IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Abrar Hussain
02/10/2024

Outline

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

Executive Summary

Summary of methodologies:

- Data Collection through API
- Data Collection through Web Scraping
- Data Wrangling
- Exploratory Data Analysis using SQL
- Exploratory Data Analysis using Pandas and Numpy
- Interactive Visual Analysis using Folium and Dash
- Machine Learning Prediction

Summary of all results:

- EDA Results
- Folium and Dashboard Results
- Predictive Analysis Results

Introduction

- In this project Spacex Launch Data is extracted using Spacex API and Web Scraping Wikepedia Articles Tables. The main aim of the project is to find whethar a first stage rocket of a Falcon 9 rocket can land successfully. A company like SPACEX can save a lot of money if it is able to reuse the first stage of the rocket.So predicting whether a first stage can land or not can be really helpful. The projects starts with using API, Webscraping for data collection, then using SQL and python visualization tools for EDA. Then creating a interactive application using Dash and plotly for better analysis. Then machine learning is applied to train the model and check for accuracy.
- Problems you want to find answers: The problem we are trying to answer to find out whether a first stage of a rocket will safely land or not.

Section 1

Methodology

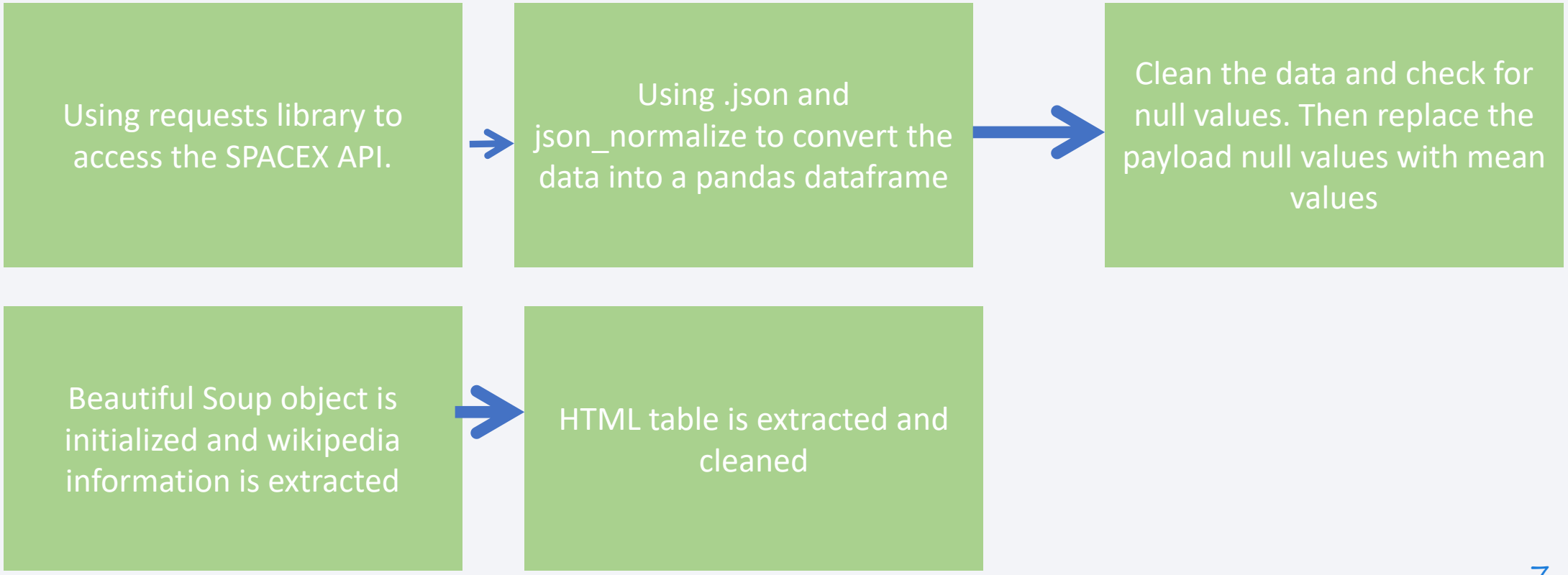
Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SPACEX API and Webscraping.
- Perform data wrangling
  - One Hot Encoding was used
- 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

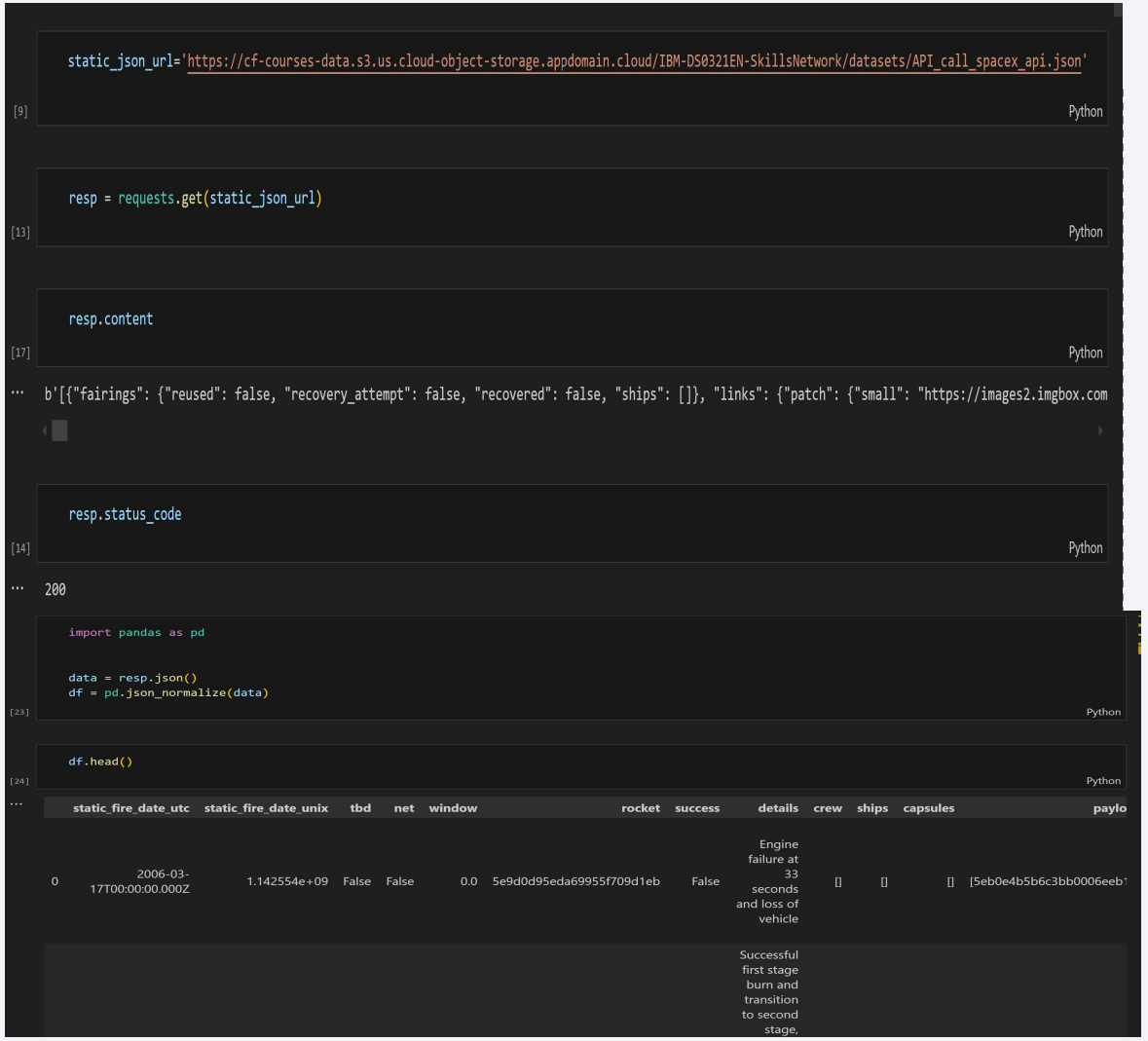
- The data is collected using SPACEX API and Web Scraping.



Data Collection – SpaceX API

- We used REST API to call SPACEX API to downloading data and ready it.

Link: https://github.com/AbrarHussain123/Spacex-FirstStage-Rocket-Landing-Prediction-IBM/blob/main/Data_Collection_Using_API.ipynb



Data Collection - Scraping

- Beautiful Soup library is used for Web Scraping

- Link:https://github.com/AbrarHussain123/Spacex-FirstStage-Rocket-Landing-Prediction-IBM/blob/main/Data_Collection_Using_WebScraping.ipynb

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [6]:
```
response = requests.get(static_url)
```

In [7]:
```
soup = BeautifulSoup(response.text,"html.parser")
```

In [26]:
```
soup.title
```

Out[26]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

In [8]:
```
soup.title.string
```

Out[8]:
```
'List of Falcon 9 and Falcon Heavy launches - Wikipedia'
```

In [9]:
```
html_tables = soup.find_all('table')
```

In [11]:
```
len(html_tables)
```

Out[11]:
```
26
```

In [12]:
```
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
<table class="wikitable plainrowheaders collapsible" style="width: 100%;">
<tbody><tr>
<th scope="col">Flight No.
</th>
<th scope="col">Date and<br/>time (<a href="/wiki/Coordinated_Universal_Time" title="Coordinated Universal Time">UTC</a>)
</th>
<th scope="col"><a href="/wiki/List_of_Falcon_9_first-stage_boosters" title="List of Falcon 9 first-stage boosters">Version,<
br/>Booster</a> <sup class="reference" id="cite_ref-booster_11-0"><a href="#cite_note-booster-11"><span class="cite-bracket">
```

Data Wrangling

- The following steps are part of the Data Wrangling process:

- The data is imported and then , checked for null values.
- Then different labels in class column is converted to 1 - Success and 0- Failure
- Then One Hot Encoding is applied to the data.

Link: https://github.com/AbrarHussain123/Spacex-FirstStage-Rocket-Landing-Prediction-IBM/blob/main/Data_Wrangling.ipynb

EDA with Data Visualization

- The plots plotted are : Catplot, Scatterplot, Barplot etc.

Link: https://github.com/AbrarHussain123/Spacex-FirstStage-Rocket-Landing-Prediction-IBM/blob/main/EDA_and_Visualization.ipynb

EDA with SQL

- The SQL Queris used are:
- Query using SUM()
- Query using MAX()
- Query using MIN()

Link: https://github.com/AbrarHussain123/Spacex-FirstStage-Rocket-Landing-Prediction-IBM/blob/main/EDA_using_SQL.ipynb

Build an Interactive Map with Folium

- Folium was used for creating maps. The lat and long values were made into a list and plotted. The labels were plotted along with the coordinates.

Link: https://github.com/AbrarHussain123/Spacex-FirstStage-Rocket-Landing-Prediction-IBM/blob/main/Map_Using_Folium.ipynb

Build a Dashboard with Plotly Dash

- Scatter point for different launch points was plotted. The graphs plotted Payload vs Class(0 or 1).

Link: https://github.com/AbrarHussain123/Spacex-FirstStage-Rocket-Landing-Prediction-IBM/blob/main/Dash_App.py

Predictive Analysis (Classification)

- The models used are Logistic Regression, SVC(), DecisionTree Classifier and KNN.

Link: https://github.com/AbrarHussain123/Spacex-FirstStage-Rocket-Landing-Prediction-IBM/blob/main/Model_Training.ipynb

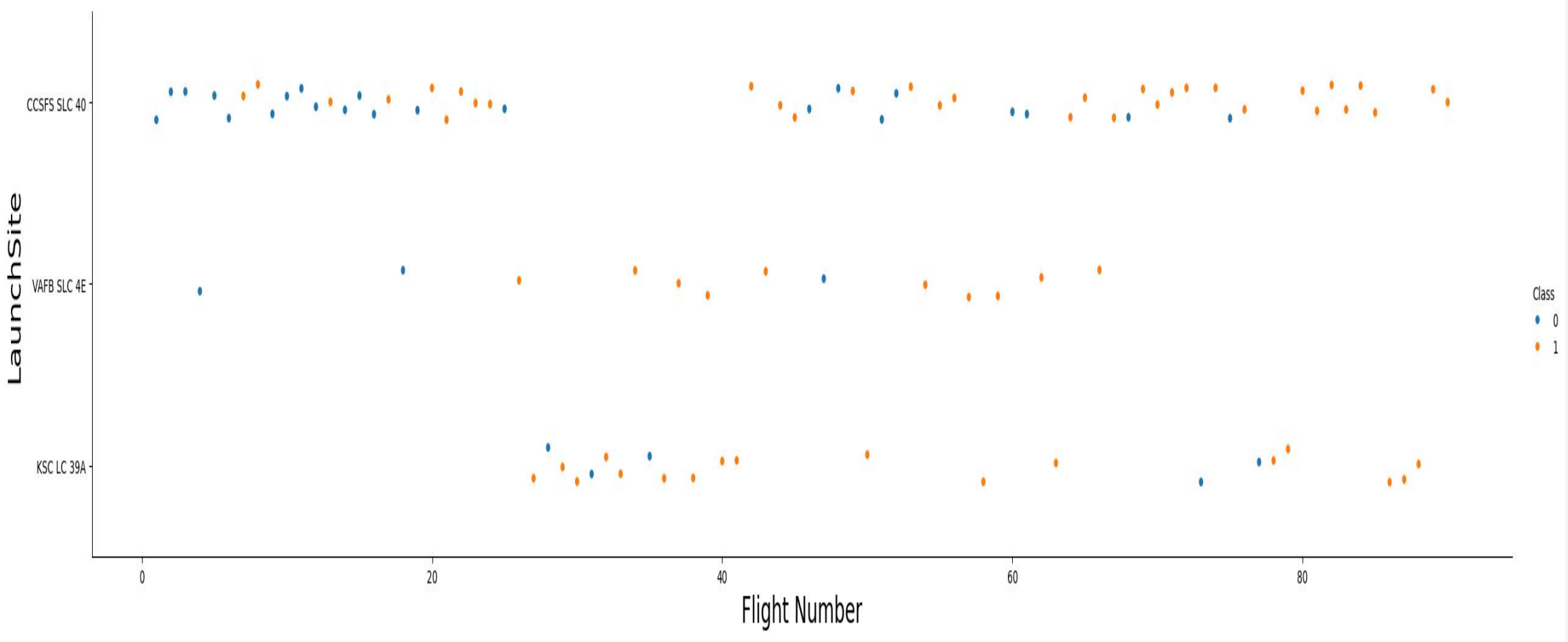
Results

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

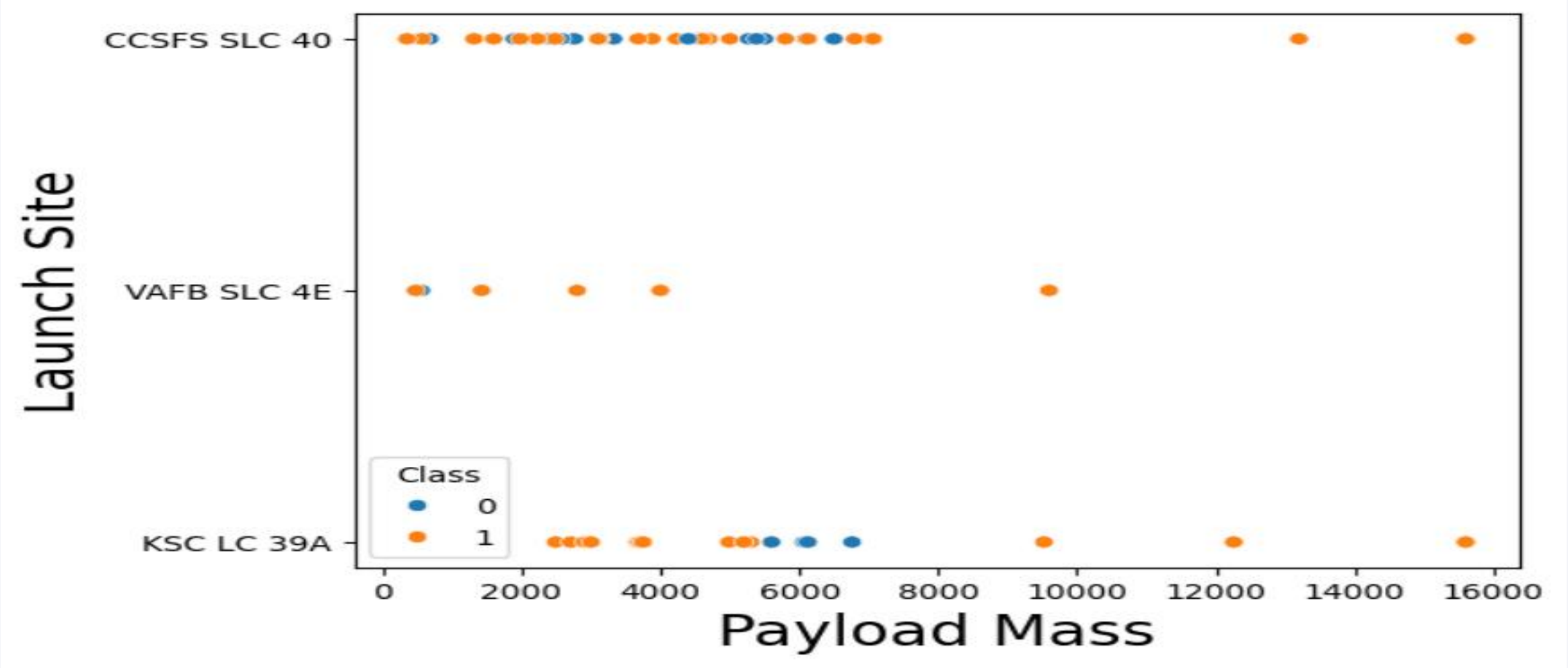
Section 2

Insights drawn from EDA

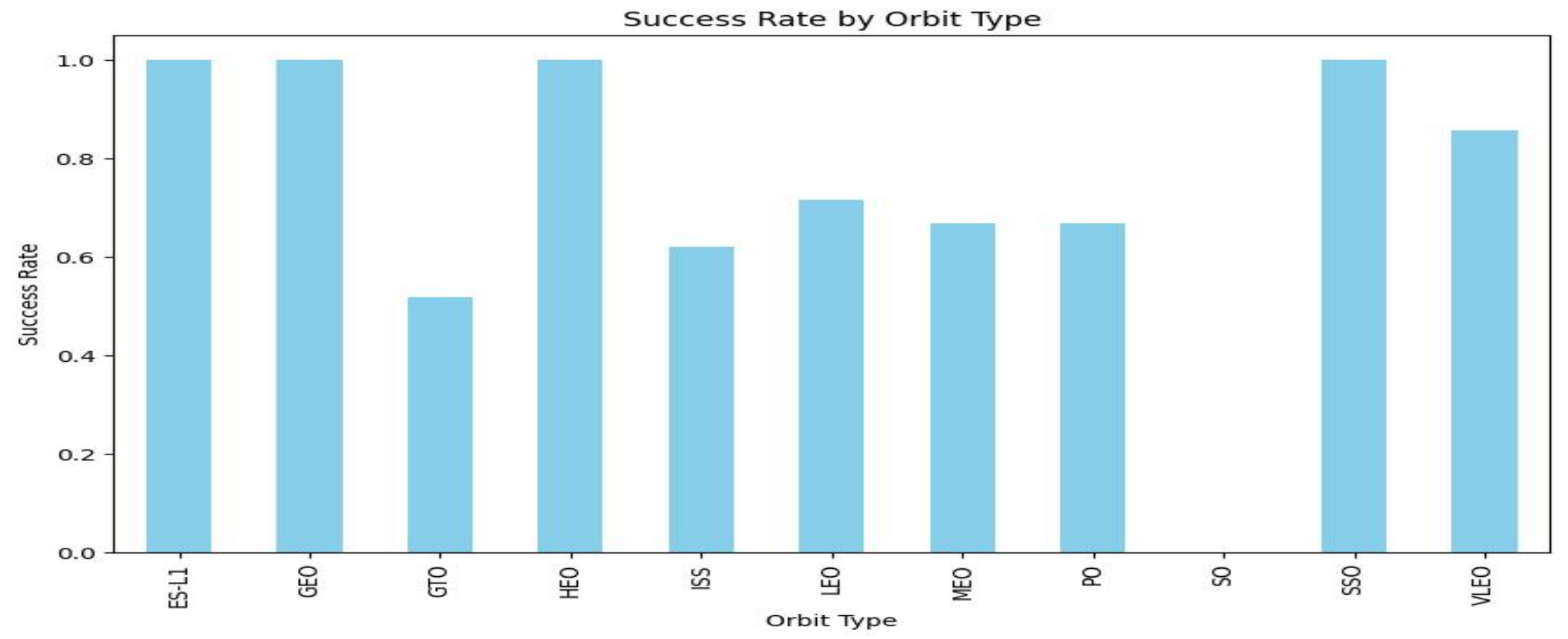
Flight Number vs. Launch Site



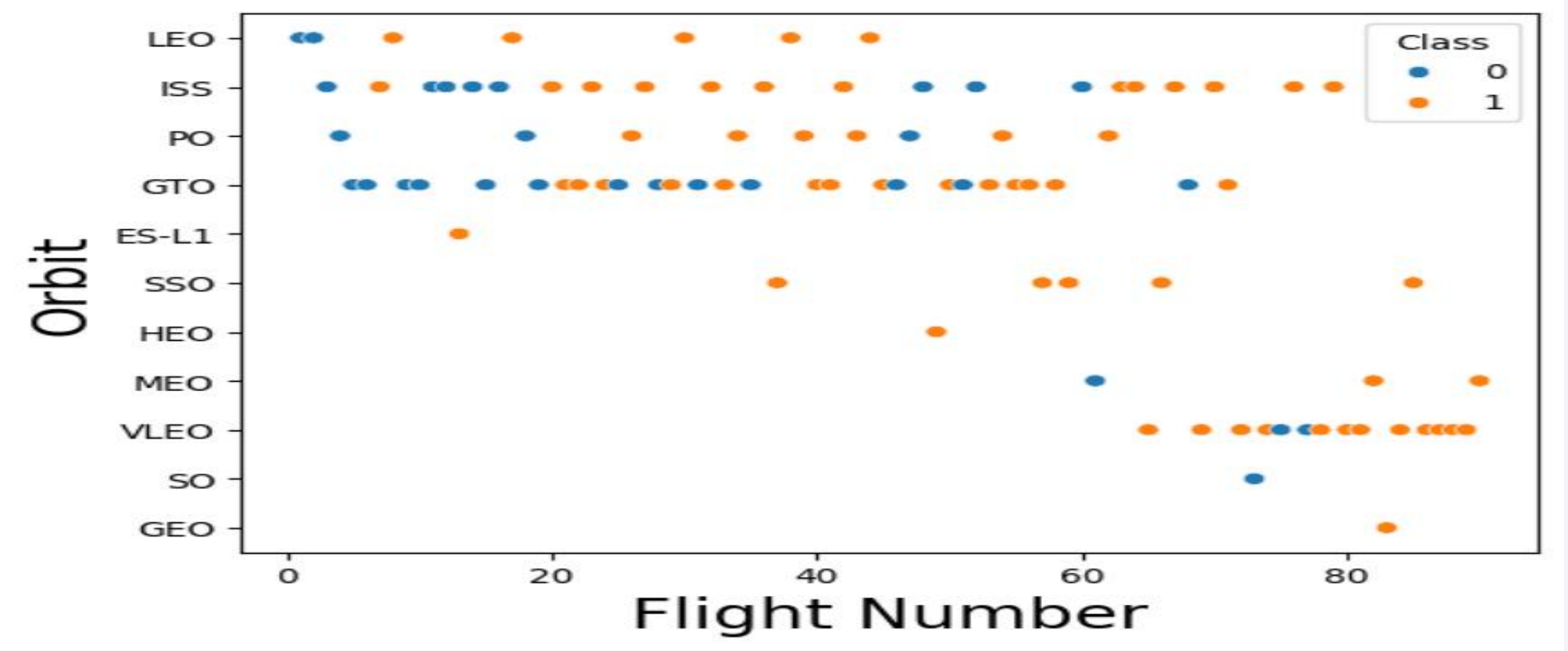
Payload vs. Launch Site



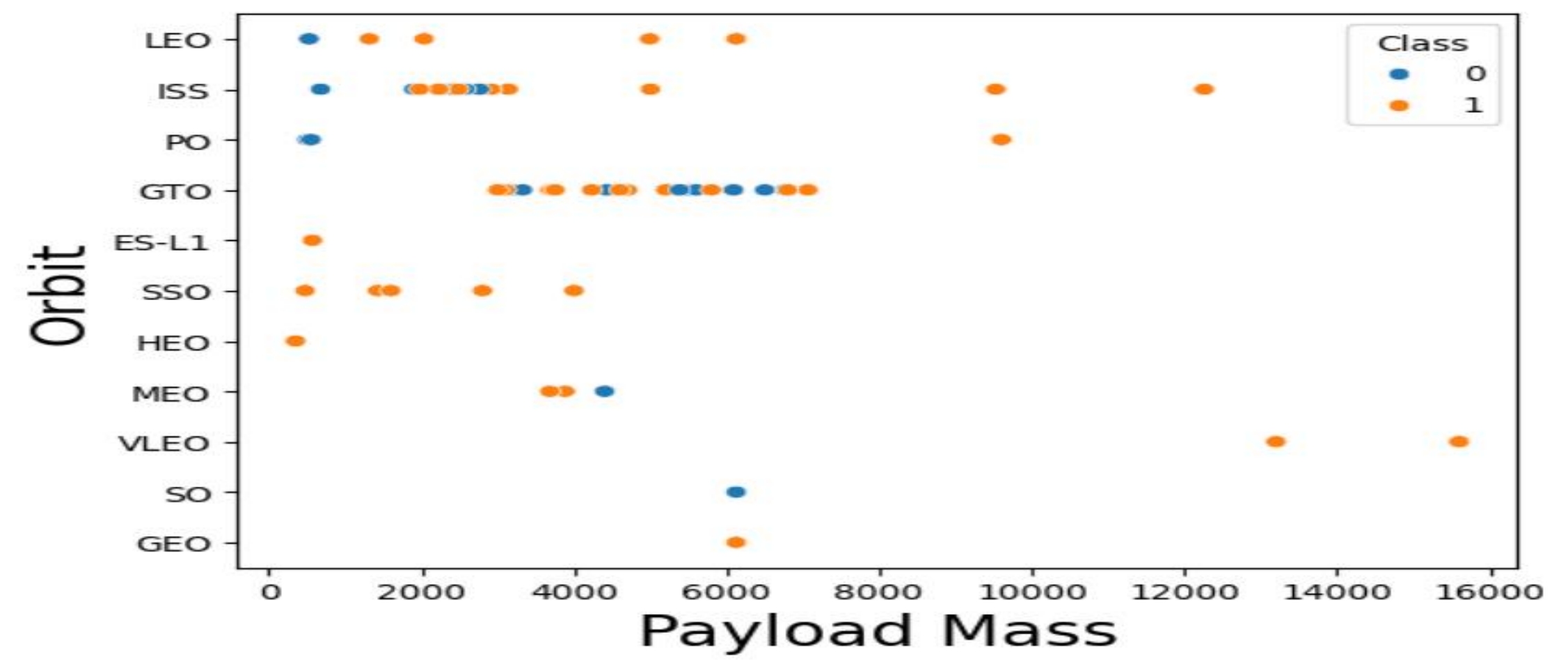
Success Rate vs. Orbit Type



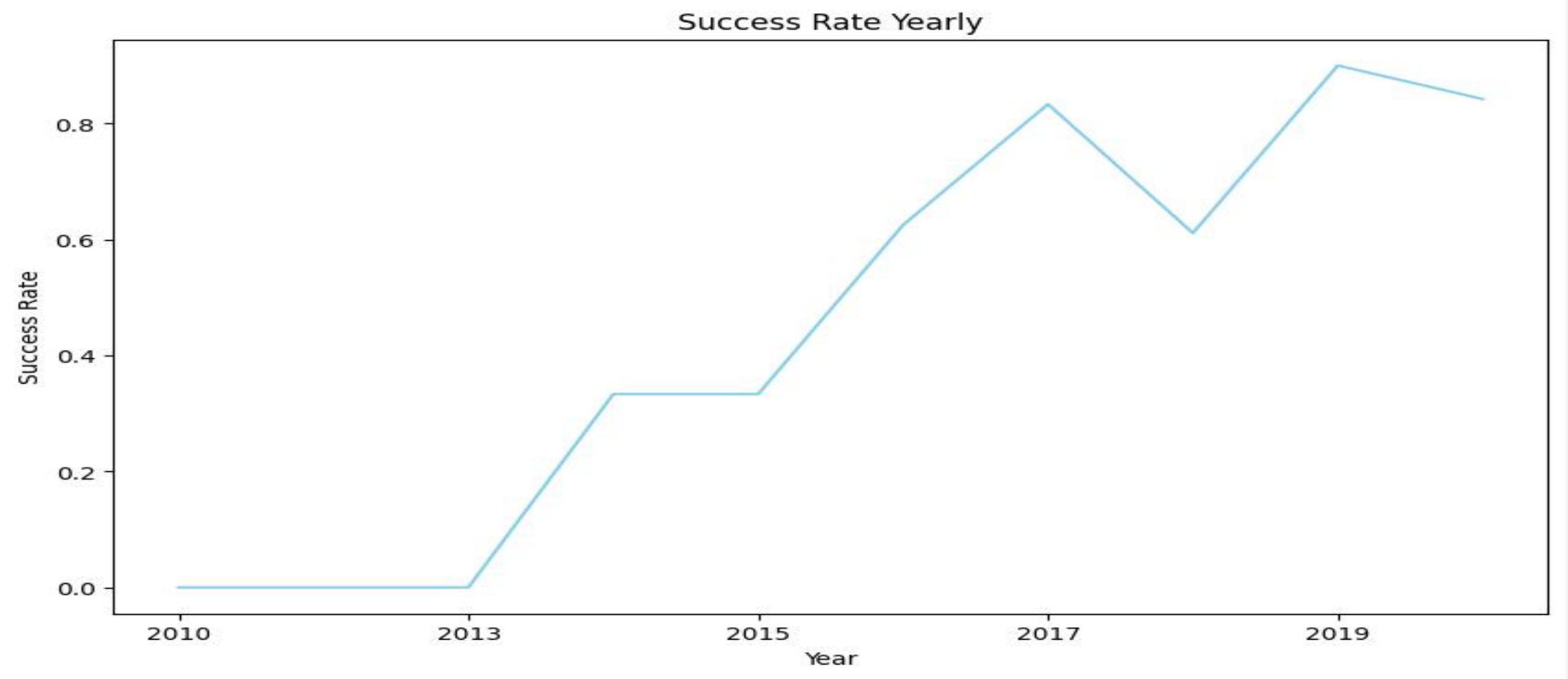
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

```
[19] %sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

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

[23] Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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 (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	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 SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

[24] Python

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

[28] Python

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

AVG(PAYLOAD_MASS_KG_)
2534.6666666666665

First Successful Ground Landing Date

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

[31] Python

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

MIN(Date)
2015-12-22

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
```

[32] Python

```
 * sqlite:///my_data.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 SELECT Mission_Outcome, COUNT(*) AS Total_Count FROM SPACEXTBL GROUP BY Mission_Outcome;
```

[33] Python

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

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

Boosters Carried Maximum Payload

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

[34] Python

```
 * sqlite:///my_data.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 SELECT Booster_Version,Launch_Site,substr(Date, 6,2) AS Month FROM SPACEXTBL WHERE substr(Date,0,5)='2015'
%sql AND Landing_Outcome = 'Failure (drone ship)'
```

[35] Python

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

Booster_Version	Launch_Site	Month
F9 v1.1 B1012	CCAFS LC-40	01
F9 v1.1 B1015	CCAFS LC-40	04

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

```
%%sql SELECT Landing_Outcome,COUNT(*) AS Outcome_Count
FROM
    SPACEXTBL
WHERE
    Date >= '2010-06-04'
    AND Date <= '2017-03-20'
GROUP BY
    Landing_Outcome
ORDER BY
    Outcome_Count DESC;
```

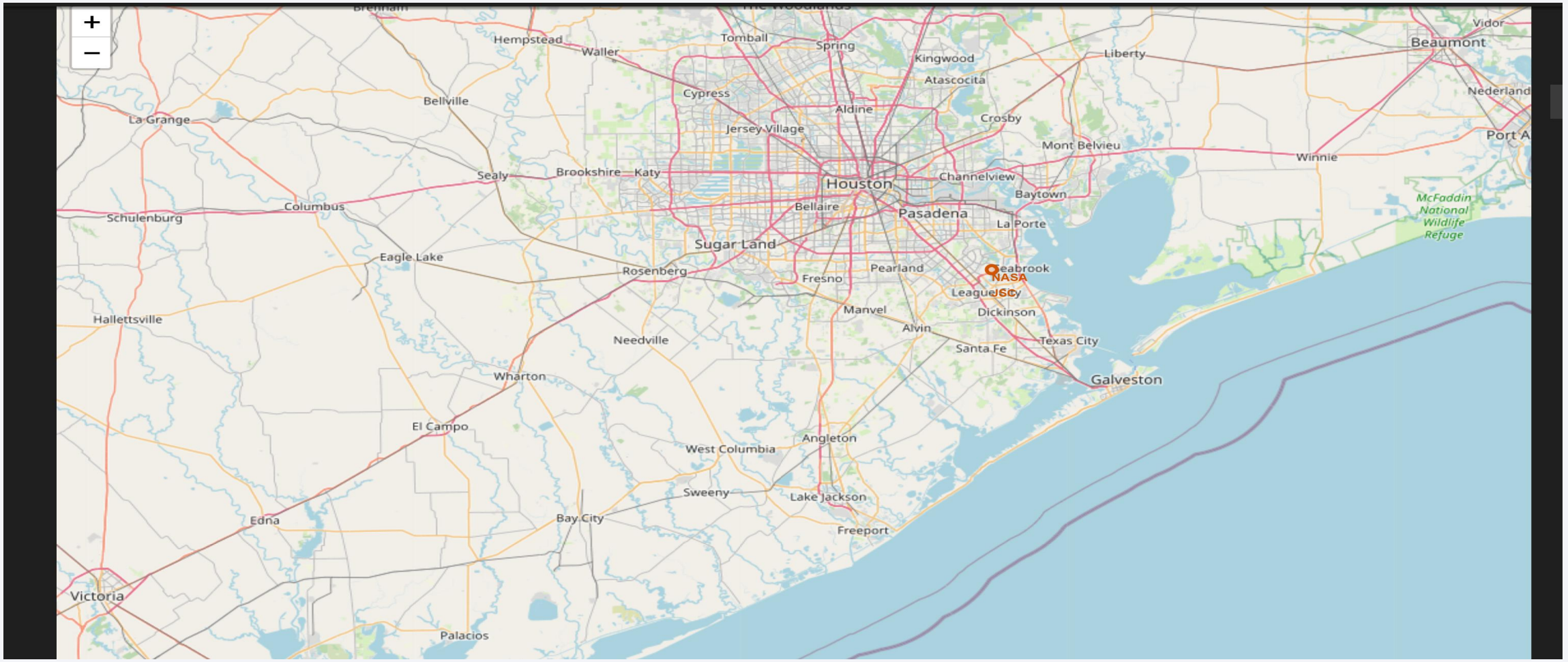
[36] Python

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

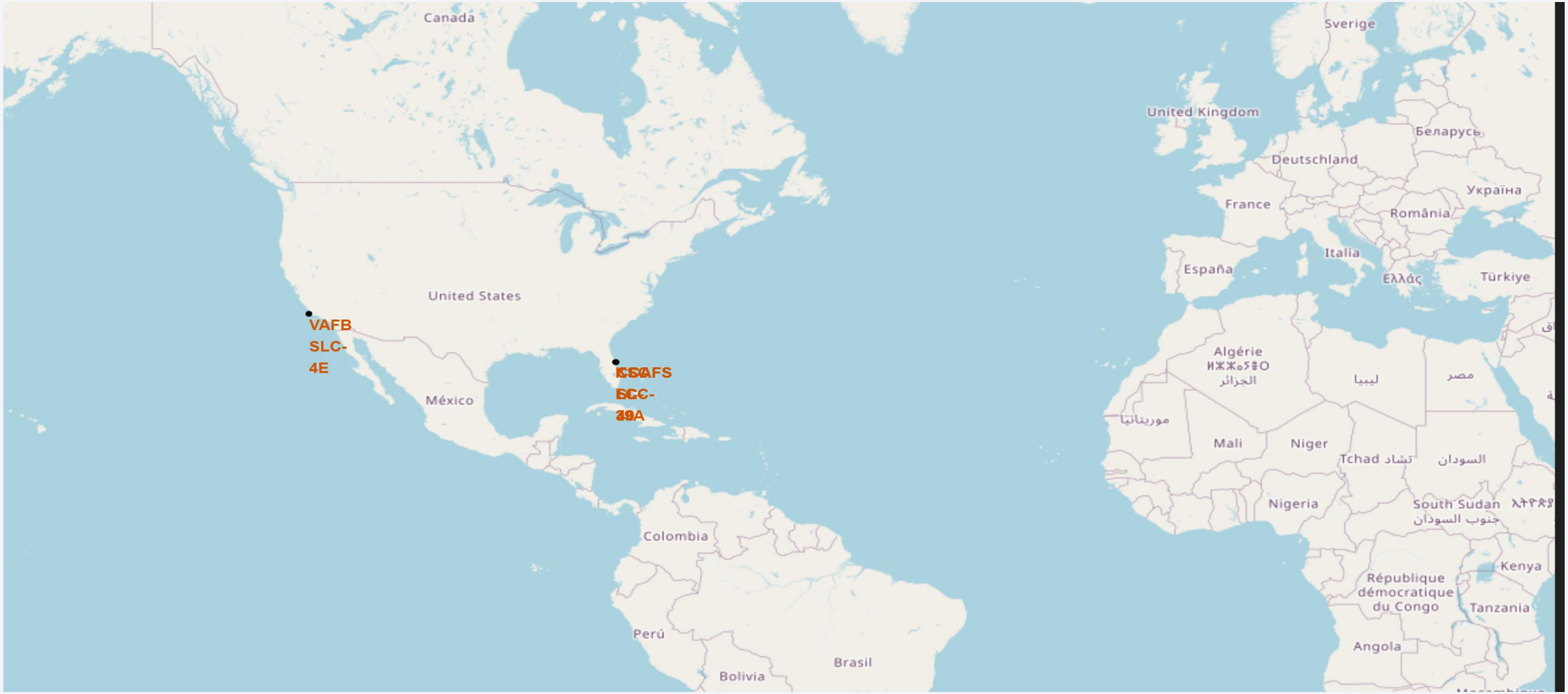
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1


Section 3
Launch Sites
Proximities Analysis

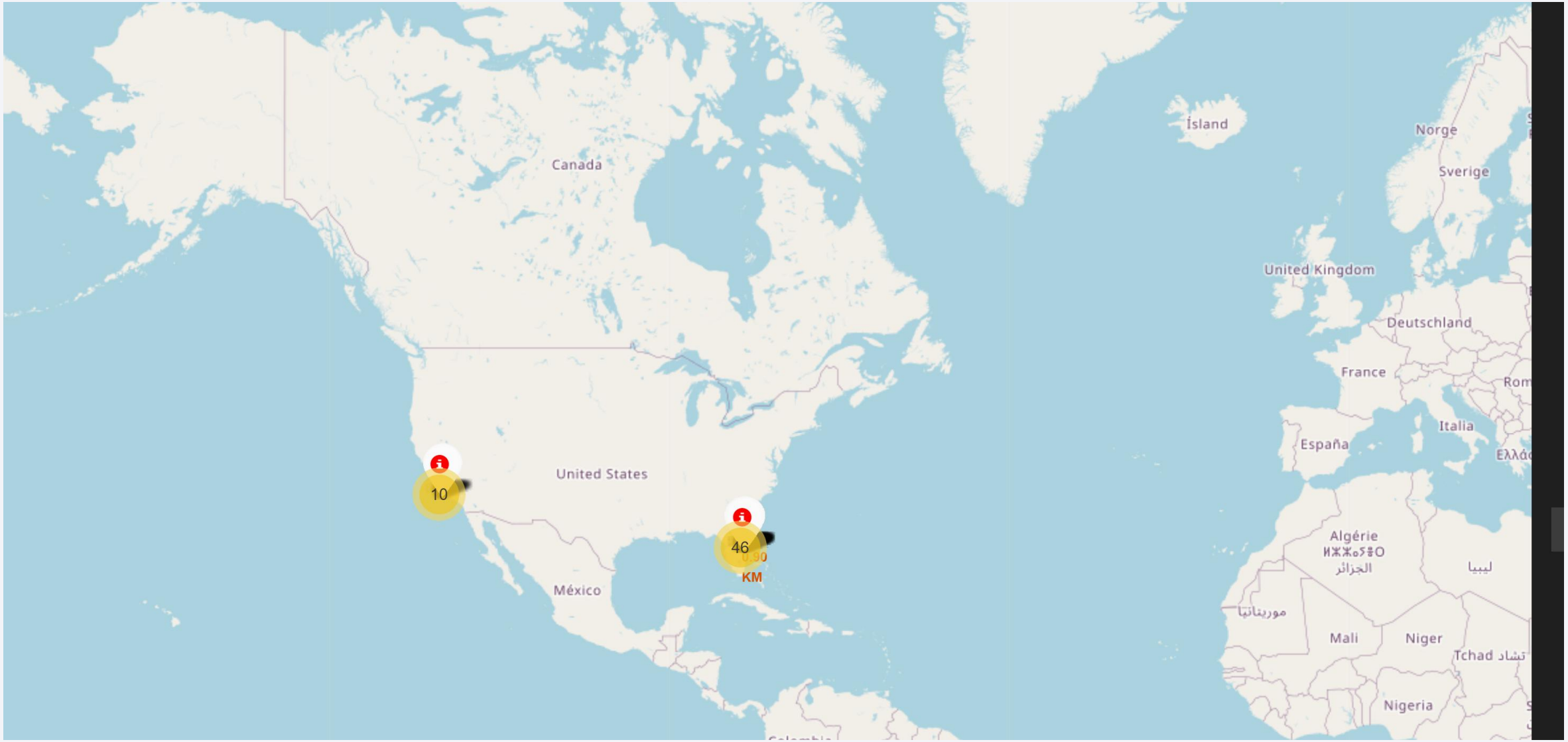
NASA JOHNSON SPACE CENTER IN HOUSTON

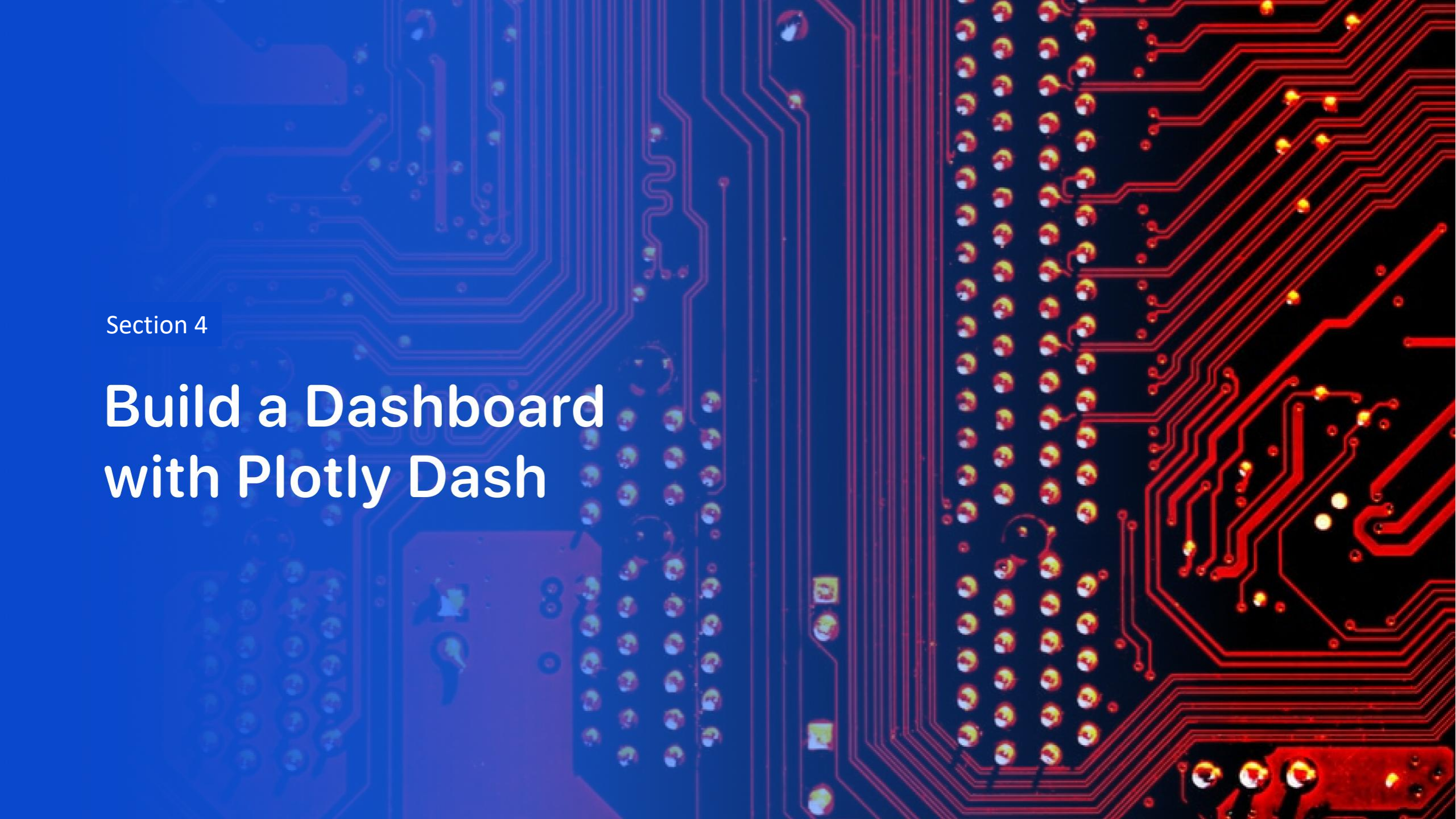


DIFFERENT LAUNCH SITES FOR SPACEX

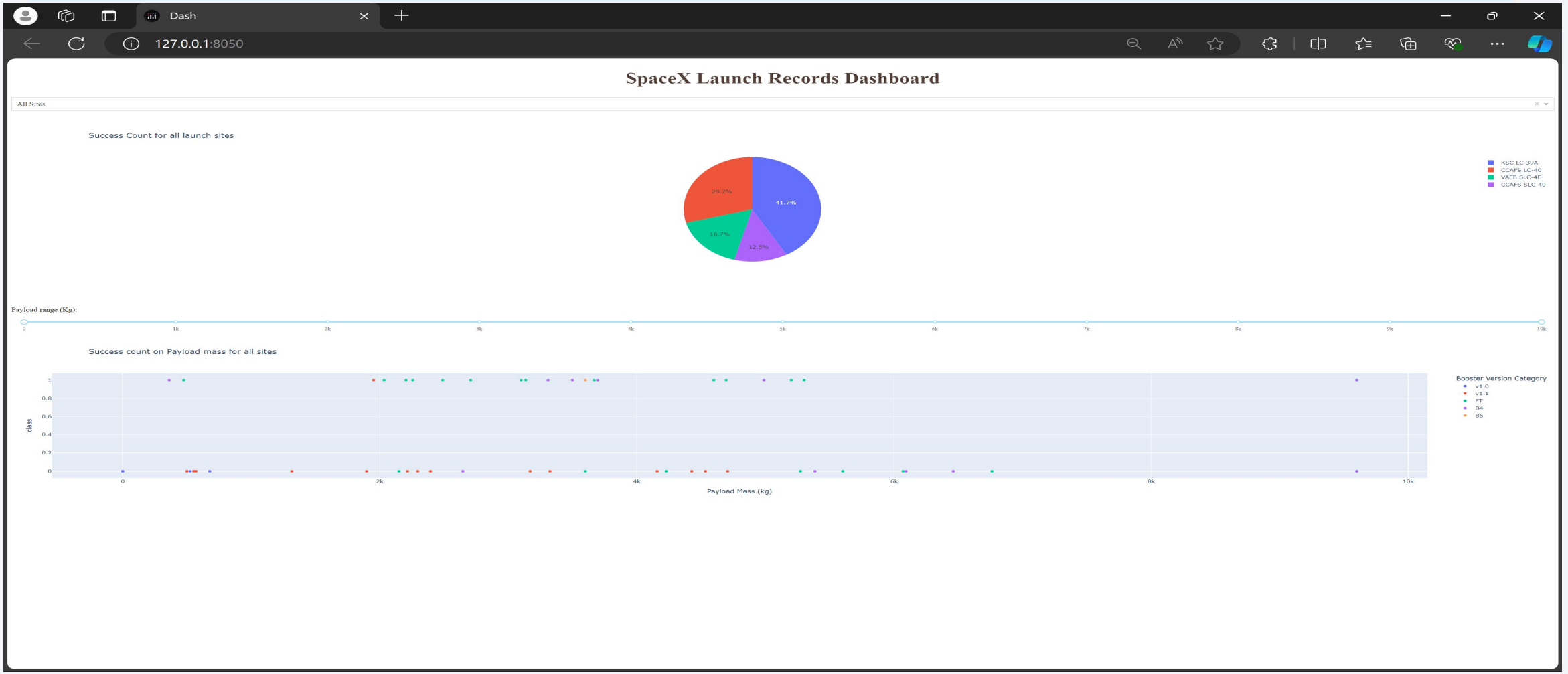


DIFFERENT LAUNCH SITES AND THE NUMBER OF LAUNCHES(SUCCESS AND FAILURE).

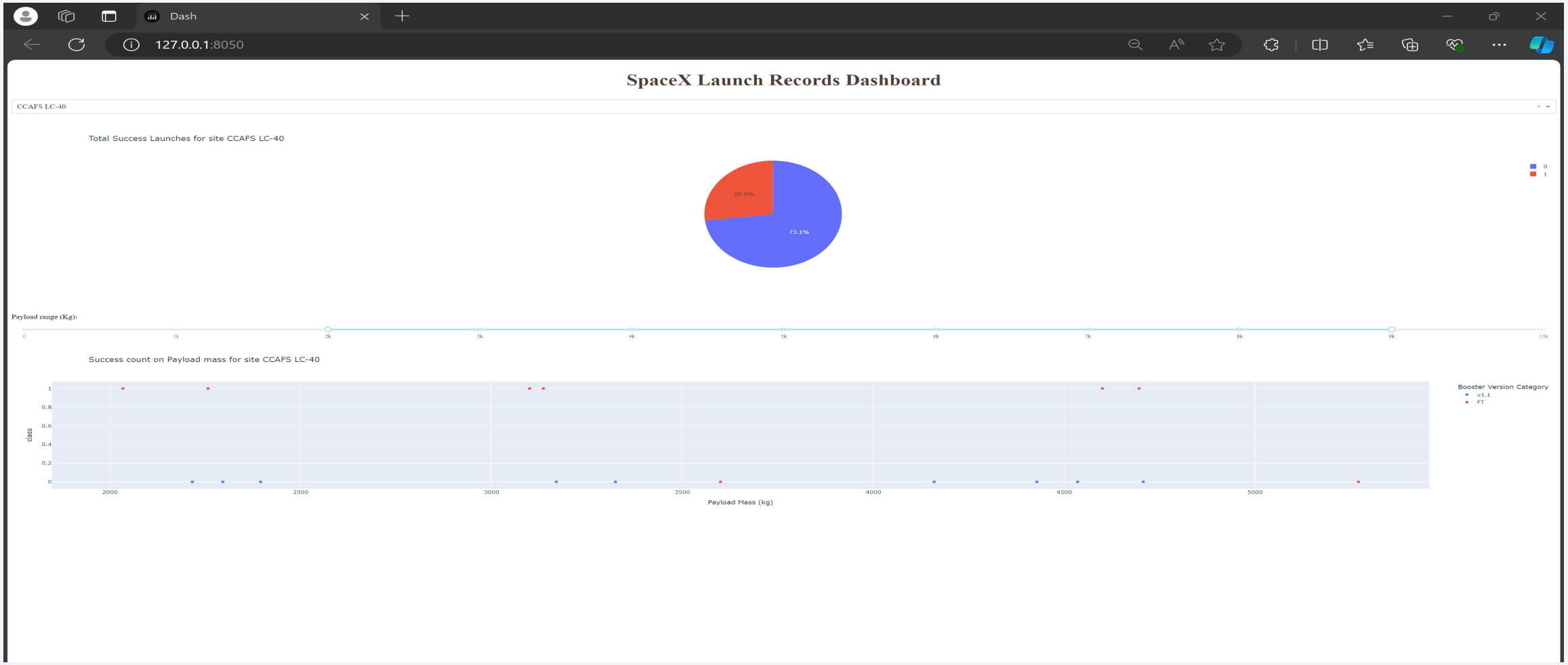



Section 4
Build a Dashboard with Plotly Dash

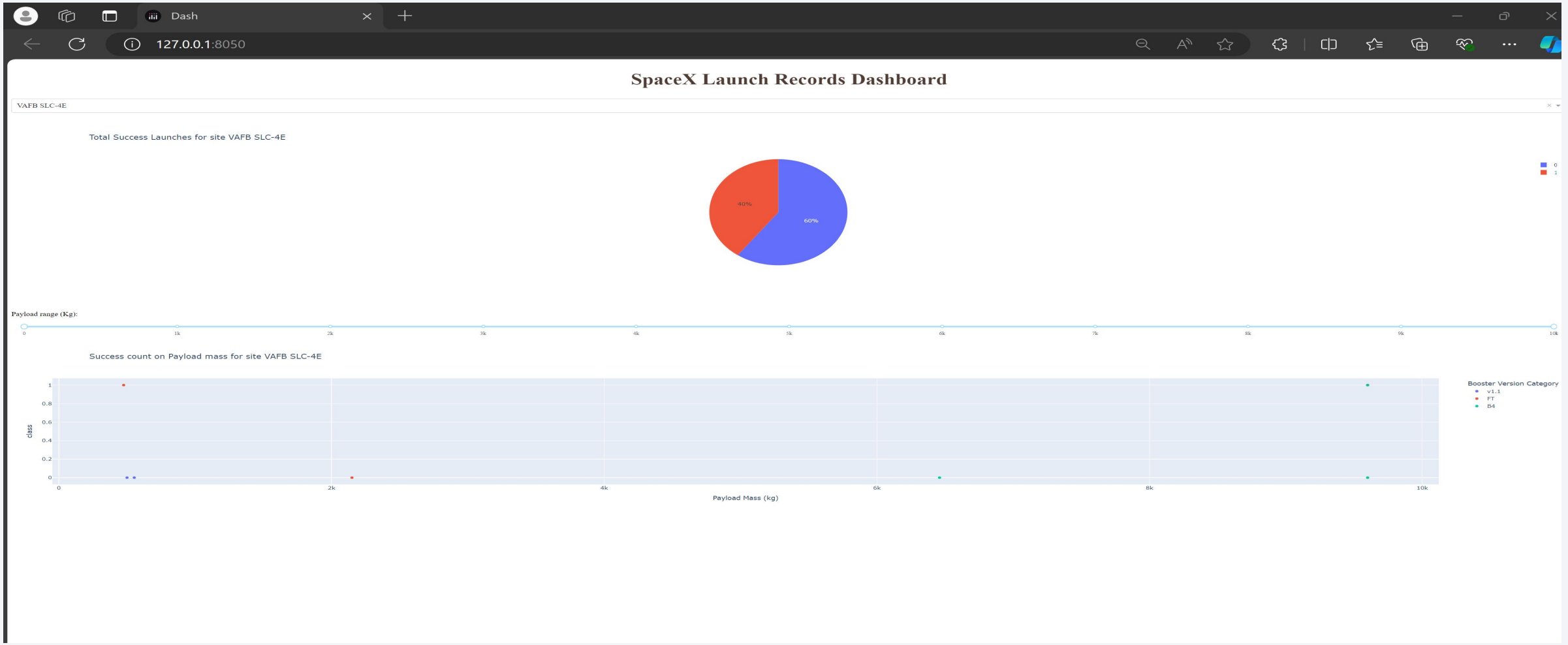
Dashboard - All Sites



Dashboard - CCAFS LC-40 AND PAYLOAD > 2K AND <9K



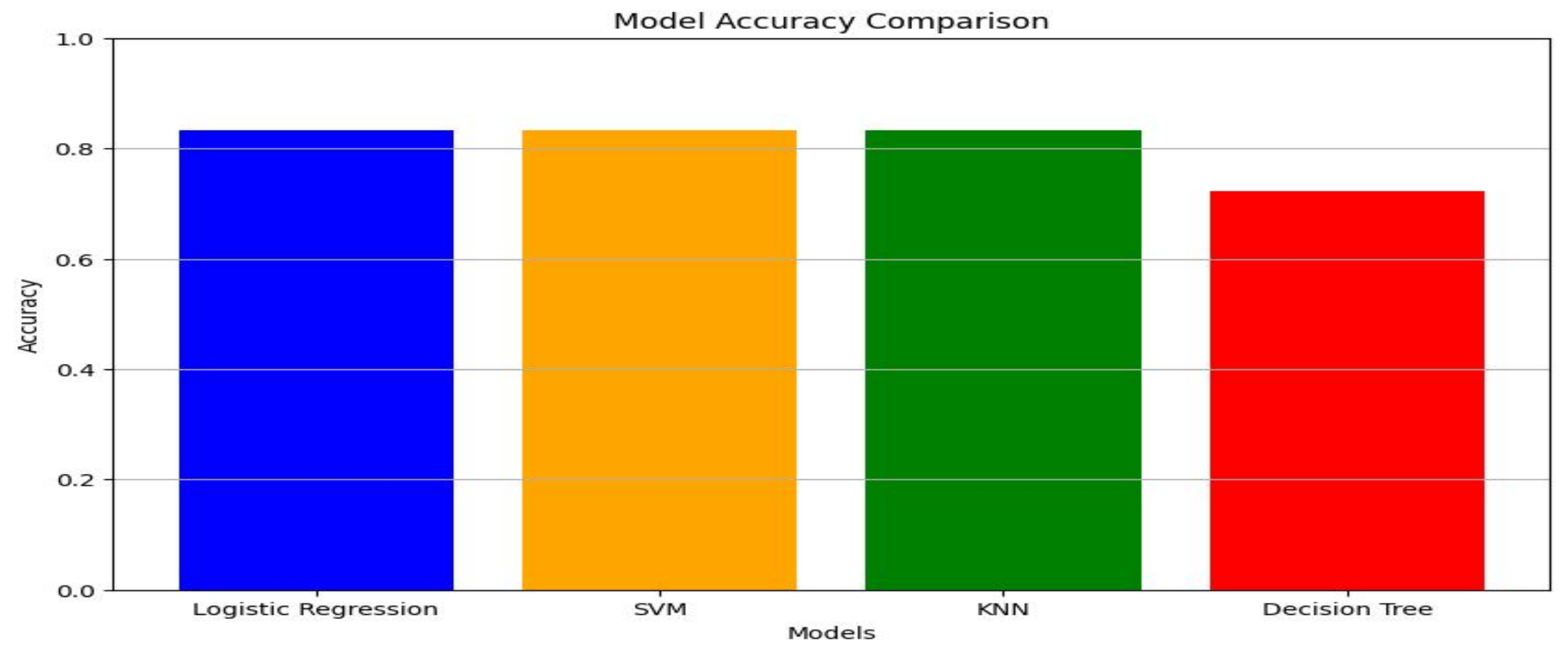
Dashboard VAFB SLC-4E



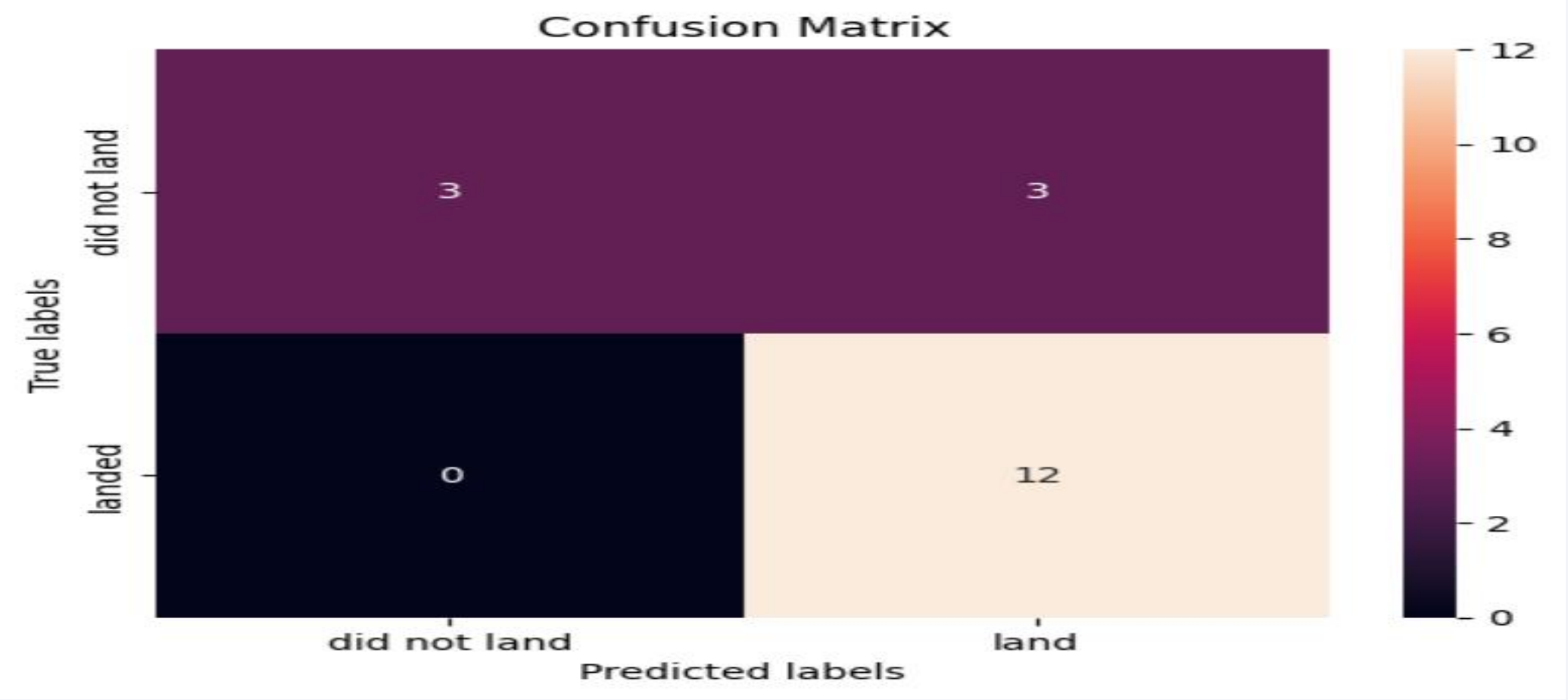
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- Logistic Regression,KNN and SVM gave similar accuracies.

Thank you!