IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Krushanthan Ragunathan
June 18th, 2024

Outline

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

Executive Summary

Use machine learning models to predict whether the first stage of a Falcon 9 rocket will successfully land based on data collected from previous launches.

- Summary of methodologies
  - Data Collection using SpaceX API
  - Data collection using a web scraping approach
  - Data wrangling
  - Expletory Data Analysis with SQL commands
  - Data visualization using Dash
  - Interactive data visualization analysis using Folium
  - Prediction using machine learning models and algorithms

- Summary of all results
  - Results from expletory data analysis and interactive maps
  - Outputs derived from the machine learning models and key parameters
  - Performance comparison results of the machine learning models

Introduction

- Project background and context

SpaceX is promising to revolutionize the aerospace industry with low-cost launches. Falcon 9's stage rocket can land back on Earth successfully and is ready for the next mission. This approach will significantly reduce the cost for every launch mission.

- Problems you want to find answers
  - What attributes play a crucial role in a successful landing?
  - What are the external operating conditions determining a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and Web Scrapping methods.
- Perform data wrangling
  - Outcome values were summarized and converted to True/ False indicators.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- Methods explained below were followed during the Data Collection
  - Space X API was invoked first to retrieve the data
  - Using the **json()** function, the API response was formatted and converted to a Data Frame with **.json_normalize**.
  - Using the web scraping method, Falcon 9 launch records were acquired from Wikipedia using **BeautifulSoap**.
  - Wikipedia data then converted to Data Frame for further analysis.

Data Collection – SpaceX API

SpaceX API calls notebook (Please click the link view the Notebook)

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
response = requests.get(static_json_url)
print(response.status_code)

data=pd.json_normalize(response.json())
print(data.head())

# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]
```

Data Collection - Scraping

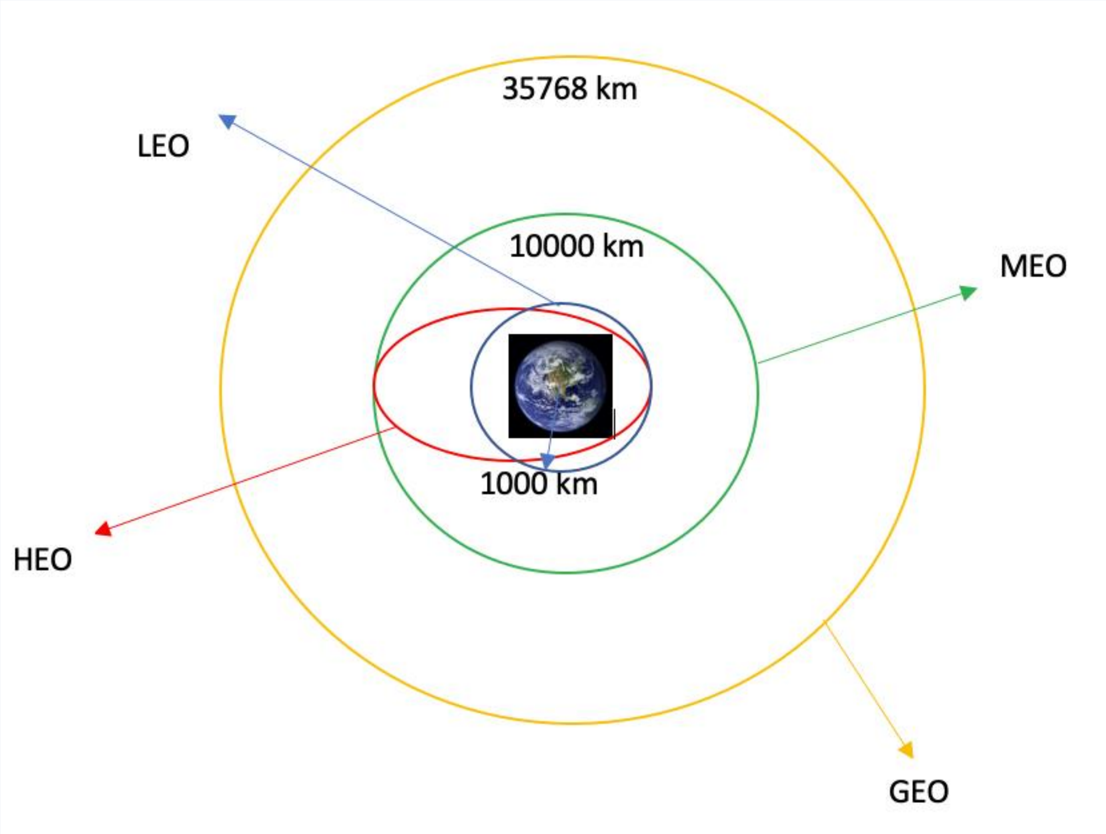
[Web scraping notebook]
(Please click the link to view the Notebook)

```python
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"

# Request the HTML page from the above URL and get a `response` object
response = requests.get(static_url)
soup = BeautifulSoup(response.text)
print(soup.title)


# Find all tables and assign the result to a new list called `html_tables`
html_tables = soup.find_all('table')
# Let's print the third table and check its content
first_launch_table = html_tables[2]
#print(first_launch_table)

column_names = []
html_headers = first_launch_table.find_all('th')
for header in html_headers:
    name = extract_column_from_header(header)
    if str(name) != 'None' and len(str(name)) > 0:
        column_names.append(name)

#print(column_names)

launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']
```

Data Wrangling

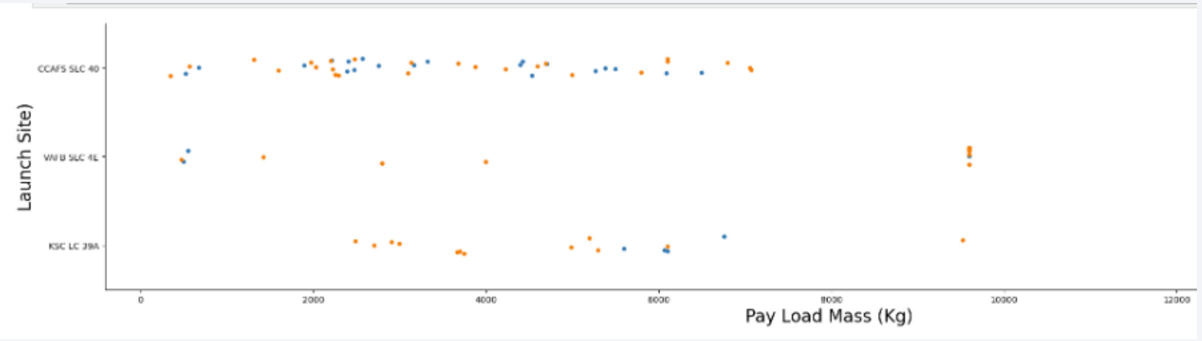
- Calculated the number of launches at each site and the number and occurrence.
- Calculated the number and occurrence of mission outcomes of the orbits.
- Created landing outcome label from the outcome column and exported the results to a CSV file.

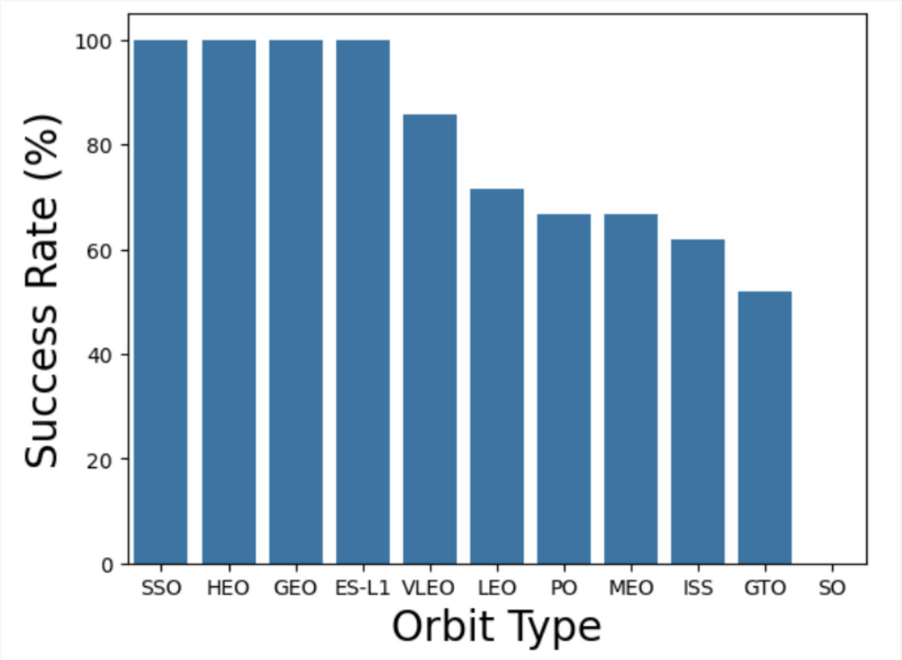
[Data wrangling related notebook]



EDA with Data Visualization

- Explored the data for the given attributes.
  - Flight Number vs Launch Site
  - Pay Load vs Launch Site
  - Orbit Type vs Success Rate
  - Flight Number vs Orbit Type
  - Pay Load vs Orbit Type
  - Launch Year vs Success Rate





[Data visualization notebook]()

EDA with SQL

- Performed the queries below after successfully loading the CSV file into a database.
    - Names of the unique launch sites
    - Launch Site names begin with “CCA.”
    - Total payload mass carried by boosters launched by NASA (CRS)
    - Average payload mass carried by booster version F9 v1.1
    - Total number of successful and failed mission outcomes
    - Failed landing outcomes in drone ship, their booster version, and launch site names

EDA with SQL notebook

Build an Interactive Map with Folium

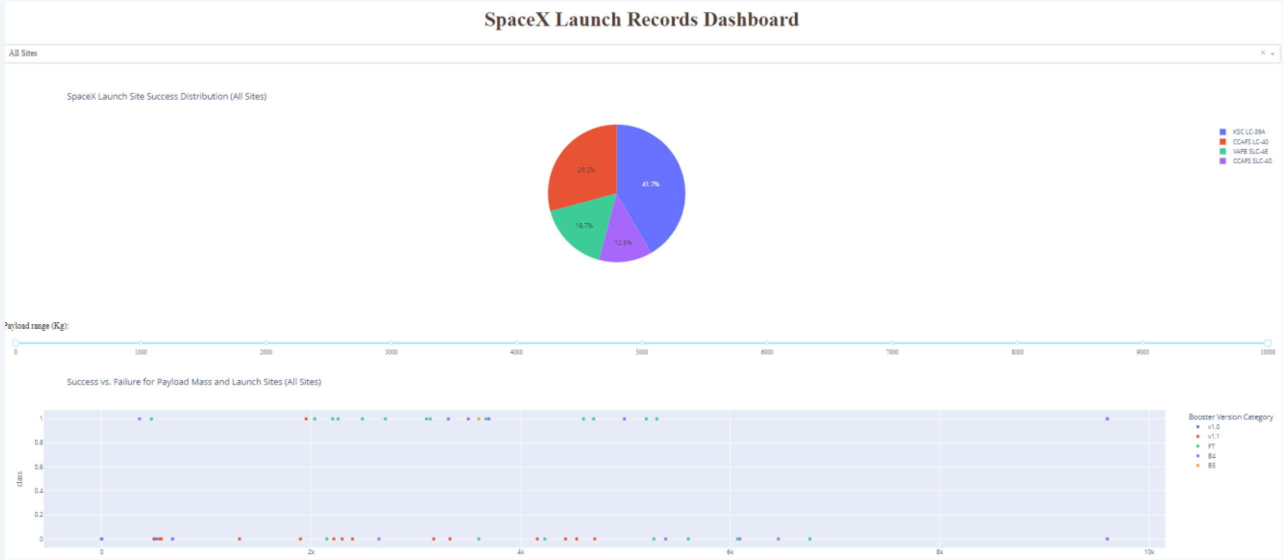
- All launch sites have been marked with map objects such as markers and circles to indicate the success or failure of the launch outcome.
- We've classified the launch outcomes into two categories, Class=1 for Success and 0 for Failure. This classification is visually represented on the interactive map with color indicators, green for Success and red for Failure, making it easy to interpret the data.
- Calculated distance between the launch sites and their proximities, such as nearest coastal lines, railroad, city, and highways.

Interactive map with the Folium map (Please refer to the screenshot folder to review map outcome)

Build a Dashboard with Plotly Dash

The mentioned plots were created using Plotly Dash.

- Pie Chart – to show the total launch by sites and success/ failures in selected sites.
- Scatter Plot – to show the relationship between Outcome and Payload Mass (kg) for different booster versions.



[Plotly Dash lab]

Predictive Analysis (Classification)

- Summary of the model development used to predict the success rate of the first stage launch.
  - Create a NumPy array from the column Class in data.
  - Data standardization.
  - Train and Test data split.
  - Identify the best Hyperparameter for Logistic Regression, SVM, Decision Tree, and KNN classifiers using GridSearchCV.
  - Calculate the accuracy of each model to compare the model performance.

- [Predictive Analysis Notebook]

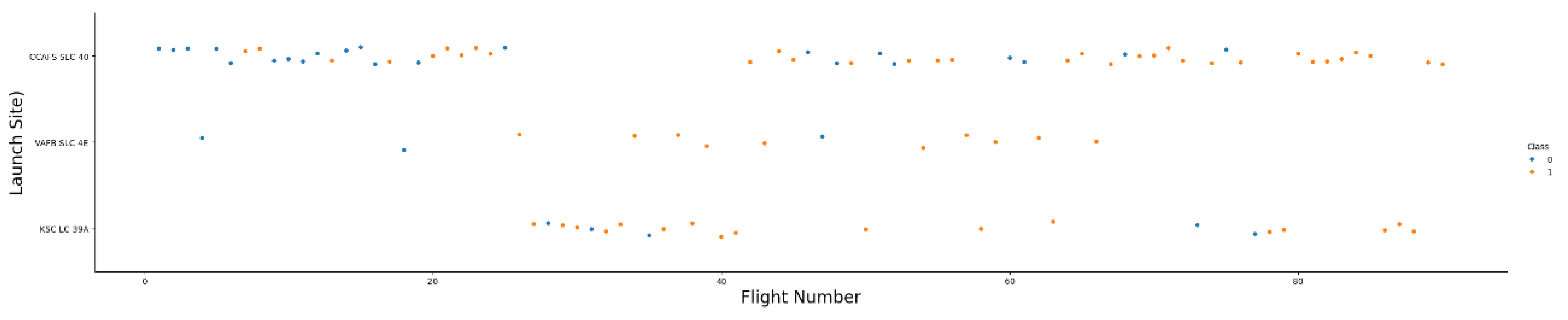
Results

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

Section 2

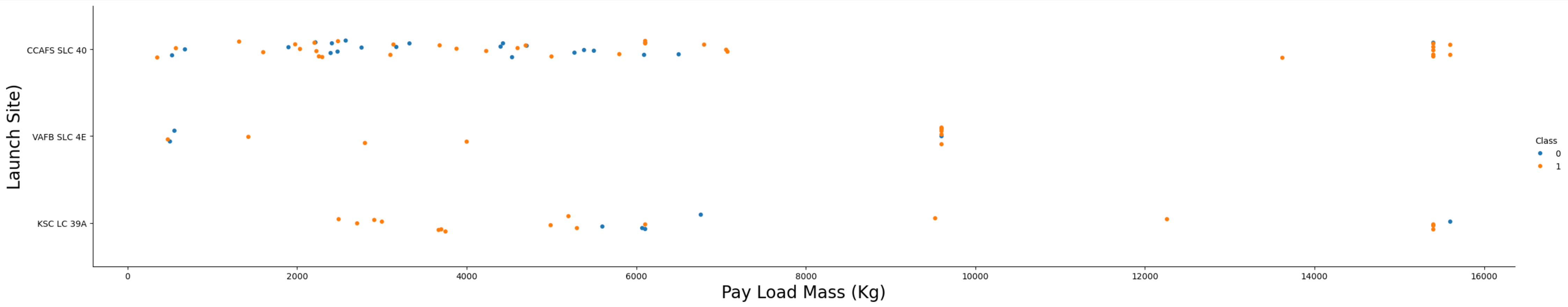
Insights drawn from EDA

Flight Number vs. Launch Site



- With time (assuming flight number increases over time), each site's success rate has increased.
- Launch Site CCAFS SLC 40 plays a significant role in the mission data.

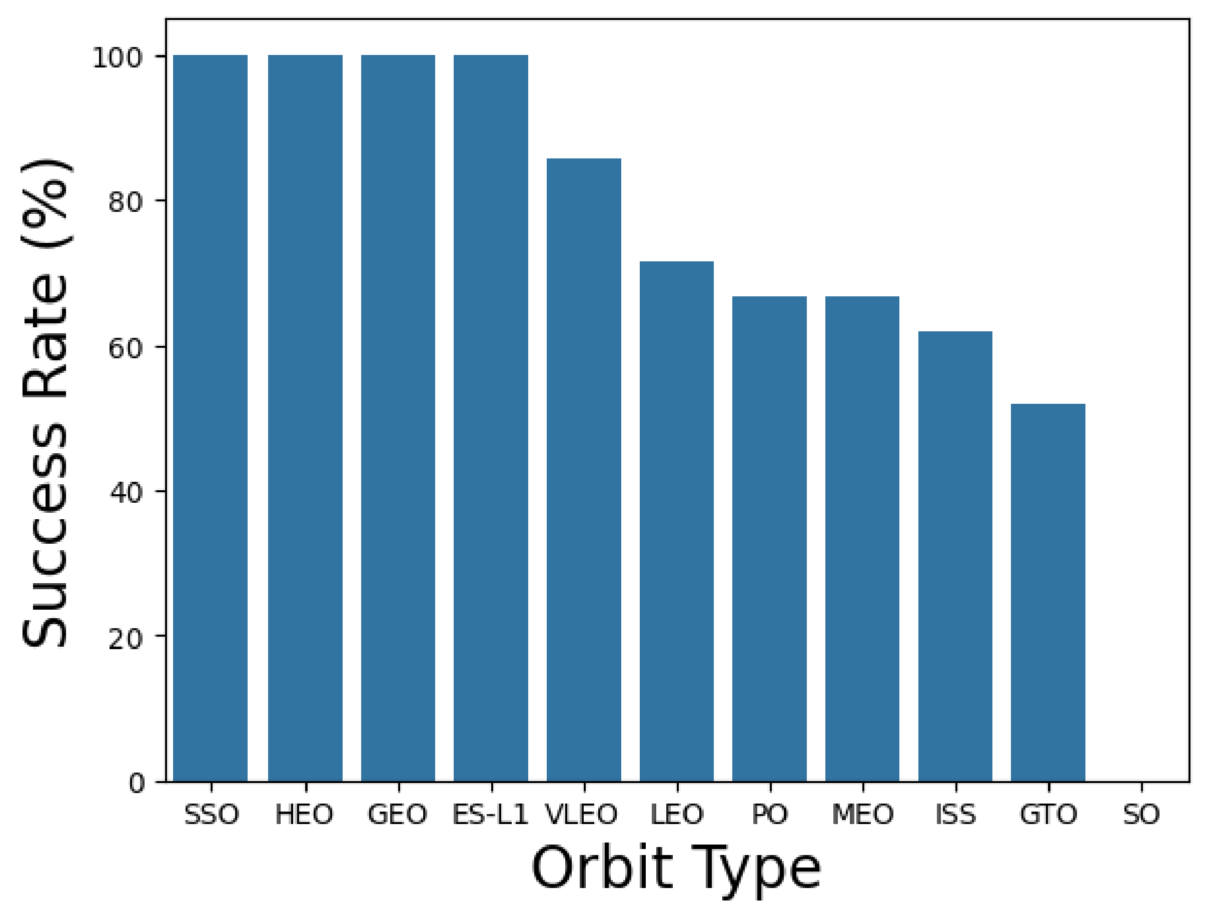
Payload vs. Launch Site



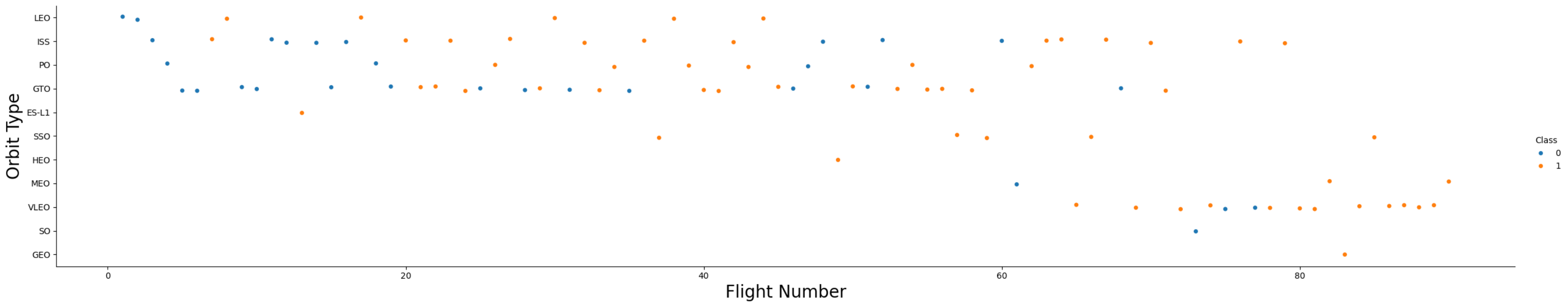
- The majority of the rockets launched had a payload of less than 7500kg.
- Out of 7 missions carried out with a payload between 7500-15750kg, six were successful.

Success Rate vs. Orbit Type

- From the plot results, we can say that SSO, HEO, GEO, and ES-L1 had the most successful missions.
- Adding the number of missions for each Orbit will give a better understanding of the success rate for comparison.

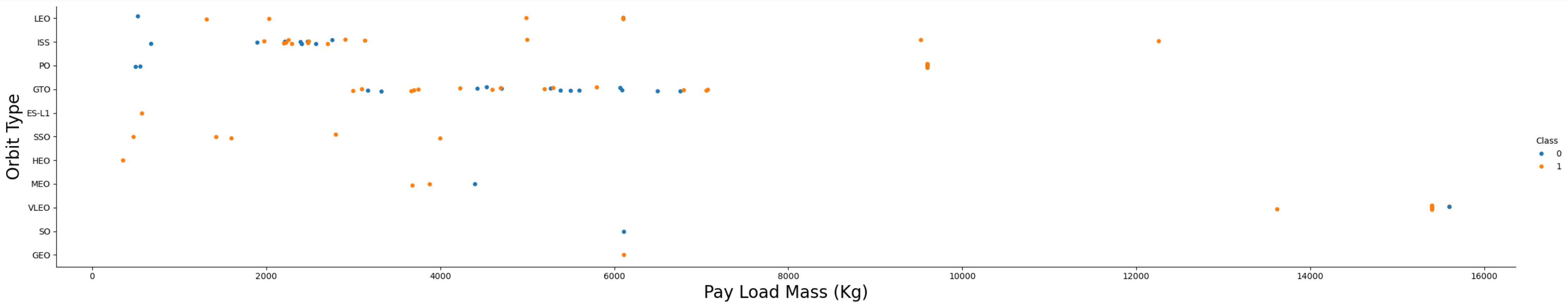


Flight Number vs. Orbit Type



- As pointed out in the previous slide, Orbits has a higher success rate but fewer launches than others.
- There is no significant relationship between Orbit Type and Flight Number.

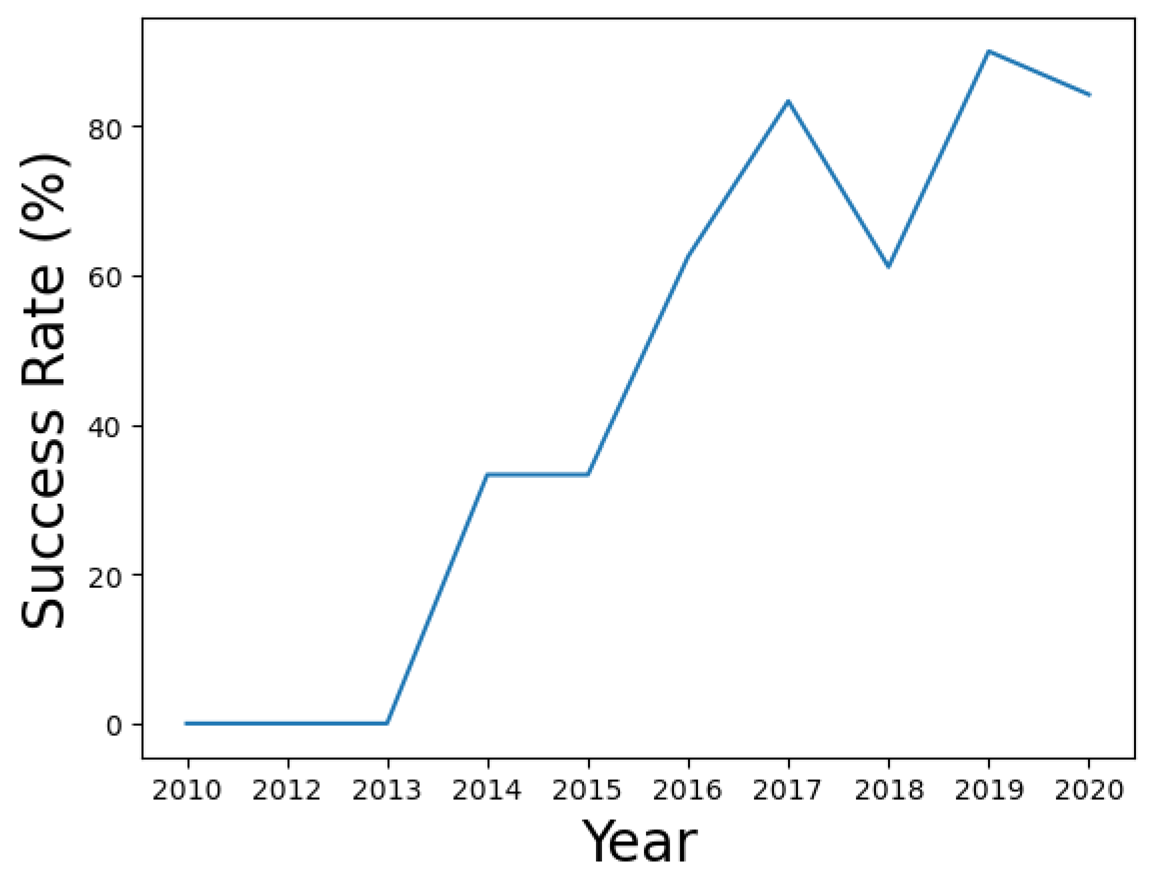
Payload vs. Orbit Type



- With heavy payloads, the success rate is high for PO, LEO, and ISS.
- However, for GTO, we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are there here.

Launch Success Yearly Trend

- Success rate has continuously shown an upward trend since 2013.



All Launch Site Names

- The **DISTINCT** keyword in the query brings the unique launch sites from SpaceX Data.
- Four unique launch sites were found.

```
%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'

- “**LIKE ‘CCA%’**” in the Where clause will list down the Launch Sites name starting with “CCA.”

```
%sql select Launch_Site from SPACEXTABLE where Launch_Site like 'CCA%' LIMIT 5
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- **SUM** keyword will calculate the total of the PAYLOAD_MASS_KG column.

```
%sql Select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer = 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- **AVG** keyword is used here to calculate the average of the PAYLOAD_MASS_KG column.

```
%sql Select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- The **MIN** keyword finds the least value from the Date column.

```
%sql select MIN(Date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
```

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

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The condition used in the WHERE clause will filter the successful landings on drone ships. Then, the AND condition is used to find the successful landings with a payload mass greater than 4000kg but less than 6000kg.

```
%sql select distinct Booster_Version
from SPACEXTABLE
where Landing_Outcome = 'Success (drone ship)'
and PAYLOAD_MASS__KG_ > 4000
and PAYLOAD_MASS__KG_ < 6000
```

Out[18]:

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

Total Number of Successful and Failure Mission Outcomes

- There are multiple keywords used here.
  - Distinct – find the unique Mission Outcomes.
  - Count – total number of each unique mission outcome.
  - Group by-group, the results are based on the mission outcome value.

In [19]:

```
%sql select distinct Mission_Outcome, count(*) from SPACEXTABLE group by Mission_Outcome
```

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

Out[19]:

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

Boosters Carried Maximum Payload

- The **MAX** keyword is used to find the largest value available for the Payload column.

```
%sql select Booster_Version
from SPACEXTABLE
where PAYLOAD_MASS__KG_ = (
        select max(PAYLOAD_MASS__KG_)
        from SPACEXTABLE
    )
```

Out[20]:

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

- **Substr** function is used to pick the Month/ Year from the full date value.

```
%sql select substr(Date, 6,2) as Month,
Landing_Outcome,
Booster_Version, Launch_Site
from SPACEXTABLE
where Landing_Outcome = 'Failure (drone ship)'
and substr(Date,0,5) = '2015'
```

Out[21]:

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

- Keyword **DESC** is used to present the results in descending order.

```
%sql select Landing_Outcome, count(*) as 'Count'
from SPACEXTABLE
where Date between '2010-06-04' and '2017-03-20'
group by Landing_Outcome
order by Count desc
```

Out[22]:

Landing_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

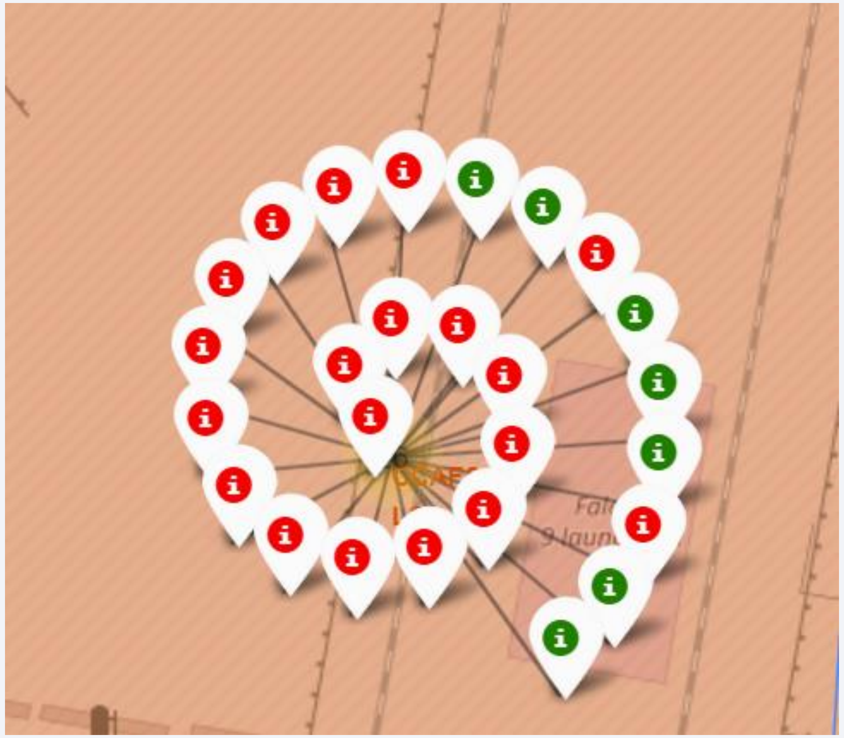
Section 3

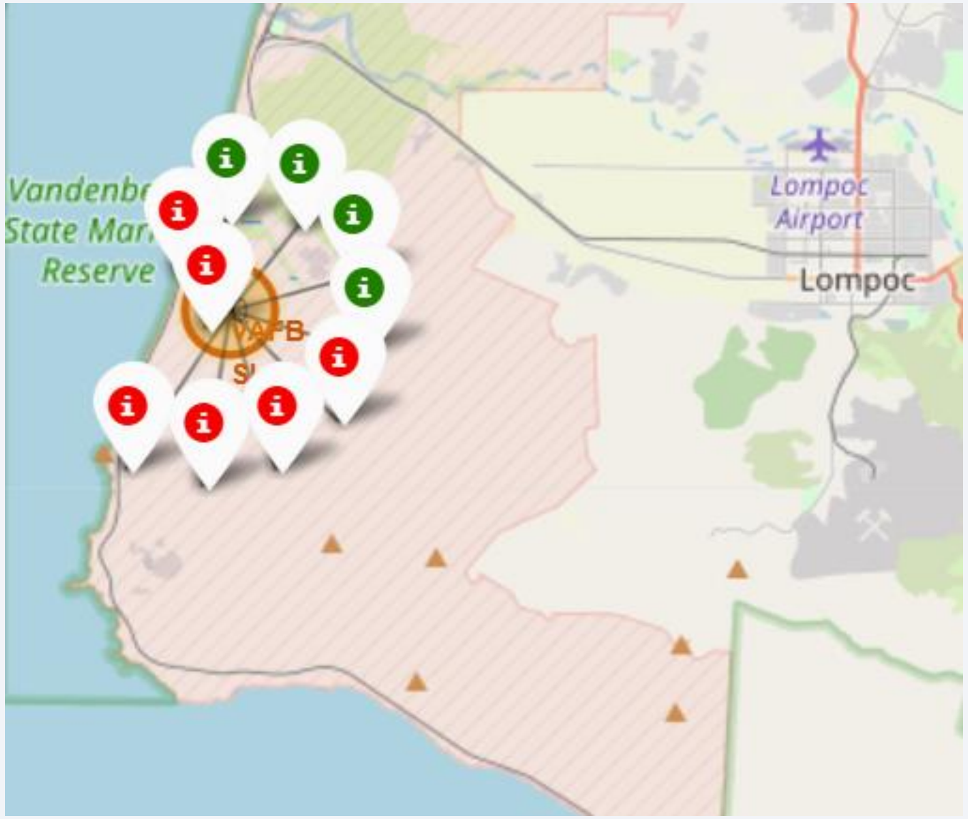
Launch Sites Proximities Analysis

All Launch Sites

- All the launch sites are situated only in the United States.
- All of them are strategically constructed near the coastal lines in CA and FL.

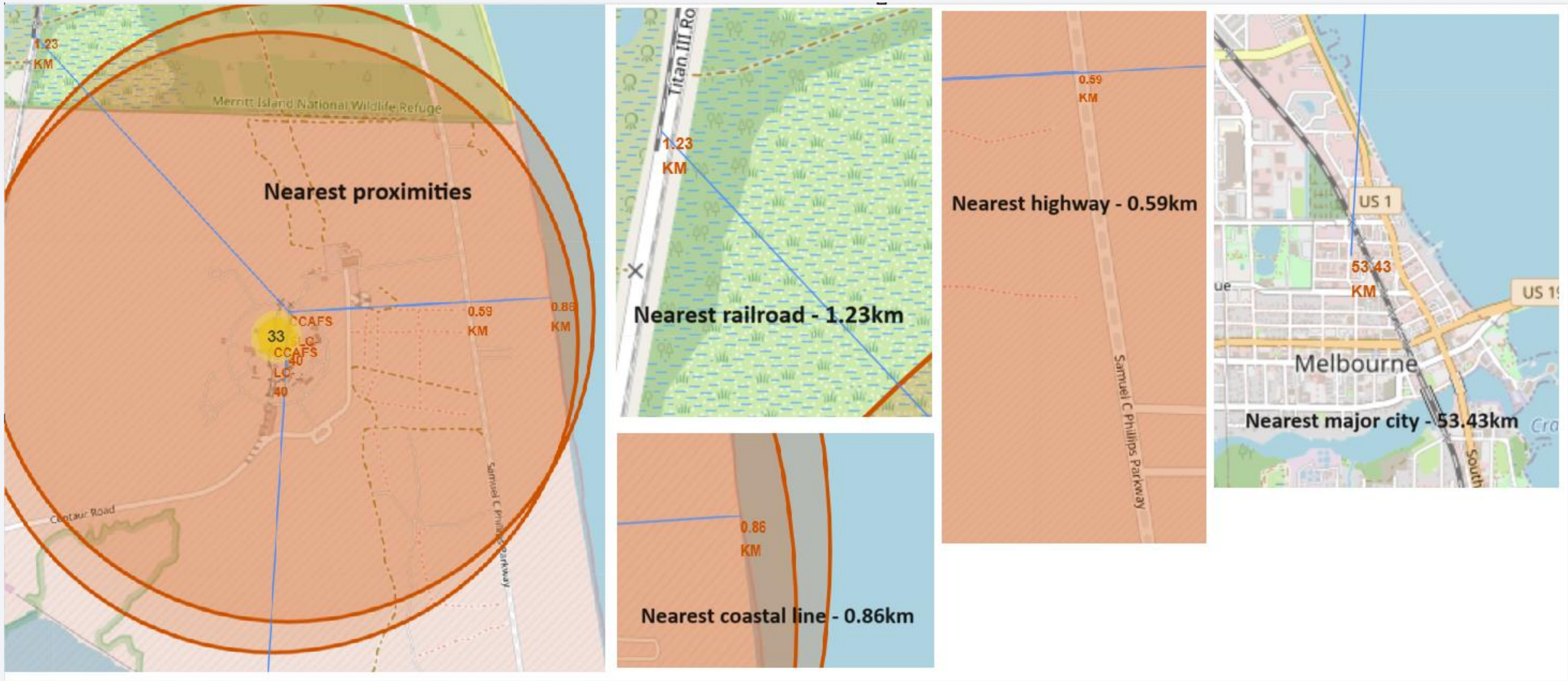
Success/ Failure Markers





- The above map indicates successes and failures at a launch site with color-coded markers (Green-success, red-failure).

Launch Sites with Proximities

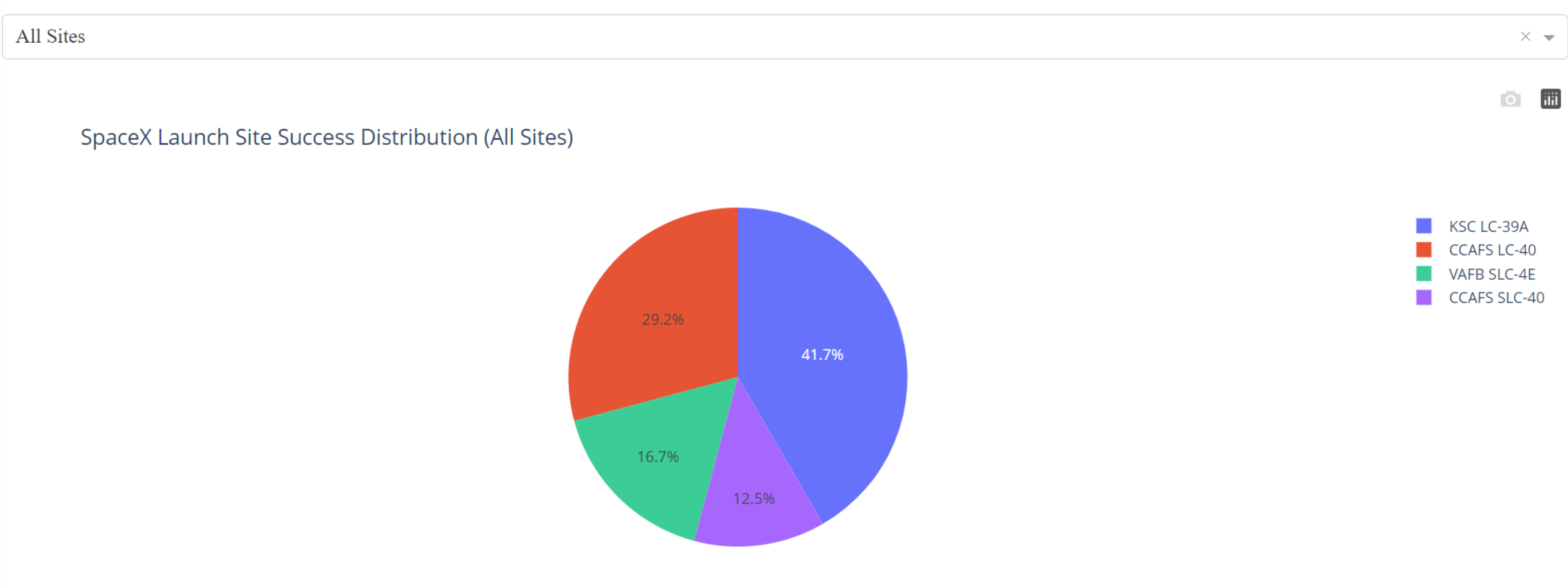


- The launch site is very near landmarks such as the railroad and highways, ensuring easy access to supplies.
- Meanwhile, the launch site is far from the nearest major city but near the coastal line to avoid disasters.

Section 4

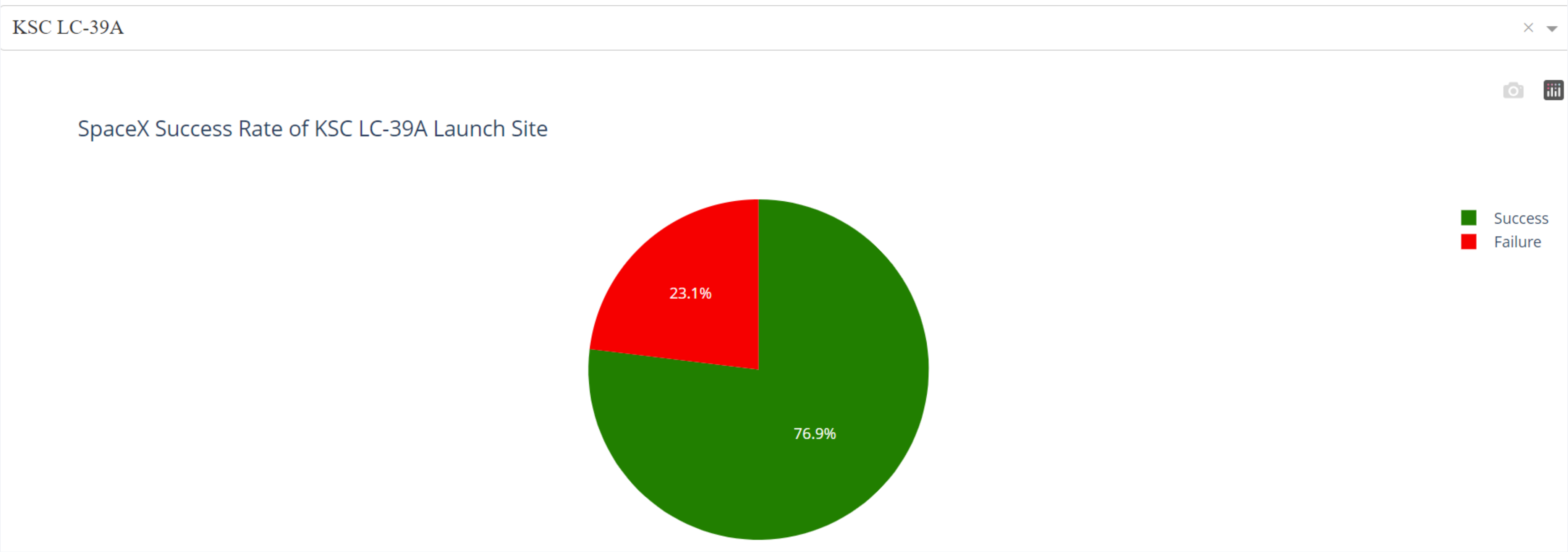
Build a Dashboard with Plotly Dash

Successful Missions for All Sites



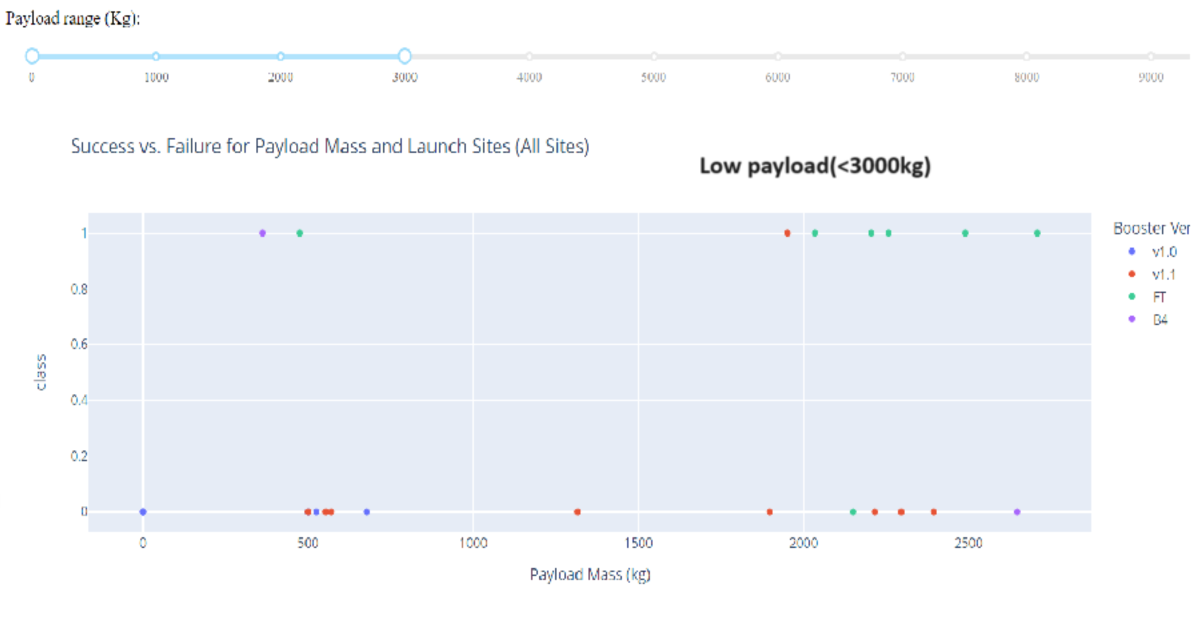
- Launch Site KSC LC-39A has the highest success rate of missions, followed by CCAFS LC-40.

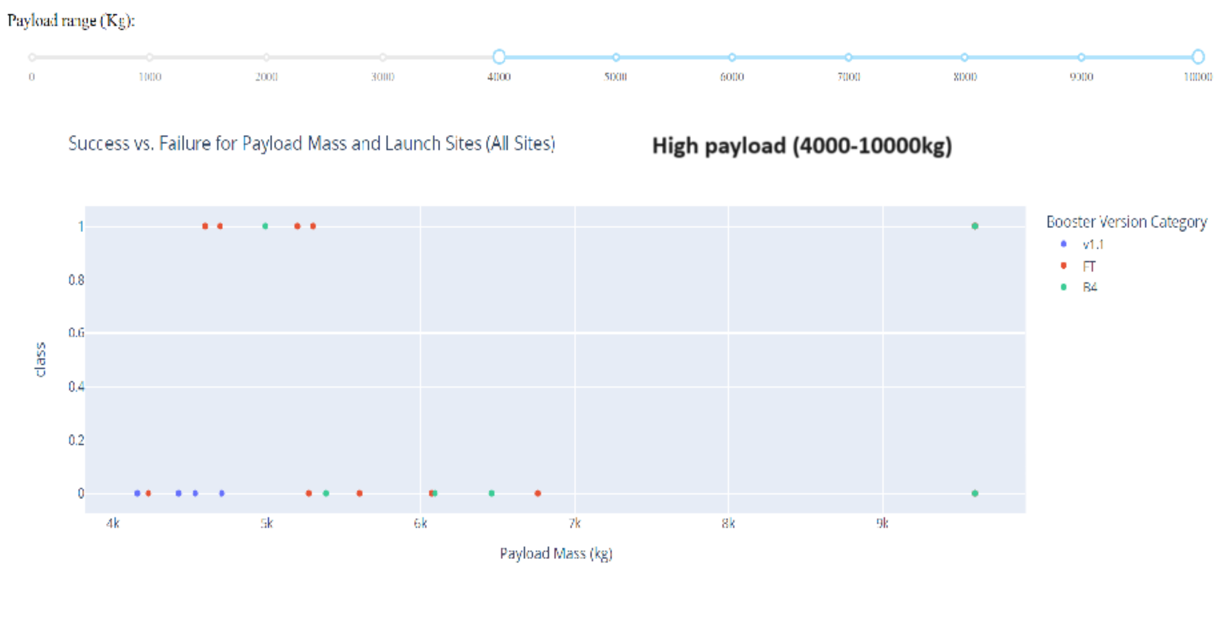
Launch Site drill down



- Most successful Launch Site KSC LC-39A has 76.9% success rate.

Payload vs Launch Outcome Analysis





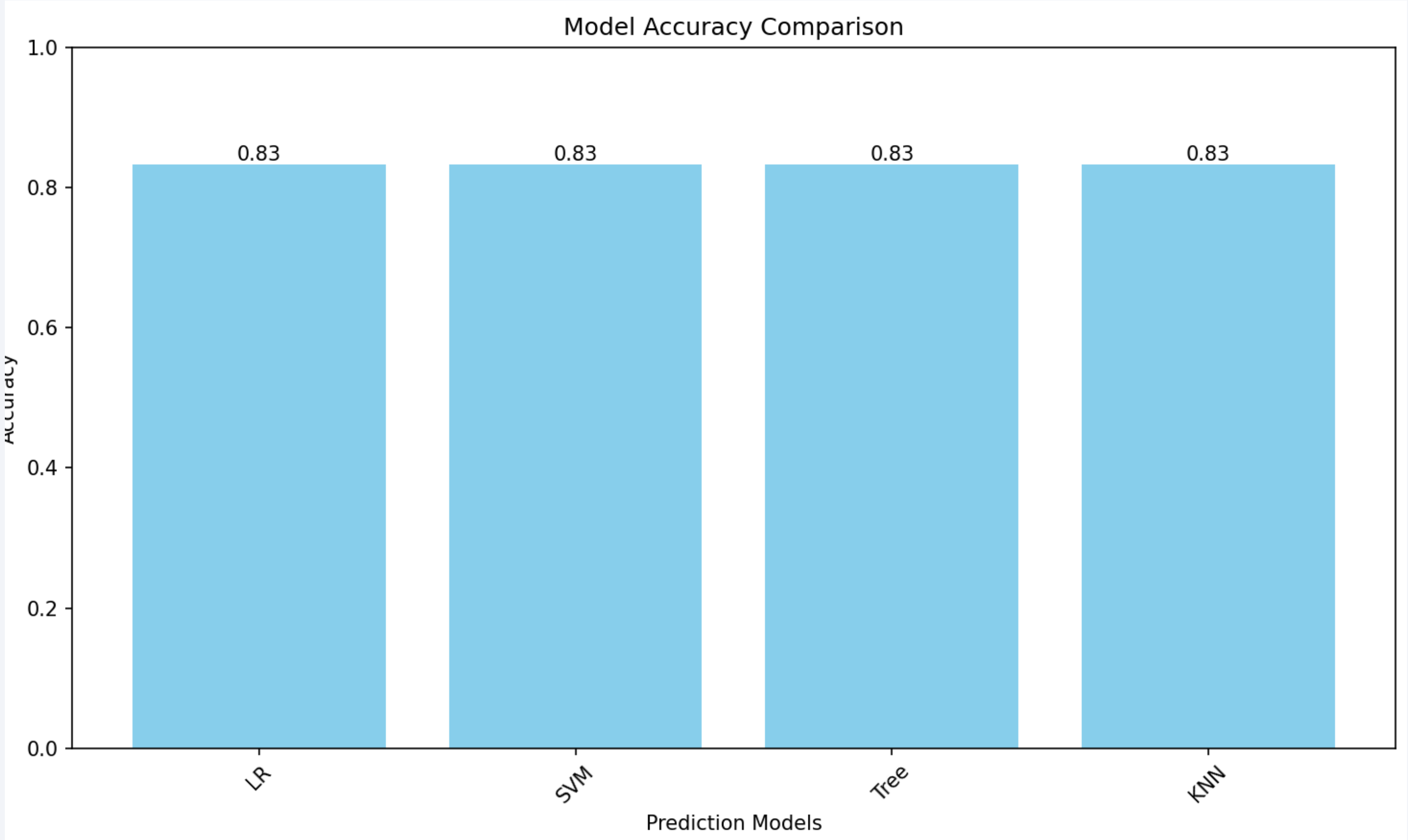
- Payload < 3000kg has a higher success rate than higher payload missions.
- Based on the pattern, low to medium amounts of payload yielded successful missions.
- Thus, prove that, as we have seen in slide #19, payload < 7500kg has a high success rate.

Section 5

Predictive Analysis (Classification)

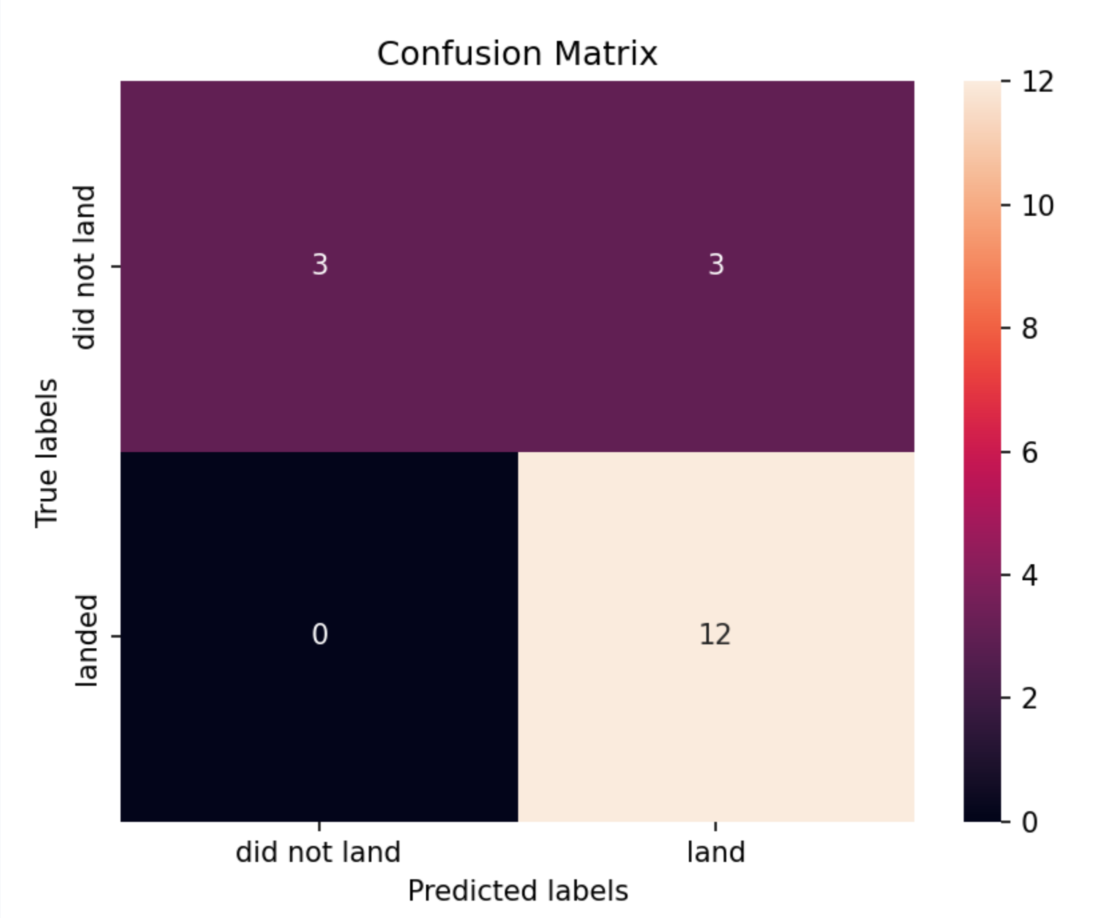
Classification Accuracy

- All four of the models have achieved the same accuracy level.



Confusion Matrix

- All four models have shown the same pattern.
- **True Positive** – Model correctly predicted 12 times
- **False Positive** – The model incorrectly predicted three times when it was not.
- **True Negative** – The model correctly predicted three unsuccessful landings.
- **False Negative** – The model incorrectly predicted unsuccessful landings 0 times when they were successful.



Conclusions

- All models have shown the same level of accuracy 83 .33%, from our testing.
- Based on the confusion matrix, all four models have the highest rating for predicting successful landings (12 true positives).
- Fine-tuning is needed to address the false positive outcome.
- Our Machine Learning models can be deployed to predict the first stage of the landing and predict the cost for the launch mission.

Appendix

- All the Python code, Notebooks, Data Sets and Screenshots can be found here: https://github.com/rkrushanthan/IBM-Data-Science-Capstone

Thank you!