IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Martin Angelov Ganchev
23.10.2023

Outline

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

Executive Summary

- Summary of methodologies
    1. Data Collection via SpaceX API
    2. Web Scraping Data
    3. Data Wrangling
    4. Exploratory Data Analysis with SQL
    5. Exploratory Data Analysis with Python Libraries
    6. Launch Sites Analyses with Folium
    7. Interactive Dashboards
    8. Machine Learning for Landing Prediction
- Summary of all results
    1. Data Analysis Results
    2. Visualised Data and Interactive Dashboard
    3. Predictive Analyses

Introduction

Project background and context

- SpaceX is a successful company in the space business that has numerous accomplishments related to sending spacecrafts to space. SpaceX provides online advertisements about Falcon 9 rocket launches that cost around 62 million dollars each. Hence, through data science it can be analysed if one of these launches lands successfully that can help better understand the cost for the launches. The analysis is done on the first stage which is crucial for the launches, and it depends on many factors of the first stage if it lands successfully, does not land, or is sacrificed.

Problems you want to find answers

- In this project, data science is utilised to predict if the first stage will land successfully by using data provided from previous rocket launches.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

Data was initially collected through SpaceX API by making a request by using functions that support the extraction of information from the API by using identification numbers and requesting rocket launch data.

The data was then parsed using a **get** request and then converted into a **pandas** data frame.

Webscraping was used to collect data from Wikipedia records using **BeautifulSoup** which was also converted into a **pandas** data frame afterward.

Data Collection – SpaceX API

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_
```

We should see that the request was successfull with the 200 status response code

```
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
data = pd.json_normalize(response.json())
```

- Link to NoteBook:

https://github.com/GanchevM00/Capstone-Project/blob/main/1.Data%20collection.ipynb

Data Collection - Scraping

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

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use  method with the provided static_url
data  = requests.get(static_url).text
print(data)

# assign the response to a object
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

- Link to NoteBook: https://github.com/GanchevM00/Capstone-Project/blob/main/2.%20Webscraping.ipynb

Data Wrangling

- Data was filtered using **BoosterVersion** to keep only Falcon 9 launches, dealt with missing values in the columns and replaced data using the mean were required.

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`:

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
df['Class'] = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
df['Class'].value_counts()
```

```
1    60
0    30
Name: Class, dtype: int64
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

```
landing_class=df['Class']
df[['Class']].head(8)
```

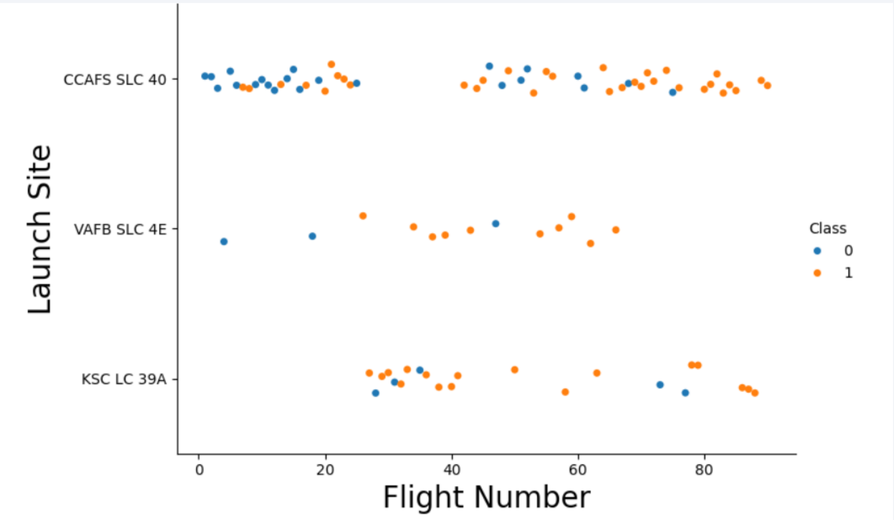
	Class
0	0
1	0

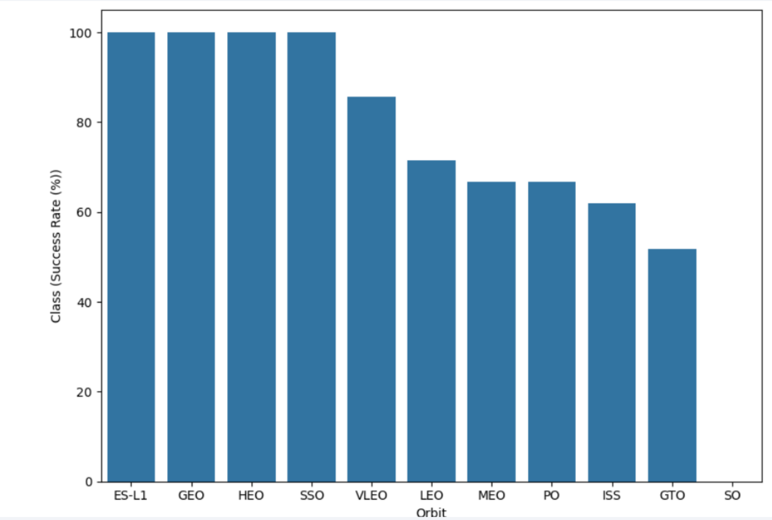
- Link to Notebook:

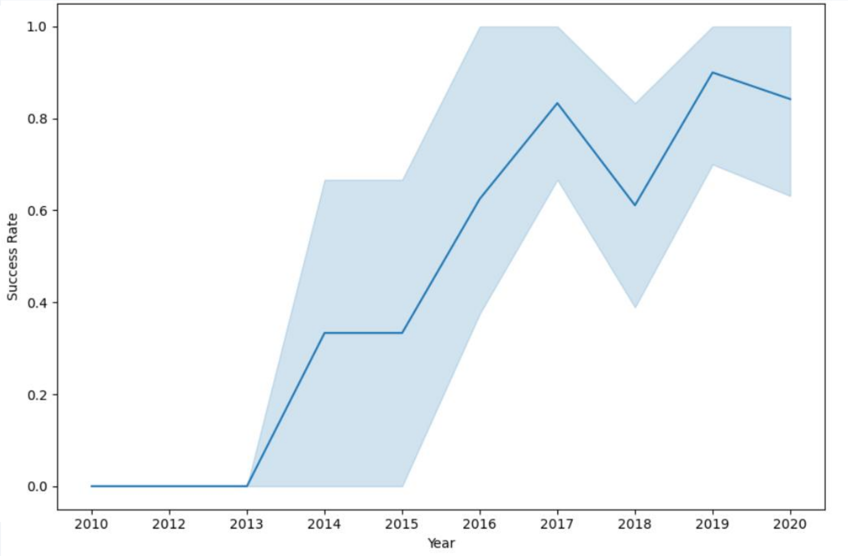
https://github.com/GanchevM00/Capstone-Project/blob/main/3.%20Data%20Wrangling.ipynb

EDA with Data Visualization

- EDA with Data Visualization was performed using **Pandas** and **Matplotlib.**
    1. Scatterplots were used to visualise the relationship between Flight Number and Launch Site, Payload and Launch Site, FlightNumber and Orbit Type and Payload and Orbit Type.
    2. Bar chart was used to visualise the relationship between success rate of each orbit type.
    3. Line plot was used to visualise the yearly trend of launch success.







- Link to Notebook:

https://github.com/GanchevM00/Capstone-Project/blob/main/5.%20Data%20Visualisation.ipynb

EDA with SQL

- The following SQL queries were used:

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

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

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

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_), Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

Hint:Use min function

```
%sql SELECT MIN(date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'
```

EDA with SQL continued…

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_ BETWEEN 4001
```

List the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS OUTCOME FROM SPACEXTBL GROUP BY MISSION_OUTCOME
```

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

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

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.

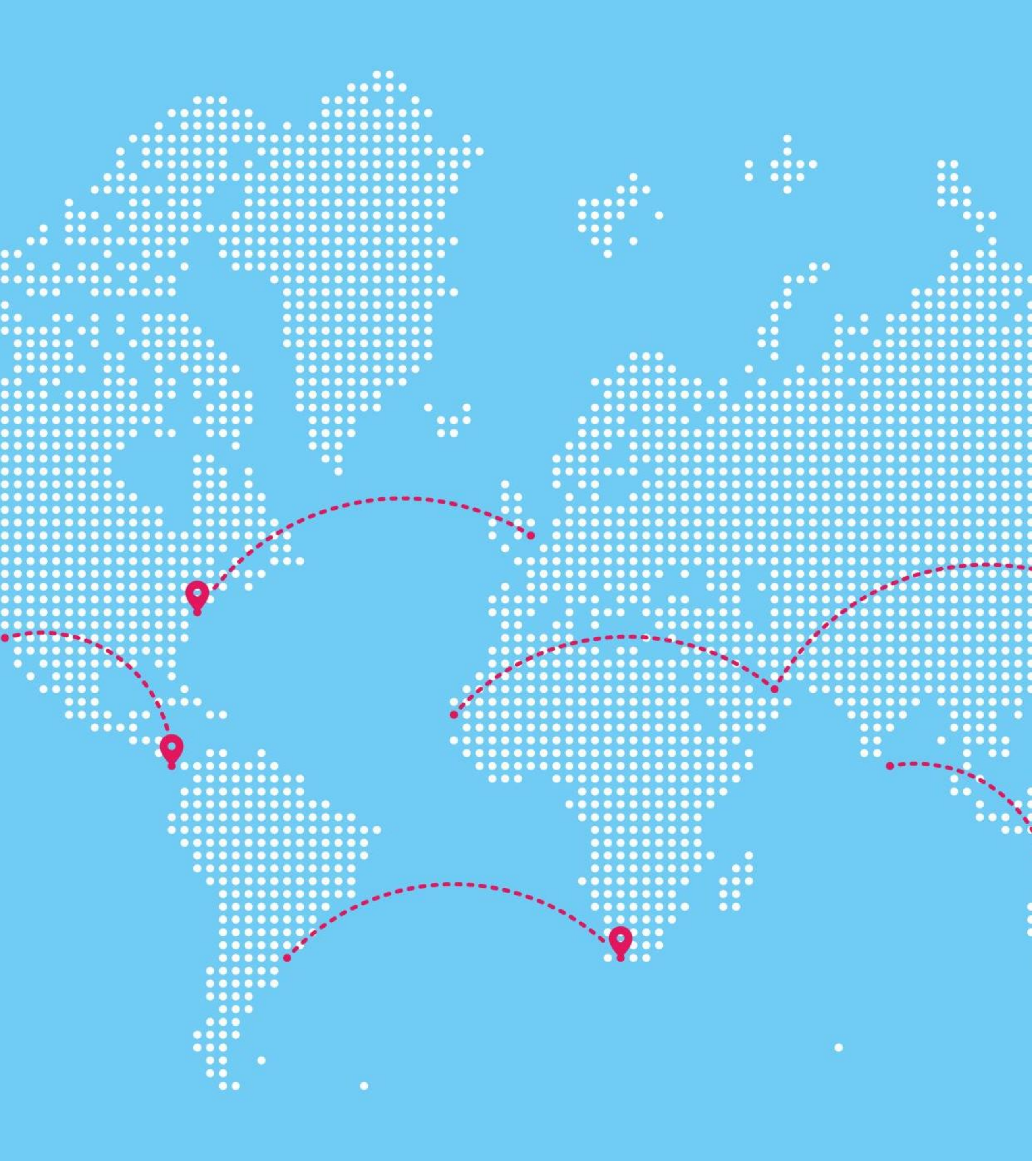
```
%sql SELECT substr(Date,6,2), Booster_Version, Launch_Site, Payload, PAYLOAD_MASS_+KG_, Mission_Outcome,Landing_Outcome FROM
```

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.

```
%sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDEF
```

- Link to Notebook:

https://github.com/GanchevM00/Capstone-Project/blob/main/4.%20SQL%20EDA.ipynb



Folium maps were used to:

- Mark all launch sites
- Create map objects as markers and circles to mark the success or failure of each site.
- Created objects as line to check the distance to the nearest coast from the launch site.

- Link to NoteBook:

https://github.com/GanchevM00/Capstone-Project/blob/main/6.%20Folium%20Maps.ipynb

Build a Dashboard with Plotly Dash

- Plotly Dash was used to create interactive dashboard that:
    - Lists launch sites with Drop-down menu
    - Renders success-pie-charts based on select site.
    - Displays a Range Slider to select Payload.
    - Renders success-payload-scatter-chart scatter plot.
- Link to NoteBook:

https://github.com/GanchevM00/Capstone-Project/blob/main/7.%20Dash.py

Predictive Analysis (Classification)

- Predicitve analysis was conducted with the following steps:
  - Creating NumPy array from the column Class in data by assigning it to variable Y as the outcome variable.
  - Standardise the feature dataset by transforming it.
  - Split data into training and testing sets using **train_test_split** from **sklearn.model_selection** with **test_size** parameter as 0.2 and **random_state** as 2.
  - Create object for each algorithm and create GridSearchCV for each.
  - Fit the training data into the GridSearch object for each to find the best Hyperparameter (**cv=10**).
  - Display the best parameters using the data attribute **best_params_** and the accuracy using **best_score_**
  - Use the method score to calculate the accuracy on the test data for each model and plot a confusion matrix.
- Link to Notebook:

https://github.com/GanchevM00/Capstone-Project/blob/main/8.Machine%20Learning.ipynb

Results

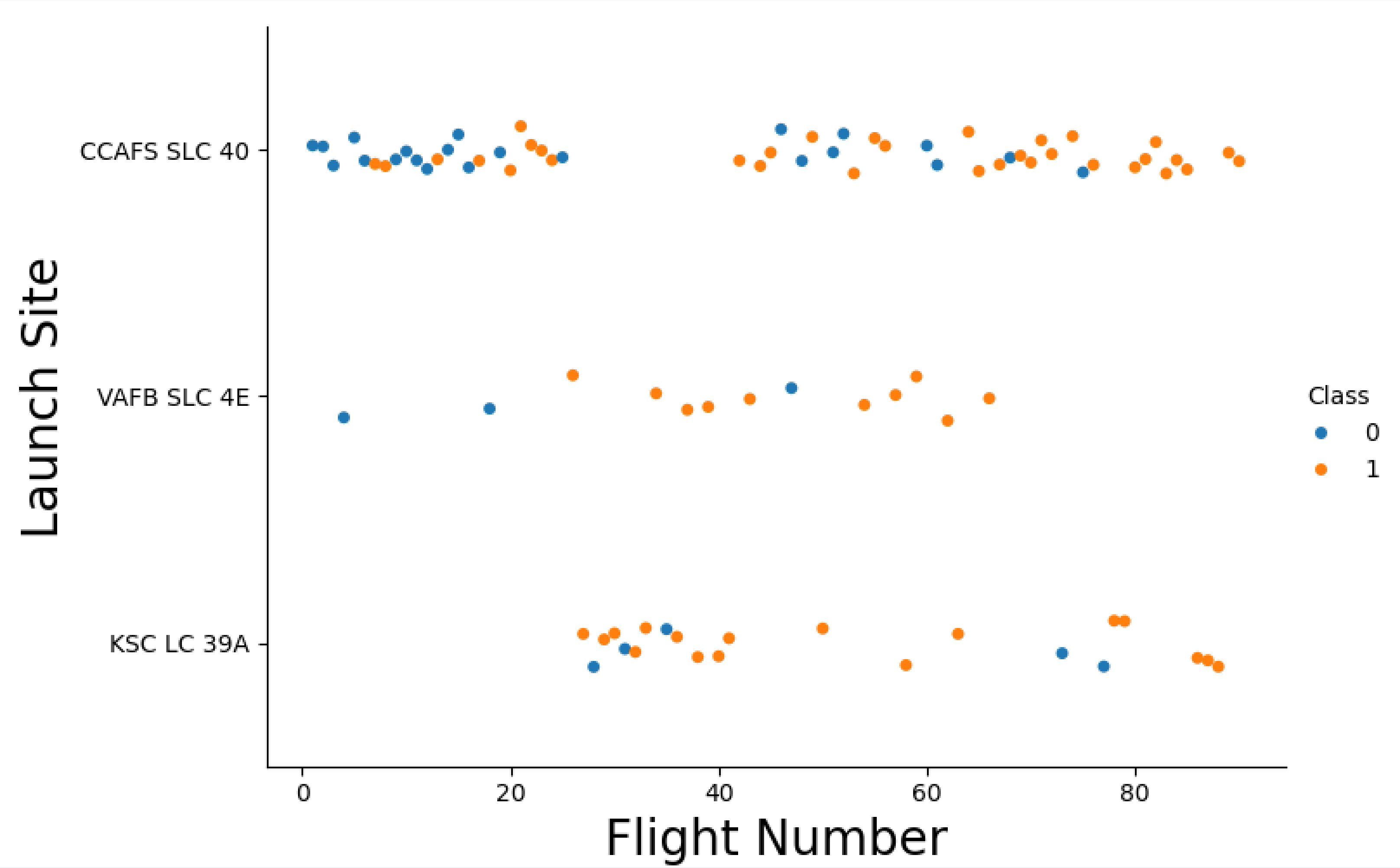
The exploratory data analysis results show that the success rate of Falcon 9 launches was **66.66%.**

The predictive analyses results showed that all models have the same accuracy of **83.33%.**

Section 2

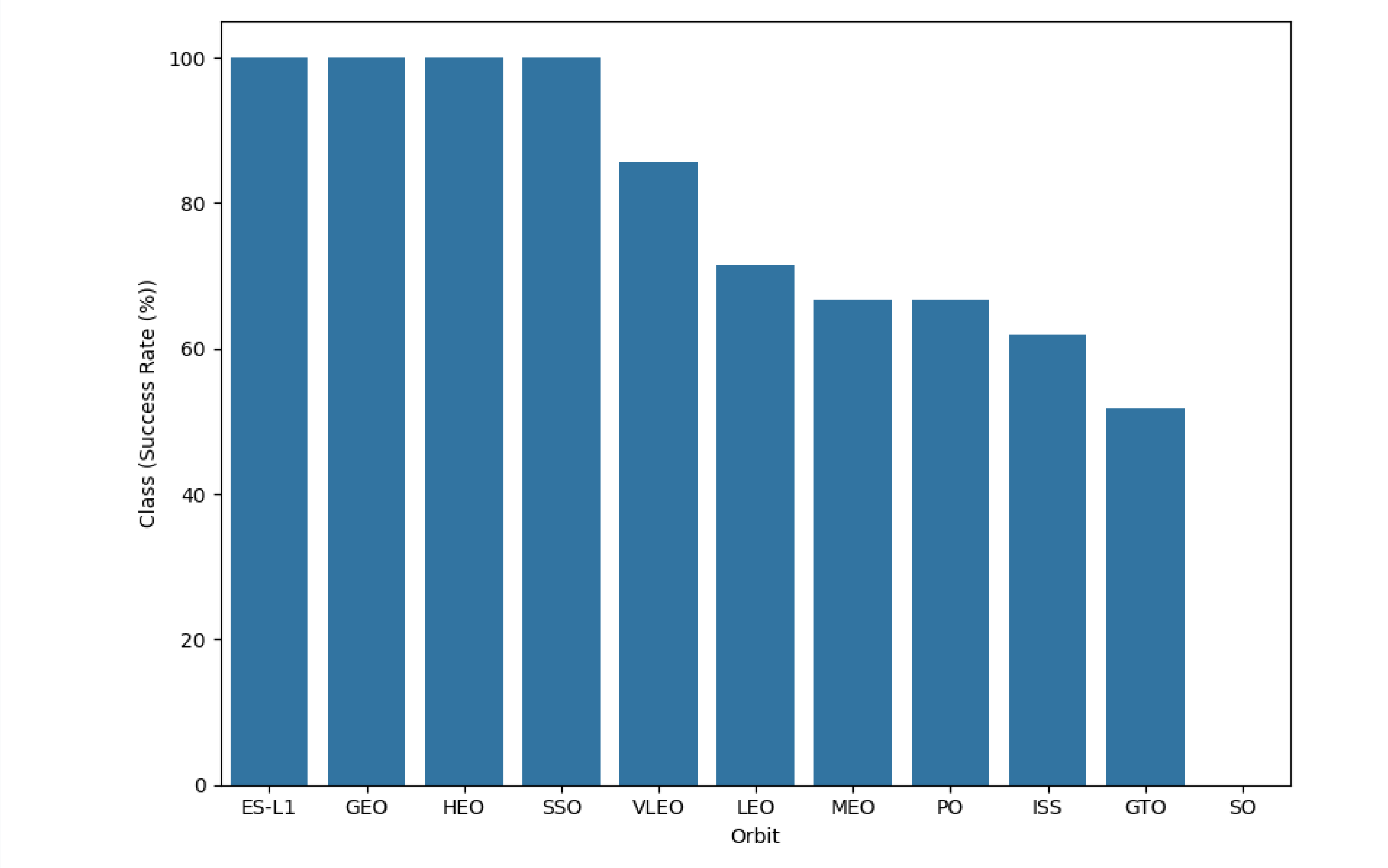
Insights drawn from EDA

Flight Number vs. Launch Site Scatterplot

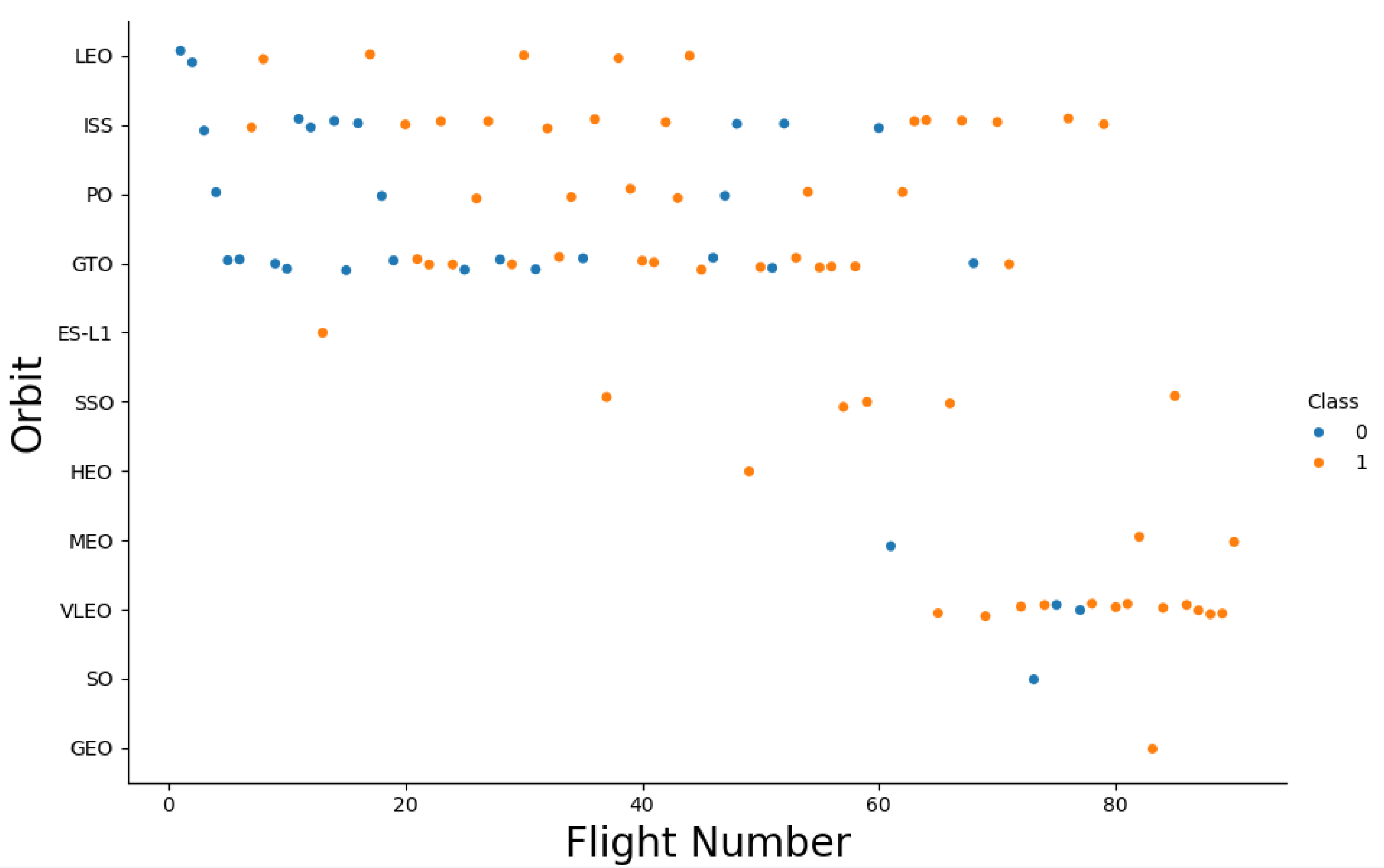


Payload vs. Launch Site Scatterplot

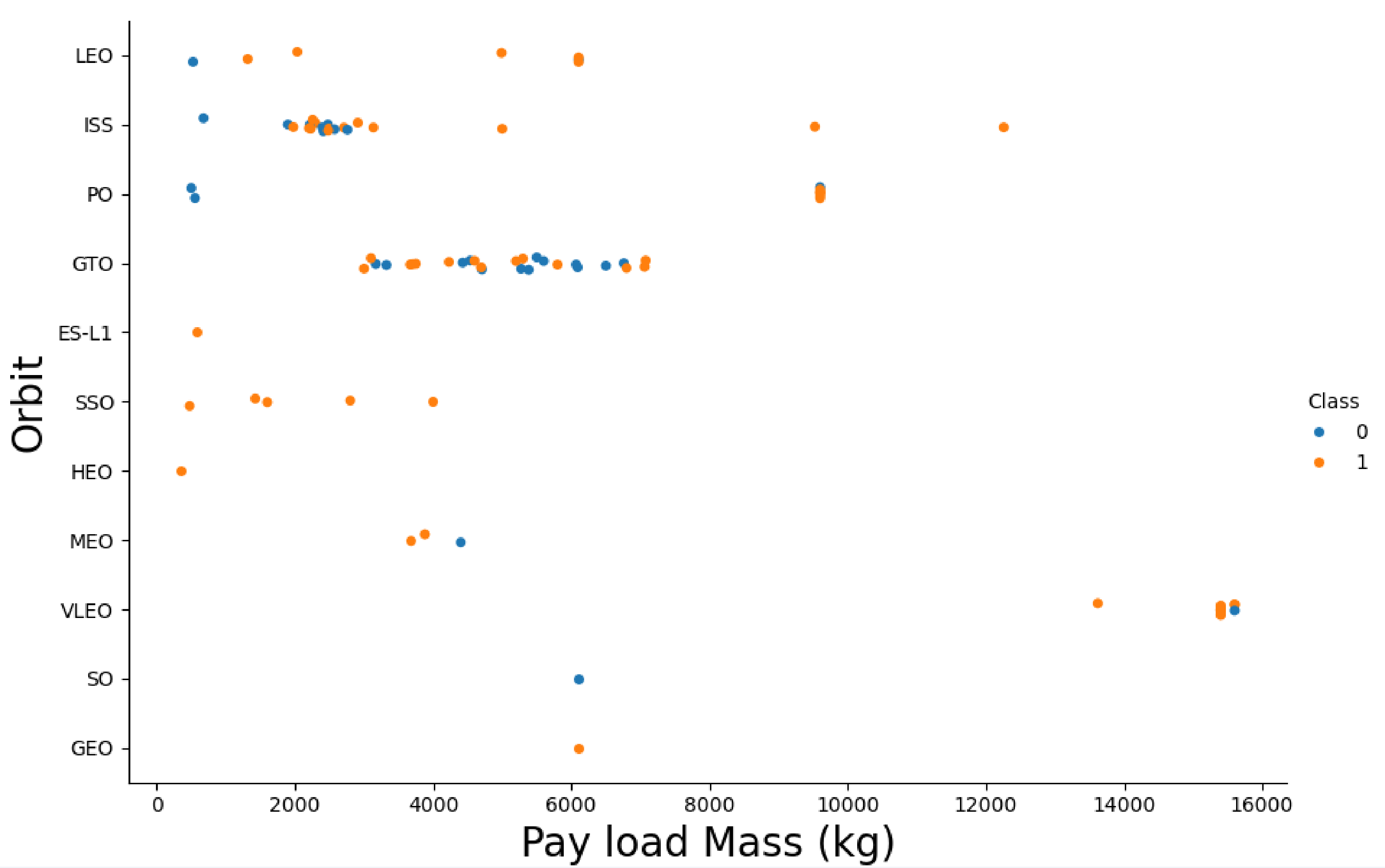
Success Rate vs. Orbit Type Bar Chart



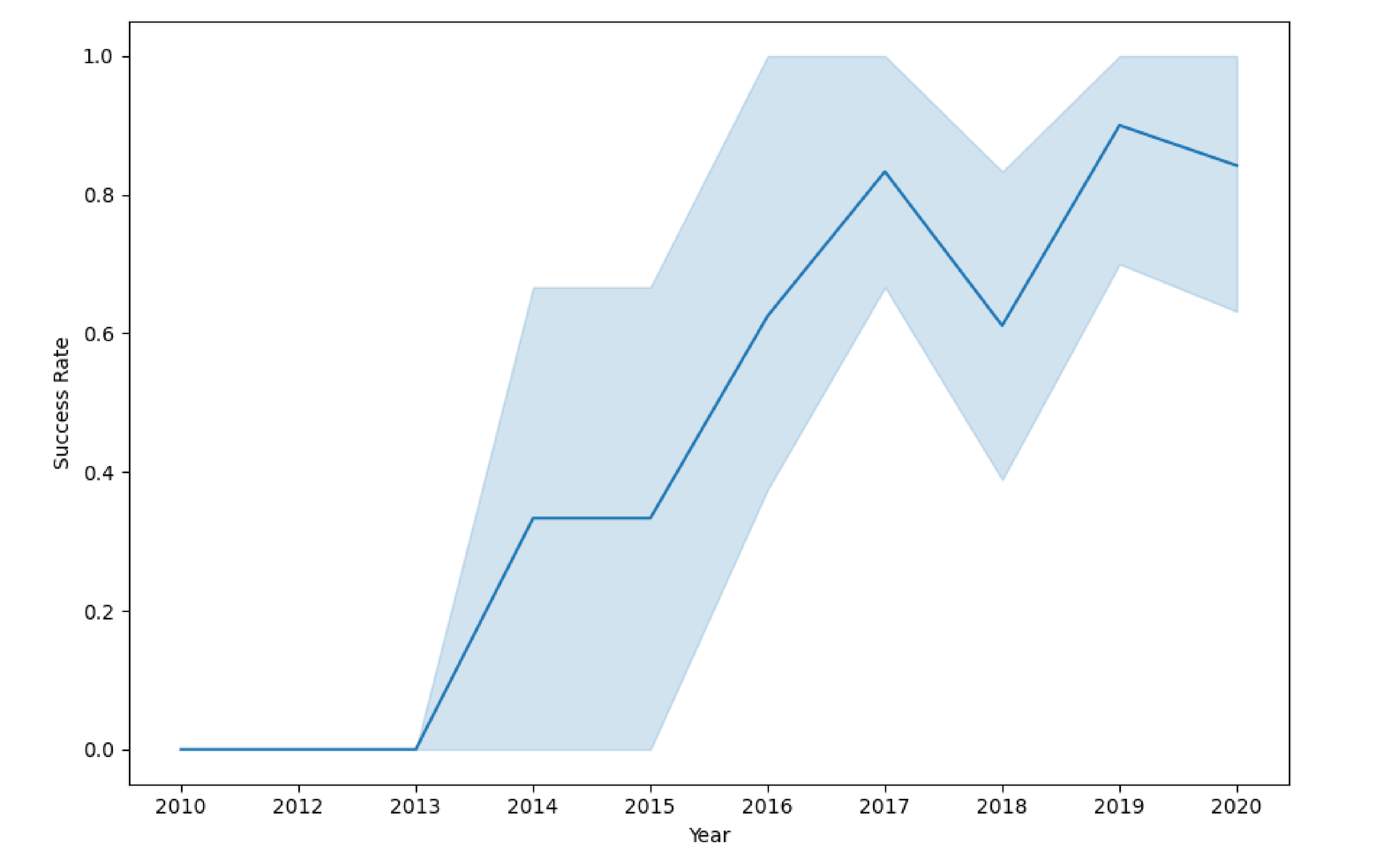
Flight Number vs. Orbit Type Scatter Point



Payload vs. Orbit Type Scatter Point



Launch Success Yearly Trend Line Chart



All Launch Site Names SQL query

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

In [8]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[8]:

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

Launch Site Names Begin with 'CCA' SQL query

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

```
%sql SELECT * FROM SPACEXTBL 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
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass SQL Query

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1 SQL query

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_), Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

AVG(PAYLOAD_MASS__KG_)	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

First Successful Ground Landing Date SQL query

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

Hint:Use min function

```
%sql SELECT MIN(date) FROM SPACEXTBL 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 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

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME='Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4001 
```

```
 * 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 SQL query

List the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS OUTCOME FROM SPACEXTBL GROUP BY MISSION_OUTCOME
```

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

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

Boosters Carried Maximum Payload SQL query

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

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

```
 * 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 SQL Query

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.

```
%sql SELECT substr(Date,6,2), Booster_Version, Launch_Site, Payload, PAYLOAD_MASS__KG_, Mission_Outcome,Landing_Outcome FROM
```

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

substr(Date,6,2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing_Outcome
10	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

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

Task 10

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.

```
%sql SELECT LANDING_OUTCOME, COUNT(*) AS COUNT_LAUNCHES FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROU
```

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

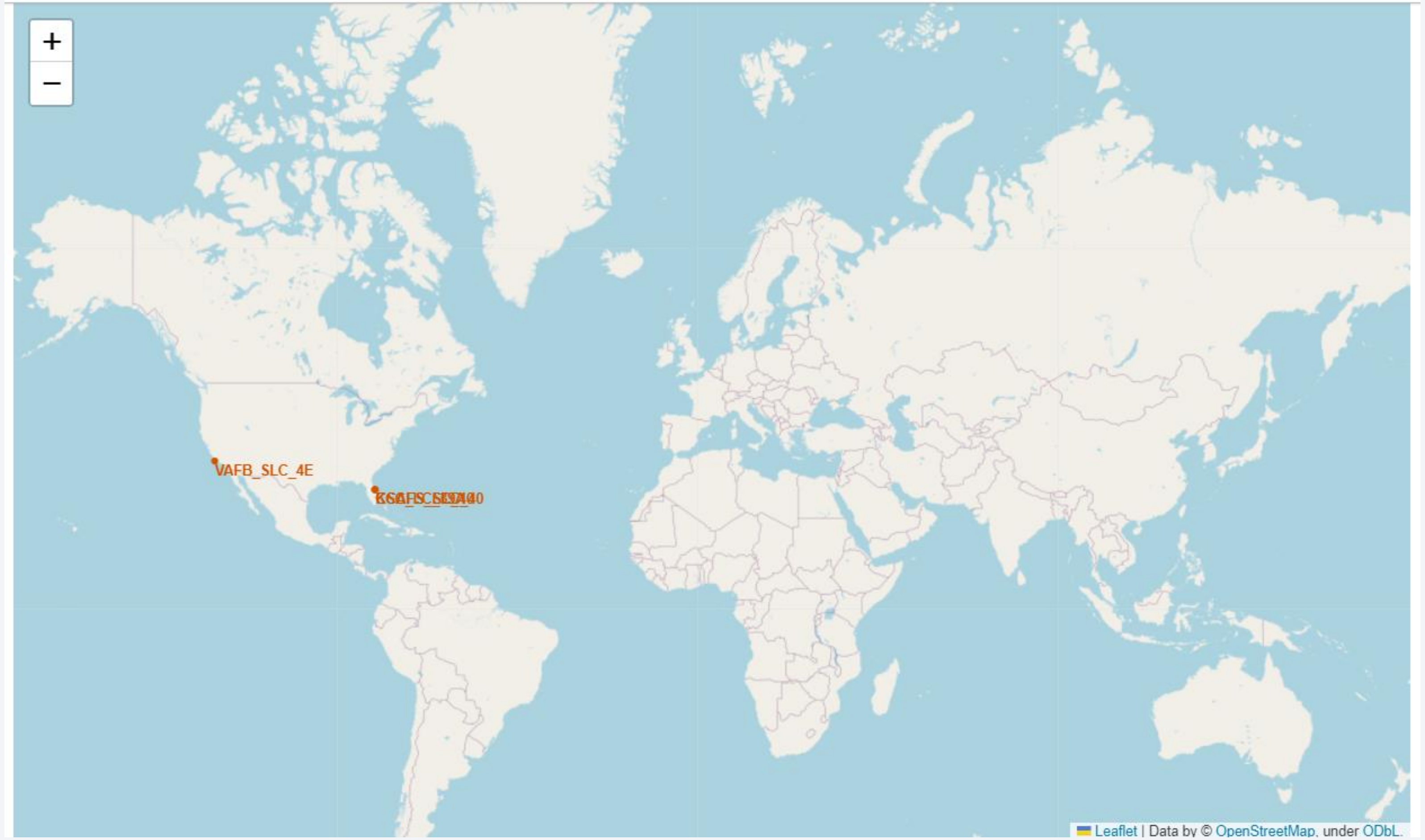
Landing_Outcome	COUNT_LAUNCHES
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

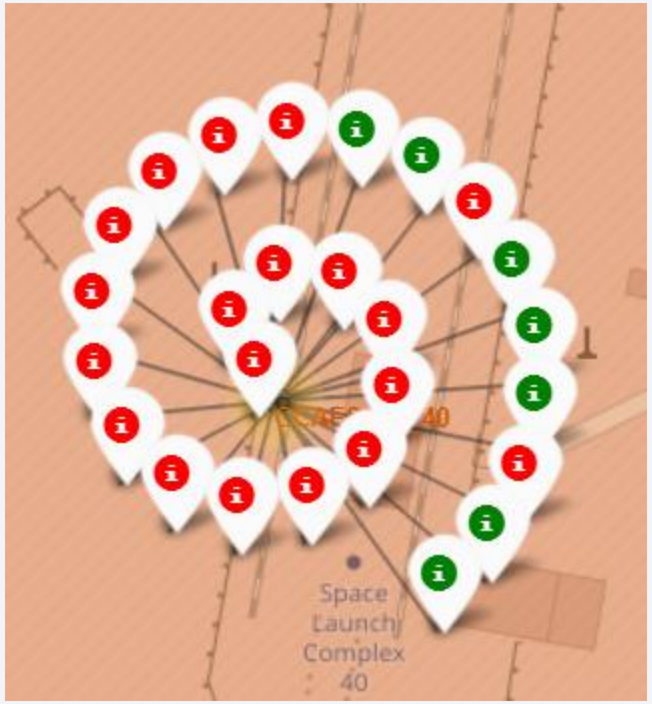
All launch sites’ location on global map

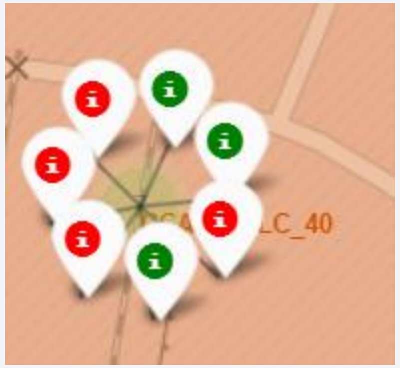
All launch sites are based in the US.

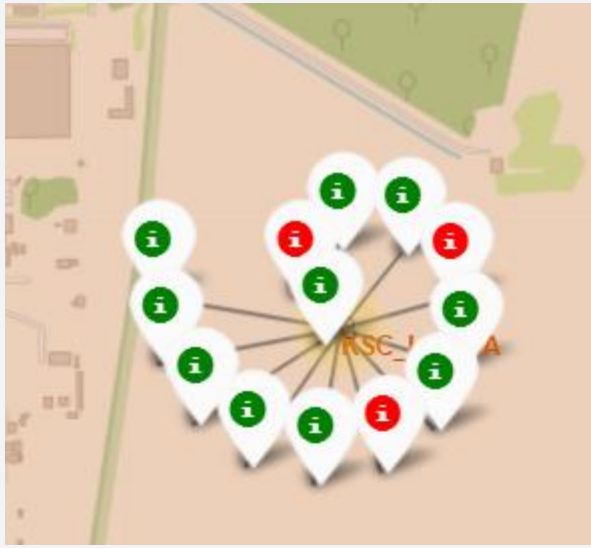


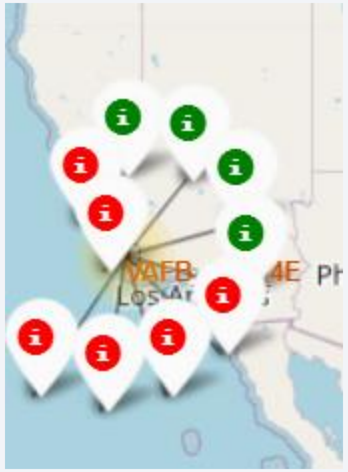
Launch Outcomes of Each Site

- Successful outcomes are in green marker, unsuccessful are in red marker.
- Different locations have different success rate.







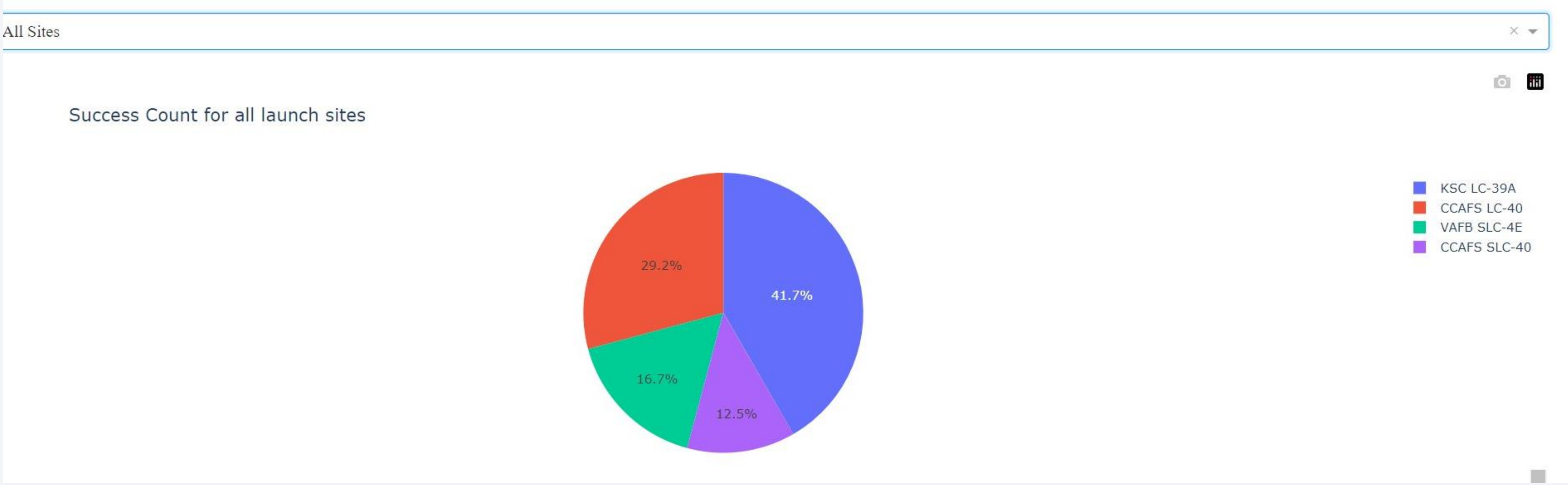


Coastline proximity to a launch site

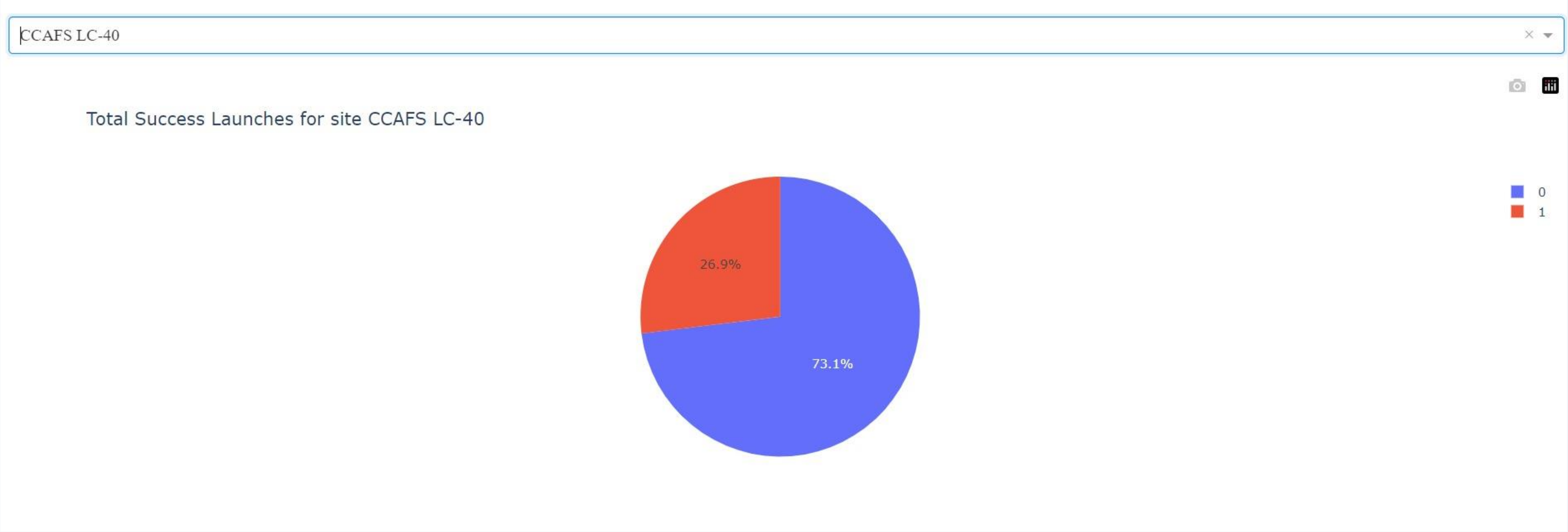
Section 4

Build a Dashboard with Plotly Dash

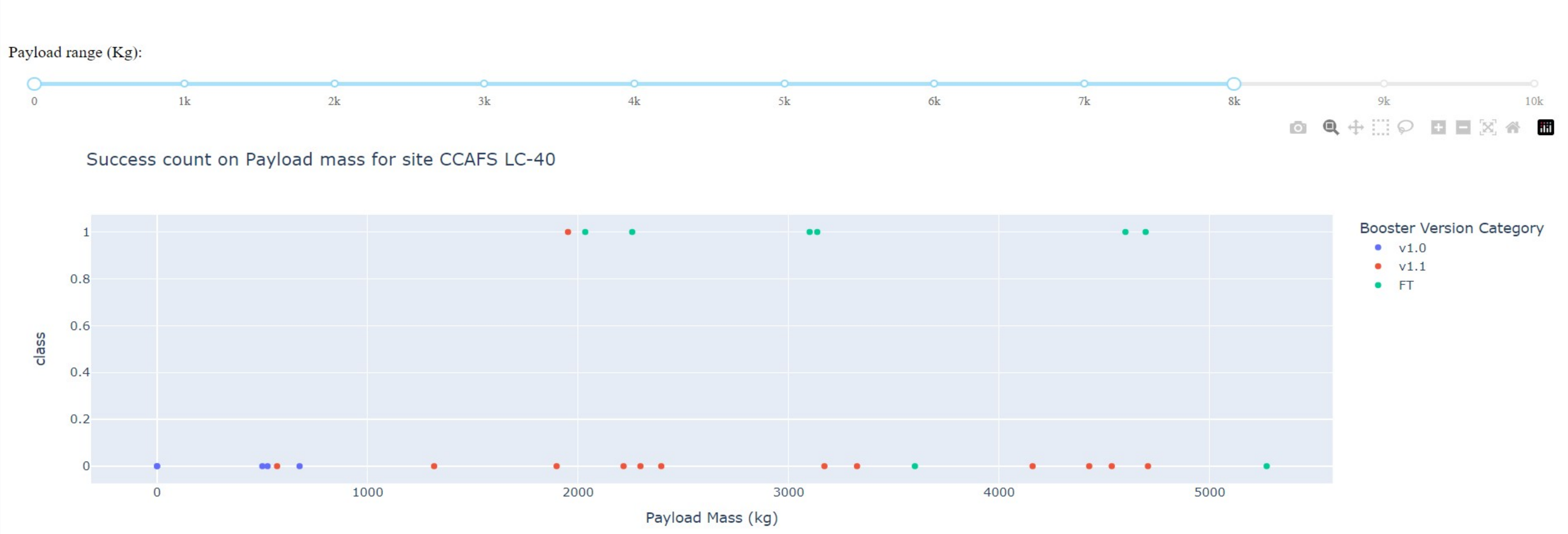
Success Count of All Sites Pie Chart in Dash



Highest Launch Success Ratio Pie Chart in Dash



Payload vs. Launch Outcome Scatter Plot in Dash

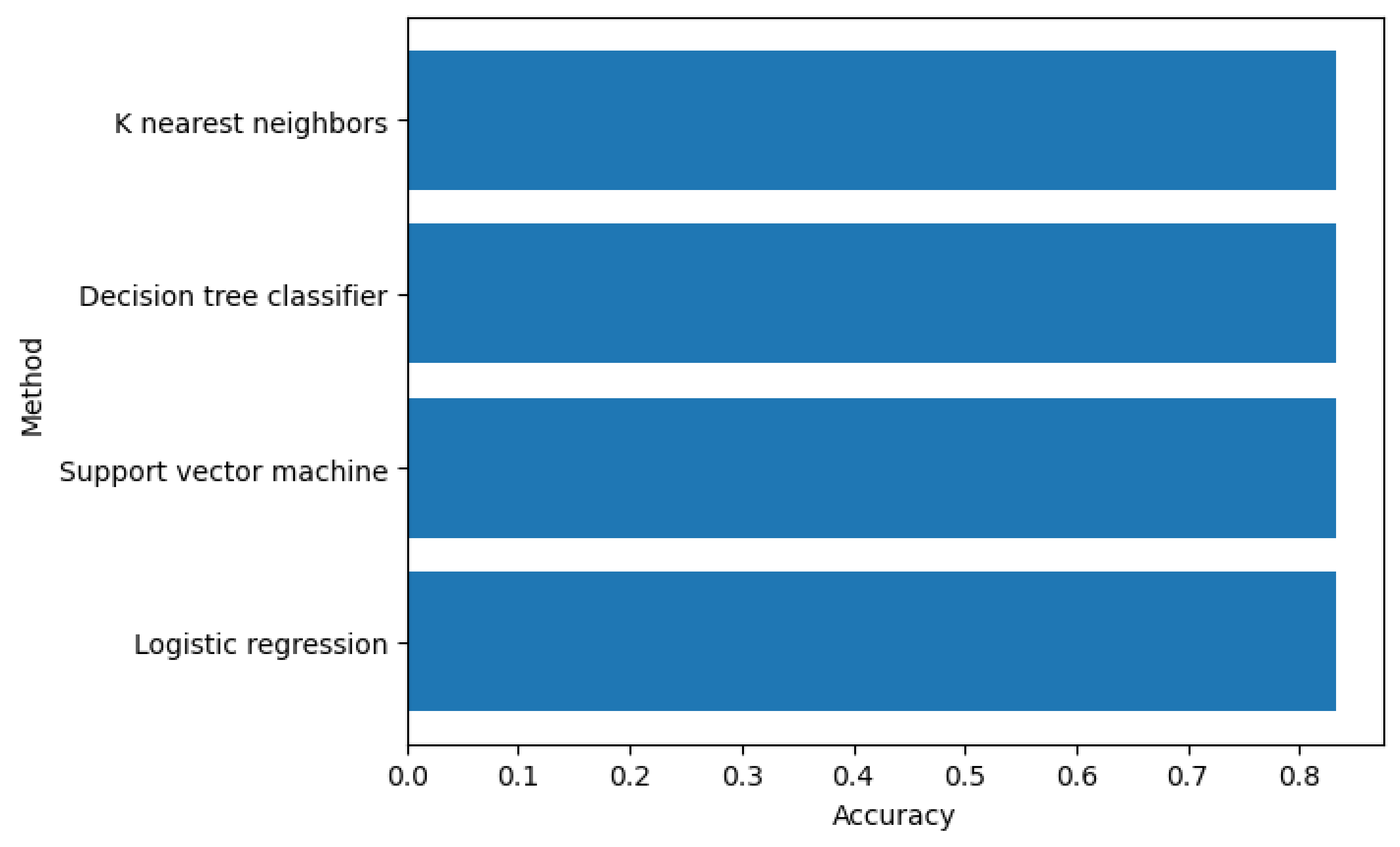


Section 5

Predictive Analysis (Classification)

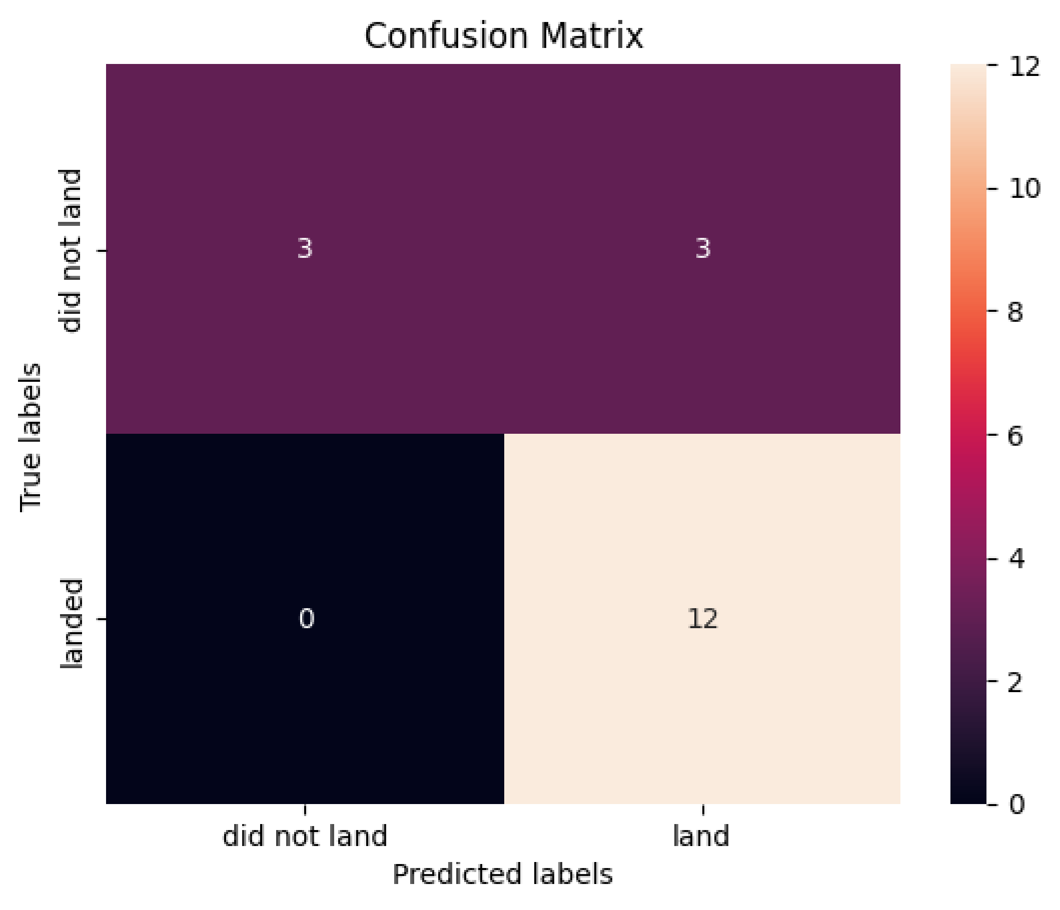
Classification Accuracy

- All models showed same accuracy of 83.33%.



Confusion Matrix

- All models have the same confusion matrix due to the same classification accuracy.



Conclusions

- The success rate is different for each of the launch site. One launch site has **60%** success rate while the other two has **77%**.
- The success rate increases with the number of launches.
- The highest success rate is observed in orbits ES-L1, GEO, HEO and SSO of **100%**.
- Lighter payloads showed more success than heavy payloads.
- All predictive models showed the same classification accuracy of **83%**.

Appendix

- Link to GitHub Repository:

https://github.com/GanchevM00/Capstone-Project/tree/main

Thank you!