IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mohana Priya Eagambaram
Feb-2-20124

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
- -Build Dashboard with Plotly Dash
- -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

1. What factors determine if the rocket will land successfully?
2. The interaction amongst various features that determine the success rate of a successful landing.
3. 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

Data wrangling is the process of transforming and structuring data from one raw form into standard format. Here one hot encoding was applied to categorical features to proceed further data analysis.

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

To build machine learning pipeline we preprocess the data and splitting into train test datasets for hyperparameter. Using hyperparameter we will test models like Logistic Regression, Support Vector Machine, Decision tree classifier and K nearest Neighbour for Accuracy.

Data Collection

Data was collected using SpaceX API and web scraping from Wikipedia.

- Here we used the get request to the SpaceX API to collect data,
- Clean the requested data and
- Did some basic data wrangling and formatting.

Data Collection – SpaceX API

We used get request to SpaceX API and data is cleaned and formatted using data wrangling methods.

GITHUB link:

Data_Collection_SpaceX_API

1. Get request for rocket launch data using API

2. Convert Json result into a Dataframes using Json normalize method.

3. Data Cleaning or formatting

4. Filling missing values in data frames.

Data Collection - Scraping

Webscraping is performed to collect Falcon 9 historical launch records from Wikipedia using Beautifulsoup. The process involved in webscraping are

- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame
- https://github.com/Mohanapriya-146/IBM_DataScience_capstone_SpaceX/blob/main/jupyter-labs-webscraping%20(1).ipynb

1. Perform an HTTP GET method to request the Falcon9 Launch HTML page
2. Create a BeautifulSoup object from the HTML response
3. Collect all relevant column names from the HTML table header
4. Create an empty dictionary with keys from the extracted column names
5. Create an empty dictionary with keys from the extracted column names
6. Fill up the launch dictionary with launch records extracted from table rows.
7. Parsing the Launch HTML tables.
8. Dictionary will be converted into a Pandas dataframe and exported into csv file.

Data Wrangling

Exploratory Data Analysis (EDA) is performed to find some patterns in the data and determine what would be the label for training supervised models.

Calculate the number of launches on each site.

↓

Calculate the number and occu

↓

Calculate the number and occurence of mission outcome per orbit type using value_counts method

↓

Create a landing outcome label from Outcome column using landing class variable.

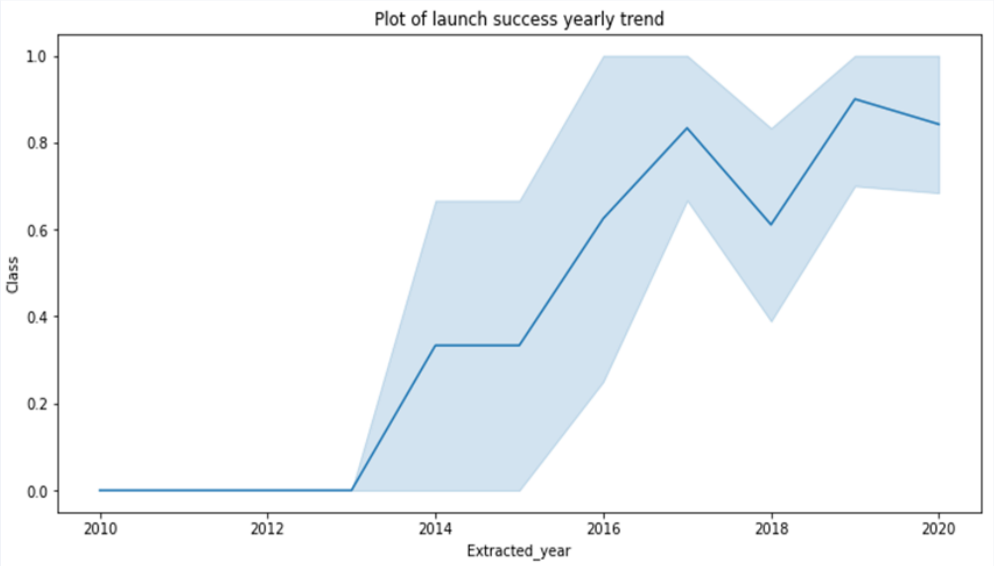
↓

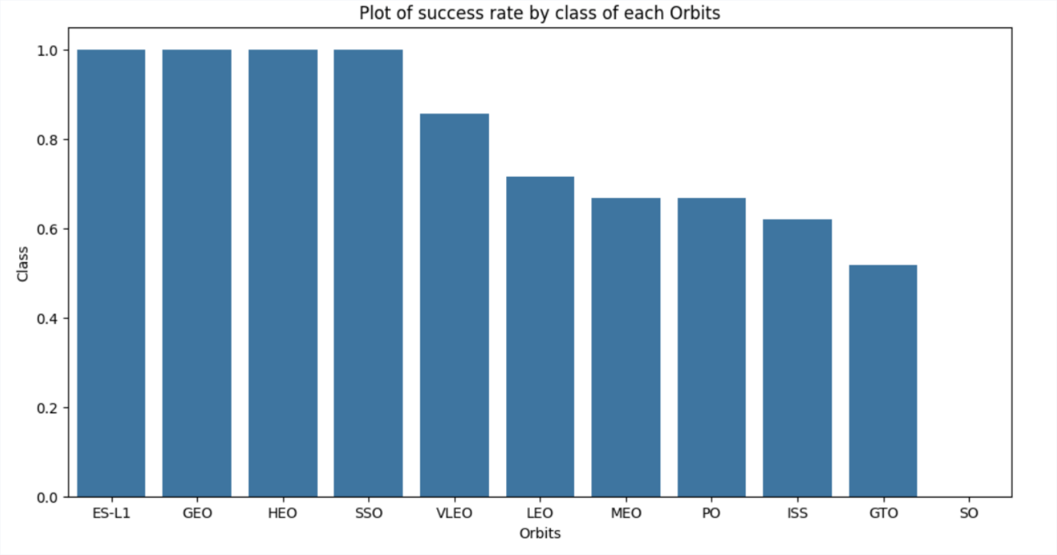
Export to csv file.

https://github.com/Mohanapriya-146/IBM_DataScience_capstone_SpaceX/blob/main/labs-jupyter-spacex-Data%20wrangling%20(1).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.





- https://github.com/Mohanapriya-146/IBM_DataScience_capstone_SpaceX/blob/main/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb

EDA with SQL

We applied EDA with SQL to get insight from the data. We wrote the following queries to find out

-The names of unique launch sites in the space mission.

-The total payload mass carried by boosters launched by NASA (CRS)

-The average payload mass carried by booster version F9 v1.1

-The total number of successful and failure mission outcomes

-The failed landing outcomes in drone ship, their booster version and launch site names.

- https://github.com/Mohanapriya-146/IBM_DataScience_capstone_SpaceX/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We 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 and answered some question as below

  -Are launch sites near railways, highways and coastlines.

  -Do launch sites keep certain distance away from cities.

https://github.com/Mohanapriya-146/IBM_DataScience_capstone_SpaceX/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- We have built an interactive dashboard with Plotly dash which contains dropdown list, range slider and graphs.
- 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.

https://github.com/Mohanapriya-146/IBM_DataScience_capstone_SpaceX/blob/main/SpaceX_Dashboard_App.py

Predictive Analysis (Classification)

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 from Logistic Regression, K nearest Neighbour, Decision tree classification.

https://github.com/Mohanapriya-146/IBM_DataScience_capstone_SpaceX/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

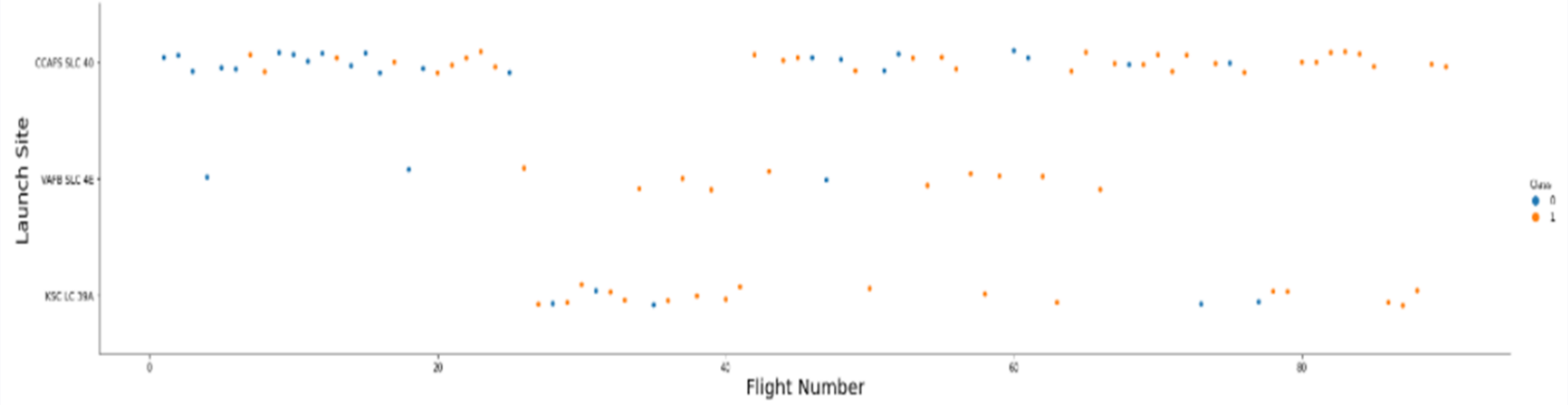
Results

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

Section 2

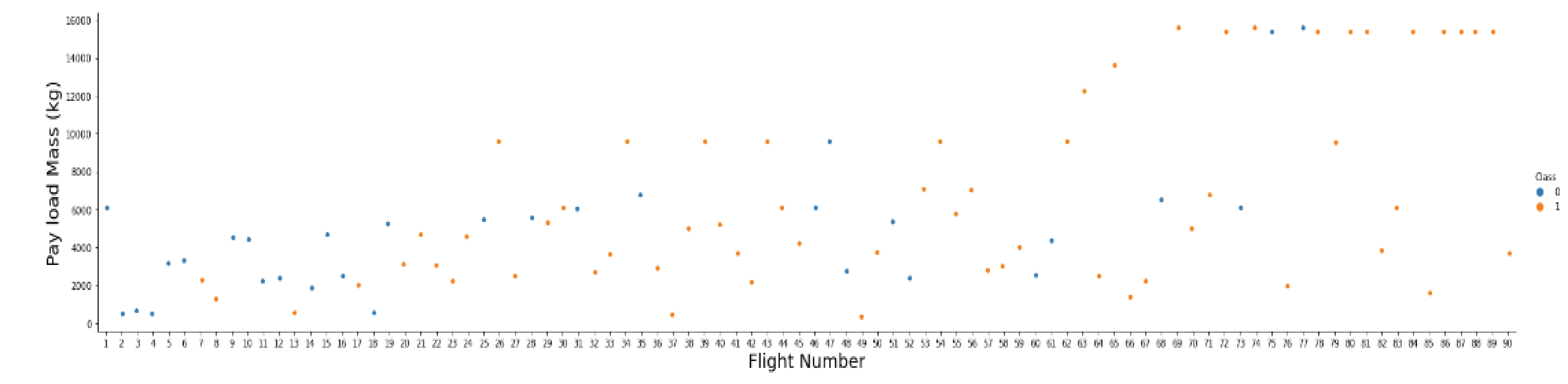
Insights drawn from EDA

Flight Number vs. Launch Site



- From the above Scatter 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

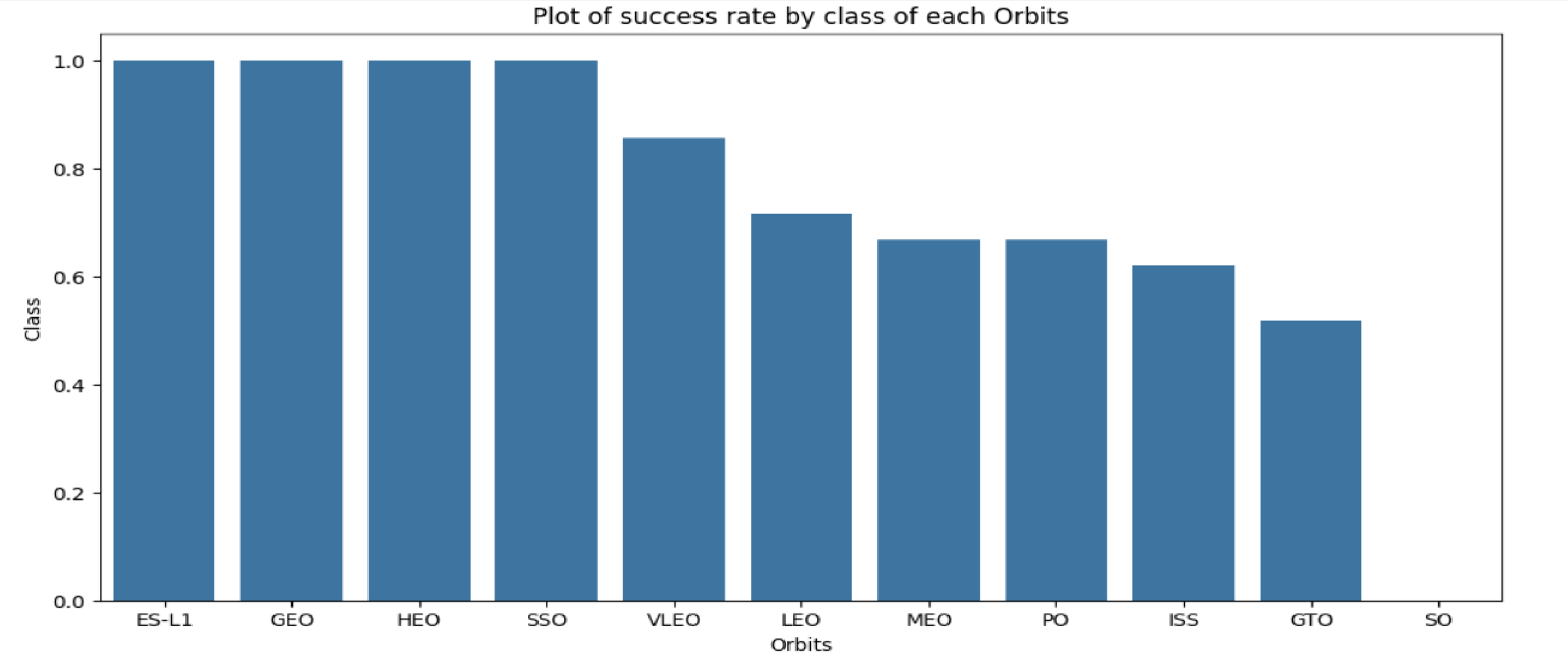


Above Scatter plot is plotted to find effect of flight number and payload mass in Launch outcome.

We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.

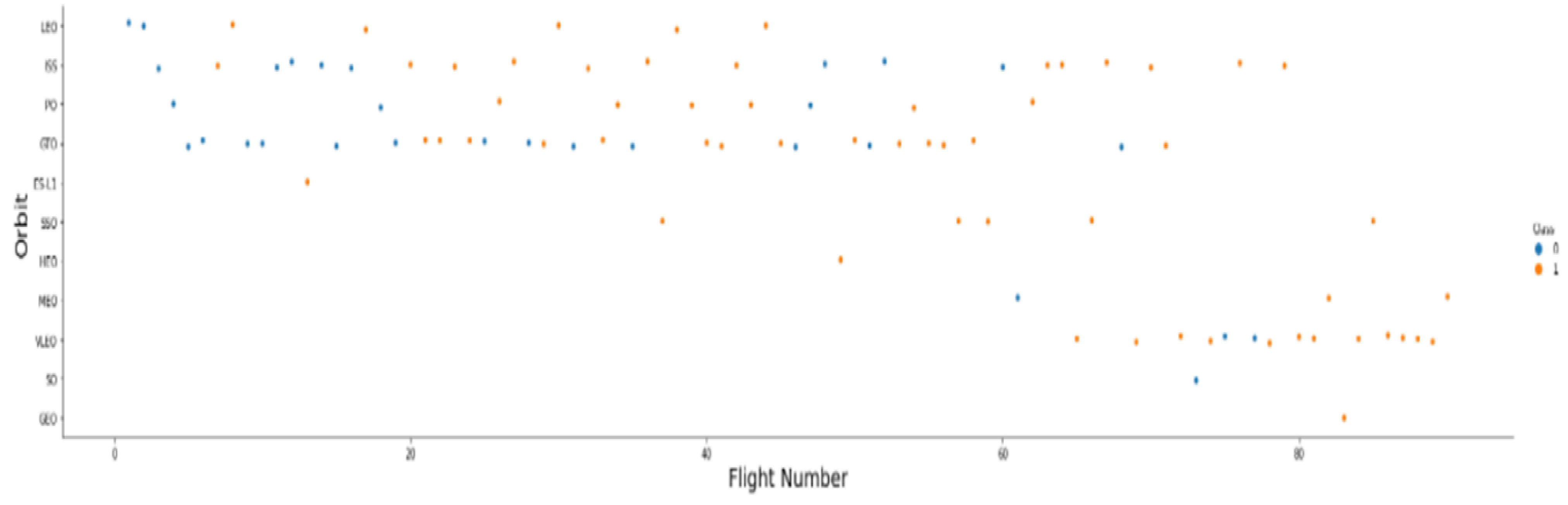
Success Rate vs. Orbit Type

- Bar chart for the success rate of each orbit type shows that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



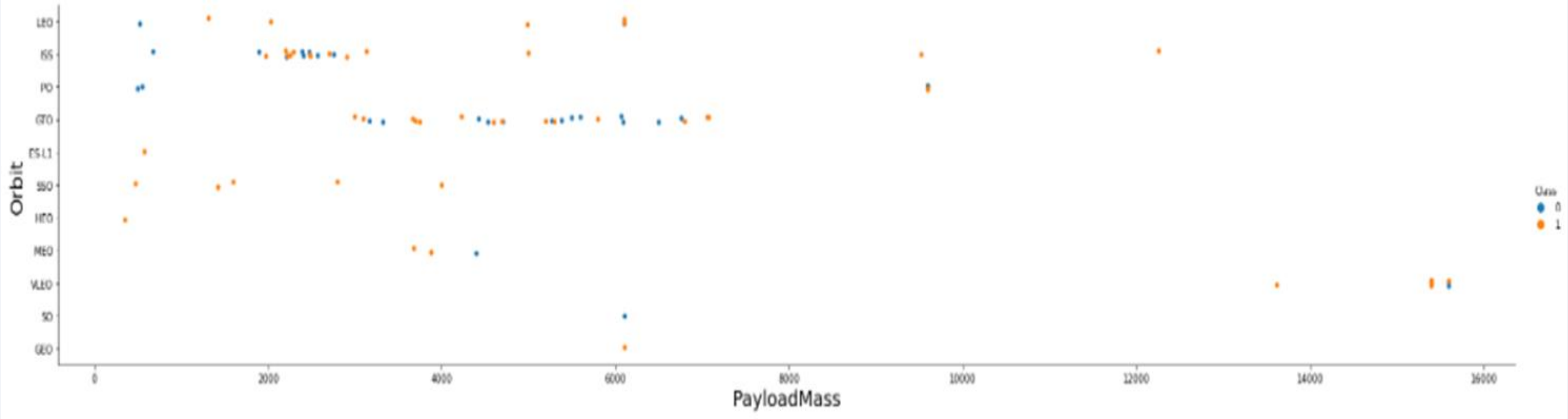
Flight Number vs. Orbit Type

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.



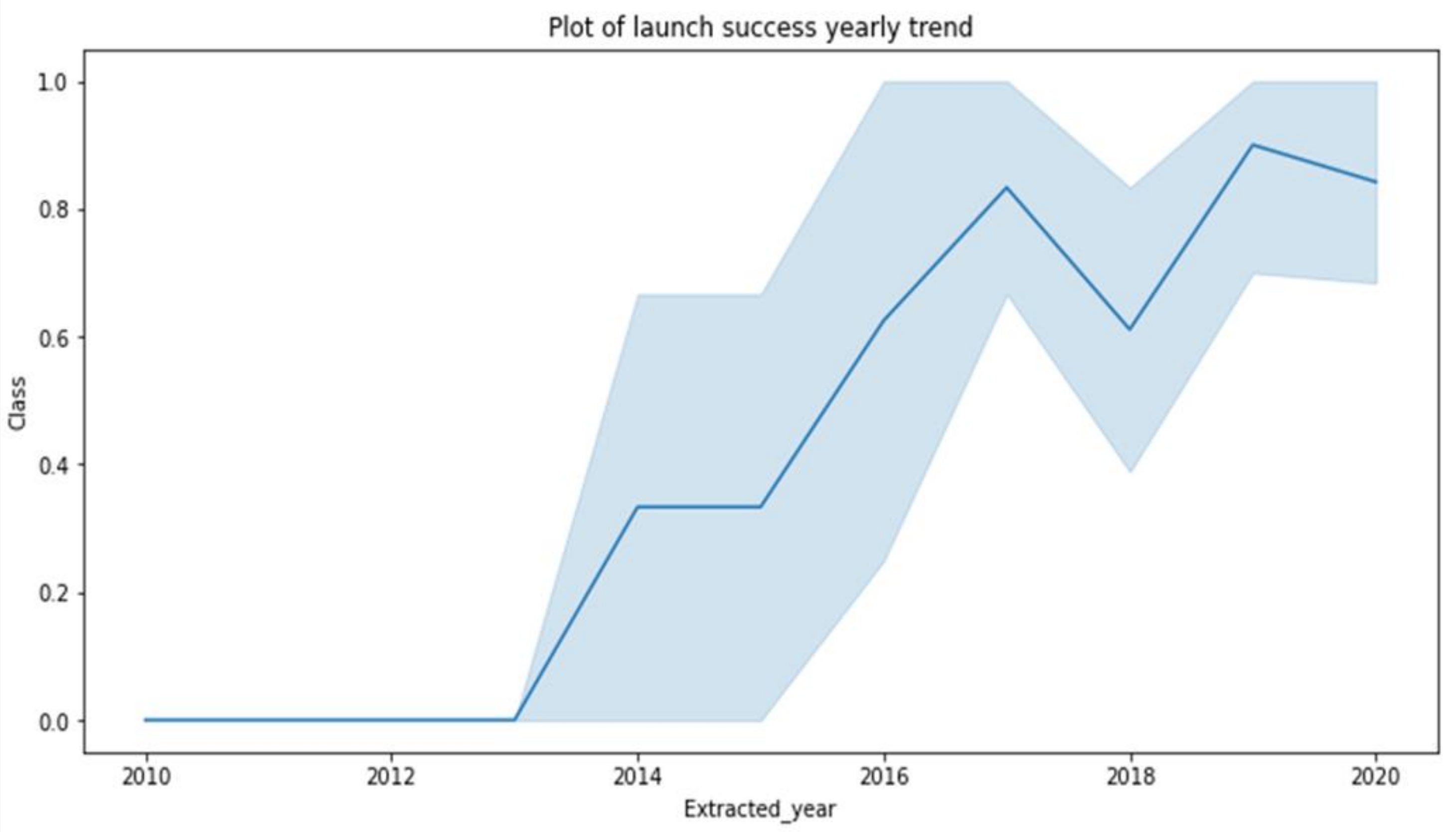
Payload vs. Orbit Type

With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. However for GTO we cannot distinguish this well as both positive landing rate and negative landing both are here.



Launch Success Yearly Trend

Line chart of yearly average success rate shows that the success rate since 2013 kept increasing till 2020.



All Launch Site Names

We used distinct keyword to find all the names of Launch sites as below from SpaceX table.

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

```
]: %sql select distinct(Launch_site) from SPACEXTABLE;

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

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'

[17]:
```
%sql select * from SPACETABLE WHERE LAUNCH_SITE LIKE 'CCA%'LIMIT 5;
```

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

[17]:

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

Above SQL Query filters Launch site which starts with 'CCA' using LIKE Keyword.

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 kg using the below SQL query.

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

```
[12]: %sql SELECT SUM(PAYLOAD_MASS__KG_) as TOTAL_PAYLOAD FROM SPACEXTABLE WHERE CUSTOMER='NASA (CRS)';
```

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

[12]:

TOTAL_PAYLOAD
45596

Average Payload Mass by F9 v1.1

The average payload mass carried by booster version F9 v1.1 is calculated as 2534.66 kg using the below SQL query.

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

[13]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXTABLE WHERE BOOSTER_VERSION LIKE 'F9 v1.1%';
```

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

[13]:

AVG_PAYLOAD
2534.6666666666665

First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015.

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

Hint:Use min function

```
[32]: %sql SELECT MIN(DATE) FROM SPACEXTABLE WHERE LANDING_OUTCOME='Success (ground pad)';
```

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

[32]:

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 using LIKE keyword in SQL query.

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

```
•[38]: %sql SELECT BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE LANDING_OUTCOME LIKE'%Success (drone ship)%'and
PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

[38]:

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

Total number of successful mission outcomes is 100 and Failure is 1.

List the total number of successful and failure mission outcomes

```
[17]: %sql SELECT COUNT(MISSION_OUTCOME) AS TOTAL_SUCCESS FROM SPACEXTABLE where MISSION_OUTCOME LIKE 'SUCCESS%';
```

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

[17]:

TOTAL_SUCCESS
100

```
[18]: %sql SELECT COUNT(MISSION_OUTCOME) AS TOTAL_FAILURE FROM SPACEXTABLE where MISSION_OUTCOME LIKE 'FAILURE%';
```

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

[18]:

TOTAL_FAILURE
1

Boosters Carried Maximum Payload

- Below are the Booster versions which have carried maximum payload mass.

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

•[19]:
```
%sql SELECT BOOSTER_VERSION , PAYLOAD_MASS__KG_ FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE order by BOOSTER_VERSION);
```

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

[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

2015 Launch Records

- We used a combinations of the **WHERE** clause, **LIKE**, **AND**, conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

21]:
```
%sql SELECT substr(Date, 6,2) AS MONTHNAME,LANDING_OUTCOME, BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTABLE
WHERE LANDING_OUTCOME LIKE '%Failure (drone ship)%' and DATE LIKE '%2015%';
```

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

21]:

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

- 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

[29]:
```
%sql SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS RANK FROM SPACEXTABLE WHERE
LANDING_OUTCOME IN ('Failure (drone ship)','Success (ground pad)') AND
DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING_OUTCOME ORDER BY COUNT(LANDING_OUTCOME) DESC;
```

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

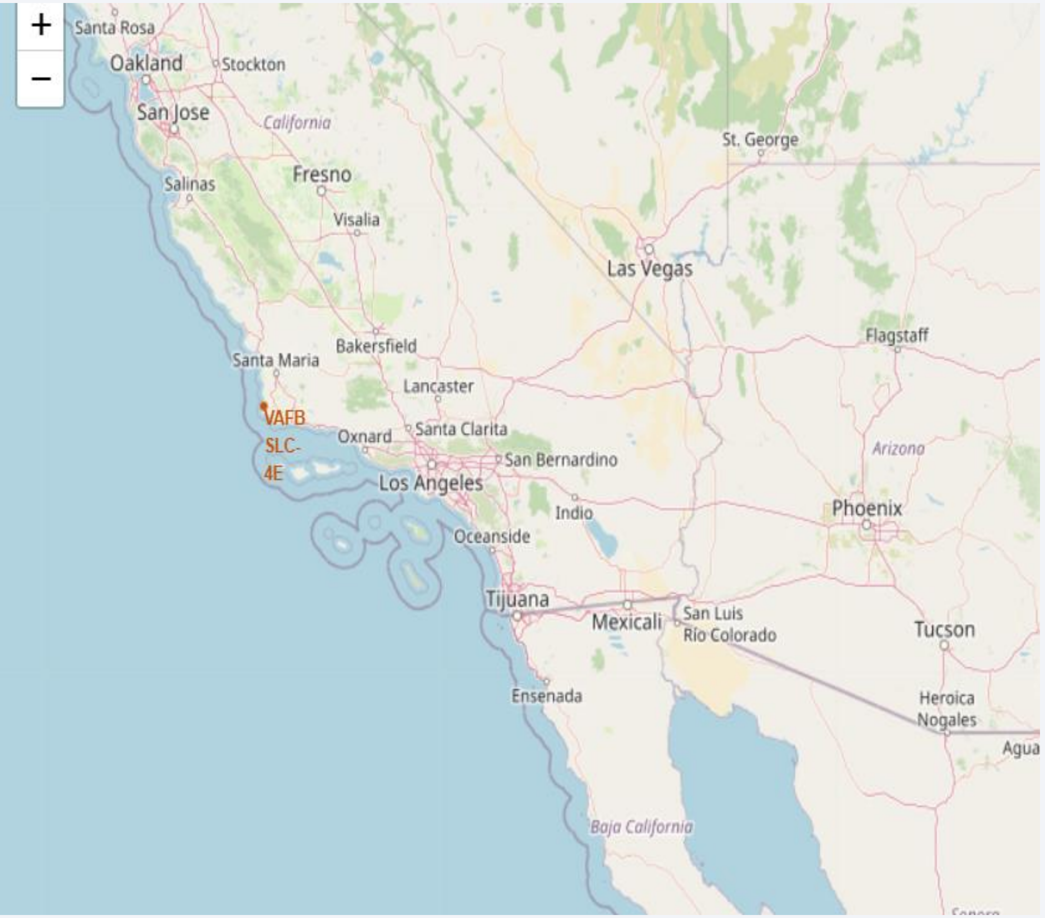
[29]:

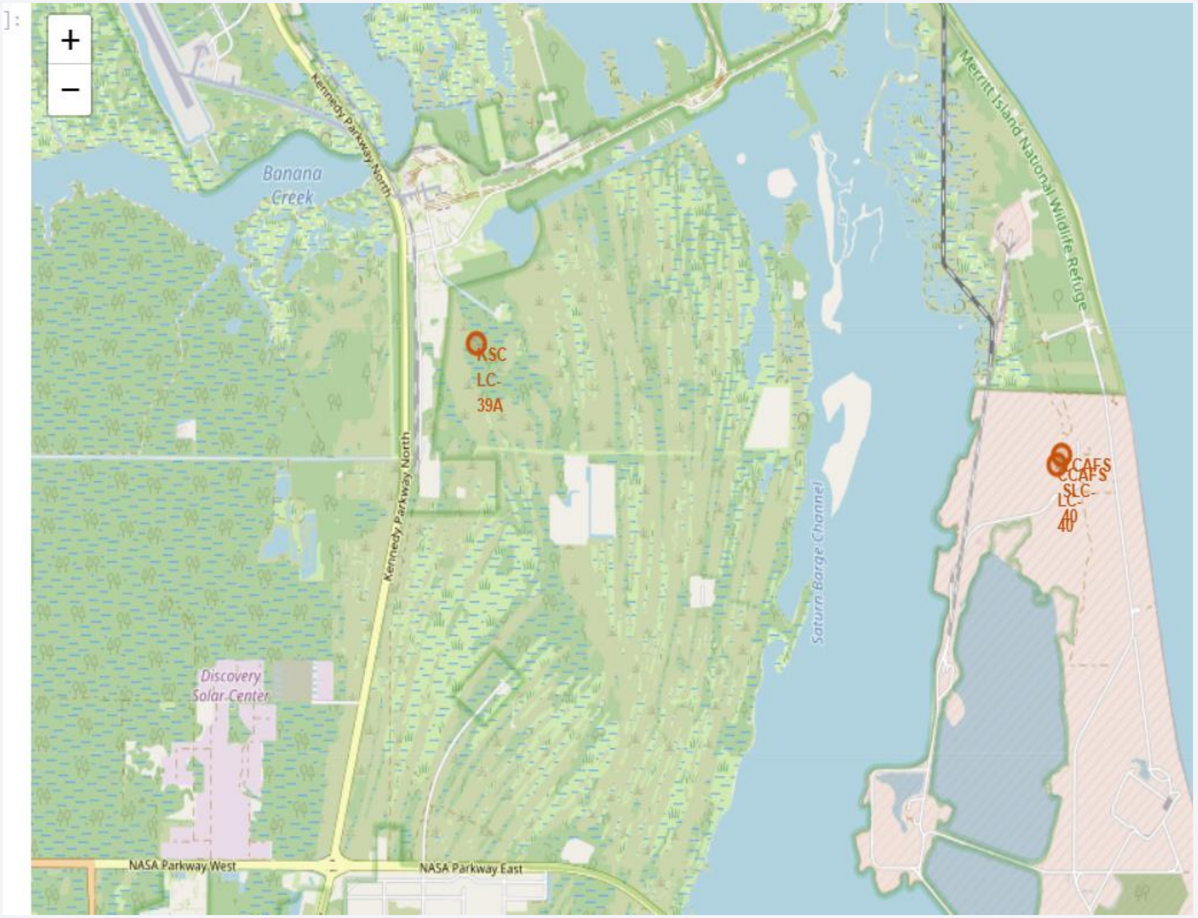
Landing_Outcome	RANK
Failure (drone ship)	5
Success (ground pad)	3

Section 3

Launch Sites Proximities Analysis

Launch Sites in global map

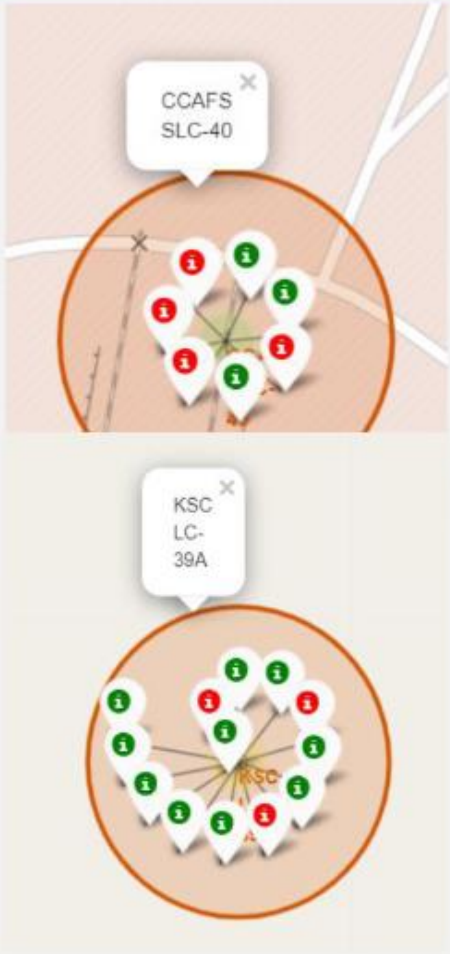




Launch site VAFB SLC-4E is located near California coast

Remaining Three launch sites CCAFS LC-40, CCAFS SLC-40, KSC LC-39A are located near Florida Coast

Markers showing launch sites with color labels



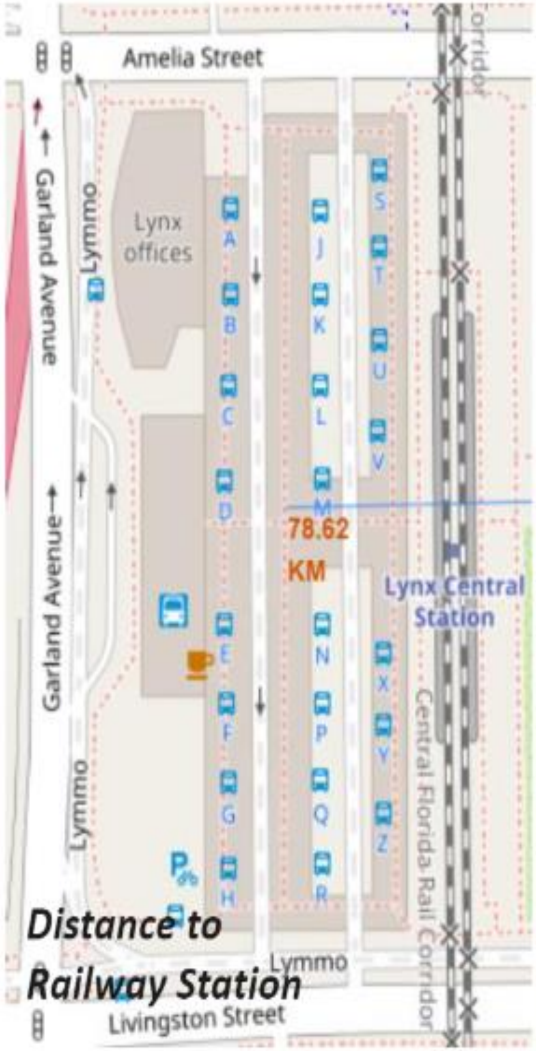
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

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

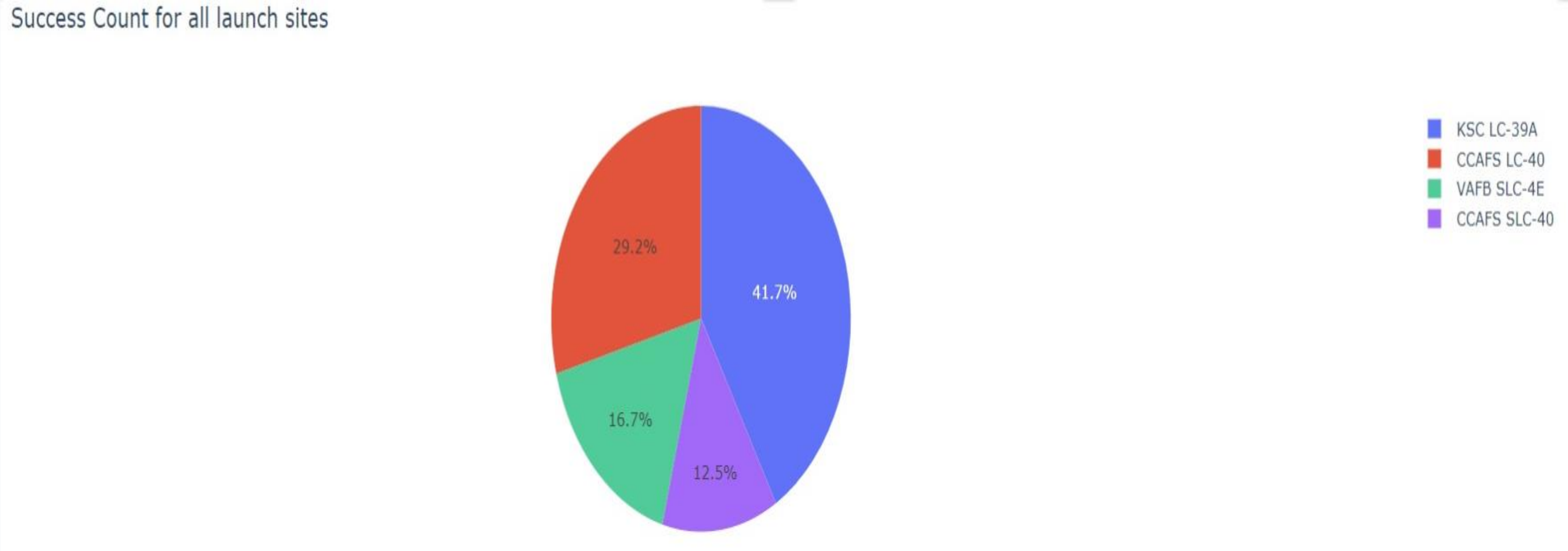
Launch Site distance to landmarks

•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 4

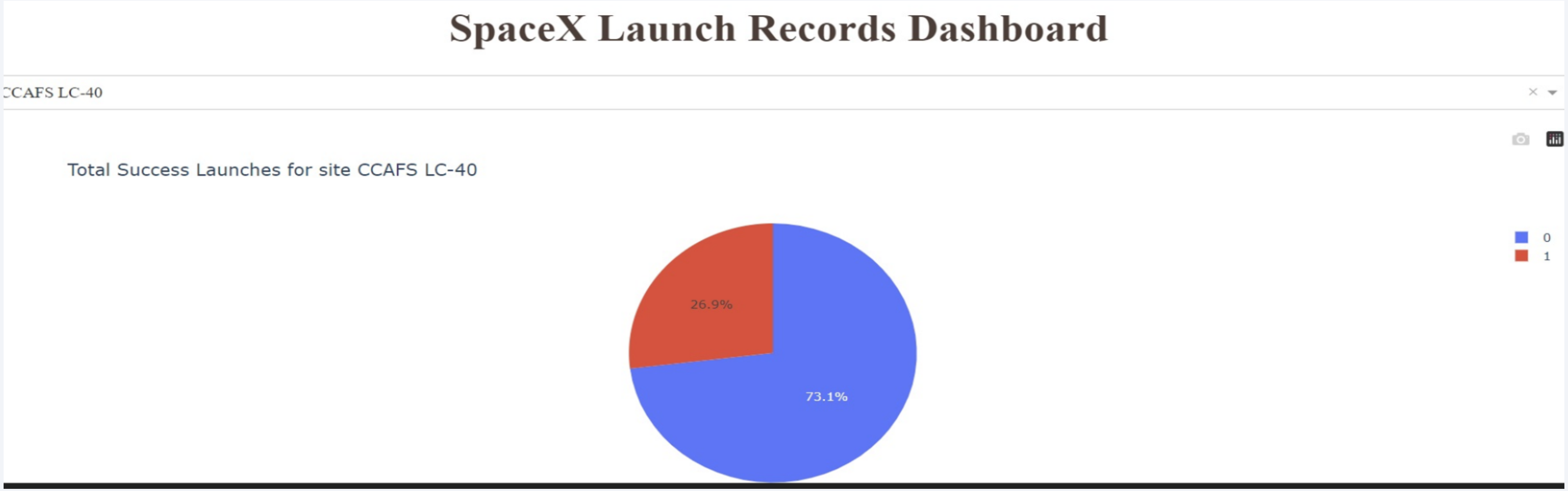
Build a Dashboard with Plotly Dash

Pie Chart for success count for all launch sites



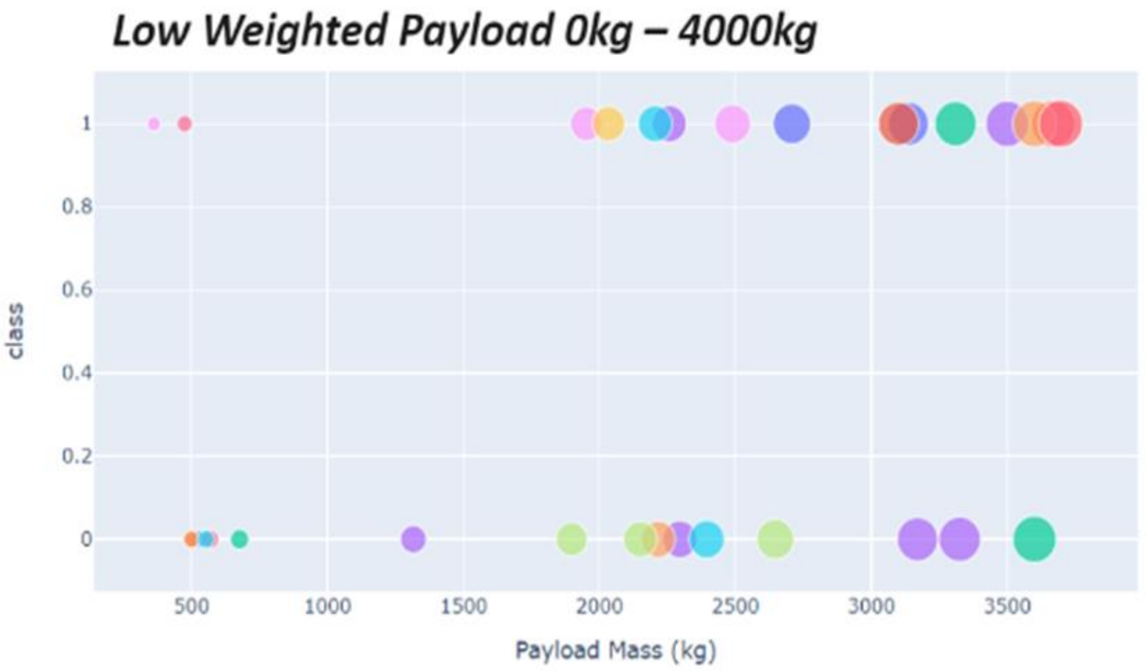
From the above pie chart we can see KSC LC-39A had most successful launches of all.

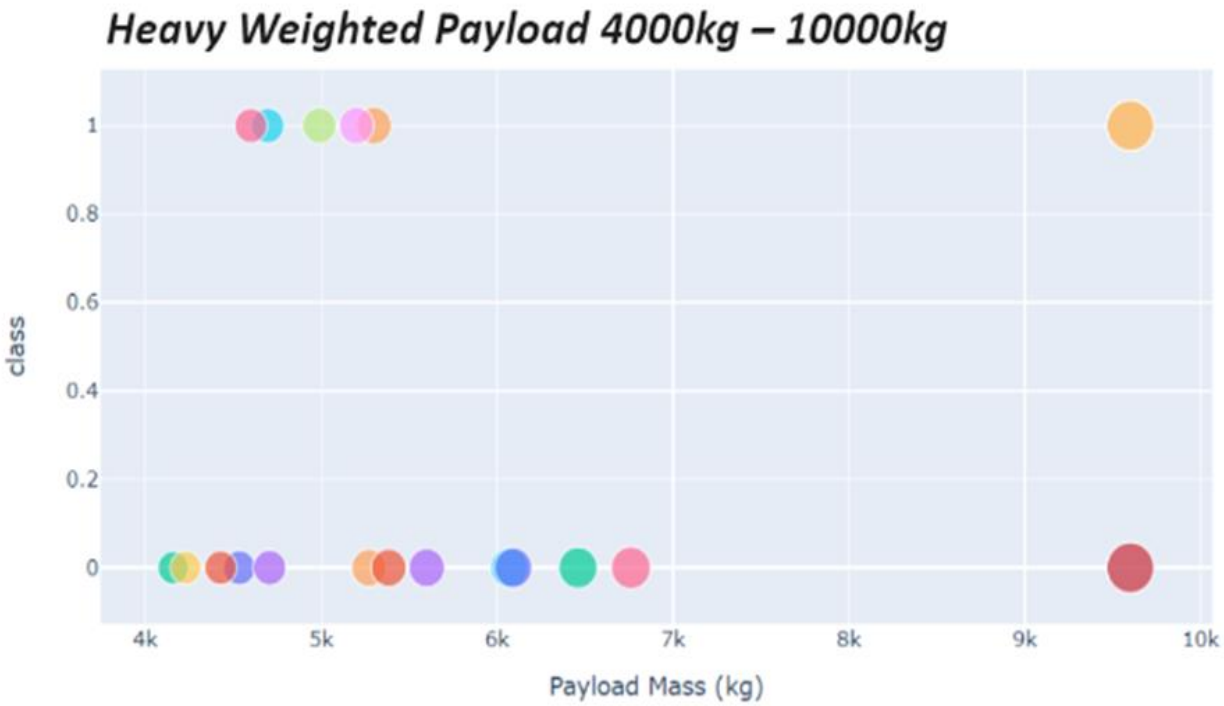
Pie chart showing the Launch site with the highest launch success ratio



From the above chart CCAFS LC-40 achieved 73.1% Success and 26.9% 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

Section 5

Predictive Analysis (Classification)

Classification Accuracy

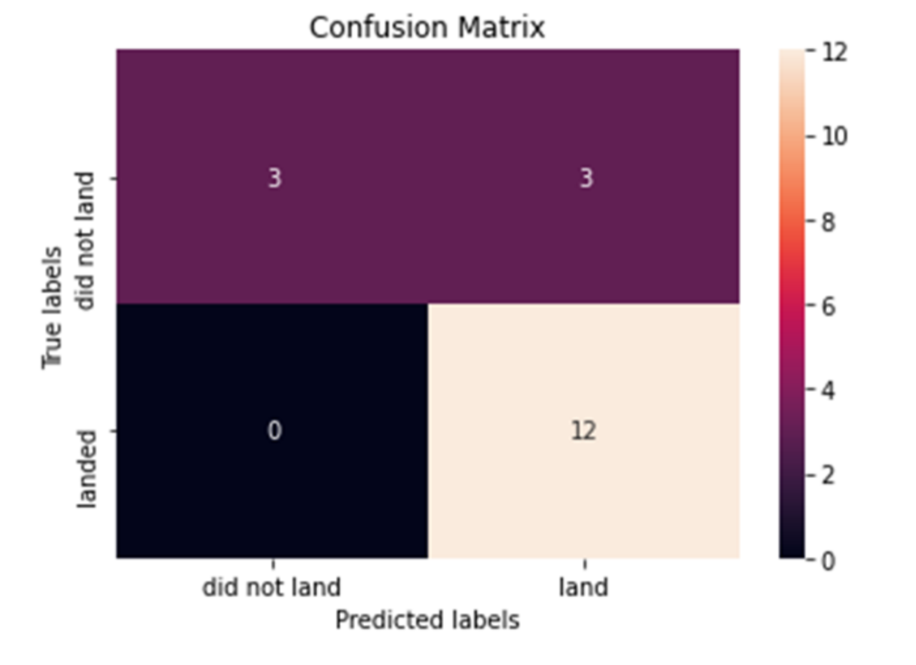
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

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

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.
- Success rate for light weighted payload is greater than heavy weighted payload.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- CCAFS LC-40 had the most successful launches of all Launch sites.
- The Decision tree classifier is the best machine learning algorithm because of higher classification accuracy compared to others.

Appendix (Reference links)

Datasets in CSV file:

1. Dataset1.csv
2. Dataset2.csv
3. Dataset3.csv

Dashboard with Plotly Dash:

1. PieChart_AllSites
2. Launch_Dropdown
3. Success_count_based_on_Payload
4. Range_Slider

Thank you!