IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Vaishnavi Patel
23-06-2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
  - Data Collection
    - Through API and Web scraping
  - Data munging or wrangling
  - Exploratory Data Analysis (EDA) with SQL and Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction with LR, SVM, Decision Tree, KNN
- Summary of all results
  - Screenshots of Data collection methods and EDA analysis
  - Screenshots for Data visualization
  - Screenshots for Machine Learning Result

Introduction

- **Project background and context**
  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars. SpaceX can reuse the first stage. If we can determine the if first stage will land, we can determine the cost of a launch.
- **Problems you want to find answers**
  - As Data Scientists at SpaceY : Determine the price of each launch, create dashboard for team
  - Train a machine learning model and predict if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Gathered SpaceX launch data from an API and Web scraping from Wikipedia
- Perform data wrangling
  - Data wrangling was performed using One-hot encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Accuracy of LR, SVM,Tree and KNN model was found and than predictive analysis is done

Data Collection

- Data is collected by two method: Using API and Web scraping.

API

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- The GitHub URL of the completed SpaceX API calls notebook as an external reference and peer-review purpose:

https://github.com/Vaishnavi179/Data_Science/blob/main/Data_Collection_API.ipynb

Imported required libraries

- Requests, pandas, NumPy, datetime

Static_json_url

- Data frame :data : pd.json_normalize(response.json())

Filtered dataset :

- data_falcon9

Data Collection - Scraping

- The GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

https://github.com/Vaishnavi179/Data_Science/blob/main/Data_collection_web_scrapping.ipynb

Imported required libraries

- Requests, pandas, sys, BeautifulSoup

Soup object

- Parse the details

Filtered dataset :

- Df = pd. DataFrame(launch_dict)

Data Wrangling

- The GitHub URL of the completed data wrangling notebook:

https://github.com/Vaishnavi179/Data_Science/blob/main/Data_Wrangling.ipynb

Libraries and Datafile

- Pandas, NumPy, df dataset

Value_counts

- Lanuch_site, Orbit , landing outcome

Landing class :

- 0 if bad outcome, 1 for success rate

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts:
  - Scatter Plots
  - Bar Plots
  - Line charts
- To find the relationship between two variable scatter plot gives basic insights. Bar plot will visualize success rate of each orbit. Line charts will show Lunch success rate yearly.
- Add the GitHub URL:
- https://github.com/Vaishnavi179/Data_Science/blob/main/EDA_VIZ.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - Unique Launch Site in space mission
  - Average payload mass
  - First successful landing outcome
  - Total number of successful and failure mission outcomes
- GitHub URL of your completed EDA with SQL notebook:

https://github.com/Vaishnavi179/Data_Science/blob/main/EDA%20SQL%20IN%20PYTHON.ipynb

Build an Interactive Map with Folium

- We marked map objects such as markers, circles, lines are marked for success or failure of launches for each site on the folium map.
- Color-labeled outcome
- Distance between a launch site to proximities are seen in map.
- Add the GitHub URL of your completed interactive map with Folium map: https://github.com/Vaishnavi179/Data_Science/blob/main/Launch_Site_location_with_folium.ipynb

Build a Dashboard with Plotly Dash

- Pie chart, Slider and scatter plots are added in Dashboard.
- GitHub URL of your completed Plotly Dash lab,:
- https://github.com/Vaishnavi179/Data_Science/blob/main/Dashboard_plotly.ipynb

Predictive Analysis (Classification)

- NumPy, pandas, libraries are used for data loading and manipulation on data.
- Using hyperparameters, GridSearchCV are used to build machine learning algorithms.
- GitHub URL of your completed predictive analysis:
- https://github.com/Vaishnavi179/Data_Science/blob/main/ML_Prediction.ipynb

Results

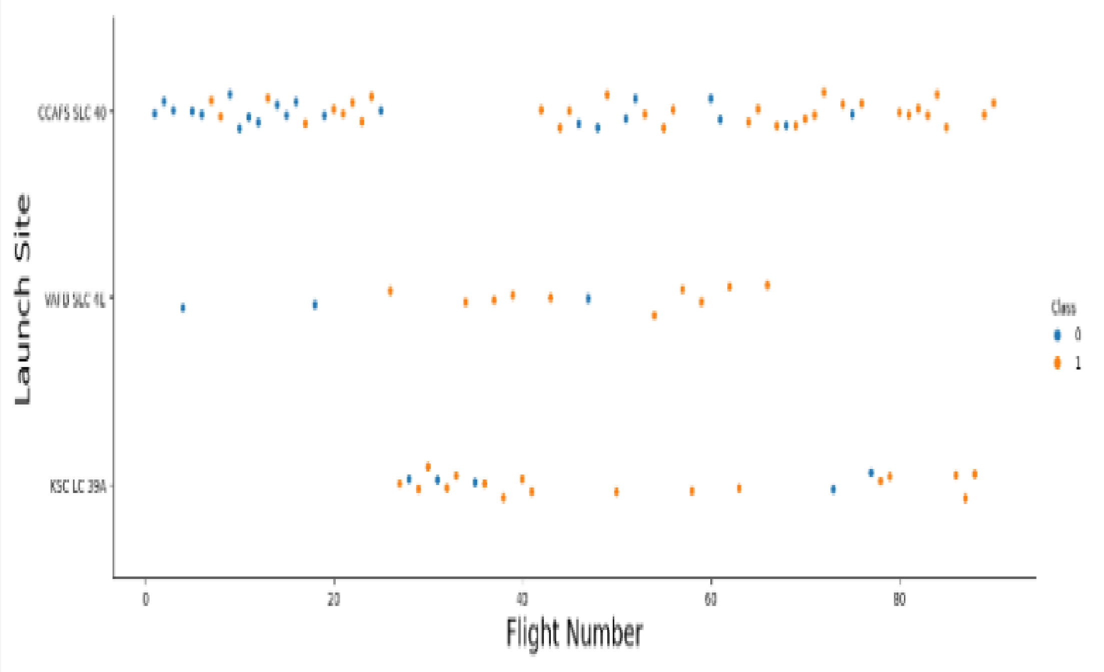
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

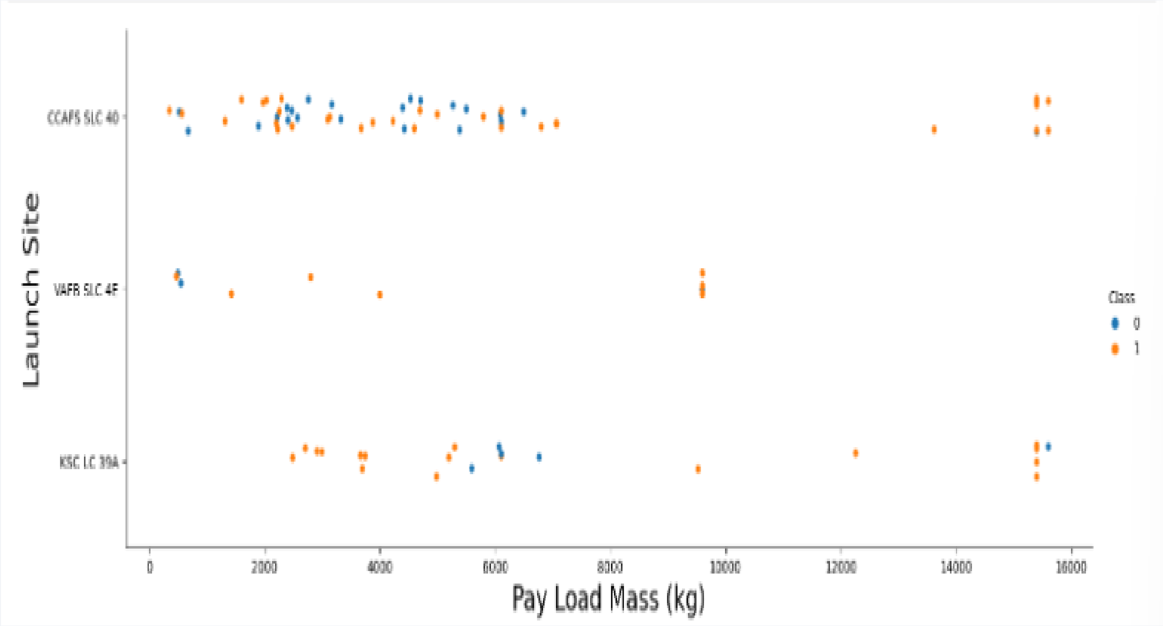
- Show a scatter plot of Flight Number vs. Launch Site



- Explanations
  - KSC LC 39A has flight numbers between 20 to 100.
  - VAFB SLS 4E has more successful flights.
  - CCAFS SLC 40 has maximum flights.

Payload vs. Launch Site

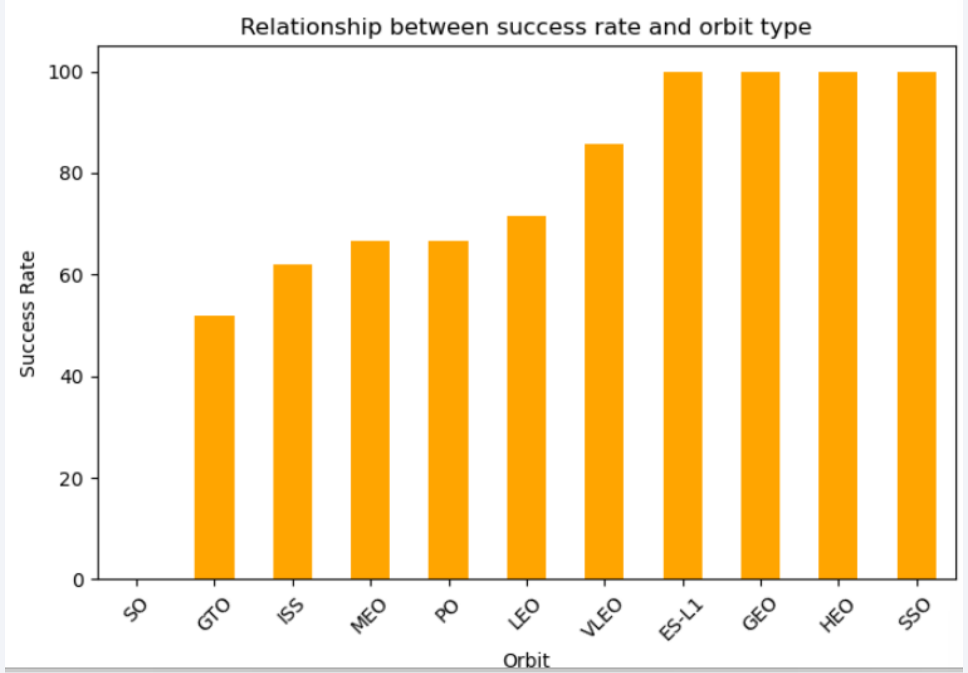
- Show a scatter plot of Payload vs. Launch Site



- Explanations
    - VAFB-SLC lunch site there are no rockets launched for heavy payload mass > 10000.

Success Rate vs. Orbit Type

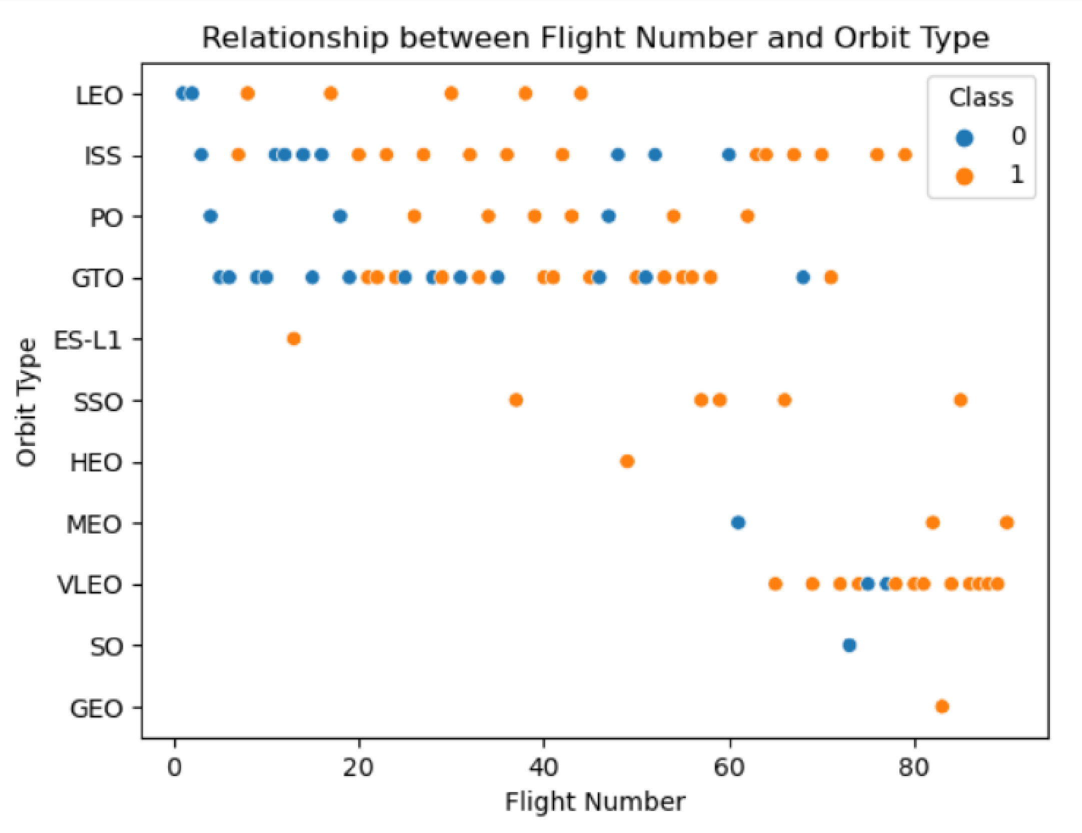
- Show a bar chart for the success rate of each orbit type



- Explanations
  - ES-L1, GEO , HEO and SSO orbits have high success rate.

Flight Number vs. Orbit Type

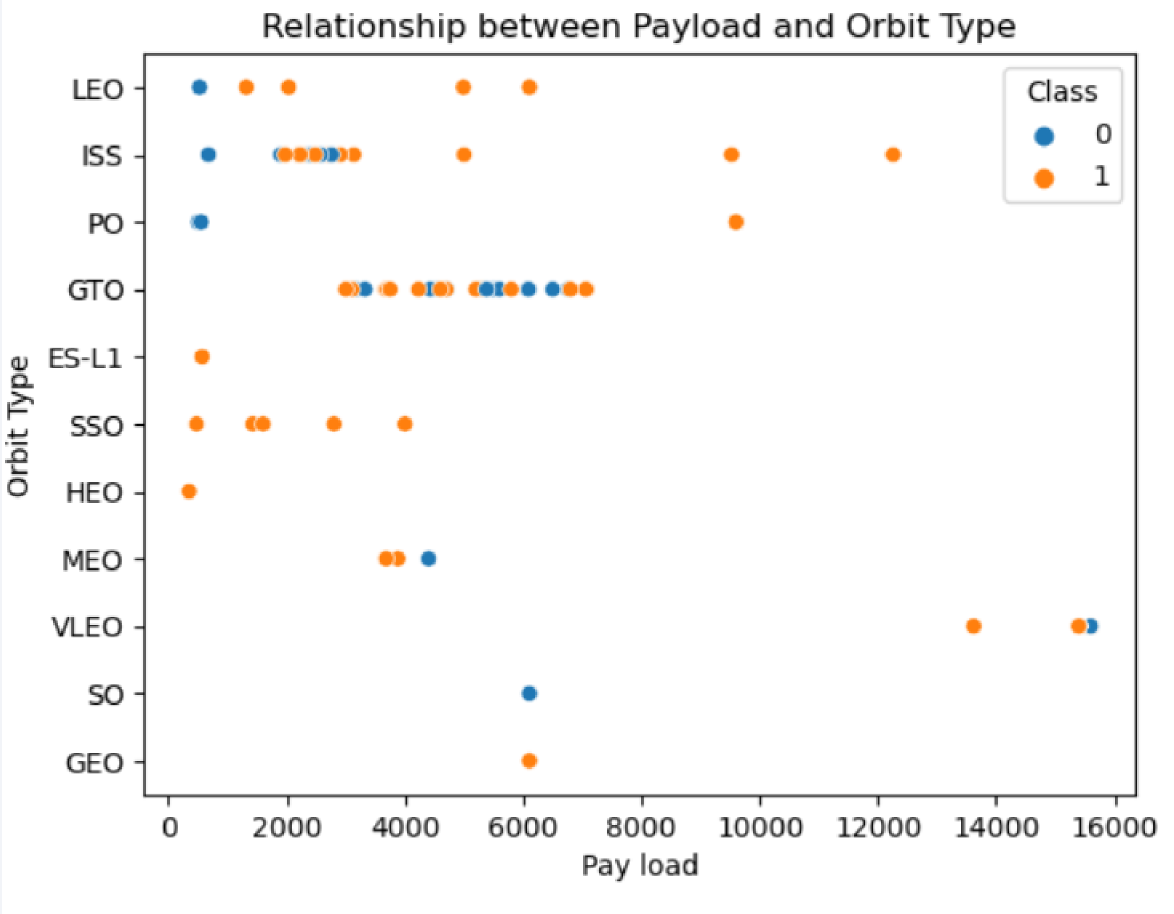
- Show a scatter point of Flight number vs. Orbit type



- Explanations
  - For LEO orbit the success appears related to the number of flights.

Payload vs. Orbit Type

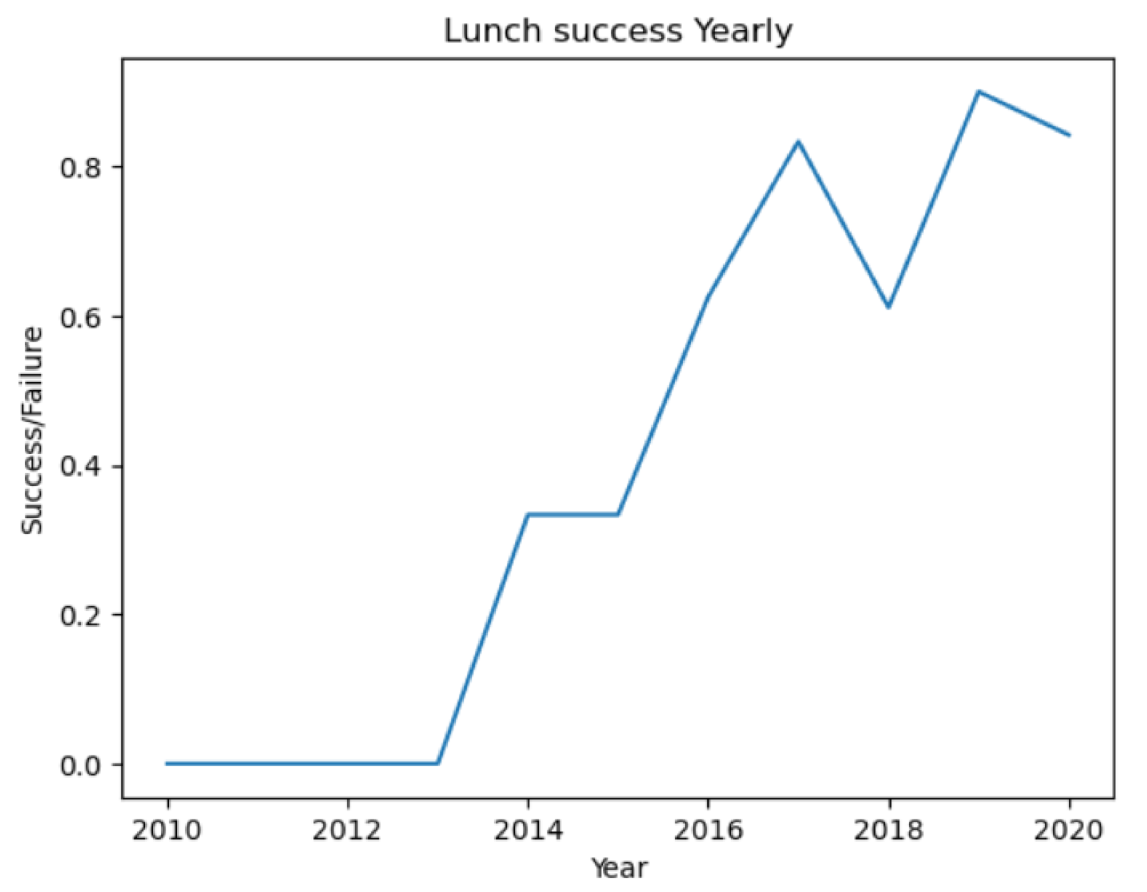
- Show a scatter point of payload vs. orbit type



- Explanations
  - LEO , ISS and Polar orbits have more positive landing with heavy payloads.

Launch Success Yearly Trend

- Show a line chart of yearly average success rate



- Explanations
  - Success rate is kept increasing after 2013.

All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
```

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

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

- Explanation:
  - DISTINCT function gives all the unique name of launch site.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

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

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

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

Launch_Site
CCAFS LC-40
CCAFS SLC-40

- Explanation: There are two Launch Site starts with CCAFS LC – 40 and CCAFS SLC-40

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

Task 3

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

```
%sql SELECT SUM(Payload_Mass__KG_) AS Total_Payload_Mass FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

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

Total_Payload_Mass
45596.0

- Explanation: SUM aggregate function is used to find total payload mass which is 45596 kg.

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

Task 4

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

```
%sql SELECT AVG(Payload_Mass__KG_) AS average_payload_mass FROM SPACEXTBL WHERE Booster_Version = 'F9
```

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

average_payload_mass
2928.4

Explanation : AVG aggregate function is used to find average payload mass for booster version F9 v1.1 which is 2928.4 kg

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

Task 5

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

Hint:Use min function

```
6]: %sql SELECT MIN(Date) AS first_successful_landing_date FROM SPACEXTBL WHERE Landing_Outcome = 'Success
```

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

6]:

first_successful_landing_date
01/08/2018

- Explanation: Min function is used.

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
- Present your query result with a short explanation here

Task 6

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_
```

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

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

- Explanation:

AND function is used to apply condition on payload mass.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

Task 7

List the total number of successful and failure mission outcomes

```
Success = %sql SELECT COUNT(Mission_Outcome) AS Success FROM SPACEXTBL WHERE Mis
Failure = %sql SELECT COUNT(Mission_Outcome) AS Failure FROM SPACEXTBL WHERE Mis
print("Success count", Success)
print("Failure count", Failure)
```

```
 * sqlite:///my_data1.db
Done.
 * sqlite:///my_data1.db
Done.
Success count +---------+
| Success |
+---------+
|   100   |
+---------+
Failure count +---------+
| Failure |
+---------+
|    1    |
+---------+
```

- Explanation: Total Successful count is 100 and failure is 1.

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

Task 8

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

```
%sql SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE Payload_Mass__KG_ = (SE
```

```
 * 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

- Explanation: Distinct function gives unique booster version name and WHERE clause will apply condition on payload mass.

2015 Launch Records

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

```
%sql SELECT substr(Date, 4, 2) AS Month, Landing_Outcome, Booster_Version, Launcl
```

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

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

- Explanation: substr gives month.

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

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

In [19]:
```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

Out[19]:

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

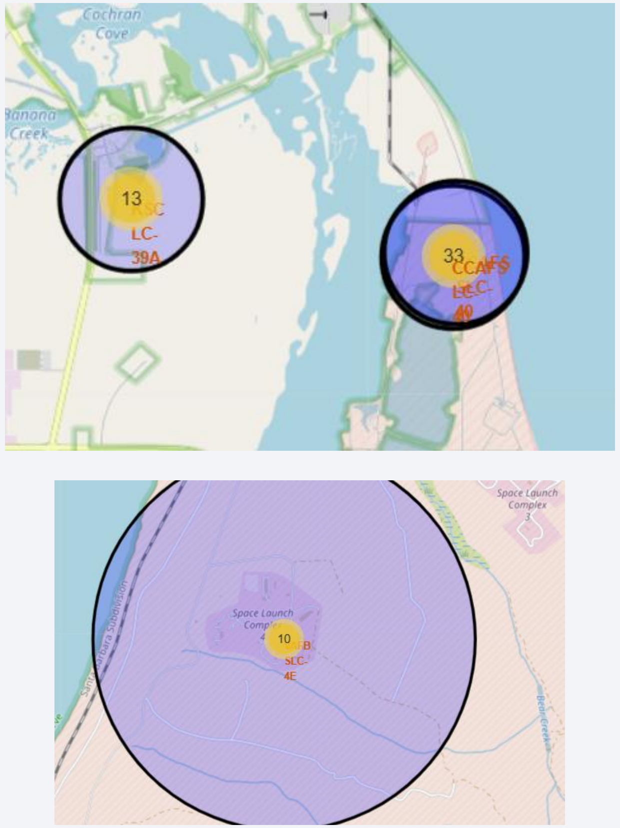
Section 3

Launch Sites Proximities Analysis

Launch sites on Map

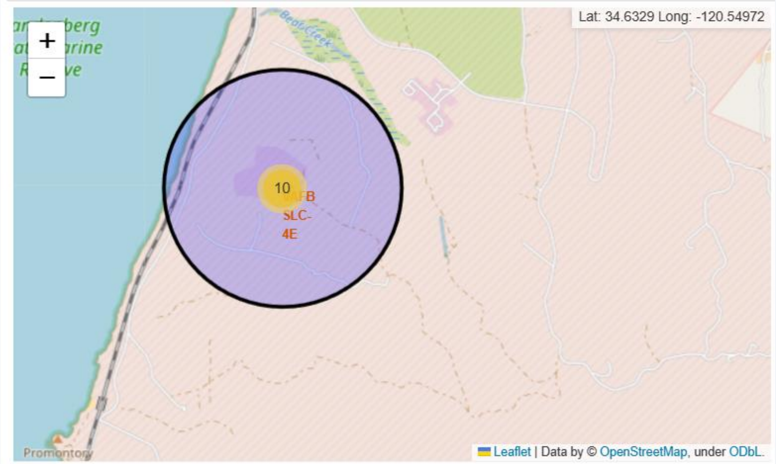
- SpaceX launch sites are in US of America coasts, Florida and California.

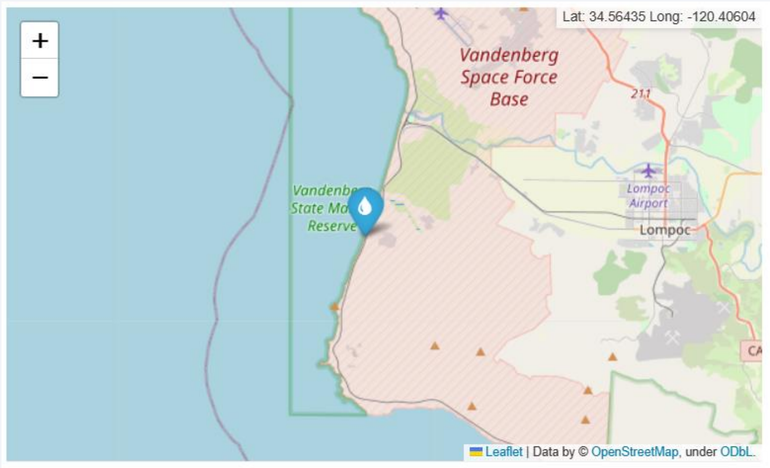
Marker Cluster to Map



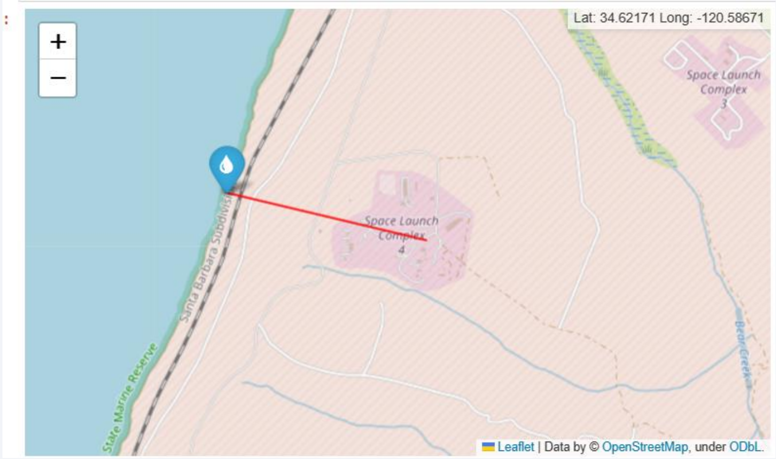
- Marker cluster on map.

Proximities and Distance to nearest coastal line





- Distance : VSF – to -coastline

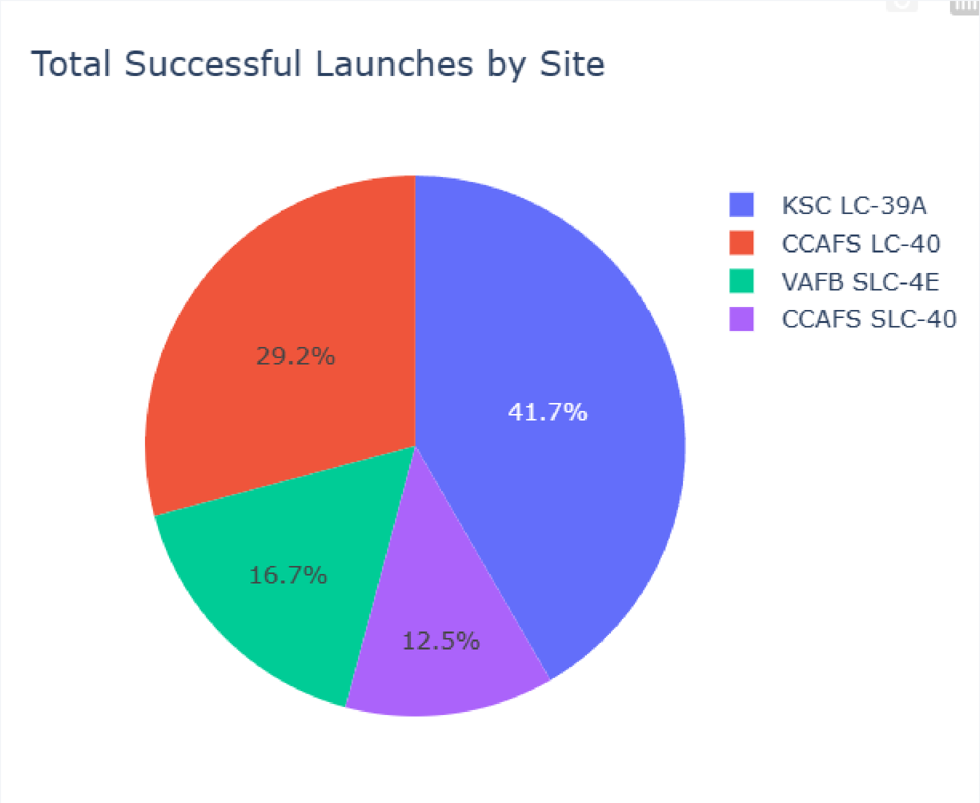


- Launch sites are not in close proximity to railways, highways, coastlines.
- Launch sites keep certain distance away from cities.

Section 4

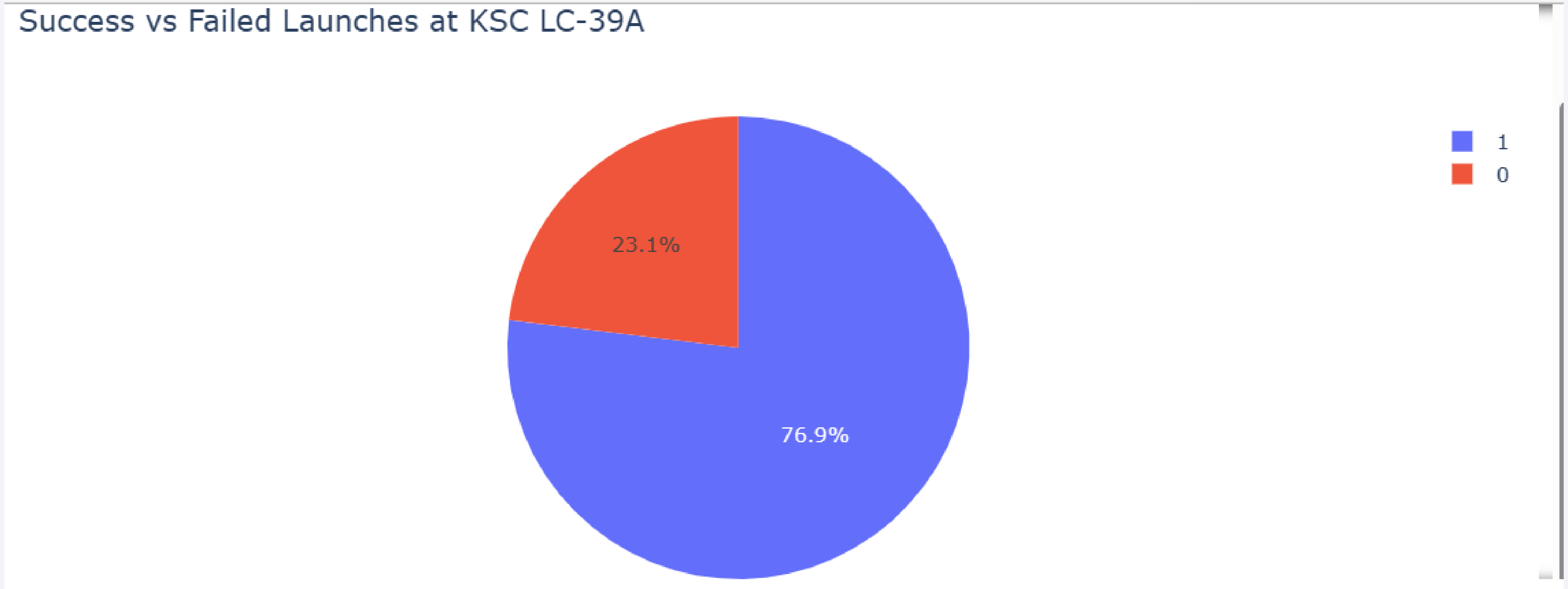
Build a Dashboard with Plotly Dash

Launch Success % by each site with Pi-Chart



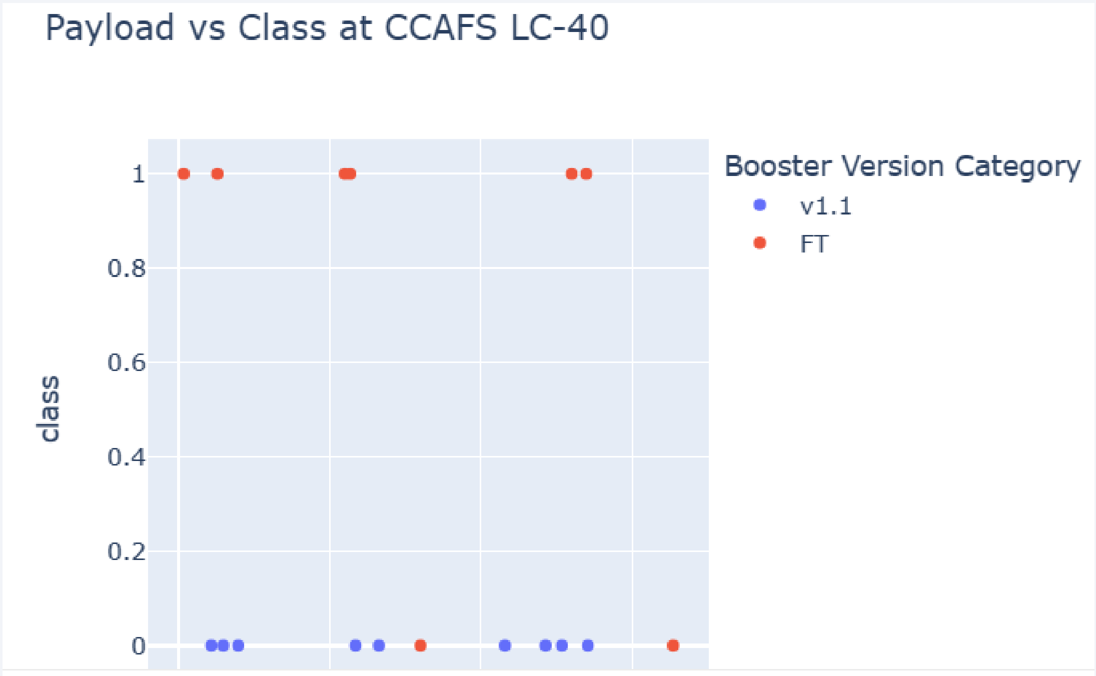
- KSC LC-39A Launch Site has most successful launches of 41.7%.

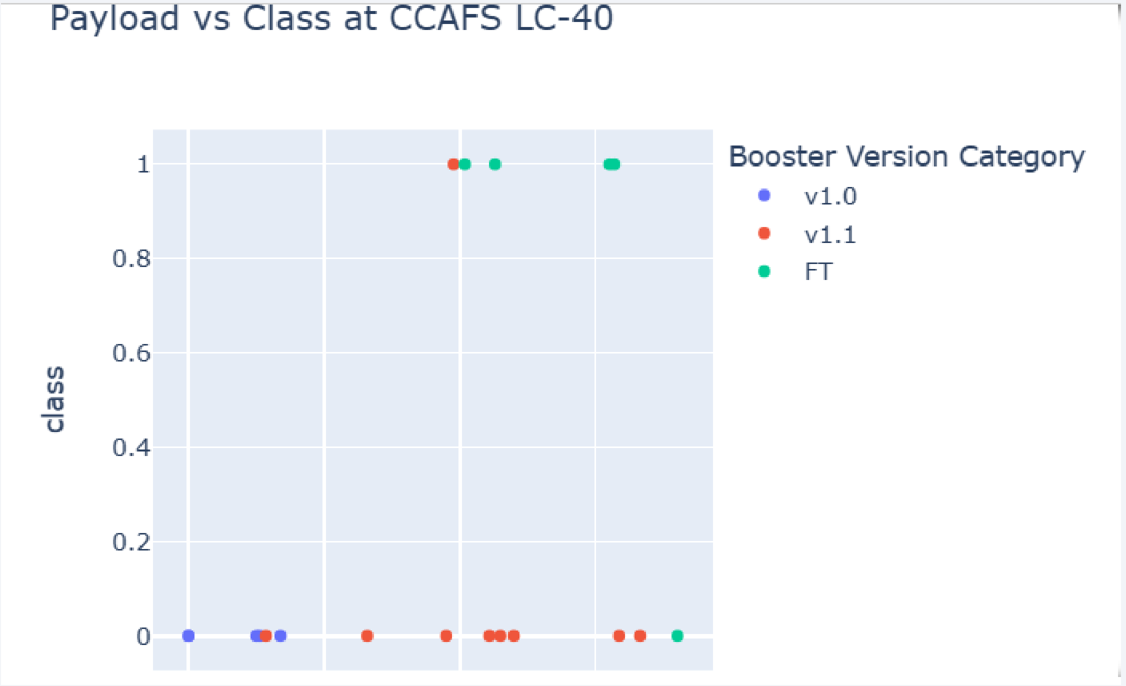
Success-Failure ratio for KSC LC-39A



- Maximum success %: 76.9 %
- Failure: 23.1 %

Payload vs Launch Outcome with Range slider





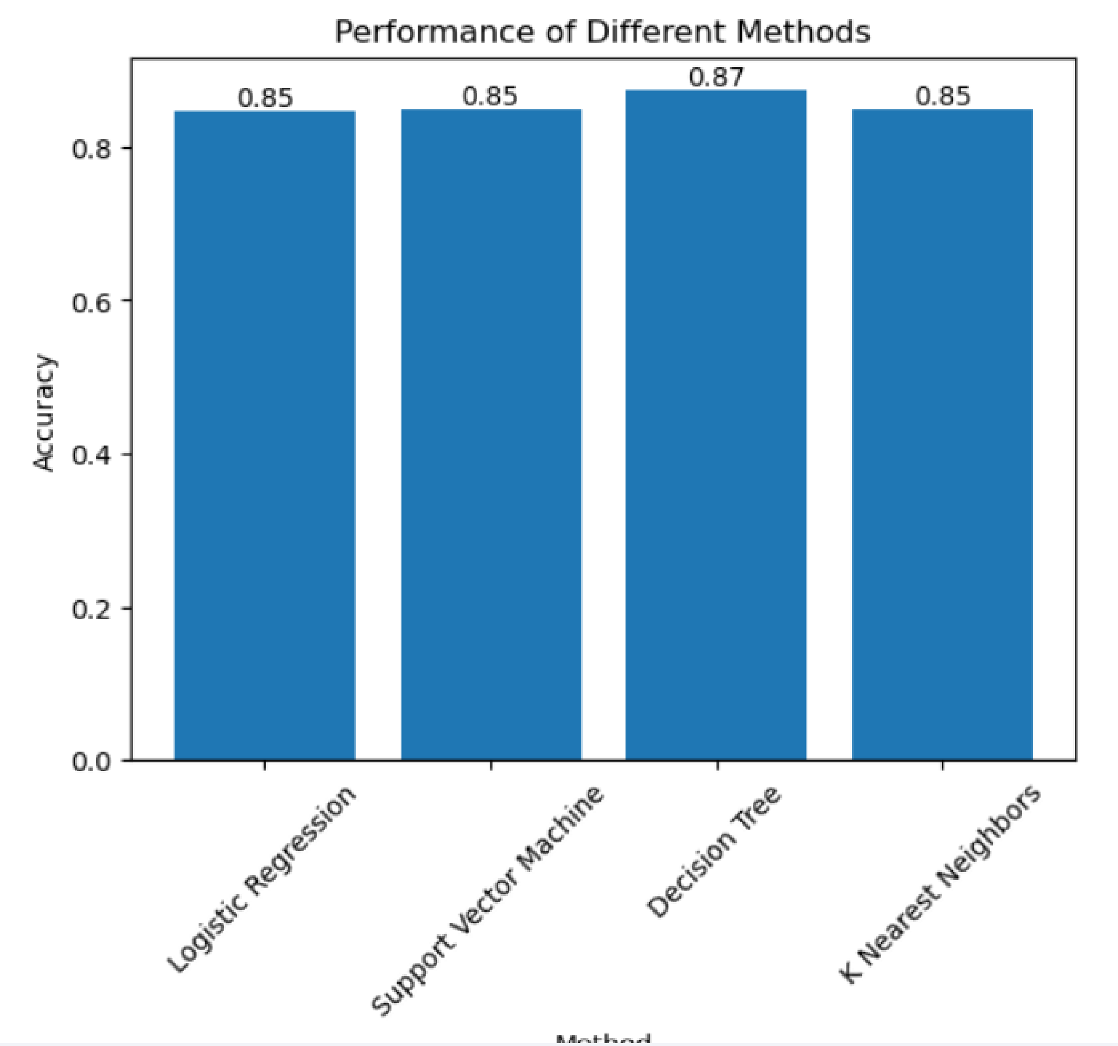
- Success rate for low weighted payloads launces is higher than heavy weighted payloads launches.

Section 5

Predictive Analysis (Classification)

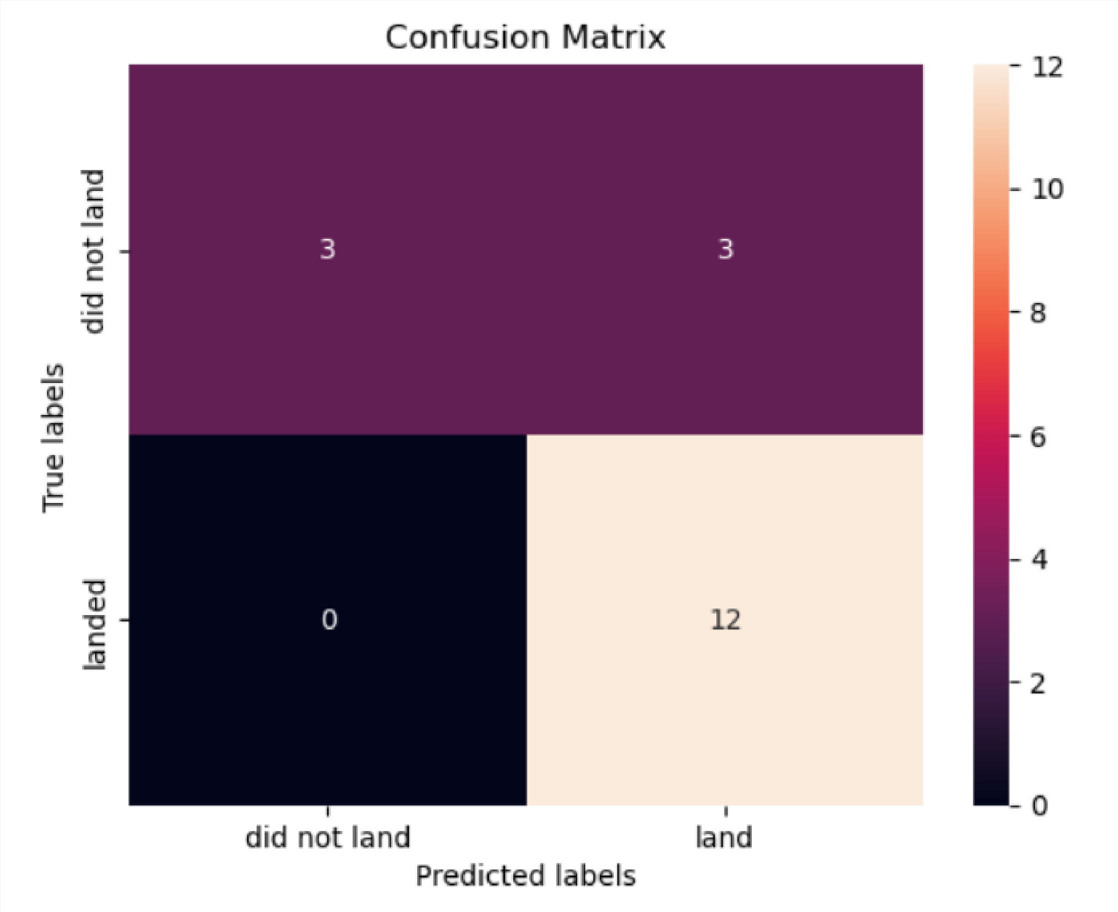
Classification Accuracy

- Best performing method: Decision Tree
- https://github.com/Vaishnavi179/Data_Science/blob/main/ML_Prediction.ipynb



Confusion Matrix

- Confusion matrix of Decision Tree
- True Negative (TN): 3
- False Positives (FP): 3
- False Negative (FN): 0
- True Positive (TP): 12
- Accuracy: 83.33%
- Precision: 80%
- Recall: 100%
- Specificity: 50%
- F1 Score: 88.89%



Model can classify, predicts, identify samples correctly. F1 score is harmonic mean of precision and recall. It combines both precision and recall into a single metric.

Conclusions

From the EDA, Visualization and Dashboard creation we can conclude that,

- More amount of flight number will impact greater success rate at launch site.
- Launch success rate increases in 2013 till 2020.
- VLEO, SSO,HEO,GEO,ES-L1 orbits have maximum success rate.
- KSC-LC-39A had most successful launches of any site.
- The machine learning algorithm predicts the decision tree as best preforming model with high accuracy of 87%.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!