SpaceX Falcon 9 Rocket

Pragati Amrutlal Mistry

2 March, 2022

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- 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 Visual Analytics result
  - Predictive Analytics result

INTRODUCTION

- Project Scenario and Overview
  - SpaceX 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 SpaceX 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 SpaceX for a rocket launch.
- Setting up the Problems
  - What are the factors that determine if the rocket will land successfully?
  - The interaction between the various features that determine the success rate of successful landing
  - Operating conditions to ensure a successful landing.

METHODOLOGY

Executive Summary

- Data Collection methodology:
  - Data was collected from SpaceX REST API and web scraping from Wikipedia
- Data Wrangling:
  - One-hot encoding was applied to categorical features
- Exploratory Data Analysis (EDA) using SQL and Visualization
- Interactive visualization analytics using Folium and Plotly Dash
- Predictive analysis using classification models
  - How to build, tune and evaluate classification models

Data Collection Methodology

The data was collected through various methods:

- We will perform a get request to the SpaceX API using the requests library to obtain the launch data.
- Next, we convert the response content JSON to a pandas dataframe using json_normalize function.
- Then, we cleaned the data, checked for missing values and fill in missing values where necessary.
- We then used Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records.
- The objective was to extract the launch records as HTML table, parse the table and convert it to pandas dataframe for future analysis.

Data Collection SpaceX API

- We used the get request to the SpaceX API, cleaned the requested data and perform some data wrangling and formatting.
- This is the link to the notebook

[testrepo/API.ipynb at main · Pragati2411/testrepo (github.com)]()

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [7]: response = requests.get(spacex_url)
```

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

```
In [11]: # Use json_normalize meethod to convert the json result into a dataframe
         data = pd.json_normalize(response.json())
```

Calculate below the mean for the PayloadMass using the .mean(). Then use the mean and the .replace() function to replace np.nan values in the data with the mean you calculated.

```
In [32]: # Calculate the mean value of PayloadMass column
         Mean_PyaloadMass = data_falcon9.PayloadMass.mean()
         # Replace the np.nan values with its mean value
         data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan)
```

You should see the number of missing values of the PayLoadMass change to zero.

```
In [34]: data_falcon9.isnull().sum()
```

Data Collection- Web Scraping

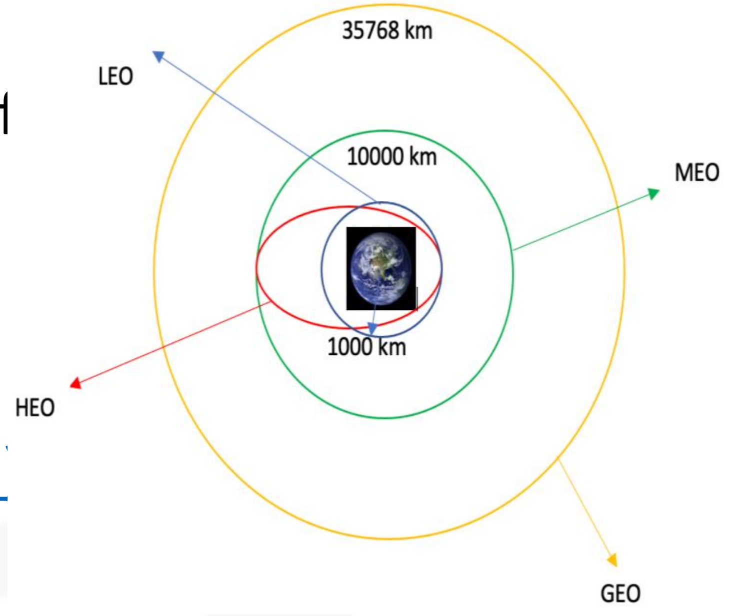
- We used BeautifulSoup object for web scraping Falcon 9 Launch records.
- We created a data frame by parsing the launch HTML tables.
- The link to the notebook is

[testrepo/data wrangling.ipynb at master · Pragati2411/testrepo (github.com)]()

Data Wrangling

- We performed Exploratory Data Analysis and determined the training labels.
- We calculated the number of launch on each sites, the number and occurrence of each orbit and the number and occurrence of mission outcome per orbit type.
- We also created a landing outcome label from Outcome column
- This is link to the notebook

[testrepo/EDA.ipynb at master · Pragati2411/testrepo (github.com)]



EDA with SQL

- We loaded the SpaceX dataset into the corresponding table in Db2 database.
- We applied EDA with SQL to get insight from the data and we executed SQL queries to answer the following:
  a. The names of the unique launch sites in the space
  b. The total payload mass carried by boosters launched by NASA(CRS)
  c. The average payload mass carried by booster version F9 v1.1
  d. The total number of successful and failure mission outcomes
  e. The failed landing outcomes in drone ship, their booster version and launch site names

- The link to the notebook is

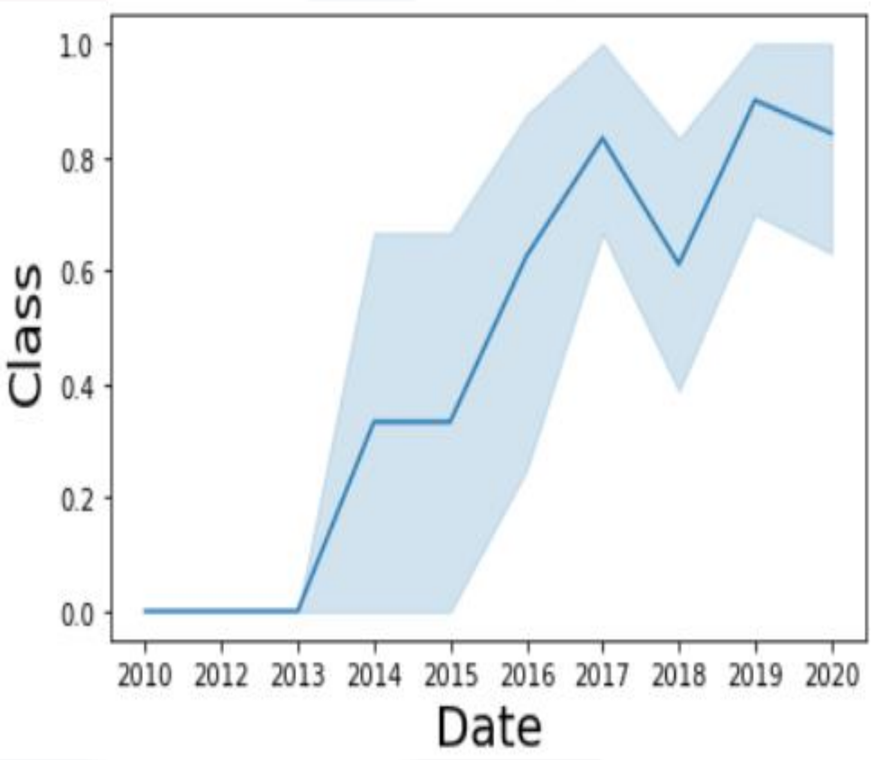
testrepo/EDA with SQL.ipynb at master · Pragati2411/testrepo (github.com)

EDA with Data Visualization

- We visualized the relationship between flight number and launch sites, payload and launch site, success rate of each orbit type, flight number and orbit type, payload and orbit type , the launch success yearly trend.

- This is the link of notebook

testrepo/EDA with Data Visualization.ipynb at master · Pragati2411/testrepo (github.com)



Interactive Data visualization with Folium

- We marked all launch sites, added map objects like markers, circles for each site, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes to class 0 (failure) and 1 (success).
- We used color-labeled marker clusters to identify which launch sites have relatively high success rate.
- The link to the notebook is

[testrepo/Interactive Visual Analytics.ipynb at master · Pragati2411/testrepo (github.com)]

Interactive Data Visualization with Plotly Dash

- We built an interactive dashboard with Plotly Dash
- We also built pie charts showing the total launches by certain sites
- We built a scatter graph showing the relationship between Outcome and Payload Mass(kg) for different booster version.
- The link to the notebook is

testrepo/plotly dash at master · Pragati2411/testrepo (github.com)

Predictive Analysis(Classification)

- We loaded the data using numpy and pandas, transformed the data, split it using test_train_split function.
- 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.
- We obtained the best perfoming classification model.
- The link to the notebook is

[testrepo/Machine Learning Prediction.ipynb at master · Pragati2411/testrepo (github.com)](https://github.com/Pragati2411/testrepo)

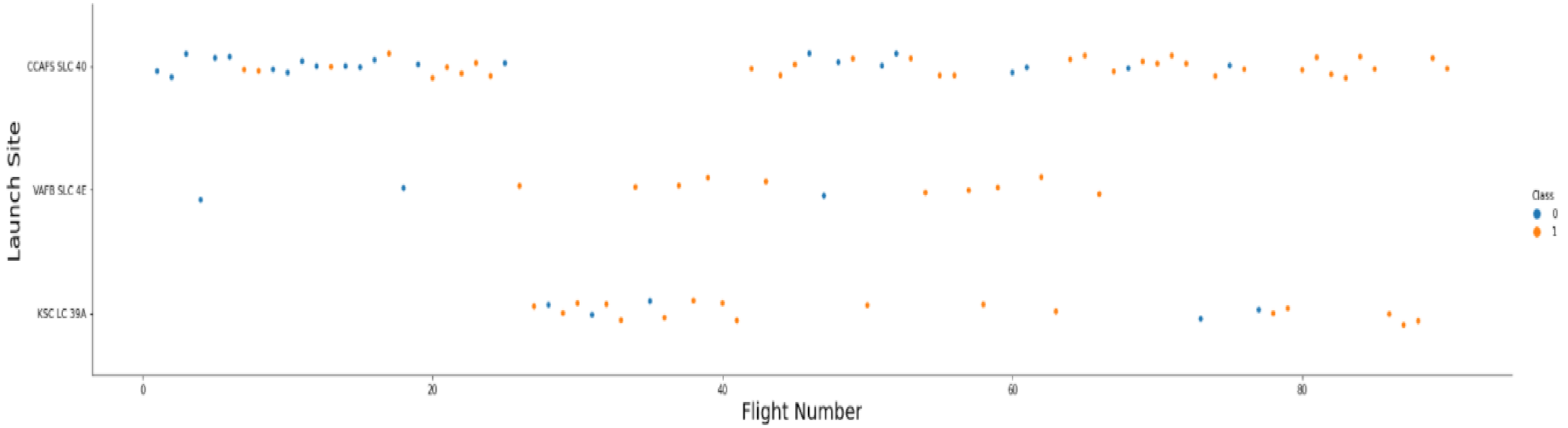
RESULTS

- Exploratory Data Analysis results
- Interactive Analytics results
- Predictive Analytics results

Exploratory Data Analysis Results

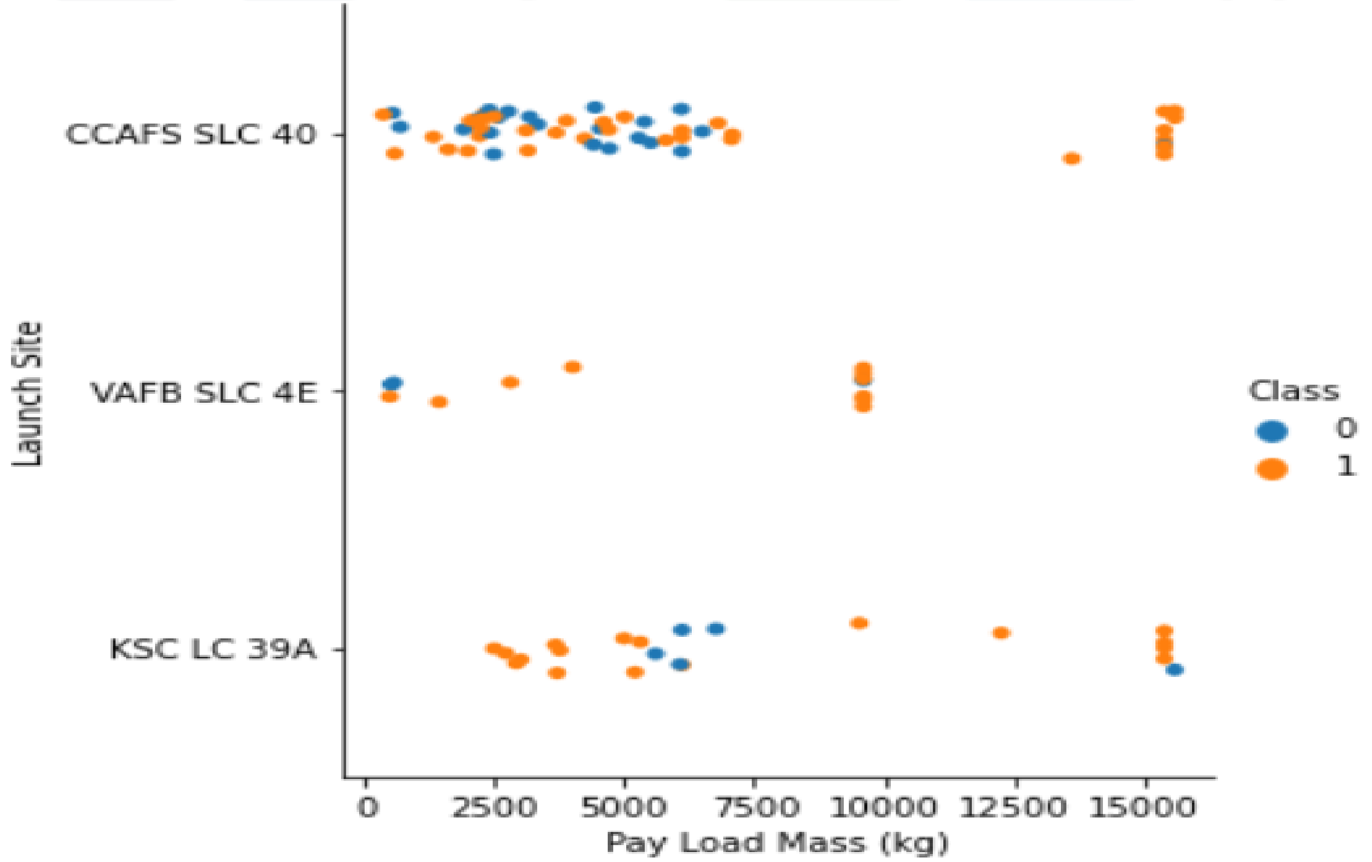
Flight number vs. Launch Site

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



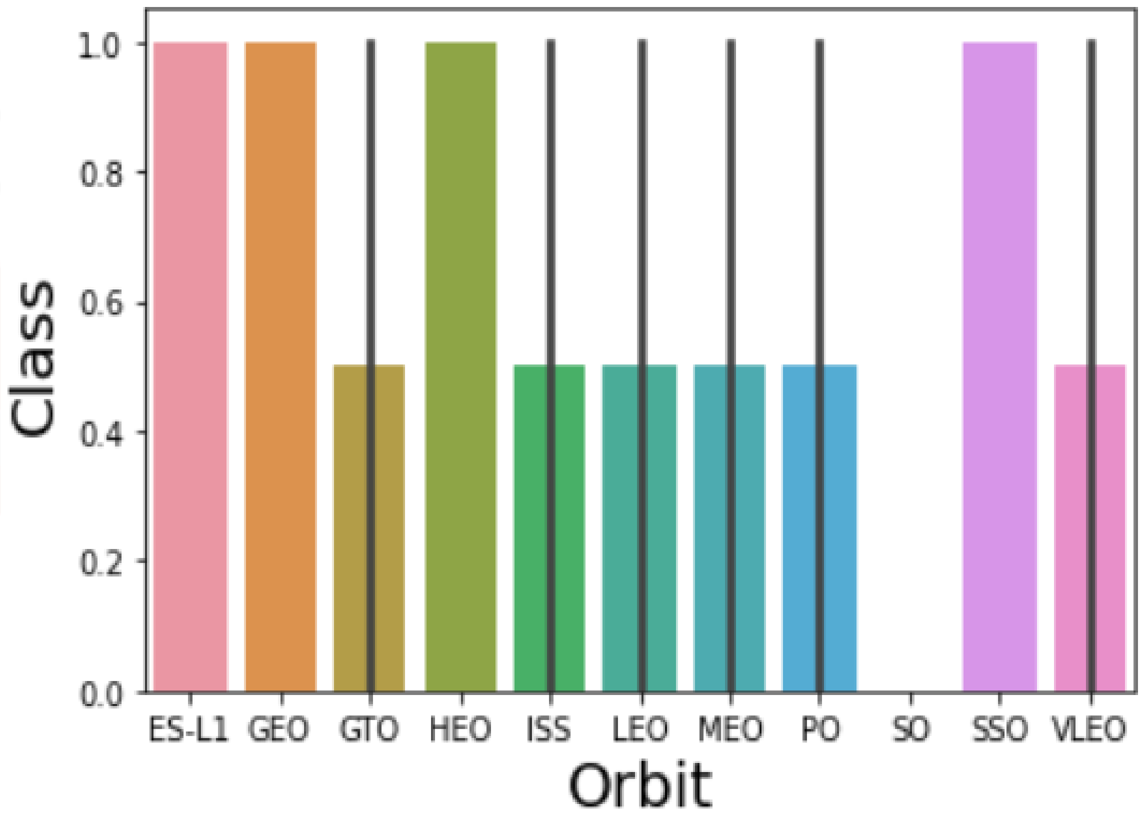
Payload vs. Launch Site

- The greater the Payload Mass for launch site CCAFS SLC 40, the higher the success rate for the rocket.



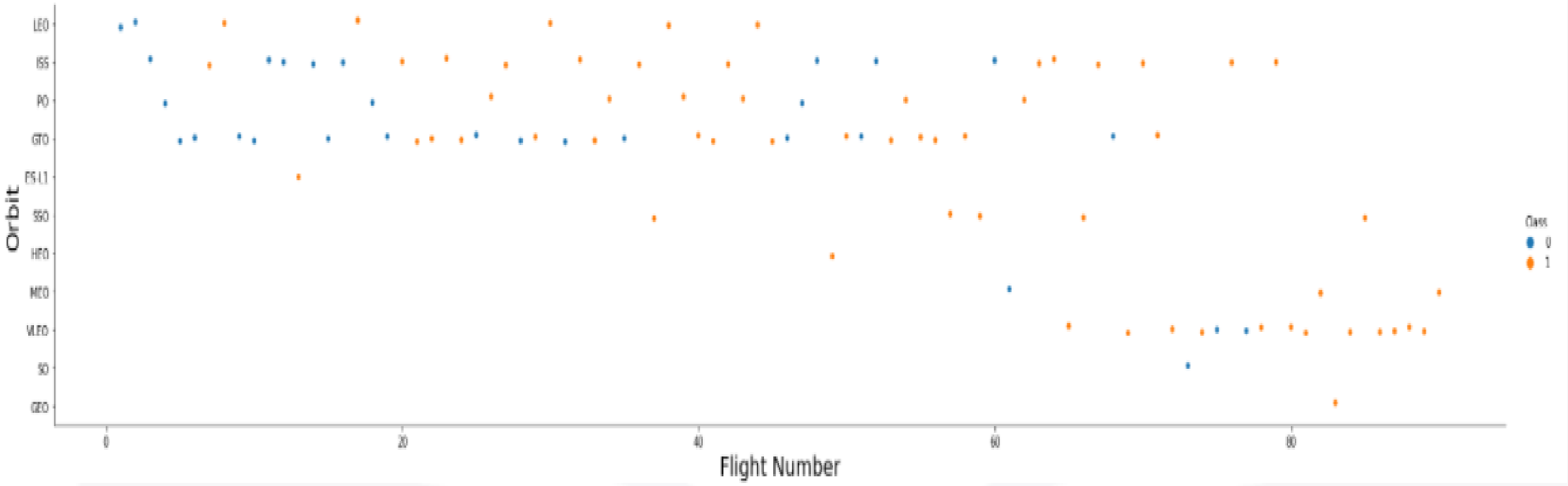
Success Rate vs. Orbit Type

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



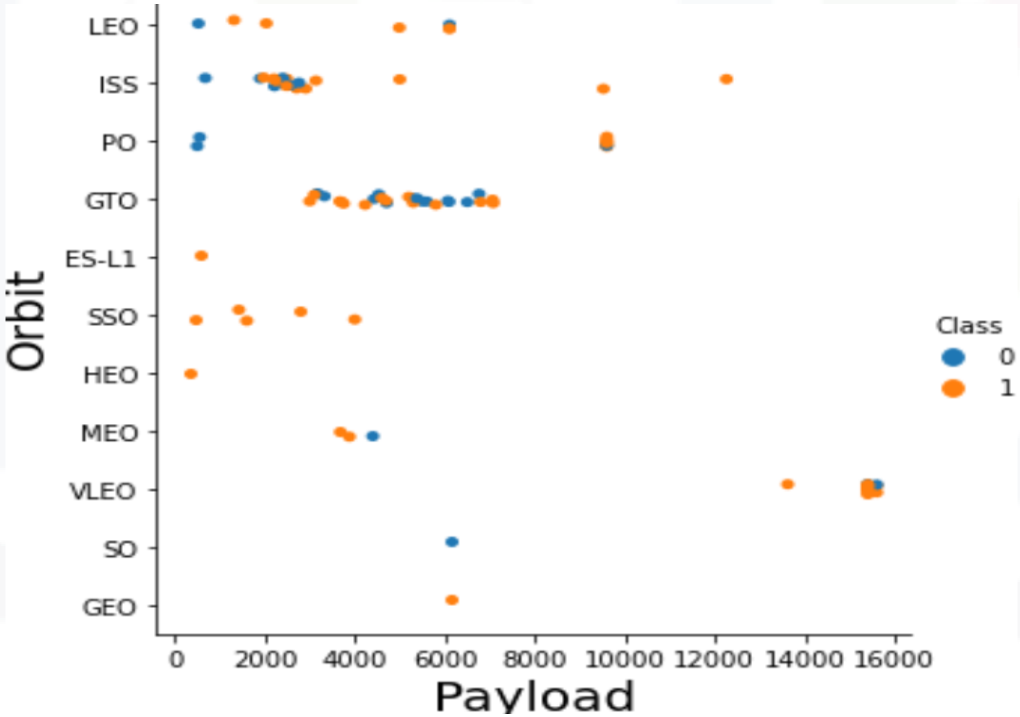
Flight Number vs. Orbit Type

- We can say 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 orbit.



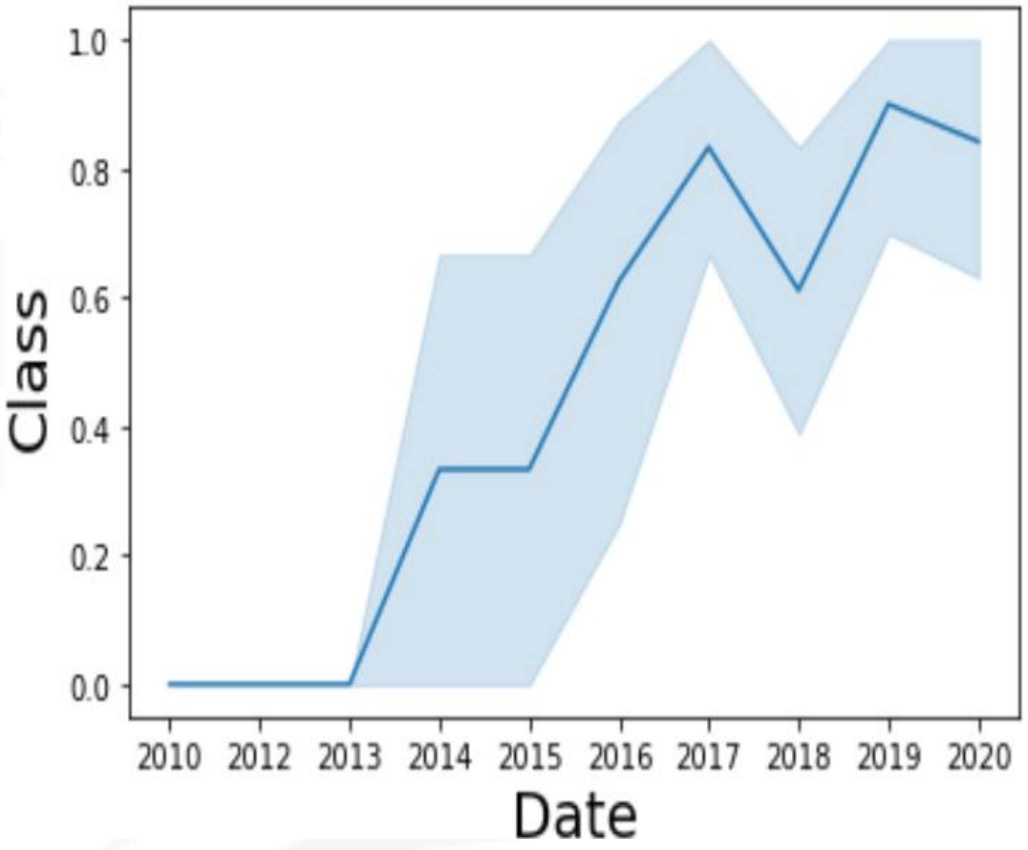
Payload vs. Orbit Type

- 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 say that success rate since 2013 kept on increasing till 2020.



Unique Launch Sites Names

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

In [7]:
```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
```

```
 * ibm_db_sa://mwc64446:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[7]:

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

Launch Sites begin with 'CCA'

We used this query to display 5 records of launch sites begin with 'CCA'.

```
In [8]: sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;

 * ibm_db_sa://mwc64446:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[8]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
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
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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

In [11]:
```
%sql select sum(payload_mass__kg_) from SPACEXTBL WHERE customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://mwc64446:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[11]:

1
45596

Average Payload Mass by F9 v1.1

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

In [13]:
```
%sql select avg(payload_mass__kg_) from SPACEXTBL WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://mwc64446:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[13]:

1
2928

First Successful Ground Landing Date

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

Hint:Use min function

In [15]:
```
%sql select min(date) from SPACEXTBL WHERE landing__outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://mwc64446:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[15]:

1
2015-12-22

Successful Drone Ship Landing

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

In [17]:
```
%sql select booster_version from SPACEXTBL where landing__outcome = 'Success (drone ship)'\
and payload_mass__kg_ between 4000 and 6000
```

```
 * ibm_db_sa://mwc64446:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[17]:

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

Total Number of Successful and Failure Outcomes

List the total number of successful and failure mission outcomes

In [18]:
```
%sql select mission_outcome, count(mission_outcome) from SPACEXTBL GROUP BY mission_outcome
```

```
 * ibm_db_sa://mwc64446:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdom
ain.cloud:30376/bludb
Done.
```

Out[18]:

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

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

In [19]:
```
%sql select booster_version, payload_mass__kg_ from SPACEXTBL\
where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)
```

```
 * ibm_db_sa://mwc64446:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[19]:

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

Launch Records for Year 2015

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

In [24]:
```
%sql select booster_version, launch_site from SPACEXTBL where landing__outcome = 'Failure (drone ship)' and year(DATE)
```

```
 * ibm_db_sa://mwc64446:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[24]:

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40
F9 v1.1 B1017	VAFB SLC-4E
F9 FT B1020	CCAFS LC-40
F9 FT B1024	CCAFS LC-40

Rank Landing Outcomes

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 [25]:
```
%sql select count(landing__outcome), landing__outcome from SPACEXTBL \
where DATE between '2010-06-04' and '2017-03-20' group by landing__outcome\
order by count(landing__outcome) desc
```

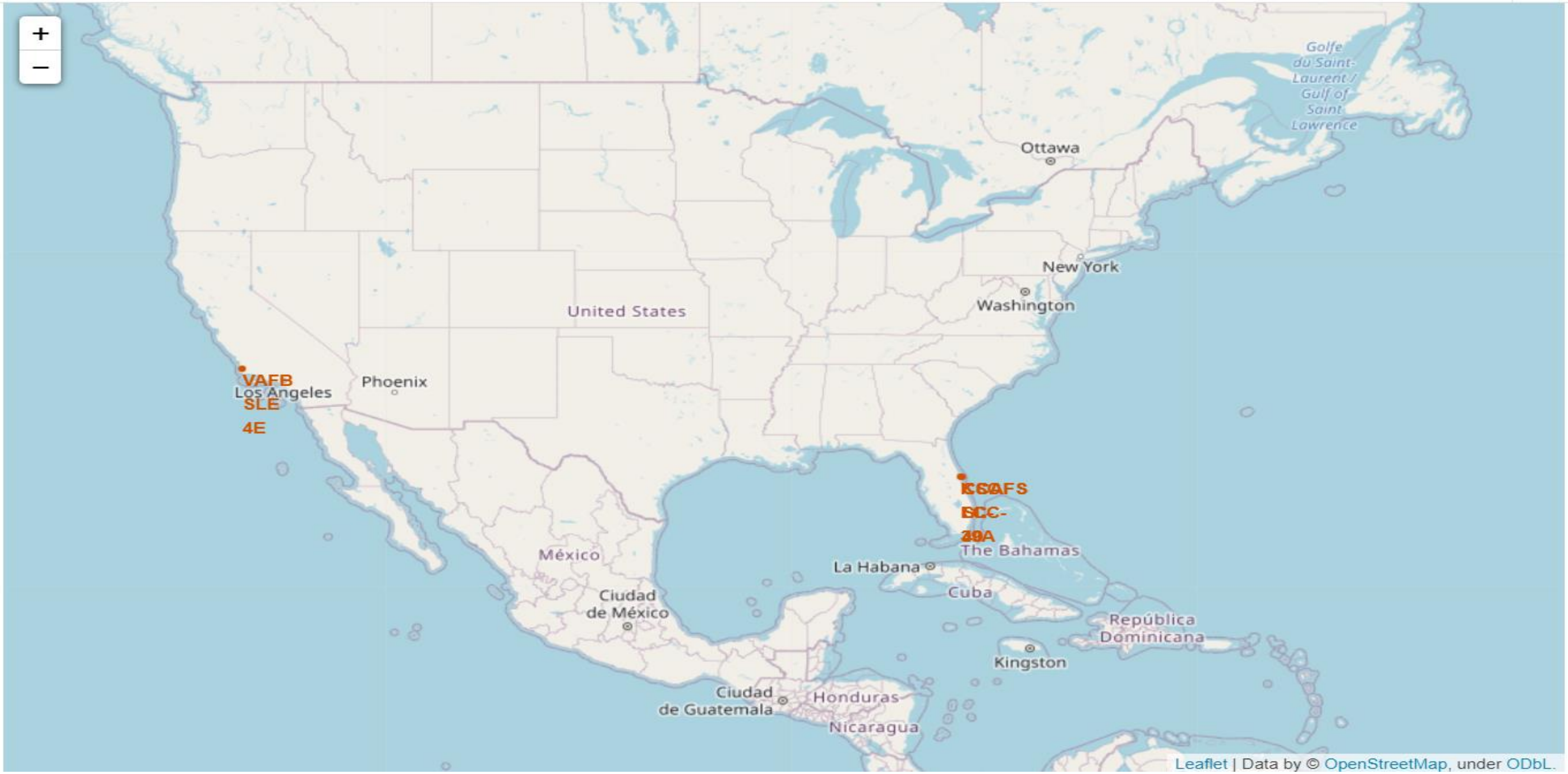
```
 * ibm_db_sa://mwc64446:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[25]:

1	landing__outcome
10	No attempt
5	Failure (drone ship)
5	Success (drone ship)
3	Controlled (ocean)
3	Success (ground pad)
2	Failure (parachute)
2	Uncontrolled (ocean)
1	Precluded (drone ship)

Launch Sites Locations Analysis with Folium

All launch sites global map markers

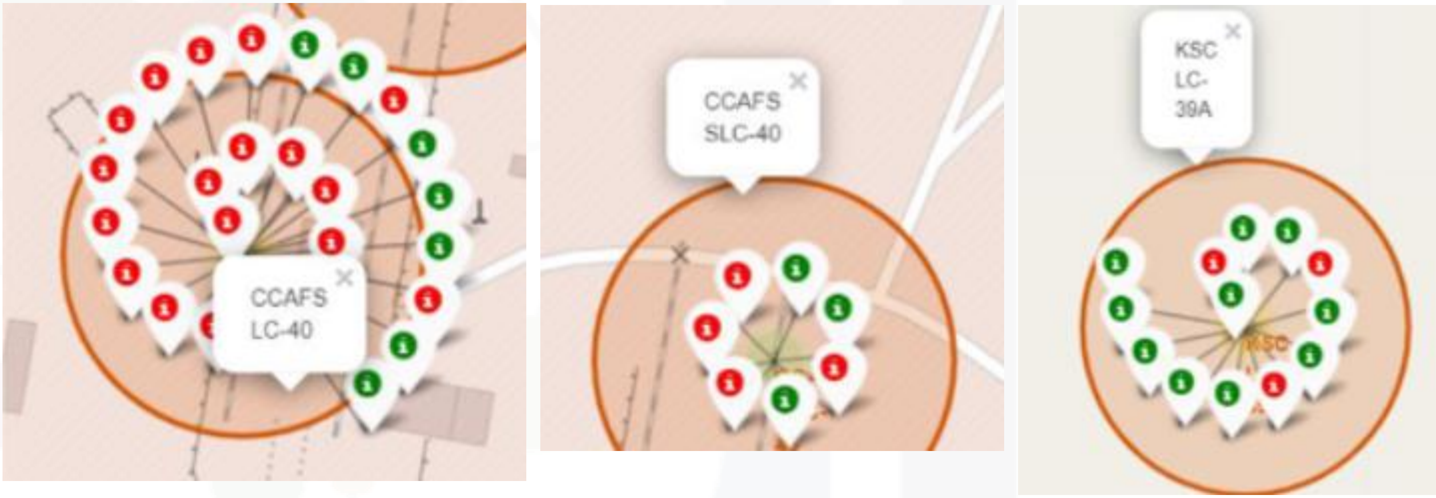


We can see that SpaceX launch sites are located in USA coasts Florida and California

Marker showing launch sites with color labels

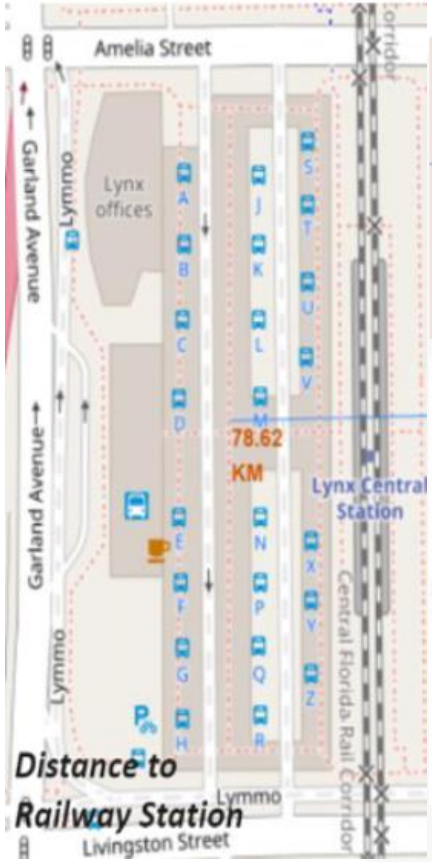
California Launch Site

Florida Launch Sites



Green marker shows successful launches and Red marker shows Failure

Launch Sites Distance to Landmarks



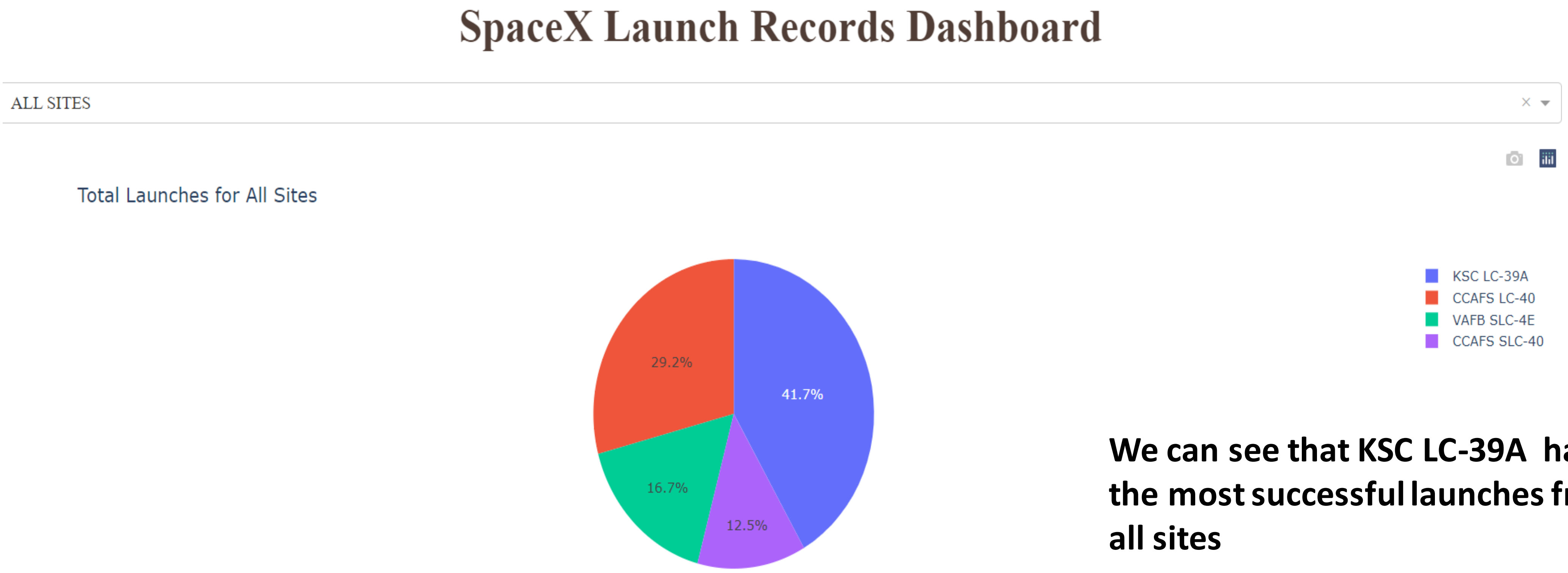
1. Are launch sites in close proximity to railways? No
2. Are launch sites in close proximity to highways? No
3. Are launch sites in close proximity to coastlin? Yes
4. Do launch sites keep a certain distance away from cities? Yes

Build A Dashboard with Plotly Dash

IBM Developer

SKILLS NETWORK

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 sites

Scatter Plot of Payload vs Launch Outcome for all sites with different payload selected in range slider

We can see that the success rates for low weighted payloads is higher than in high weighted payloads

Predictive Analysis

Results

Classification Accuracy and Confusion Matrix

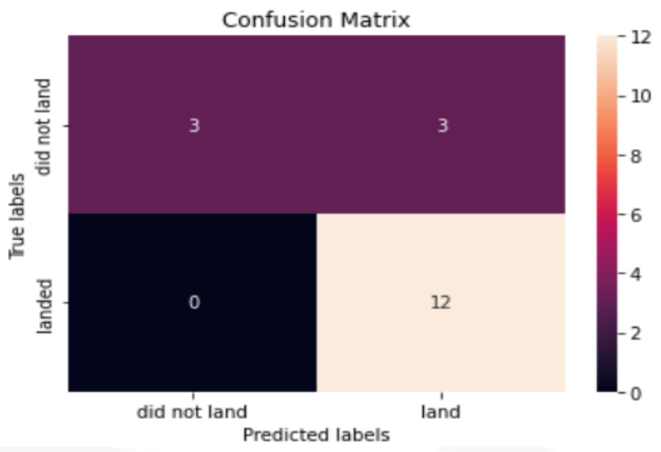
Calculate the accuracy of tree_cv on the test data using the method `score`:

In [24]:
```
tree_cv.score(X_test, Y_test)
```

Out[24]: `0.8333333333333334`

We can plot the confusion matrix

In [25]:
```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



The Decision Tree Classifier is the model with the highest classification accuracy

CONCLUSION

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started increasing in 2013 till 2020.
- Orbits ES-L1, GEO, HEO,SSO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- Then Decision Tree Classifier is the best machine learning algorithm for this task.