IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Sakshi.M>
<10-11-2022>

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 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.

The goal of this 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 or not?
  - What operating conditions are required to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical 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
  - How to build, tune, evaluate classification models

Data Collection

- Data collection was done using get request to the SpaceX API.
- Next, the response content was decoded as Json using .json() function call and turned it into a pandas dataframe using .json_normalize().
- Then the data was cleaned and checked, for missing values.
- In addition, web scraping was performed for Falcon 9 launch records 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

- Here the GET request was used to the SpaceX API to collect data, clean the requested data and to do some basic data wrangling and formatting.
- The link to the notebook is https://github.com/MSakkshi/Sak/blob/master/week%201-%20data%20collection.ipynb

In [50]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

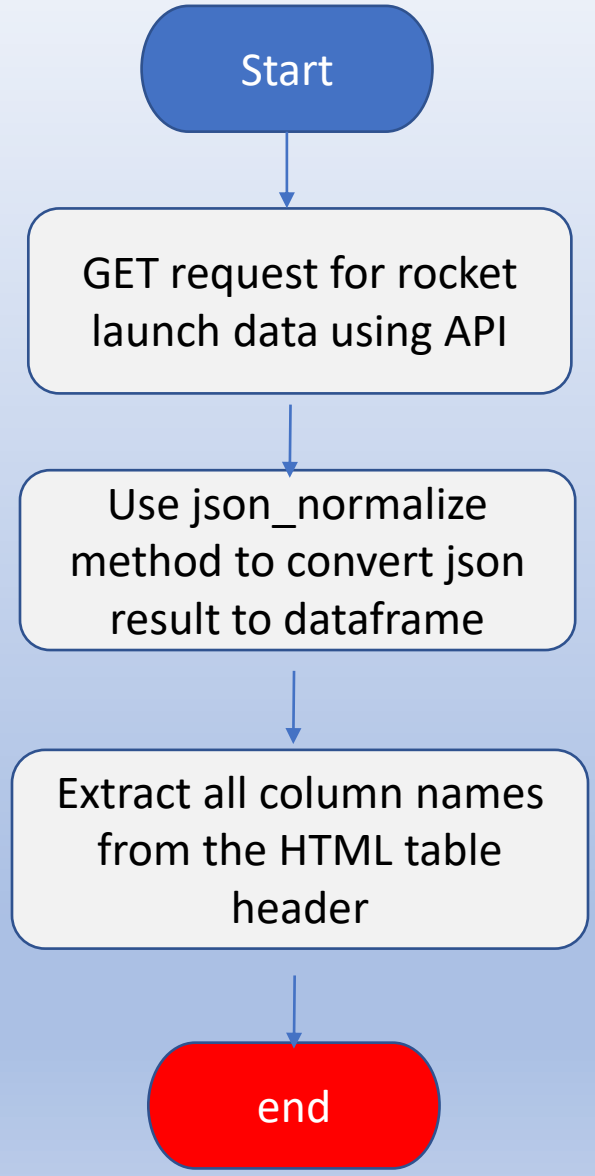
We should see that the request was successfull with the 200 status response code

In [51]:
```
response.status_code
```

Out[51]:
```
200
```

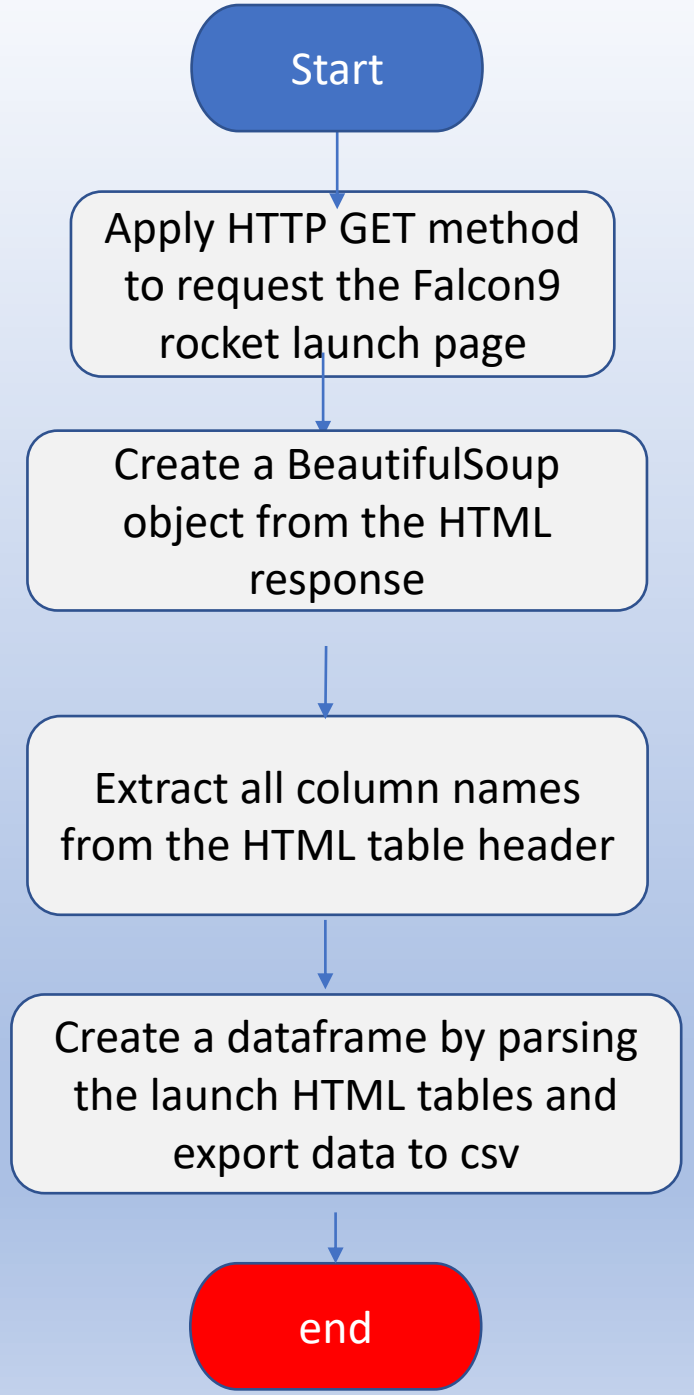
Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

In [52]:
```
# Use json_normalize meethod to convert the json result into a dataframe
static_json_df = response.json()
data = pd.json_normalize(static_json_df)
```



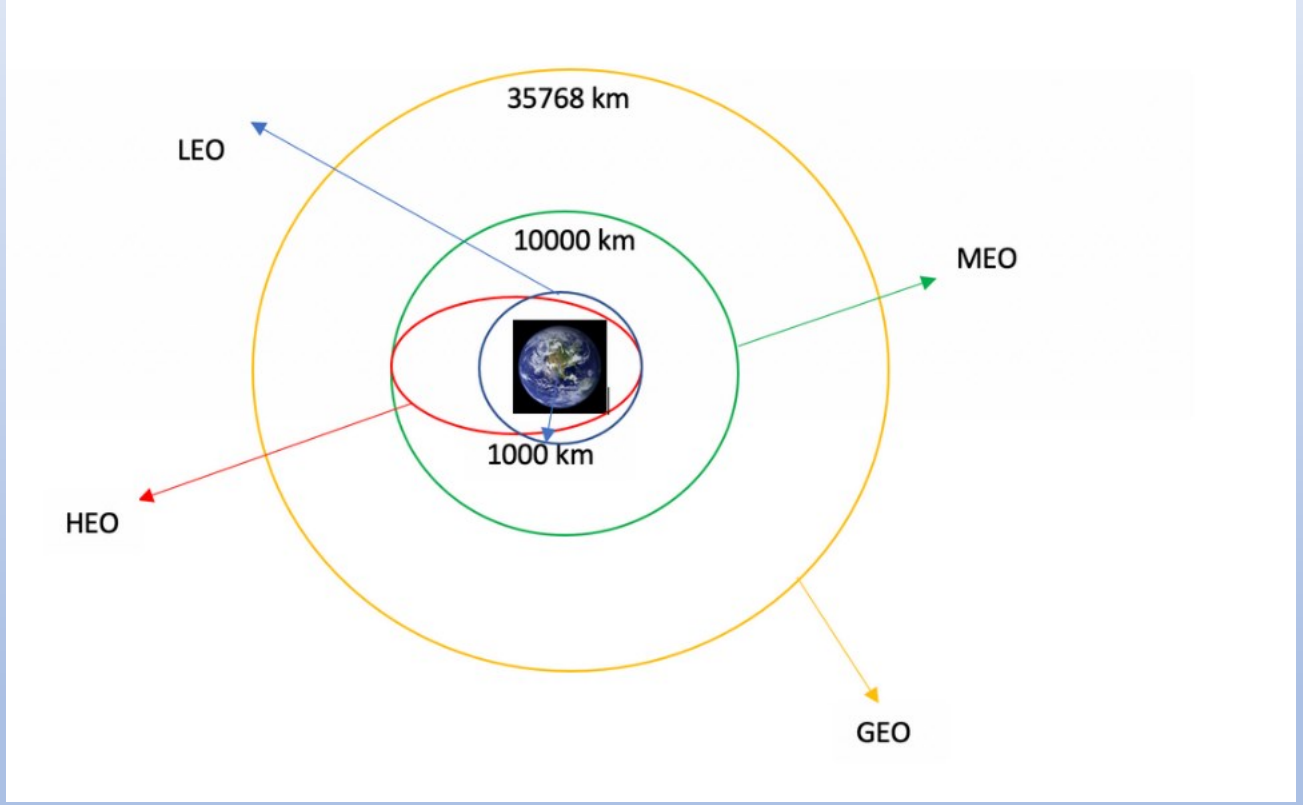
Data Collection - Scraping

- In this web scrapping is applied to webscrap Falcon 9 launch records with BeautifulSoup
- Then the table is parsed and converted into a pandas dataframe.
- The link to the notebook is https://github.com/MSakkshi/Sak/blob/master/week%201-%20data%20collection%20with%20web%20scraping.ipynb



Data Wrangling

- In this performed exploratory data analysis was performed and determined the training labels.
- Also the number of launches at each site was calculated, and the number and occurrence of each orbits
- Landing outcome label was created from outcome column and exported the results to csv.
- The github link to the notebook is https://github.com/MSakkshi/Sak/blob/master/week%201-data%20wrangling.ipynb

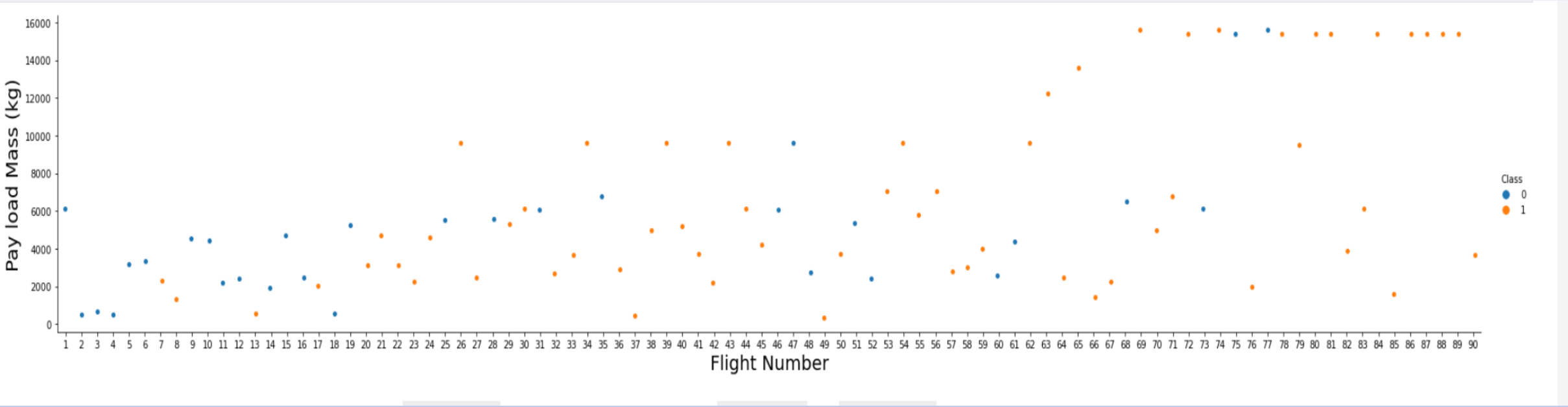


EDA with Data Visualization

- In this the data was explored 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.
- The charts which were plotted during EDA with data visualization includes(summarized) as follows

1. **Flight Number vs PayloadMass(Scatter point chart):**

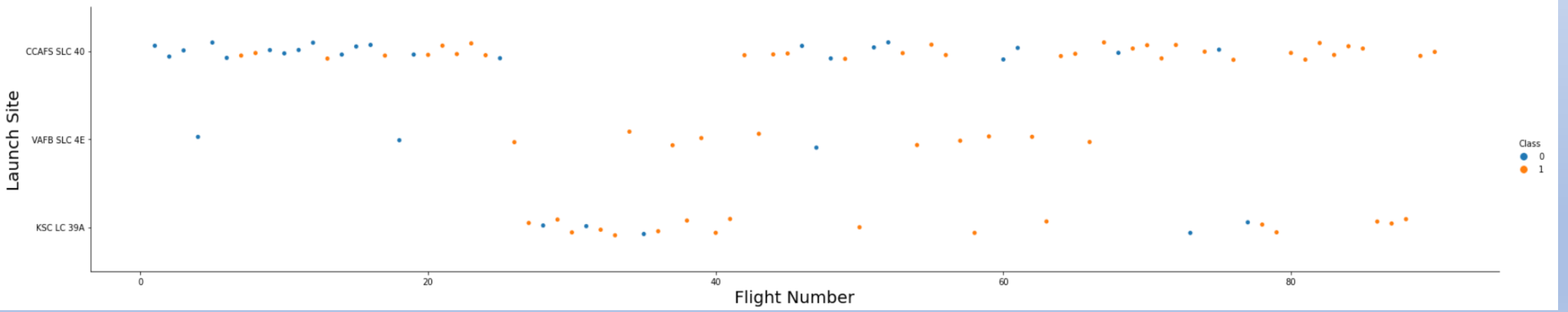
Purpose: To find out how the Flight Number (indicating the continuous launch attempts) and Payload variables would affect the launch outcome.


Pay load Mass (kg)
Flight Number
Class
0
1

EDA with Data Visualization

2. Flight number vs Launch sites

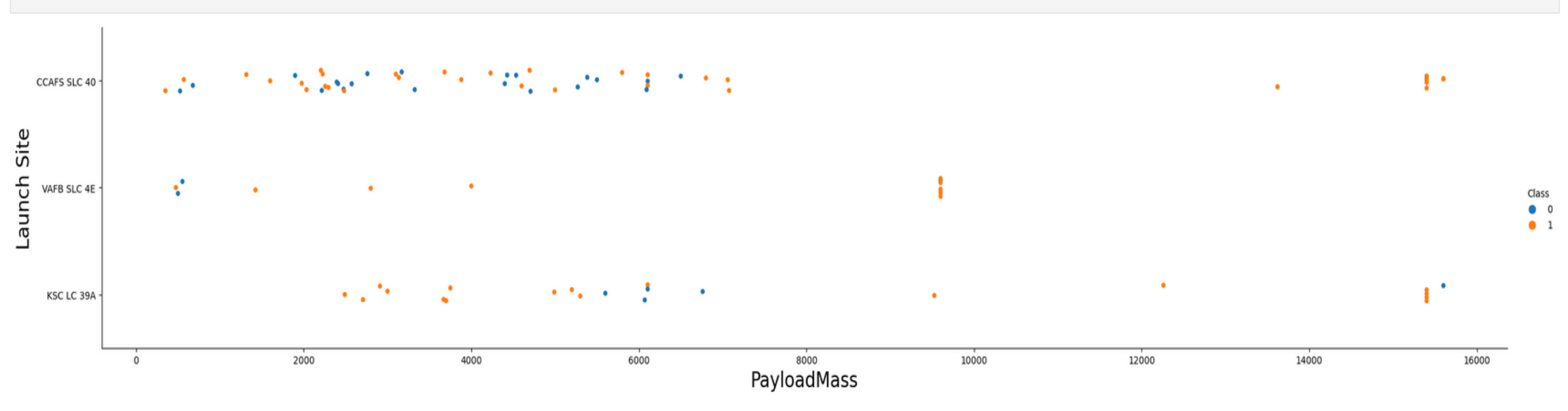
Purpose: To visualize the relationship between the flight numbers and the launch sites using the function catplot.



EDA with Data Visualization

3. Launch sites vs payload mass(Scatter point chart)

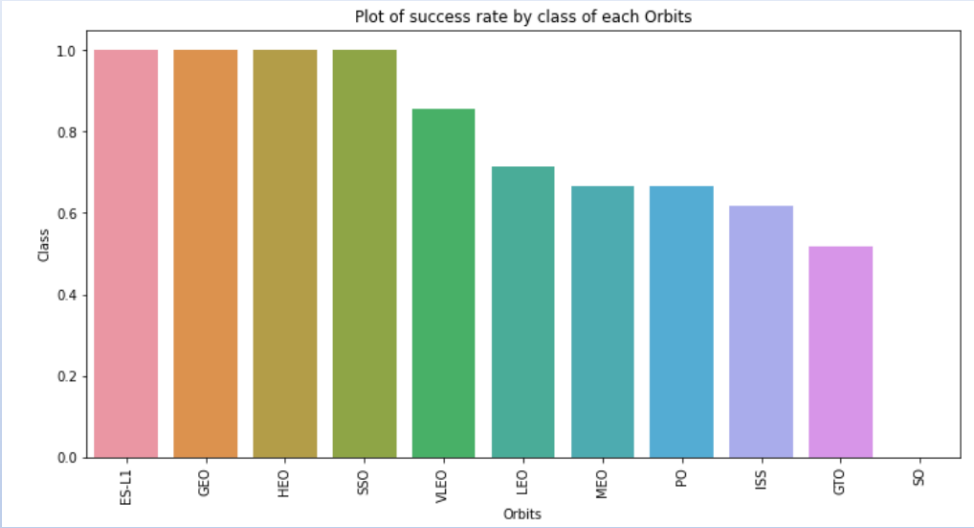
Purpose: To observe if there is any relationship between launch sites and their payload mass

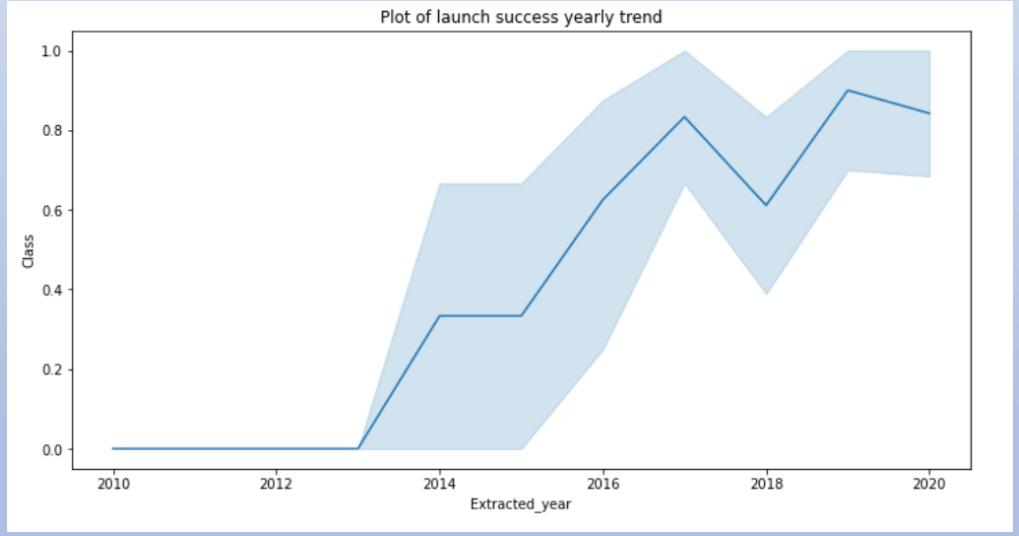


EDA with Data Visualization

4. Bar chart for the success rate of each orbit.

5. A line chart to visualize the launch success yearly trend

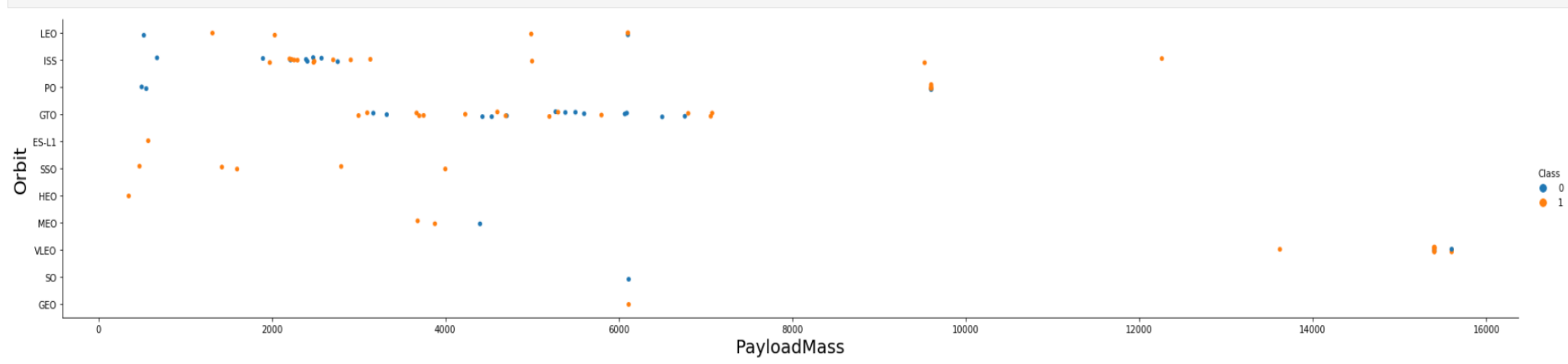




EDA with Data Visualization

6. Payload vs Orbit type

Purpose: To reveal the relationship between payload and orbit type.



EDA with Data Visualization

GitHub URL of completed EDA with data visualization notebook:

https://github.com/MSakkshi/Sak/blob/master/week%202-%20EDA%20with%20data%20visualization.ipynb

EDA with SQL

- EDA with SQL was applied to get insight from the data. The summary of the SQL queries performed includes finding out the:
  - The names of unique launch sites in the space mission.
  - 5 records where launch sites begin with string “CCA”
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The date when the first successful landing outcome in ground pad was achieved.
  - List of the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - The total number of successful and failure mission outcomes.
  - The names of the booster versions which have carried the maximum payload mass
  - the records which displays the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
  - The count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

EDA with SQL

The link to EDA with SQL notebook is:
https://github.com/MSakkshi/Sak/blob/master/jupyter_labs_eda_sql_coursera_sqllite.ipynb

Build an Interactive Map with Folium

Markers of all Launch Sites:

❖ Added Marker with Circle, Popup Label and Text Label of NASA Johnson Space Center using its latitude and longitude coordinates as a start location.

❖ Added Markers with Circle, Popup Label and Text Label of all Launch Sites using their latitude and longitude coordinates to show their geographical locations and proximity to Equator and coasts.

Coloured Markers of the launch outcomes for each Launch Site:

❖ Markers were created for all launch records. If a launch was successful (class=1), then a green marker was used and if a launch was failed, then a red marker (class=0) was used

Build an Interactive Map with Folium

Distances between a Launch Site to its proximities:

- Added coloured Lines to show distances between the Launch Site and its proximities like Railway, Highway, Coastline and Closest City.

The GitHub URL of the notebook is https://github.com/MSakkshi/Sak/blob/master/week%203-%20interactive%20visual%20analytics%20with%20folium.ipynb

Build a Dashboard with Plotly Dash

Plots/graphs and interactions that have added to the dashboard:

- **Launch Sites Dropdown List**:

  Added a dropdown list to enable Launch Site selection.

- **Pie Chart showing Success Launches (All Sites/Certain Site):**

  Added a pie chart to show the total successful launches count for all sites and the Success vs. Failed counts for the site, if a specific Launch Site was selected.

Build a Dashboard with Plotly Dash

- **Slider of Payload Mass Range：**

  Added a slider to select Payload range.

- **Scatter Chart of Payload Mass vs. Success Rate for the different Booster Versions**:

  Added a scatter chart to show the correlation between Payload and Launch Success.

- **The github url is** https://github.com/MSakkshi/Sak/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- In this, first the data was loaded using numpy and pandas, then it was transformed and split into into training and testing.
- Next different machine learning models were built and different hyperparameters were defined.
- One of the main aims was to find the best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Accuracy was used as the metric for the model.
- And then the best performing classification model was found.
- The link to the notebook is https://github.com/MSakkshi/Sak/blob/master/week%204-%20Machine%20learning%20prediction.ipynb

Find the method performs best:

In [29]:
```
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'}
```

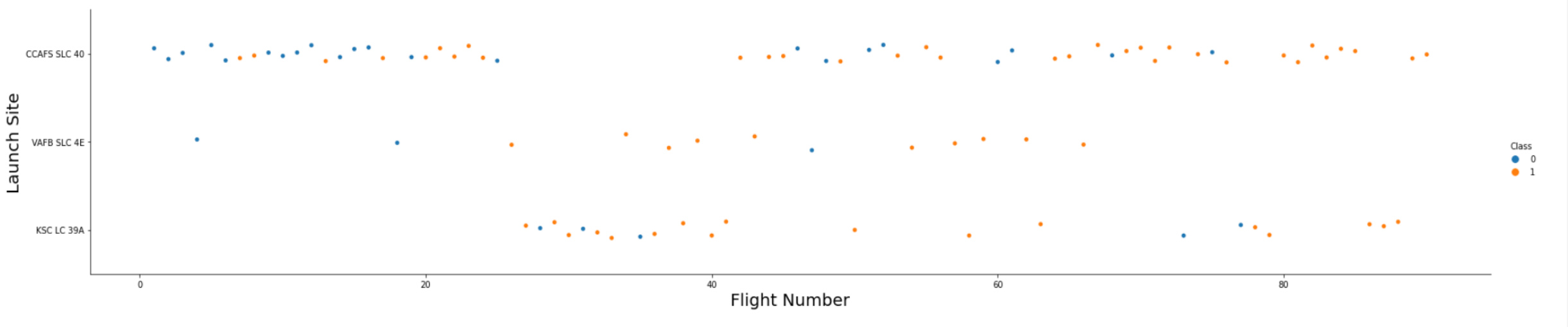
Results

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

Section 2

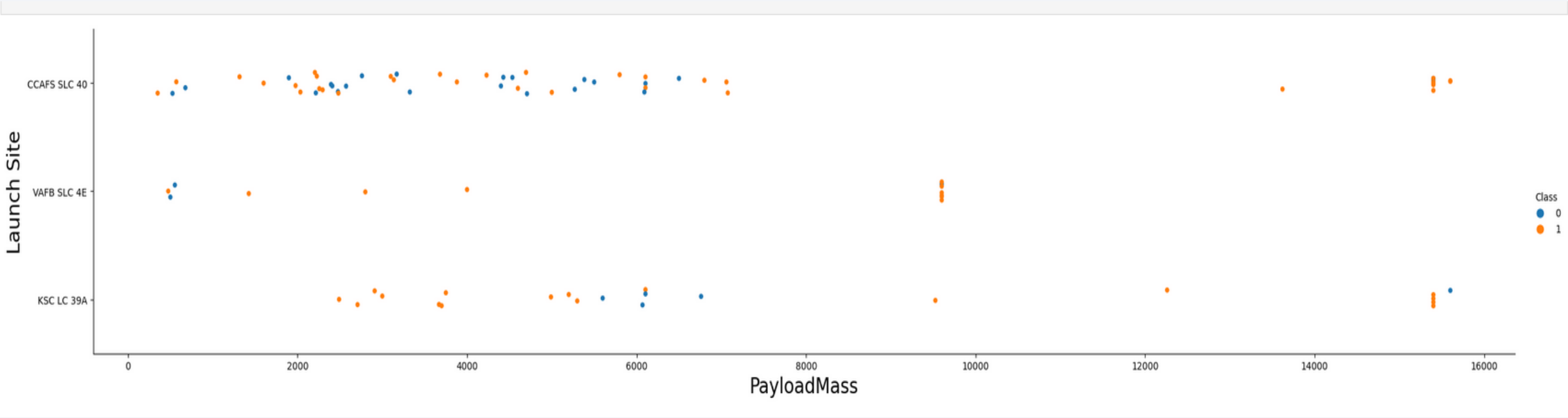
Insights drawn from EDA

Flight Number vs. Launch Site



- From the plot, it can be observed that the larger the flight amount at a launch site, the greater the success rate at a launch site.

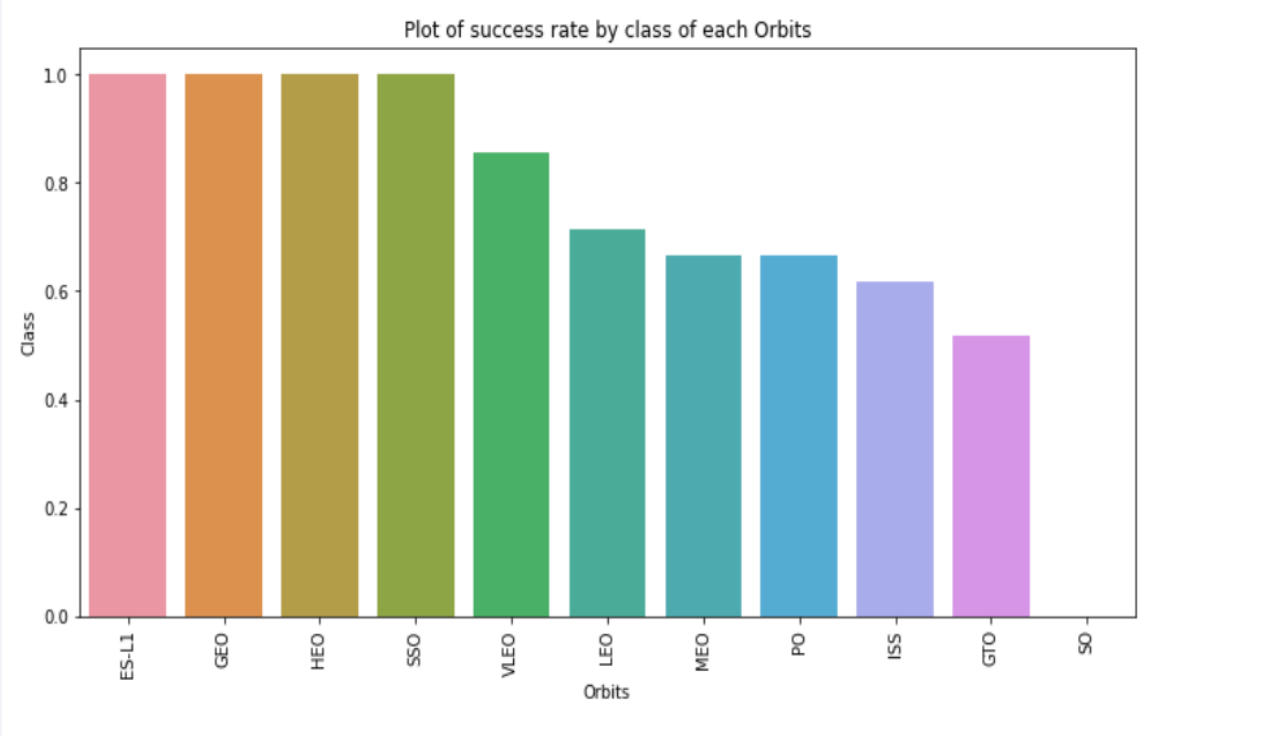
Payload vs. Launch Site



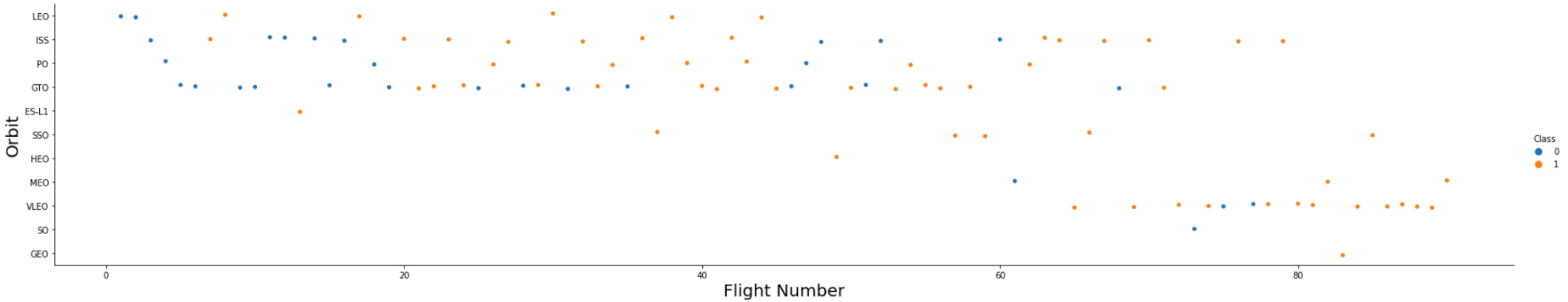
- From the above graph it can be observed greater the payload mass for launch site the higher the success rate for the rocket.

Success Rate vs. Orbit Type

- From the bar chart, it can be observed that ES-L1, GEO, HEO, SSO, VLEO have the more success rate.

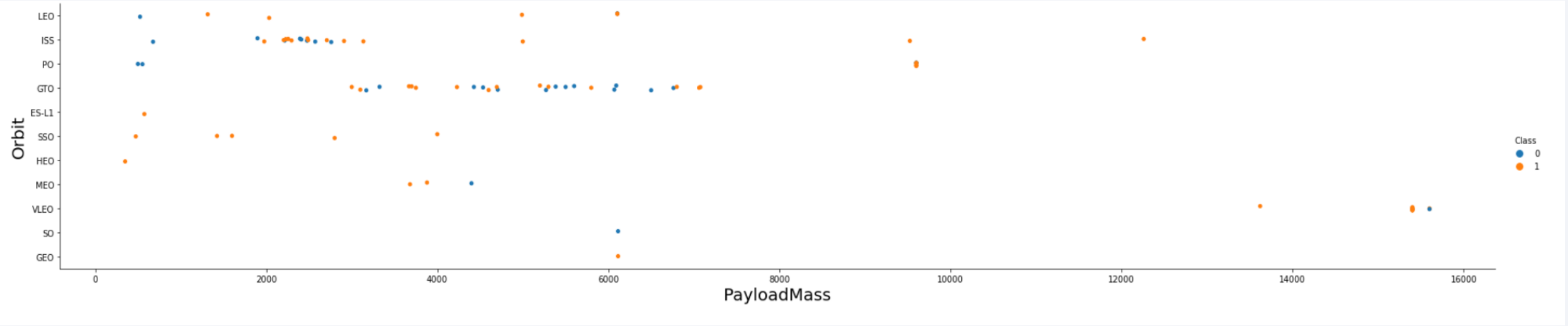


Flight Number vs. Orbit Type



- The plot below shows the Flight Number vs. Orbit type. It can be observed 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.

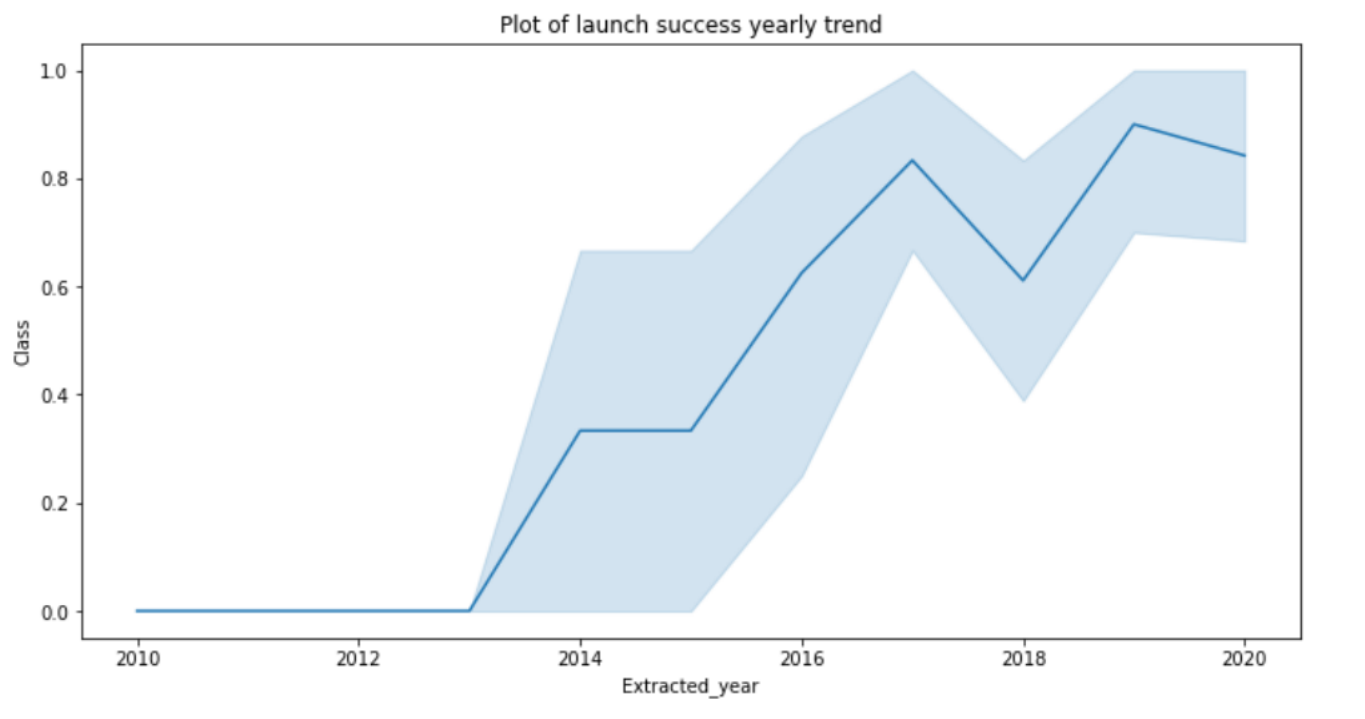
Payload vs. Orbit Type



- It can be observed that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.

Launch Success Yearly Trend

- From the plot, it can be observed that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- Here in this key word **DISTINCT** is used to show only unique launch sites from the SpaceX data.

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

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

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
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- The above screenshot depicts the query used to display 5 records where launch sites begin with `CCA`

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```

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

Total Payload Mass by NASA (CRS)
45596

- In this it can be observed that the total payload carried by boosters from NASA is 45596 using the query above.

Average Payload Mass by F9 v1.1

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

```
[ ] %sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster Version F9 v1.1" FROM SPACEXTBL \
    WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

Average Payload Mass by Booster Version F9 v1.1

2928.4

- The average payload mass carried by booster version F9 v1.1 is 2928.4

First Successful Ground Landing Date

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

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad" FROM SPACEX \
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

First Succesful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Here in the query **WHERE** clause was used to filter for boosters which have successfully landed on drone ship and **AND** clause was applied condition to determine successful landing with payload mass greater than 4000 but less than 6000

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

```
%sql select BOOSTER_VERSION from SPACEXTBL where Landing__Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

From the screenshot it can be seen that the number of successful mission outcomes is 100 and failure mission outcomes is 1.

List the total number of successful and failure mission outcomes

In [45]:
```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)'
```

Done.

Out[45]:

1
100

Boosters Carried Maximum Payload

- Here a subquery is used to list the names of the booster_versions which have carried the maximum payload mass

```
[ ] %sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEXTBL\
    WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- Here the combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions are used 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

- The query and the result of ranking the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 is as shown below.

Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

```
In [48]: %sql select * from SPACEXTBL where Landing__Outcome like 'Success%' and (DATE between '2010-06-04' and '2017-03-20') order by date desc
```

Out[48]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

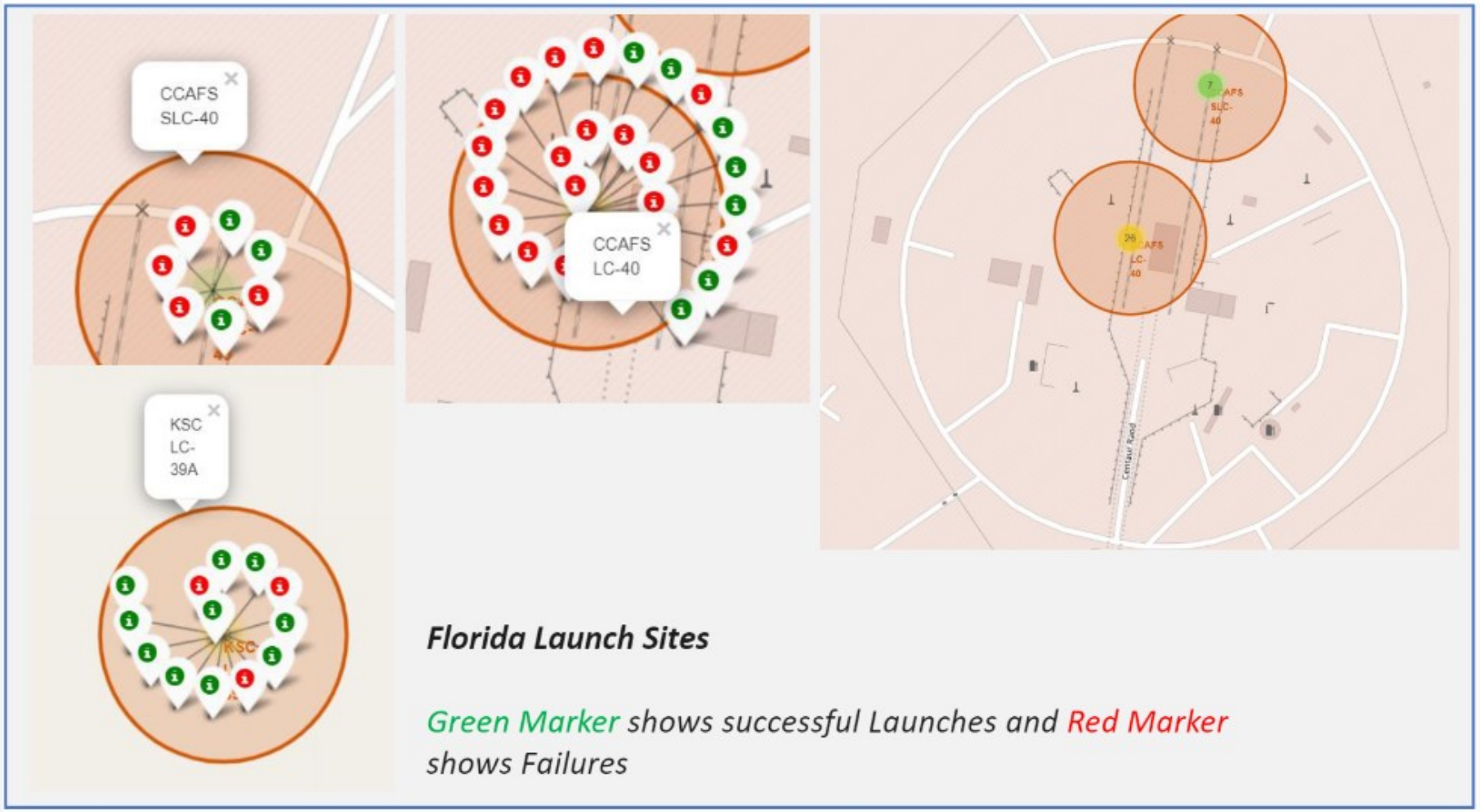
Section 3

Launch Sites Proximities Analysis

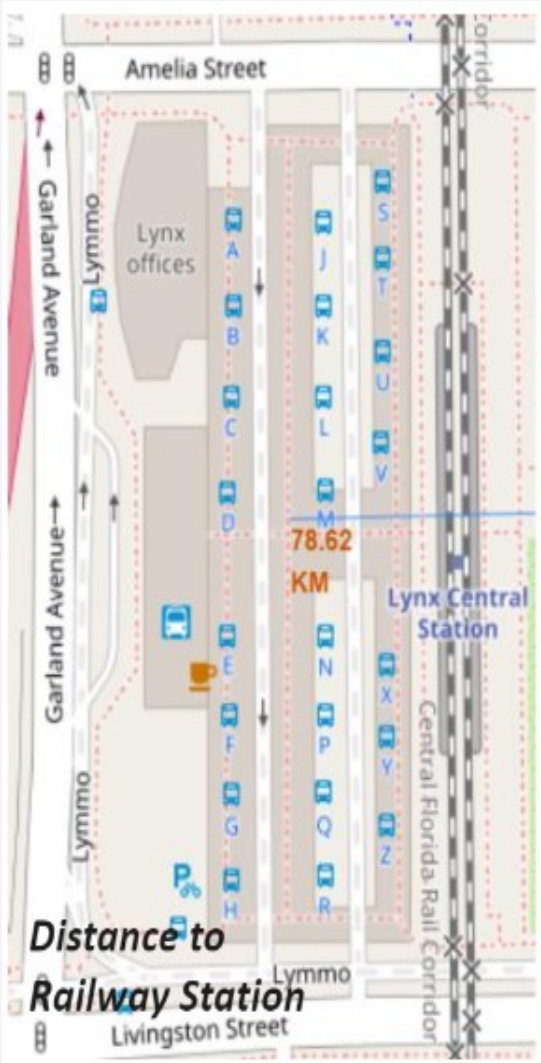
<All launch sites global map markers>

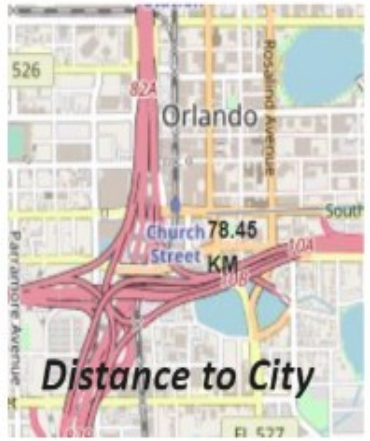
We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

<Markers showing launch sites with color labels>

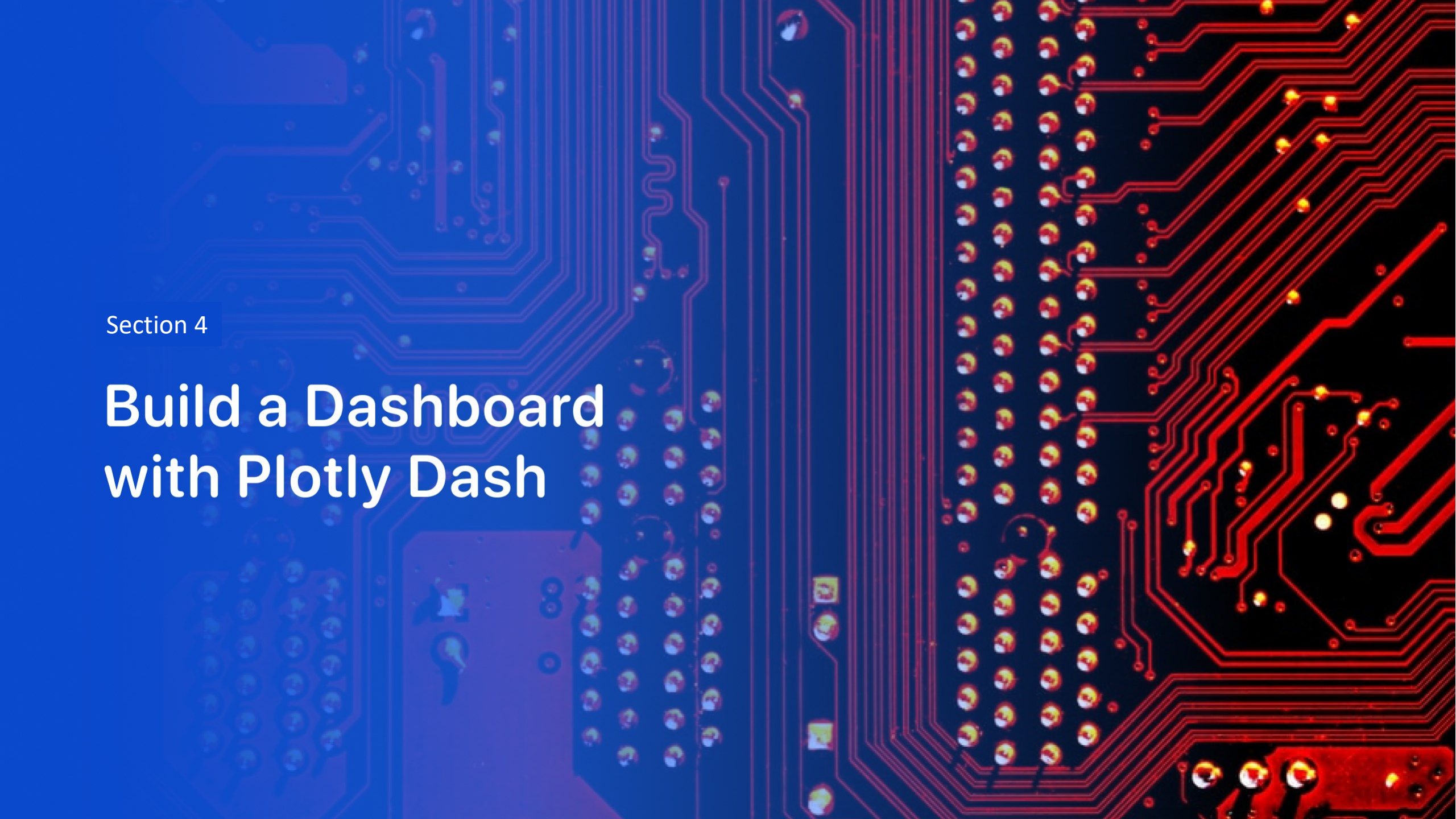


<Launch Site distance to landmarks>

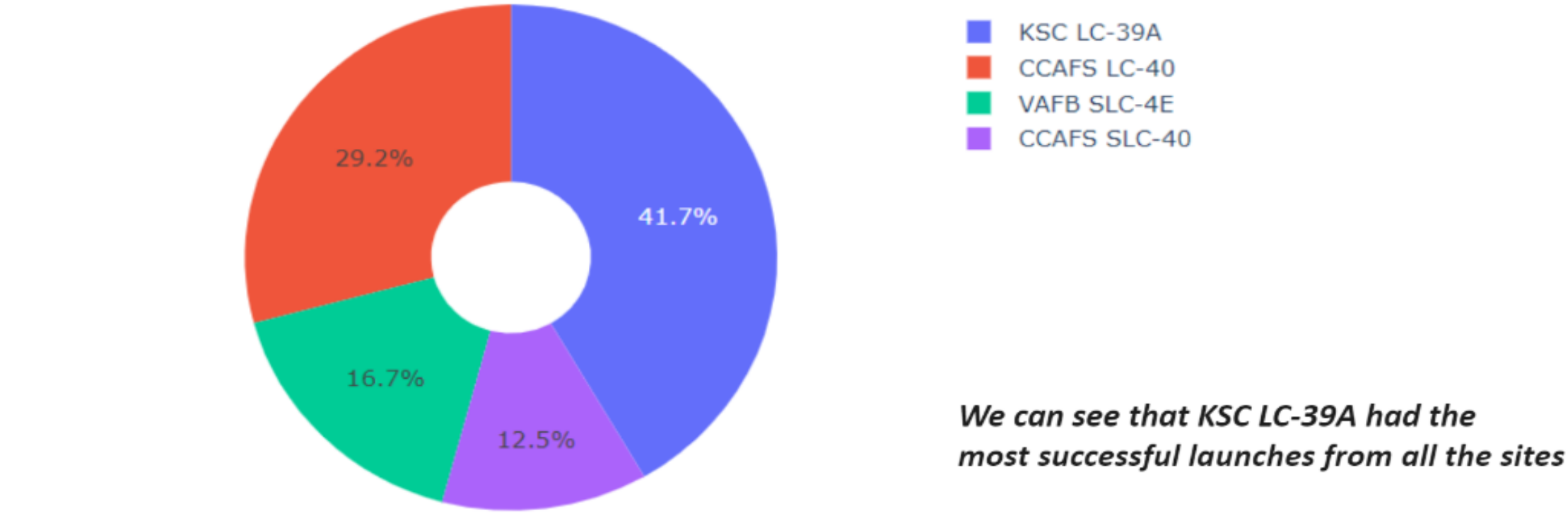




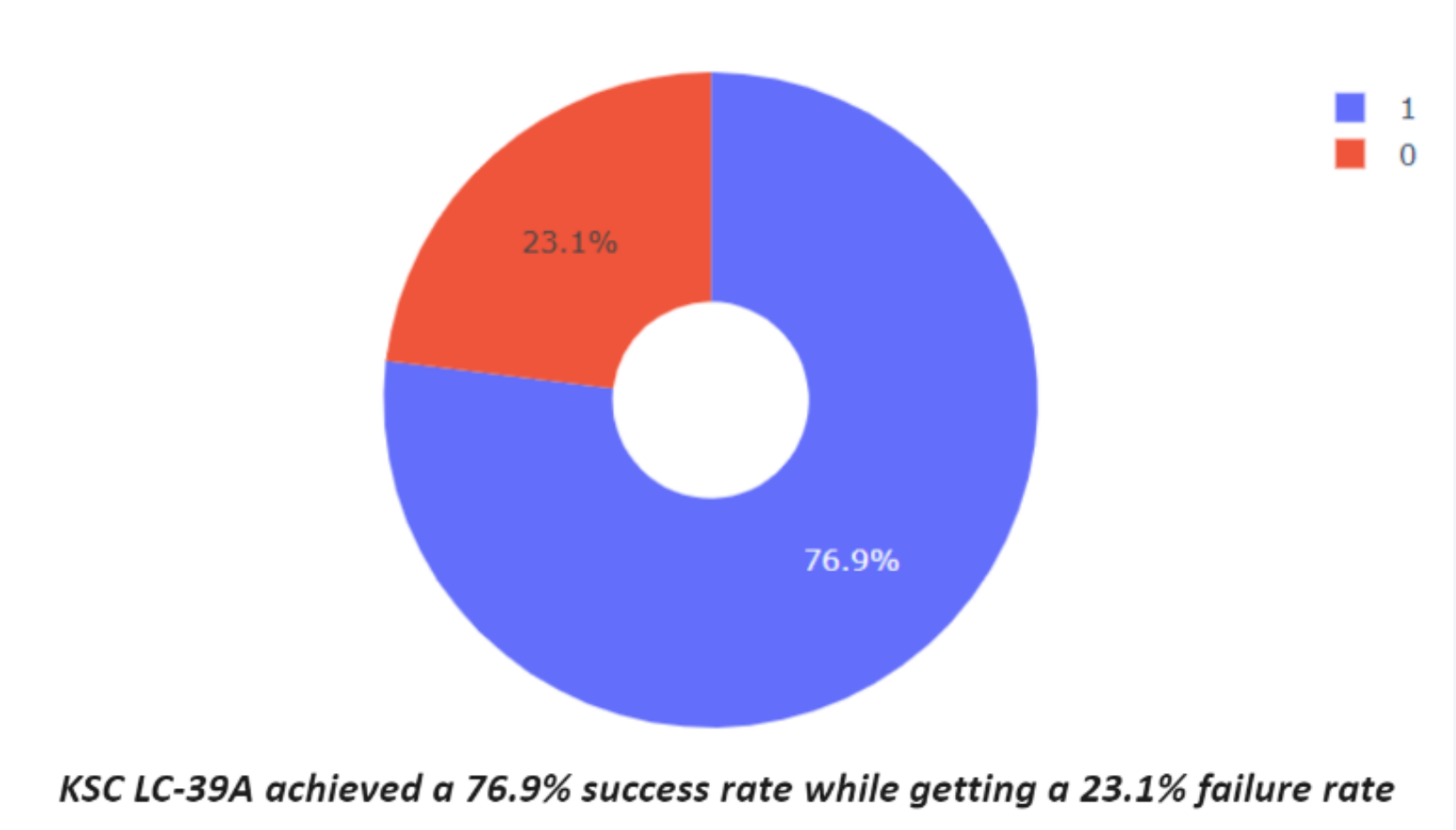
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

<Pie chart showing the success percentage achieved by each launch site>

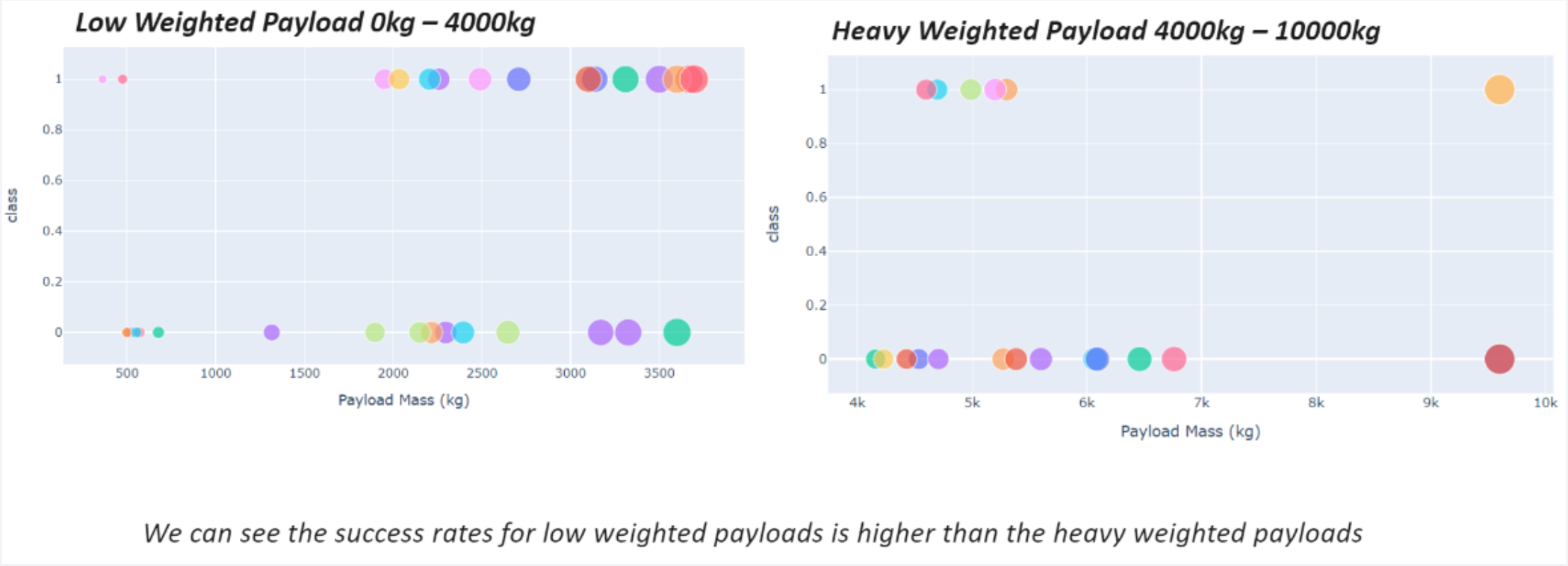


<Pie chart showing the Launch site with the highest launch success ratio>



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

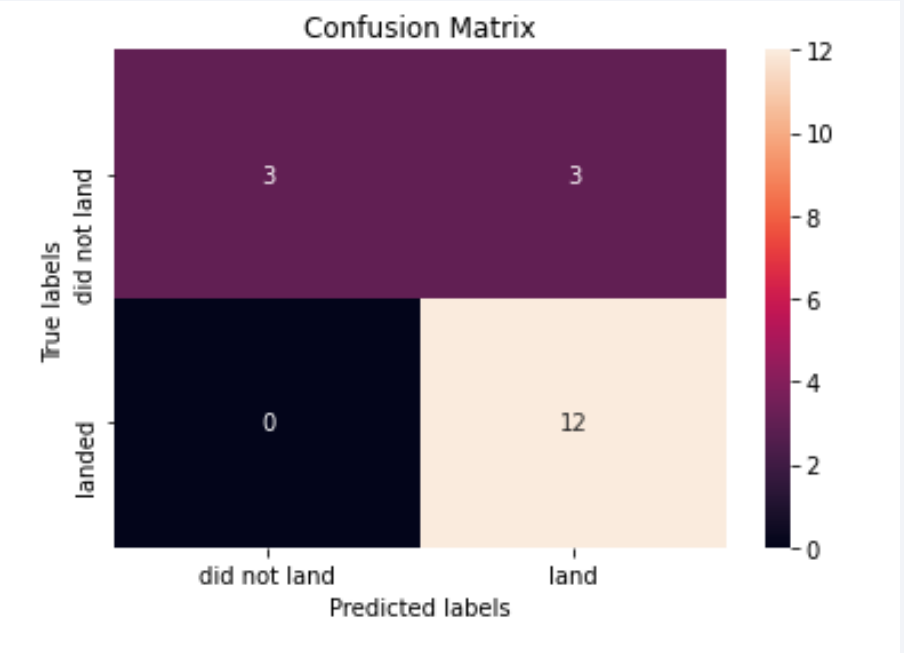
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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 0.873214285714285 6
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.



Conclusions

- 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!