IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Swaroop Ratan Karmankar
25.02.2023

Outline

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

Executive Summary

- **Summary of methodologies**
  - ➢ Data Collection
  - ➢ Data Wrangling
  - ➢ EDA with SQL
  - ➢ EDA with Data Visualization
  - ➢ Using folium to build interactive map
  - ➢ Using plotly to build interactive dashboard
  - ➢ Prediction using ML algorithms
- **Summary of all results**
  - Building Interactive Dashboards
  - Predicting the success of the Space X

Introduction

- **Project background and context**

  The project was to predict whether the first stage landing of the Falcon 9 will be successful.

  We can determine the cost of the launch if we found out if the first stage will be successful or not.

- **Problems you want to find answers**

  What are the favorable conditions for the Falcon 9 first stage landing to be successful.

  What feature makes the best affect on the first stage success landing.

  What are the different factors and which launch sites and other features has the relationship with the landing outcome.

Section 1

Methodology

Methodology

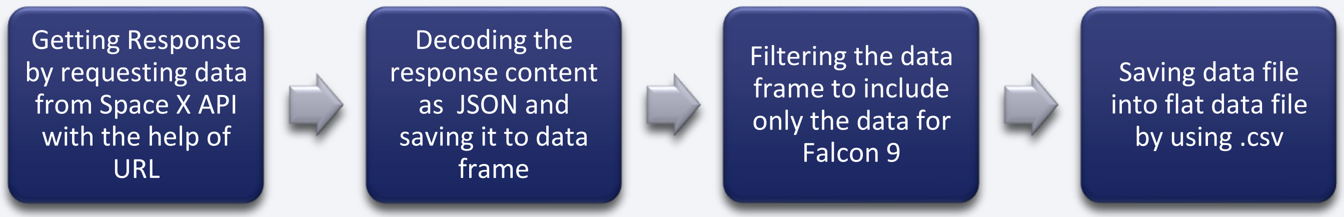
Executive Summary

- Data collection methodology:
    - Space X Rest Api
    - Web Scraping from Wikipedia
- Perform data wrangling
    - Describe how data was processed
- 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

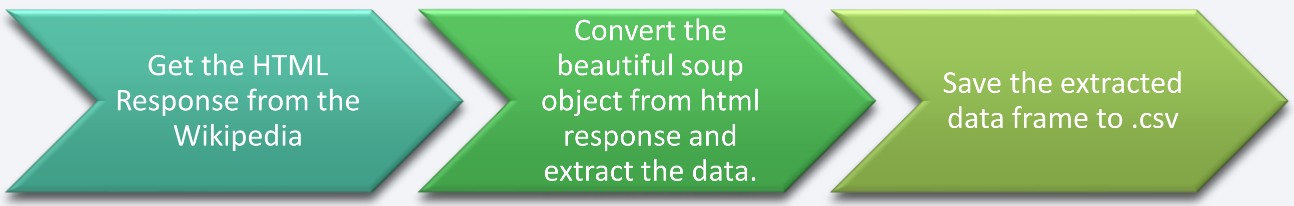
Data Collection

Using REST API:

Collected the data set from the Space X Rest API where in we got the multiple information including time, launches, launch pad, rockets, success, failures, crew, ships, booster version, payload, reused, serial number etc.



Using Web Scraping:



Surfing through the Wikipedia we found out the table which was providing the necessary information about the Falcon 9.
We collected the information from here through Web Scraping.

Data Collection – SpaceX API

GitHub - Data Collection Using API

1 . Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

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

2. Decoding response content and converting into Data Frame

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

3. Applying filtering functions

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

4. Creating dictionary and converting it to data frame

```
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}
```

```
df = pd.DataFrame.from_dict(launch_dict)
```

5. Filtering the dataset for only Falcon 9

```
data_falcon9 = df[df['BoosterVersion'] == 'Falcon 9']
```

6. Saving the dataset as .csv

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

Data Collection - Scraping

[GitHub - Data Collection Using Web Scraping]

1. Getting Response from URL

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(static_url)
```

2. Creating Beautiful Soup Object

```
soup = BeautifulSoup(response.text)
```

3. Finding the particular table using soup object

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

```
first_launch_table = html_tables[2]
```

4. Finding column names

```
column_names = []
for i in first_launch_table.find_all('th'):
    x = extract_column_from_header(i)
    if x is not None and len(x):
        column_names.append(x)
```

5. Creating dictionary

```
the_launch_dict= dict.fromkeys(column_names)

del the_launch_dict['Date and time ( )']

the_launch_dict['Flight No.'] = []
the_launch_dict['Launch site'] = []
the_launch_dict['Payload'] = []
the_launch_dict['Payload mass'] = []
the_launch_dict['Orbit'] = []
the_launch_dict['Customer'] = []
the_launch_dict['Launch outcome'] = []
# Added some new columns
the_launch_dict['Version Booster']=[]
the_launch_dict['Booster landing']=[]
the_launch_dict['Date']=[]
the_launch_dict['Time']=[]
```

6. Appending the information in the dictionary keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into the_launch_dict with key `Flight No.`
            the_launch_dict['Flight No.'].append(flight_number)
#             print(flight_number)
```

7. Converting dictionary to data frame

```
df=pd.DataFrame(the_launch_dict)
```

8. Saving the data frame to csv

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

Data Wrangling

[GitHub Link For Data Wrangling]

We performed EDA first and then convert the outcomes depending on their basis such as True * means landed successfully and False * means not landed successfully into training labels as 1 and 0 respectively.

We have performed different steps :

1. Calculated number of launches from each sites.
2. Calculated number and occurrence of each orbit.
3. Calculate the number and occurence of mission outcome per orbit type
4. Create a landing outcome label from Outcome column

Hence forth we downloaded the dataset into csv.

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

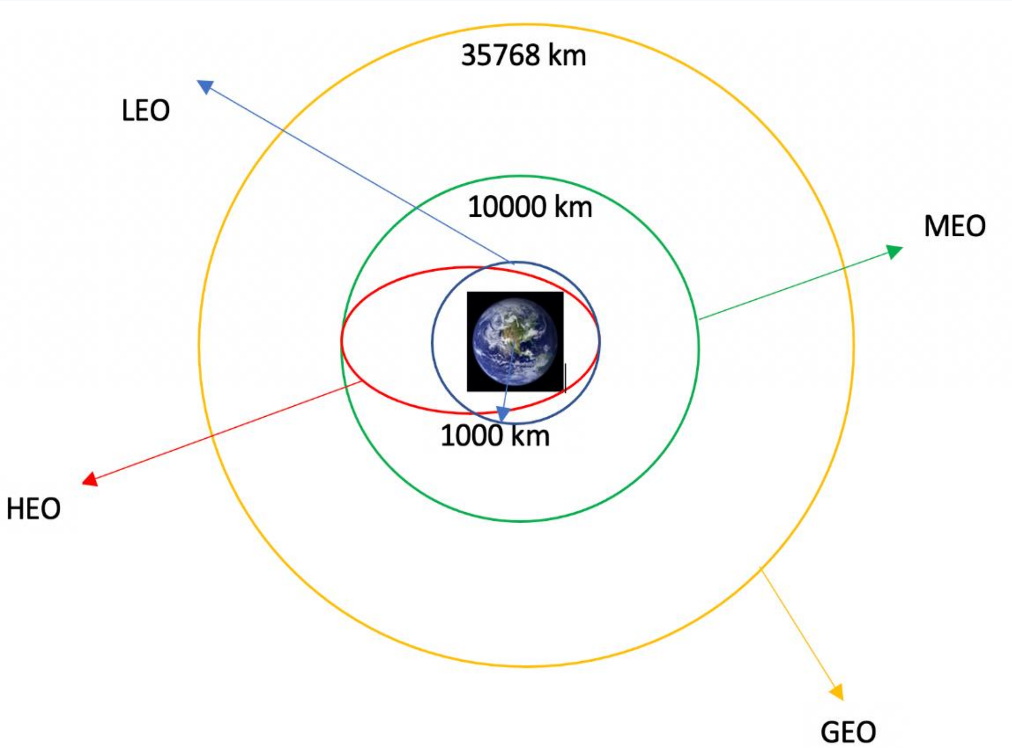


Fig. Some common orbit types used in SpaceX

EDA with Data Visualization

We have drawn some graphs such as Scatter plots, bar charts and line charts to visualize how the features are related to each other.

EDA with Data Visualization

Drawn scatter plots between:

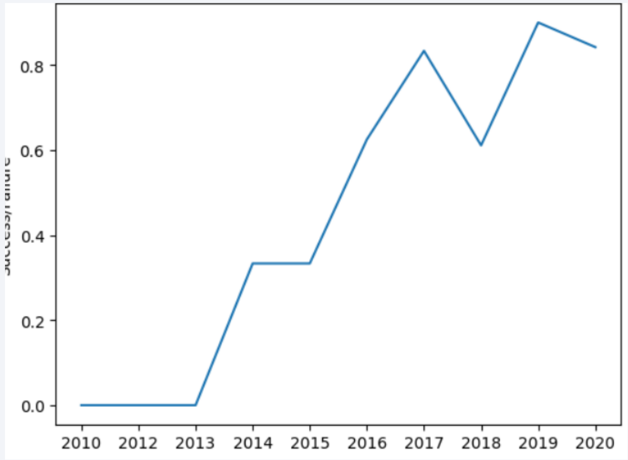
- Flight Number and Launch Site
- Payload and Launch Site
- Flight Number and Orbit Type
- Payload and Orbit Type

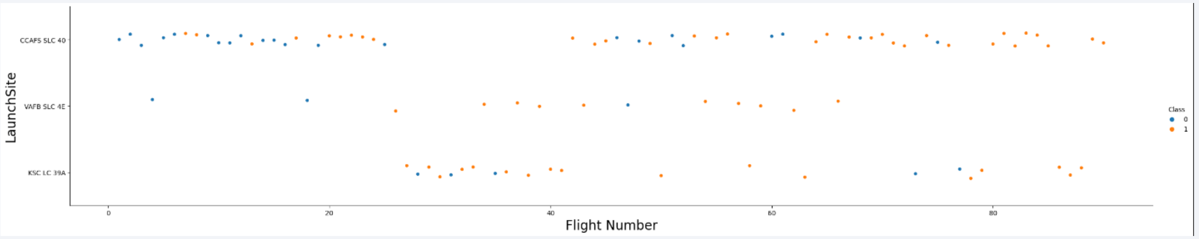
Drawn line chart between:

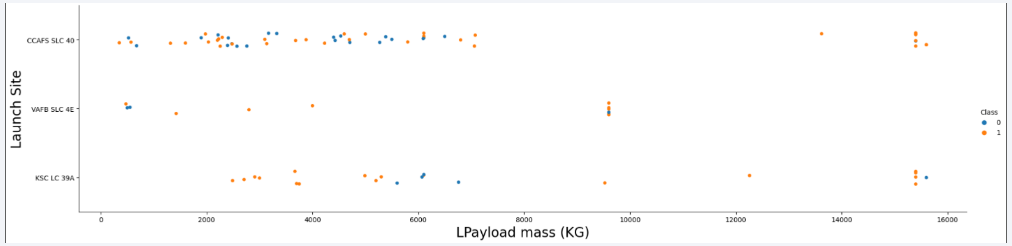
- Success rate vs Year

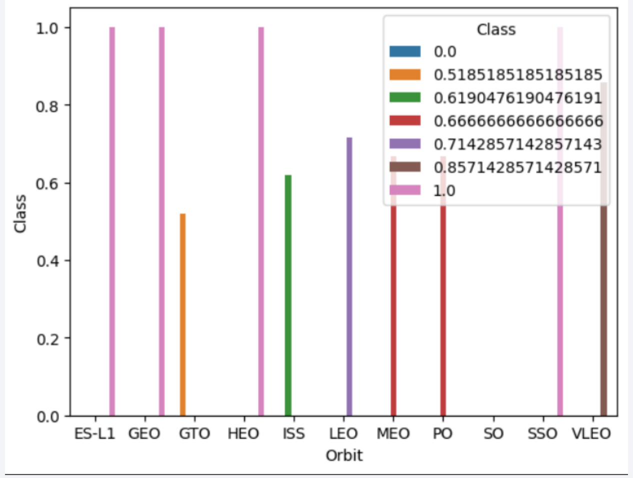
Drawn bar chart between:

- Success rate vs Orbit Types









EDA with SQL

EDA with SQL

We have loaded the data into the PostgreSQL in the notebook itself.

Later on we performed SQL queries on the dataset to get the answer of some of the questions such as:

- Names of Unique Launch Sites
- 5 records where launch sites begin with 'CCA'
- Total Payload mass carried by boosters launched by NASA
- Total Payload Mass carried by booster version F9 V1.1
- Date when the first successful landing happened in the ground pad.
- Names of booster which have success in drone ship and payload mass greater then 4000 but less than 6000.
- Total number of successful and failure mission outcomes.
- Name of booster versions which have carried maximum payload mass.
- Rank the count of successful landing outcome between a specific date.

```
%%sql

select min(date) as First_successfull_Landing, "landing _Outcome" from spacextbl
where "Landing _Outcome" = 'Success (ground pad)';
```

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

First_successfull_Landing	Landing _Outcome
01-05-2017	Success (ground pad)

```
%%sql

SELECT Booster_Version from SPACEXTBL
WHERE Payload_Mass__Kg_ = (Select max(Payload_Mass__Kg_) from spacextbl);
```

```
%%sql

select BOOSTER_VERSION, "Landing _Outcome", Payload_mass__kg_ from SPACEXTBL
where "Landing _Outcome"='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

Build an Interactive Map with Folium

GitHub Link to Folium Maps

We marked all launch sites by using the longitude and latitude points of each of the launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.

We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure as the RED markers and 1 for success, which will be the GREEN markers.

Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.

We calculated the distances between a launch site to its proximities using Haversine's formula to find the trend pattern

We answered some question for instance:

- * Are launch sites in close proximity to railways? NO
- * Are launch sites in close proximity to highways? NO
- * Are launch sites in close proximity to coastline? YES
- * Do launch sites keep certain distance away from cities? YES

.

Build a Dashboard with Plotly Dash

GitHub Link for Plotly Dash

We built an interactive dashboard with Plotly dash

We plotted pie charts showing the total launches by a certain sites

We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

Predictive Analysis (Classification)

GitHub Link for ML predictions analysis

Building the model:

1. Loading the dataset using the pandas and Numpy
2. Transforming the dataset
3. Splitting the dataset using the train_test_split
4. Deciding the type of algorithm which we want to use
5. Set our parameters and algorithm to GridSearchCV and then fitting the dataset into GridSearchCV objects and training them

Evaluating Model:

- We are using different metrics to evaluate the model, such as checking the accuracy or f1-score or Jaccard-index of the model.
- Plotting confusion matrix for the models

Improving Models:

We use Feature Engineering and Algorithm tuning to improve our models.

Thus we found the best performance classification model by going through all the accuracy and other metrics score. The higher the accuracy the better the model performs.

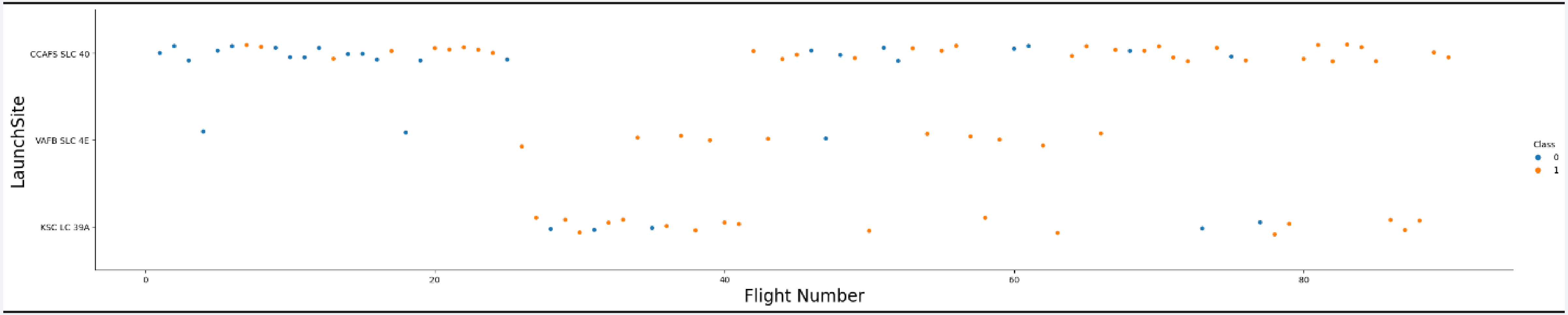
Results

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

Section 2

Insights drawn from EDA

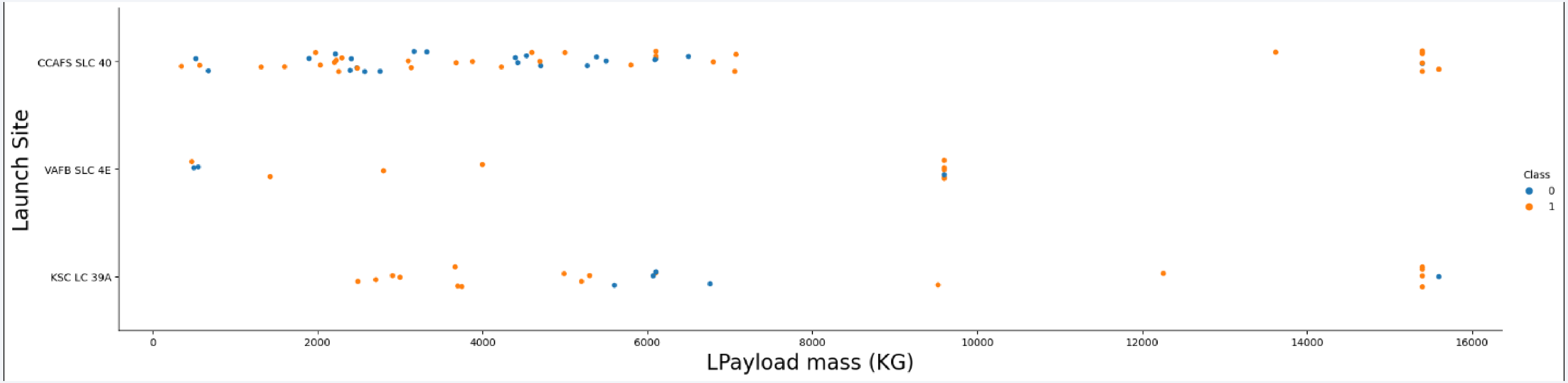
Flight Number vs. Launch Site



It shows that the more the numbers of flights the more the success rates.

Many numbers of flights were launched from CCAPS SLC 40 Launch Sites.

Payload vs. Launch Site



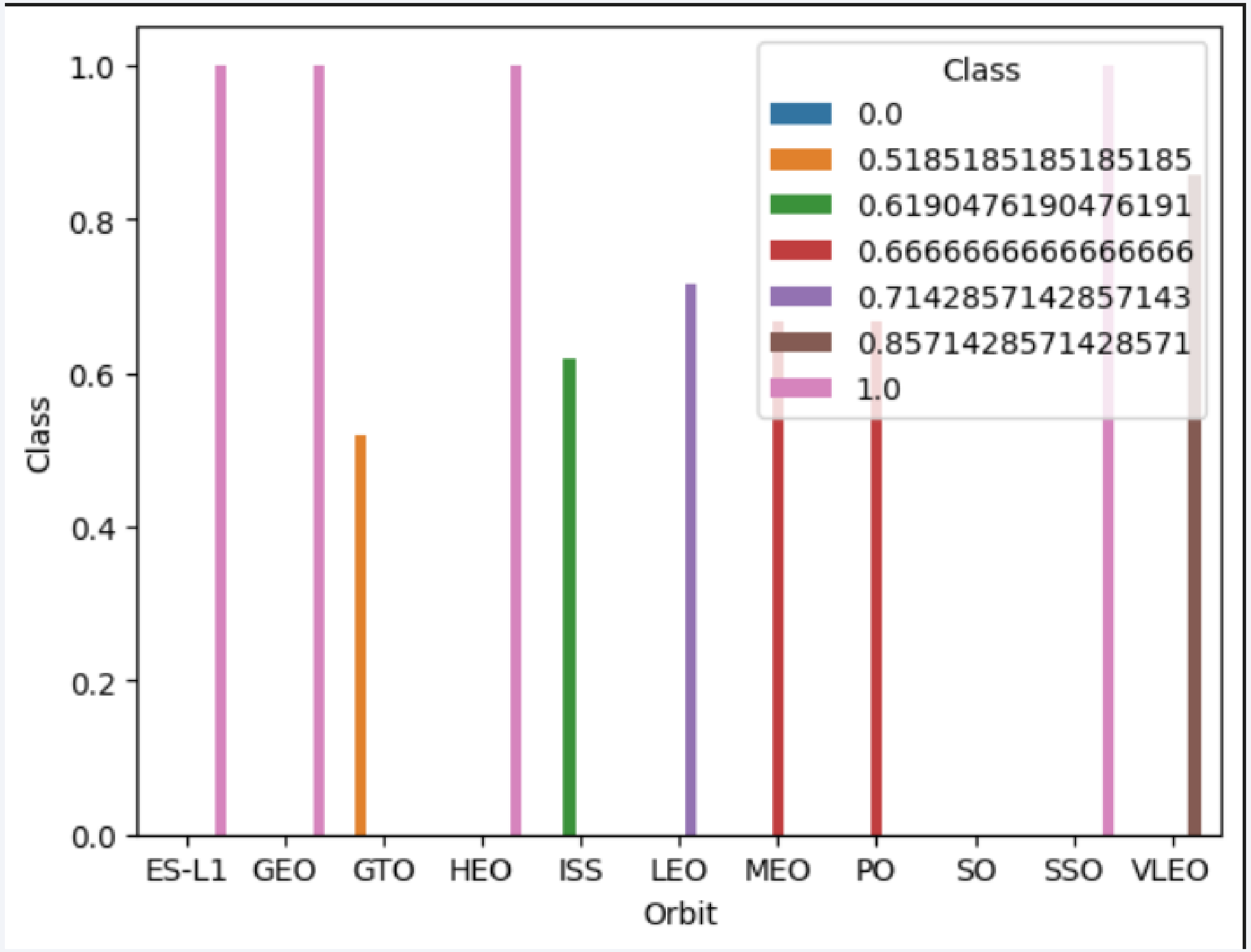
We found that the greater the Payload Mass the greater is the success rate.

But still it is not clear, whether the success rate of Launch site depends on payload mass

Success Rate vs. Orbit Type

The success rate of the orbits ES-L1, GEO, HEO and SSO are higher then the other orbits.

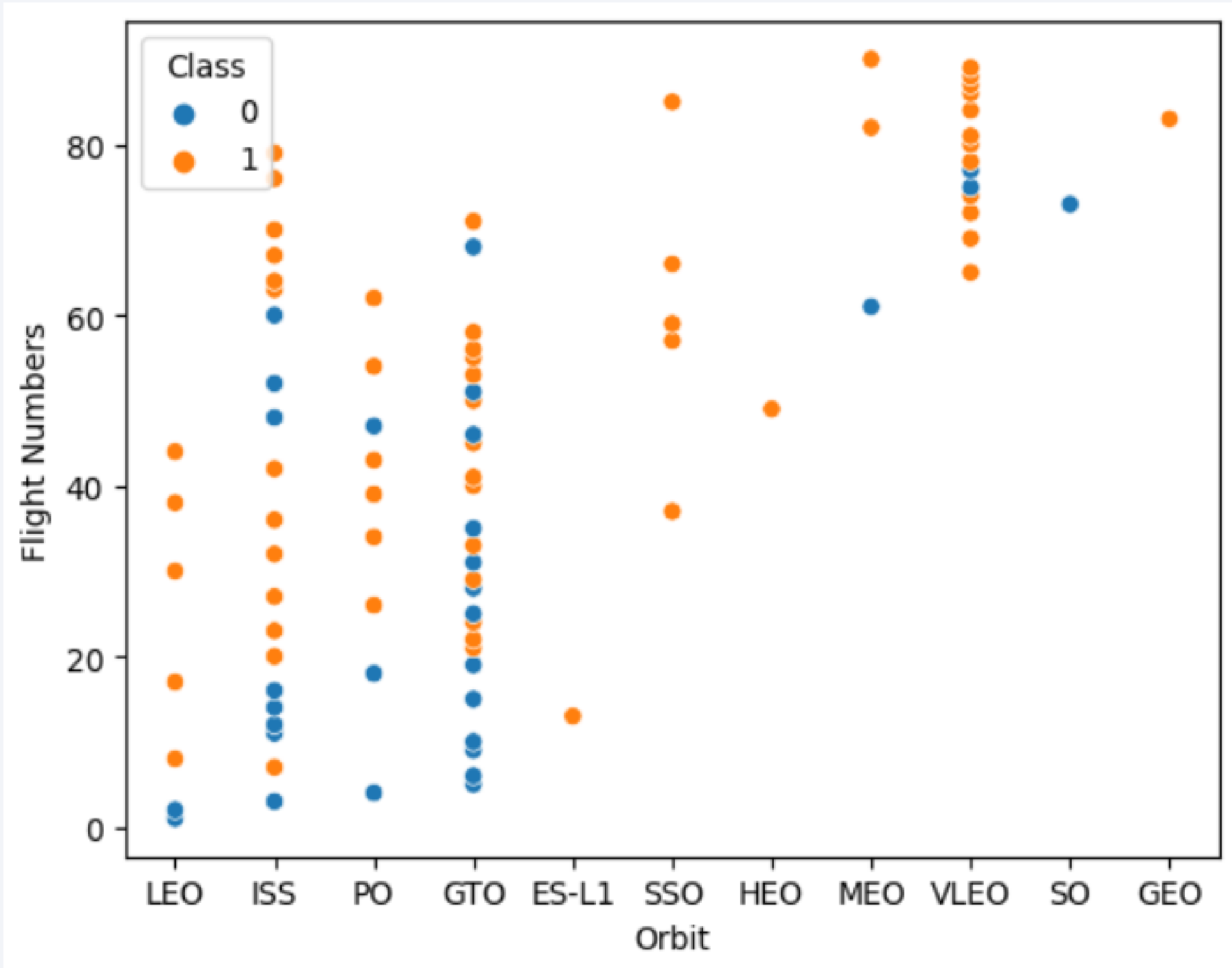
The least 3 orbits with the success rates are ISS, GTO and SO.



Flight Number vs. Orbit Type

From this plot we are able to conclude for some orbits like LEO, number of flights is related to Success appears.

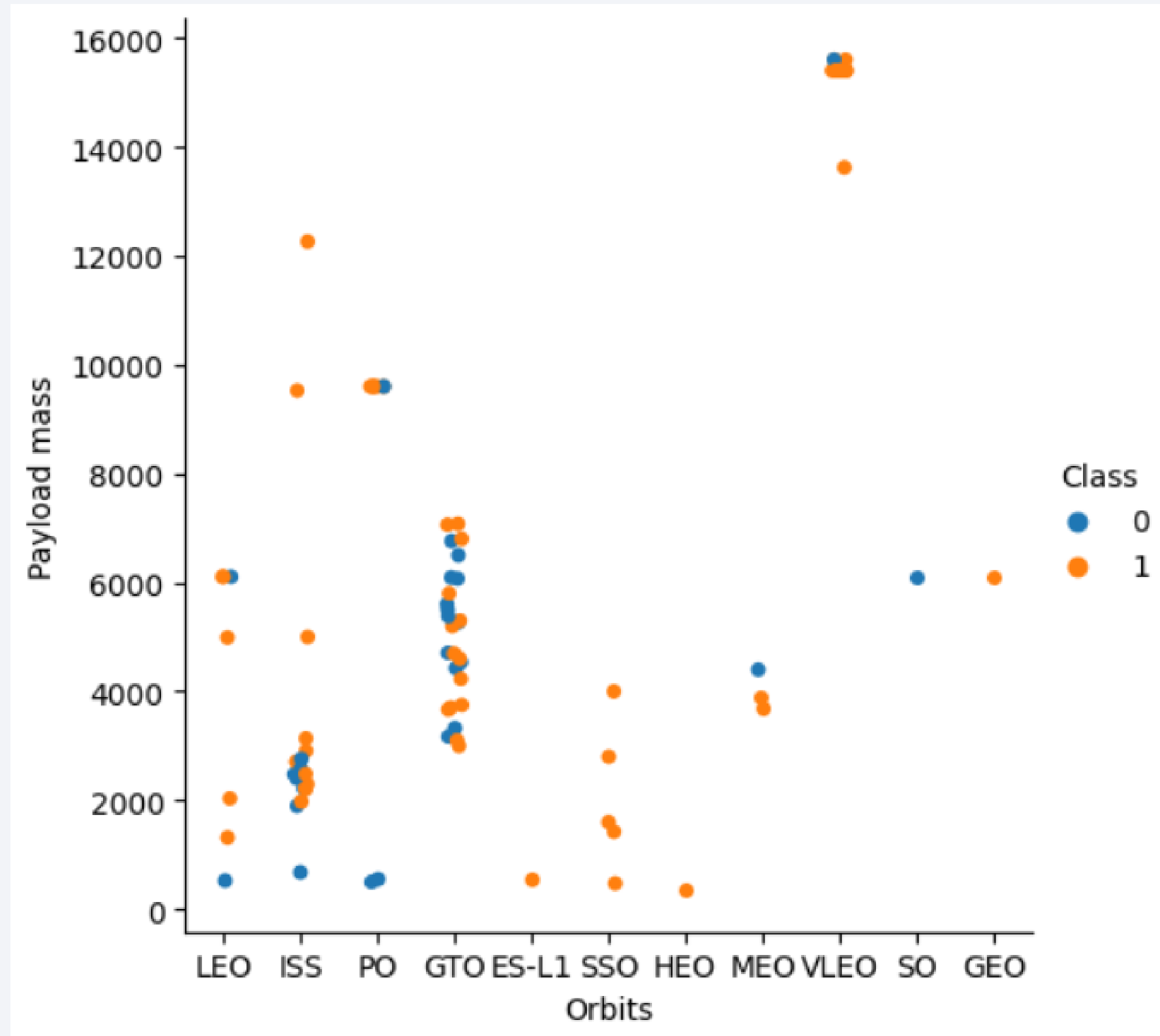
But we can’t see any relationship between the others such as GTO, ISS.



Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

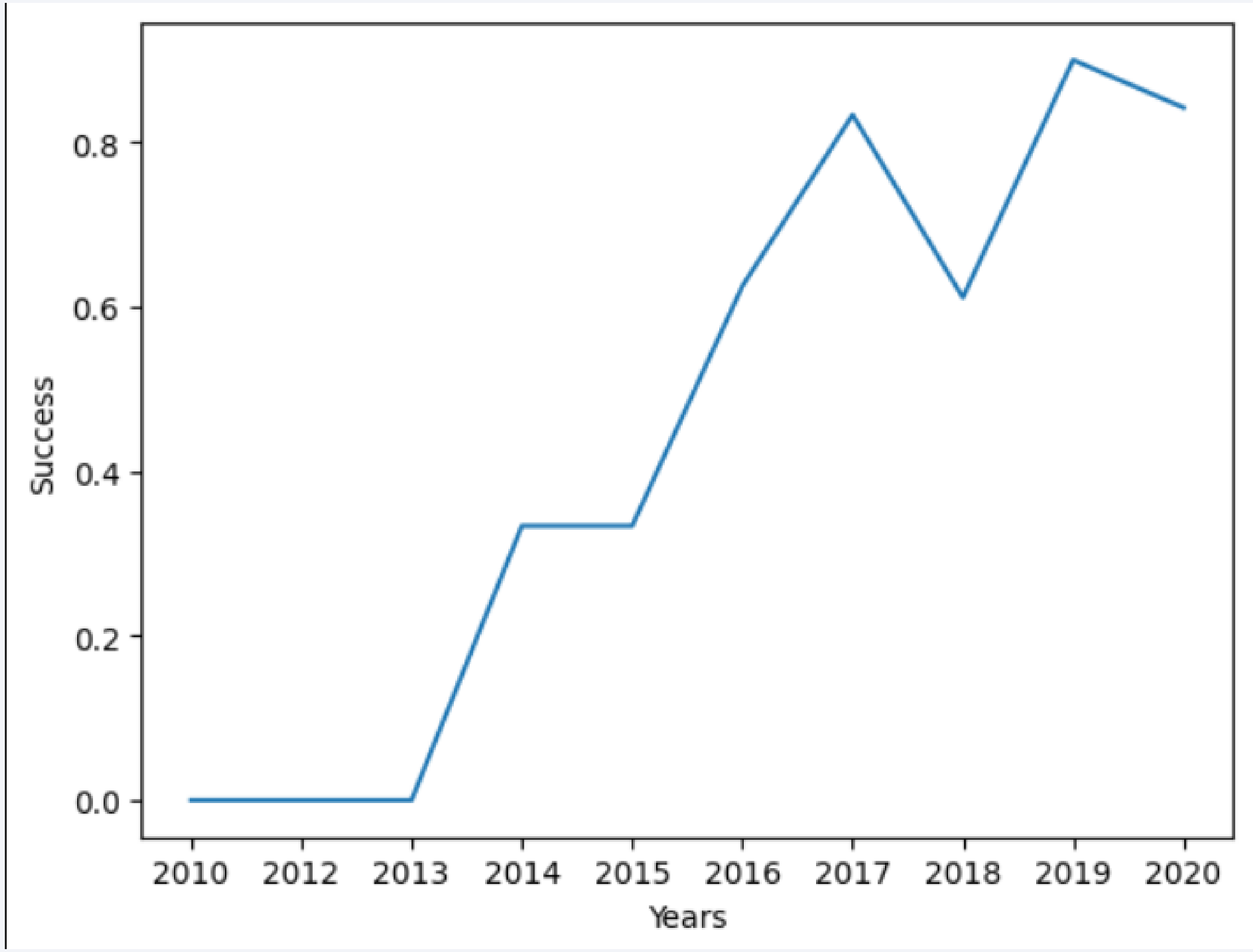
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



Launch Success Yearly Trend

- We can observe the trend line of the success rate through the years.

- The success rate kept on increasing from the year 2013, and kept on rising.



All Launch Site Names

```
%%sql

SELECT DISTINCT Launch_Site from SPACEXTBL;
```

We use the DISTINCT keywords to find the unique names of the Launch sites.

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;
```

We used the '%' character at the end of the string 'CCA' to consider any other character after that. Also we used LIMIT keyword to limit the result to 5.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	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_) as Total_Payload_Mass from SPACEXTBL where Customer = 'NASA (CRS)';
```

We used the SUM aggregate function to total the sum of the payload mass.

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

```
%%sql

SELECT AVG(PAYLOAD_MASS__KG_) AS Average_Payload_mass from SPACEXTBL where Booster_Version = 'F9 v1.1'
```

We used the AVG (average)aggregate function to calculate the average of the payload mass carried by F9 v1.1

Average_Payload_mass
2928.4

First Successful Ground Landing Date

```
%%sql

select min(date) as First_successfull_Landing, "landing _Outcome" from spacextbl
where "Landing _Outcome" = 'Success (ground pad)';
```

We used the MIN() keyword around date to find the minimum of the date and we added the WHERE clause containing the successful ground pad.

First_successfull_Landing	Landing _Outcome
01-05-2017	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql

select BOOSTER_VERSION, "Landing _Outcome", Payload_mass__kg_ from SPACEXTBL
where "Landing _Outcome"='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

We used the WHERE clause to find the success drone ship from the Landing outcome.

Also we used BETWEEN keyword to consider the payload mass between 4000 to 6000.

Booster_Version	Landing _Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

```
%sql select Mission_Outcome, count(Mission_Outcome) from spacextbl group by Mission_Outcome;
```

We used the GROUP keyword to group the data by the Mission Outcome column so as to count. Also we used COUNT keyword to count the numbers of outcomes.

Mission_Outcome	count(Mission_Outcome)
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);
```

We have used the SUBQUERY here to find out the Boosters with the maximum payload mass with the help of the MAX keyword and WHERE clause.

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 substr(Date,4,2) as Month, Booster_Version, Launch_Site, "Landing _Outcome" from spacextbl
where substr(Date, 7, 4)='2015' and "landing _Outcome" = "Failure (drone ship)";
```

We made the use of AND keyword and the SUBSTR() to get the months of the date with WHERE clause.

Month	Booster_Version	Launch_Site	Landing _Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

```
%%sql

select "landing _outcome", count(*) from spacextbl
where "landing _outcome" like "Success%" and Date between '04-06-2010' and '20-03-2017'
group by "Landing _Outcome";
```

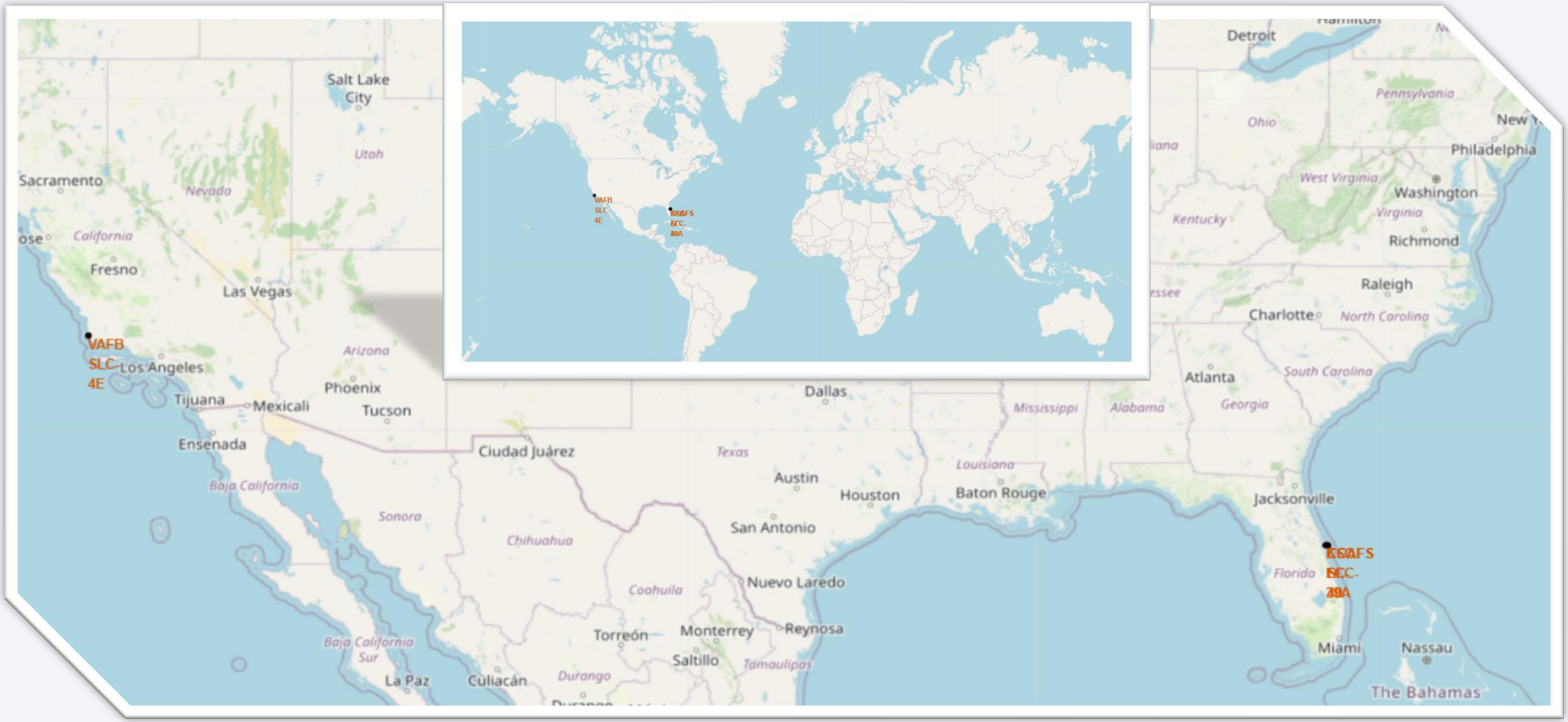
We have used COUNT keywords to count the number of outcomes, then giving the conditions in WHERE clause wherein the success should be selected BETWEEN the dates given, and GROUPING the data by the Landing Outcome column.

Landing _Outcome	count(*)
Success	20
Success (drone ship)	8
Success (ground pad)	6

Section 3

Launch Sites Proximities Analysis

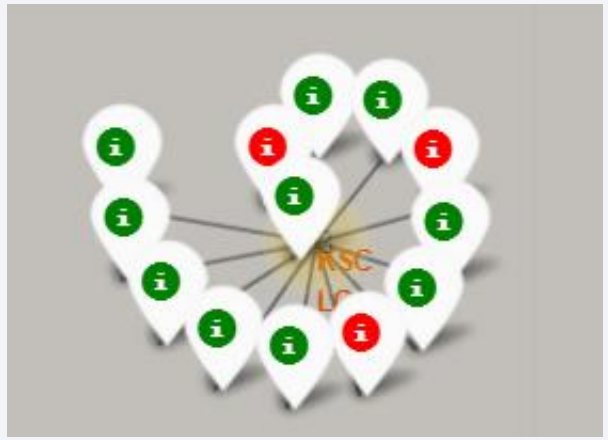
Launch sites Global Map Markers



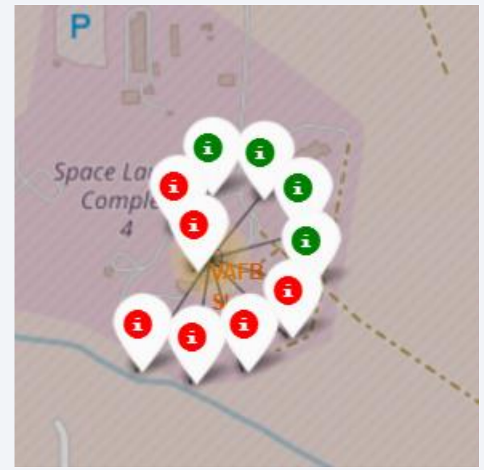
We have mapped the markers for our Launch Sites and we could see it is in **United States of America.**

One of them is situated on in the **Florida** and the other in the **California**.

Markers on each sites



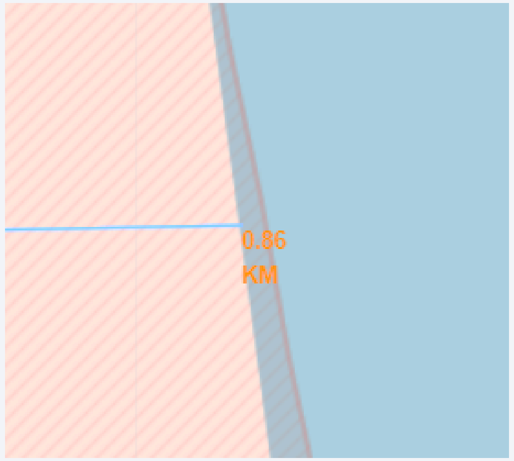
Markers marked on the Florida Site

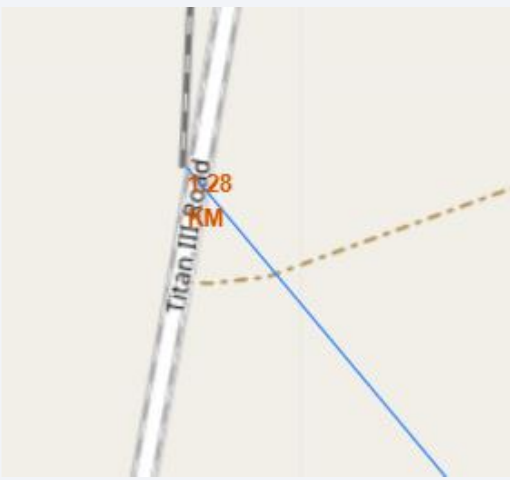


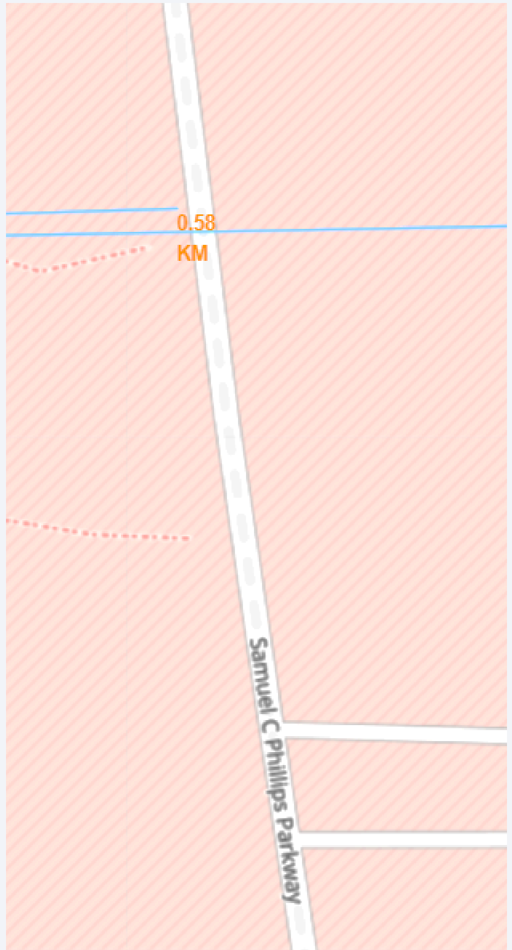
California Site

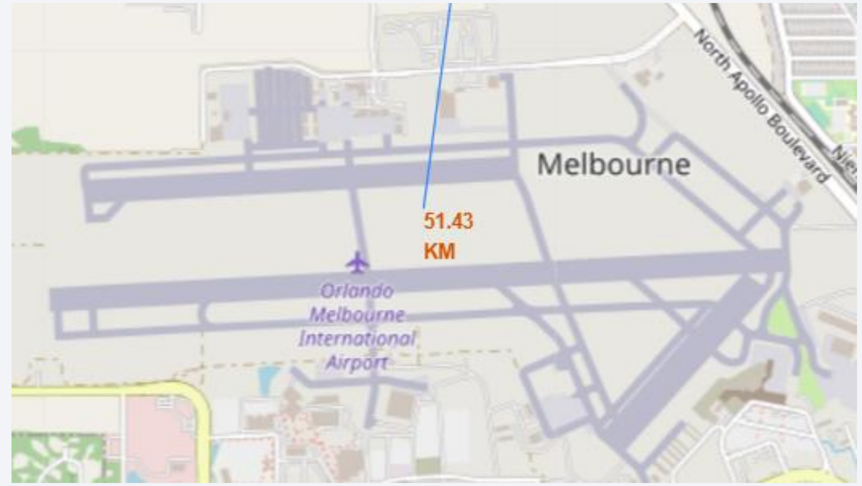
The markers mapped with the Green indicates Success while the Red ones indicate failure on the sites.

Distances measured







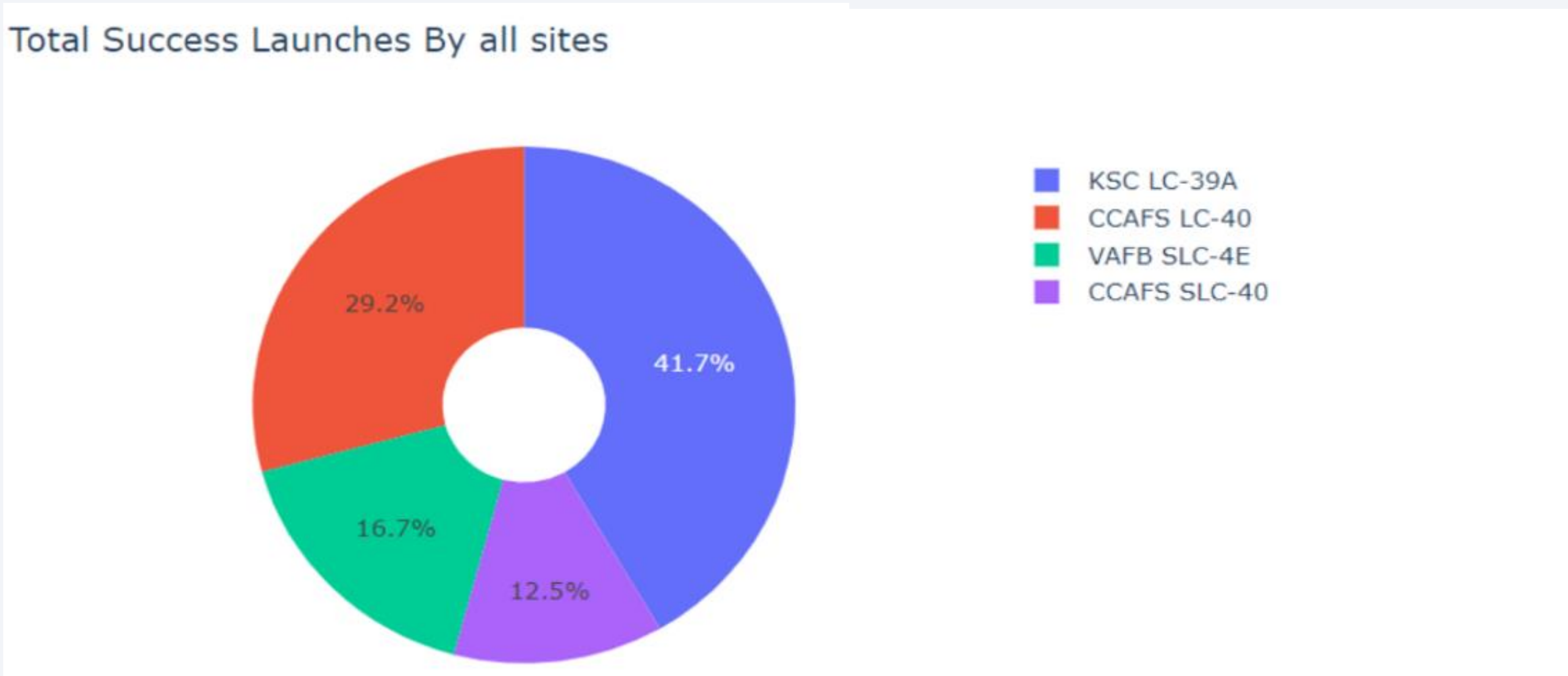


Here are some of the distance calculated from the CCAFS-SLC-40 to the nearest highway, airport, coastline and railway line.

Section 4

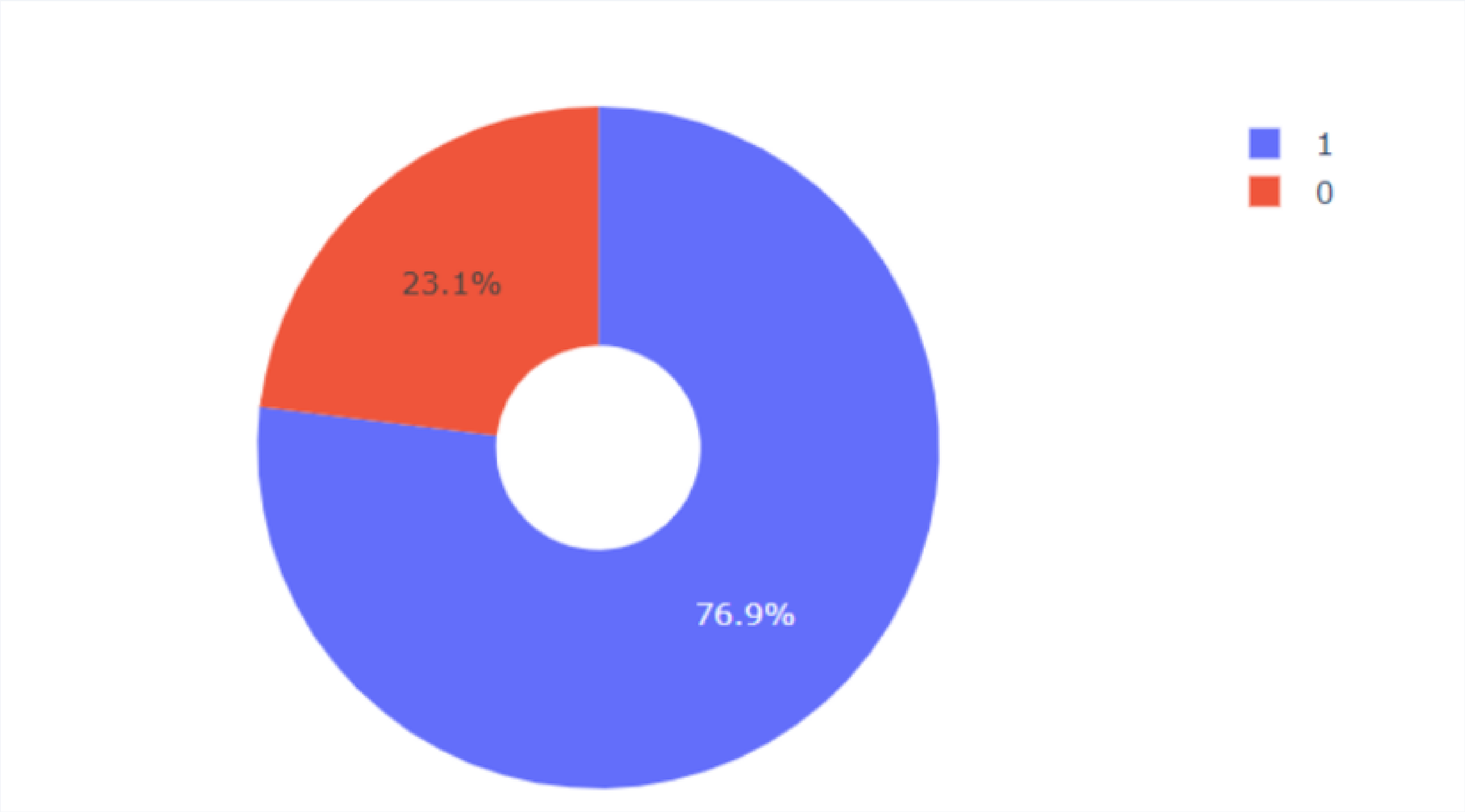
Build a Dashboard with Plotly Dash

Success percentages of each sites



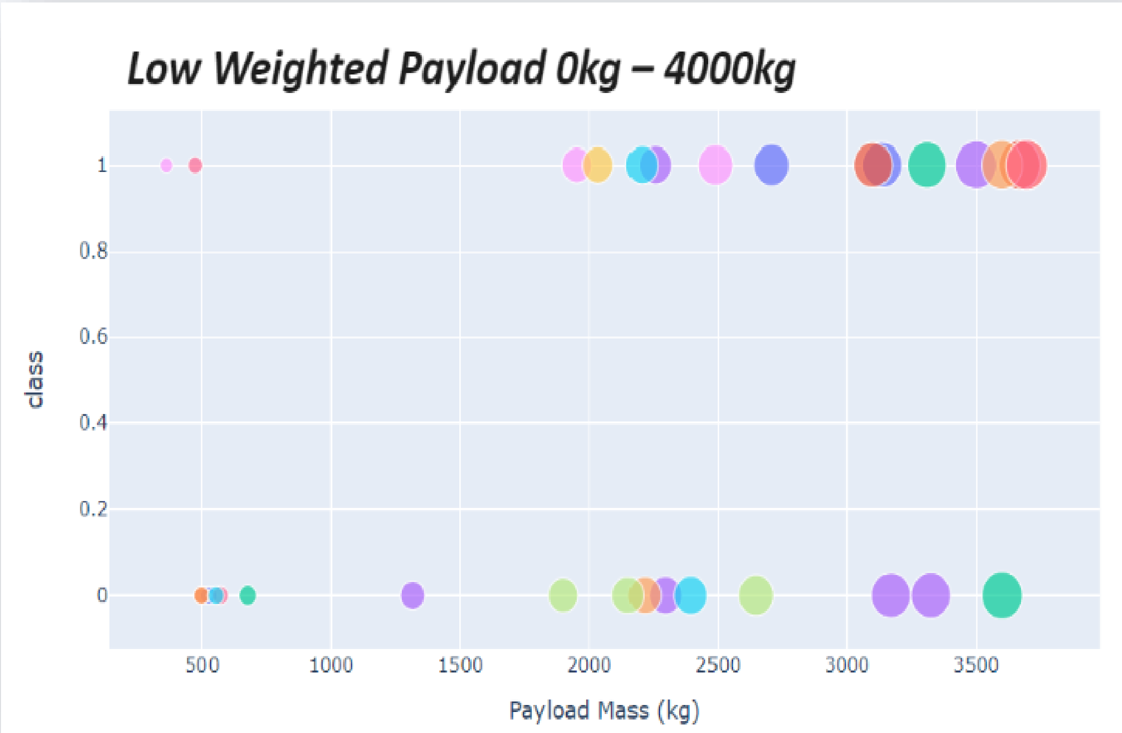
All the unique launch sites which we displayed in the SQL sheet percentages are mentioned here in the plotly dashboard. The highest percentage is 41.7 % of the KSC LC-39A site.

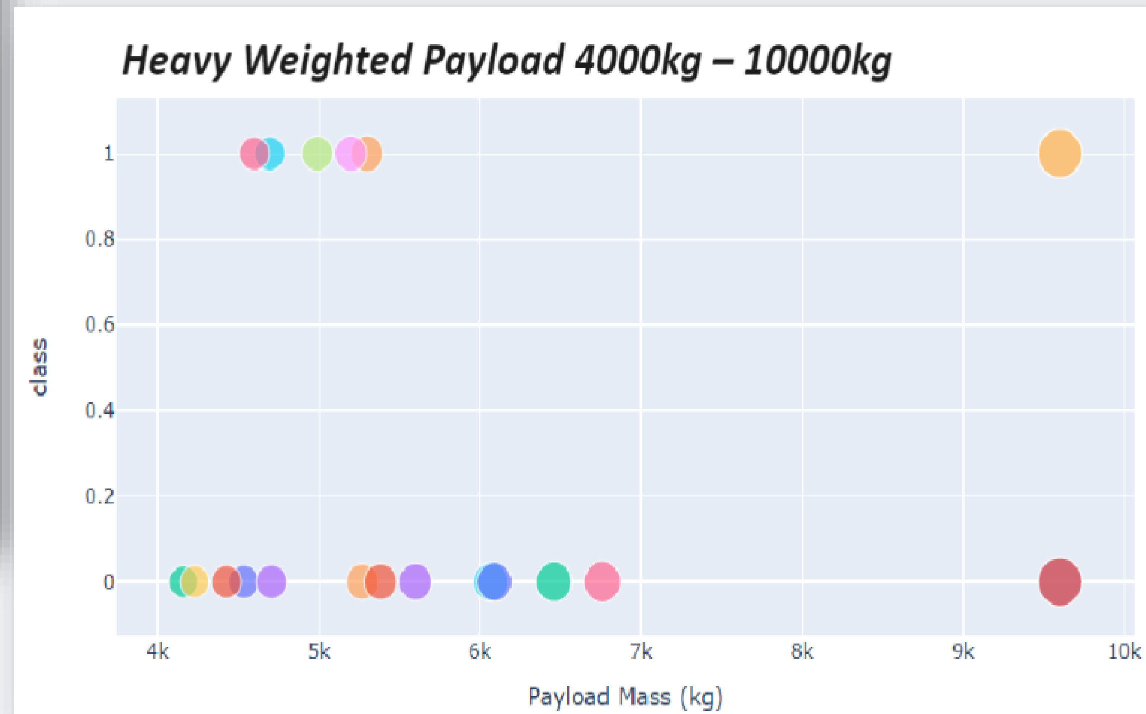
KSCLC-39A success rates ratio



KSC LC-39A site has launched 76.9 % success rate and 23.1 % failure rates.

Payload vs Launch Outcome





From this dashboard we can say that if there is low weight payload then the success rate will be higher.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

We have applied different ML algorithms to develop a model. We will use the following code to determine for which algorithm do we have the maximum accuracy.

```
bestalgorithm = max(models, key=models.get)
```

From the below we can see that we have Decision Tree as its best model with the accuracy score of 87 % wherein max depth = 6.

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

The above score was achieved by using GridSearchCV and hyper tuning the parameters.

Here is the representation of their scores and the algorithms used.

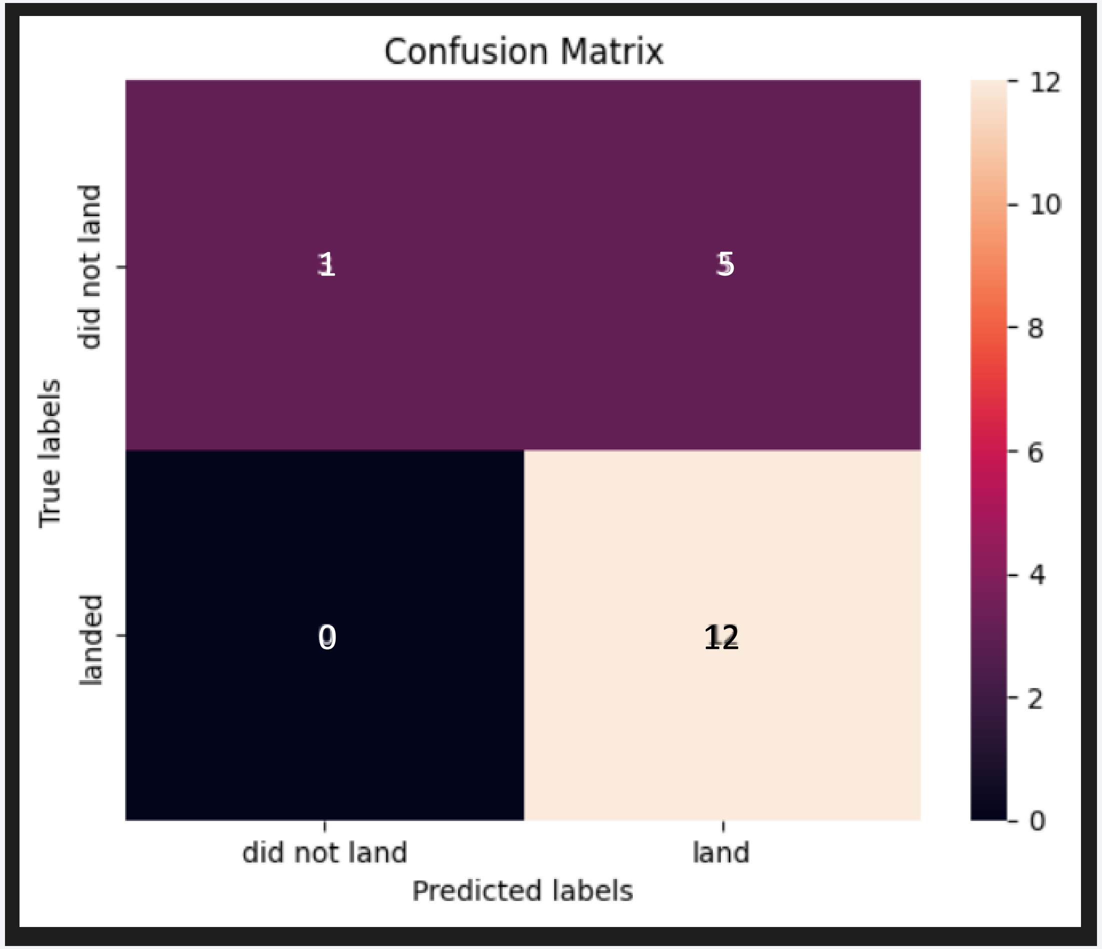
	Algorithm	Acc Score
0	Logistic Regression	0.846429
1	Decision Tree	0.873214
2	KNN	0.848214
3	SVM	0.848214

Confusion Matrix

Confusion Matrix is one of the metrics to find how well the algorithm is working based on its TP, TN, FP and FN cell.

Here in we can see that, our model is performing well as it is correctly predicting the TP values, that means, if the actual label is landed then the model is predicting it's landing positively.

As well as for the FP as well it has predicted the labels approximate correctly.



Conclusions

- By analysing most of the plots, it came to know that, the more the flight numbers on the launch site, the greater the success rates. So we can say that Flight numbers are directly proportional to success rate.
- Launch outcome will be more successful if the weight carried comes under the category of Low weighted payload (0 kg – 4000 kg).
- KSC LC-39A launch sites has been proved more successful then the other with the 41.7 %.
- If the launch is planned on any one of the ES-L1, GEO, HEO and SSO orbit, it can result in success, as these orbits are having highest success rates.
- Launch success rates of Space X has been increasing since 2013 and it may keep rising high as the time passes by.

Thank you!