IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection through API
  - Data collection through web scrapping
  - Data wrangling
  - Exploratory Data Analysis (EDA) with SQL, Pandas, and Matplotlib
  - Interactive visual analytics and dashboarding
  - Predictive analysis using Machine Learning (ML)
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics results
  - Predictive analysis results

Introduction

SpaceX is a private aerospace manufacturer that established in 2002 by entrepreneur Elon Musk. SpaceX has a mission to revolutionize space technology and make space exploration more affordable and cost-efficient. One of their greatest project is Falcon 9, which cost 62 million dollars while other providers can up to 165 million dollars. This can be achieved because SpaceX utilized the reuse of the first stage. By doing this, the can compress their budget.

As a data scientist, the goal of this project is to make a prediction about the landing outcome of the first stage in the future from the existing data using Machine Learning (ML).

Problem statements:

1. Identifying factors that can affect the landing outcome
2. Relationship between each variable
3. The best conditions to increase the probability of successful landing

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data is collected through API and web scrapping
- Perform data wrangling
  - Describe is processed by performing one-hot-encoding for 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 of SpaceX Falcon 9 project is collected through the API, then the additional data is collected by web scrapping from Wikipedia
- The data collection using API is started using the get_request function then the data is turned into pandas data frame.
- For the web scrapping, the BeautifulSoup function is used to extract the information from HTML table.

Data Collection – SpaceX API

Perform data collection by using get_request

Perform function json_normalize to convert json result into dataframe

Perform data preparation such as cleaning missing values

```
In [9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-
```

```
In [10]: response.status_code
```

```
In [11]: # Use json_normalize meethod to convert the json result into a dataframe
         data = response.json()
         data = pd.json_normalize(data)
```

```
# 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 multip
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Data Collection - Scraping

Request the Falcon 9 launch wiki from URL

Extract all column and variables from HTML table header

Create data frame by parsing the launch HTML tables

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
BeautifulSoup = BeautifulSoup(response, 'html.parser')
```

```
# Use soup.title attribute
BeautifulSoup.title
```

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
temp = BeautifulSoup.find_all('th')
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
for x in range(len(temp)):
    try:
        name = extract_column_from_header(temp[x])
        if (name is not None and len(name)>0):
            column_names.append(name)
    except:
        pass
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

Data Wrangling

- Data wrangling process is conducted to perform cleaning the data and make sure there are no messy data.
- In this process, I performed:
    1. Replacing the missing values with the mean value
    2. Calculating the number of launches site
    3. Calculating the number of orbit
    4. Calculating the number and occurrence of mission outcomes
    5. Create a variable that represent the landing outcome in binary terms

EDA with Data Visualization

- EDA with data visualization process is conducted to gain better understanding how each variable correlate with each other.
- In this process, I performed:
    1. Created a scatterplot of flight number vs payload mass
    2. Visualize the relationship between flight number and launch site
    3. Visualize the relationship between payload and launch site
    4. Visualize the relationship of success rate of each orbit type
    5. Visualize the relationship between flight number and orbit type
    6. Visualize the relationship between payload and orbit type
    7. Visualize the yearly trend of successful launch
    8. Performed one-hot-encoding to categorical value

EDA with SQL

- EDA with SQL is conducted to gain better understanding of specific information of the data.
- In this process, I performed:
    1. Display unique launch site
    2. Launch site begin with CCA-
    3. Total payload mass carried by NASA
    4. Average payload mass carried by booster version F9 v1.1
    5. The date when the successful landing outcome on ground pad was achieved
    6. Names of the boosters which have success in drone ship with payload mass more than 4000 and less than 6000
    7. Total number of successful and failure mission
    8. Name of booster which have carried maximum payload mass
    9. Records in year 2015
    10. Rank the count of landing outcome between the date 2010-06-04 and 2017-03-20

Build an Interactive Map with Folium

- In this task, I built an interactive map with Folium.
- In this process, I performed:
    1. Mark the coordinate of each launch site and add a circle mark
    2. Assign the launch outcome to classes 0 and 1 with red and green markers
    3. Measure the distance of the launch sites to various landmark

Build a Dashboard with Plotly Dash

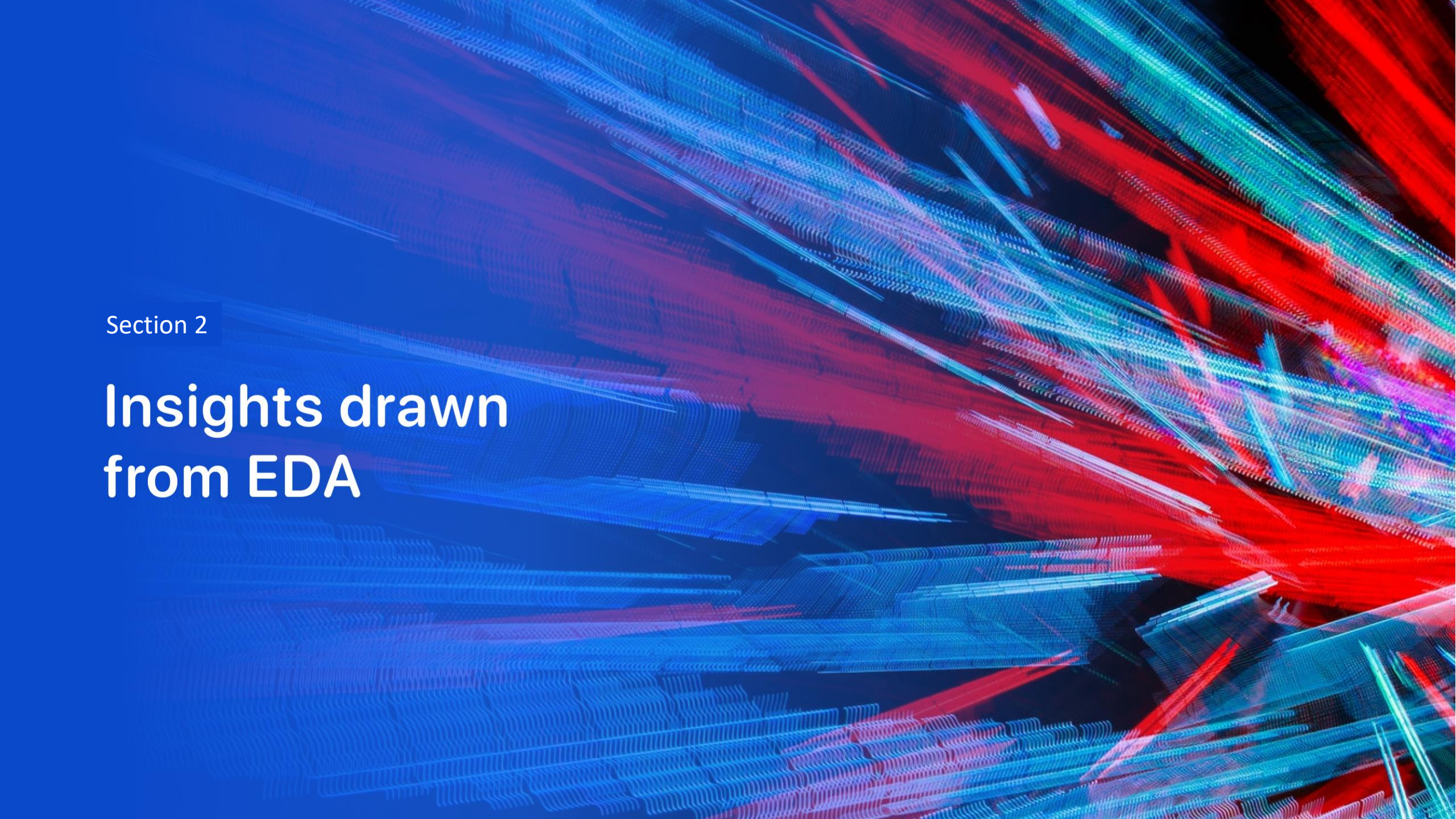
- In this task, I built a dashboard using plotly dash in order to create more interactive visualization.
- In this process, I performed:
    1. Created pie chart that shows total launches per each site
    2. Created scatter plot to show the relationship between outcome and payload mass for different booster version

Predictive Analysis (Classification)

- In this task, I built a machine learning model to predict the outcome of launch.
- The step of this process:
    1. Building the model
    2. Evaluate the model
    3. Improving the model
    4. Find the best model

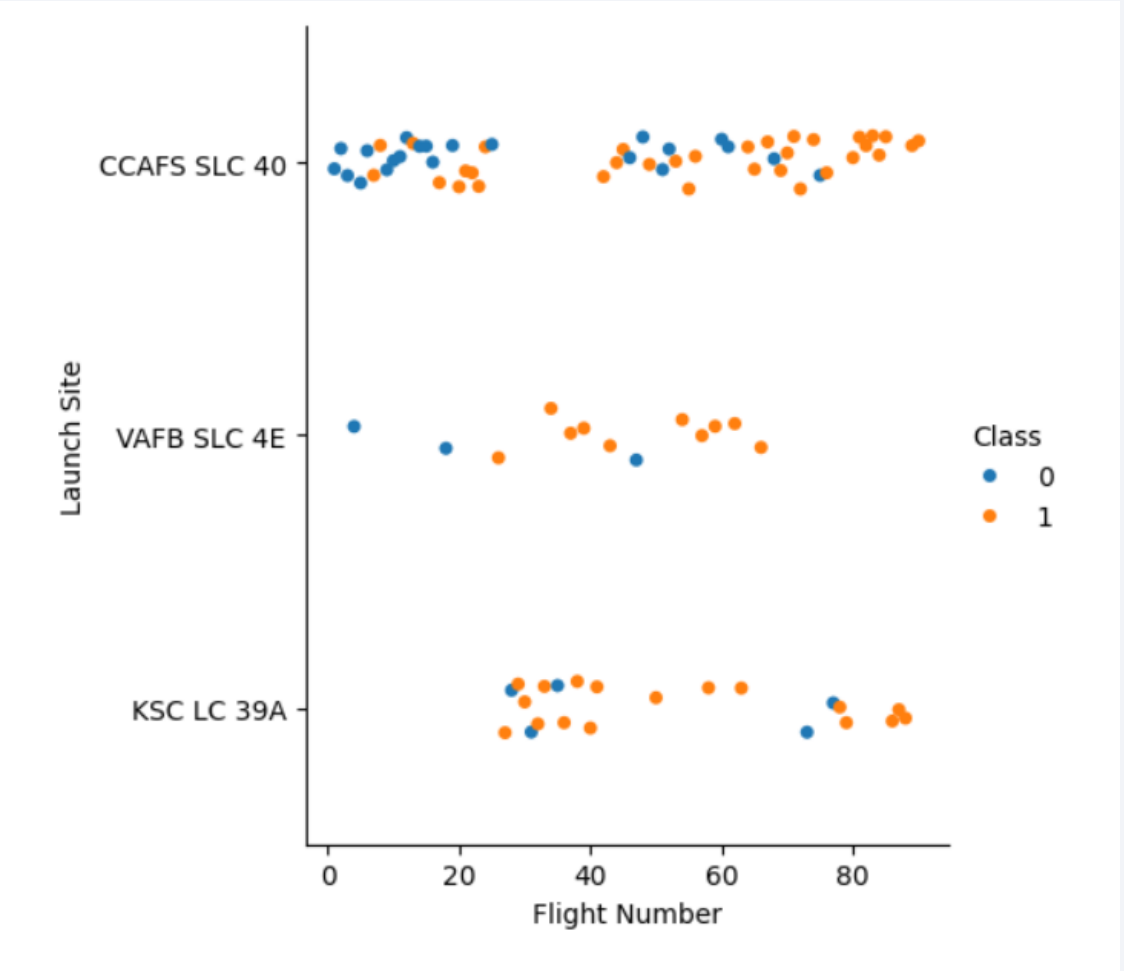
Results

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


Section 2
Insights drawn from EDA

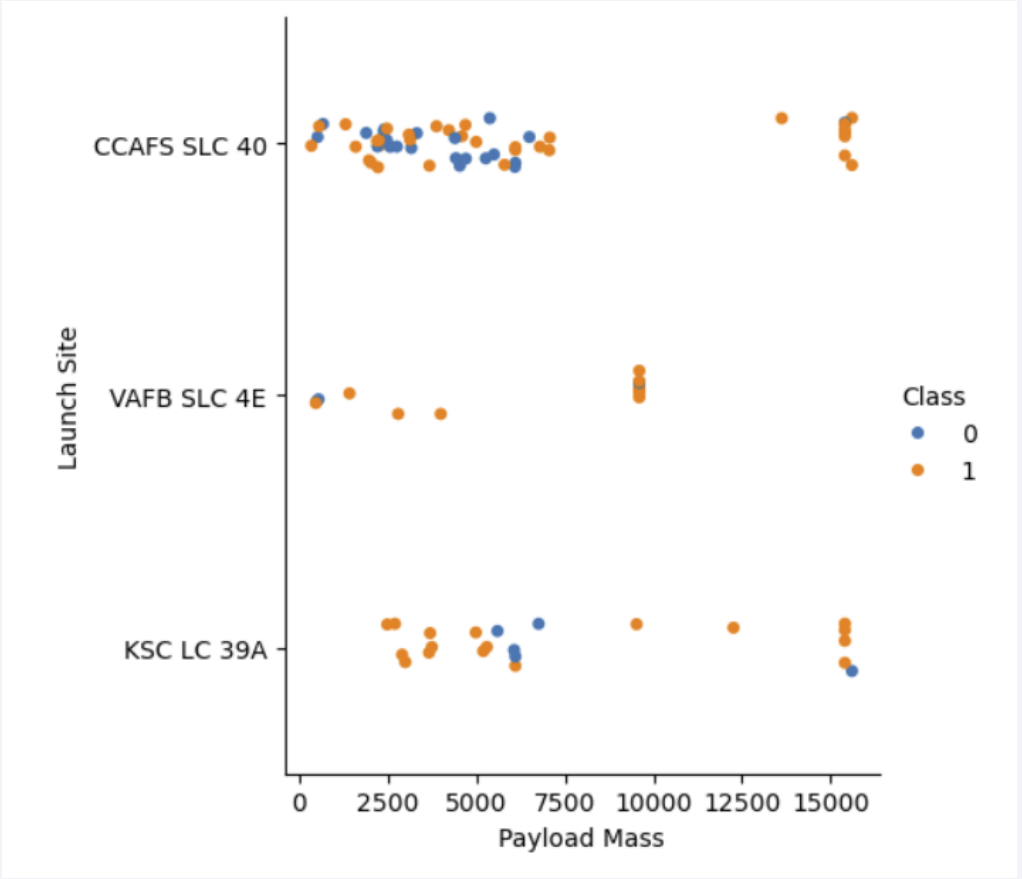
Flight Number vs. Launch Site

- The figure shows the relationship between launch site and flight number.
- From the graph, it can be seen that as the flight number increases, the successful launch outcome can be more obtained.



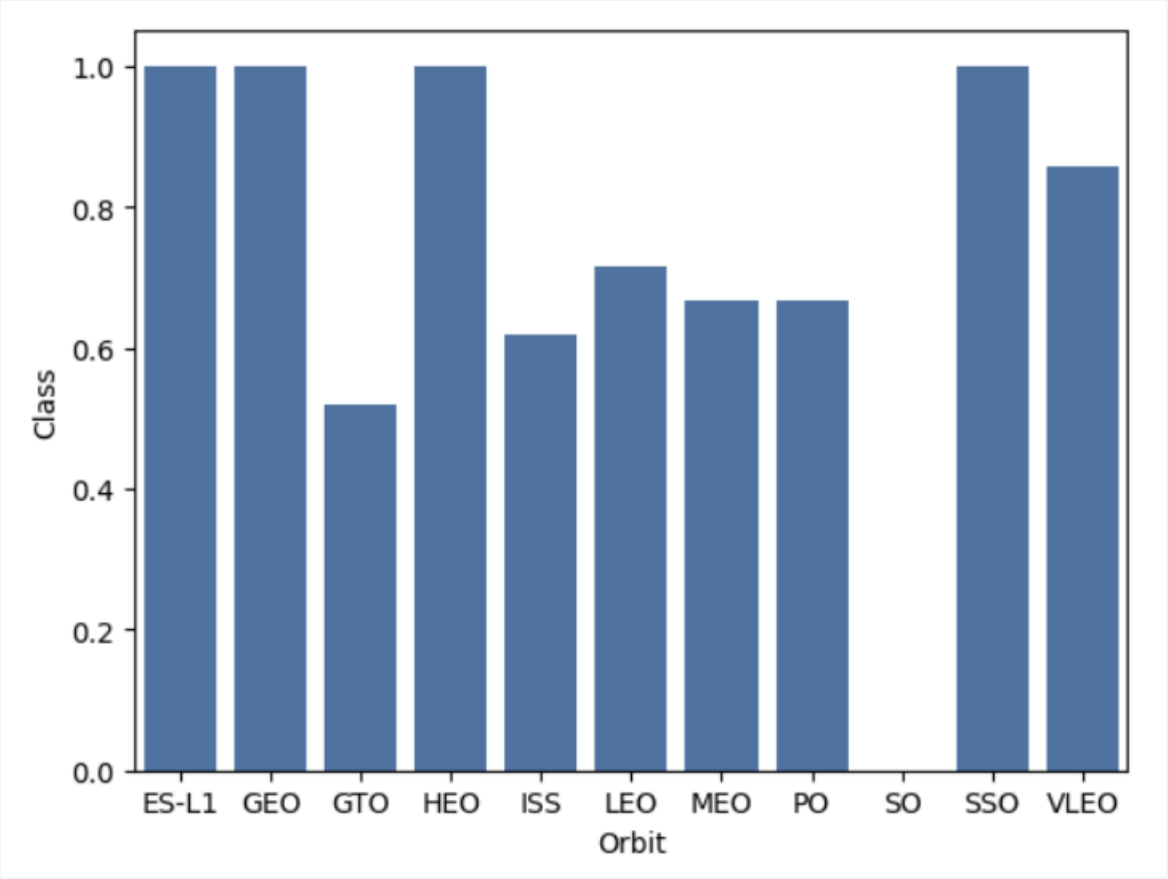
Payload vs. Launch Site

- The figure shows the relationship between launch site and payload mass.
- From the graph, it can be seen that the launch tends to be more successful when the payload is more than 7000 kg.



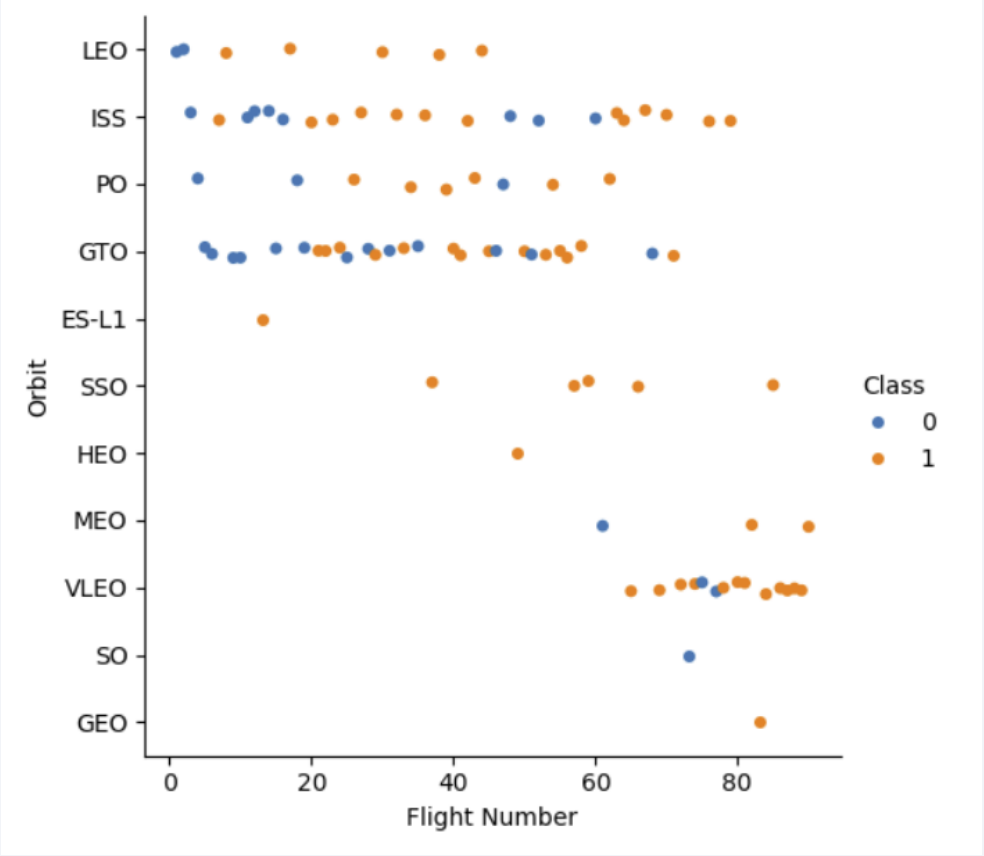
Success Rate vs. Orbit Type

- The figure shows the relationship between the success rate and the orbit type.
- From the figure, it can be seen that there are 4 orbit that have 100% success rate and 1 orbit that has zero percentage success rate.



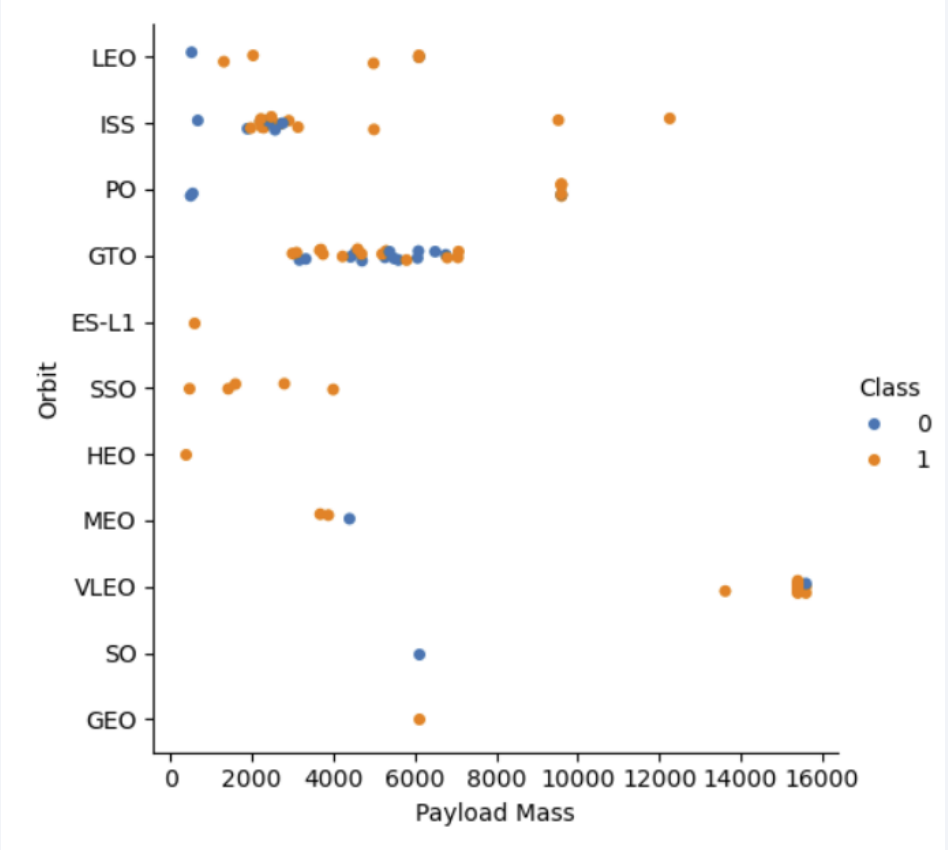
Flight Number vs. Orbit Type

- The figure shows the relationship between the flight number vs orbit type.
- From the figure, it can be seen that as the flight number increases, the success rate of the launch tends to increase.
- However, for the GTP orbit has no clear relationship as the launch outcome are fluctuated.



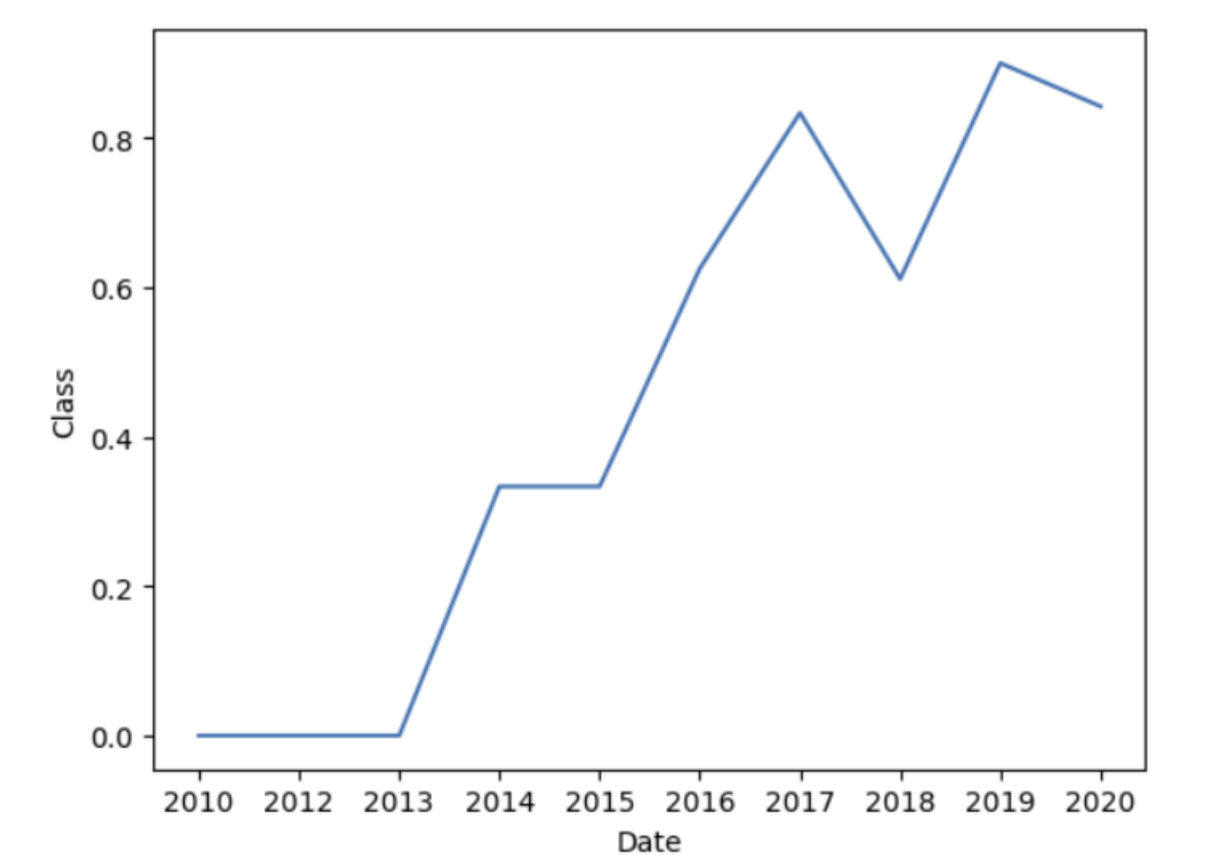
Payload vs. Orbit Type

- The figure shows the relationship between the orbit and payload mass.
- From the figure, it can be seen the percentage of the success rate is increased when the payload is below 7000 kg.
- However, the VLEO and PO orbit has successful tendency at higher payload mass.



Launch Success Yearly Trend

- The figure shows the launch success yearly trend.
- From the figure, it can be concluded the successful rate of the launch has tendency to increase until 2020.



All Launch Site Names

- Figure below shows all launch site names.
- I used DISTINCT function to generate unique names of the launch sites.

```
[19]: %sql SELECT DISTINCT LAUNCH_SITE as "Launch Site" FROM SPACEXTBL
 * sqlite:///my_data1.db
Done.
```

[19]:

Launch Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

- Figure below shows sites name begin with ‘CCA’.
- I used WHERE function to generate launch site names that begin with ‘CCA’ and used LIMIT function to show only 5 names.

```
[47]: %sql SELECT LAUNCH_SITE FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
 * sqlite:///my_data1.db
Done.
```

[47]:

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

Total Payload Mass

- Figure below shows total payload mass by boosters NASA.
- From the code, the total payload mass carried by NASA is 45596 kg.

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

```
[14]: %sql SELECT SUM (PAYLOAD_MASS__kg_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'
```

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

[14]:

SUM (PAYLOAD_MASS__kg_)
45596

Average Payload Mass by F9 v1.1

- Figure below the code to calculate the average payload mass by F9 v1.1
- I used AVERAGE function to calculate the average value of the payload mass with WHERE function to filter.
- The average payload mass carried by the F9 v1.1 is 2928.4 kg

```
[17]: %sql SELECT AVG(PAYLOAD_MASS__kg_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
 * sqlite:///my_data1.db
Done.
[17]: AVG(PAYLOAD_MASS__kg_)
      2928.4
```

First Successful Ground Landing Date

- Figure below the code to show the date the launch was successful for the first time.
- I used MIN function to determine the lowest time and WHERE to filter the landing outcome.

```
[22]: %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)'
 * sqlite:///my_data1.db
Done.
[22]: MIN(DATE)
      2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- Figure below the code to show the successful drone ship landing with payload between 4000 and 6000 kg.
- I used the WHERE function to filter only to Success drone ship and I used conditional formatting to filter the payload mass between 4000 and 6000 kg.

```
[27]: %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.
```

[27]:

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

Total Number of Successful and Failure Mission Outcomes

- Figure below shows the number of successful and failure mission outcomes.
- From the result, the successful missions were 100 missions, while the failure was only one.

List the total number of successful and failure mission outcomes

```
[36]: %sql SELECT COUNT(MISSION_OUTCOME) AS "succesful mission "FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%';
```

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

[36]:

succesful mission
100

```
[37]: %sql SELECT COUNT(MISSION_OUTCOME) AS "failure mission "FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Failure%';
```

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

[37]:

failure mission
1

Boosters Carried Maximum Payload

- Figure below shows the list of boosters that carried maximum payload.

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

```
[38]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__kg_ = (SELECT MAX(PAYLOAD_MASS__kg_) FROM SPACEXTBL)
 * sqlite:///my_data1.db
Done.
```

[38]:

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

- Below figure shows the failure launches record in 2015.
- From the result, the failures were happened at first and fourth month of 2015 with the booster F9 v1.1 B1012 and B1015

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.

[43]:
```
%sql SELECT substr(Date, 6,2) AS MONTH, LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE substr(Date,0,5)='2015' AND LANDING_OUTCOME =
```

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

[43]:

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

- Below figure shows the rank of 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.

[46]:
```
%sql SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING_OUTCOME \
ORDER BY COUNT(LANDING_OUTCOME) DESC ;
```

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

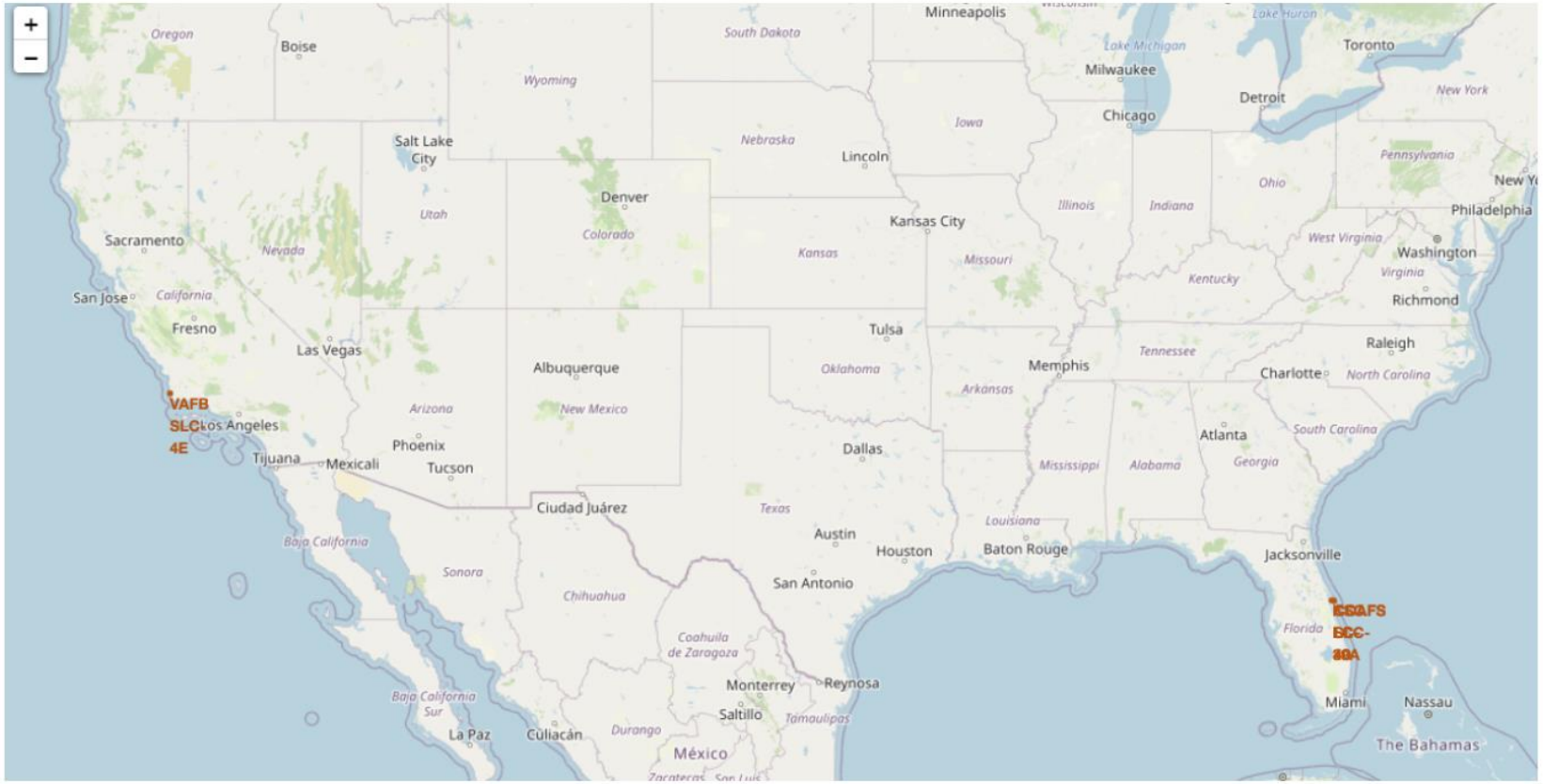
[46]:

Landing_Outcome	COUNT(LANDING_OUTCOME)
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

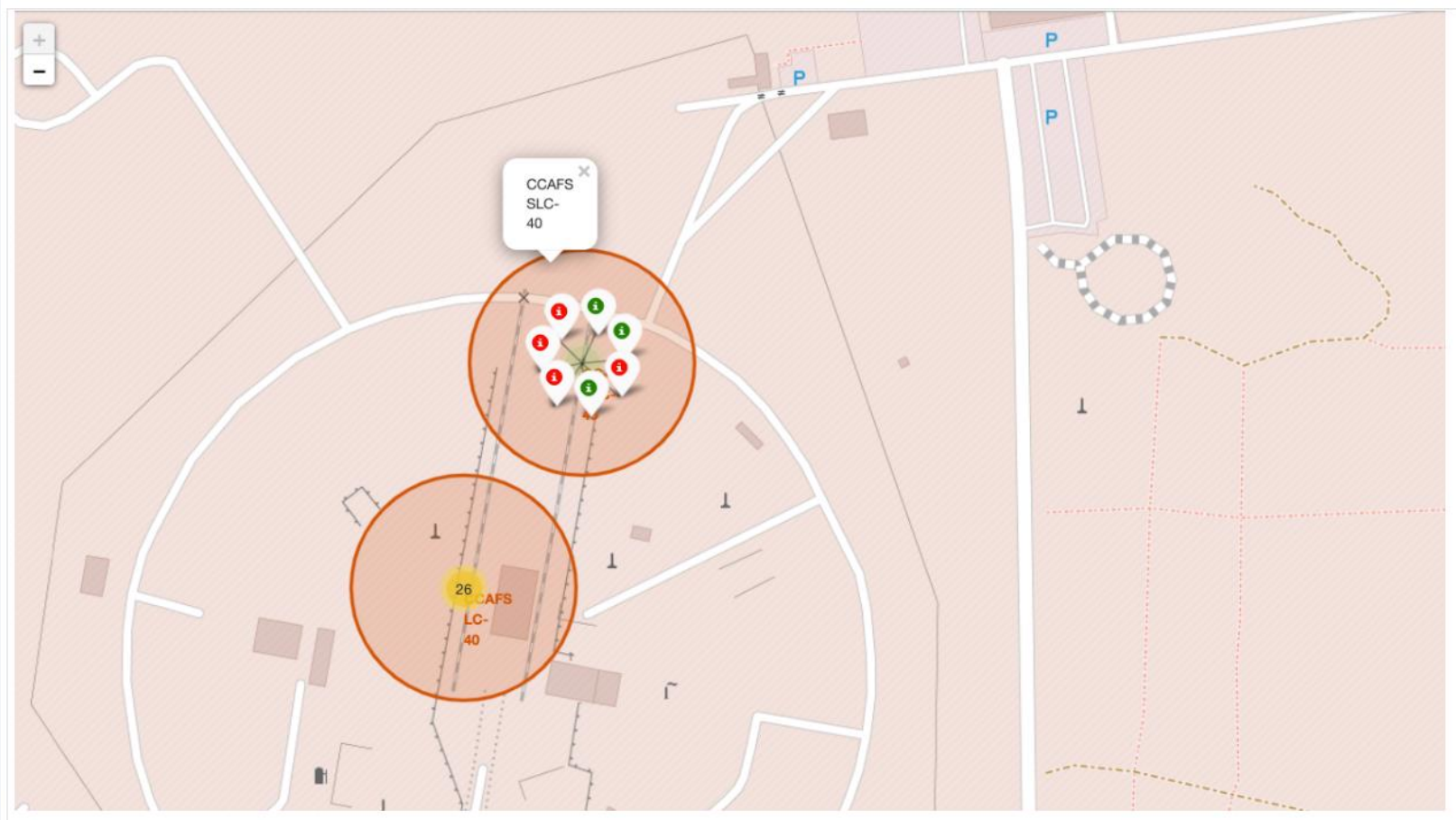
Launch Sites Proximities Analysis

Mark All Launch Sites

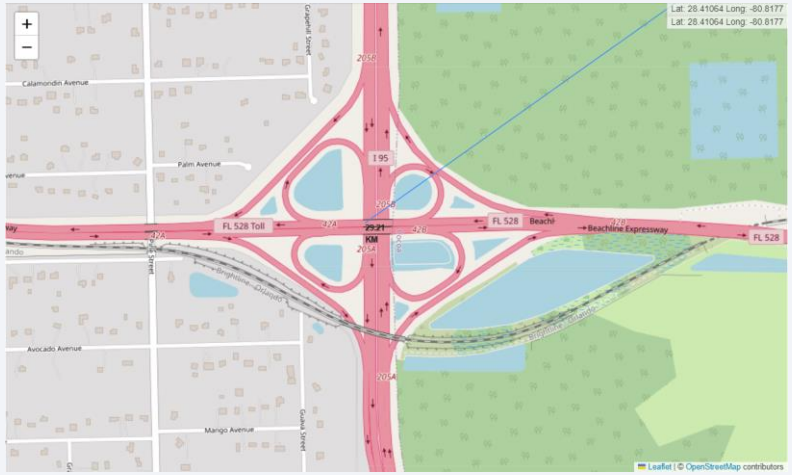


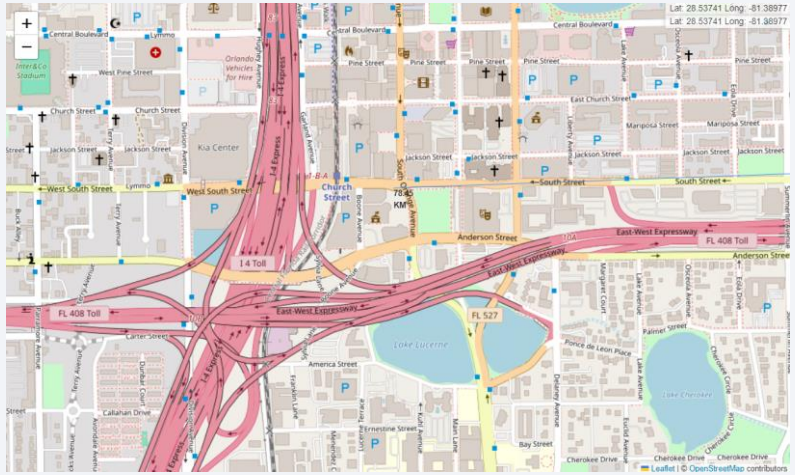
From the results, it can be seen that the launch sites is located on the west and east of USA.

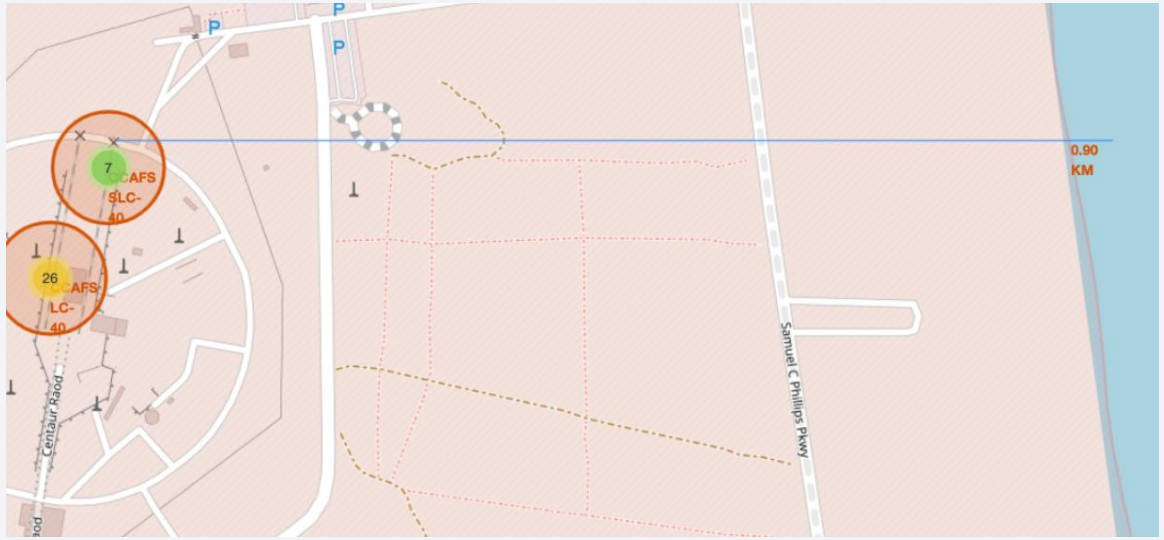
Launch Sites with Color Labels

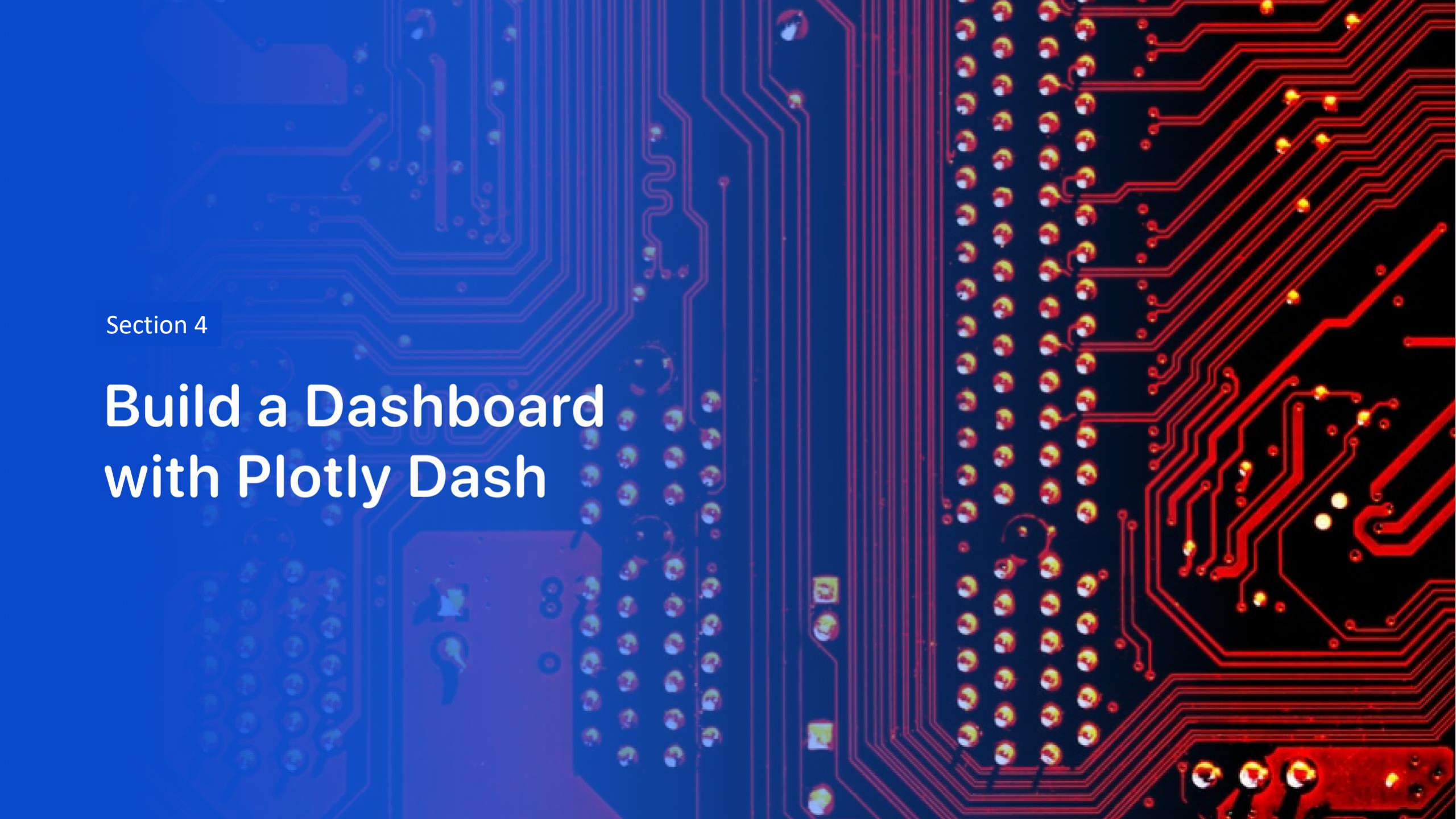


Launch Sites Distances to Landmarks

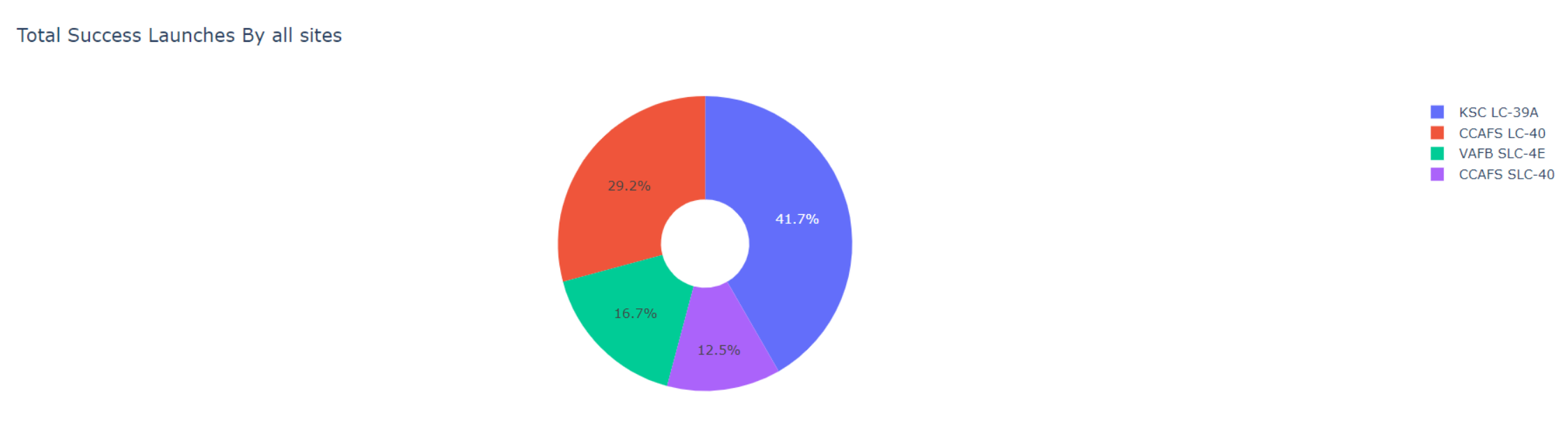






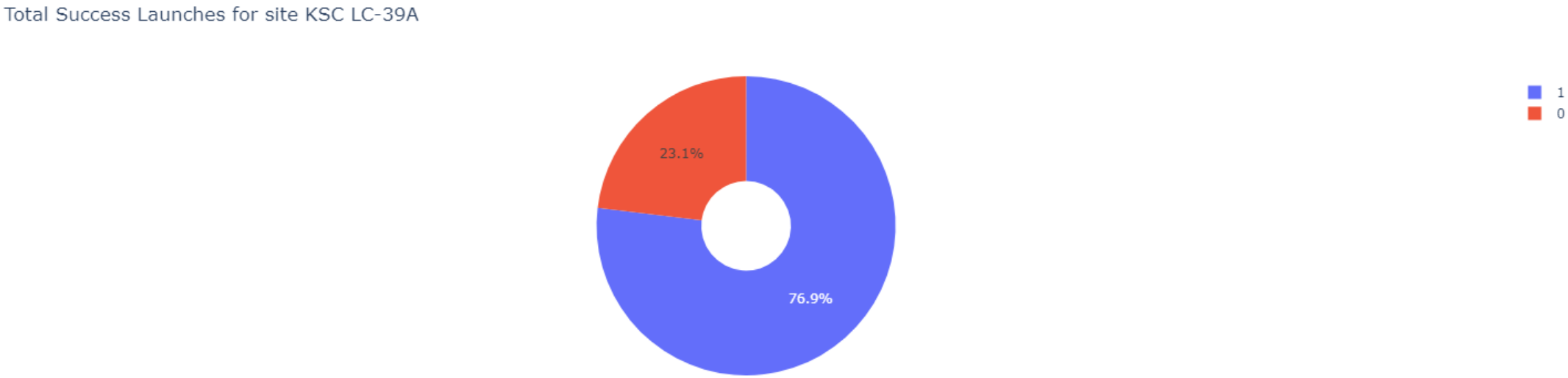

Section 4
Build a Dashboard with Plotly Dash

The Success Rate Percentage of Each Launch Site



From the results, it can be seen the KSC LC-39A has the most successful launch.

Total Success Launches for site KSC LC-39A



From the results, it can be seen the 76,9% of launch records in KSC LC-39A is successful.

Relationship Class vs Payload Mass of Each Launch Site

From the results, it can be seen the most of the launch sites have successful launches when the payload mass is below 5500 kg.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

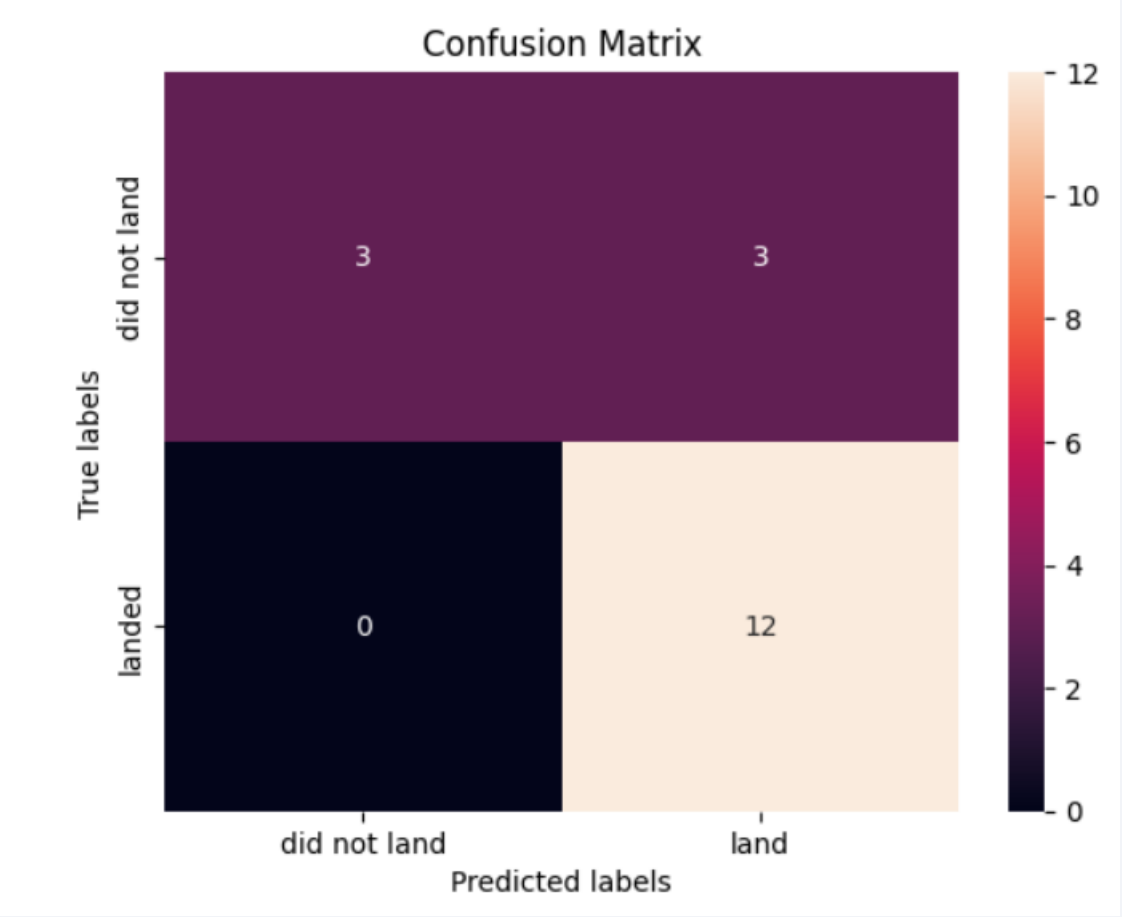
Here is the code to determine which model has highest accuracy.

[33]:
```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.875
Best Params is : {'criterion': 'gini', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'ra
ndom'}
```

Confusion Matrix

- Below is the confusion matrix.



Conclusions

- The best machine learning model is Tree with the score of accuracy is 0,85
- The KSC LC-39A has highest launch successful rate.
- SSO, GEO, ES-L1, and HEO orbit have 100% launch successful rate, while the SO orbit is the orbit that has no successful launch.
- From the launch success yearly trend, it can be concluded the launch success rate is increasing since 2013 until 2020.
- The low payload mass perform better compare to high payload mass. It can be seen that the number of successful launch is bigger in low payload mass.

Thank you!