IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yug Dahiya
5 May, 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using API and web scraping
  - Data Wrangling
  - Exploratory Data Analysis (EDA) and Visualization
  - EDA with SQL
  - Launch Site on Map using Folium
  - Dashboard with Plotly
  - Predictive analysis
- Summary of all results
  - EDA Results
  - Interactive maps and dashboard
  - Predicted Results

Introduction

- Project background and context

  In this project, we aim to find the outcome of the first stage of the Falcon 9. SpaceX says on its website that the Falcon 9 rocket launch cost 62 million dollars. Other providers cost upward of 165 million dollars each. The price difference is explained by the fact that SpaceX can reuse the first stage. By finding out whether the first stage will land or not, we can determine its cost. This information can prove crucial for any other companies that want compete with SpaceX.

- Problems you want to find answers
  - What are the main feature that affects the successful landing?
  - How each characteristic will affect the success rate of the landing?
  - What are the best feature that will help SpaceX to achieve the best success rate?

Section 1

Methodology

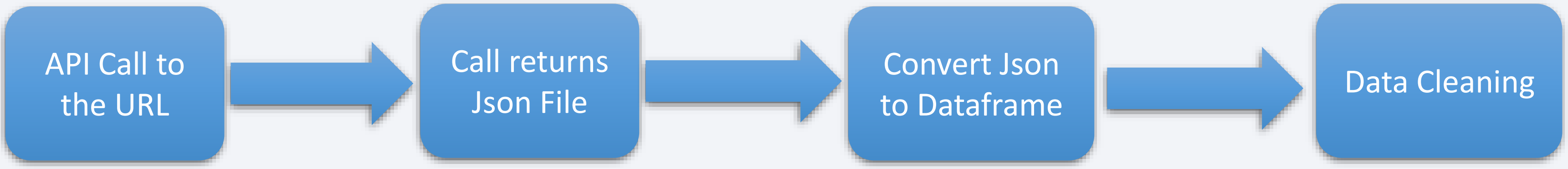
Methodology

Executive Summary

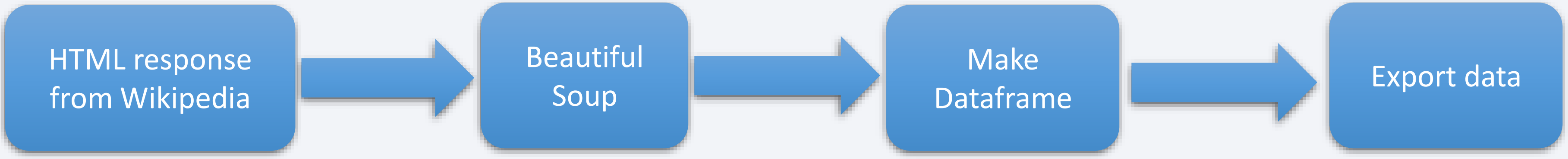
- Data collection methodology:
  - SPACEX Rest API
  - Web Scraping from Wikipedia
- Perform data wrangling
  - Dropped unnecessary columns
  - One Hot encoding for Classification modes
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Datasets are collected from Rest SpaceX API and webscraping Wikipedia.
  - Information obtained by API are on rockets, payload, launches and launch sites were collected.
    - The Space X REST API URL is api.spacexdata.com/v4/



- Information obtained by webscraping the Wikipedia are launches, payload information and landing
  - URL https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



Data Collection – SpaceX API

1.Getting response from API

```
response = requests.get(spacex_url)
```

2. Normalize JSON

```
data=pd.json_normalize(response.json())
```

3. Transform Data

```
BoosterVersion[0:5]
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

4. Combine columns into Dictionary

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

5. Create

```
data=pd.DataFrame(launch dict)
```

6. Filter Dataframe

```
data_falcon9=data[data['BoosterVersion']!='Falcon 1']
```

7. Export to CSV

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


IPNB File

Data Collection - Scraping

1.Getting response from HTML

```
response=requests.get(static_url).text
```

2.Create BeautifulSoup Object

```
soup=BeautifulSoup(response, 'html.parser')
```

3. Find Tables

```
html_tables = soup.find_all("table")
```

4.Extract 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
```

[IPNB File]

5. Create Dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Add Data

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is a
        if rows.th:
            if rows.th.string:
```

Rest of the code in the IPNB

7. Make Dataframe

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

8. Export to CSV

```
df.to csv('spacex web scraped.csv', index=False)
```

Data Wrangling

- In the dataset we can see that there are several cases when the booster did not land.
  - True Ocean, True RTLS, True ASDS means the mission was successfully.
  - False Ocean, False RTLS, False ASDS means mission was a failure.
- We need to convert string variables into categorical values in which all the true values are equal to 1 and all the false values are 0 concluding the mission was successful and failure respectively.

1. Calculate total number of launches for each site

```
df['LaunchSite'].value_counts()
LaunchSite
CCAFS SLC 40     55
KSC LC 39A       22
VAFB SLC 4E      13
Name: count, dtype: int64
```

2 Calculate the number and occurrence of each orbit

```
df['Orbit'].value_counts()
Orbit
GTO       27
ISS       21
VLEO      14
PO         9
LEO        7
SSO        5
MEO        3
HEO        1
ES-L1      1
SO         1
GEO        1
Name: count, dtype: int64
```

3 total number and occurrence of mission

```
landing_outcomes=df['Outcome'].value_counts()
Outcome
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: count, dtype: int64
```

4 Creating a landing outcome label from outcome column

```
landing_class = df['Outcome'].replace({'False Ocean': 0,
'False ASDS': 0, 'None None': 0, 'None ASDS': 0, 'False RTLS': 0,
'True ASDS': 1, 'True RTLS': 1, 'True Ocean': 1})
df['Outcome'] = df['Outcome'].astype(int)
df['Class']=landing_class
```

5 Export to csv

```
df.to_csv("dataset_part_2.csv", index=False)
```


IPNB File

EDA with Data Visualization

- There were a variety of graphs used in the visualizing the relationship between variables
- **Scatter Plot**: Shows the correlation between the variables
  - Flight Number VS Payload Mass
  - Flight Number VS Launch site
  - Payload Mass VS Launch Site
  - Orbit Vs Flight Number
  - Payload VS Orbit Type
  - Payload VS Payload Mass
- **Line Graph**: Line graphs can show the variables and their trends which can help in predicting unseen data
  - Success rate VS Year
- **Bar Graph:** Bar graph shows the relationship between numerical and categorical values
  - Success Rate VS Orbit

[LINK TO CODE]

EDA with SQL

- We used many SQL queries to filter and understand data more clearly
    - 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 all the booster versions that have carried the maximum payload mass. Use a subquery.
    - 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.

LINK TO CODE

Build an Interactive Map with Folium

- Folium Map Object is a map centered on NASA Johnson Space Center at Huston , Texas.
  - Red circle at NASA Johnson Space Center's coordinate with label showing its name.
  - Red circles at each launch site coordinates with labels shows launch site name.
  - Clustered points displays diverse information for the coordinates.
  - Green markers shows successful landing and Red markers shows unsuccessful landing.
  - Markers to show distance between launch site and nearest key location (railway, coast, highway) and a line to visually present the distance

LINK TO CODE

Build a Dashboard with Plotly Dash

- There were two graphs and two interaction added in the dashboard

  - Dropdown menu: Dropdown menu gives user the option to select between all or any particular launch site.

  - RangeSlider: Allows users to drag the slider to adjust the payload mass according to them.

  - Pie Chart: Shows the success rate and failure rate of all of any of the launch site.(Can be chosen using the dropdown menu)

  - Scatter Plot: Shows the relationship between the payload mass and Success. (Payload value can be adjusted using the RangeSlider)

LINK TO CODE

Predictive Analysis (Classification)

Data Preparation

- Load and Normalize data
- Split Train and Test Data

Model Preparation

- Selection of Algorithm
- Set Parameters to GridSearchCV()
- Train GridSearch() model

Model Evaluation

- Get best Hyperparameters for each model

Complete and test Model Accuracy and plot confusion matrix

Model Comparison

- Compare model accuracy
- Choose model with best accuracy

LINK TO CODE

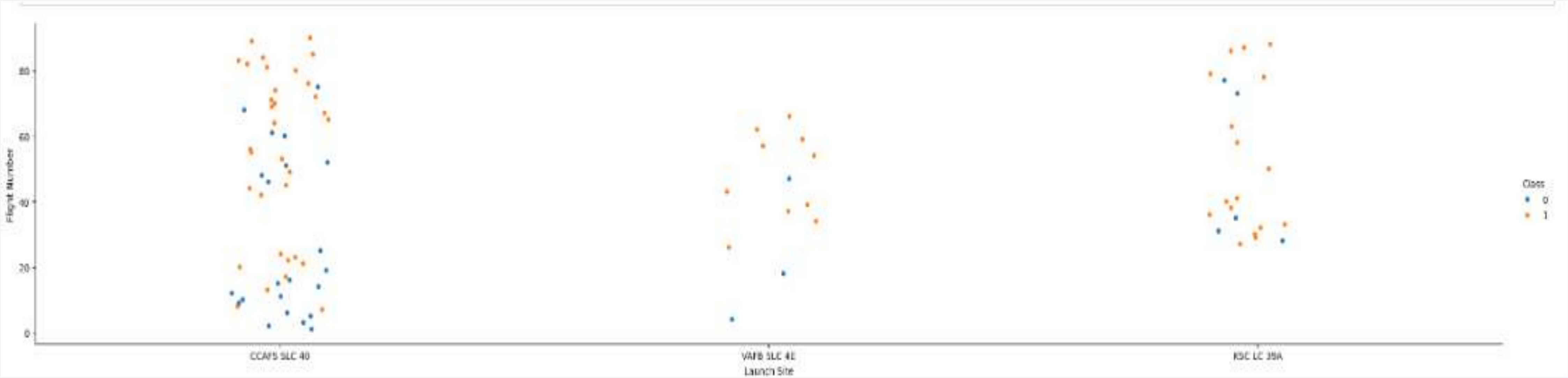
Results

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

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



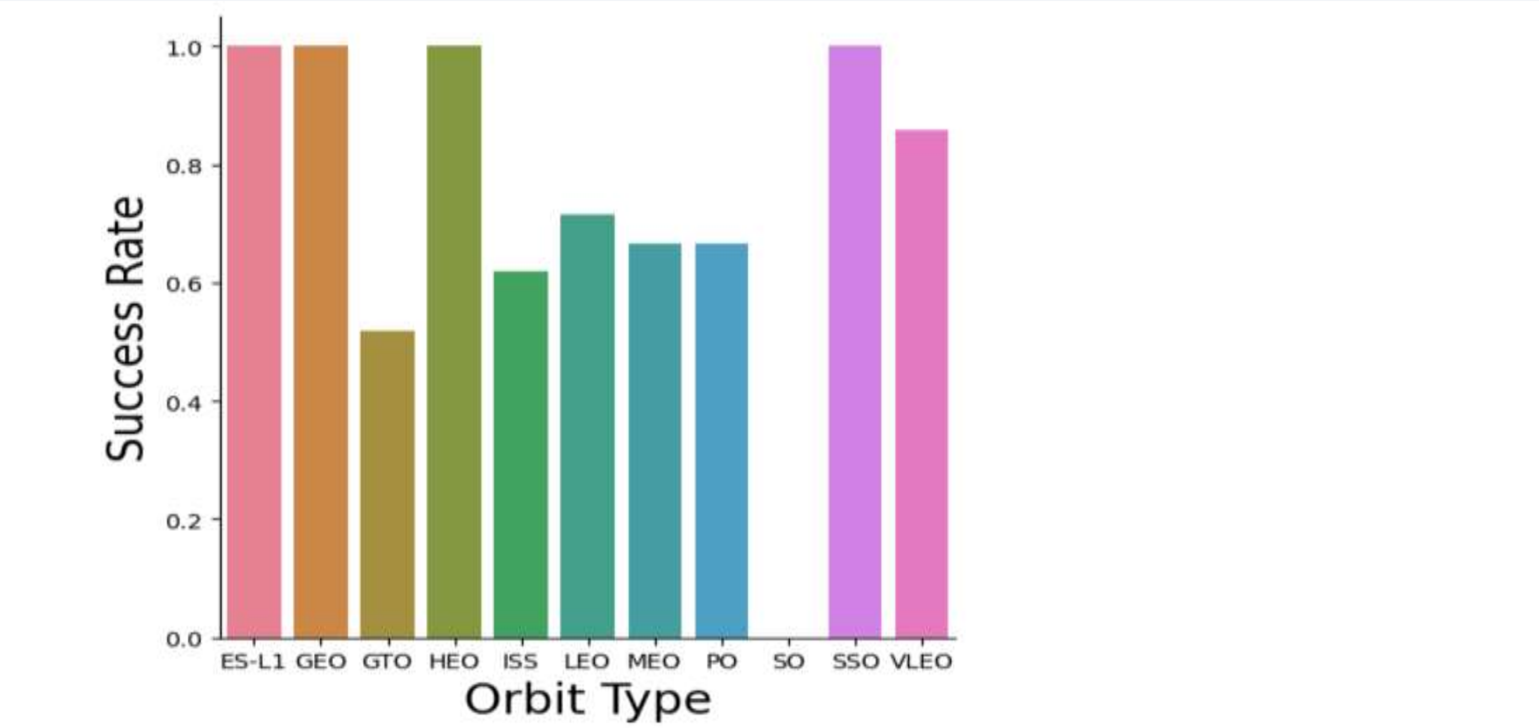
We can observe that CCAFS SLC-40 launch site has the most flights and successful flights can be seen as the flight number increases

Payload vs. Launch Site

Most of the flights have been launched when the payload mass ranges from 0 to 7000.

Different sites have different payload ranges for more successful launches.

Success Rate vs. Orbit Type

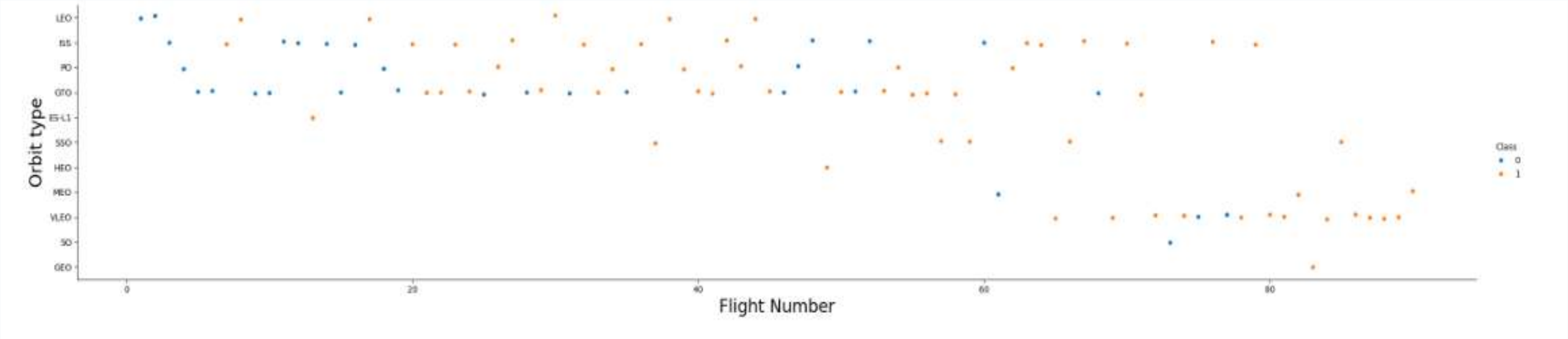


This bar chart shows the success rates for all the orbit types.

We can observe that ES-L1,GEO, HEO and SSO has the success rate of 1.

The only orbit type with 0 success rate is SO.

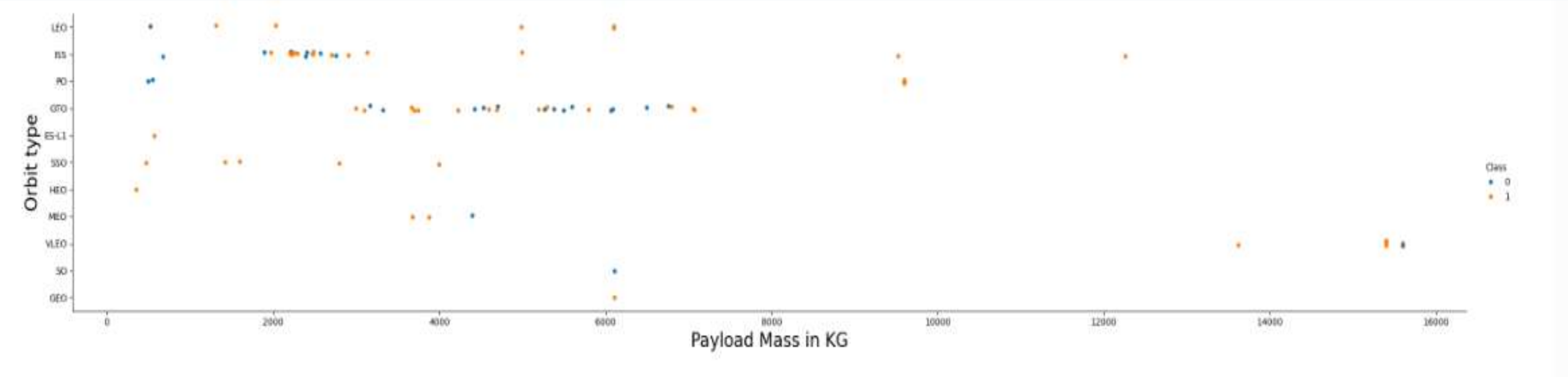
Flight Number vs. Orbit Type



You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

Flights launched in orbits like SO or GEO appears to be late than the others and have more successes.

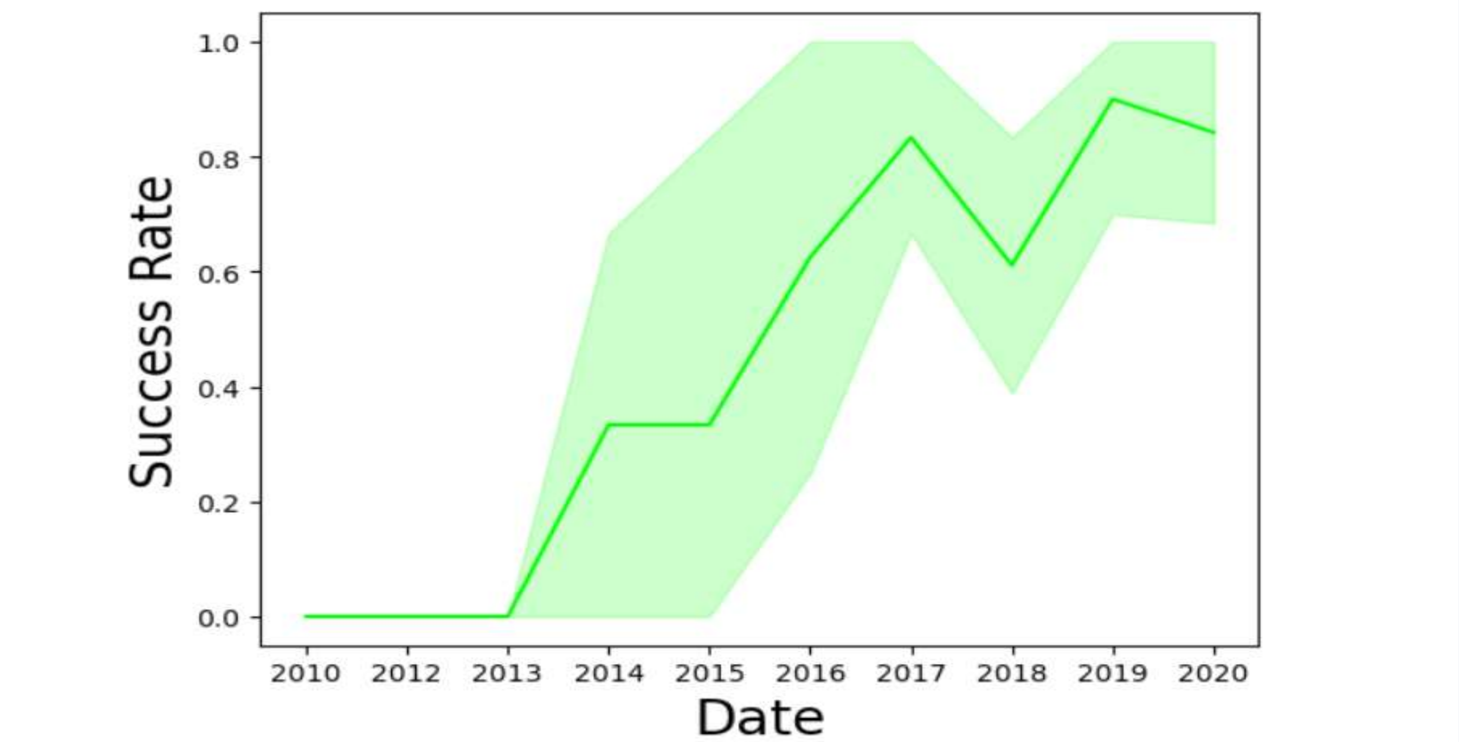
Payload vs. Orbit Type



Most of the flights are launched when the payload mass ranges between 0 to 7000.

The weight of payload influences the success rate. LEO has more success with high payload while GTO and SSO seems to be more successful with low payload.

Launch Success Yearly Trend



We can clearly observe that the launch success has significantly increased from 2013

All Launch Site Names

SQL Query

```
%sql Select distinct Launch_Site  from SPACEXTABLE
```

Output

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

This query returns all the distinct launch sites from the SpaceXtable.

Selecting all the different launch sites is done by the keyword ‘Distinct’

Launch Site Names Begin with 'CCA'

SQL Query

```
%sql select * from SPACEXTABLE where Launch_Site like('CCA%') limit 5
```

Output

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

All the launch sites that begin with CCA are called using this query.

Filter is done by adding where clause and like(‘CCA%’).

To get 5 result ‘Limit 5 is used’

Total Payload Mass

SQL Query

```
%sql SELECT SUM(payload_mass__kg_) FROM SPACEXTABLE WHERE customer = 'NASA (CRS)';
```

Output

SUM(payload_mass__kg_)
45596

Total payload mass can be found out using this query.

The keyword ‘sum’ lets you calculate the total payload mass.

Average Payload Mass by F9 v1.1

SQL Query

```
%sql SELECT avg(payload_mass__kg_) from SPACEXTABLE where Booster_Version="F9 v1.1"
```

Output

avg(payload_mass__kg_)
2928.4

By this query, we are finding the average payload mass for the Booster version F9 v1.1.

The keyword ‘avg’ is used to find the average

First Successful Ground Landing Date

SQL Query

```
%sql SELECT min(date) from SPACEXTABLE where Landing_Outcome like "%Success%"
```

Output

min(date)
2015-12-22

This query gives the first date on which the landing was successful

Keyword ‘min’ is used to find the earliest date;
Where clause is used to filter data;
Like keyword is used find the success outcome.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' and payload_mass__kg_ BETWEEN 4000 and 6000;
```

Output

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

We used this query to find all the booster version of all the successful landing on drone ship in which the payload mass ranges between 4000 and 6000.

Where clause is used to filter the dataset; ‘BETWEEN’ keyword is used to further filter all the data lying between 4000 and 6000 payload mass.

Total Number of Successful and Failure Mission Outcomes

SQL Query

```
%sql select count(mission_outcome), mission_outcome from SPACEXTABLE group by mission_outcome
```

Output

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

This query gives the count of all the success and failure outcomes.

Keyword ‘count’ is used to count all the mission outcomes;
Group by is used to group all the same data.

Boosters Carried Maximum Payload

SQL Query

```
%sql select Booster_Version from SPACEXTABLE where payload_mass__kg_=(select max(payload_mass__kg_) from SPACEXTABLE)
```

Output

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

To find all the booster that carried the maximum payload, we use this query.

Where clause is used to filter dataset by finding maximum payload mass ;
The ‘max’ keyword is used in a sub-query to find the maximum value of payload mass.

2015 Launch Records

SQL Query

```
%sql SELECT substr("DATE",6,2) as month, Booster_Version, Launch_Site FROM SPACEXTABLE where Landing_Outcome= "Failure (drone ship)" and substr("DATE", 0, 5)='2015
```

Output

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

In this query we found out all the failure landing outcomes in drone ship records of 2015;

Substr("DATE",6,2) is used to find the month and
Substr("DATE",0,5) is used to find the year;

Where clause is used to filter the dataset according to Failure (drone ship).

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

SQL Query

```
%sql SELECT LANDING_OUTCOME, count(Landing_Outcome) FROM SPACEXTABLE WHERE "DATE" >= '2010-06-04' <='2017-03-20' AND
Landing_Outcome LIKE "%Success%" or Landing_Outcome LIKE "%Failure%" group by Landing_Outcome ORDER BY count(Landing_Outcome) DESC;
```

Output

Landing_Outcome	count(Landing_Outcome)
Success	38
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Failure	3
Failure (parachute)	2

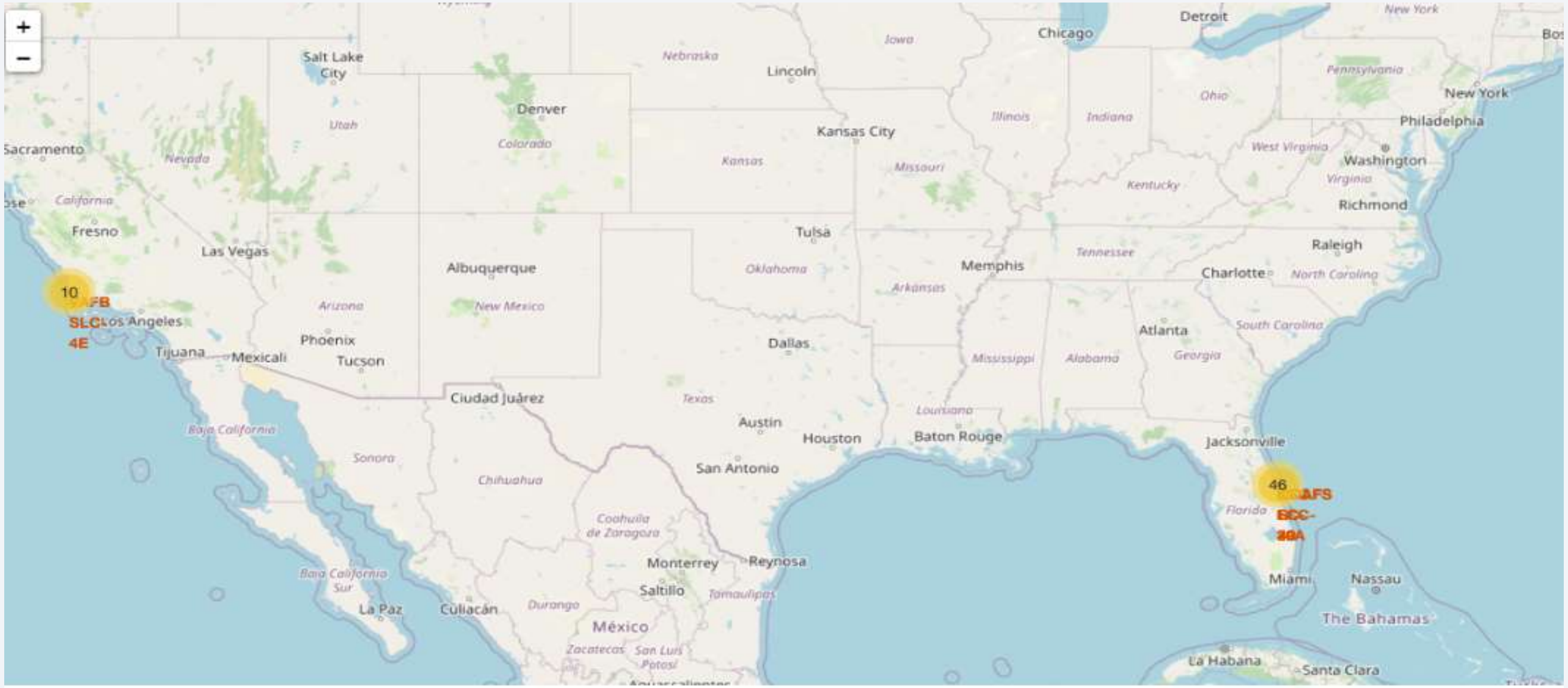
This query ranks the landing outcomes, success or failure between 2010-06-04 and 2017-03-20.

The dataset is filter according to the dates and landing outcomes then grouped and ordered by landing outcome

Section 3

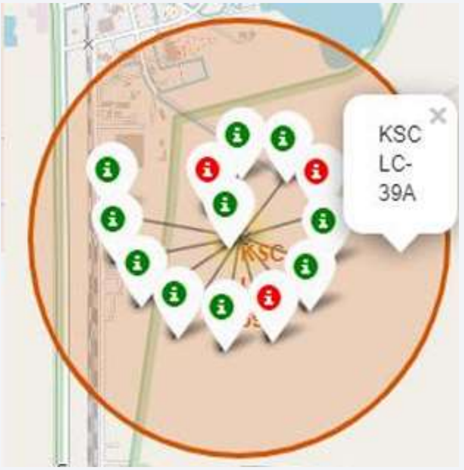
Launch Sites Proximities Analysis

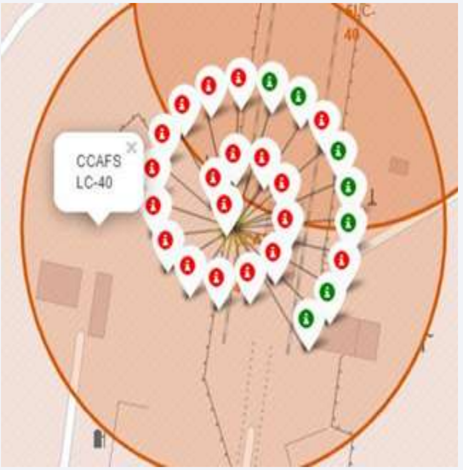
Folium Map- Ground Stations

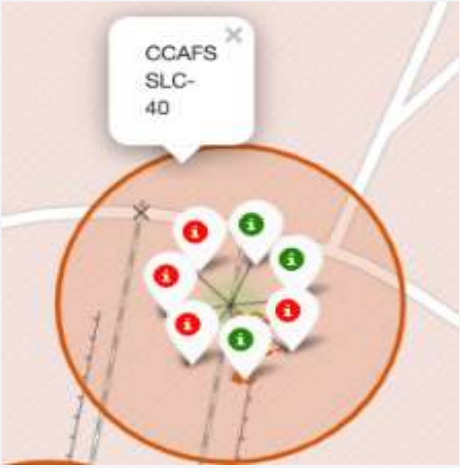


All the launch sites of SpaceX are marked. All them are very near to the coast

Folium Map- Labeled Markers



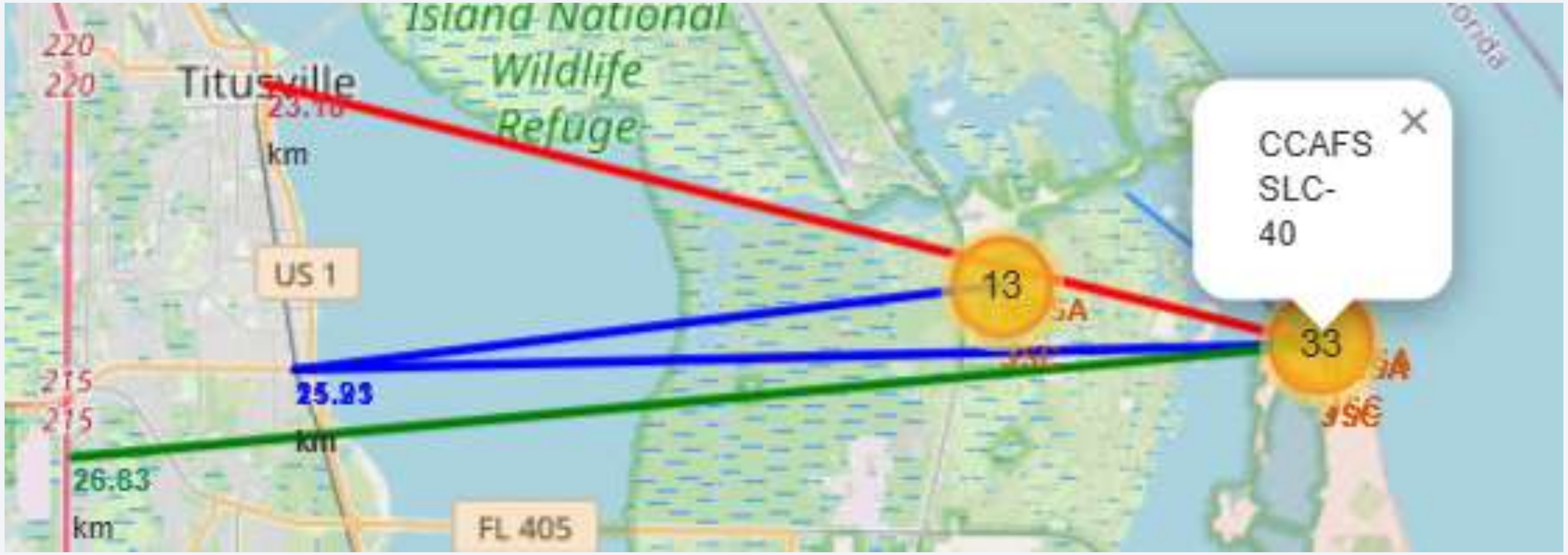




Green labels represent successful launches and Red labels represent failed launches.

KSC LC-39A has the most success rate and CCAFS LC-40 although has most launches but most of them are not successful.

Folium Map- Nearest proximities from CCAFS SLC-40

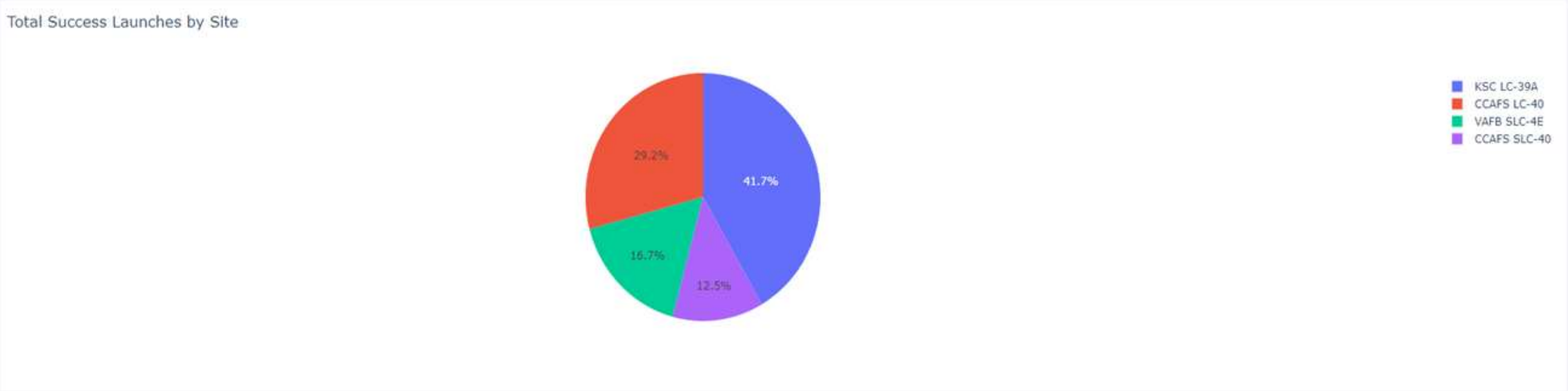


Coastline is in close proximity.
City , Highway and Railways are not in close proximity

Section 4

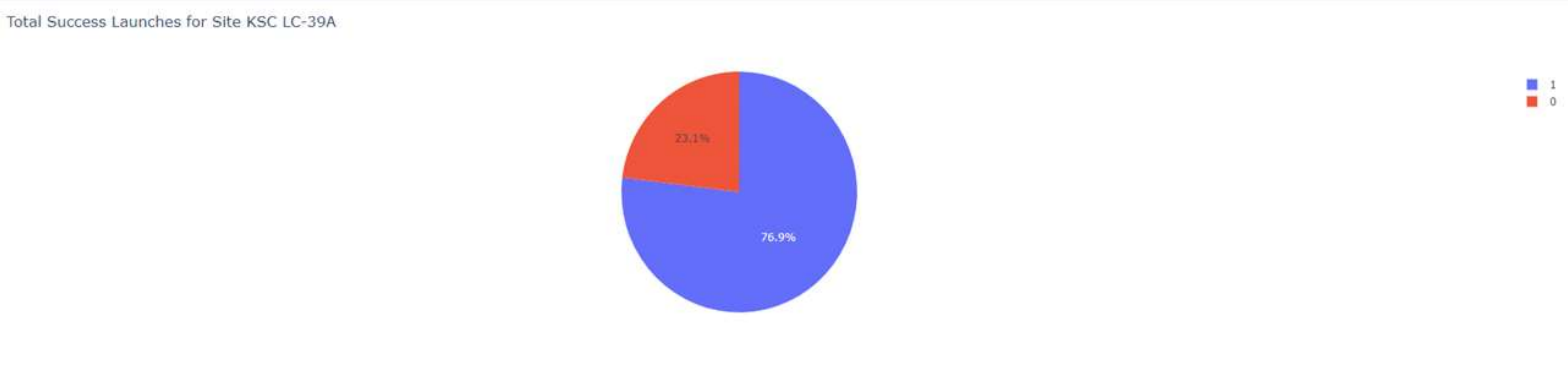
Build a Dashboard with Plotly Dash

Dashboard- Success of Sites



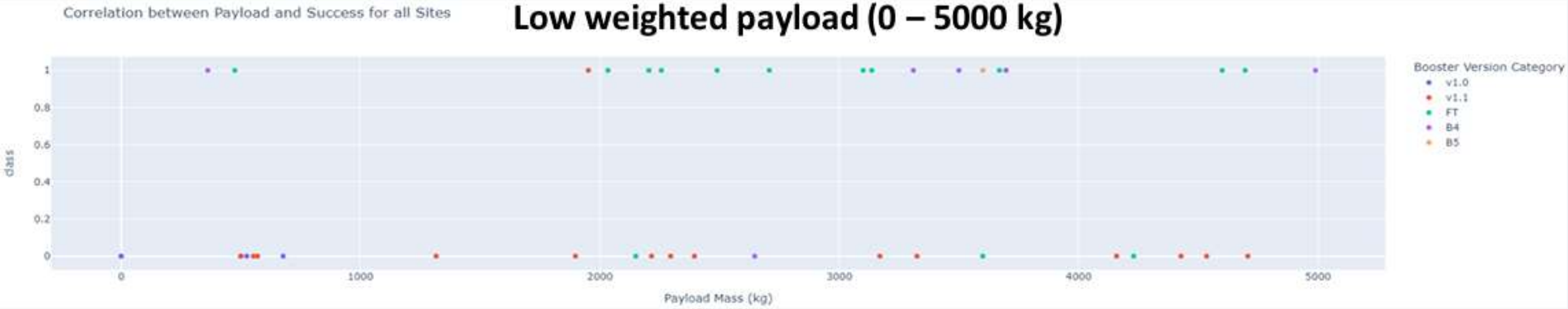
We can observe that KSC LC-39A have had the most success of all

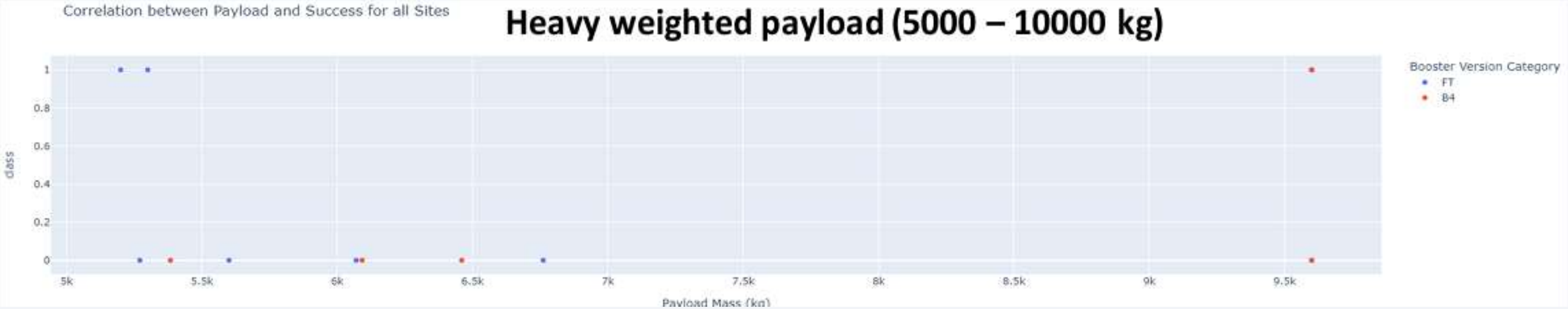
Dashboard- Successful Launches for KSC LC-39A



Site KSC LC-39A has 76.9% successful launches and 23.1% failed launches

Dashboard: Payload Mass VS Outcome for low and high payload mass



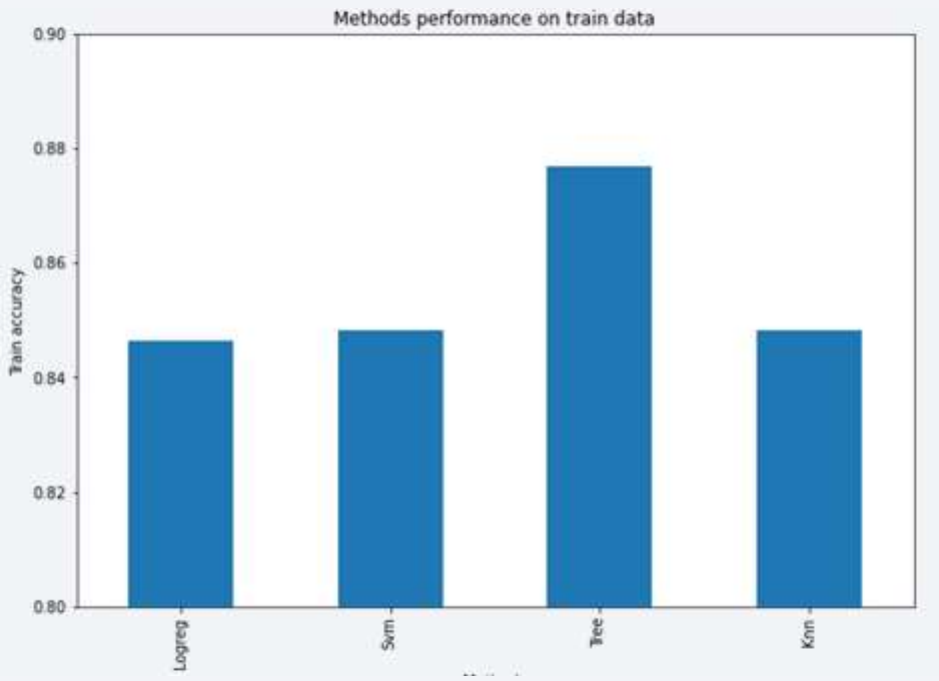


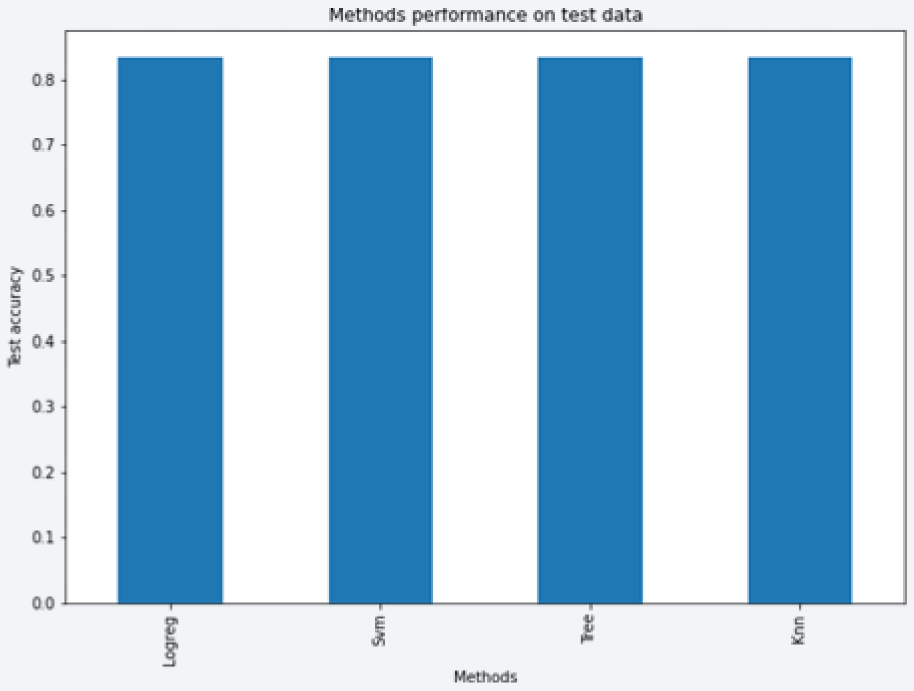
When Payload Mass ranges between 0-5000 kg, there appears to be more successful outcomes as compared when the payload mass is between 5000 and 10000 kg.
Most of the outcomes are seen till 6000 kg

Section 5

Predictive Analysis (Classification)

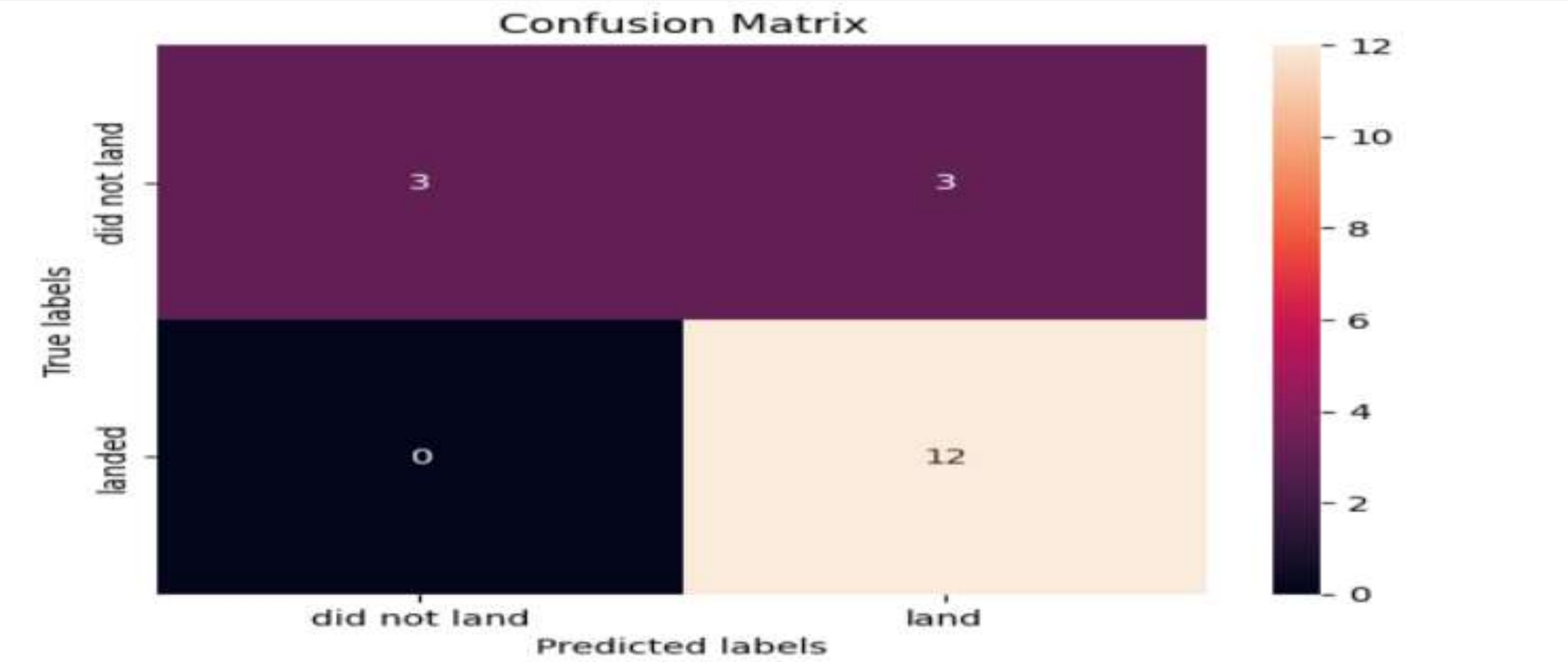
Classification Accuracy





All of the models performed really well and similarly but the Decision tree just came on top because of its train accuracy

Confusion Matrix

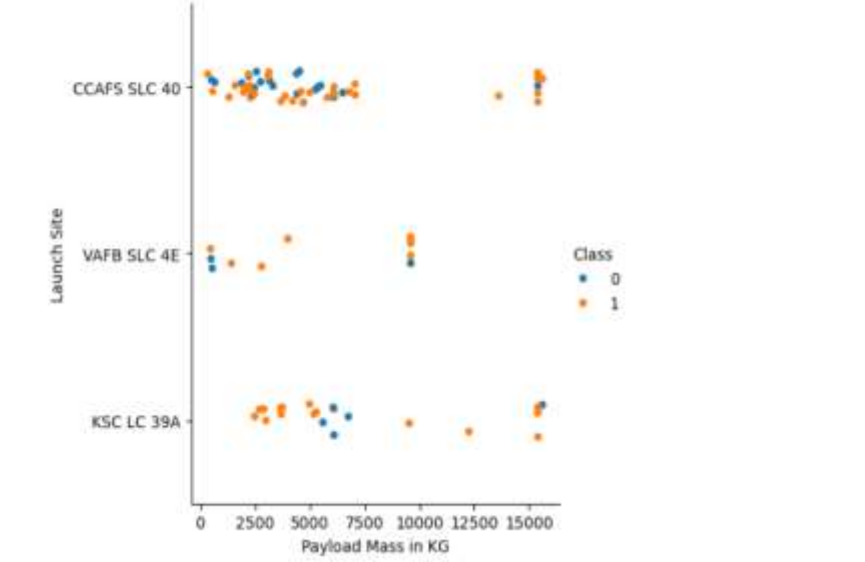


Confusion matrix of Decision tree can distinguish different classes. The problem of this model is false positives

Conclusions

- There are many features that affect the success and failure rate of the launches such as launch site, orbit type , payload mass. From 2013 there has been a massive jump in the success rate of the launches, which can also indicate that the previous knowledge about the failures may have helped.
- GEO, HEO, SSO are found out to be the orbits with most success rate.
- Low payload masses usually have better success rate for most of the orbit types. Only a few orbit types actually provide better success numbers. This could be because of the distance of the orbits.
- We saw that CCASF LC-40 had the most amount of launches from all the four sites. But the success rate was very low. KSC LC-39A do not have as many launches as CCASF LC-40 but the success rate was very high. This could mean that KSC LC-40 is the most successful launch site.
- We used four different pipeline models to predict whether the first stage will land or not. All of the models performed really good and were almost identical. But Decision Tree just takes the edge over the others because of its good train accuracy results

Appendix



	FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	ReusedCount	L1	Orbit_GEO	...	Serial_B1048	Serial_B1049	Serial_B1050	Serial_B1051	Serial_B1054
0	1.0	6104.959412	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
1	2.0	525.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
2	3.0	677.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
3	4.0	500.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
4	5.0	3170.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
85	86.0	15400.000000	2.0	1.0	1.0	1.0	5.0	2.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
86	87.0	15400.000000	3.0	1.0	1.0	1.0	5.0	2.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
87	88.0	15400.000000	6.0	1.0	1.0	1.0	5.0	5.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0
88	89.0	15400.000000	3.0	1.0	1.0	1.0	5.0	2.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
89	90.0	3681.000000	1.0	1.0	0.0	1.0	5.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0

Thank you!