IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mohit Punasiya
08-08-2022

GitHub : https://github.com/MohitPunasiya/SpaceX_FirstStage_Recovery_Prediction

Outline

Executive Summary

- The methodology involves web-scrapping launches info using SpaceX API, Data cleaning, feature engineering, Exploratory data analysis through interactive visual analytics. And at last model training, testing and fine tuning.
- The key observations were best performance by logistics regression and SVM model with Testing accuracy of 94%. Launch success rate dependence on payload mass and launching site.

Introduction

- **SpaceX** designs, manufactures and launches advanced rockets and spacecraft. They provide this services at much lower cost than market primary reason being the SpaceX's Technology to reuse the first stage. There is a need for a model which can predict the success or failure of this launches. This model outcomes can potentially influence decisions of those who are on the other side of the business.   This project aims to explore data driven approach to find out success prediction of first stage of Falcon 9 rocket.

- The question that would be addressed

➢ Can machine learning based model give promising results in predicting the success of landing of falcon 9 first stage?

➢ What are the variables(Launch site, payload mass) that the success rate depends upon?

➢ Which models perform better?

Rocket stages

- This image provides context of what, is meant by first stage and payload of rocket

Source : https://www.quora.com/What-do-we-mean-by-rocket-stages

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data was collected via web-scrapping from SpaceX REST API ; api.spacexdata.com/v4/
- Perform data wrangling
    - Data wrangling included filter Falcon 9 boosters only, missing data imputation for payload mass & one hot bit encoding
- 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

- ❑ Web-scrapping through SpaceX REST API (api.spacexdata.com/v4/)
- ❑collect publicly available data about Falcon 9 launches from the SpaceX API and Wikipedia

Data Collection – SpaceX API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url).json()
```

Getting Responses from API → Converting Responses to . JSON file → Apply custom functions to clean data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

↓

Assign list to dictionary then dataframe

```
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}
```

← Filter dataframe & export to flat files

```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

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

GitHub URL : https://github.com/MohitPunasiya/SpaceX_FirstStage_Recovery_Prediction/blob/main/1_DataCollection.ipynb

Data Collection - Scraping

Getting Response from HTML
Creating Beautiful soup object and Find tables
Getting Column Names
Creation of dictionary
Appending Data to keys
Converting to dictionaries and to .CSV
Data Frame to

```
page = requests.get(static_url)
```

```
soup = BeautifulSoup(page.text, 'html.parser')
html_tables = soup.find_all('table')
```

```
launch_dict= dict.fromkeys(column_names)

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

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(
   # get table row
    for rows in table.find_all("tr")
        #check to see if first table
```

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

- Getting HTML Response from wiki media > Data extraction using Beautiful Soup > (html tables to Creation of dictionary)
- GitHub URL to notebook : https://github.com/MohitPunasiya/SpaceX_FirstStage_Recovery_Prediction/blob/main/webscraping.ipynb

Data Collection - Scraping

Getting Response from HTML
Creating Beautiful soup object and Find tables
Getting Column Names
Creation of dictionary
Appending Data to keys
Converting to dictionaries and to .CSV
Data Frame to

```
page = requests.get(static_url)
page.status_code
```

```
soup = BeautifulSoup(page.text,'html.parser')
```

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

```
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
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
```

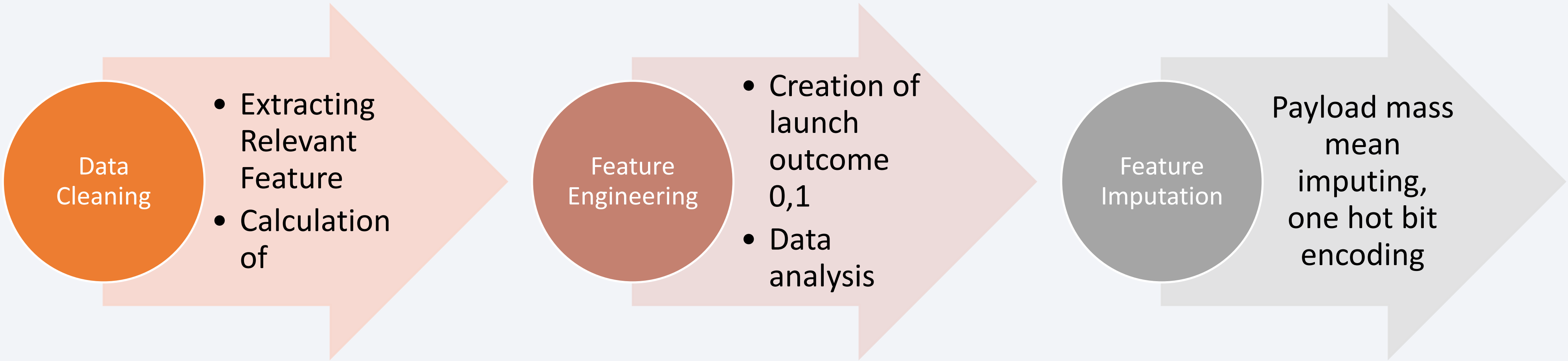
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

- Getting HTML Response from wiki media > Data extraction using Beautiful Soup > (html tables to Creation of dictionary)

- GitHub URL to notebook : https://github.com/MohitPunasiya/SpaceX_FirstStage_Recovery_Prediction/blob/main/webscraping.ipynb

Data Wrangling

- As part of data wrangling extraction, elimination and filtering of relevant feature was done
- Next missing values were identified and were taken care of using mean filling

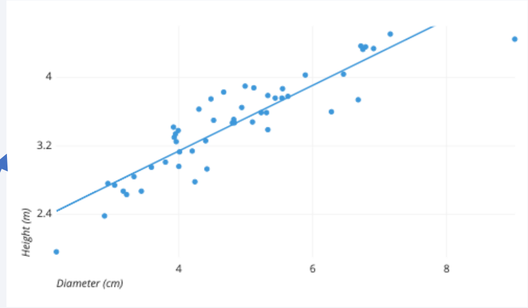


GitHub URL :
https://github.com/MohitPunasiya/SpaceX_FirstStage_Recovery_Prediction/blob/main/DataWrangling.ipynb

EDA with Data Visualization

- Charts that were plotted

Scatter Graph	**Flight Number VS Payload Mass** **Flight Number VS Launch Site** **Payload VS Launch Site** **Orbit VS Flight Number** **Payload VS Orbit Type** **Orbit VS Payload Mass**
Bar Graph	Mean VS Orbit
Line Graphs	Success Rate VS Year
Folium Maps	Maps were created using Folium to locate launch site location on graph and were marked by their success or failure



GitHub Link :
https://github.com/MohitPunasiya/SpaceX_FirstStage_Recovery_Prediction/blob/main/Pholium_launch_site_location.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 acheived.**
- **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. Use a subquery**
- **List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015**

- GitHub Link : https://github.com/MohitPunasiya/SpaceX_FirstStage_Recovery_Prediction/blob/main/EDA_SQL.ipynb

Build an Interactive Map with Folium

- Markers, Popups, Marker customization , Marker Cluster, were use ful functions that were used to visualize launch sites
- To Understand the location of launch sites and their relation ship with the success rate of launches this task was performed.
- The launches were colored red (Success) and Green(Failure)
- A line was plotted to connect coast to launch site and to calculate distance in between them

- GitHub URLhttps://github.com/MohitPunasiya/SpaceX_FirstStage_Recovery_Prediction/blob/main/Pholium_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Used Plotly dash dashboard to play around with data and visualized various parametric influences
- Graphs

- Pie chart to show the total launches by a certain site / all sites

- Scatter Graph Showing the relationship with Outcome and payload mass (KG) for the different booster versions

- GitHub URL : https://github.com/MohitPunasiya/SpaceX_FirstStage_Recovery_Prediction/blob/main/PlotlyDash_for_project.ipynb

Predictive Analysis (Classification)

Model building

Hyper-parameter tuning using GridSearchCV

Model Evaluation

Dataset preparation Train-Test-Split → Model Building and Fine Tuning → Model Evaluation

GitHub URL :
https://github.com/MohitPunasiya/SpaceX_FirstStage_Recovery_Prediction/blob/main/Prediction_Model.ipynb

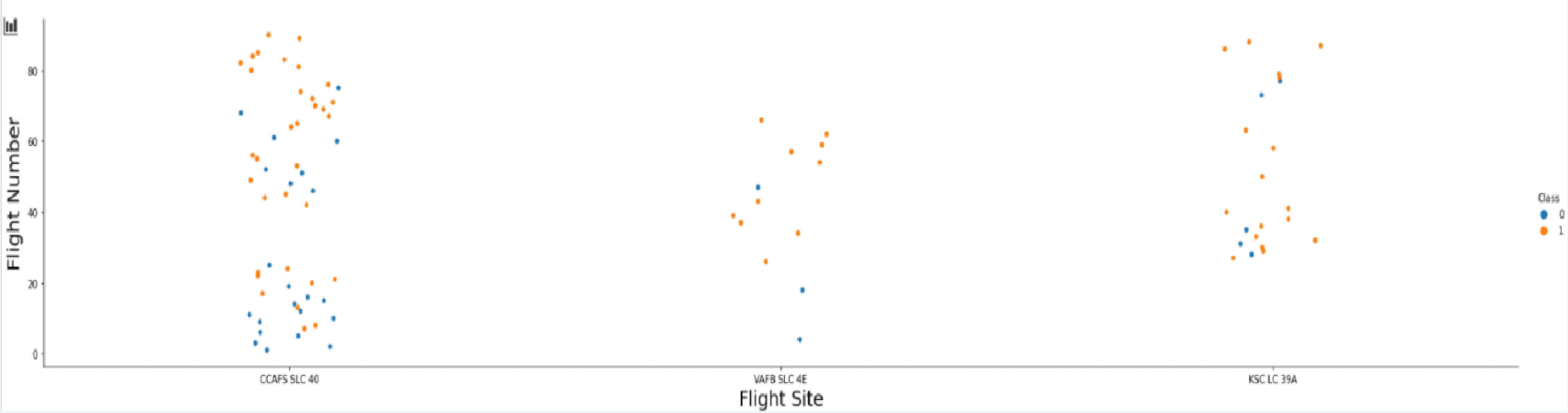
Results

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

Section 2

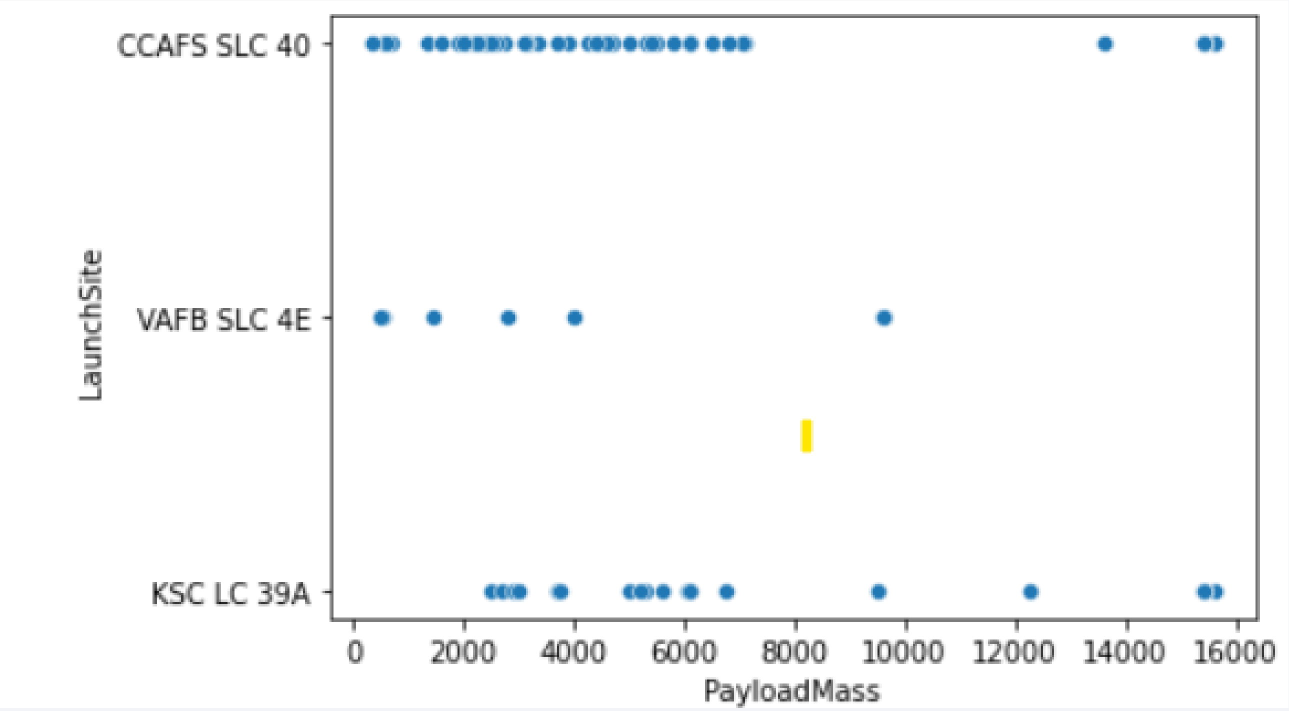
Insights drawn from EDA

Flight Number vs. Launch Site



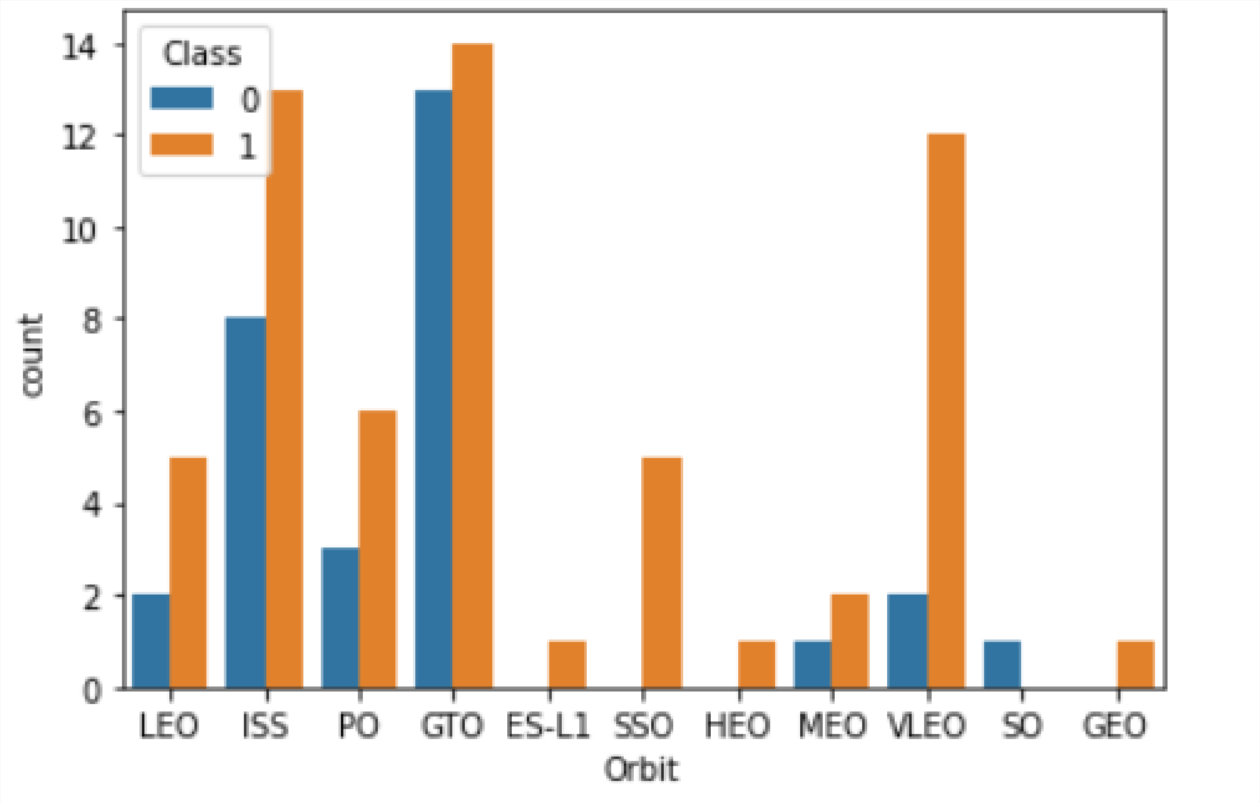
One Launch site has been used multiple times this site also has higher success rate. This might be due to the fact that the initial testing were done on this site

Payload vs. Launch Site



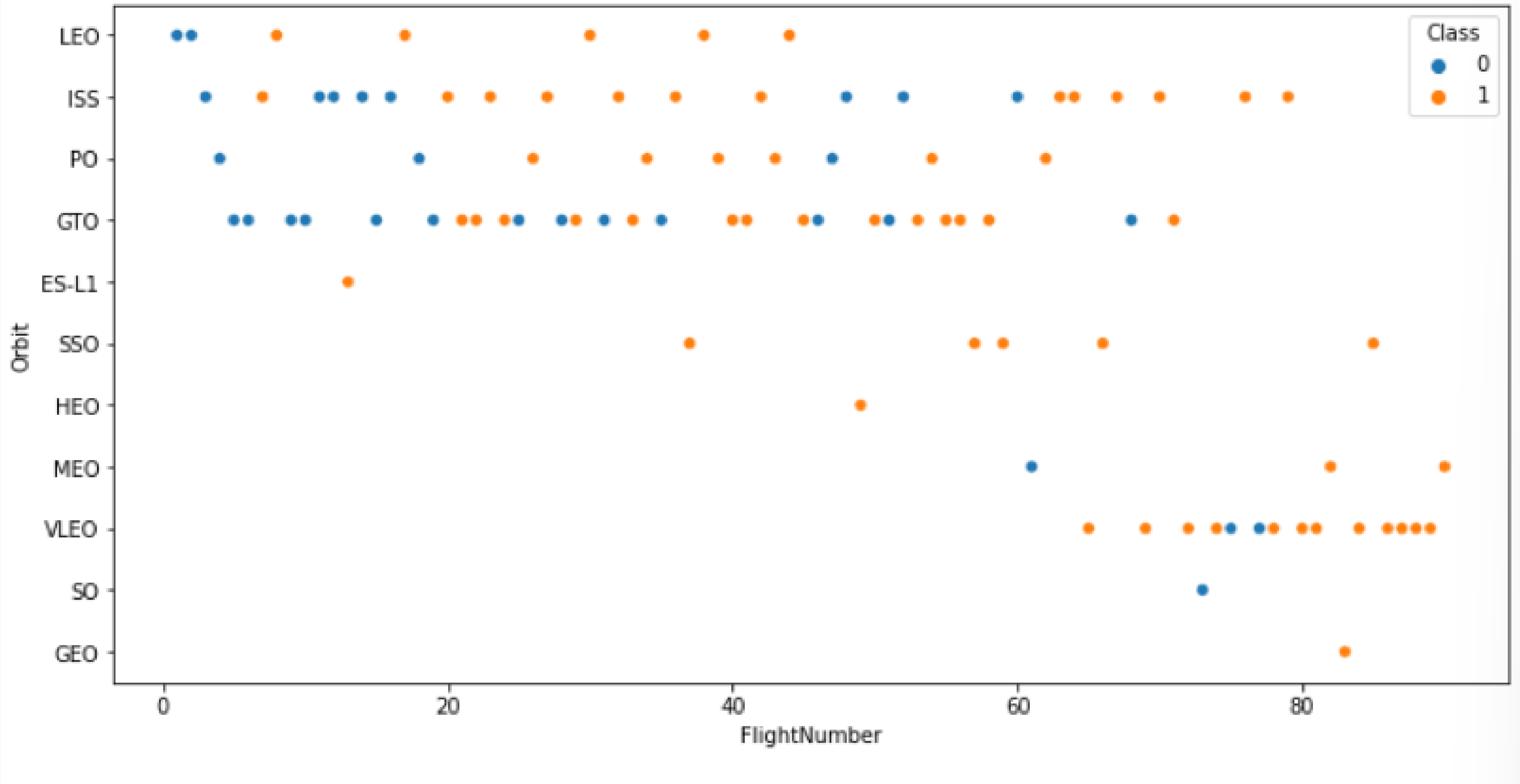
Most of the launches have been tried out at CCAFS SLC 40 and more than 85% launches that have tested have payload mass less than 8000

Success Rate vs. Orbit Type



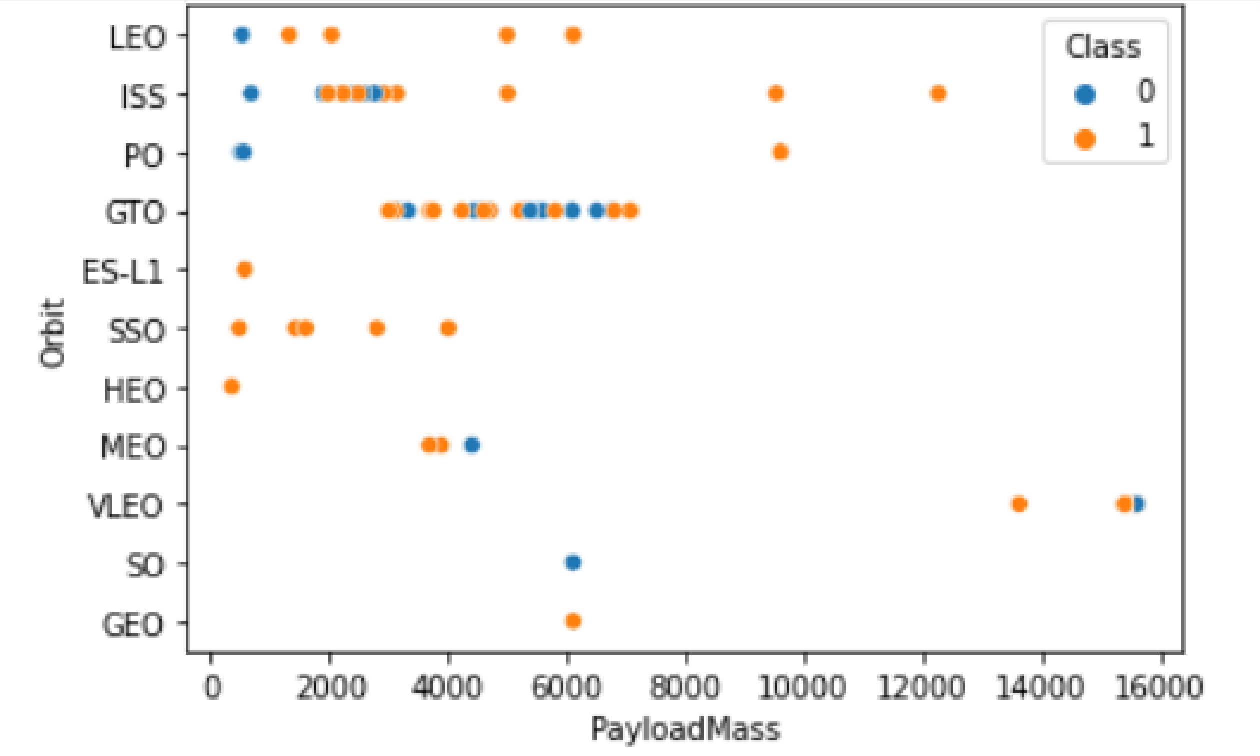
If we look at count plot above we can observe that orbit "VLEO" shows highest success rate. And GTP has success rate of almost 50 % while some of the launch site such as ESL1 and SSO Orbit have been sent lesser rockets but success rates are 100 %

Flight Number vs. Orbit Type



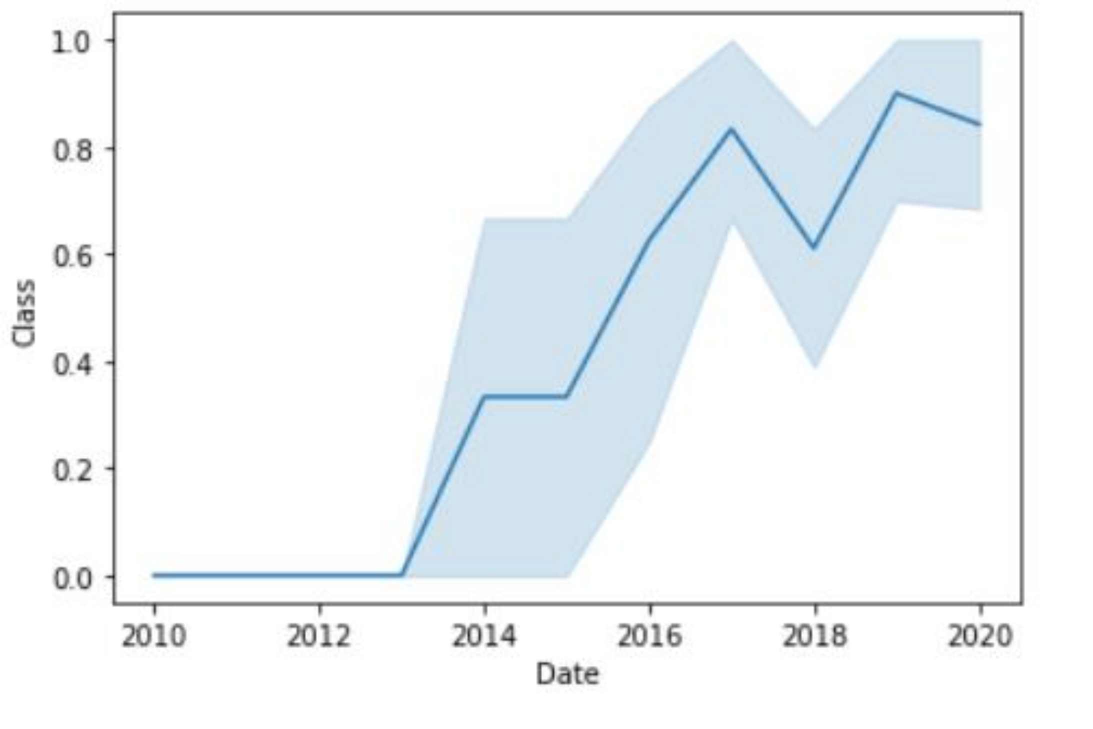
You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



VLEO orbit have been tested for heaviest payload mass, and GTO orbit the pay load mass sent is between 2500 to 7000, while there is no particular trend in success rate here.

Launch Success Yearly Trend



- observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

SQL Query : select distinct Launch_Site from SpaceXTable

Unique Launch Sites
CCAFS LC-40
CCAFS SLC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Success rates analysis will be done for this launch sites individually later
Here function worth notice is DISTINCT

Launch Site Names Begin with 'CCA'

Query : Select TOP 5 * from SpaceXTable Where Launch_Site Like ‘CCA%’

	Date	Time_UTC	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	19-02-2017	2021-07-02 14:39:00.0000000	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
1	16 03 2017	2021 07 02 06:00:00.0000000	F9 FT B1030	KSC LC 39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2	30-03-2017	2021-07-02 22:27:00.0000000	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
3	01-05-2017	2021-07-02 11:15:00.0000000	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
4	15-05-2017	2021-07-02 23:21:00.0000000	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Use of function like is to find out the rows which has similar to “CCA%” string and % sign sets the limitation for search

Total Payload Mass

- Total Payload Mass was coming out to be 45596

query

Select SUM(Paylod_mass_KG_) TotalPayloadMass From SpaceXTable where Customer = ‘NASA(CRS)’

	Total Payload Mass
0	45596

Average Payload Mass by F9 v1.1

- Query : select AVG(Payload_mass_kg_) AveragePayloadMass From SpaceXTable where Booster_Version = 'F9 v1.1'

	Average Payload Mass
0	2928

First Successful Ground Landing Date

Query

Select Date SLO form SpaceXTable where Landing_Outcome = ‘Success(ground pad)‘

	Date which first Successful landing outcome in drone ship was acheived.
0	06-05-2016

The first success ful launch date for ground pad landing was 06-05-2016

Successful Drone Ship Landing with Payload between 4000 and 6000

- Select Booster_Version from spaceXTable Where Landing_Outcome = 'Success (Ground pad)' And Payload_Mass_kg Between 4000 and 6000

	Date which first Successful landing outcome in drone ship was acheived.
0	F9 FT B1032.1
1	F9 B4 B1040.1
2	F9 B4 B1043.1

Between function can be used to set variable selection with in limit

Total Number of Successful and Failure Mission Outcomes

- SELECT (SELECT Count(Mission_Outcome)) from SpaceXTable where Mission_Outcome Like ‘%Success%’ as Successful_Mission_Outcomes, (Select Count(Mission_Outcome)) from SpaceXTable Where Mission_Outcome Like “%Failure%” as Failure_Mission_Outcomes

	Successful_Mission_Outcomes	Failure_Mission_Outcomes
0	100	1

Boosters Carried Maximum Payload

Query

Select payload_mass from spaceXTable order by desc limit 1

OR

Select MAX(Payload_mass) form SpaceXTable limit 1

	Booster_Version	Maximum Payload Mass
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
...	...	...
92	F9 v1.1 B1003	500
93	F9 FT B1038.1	475
94	F9 B4 B1045.1	362
95	F9 v1.0 B0003	0
96	F9 v1.0 B0004	0

97 rows × 2 columns

2015 Launch Records

- Query

Select * from SpaceXTable where date between 01-01-2015 and 01-01-2016

Month	Booster_Version	Launch_Site	Landing_Outcome
January	F9 FT B1029.1	VAFB SLC-4E	Success (drone ship)
February	F9 FT B1031.1	KSC LC-39A	Success (ground pad)
March	F9 FT B1021.2	KSC LC-39A	Success (drone ship)
May	F9 FT B1032.1	KSC LC-39A	Success (ground pad)
June	F9 FT B1035.1	KSC LC-39A	Success (ground pad)
June	F9 FT B1029.2	KSC LC-39A	Success (drone ship)
June	F9 FT B1036.1	VAFB SLC-4E	Success (drone ship)
August	F9 B4 B1039.1	KSC LC-39A	Success (ground pad)
August	F9 FT B1038.1	VAFB SLC-4E	Success (drone ship)
September	F9 B4 B1040.1	KSC LC-39A	Success (ground pad)
October	F9 B4 B1041.1	VAFB SLC-4E	Success (drone ship)
October	F9 FT B1031.2	KSC LC-39A	Success (drone ship)
October	F9 B4 B1042.1	KSC LC-39A	Success (drone ship)
December	F9 FT B1035.2	CCAFS SLC-40	Success (ground pad)

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

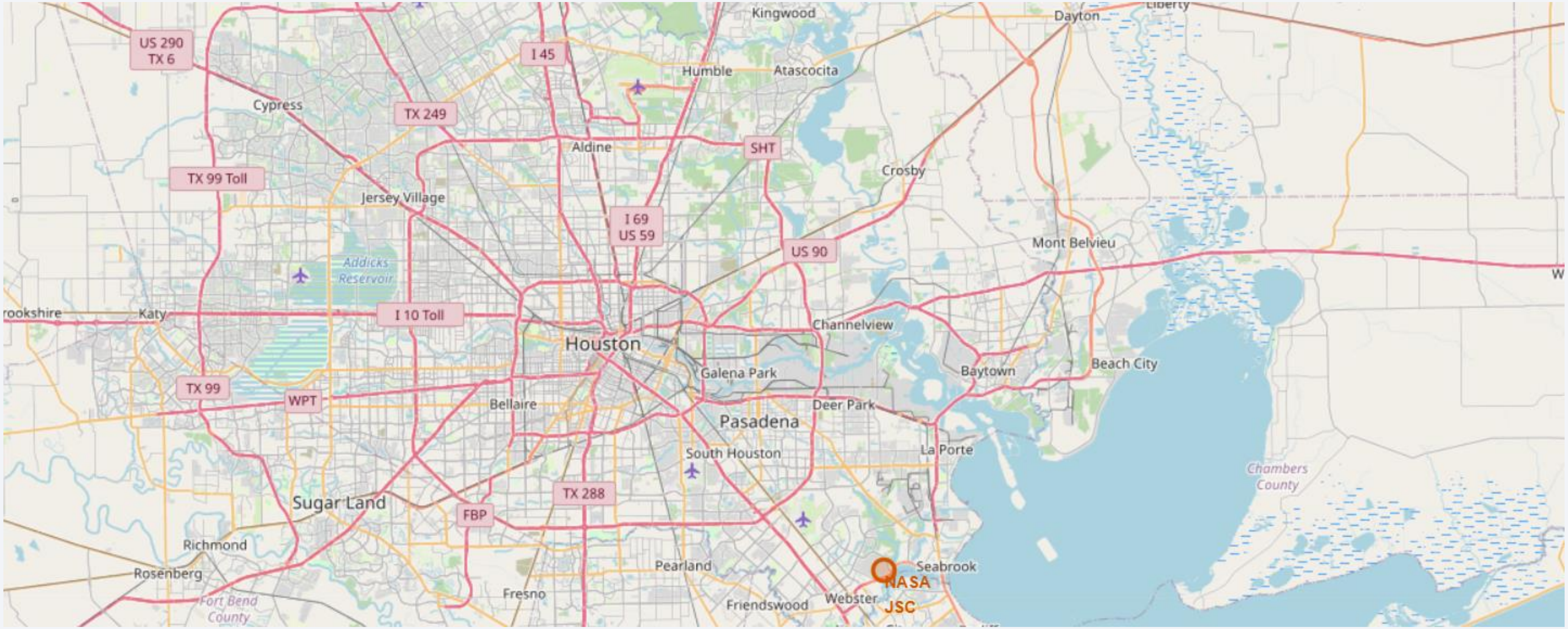
- Select LandingOutcome from SpaceXTable where Date Between 2010-06-04 and 2017-03-20

	Successful Landing Outcomes Between 2010-06-04 and 2017-03-20
0	34

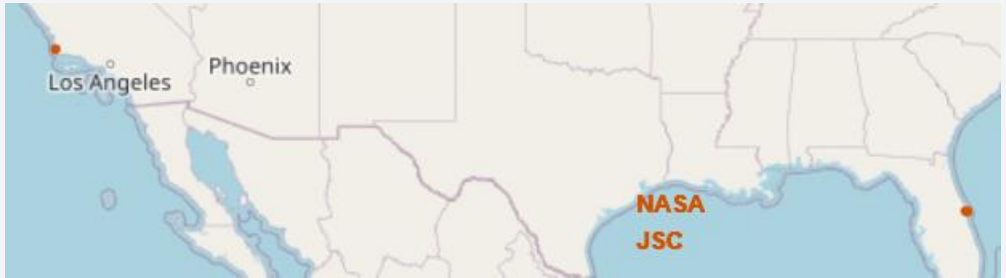
Section 3

Launch Sites Proximities Analysis

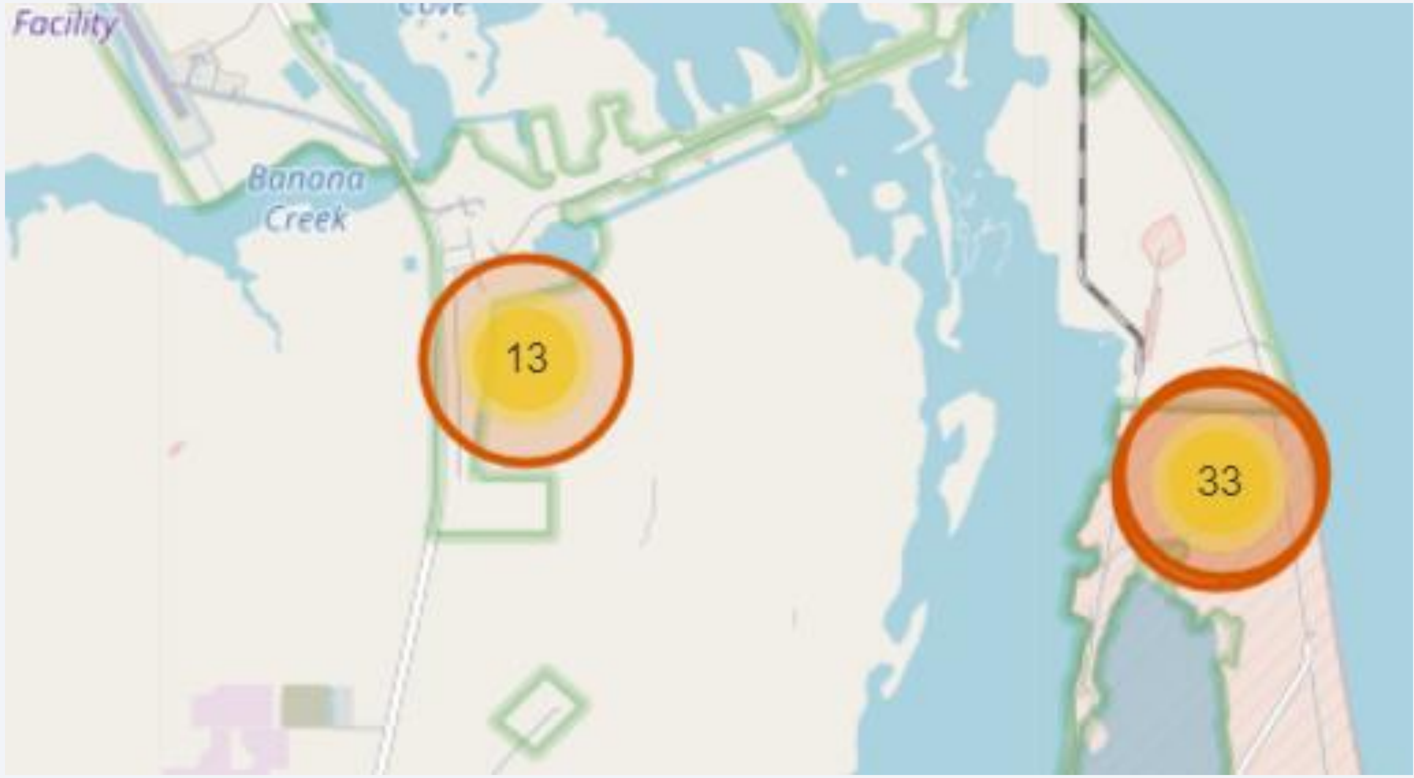
Initial Site Map Object



Space Centre Zoomed IN

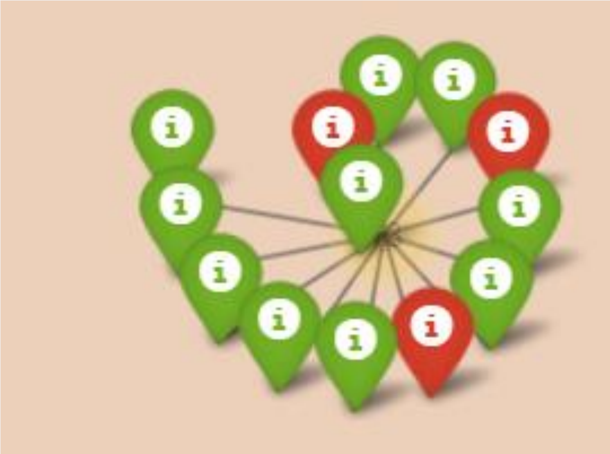


<Launch Site Marking With Number of Launches>



Location of Two Distinct Launch Sites

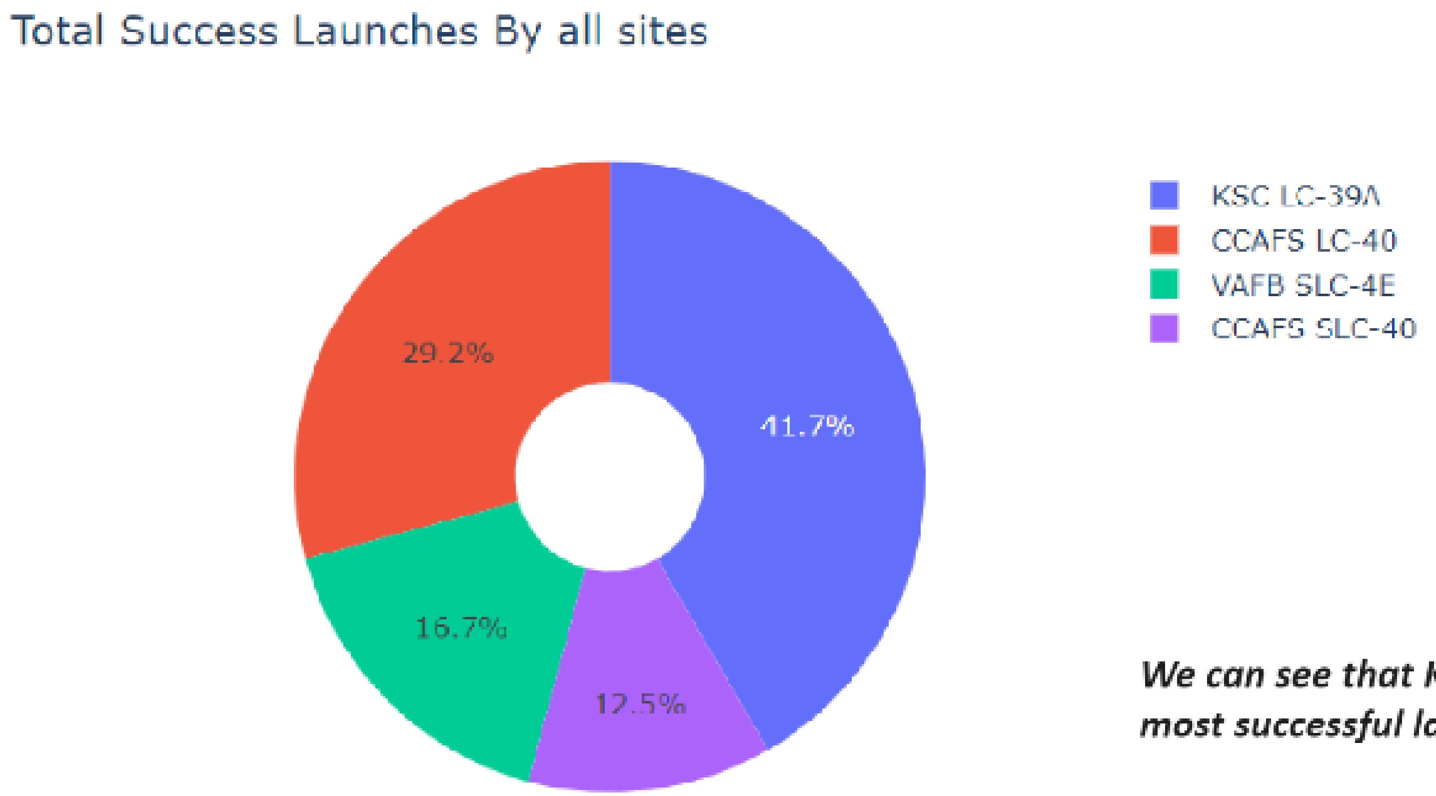
<Successful Launch Depiction using Marker Cluster>



Section 4

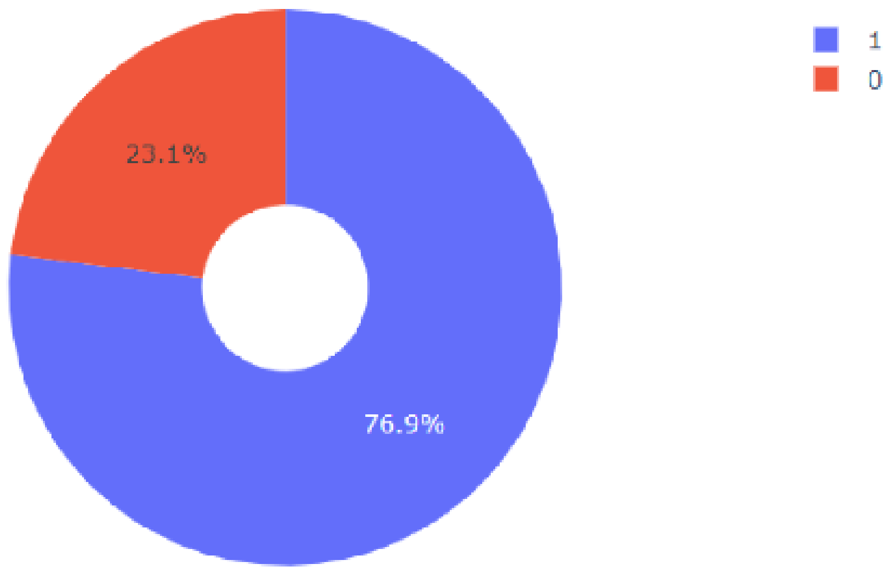
Build a Dashboard with Plotly Dash

<Plotly Dash board Success by Site>



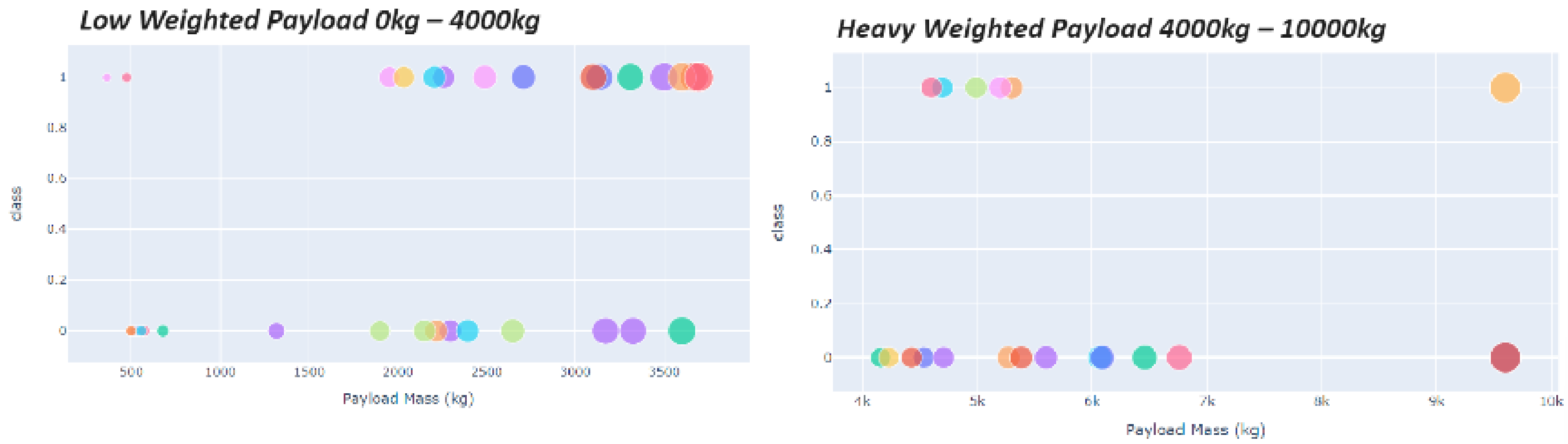
We can see that KSC LC-39A had the most successful launches from all the sites

<Pie Chart for KSC LC-39A Launch pad>



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

<Two Payload Category using Plotly Dash>



We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

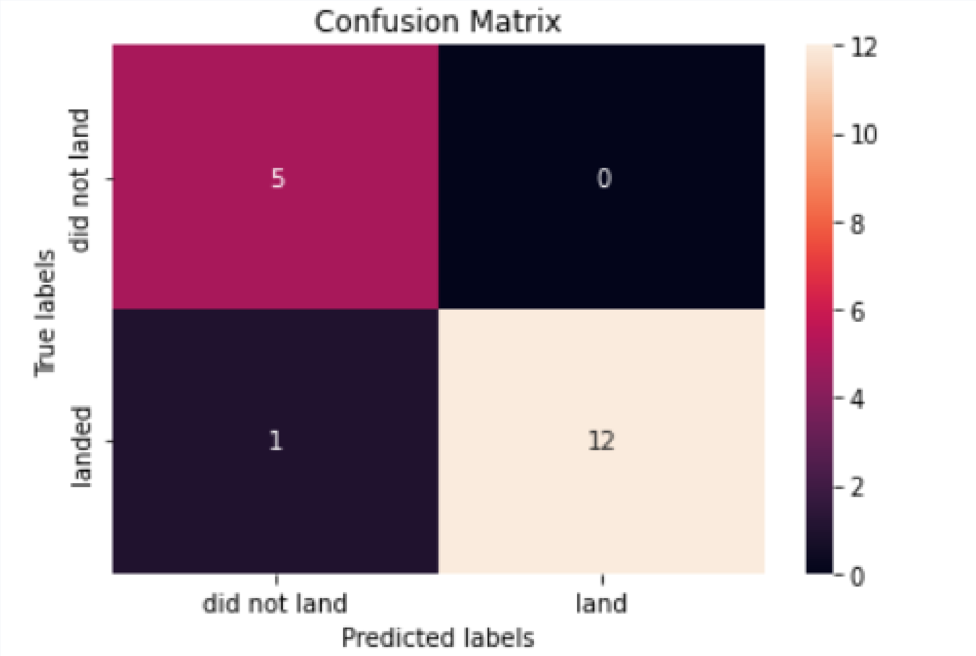
Predictive Analysis (Classification)

Classification Accuracy

- As evident from the plot that Logistics Regression and SVM has best accuracies

Confusion Matrix

- Logistics Regression Model



- Support Vector Regression

Conclusions

- Best performing model were logistics regression, SVM with 94% accuracy by both models while decision tree and knn gave accuracies of 72% and 88.88% respectively
- Few launch sites such as KSC LC-39A have seen better success rate in Past
- Orbits GEO,HEO and SSO, VLEO has better success rate
- This data driven models can be used for decision making purpose as the performance was satisfactory

Appendix

Thank you!