



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ashutosh Dinesh Pathak
27TH OCT 2022



Outline

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

Executive Summary

- Summary of methodologies
 - Data Collection through API
 - Data Collection with Web Scraping
 - Data Wrangling
 - Exploratory Data Analysis with SQL
 - Exploratory Data Analysis with Data Visualization
 - Interactive Visual Analytics with Folium
 - Machine Learning Prediction
- Summary of all results
 - Exploratory Data Analysis result
 - Interactive analytics in screenshots
 - Predictive Analytics result

Introduction

Project background and context

- Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars;
- other providers cost upward of 165 million dollars each, much of the savings is because
- Space X can reuse the first stage. Therefore, if we can determine if the first stage will land,
- we can determine the cost of a launch. This information can be used if an alternate company
- wants to bid against space X for a rocket launch. This goal of the project is to create a
- machine learning pipeline to predict if the first stage will land successfully.

Problems you want to find answers

- What factors determine if the rocket will land successfully?
- The interaction amongst various features that determine the success rate of a successful landing.
- What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
 - Describe how data was collected
- 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

The data was collected using various methods

- Data collection was done using get request to the SpaceX API.
- Next, we decoded the response content as a Json using `.json()` function call and turn it into a pandas dataframe using `.json_normalize()`.
- We then cleaned the data, checked for missing values and fill in missing values where necessary.
- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
- The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting
- The link to the notebook is [https://github.com/Asdjamez/Ibm_capstone/blob/master/Final Assignment.ipynb](https://github.com/Asdjamez/Ibm_capstone/blob/master/Final_Assignment.ipynb)

1. Getting Response from API

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

2. Converting Response to a .json file

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

3. Apply custom functions to clean data

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

```
getBoosterVersion(data)
```

4. 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}
```

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

5. Filter dataframe and export to flat file (.csv)

```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]  
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```


Data Collection - Scrapping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is https://github.com/Asdjamez/bm_capstone/blob/master/Final%20Assignment%20Webscrapping.ipynb

1 .Getting Response from HTML

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

2. Creating BeautifulSoup Object

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

3. Finding tables

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

4. Getting 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
```

5. Creation of dictionary

```
launch_dict= dict.fromkeys(column_names)

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

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'] = []
launch_dict['Version Booster'] = []
launch_dict['Booster landing'] = []
launch_dict['Date'] = []
launch_dict['Time'] = []
```

6. Appending data to keys (refer) to notebook block 12

```
In [12]: 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
```

7. Converting dictionary to dataframe

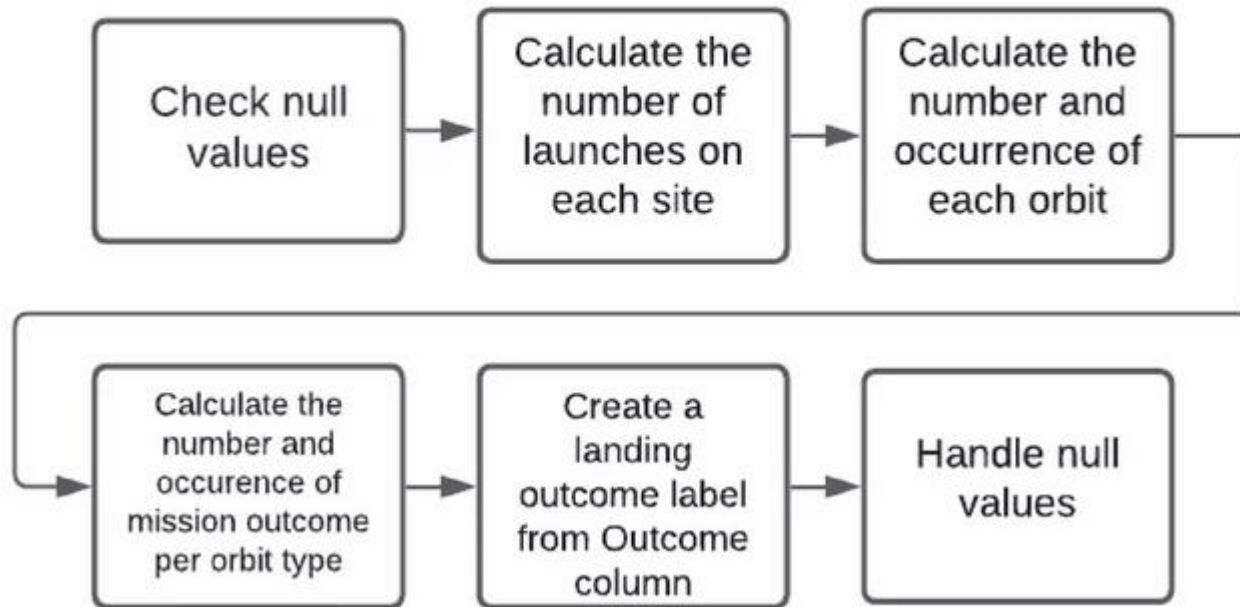
```
df = pd.DataFrame.from_dict(launch_dict)
```

8. Dataframe to .CSV

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

Data Wrangling

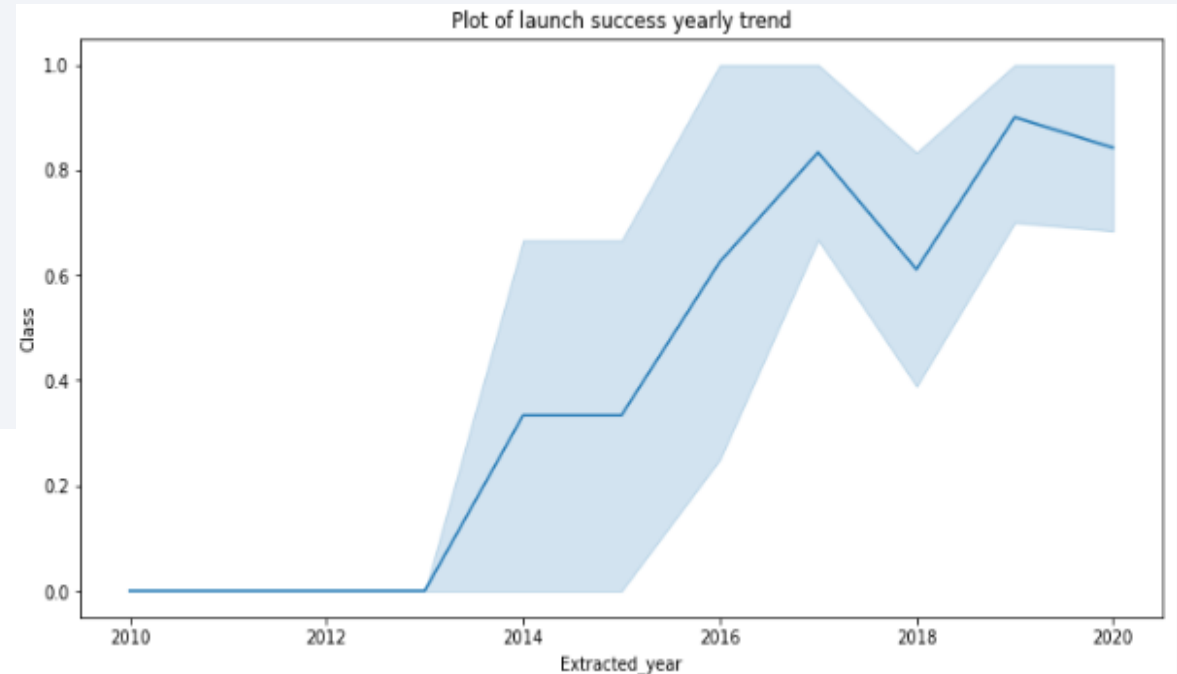
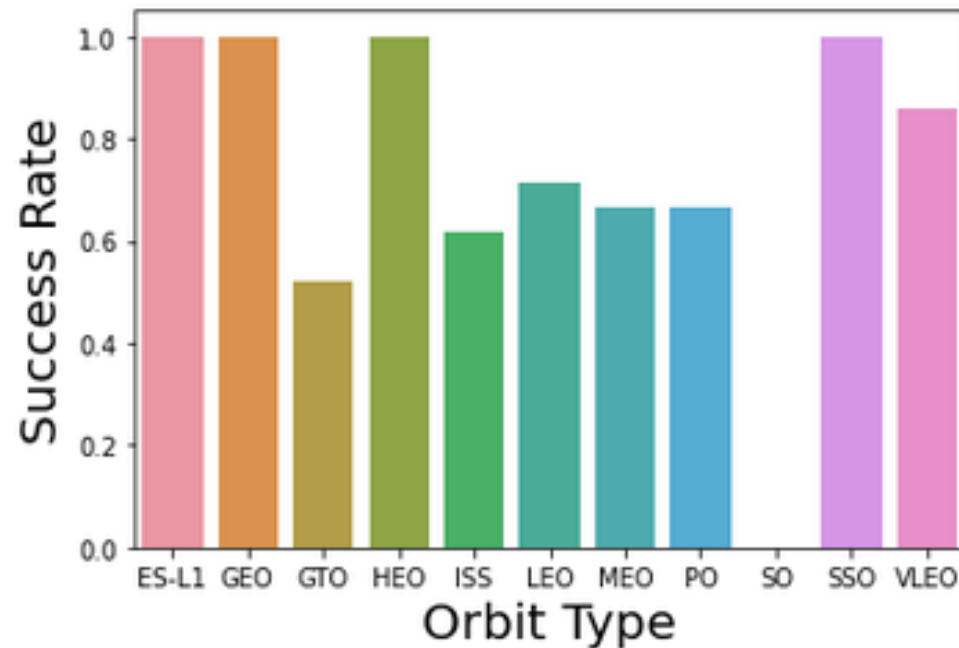
EDA analysis



- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits.
- We created landing outcome label from outcome column and exported the results to csv.
- The link to Github Notebook is https://github.com/Asdjamez/Ibm_capstone/blob/master/Data%20Wrangling.ipynb

EDA with Data Visualization

We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.



The link to the notebook is

https://github.com/Asdjamez/lbm_capstone/blob/master/EDA%20with%20Viz.ipynb

EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
 - The names of unique launch sites in the space mission.
 - The total payload mass carried by boosters launched by NASA (CRS)
 - The average payload mass carried by booster version F9 v1.1
 - The total number of successful and failure mission outcomes
 - The failed landing outcomes in drone ship, their booster version and launch site names.
- The link to the notebook is
https://github.com/Asdjamez/lbm_capstone/blob/master/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- We marked all 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, and 1 for success.
- 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. We answers some question for instance:
 - Are launch sites near railways, highways and coastlines.
 - Do launch sites keep certain distance away from cities.

Build a Dashboard with 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.
- The link to the notebook is https://github.com/Asdjamez/lbm_capstone/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- The link to the notebook is https://github.com/Asdjamez/lbm_capstone/blob/master/SpaceX%20Machine%20Learning%20Prediction.ipynb

Results

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

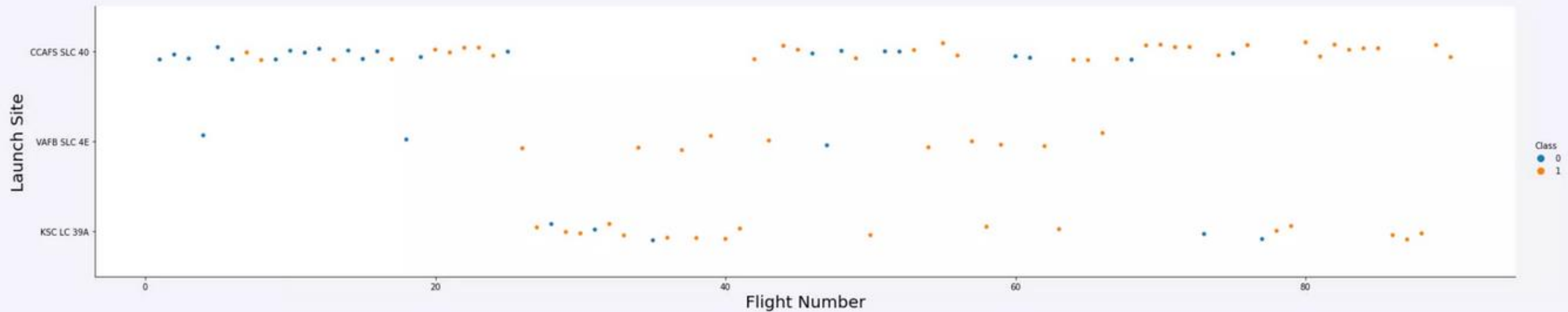
The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of red and cyan. A faint, light blue grid pattern is also visible, particularly in the lower-left quadrant. The overall effect is dynamic and technological.

Section 2

Insights drawn from EDA

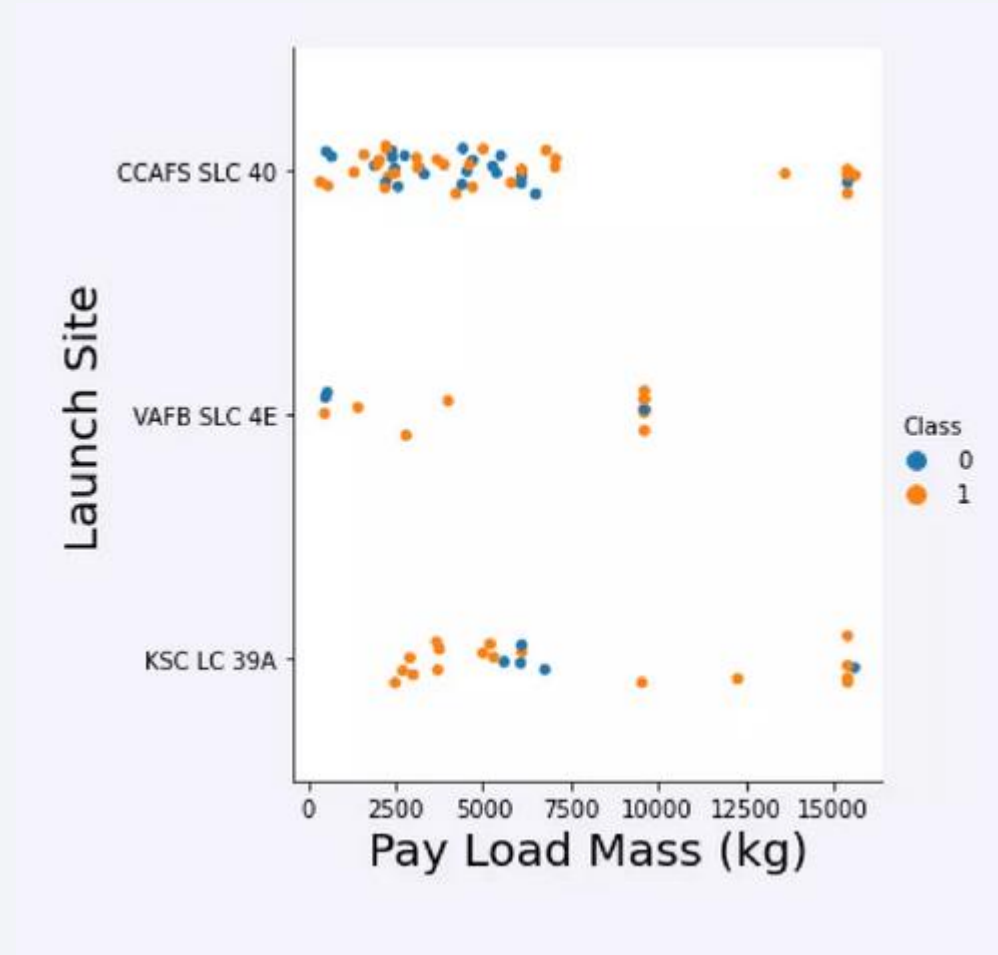
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



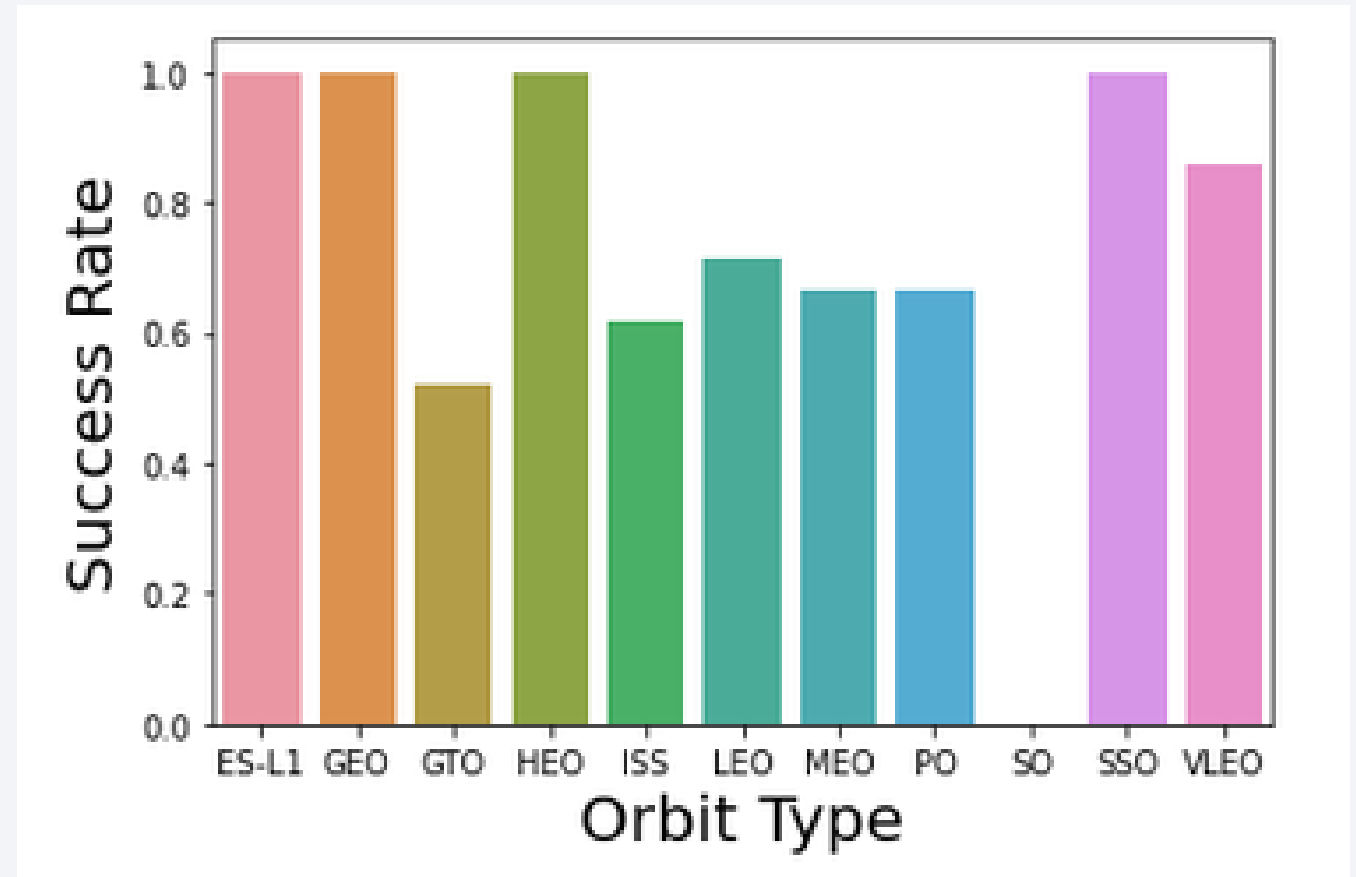
Payload vs. Launch Site

- The majority of rocket with lower payload mass have been launched from CCAFS SLC 40.



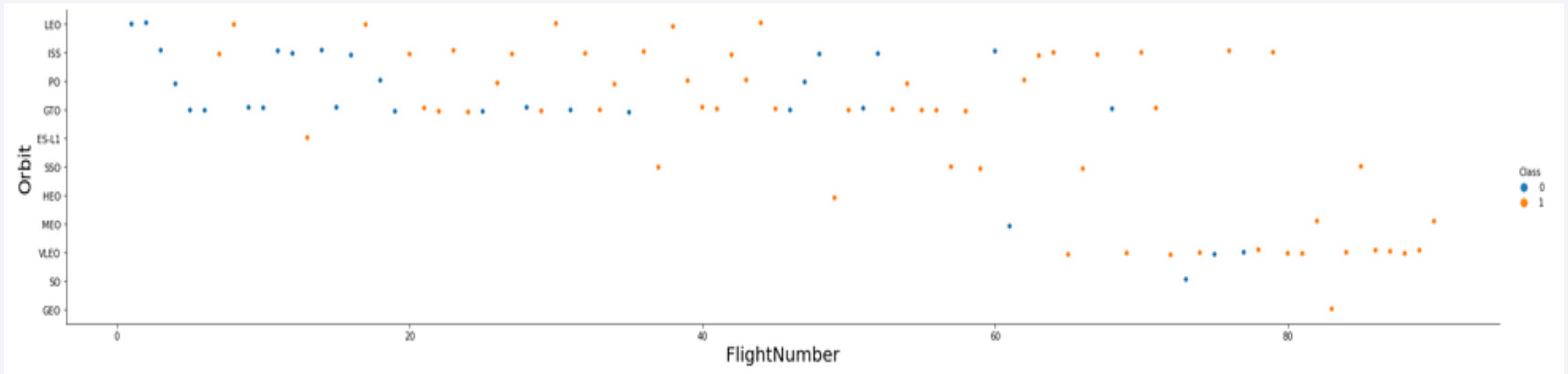
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate



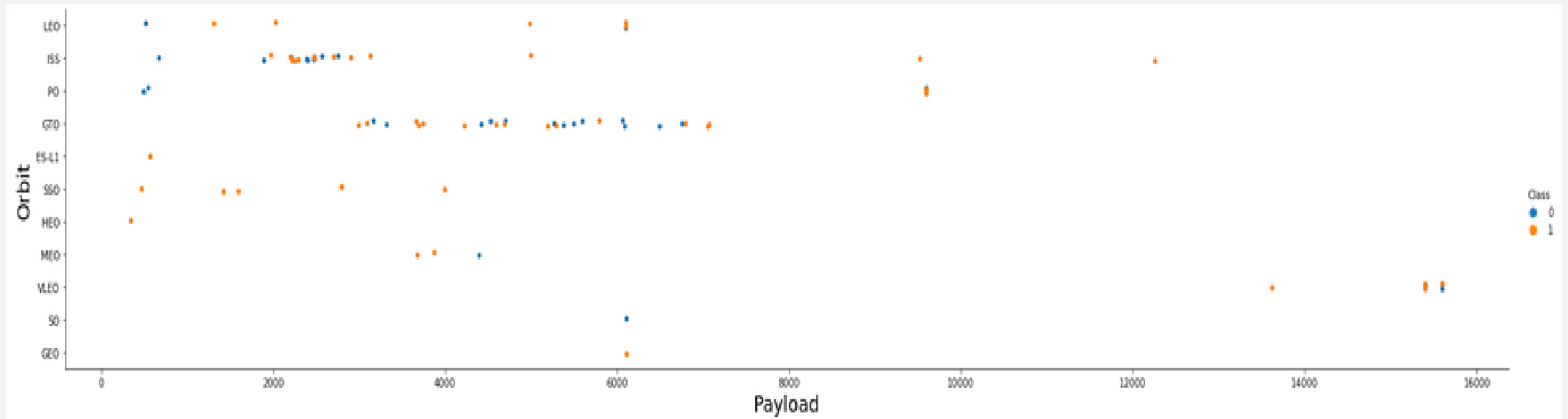
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



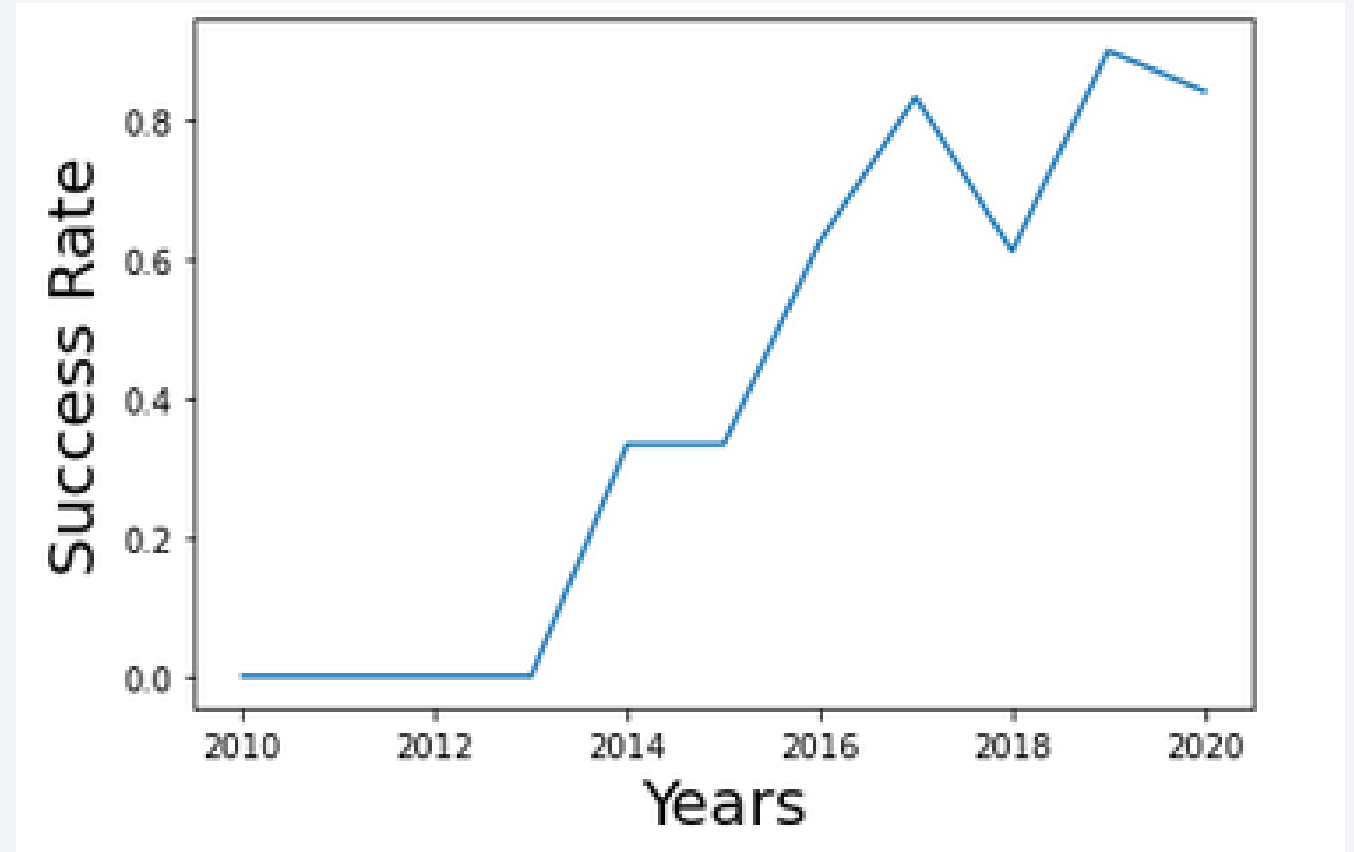
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, we can observe that the success rate is increasing from 2013 and again recovered its success rate after 2018 sudden decrease.



All Launch Site Names

Using below command in python we got all distinct Launch Site Names

```
%sql SELECT UNIQUE(LAUNCH_SITE) FROM SpaceX;
```

```
Out[6]: launch_site
```

```
CCAFS LC-40
```

```
CCAFS SLC-40
```

```
KSC LC-39A
```

```
VAFB SLC-4E
```


Launch Site Names Begin with 'CCA'

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

In [7]:

```
%%sql
SELECT
LAUNCH_SITE
FROM
Spacex
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5;
```

```
* ibm_db_sa://tbg74640:***@21fecfd8-47b7-4937-840d-d791c
Done.
```

Out[7]:

launch_site

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

Total Payload Mass

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

```
%%sql
SELECT
SUM(PAYLOAD_MASS__KG_) AS "TOTAL PAYLOAD"
FROM
Spacex
WHERE
Customer LIKE 'NASA%';
```

```
* ibm_db_sa://tbg74640:***@21fecfd8-47b7-4937-840d-d791d0218660.l
Done.
```

TOTAL PAYLOAD

99980

Average Payload Mass by F9 v1.1

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

In [13]:

```
%%sql
SELECT
AVG(PAYLOAD_MASS__KG_) AS "AVG PAYLOAD"
FROM SpaceX
WHERE Booster_Version = 'F9 v1.1' ;
```

```
* ibm_db_sa://tbg74640:***@21fecfd8-47b7-4937-840d-d791d02
Done.
```

Out[13]:

AVG PAYLOAD

2928

First Successful Ground Landing Date

List the date when the first successful landing outcome in ground pad was achieved.

Hint: Use min function

```
%sql SELECT MIN(DATE) FROM SpaceX WHERE LANDING__OUTCOME = 'Success (grc
```

```
* ibm_db_sa://tbg74640:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90:
Done.
```

```
1
```

```
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 DISTINCT BOOSTER_VERSION
FROM SpaceX
WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME = 'Success (drone ship)';
```

```
* ibm_db_sa://tbg74640:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od81cg.databases.appdomain.
Done.
```

booster_version

F9 FT B1021.2

F9 FT B1031.2

F9 FT B1022

F9 FT B1026

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%%sql
SELECT MISSION_OUTCOME AS MISSION,
COUNT(*) AS OUTCOME
FROM SpaceX
GROUP BY MISSION_OUTCOME;
```

```
* ibm_db_sa://tbg74640:***@21fecfd8-47b7-4937-840d-d791
Done.
```

mission	outcome
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

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

```
%%sql
SELECT DISTINCT BOOSTER_VERSION
FROM SpaceX
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SpaceX);
```

```
* ibm_db_sa://tbg74640:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od81cg.dat
Done.
```

booster_version

F9 B5 B1048.4

F9 B5 B1048.5

F9 B5 B1049.4

F9 B5 B1049.5

F9 B5 B1049.7

F9 B5 B1051.3

F9 B5 B1051.4

F9 B5 B1051.6

F9 B5 B1056.4

F9 B5 B1058.3

F9 B5 B1060.2

F9 B5 B1060.3

2015 Launch Records

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

```
%%sql
SELECT BOOSTER_VERSION, LAUNCH_SITE
FROM SpaceX
WHERE LANDING__OUTCOME='Failure (drone ship)' AND EXTRACT(YEAR FROM DATE)='2015';
```

```
* ibm_db_sa:///tbg74640:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od81cg.databa:
Done.
```

booster_version	launch_site
-----------------	-------------

F9 v1.1 B1012	CCAFS LC-40
---------------	-------------

F9 v1.1 B1015	CCAFS LC-40
---------------	-------------

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

```
%%sql
SELECT LANDING__OUTCOME, COUNT(*) AS qty
FROM SpaceX
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY 2 DESC;
```

```
* ibm_db_sa://tbg74640:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od81cg.databases.appdomain.cloud:31864/bludb
Done.
```

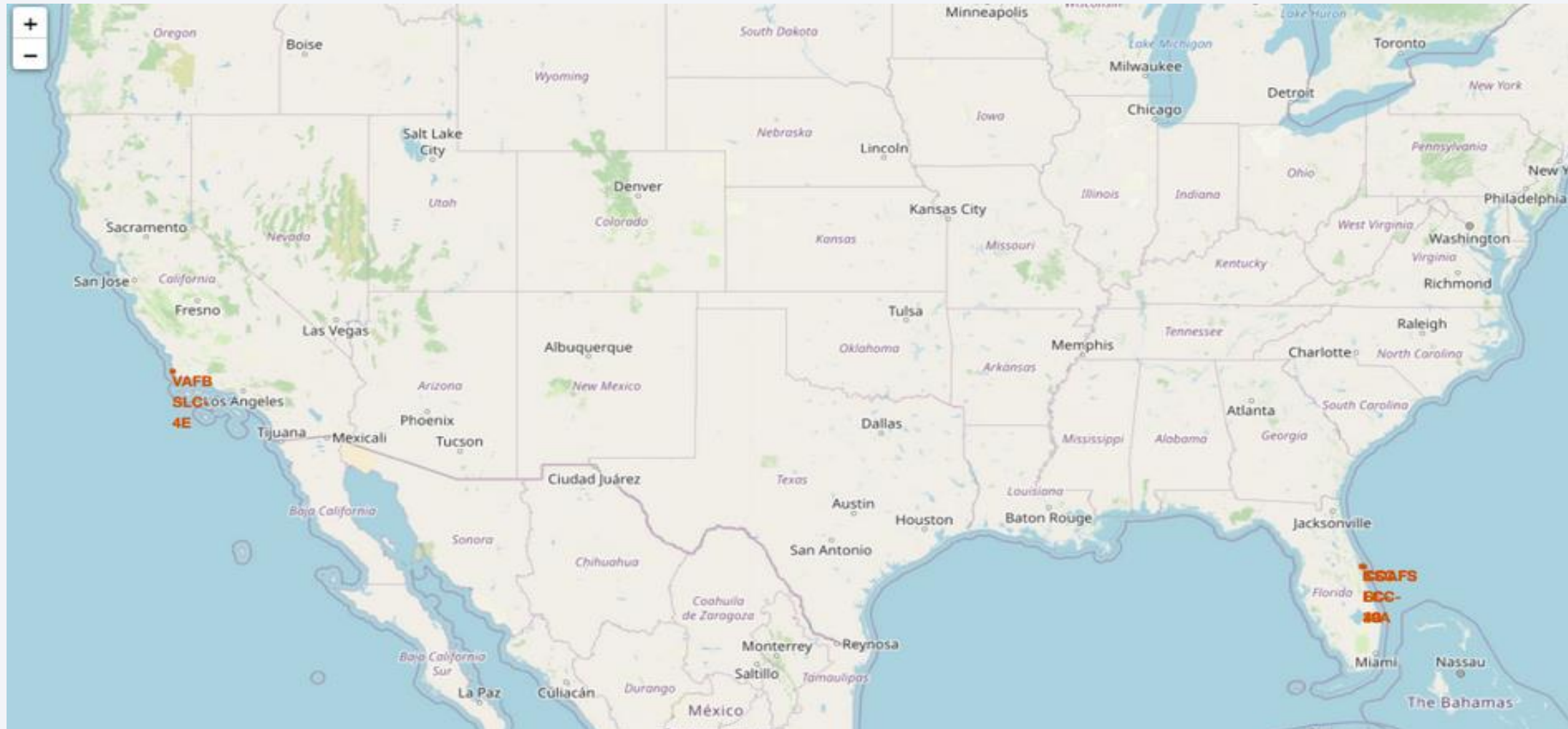
landing__outcome	qty
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

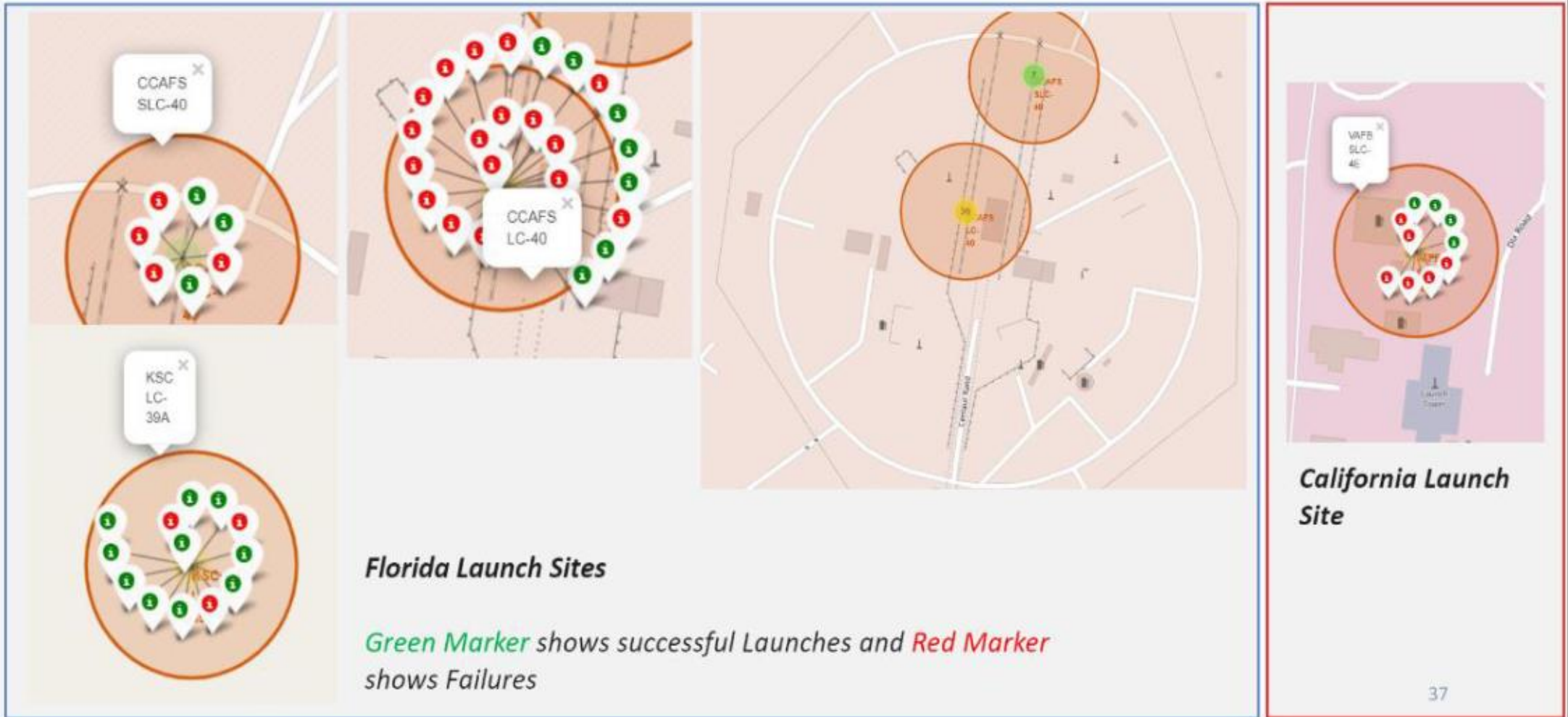
Section 3

Launch Sites Proximities Analysis

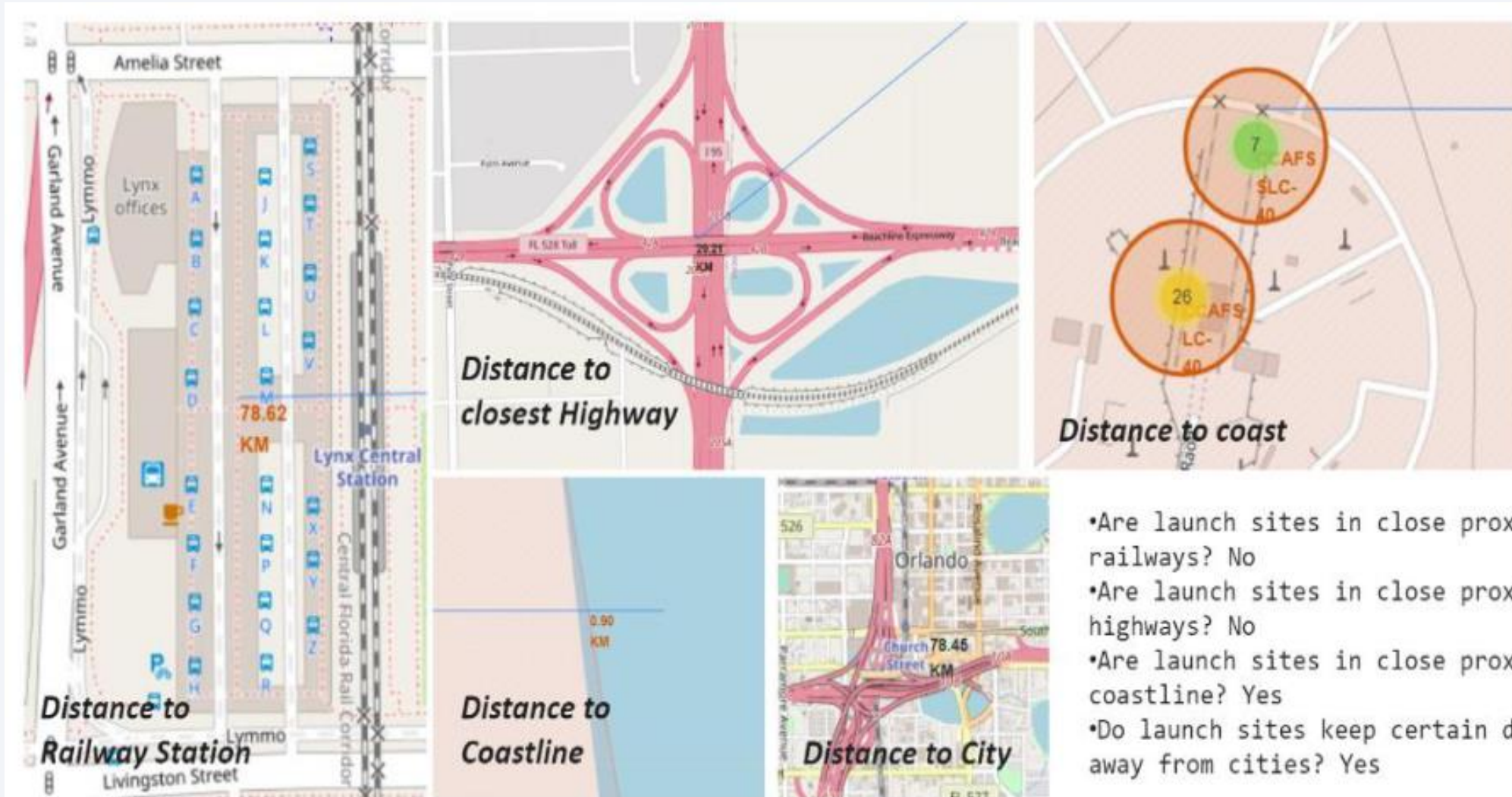
All Launch Sites



Launch Site Marker Label



Launch Site distance to landmarks



- 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

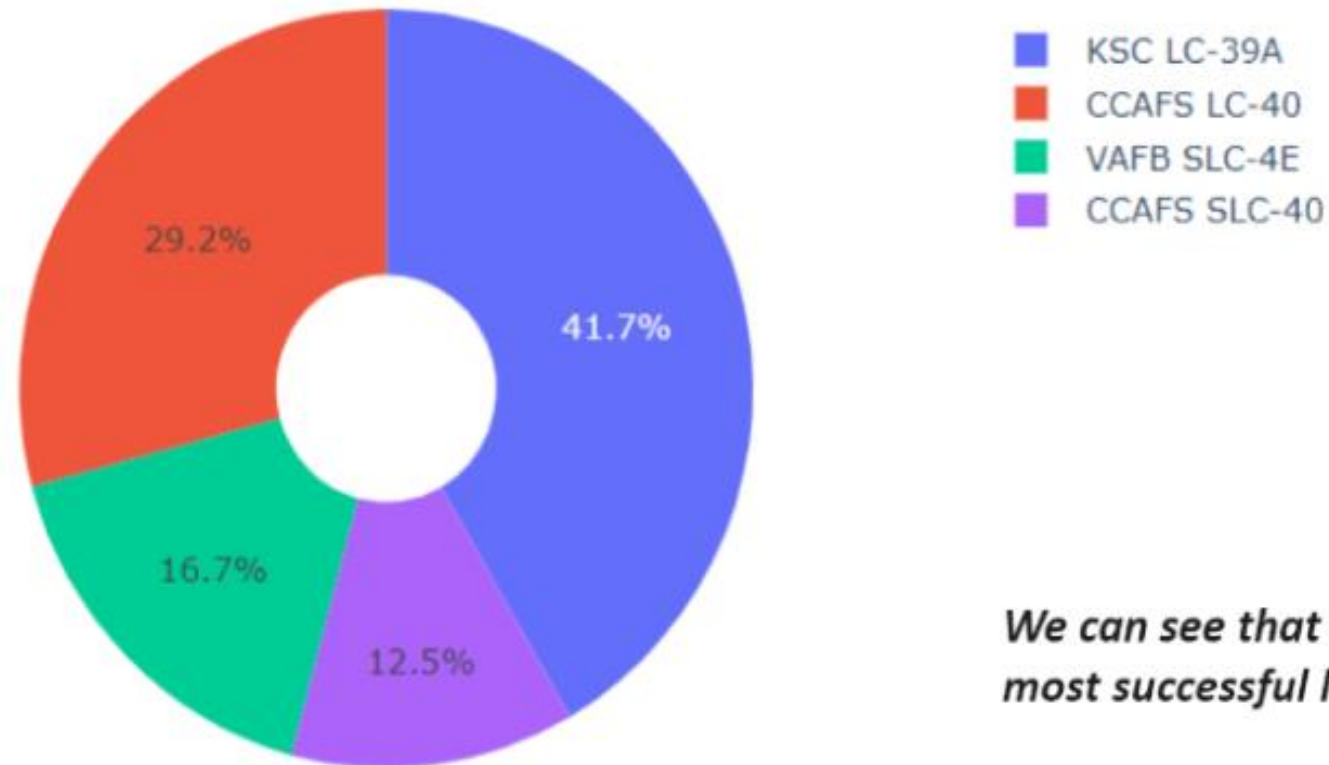


Section 4

Build a Dashboard with Plotly Dash

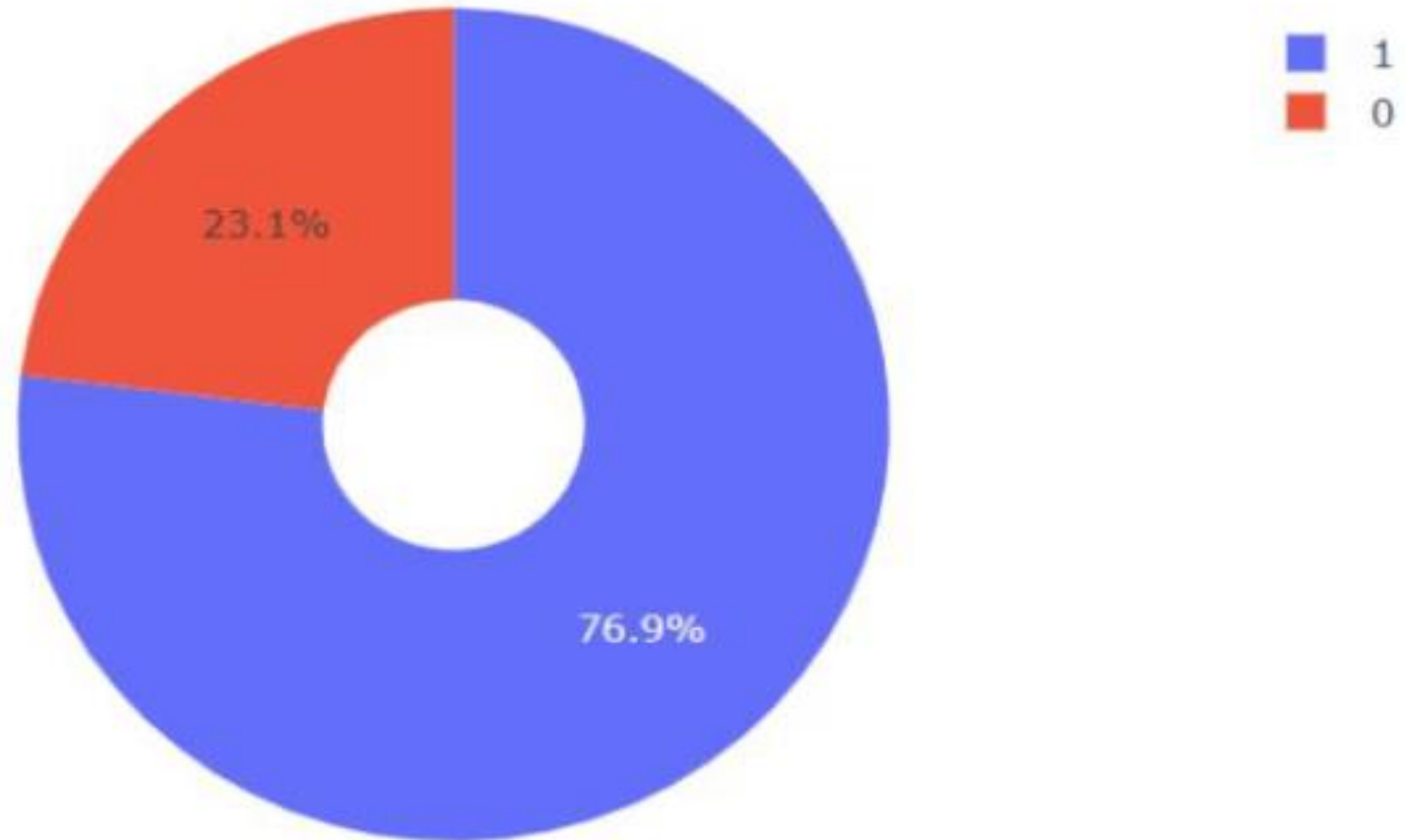
Pie chart showing the success percentage achieved by each launch site

Total Success Launches By all sites



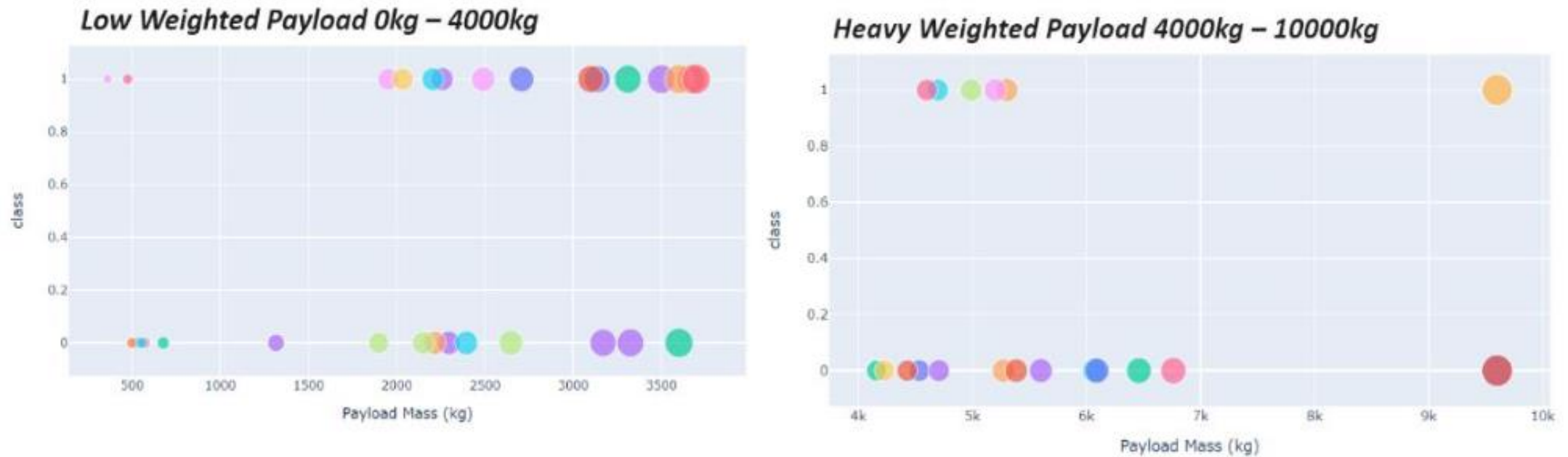
We can see that KSC LC-39A had the most successful launches from all the sites

Pie chart showing the Launch site with the highest launch success ratio



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

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

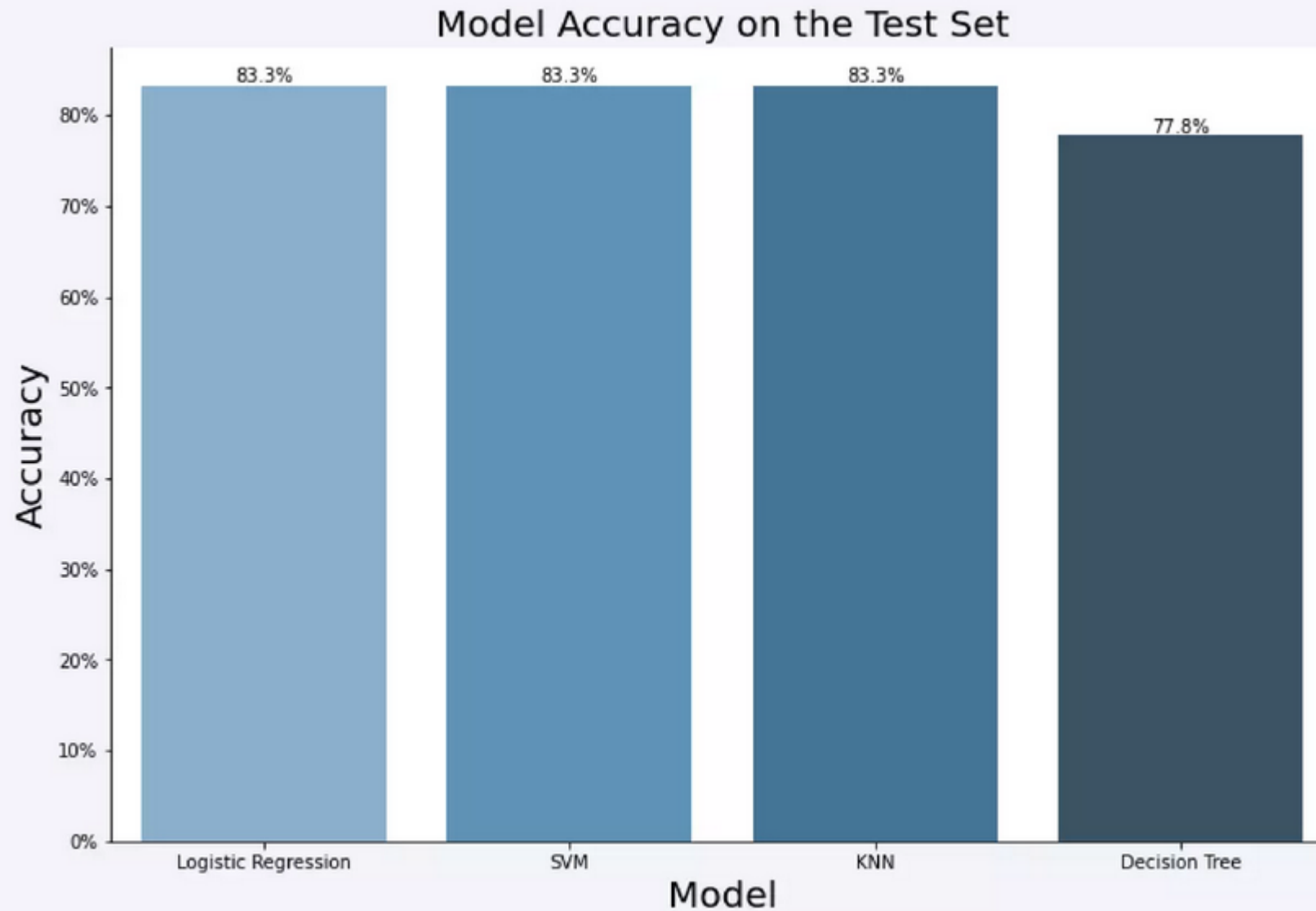


We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

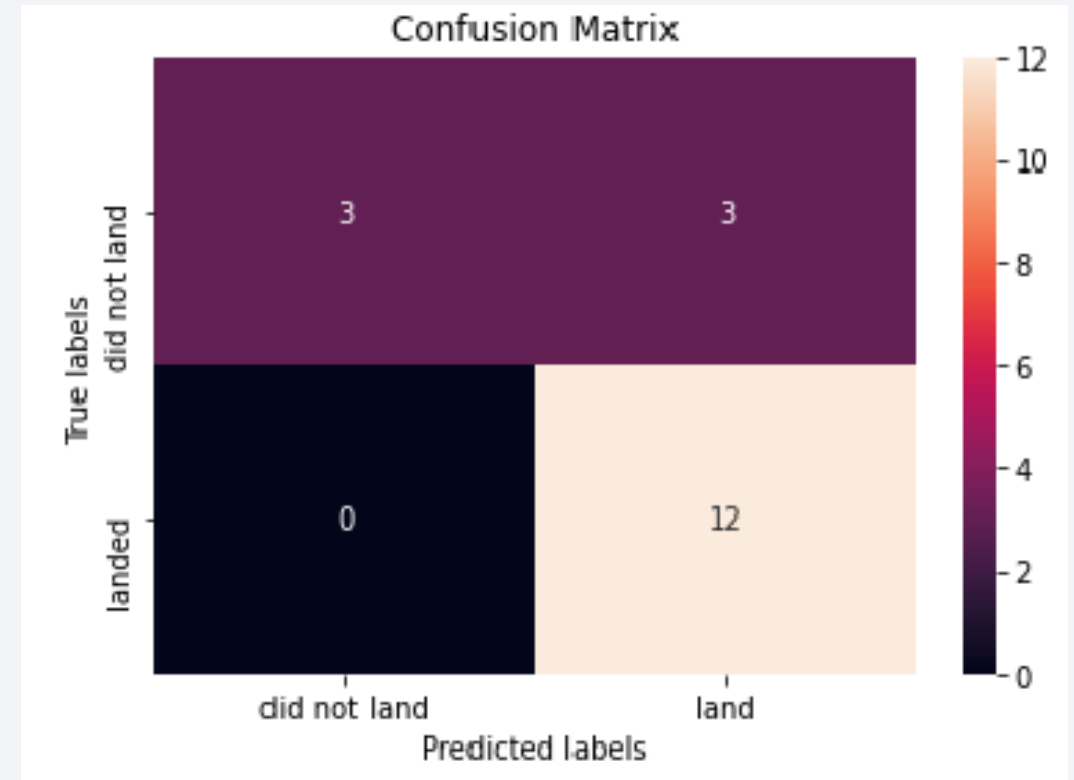
Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!

