IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

EMRE CHATZI SERIF
15 FEBRUARY 2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

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 build 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
  - Clean the dataset
  - Deal with missing values
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Try different models in order to find the best model with the highest accuracy for prediction

Data Collection

- The data was collected using two methods: SpaceX API and Web Scraping

SpaceX API Method:

- 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.

Web Scraping Method:

- 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 Method

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook is:

https://github.com/chatziserif/SpaceX_Capstone_Project_IBM/blob/main/1-SpaceX_API.ipynb

SPACEX API
1. GET REQUEST TO THE SPACEX API
2. CONVERT THE JSON RESULT INTO A DATAFRAME
3. FILTER DATAFRAME TO ONLY FALCON 9 LANCHES AND DATA WRANGLING
4. EXPORT TO CSV

Data Collection – Web Scraping Method

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The Git hub link is:

https://github.com/chatziserif/SpaceX_Capstone_Project_IBM/blob/main/2-SpaceX_WebScraping.ipynb

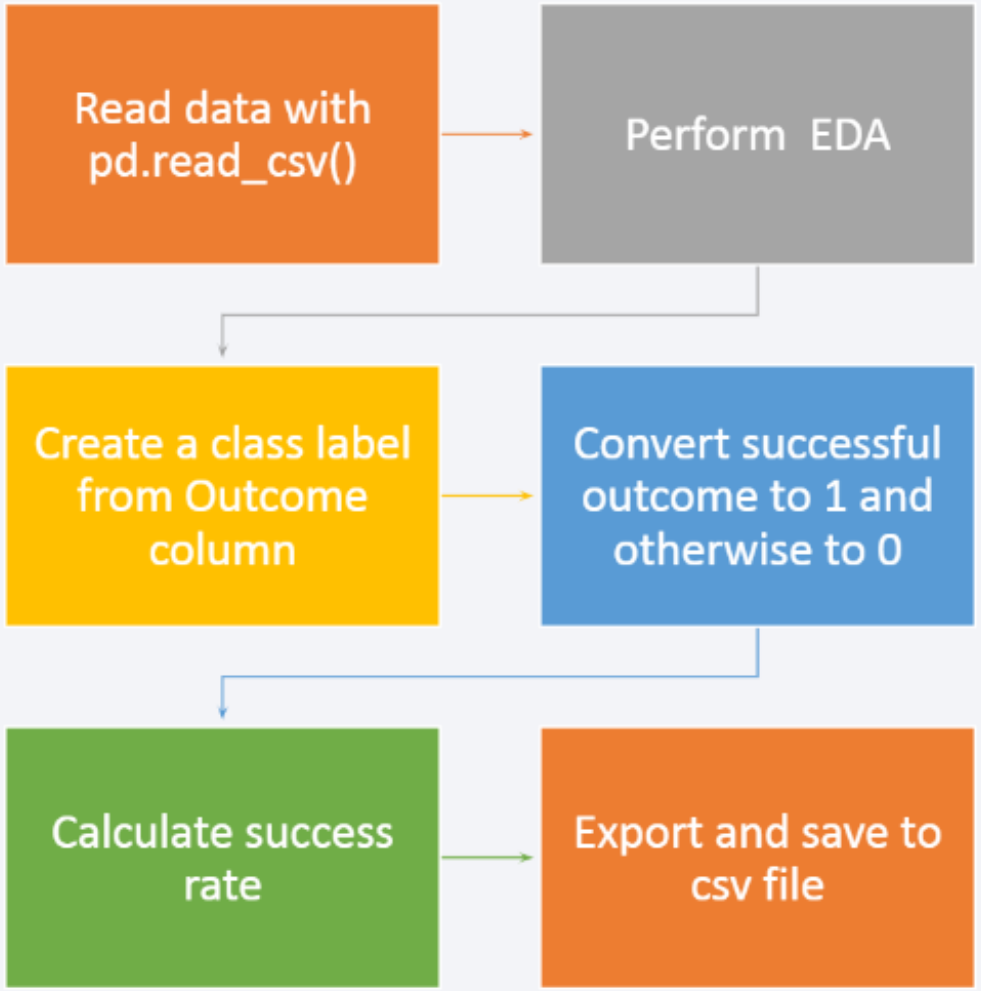
WEB SCRAPING
1.REQUEST THE FALCON 9 WIKIPEDIA PAGE FROM ITS URL
2. EXTRACT ALL COLUMN/VARIABLE NAMES FROM THE HTML TABLE HEADER
3. GENERATE A DATAFRAME BY PARSING THE LAUNCH HTML TABLES
4. EXPORT TO CSV

Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We generated landing outcome label from outcome column and exported the results to csv.

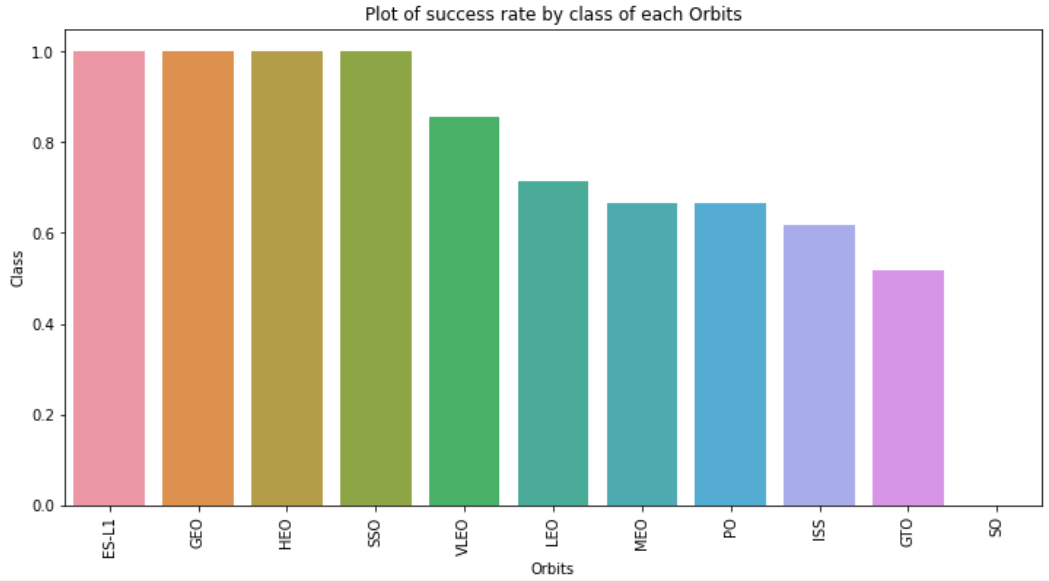
- Github link:

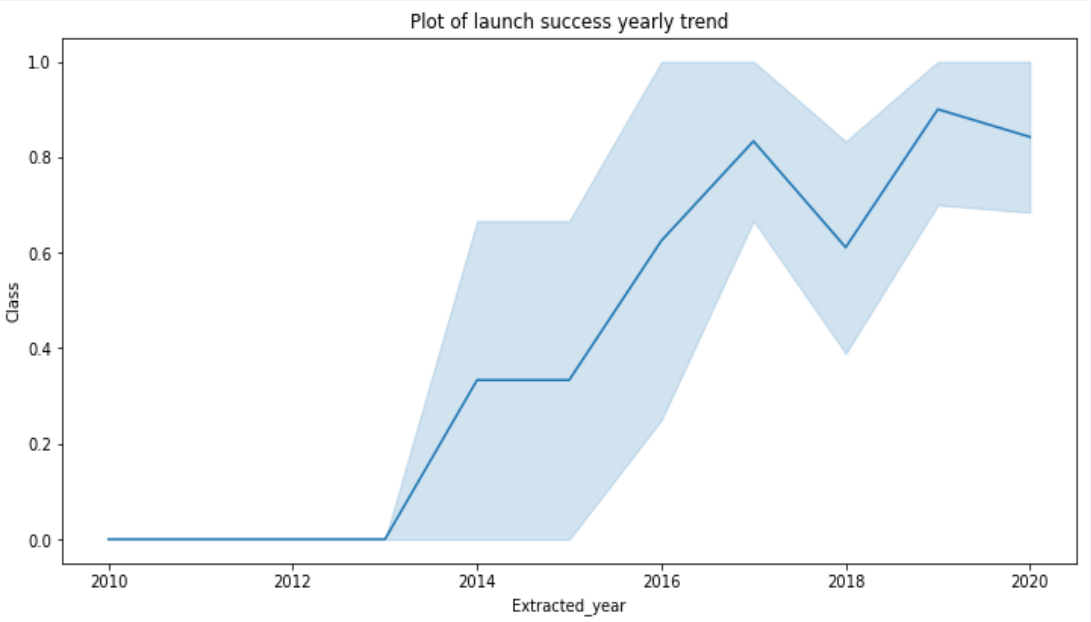
https://github.com/chatziserif/SpaceX_Capstone_Project_IBM/blob/main/3-SpaceX_DataWrangling.ipynb



Exploratory Data Analysis(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.





- The link to the notebook is

https://github.com/chatziserif/SpaceX_Capstone_Project_IBM/blob/main/5-SpaceX_Exploring_PreparingData.ipynb

Exploratory Data Analysis(EDA) with SQL

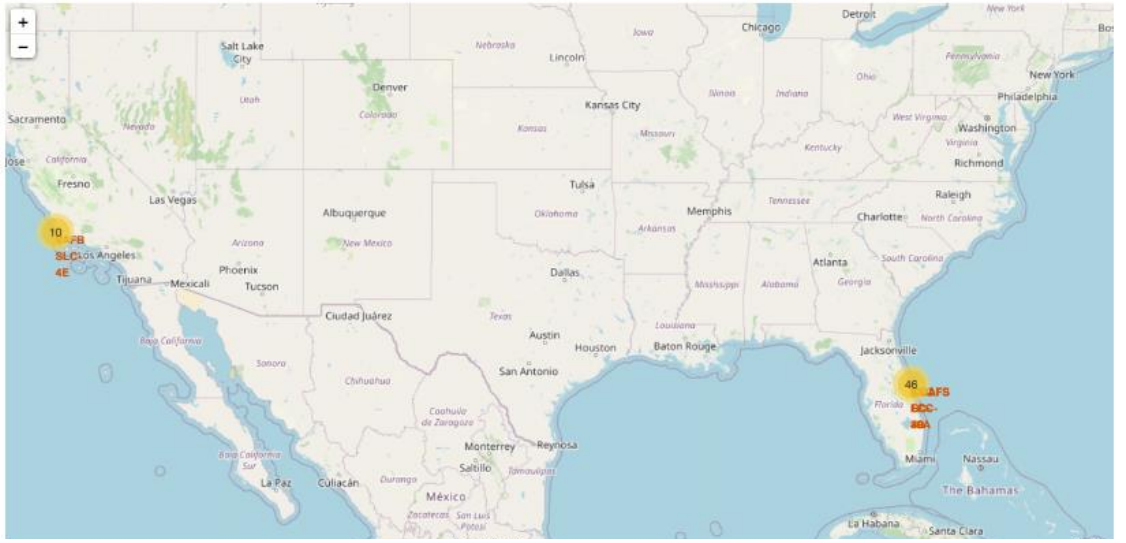
- Exploratroy Data Analysis conducted usign SQL involved the following:
  - Display unique launch sites in the space mission
  - Explore launch sites that begin with 'CCA'
  - Display the total payload mass carried by boosters launched by NASA(CRS)
  - Display average payload mass carried by booster version F9 v.1.1
  - Explore the first successful landing outcome in ground pad
  - Display the successful boosters in drone ship and have a payload between 4000kg and 6000kd
  - Display the total number of successful and failed mission outcomes
- The link to the notebook is:

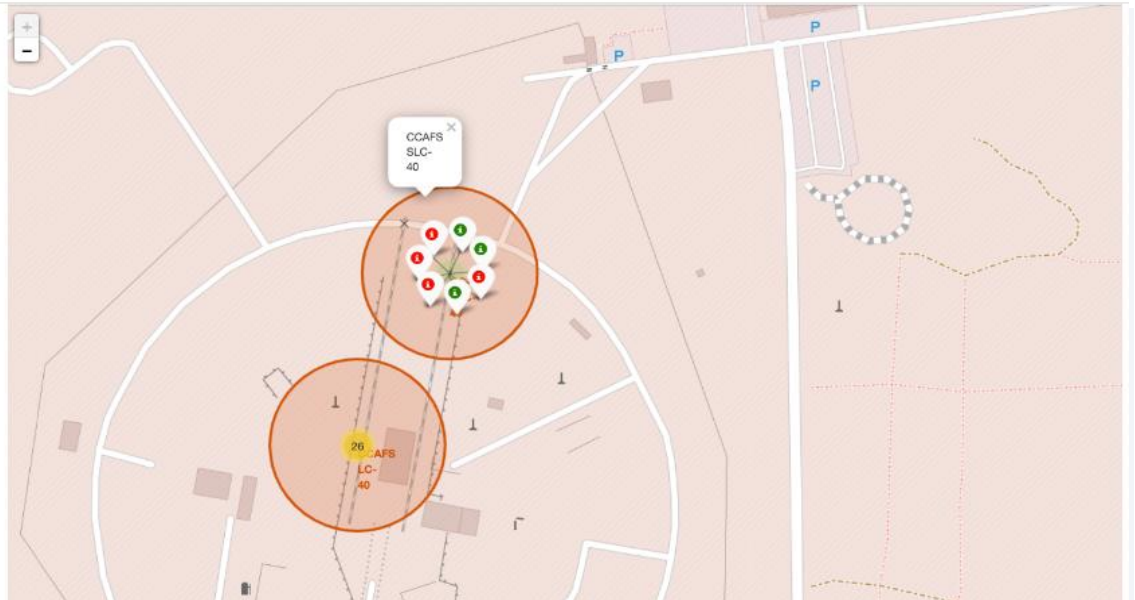
https://github.com/chatziserif/SpaceX_Capstone_Project_IBM/blob/main/4-SpaceX_SQL.ipynb

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.
- The Git hub link:

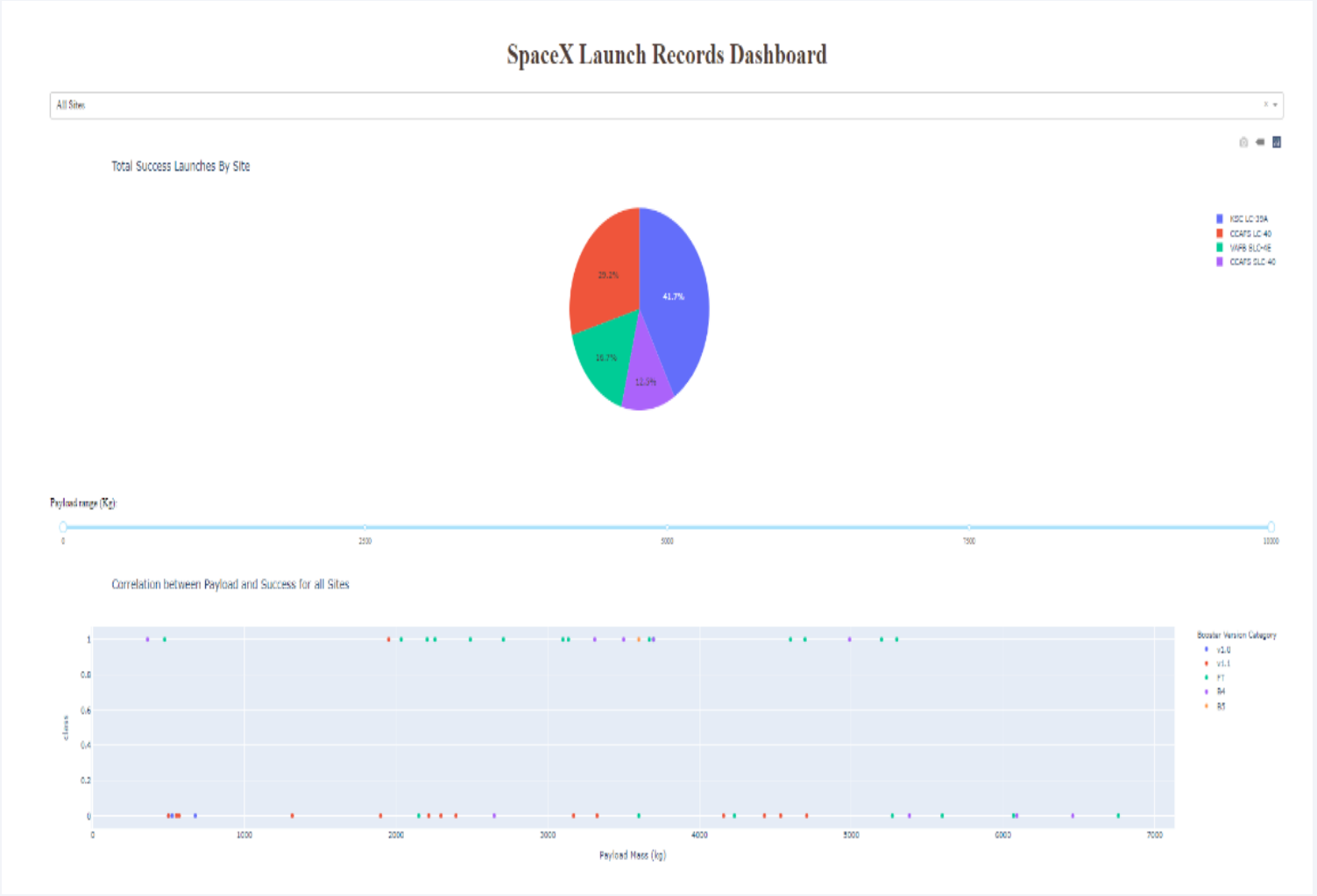
https://github.com/chatziserif/SpaceX_Capstone_Project_IBM/blob/main/6-Folium_LaunchSitesLoctions.ipynb





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.

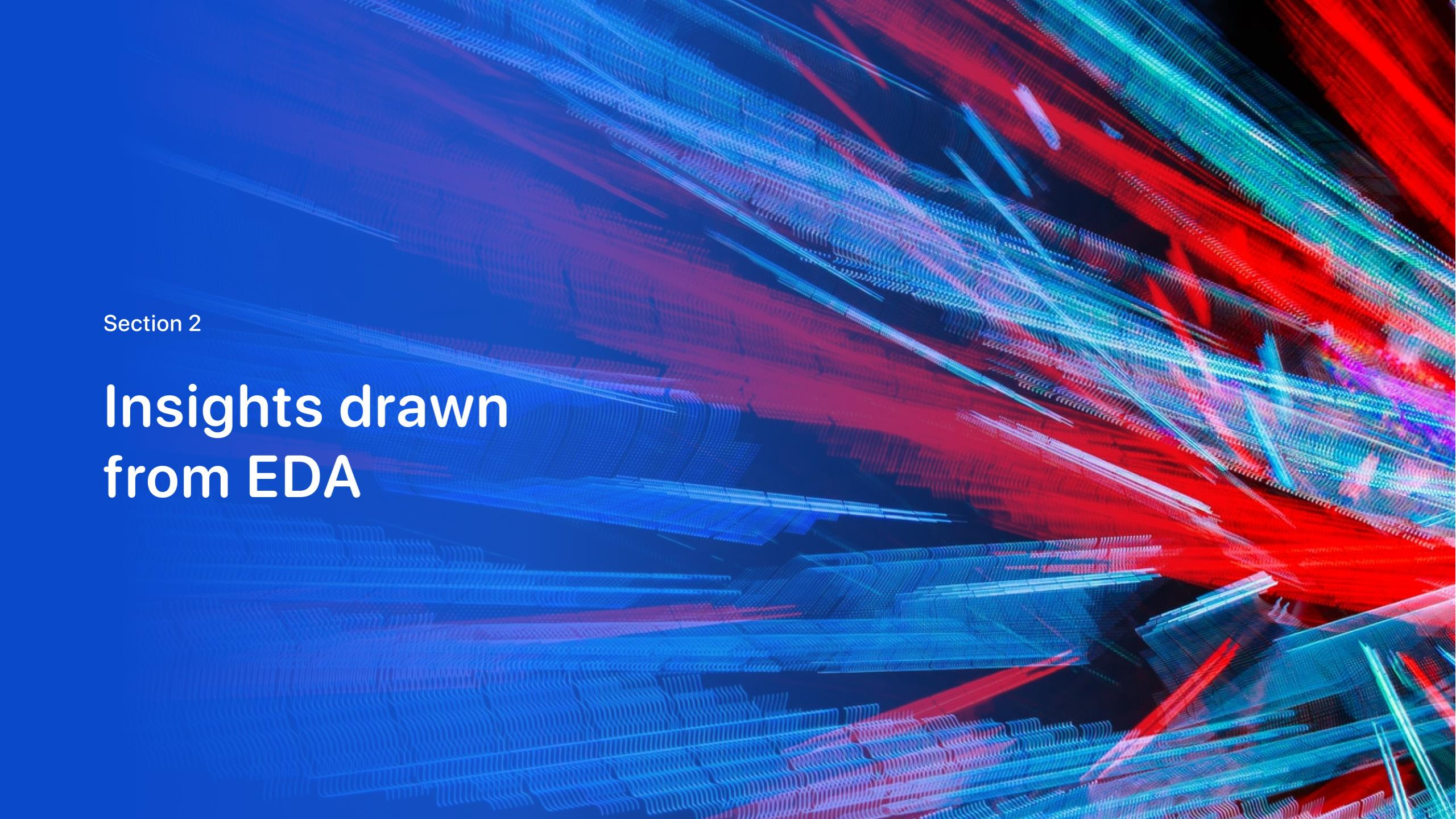


Predictive Analysis (Classification) Machine Learning

- 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 https://github.com/chatziserif/SpaceX_Capstone_Project_IBM/blob/main/7-SpaceX_MachineLearningPrediction.ipynb

Results

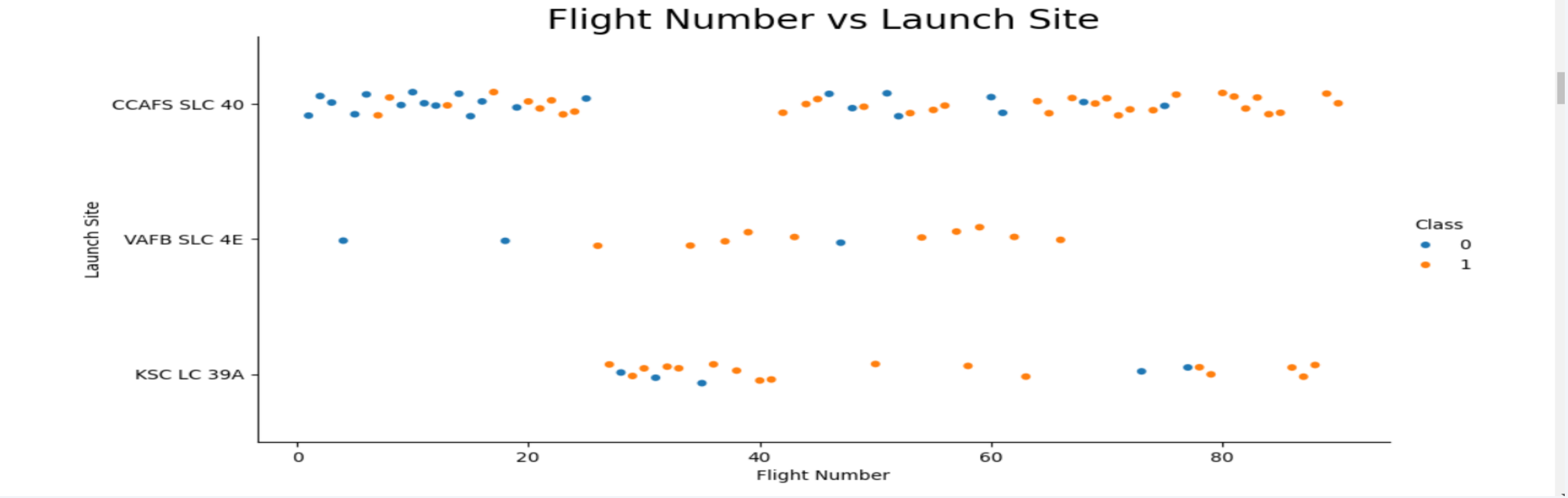
- The exploratory data analysis (EDA) has shown us that successful landing outcomes are somewhat correlated with flight number. It was also apparent that successful landing outcomes have had a significant increase since the year 2013.
- All launch sites are located near to coast line. This makes it easier to test rocket landings in the water.
- The machine learning was able to predict the landing success of rockets and the best model was decision tree.



Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



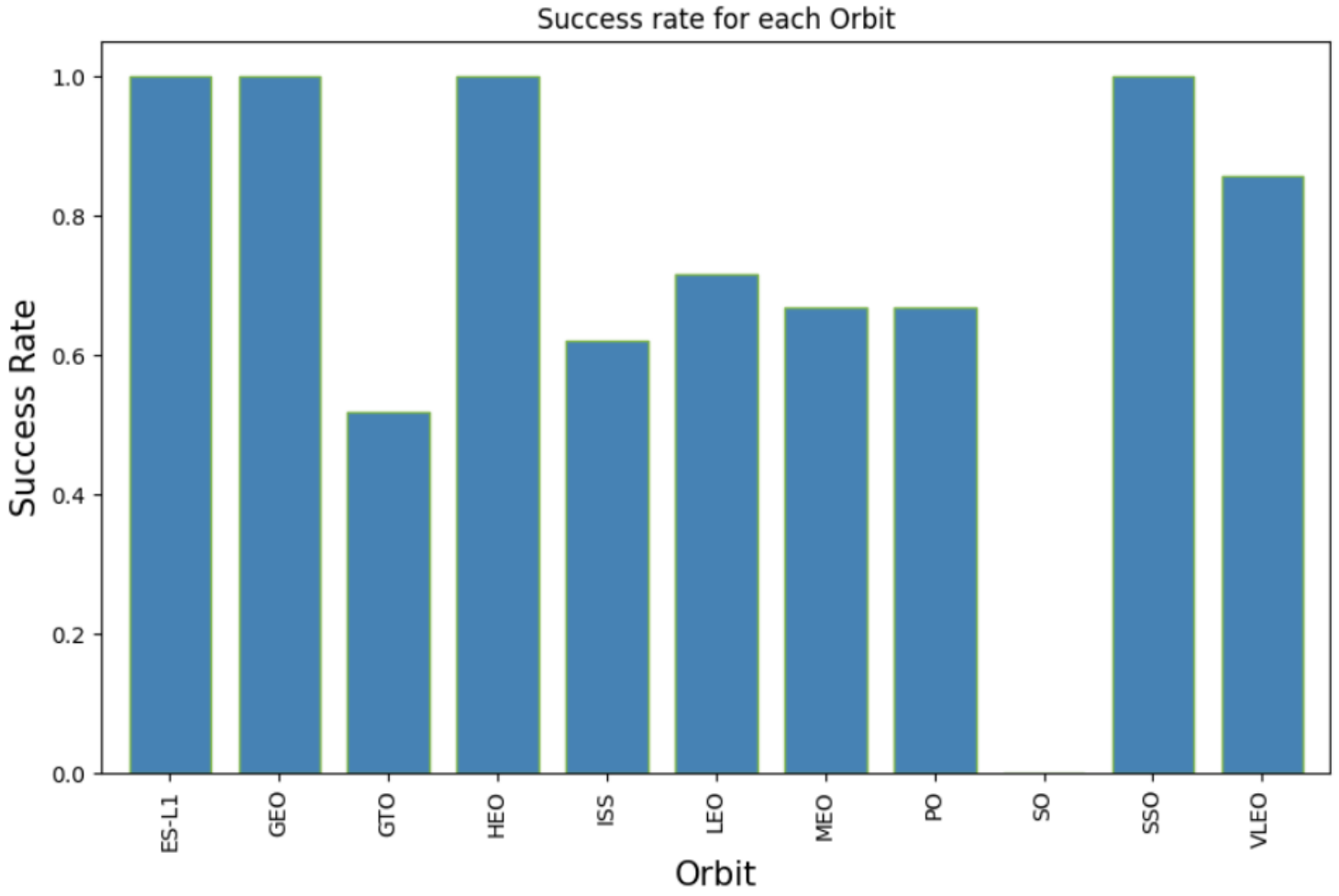
- The plot displays valuable info about:
  - Flight numbers
  - Number of flights per Launch sites
  - Success/Failure per launch site

Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- It is also observed that the orbit **SO** has the least success rate .

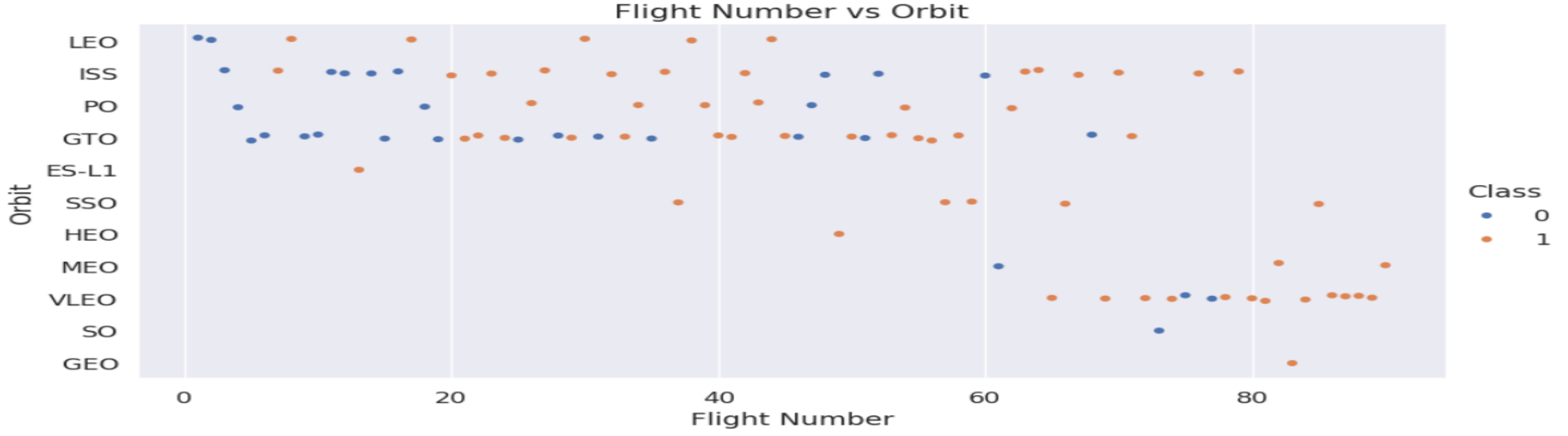
Success rate may strongly depend on both:

- Payload mass
- Orbit

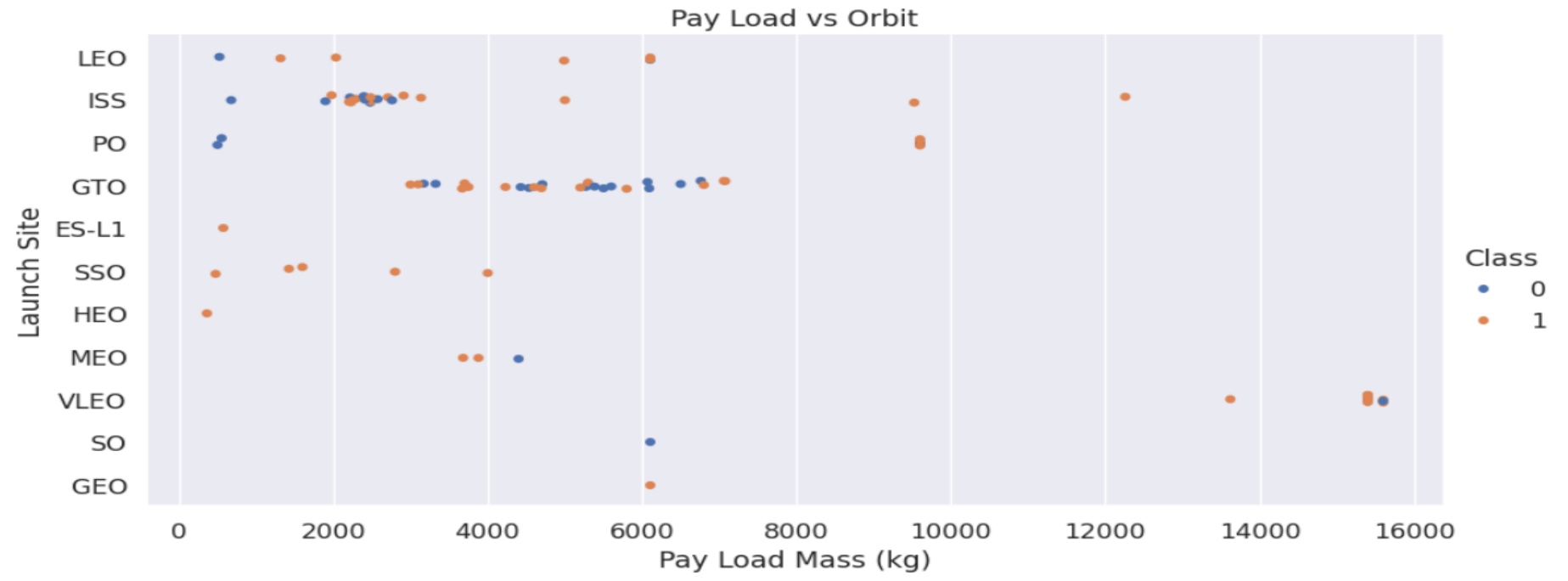


Flight Number vs. Orbit Type

- The plot brings additional info:
  - Number of flights per orbit
  - Success rate per orbit
- 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.
- dk



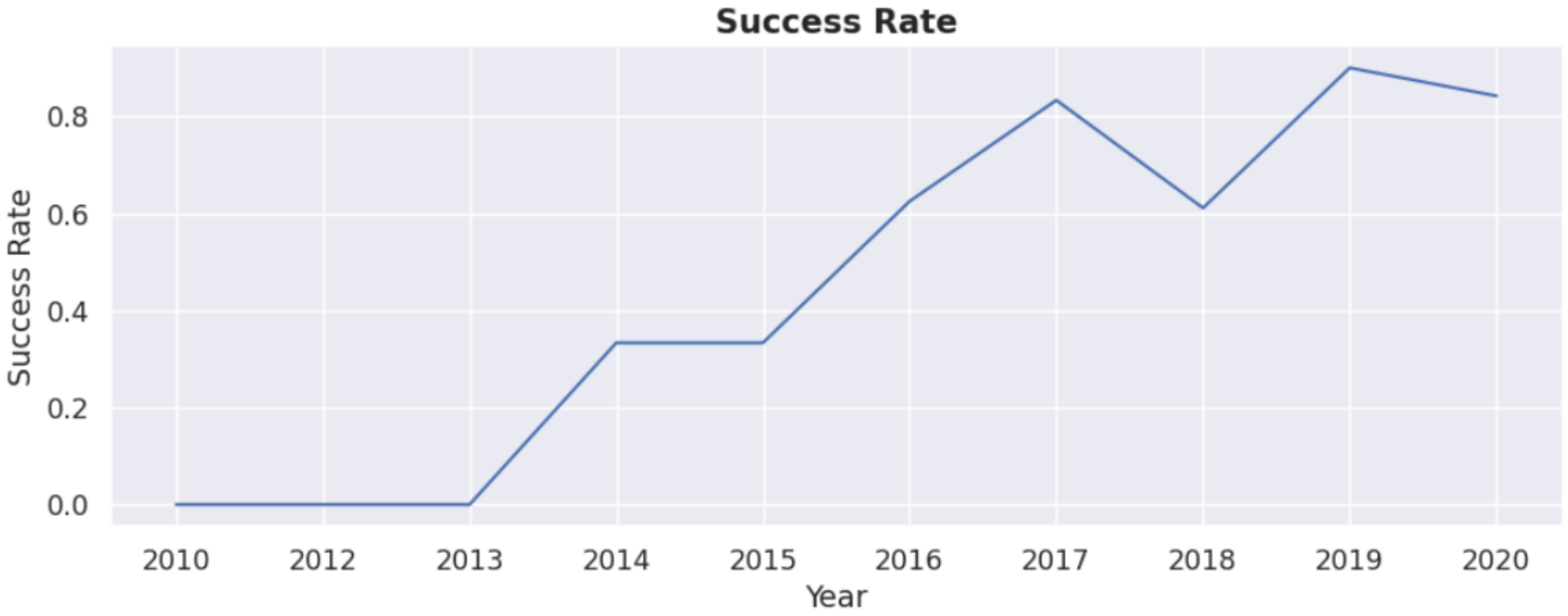
Payload vs. Orbit Type



- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.
- However For GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

- 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.
- There are 4 unique launch sites:
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40

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

```
[11]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTABLE;
```

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

```
[11]: ,,,,,,,,,,,,,,,,,,,,
```

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

Launch Site Names Begin with 'CCA'

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

[12]:
```
%sql SELECT * FROM SPACEXTABLE WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Done.
```

[12]:

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

- We used the query above to display 5 records where launch sites begin with `CCA`. As we can see, there are other organizations besides SpaceX that were testing their rockets.

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)

```
[16]: %sql SELECT SUM(payload_mass__kg_) FROM SPACEXTABLE WHERE customer = 'NASA (CRS)';
```

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

```
[16]: ,,,,,,,,,,
```

SUM(payload_mass__kg_)
45596

Average Payload Mass by F9 v1.1

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

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

```
[23]: %sql SELECT AVG(payload_mass__kg_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';
```

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

```
[23]: ,,,,,,,,,,
```

AVG(payload_mass__kg_)
2928.4

First Successful Ground Landing Date

- We observe that the date of the first successful landing outcome on ground pad was 22nd December 2015. We obtain this information using the query below:

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

```
[14]: %sql SELECT MIN(DATE) AS FIRST_SUCCESSFULL_LANDING_DATE FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';
```

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

[14]:

FIRST_SUCCESSFULL_LANDING_DATE
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

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

```
[25]: %sql SELECT DISTINCT Booster_Version, Customer, Landing_Outcome, PAYLOAD_MASS__KG_  FROM SPACEXTABLE \
      WHERE Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000;
```

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

[25]:

Booster_Version	Customer	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	SKY Perfect JSAT Group	Success (drone ship)	4696
F9 FT B1026	SKY Perfect JSAT Group	Success (drone ship)	4600
F9 FT B1021.2	SES	Success (drone ship)	5300
F9 FT B1031.2	SES EchoStar	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

- We used **COUNT** function for **mission_outcome** column to obtain the total number of successful and failure mission outcomes.

List the total number of successful and failure mission outcomes

```
[40]: %sql SELECT COUNT(mission_outcome) FROM SPACEXTABLE ;
```

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

[40]:

COUNT(mission_outcome)
101

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery with **WHERE** and **MAX()** functions to payload_mass_kg_ column.
- We observe that 12 boosters have carried the maximum payload mass of 15600kg.

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

```
[17]: %sql SELECT booster_version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
 * sqlite:///my_data1.db
Done.
```

[17]:

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

```
[18]: %sql SELECT max(PAYLOAD_MASS__KG_) FROM SPACEXTABLE;
 * sqlite:///my_data1.db
Done.
```

[18]:

max(PAYLOAD_MASS__KG_)
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
- We observe that 2 boosters **F9 v1.1 B1012** and **F9 v1.1 B1015** failed to land in 2015.

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

[19]:
```
%sql SELECT Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Failure (drone ship)'  \
AND Date BETWEEN '2015-01-01' AND '2015-12-31';
```

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

[19]:

Booster_Version	Launch_Site	Landing_Outcome
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
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.
- We observe that the number of successful landings have increased since 2015.

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) from 2010-06-04 to 2017-03-20

[22]:
```
%sql SELECT Landing_Outcome, COUNT(Landing_Outcome), Date FROM SPACEXTABLE \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome \
ORDER BY COUNT(Landing_Outcome) DESC;
```

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

[22]:

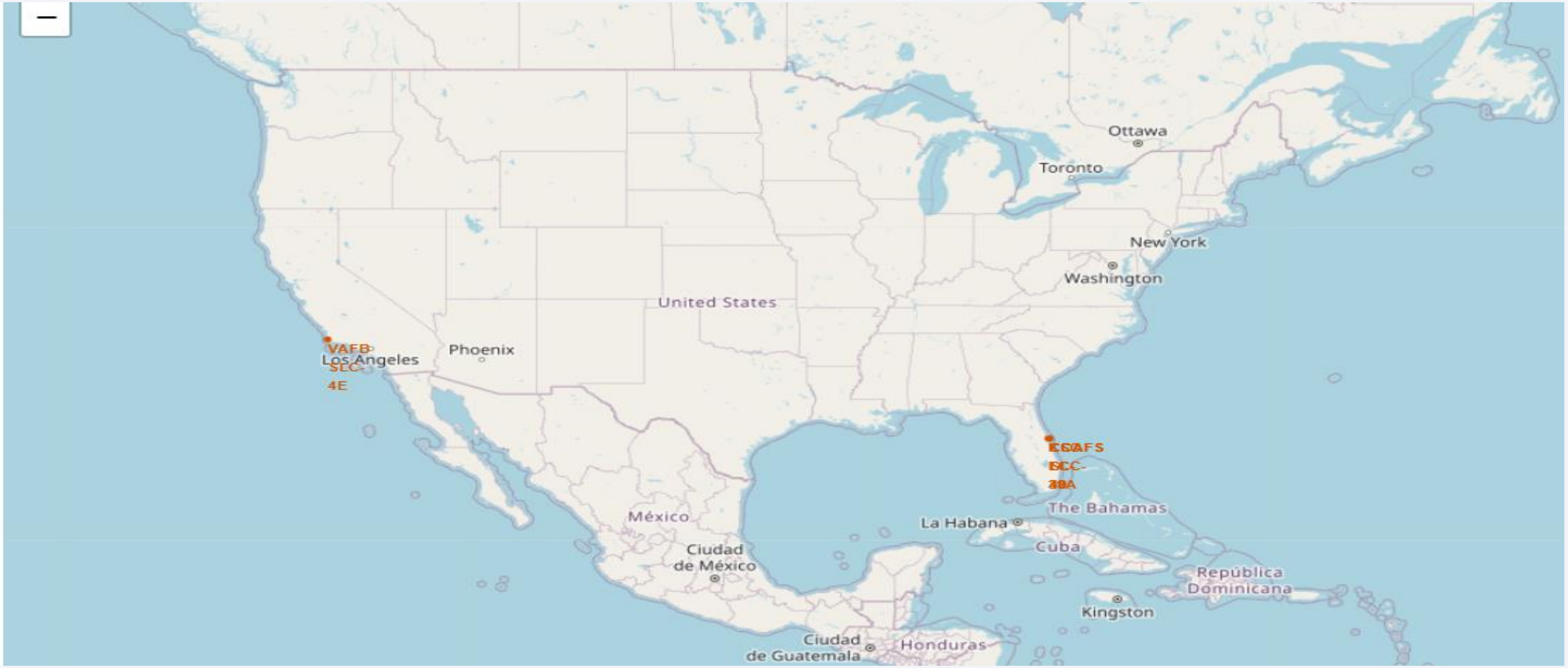
Landing_Outcome	COUNT(Landing_Outcome)	Date
No attempt	10	2012-05-22
Success (drone ship)	5	2016-04-08
Failure (drone ship)	5	2015-01-10
Success (ground pad)	3	2015-12-22
Controlled (ocean)	3	2014-04-18
Uncontrolled (ocean)	2	2013-09-29
Failure (parachute)	2	2010-06-04
Precluded (drone ship)	1	2015-06-28

Section 4

Launch Sites Proximities Analysis

All launch sites global map markers

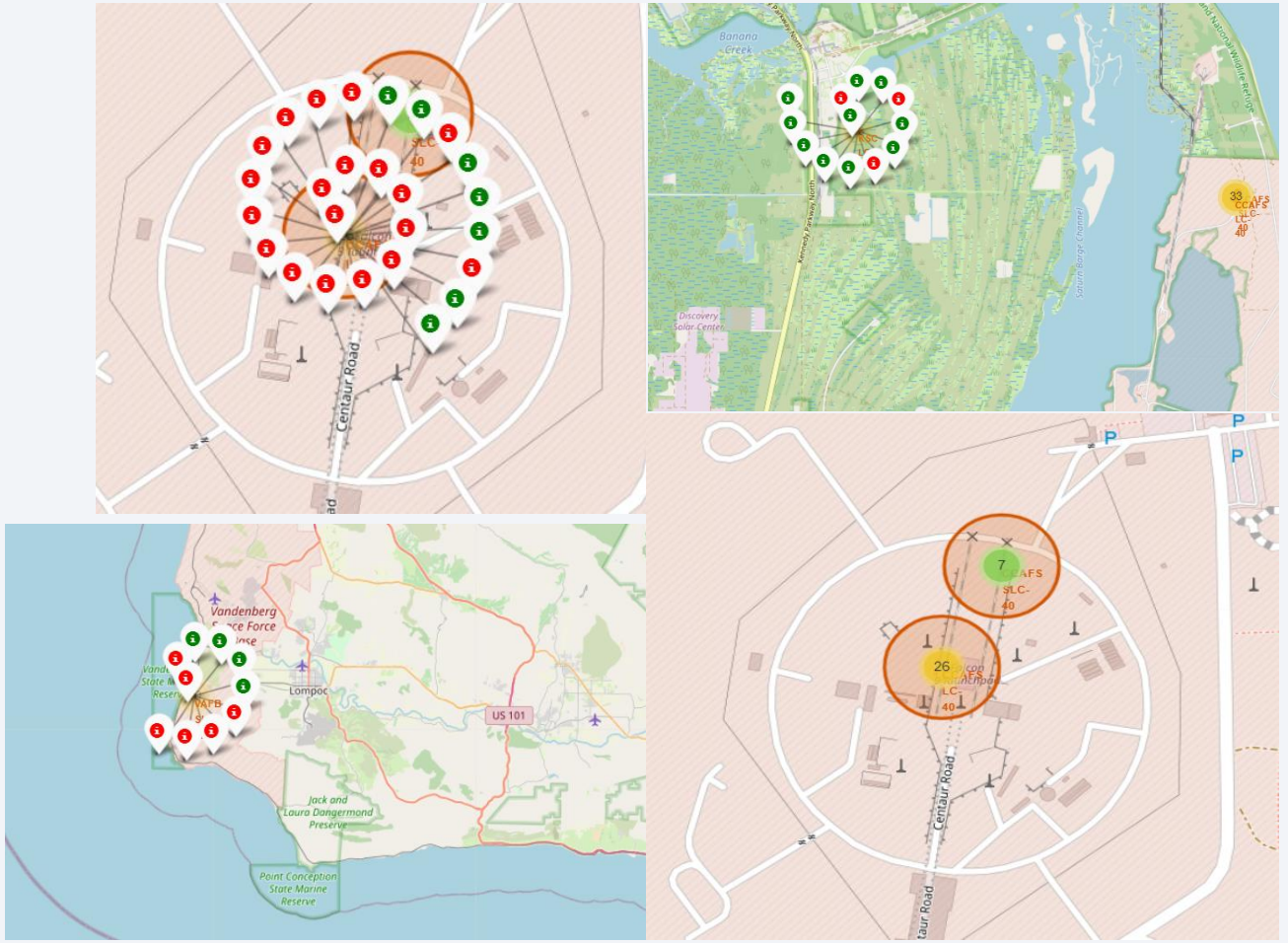
We can see that the SpaceX launch sites are in the United States of America coasts(Florida and California).Also, there are in proximity to the Equator line.



Markers showing launch sites with color labels

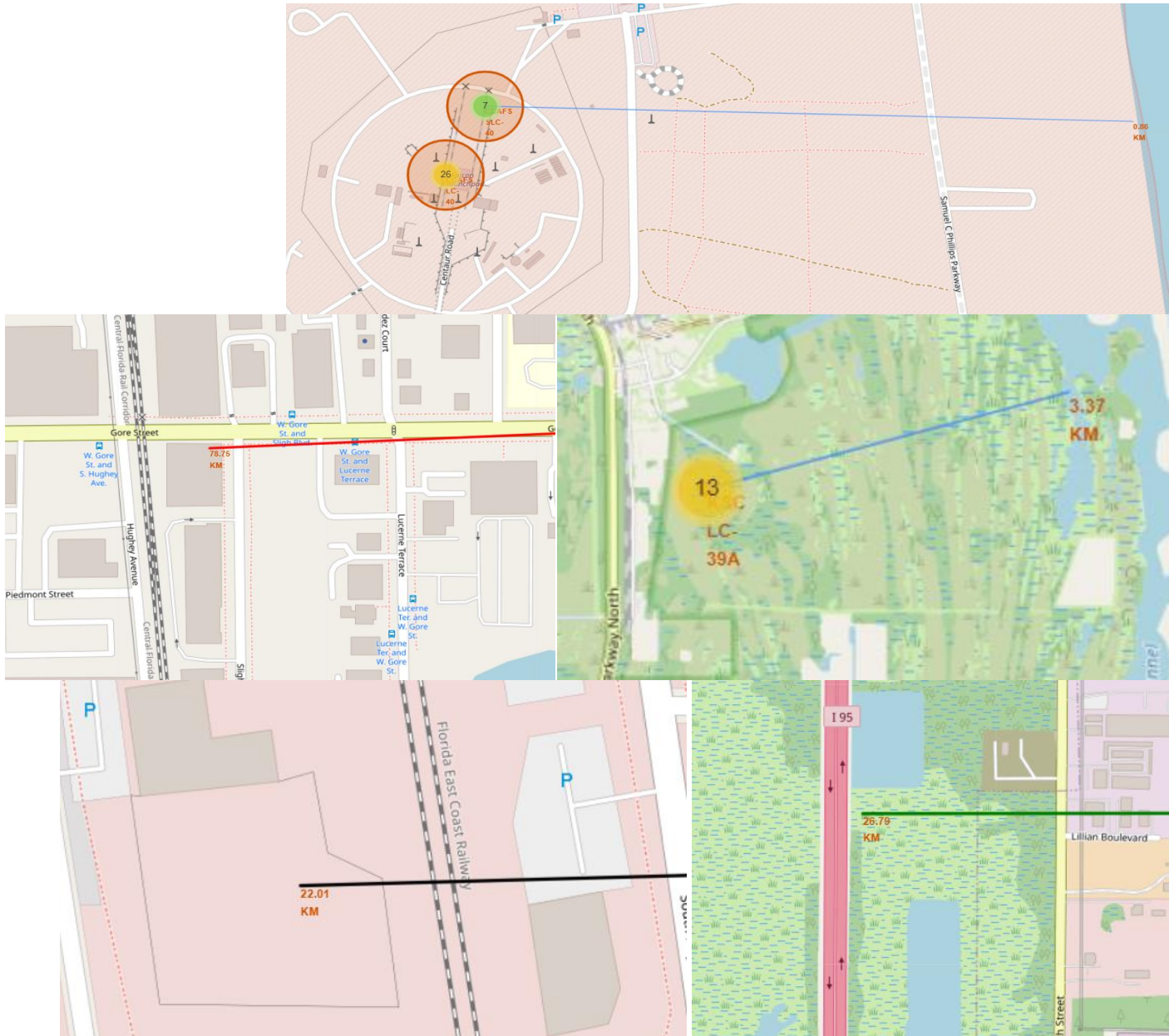
- The figure shows the launch outcomes for various launch sites:
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40
  - CCAFS LC-40

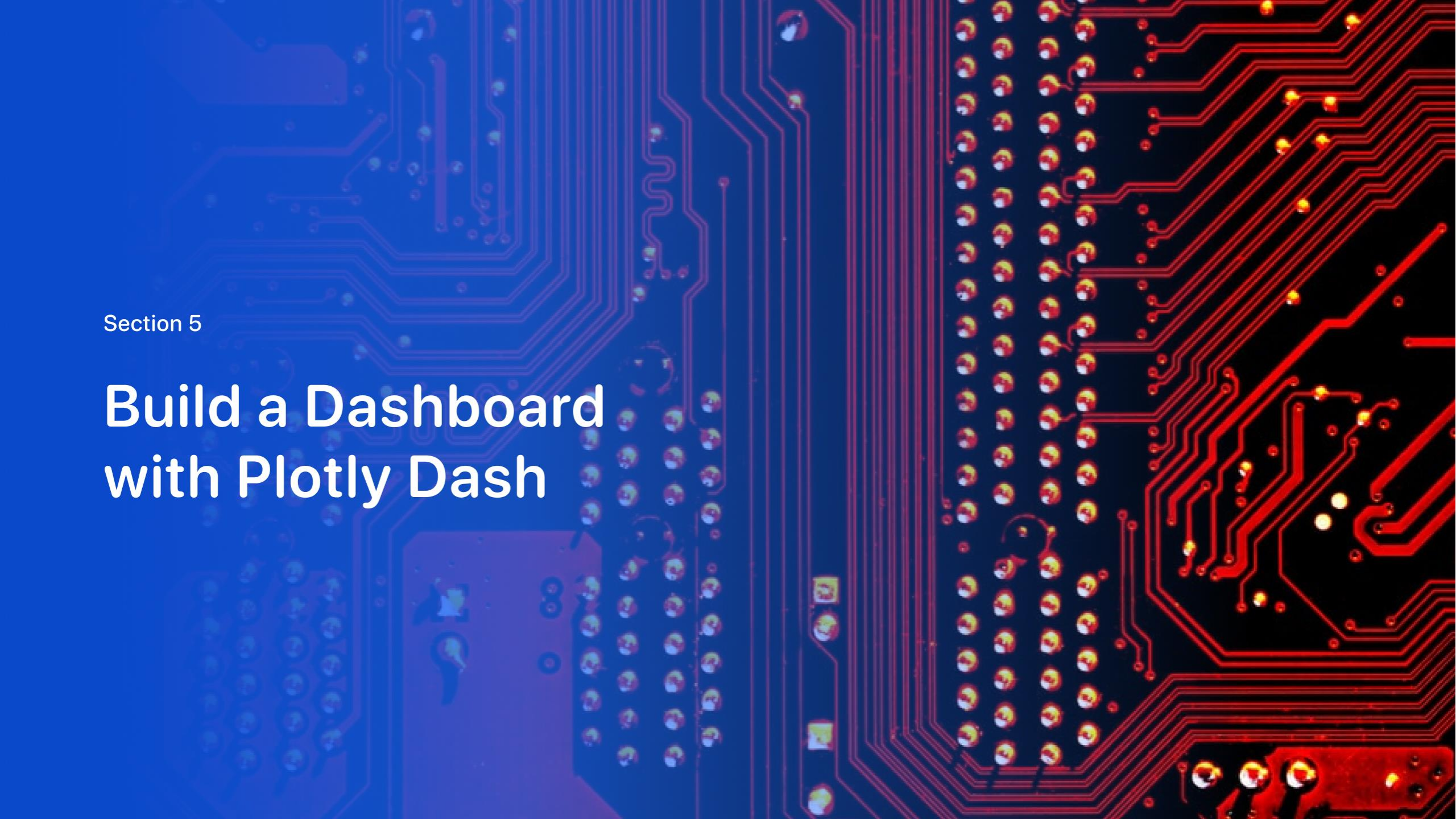
- Left coast site has 10 trails and right coast site has 46 trails.
- Red icons indicate failed outcomes and the green icons indicate successful outcomes.



Launch Site distance to landmarks

- KSC LC-39A is 3.37 km far from the coast.
- Distance from CCAFS_SLC40 to:
  - Closest coast: 860m
  - Florida East Coast Railway: 22 km
  - Highway I 95: 26.8 km
  - Orlando: 78 km
- Launch sites are close to coasts for safety issues.
- Launch sites are relatively far from populated areas for protecting population from serious incidents at lift off: explosion on the launch pad.



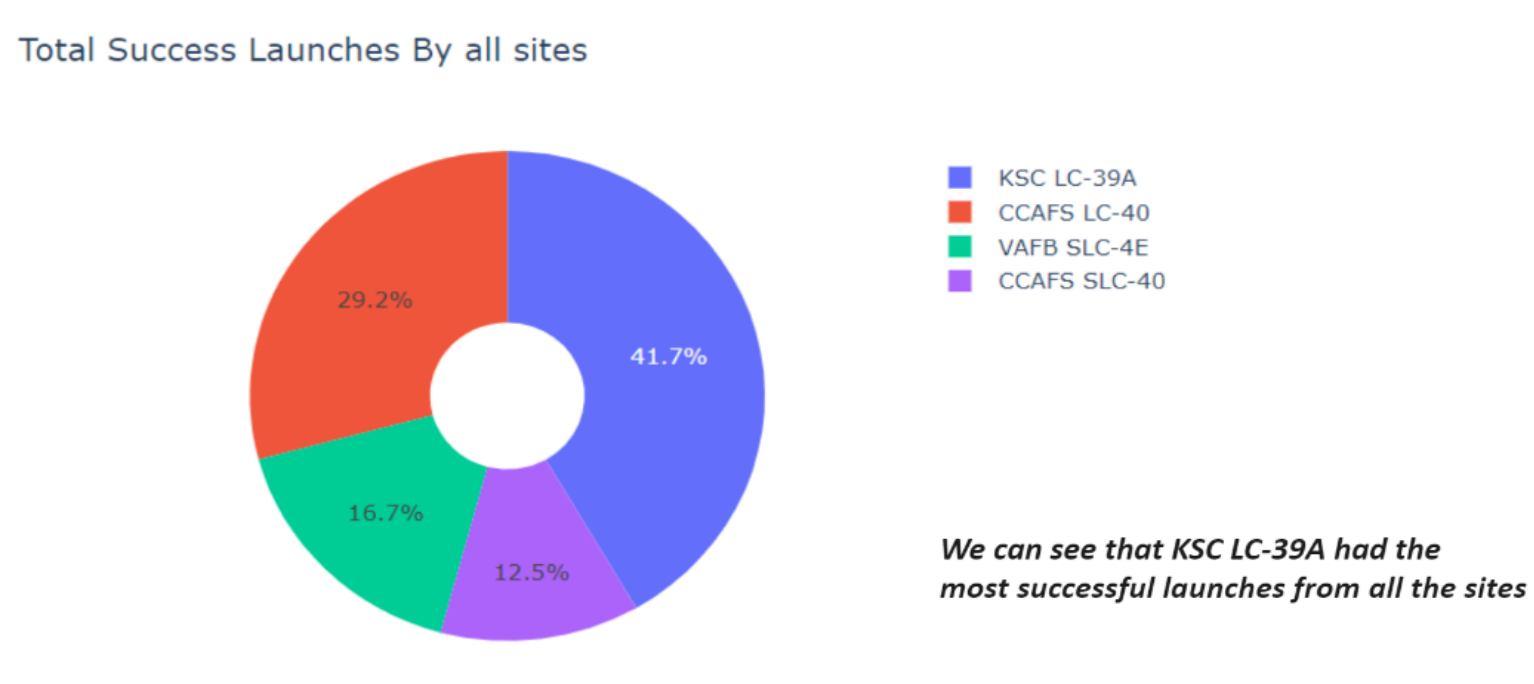


Section 5

Build a Dashboard with Plotly Dash

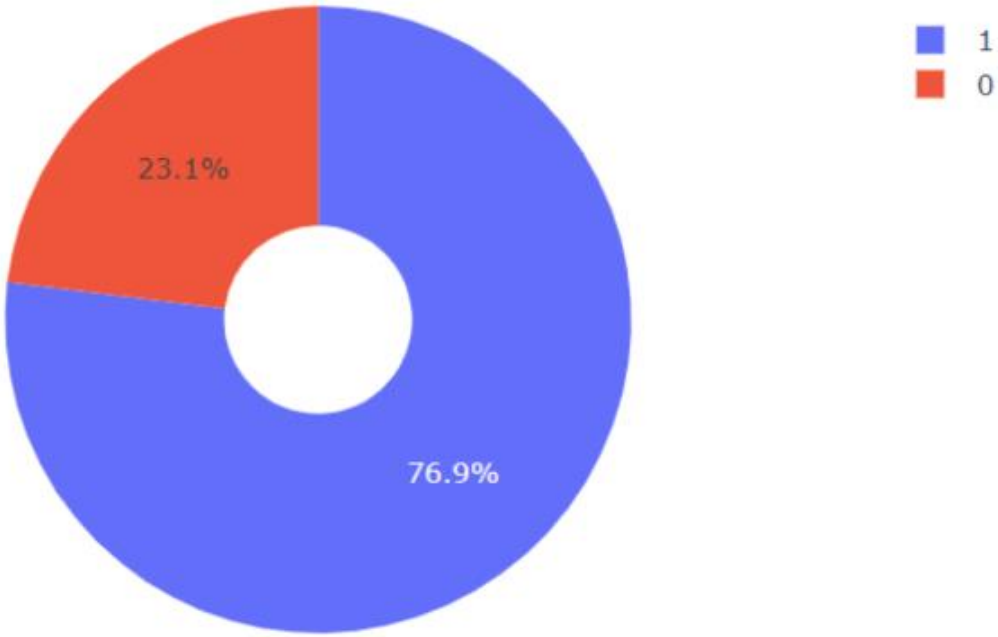
Pie chart showing the success percentage achieved by each launch site

It is shown that KSC LC-39A has the largest success rate with about 41.7% of the total success ratio with other sites.



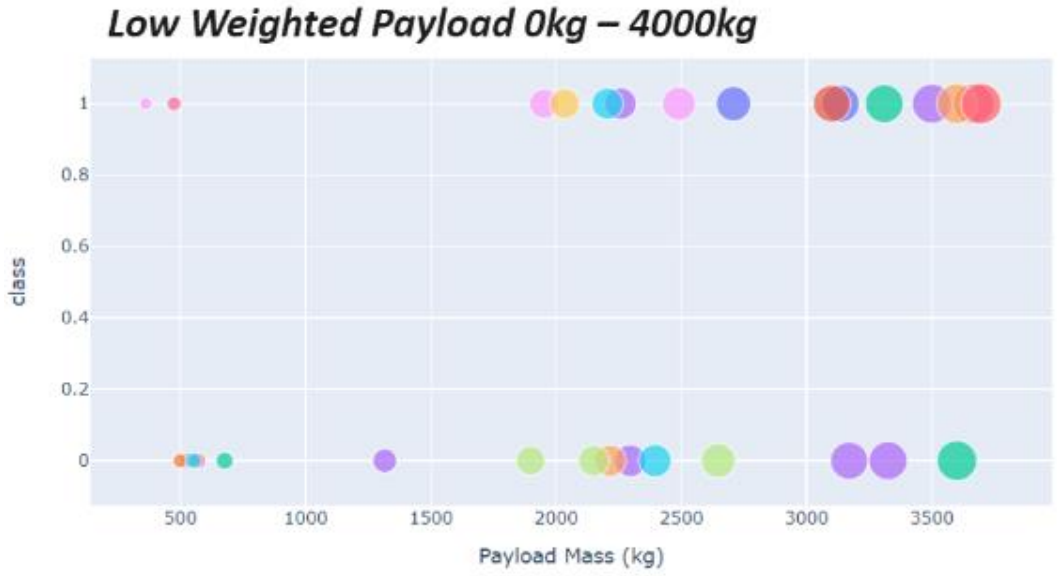
Pie chart showing the Launch site with the highest launch success ratio

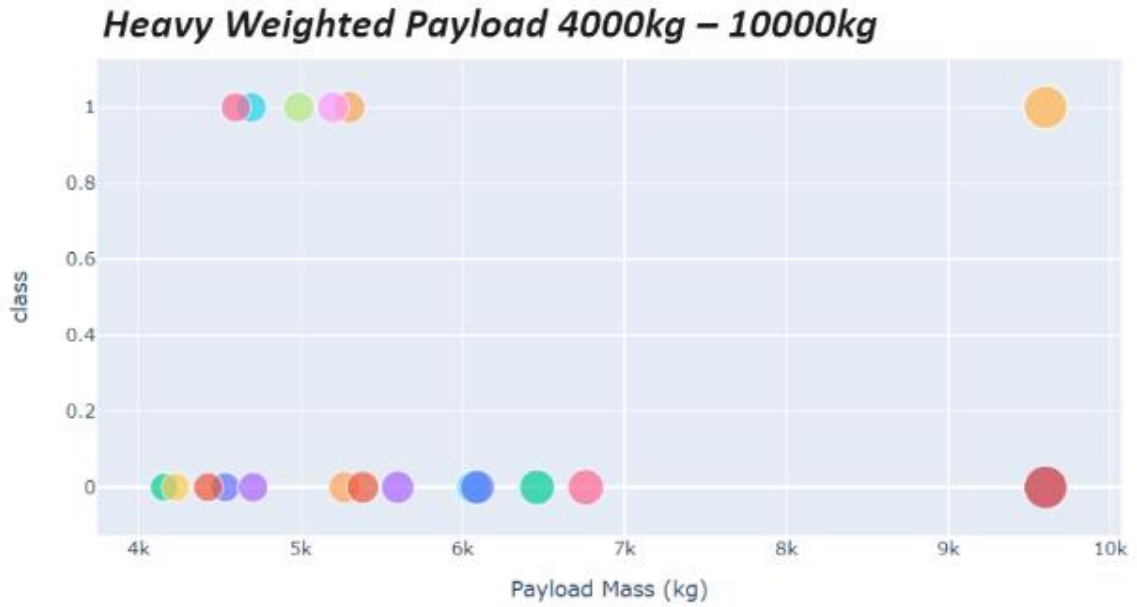
As we see KSC LC 39A has a 76.9% success rate while getting a 23.1% failure rate.



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

It appears that the payload range between 2000kg and 4000 kg has the highest success rate

Section 6

Predictive Analysis (Classification)

Classification Accuracy

- In our case, decision tree classifier is the model with the highest classification accuracy

Task 8: Decision Tree

Build a decision tree classifier object then build a `GridSearchCV` object `tree_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.

[34]:
```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

[35]:
```
tree_cv = GridSearchCV(tree, parameters, cv=10)
tree_cv.fit(X_train, Y_train)
```

```
/lib/python3.11/site-packages/sklearn/model_selection/_validation.py:425: FitFailedWarning: ...
```

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

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'}
accuracy : 0.8892857142857145
```

Task 9

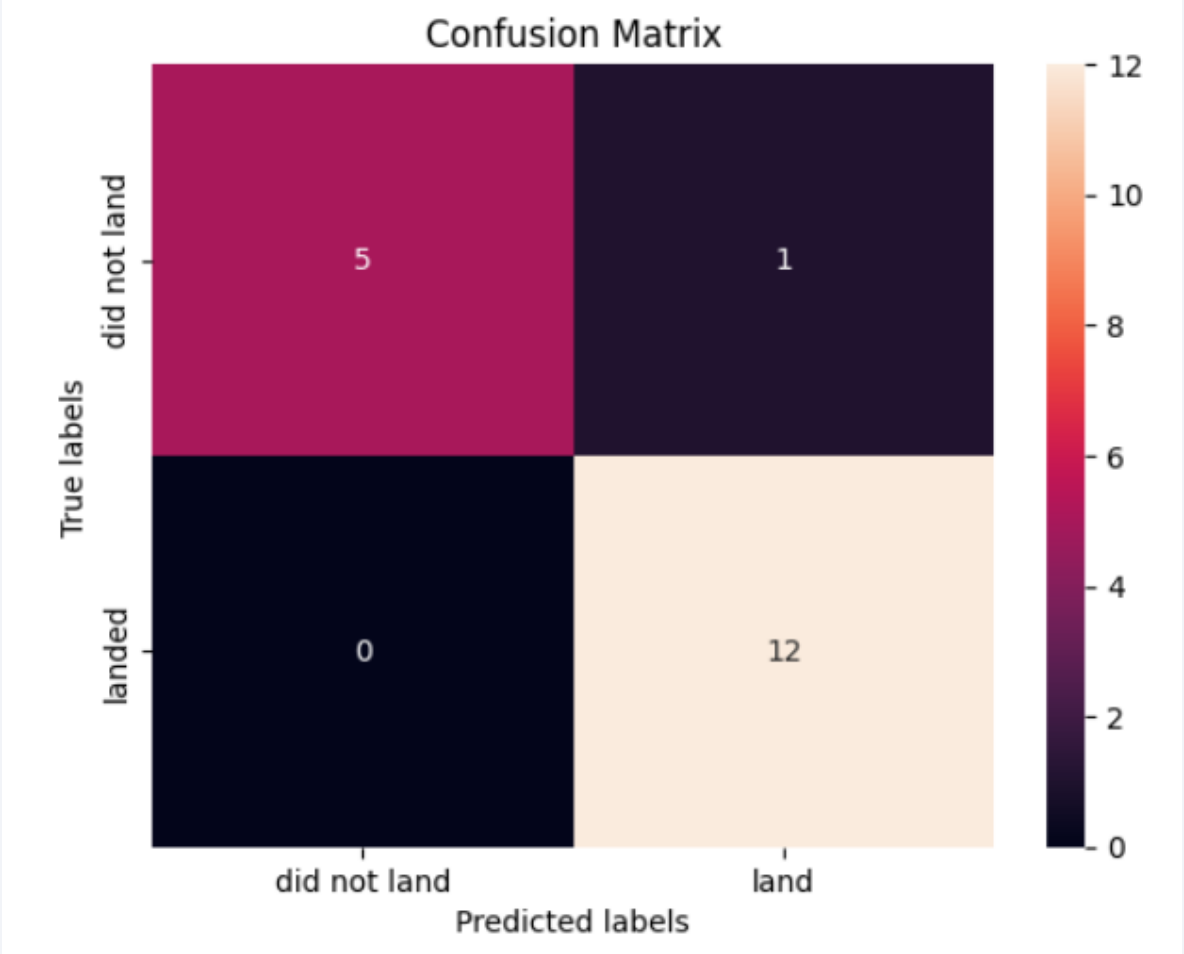
Calculate the accuracy of tree_cv on the test data using the method `score:`

[28]:
```
tree_cv.score(X_test, Y_test)
```

[28]:
```
0.9444444444444444
```

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. We have just one misprediction with 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. Payload mass is also associated with the success rate, the more massive the payload, the more likely the first stage will return. Especially for the **CCAFS SLC 40** launch site.
- There has been an increase in success rate since 2013 and kept increasing 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!