization

- The following SQL queries were performed:

    - Names of the unique launch sites in the space mission;

    - Top 5 launch sites whose name begins with the string 'CCA';

    - Total pay load 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 in drone ship and have 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 out comes in droneship, their booster versions, and launch site names for in year 2015; and

    - Rank of the count of landing outcomes (such as Failure (droneship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20.

EDA with SQL

- To explore data, scatterplots and bar plots were used to visualize the relationship between pair of features:

• Payload Mass X Flight Number, Launch Site X Flight Number, Launch Site X Payload Mass, Orbit and Flight Number, Payload and Orbit

• Source code: EDA with SQL
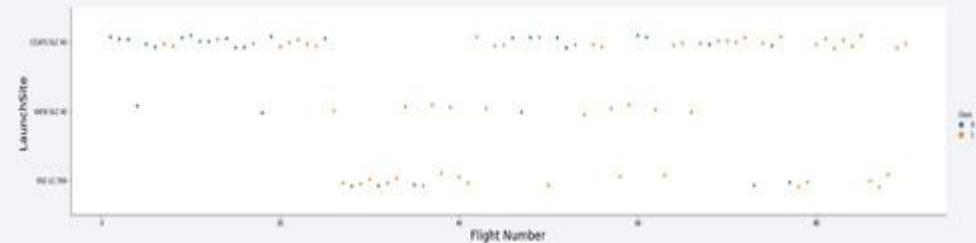
Build an Interactive Map with Folium

- 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 coordinate, like launches in a launch site; and

- Lines are used to indicate distances between two coordinates.

- Source code: Build an Interactive Map with Foliu

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 here.](#)

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 here.

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 v1.1 booster is 2,928 kg;

    - The first success landing outcome happened in 2015 fiver year 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 drone ships in 2015: F9 v1.1 B1012 and F9 v1.1 B1015;

    - The number of landing outcomes became as better as years passed.

Results

• Using interactive analytics was possible to identify that launch sites use to be in safety places, near sea, for example and have a good logistic infrastructure around.

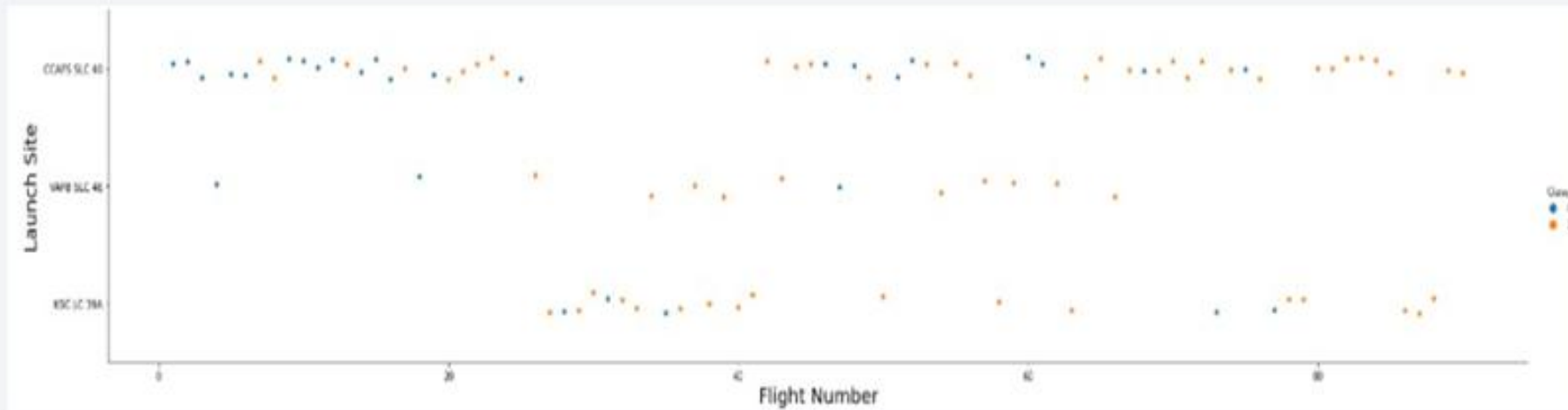• Most launches happens at east cost launch sites.

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.

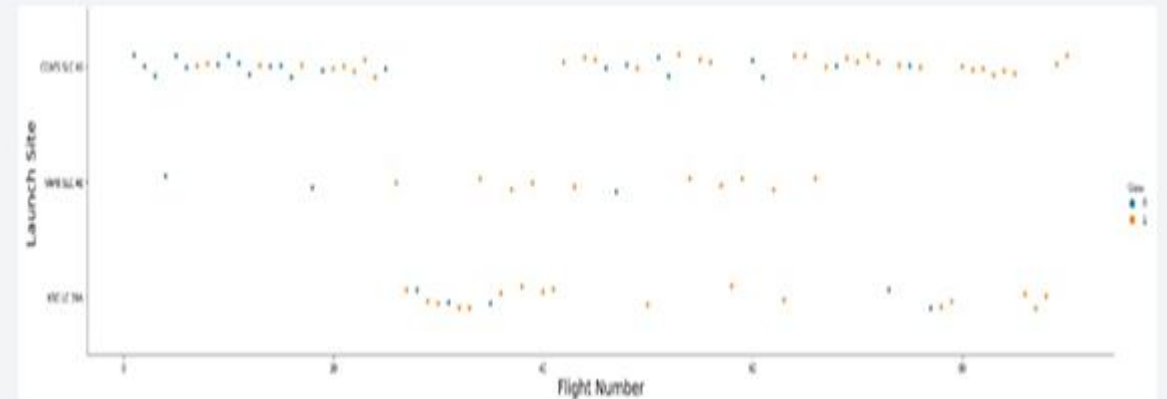# Payload vs. Launch Site

- Payload vs. Launch Site

Success Rate vs. Orbit Type

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

Plot of success rate by class of each Orbits
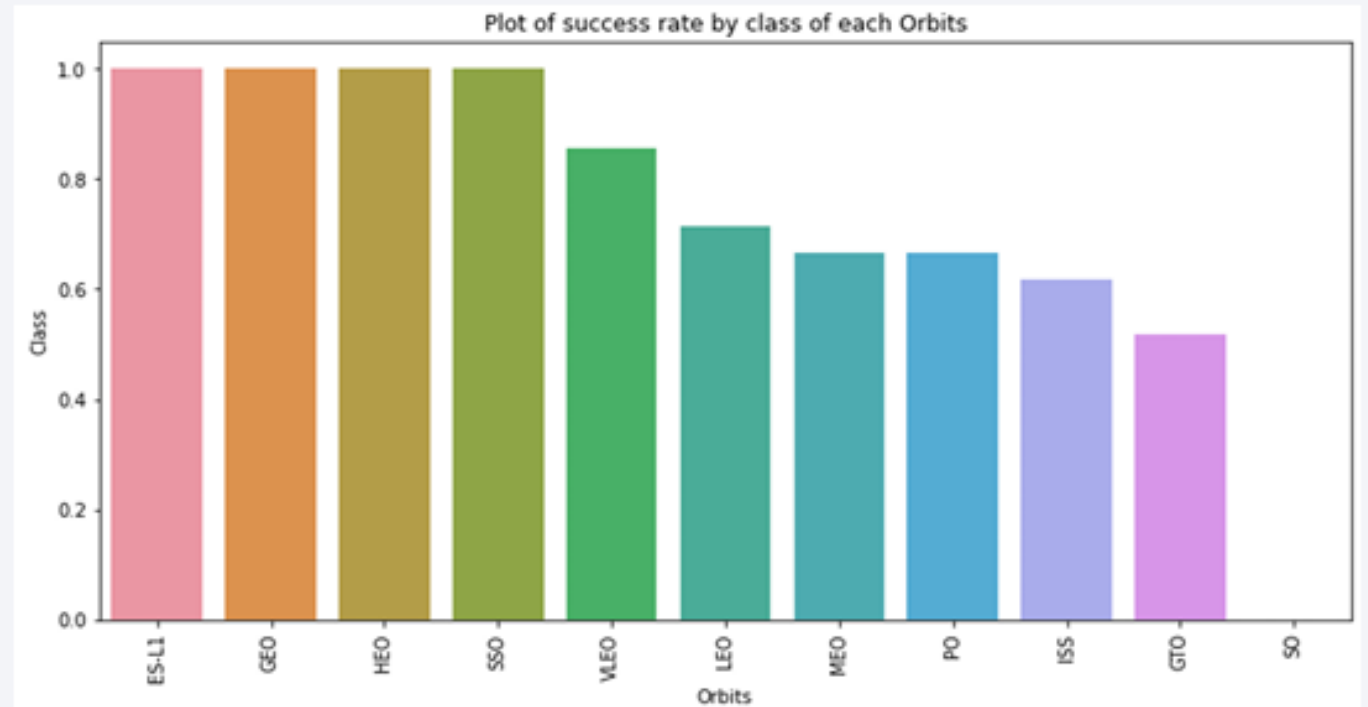
Flight Number vs. Orbit Type

• 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.
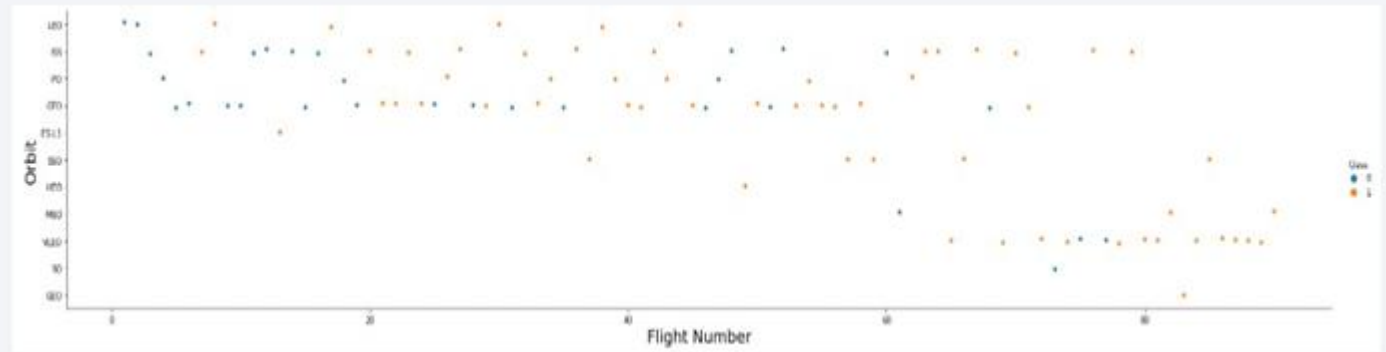
Payload vs. Orbit Type

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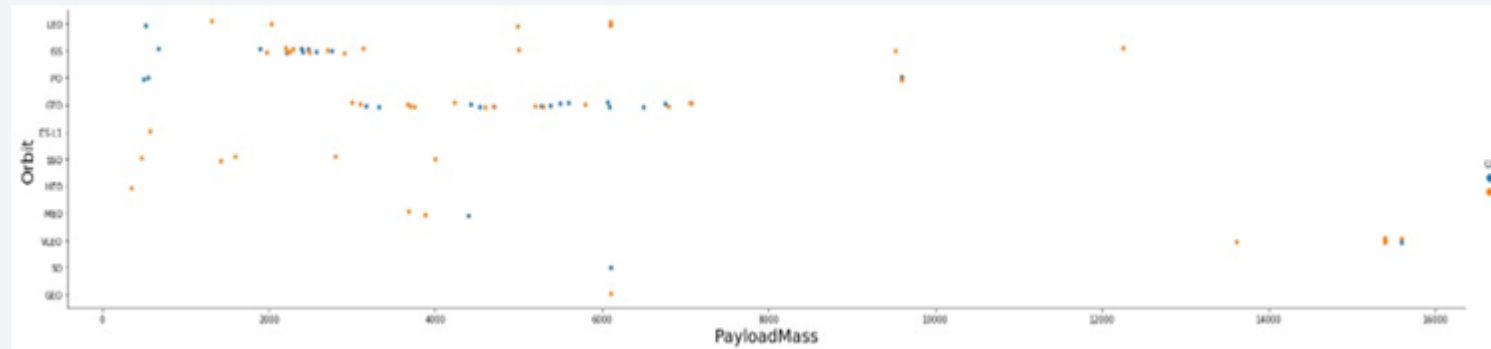
Launch Success Yearly Trend

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

Plot of launch success yearly trend
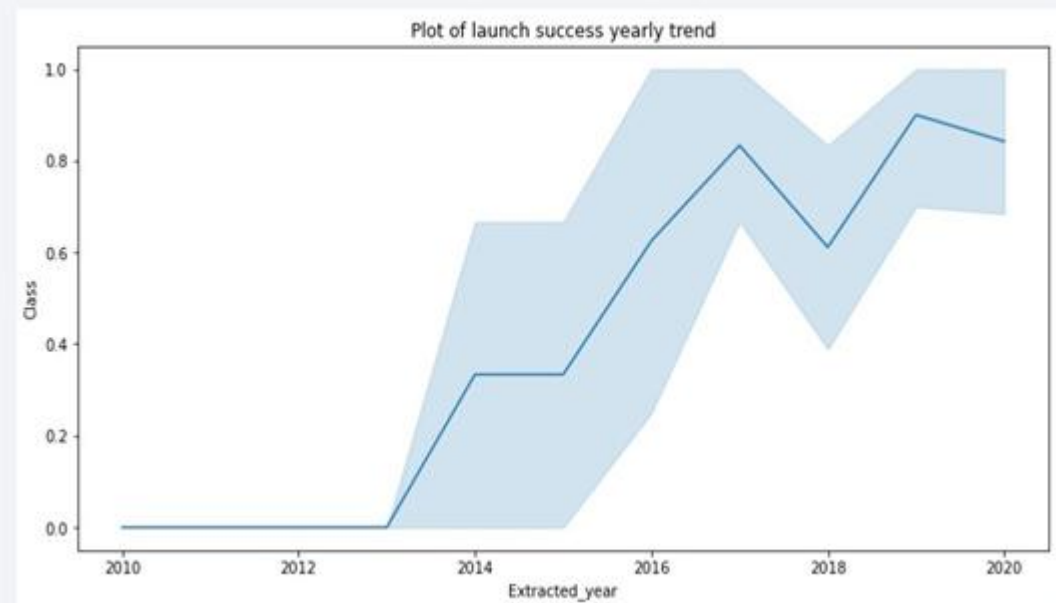
All Launch Site Names

- • 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'

- • 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'

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

	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 subqu

```
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)
```

	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)
```

	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

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

Markers showing launch sites with color labels

Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site
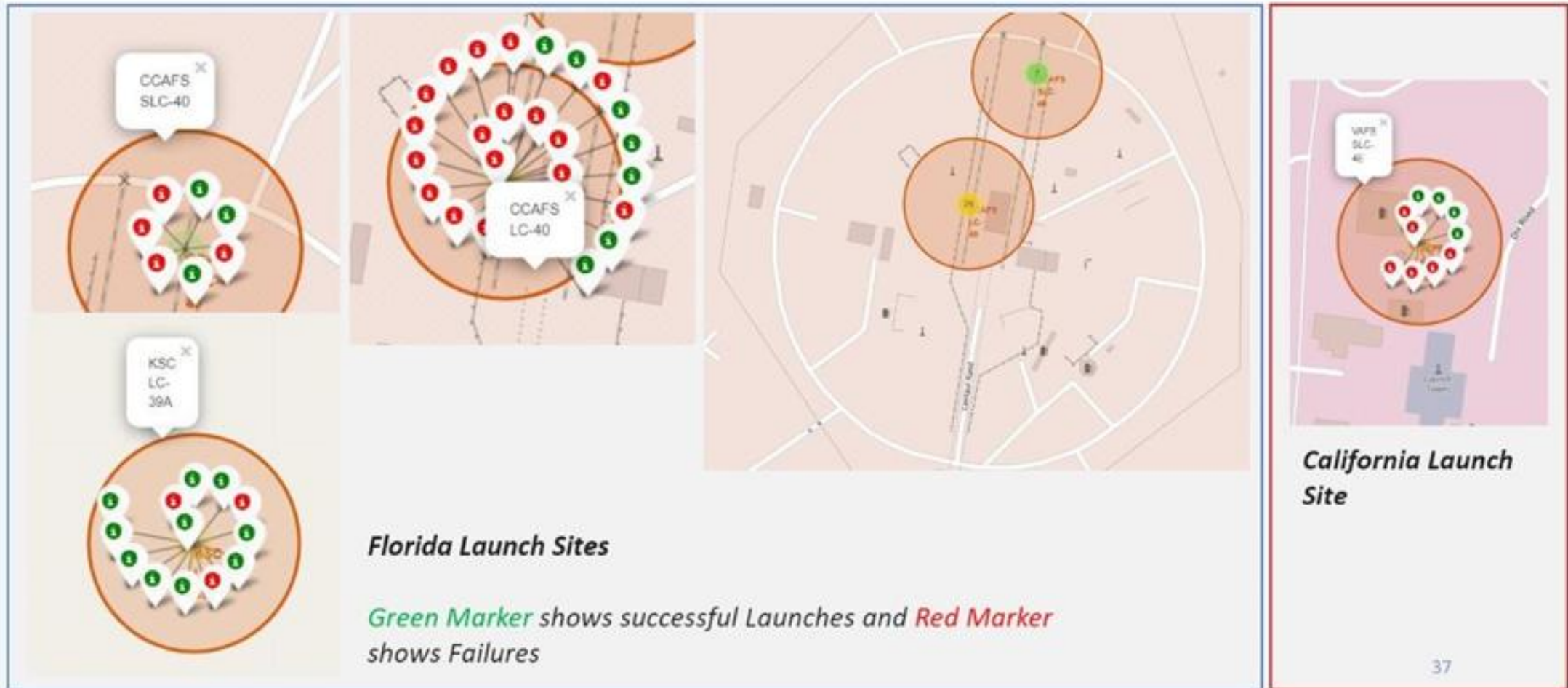
Launch Site distance to landmarks

Distance to Railway Station

Distance to closest Highway

Distance to Coastline

Distance to City

Distance to coast

•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
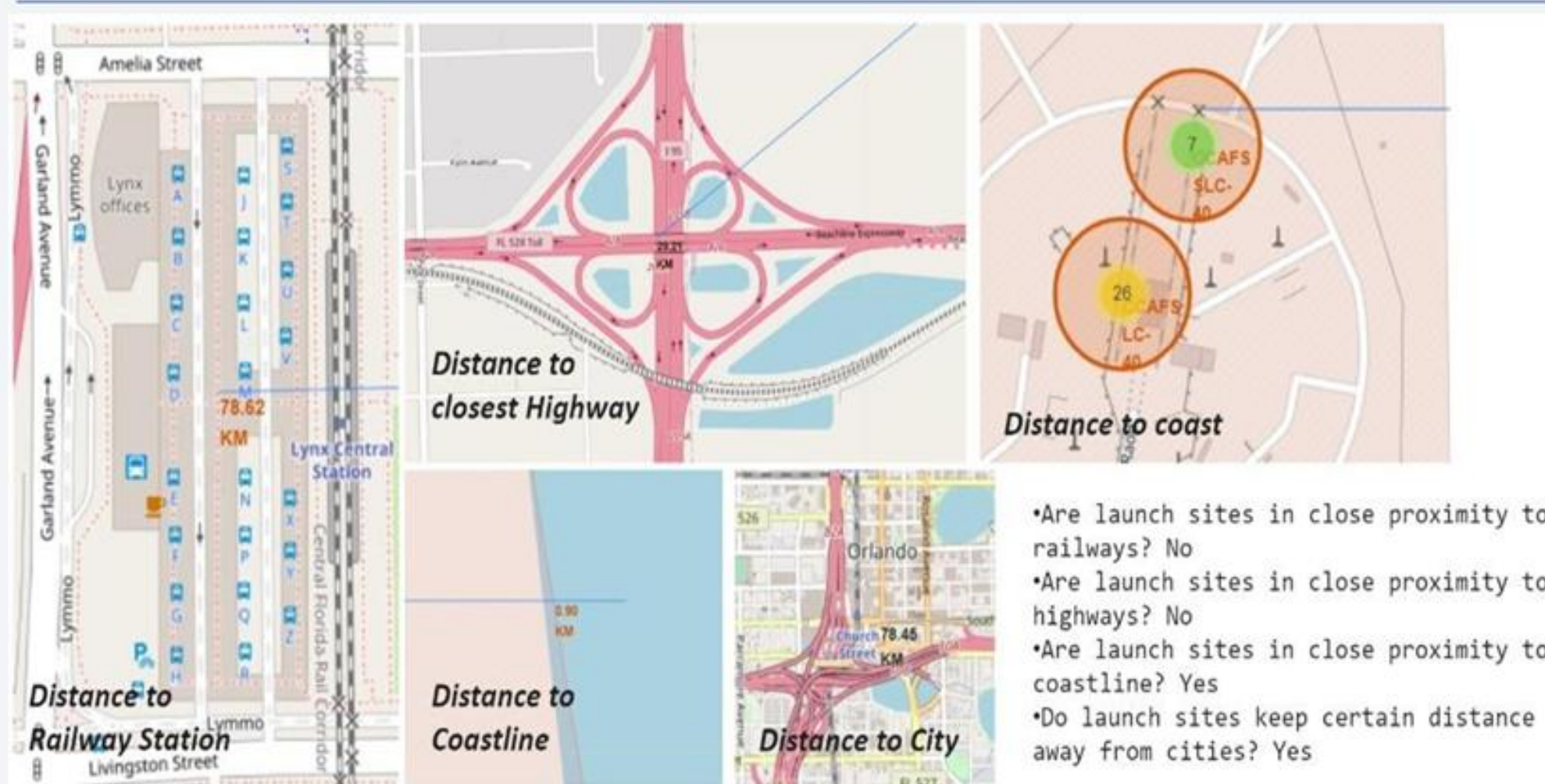
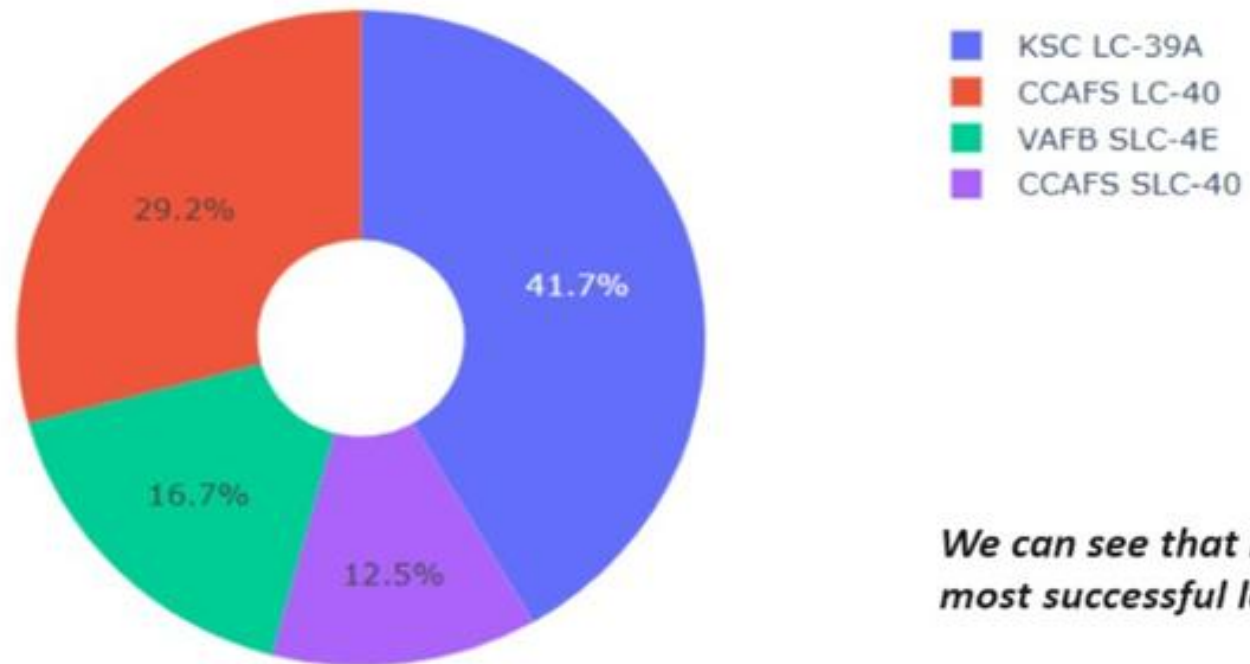Section 4

Build a Dashboard
with Plotly Dash

# Pie chart showing the success percentage achieved by each launch site

Total Success Launches By all sites

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

41.7%
29.2%
16.7%
12.5%

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
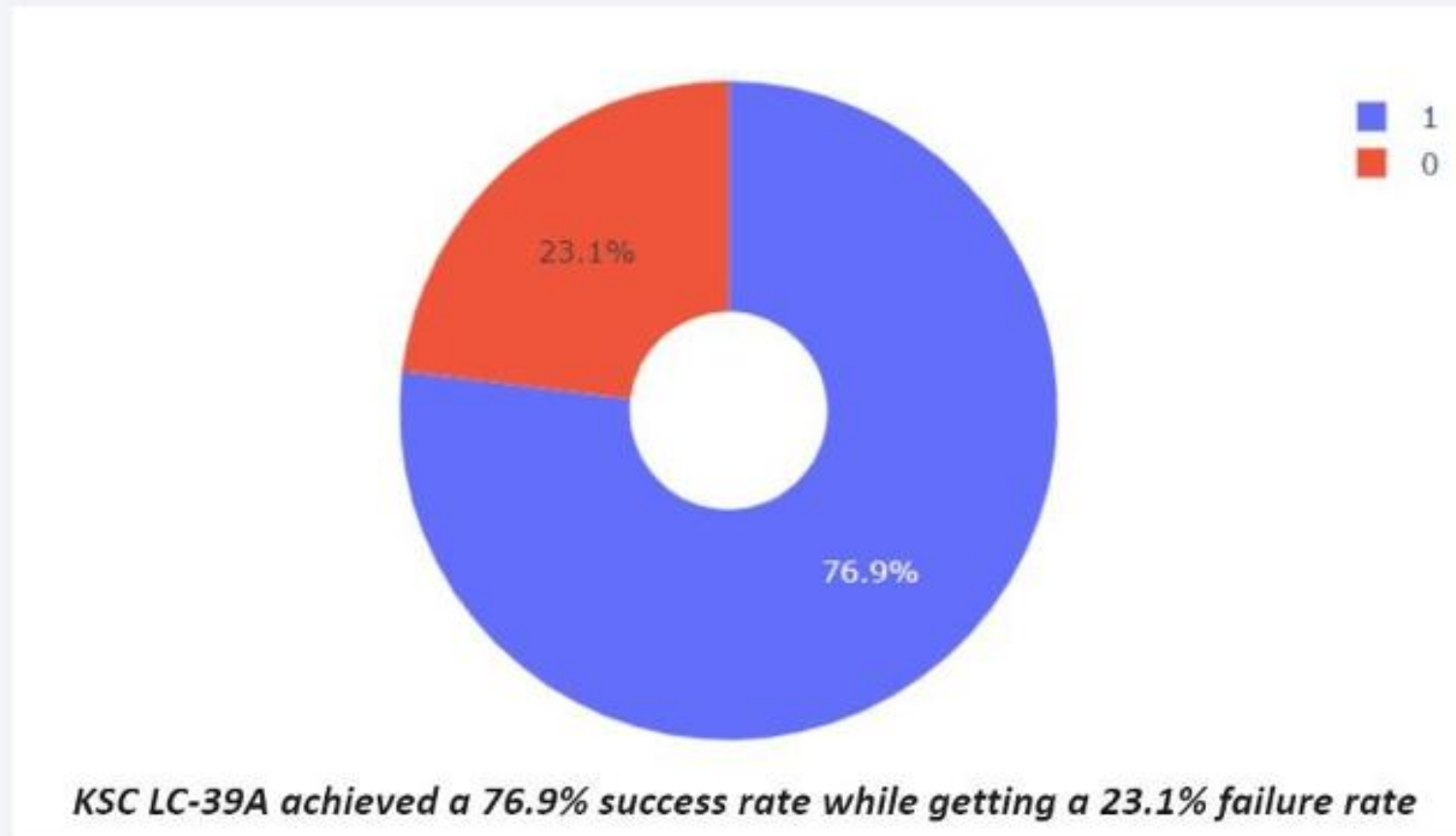
Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider
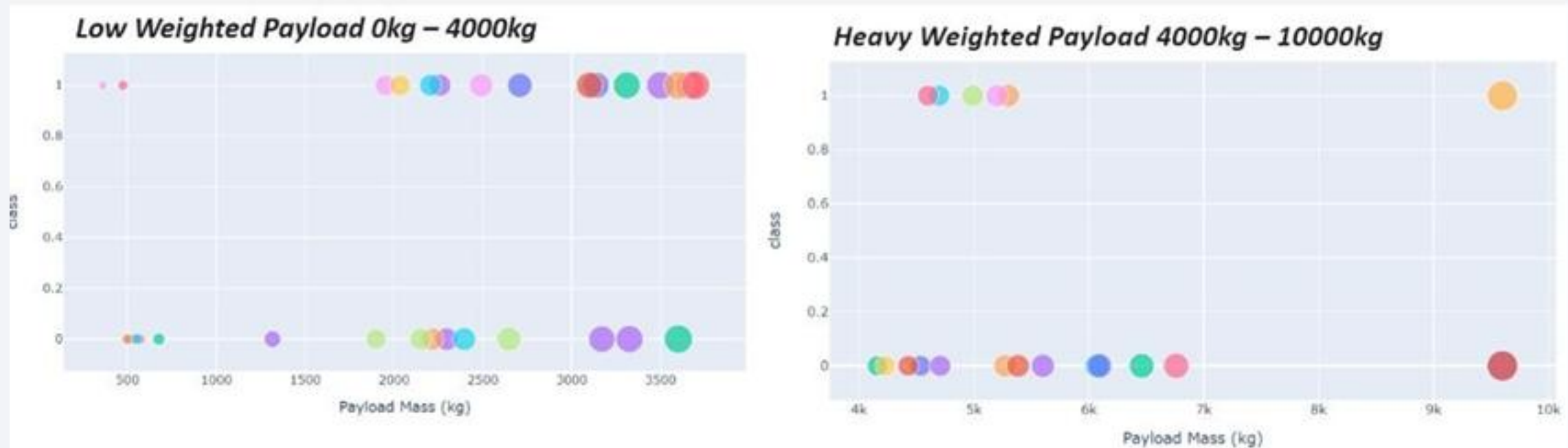
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- 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

- 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
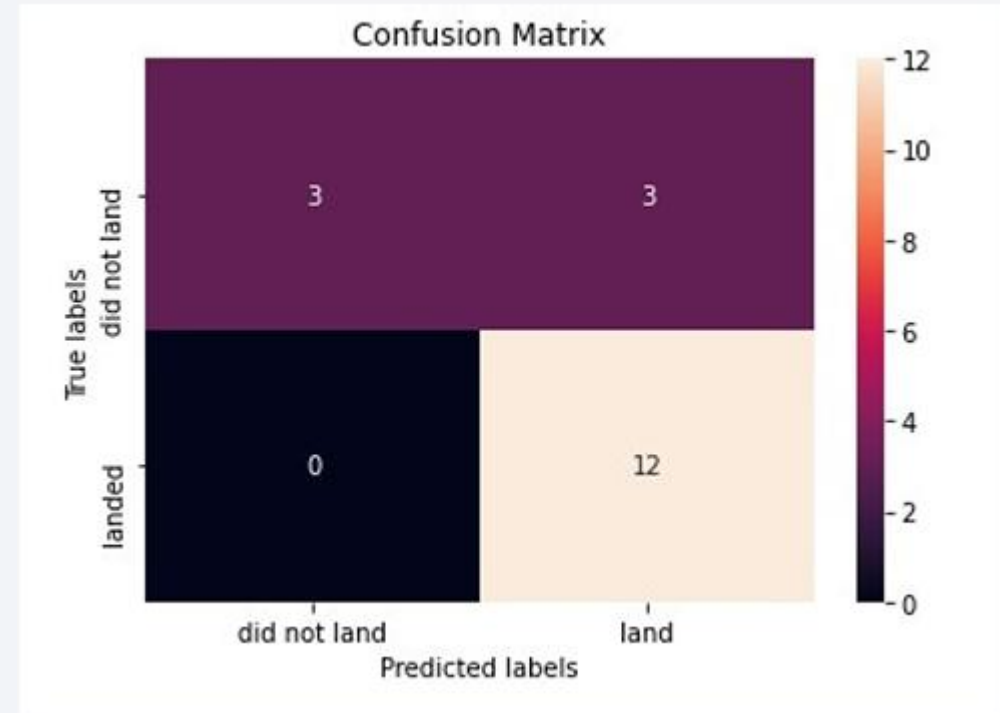
Conclusions

- We can conclude that:

    • The larger the flight amount at a launch site, the greater the success rate at a launch site.

    • Launch success rate started to increase in 2013 till 2020.

    • Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate. • KSC LC-39A had the most successful launches of any sites.

    • The Decision tree classifier is the best machine learning algorithm for this task

Thank you!