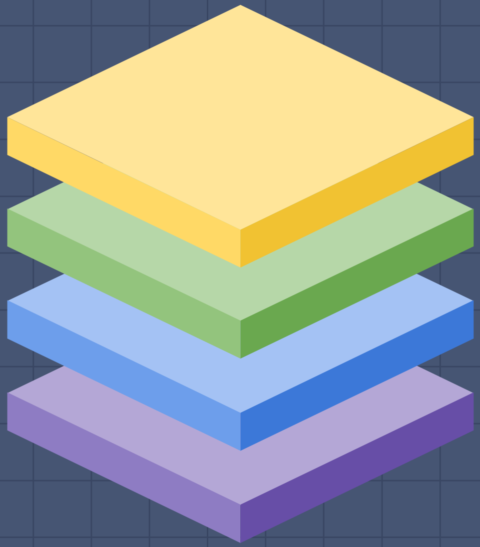
Winning Space Race with Data Science

Carlos Alejandro Arreola Trujillo

17 Oct 2021

Outline

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



Executive Summary

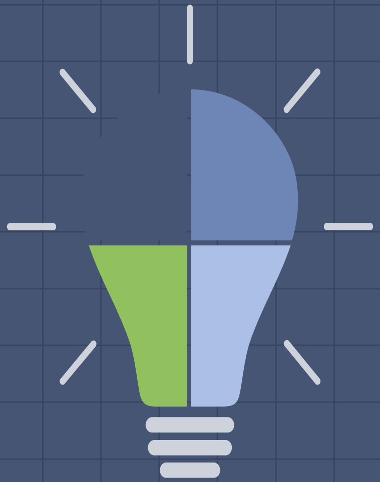
- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - EDA with Data Visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a dashboard in Plotly Dash
  - Predictive Analysis (Classification)
- Summary of all results
  - Exploratory Data Analysis
  - Interactive Analytics in SS
  - Predictive Analysis Results

Introduction

The main objective for this project is to be able to predict if the Falcon 9 from Space X will land have a successful first landing stage.

The main reason why Space X has been able to advertise their launches to be $62million USD; other providers costs upward of $165million USD each. Much of the savings' difference is due to Space X can reuse the first stage.

Therefore, we can know the cost of the launch if we determine if the first stage will land. This could be used by any other company who wants to bid against SpaceX rocket launch.



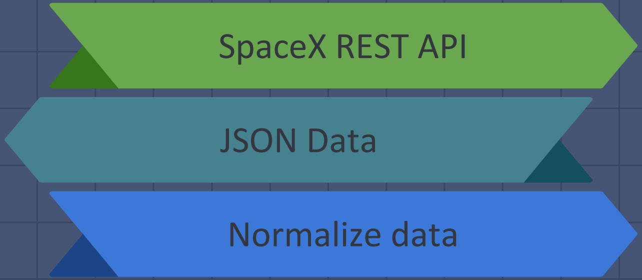
Methodology

Section 1

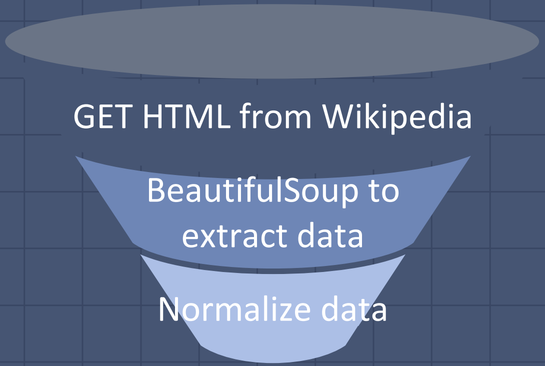
Summary of Methodology

The following data sets were collected by:

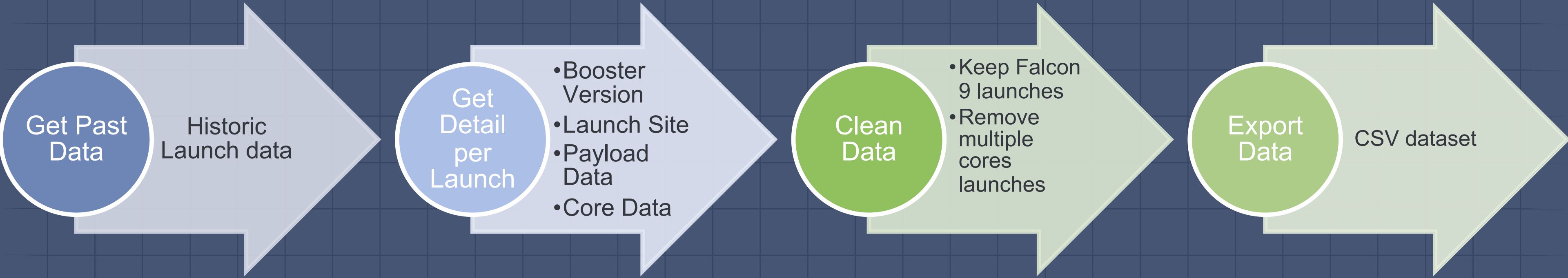
- Launch public data form SpaceX Rest API.
  - [Rocket used, payload delivered, launch specifications, landing specifications, landing outcome]
  - URL api.spacexdata.com/v4/



- Web Scraping Wikipedia using BeautifulSoup
  - Falcon 9 and Falcon Heavy Launches Records



Data Collection - SpaceX API



GitHub

Notebook API Collection Data

Data Collection – Web Scraping

Wikipedia - List of Falcon 9 and Falcon Heavy Launches

Extract Falcon 9 launch records HTML from the Wikipedia page above

Step 1: Request the Falcon9 Launch Wiki page from URL and create a BeatifulSoup object from the HTML response

```python
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_l
# use requests.get() method with the provided static_url
# assign the response to a object
raw_soup = requests.get(static_url)
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(raw_soup.content,'lxml')
soup
```

Python

Step 2: Extract all column names and tables using BeautifulSoup

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 the end of this lab

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

Python

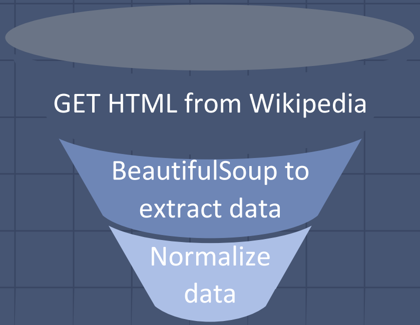
Step 3: Create a data frame by parting the launch HTMLtables trough a dictionary

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

```python
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
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'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```



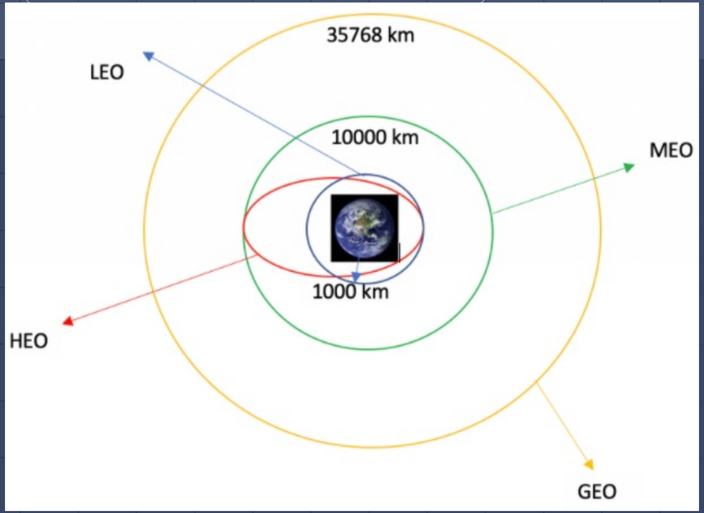
GitHub

Notebook Web Scraping

Data Wrangling

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.



1 Calculate number of launches on each site

2 Calculate number of occurrence of each orbit

3 Calculate number and occurrence of mission outcome per orbit type

4 Create a landing outcome label from Outcome column

GitHub

Notebook Data Wrangling

EDA with Data Visualization

- ❑ Scatter Graphs (outcome used as hue):
  - FlightNumbes vs PayloadMass
  - FlightNumber vs LaunchSite
  - Payload vs Launch Site
  - Relationship between FlightNumber and Orbit type
  - Relationship between Payload and Orbit type

- ❑ Bar Charts:
  - Relationship between Success rate and Orbit type

- ❑ Line Graphs:
  - Success Rate by Year

- ❑ Scatter plots show the correlation between two variables and how it affects an outcome

- ❑ Bar graphs are easy to compare sets of data between different groups

- ❑ Line graphs are used to show easily trends on data variables

GitHub

Notebook EDA Data Visualization

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch site begin with ‘CCA’
- Calculate Total Payload Mass carried by boosters launched by NASA (CRS)
- Calculate Average Payload Mass carried by booster version F9 v1.1
- List date of first successful landing in ground pad
- List names of the booster which have success in drone ship and have a payload mass > than 4000 but < than 6000
- List Total of number of successful and failure mission outcomes
- List names of the booster version which have carried the maximum payload mass
- List the failed outcomes in drone ship, their booster versions and launch site names in 2015
- Total of landing outcomes (Drone Ship or Ground pad) between 2010-2017

GitHub

Notebook EDA with SQL

Interactive Map with Folium

- Using the latitude and longitude coordinates at each launch and add a Circle Marker around each site with a label of it's name.
- Assigning Launch Outcomes (failures and successes) to 1, 0 classes with green and red markers on the map
- Using 'Havernesine's formula we calculate the distance between multiple landmarks to find various trends to measure patterns

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

GitHub

Notebook Folium Interactive Map

Dashboard with Plotly Dash

- Launch Site Drop-down Input Component

- Success Pire Chart based on the selected site dropdown

Size of the circle can be made proportional to the total quantity it represents

- Range slider to select Payload

- Success Payload Scatter Chart

Showing the relationship of the outcome with PayloadMass for the different Booster Versions

GitHub

Notebook Folium Interactive Map

Predictive Analysis (Classification)

Building Model

- Transform Data
- Split in training and test data (80-20)
- ML Algorithms to be used
- Set parameters and algorithms to GridSearchCV
- Fit and Train

Model Evaluation

- Accuracy for each model
- Tuned Hyperparameters
- Confusion Matrix

Improving Model

- Tuning and Feature Engineering

Finding best performing Classification Model

- Calculate accuracy score for each model

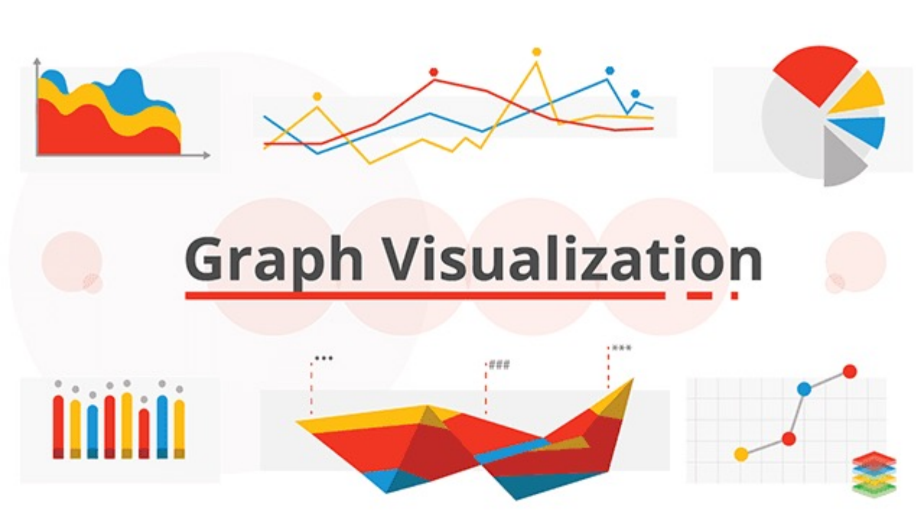
GitHub

Notebook Classification Model

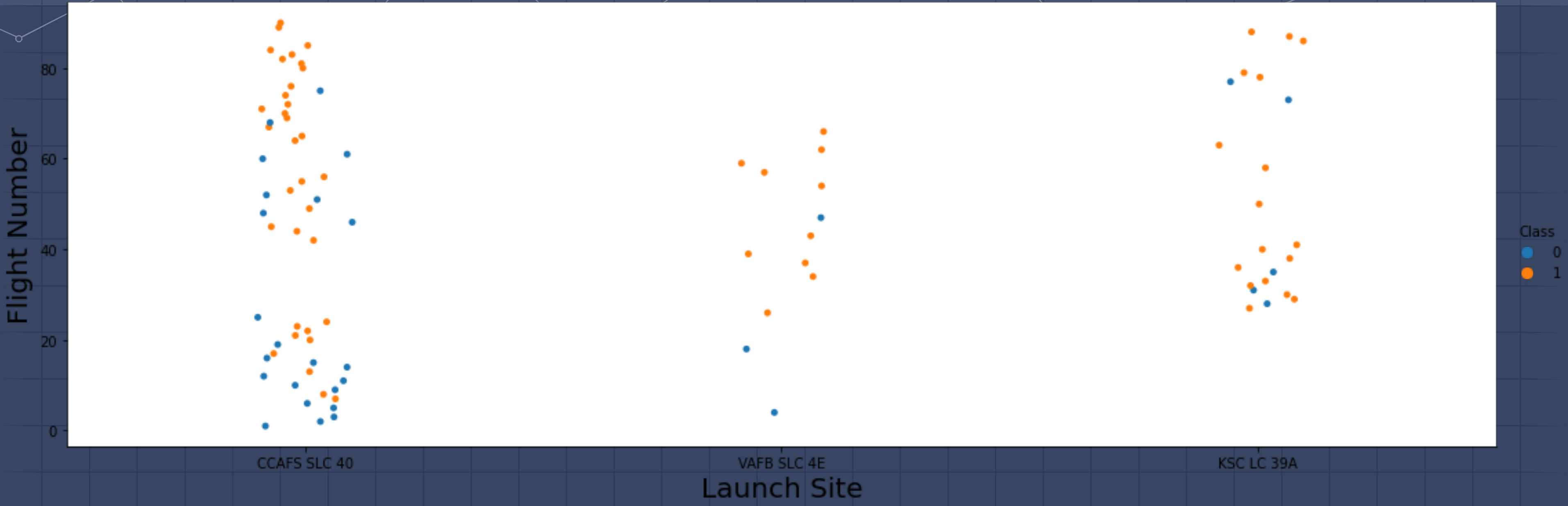
Results

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

EDA with Visualization

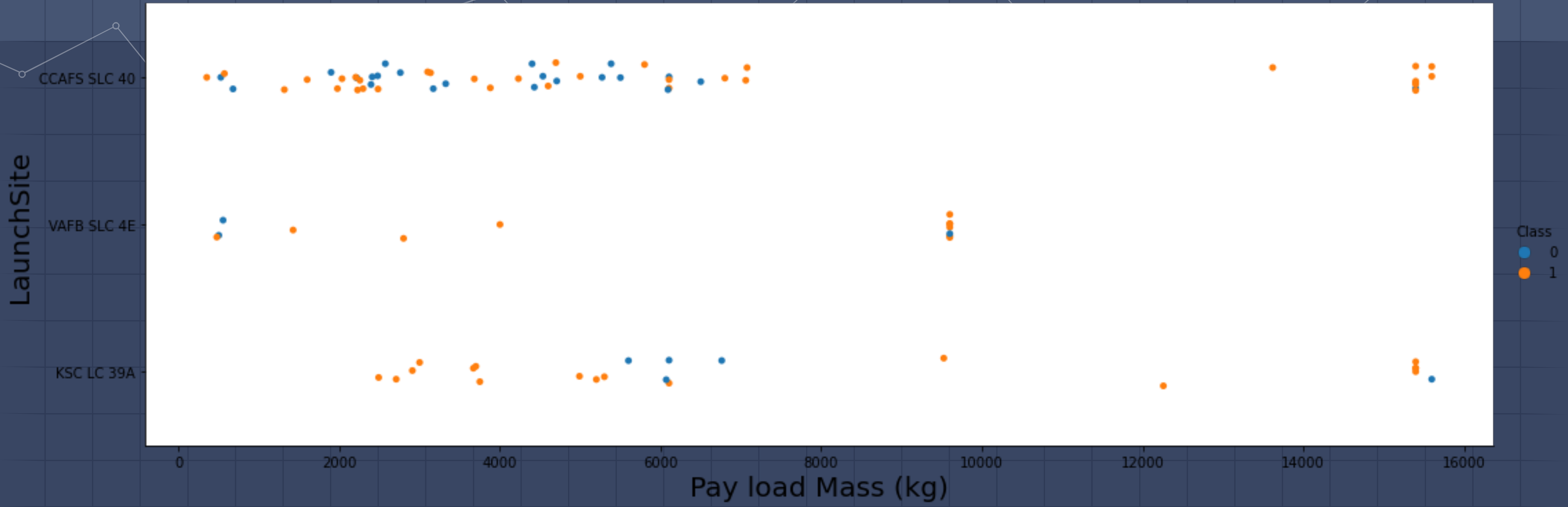


Flight Number vs Flight Site



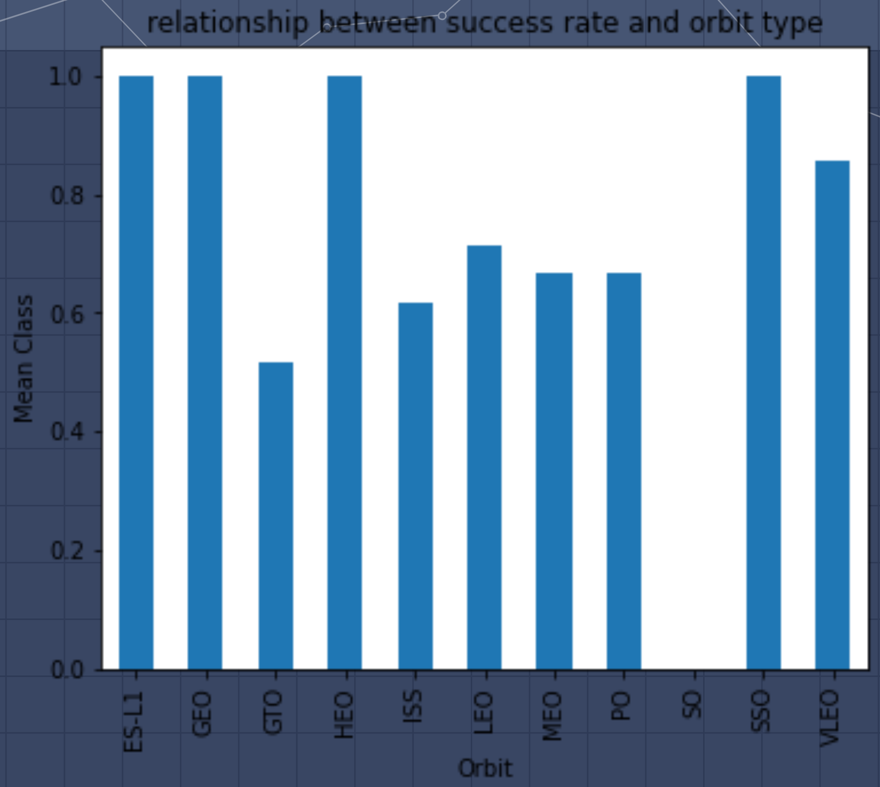
The more amount of flights at a Launch site there is a higher success rate

Payload Mass vs Launch Site



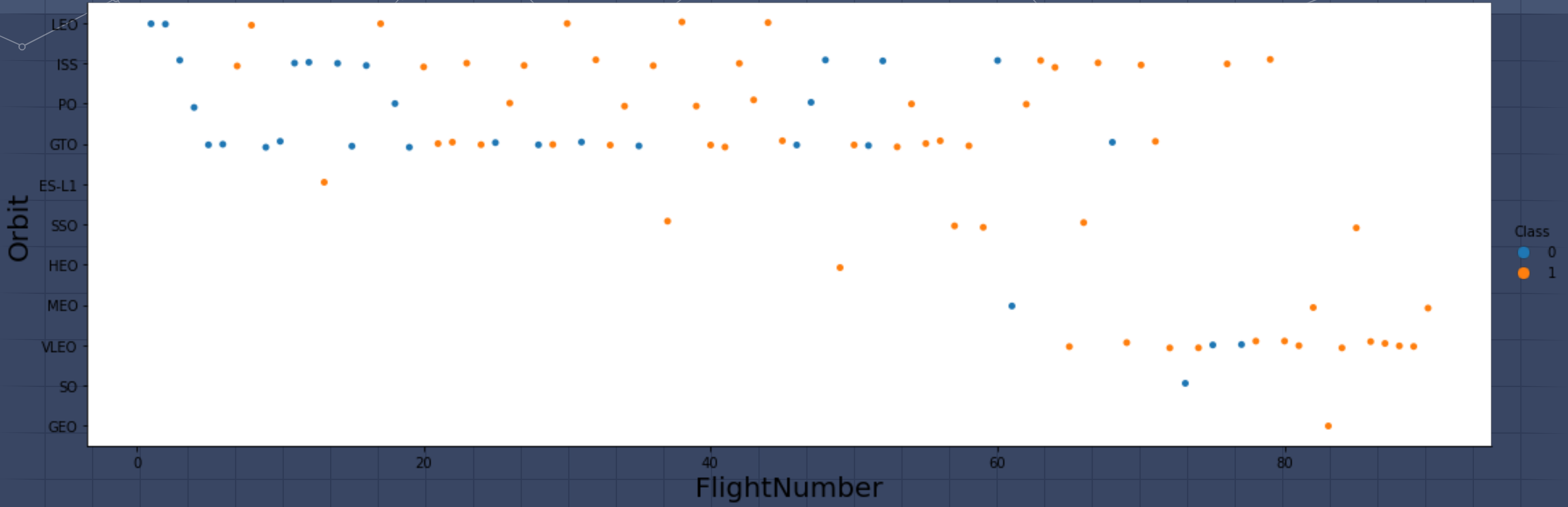
The more amount of flights at a Launch site there is a higher success rate

Bar Chart: Success Rate vs Orbit type



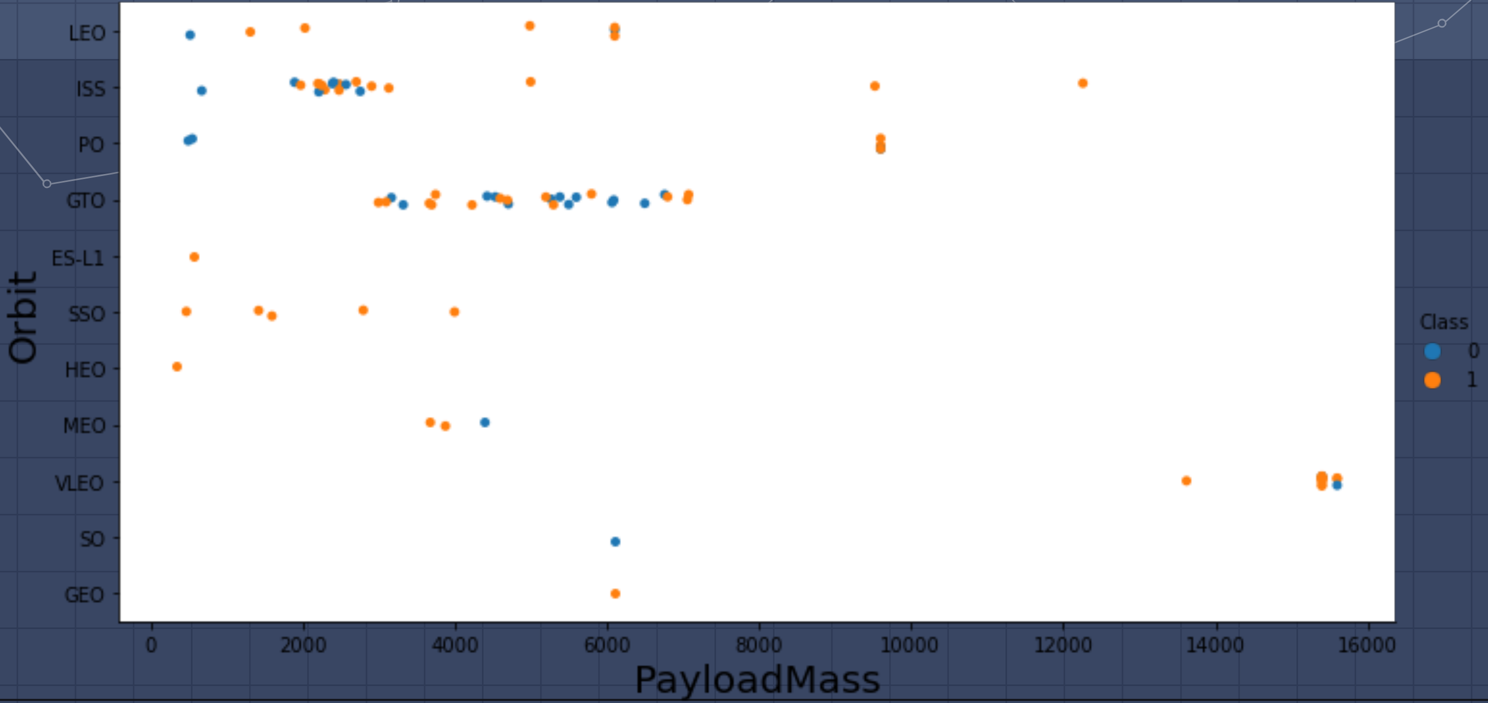
Orbit ES-L1, GEO, HEO, SSO VLEO

Flight Number vs Orbit type



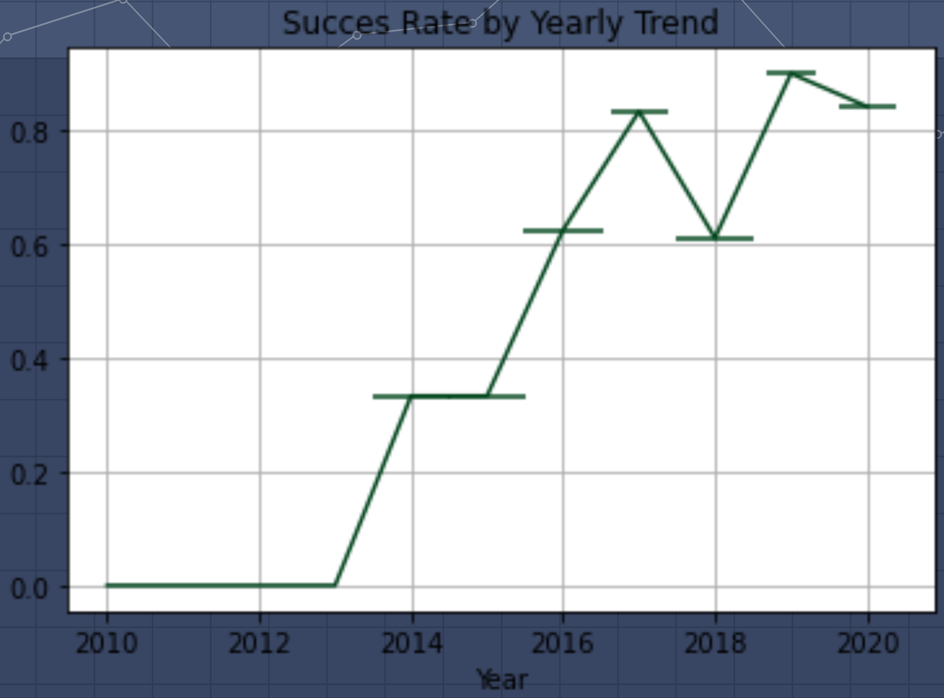
LEO orbit Success appears to be related to the number of flights, in the other hand appears to be no relationship between it and GTO orbit

Payload vs Orbit type



Heavy Payload's appear to have negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits

Success Rate Trend by Year



Success Rate keep increasing from 2013 up to 2020

EDA with SQL

1)Display Names of the Launch Sites

```
%sql SELECT Unique("Launch_Site") FROM "FCR91149"."DATASPACEX";
```

Python

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

2) List 5 Records where Launch Site begin with CCA

```
%sql SELECT * FROM "FCR91149"."DATASPACEX" WHERE "Launch_Site" like 'CCA%';
```

Python

Date	Time__UTC_	Booster_Version	Launch_Site	Payload	payload_mass__kg_	Orbit
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO
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)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)

3)Calculate Total Payload mass carried by boosters Launched by Nasa (CRS)

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") as "Total Payload" FROM
"FCR91149"."DATASPACEX" WHERE "Customer" = 'NASA (CRS)';
```

Python

Total Payload
45596

4)Calculate Average Payload mass carried by booster version F9 v1.1

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") as "Average Payload"
FROM "FCR91149"."DATASPACEX" WHERE "Booster_Version" LIKE 'F9 v1.1%';
```

Python

Average Payload
38020

5)When was achieved the first successful landing outcome in ground pad?

```
%sql SELECT MIN("Date") FROM "FCR91149"."DATASPACEX"
WHERE UPPER("Landing__Outcome") LIKE '%SUCCESS%';
```

1
2015-12-22

6)List the boosters which have success in drone shipping and have payload mass greater than 4000 but less than 6000

```
%sql SELECT UNIQUE("Booster_Version") FROM "FCR91149"."DATASPACEX"
WHERE ("Landing__Outcome" = 'Success (drone ship)') AND (
    ("PAYLOAD_MASS__KG_">4000)AND("PAYLOAD_MASS__KG_"<6000))
```

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

7) List Total Number of successful and failure mission outcomes

```
%sql SELECT UNIQUE("Mission_Outcome"),COUNT("Orbit") as "Total Number"
FROM "FCR91149"."DATASPACEX" WHERE UPPER("Landing__Outcome")
LIKE '%SUCCESS%' Group by "Mission_Outcome";
```

Mission_Outcome	Total Number
Success	60
Success (payload status unclear)	1

8) List the names of Booster version which have carried the maximum payload mass

```
%sql SELECT UNIQUE("Booster_Version") FROM "FCR91149"."DATASPACEX"
WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_")
                            FROM "FCR91149"."DATASPACEX")
```

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

9) List the failed landings in drone ship, their boost versions and launch site names for in year 2015

```
%sql SELECT UNIQUE("Mission_Outcome"),COUNT("Orbit") as "Total Number"
FROM "FCR91149"."DATASPACEX" WHERE UPPER("Landing__Outcome")
LIKE '%SUCCESS%' Group by "Mission_Outcome";
```

Mission_Outcome	Total Number
Success	60
Success (payload status unclear)	1

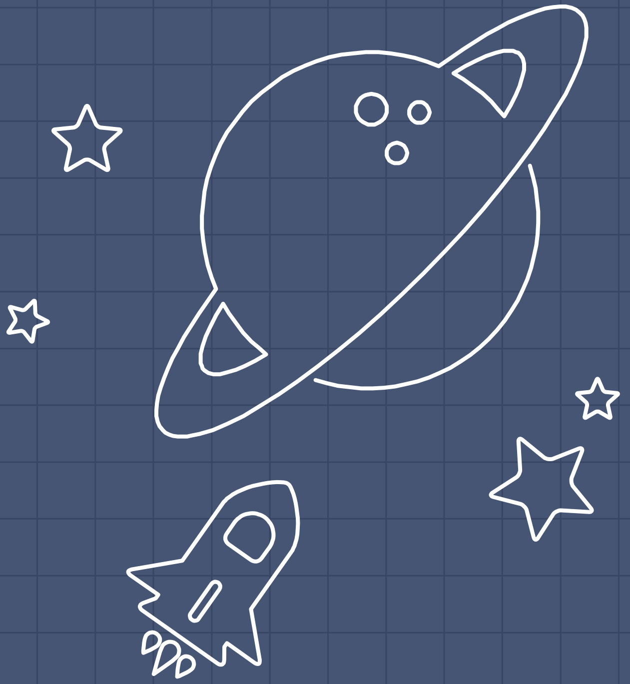
10) Count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order

In [12]:

```
%sql SELECT ROW_NUMBER() OVER(ORDER BY "Date" desc) RowNumber,
"Landing__Outcome","Mission_Outcome","Launch_Site"
FROM "FCR91149"."DATASPACEX"
WHERE ("Landing__Outcome" = 'Failure (drone ship)' OR
        "Landing__Outcome" = 'Success (ground pad)') AND (
    "Date" BETWEEN '2010-06-04' AND '2017-03-20')
```

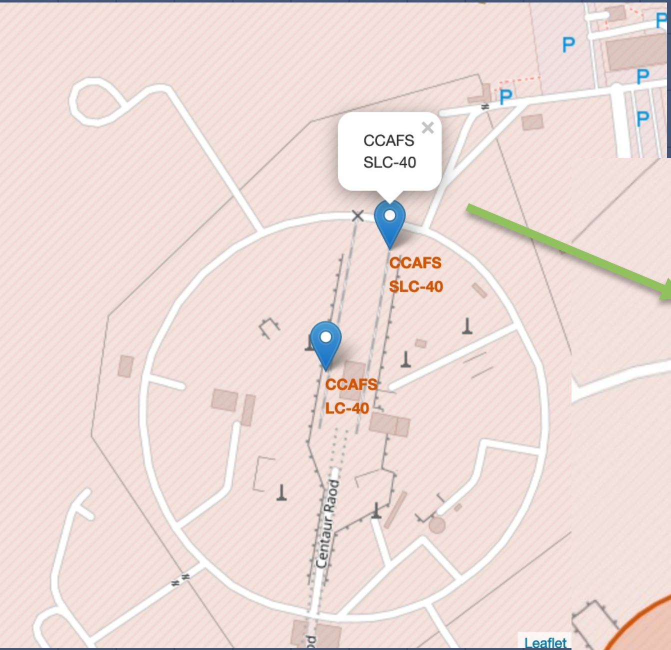
ROWNUMBER	Landing__Outcome	Mission_Outcome	Launch_Site
1	Success (ground pad)	Success	KSC LC-39A
2	Success (ground pad)	Success	CCAFS LC-40
3	Failure (drone ship)	Success	CCAFS LC-40
4	Failure (drone ship)	Success	CCAFS LC-40
5	Failure (drone ship)	Success	VAFB SLC-4E
6	Success (ground pad)	Success	CCAFS LC-40
7	Failure (drone ship)	Success	CCAFS LC-40
8	Failure (drone ship)	Success	CCAFS LC-40

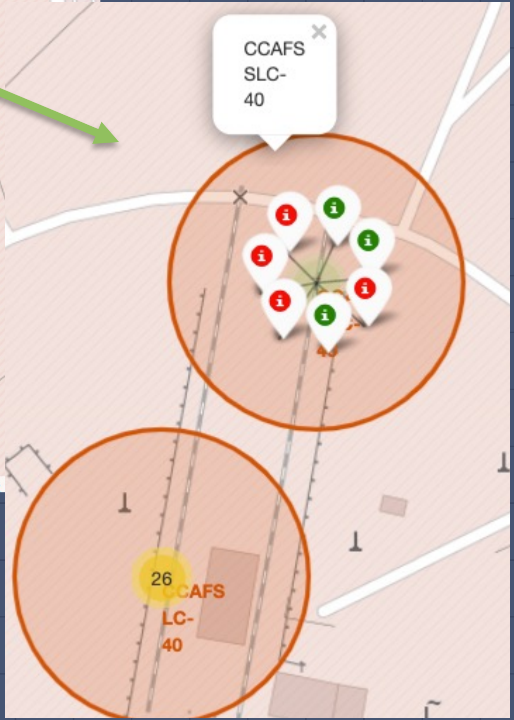
Interactive Map with Folium




+
−
VAFB
SLC-4E
Utah
Colorado
Kansas City
Nevada
Kansas
Missouri
West Virginia
Kentucky
Fresno
Las Vegas
Tulsa
Albuquerque
Oklahoma
Tennessee
Memphis
Charlotte
Arkansas
Arizona
New Mexico
Los Angeles
Atlanta
South Carolina
Phoenix
Dallas
Tijuana
Mexicali
Tucson
Mississippi
Alabama
Georgia
Ciudad Juárez
Texas
Louisiana
Baja California
Austin
Houston
Baton Rouge
Sonora
San Antonio
Chihuahua
Florida
Coahuila de Zaragoza
Torreón
Monterrey
Reynosa
Baja California Sur
Saltillo
Tamaulipas
Miami
La Paz
Culiacán
Durango
México
Zacatecas
San Luis Potosí
La Habana
Aguascalientes
Cuba
Mérida
León
Jalisco
Cancún
Ciudad de México
Morelia
Veracruz
Puebla
Leaflet | Data by © OpenStreetMap, under ODbL.

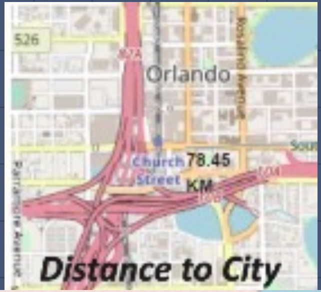
Florida Launch Site
Zoom on click

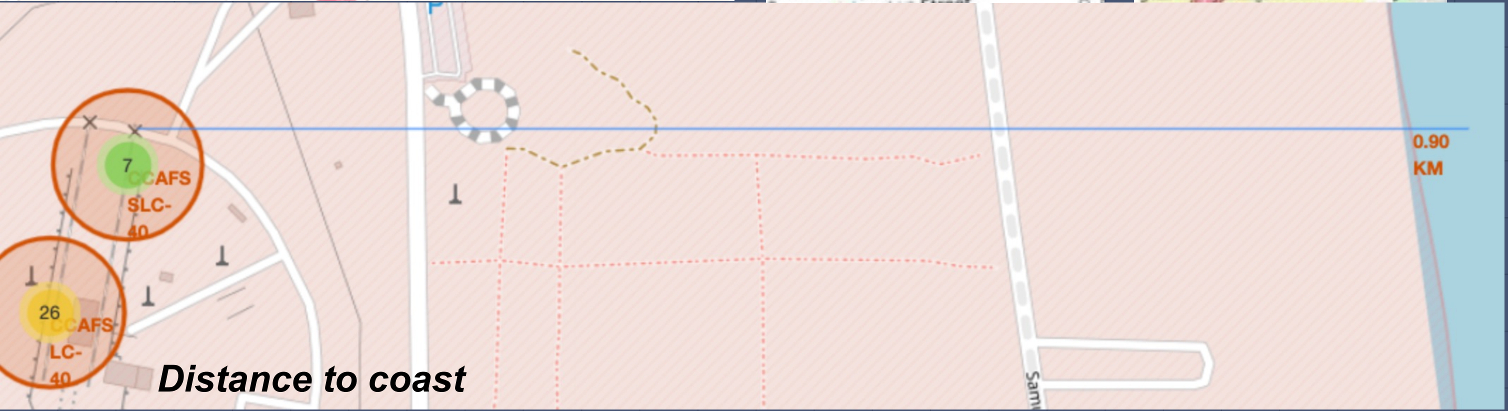




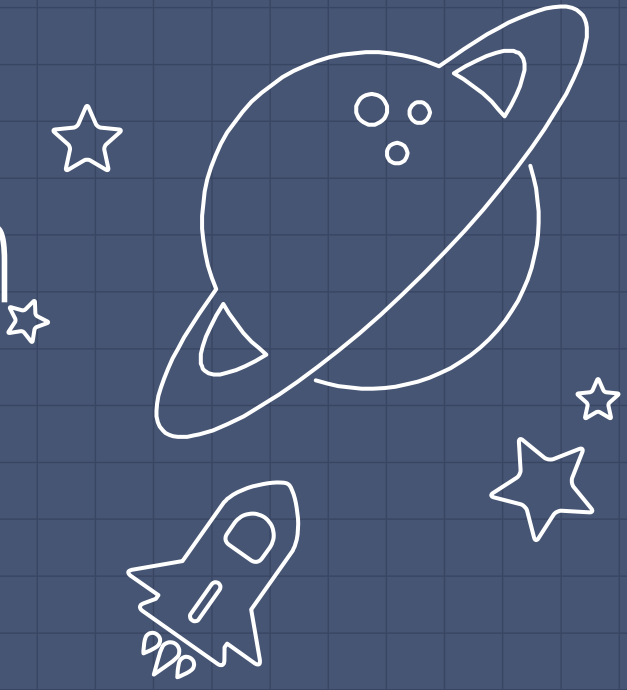
California Launch Site

Launch Site distance to landmarks to find trends





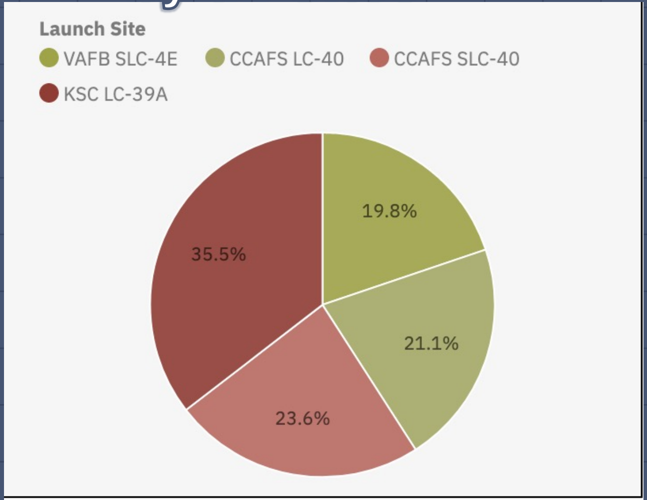
Interactive Dashboard with IBM Cognos Dashboard



IBM Cognos Dashboard Embedded

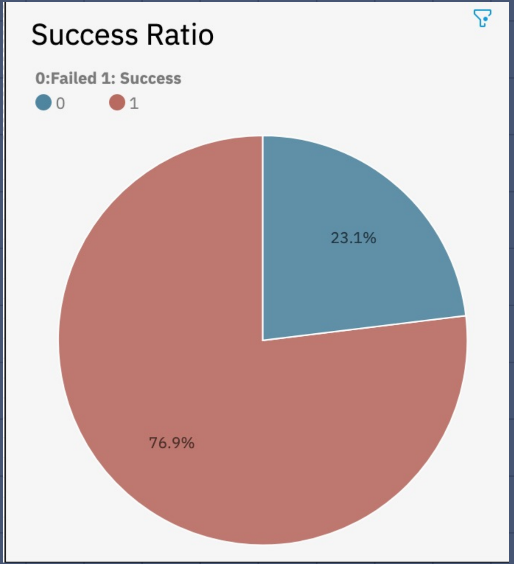
Click on logo to go to dashboard

Success Launches by all Sites

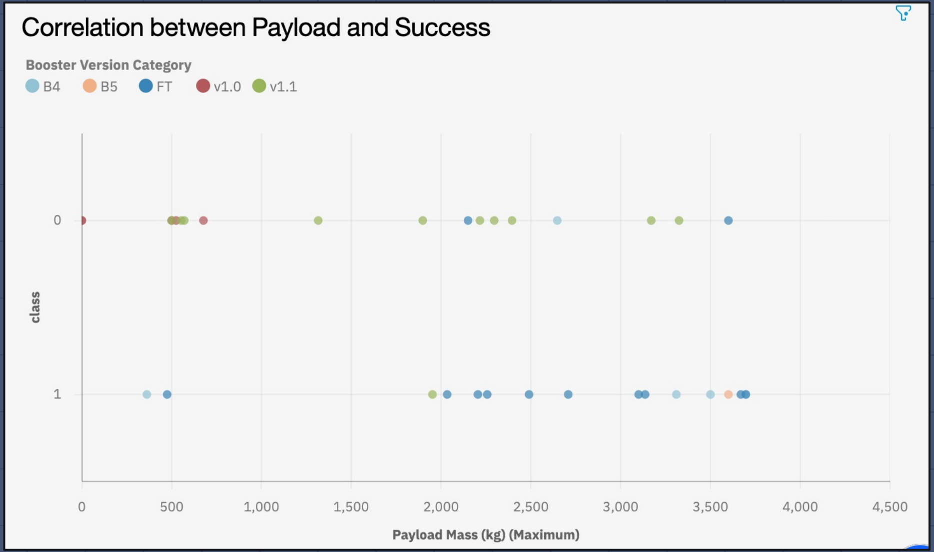


Launch Site with Highest Success Ratio

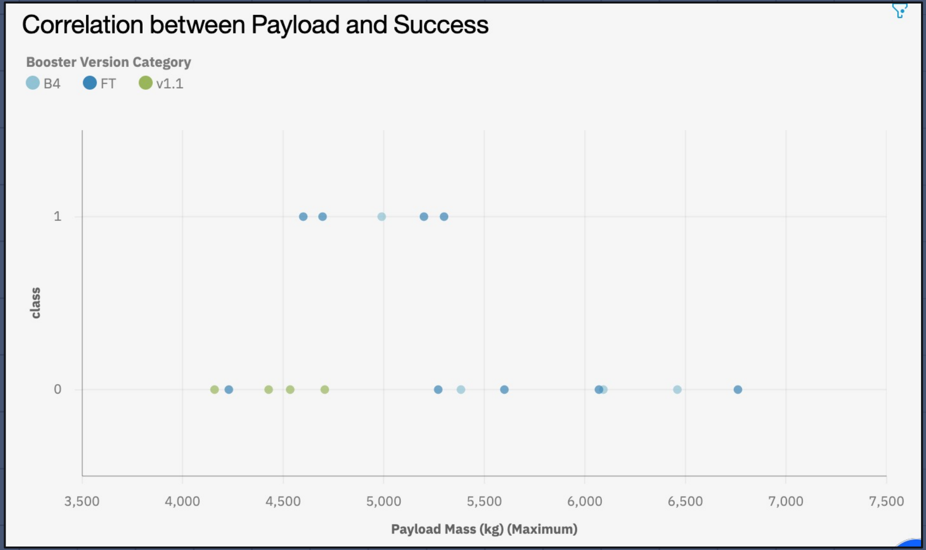
KSCLC – 39A



Low Weighted Payload (<4000kg)



Heavy weighted Payload (>4000kg)



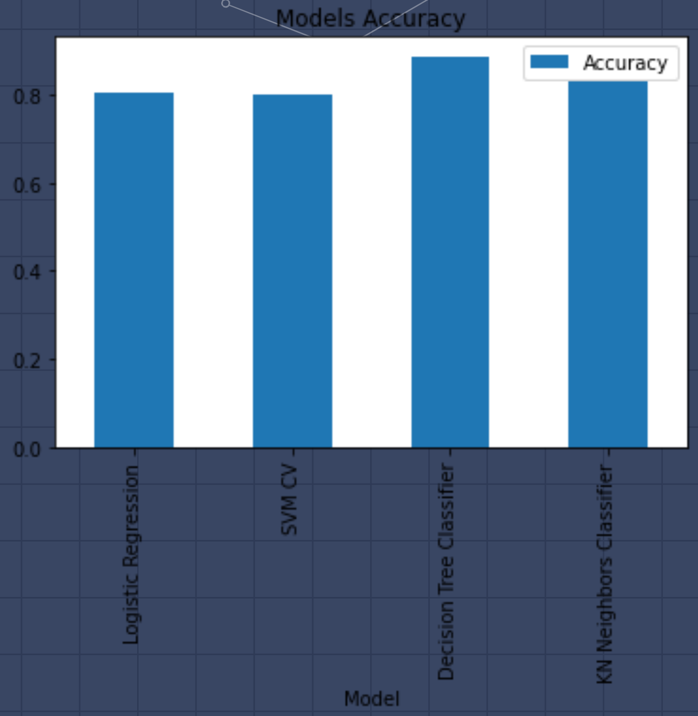
Predictive Analysis (Classification)

Model Selection

Models Accuracy

Model	Accuracy	Hyper Params
Logistic Regression	0.803571	{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
SVM CV	0.801786	{'C': 1.0, 'gamma': 0.03162277660168379, 'kern...
Decision Tree Classifier	0.887500	{'criterion': 'entropy', 'max_depth': 10, 'max...
KN Neighbors Classifier	0.830357	{'algorithm': 'auto', 'n_neighbors': 7, 'p': 1}

The Decision Tree Classifier wins as the model with the highest accuracy.



Confusion Matrix

Having a look at the confusion matrix we can see out model can distinguish between the classes having a slight error on False Negative's

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

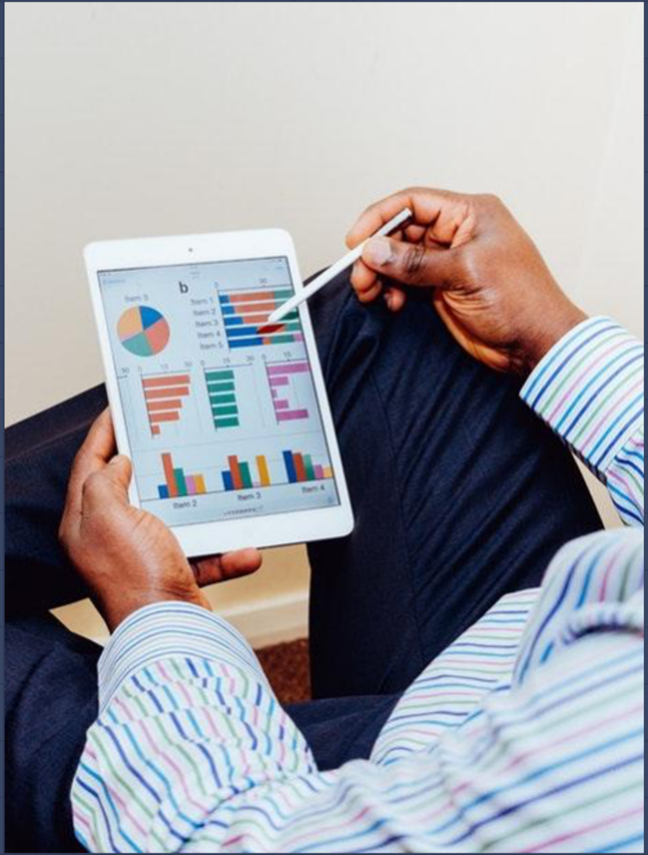
Conclusion

For this data solution the best model to work with is the Tree Classifier Algorithm.

KSC LC-39A had the most successful launches from all sites.

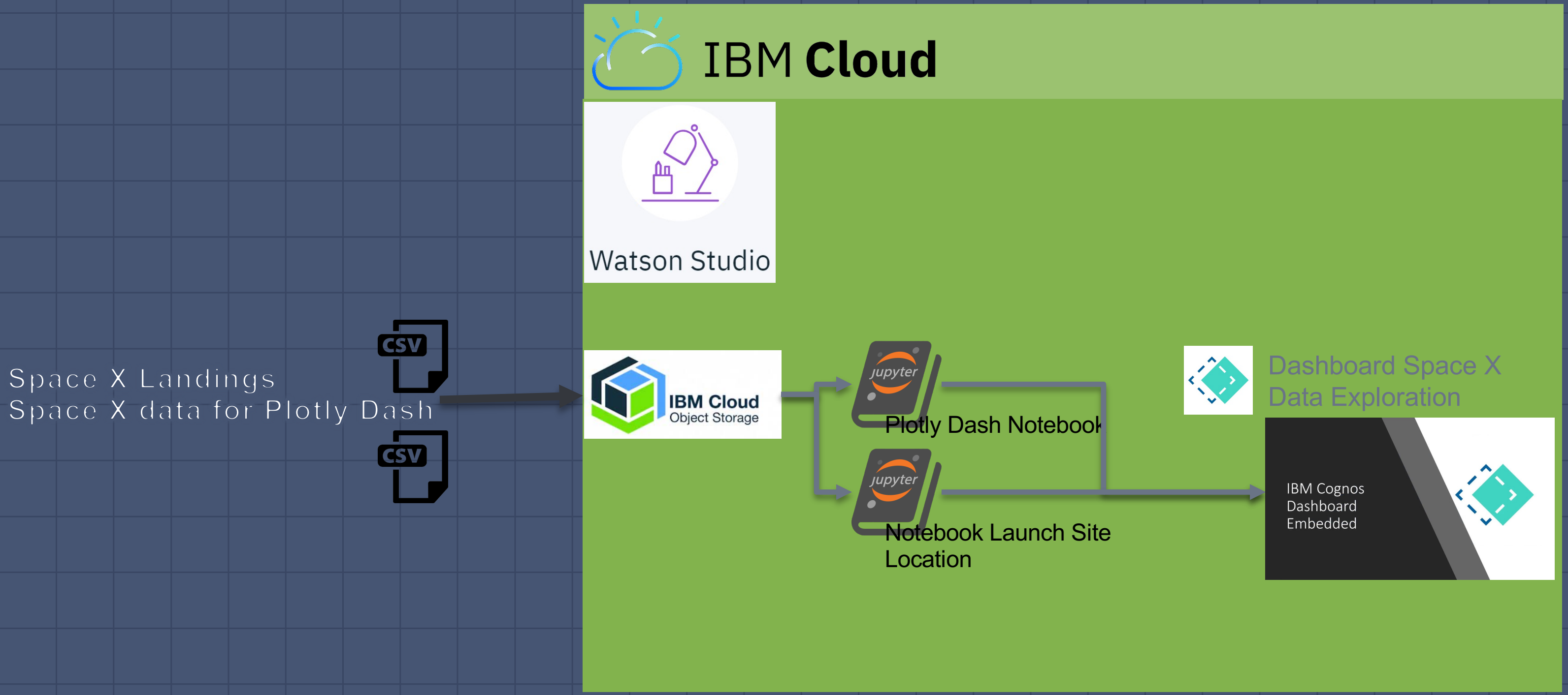
Orbits SSO, ES-L1, GEO, HEO have the best success rates.

Low payloads perform better landings than the heavier ones.



Appendix

Dashboard in Watson Studio Architecture



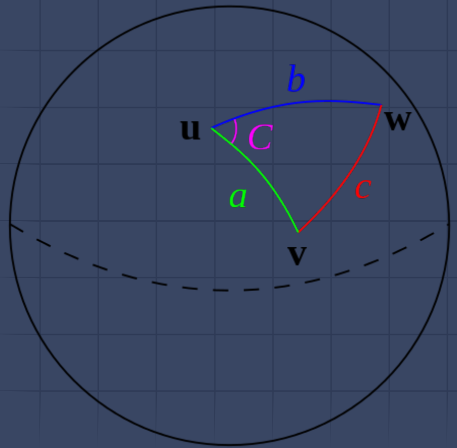
Haversine Formula

The haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes. Important in navigation, it is a special case of a more general formula in spherical trigonometry, the law of haversines, that relates the sides and angles of spherical triangles.

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi 1 \cdot \cos\varphi 2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 \cdot \operatorname{atan2}\left(\sqrt{a}, \sqrt{(1-a)}\right)$$

$$d = R \cdot c$$



THANK YOU