Winning Space Race with Data Science

Nguyen Phu Quoc

September 2021

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

• Project background and context

We predicted if the Falcon 9 first stage will land successfully. SpaceX 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 SpaceX 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 SpaceX for a rocket launch.

• Common problems that needed solving

- What influences if the rocket will land successfully?
- The effect each relationship with certain rocket variables will impact in determining the success rate of a successful landing.
- What conditions does SpaceX have to achieve to get the best results and ensure the best rocket success landing rate.

Methodology

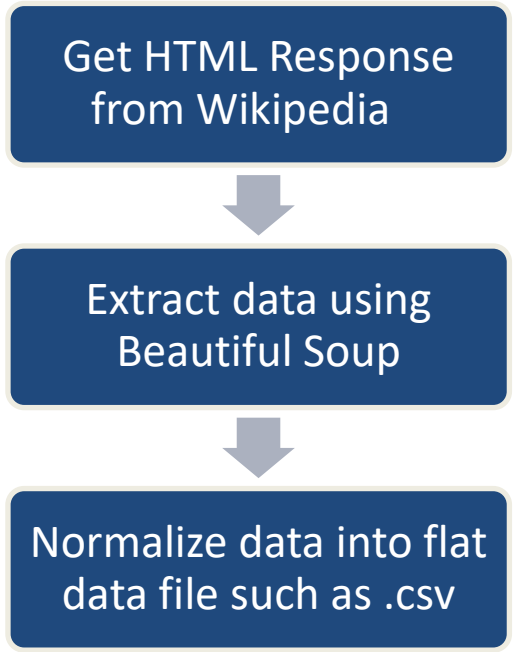
- Data collection methodology:
  - SpaceX Rest API
  - (Web Scrapping) from Wikipedia
- Performed data wrangling (Transforming data for Machine Learning)
  - One Hot Encoding data fields for Machine Learning and dropping irrelevant columns
- Performed exploratory data analysis (EDA) using visualization and SQL

  - Plotting : Scatter Graphs, Bar Graphs to show relationships between variables to show patterns of data.
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models
  - Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
  - Find the method performs best using test data

Section 1: Methodology

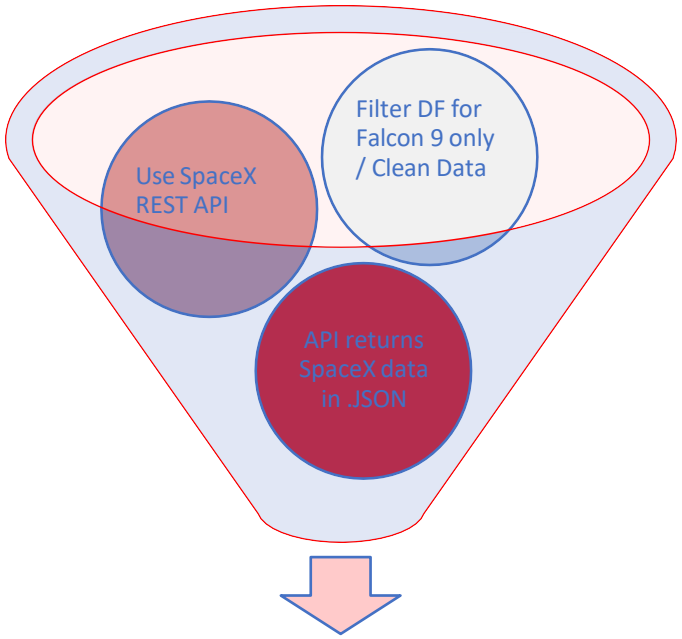
Methodology

- The following datasets was collected by
- We worked with SpaceX launch data that is gathered from the SpaceX REST API.
- This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- Our goal is to use this data to predict whether SpaceX will attempt to land a rocket or not.
- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/.
- Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using BeautifulSoup.



Data collection – SpaceX API

GitHub



Normalize data into flat data file such as .csv

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)
```

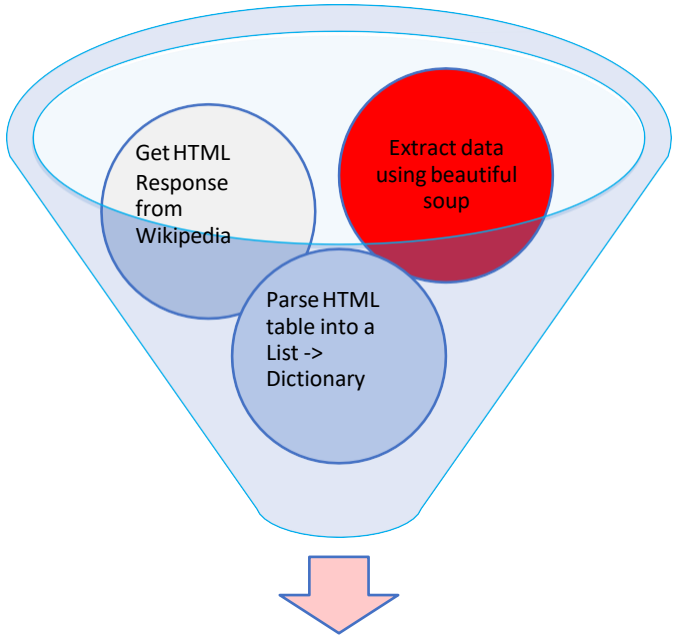
5. Filter dataframe and export to flat file (.csv)

```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

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

Data collection – Scrapping

GitHub



Normalize data into flat data file such as .csv

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 irrelvant 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_scraped.csv', index=False)
```

Data Wrangling

GitHub

Introduction

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False
ASDS means the mission outcome was unsuccessfully landed on a drone ship.
We mainly convert those outcomes into Training Labels with 1 means the booster successfully
landed 0 means it was unsuccessful.

Perform EDA on dataset

Calculate the number of launches at each site

Calculate the number and occurrence of each orbit

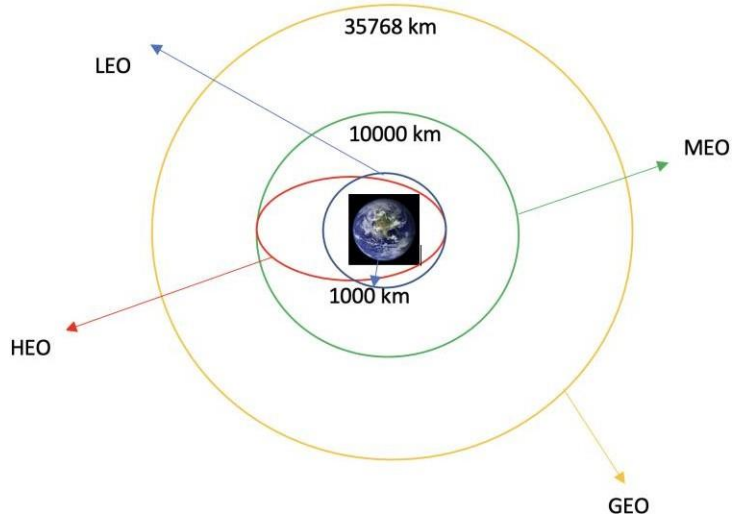
Calculate the number and occurrence of mission outcome per orbit type

Export dataset as .CSV

Create a landing outcome label from Outcome column

Work out success rate for every landing in dataset

Each launch aims to an dedicated orbit, and here are some common orbit types:



SPACEX

EDA with Data Visualization

GitHub

Scatter Graphs being drawn:

Flight Number VS. Payload Mass Flight Number VS. Launch Site Payload VS.

Launch Site

Orbit VS. Flight Number

Payload VS. Orbit Type

Orbit VS. Payload Mass

Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation . Scatter plots usually consist of a large body of data.

Bar Graph being drawn:

Mean VS. Orbit

A bar diagram makes it easy to compare sets of data between different groups at a glance. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes. Bar charts can also show big changes in data over time.

Line Graph being drawn:

Success Rate VS. Year

Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded

EDA with SQL

GitHub

Performed SQL queries to gather information about the dataset.

For example of some questions we were asked about the data we needed information about. Whichwe are using SQL queries to get the answers in the dataset:

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string'KSC'
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date where the successful landing outcome in drone ship wasachieved.
- Listing the names of the boosters which have success in ground pad and have payload mass greater than4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster_versions which have carried the maximum payloadmass.
- Listing the records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
- Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

Building an interactive map with Folium

To visualize the Launch Data into an interactive map. We took the Latitude and Longitude Coordinates at each launch site and added a Circle Marker around each launch site with a label of the name of the launch site.

We assigned the dataframe launch_outcomes(failures, successes) to *classes 0 and 1* with Green and Red markers on the map in a MarkerCluster()

Using Haversine's formula we calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks

Example of some trends in which the Launch Site is situated in.

- 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

Used Python Anywhere to host the website live 24/7 so your can play around with the data and view the data

- The dashboard is built with Flask and Dash web framework.

  Graphs

  - Pie Chart showing the total launches by a certain site/all sites

    - display relative proportions of multiple classes of data.

    - size of the circle can be made proportional to the total quantity it represents.

Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions

- It shows the relationship between two variables.
- It is the best method to show you a non-linear pattern.
- The range of data flow, i.e. maximum and minimum value, can be determined.
- Observation and reading are straightforward.

Predictive analysis (Classification)

GitHub

BUILDING MODEL

- Load our dataset into NumPy and Pandas
- Transform Data
- Split our data into training and test data sets
- Check how many test samples we have
- Decide which type of machine learning algorithms we want to use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCV objects and train our dataset.

EVALUATING MODEL

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

IMPROVING MODEL

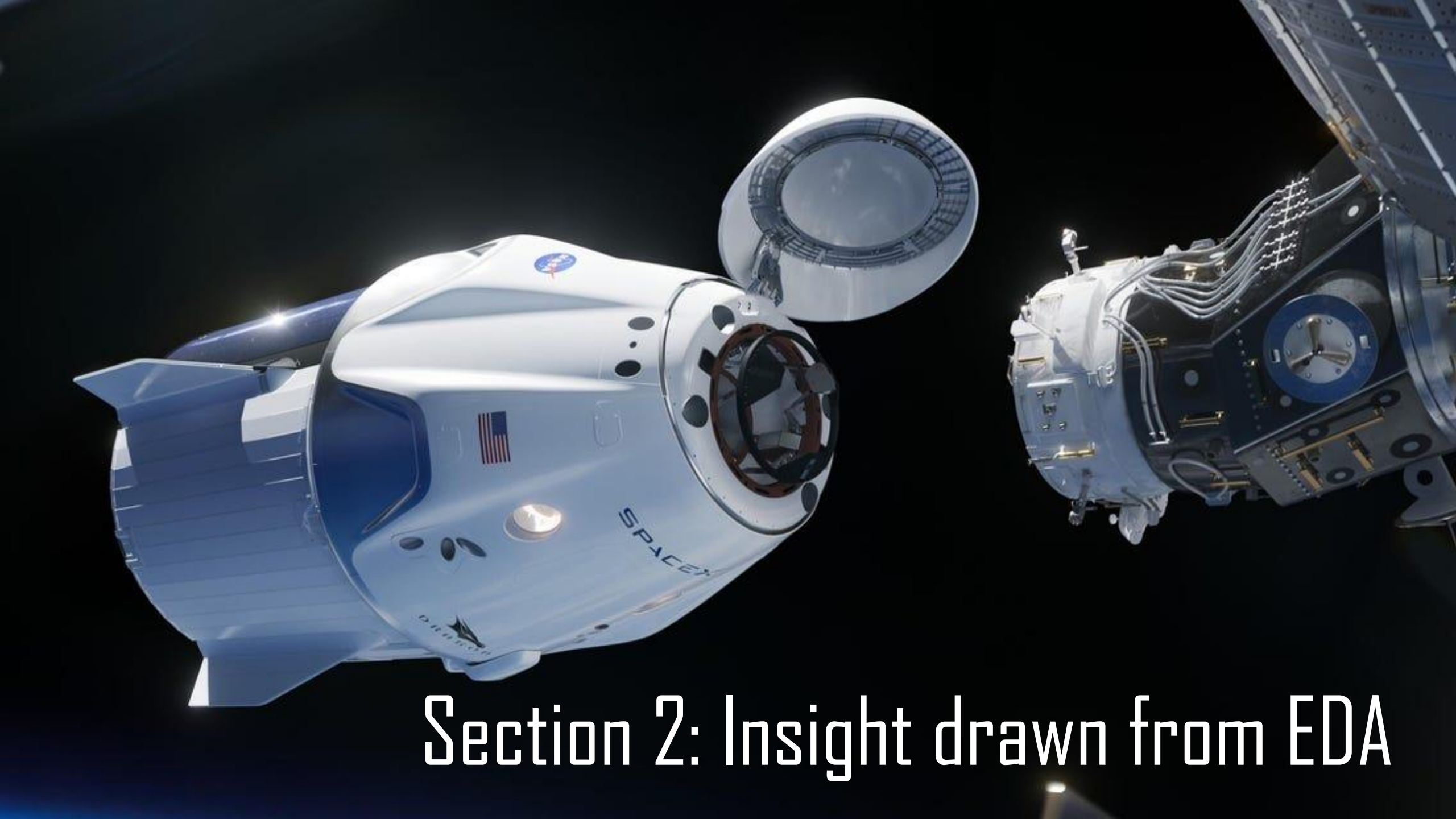
- Feature Engineering
- Algorithm Tuning

FINDING THE BEST PERFORMING CLASSIFICATION MODEL

- The model with the best accuracy score wins the best performing model
- In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook.

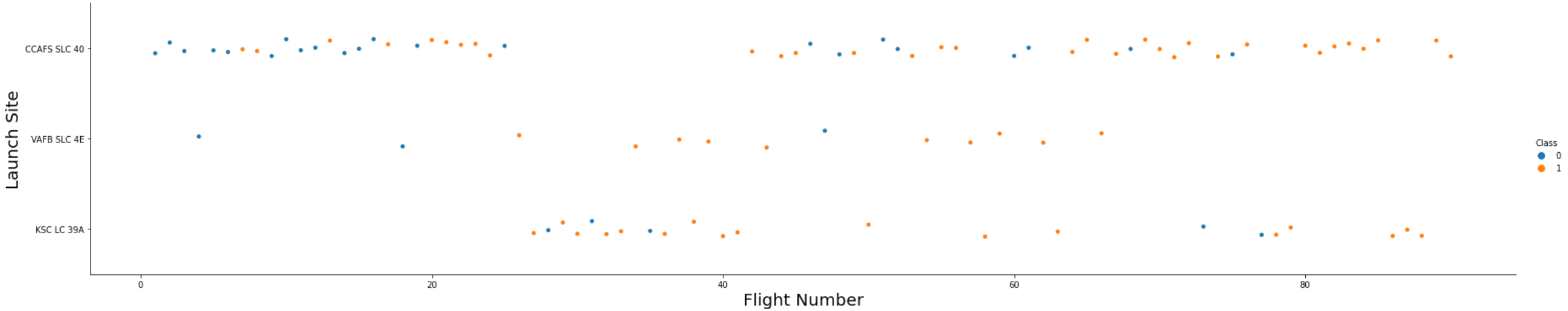
Results

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



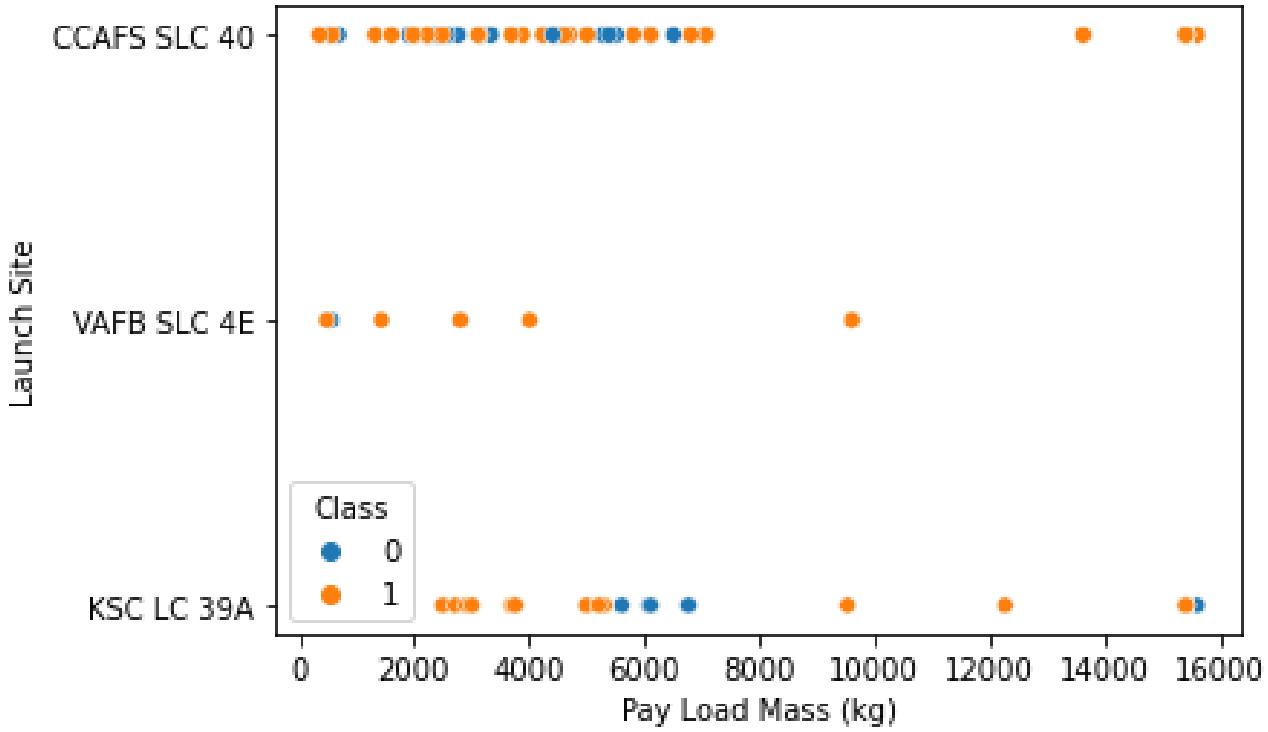
Section 2: Insight drawn from EDA

Flight Number vs Launch Site



The more amount of flights at a launch site the greater the success rate at a launch site.

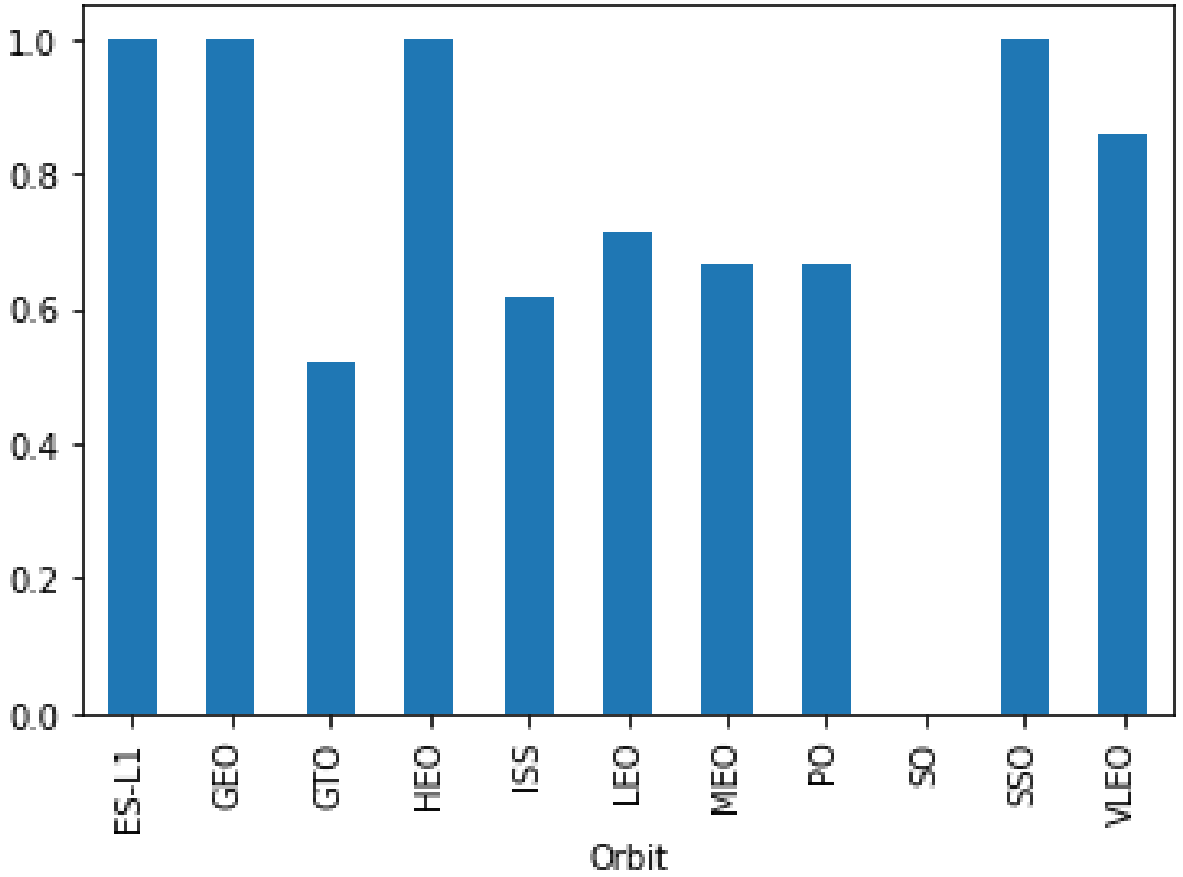
Payload vs. Launch Site



The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket. There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependant on Pay Load Mass for a success launch.

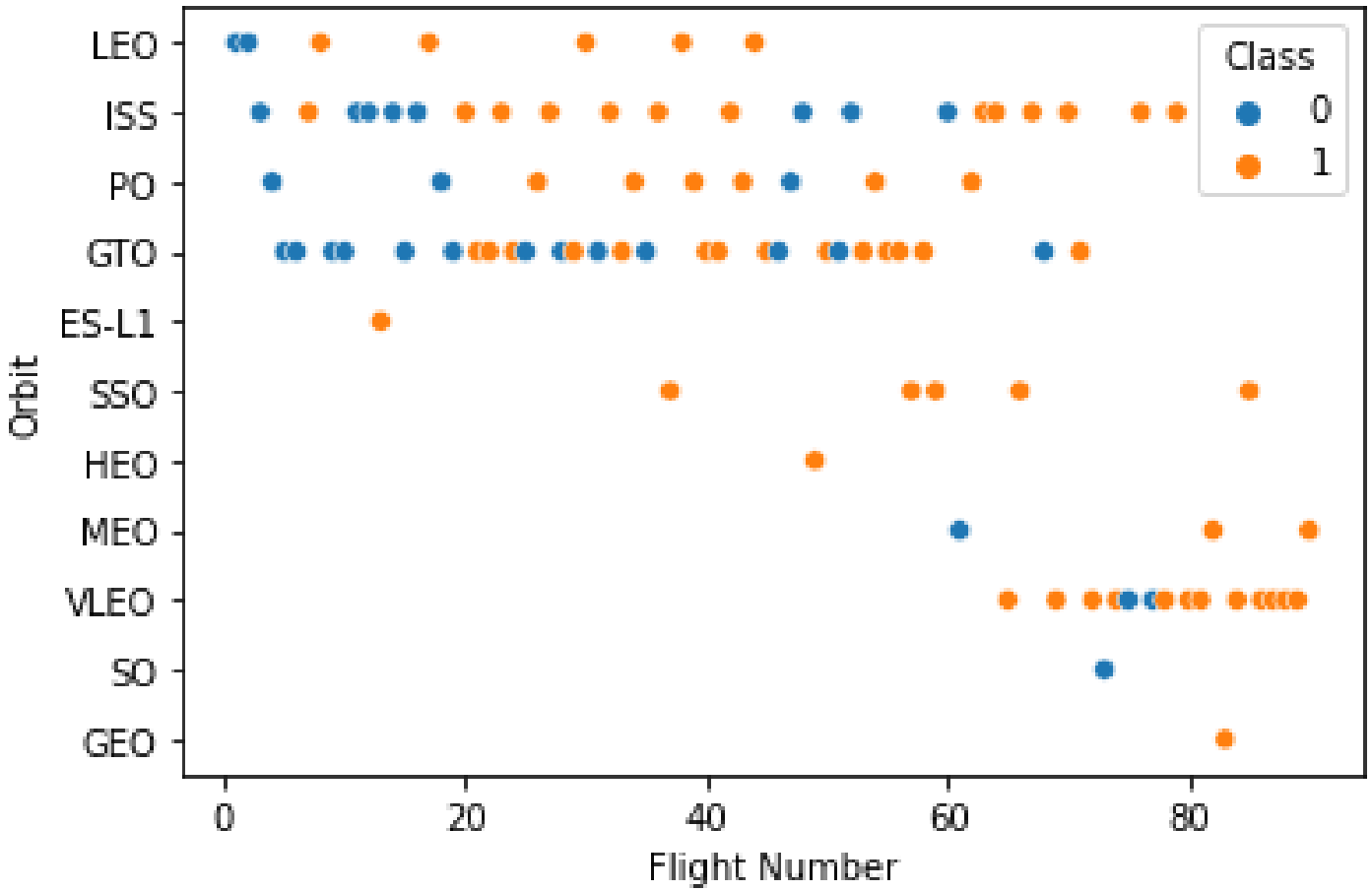
Success rate vs. Orbit type

Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate



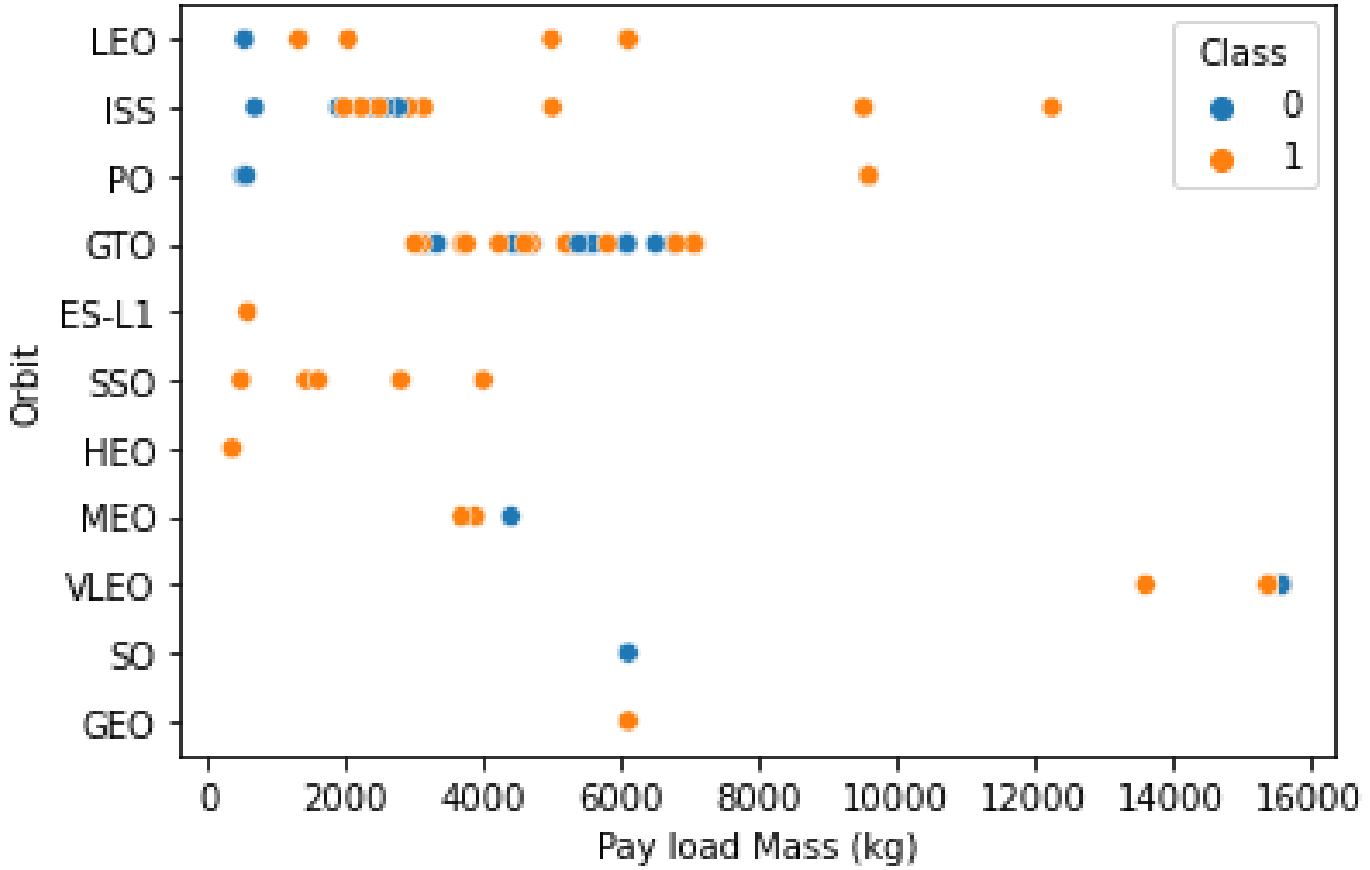
Flight Number vs. Orbit type

You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



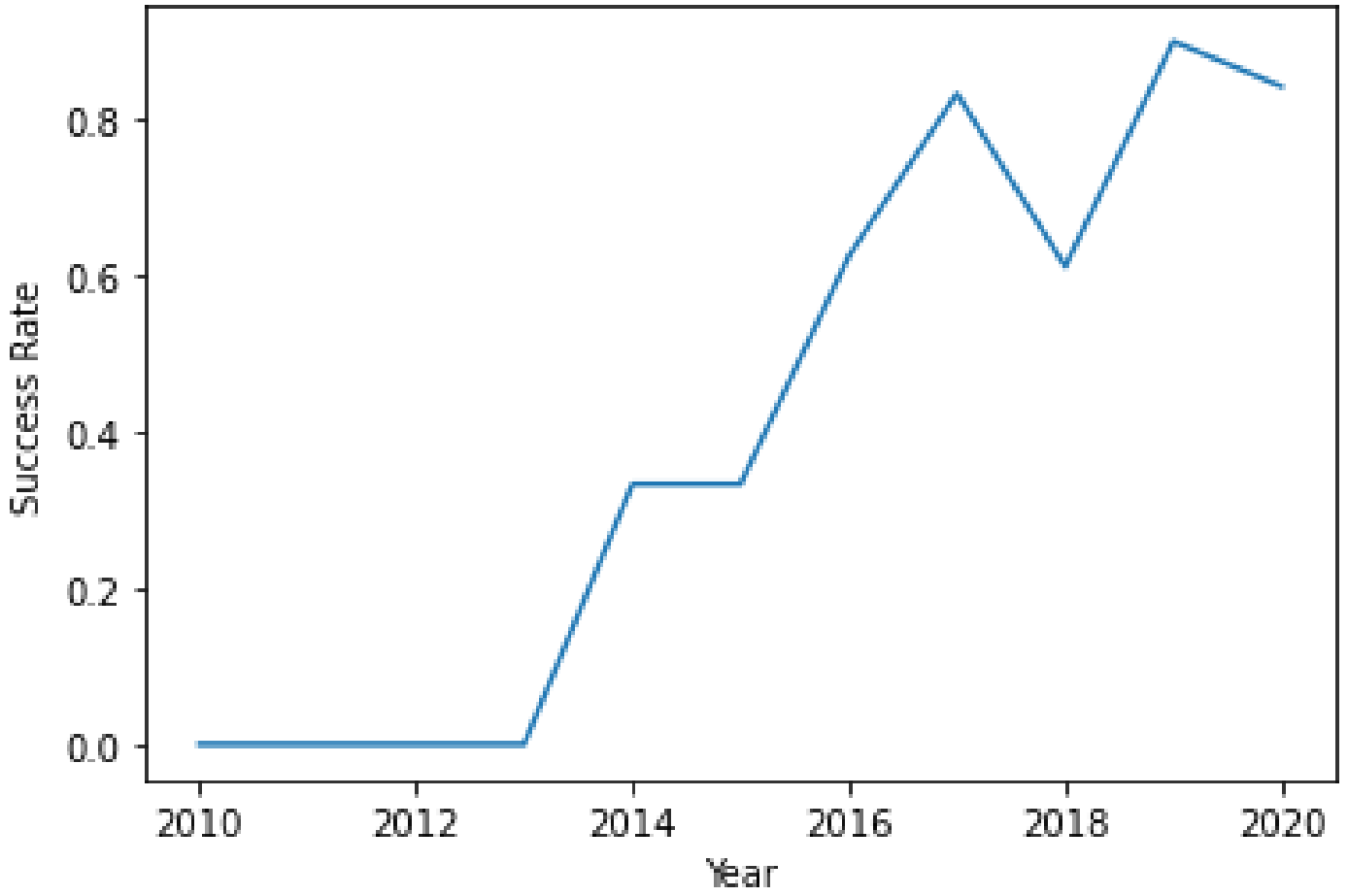
Payload vs. Orbit type

You should observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.



Launch success yearly trend

As can be observed that the success rate since 2013 kept increasing till 2020



All Launch Site Names

Within a total of 90 launch sites, 55 launches belong to site CCAFS SLC 40, 22 and 13 belongs to KSC LC 39A and VAFB SLC 4E, respectively.

```
In [5]: # Apply value_counts() on column LaunchSite
        df['LaunchSite'].value_counts()

Out[5]: CCAFS SLC 40    55
        KSC LC 39A      22
        VAFB SLC 4E     13
        Name: LaunchSite, dtype: int64
```

Launch Site Names Begin with 'CCA'

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

In [5]:
```
%sql select * from SPACEXTBL where launch_site like 'CCA%' limit 5
```

```
 * ibm_db_sa://xwk47039:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[5]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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)
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Launch Site Name begins with 'CCA' is CCAFS LSC 40

Total Payload Mass

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

In [6]:
```
%sql select sum(payload_mass__kg_) total_payload_mass from SPACEXTBL where customer like '%NASA%'
```

```
 * ibm_db_sa://xwk47039:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[6]:

total_payload_mass
37249

Total payload mass carried by boosters launched by NASA (CRS) is 37,249

Average Payload Mass by F9 v1.1

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

In [28]:
```
%sql select avg(payload_mass__kg_) Average_payload_mass from SPACEXTBL where booster_version like 'F9 v1.1%'
```

```
 * ibm_db_sa://xwk47039:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[28]:

average_payload_mass
3226

Average Payload Mass by F9 v1.1 is 3,226

First Successful Ground Landing Date

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

Hint:Use min function

In [26]:
```
%sql select min(DATE) DATE from SPACEXTBL where landing__outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://xwk47039:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[26]:

DATE
2017-01-05

First successful ground landing date is 05 January 2017

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

In [36]:
```
%sql select distinct booster_version, landing__outcome from SPACEXTBL where landing__outcome = 'Success (drone ship)' and payloa
d_mass__kg_ between '4000' and '6000'
```

```
 * ibm_db_sa://xwk47039:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[36]:

booster_version	landing__outcome
F9 FT B1031.2	Success (drone ship)
F9 FT B1022	Success (drone ship)

Successful Drone Ships Landing with Payload between 4000 and 6000 are F9 FT B1031.2 and F9 FT B1022

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [23]:
```
%sql select COUNT(*) total_number from SPACEXTBL where landing__outcome like 'Failure%' or landing__outcome like 'Success%'
```

```
 * ibm_db_sa://xwk47039:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[23]:

total_number
32

Total Number of Successful and Failure Mission Outcomes is 32.

Boosters Carried Maximum Payload

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

In [21]:
```
%sql select distinct booster_version, payload_mass__kg_ from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)
```

```
 * ibm_db_sa://xwk47039:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[21]:

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600

Boosters Carried Maximum Payload are F9 B5 B1048.4, F9 B5 B1049.4, F9 B5 B1049.5, F9 B5 B1058.3, F9 B5 B1060.2

2015 Launch Records

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

In [30]:
```
%sql select landing__outcome, booster_version, launch_site from SPACEXTBL where year(DATE) = '2015'
```

```
 * ibm_db_sa://xwk47039:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[30]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
No attempt	F9 v1.1 B1014	CCAFS LC-40

There are 2 landing outcomes of interests which are Failure (drone ship) and Controlled (ocean)

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 [42]:
```
%sql select landing__outcome, count(landing__outcome) count_of_landing_outcomes from SPACEXTBL  where DATE between '2010-06-04' and '2017-03-20' group by landing__outcome order by count(landing__outcome) desc
```

```
 * ibm_db_sa://xwk47039:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[42]:

landing__outcome	count_of_landing_outcomes
No attempt	7
Failure (drone ship)	2
Success (drone ship)	2
Success (ground pad)	2
Controlled (ocean)	1
Failure (parachute)	1

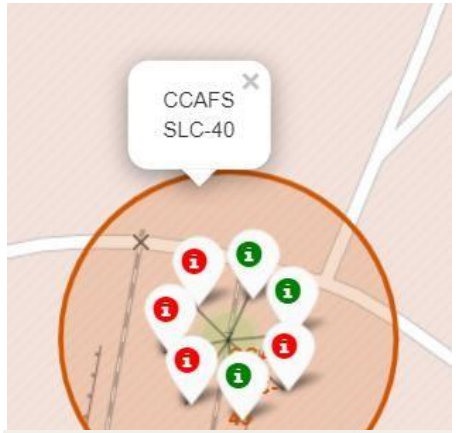
The descending order of landing outcome should be No attempt, Failure (drone ship), Success (drone ship), Success (ground pad), Controlled (ocean), Failure (parachute)

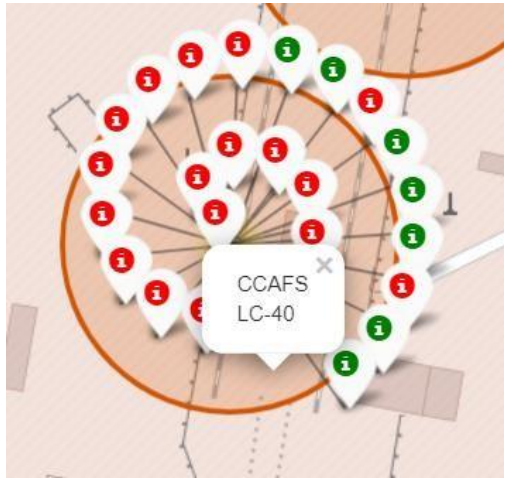
Section 4
Launch Sites Proximities Analysis

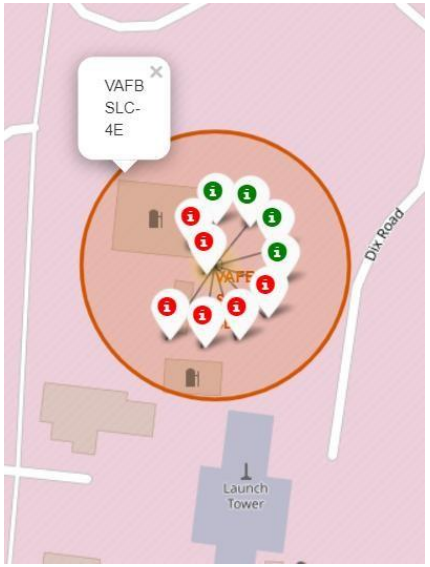
All launch sites global map markers

We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

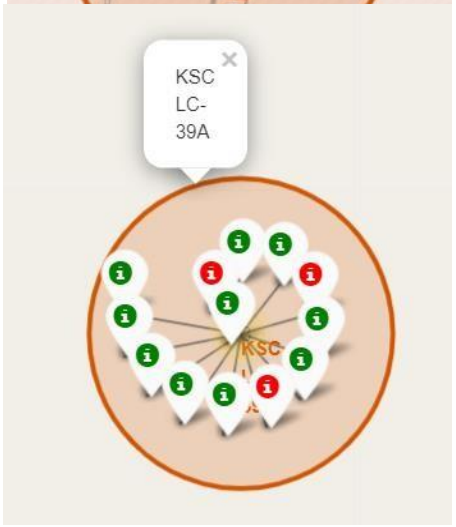
Colour Labelled Markers







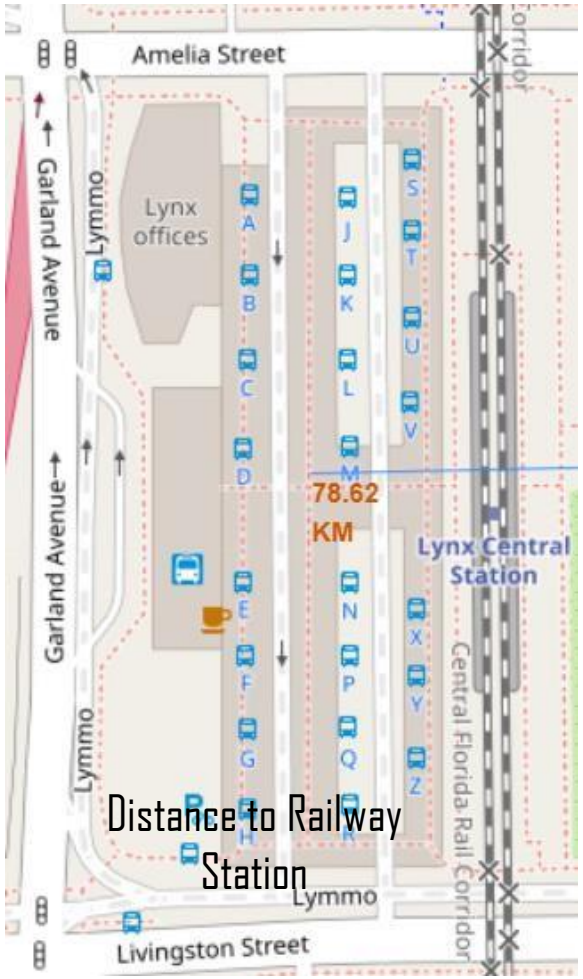
California Launch Site

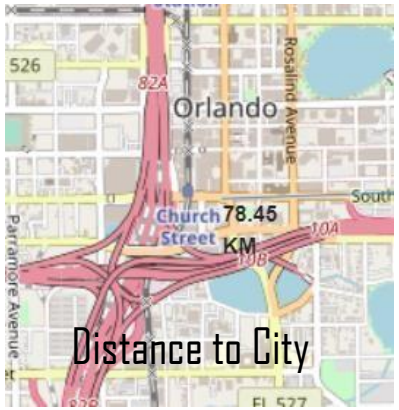


Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

Working out Launch Sites distance to landmarks



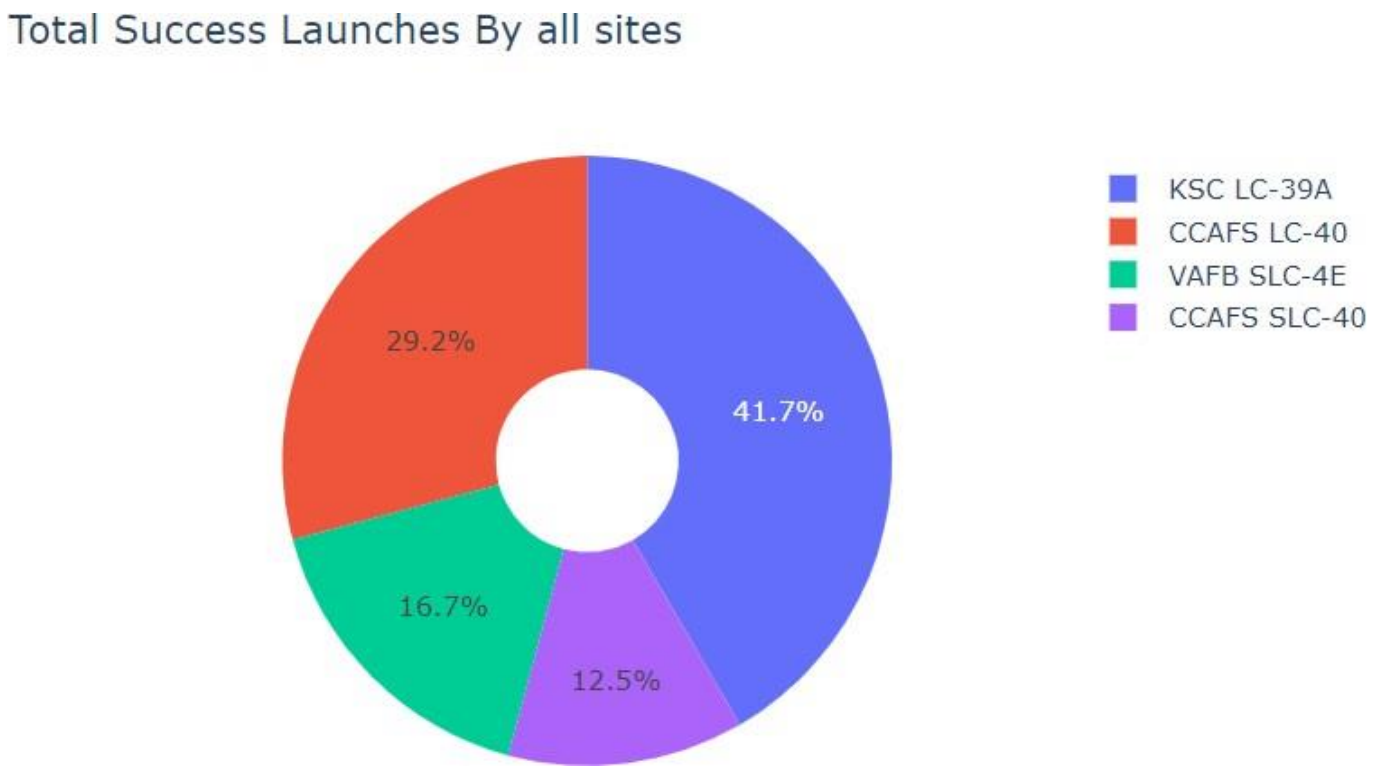


Are launch sites in close proximity to railways? No
Are launch sites in close proximity to highways? No
Are launch sites in close proximity to coastline? Yes
Do launch sites keep certain distance away from cities? Yes

SPACEX

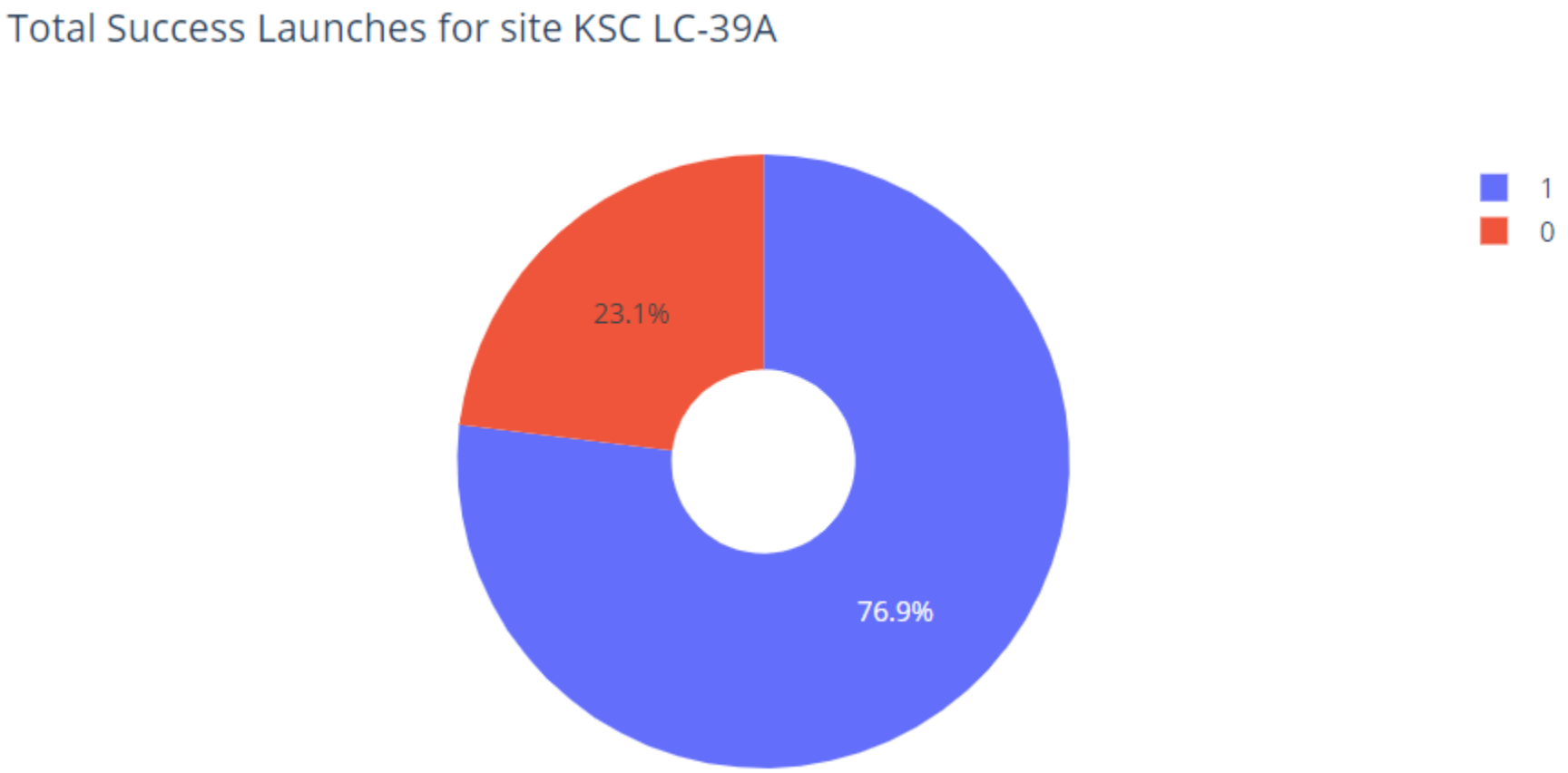
Section 5
Building a Dashboard with Plotly Dash

DASHBOARD – Pie chart showing the success percentage achieved by each launch site



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

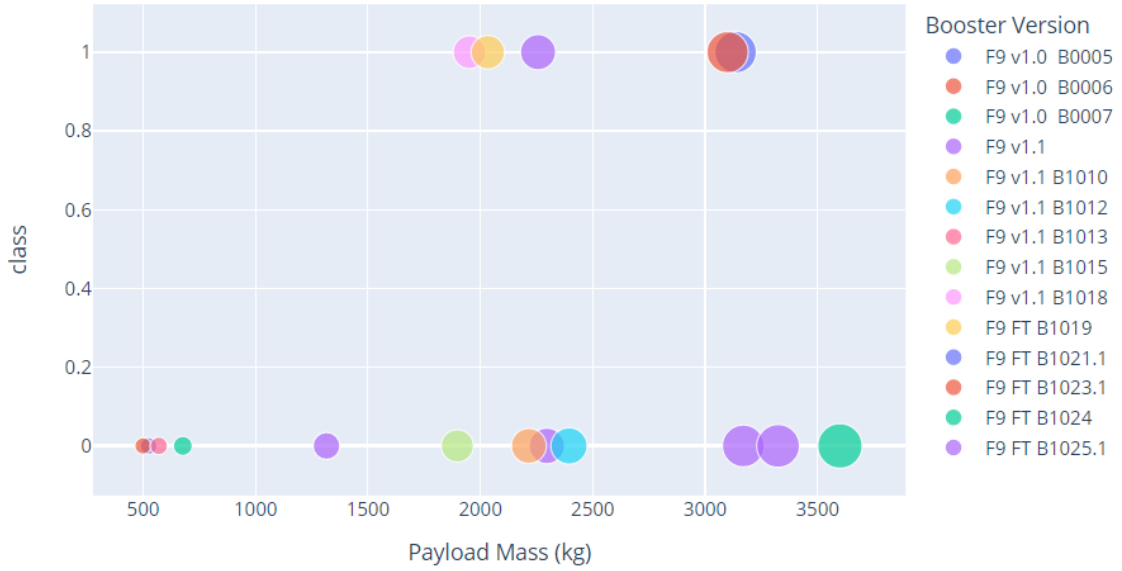
DASHBOARD – Pie chart for the launch site with highest launch success ratio



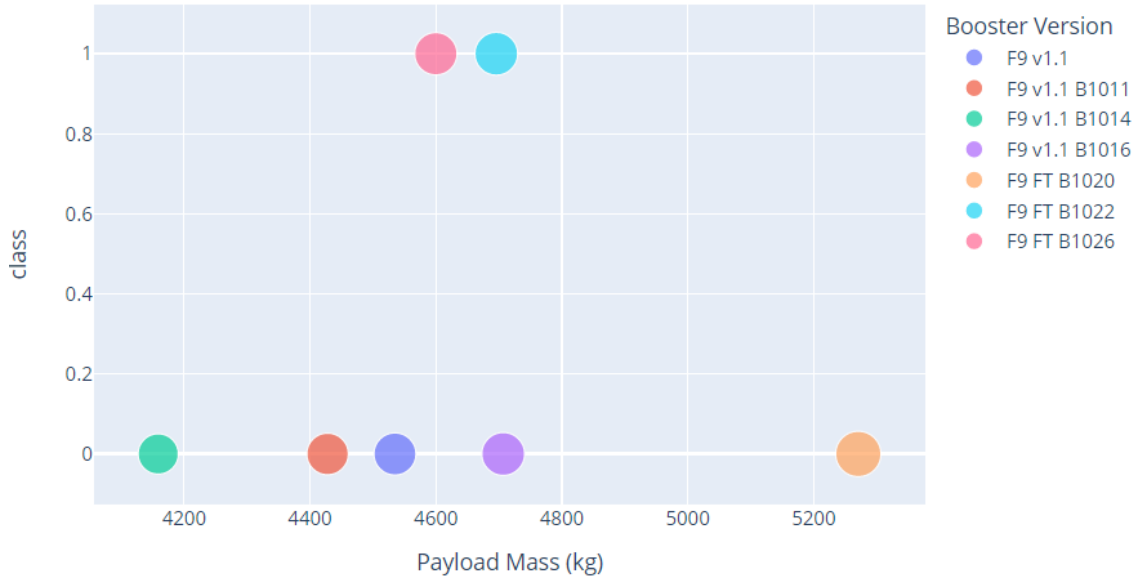
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

DASHBOARD – Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

Low Weighted Payload 0kg – 4000kg



Heavy Weighted Payload 4000kg – 10000kg



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

Section 6
Predictive analysis (Classification)

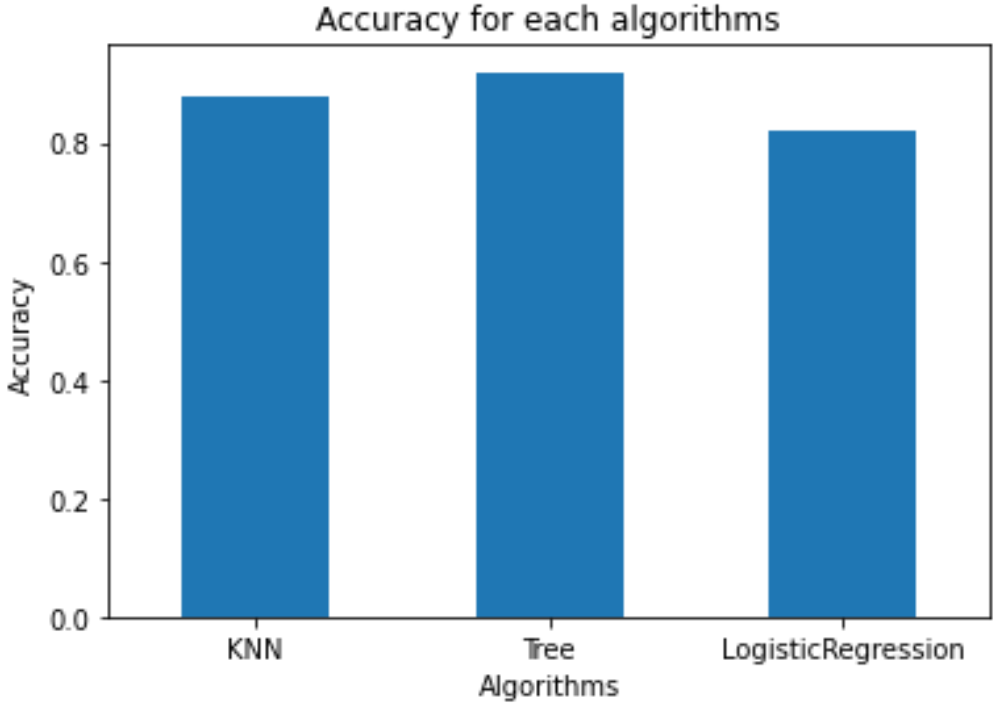
Classification Accuracy using training data

As you can see our accuracy is extremely close but we do have a winner its down to decimal places! using this function

The tree algorithm wins!!

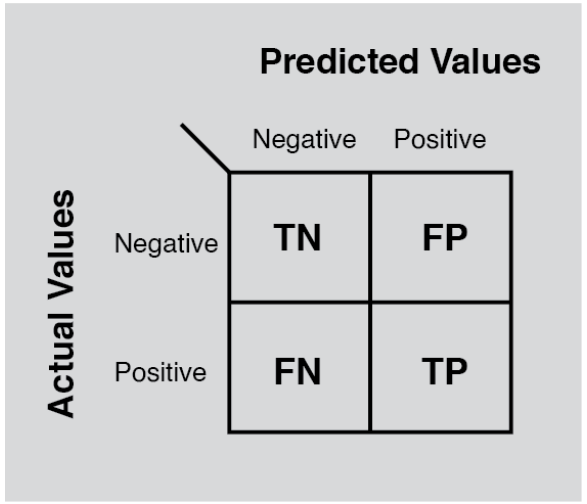
```
Best Algorithm is Tree with a score of 0.9196428571428573
Best Params is : {'criterion': 'entropy', 'max_depth': 4,
'max_features': 'auto', 'min_samples_leaf': 4,
'min_samples_split': 10, 'splitter': 'best'}
```

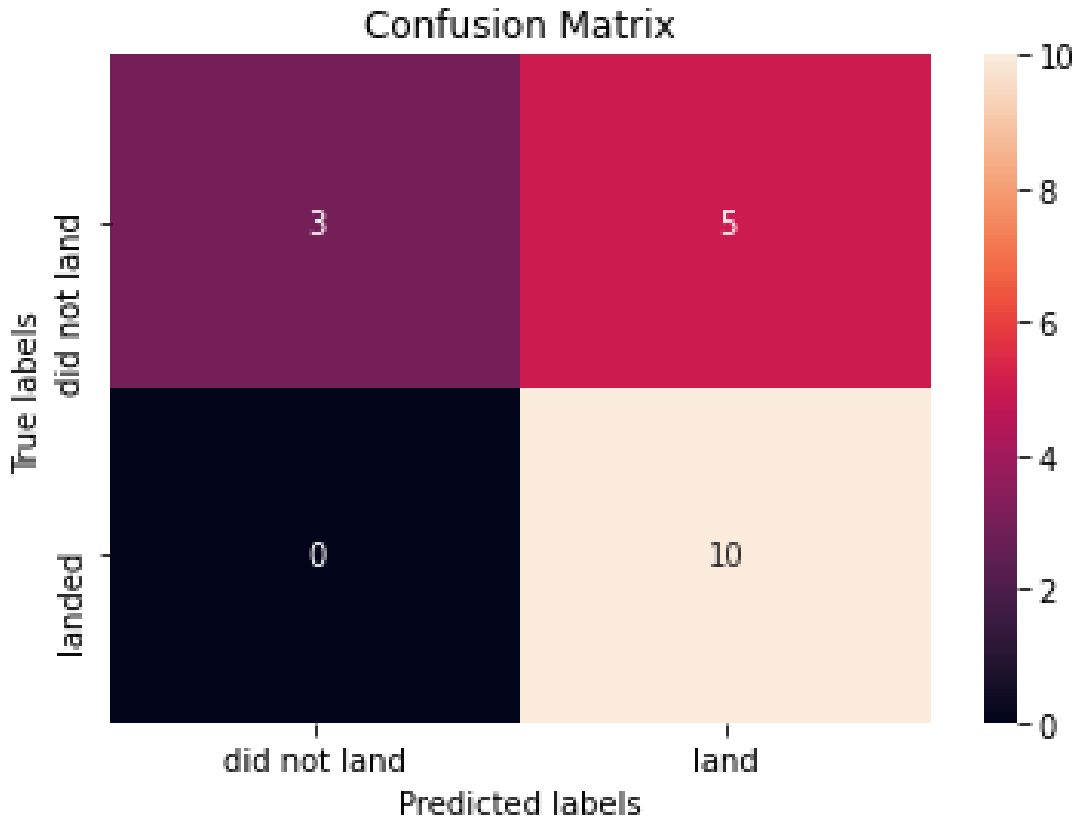
After selecting the best hyperparameters for the decision tree classifier using the validation data, we achieved 83.33% accuracy on the test data.



Confusion Matrix for the Tree

Examining the confusion matrix, it can be seen that Tree can distinguish between the different classes and that the major problem is false positives.





Conclusion

- The Tree Classifier Algorithm is the best for Machine Learning for this dataset
- Low weighted payloads perform better than the heavier payloads
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches
- We can see that KSC LC-39A had the most successful launches from all the sites
- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Appendix

For further details and notebook, please visit my GitHub


SPACEX
Thank you!