IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

MANISH N
06 August 2024

Outline

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

Executive Summary

- **Summary of methodologies**
- Data collection
- Data Wrangling
- EDA Data visualization.
- EDA with SQL .
- Predictive Analysis
- Interactive Map With Folium
- Interactive Dashboard Using Plotly

Results Summary

- Exploratory Data Analysis
- Predictive Analysis results

Introduction

INTRODUCTION

- **Background : SpaceX a rocket company launches satellites at low price like 70% less than their competitor since they land their satellites for reusing them to launch**

- **Problem : We use the previous data of launches of Falcon 9 rocket to predict the probability of the booster landing back to the pad influenced/correlated with the space launch site,the payload orbit , mass , landing pad location and the version of the booster**

Section 1

Methodology

Methodology

- Data Collection -API & Web Scraping.
- Data wrangling-Extracting Load & Transform .
- Cleaning data to values that we can use -example labels to dummy integers .
- EDA with visualization and Sql.
- Interactive with Folium and Plotly Dash .
- Predictive Analysis -using Machine Learning Models .

Data Collection

DATA COLLECTION

- REST API : Using the REST API we extract the data in form of JSON and transform it to a dataframe using inbuilt python pandas method normalize.

- WEB SCRAPING : Web scraping spacex launches from wikipedia and converting it into a dataframe.

Data Collection – SpaceX API

REST API

```
json_data=requests.get(static_json_url).json()
```

Make request

```
# Use json_normalize meethod to convert the json result in
data=pd.json_normalize(json_data)
```

Normalize to df

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
':LaunchSite
utcome,
lights,
GridFins,
used,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
Longitude': Longitude,
Latitude': Latitude}
```

Create a dictionary for creating a dataframe from the dataset collected

Filter Falcon 9 only

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9=df_launch[df_launch['BoosterVersion']!='Falcon 1']
```

Save to CSV

```
data_falcon9.to_csv('dataset_part_1.csv',index=False)
```

Data Collection - Scraping

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

Get content of the wiki

Loop Through and add column names

```
column_names = []
temp = soup.find_all('th')
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
```

Create a Dataframe for the important columns

Loop through the request content and extract data

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Save to cvs

Data Wrangling

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

EXPLORATORY DATAANALYSIS

Calculate the number of launches on each site

Calculate the number and occurrence of each orbit

Calculate the number and occurence of mission outcome per orbit type

Create a landing outcome label from Outcome column

Save to CSV

Calculate the % of Null Values

```
df.isnull().sum()/df.count()*100
```

EDA with Data Visualization

Through EDA on the data from APi and Wiki,we will find some insights on :

- **Flight number & Launch Sites-Visualizing the launch from every site .**
- **Payload & Launch Sites-Payload launch from sites**
- **Success rate & Orbit type-Success rate compared to the orbit type**
- **Flight number & Orbit Type -Type of orbit for each launch**
- **Payload & Orbit type -Payload and the orbit .**
- **Trend of success rate-Trend of the success rate over the years .**

EDA with SQL

- **Exploratory Data Analysis on the follow criteria:**
- **Unique Sites**
- **Max Payload**
- **Average Payload**
- **Day when First Success Landing**
- **Success and Failures count**
- **Boosters With Max Payload**

Build an Interactive Map with Folium

Visualization of the launches for every site and every launch in a Interactive Map

- *Visualization Of :*

❖ Launch Sites

❖ Visualize the launches on the map base on Fail or Success

Build a Dashboard with Plotly Dash

INTERACTIVE WITH DASH

- **Visualization of the Launches from Site in Dashboard**

- ***Visualization of:***

❖ **Success Launch Launch Sites**

❖ **Visualize payload from different sites with rangeSlider for interacting with the plot .**

Predictive Analysis (Classification)

- Through Models , tuned for best performance we go the insights on the probability if a launch being success or a failure .

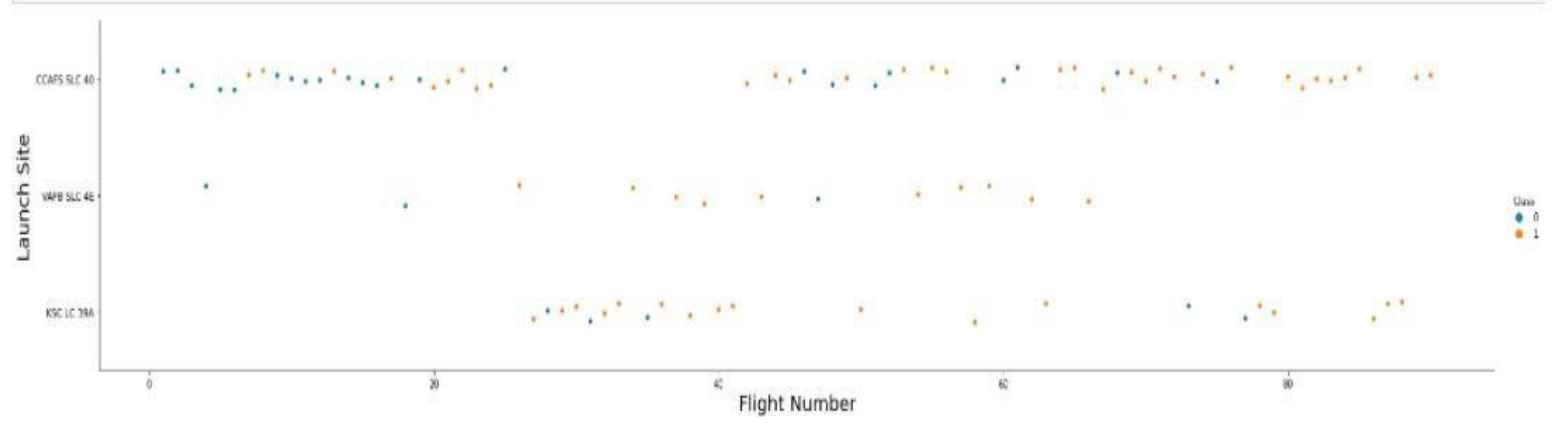
Models used include:

- KNeighboursClassfier
- Decision Tree
- Logistic Regression
- Support Vector Machine

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



From the Visualization we can concluded that:

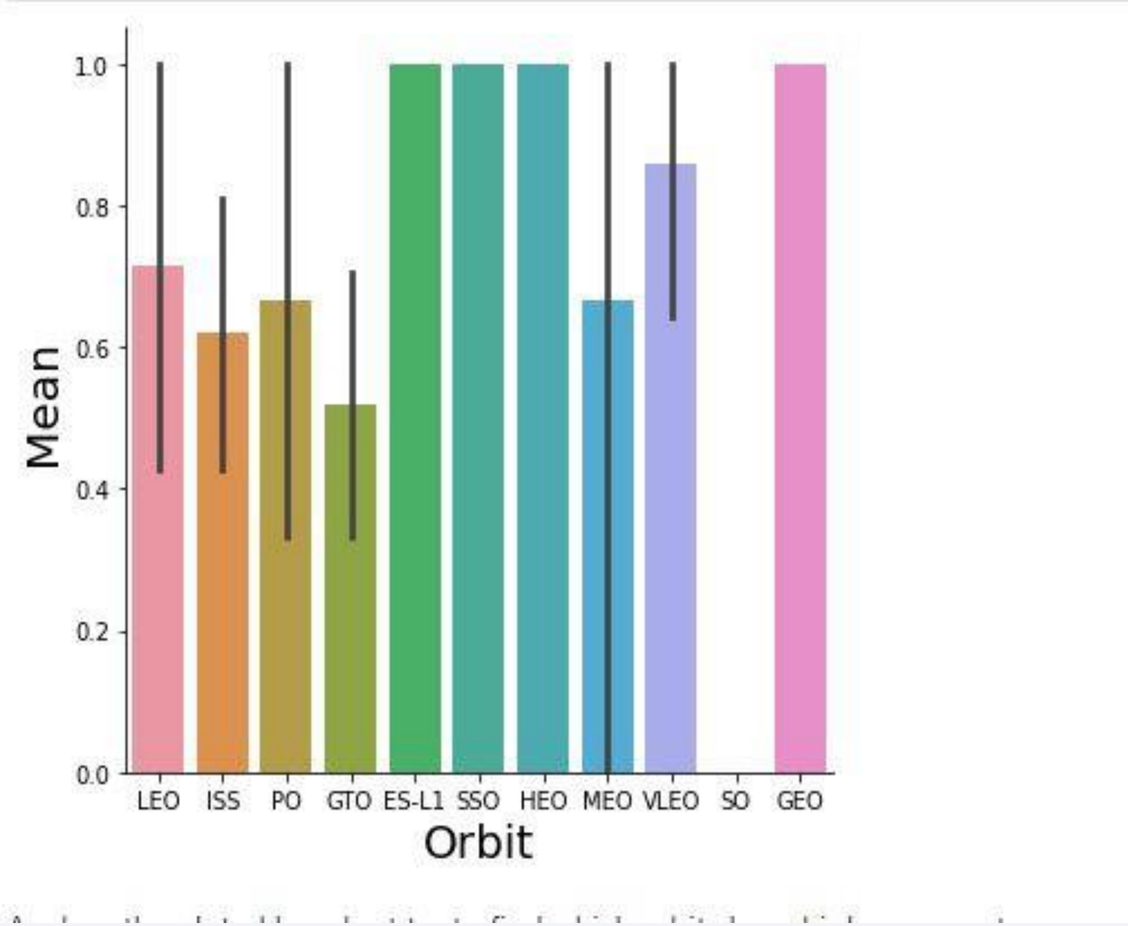
- **VAFB SLC 4E has Low Payload launches**
- **CCAFS SLC 40 has more Higher Payload Launches and Low Payload Lauches** .

Payload vs. Launch Site

From the Visualization we can concluded that:

- **VAFB SLC 4E has Low Payload launches**
- **CCAFS SLC 40 has more Higher Payload Launches and Low Payload Lauches** .

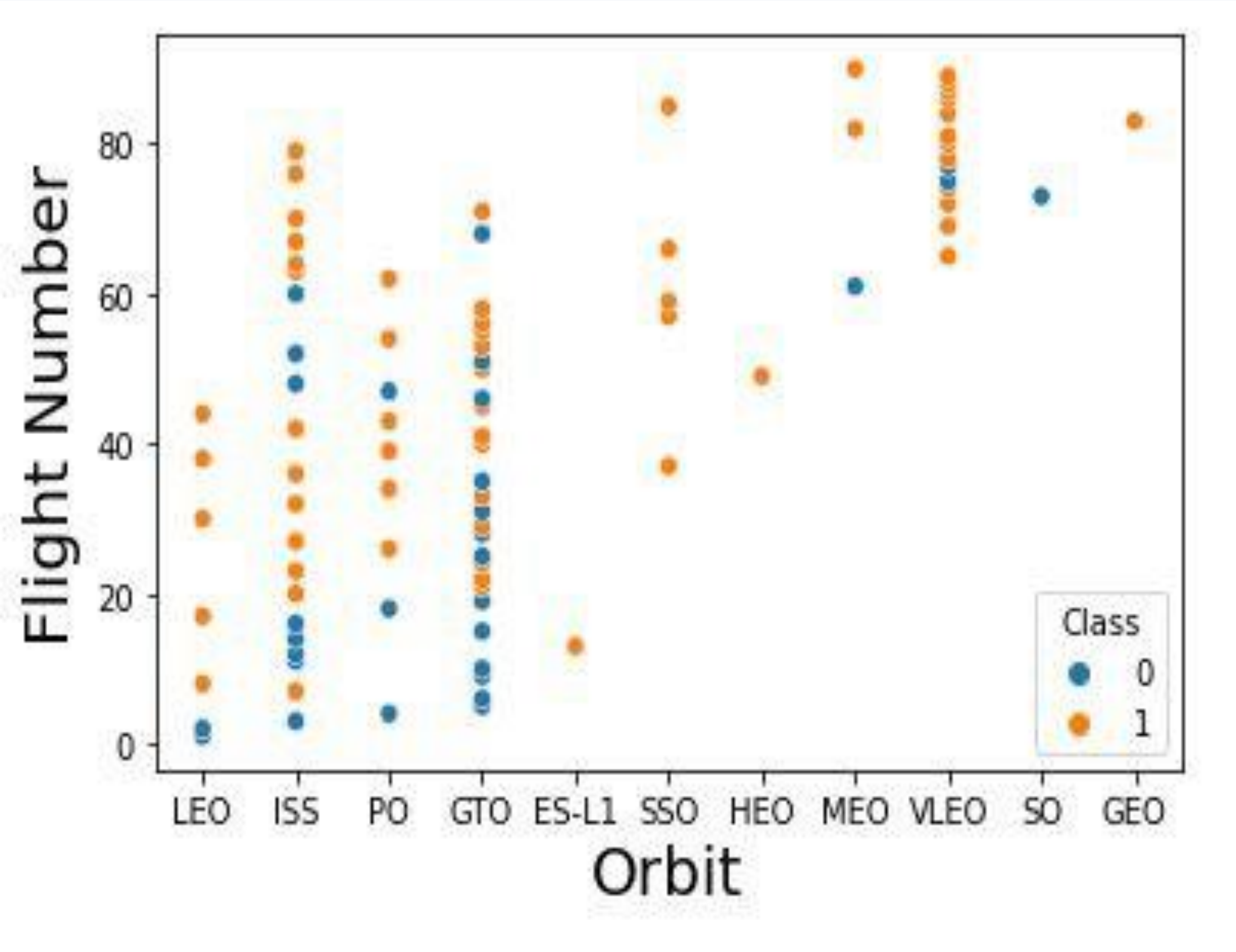
Success Rate vs. Orbit Type



From the Visualization we can concluded that:

- **GEO,HEO & ES-L1,SS) have high success rate .**

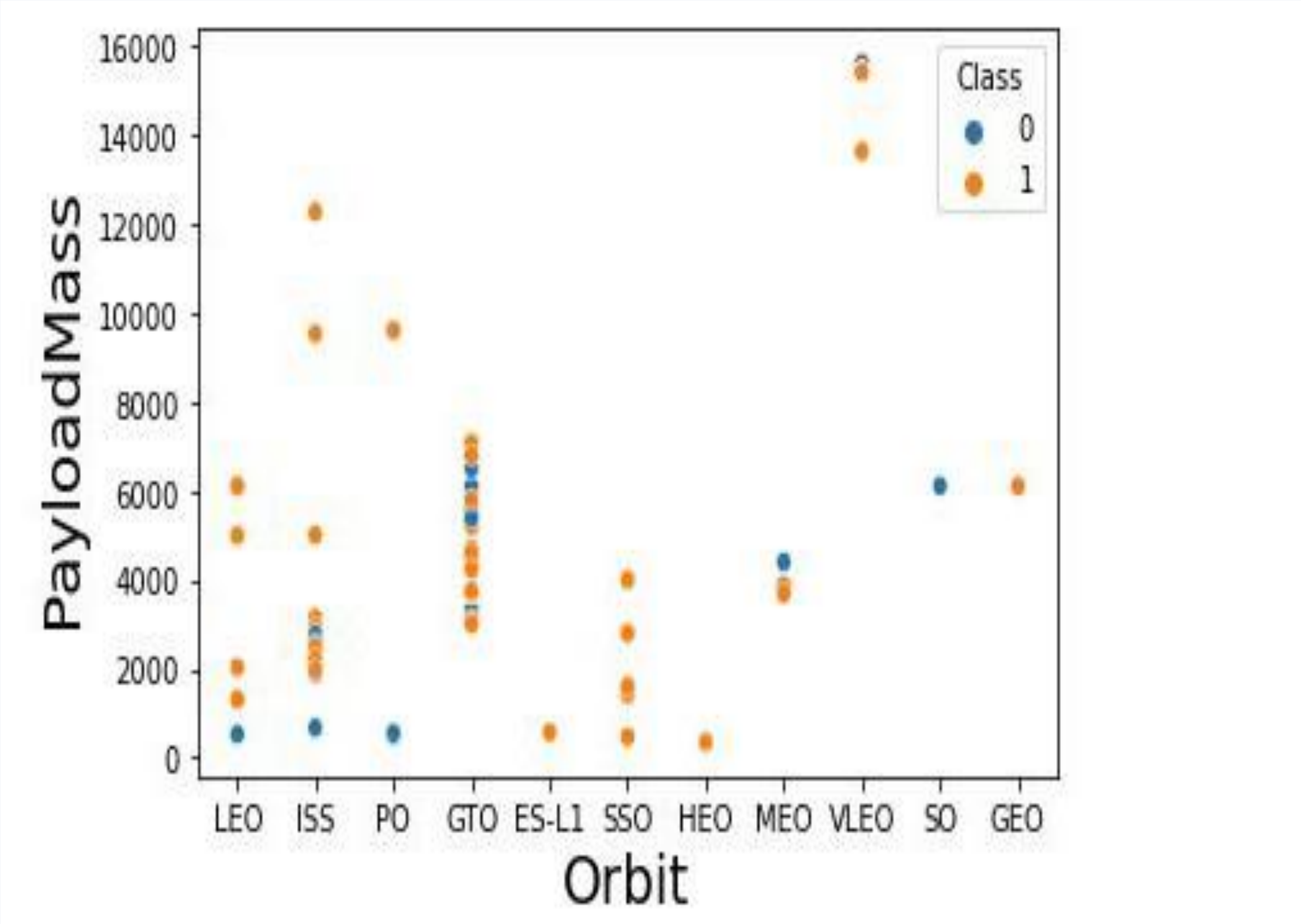
Flight Number vs. Orbit Type



From the Visualization we can concluded that:

- **Most Flight are to ISS,PO,GTO and VLEO**
- **MOST fails are for ISS,GTO**
- **SSO & VLEO has high success rate .**

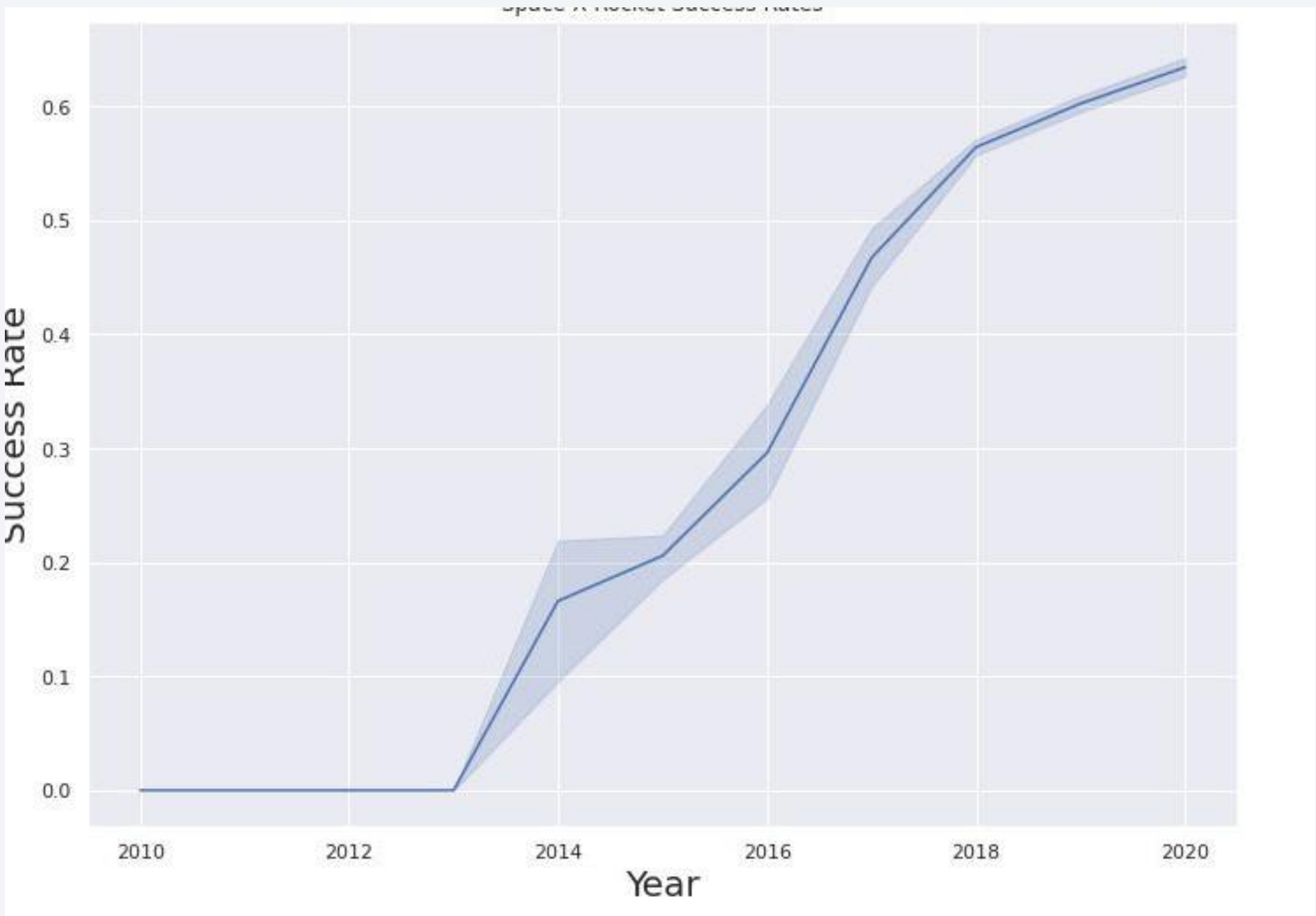
Payload vs. Orbit Type



From the Visualization we can concluded that:

- **Higher Payload are to the VLEO**
- **Least Payload are for HEO,ISS,PO,ES-L1**
- **GTO has average payload size .**

Launch Success Yearly Trend



The rate of success of the launches increase over time since to the data collected from the previous fails and success launches .

All Launch Site Names

❖ **Sites that SpaceX operates in are:**

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

Launch Site Names Begin with 'CCA'

```
[53]: # Query to get records where launch sites begin with 'CCA'
result = %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5

# Convert the result to a DataFrame for better display
df_result = pd.DataFrame(result)
print(df_result)
```

```
 * sqlite:///my_data1.db
Done.
         Date Time (UTC) Booster_Version  Launch_Site  \
0  2010-06-04   18:45:00  F9 v1.0  B0003  CCAFS LC-40
1  2010-12-08   15:43:00  F9 v1.0  B0004  CCAFS LC-40
2  2012-05-22    7:44:00  F9 v1.0  B0005  CCAFS LC-40
3  2012-10-08    0:35:00  F9 v1.0  B0006  CCAFS LC-40
4  2013-03-01   15:10:00  F9 v1.0  B0007  CCAFS LC-40
```

Total Payload Mass

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

```
%%sql
select sum(payload_mass__kg_) from SPACE  where customer LIKE '%CRS%'
```

```
 * ibm_db_sa://cdp97036:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/bludb
Done.
```

1
48213

Average Payload Mass by F9 v1.1

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

```
%%sql
select avg(payload_mass__kg_) from SPACE  where booster_version='F9 v1.1'
```

```
 * ibm_db_sa://cdp97036:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/bludb
Done.
```

1
2928

First Successful Ground Landing Date

```
[79]: import sqlite3
import pandas as pd

# Connect to SQLite database
con = sqlite3.connect("my_data1.db")

# Define the updated query with correct column names
query = """
SELECT MIN(date) AS First_Successful_Landing_Date
FROM SPACEXTABLE
WHERE Landing_Outcome LIKE '%Success%' AND Landing_Outcome IS NOT NULL
"""

# Execute the query and load the result into a DataFrame
df_result = pd.read_sql_query(query, con)

# Print the result
print(df_result)

# Close the connection
con.close()
```

```
  First_Successful_Landing_Date
0                    2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

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   SPACE  where landing__outcome LIKE '%drone ship%' and payload_mass__kg_>=4000 and payload_mas
```

```
 * ibm_db_sa://cdp97036:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/bludb
Done.
```

booster_version
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%%sql
SELECT Count(mission_outcome) from SPACE where mission_outcome LIKE '%Success%'
```

```
 * ibm_db_sa://cdp97036:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/bludb
Done.
```

1
100

Boosters Carried Maximum Payload

```
%%sql

SELECT  booster_version FROM Space  where  payload_mass__kg_ = (Select Max(payload_mass__kg_) from space)
```

```
 * ibm_db_sa://cdp97036:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/bludb
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,landing__outcome,DATE from space where landing__outcome LIKE '%drone%' AND DATE  LIKE '%
```

```
 * ibm_db_sa://cdp97036:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/bludb
Done.
```

booster_version	launch_site	landing__outcome	DATE
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)	10-01-2015
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)	14-04-2015
F9 v1.1 B1018	CCAFS LC-40	Precluded (drone ship)	28-06-2015

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

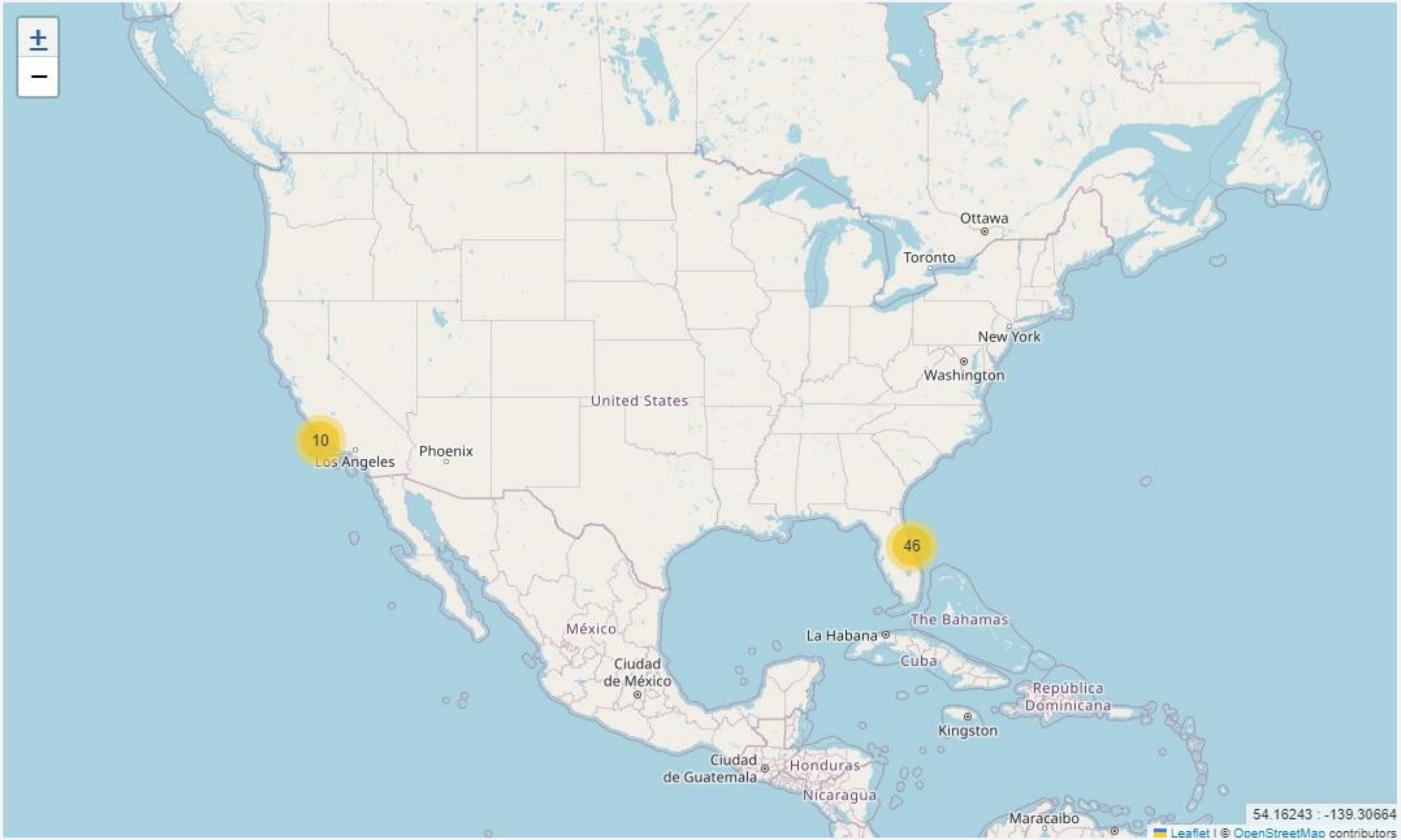
[83]:
```
import sqlite3
import pandas as pd

# Connect to SQLite database
con = sqlite3.connect("my_data1.db")

# Define the query to count landing outcomes between the specified dates and rank them
query = """
SELECT landing_outcome, COUNT(*) AS count
FROM SPACEXTABLE
WHERE date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY count DESC
"""

# Execute the query and load the result into a DataFrame
df_result = pd.read_sql_query(query, con)

# Print the result
print(df_result)

# Close the connection
con.close()
```

```
          Landing_Outcome  count
0              No attempt     10
1    Success (drone ship)      5
2    Failure (drone ship)      5
3   Success (ground pad)       3
4     Controlled (ocean)       3
5   Uncontrolled (ocean)       2
6    Failure (parachute)       2
7  Precluded (drone ship)      1
```

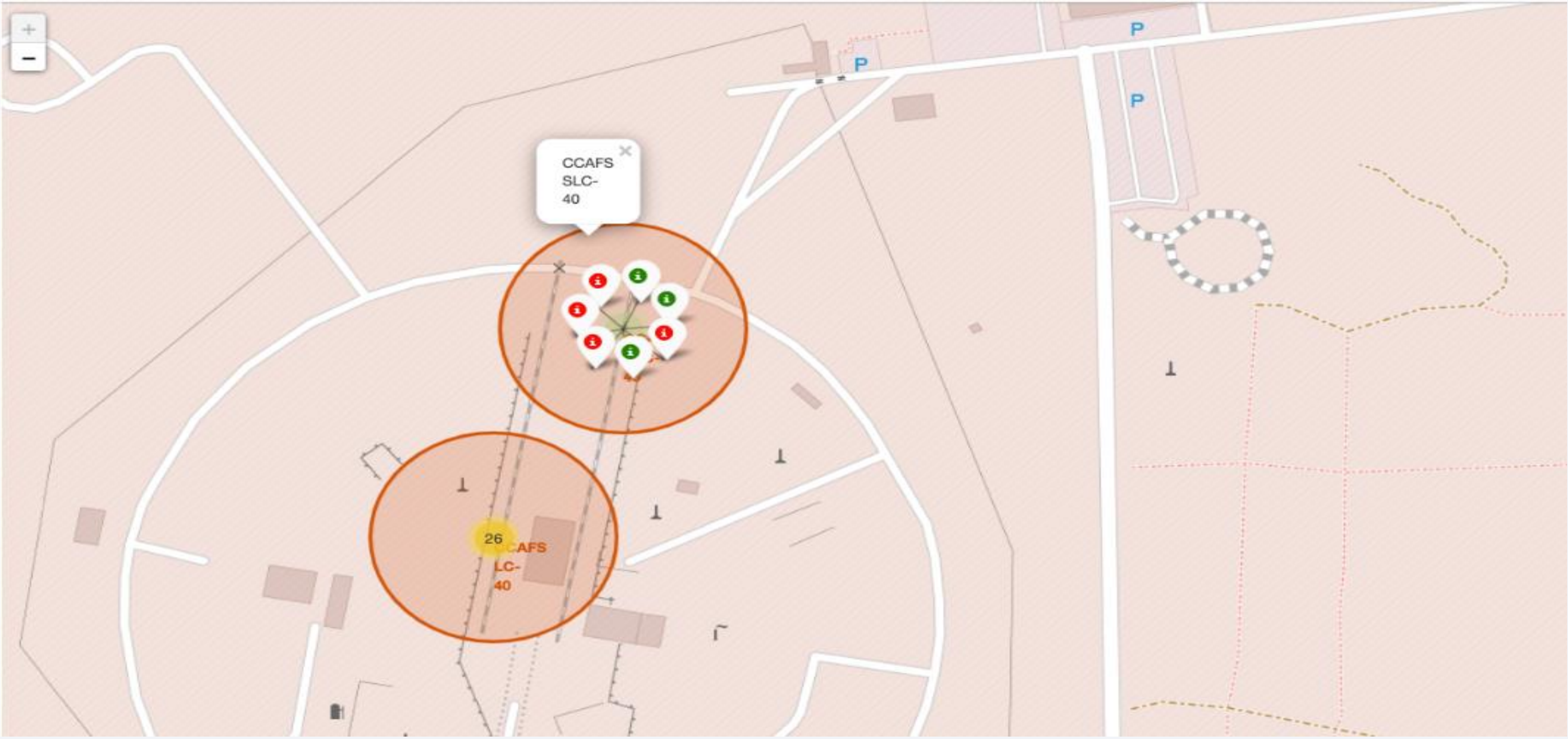
Section 3

Launch Sites Proximities Analysis

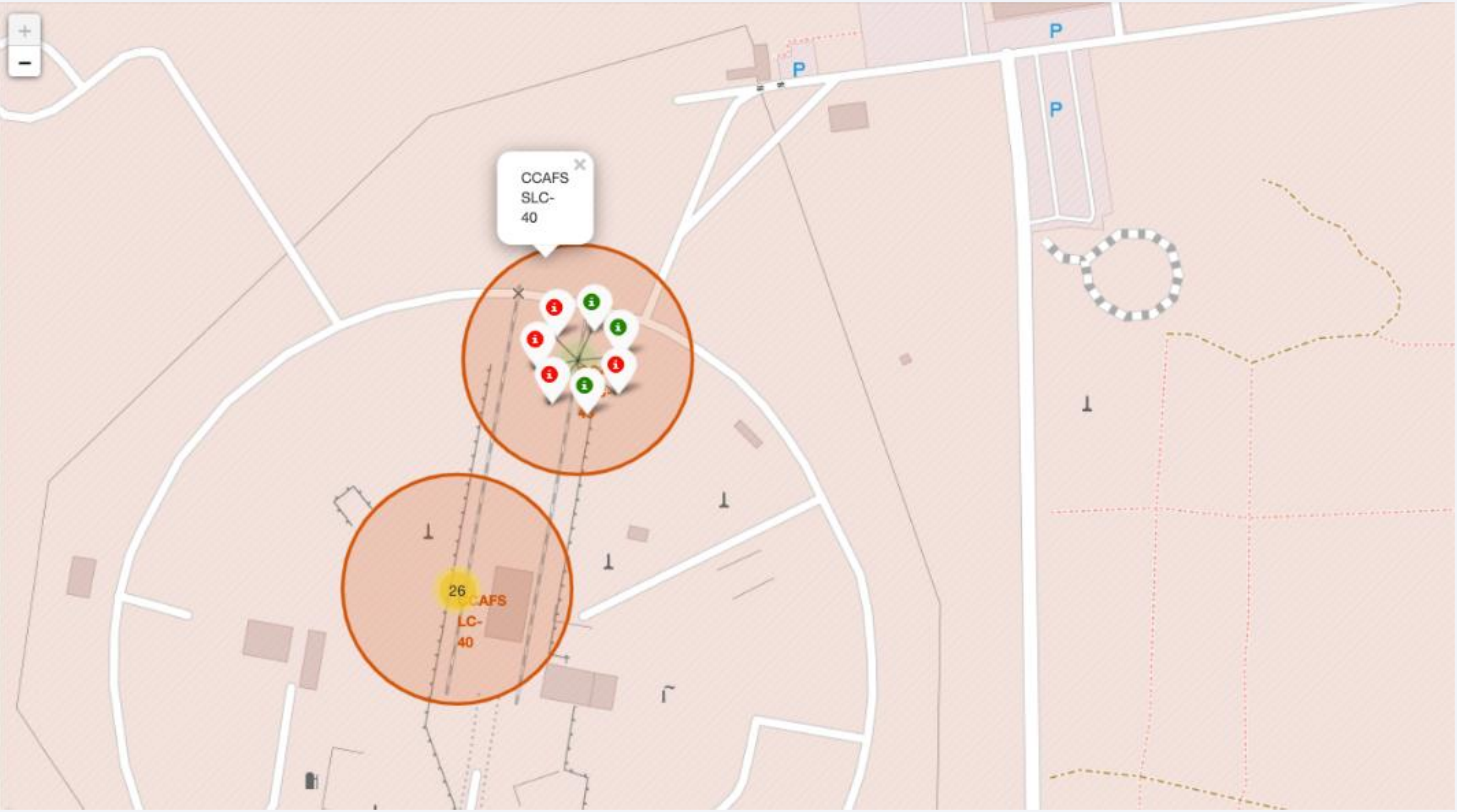
ALL LAUNCH SITES



Labeled Launch Outcomes



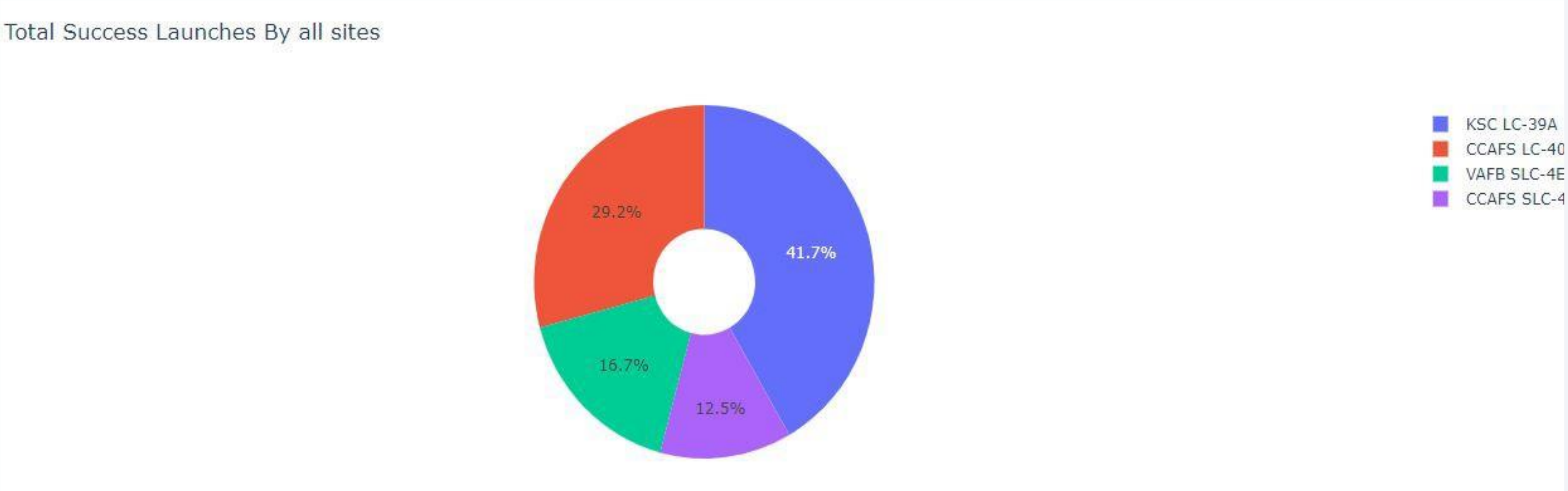
Launch Site Proximities



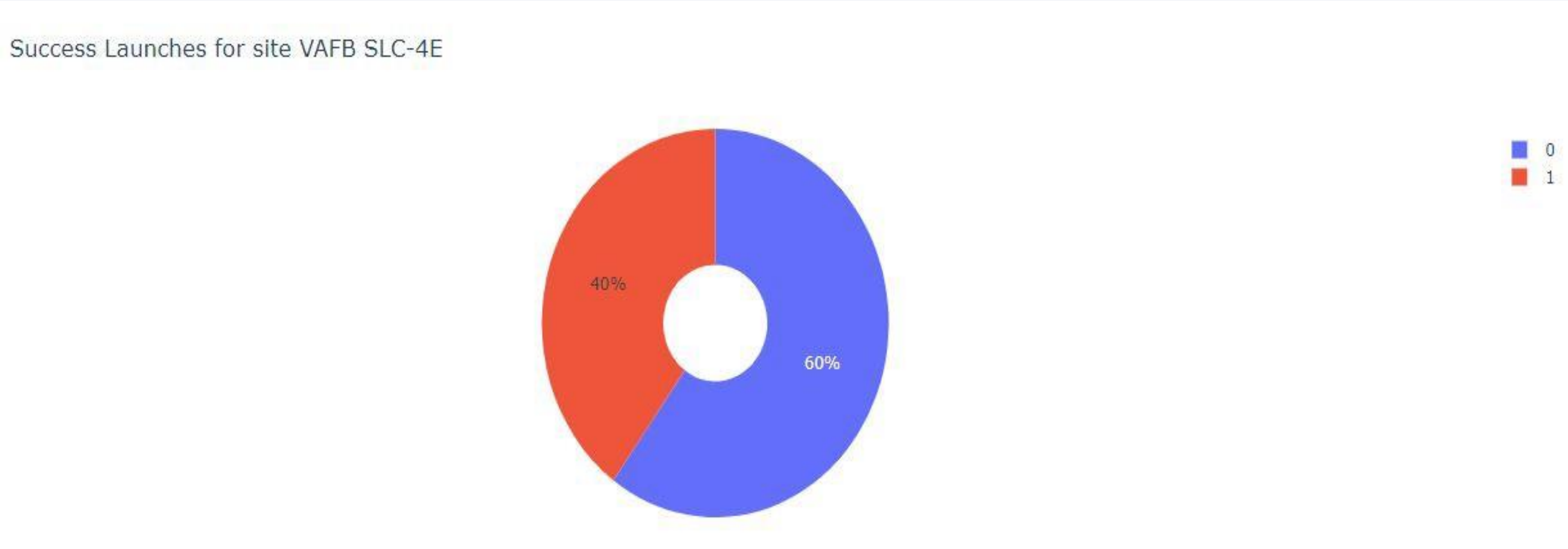
Section 4

Build a Dashboard with Plotly Dash

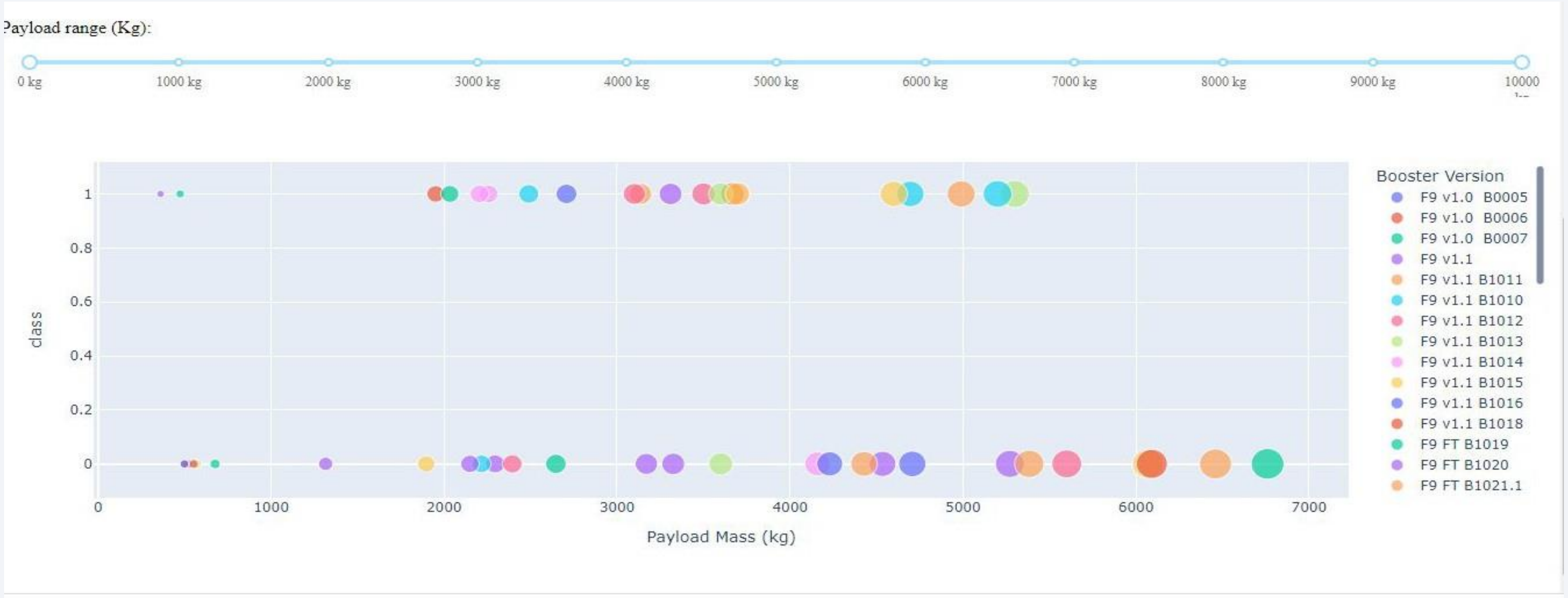
Success Launches By Sites



Launch By Site



Range Slider



Using the range slider we can view the sites that failed and succeed for each booster version and the Payload they were carrying .

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Best Model Prediction

After Analyzing all the Models,the KNN was the best Model with accuracy of 77% and best Score of 87%

```
parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
              'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
              'p': [1,2]}

KNN = KNeighborsClassifier()
gscv=GridSearchCV(KNN,parameters,scoring="accuracy",cv=10)
KNN_cv=gscv.fit(X_train,y_train)
```

```
print("Accuracy",KNN_cv.score(X_test,y_test))
```

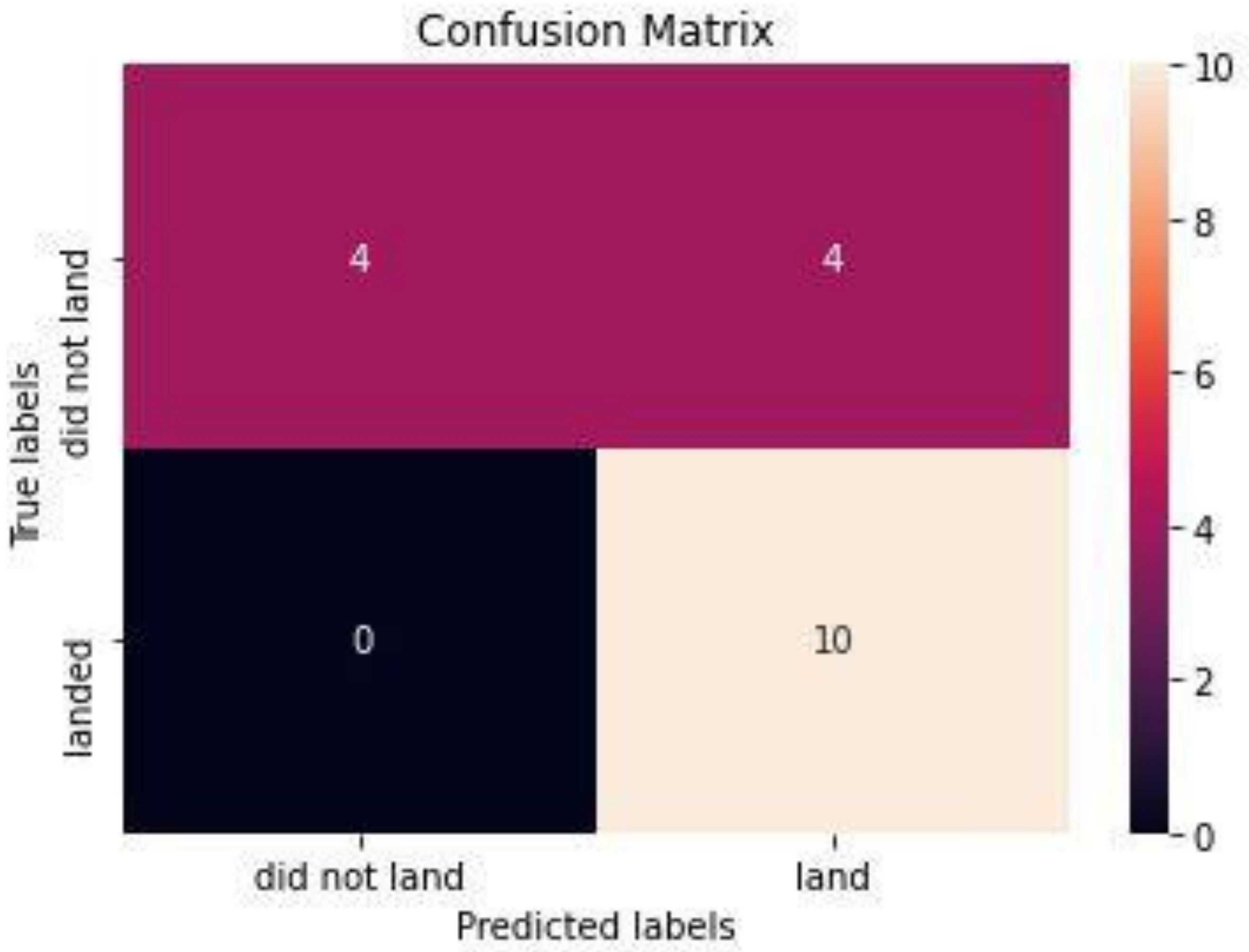
```
Accuracy 0.7777777777777778
```

```
print("tuned hpyerparameters :(best parameters) ",KNN_cv.best_params_)
print("accuracy :",KNN_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 4, 'p': 1}
accuracy : 0.8767857142857143
```

Confusion Matrix

```
yhat = KNN_cv.predict(X_test)
plot_confusion_matrix(y_test,yhat)
```



Conclusions

- **Comprehensive Data Analysis**: We meticulously collected and analyzed SpaceX launch data to understand key metrics and trends. This included evaluating launch outcomes, rocket types, payloads, and launch sites. By cleaning and transforming the data, we were able to derive meaningful insights that informed our subsequent analyses.
- **Effective Visualization and Exploration**: Through exploratory data analysis (EDA), we visualized various aspects of the launch data, including success rates, payload masses, and launch site distributions. Tools like Matplotlib, Seaborn, and interactive Folium maps were used to provide a clear and comprehensive view of the data, allowing us to identify patterns and anomalies effectively.
- **Insightful Predictive Analysis**: Predictive models were employed to classify launch outcomes and assess the potential success of future missions. We evaluated different algorithms and selected the best-performing model to predict launch outcomes based on historical data, providing actionable insights for future mission planning.
- **Interactive Dashboards and Reports**: We created interactive dashboards using Plotly Dash and detailed reports that encapsulate our findings. These tools enabled dynamic exploration of the data and presented the results in a user-friendly manner, facilitating a deeper understanding of the factors influencing launch outcomes and site performance.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!