IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<G.UMA DEVI>
<15.02.2023>

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

i) What are the interactions among various features that determine landing success rates?

ii) How can the landing program be successful if certain conditions are met?

iii) What is predicting if the first stage of Space X Falcon 9 rocket will land successfully?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - ❖ Data was collected using SpaceX API
    - ❖ Web scraping from Wikipedia.
- Perform data wrangling
    - ❖ Dropping null columns
    - ❖ 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

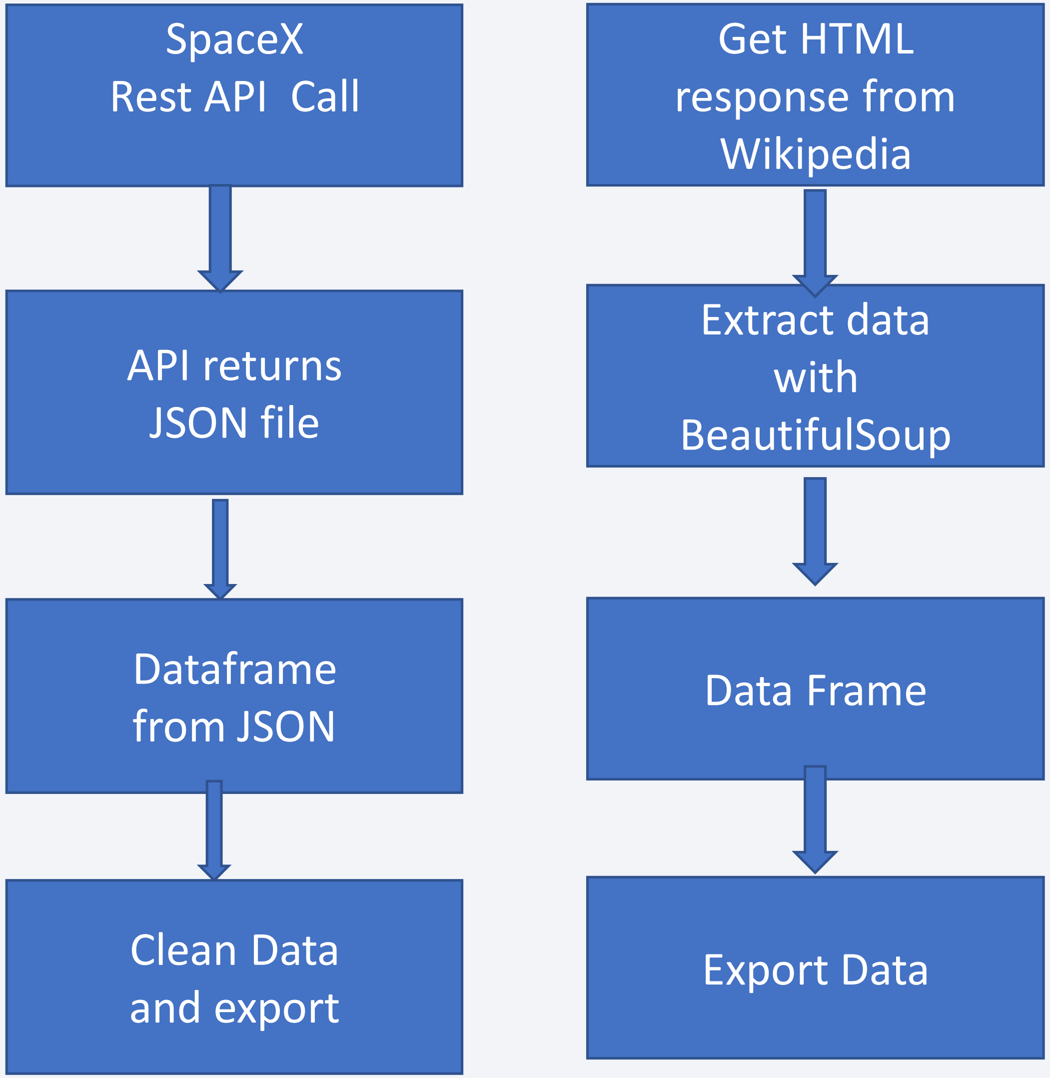
- The data was collected using various methods
- Data collection was done using get request to the SpaceX API.
- Next, we decoded the response content as a Json using .json() function call and turn itinto a pandas dataframe using .json_normalize().
- We then cleaned the data, checked for missing values 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

Datasets are collected from Rest SpaceX API and webscrapping Wikipedia. The information obtained by the API are rocket, launches, payload information. 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

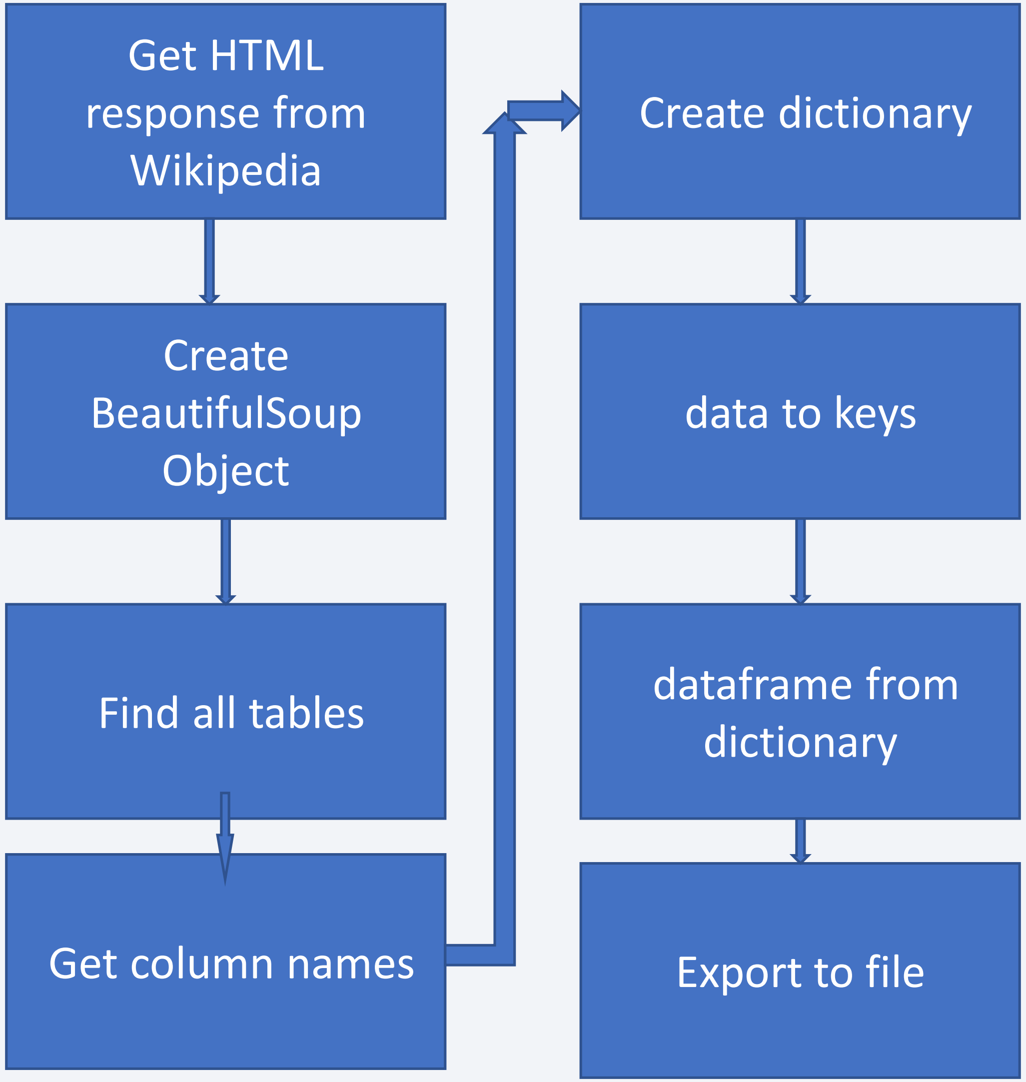
https://github.com/uma1310/Applied-Data-Science-Capstone.git/Applied-Data-Science-Capstone/blob/main/DataCollection_API.ipynb



Data Collection - Scraping

We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
We parsed the table and converted it into a pandas dataframe.

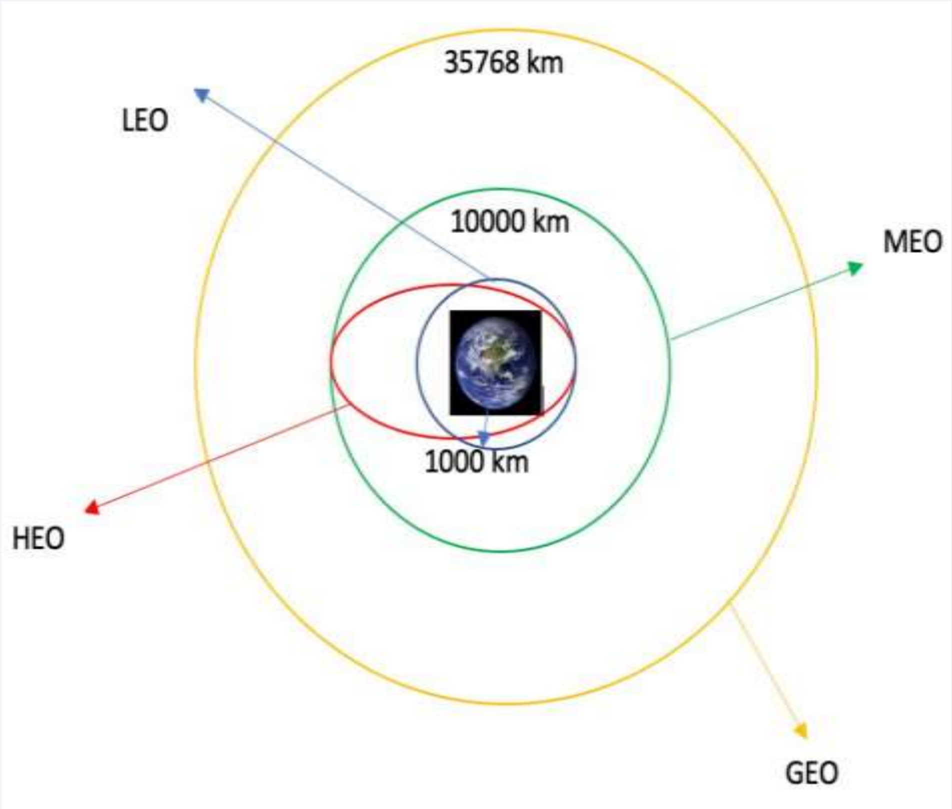
- The link to the notebook is https://github.com/uma1310/Applied-Data-Science-Capstone/blob/main/DataCollection_Webscraping.ipynb



Data Wrangling

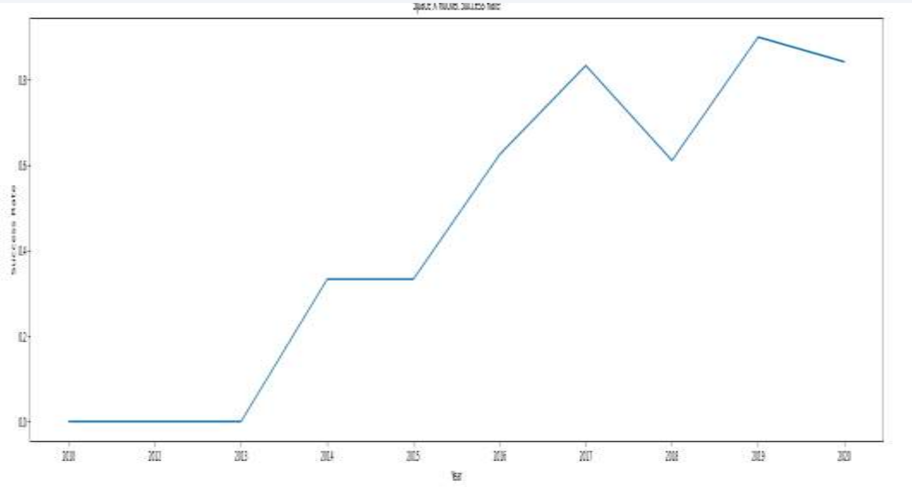
- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is

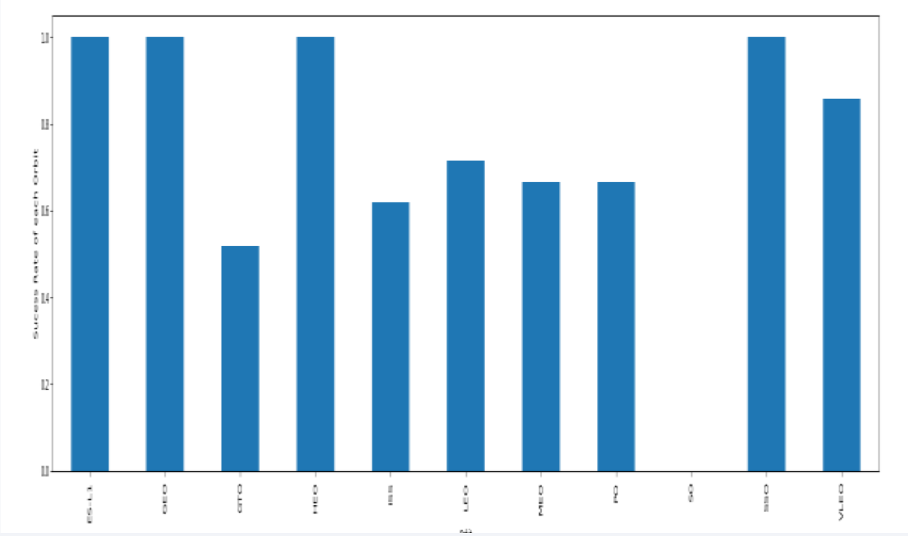
https://github.com/uma1310/Applied-Data-Science-Capstone/blob/main/DataWrangling_jupyterlite.ipynb



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/uma1310/Applied-Data-Science-Capstone/blob/main/EDA_DataVisualization.ipynb

EDA with SQL

- We performed SQL queries to gather and understand data from dataset:
- Displaying the names of the unique lauunch sites in the space mission.
- Display 5 records where launch sites begin with the string 'CCA'
- 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.
- List the records which will display the month names, faiilure landing_ouutcomes in drone ship, booster versions, launch_site for the months in year 2015.
- Rank the count of successful landiing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

The link to the notebook is **https://github.com/uma1310/Applied-Data-Science-Capstone/blob/main/Eda-SQL.ipynb**

Build an Interactive Map with Folium

- Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas
- Red circle at NASA Johnson Space Center's coordinate with label showing its name (folium.Circle, folium.map.Marker).
- Red circles at each launch site coordinates with label showing launch site name (folium.Circle, folium.map.Marker, folium.features.DivIcon).
- The grouping of points in a cluster to display multiple and different information for the same coordinates (folium.plugins.MarkerCluster).
- Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing.(folium.map.Marker, folium.Icon).
- Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them. (folium.map.Marker, folium.PolyLine, folium.features.DivIcon)
- These objects are created in order to understand better the problem and the data.

The link to the notebook is **https://github.com/uma1310/Applied-Data-Science-Capstone/blob/main/Launch_Site_Location.ipynb**

Build a Dashboard with Plotly Dash

- Dashboard has dropdown, pie chart, rangeslider and scatter plot components
- Dropdown allows a user to choose the launch site or all launch sites (dash_core_components.Dropdown).
- Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component.
- Rangeslider allows a user to select a payload mass in a fixed range.
- Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass.
- The link to the notebook is **https://github.com/uma1310/Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py**

Predictive Analysis (Classification)

- Data Preparation
  - ➢ loaded the data using numpy and pandas,
  - ➢ transformed the data,
  - ➢ split our data into training and testing.
- Built different machine learning models and tune different hyperparameters using GridSearchCV.
- Model Evaluation as the metric for our model, improved the model using feature engineering and algorithm tuning. and Plot Confusion Matrix
- Model comparison
  - ➢ Comparison of models according to their accuracy
  - ➢ The model with the best accuracy

The link to the notebook is https://github.com/uma1310/Applied-Data-Science-Capstone/blob/main/Machine_Learning_Prediction.ipynb

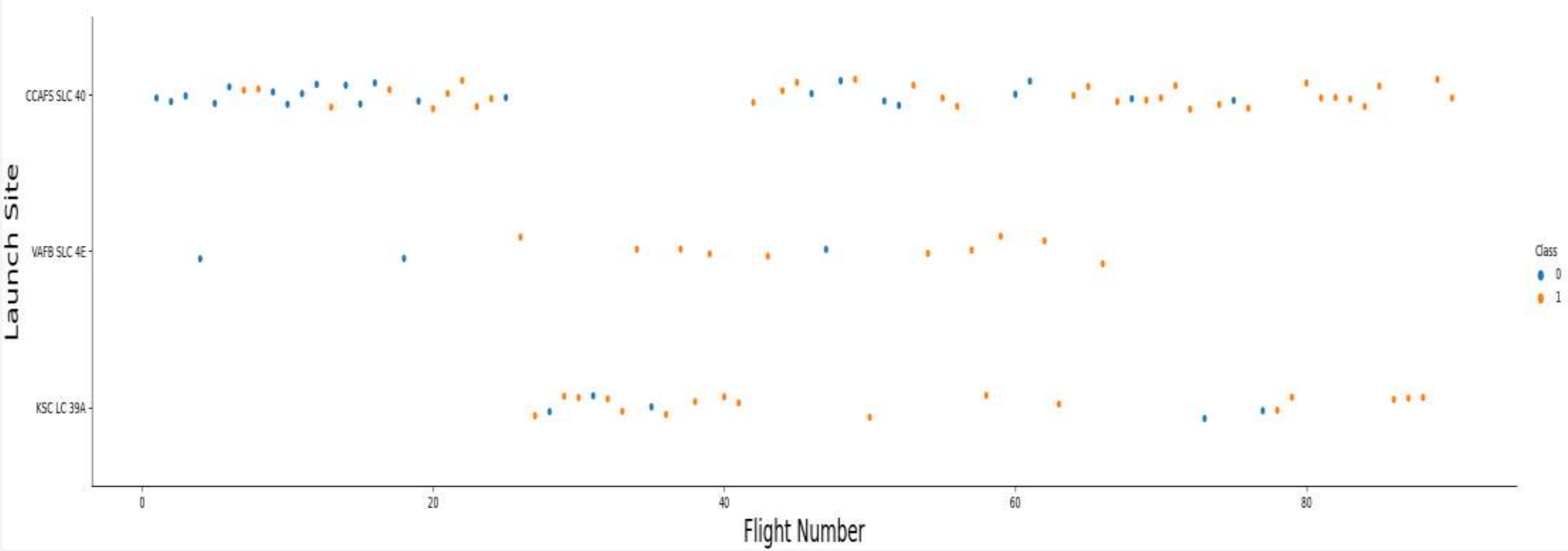
Results

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

Section 2

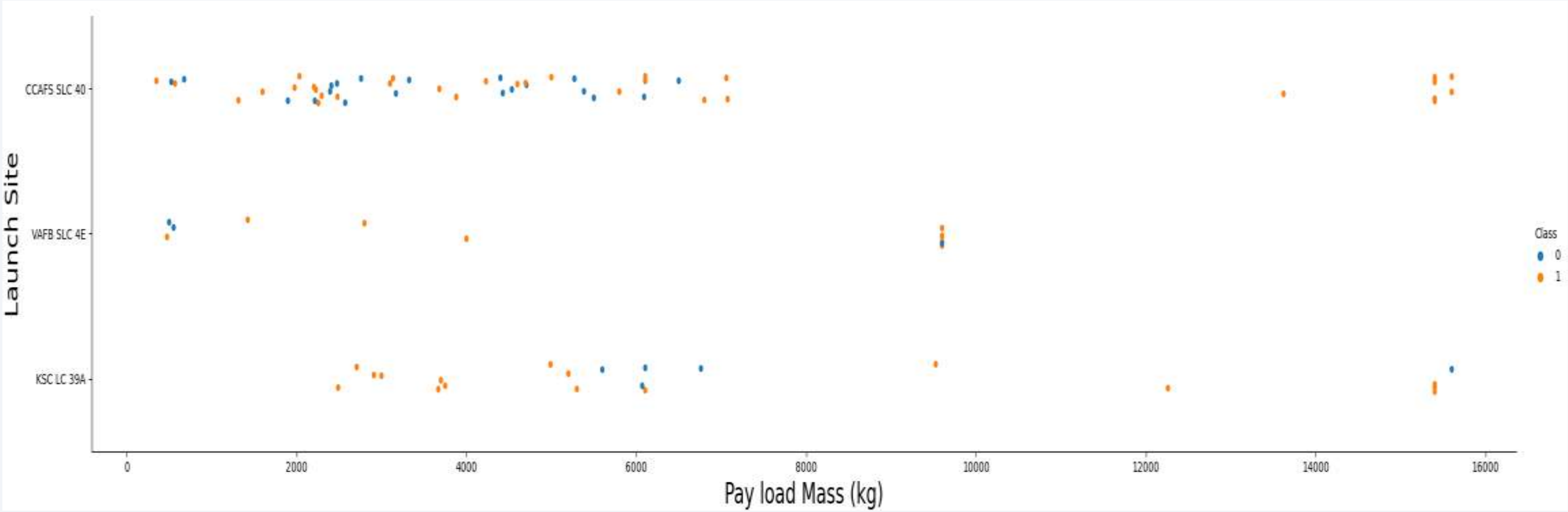
Insights drawn from EDA

Flight Number vs. Launch Site



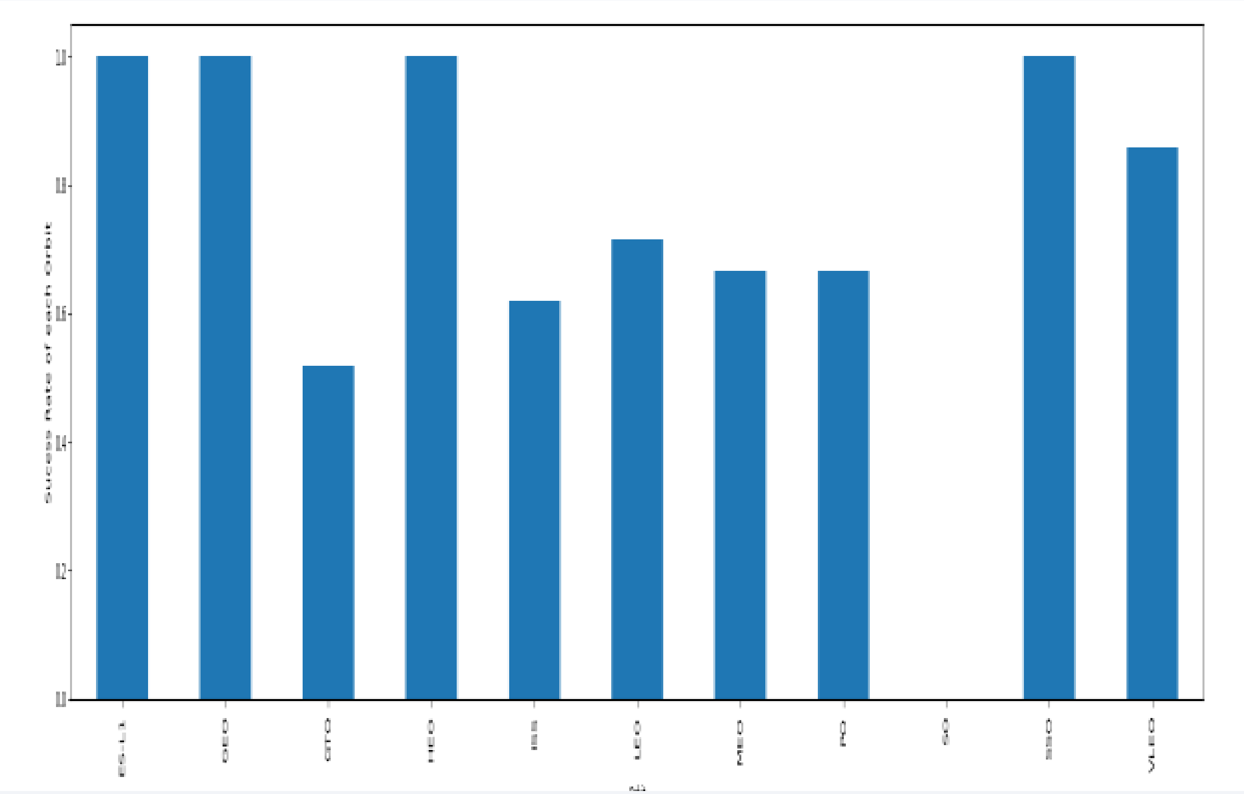
The plot, shows the larger the flight amount at a launch site, the greater the success rate at a launch site.

Payload vs. Launch Site



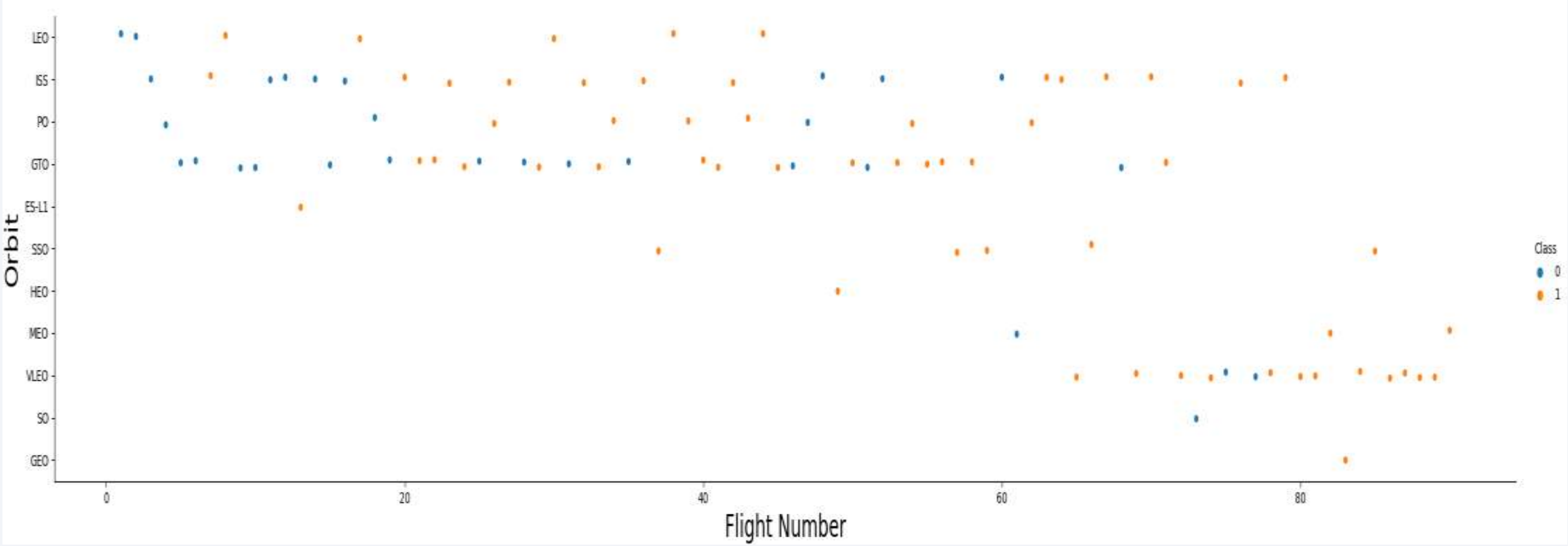
The plot shows depending on the launch site, a heavier payload may be a consideration for a successful landing. If a too heavy payload can make a landing fail.

Success Rate vs. Orbit Type



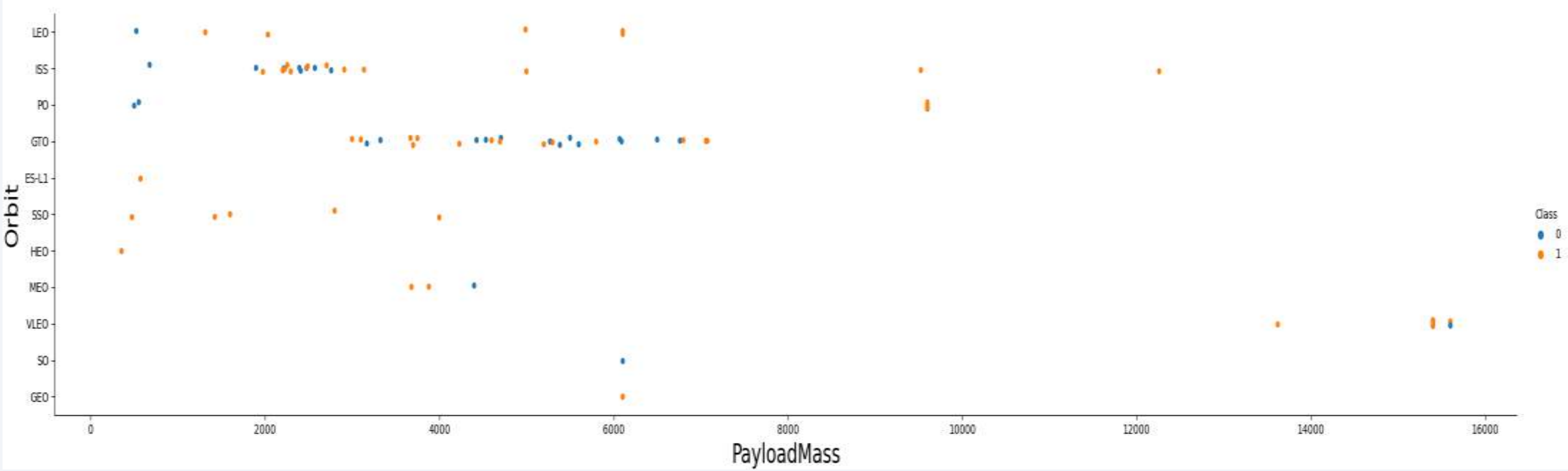
- This plot shows the success rate for different orbit types. We note that ES-L1, GEO, HEO, SSO have the

Flight Number vs. Orbit Type



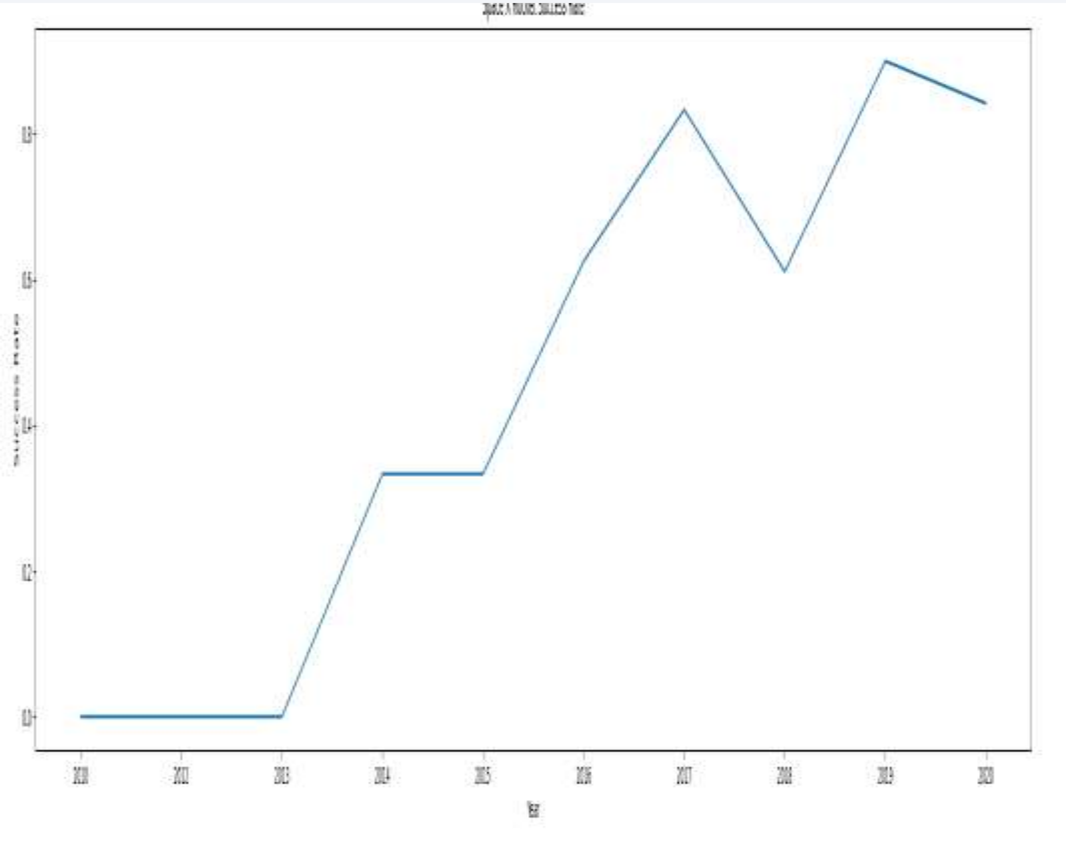
This plot shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.

Payload vs. Orbit Type



The weight of the payloads can have a great influence on the success rate of the launches in certain orbits. That with heavy payloads, the successful landing are more for PO, LEO and ISS orbits

Launch Success Yearly Trend



This plot, observe that success rate

- since 2013 kept on increasing till 2020.

All Launch Site Names

SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL

Task 1

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

In [6]:
```
%sql SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

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

Out[6]:

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

Explanation

The use of **DISTINCT** in the query allows to remove duplicate LAUNCH_SITE.

Launch Site Names Begin with 'CCA'

SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5

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

In [7]:
```
%sql SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5
```

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

Out[7]:

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

Explanation

The WHERE clause followed by LIKE clause filters launch sites that contain the substring CCA. LIMIT 5 shows 5 records from filtering

Total Payload Mass

SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'

```
In [8]: %sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'

 * sqlite:///my_data1.db
Done.

Out[8]: SUM("PAYLOAD_MASS_KG_")
                          45596
```

Explanation

This query returns the sum of all payload masses where the customer is NASA (CRS)

Average Payload Mass by F9 v1.1

SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'

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

```
In [29]: %sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'

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

Out[29]:

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

Explanation

This query returns the average of all payload masses where the booster version contains the substring F9 v1.1.

First Successful Ground Landing Date

SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'

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

Hint:Use min function

```
In [30]: %sql SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'

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

Out[30]:

MIN("DATE")
01-05-2017

Explanation

With this query, we select the oldest successful landing.The WHERE clause filters dataset in order to keep only records where landing was successful. With the MIN function.

Successful Drone Ship Landing with Payload between 4000 and 6000

SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (drone ship)' \
AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 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;
```

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

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

Explanation

- This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg. The WHERE and AND clauses filter the dataset.

Total Number of Successful and Failure Mission Outcomes

SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE

List the total number of successful and failure mission outcomes

In [12]:

```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, \
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

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

Out[12]:

SUCCESS	FAILURE
100	1

Explanation

The first subquery counts the successful mission. The second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered.

Boosters Carried Maximum Payload

SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)

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

In [13]:
```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

Out[13]:

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

Explanation

A subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version with the heaviest payload mass.

2015 Launch Records

SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'

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, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

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

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

Explanation

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

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

SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%' GROUP BY "LANDING _OUTCOME" ORDER BY COUNT("LANDING _OUTCOME") DESC ;

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

In [15]:

```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

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

Out[15]:

Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

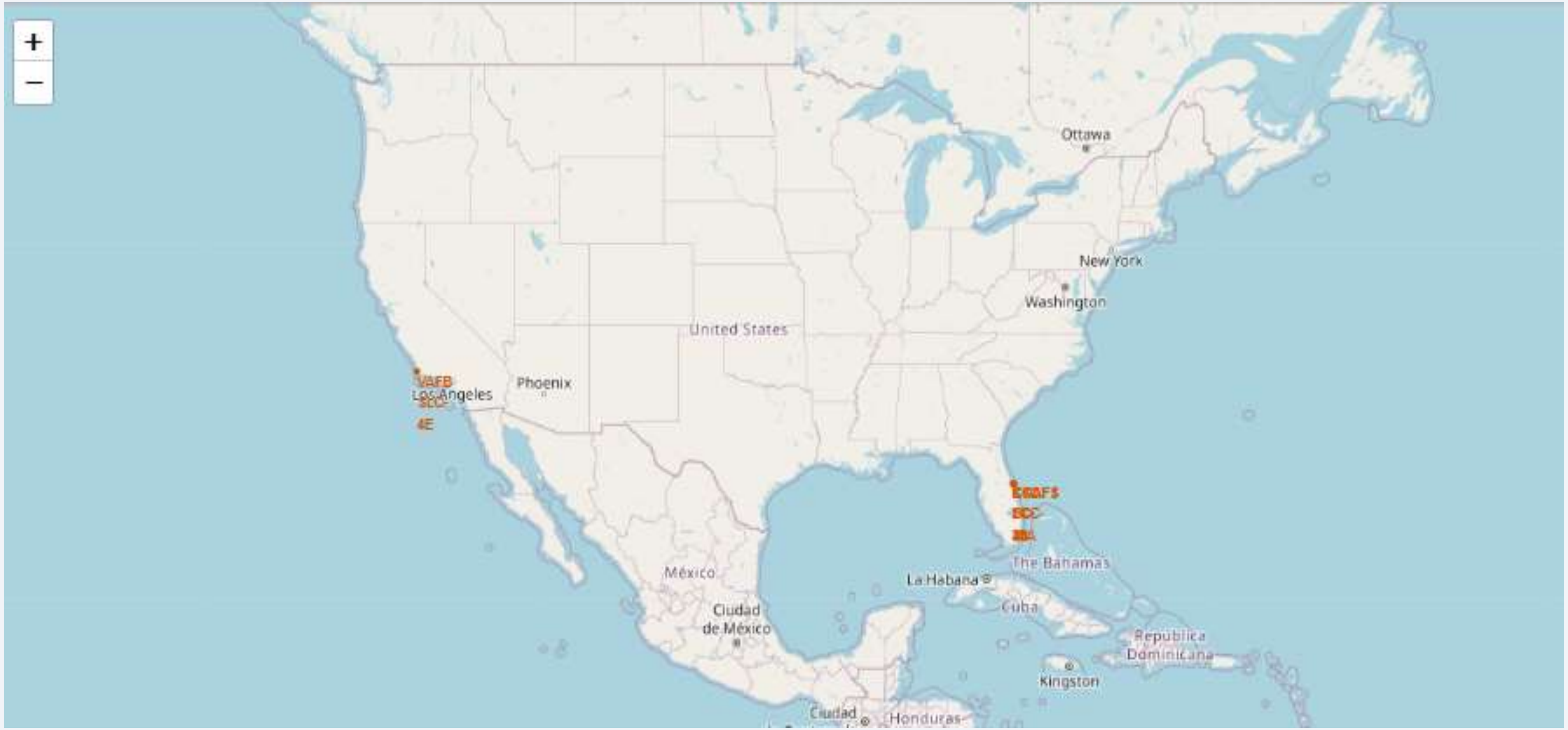
Explanation

This query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

Section 3

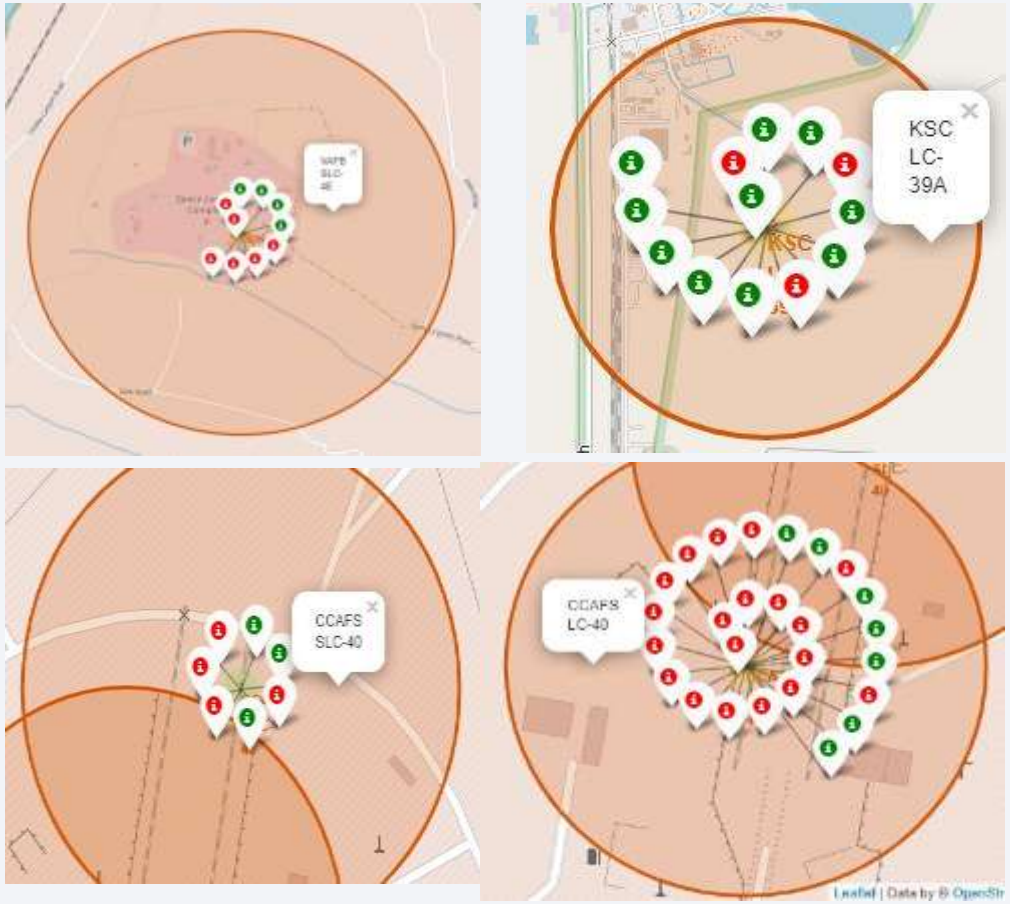
Launch Sites Proximities Analysis

Folium Map – Ground Stations



Space X launch sites are located on the coast of the United States

Folium Map – Color Labeled Markers(Red and Green)

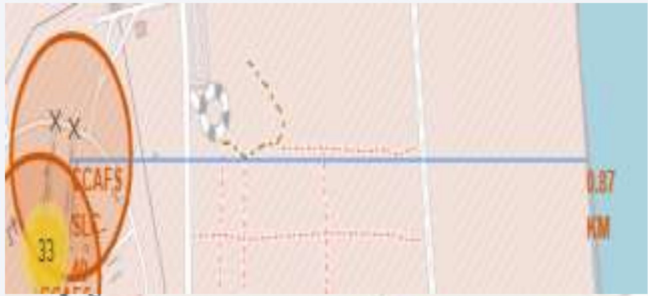


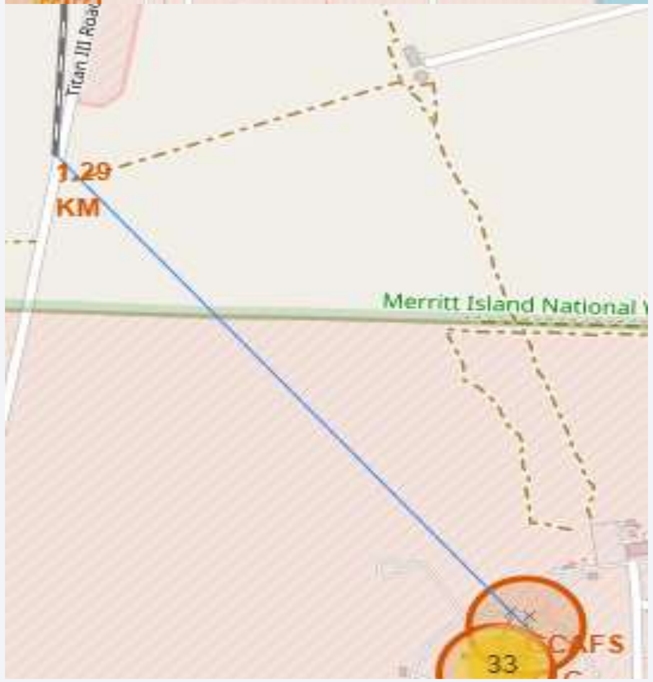
Green marker represents successful launches.

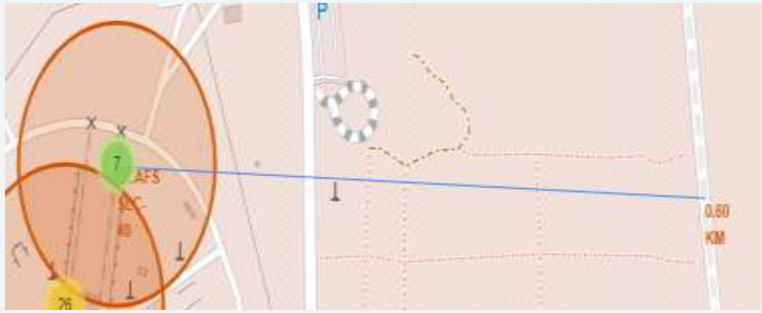
Red marker represents unsuccessful launches.

KSC LC-39A has a higher launch success rate.

Folium Map – Distances between CCAFS SLC-40 and proximities







Is CCAFS SLC-40 in close proximity to railways ? Yes

Is CCAFS SLC-40 in close proximity to highways ? Yes

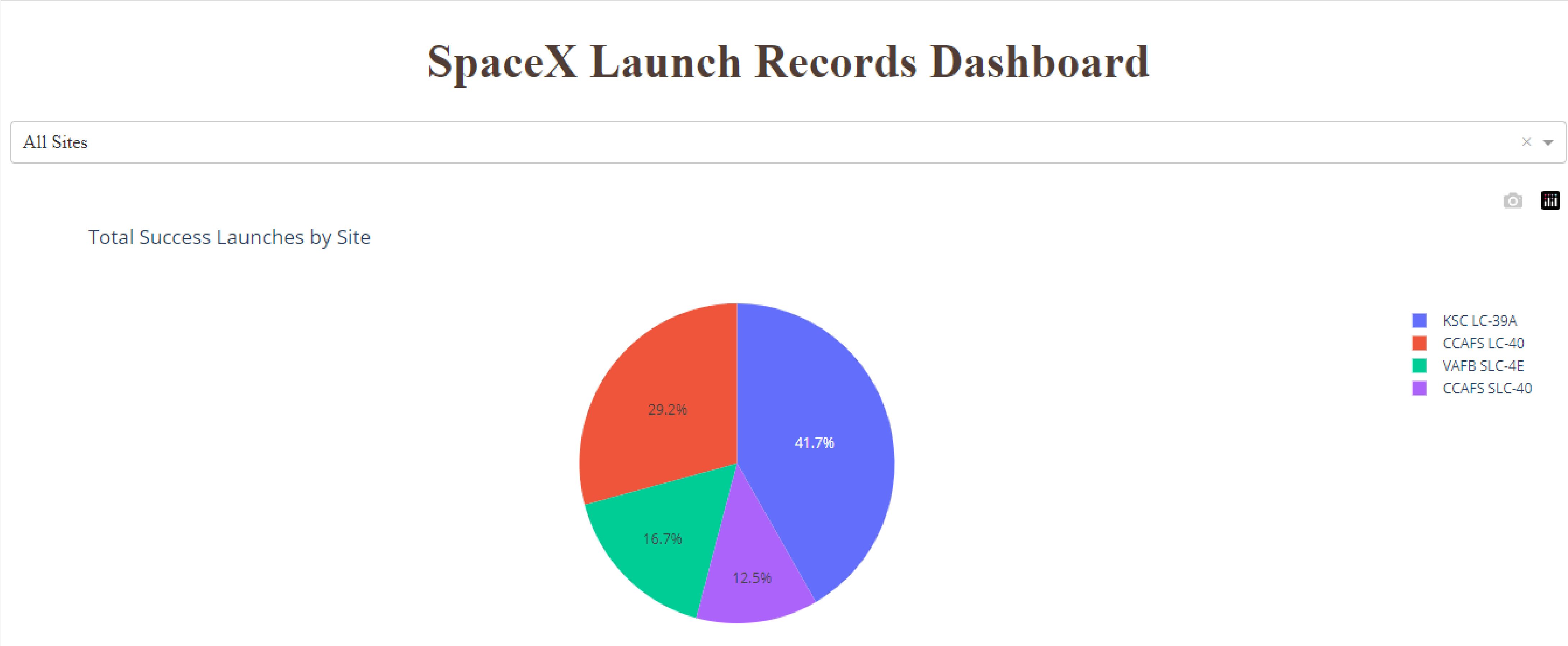
Is CCAFS SLC-40 in close proximity to coastline ? Yes

Do CCAFS SLC-40 keeps certain distance away from cities ? No

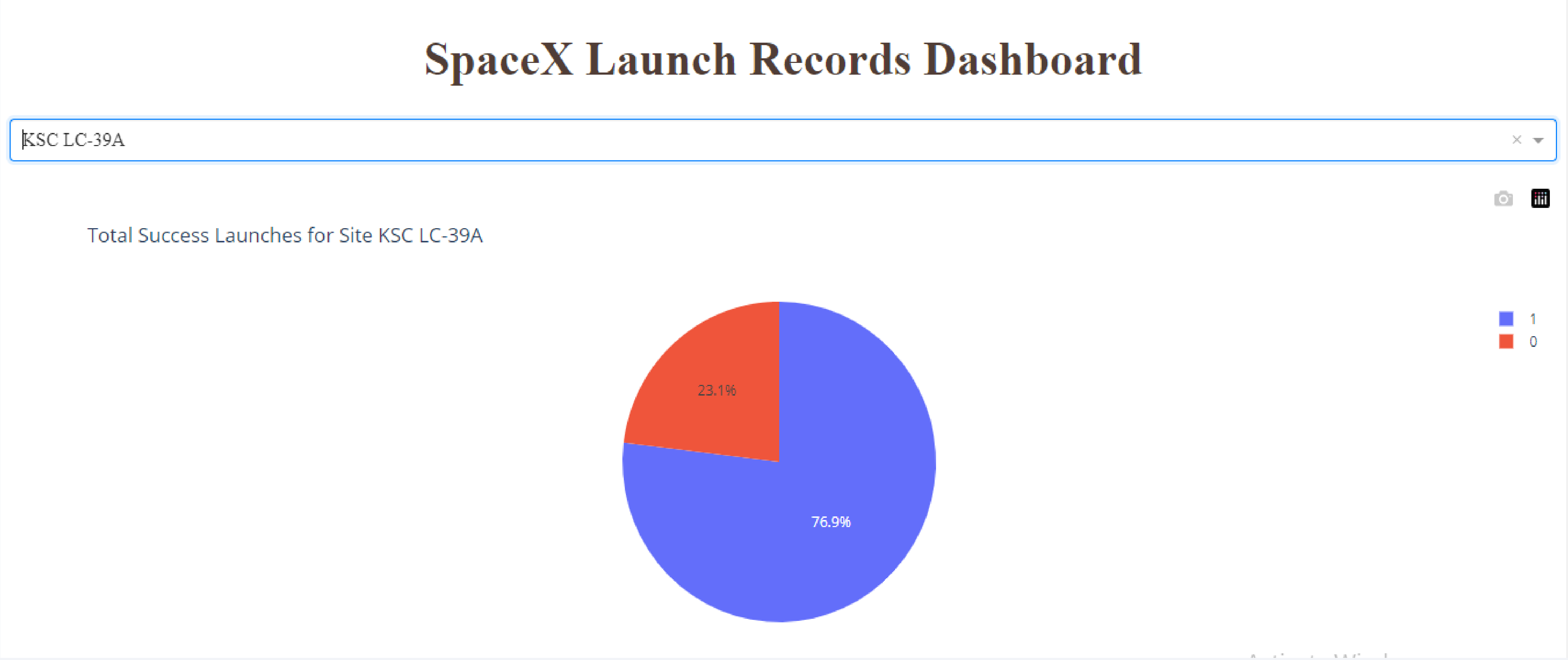
Section 4

Build a Dashboard with Plotly Dash

Dashboard – Total success by Site



Dashboard – Total success launches for Site KSC LC-39A

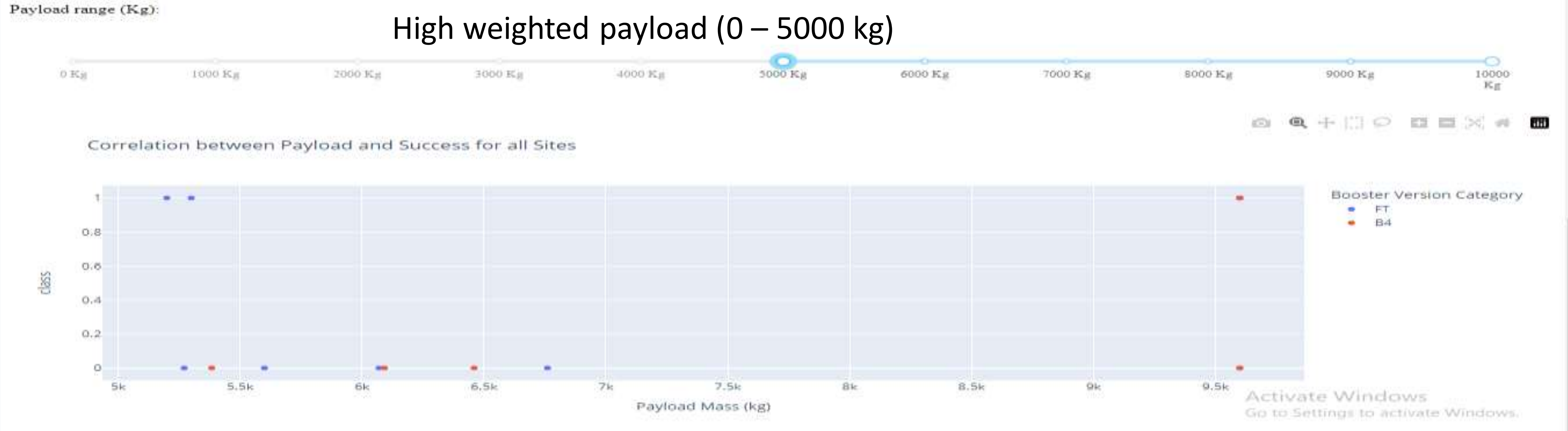


76.9% success rate and 23.1% failure rate

Dashboard – Payload mass vs Outcome for all sites with different payload mass

Low weighted payload (0 – 5000 kg)

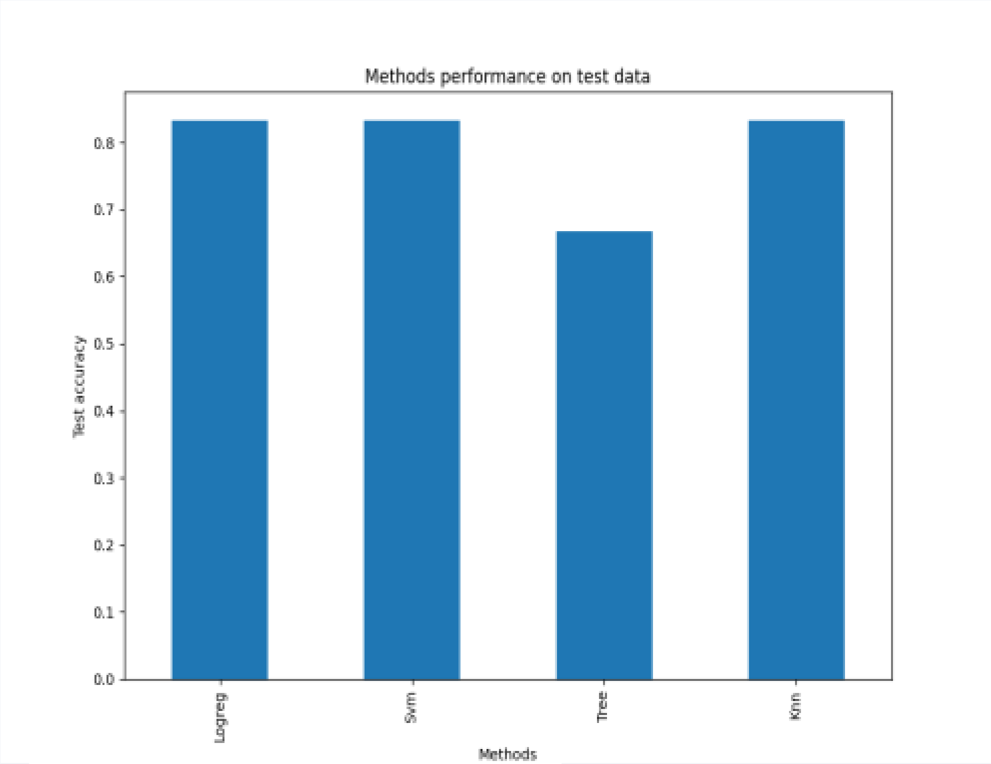
High weighted payload (0 – 5000 kg)

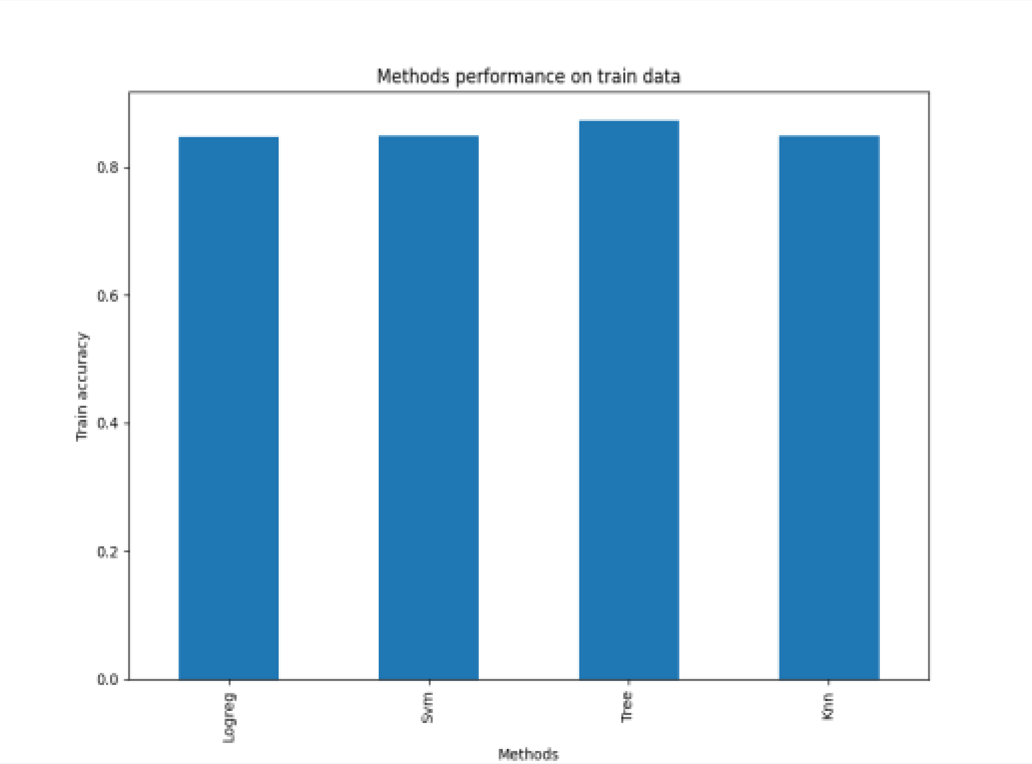


Section 5

Predictive Analysis (Classification)

Classification Accuracy



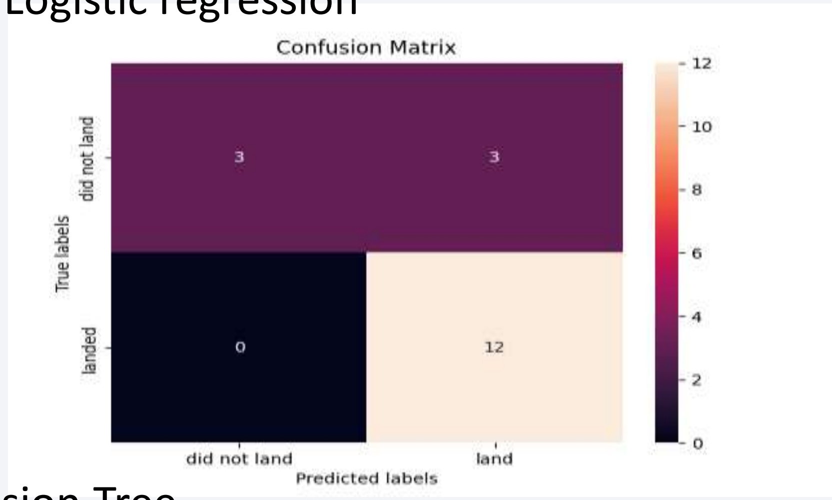


	Accuracy Train	Accuracy Test
Tree	0.873214	0.666667
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333

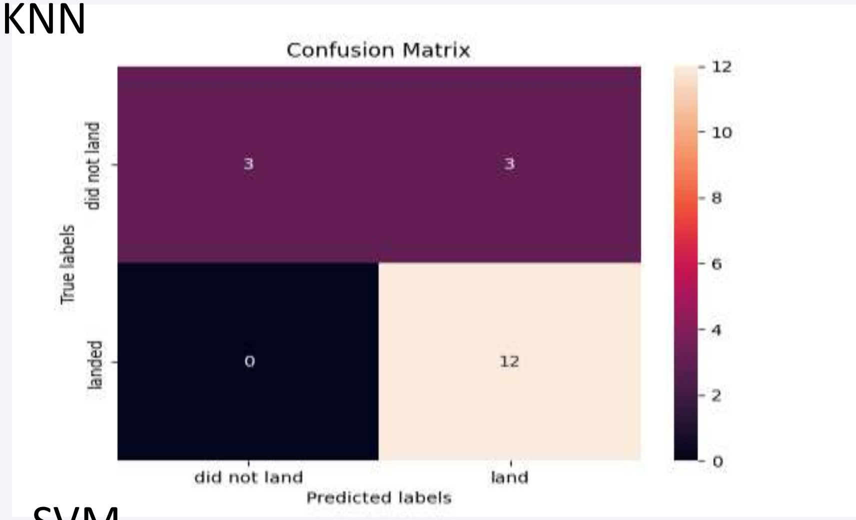
```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 8, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2,
'splitter': 'random'}
accuracy : 0.8732142857142857
```

Confusion Matrix

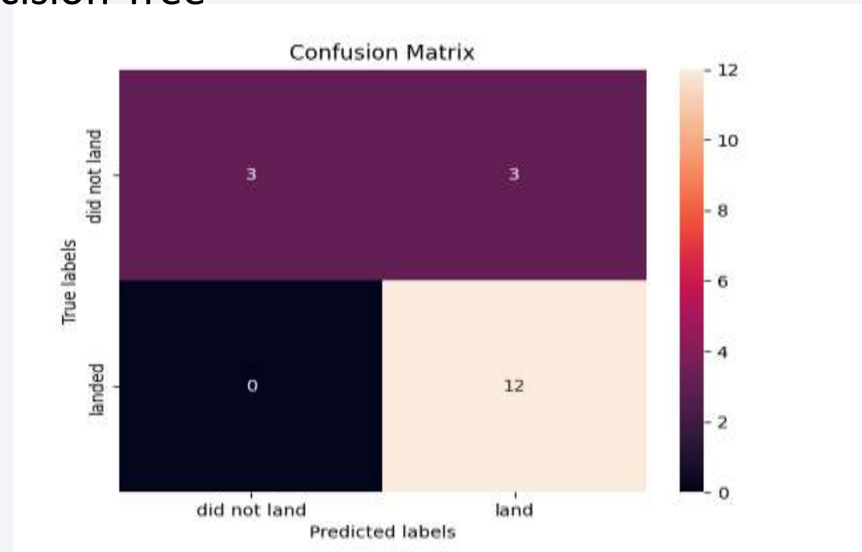
Logistic regression



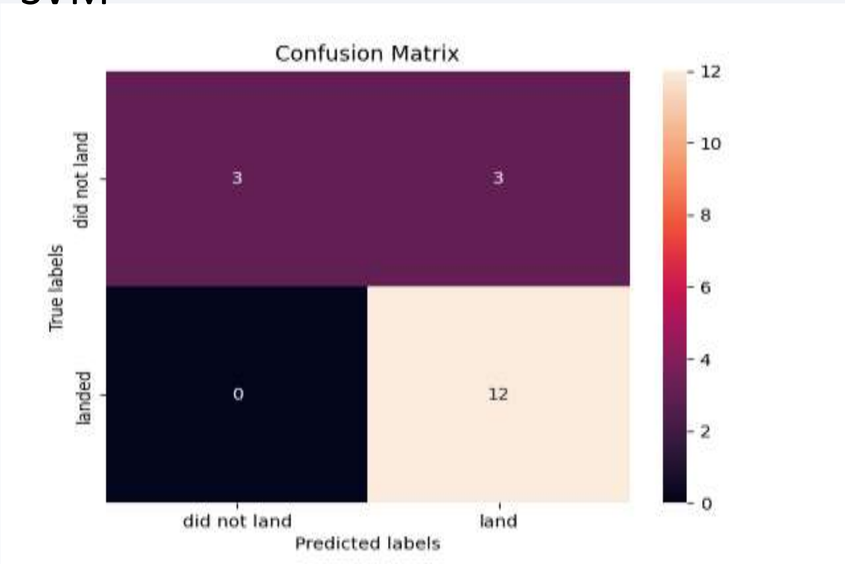
KNN



Decision Tree



SVM



The test accuracy are all equal, the confusion matrices are also identical.

Conclusions

- Several factors influence mission success, including the launch site, orbit, and, most importantly, the number of previous launches. As a result, we can assume that we gained knowledge between launches that enabled us to transition from a failed launch to a successful launch.
- GEO, HEO, SSO, and ES-L1 orbits have the highest success rates. Based on the orbit, the mass of the payload can be a success criterion. The payload mass can be light or heavy depending on the orbit. However, low weighted payloads outperform heavy weighted payloads in general.
- We can't explain why some launch sites are better than others based on the data we have right now (KSC LC-39A is the best launch site). We could obtain atmospheric or other relevant data to find an answer to this problem.
- In this dataset, we select the Decision Tree Algorithm as the best model even though the test accuracy of all models is identical. The Decision Tree Algorithm was chosen because it has a higher train accuracy.

Appendix

https://github.com/uma1310/Applied-Data-Science-Capstone

Thank you!