IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Adam Hamadi>
<11/2/2023>

Outline

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

Executive Summary

- Summary of methodologies

1. Collection of data through API.
2. Collection of data through Web-Scraping.
3. Data Wrangling.
4. Exploratory Analysis Using SQL.
5. Exploratory Analysis Using Pandas and Matplotlib.
6. Interactive Visual Analytics and Dashboard.
7. Predictive Analysis (Classification).

- Summary of all results

1. Exploratory Data Analysis.
2. Interactive Visual Data Analytics.
3. Predictive Analysis.

Introduction

- Project background and context
- Over the years, SpaceX has revolutionized the industry of commercial space exploration by producing more cost-effective and reusable rocket launches. In particular, Falcon 9 which plays a significant role in this advancement. This was possible by making the first stage of the falcon 9 rocket reusable. This led to an increase in cost savings overall. However, this reusability is not always guaranteed, so an understanding of the factors that effect the first stage must be studied to figure out, what is crucial and necessary for this stage to occur successfully.
- Problems you want to find answers
- What are the main factors that determine whether or not a rocket lands?
- Are there any identifiable patterns that effect the first stage recovery ?
- What are the conditions that must be met to assure that the stage 1 recovery happens?

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

- The Process of data collection

1. The initial part of the data collection was done using the “get request’ to the SpaceX API.
2. The, data is decoded using .json() function and then turned into pandas dataframe using the normalizer .json_normalize().
3. Next, data cleaning occurs, where all the missing values are replaced by a zero.
4. Lastly, Web scraping is preformed with use of beautiful soup in order to extract data from Wikipedia Falcon 9 launching records in the form of an HTML.

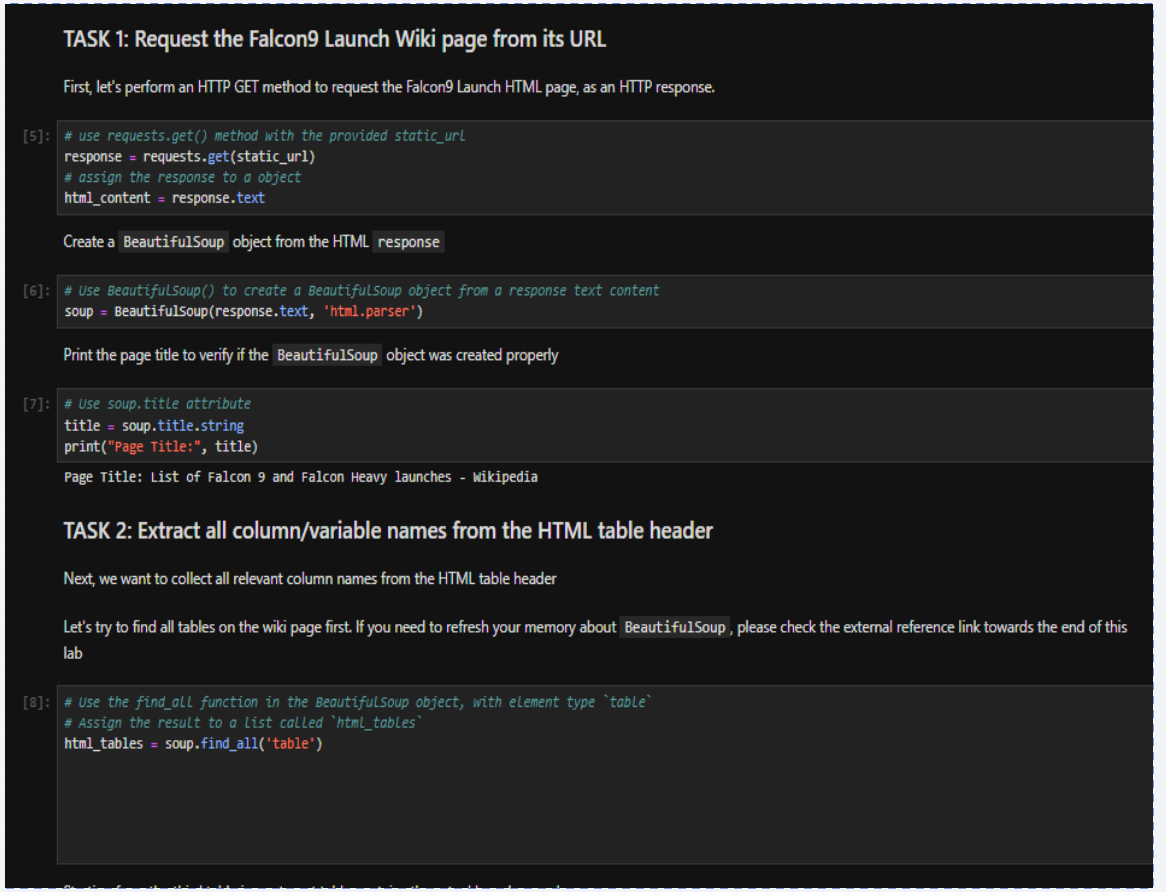
Data Collection – SpaceX API

- As seen from the figure, the request.get is used to get the data and then used .json() to decode and .json_normalize() to turn it into a pandas dataframe.

- The GitHub Link:
- https://github.com/AAhdm/Capstone-/blob/a6ebcd418da27271d6f41d85a5a395bc7d360fc0/Data%20collection.ipynb

Data Collection - Scraping

- In the process, we applied the web scrapping to the falcon 9 launging data using beautiful soup.
- Next, the table was extracted and turned into a pandas dataframe
- GitHub URL: https://github.com/AAhdm/Capstone-/blob/46b4129da2c3f3682525dc30a5b20e73a968d469/Data%20collection.ipynb



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.

[5]:
```
# use requests.get() method with the provided static_url
response = requests.get(static_url)
# assign the response to a object
html_content = response.text
```

Create a `BeautifulSoup` object from the HTML `response`

[6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

[7]:
```
# Use soup.title attribute
title = soup.title.string
print("Page Title:", title)
```

```
Page Title: List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

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

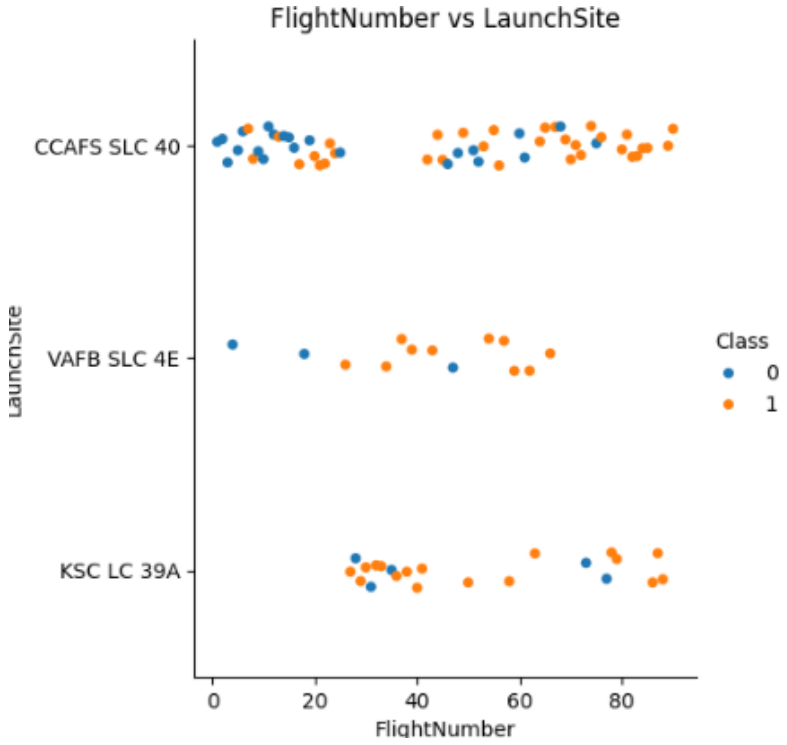
[8]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

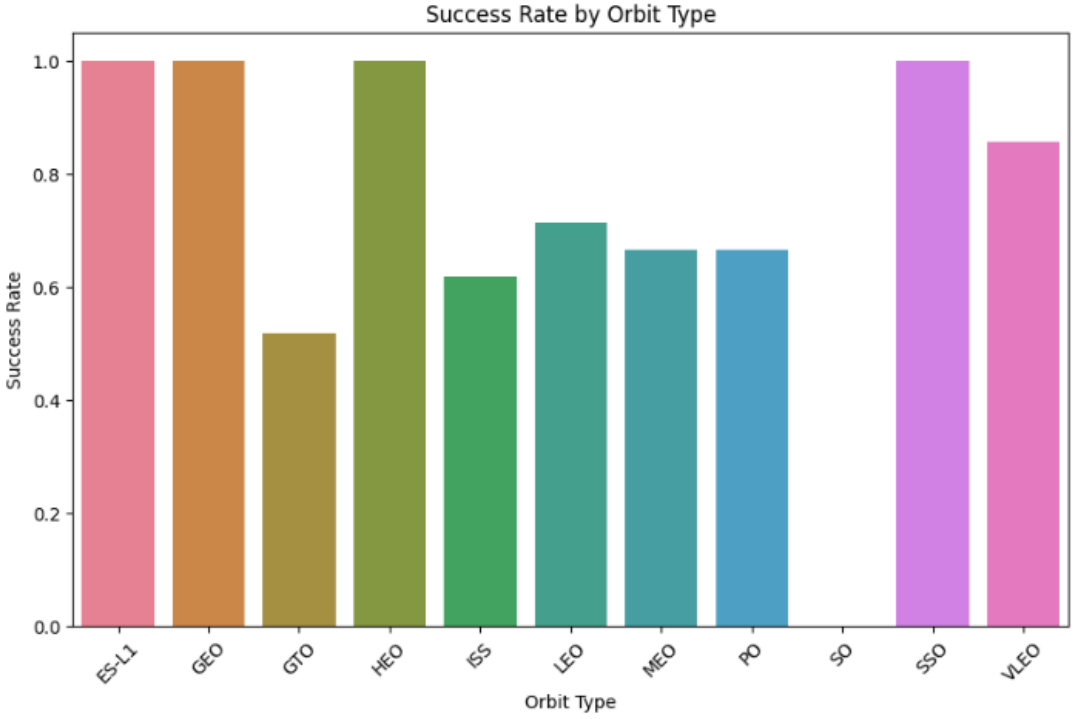
Data Wrangling

- The Process:
  - Initially, after obtaining the data a calculation of the number of launches is done on each site, as the data contains several lunching facilitates.
  - Then, calculate the number and occurrence of each orbit.
  - Next, calculation of numbers and occurrence of mission's outcome of the orbits.
  - Lastly, landing outcome label is created.
- The GitHub Link:
- https://github.com/AAhdm/Capstone-/blob/46b4129da2c3f3682525dc30a5b20e73a968d469/Data%20wrangling.ipynb

EDA with Data Visualization

- The exploration of data is preformed in this part, in order to, perform the Visualization and observe the relationship between several elements. Such Flight Number and Launch Site, Payload and Launch Site, success rate of each orbit type etc.…
- GitHub URL: https://github.com/AAhdm/Capstone-/blob/46b4129da2c3f3682525dc30a5b20e73a968d469/dataviz.ipynb





EDA with SQL

- In this part, begins by loading an SQL extension and establishing a connection with the database.
- The purpose of this is to get insight from the database. This is performed by running SQL quires to find details Such as:

1. The names of the unique launch sites in the space mission.
2. The launch sites begin with the string 'CCA'.
3. The total payload mass carried by boosters launched by NASA (CRS)
4. The date when the first succesful landing outcome in ground pad was acheived.
5. The total number of successful and failure mission outcome
6. The records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

- GitHub URL: https://github.com/AAhdm/Capstone-/blob/46b4129da2c3f3682525dc30a5b20e73a968d469/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- On the map we mark all launch sites using site's latitude and longitude and by the use of objects such as markers, circles, lines that indicate the success or failure of each site on the folium map.
- We mark the success or failure of each site on the map by adding a column "class". If 0 indicates "failure" if 1 indicates "success"
- We calculate the distance between a lunch site to its proximities, which helped distinguish between city, railway, highway through the symbols.
- GitHub URL: https://github.com/AAhdm/Capstone-/blob/46b4129da2c3f3682525dc30a5b20e73a968d469/launch_site_location.ipynb

Build a Dashboard with Plotly Dash

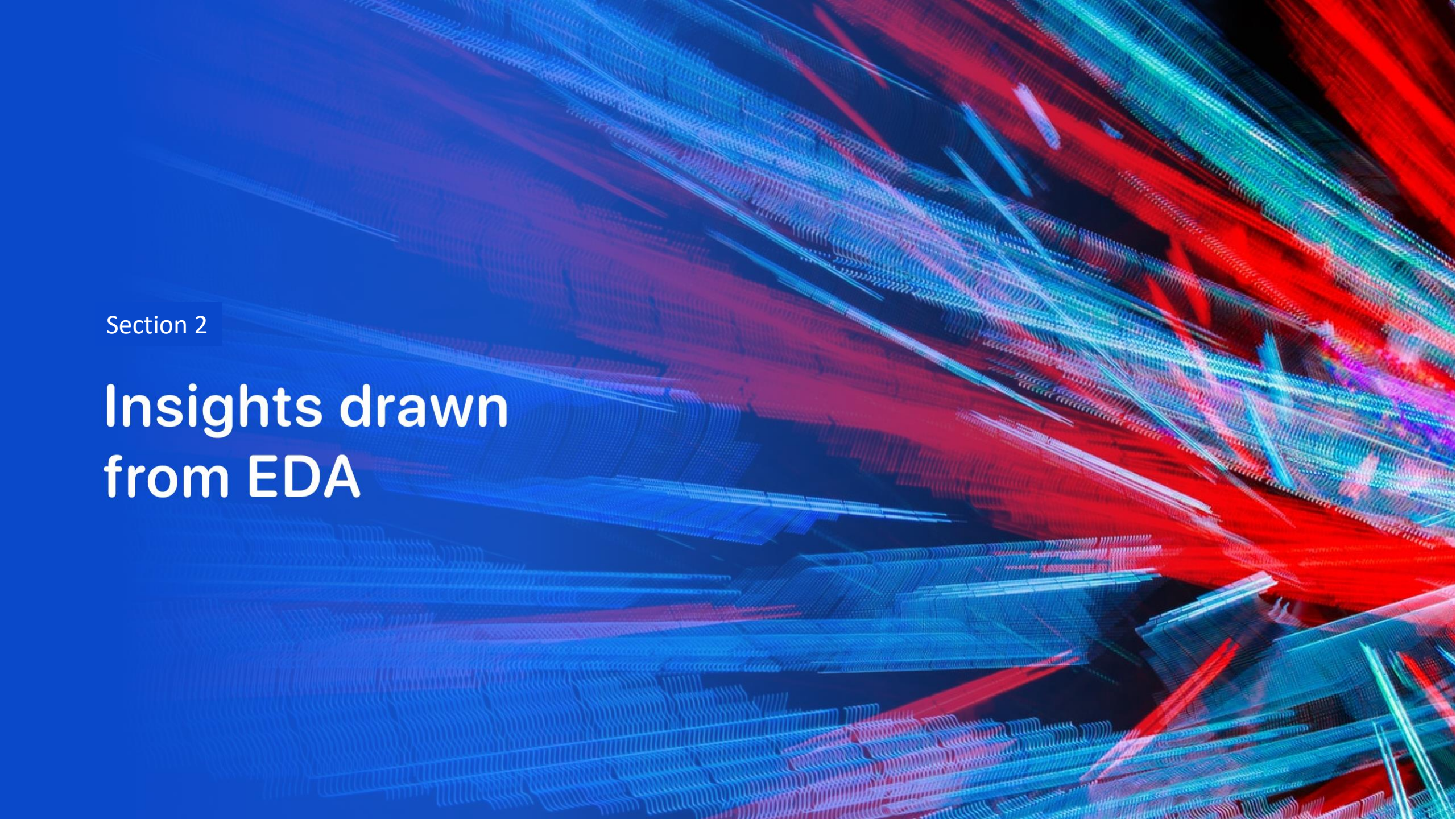
- With the use of Plotly Dash, an interactive dashboard was built. In order to,
- Plot a pie chart that show the total lunches by area.
- A scatter graph that shows the relationship between the outcome and the payload mass (Kgs)
- GitHub URL: https://github.com/AAhdm/Capstone-/blob/1a448bde333449ea09c0a667dbcfe5d7d0e9490b/Dash_app.py

Predictive Analysis (Classification)

- The data is loaded into a numpy array, then standardized and reassigned using the transform.
- Using the function train_test_split to split the data X and Y into training and test data.
- With the use GridSearchCV, different machine learning models are built with the use of different hyperparameters.
- Accuracy is calculated in order to find which of the models performed best.
- GitHub URL: https://github.com/AAhdm/Capstone-/blob/46b4129da2c3f3682525dc30a5b20e73a968d469/SpaceX_Machine_Learning_Prediction.ipynb

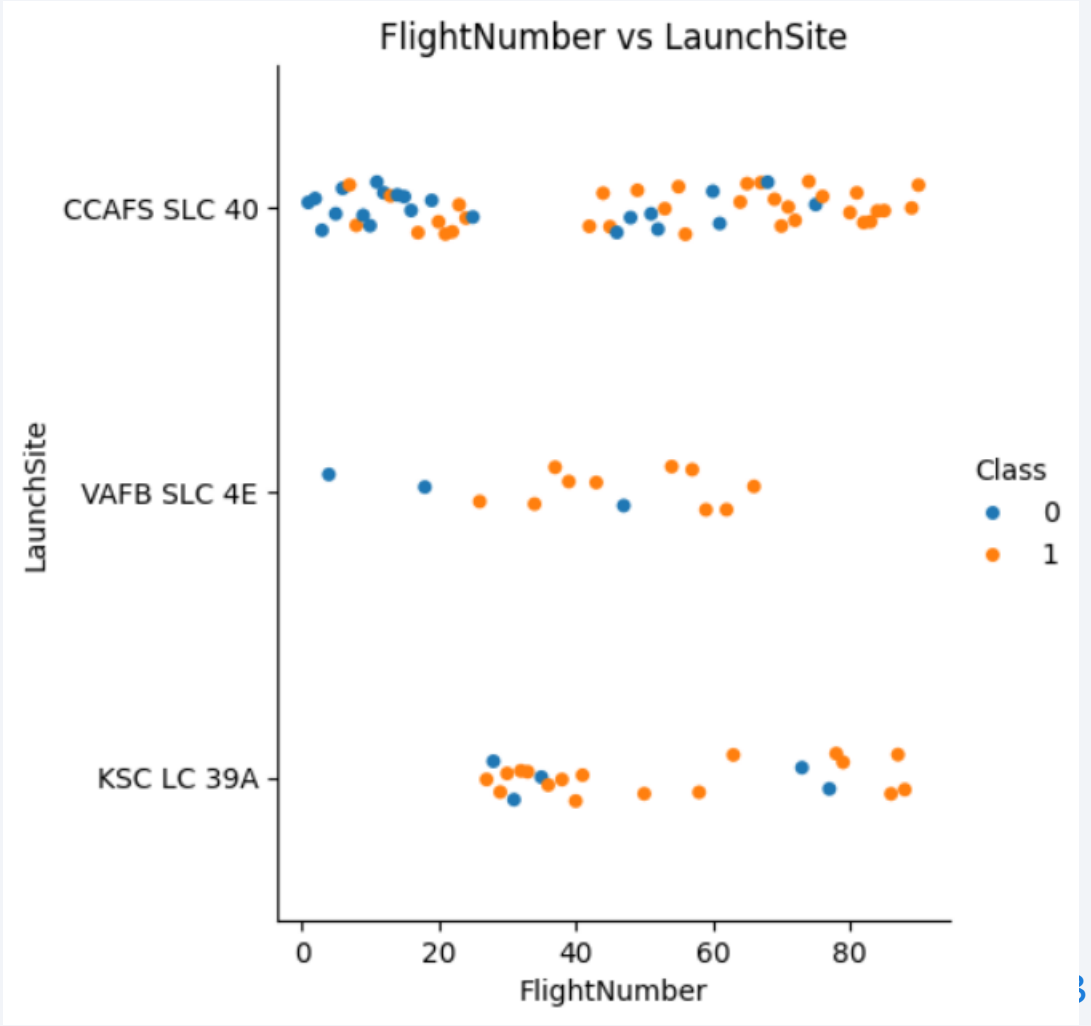
Results

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


Section 2
Insights drawn from EDA

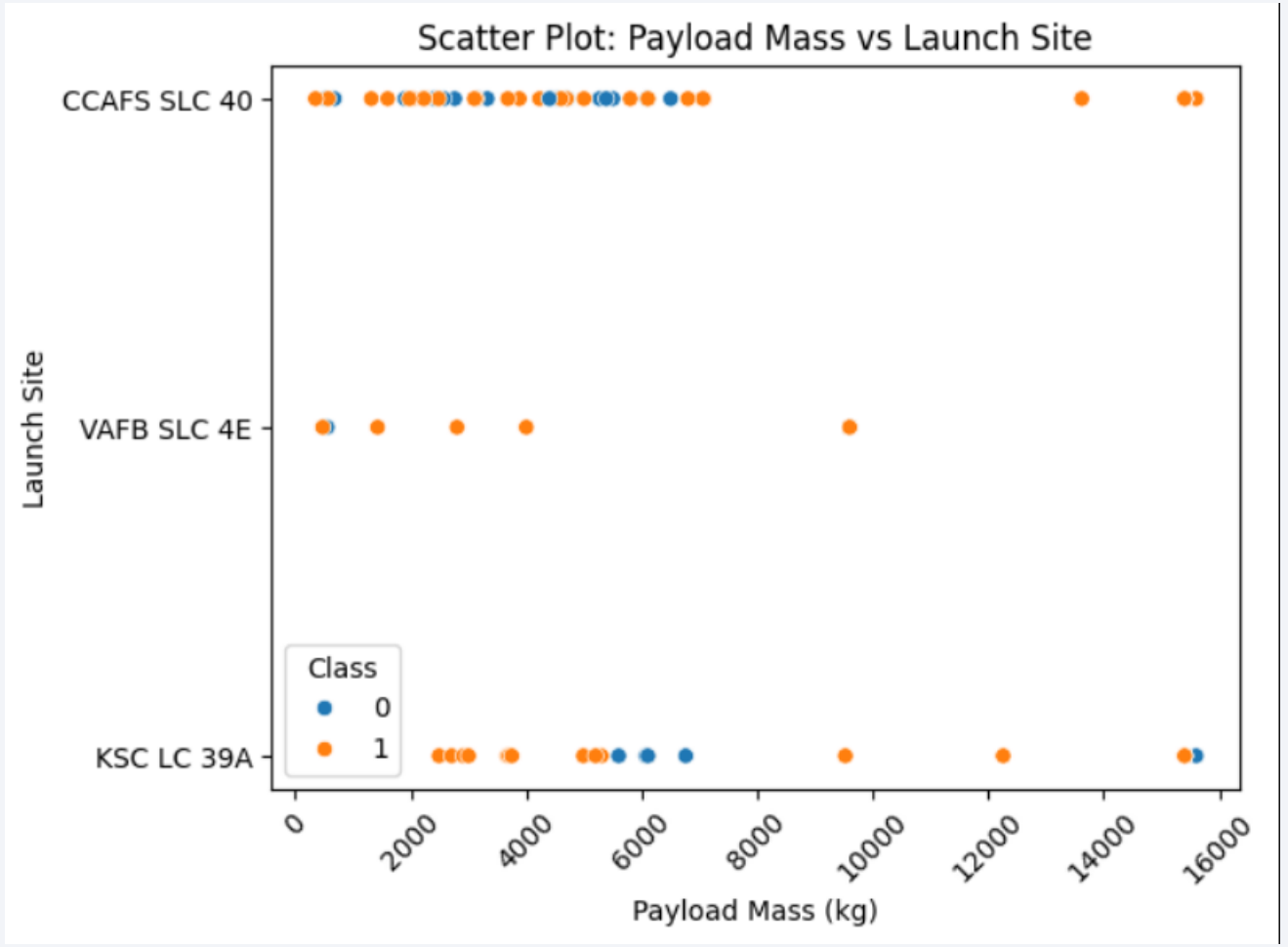
Flight Number vs. Launch Site

- It can be seen from the graph that the high the number of flights in a launch site the higher the success rate is.



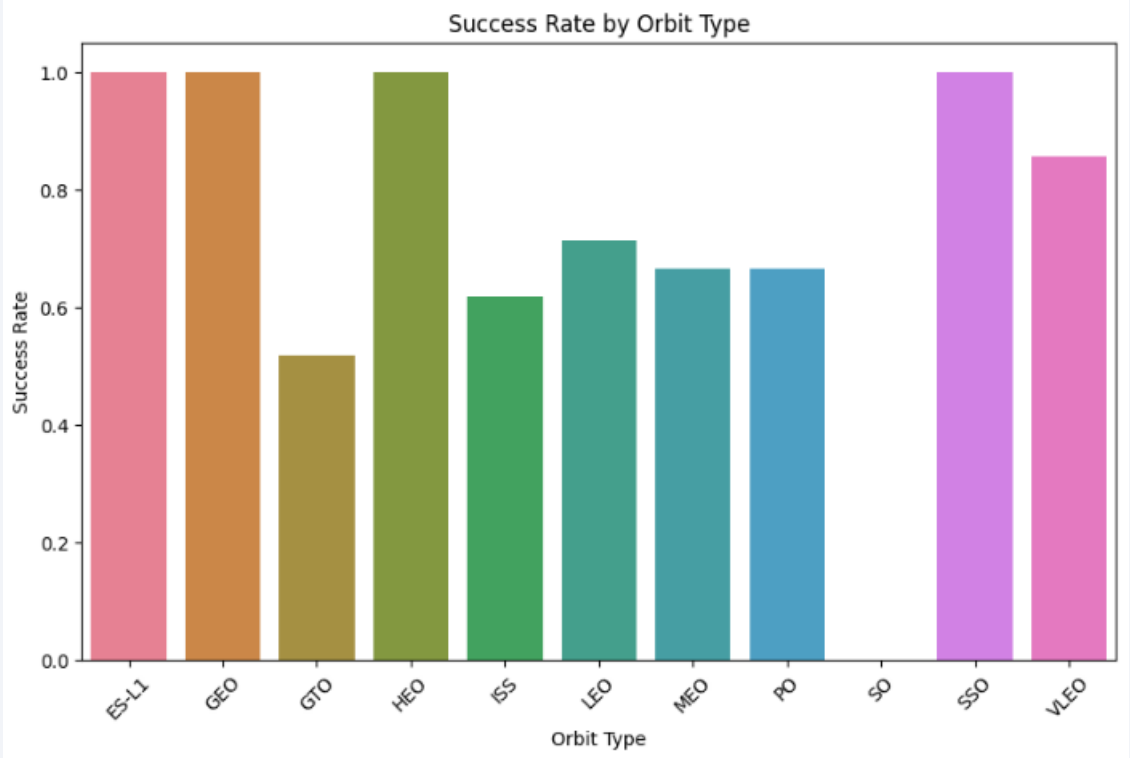
Payload vs. Launch Site

- It can be seen that for VAFB there is no rockets launched after 10000 Kgs.
- The Higher the payload Mass is the lower the chances of failure are.



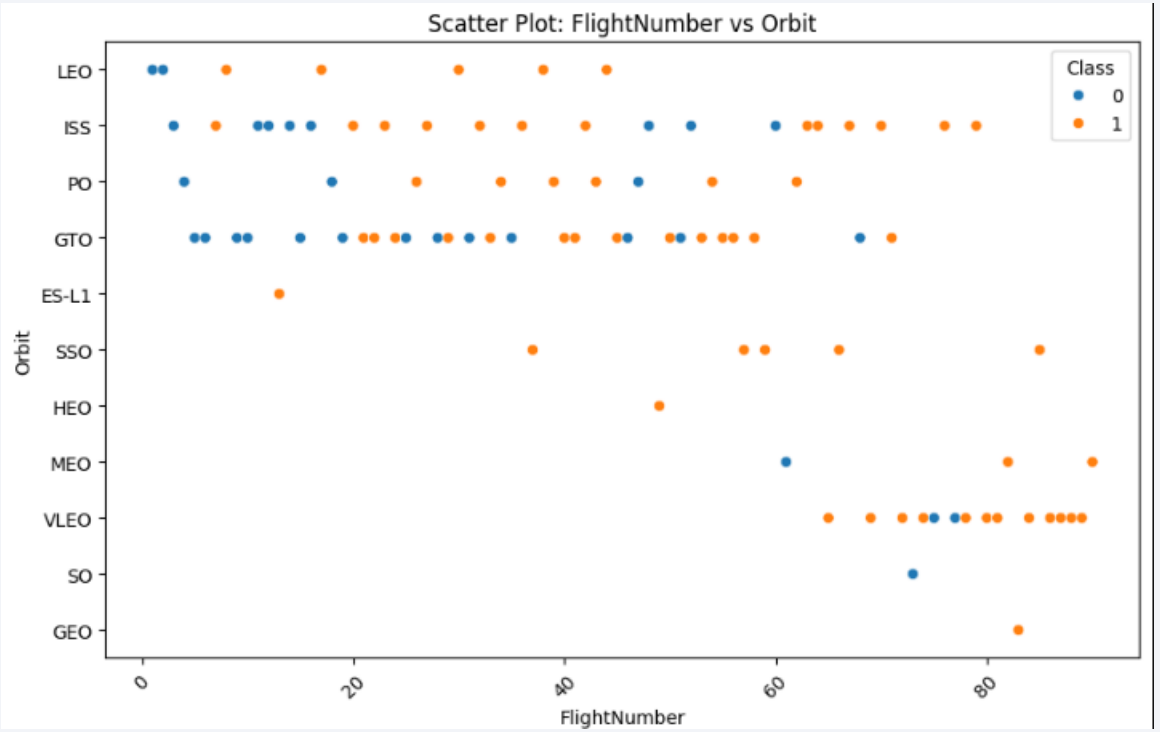
Success Rate vs. Orbit Type

- It can be seen that the most successful orbits are ES-L1, GEO, HEO, SSO.



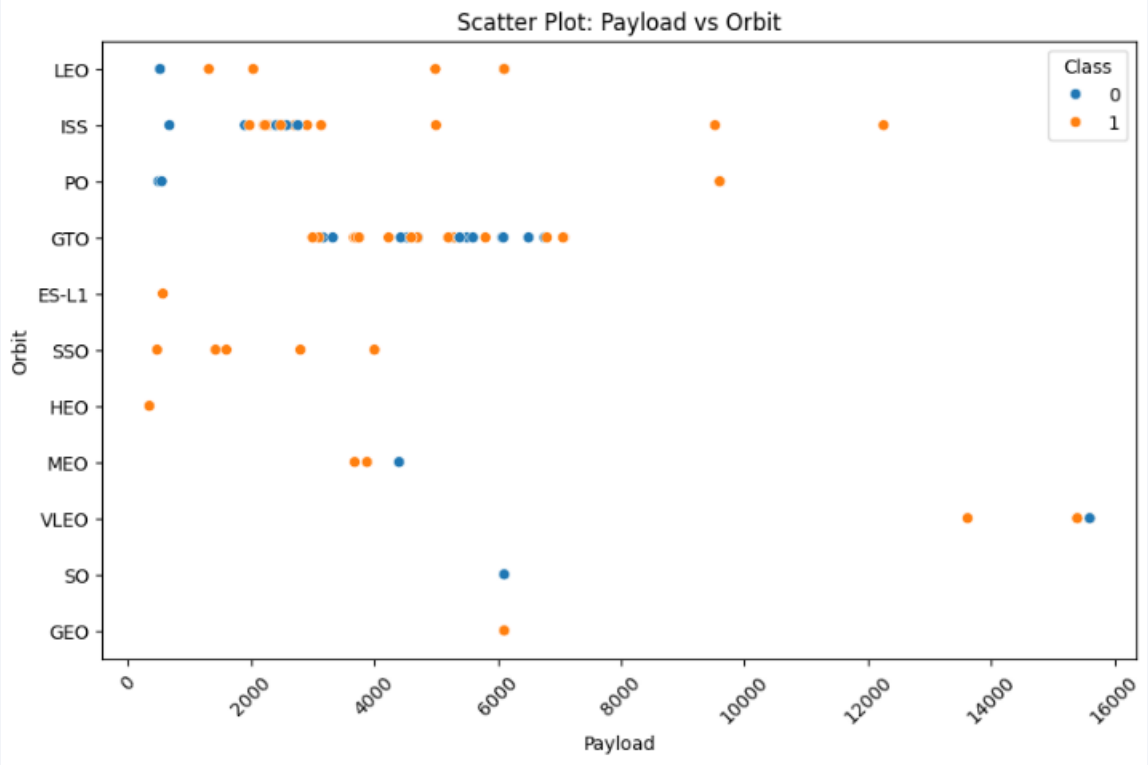
Flight Number vs. Orbit Type

- Observing the graph of flight numbers vs Orbits. It can be seen that in Leo as the number of flight increases the success increases, while in the case of GTO, there is no relationship between the two.



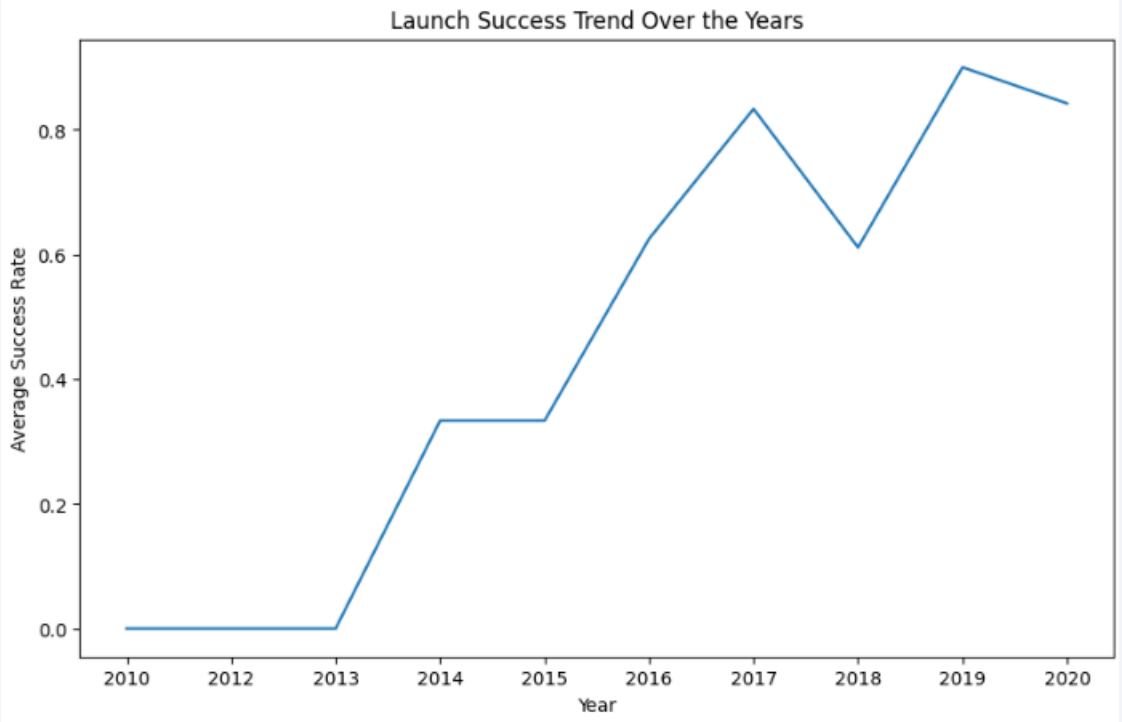
Payload vs. Orbit Type

- In the orbits LEO, ISS and PO, it can be noticed that as the payload increases success increases as well.



Launch Success Trend

- We can observe that the success rate has started increasing from the year 2013 all the way to 2020.



```
[10]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
 * sqlite:///my_data1.db
Done.
[10]:
```

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

All Launch Site Names

- With the use of the word DISTINCT. We get only the names of the unique launch_sites.

Task 2

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

In [10]:

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

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

Launch Site Names Begin with 'CCA'

- The query above is used to display a limited number of launch sites that begin with the string ‘CCA’.

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

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

SUM(PAYLOAD_MASS__KG_)
45596

Total Payload Mass

- From the query above we Calculated the total payload carried by boosters from NASA. It came out t be 45596 KGs.

Task 4

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

In [16]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) \
     FROM SPACEXTABLE \
     WHERE Booster_Version = 'F9 v1.1';
```

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

Out[16]:

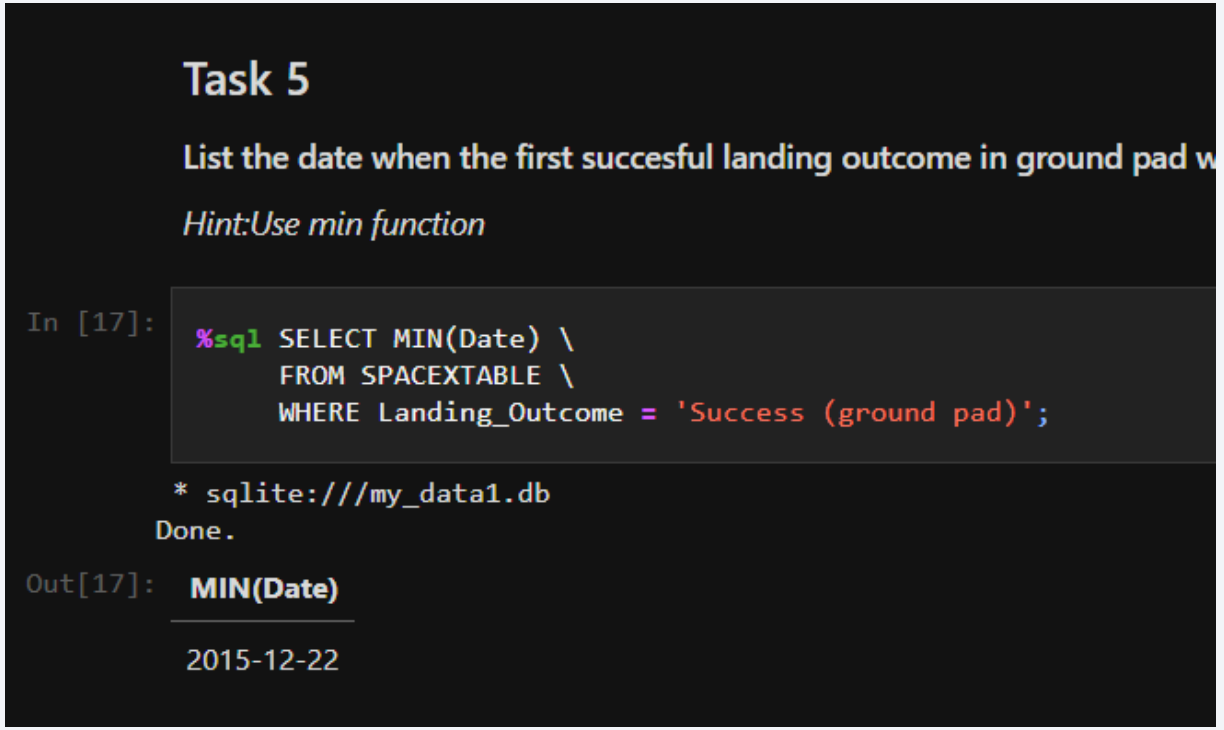
AVG(PAYLOAD_MASS__KG_)
2928.4

Average Payload Mass by F9 v1.1

- In the quary above, we calculated the average payload mass carried by booster version F9 v1.1 which came out to be 2928.4 KGs.

First Successful Ground Landing Date

- The query shows that the dates of the first successful landing outcome on ground pad is 2015 -12 -22



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

[12]:
```
%sql SELECT Booster_Version \
FROM SPACEXTABLE \
WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

[12]:

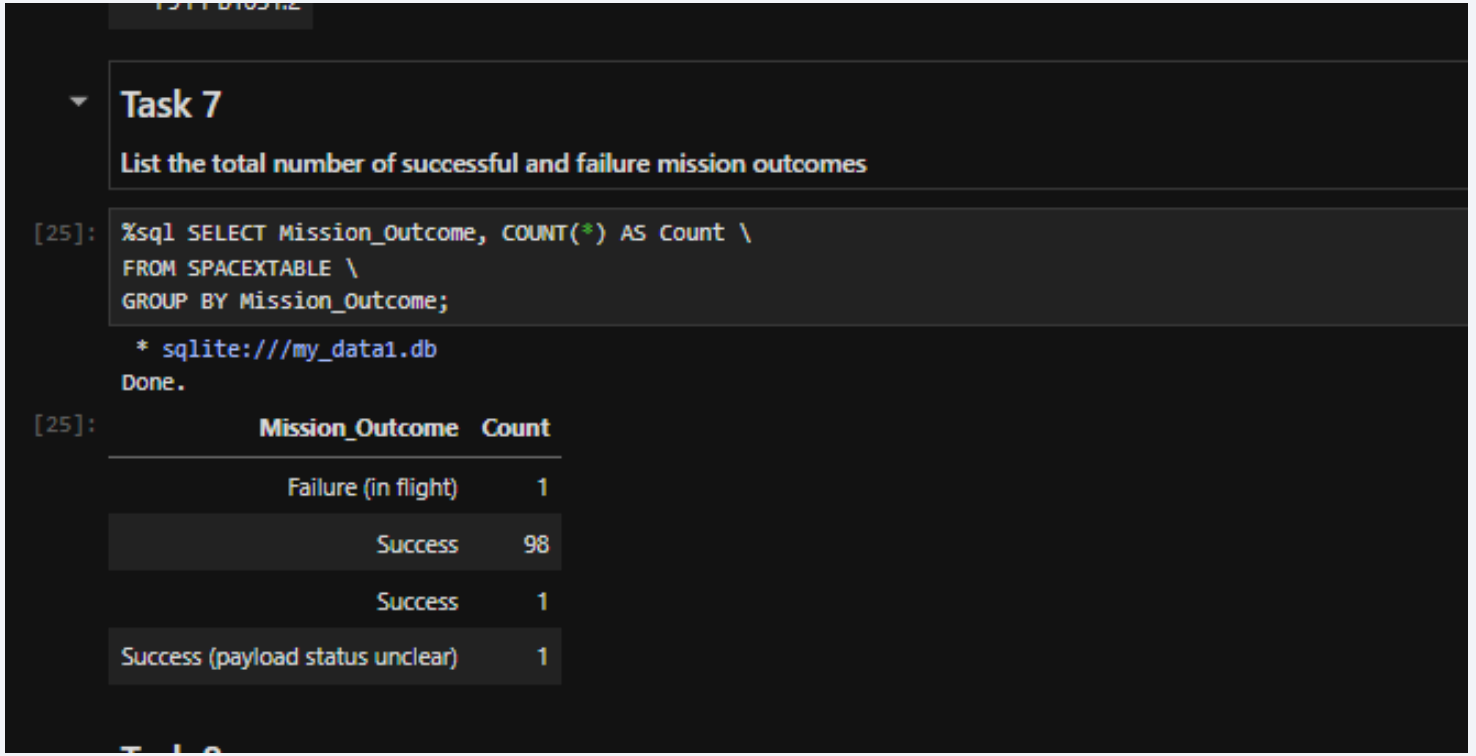
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Successful Drone Ship Landing with Payload between 4000 and 6000

- In the query, in order to get the List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000. we specified that the payload is between 4000 and 6000. As well as the outcome must be successful.

Total Number of Successful and Failure Mission Outcomes

- The query calculates the total number of successful and failure mission outcomes.



Task 7

List the total number of successful and failure mission outcomes

[25]:
```
%sql SELECT Mission_Outcome, COUNT(*) AS Count \
FROM SPACEXTABLE \
GROUP BY Mission_Outcome;
```

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

[25]:

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

Boosters Carried Maximum Payload

- In this query we list the names of the booster version which have carried the maximum payload.

Task 8

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

[29]:
```
%sql SELECT Booster_Version \
FROM SPACEXTABLE \
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE);
```

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

[29]:

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

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 ye

[40]:
```
%sql SELECT substr("Date", 6, 2) AS "Month", [Landing_Outcome] , Booster_Version , Launch_Site \
FROM SPACEXTABLE \
WHERE substr("Date", 0, 5) = '2015' AND [Landing_Outcome] = 'Failure (drone ship)';
```

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

[40]:

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

2015 Launch Records

- In the query we display the month, the failure landing_outcome, booster_version and launch_site for the year 2015.

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.

[60]:
```
%sql SELECT [Landing_Outcome], COUNT(*) AS count_outcomes \
FROM SPACEXTABLE \
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY [Landing_Outcome] ORDER BY count_outcomes DESC;
```

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

[60]:

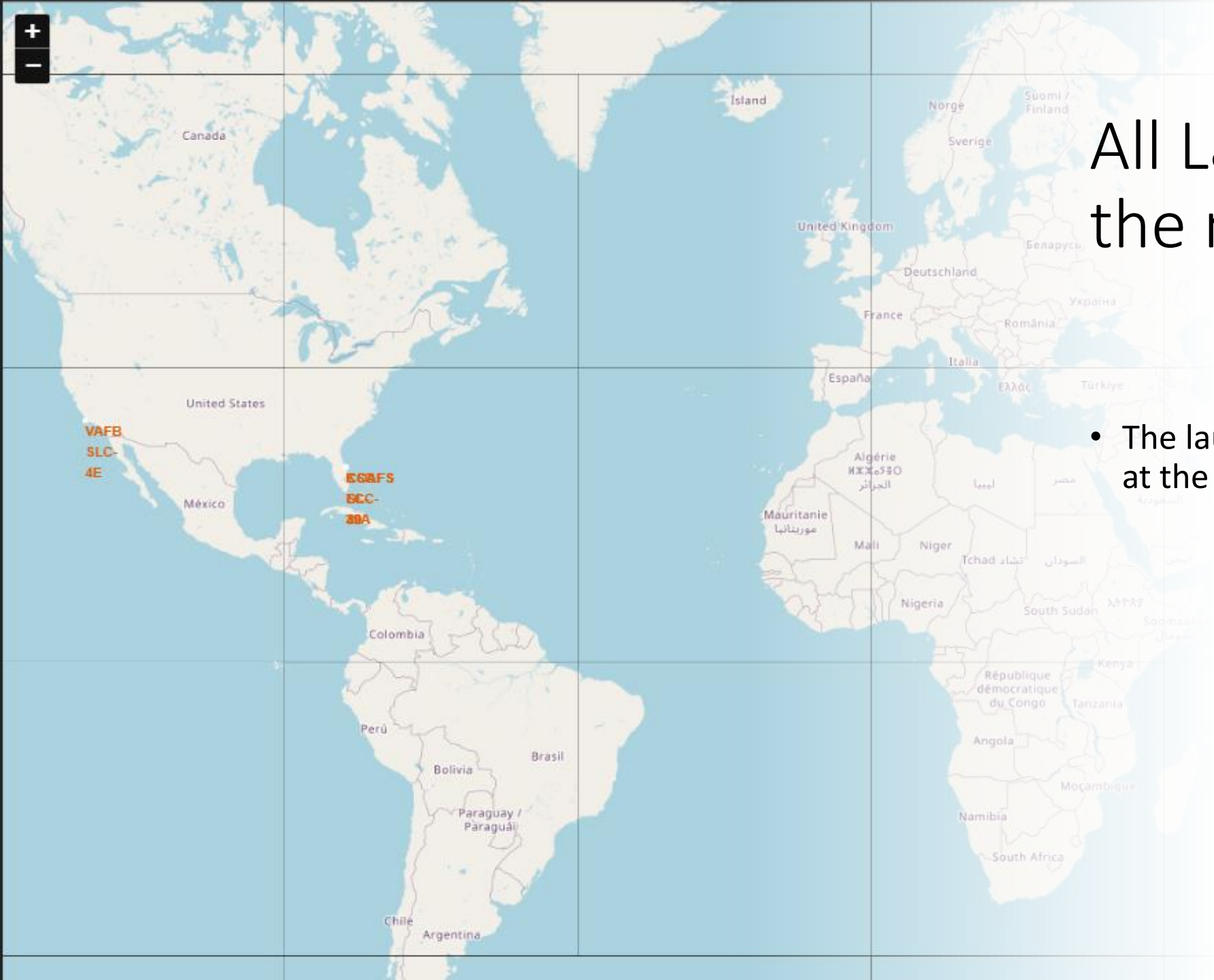
Landing_Outcome	count_outcomes
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3

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

- This query we select the landing_outcome, and count in order to count the landing outcome. In the date between 2010-06-04 and 2017-03-20.

Section 3

Launch Sites Proximities Analysis

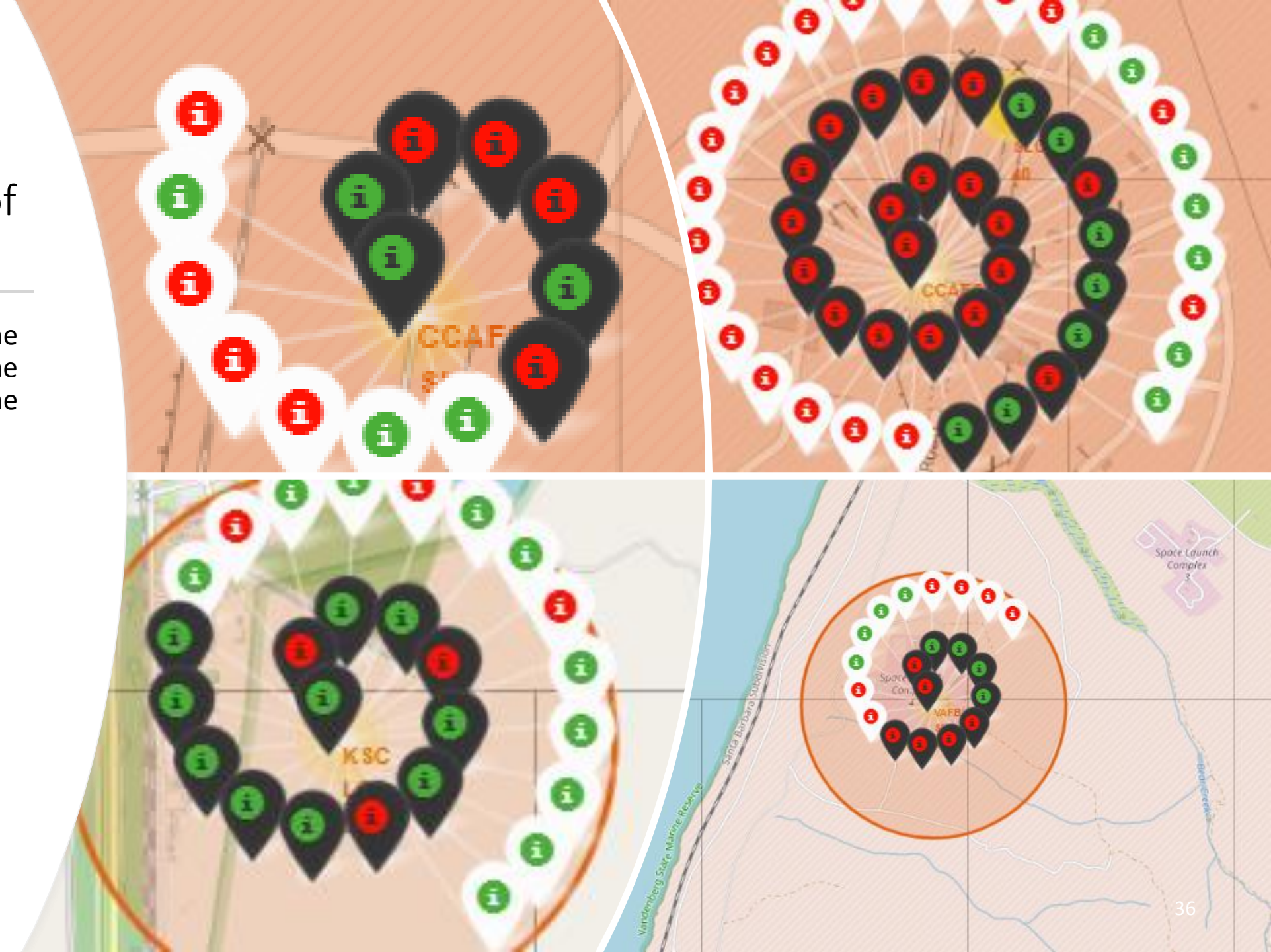


All Launch sites at the map

- The launch sites of Space X are at the west of the USA.

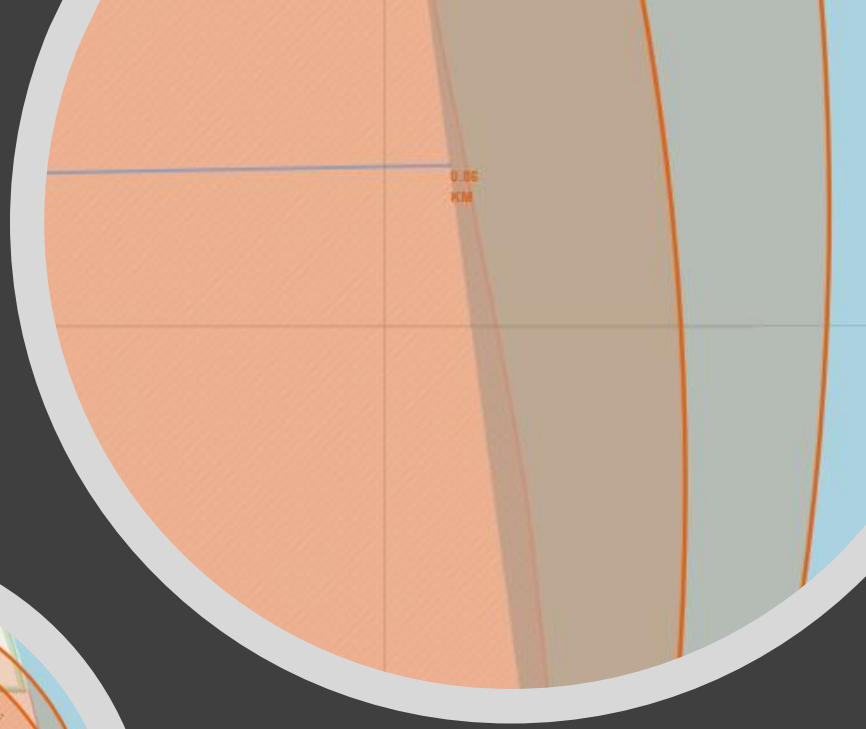
Marked areas representing the success or failure of the launch

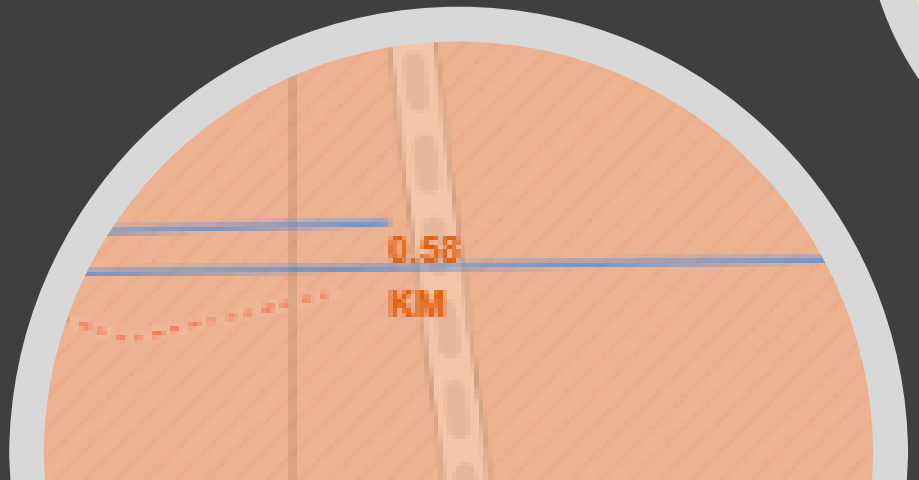
- The green marker in the Map represesnts the success while the red one represents the failure.

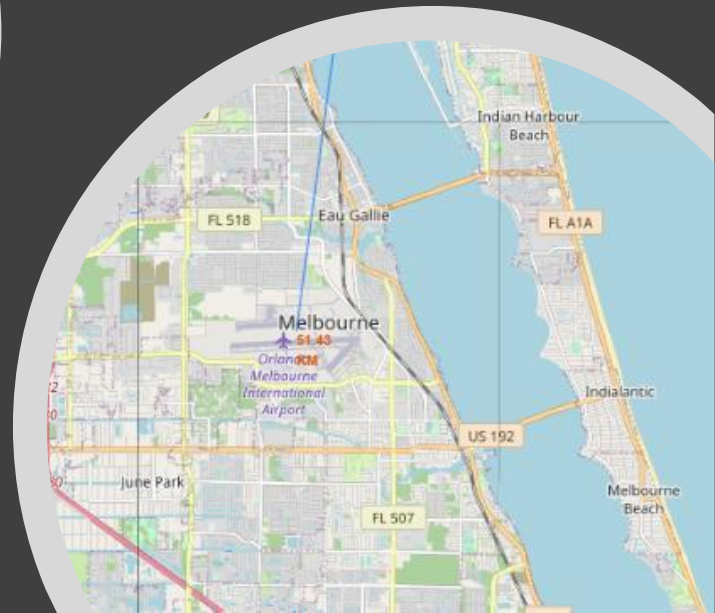


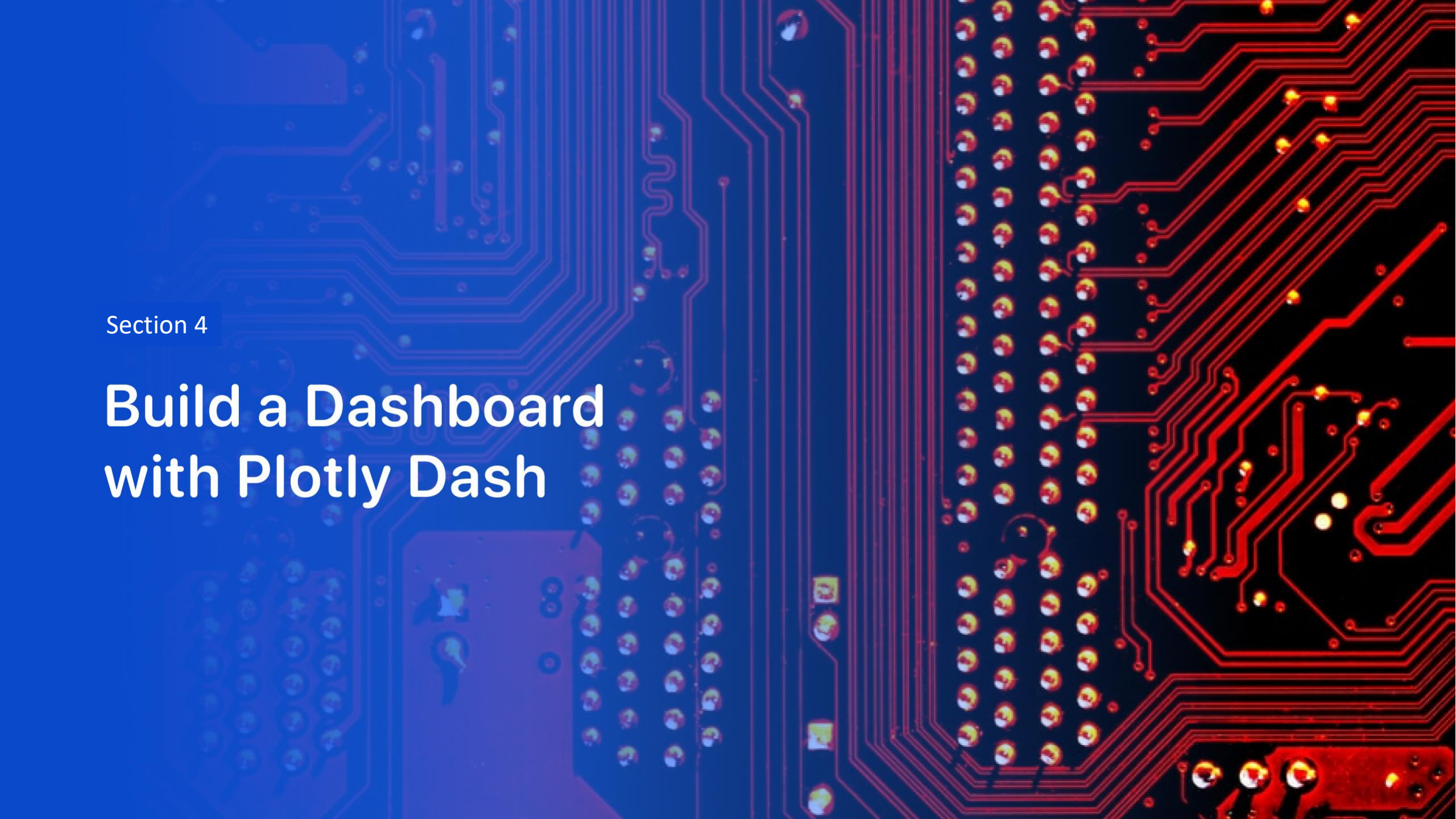
Launch Site distance to landmarks

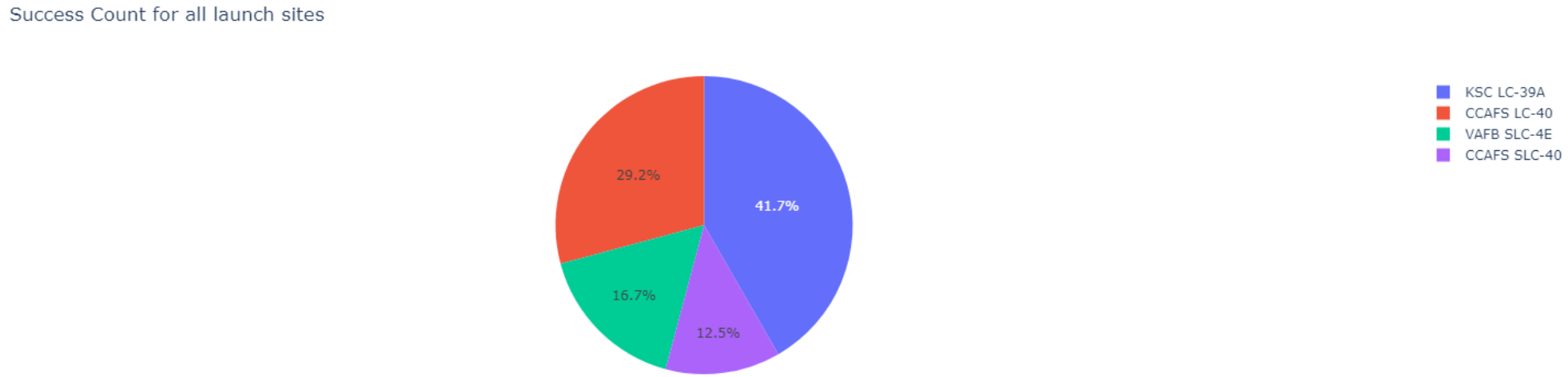
- The Screenshots show the closest City, Coastline, Highway, railway to the station.








Section 4
Build a Dashboard with Plotly Dash

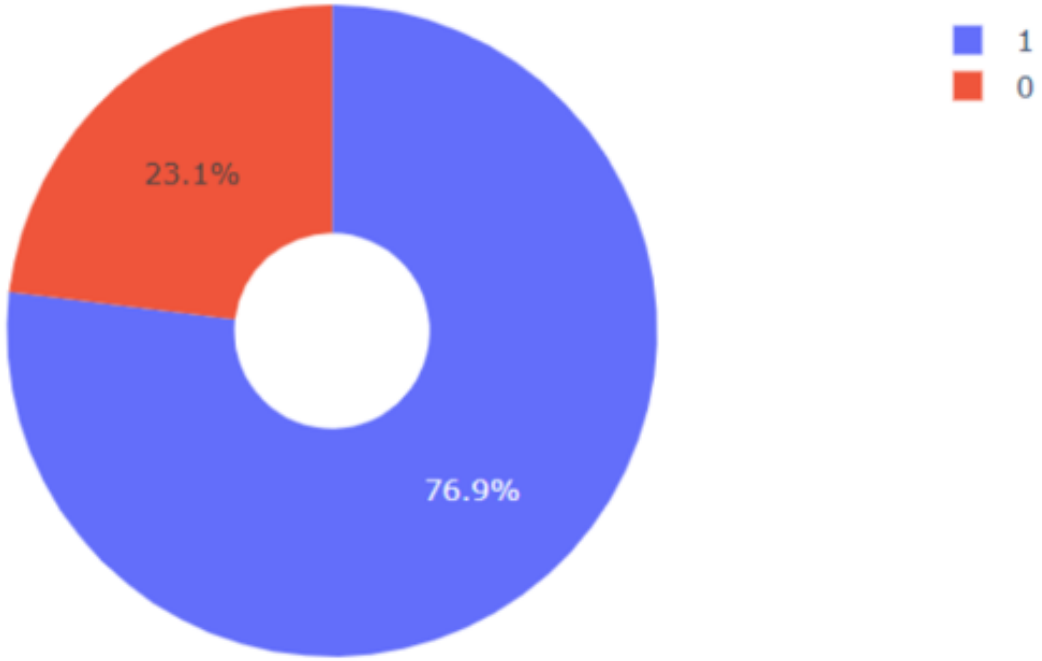


- We can notice that KSC LC-39A is the most successful of all the sites

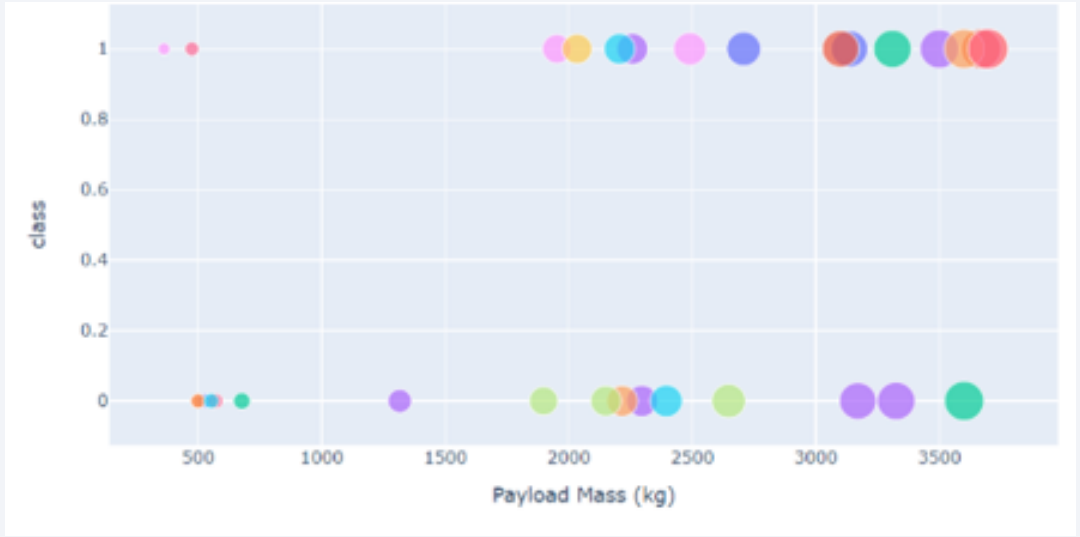
Success count of all launch sites

Launch site with the highest success

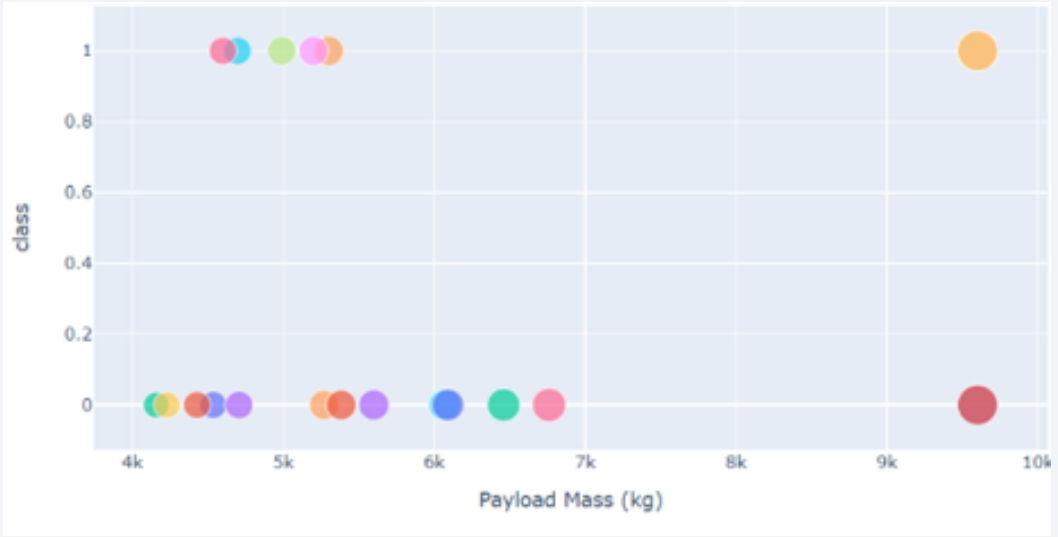
- KSC LC-39A has 76.9 success rate and 23.1 Failure rate



Scatter Graph of Payload Vs outcome for different ranges of payload



Payload ranges from 0 to 4000 KGs



Payload ranges from 4k to 10k Kgs

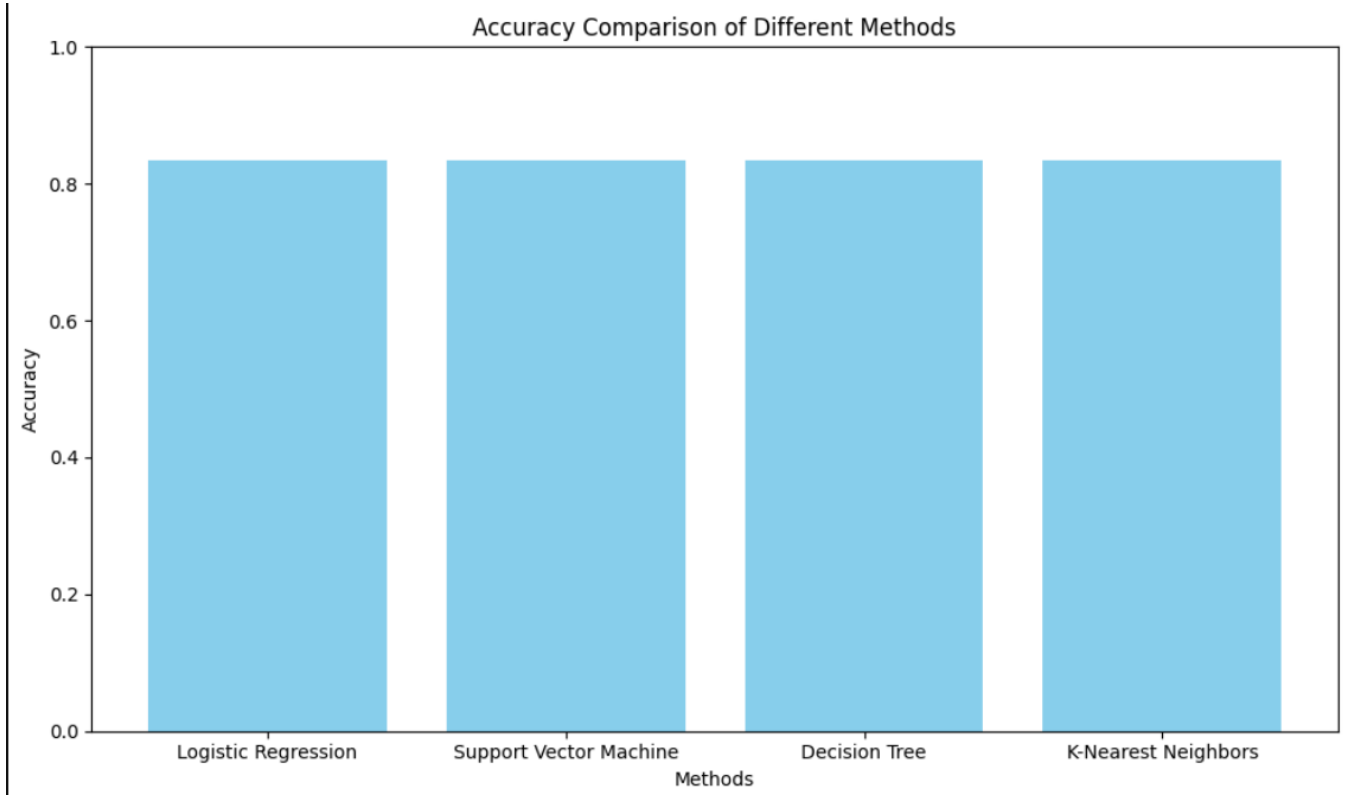
The higher the payload the lower the success rate

Section 5

Predictive Analysis (Classification)

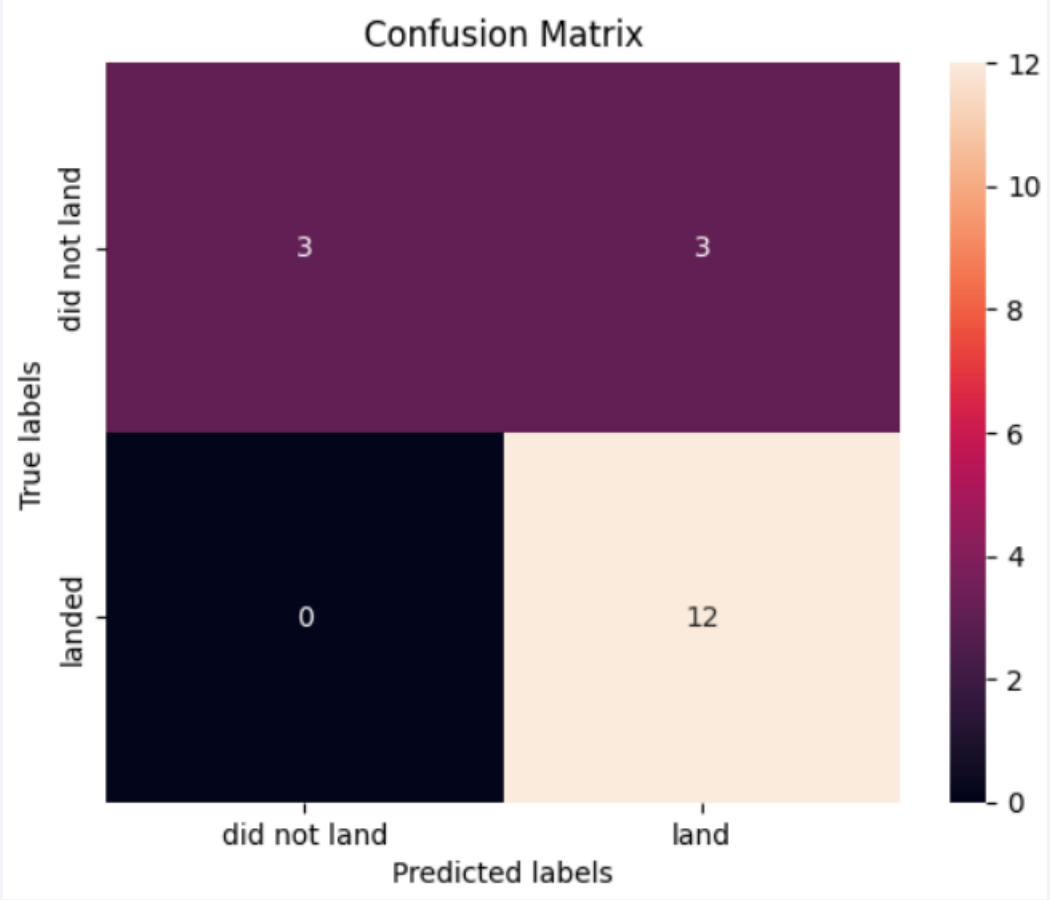
Classification Accuracy

- We can observe that the accuracy of all the models is equal. Making them all fit for use and equal.



Confusion Matrix

- We can see in the confusion Matrix that 12 were predicted correctly in the case of success and 3 were predicted correctly for the case of failure.



Conclusions

- In Conclusion:
- ES-L1, GEO, HEO, SSO are the most successful orbits.
- There 4 unique launch sites CCAFS LC-40, VAFB SLC-4E, KSC LC-39AC, CAFS SLC-40
- The trend of launch success started in the year 2013.
- KSC LC-39A is the launch site with the highest success rate.
- All 4 of the methods used for machine learning are fit to do the job.

Thank you!