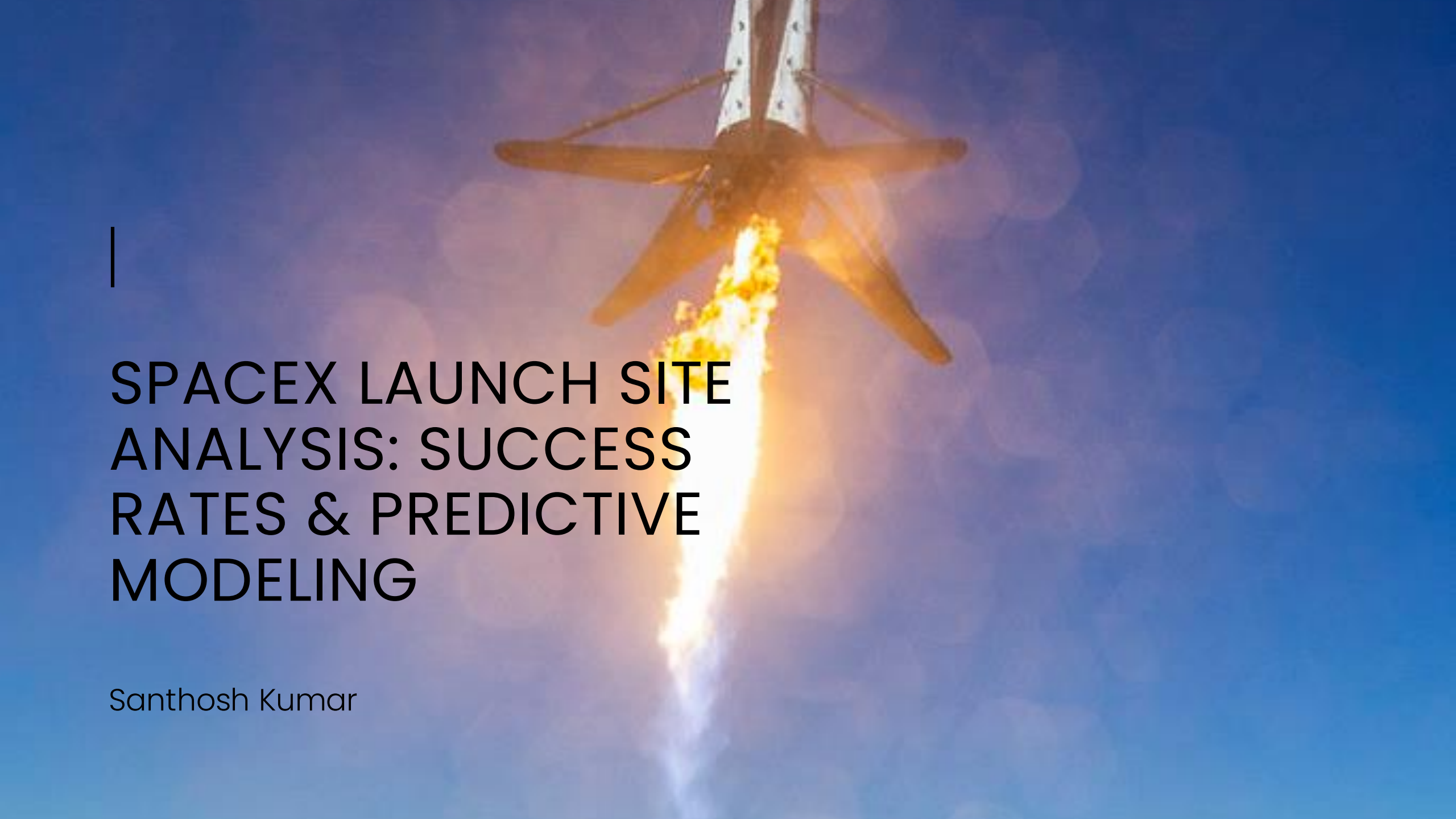


SPACEX LAUNCH SITE ANALYSIS: SUCCESS RATES & PREDICTIVE MODELING

Santhosh Kumar

EXECUTIVE SUMMARY SLIDE:

- Briefly introduce the project goal: analyzing SpaceX launch data to study success rates across launch sites and predict mission outcomes.
- Highlight the importance: Data-driven insights can improve launch planning and reduce risk.
- Mention the approach: Exploratory Data Analysis, interactive visualization (Folium/Plotly), and Machine Learning.
- End with key outcomes: Identification of the best-performing launch sites and predictive insights on future launches.

INTRODUCTION:

- SpaceX is a leader in commercial space exploration, with multiple launch sites and missions.

Focus of Study:

- Compare success rates across different launch sites.
- Explore payload and booster factors affecting outcomes.
- Predict the likelihood of a successful launch at a given site.

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
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

```
response=requests.get(static_json_url)
```

```
response.status_code
```

```
200
```

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

```
# Use json_normalize meethod to convert the json result into a dataframe
response=response.json()
data = json_normalize(response)
```

Using the dataframe `data` print the first 5 rows

```
# Get the head of the dataframe
data.head(2)
```

	static_fire_date_utc	static_fire_date_unix	tbd	net	window	rocket	success	details	crew	ships	capsules	payloads
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	False	0.0	5e9d0d95eda69955f709d1eb	False	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b6c3bb0006eeb1e1]

DATA COLLECTION – SPACEX API:

- Utilized SpaceX REST API to collect structured launch data.
- Normalized JSON responses into tabular format using pandas.
- Extracted relevant features: payload mass, orbit, launch site, and outcome.
- Stored data in CSV for downstream analysis and visualization.

- GIT link

DATA COLLECTION- WEB SCRAPING:

- Applied web scraping techniques to collect launch data from SpaceX/Wikipedia.
- Automated extraction of tabular data using BeautifulSoup.
- Cleaned and structured the scraped data into tabular format.
- Exported dataset into CSV for further analysis and visualization.

GIT link

```
        # TODO: Append the launch_outcome into launch_dict with key `Launch outcome`
        launch_outcome = list(row[7].strings)[0]
        launch_dict['Launch outcome'].append(launch_outcome)
        print(launch_outcome)

        # Booster Landing
        # TODO: Append the launch_outcome into launch_dict with key `Booster Landing`
        booster_landing = landing_status(row[8])
        launch_dict['Booster landing'].append(booster_landing)
        print(booster_landing)
```

```
1
4 June 2010
18:45
F9 v1.07B0003.18
CCAFS
Dragon Spacecraft Qualification Unit
Dragon Spacecraft Qualification Unit
LEO
SpaceX
Success

Failure
2
8 December 2010
15:43
F9 v1.07B0004.18
CCAFS
Dragon
Dragon
LEO
NASA
Success
Failure
3
22 May 2012
07:44
F9 v1.07B0005.18
CCAFS
Dragon
```

```
df.head(5)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longit
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.57
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.57
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.57
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.61
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.57

We can use the following line of code to determine the success rate:

```
df["Class"].mean()
```

```
np.float64(0.6666666666666666)
```

We can now export it to a CSV for the next section,but to make the answers consistent, in the next lab we will provide data in a pre-selected date range.

```
df.to_csv("dataset_part_2.csv", index=False)
```

DATA WRANGLING METHODOLOGY:

- Performed data inspection to identify inconsistencies and missing values.
- Applied cleaning steps: removing duplicates, handling nulls, standardizing formats.

- GIT Link

Exploratory Data Analysis

EDA WITH DATA VISUALIZATION:

- Visualized the relationship between flight number, launch site and other variables in the data-frame

- Used scatter plot, bar plot, line plot to visualize the relationships between the variables

- GIT LINK

EDA with SQL:

- Used SQL in python to get insights from data like unique launch sites, total number of successful and failure mission outcomes
- GIT Link

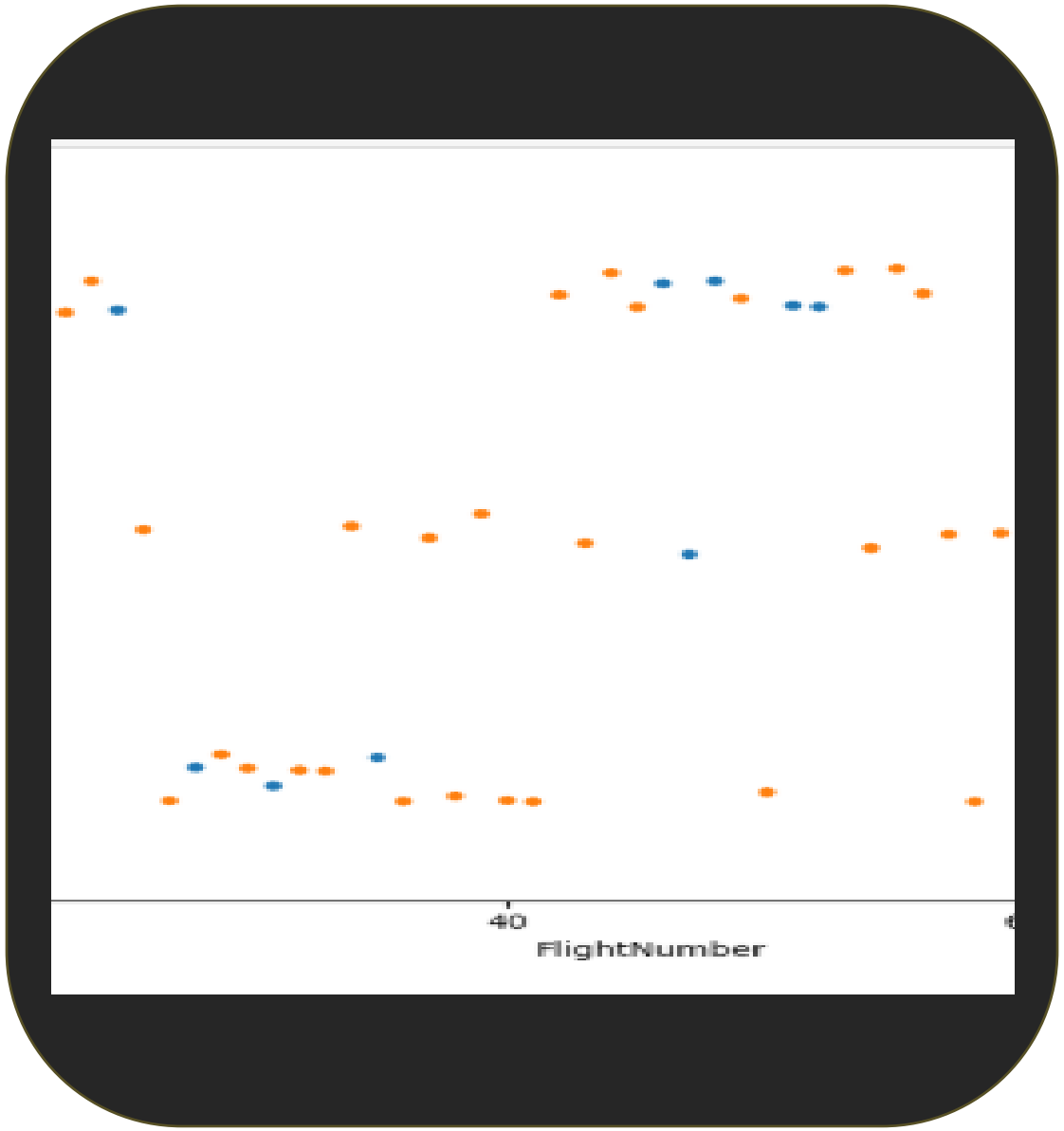
Interactive visual analytics:

- Built an interactive dashboard to get insights from collected data. Created dashboard of pie chart with Successful launch details based on launching site and scatterplot with Payload mass and class.

EDA WITH PREDICTIVE ANALYSIS METHODOLOGY:

- Built multiple classification models including Logistic Regression, KNN, Decision Tree, and SVM.
- Evaluated models using cross-validation, accuracy, precision, recall, and F1-score.
- Applied GridSearchCV for hyperparameter optimization to improve model performance.
- Selected the best-performing model based on highest validation accuracy and balanced metrics.
- The optimized model was then used for predicting SpaceX launch outcomes with high reliability.
- GIT link

EDA WITH VISUALIZATION USING PYTHON

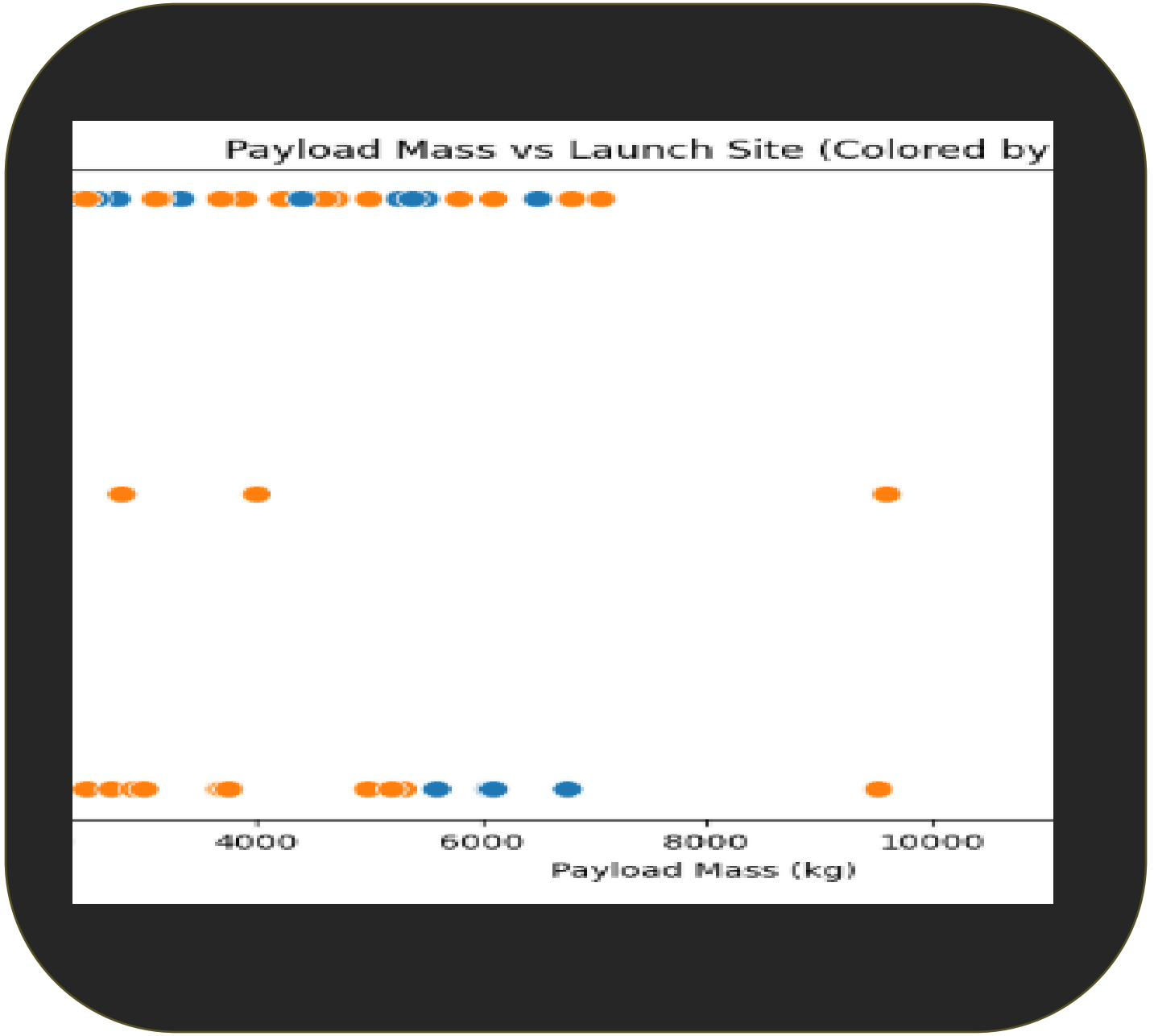


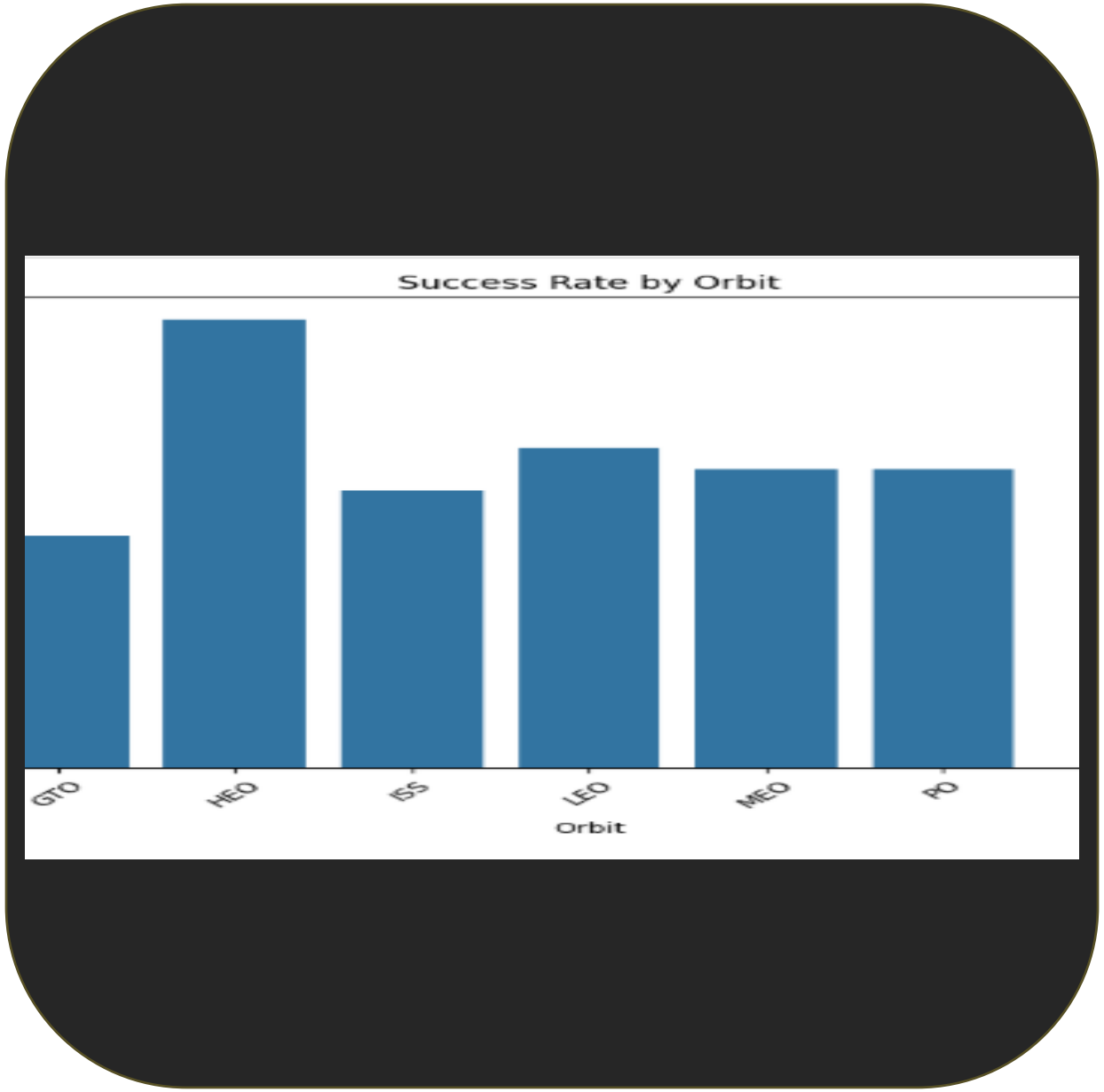
RELATIONSHIP BETWEEN FLIGHT NUMBER AND LAUNCH SITE:

This Scatter plot visualizes the relation between various flight number and their outcome on the launch site.

PAYLOAD VS. LAUNCH SITE SCATTER CHART:

This chart explains the relationship between the payload and launch sites based on the class



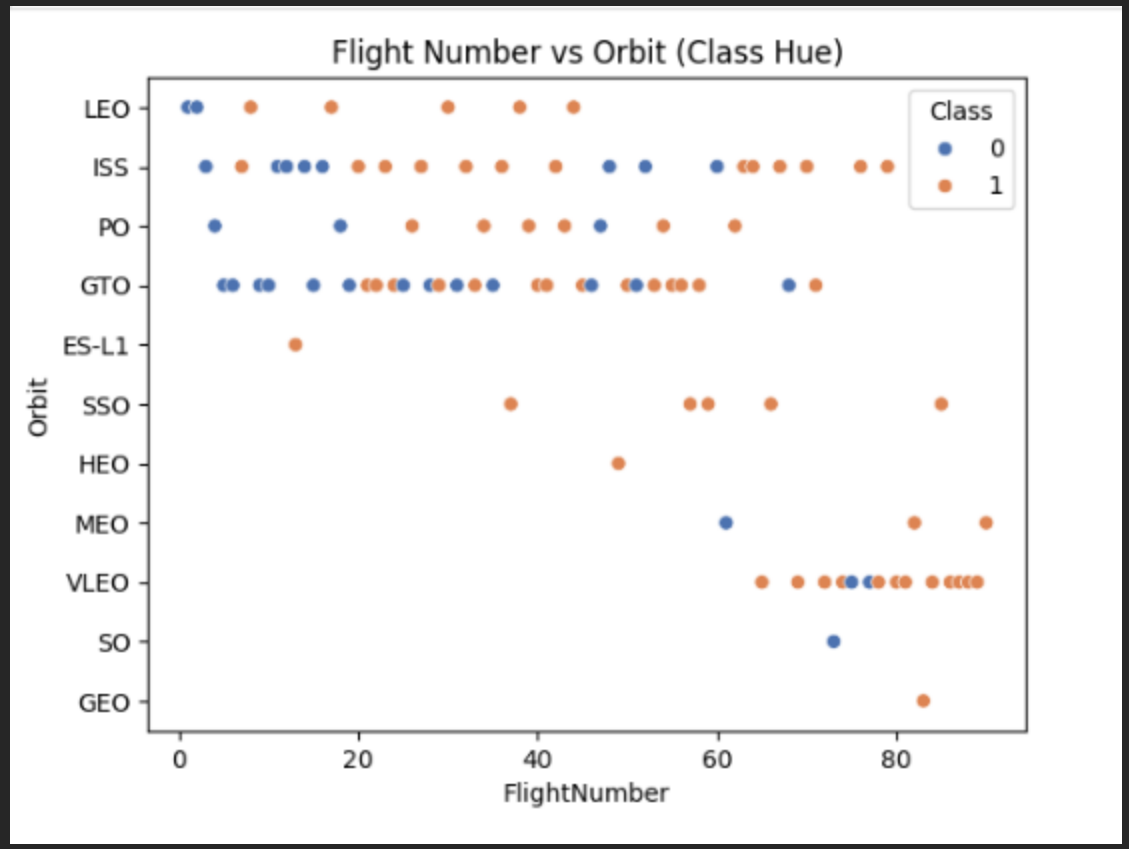


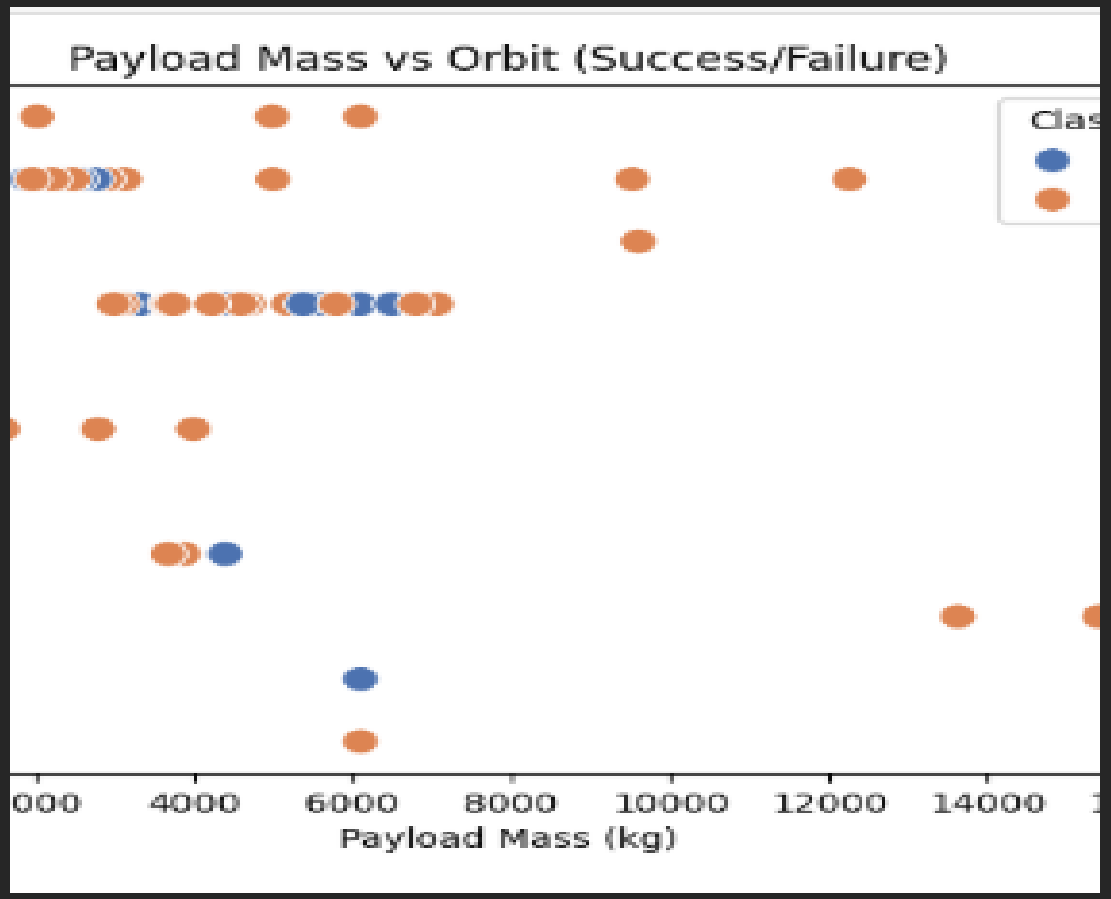
SUCCESS RATE VS. ORBIT TYPE BAR CHART:

- This bar chart explains the relationship between the success rate of the launch and the orbit it is launched to.

FLIGHT NUMBER VS. ORBIT TYPE SCATTER CHART:

- This scatterplot explains the relationship between the flight number and the orbit based on the class



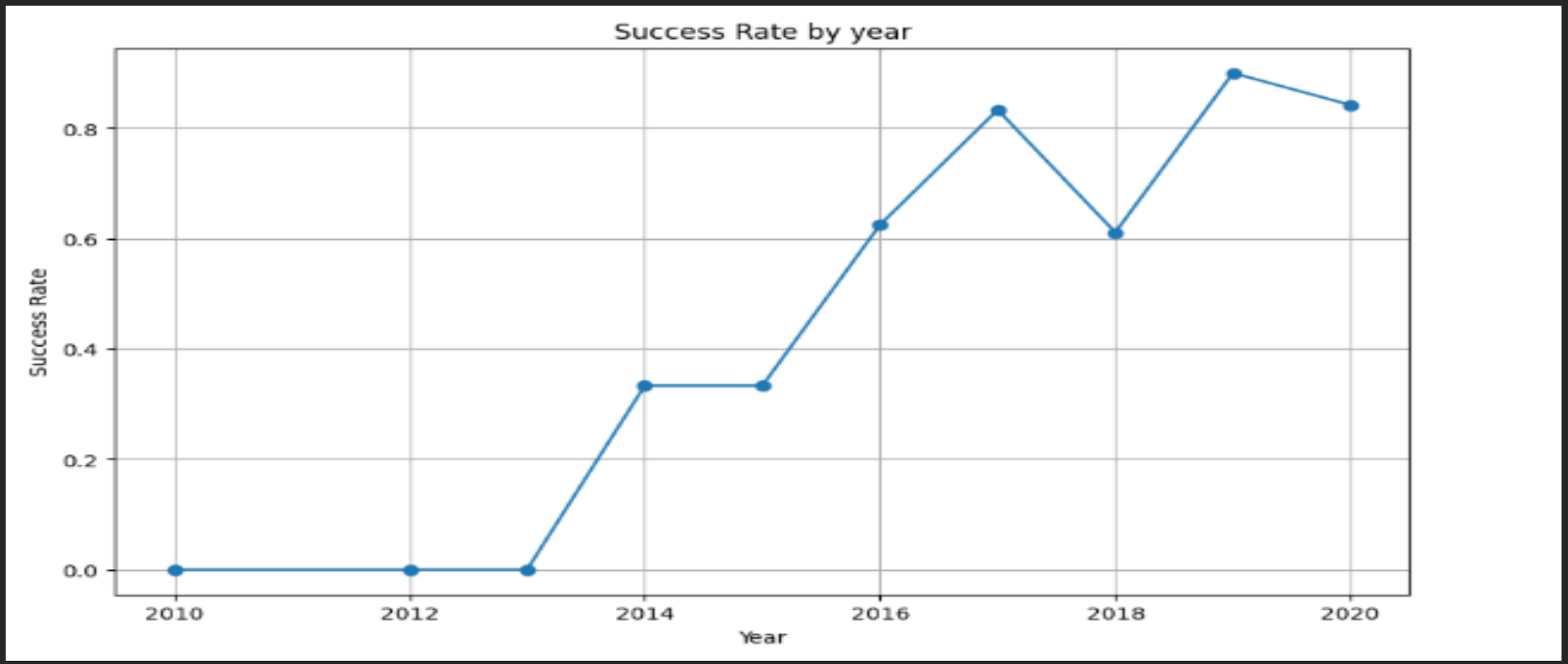


PAYLOAD VS. ORBIT TYPE SCATTER CHART :

- This scatterplot shows the relationship between the payload mass and the orbit based on the class whether the launch was successful or not.

LAUNCH SUCCESS YEARLY TREND LINE :

- This line plot shows the relationship between the successful launch trend line on yearly basis



EDA WITH SQL

LAUNCH SITE NAMES:

- Retrieved all launch sites from the dataframe using an SQL query.

```
%sql select * from spacextable 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
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

LAUNCH SITE NAMES BEGIN WITH `CCA`:

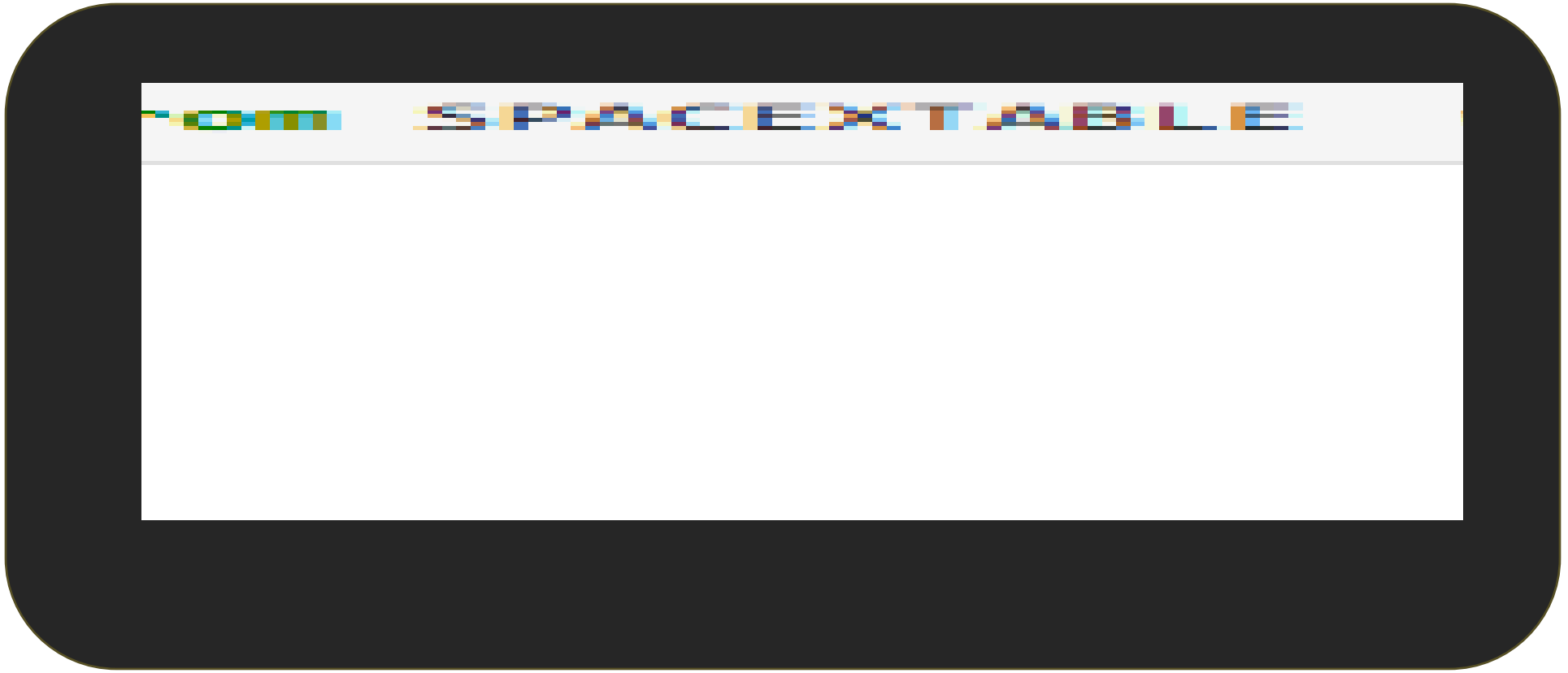
RETRIEVED ALL LAUNCH SITE NAMES BEGINNING WITH CCA USING AN SQL QUERY.

TOTAL PAYLOAD MASS CARRIED BY BOOSTERS:

- Calculated the total payload mass carried by all boosters using an SQL query.

AVERAGE PAYLOAD MASS BY F9 V1.1:

- Calculated the average payload mass specific to the F9 v1.1 version.



FIRST SUCCESSFUL GROUND LANDING DATE:

- Identified the date of the first successful ground landing.

```
[18]: %sql select min(date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
 * sqlite:///my_data1.db
Done.
[18]: min(date)
2015-12-22
```

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000:

- Used the SQL query to retrieve successful drone ship landings with payloads between 4000 and 6000 kg.

```
%sql select Booster_Version from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 AND 6000;
```

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

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

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES:

- Used an SQL query to determine the total count of successful and failed mission outcomes.

```
%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) AS Total FROM SPACEXTABLE GROUP BY Mission_Outcome;
```

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

Mission_Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

BOOSTERS CARRIED MAXIMUM PAYLOAD:

- Used an SQL query to identify the boosters that carried the maximum payload.

```
%sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = ( SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE );
```

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

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:

- Used an SQL query to identify the records for months in 2015

```
%sql SELECT substr(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE substr(Date, 1, 4) = '2015' AND Landing_Ou
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

RANK SUCCESS COUNT BETWEEN 2010-06-04 AND 2017-03-20:

- Used an SQL query to Rank the count of successful landings between 2010-06-04 and 2017-03-20

```
%sql SELECT Landing_Outcome, COUNT(*) AS Outcome_Count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER E
```

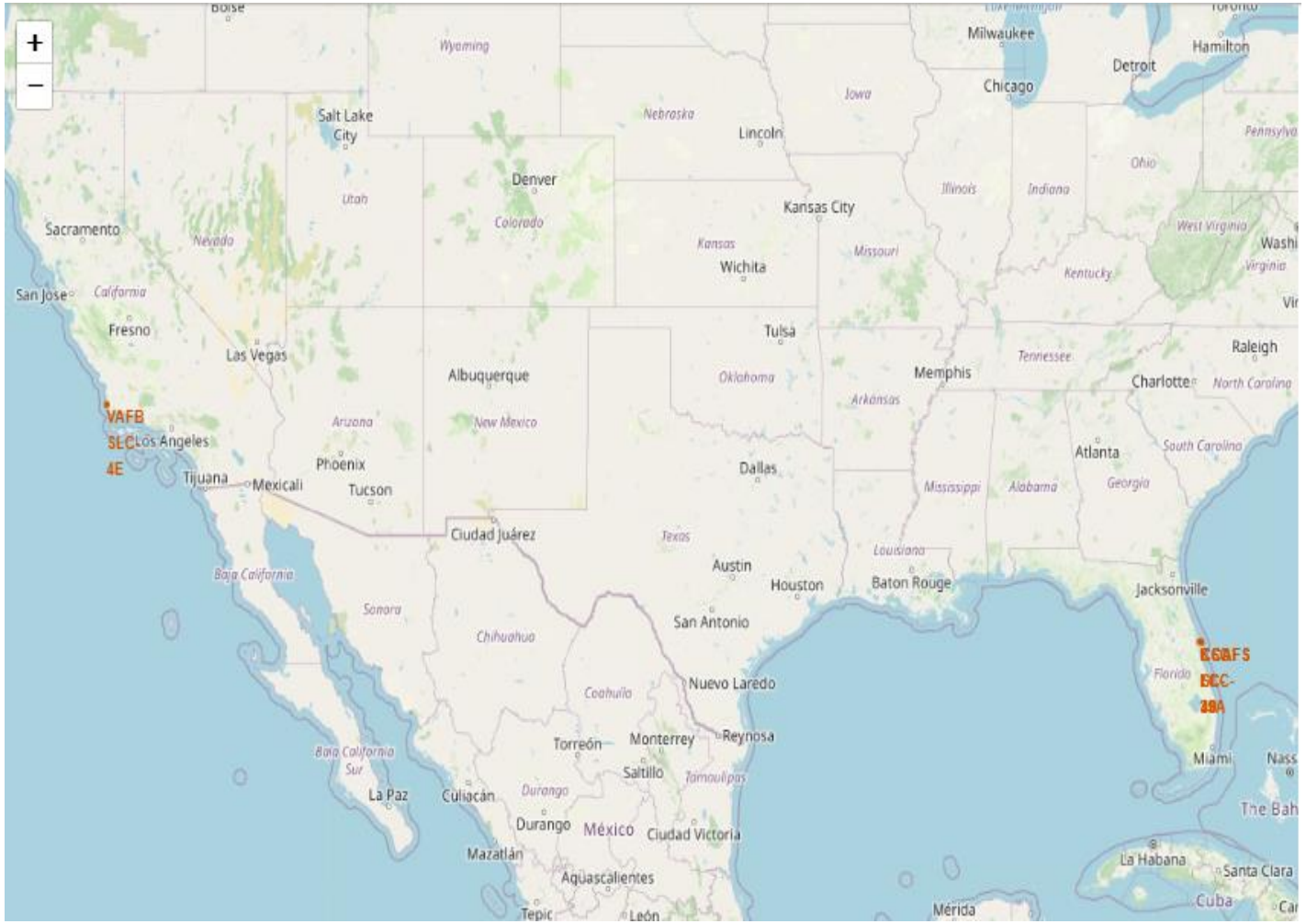
```
 * sqlite:///my_data1.db
Done.
```

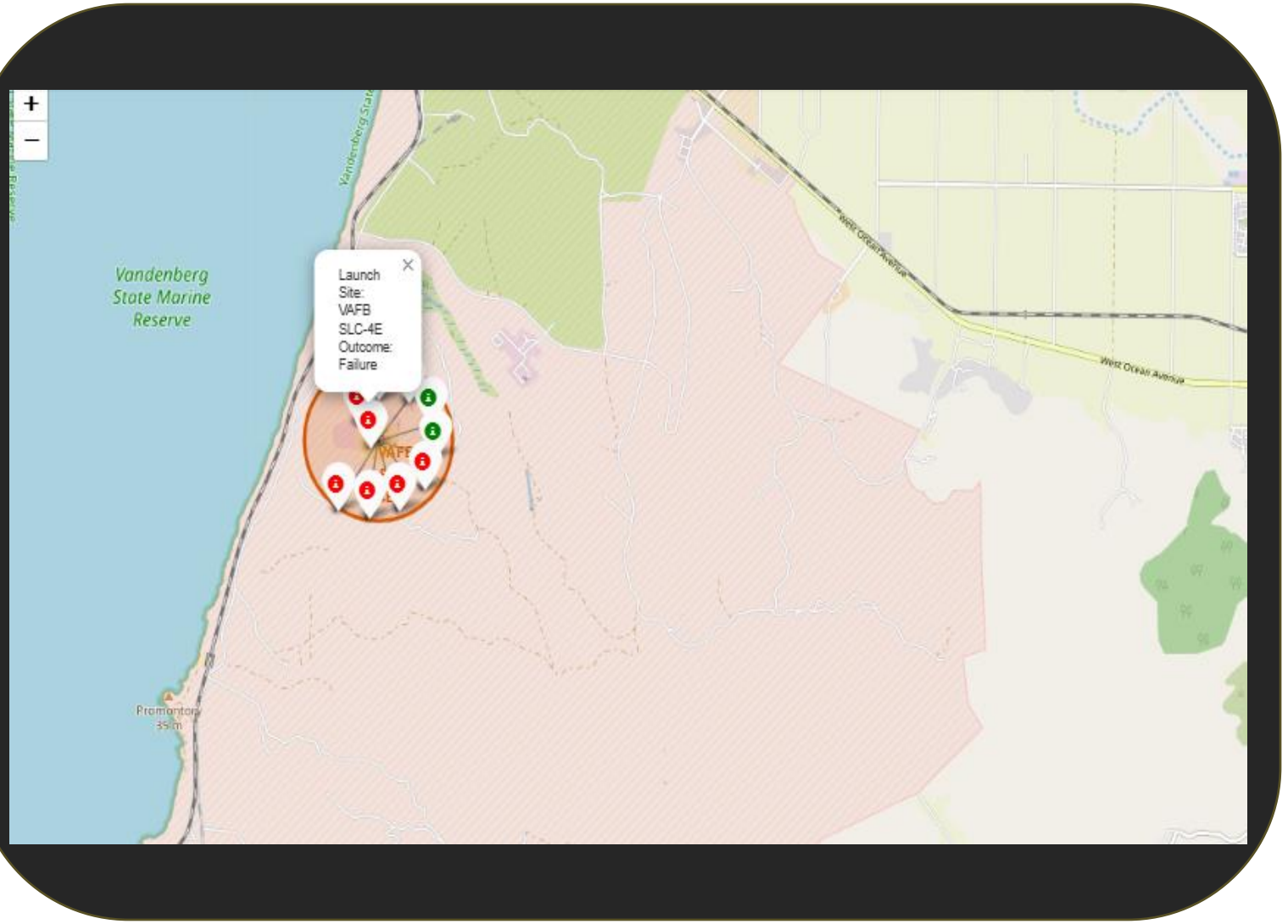
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

EDA WITH INTERACTIVE MAP USING FOLIUM

LAUNCH SITES' MARKERS ON A GLOBAL MAP:

- We applied Folium to show the exact locations of the launch pads
- Totally 3 locations were added to the map,
- We can see that the locations of the launch pads were positioned near the coastal area

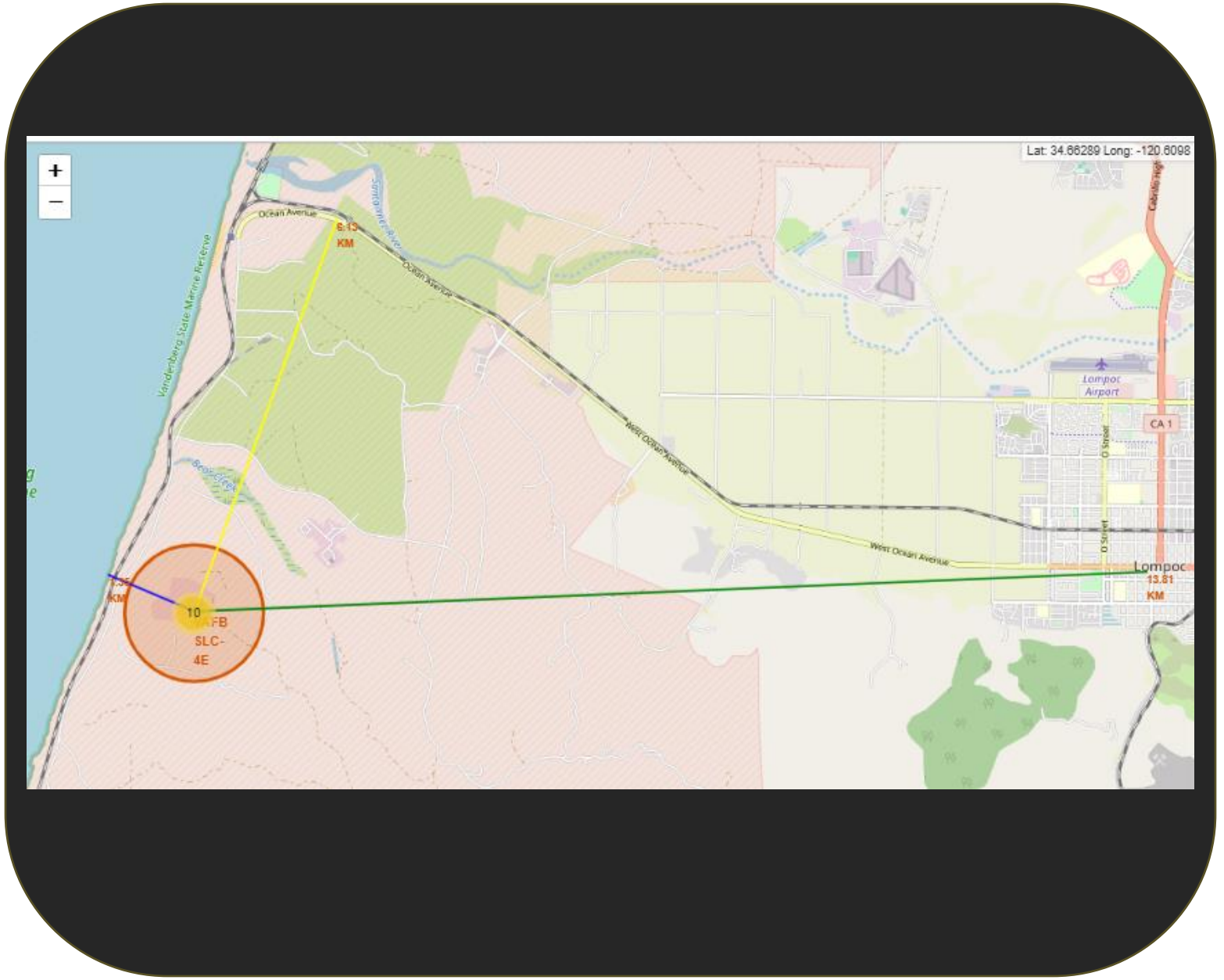




- Created markers for all launch records. If a launch was successful (`class=1`),
- Used a green marker and if a launch was failed, we use a red marker (`class=0`)

FOLIUM POSITIONING:

- Used folium to dram line to measure the distance between coastal line, nearest city, railway line

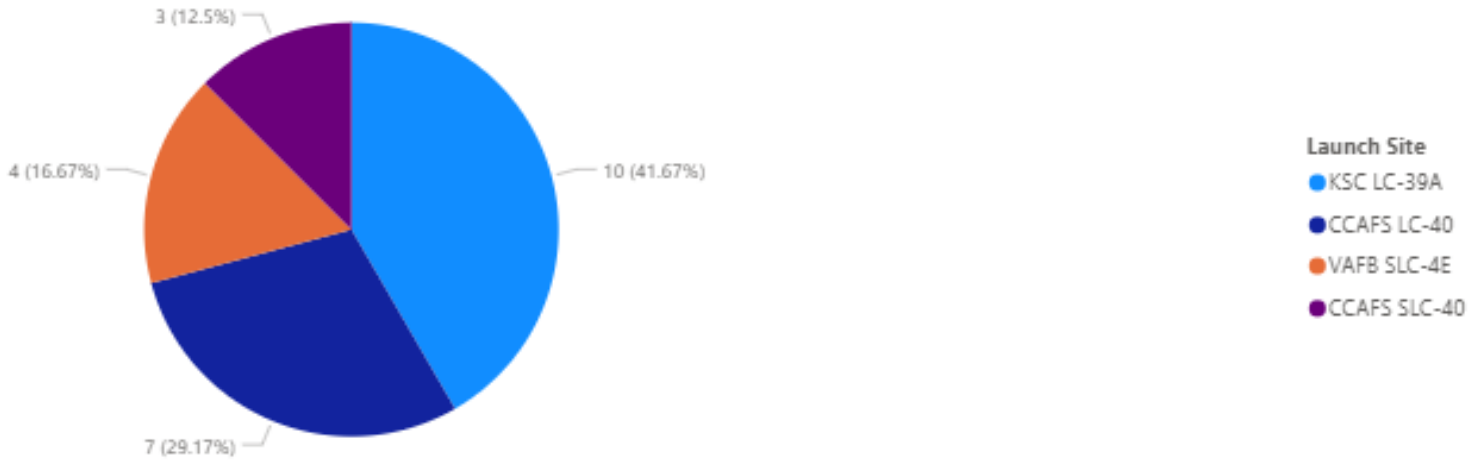



EDA WITH
INTERACTIVE DASHBOARD

LAUNCH SUCCESS COUNT FOR ALL SITES:

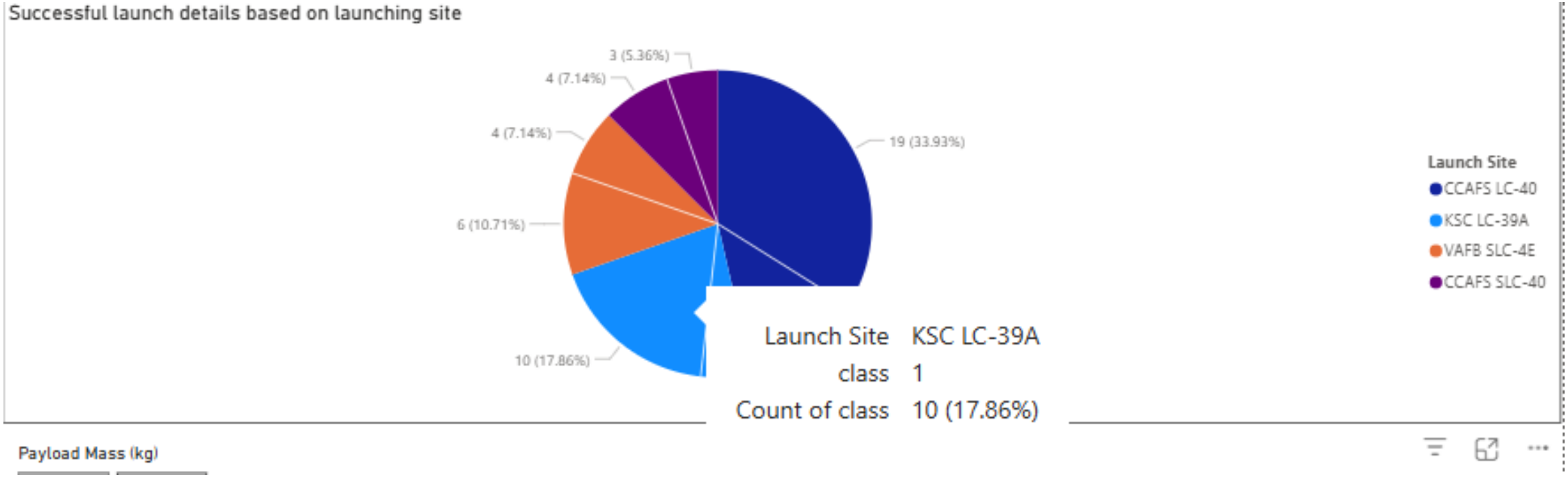
- Created an interactive dash board to display launch success count for all sites in a pie chart

Successful launch details based on launching site



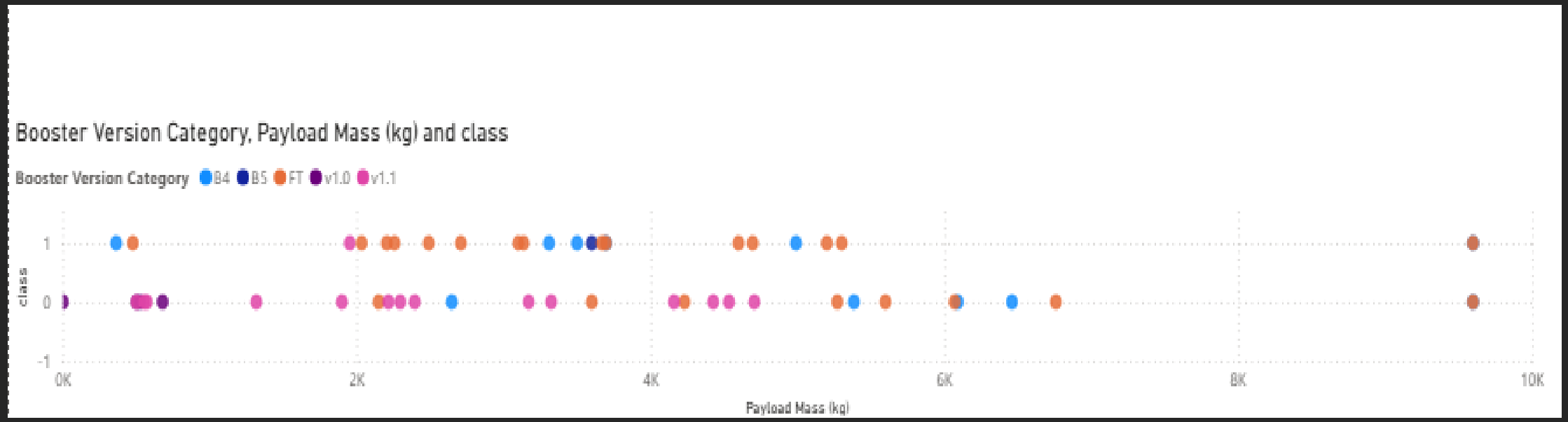
LAUNCH SITE WITH HIGHEST SUCCESS RATE:

- Using interactive dash board we found that launch site KSC LC-39A has highest success rate comparing to other launch sites



PAYLOAD VS. LAUNCH OUTCOME:

- Created a scatterplot to display launch outcomes plotted against payload mass, differentiated by booster versions



PREDICTIVE ANALYSIS USING MACHINE LEARNING

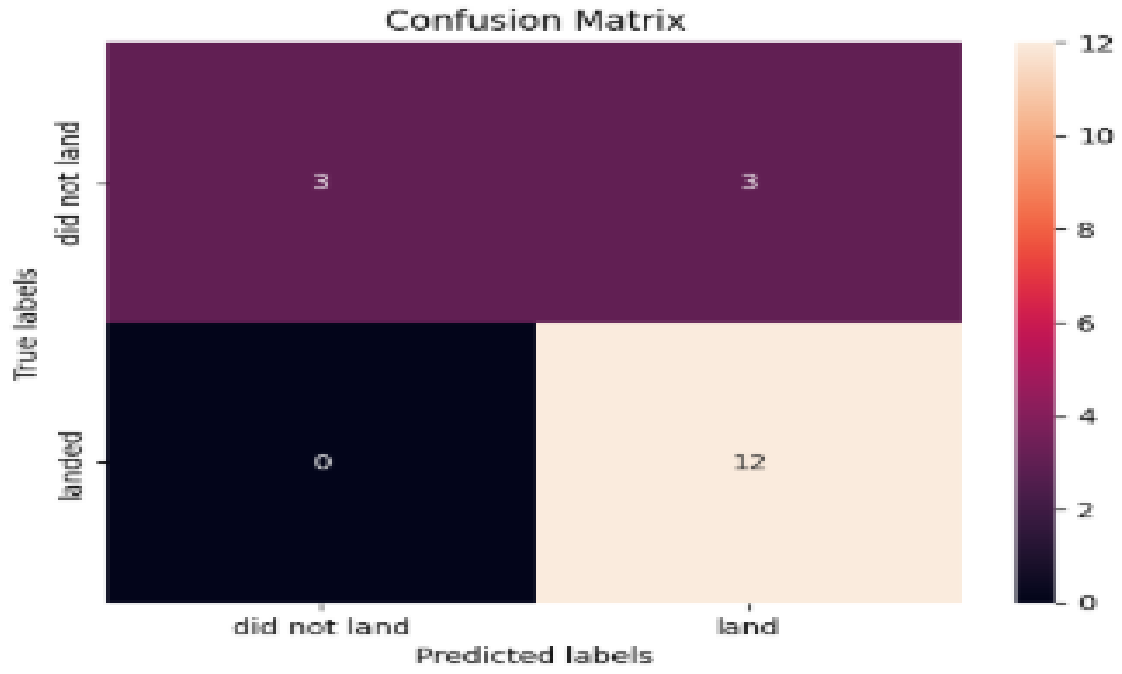
SUPPORT VECTOR MACHINE MODEL:

```
svm_test_accuracy = svm_cv.score(X_test, Y_test)
print("SVM Test Accuracy:", svm_test_accuracy)
```

```
SVM Test Accuracy: 0.8333333333333334
```

We can plot the confusion matrix

```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



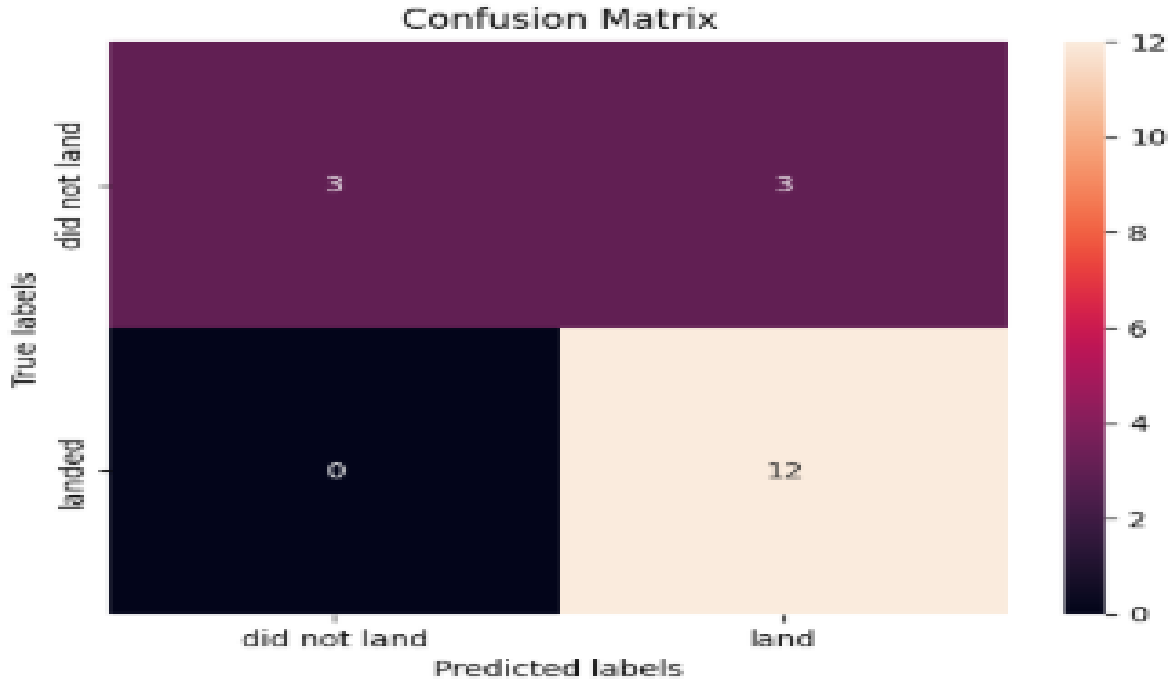
LOGISTIC REGRESSION MODEL:

```
test_accuracy = logreg_cv.score(X_test, Y_test)
print("Test Accuracy:", test_accuracy)
```

```
Test Accuracy: 0.8333333333333334
```

Lets look at the confusion matrix:

```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



DECISION TREE CLASSIFICATION MODEL:

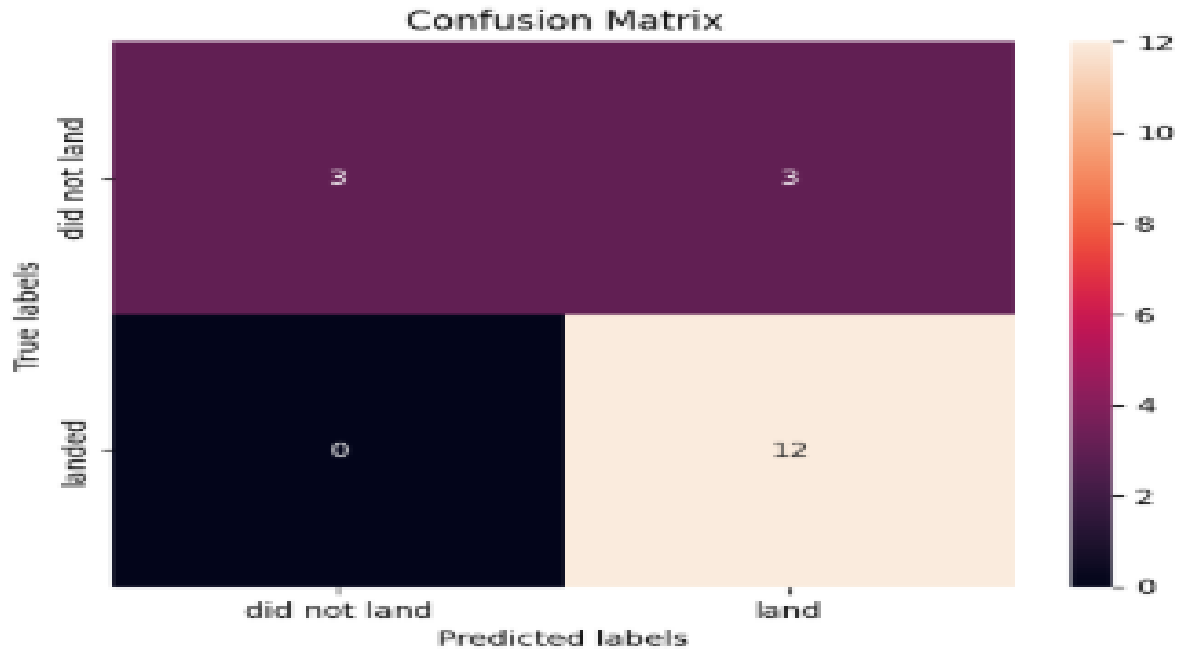
(BEST MODEL)

```
tree_test_accuracy = tree_cv.score(X_test, Y_test)
print("Decision Tree Test Accuracy:", tree_test_accuracy)
```

```
Decision Tree Test Accuracy: 0.8333333333333334
```

We can plot the confusion matrix

```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



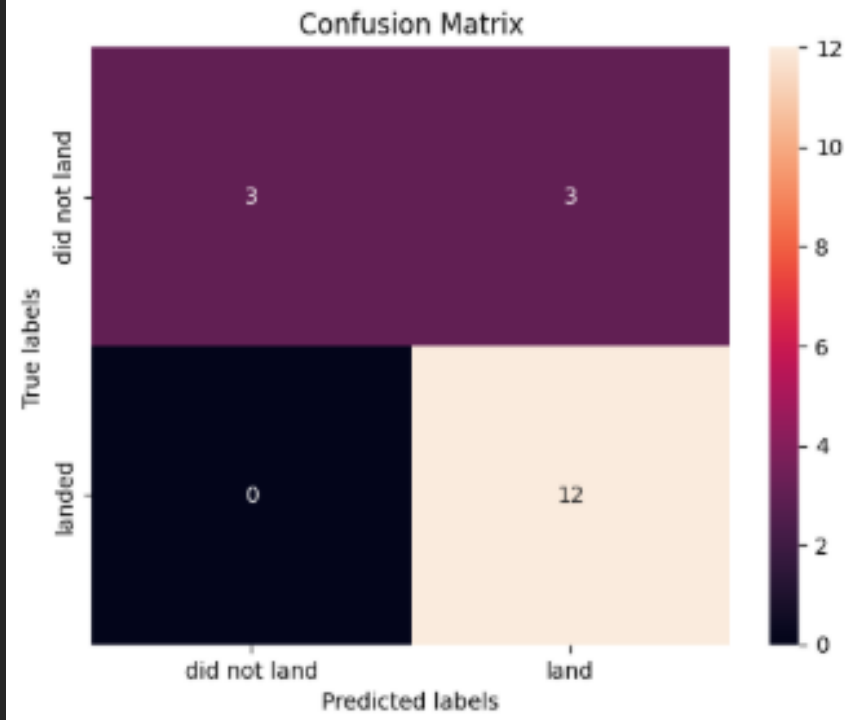
K NEAREST NEIGHBOURS MODEL:

```
knn_cv_accuracy = knn_cv.score(X_test, Y_test)
print("K nearest Neighbours Test Accuracy:", knn_cv_accuracy)
```

```
K nearest Neighbours Test Accuracy: 0.8333333333333334
```

We can plot the confusion matrix

```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



BEST MODEL:

- On performing analysis, we found that the test accuracy for all the model is around 83.3%
- However, on looking at training set accuracy, Decision tree has best accuracy of 88.75%
- So, based on this, we can conclude that Decision tree model is best out of all model

```
                 Model  Test Accuracy
0  Logistic Regression       0.833333
1                  SVM       0.833333
2        Decision Tree       0.833333
3                  KNN       0.833333
Best Model: Logistic Regression with accuracy: 0.8333333333333334
```

CONCLUSION:

- From the line graph we found that the success rate increased exponentially over time

- Launch site VABF SLC-4E has more success rate when compared to other sites with place

- Decision tree classification model performs slightly better than other models


THANK YOU !!!!