Winning Space Race with Data Science

Palani Vigneshwar
16/03/2024

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- **Summary of all results**
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result from Machine Learning Lab

Introduction

Since SpaceX can reuse the first stage it offers rocket launches as low as 62 million dollars; while other providers cost upward of 165 million dollar each.

The goal of this project is to create the machine learning pipeline to predict the landing outcome of the first stage in the future.This project is crucial in identifying the right price to bid against SpaceX for a rocket launch.

However following problems needs to be addressed:

1. Identifying all factors that influence the landing outcome.
2. The relationship between each variables and how it is affecting the outcome.
3. The best condition needed to increase the probability of successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Web Scraping was done using Wikipedia.
- Perform data wrangling
  - One hot encoding was applied on the 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

Data collection is the process of collecting and analyzing information on relevant variables in a predetermined, methodical way so that one can respond to specific research questions, test hypotheses, and assess results.

For web scraping from Wikipedia, we will use the BeautifulSoup to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for further analysis

Data Collection - Scraping

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

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

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

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

```
7]: # Use soup.title attribute
    soup.title
```

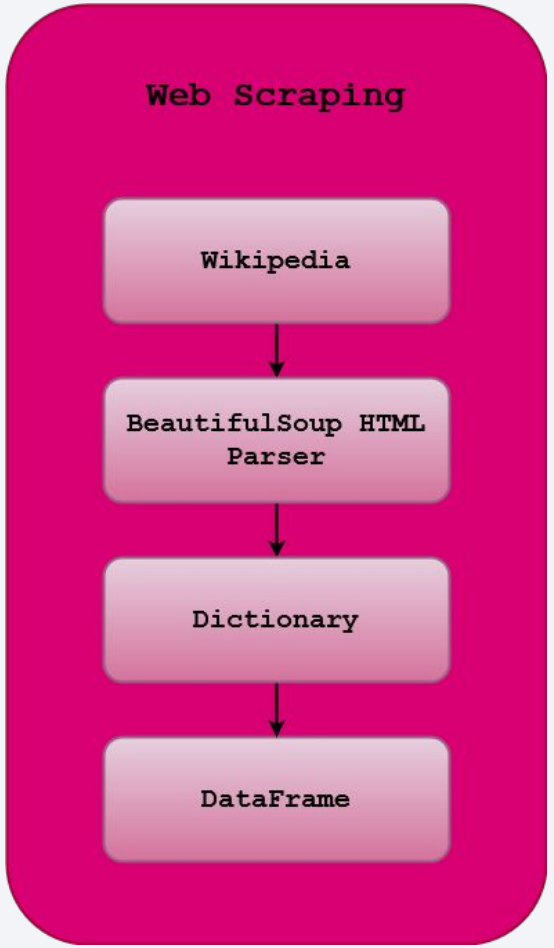
```
7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

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('tr')
    html_tables
```



https://github.com/palanivigneshwar/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb

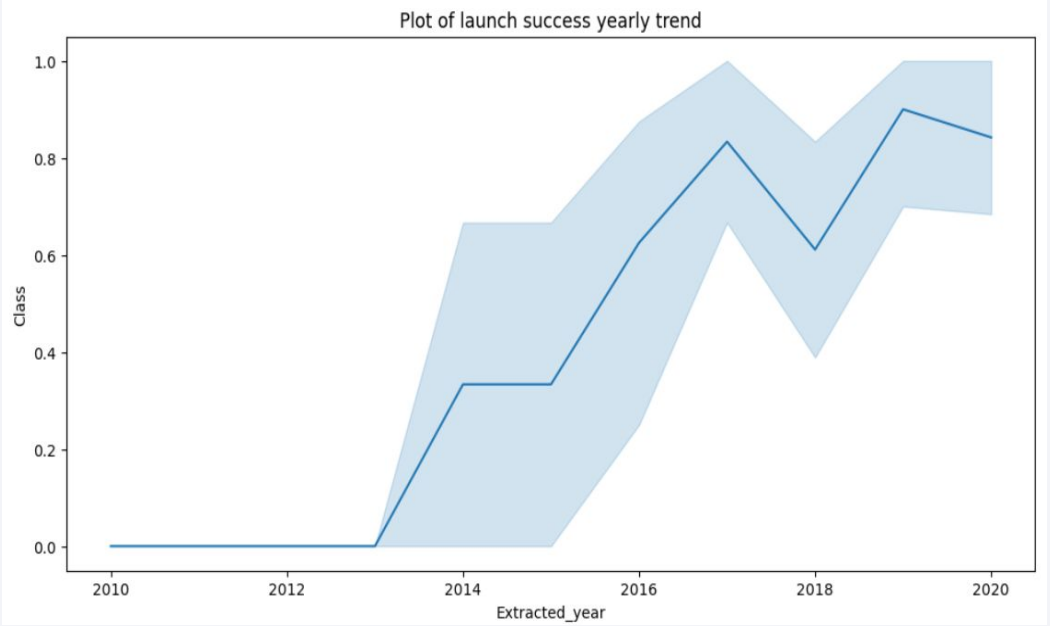
Data Wrangling

- Data Wrangling is the process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis (EDA).
- We calculated number of occurrences of each outcome.
- We modified the landing outcome label from 8 to 2.(0 for Failure and 1 for Success)

https://github.com/palanivigneshwar/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Scatter Plot:
    1. Flight Number vs. Payload Mass
    2. Flight Number vs Launch Site
    3. Payload Mass vs Launch Site
    4. Flight Number vs Orbit type
    5. Payload vs Orbit type
- Bar Plot:
    1. Success Rate in each orbit
- Line Chart
    1. Launch success yearly trend



https://github.com/palanivigneshwar/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

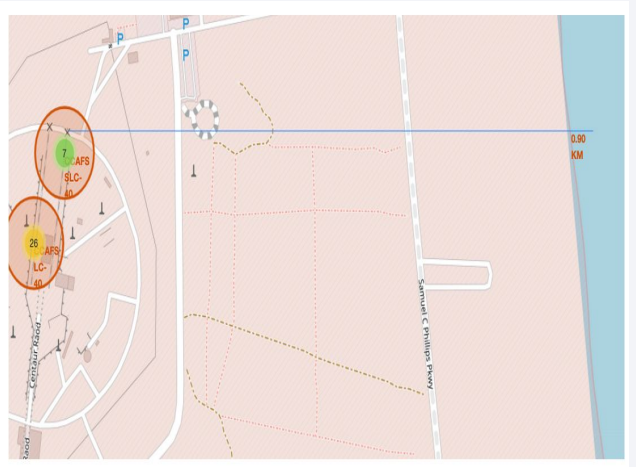
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass.
- 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.
- 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.

https://github.com/palanivigneshwar/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- To visualize the launch data into an interactive map. We took the latitude and longitude coordinates at each launch site and added a circle marker around each launch site with a label of the name of the launch site.
- We then assigned the dataframe launch_outcomes (failure,success) to classes 0 and 1 with Red and Green markers on the map in MarkerCluster().
- We calculate distances between the launching sites to its proximities. From this we determine whether launching sites are close to railways, highways and coastlines or do launch sites keep certain distance from cities.

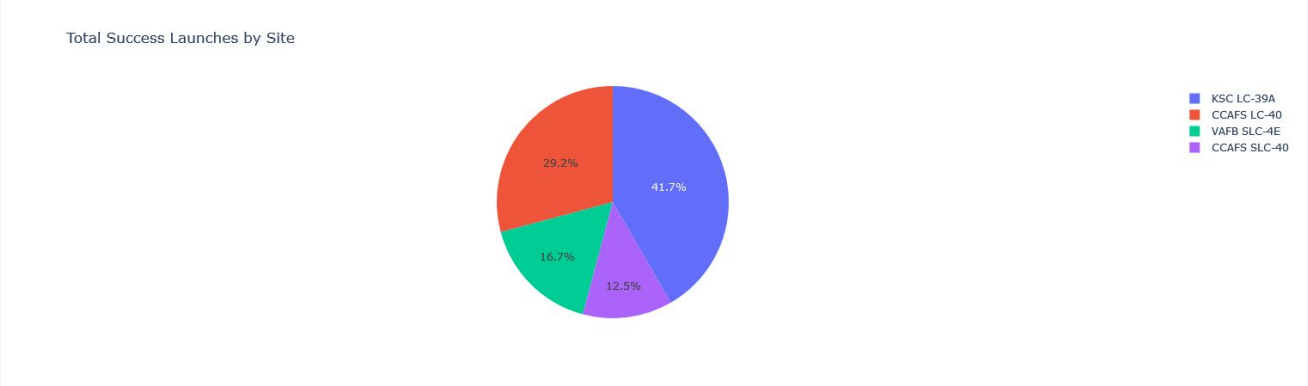
https://github.com/palanivigneshwar/Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

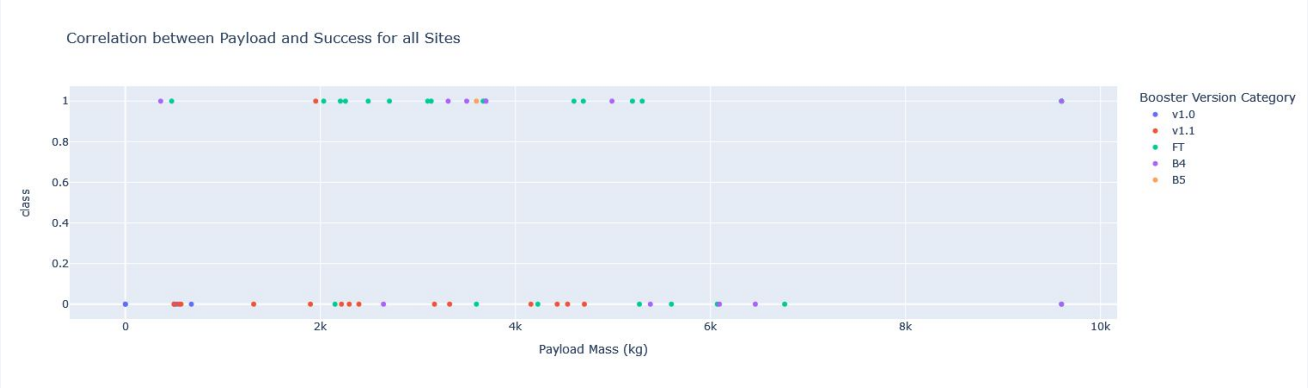


Build a Dashboard with Plotly Dash

- Plotly dash was used for making interactive dashboard.
- Success / Failure pie charts was plotted for certain sites.
- We then plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

https://github.com/palanivigneshwar/Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py





Predictive Analysis (Classification)

- Building Model

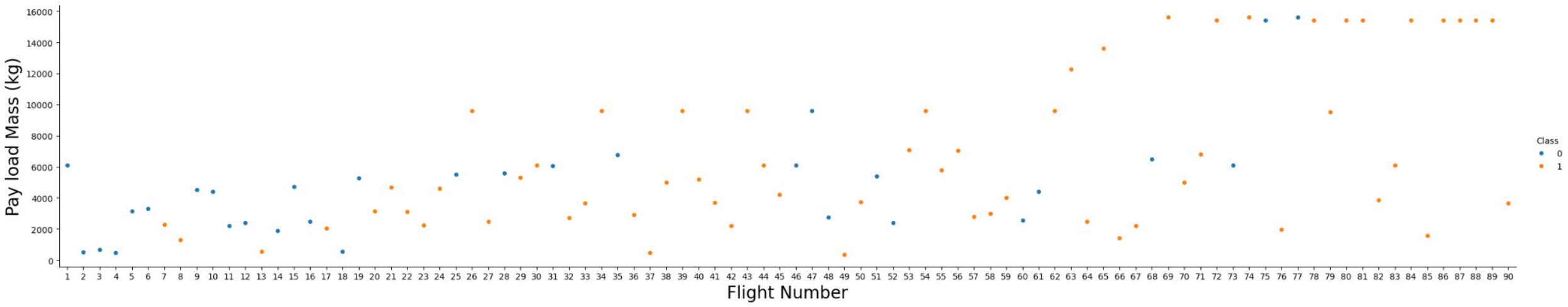
1. Import pandas, numpy and relevant libraries.
2. Import dataset using pandas to dataframe.
3. Standardize your independent features.
4. Perform train test split to dataframe (75-25).
5. Set parameters to gridsearchCV.

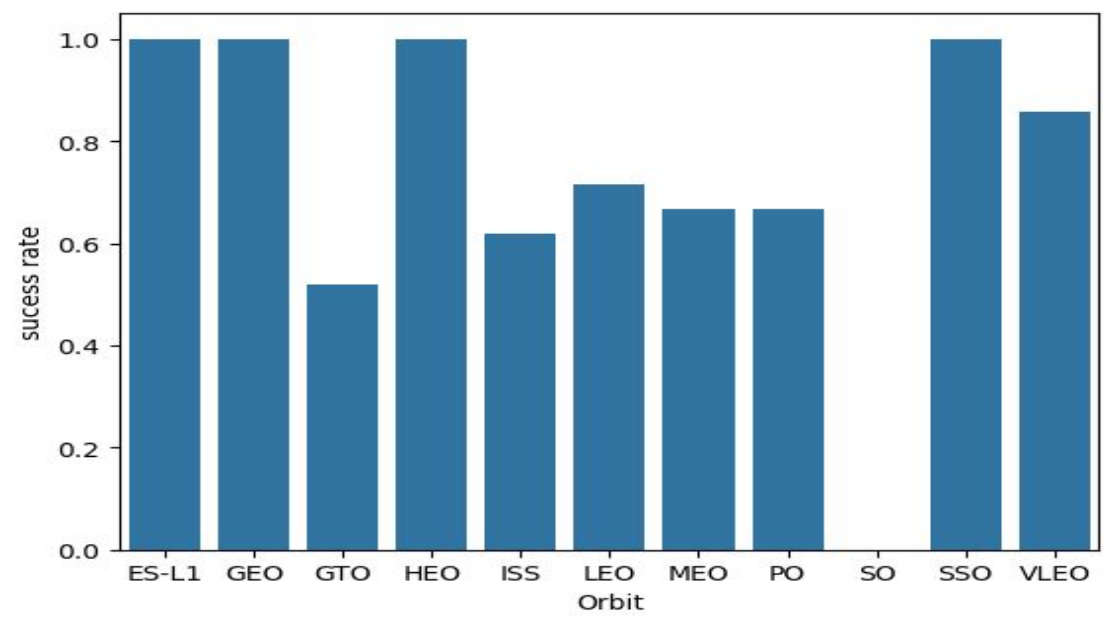
- Evaluating the Model

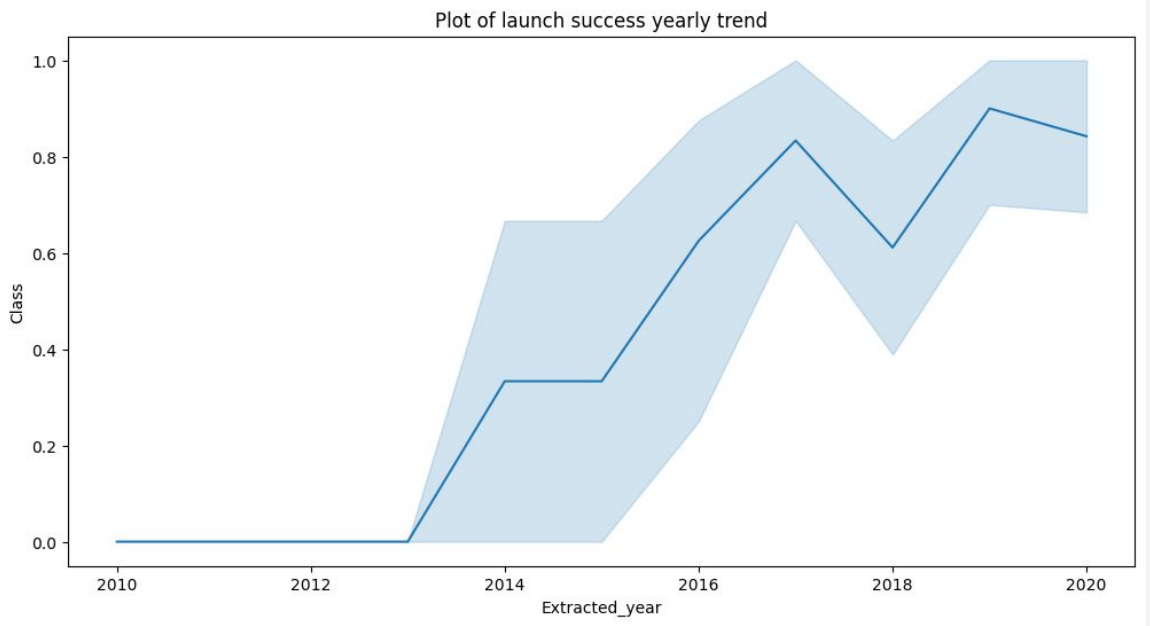
1. Use machine learning models to predict the dependent feature with the help of independent feature.
2. Use appropriate accuracy parameter to evaluate your model.
3. Select the model with the highest accuracy.

https://github.com/palanivigneshwar/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

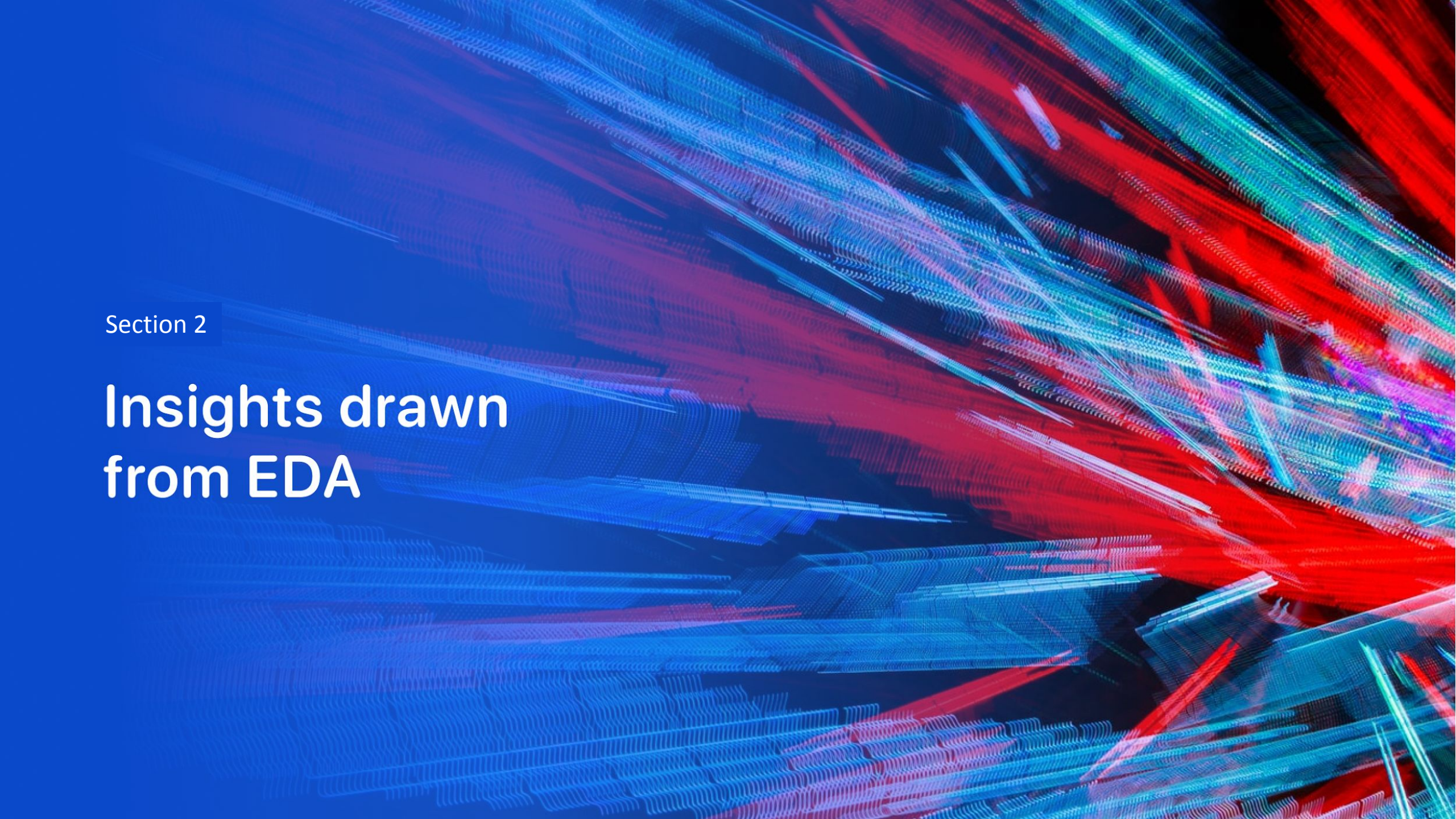

Pay load Mass (kg)
Flight Number
Class
0
1


sucess rate
Orbit
ES-L1
GEO
GTO
HEO
ISS
LEO
MEO
PO
SO
SSO
VLEO

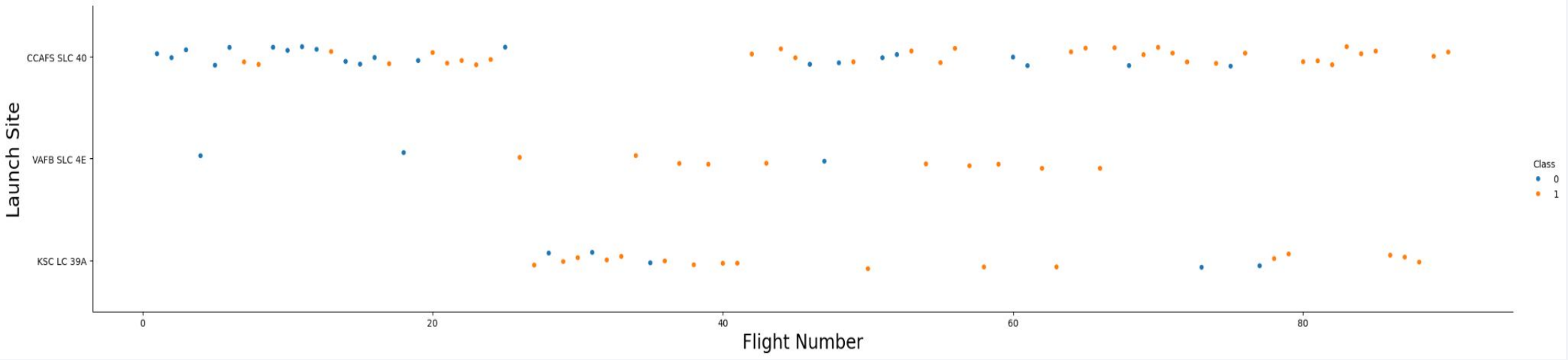

Plot of launch success yearly trend
Class
Extracted_year

Section 2

Insights drawn from EDA

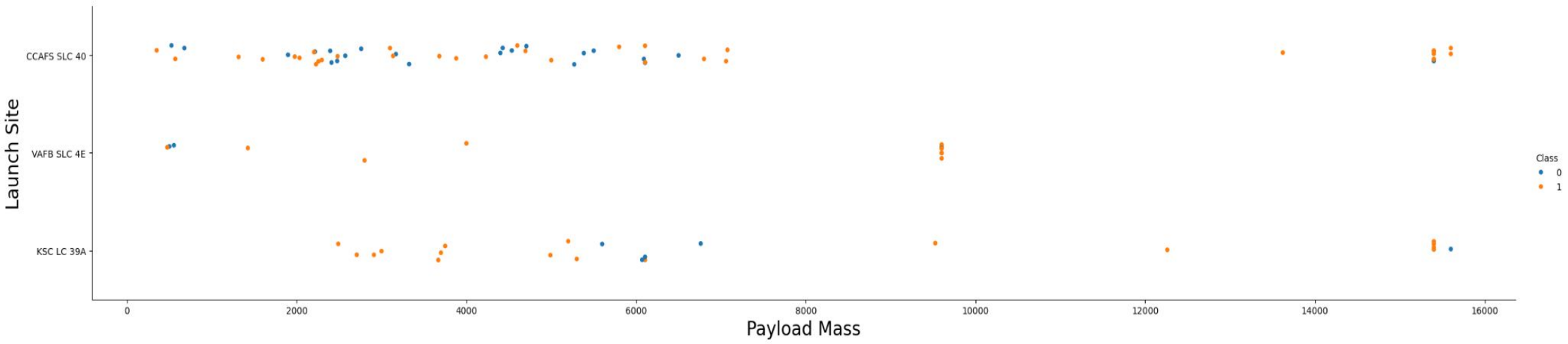


Flight Number vs. Launch Site



This scatter plot shows that the larger the flights amount of the launch site, the greater the the success rate will be. However, site CCAFS SLC40 shows the least pattern of this.

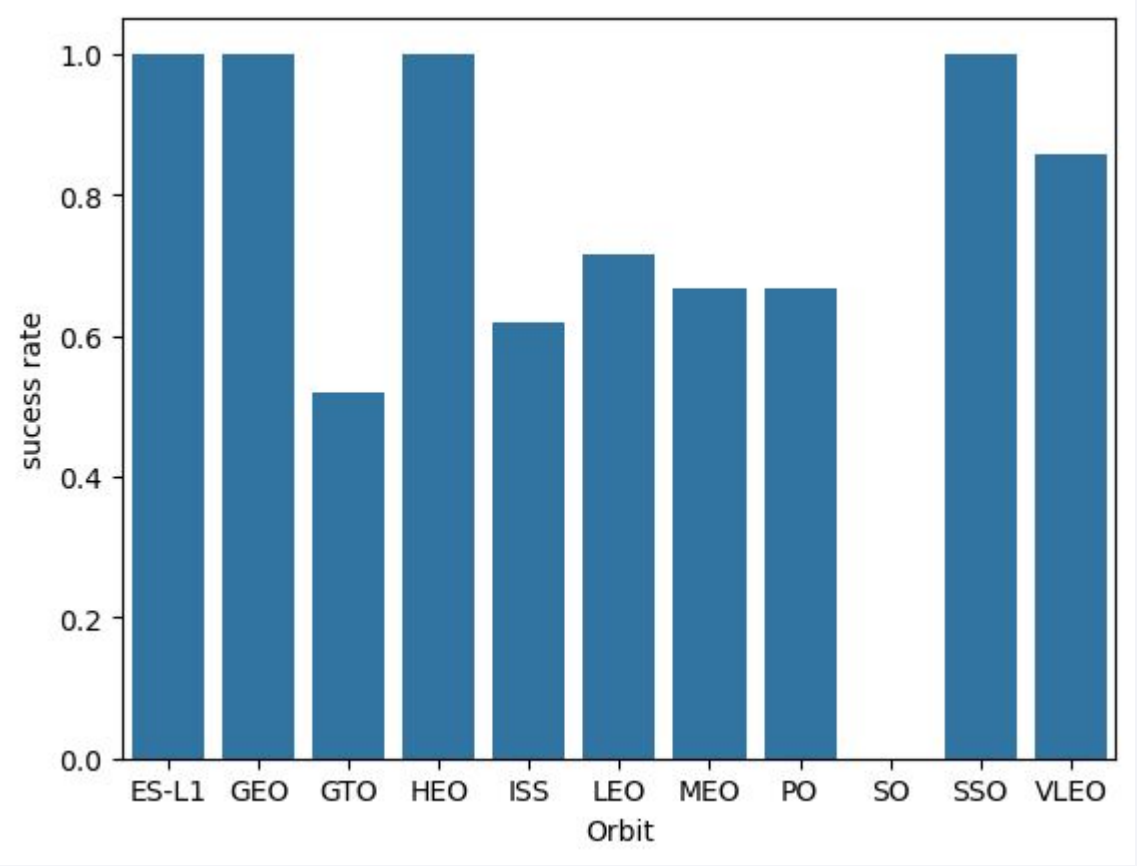
Payload vs. Launch Site



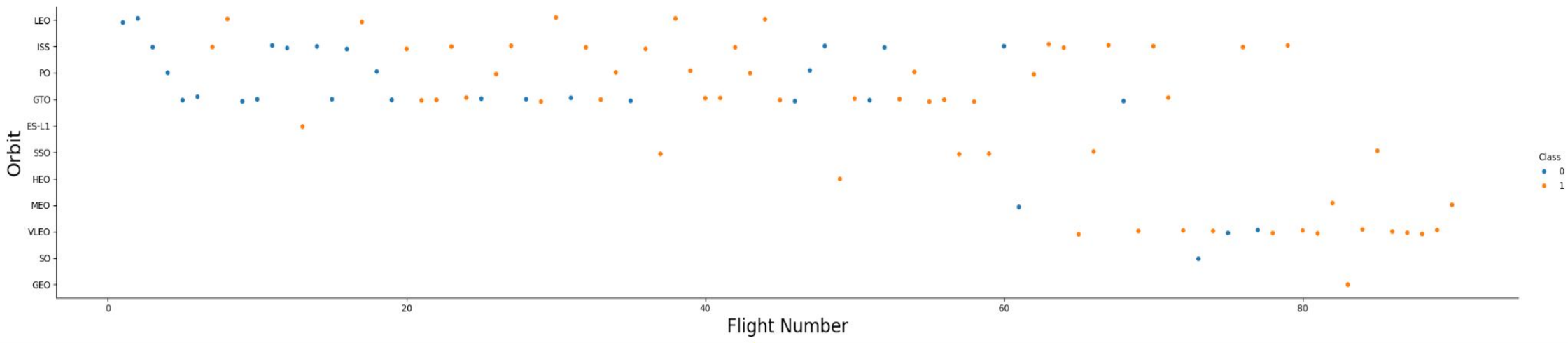
This scatter plot shows once the payload mass is greater than 7000 kg, the probability of the success rate will be highly increased. However, there is no clear pattern to say the launch site is dependent to the payload mass for the success rate.

Success Rate vs. Orbit Type

ES-L1, GEO, HEO, and SSO have highest success rate. SO has the worst success rate.

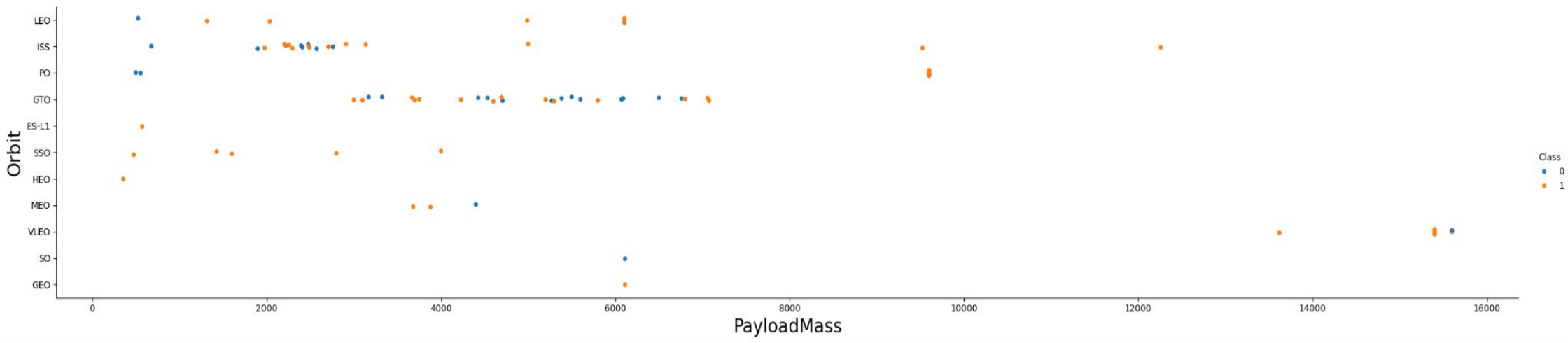


Flight Number vs. Orbit Type



This scatter plot shows that generally, the larger the flight number on LEO orbits, the greater the success rate however for GTO orbit which depicts no relationship between both attributes.

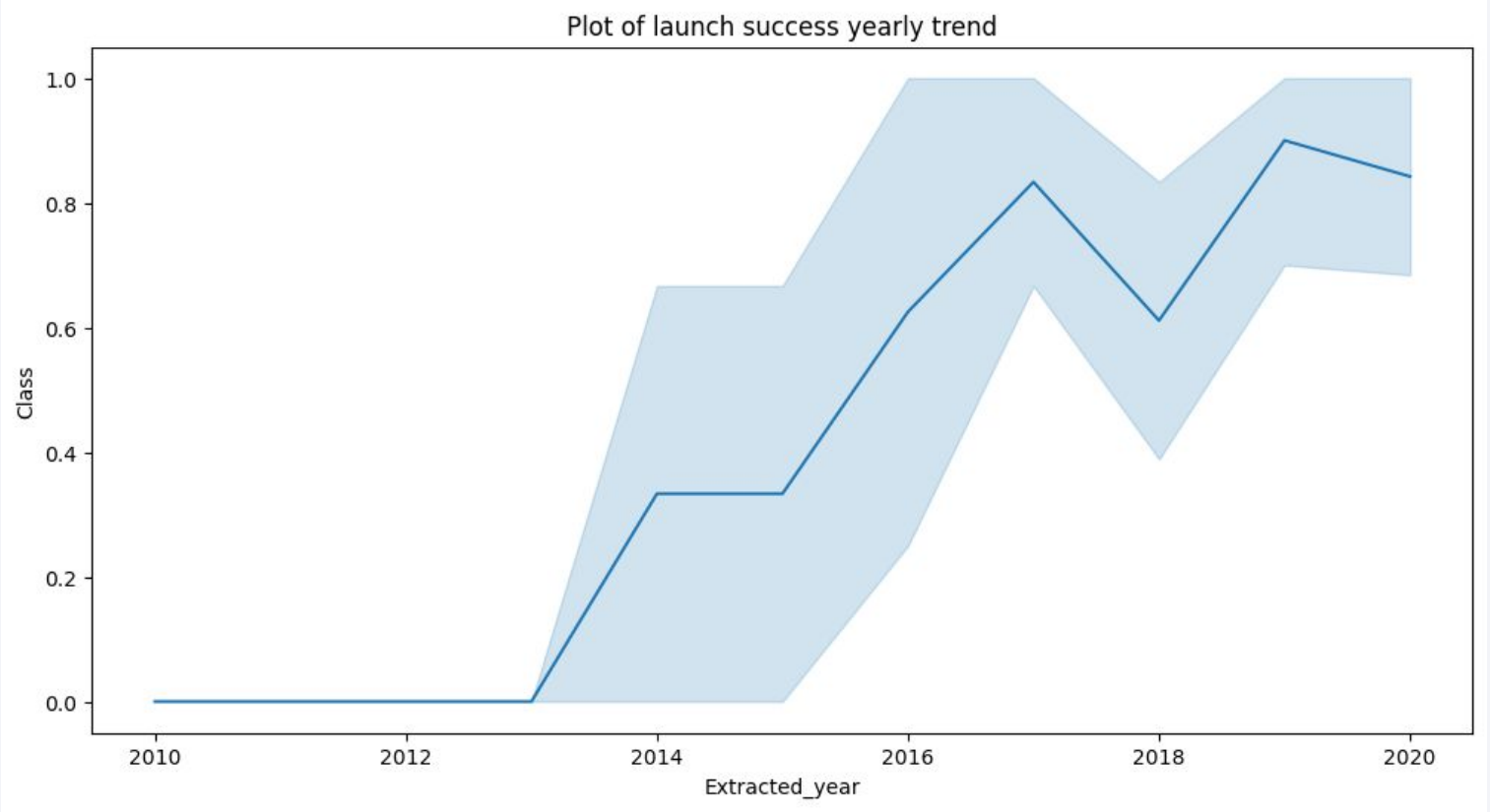
Payload vs. Orbit Type



- Heavy payloads successful landings for PO, LEO, and ISS orbits.

Launch Success Yearly Trend

This figures clearly depicted and increasing trend from the year 2013 until 2020.



All Launch Site Names

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

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

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

The query returns distinct names of launch sites from the table with the help of **DISTINCT** keyword.

Launch Site Names Begin with 'CCA'

We used the query above to display records where launch sites begin with ‘CCA’ with the help of **LIKE** keyword.

```
%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_Outc
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (paracl
2010-12-08	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 (paracl
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No atte
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No atte
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No atte

Total Payload Mass

We calculated the total payload carried by boosters from NASA as 45596 with the help of **SUM** function.

```
%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

We calculated the average payload mass carried by booster version F9 v1.1 as 2534.67 using **AVG** function.

```
%sql SELECT AVG(PAYLOAD_MASS__kg_) FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE "F9 v1.1%";
```

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

AVG(PAYLOAD_MASS__kg_)
2534.6666666666665

First Successful Ground Landing Date

```
%sql SELECT MIN(DATE) AS "First Successful Landing" FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

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

First Successful Landing
2015-12-22

We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015 using **MIN** function.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%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.
```

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

We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful Mission", \
    sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission" \
FROM SPACEXTBL;
```

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

Successful Mission	Failure Mission
100	1

We used wildcard like '%' to filter for WHERE Mission_Outcome was a success or a failure. We also used subqueries to show the results in the single line.

Boosters Carried Maximum Payload

We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX** function.

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

Booster Versions which carried the Maximum Payload Mass
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 SELECT substr(Date, 6,2) AS "Month_Names",Landing_Outcome,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL \
WHERE substr(Date,0,5)='2015' AND LANDING_OUTCOME = 'Failure (drone ship)';
```

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

Month_Names	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

We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015.

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

```
%sql SELECT LANDING_OUTCOME as "Landing Outcome", COUNT(LANDING_OUTCOME) AS "Total Count" FROM SPACEXTBL \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING_OUTCOME ORDER BY COUNT(LANDING_OUTCOME) DESC ;
```

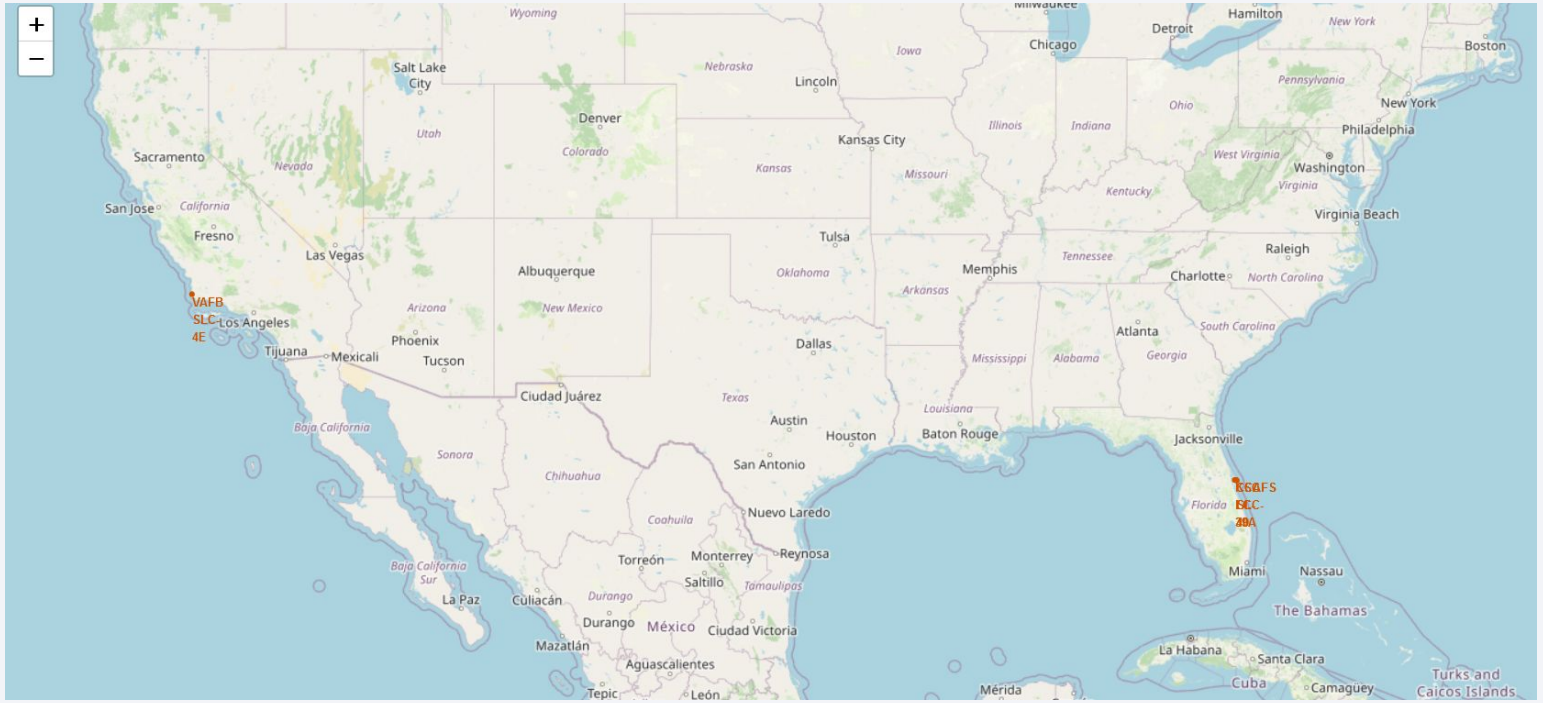
```
 * sqlite:///my_data1.db
Done.
```

Landing Outcome	Total 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

We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20. We used **GROUP BY** to get count for individual landing outcome counts and **ORDER BY** to display data in descending order.


Section 3
Launch Sites
Proximities Analysis

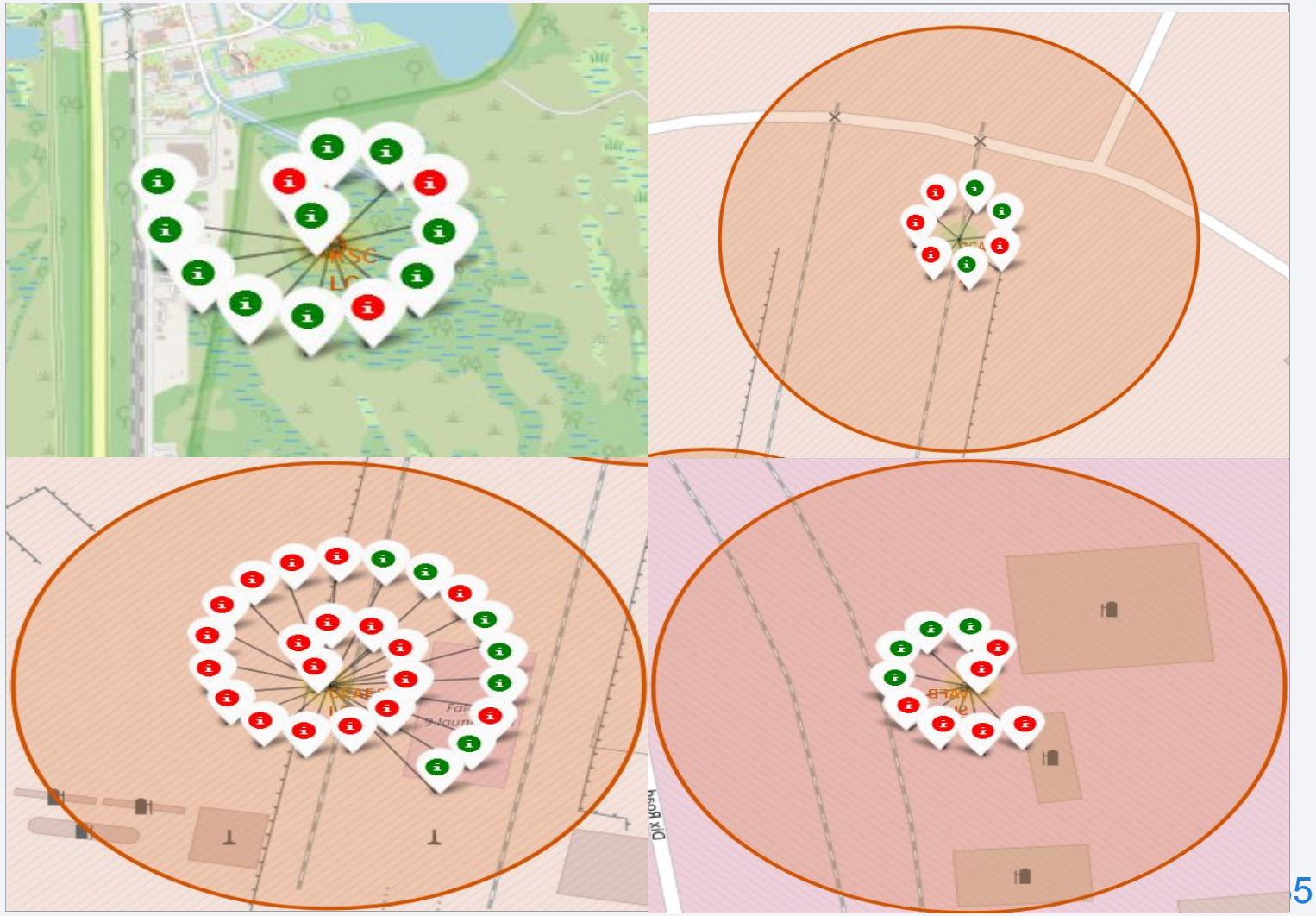
Location of all the Launch Sites



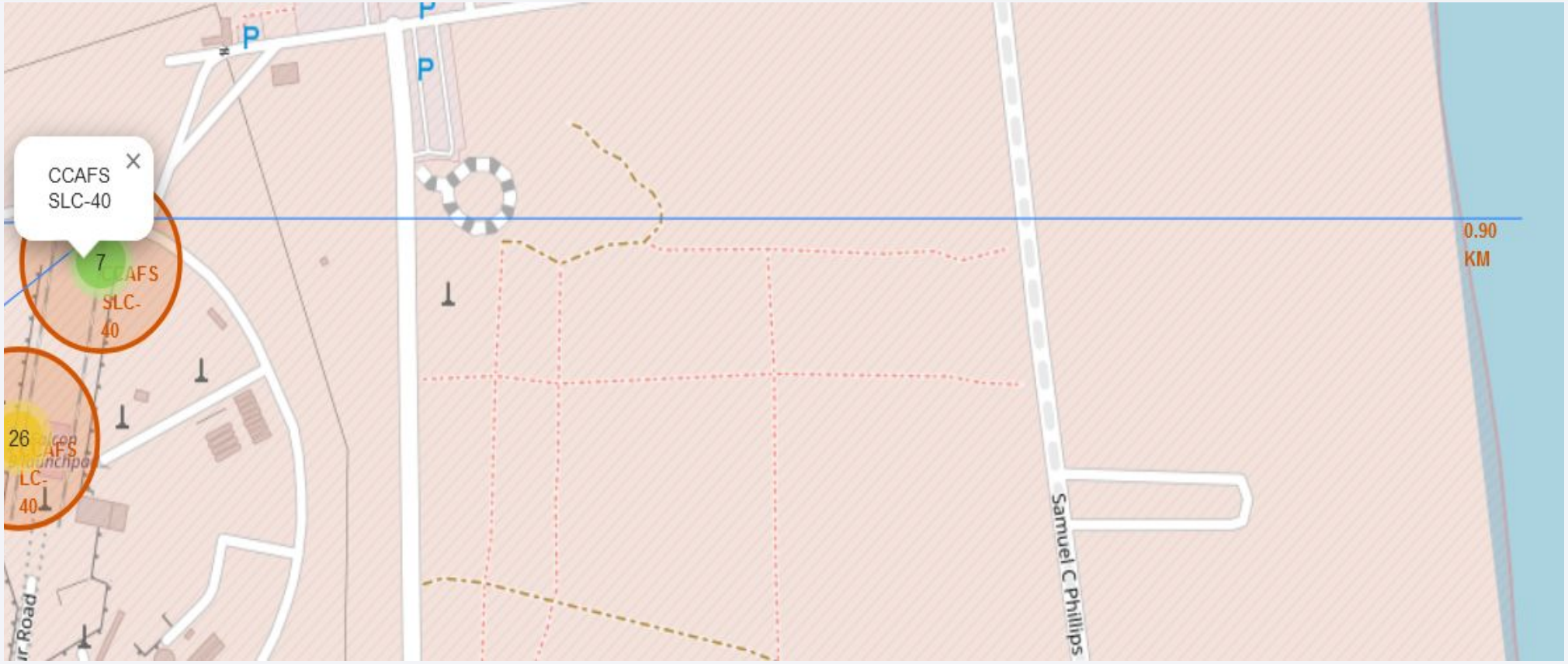
Launch Site	Latitude	Longitude	Direction
CCAFS LC-40	28.56230197	-80.57735648	South-East
CCAFS SLC-40	28.56319718	-80.57682003	South-East
KSC LC-39A	28.57325457	-80.64689529	South-East
VAFB SLC-4E	34.63283416	-120.6107455	South-West

Display Launch Outcome by Color

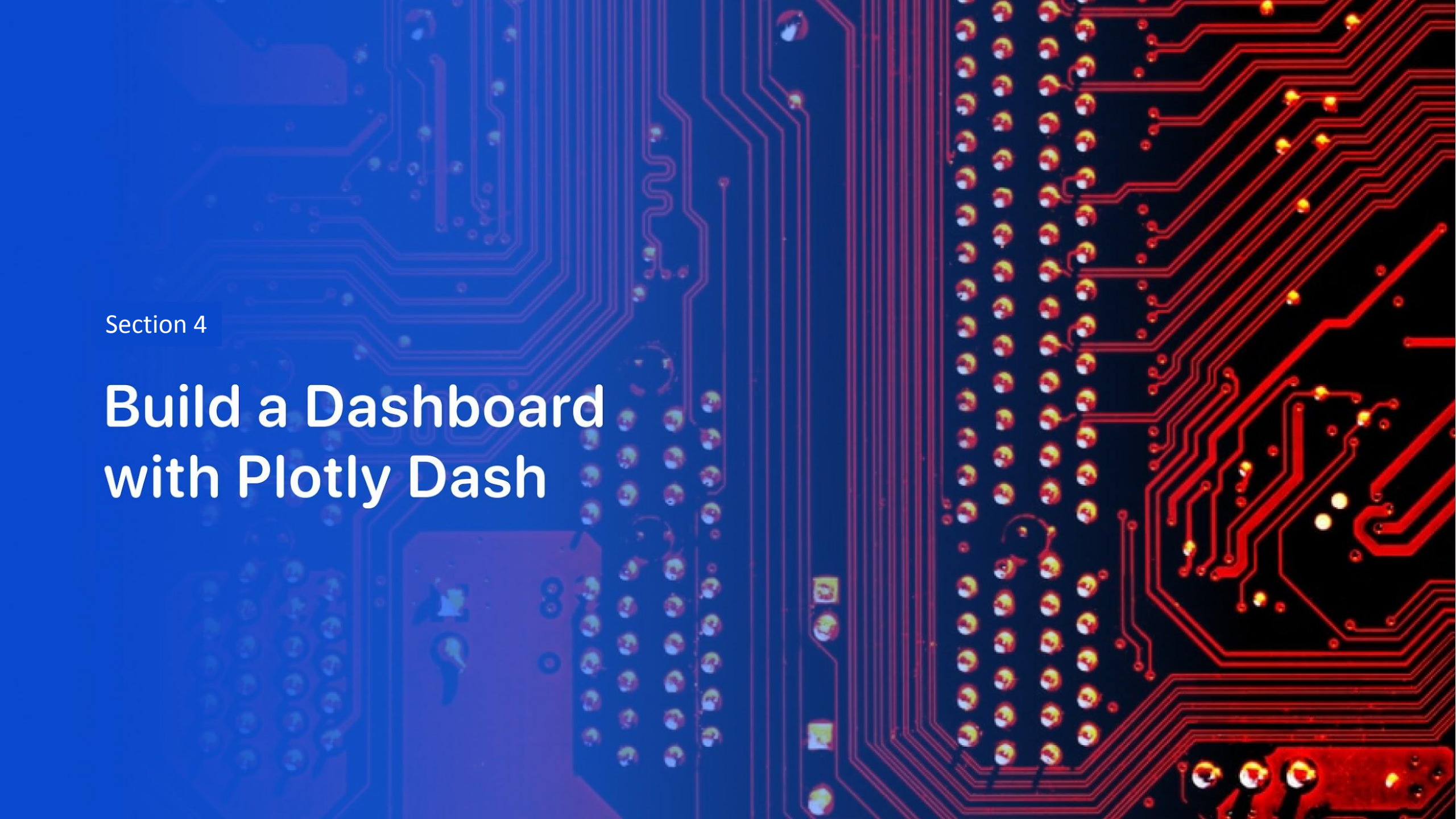
Green Markers show successful launches and Red Markers show failed launches.



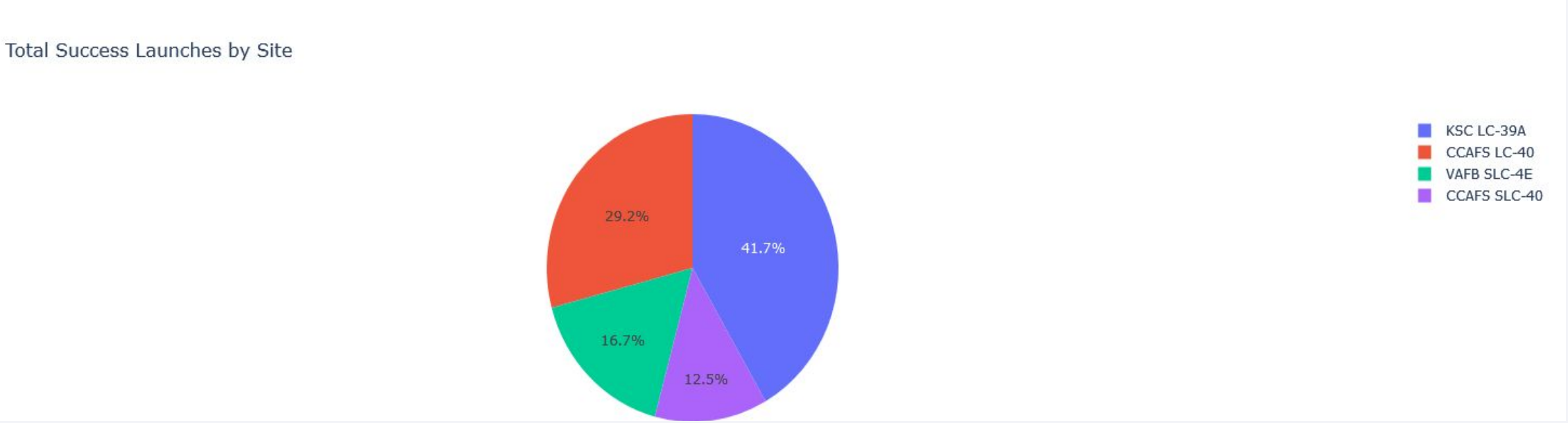
Launch Sites Distance to Landmarks



CCAFS SLC-40 is 0.9 KM from sea.


Section 4
Build a Dashboard with Plotly Dash

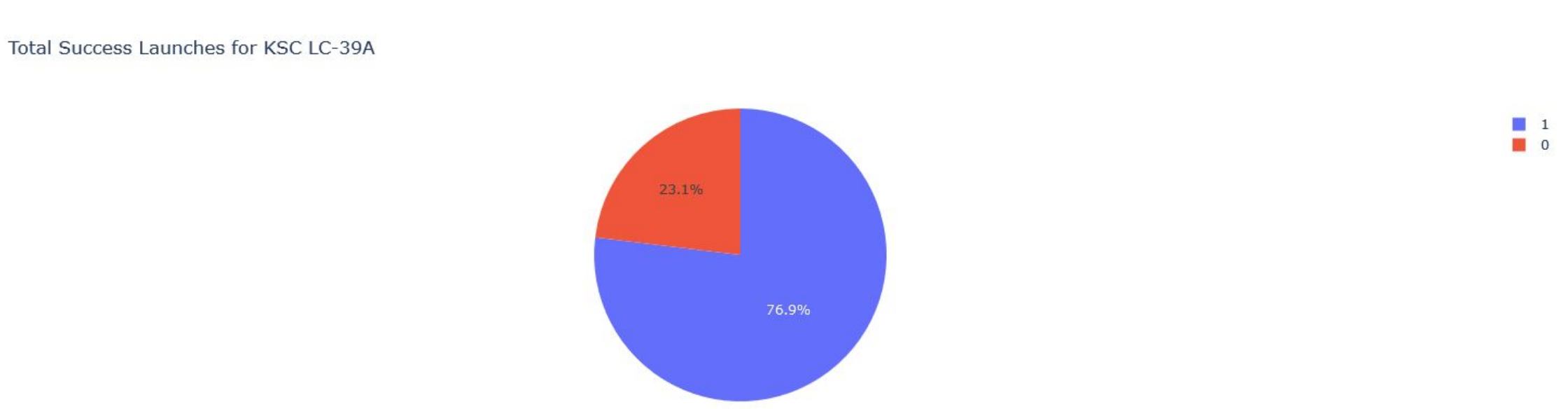
Total Success Launches for All Sites



Total Success Launches for All Sites is

- KSC LC-39A=41.7%
- CCFAS LC-40=29.2%
- VAFB SLC-4E=16.7%
- CCAFS SLC-40=12.5%

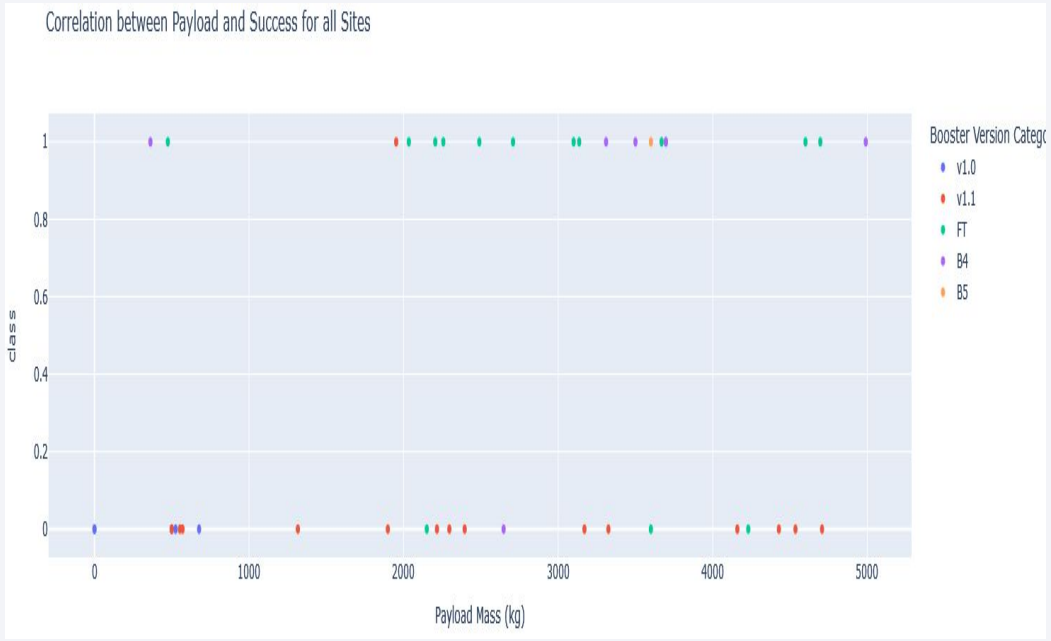
Success Ratio for KSC LC-39A

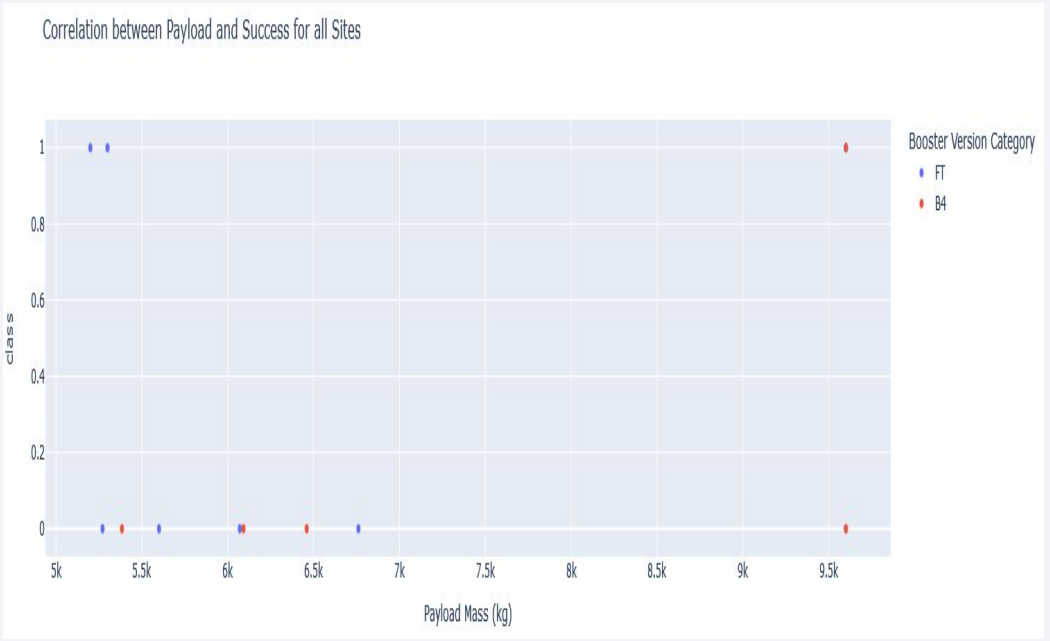


Success Rate = 76.9%

Failure Rate = 23.1%

Correlation Between Payload and Success





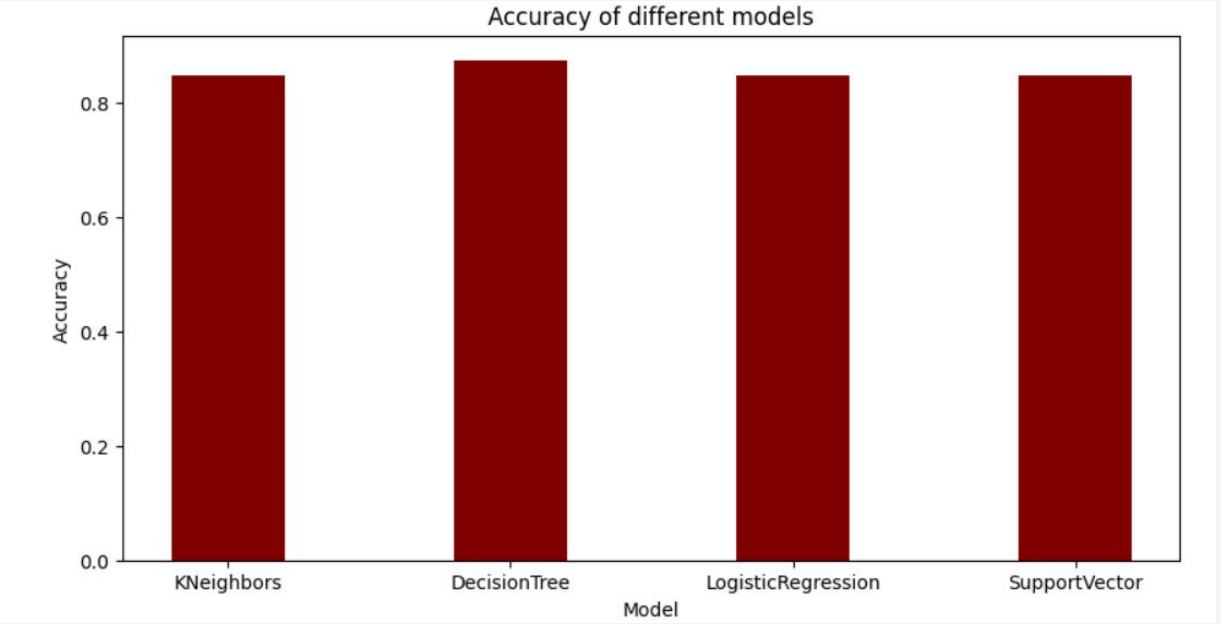
We can see that all the success rate for low weighted payload is higher than heavy weighted payload.

Section 5

Predictive Analysis (Classification)

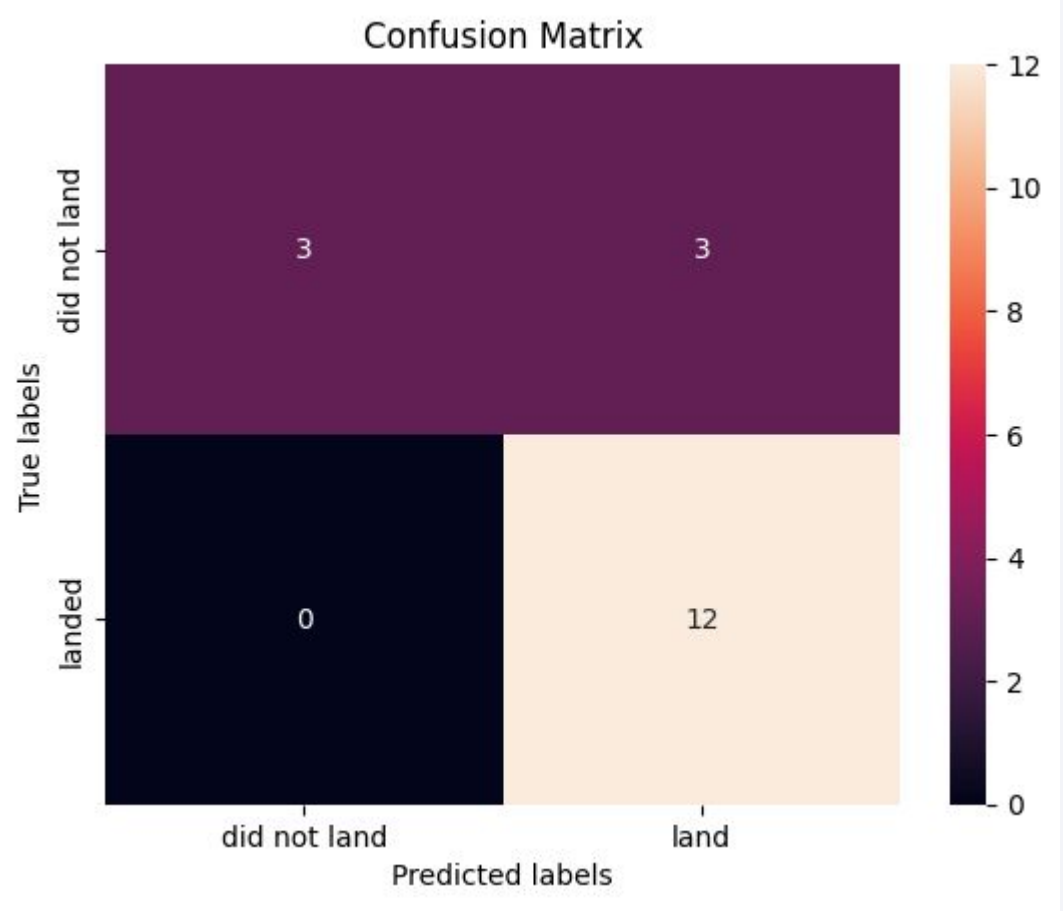
Classification Accuracy

DecisionTree has the highest classification accuracy of 87.32%.



Confusion Matrix

The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

- The Decision Tree Classifier Algorithm is the best Machine Learning approach for this dataset.
- The low weighted payloads (which define as 5000kg and below) performed better than the heavy weighted payloads.
- Starting from the year 2013, the success rate for SpaceX launches is increased, directly proportional time in years to 2020, which it will eventually perfect the launches in the future.
- KSC LC-39A have the most successful launches of any sites which is 76.9%.
- ES-L1, GEO, HEO, and SSO have the most success rate; 100% and more than 1 occurrence.

Appendix

Relevant Links

1. My Github Repository- https://github.com/palanivigneshwar/Applied-Data-Science-Capstone/tree/main
2. FlowCharts were made by- https://app.diagrams.net/

Thank you!