IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Phan Anh Kiet
Dec 10th, 2023

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:
  - 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
  - Standardize data using StandardScaler and tune models using GridSearchCV

Data Collection

- The data was collected using various methods
  - 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

- We used the get request on the SpaceX API to collect data, clean the obtained data and performed basic data wrangling and formatting.
- The link to the notebook: https://github.com/KietUwU/Data-Science---Coursera-Capstone/blob/main/Hands-on%20Lab:%20Complete%20the%20Data%20Collection%20API%20Lab.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)
```

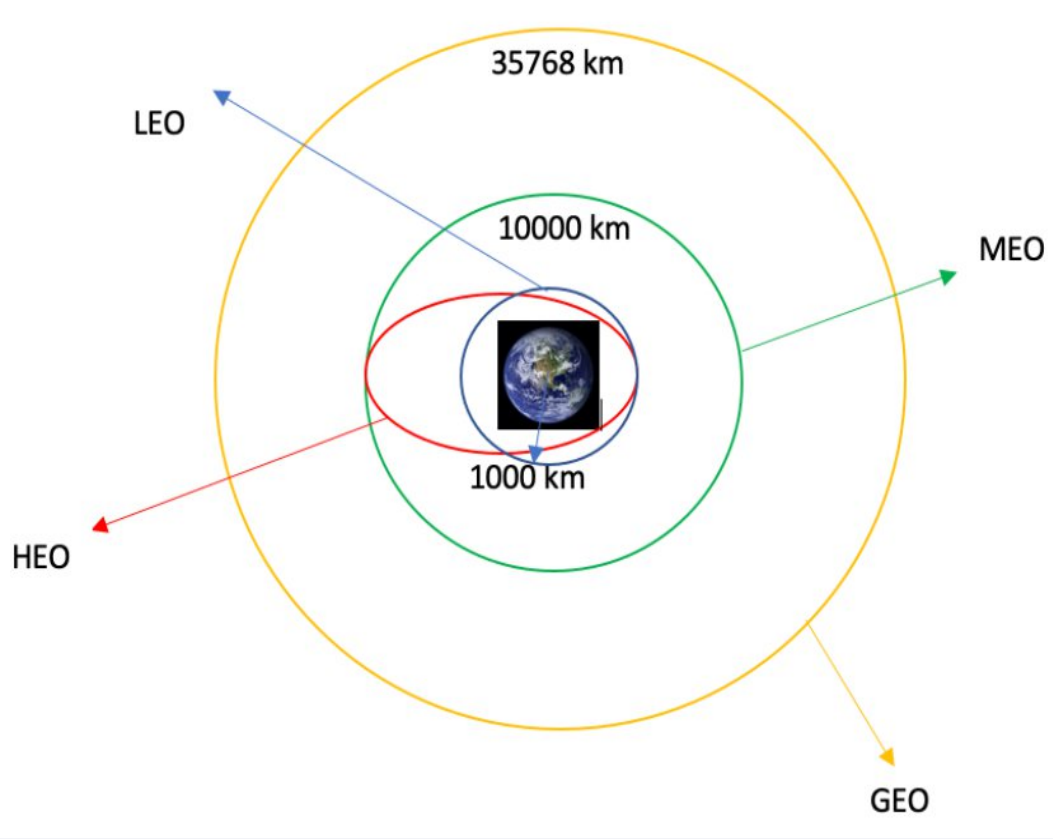
3. We then performed data cleaning and filling in the missing values

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
```

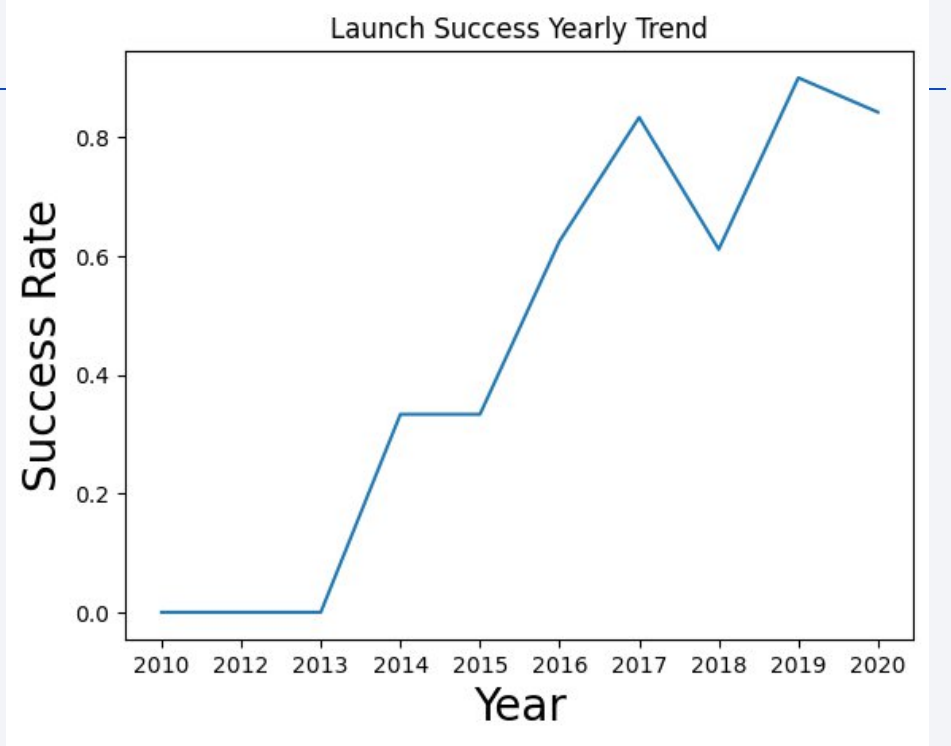
Data Wrangling

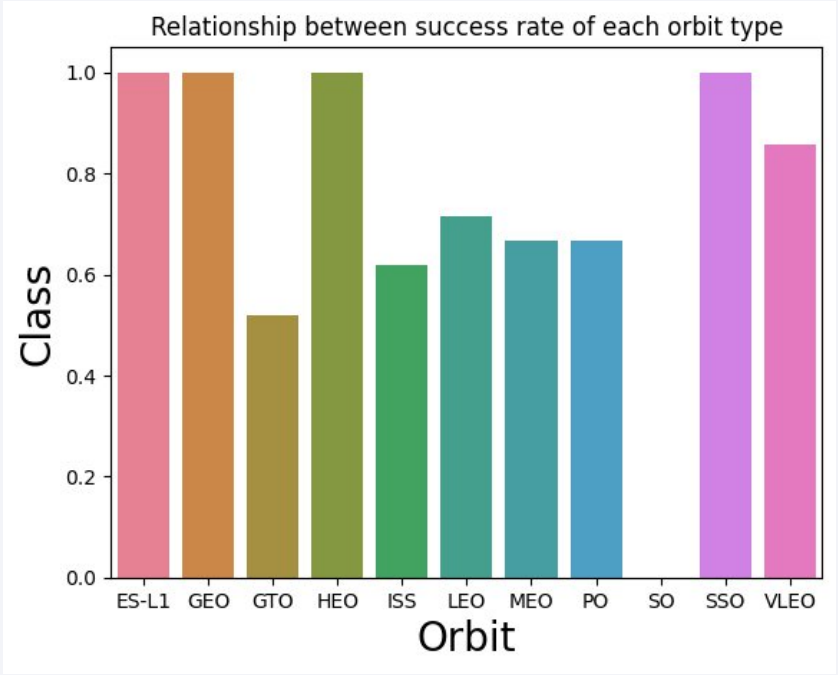


- Perform exploratory data analysis and determined the training labels.
- Calculate the number of launches at each site, and the number and occurrence of each orbits
- Create landing outcome label from outcome column and exported the results to .csv file.
- The link to the notebook: https://github.com/KietUwU/Data-Science---Coursera-Capstone/blob/main/Hands-on%20Lab%3A%20Data%20Wrangling.ipynb

EDA with Data Visualization

- Visualize 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.
- These graphs help us visualize the factors that affect the success rate of a launch and to which extent.





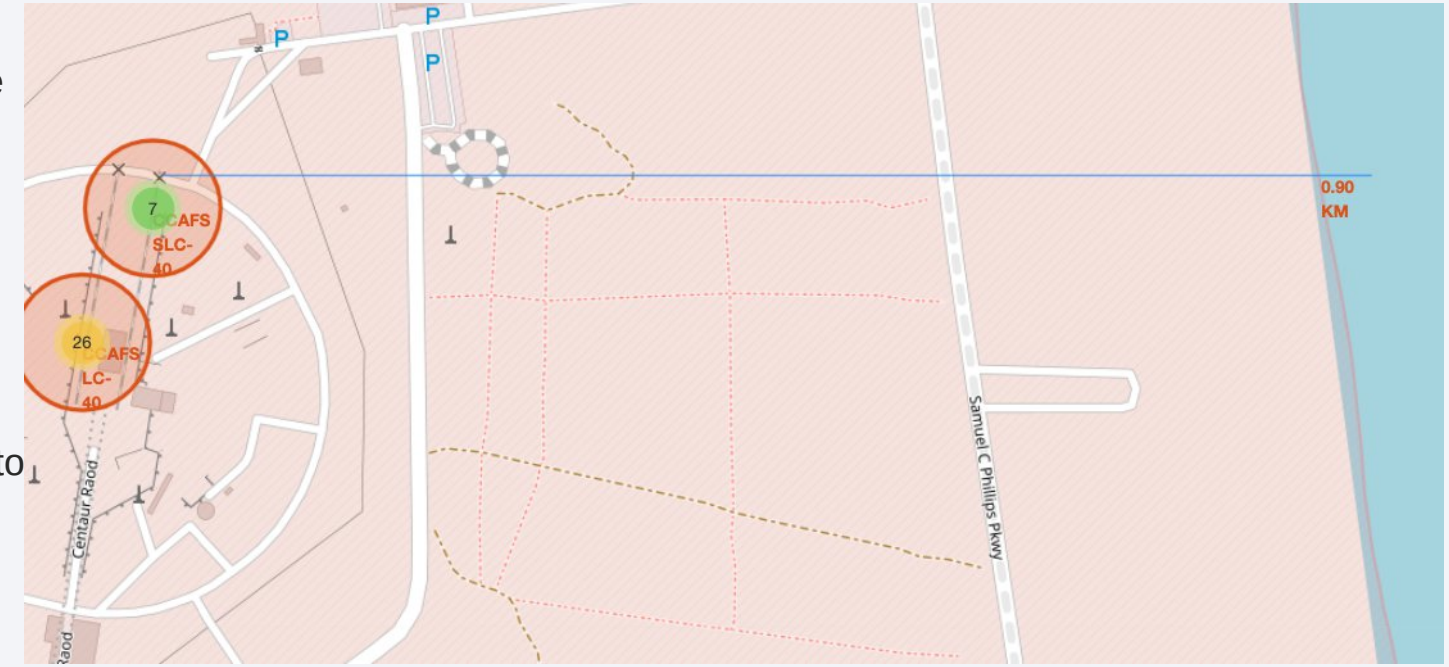
- The link to the notebook: https://github.com/KietUwU/Data-Science---Coursera-Capstone/blob/main/EDA%20with%20Visualization%20Lab.ipynb

EDA with SQL

- There are 10 SQL query statements, their purpose is to highlight features and trends in the dataset:
  - Display the names of the unique launch sites in the space mission
  - 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 successful landing outcome in ground pad was achieved.
  - 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.
- The link to the notebook: https://github.com/KietUwU/Data-Science---Coursera-Capstone/blob/main/Hands-on%20Lab%3A%20Complete%20the%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

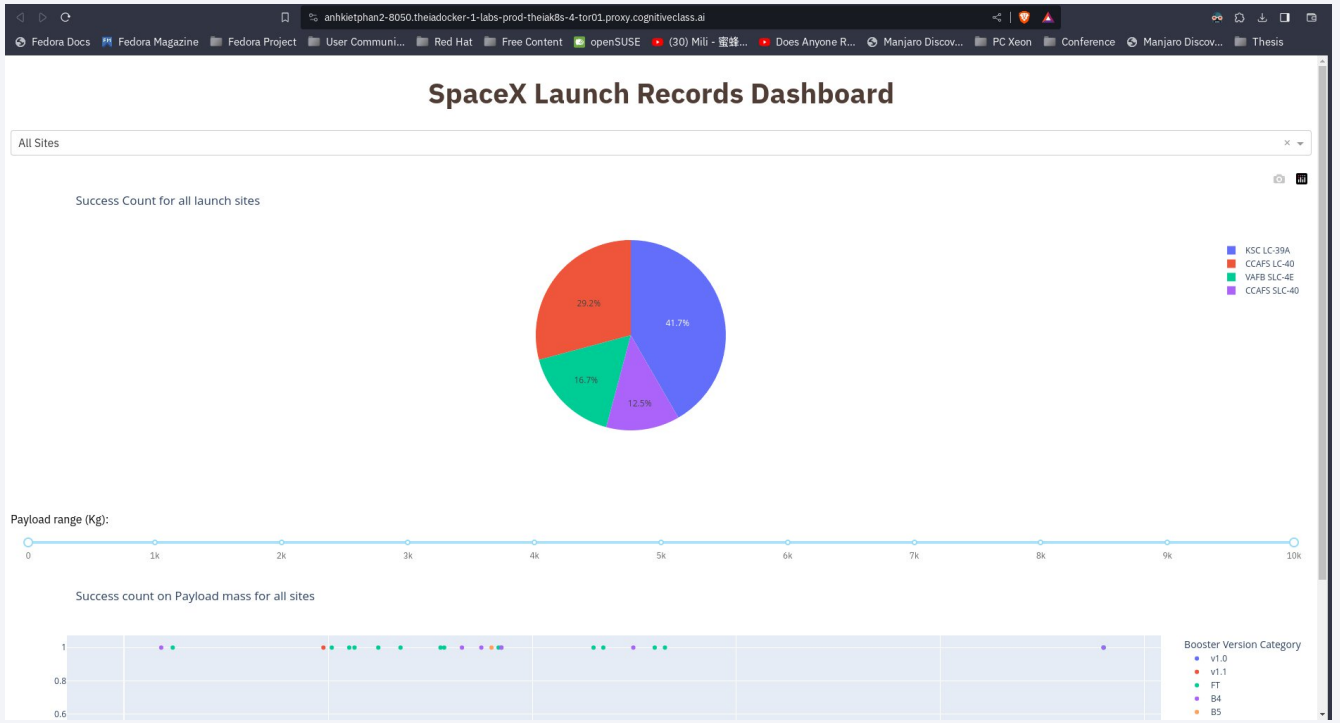
- Mark all launch sites, and add map objects such as markers, circles, lines to mark the success or failure of launches for each site on the Folium map.
- Assign the feature launch outcomes (failure or success) to class 0 and 1.
- Using the color-labeled marker clusters, identify which launch sites have higher success rate.
- We calculated the distances between a launch site to its proximity. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.
- The link to the notebook: https://github.com/KietUwU/Data-Science---Coursera-Capstone/blob/main/Hands-on%20Lab%3A%20Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb



Distance from a Launch Site to the Coast

Build a Dashboard with Plotly Dash

- Build an interactive dashboard with Plotly dash
- Plot an interactive pie chart showing the Success count of the Launch Sites
- Plot a scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- The link to the Github repo: https://github.com/KietUwU/Data-Science---Coursera-Capstone/blob/main/spacex_dash_app.py



Predictive Analysis (Classification)

- Standardise the data using StandardScaler, split the data into training and testing sets.
- Built different machine learning models and tune different hyperparameters using GridSearchCV:
  - Logistic Regression, SVM, Decision Tree, KNN.
- Accuracy is the metric for all models.
- The models are tuned using GridSearchCV to find the parameters giving the best Accuracy.
- The link to the notebook: https://github.com/KietUwU/Data-Science---Coursera-Capstone/blob/main/Hands-on%20Lab%3A%20Complete%20the%20Machine%20Learning%20Prediction%20lab.ipynb

Results

- The models with the most accuracy are Logistic Regression SVM and KNN, all being at 0.833.
- Decision Tree preforms the worst with an accuracy if 0.722 after tuning.

TASK 12

Find the method performs best:

In [36]:
```
Report = {'Accuracy': [Accuracy_LR, Accuracy_SVM, Accuracy_Tree, Accuracy_KNN]}
index = ['Logistic Regression', 'SVM', 'Decision Tree', 'KNN']
df_Report = pd.DataFrame(data=Report, index=index)
```

In [37]:
```
df_Report
```

Out[37]:

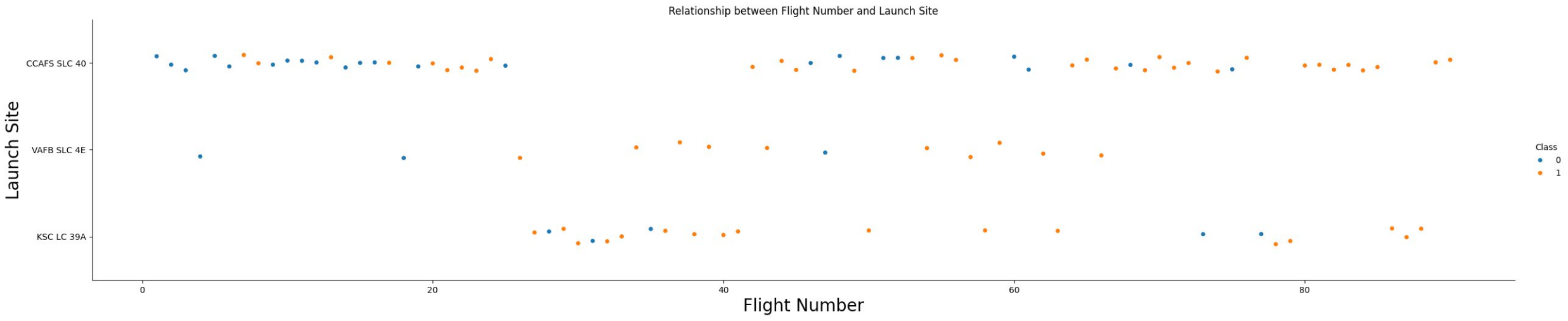
	Accuracy
Logistic Regression	0.833333
SVM	0.833333
Decision Tree	0.722222
KNN	0.833333

Section 2

Insights drawn from EDA

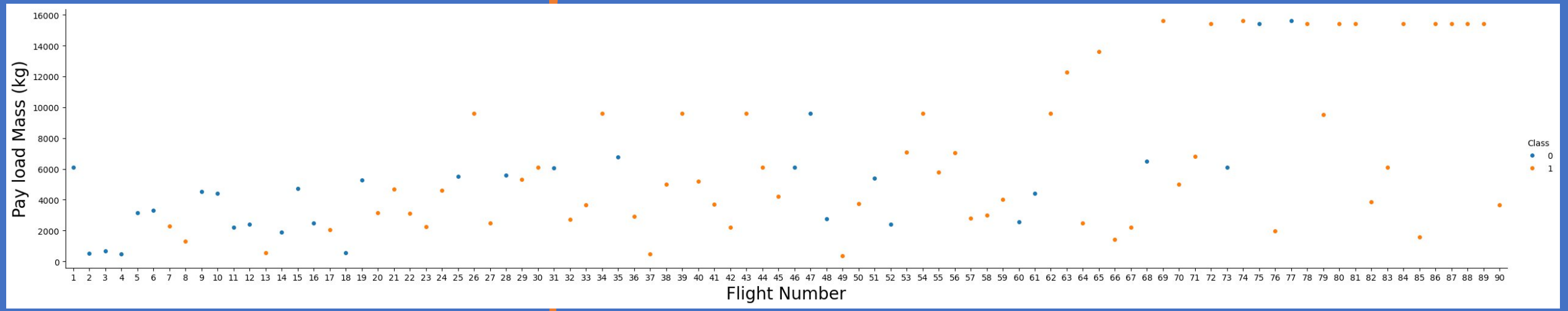
Flight Number vs. Launch Site

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



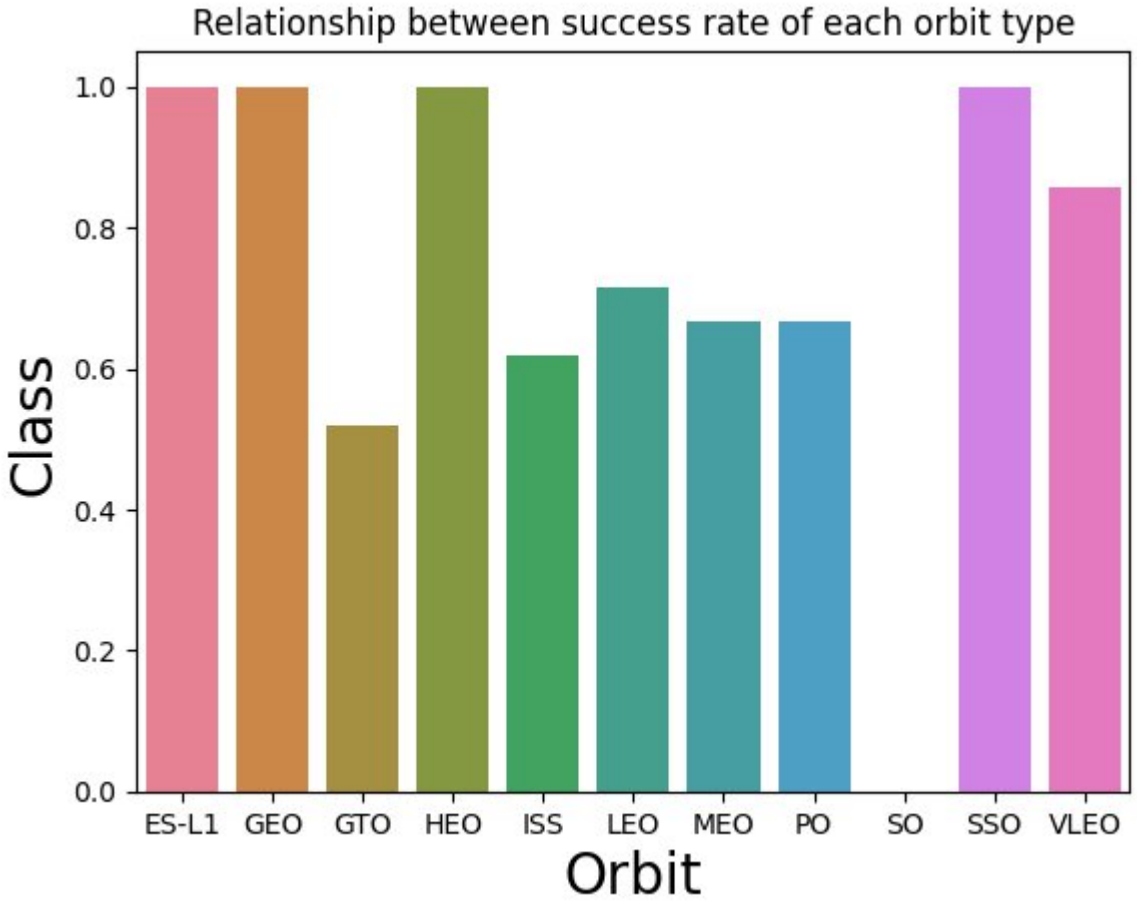
Payload vs. Launch Site

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



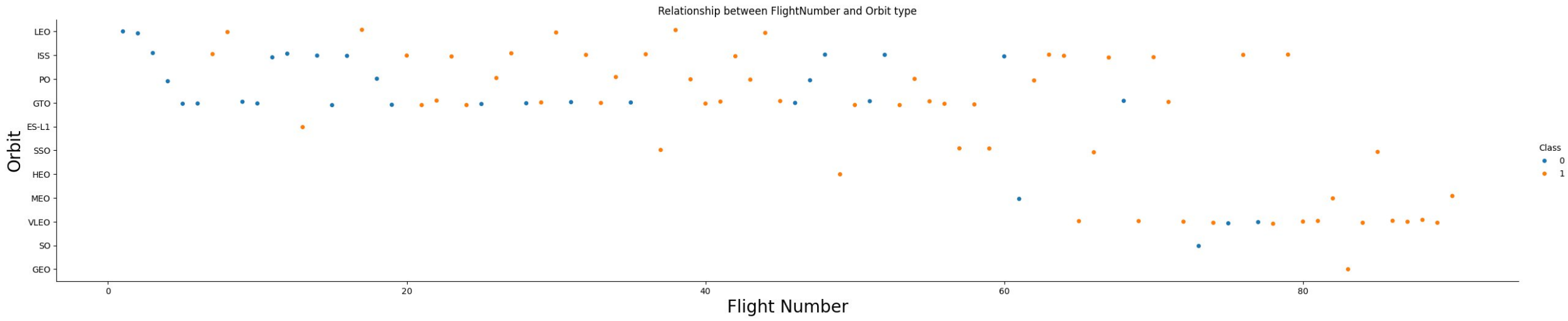
Success Rate vs. Orbit Type

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



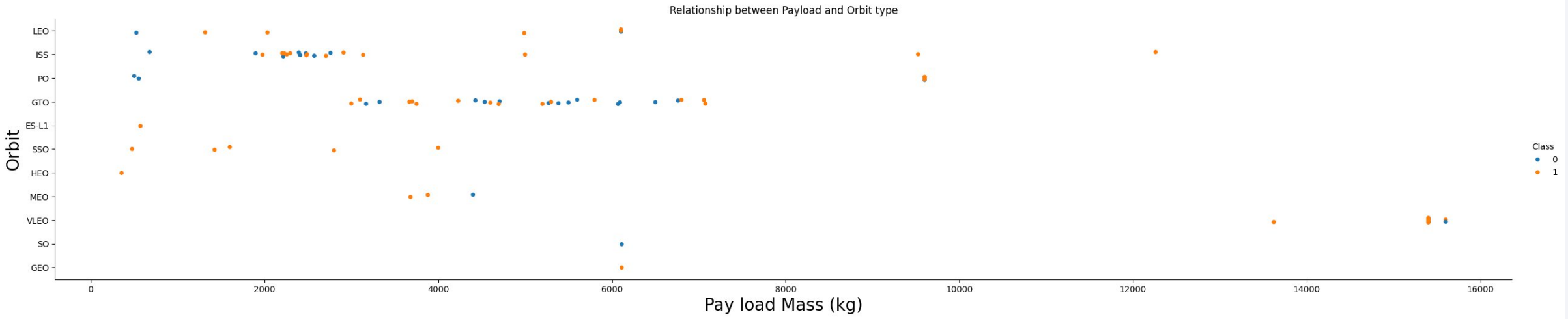
Flight Number vs. Orbit Type

- 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.



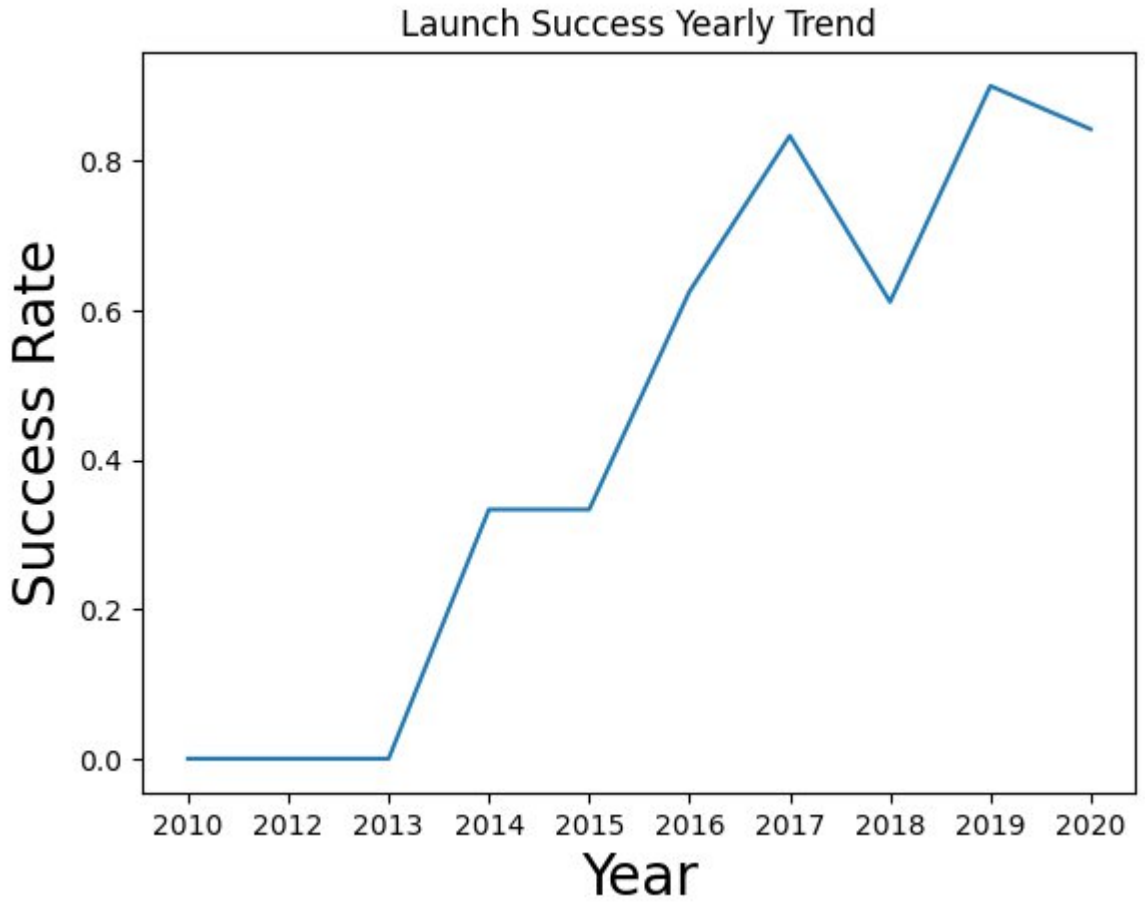
Payload vs. Orbit Type

- 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.



All Launch Site Names

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

Task 1

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

In [9]:
```
%sql select distinct "Launch_Site" from SPACEXTABLE
```

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

Out[9]:

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

Launch Site Names Begin with 'CCA'

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

Task 2

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

In [10]:
```
%sql select * from SPACEXTABLE where "Launch_Site" like 'CCA%' limit 5
```

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

Out[10]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parac
2010-08-12	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 (parac
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No att
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No att
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No att

5
2
5

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 525 using the query below

Task 3

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

In [11]:
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Customer" = 'NASA (COTS)'
```

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

Out[11]:

sum(PAYLOAD_MASS__KG_)
525

Average Payload Mass by F9 v1.1

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

Task 4

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

In [12]:
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Booster_Version" = 'F9 v1.1'
```

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

Out[12]:

avg(PAYLOAD_MASS__KG_)
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

Task 5

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

Hint:Use min function

In [13]:
```
%sql select min("Date") from SPACEXTABLE where "Landing_Outcome" = 'Success (ground pad)'
```

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

Out[13]:

min("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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

In [14]:
```
%sql select "Booster_Version" from SPACEXTABLE where "Landing_Outcome" = 'Success (drone ship)' and PAYLOAD_MASS_
```

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

Out[14]:

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

- 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

Total Number of Successful and Failure Mission Outcomes

- We used wildcard like ‘%’ to filter for **WHERE** MissionOutcome was a success or a failure.

Task 7

List the total number of successful and failure mission outcomes

In [15]:
```
%sql select count(*) from SPACEXTABLE where "Landing_Outcome" like 'Success%' or "Landing_Outcome" like 'Failure%
```

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

Out[15]:

count(*)
71

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.

Task 8

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

In [16]:
```
%sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEX
```

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

Out[16]:

Booster_Version
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

- 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

Task 9

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [17]:
```
%sql select substr("Date", 6,2) as "Month in year 2015", "Landing_Outcome", "Booster_Version", "Launch_Site" from
```

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

Out[17]:

Month in year 2015	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

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

Task 10

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.

In [18]:
```
%sql select "Landing_Outcome", count("Landing_Outcome") from SPACEXTABLE where "Date" between '2010-06-04' and '2
```

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

Out[18]:

Landing_Outcome	count("Landing_Outcome")
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
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 4

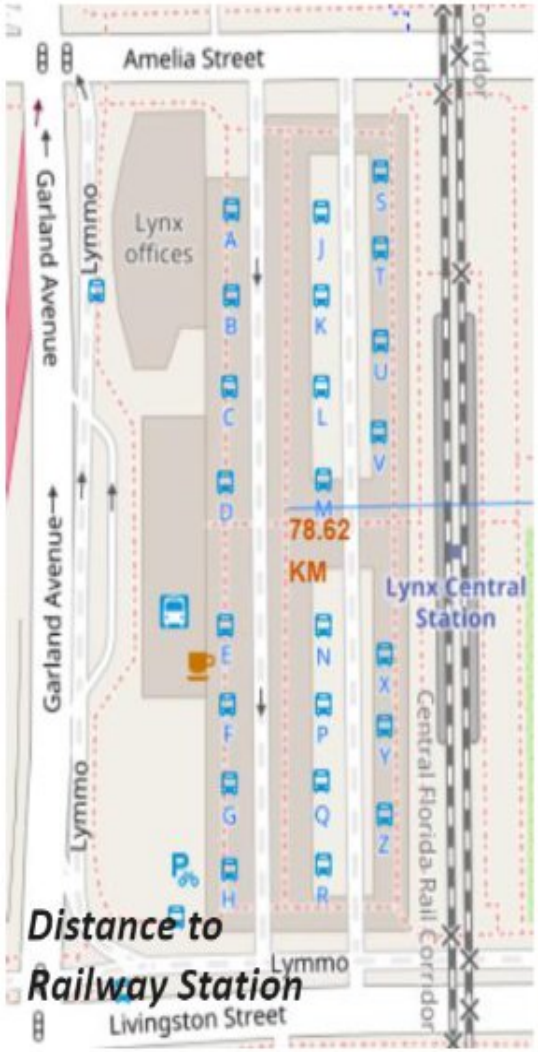
Launch Sites Proximities Analysis

All launch sites global map markers

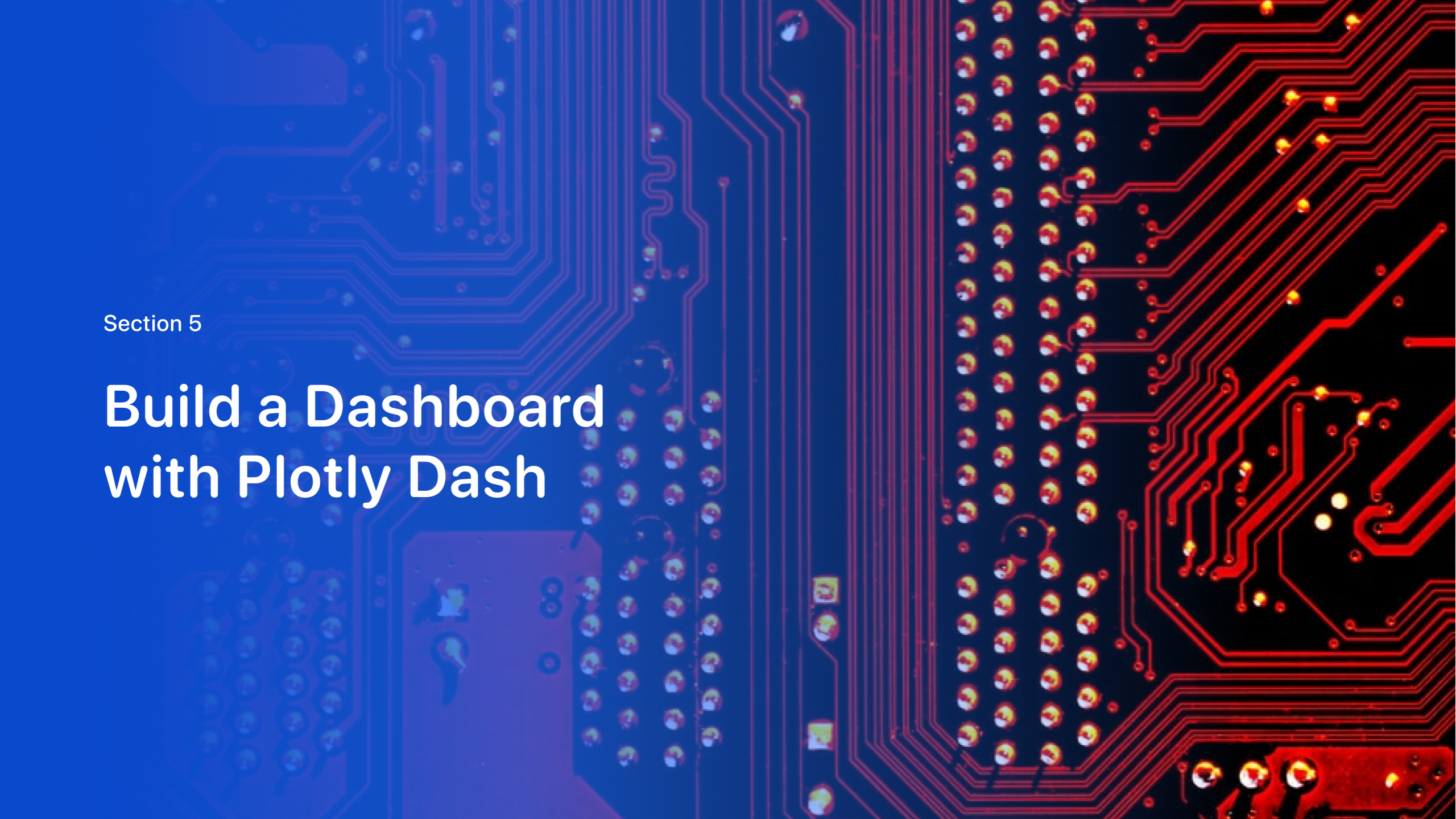
We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

Markers showing launch sites with color labels

Launch Site distance to landmarks



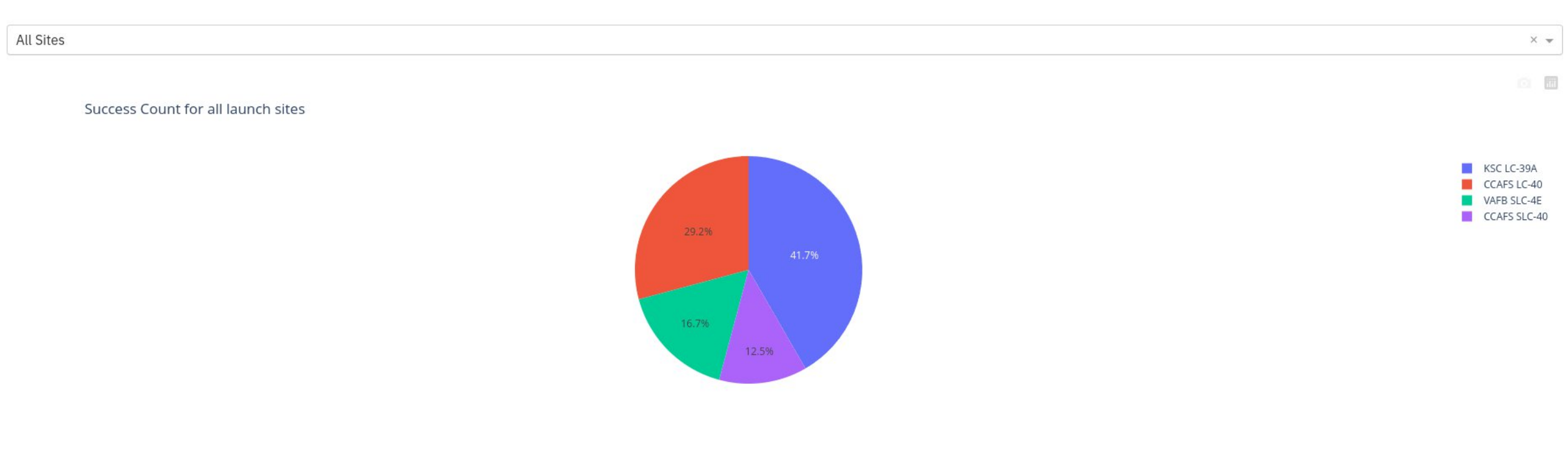
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes



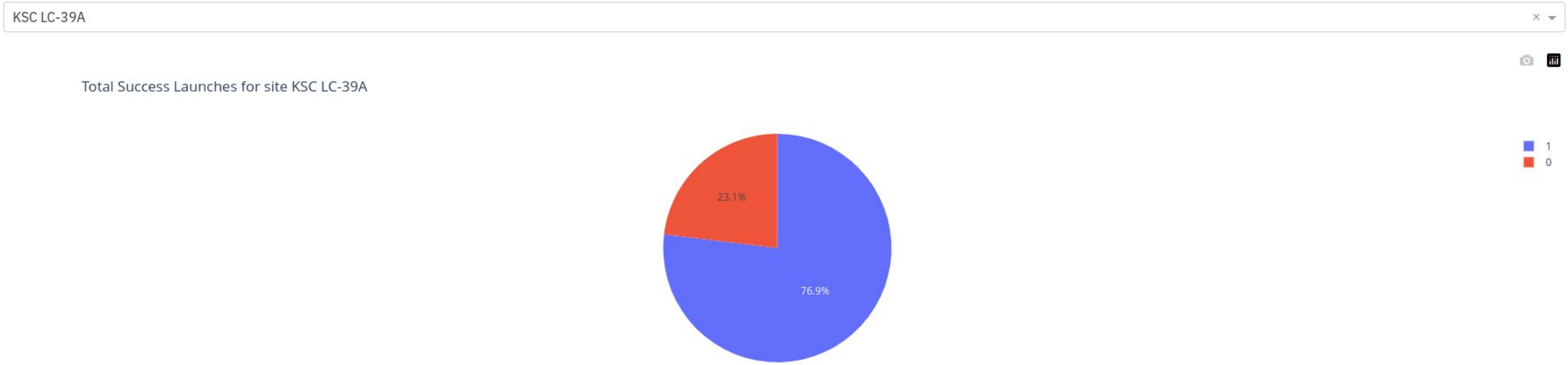
Section 5

Build a Dashboard with Plotly Dash

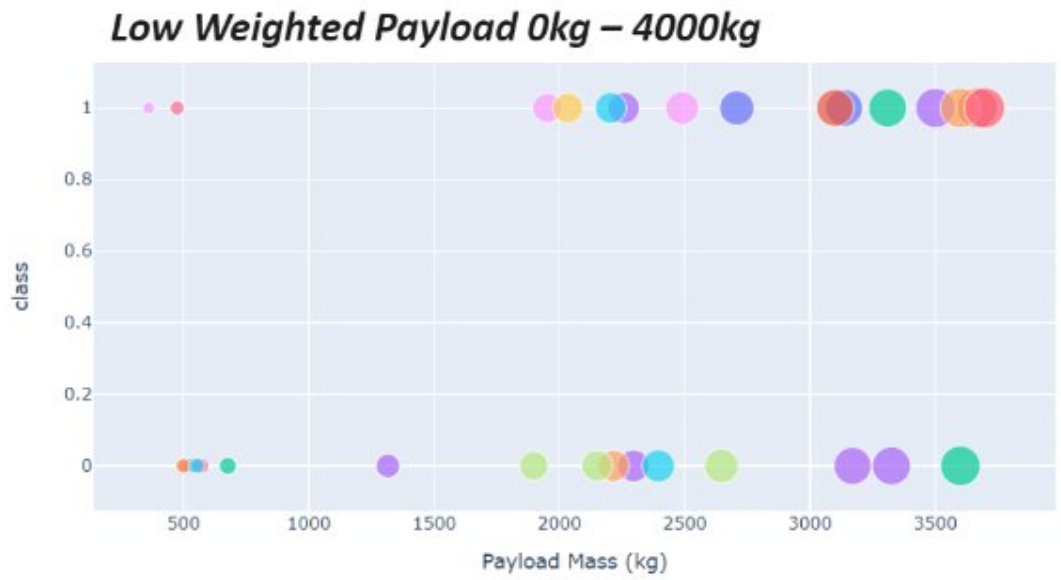
Pie chart showing the success percentage achieved by each launch site

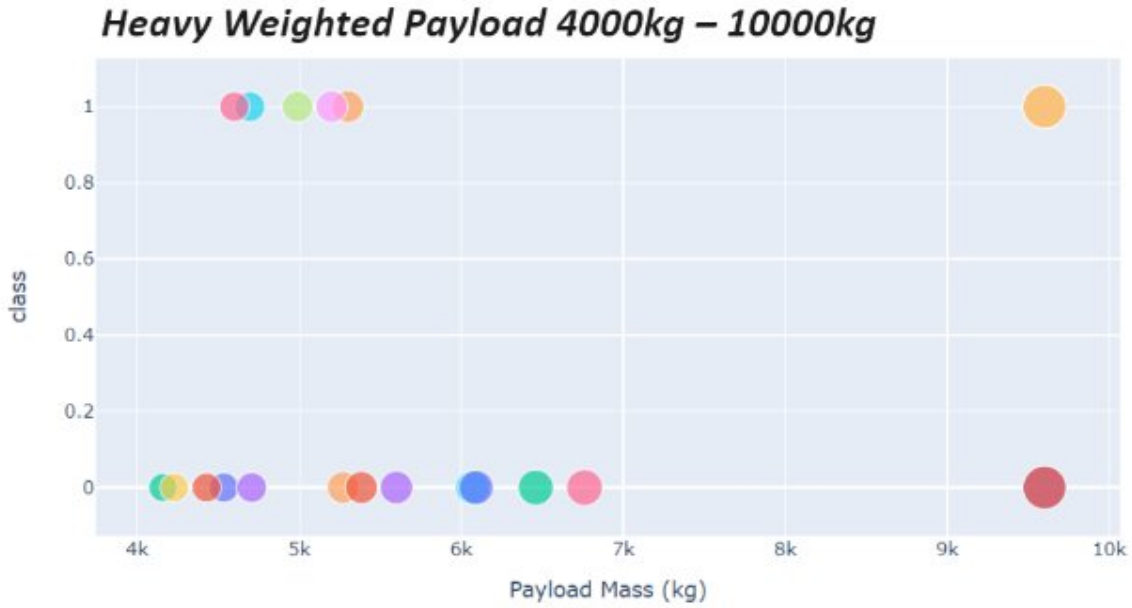


Pie chart showing the Launch site with the highest launch success ratio



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 6

Predictive Analysis (Classification)

Classification Accuracy

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

Out[26]:
```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

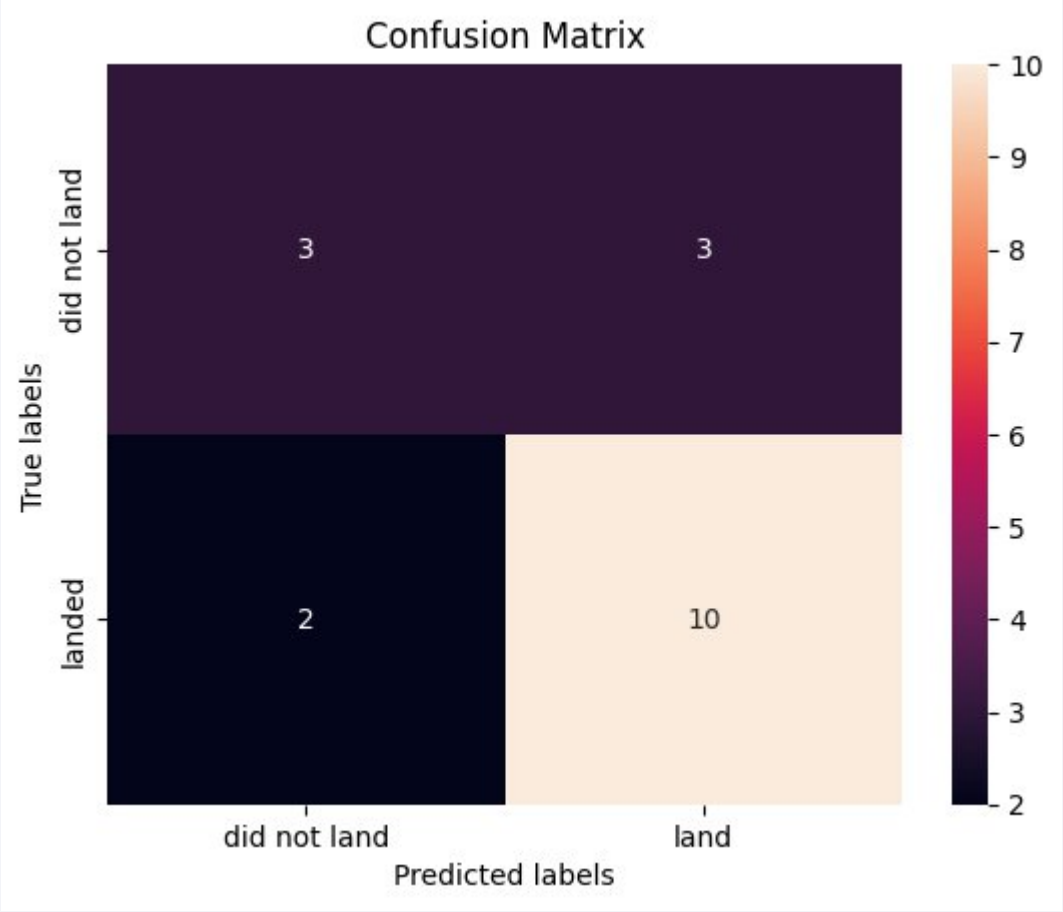
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [27]:
```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'auto', 'min_
samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}
accuracy : 0.8767857142857143
```

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!