Winning Space Race with Data Science

Bhargavi Katta
Nov 20,2022

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 (EDA)with SQL
  - Exploratory Data Analysis (EDA)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 where as other providers cost upward of 165 million dollars each. Cost difference is due to its ability to 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 to provide service at low cost.This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully

- Problems we want to find answers
  - What factors determine if the rocket will land successfully or not?
  - Which features can be chosen to make the launch successful?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and wikipedia by web scraping
- Perform data wrangling
  - One hot encoding was used for categorical values.
- 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

- Data was collected by various methods:
  - Data was collected from SpaceX API.
  - Collected data was decoded as Json using .json() function call and then into pandas data frame using .json_normalize().
  - Data was cleaned, checked for missing values and NA values were filled.
  - 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 data frame for future analysis.

Data Collection – SpaceX API

- We used get request to SpaceX API to collect data, clean basic data wrangling and formatting
- https://github.com/Bhargavik01/IBM-Data-Science/tree/master All the coding documents were available here.

Data Collection - Scraping

- We applied web scrapping to scrap Falcon 9 launch records with BeautifulSoup

- We parsed the table and converted it into a pandas data frame

- Link to the document:
- https://github.com/Bhargavik01/IBM-Data-Science/blob/master/data%20collection%20with%20web%20scraping.ipynb

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [7]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
page=requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

In [8]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup=BeautifulSoup(page.text,'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [10]:
```
# Use soup.title attribute
soup.title
```

Out[10]:
```
List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards
end of this lab

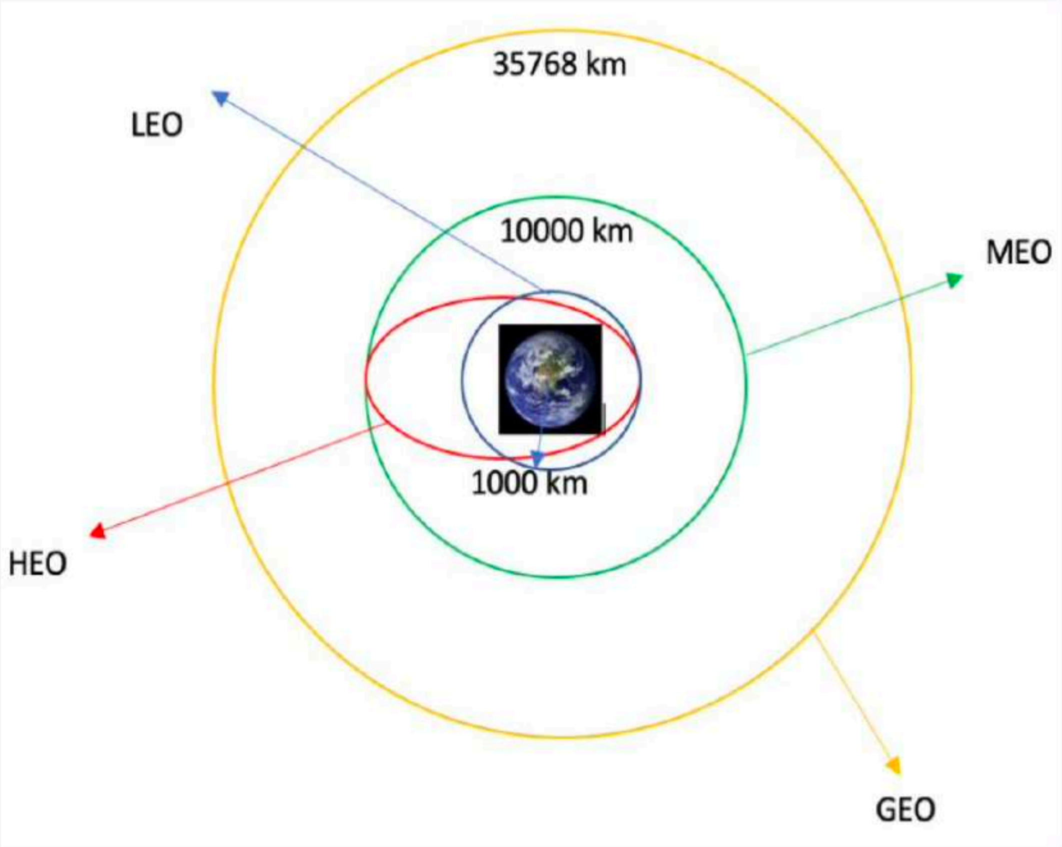
In [13]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables=soup.find_all("table")
```

Starting from the third table is our target table contains the actual launch records.

In [14]:
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

Data Wrangling

- 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 the notebook is :https://github.com/Bhargavik01/IBM-Data-Science/blob/master/Data%20wrangling%20EDA.ipynb



•

EDA with Data Visualization

Explored data by visualizing the relation 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.

Visualizations were available at:https://github.com/Bhargavik01/IBM-Data-Science/blob/master/visualization%20with%20pandas.ipynb

EDA with SQL

- Loaded the SpaceX dataset into a PostgreSQL database
- Performed EDA with SQL to get insight from the data and executed queries for the below.
  - - 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 ishttps://github.com/Bhargavik01/IBM-Data-Science/blob/master/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- 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 answered some question for instance:

Link: https://github.com/Bhargavik01/IBM-Data-Science/blob/master/Visual%20Analytics.ipynb

.

Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly dash
- Plotted pie charts showing the total launches by a certain sites
- Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

Predictive Analysis (Classification)

- Loaded the data using numpy and pandas, transformed the data, splitted data into training and testing.

- Built different machine learning models and tuned different hyperparameters using GridSearchCV.

- Used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning and found best performing model

- https://github.com/Bhargavik01/IBM-Data-Science/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

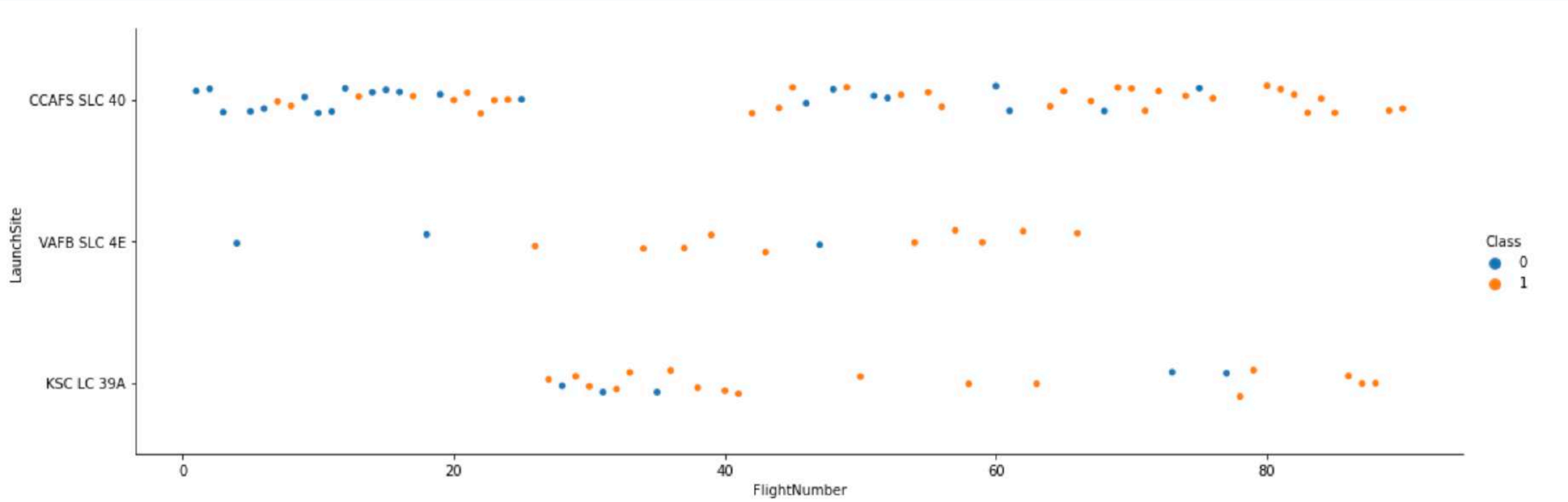
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

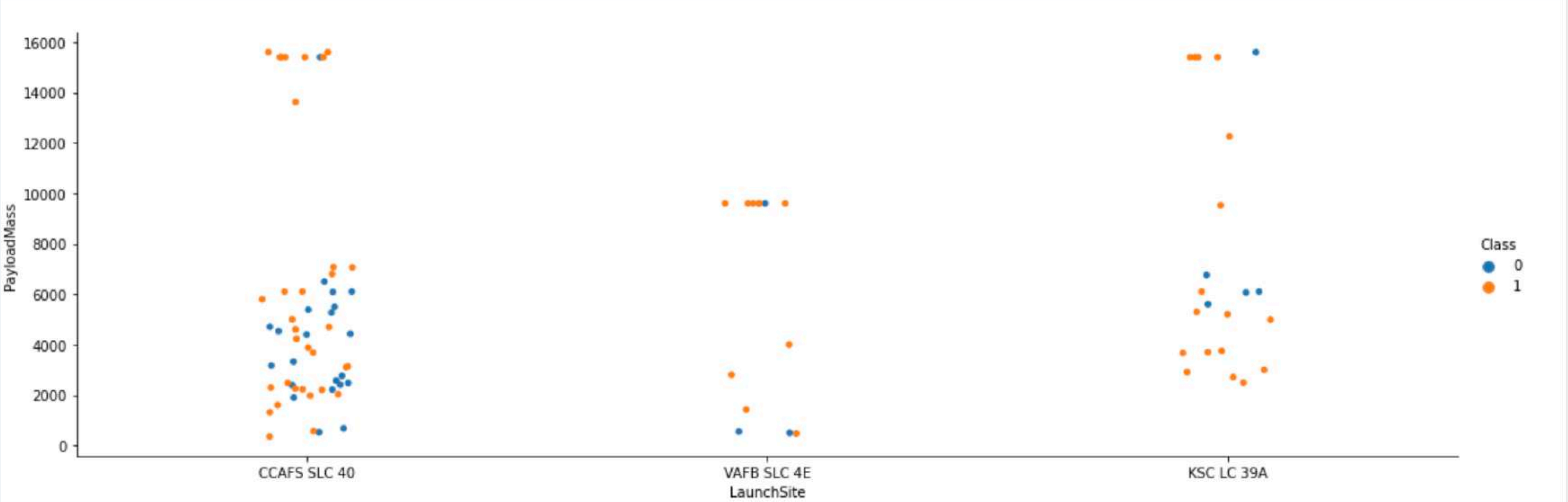
Insights drawn from EDA

Flight Number vs. Launch Site

From the plot, we found that the larger the flight amount at a launch

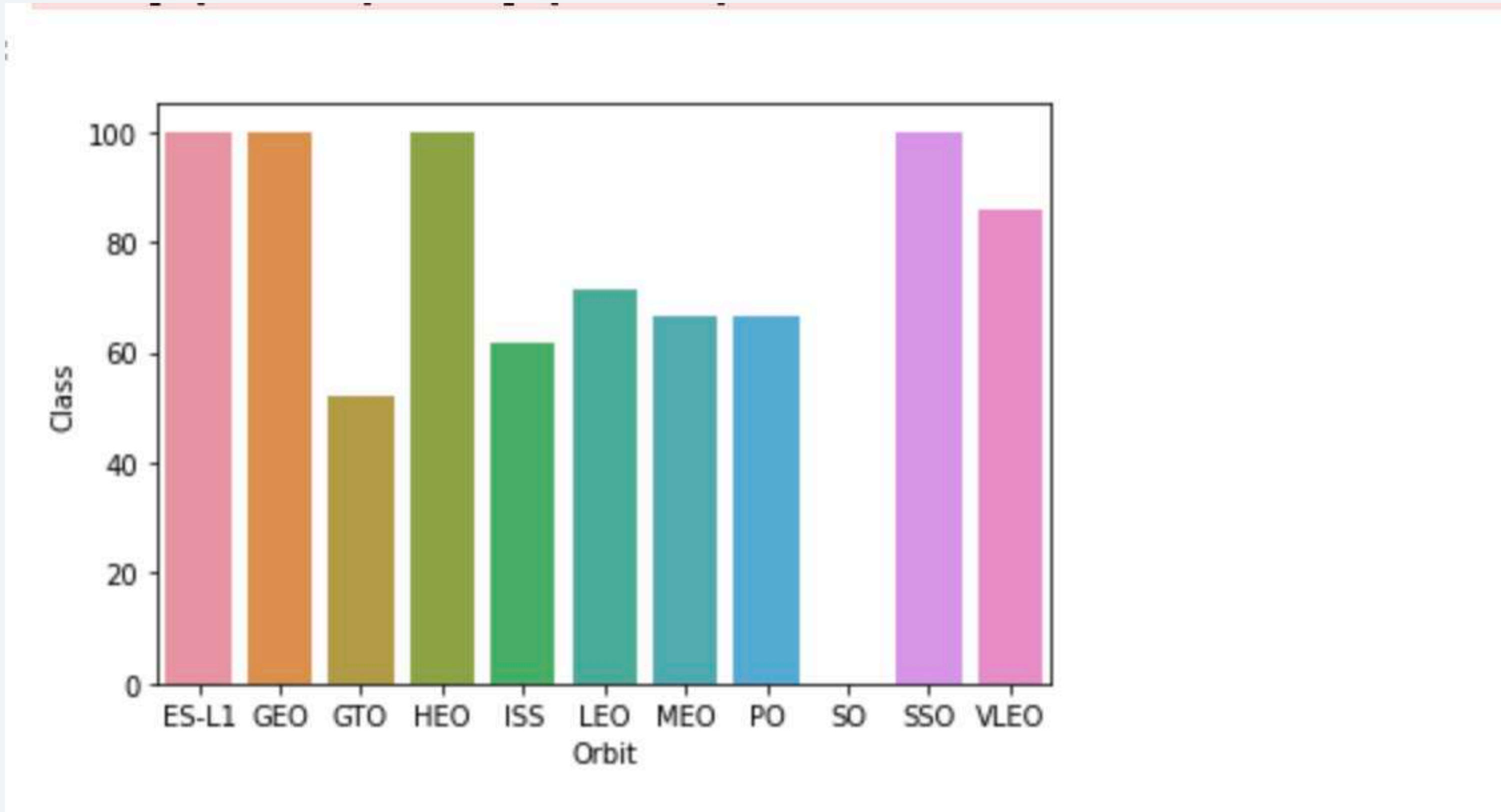


Payload vs. Launch Site



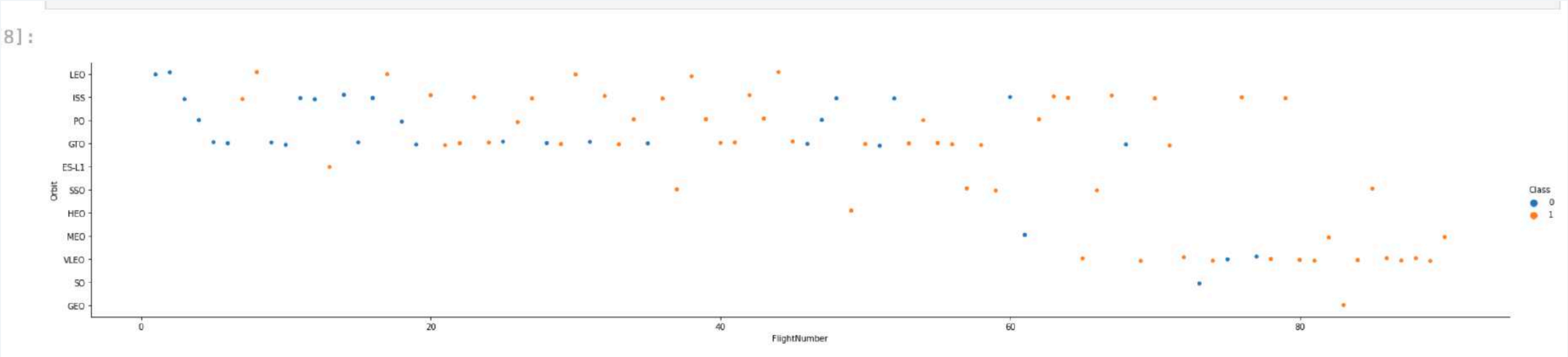
Success Rate vs. Orbit Type

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



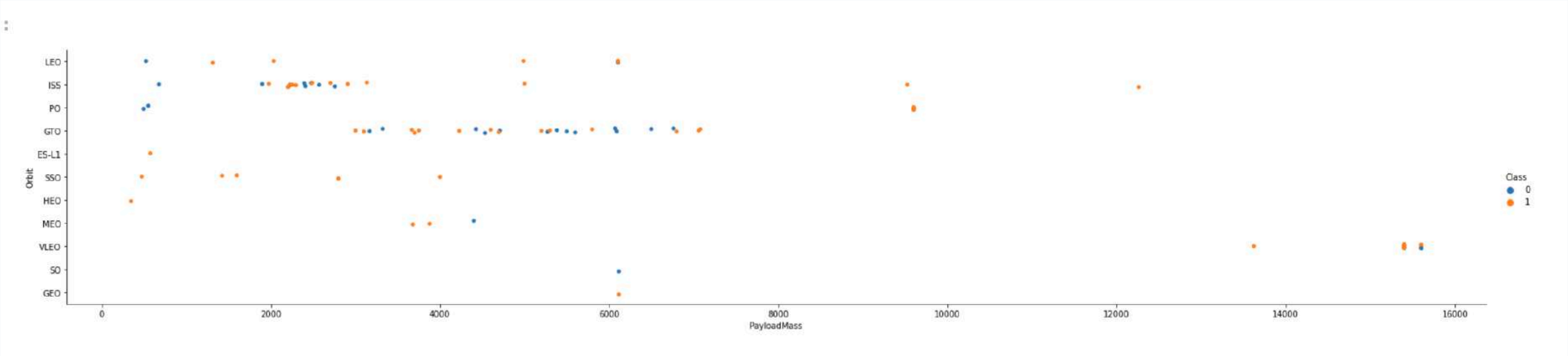
Flight Number vs. Orbit Type

The plot below shows the Flight Number vs. Orbit type. We can 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.



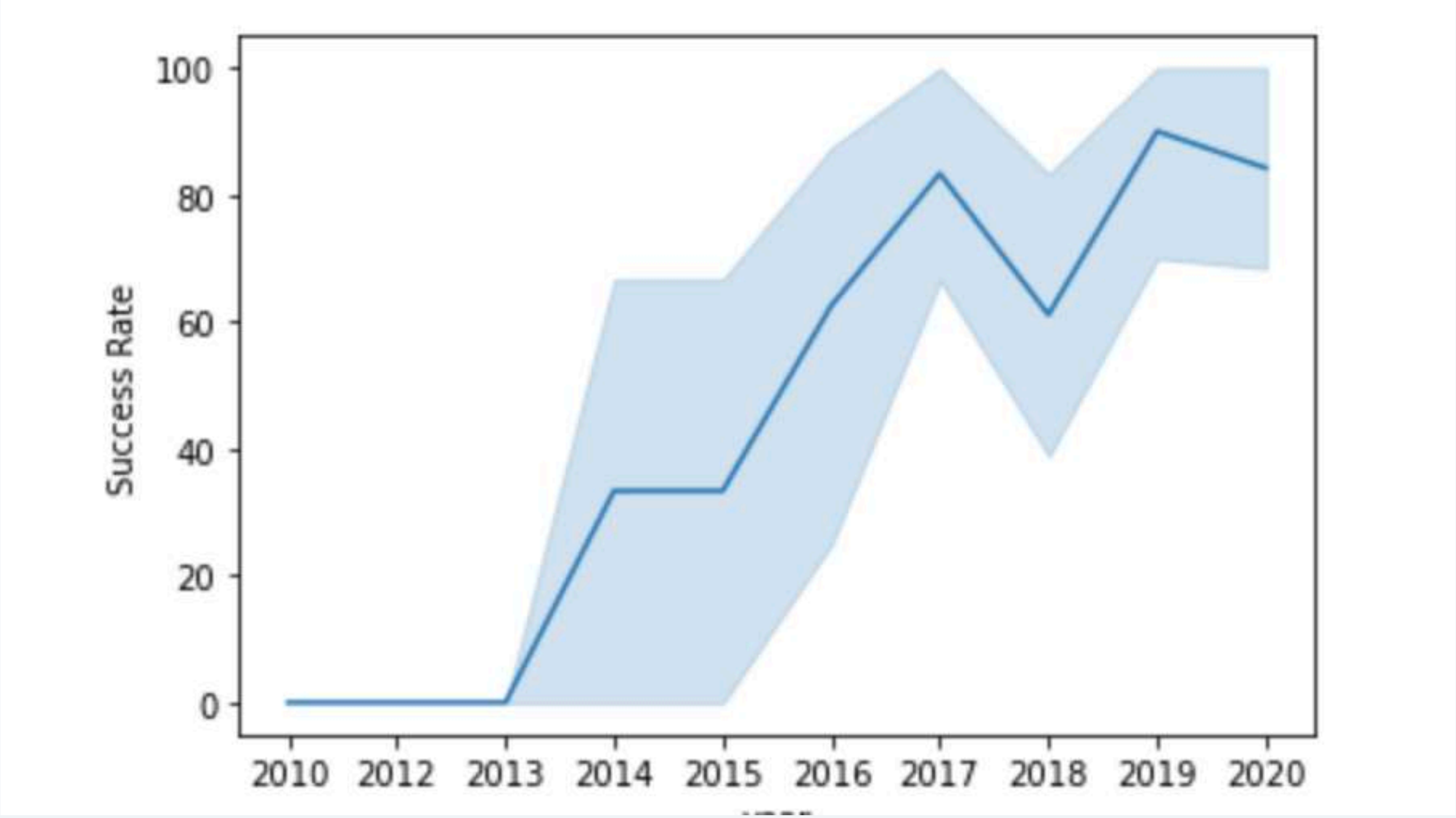
Payload vs. Orbit Type

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 success rate from 2013 kepp on increasing.



All Launch Site Names

Used the key word DISTINCT to show only unique launch sites from the SpaceX data.

-

```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- Used LIKE to get the matched launch sites

```
%sql select * from space s where s.launch_site like 'CCA%' limit 5;
```

```
 * ibm_db_sa://ff140909:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Total Payload Mass

- Used Sum(Payload_mass_kg) to get the total count from NASA(CRS)

```
%sql select sum(payload_mass__kg_) as sum from space where customer like 'NASA (CRS)';
```

```
 * ibm_db_sa://ff140909:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90108kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

SUM
45596

Average Payload Mass by F9 v1.1

- Used average(AVG) function

```
%sql select avg(payload_mass__kg_) as average from space where booster_version like 'F9 v1.1%'
```

```
 * ibm_db_sa://ff140909:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

average
2534

First Successful Ground Landing Date

- Used min(DATE) to get this.

```
%sql select min(DATE) as first from space where landing__outcome ='Success'
```

```
 * ibm_db_sa://ff140909:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90108kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

first
2018-07-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Applied many conditions to get the data

Task 6

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 booster_version from space where mission_outcome ='Success'and payload_mass__kg_ >4000 and payload_mass__kg_<6000
```

```
 * ibm_db_sa://ff140909:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

Total Number of Successful and Failure Mission Outcomes

- Used Count(*),Group by, and orderly functions

List the total number of successful and failure mission outcomes

[8]:

```
%sql SELECT mission_outcome, count(*) as Count FROM space GROUP by mission_outcome ORDER BY mission_outcome
```

```
 * ibm_db_sa://ff140909:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90108kqblod8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

[8]:

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Used sub query to get highest payload

```
%sql select booster_version from space where payload_mass__kg_ =(select max(payload_mass__kg_) as highest_payload from space)
```

```
 * ibm_db_sa://ff140909:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4

2015 Launch Records

- Used MONTHNAME(date) and LIKE to get The data

```
%sql select MONTHNAME(DATE) as Month, landing__outcome, booster_version, launch_site from space where DATE like '2015%' AND landing__outc

 * ibm_db_sa://ff140909:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

MONTH	landing__outcome	booster_version	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Used count(*) and groupie

```
%sql select landing__outcome, count(*) as count from space where Date >= '2010-06-04' AND Date <= '2017-03-20' GROUP by landing__outcome
```

```
 * ibm_db_sa://ff140909:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90108kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

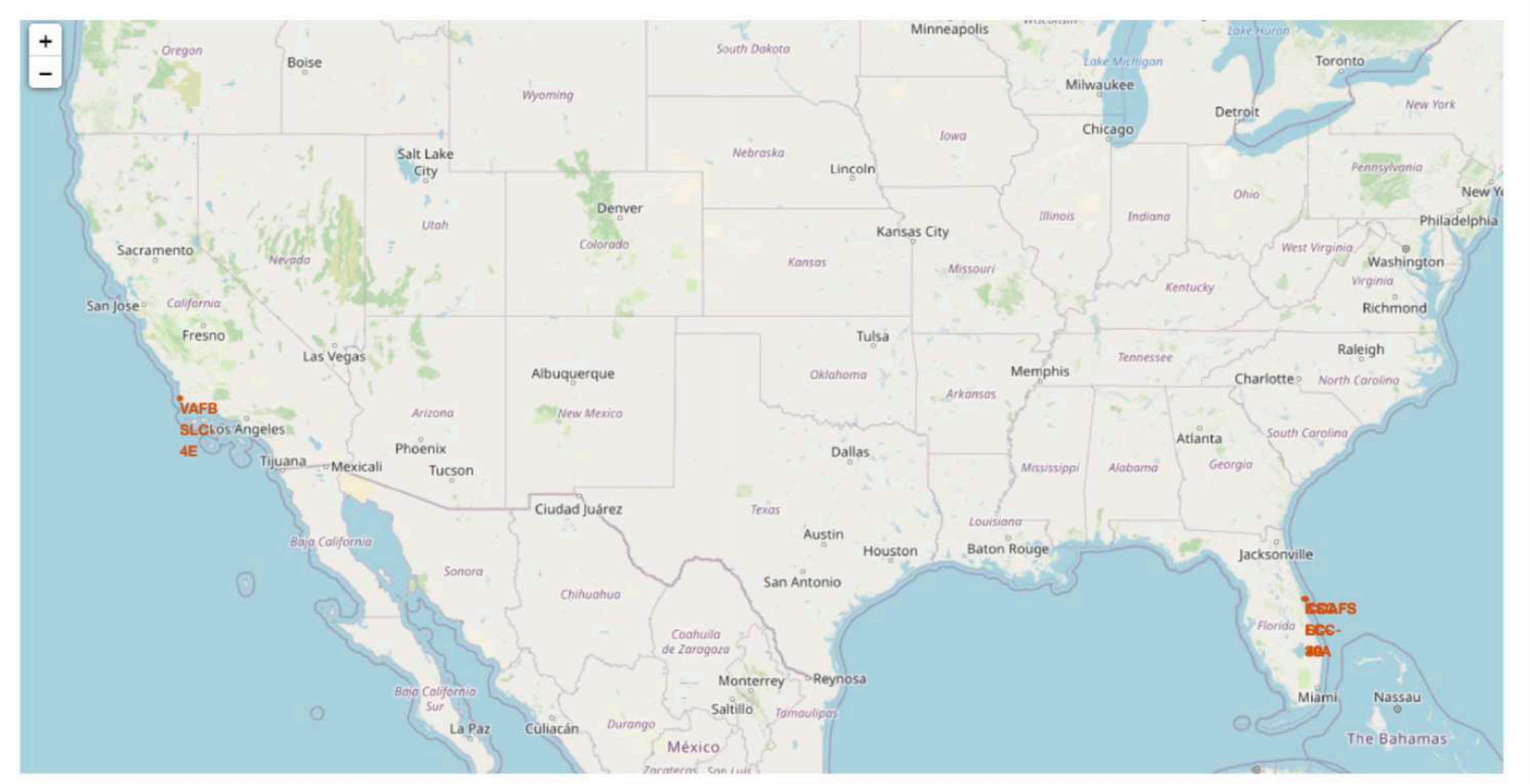
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

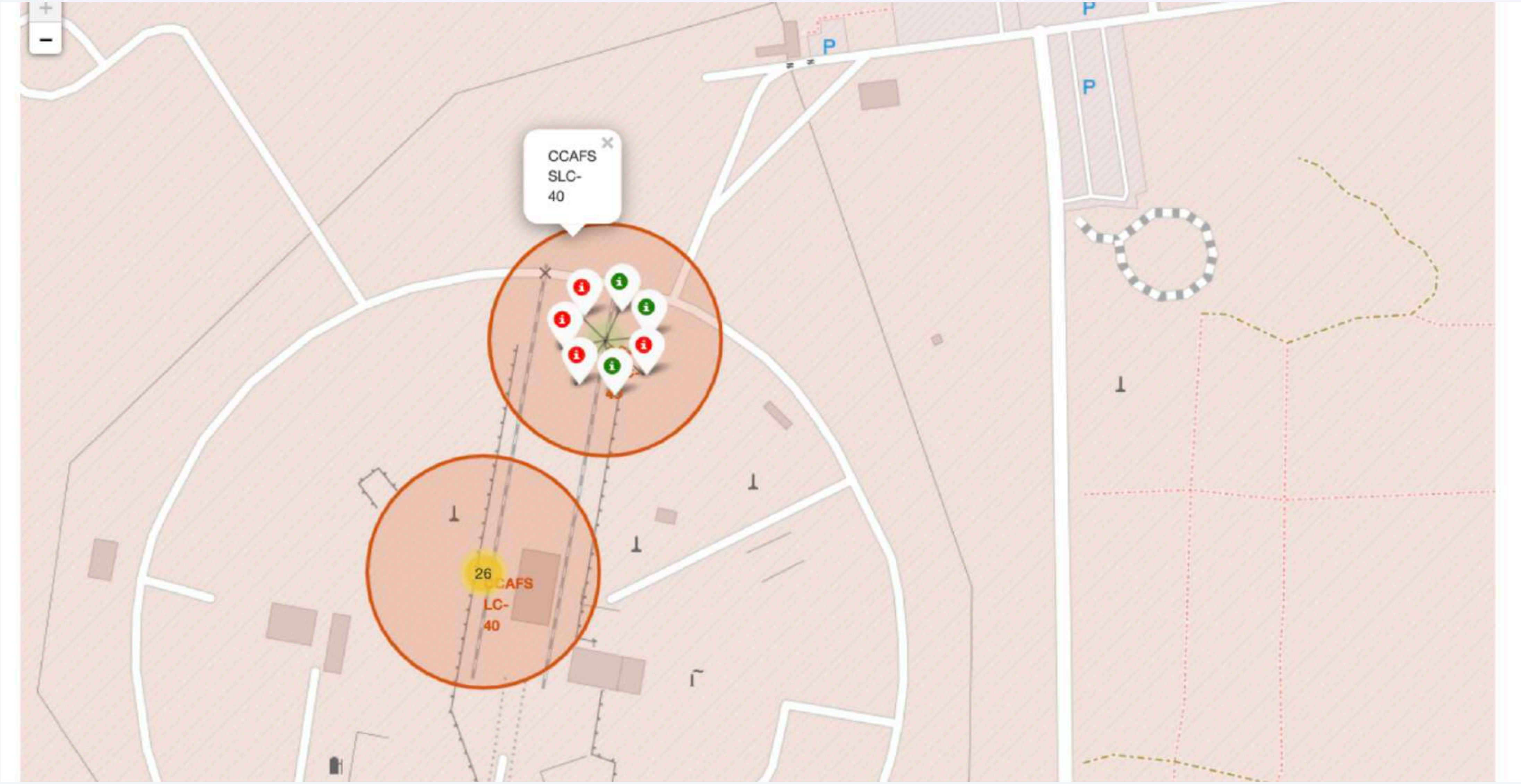
Launch Sites Proximities Analysis

Launch Sites global. Map. Markers

- Space X launch sites are located in Florida and california

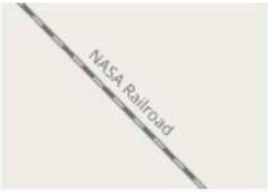


Launch sites with color labels, green=sucess,red

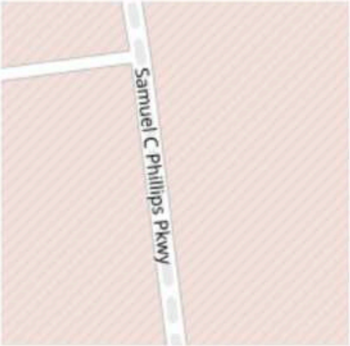


Launch Site distance to landmarks

A railway map symbol may look like this:



A highway map symbol may look like this

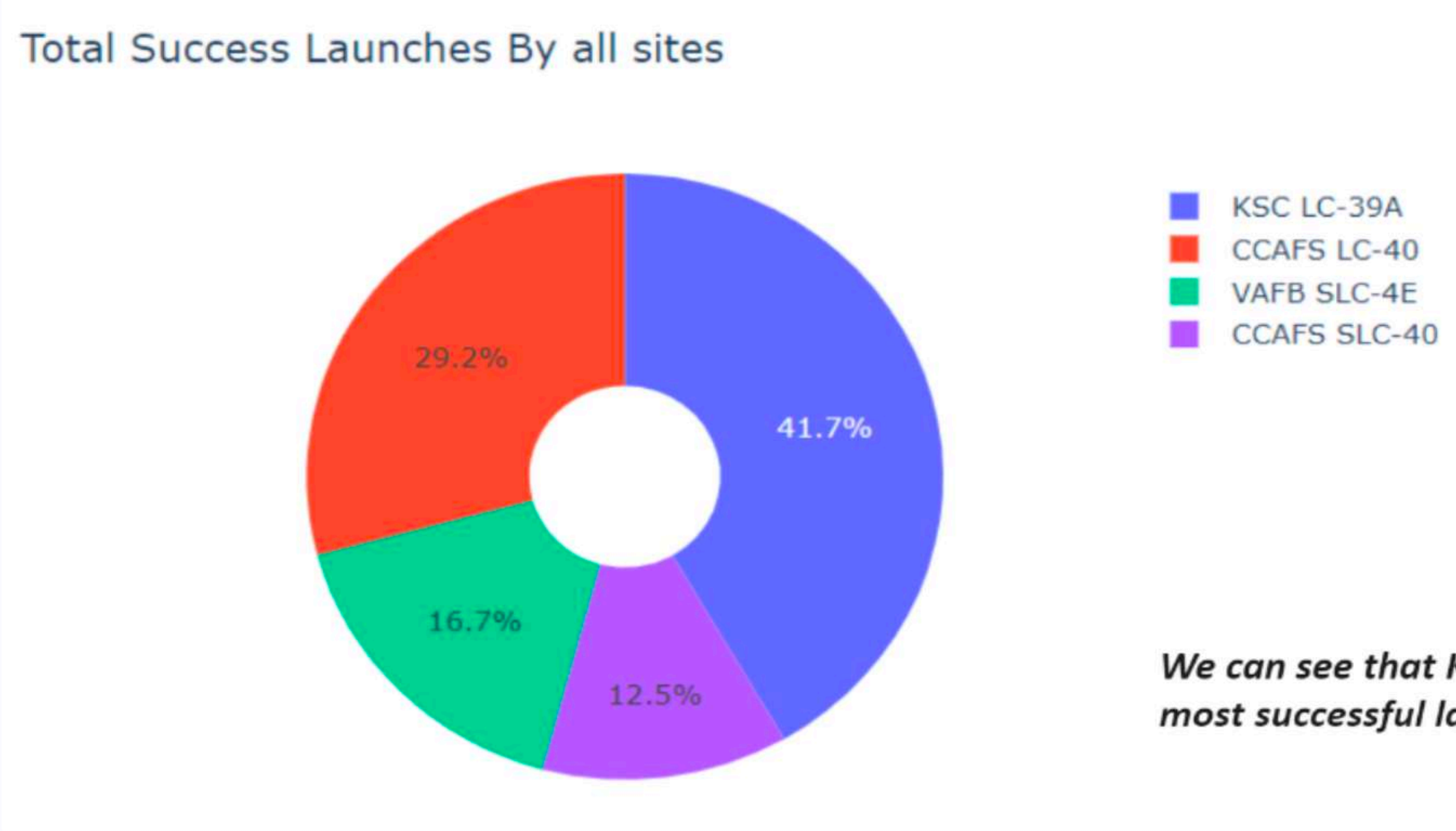


A city map symbol may look like this:

Section 4

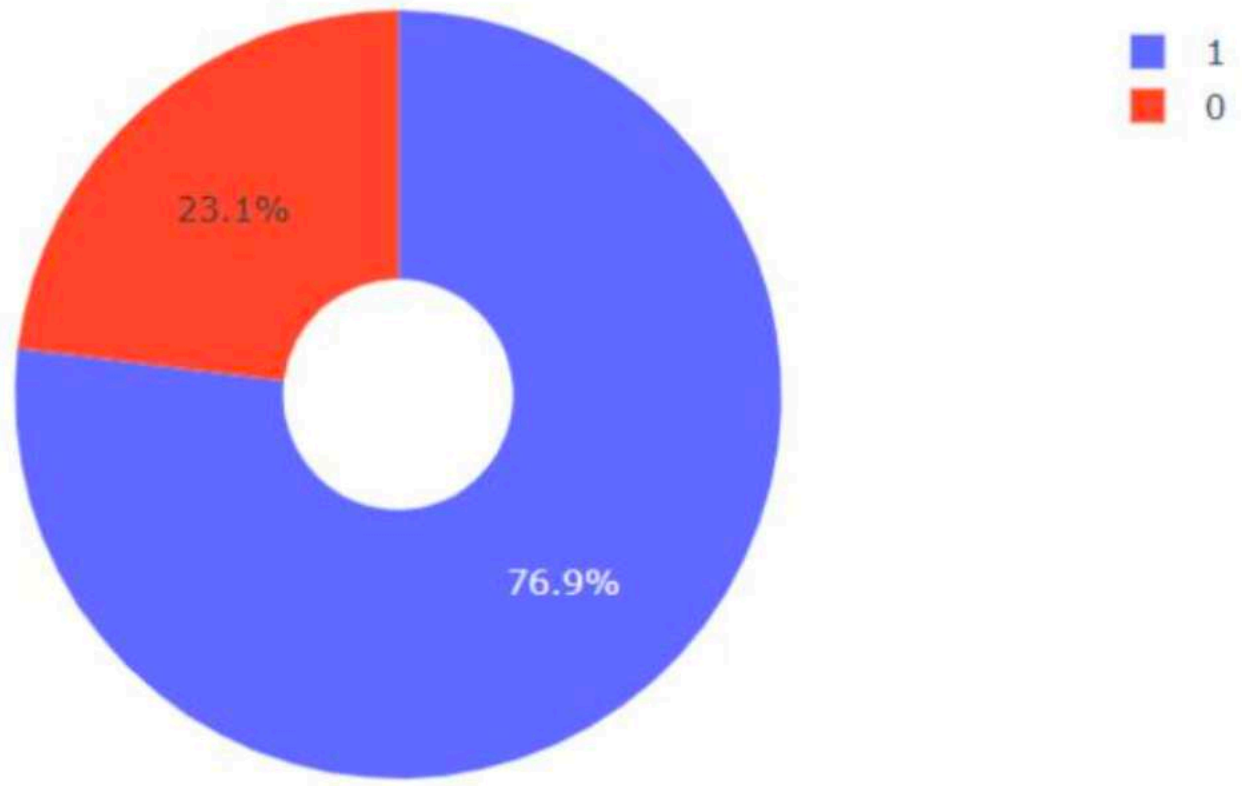
Build a Dashboard with Plotly Dash

Pie chart showing the success of launch site



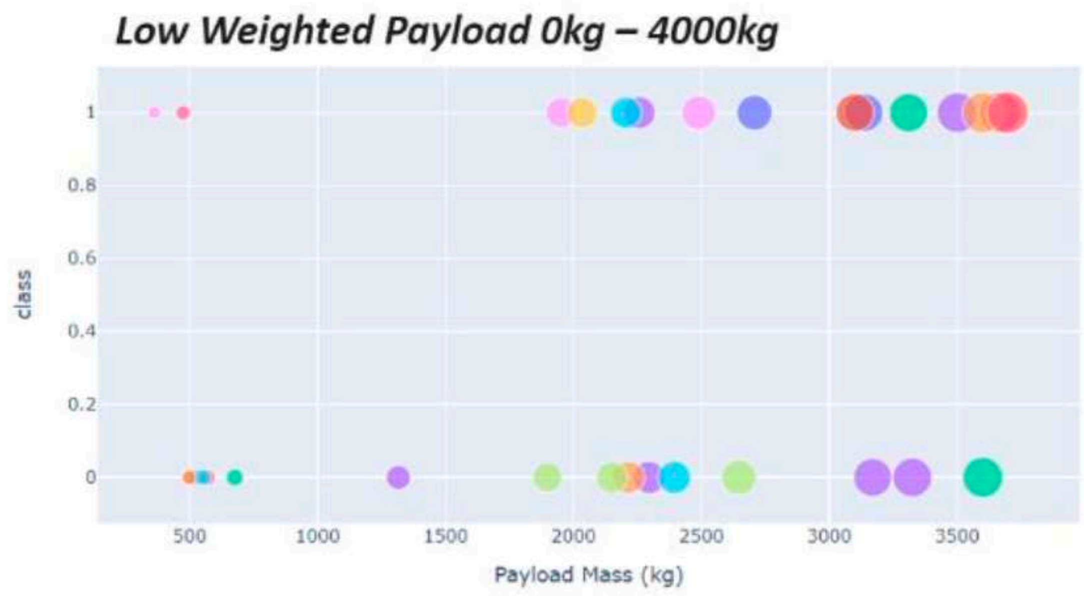
We can see that KSC LC-39A had the most successful launches from all the sites

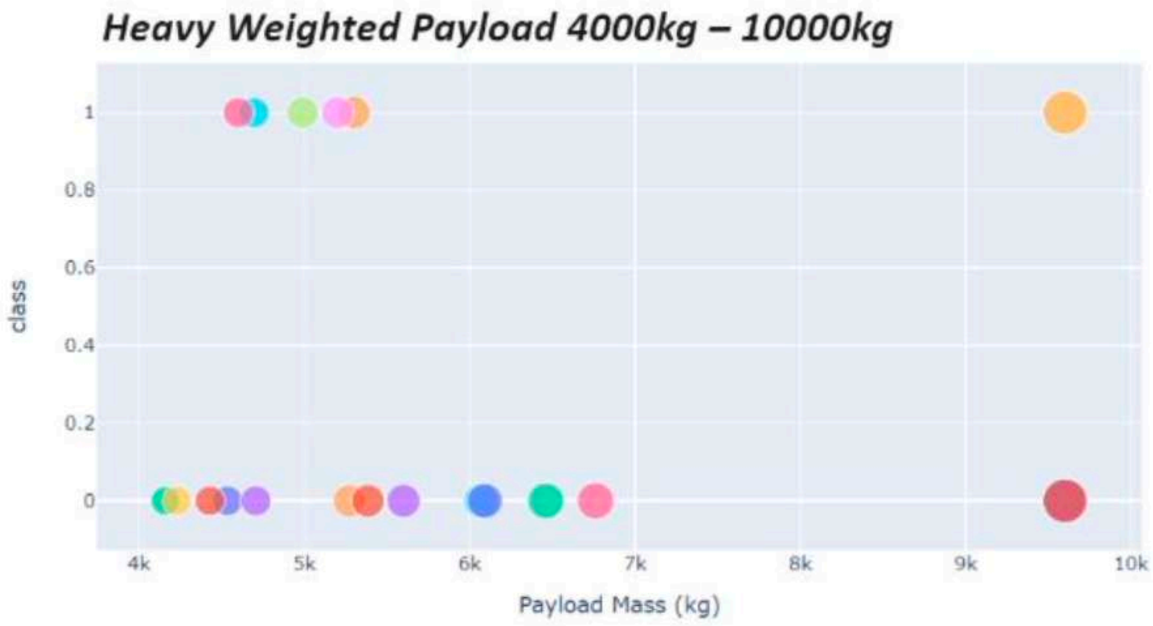
Pie chart to desire launch sites with high success ratio’s



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

Scatter plot of Payload vs Launch outcome





Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All algorithms gave similar accuracy except Decision tree

Find the method performs best:

```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test, Y_test))
```

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.5555555555555556
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```

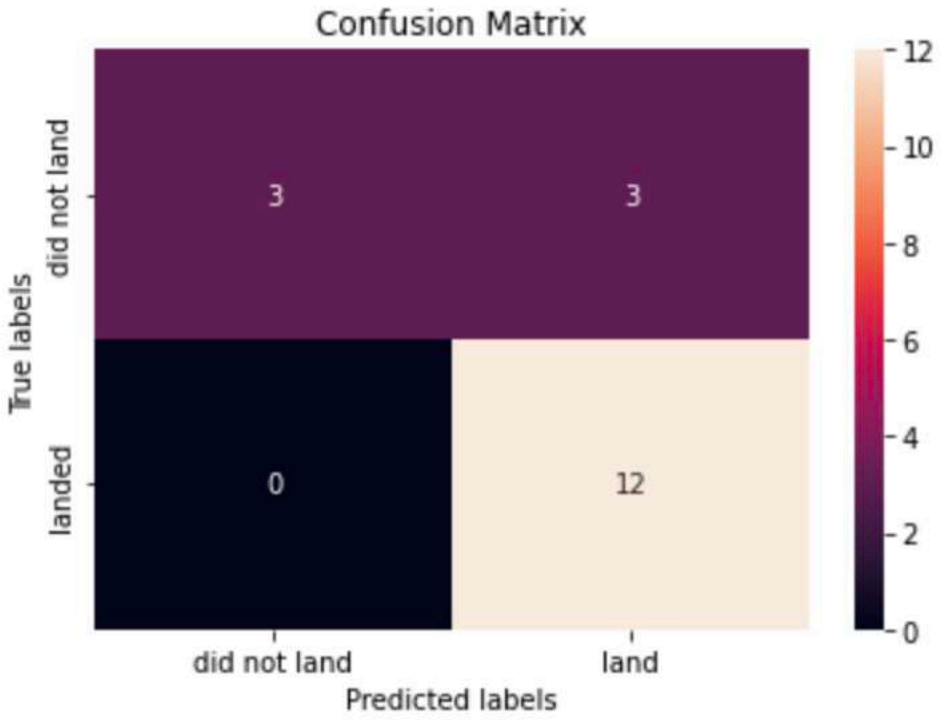
-

Confusion Matrix

The confusion matrix for the Logistic regression shows that the classifier can distinguish between the different classes.

.

```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



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!