IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rodrigo Fernandes
April 22nd, 2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- **The following methodologies were used to analyze data:**

- Data Collection using web scrapping and SpaceX API;
- Exploratory Data Analysis (EDA), including data wrangling, data visualization, and interactive visual analytics;
- Machine Learning Prediction.

- **Summary of all results**

- Machine Learning Prediction showed the best model to predict which characteristics are important to drive this opportunity in the best way, using all the collected data;
- EDA allowed to identify which features are the best to predict the success of launchings;
- It was possible to collect valuable data from public sources.

Introduction

Project background and context

- The aim of this project is to predict if the Falcon 9 first stage will successfully land. SpaceX says on its website that the Falcon 9 rocket launch cost 62 million dollars. Other providers cost upward of 165 million dollars each. The price difference is explained by the fact that SpaceX can reuse the first stage. By determining if the stage will land, we can determine the cost of a launch. This information is interesting for another company if it wants to compete with SpaceX for a rocket launch.

Desirable answers:

- What factors determine if the rocket will land successfully?
- What operating conditions need to be in place to ensure a successful landing?
- Is it possible to estimate the total cost for launches, by predicting successful landings of the first stage of rockets?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

Data was collected from the following sources:

1. SpaceX REST API (https://api.spacexdata.com/v4/rockets/)
2. Web Scraping from Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)

- Perform data wrangling
  - Dropping unnecessary columns
  - One Hot Encoding for classification models

Methodology

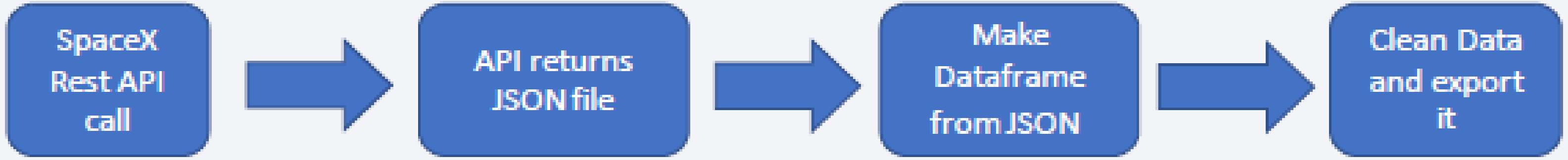
Executive Summary

- 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

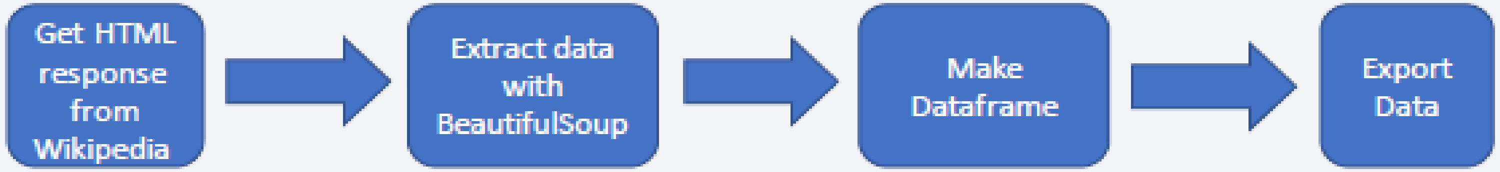
- Datasets were collected from SpaceX REST API and Web Scraping Wikipedia.
- The information obtained by the API are rocket, launches, and payload information.

    - The Space X REST API URL is api.spacexdata.com/v4/



- The information obtained by web scraping Wikipedia are launches, landing, and payload information.

    - URL is https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



Link to code

Data Collection – SpaceX API

1. Getting Response from API → **2. Convert JSON Response to a Pandas Dataframe** → **3. Transform data** →

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)
```

```
In [11]: # Use json_normalize meethod to convert the json result into a dataframe
         data = pd.json_normalize(response.json())
```

```
In [16]: # Call getBoosterVersion
         getBoosterVersion(data)
```

the list has now been update

```
In [17]: BoosterVersion[0:5]

Out[17]: ['Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 9']
```

we can apply the rest of the functions here:

```
In [18]: # Call getLaunchSite
         getLaunchSite(data)

In [19]: # Call getPayloadData
         getPayloadData(data)

In [20]: # Call getCoreData
         getCoreData(data)
```


[Link to code]()

Data Collection – SpaceX API

4. Create Dictionary with data

5. Create dataframe

6. Filter dataframe

7.Export to csv file

```
In [21]: 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}
```

```
In [22]: # Create a data from launch_dict
dataframe = pd.DataFrame(launch_dict)
```

```
In [63]: # Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = dataframe[dataframe.BoosterVersion == 'Falcon 9']
```

```
In [60]: data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

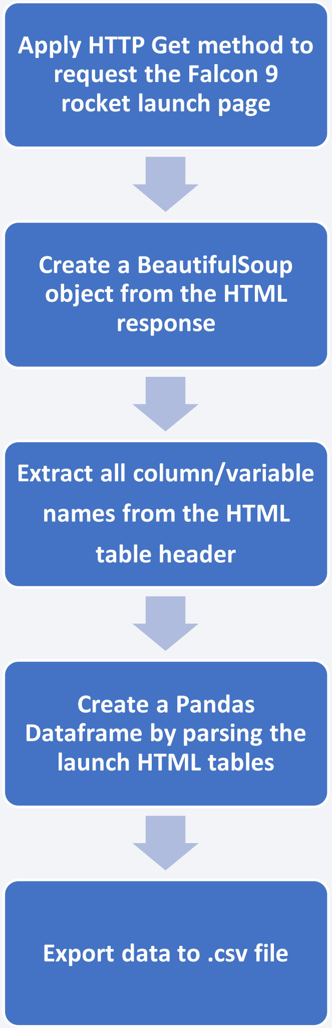
Link to code

Data Collection - Scraping

- We performed web scraping to look for Falcon 9 launch records with BeautifulSoup;
- We extracted those records from a HTML in Wikipedia;
- We parsed the table and converted it into a pandas dataframe;
- And converted the data to a .csv file.

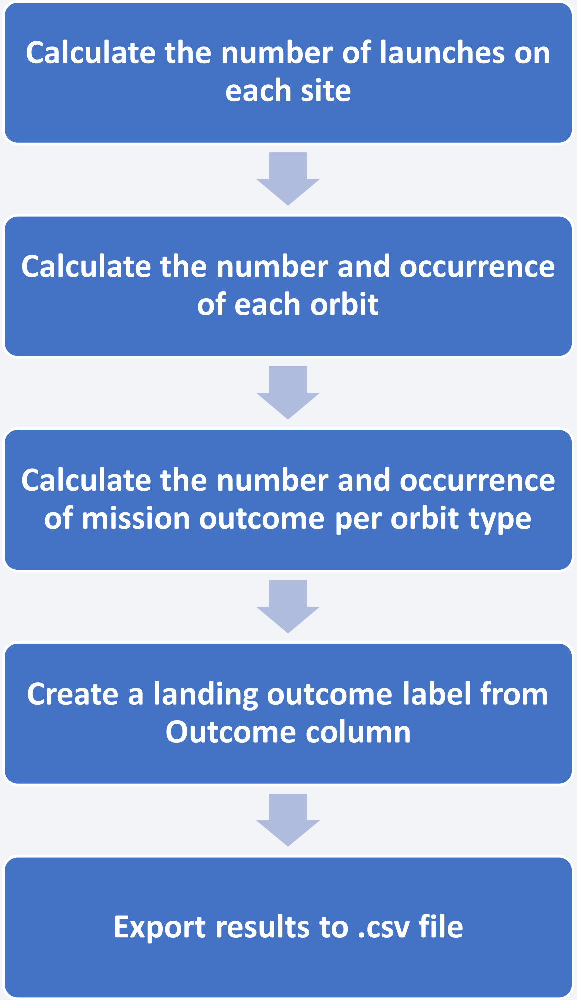
Link to code

Steps:



Data Wrangling

- We performed Exploratory Data Analysis (EDA) and determined trained labels;
- We calculated the number of launches at each site, the number and the occurrence of each orbit;
- A landing outcome label from outcome column was created;
- The results were exported to a .csv file.



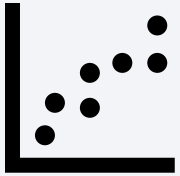
[Link to code]

EDA with Data Visualization

- To explore the relationship between pair of variables involved in the launch process, we plotted scatter charts, a bar graph, and a line graph:

Scatter Charts

- Flight Number vs. Launch Site
- Flight Number vs. PayloadMass
- Flight Number vs. Orbit Type
- PayloadMass vs. Launch Site
- PayloadMass vs. Orbit Type



Bar Graph

- Success rate vs. Orbit Type

Line Graph

- Success rate vs. Year

[Link to code]

EDA with SQL

We applied EDA with SQL to get insights from the data. We wrote SQL queries with the following objectives:

- 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 achieved;
- List the name of the boosters which have success in drone ship and have payload mass greater than 4,000 kg but less than 6,000 kg;
- List the total number of successful and failure mission outcomes;
- List the name of the booster versions which have carried the maximum payload mass;
- List the failed landing outcomes in drone ship, their booster versions, and launch site names for the year 2015;
- Rank the count of landing outcomes between the dates 2010-06-04 and 2017-03-20 in descending order.

[Link to code]()

Build an Interactive Map with Folium

We created Folium Maps and added markers, circles, marker clusters, and lines to them:

- Markers: indicate relevant points, like launch sites;
- Lines: used to indicate distances between two coordinates;
- Marker Clusters: indicate a group of events in each coordinate (e.g., launches in a site);
- Circles: indicate highlighted areas around specific coordinates.

Link to code

Build a Dashboard with Plotly Dash

- Three graphics were plotted and used to visualize data:

1. Pie Chart showing the percentage of successful launches by site;
2. Pie Chart showing the percentage of successful/failed launches per each site;
3. Mass Payload per Booster Version in successful or failed missions.

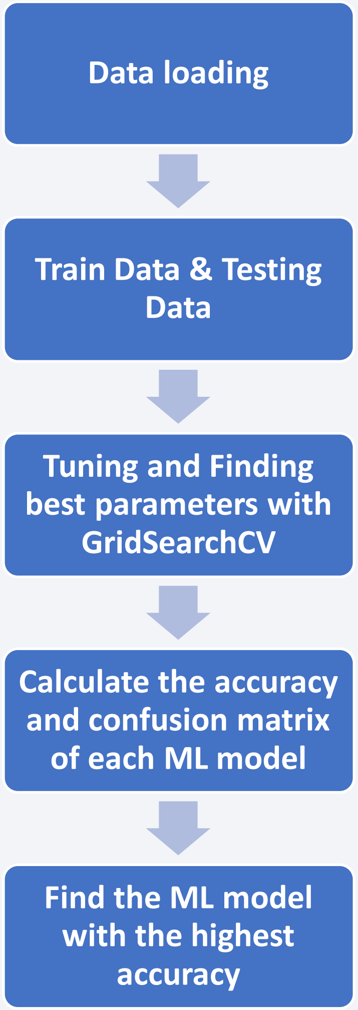
Link to code

Predictive Analysis (Classification)

- Data was loaded, transformed and splitted into two groups: training data and testing data;
- Different machine learning models were used and tuned with GridSearchCV:

➢ Logistics Regression method (**LogReg**)
➢ Support Vector Machine method (**SVM**)
➢ Decision Tree Method (**Tree**)
➢ K nearest neighbors method (**KNN**)

- Accuracy of each model was determined;
- We determined the confusion matrix of ML model;
- We determined the best performing classification model.

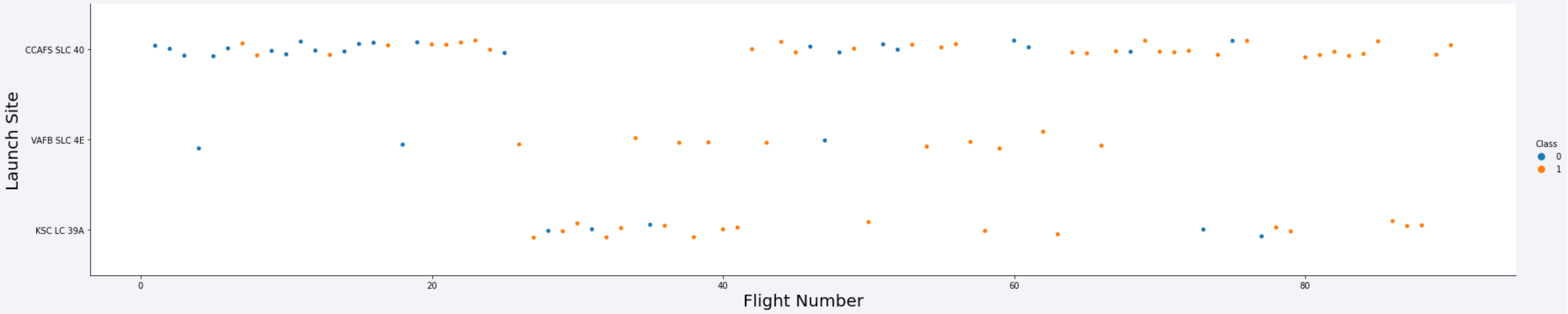


Link to code

Section 2

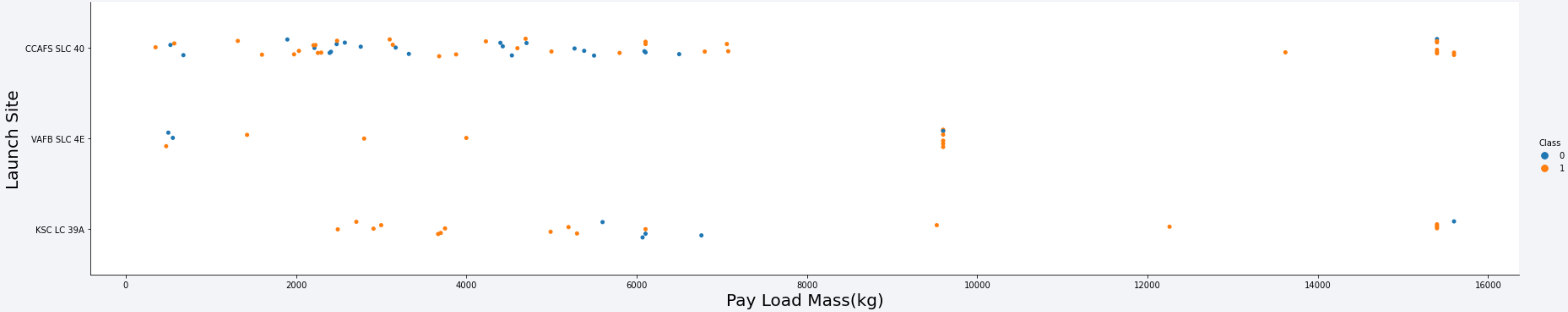
Insights drawn from EDA

Flight Number vs. Launch Site



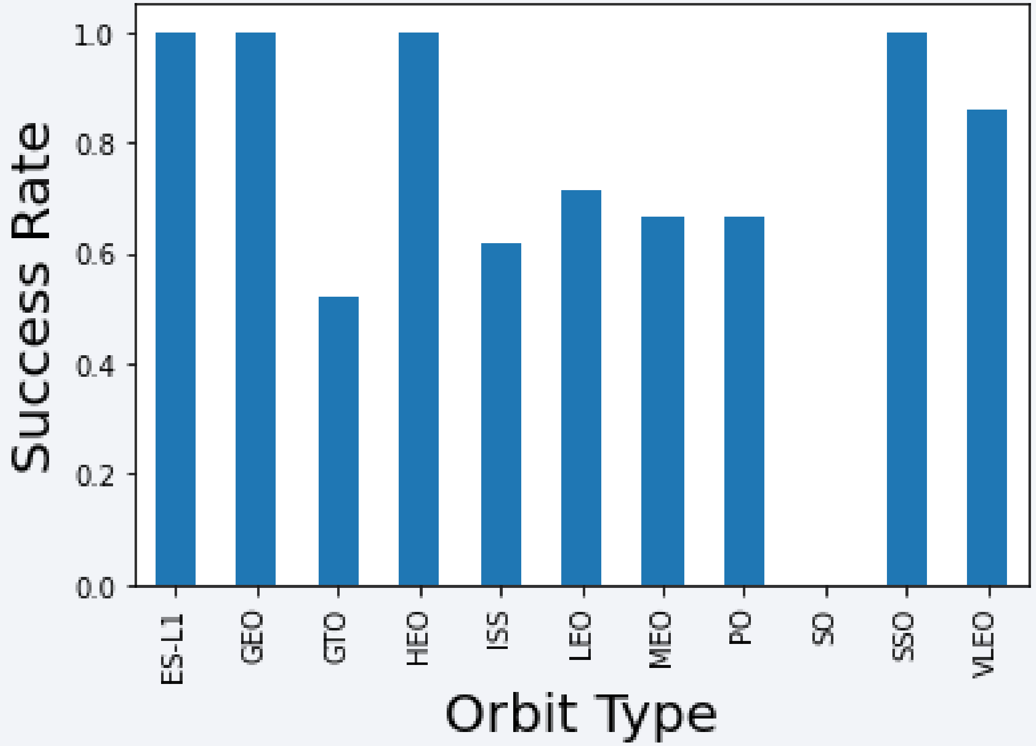
- We can infer that, for each site, the success rate increases with the number of launches performed.

Payload vs. Launch Site



- Payloads over 9,000kg (about the weight of a school bus) have an excellent success rate;
- Payloads over 10,000kg seems to be possible only on CCAFS SLC 40 and KSC LC 39A launch sites.

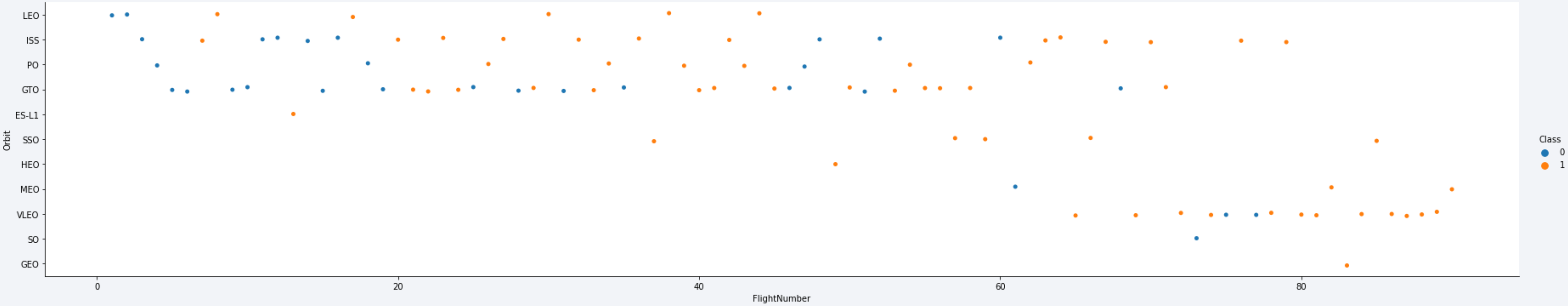
Success Rate vs. Orbit Type



Orbit	Success rate
ES-L1	1.000000
GEO	1.000000
GTO	0.518519
HEO	1.000000
ISS	0.619048
LEO	0.714286
MEO	0.666667
PO	0.666667
SO	0.000000
SSO	1.000000
VLEO	0.857143

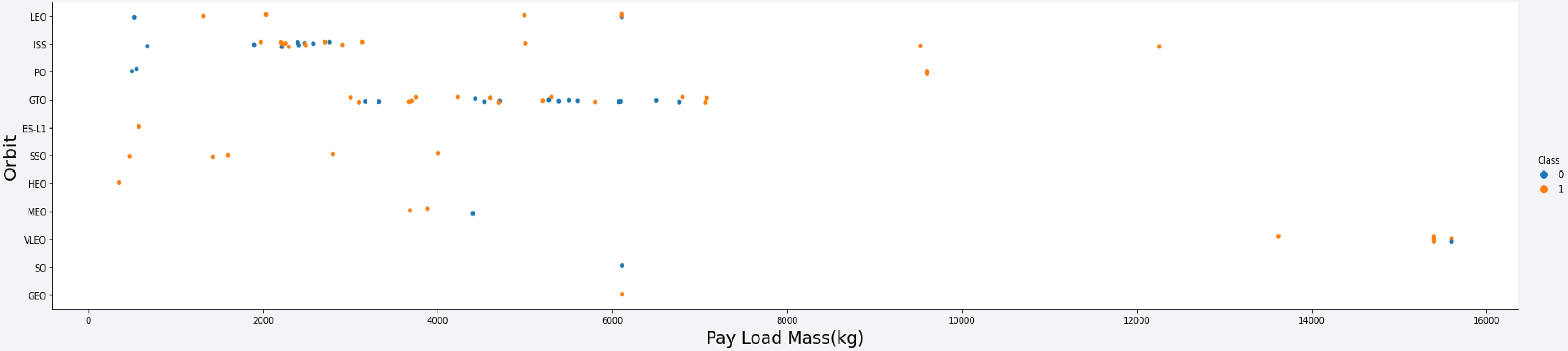
- From the results displayed above, we conclude that the best success rate (100 %) occurs in 4 orbits (ES-L1, GEO, HEO, SSO).

Flight Number vs. Orbit Type



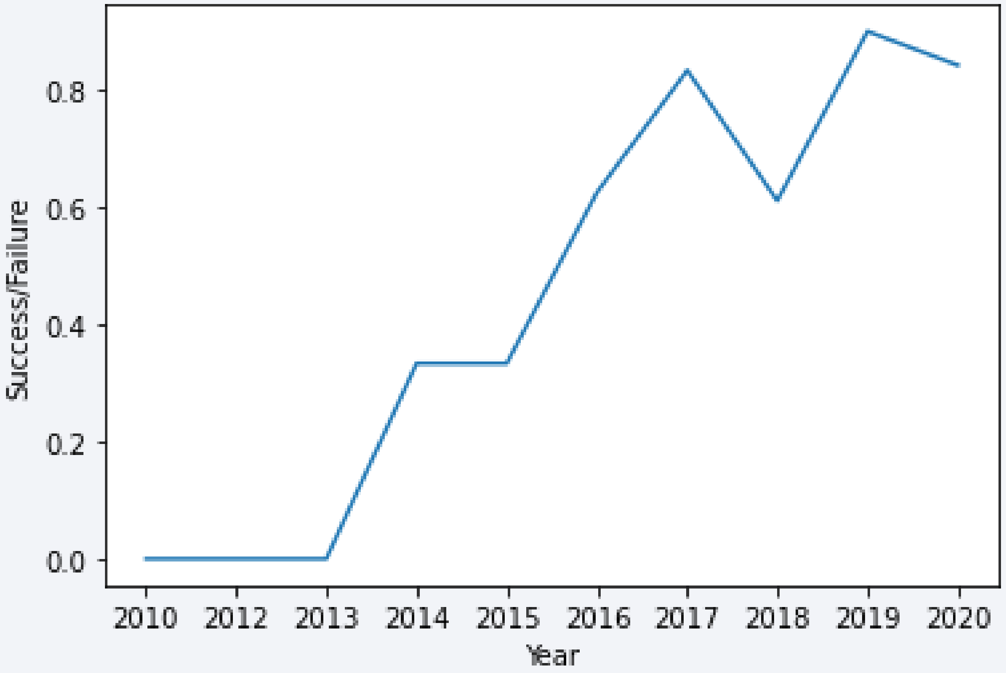
- We observe that in the LEO orbit, success is related to the number of flights;
- For some orbits (for example GTO), there is no relation between the success rate and the number of flights;
- Higher success rate of some orbits can be explained by experience acquired from previous launches.

Payload vs. Orbit Type



- The payload weight can have a significant influence on the success rate of the launches in certain orbits;
- Heavier payloads improve the success rate in LEO orbit;
- Apparently, there is no correlation between payload and success rate in GTO;
- ISS orbit has the widest range of payload mass and a good rate of success.

Launch Success Yearly Trend



- We note that the success rate started increasing in 2013 and kept until 2020;
- The first three years were a period of adjustments and technological improvements for the years that followed.

All Launch Site Names

- We used the **DISTINCT** function to show unique launch sites from the SpaceX data:

```
In [8]: %sql select distinct(LAUNCH_SITE) from "SPACEXDATASET"
```

Output:

Out[8]:

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

Launch Site Names Begin with 'CCA'

- We used the query below to display 5 records where launch site names begin with ‘CCA’:

```
In [9]: %sql select * from "SPACEXDATASET" where LAUNCH_SITE like 'CCA%' limit 5;
```

The **WHERE** clause followed by **LIKE** clause filters launch sites that contain the substring CCA.

LIMIT 5 shows 5 records from filtering.

Output

Out[9]:

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

- The query below requests the sum of all payload mass carried by boosters launched by NASA (CRS):

```
In [10]: %sql select SUM(PAYLOAD_MASS__KG_) FROM "SPACEXDATASET" WHERE CUSTOMER = 'NASA (CRS)';
```

We used the **SUM** function to retrieve the following result:

Output:

```
Out[10]:   1
       45596
```

Average Payload Mass by F9 v1.1

- The query below requests the average payload mass carried by booster version F9 v.1.1 :

```
In [11]: %sql select AVG(PAYLOAD_MASS__KG_) FROM "SPACEXDATASET" WHERE BOOSTER_VERSION LIKE 'F9 v1.1%';
```

We used the **AVG** function to retrieve the following result.

Output:

```
Out[11]:    1
         2534
```

First Successful Ground Landing Date

- The query below requests the date when the first successful landing outcome in ground pad was achieved:

```
%sql select MIN(DATE) FROM "SPACEXDATASET" WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

We used the **MIN** function to retrieve the first date.

Output

Out[12]:

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The query below requests the booster version where the landing was successful in drone ship, and payload mass is between 4000 kg and 6000 kg:

```
In [37]: %sql select BOOSTER_VERSION FROM "SPACEXDATASET" WHERE LANDING__OUTCOME = 'Success (drone ship)' \
         and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000;
```

- We used the **WHERE** and **AND** clauses to filter the dataset.

Output

Out[37]:

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- The query below requests the number of successful mission outcomes:

```
In [38]: #number of successfull mission outcomes
         %sql SELECT COUNT(MISSION_OUTCOME) FROM "SPACEXDATASET" WHERE MISSION_OUTCOME = 'Success';
```

- We used the **COUNT** function to return the number of successful outcomes.

Output

Out[38]:

1
99

Total Number of Successful and Failure Mission Outcomes

- The query below requests the number of failure mission outcomes:

```
In [39]: #number of failure mission outcomes
         %sql SELECT COUNT(MISSION_OUTCOME) FROM "SPACEXDATASET" WHERE MISSION_OUTCOME <> 'Success';
```

We used the **COUNT** function to return the number of failure outcomes.

Output

Out[39]:

1
2

Boosters Carried Maximum Payload

- The query below requests a list of the booster versions which have carried the maximum payload mass:

```
In [23]: %sql SELECT BOOSTER_VERSION as boosterversion from "SPACEXDATASET" WHERE \
         PAYLOAD_MASS__KG_=(SELECT max(PAYLOAD_MASS__KG_) from "SPACEXDATASET");
```

We used the **MAX** function in a subquery.

Output

Out[23]:

boosterversion
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

- The query below requests a list of failed landing outcomes in drone ship, their booster versions, and launch site names for the year 2015:

```
In [24]: %sql SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, DATE FROM "SPACEXDATASET" \
         WHERE LANDING__OUTCOME = 'Failure (drone ship)' and YEAR(DATE) = 2015;
```

We used the **YEAR** function to return the respective year of the dates.

Output:

Out[24]:

landing__outcome	booster_version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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

- The query below returns all landing outcomes and their occurrences, between 06/04/2010 and 03/20/2017, in descending order.

```
In [25]: %sql SELECT LANDING__OUTCOME, COUNT(*) AS number_of_launches FROM "SPACEXDATASET" \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME\
ORDER BY number_of_launches DESC;
```

We used the **COUNT, BETWEEN, GROUP BY, ORDER BY** functions to retrieve the following result:

Output

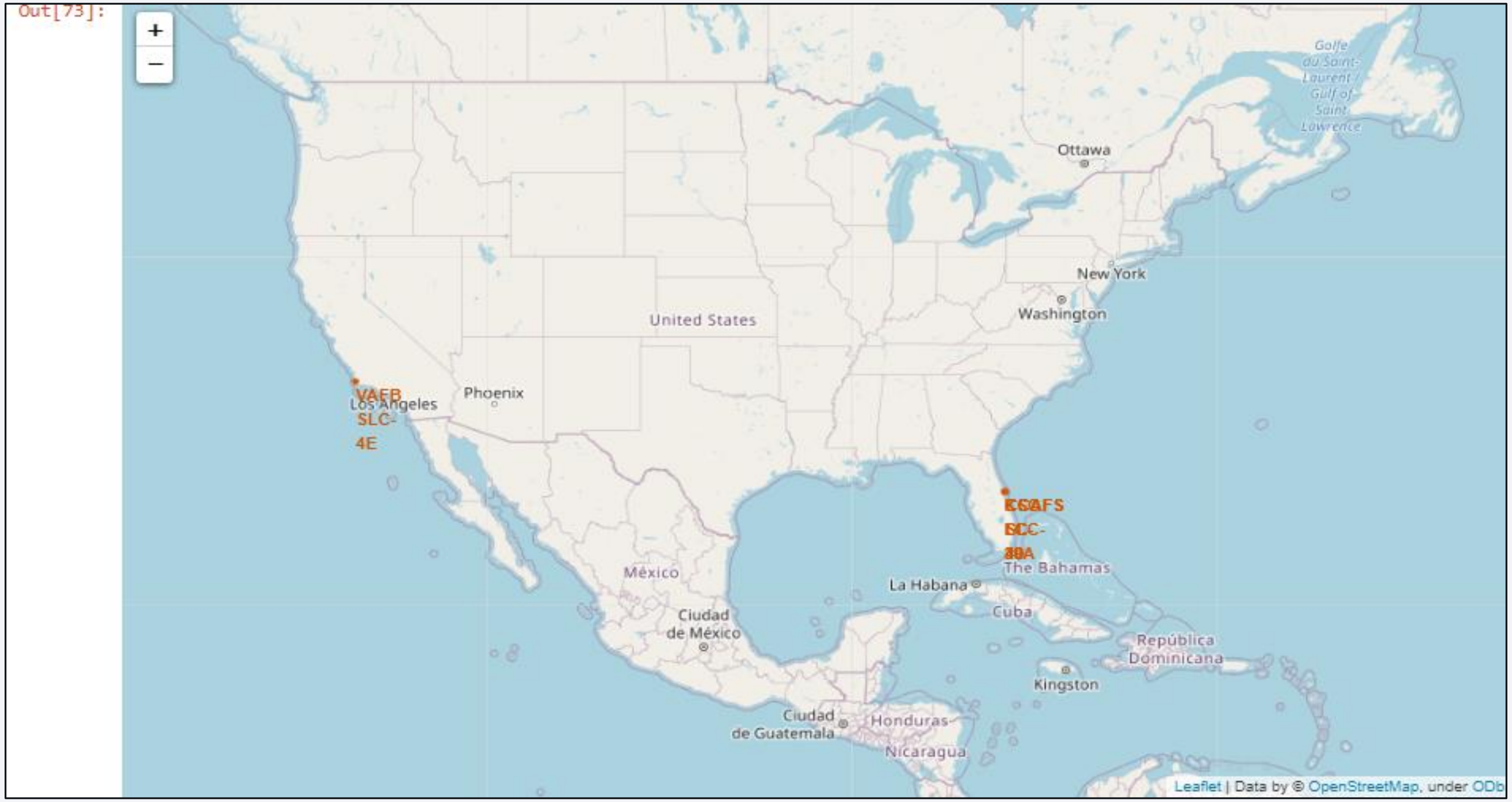
Out[25]:

landing__outcome	number_of_launches
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

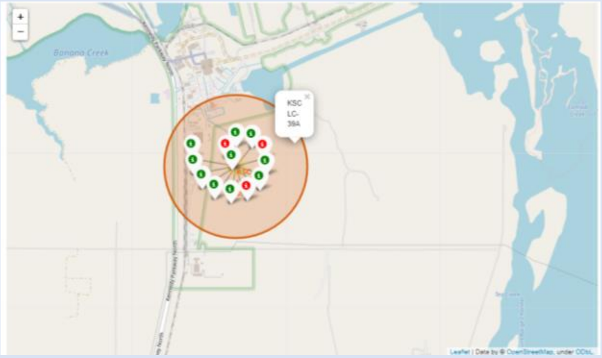
All Launch Sites



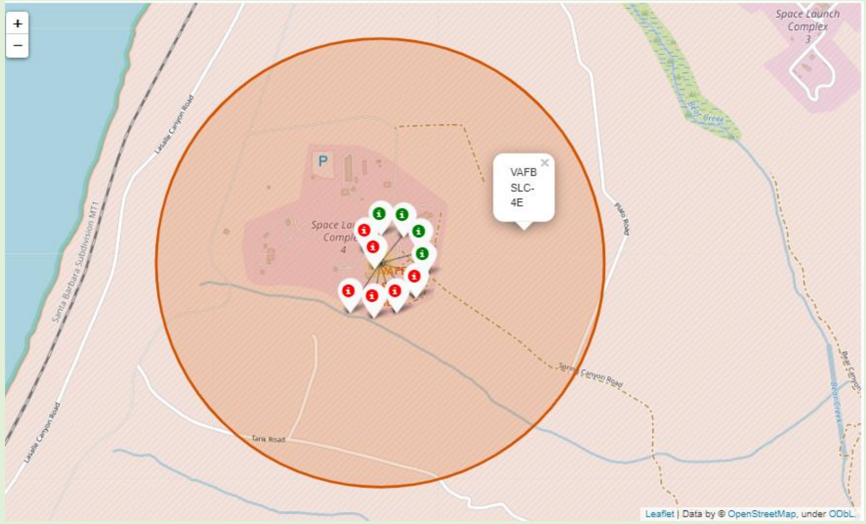
- Note that all launch sites are in very proximity to the coast of the US;
- 3 Launch Sites are in Florida, while 1 Launch Site is in California.

Launch Outcomes per Site

Florida Launch Sites



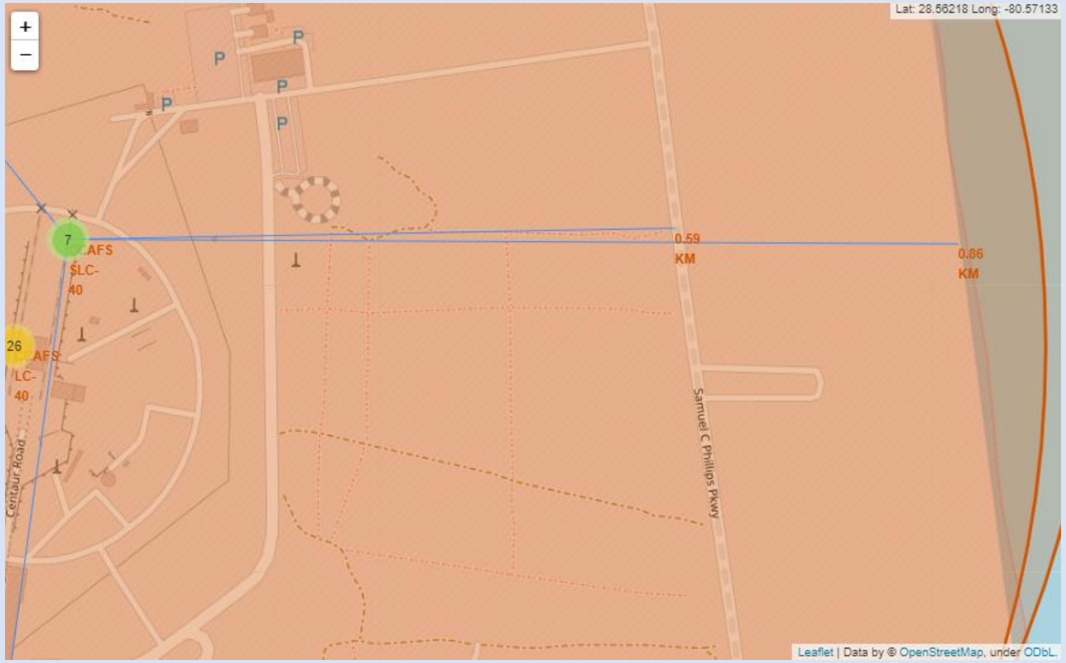
California Launch Site



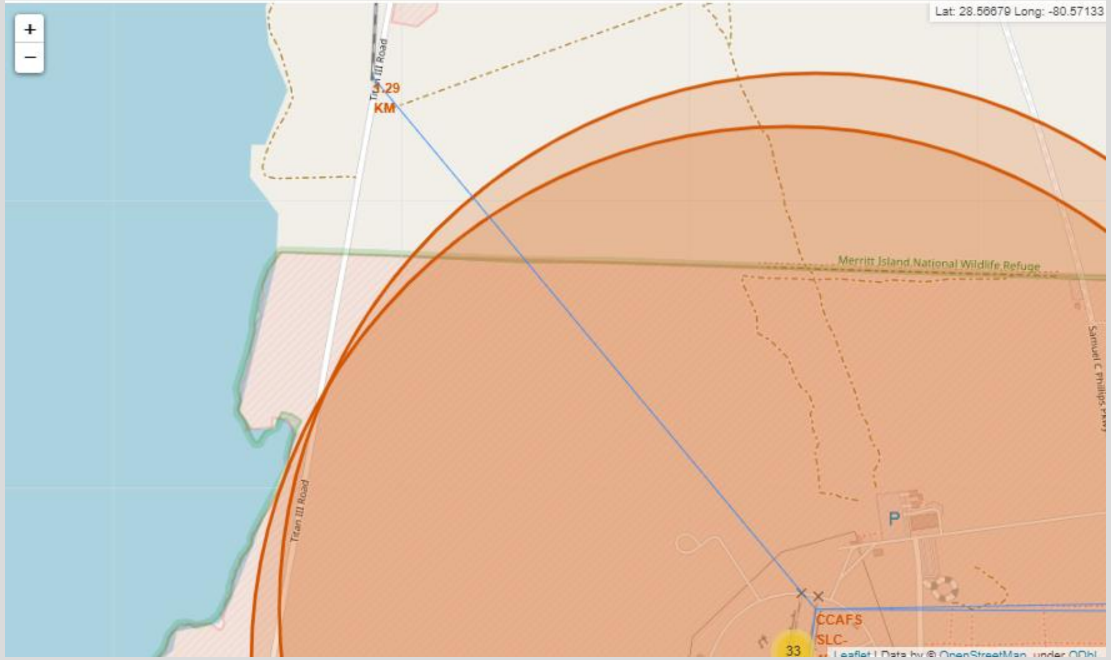
Green Marker shows successful Launches and Red Marker shows Failures

Logistics and Safety – CCAFS SLC-40

Distance to highway and coastline



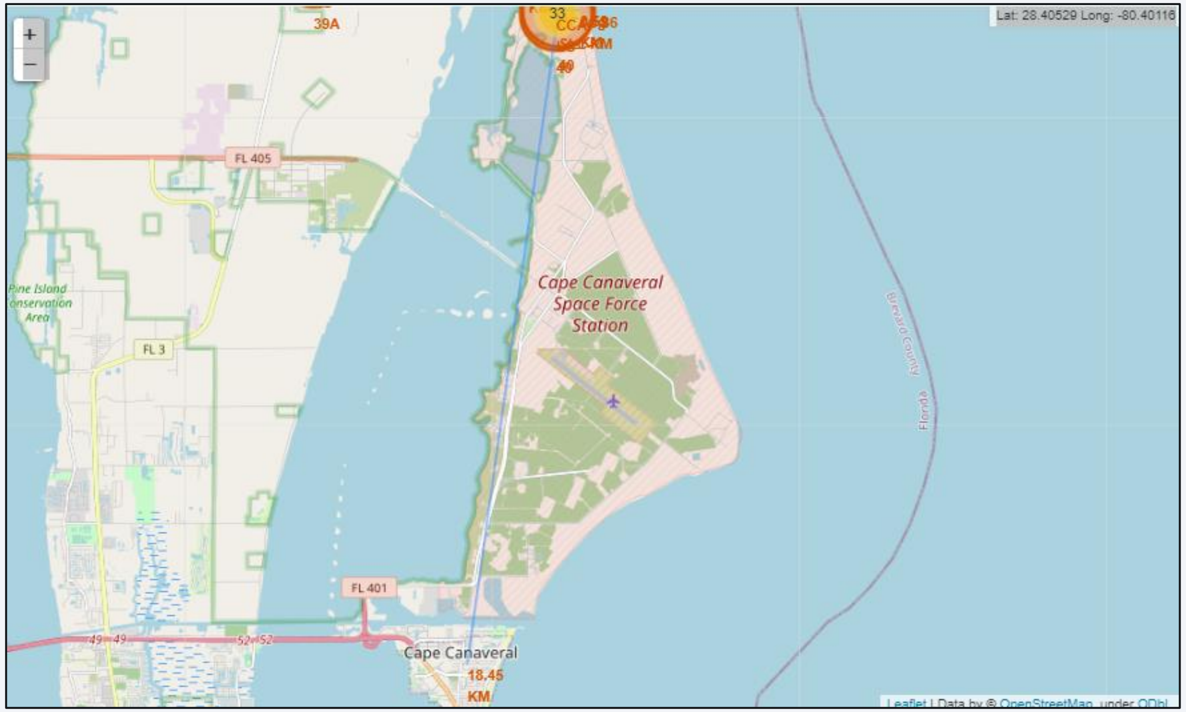
Distance to railway



- Note that the launch site is located near highways, railways, and the coastline, as shown in the maps above.

Logistics and Safety – CCAFS SLC-40

Distance to the nearest city

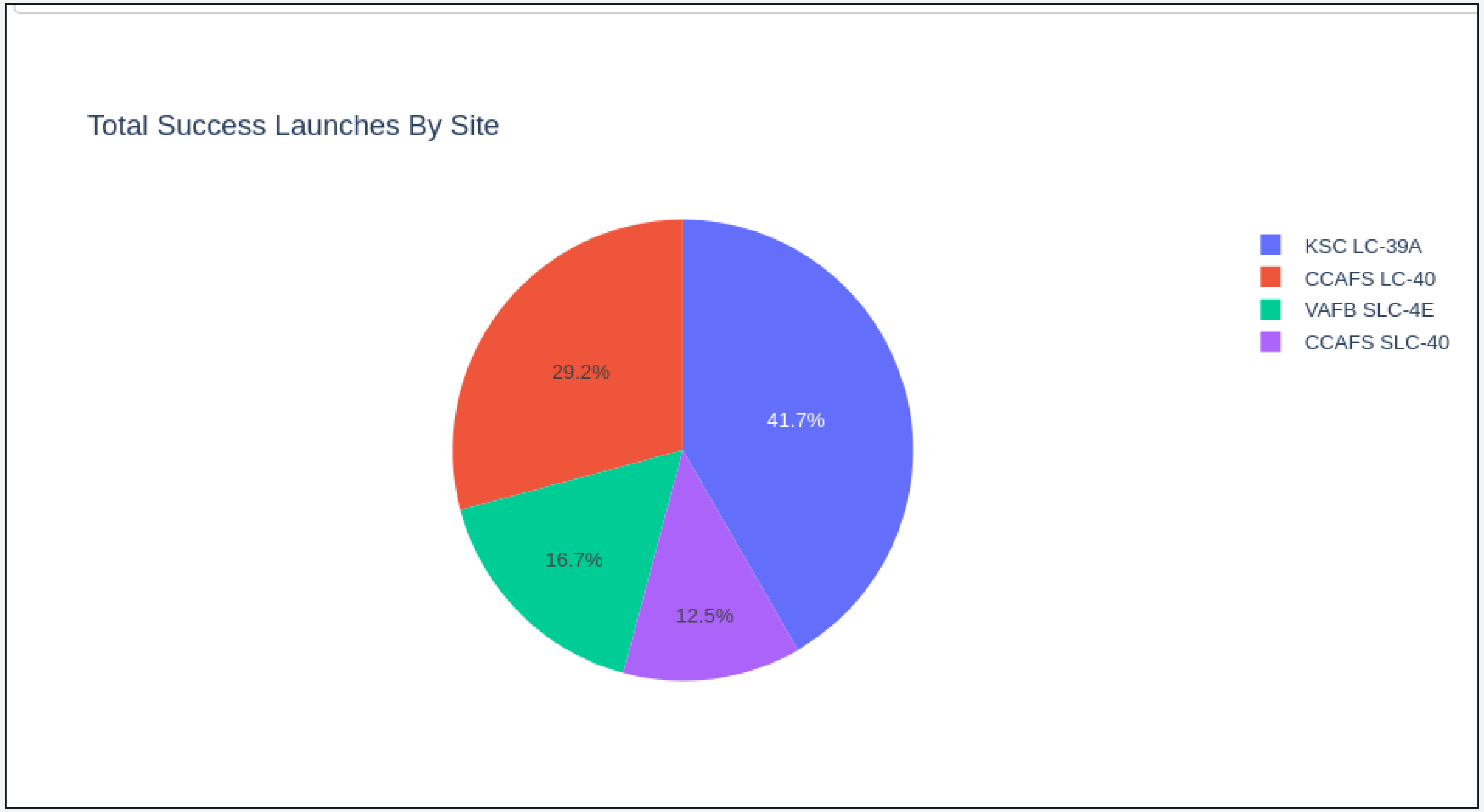


- Note that the launch site keeps a certain distance away from cities. The nearest city is Cape Canaveral/Florida (18.45 km).

Section 4

Build a Dashboard with Plotly Dash

Total Success Launches by Site

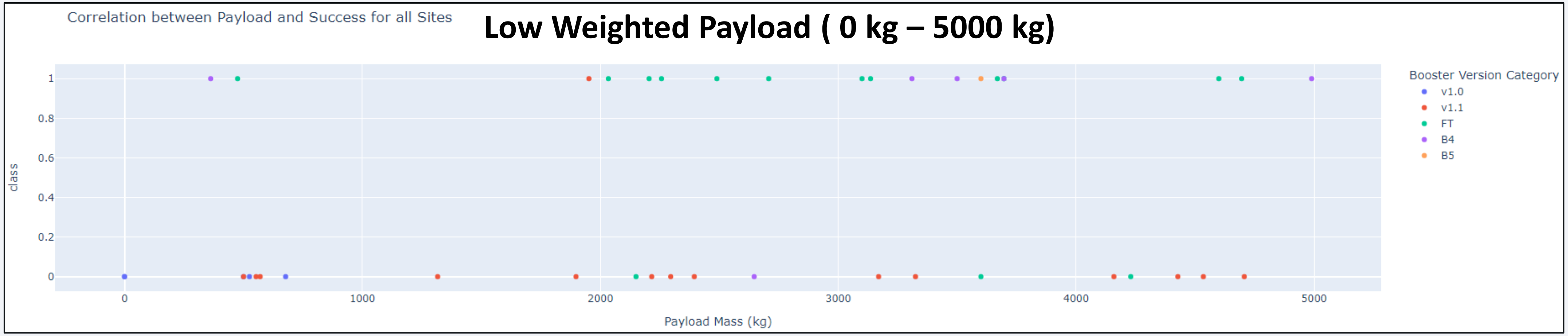


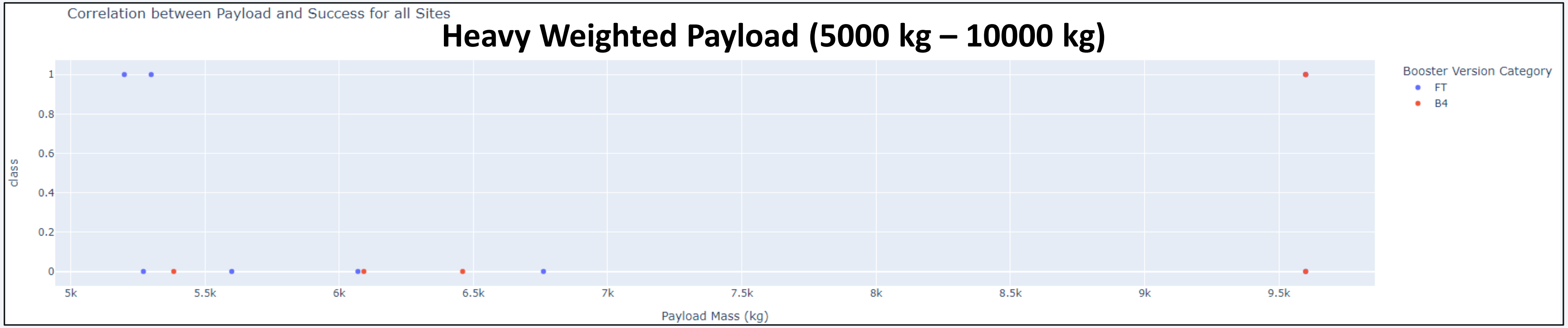
- We see that KSC LC-39A has the best launch success rate among all sites.

Total Success Launches for Site KSC LC-39A

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

Payload mass vs Launch Outcome (all launch sites)



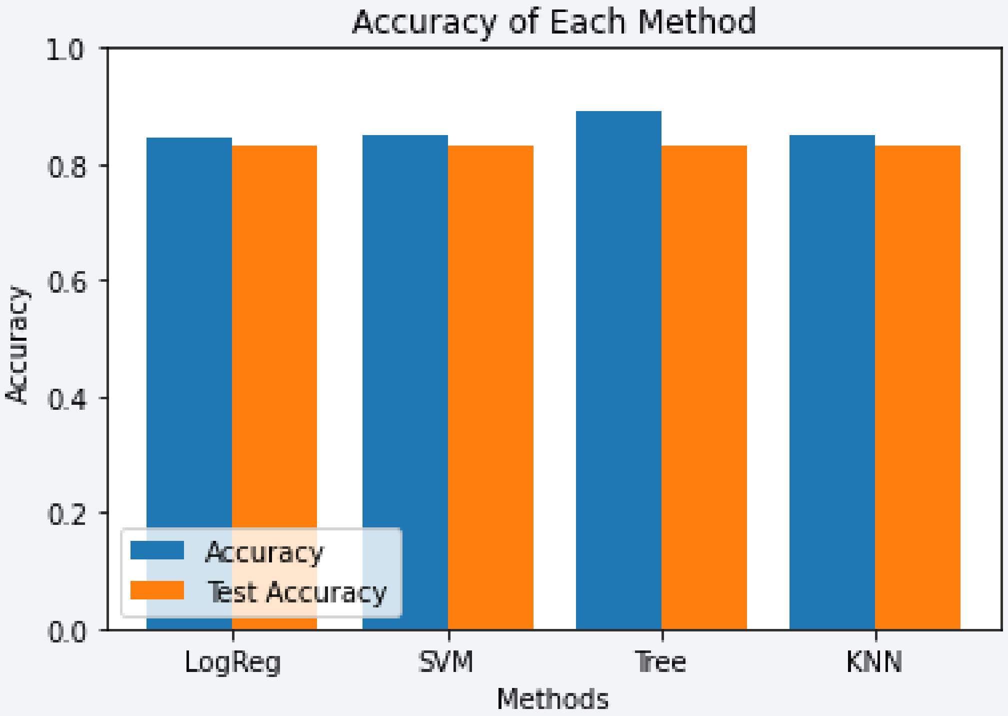


- Low Weighted payloads have a better success rate than heavily weighted payloads;
- Booster Version Category FT seems to be more successful than others.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



Model	Accuracy	TestAccuracy
LogReg	0.846429	0.833333
SVM	0.848214	0.833333
Tree	0.889286	0.833333
KNN	0.848214	0.833333

- Four classification models were tested, and their accuracies are plotted beside:

➢ Logistics Regression method (**LogReg**)
➢ Support Vector Machine method (**SVM**)
➢ Decision Tree Method (**Tree**)
➢ K nearest neighbors method (**KNN**)

- The Decision Tree classifier is the Machine Learning model with the highest classification accuracy (approximately 89 % 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., an unsuccessful landing is marked as a successful landing by the classifier.

Conclusions

- For each site, the success rate increases with the number of launches performed;
- The payload weight can have a significant influence on the success rate of the launches in certain orbits;
- Heavier payloads improve the success rate in LEO orbit;
- Low Weighted payloads have a better success rate than heavily weighted payloads;
- Booster Version Category FT seems to be more successful than others.;
- Success rate started increasing in 2013 and kept until 2020;
- Orbits ES-L1, GEO, HEO, SSO, 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!