IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ihor Hapon
26-12-2022

Outline

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

Executive Summary

- Summary of methodologies
  - Collection of Data Via API and Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis Visualization and SQL
  - Interactive Visual analytics using Folium
  - Predictive analysis using machine learning models
- Summary of all results
  - EDA results and interactive visualization dashboard
  - Comparison of predictive analysis model

Introduction

- Project background and context

  - SpaceX is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010.

  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars where's other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage.

  - The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations.

- Problems you want to find answers

  - What are the most ideal conditions to guarantee a successful landing

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - ➢ Data was collected using SpaceX API and web scraping from Wikipedia
- Perform data wrangling
  - ➢ One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - ➢ Data collected were normalized, divided into training and test data sets, and evaluated by four different classification models. The accuracy of each model was assessed using various combinations of parameters.

Data Collection

Describe how data sets were collected.

- Obtained Data from API and Web pages
- Made it into a data frame
- Filtered unnecessary data by formatting the data frame
- Converting the data frame into .csv file for further use

API method:

Web scrapping method:

Data Collection – SpaceX API

- Collecting the SpaceX-Data from the REST-Api with get request and used json_normalize to convert the collected data to a pandas dataframe.

- Link to the Notebook: https://github.com/igrikg/PythonFinalDataScience/blob/master/jupyter-labs-spacex-data-collection-api%20(3).ipynb

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

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

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

```
# Lets take a subset of our dataframe keeping only the features we want and the flight
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extr
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value i
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Data Collection - Scraping

- Extracting the HTML from Wikipedia with requests and collecting the relevant column names with BeautifulSoup find_all()
- A dictionary with the column-names as keys was then parsed with the correspondent values of the columns.
- Link to the Notebook: https://github.com/igrikg/PythonFinalDataScience/blob/master/jupyter-labs-webscraping.ipynb

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

```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup=BeautifulSoup(response.content)
```

```python
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 launch_dict with key `Flight No.`
            #print(flight_number)
            datatimelist=date_time(row[0])

            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            #print(date)

            # Time value
            # TODO: Append the time into launch_dict with key `Time`
```

Data Wrangling

- Exploratory data analysis:
  - Calculating the number of launches at each site
  - Calculating the number and occurrence of each orbits
  - Calculating the number and occurrence of mission outcome per orbit type
- Creating a landing outcome label from Outcome column
- Link to the Notebook: https://github.com/igrikg/PythonFinalDataScience/blob/master/Module1_labs-jupyter-spacex-data_wrangling_jupyter.ipynb

Number of launches on each site

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

Number and occurrence of each orbit

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
HEO       1
GEO       1
ES-L1     1
SO        1
```

Number and occurence of mission outcome per orbit type

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
```

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
```

```
In [29]: df['Class']=landing_class
         df[['Class']].head(8)
```

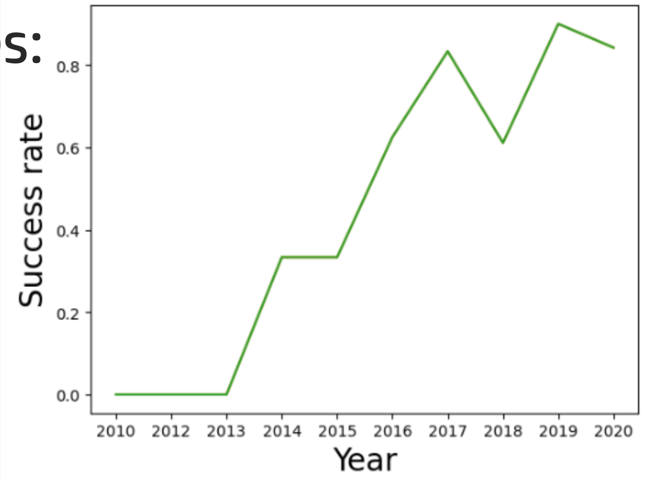
Out[29]:

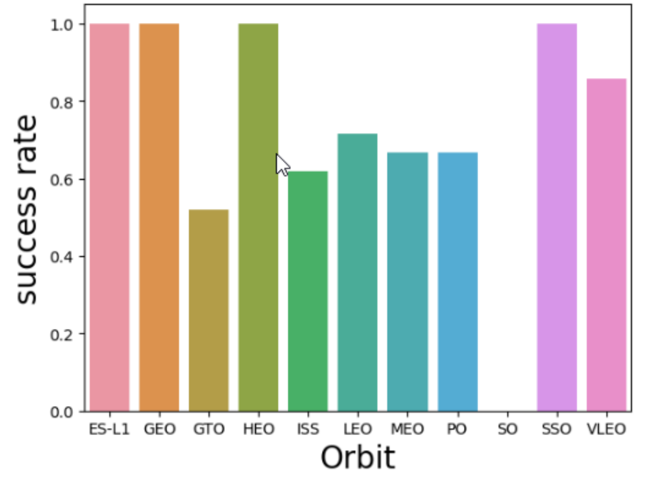
	Class
0	0
1	0
2	0
3	0
4	0

Create a landing outcome label from Outcome column

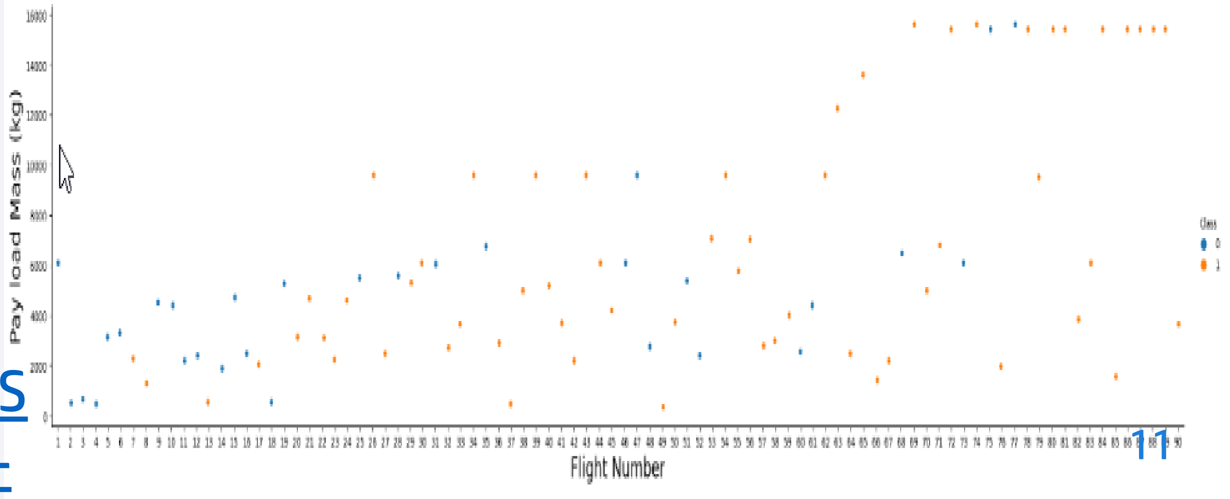
EDA with Data Visualization

- We explored the data by visualizing the relationship between different variables:
  - ➢ Flight Number and Launch Site
  - ➢ Payload and Launch Site
  - ➢ Success Rate of each orbit ype
  - ➢ Flight Number and Orbit type
  - ➢ Payload and Orbit type





- We use line plot, scatter plot, catplot, and bar plot to visualize the relationships between variables



- Link to the Notebook: https://github.com/igrikg/PythonFinalDataScience/blob/master/Module2_jupyter-labs-eda-dataviz.ipynb

EDA with SQL

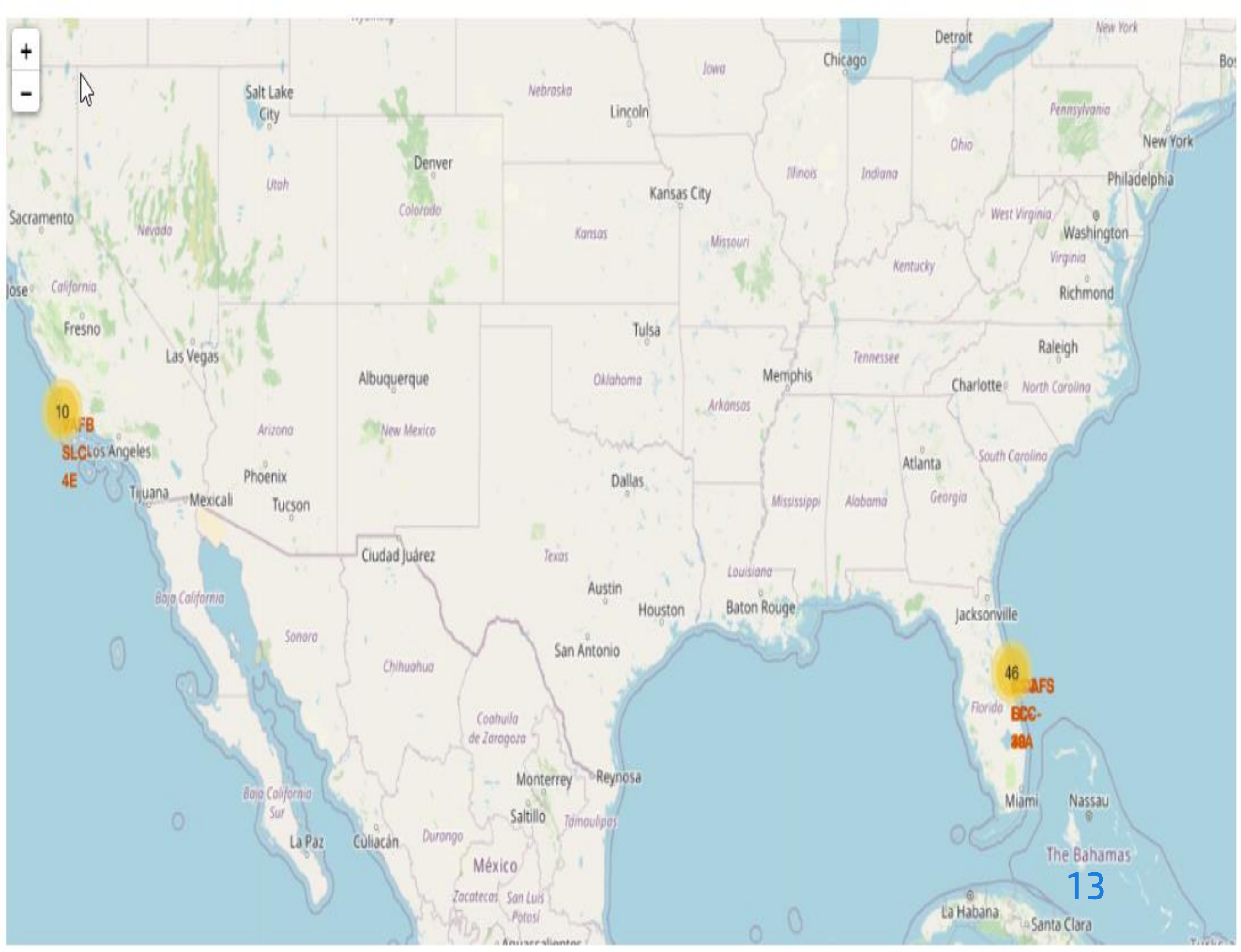
Using SQL, we had performed many queries to get better understanding of the dataset:

- ✓ 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 succesful 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 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 successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

Link to the Notebook: https://github.com/igrikg/PythonFinalDataScience/blob/master/jupyter-labs-eda-sql-coursera_sqllite%20(2).ipynb

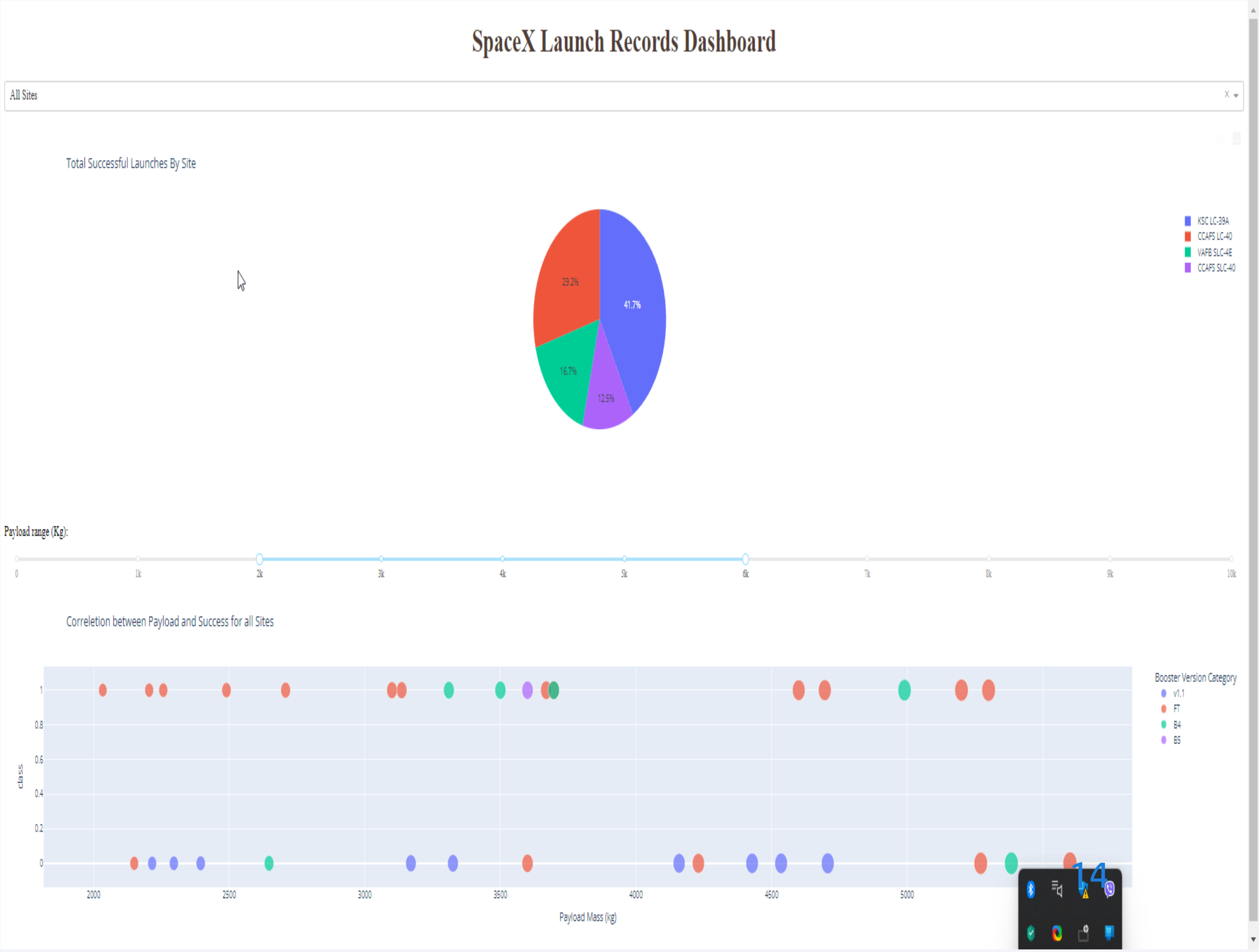
Build an Interactive Map with Folium

- Mark the success/failed launches for each site on the map
- Adding a red or green folium.Marker for each launch result to determine the launch sites with relatively high success rate
- Calculating the distances between a launch site to its proximities
  - ➢ Are launch sites near railways, highways and coastlines.
  - ➢ Do launch sites keep certain distance away from cities.
- Link to the notebook: https://github.com/igrikg/PythonFinalDataScience/blob/master/Module3_lab_jupyter_launch_site_location.ipynb



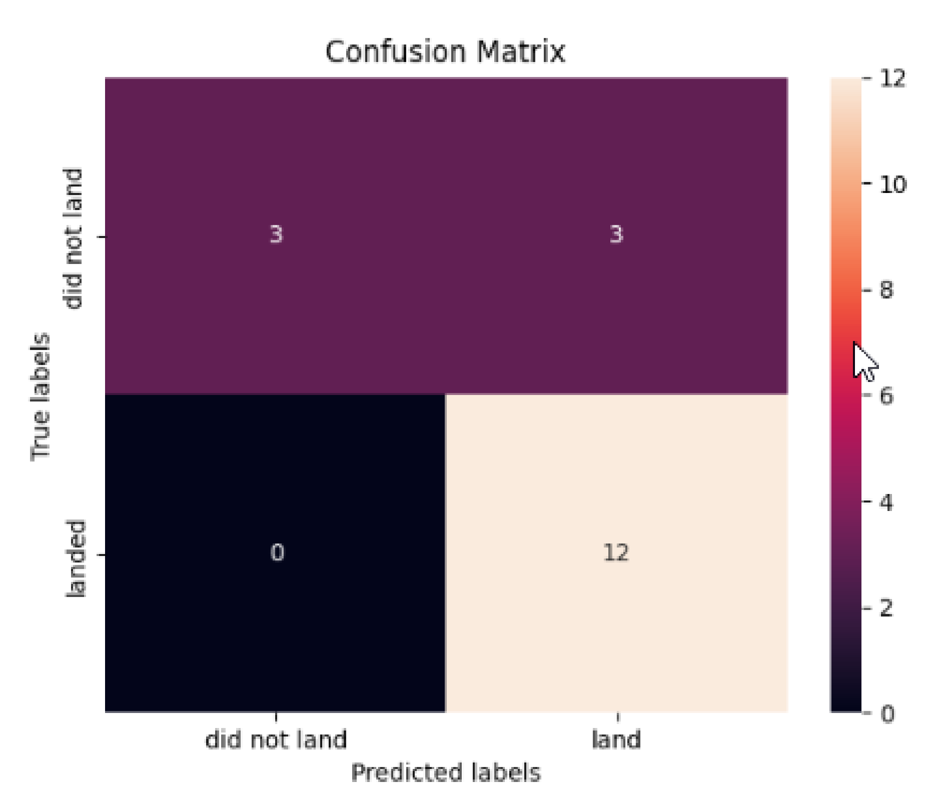
Build a Dashboard with Plotly Dash

- Interactive Dashboard with Plotly Dash
- Pie charts showing total launches by certain sites
- Scatter graph showing relationship of Outcome and Payload Mass (Kg) for different booster versions.
- Link to the notebook: https://github.com/igrikg/PythonFinalDataScience/blob/master/spacex_dash_app.py



Predictive Analysis (Classification)

- Create a NumPy array from the column Class for the labels and standardize the data for the features.
- We split the data into training and testing data using the function train_test_split. and finding the best hyperparameters with GridSearchCV()
- We calculated the accuracy of the models the method score.
- Link to the notebook: https://github.com/igrikg/PythonFinalDataScience/blob/master/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Ma.ipynb



Results

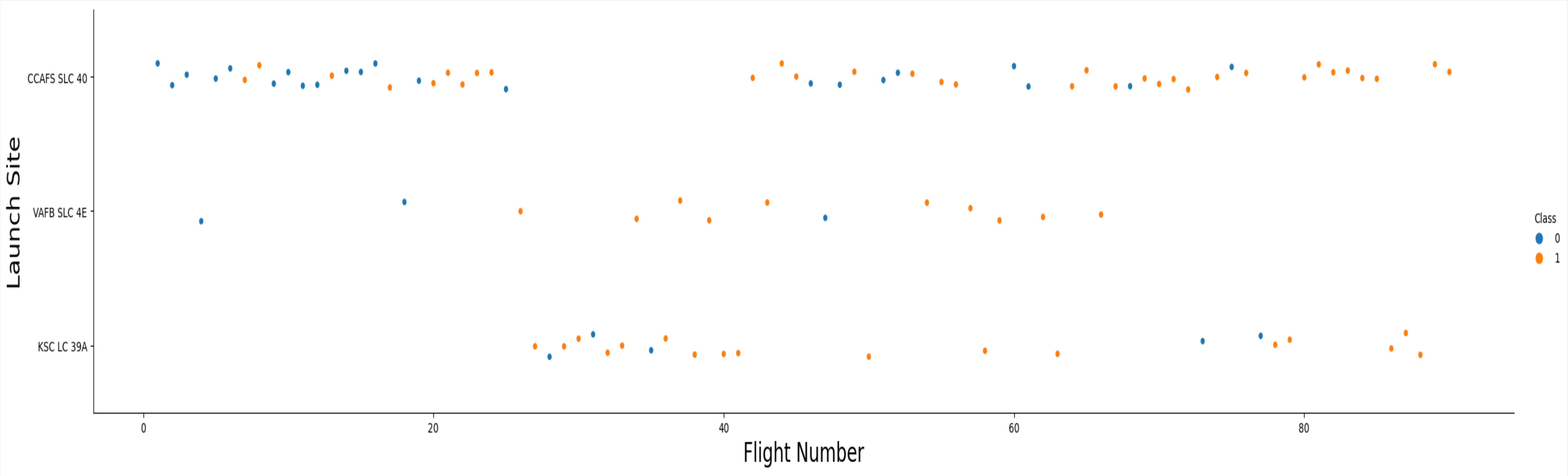
- The SVM, KNN and Logistic Regression models are the best in terms of prediction accuracy for this dataset.
- Low weighted payloads performs better than the heavier payloads.
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches.
- KSC LC 39A had the most Successful launches from all the sites.
- Orbit GEO, HEO, SSO, ES L1 has the best Success Rate

Section 2

Insights drawn from EDA

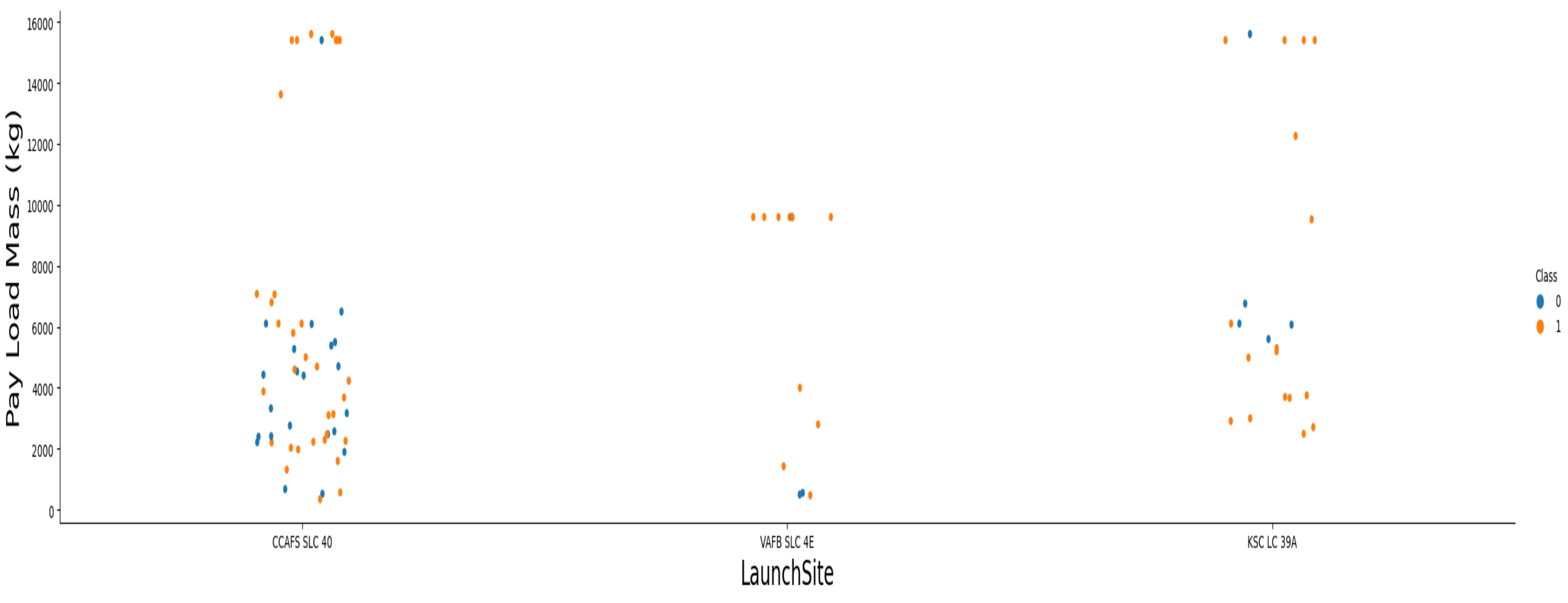
Flight Number vs. Launch Site

- According to the plot above, the most successful launch site is CCAF5 SLC 40
- In second place VAFB SLC 4E and third place KSC LC 39A;
- Also, the overall success rate is improving over time.



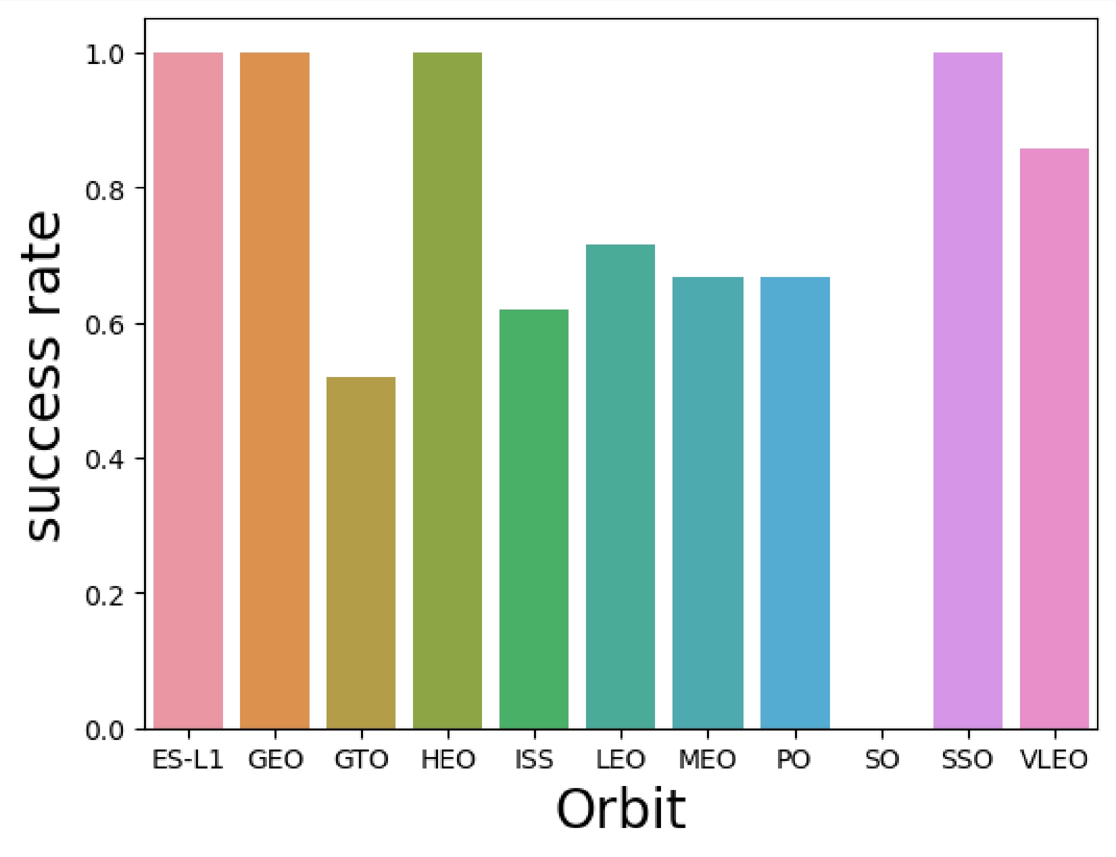
Payload vs. Launch Site

- A higher payload mass translates directly to a higher success rate.



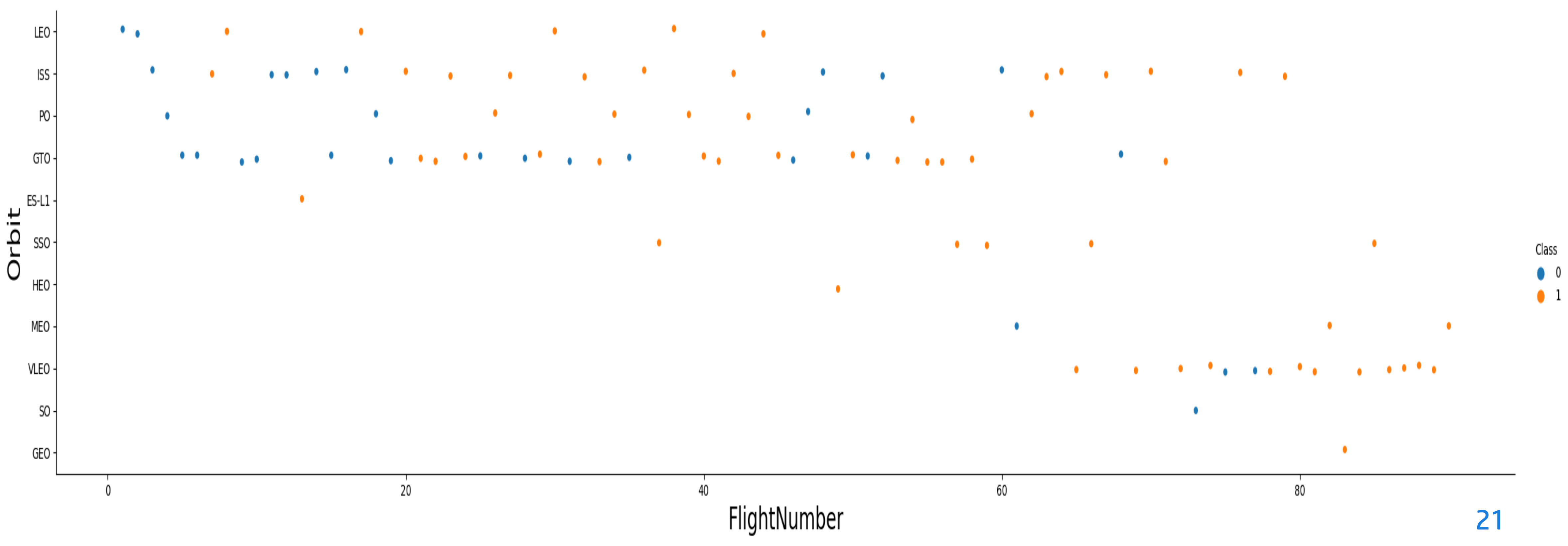
Success Rate vs. Orbit Type

- The following orbits have the highest success rates:
  - ➢ES-L1
  - ➢GEO
  - ➢HEO
  - ➢SSO



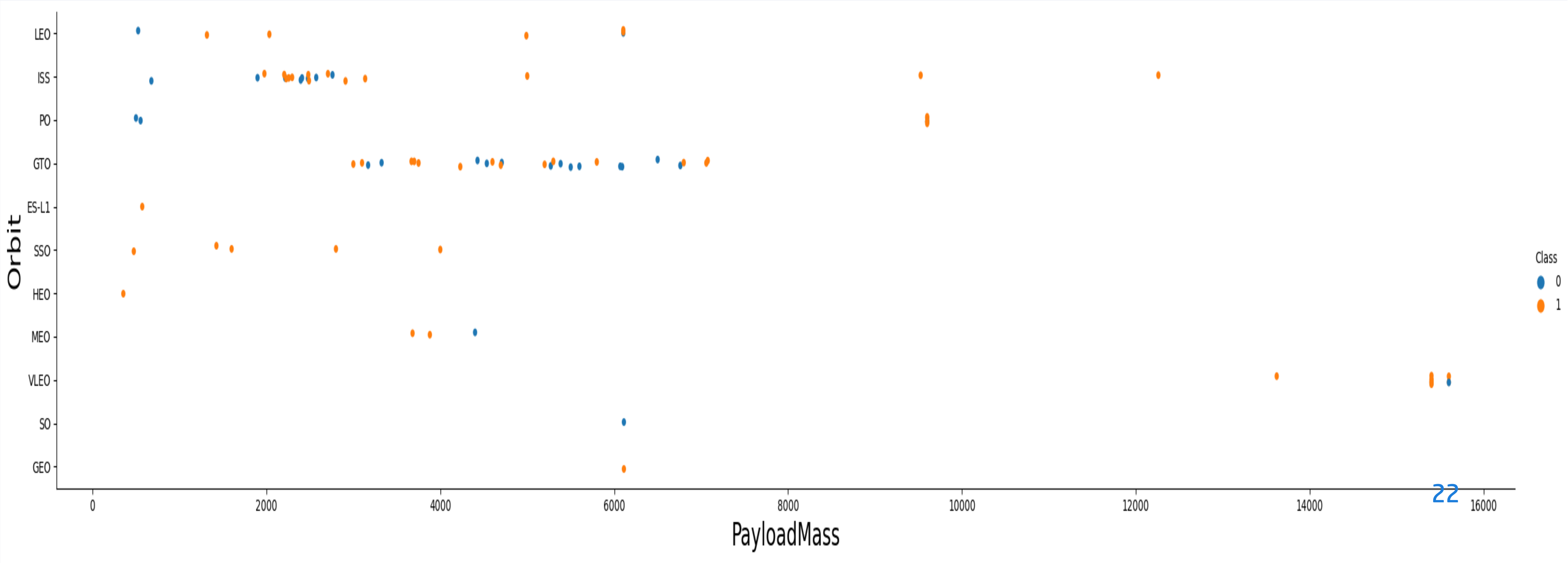
Flight Number vs. Orbit Type

- It appears, that in the LEO orbit success is related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



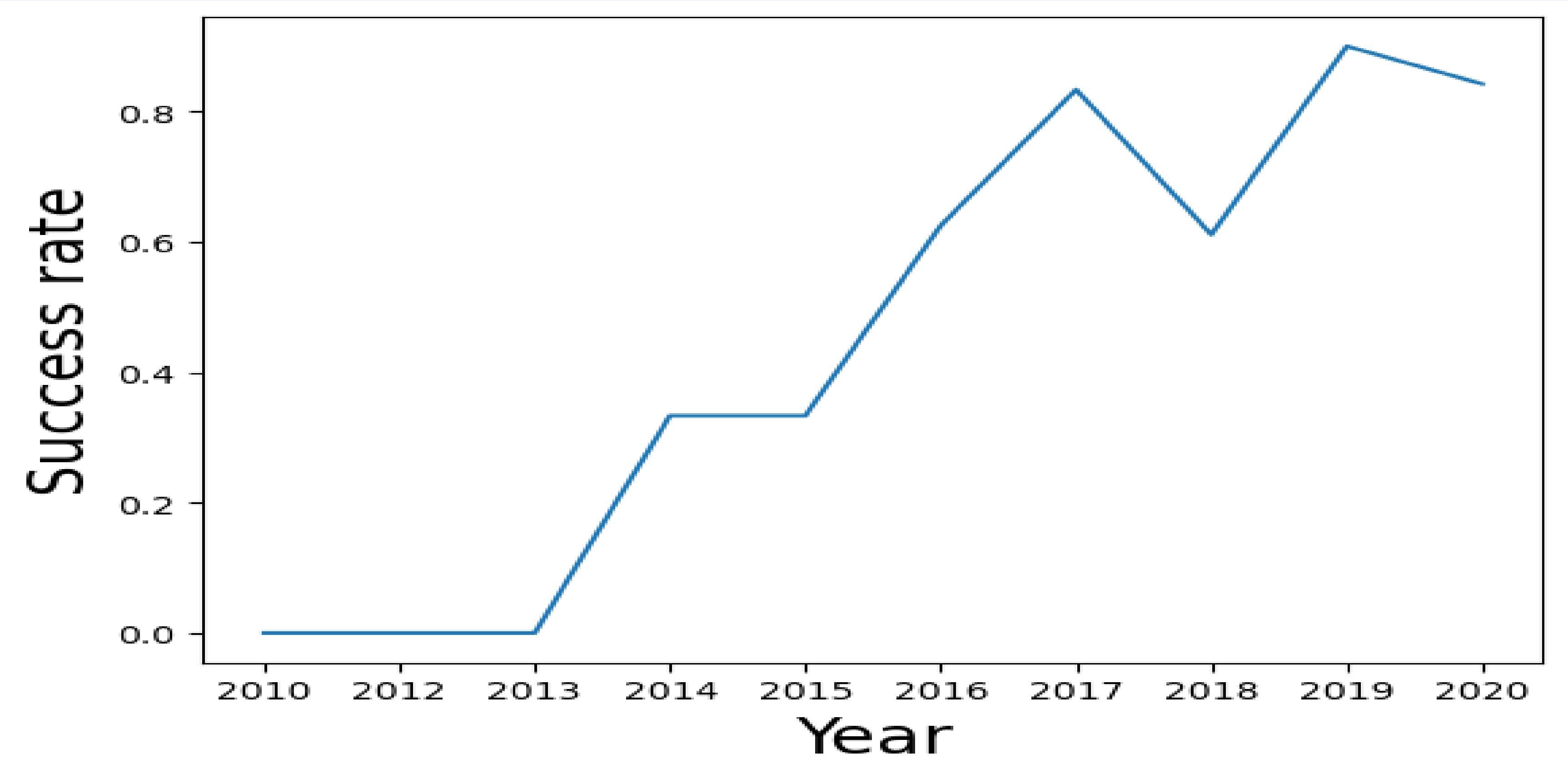
Payload vs. Orbit Type

- 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

- The success rate since 2013 kept increasing till 2020



All Launch Site Names

- Selecting the unique launch site names using the distinct key-word

```
In [14]: %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL;

 * sqlite:///my_data1.db
Done.

Out[14]:
```

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

Display 5 records where launch sites begin with the string 'CCA'

```
In [15]: %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL WHERE "Launch_Site" LIKE "CCA%";

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

Out[15]:

Launch_Site
CCAFS LC-40
CCAFS SLC-40

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

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

In [16]:
```
%sql SELECT SUM("PAYLOAD_MASS__KG_") as "Total_payload_mass" FROM SPACEXTBL WHERE "Customer" LIKE '%NASA (CRS)%';
```

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

Out[16]:

Total_payload_mass
48213

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

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

```
In [17]: %sql SELECT AVG("PAYLOAD_MASS__KG_") as "Average_payload_mass" FROM SPACEXTBL WHERE "Booster_Version" LIKE 'F9 v1.1%';
```

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

Out[17]:

Average_payload_mass
2534.6666666666665

First Successful Ground Landing Date

- The date of the first successful landing outcome on a drone ship was 01th May 2017

```
In [18]: #%sql SELECT * FROM SPACEXTBL LIMIT 10;
         %sql SELECT MIN("Date") FROM SPACEXTBL  WHERE "Landing _Outcome" LIKE "%Success%ground%" ORDER BY "Date";
         #22-12-2015

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

Out[18]:

MIN("Date")
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

In [19]:
```
#%sql SELECT * FROM SPACEXTBL LIMIT 10;
%sql SELECT "Payload"  As NAME, "PAYLOAD_MASS__KG_" FROM SPACEXTBL  WHERE "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;
```

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

Out[19]:

NAME	PAYLOAD_MASS__KG_
AsiaSat 8	4535
AsiaSat 6	4428
ABS-3A Eutelsat 115 West B	4159
Turkmen 52 / MonacoSAT	4707
SES-9	5271
JCSAT-14	4696
JCSAT-16	4600
EchoStar 23	5600
SES-10	5300
NROL-76	5300
Boeing X-37B OTV-5	4990
SES-11 / EchoStar 105	5200
Zuma	5000
GovSat-1 / SES-16	4230
SES-12	5384
Merah Putih	5800
Es hail 2	5300
SSO-A	4000
GPS III-01	4400
Nusantara Satu, Beresheet Moon lander, S5	4850
RADARSAT Constellation, SpaceX CRS-18	4200
GPS III-03, ANASIS-II	4311
ANASIS-II, Starlink 9 v1.0	5500
GPS III-04 , Crew-1	4311

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

In [20]:
```
%sql SELECT "Mission_Outcome", Count() AS "Count" FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

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

Out[20]:

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [21]:
```
%sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" IN (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);
```

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

Out[21]:

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
In [22]: #%sql SELECT DISTINCT("Landing _Outcome") FROM SPACEXTBL;

         %sql SELECT substr(Date, 4, 2) as month,Booster_Version,"Landing _Outcome" Launch_Site FROM SPACEXTBL WHERE substr(Date,7,4)='2015' and "Landing _Outcome" like "%Failure%drone%ship%";

 * sqlite:///my_data1.db
Done.

Out[22]:
```

month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	Failure (drone ship)
04	F9 v1.1 B1015	Failure (drone ship)

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

- 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

In [36]:
```
%%sql
select "Landing _Outcome", count(*) from spacextbl
WHERE
DATE(substr(Date,7,4)||'-'||substr(Date, 4, 2)||'-'||substr(Date, 1, 2))
between Date('2010-06-04') and Date('2017-03-20')
group by "Landing _Outcome" order by count("Landing _Outcome") desc;
```

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

Out[36]:

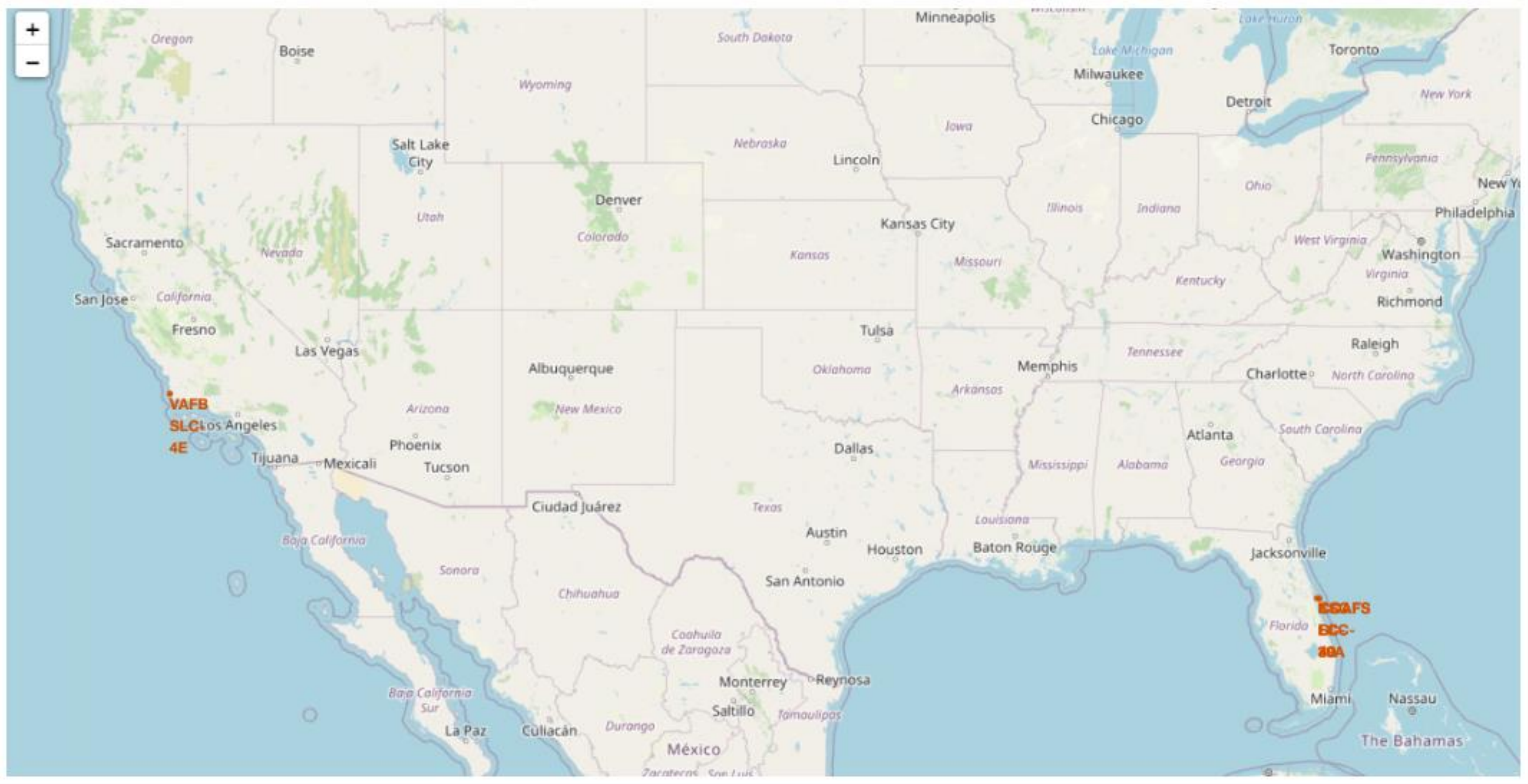
Landing _Outcome	count(*)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

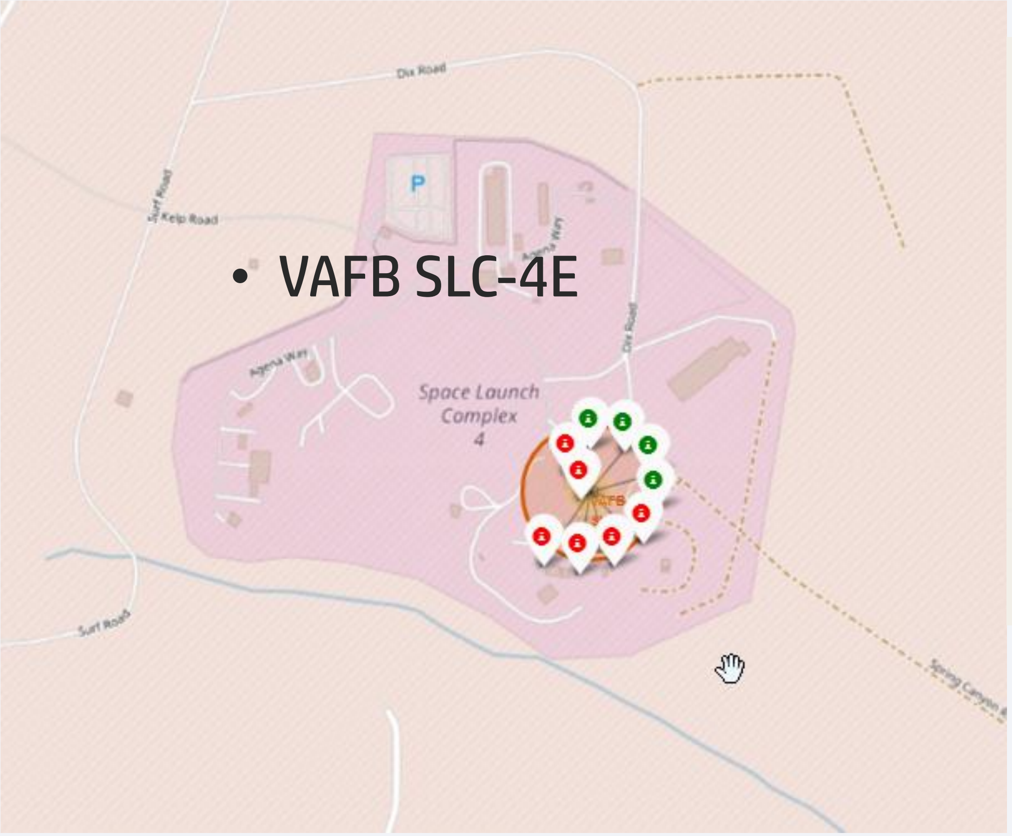
SpaceX Launch Sites

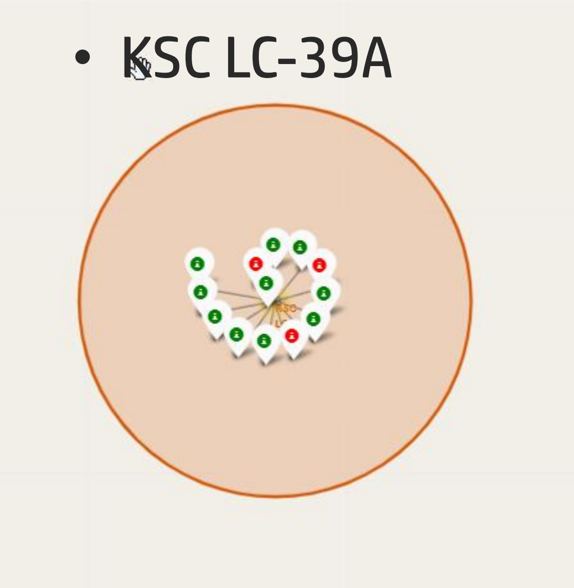
- The SpaceX launch sites are in the east and the west coast in the USA

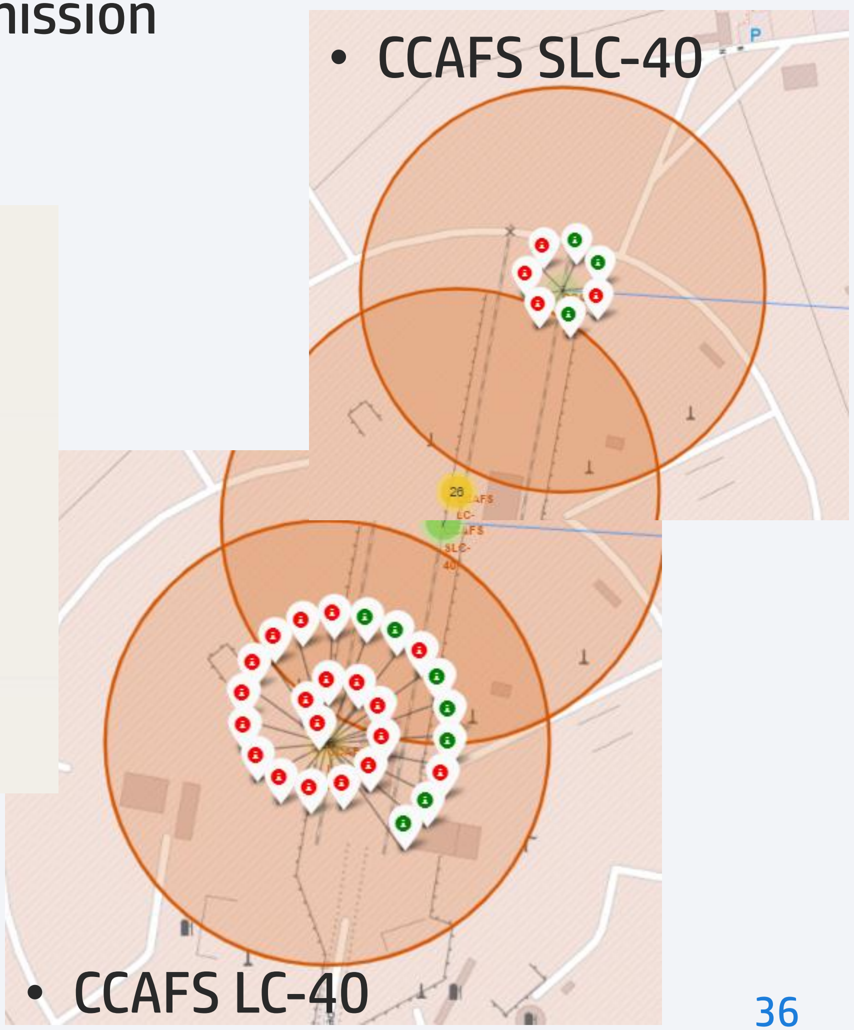


Markers showing launch sites with color labels

Green markers show mission success red markers show mission failure

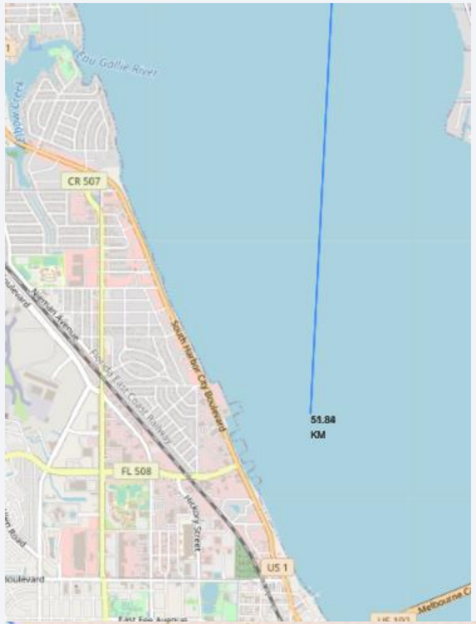


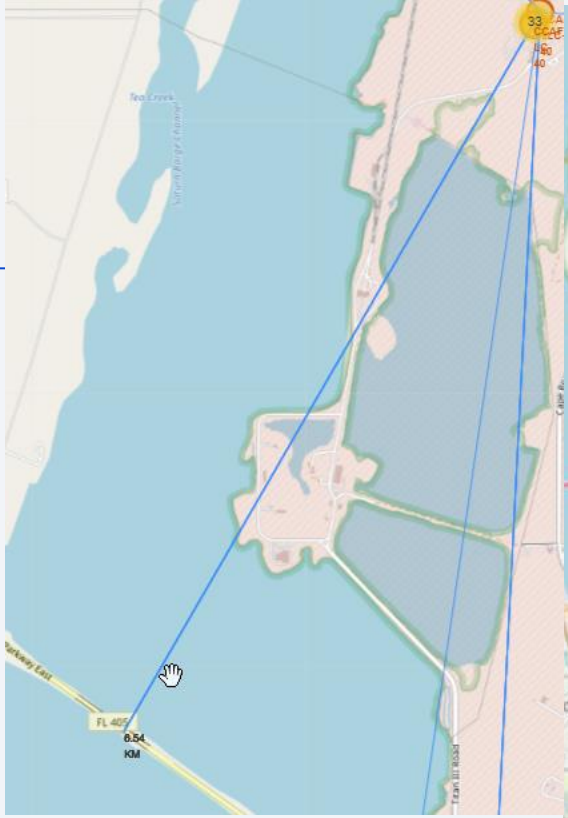


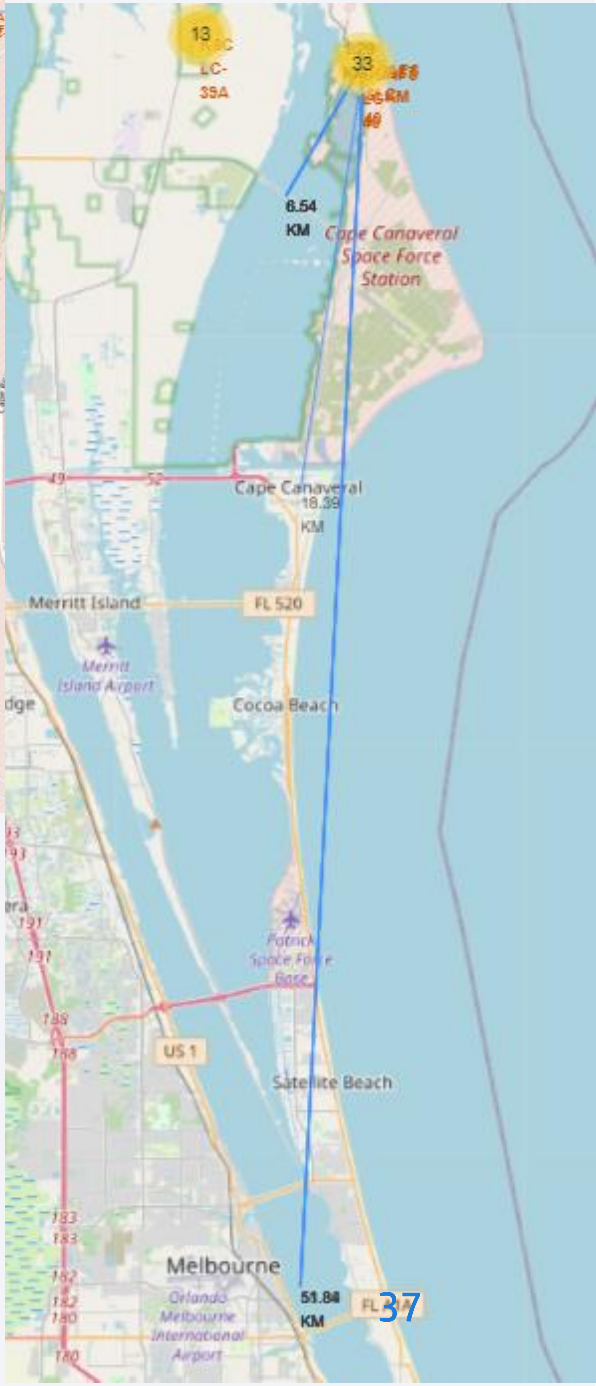


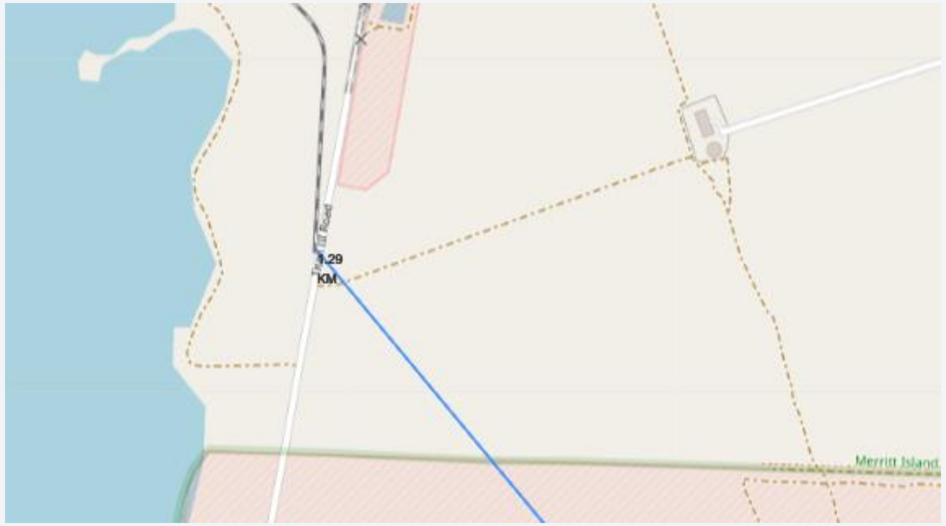
Distance to the proximities

- Calculated the distances between a launch site to its proximities





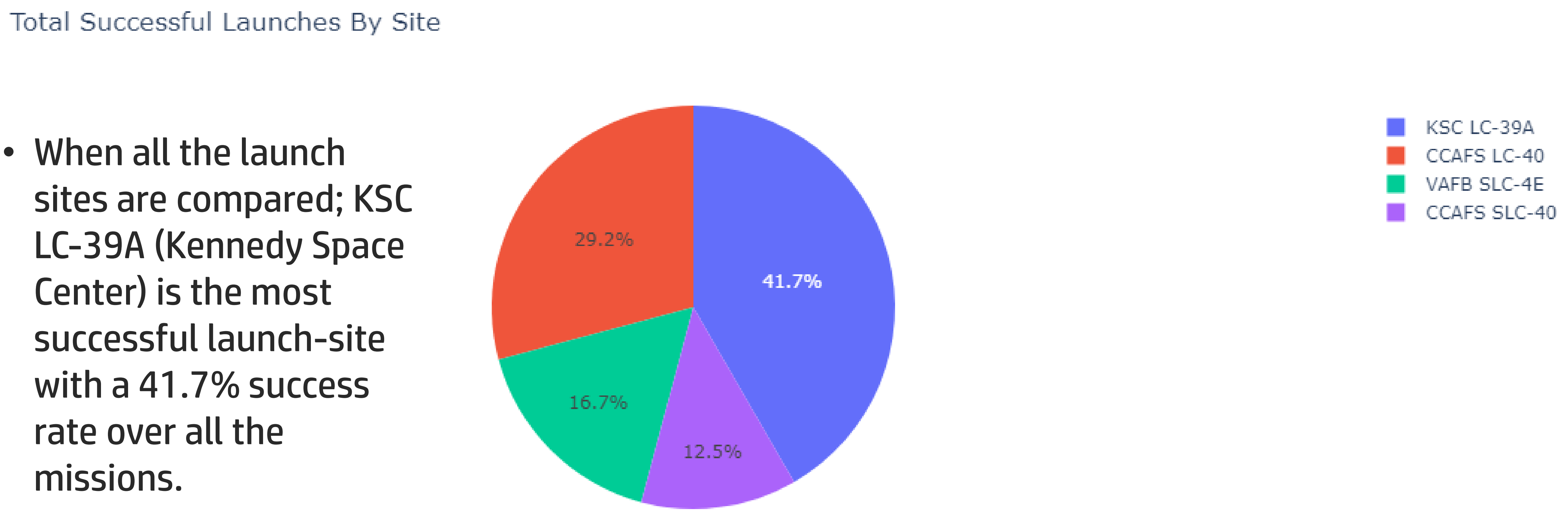




Section 4

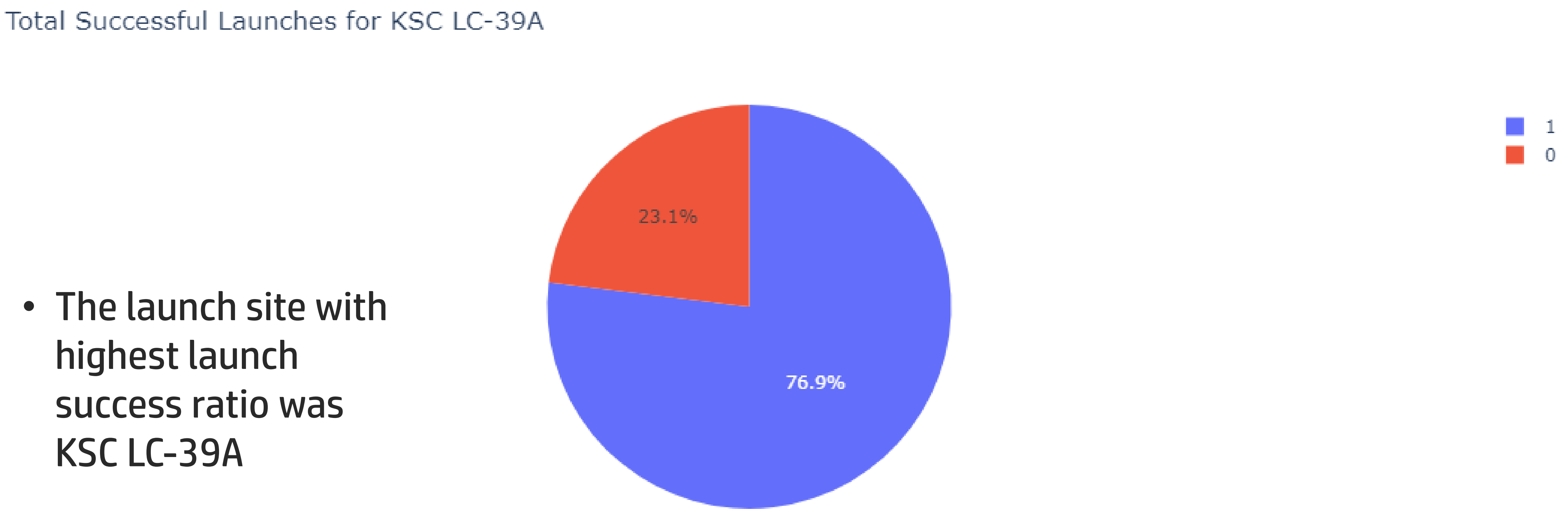
Build a Dashboard with Plotly Dash

Success percentage achieved by launch site



- When all the launch sites are compared; KSC LC-39A (Kennedy Space Center) is the most successful launch-site with a 41.7% success rate over all the missions.

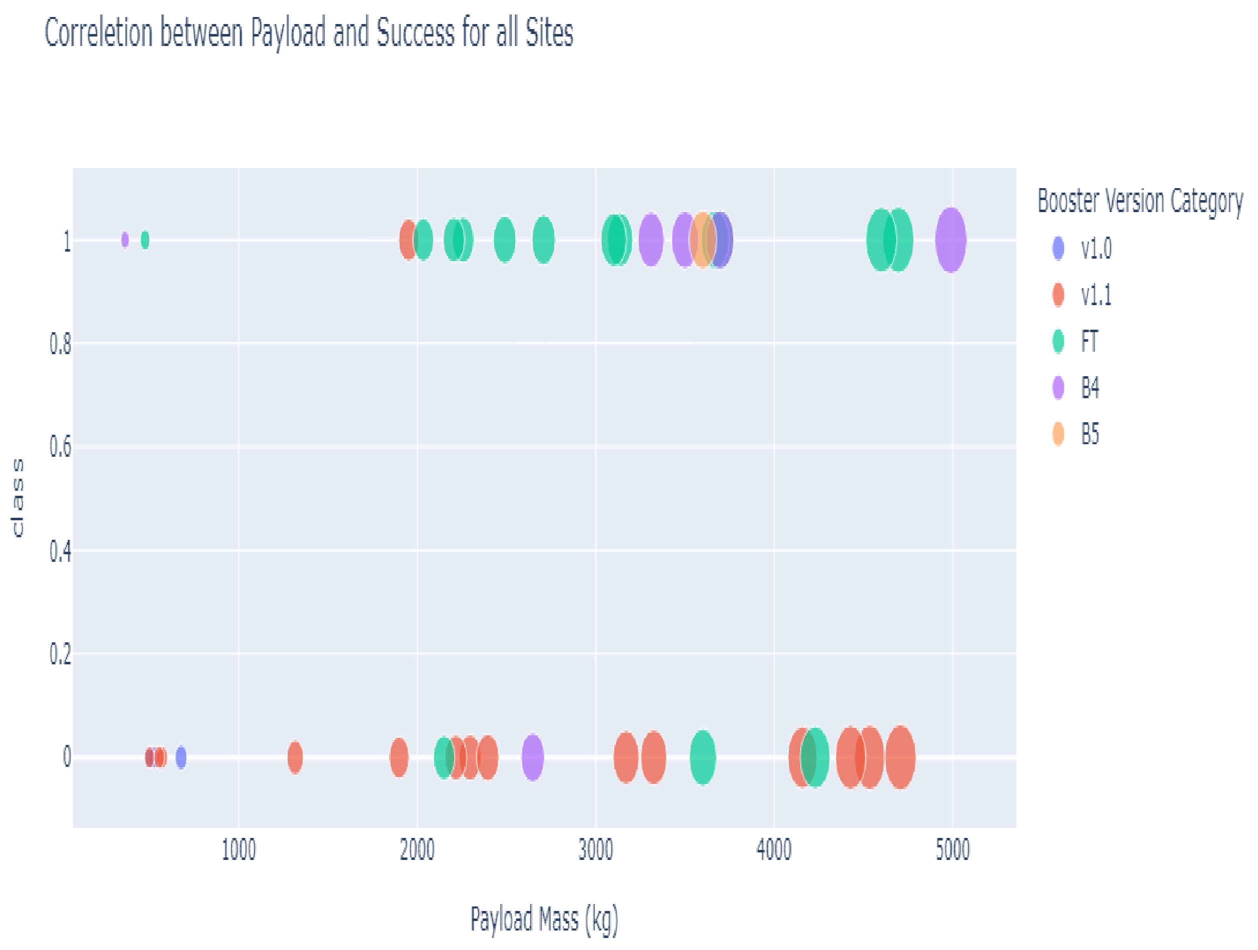
Launch Site With Highest Success Ratio

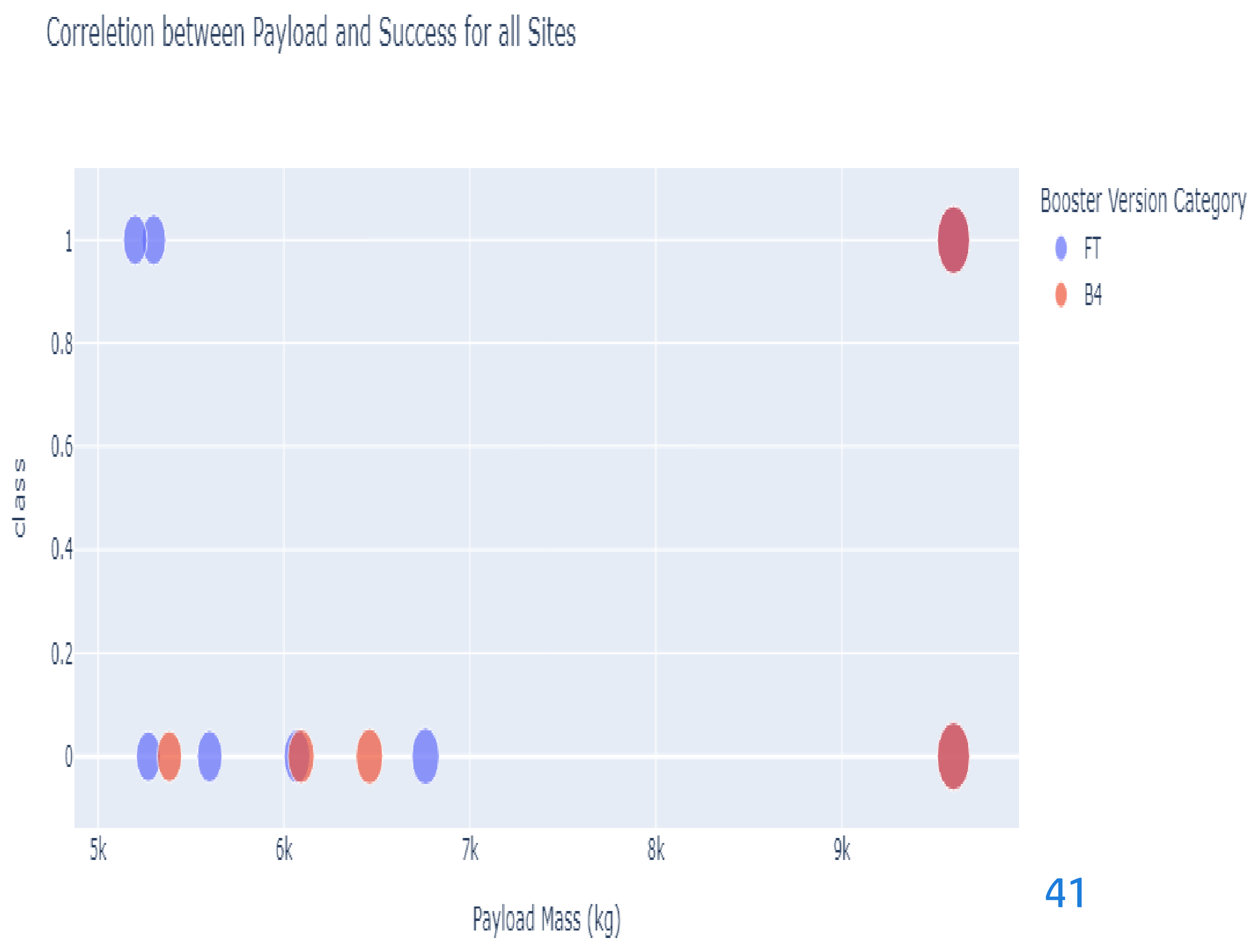


- The launch site with highest launch success ratio was KSC LC-39A

Success Rate By Payload Mass

The success rate of the payload mass from 0 – 5000 kg is much higher then the success rate from 5000 to 10000 kg



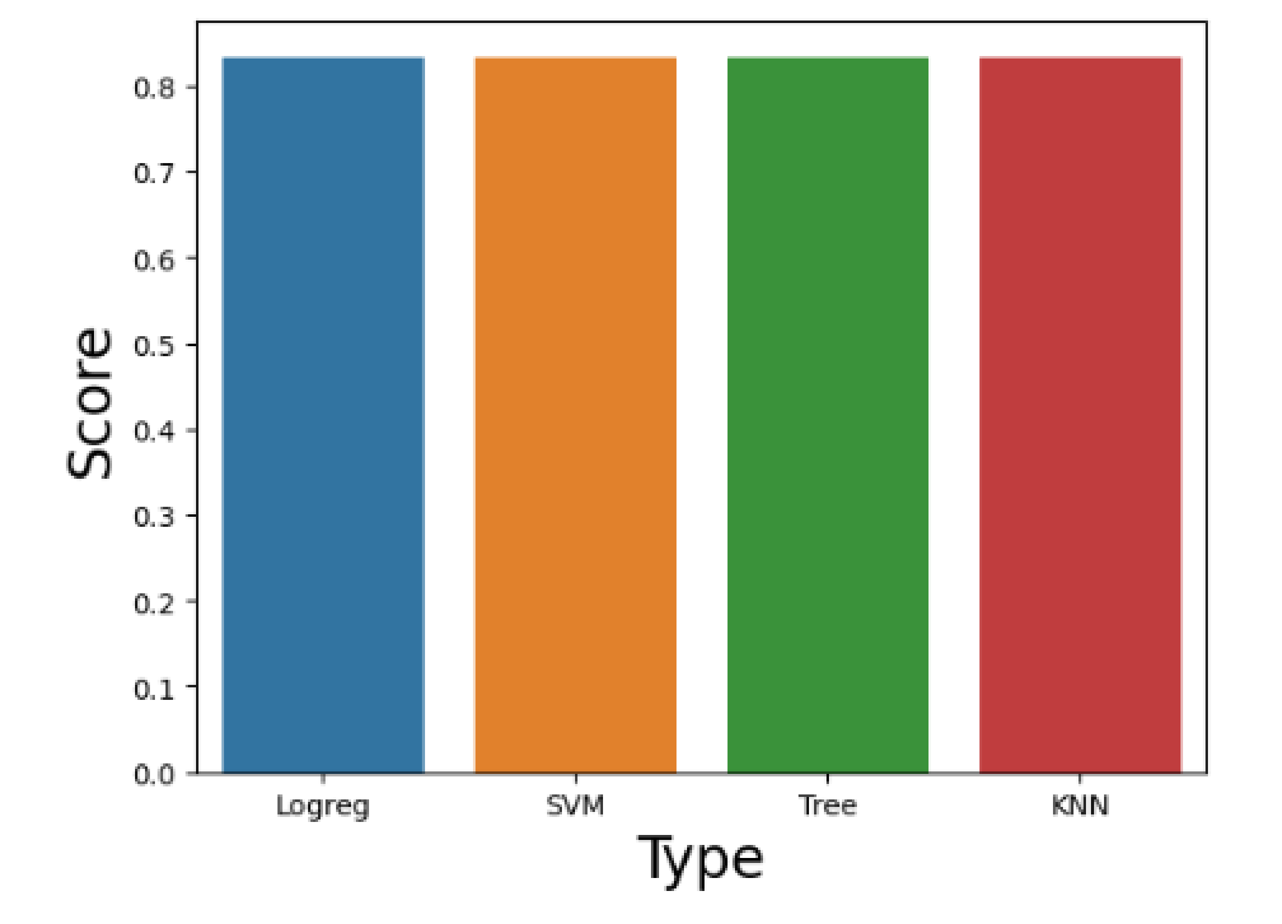


Section 5

Predictive Analysis (Classification)

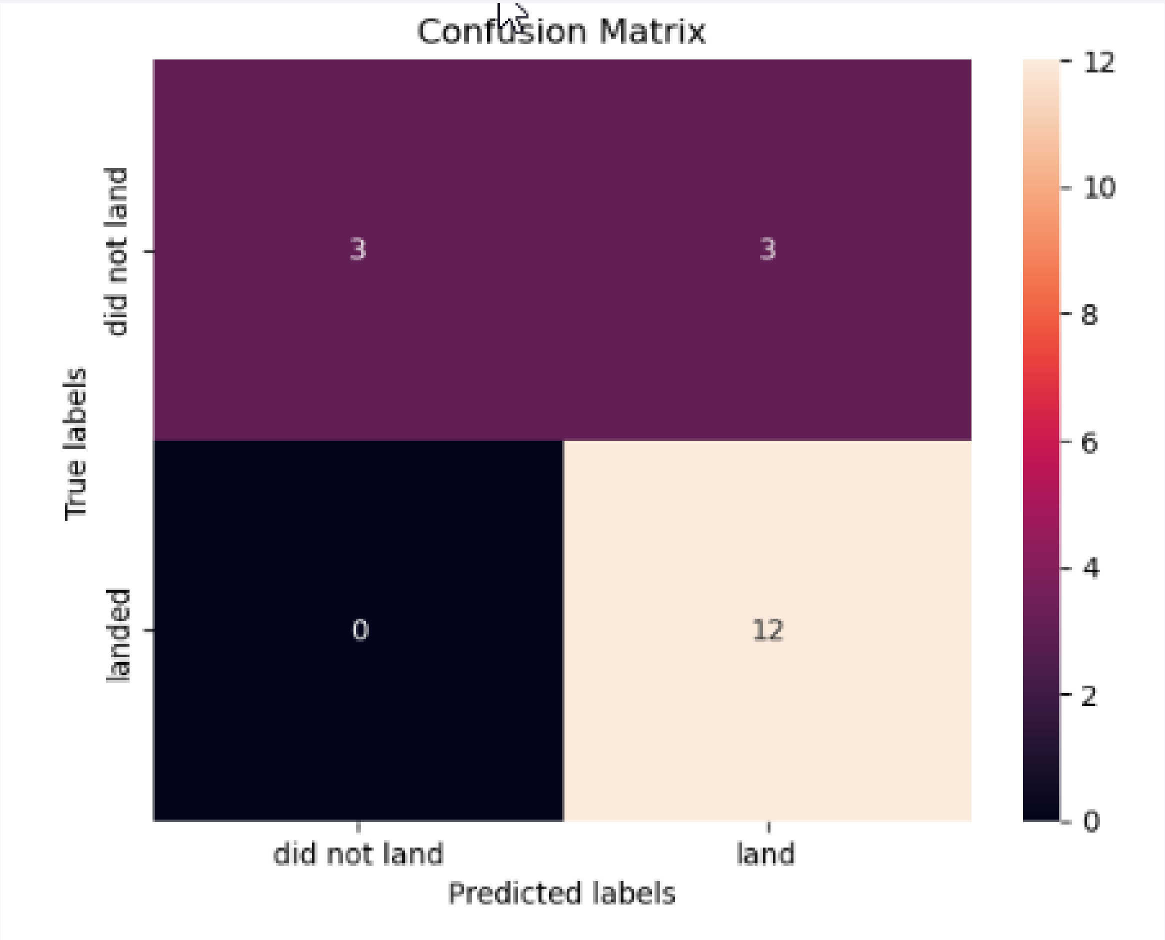
Classification Accuracy

- All the plots have similar score so there’s no one model with better accuracy score.



Confusion Matrix

- The Model predicted 12 landings correctly but couldn't predict the 3 unsuccessful landings (false positives).



Conclusions

- The larger the number of flights from a launch-site the higher the success rate
- The success rate since 2013 kept increasing till 2020
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had with 41.7% the most successful launches compared to all sites.
- The mean accuracy score of models was: 0.833333

Appendix

- Repository for all notebooks, files and data sets: https://github.com/igrikg/PythonFinalDataScience

Thank you!