IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Boyke Kurniawan
January, 14th 2022

Outline

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

Executive Summary

Summary of methodologies

- Data collection
- Data wrangling
- EDA with data visualization
- EDA with SQL
- Build an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis (Classification)

- ## Summary of all results

- Exploratory data analysis
- Interactive analytics
- Predictive analysis

Introduction

Project background and context

SpaceX advertised that Falcon 9 rocket launches on its website, with a cost of 62 million dollars Other company offer cost above 165 million dollars each, this is because SpaceX can reuse the first stage. If we can determine if the first stage will land, we can determine the cost of a launch.

Problems you want to find answers

- What aspect 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.

Section 1

Methodology

Methodology

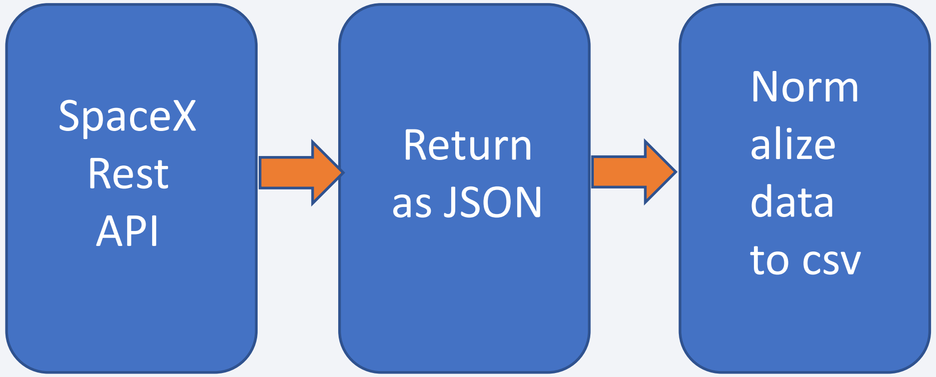
Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning
- 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

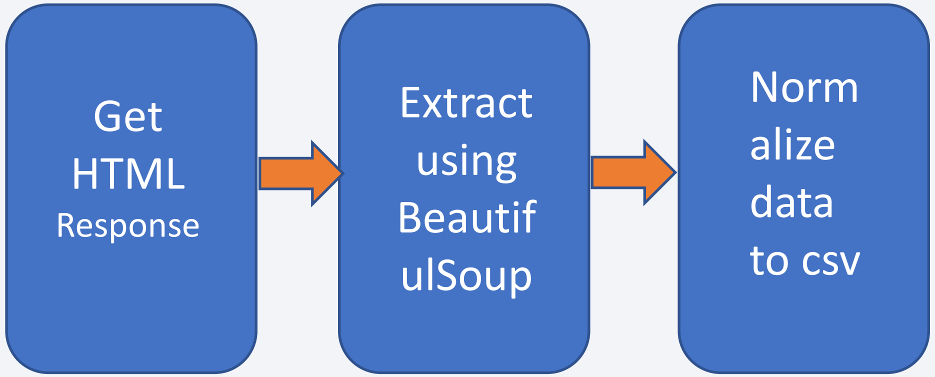
Data Collection

- We collected data from SpaceX Rest API
- Web scrapping Wikipedia using BeautifulSoup

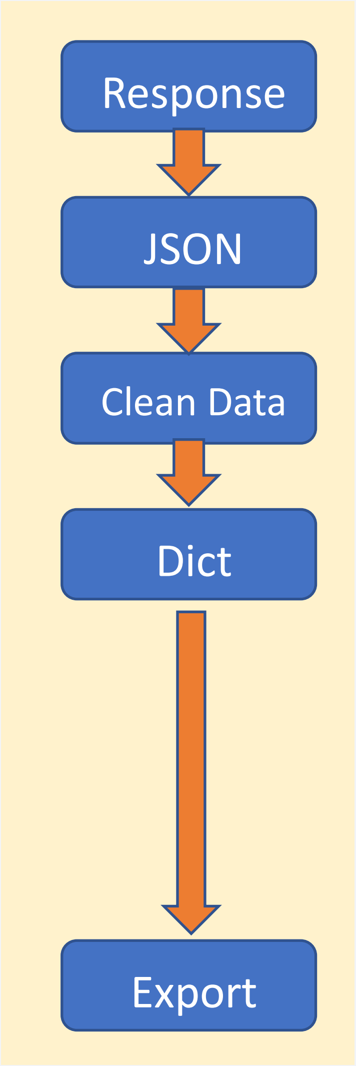
Rest API



Web Scrapping



Data Collection – SpaceX API




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

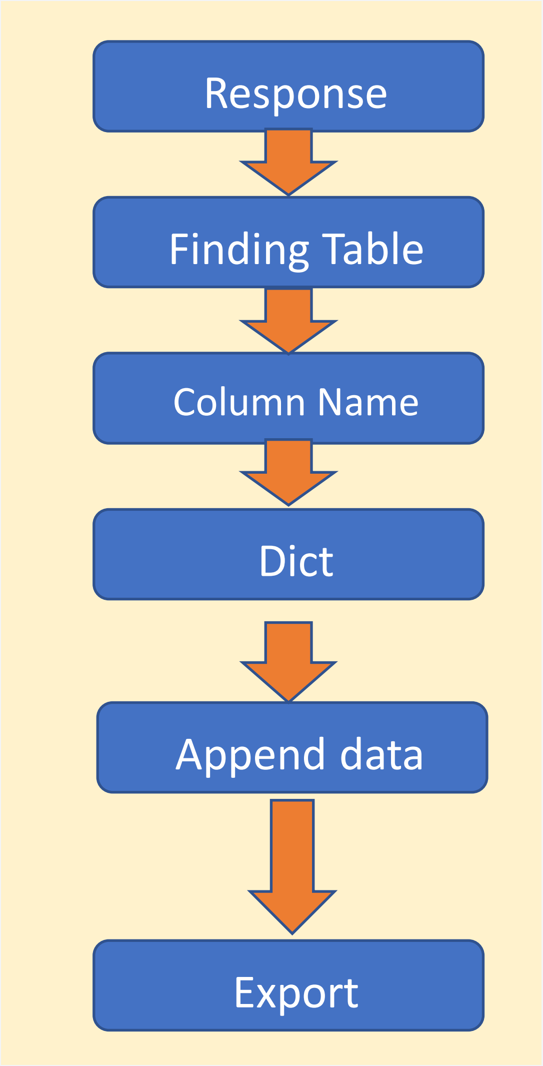
```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

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

```
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}
```

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

Data Collection - Scraping




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

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

```
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
```

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with eac
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']=[]
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading i
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.s
                flag=flight_number.isdigit()
        else:
            flag=False
```

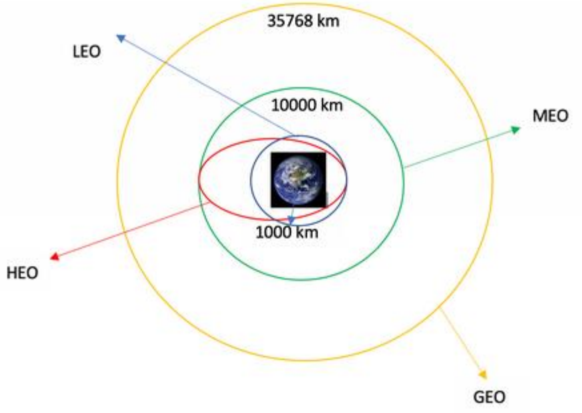
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

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.

Github

Diagram showing some common orbit types



Calculate the number of launches

Calculate the number and occurrence of each orbit

Calculate the number and occurence of mission outcome per orbit type

Export to .csv

Create a landing outcome label from Outcome column

EDA with Data Visualization

Scatter Graphs

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

Bar Graph

Mean VS. Orbit

Line Graph being drawn:

Success Rate VS. Year

Line Graph

Success Rate VS. Year

Github

EDA with SQL

Performed SQL queries

- Displaying the names of the unique launch sites in the space mission
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Displaying the date where the successful landing outcome in drone ship was achieved.
- Displaying the total number of successful and failure mission outcomes
- Displaying the names of the booster versions which have carried the maximum payload mass.
- Displaying 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.

Github

Build an Interactive Map with Folium

Visual the Launch Data into an interactive map.

Using 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.

<u>Example of some trends in which the Launch Site is situated in.</u>

- 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

Build a Dashboard with Plotly Dash

- Creating dashboard is built with Dash web framework.

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.

Pie Chart showing the total launches by 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.

Github

Predictive Analysis (Classification)

BUILDING MODEL

- Load dataset into NumPy and Pandas
- Transform Data and 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.

Github

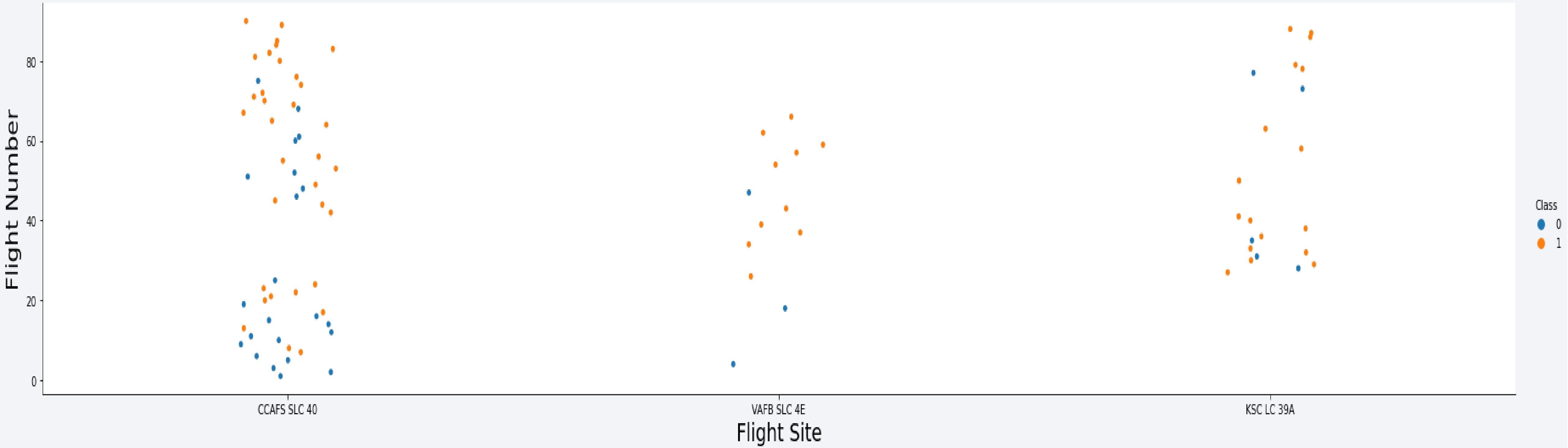
Results

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

Section 2

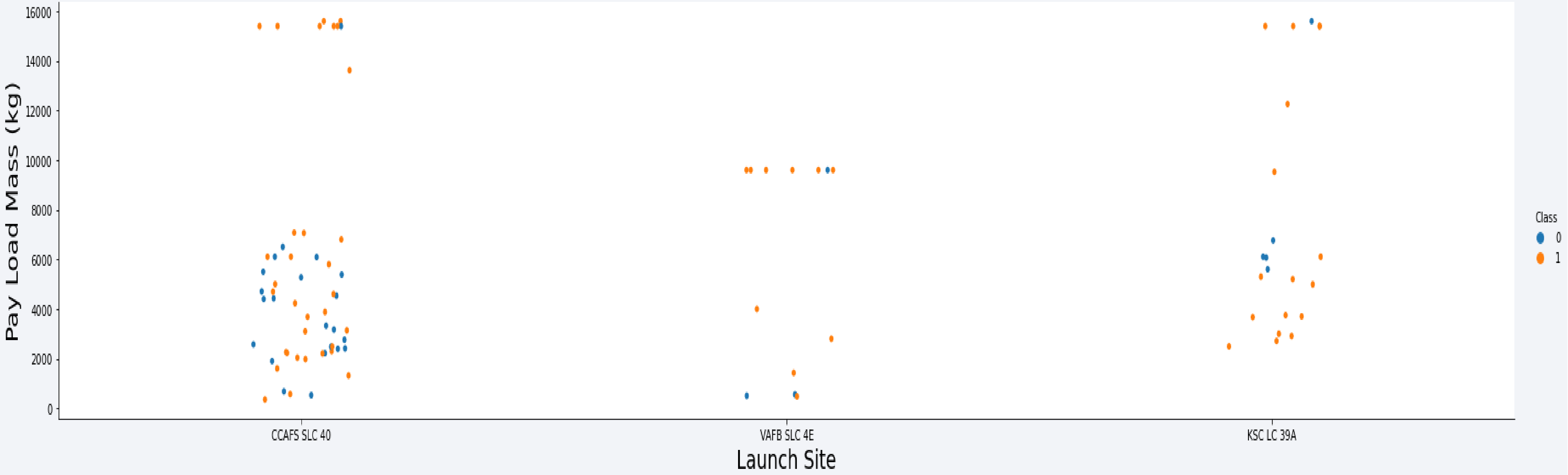
Insights drawn from EDA

Flight Number vs. Launch Site



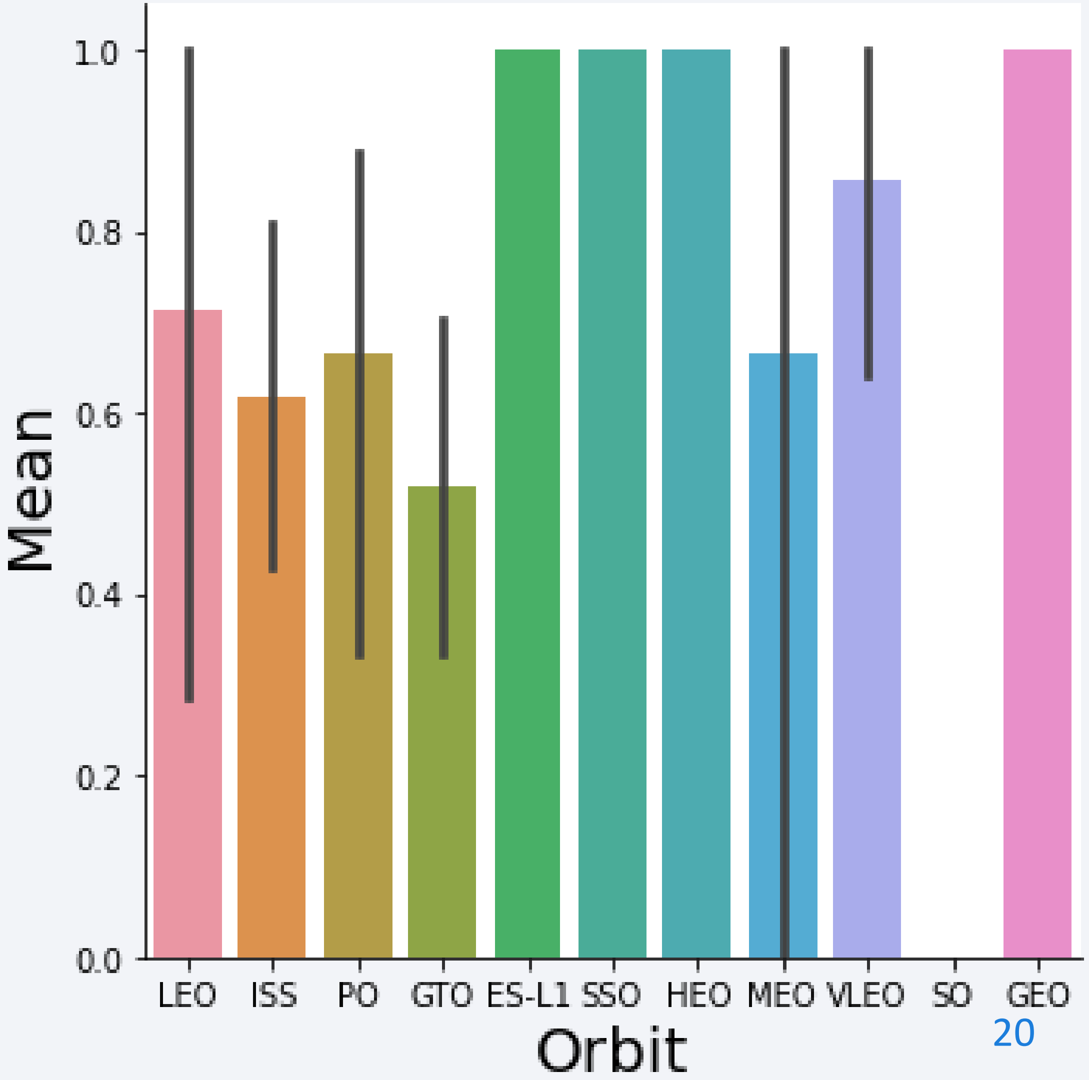
The more amount of flights at a launch site the greater the success rate at a launch site

Payload vs. Launch Site



The greater the payload mass for Launch Site CCAFS SLC 40 the highest success rate for the Rocket.

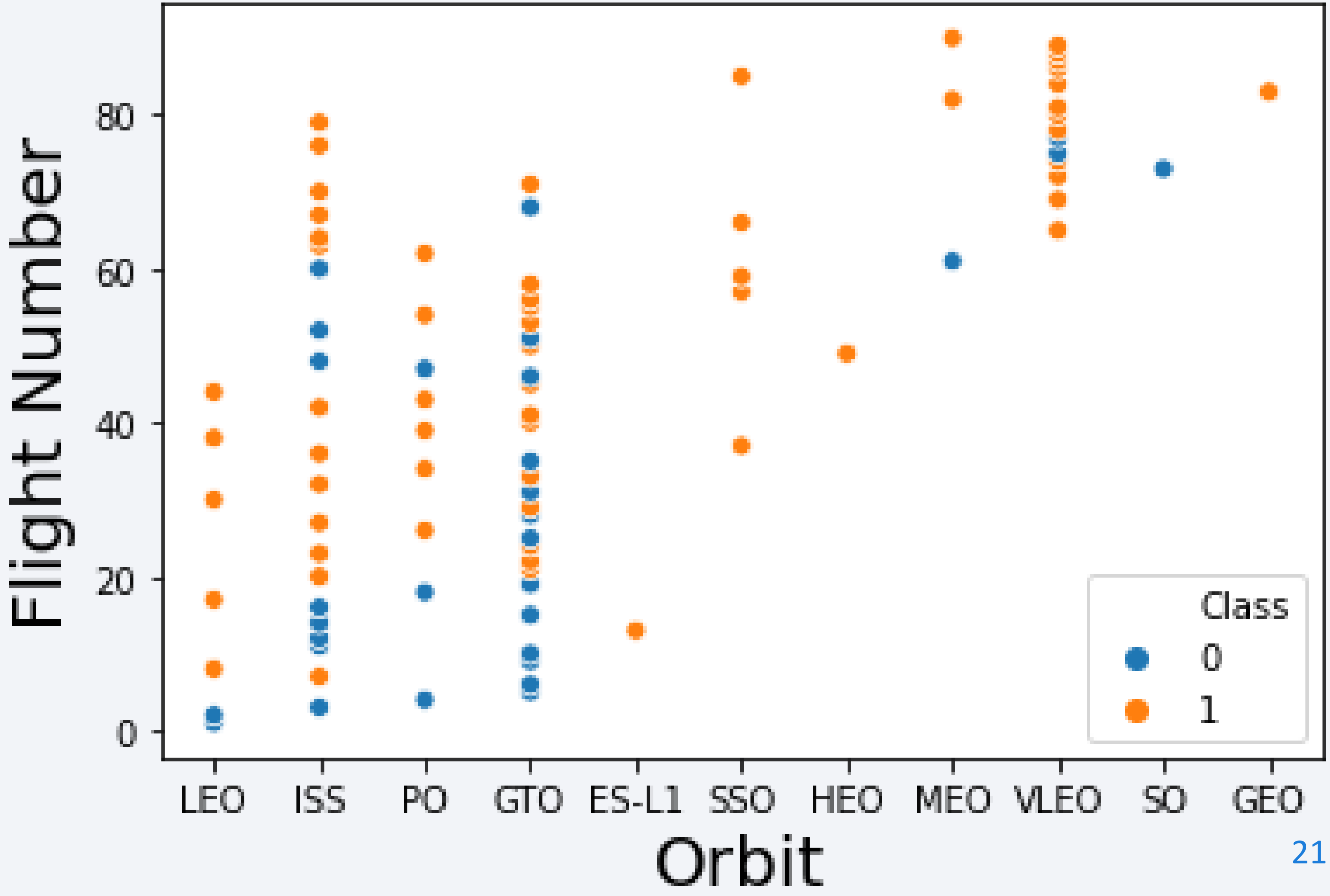
Success Rate vs. Orbit Type



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

Flight Number vs. Orbit Type

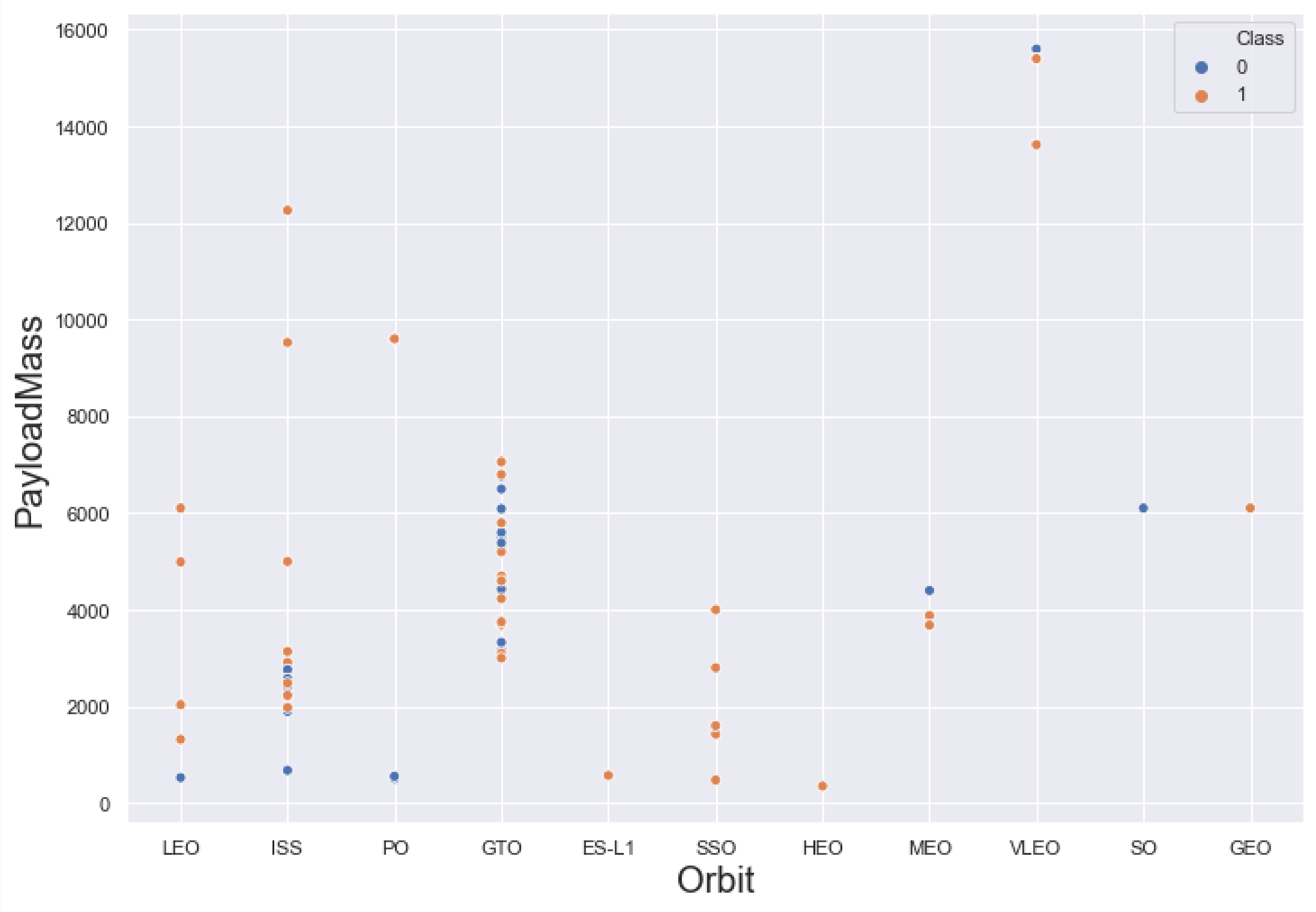
Graph showing random conclusion, LEO appears related to the number of flights; but it seems to be no relationship between flight number when in GTO orbit.



Payload vs. Orbit Type

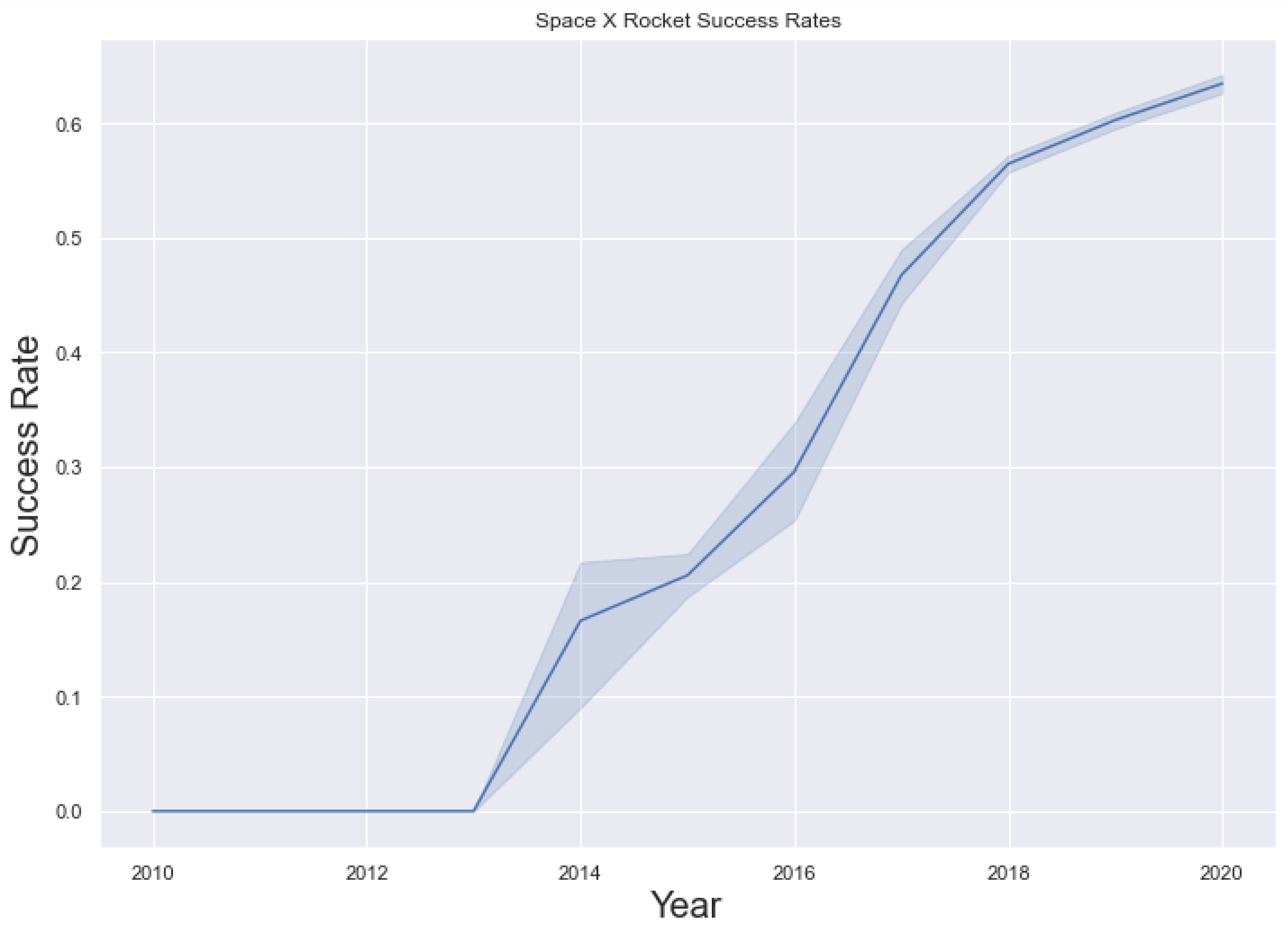
Heavy payloads have a successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this both positive and negative landing rate are present



Launch Success Yearly Trend

Sucess rate since 2013 kept increasing till 2020



All Launch Site Names

```
select distinct launch_site from spacex
```

Using distinct to select unique value

Unique launch sites
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

```
select * from spacex
where launch_site like 'CCA%'
limit 5
```

Using *where* to filter launch site and *limit* to limit 5 records

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

Using combination *sum* and *where* to filter and sum payload mass

```
select sum(payload_mass__kg_) from spacex
where customer ='NASA (CRS)'
```

1
45596

Average Payload Mass by F9 v1.1

Using combination *avg* and *where* to filter and get average payload mass

```
select avg(payload_mass__kg_) from spacex
where booster_version ='F9 v1.1'
```

1
2928

First Successful Ground Landing Date

Using combination *min* and *where* to find first successful landing outcome on ground pad

```
select min(DATE) from spacex
where landing__outcome = 'Success (ground pad)'
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Using combination where to find booster version that successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
%%sql
select booster_version, la                              oad_mass__kg_ from spacex
where landing__outcome = ':
      and payload_mass__kg                              000
```

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

```
select landing__outcome, count( landing__outcome) from spacex
group by landing__outcome
```

landing__outcome	2
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	22
recluded (drone ship)	1
Success	38
Success (drone ship)	14
uccess (ground pad)	9
Uncontrolled (ocean)	2

Using *count* and *Group by* to summarize landing outcome

Boosters Carried Maximum Payload

```
select booster_version from spacex
where payload_mass__kg_ in (select max(payload_mass__kg_) from spacex)
```

Using *subquery* to find booster_versions which have carried the maximum payload mass

booster_version
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

```
select booster_version, launch_site, landing__outcome from spacex
where year(DATE) =2015 and landing__outcome = 'Failure (drone ship)'
```

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

Using 2 conditions to find List the failed landing in 2015

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

```
select landing__outcome, count(landing__outcome) from spacex
where DATE between '2010-06-04' and '2017-03-20'
group by landing__outcome
order by count(landing__outcome) desc
```

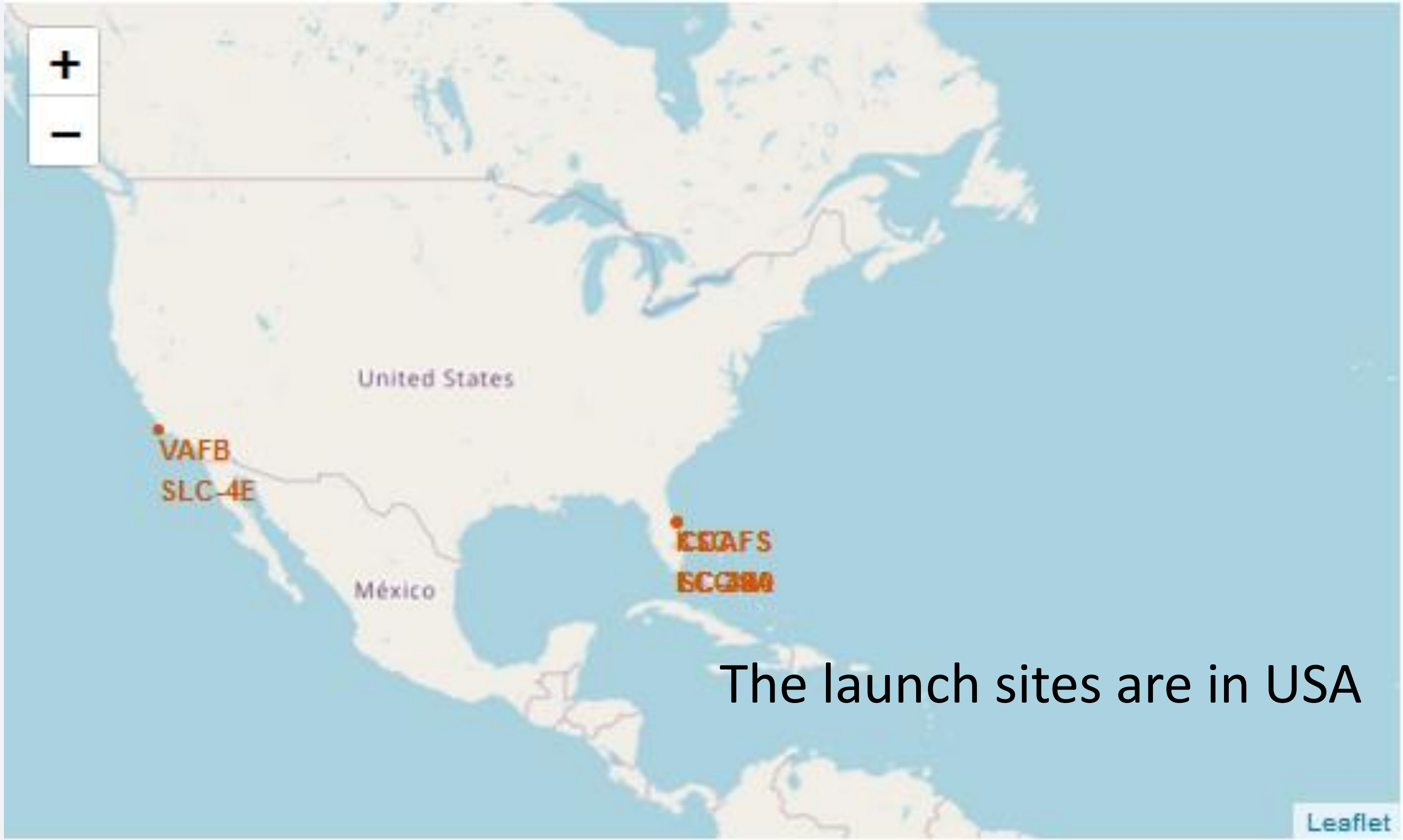
landing__outcome	2
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Find and 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

Section 4

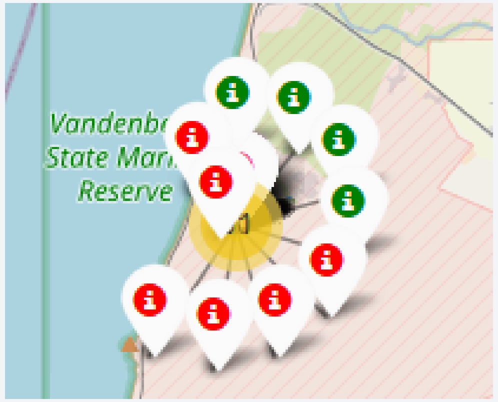
Launch Sites Proximities Analysis

Map of launch sites

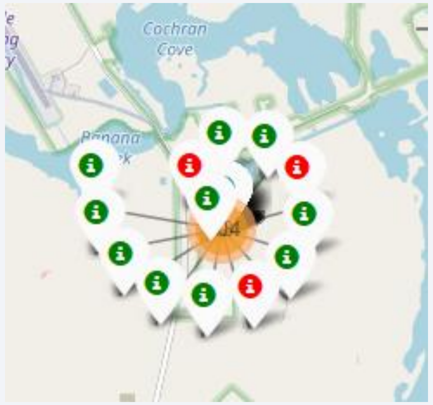


The launch sites are in USA

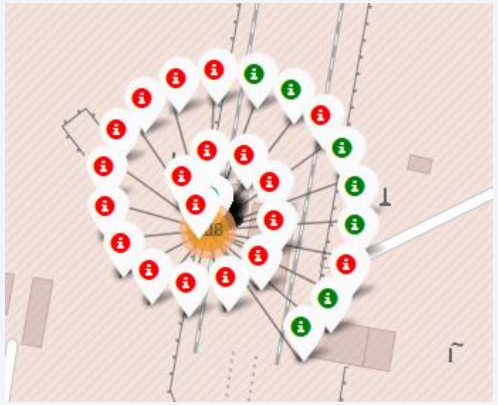
Map of Succeeded launch



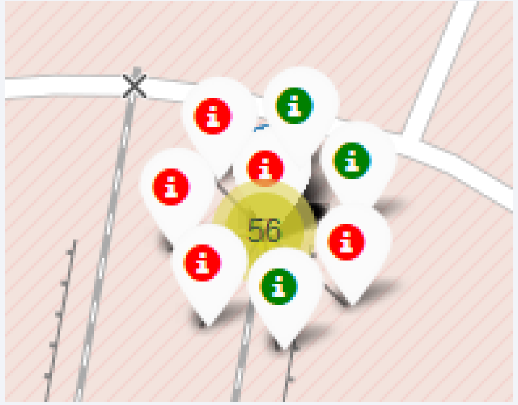
VAFB SLC-4E



KSC LC-39A



CCAFS LC-40



CCAFS SLC-40

Green are showing Succeeded launch, and red failed launch

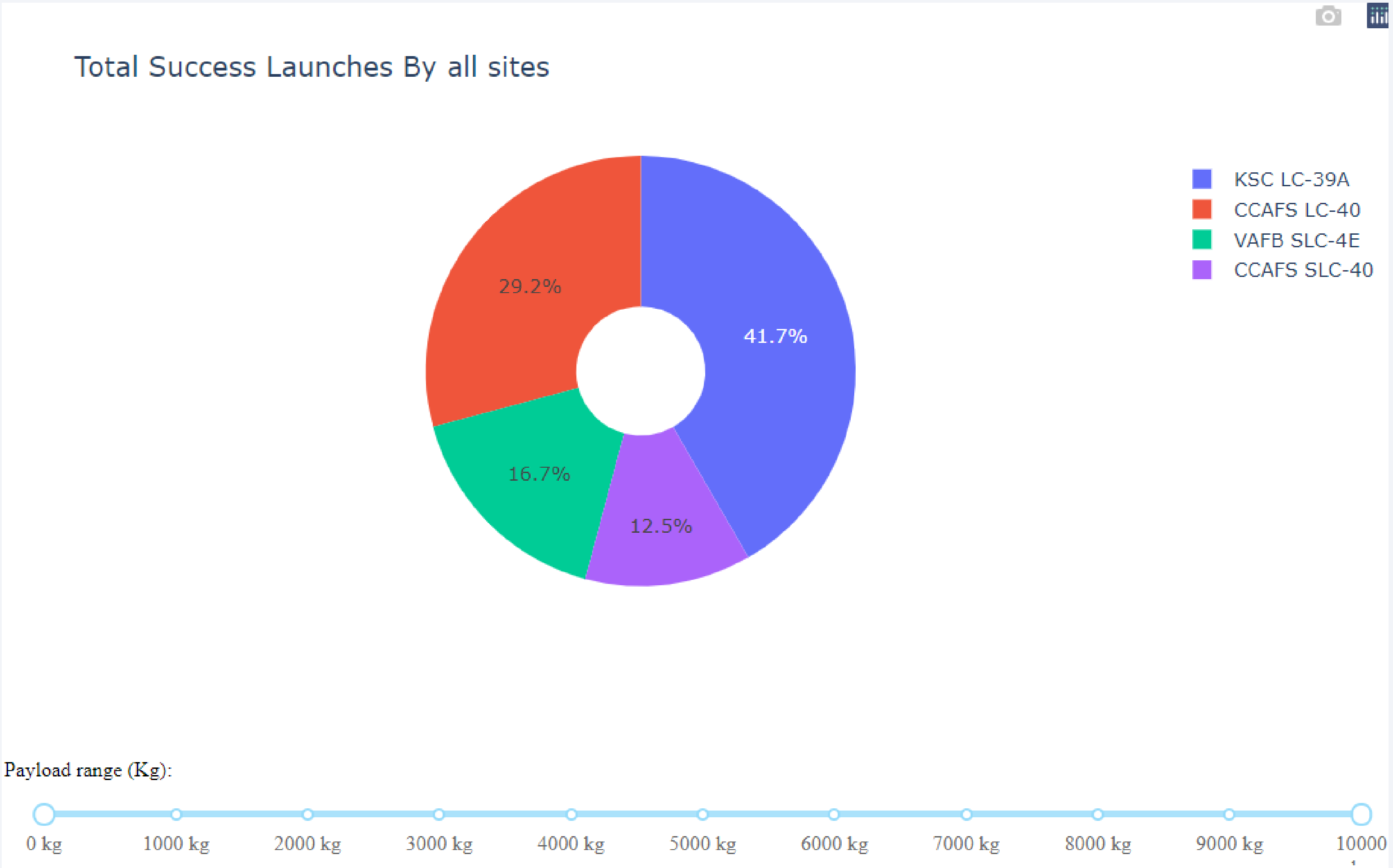
Proximity map

Creating line to measure distance between railroad, coast highway

Section 5

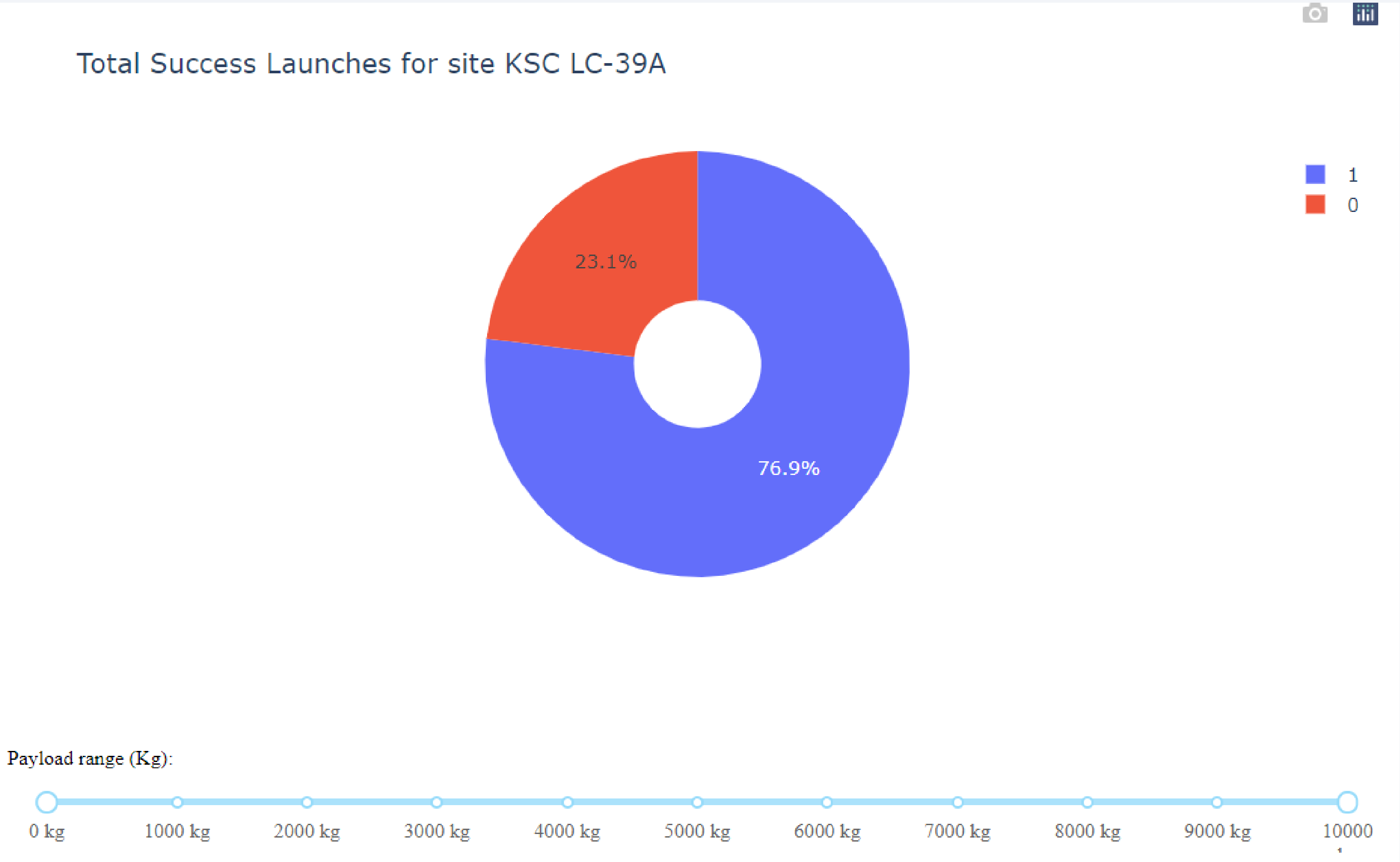
Build a Dashboard with Plotly Dash

Total Success Launches



This graph show that KSC LC 39A has the highest success rate among other sites

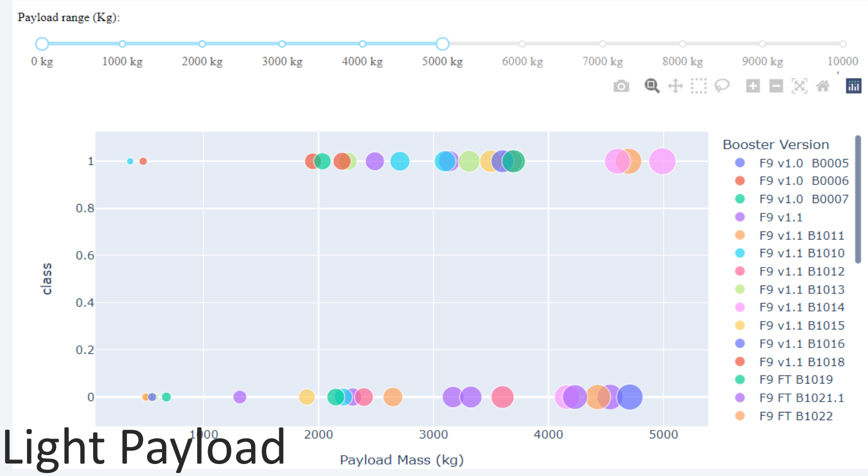
Success Rate for launch site KSC LC 39A



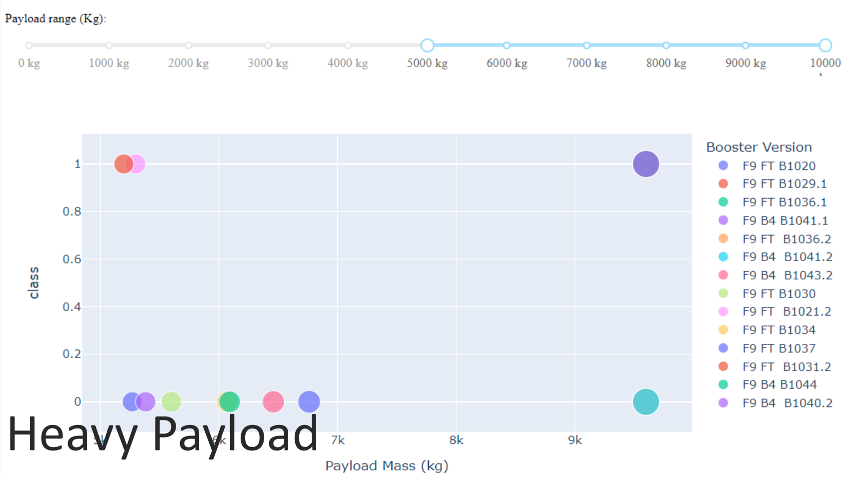
The success rate of site KSC LC 39A is 76.9%

Succesfull launch with different payload

All Payload



Light Payload



Heavy Payload

Section 6

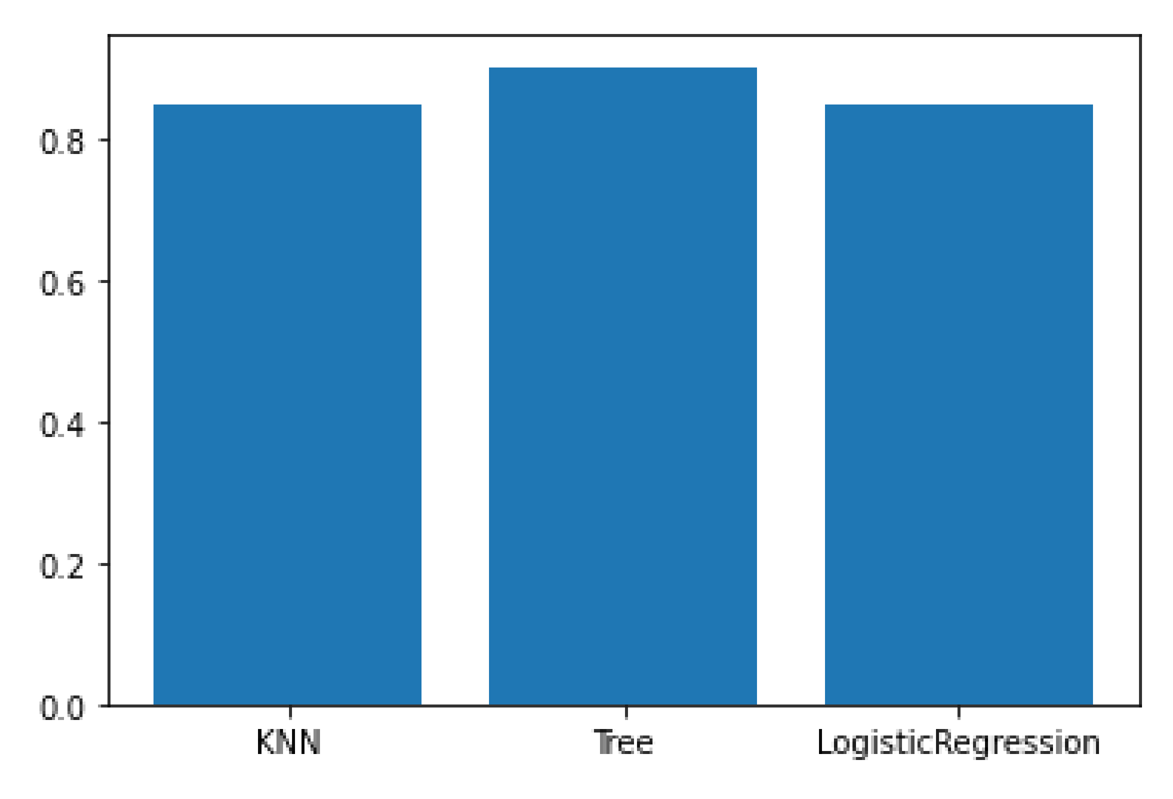
Predictive Analysis (Classification)

Classification Accuracy

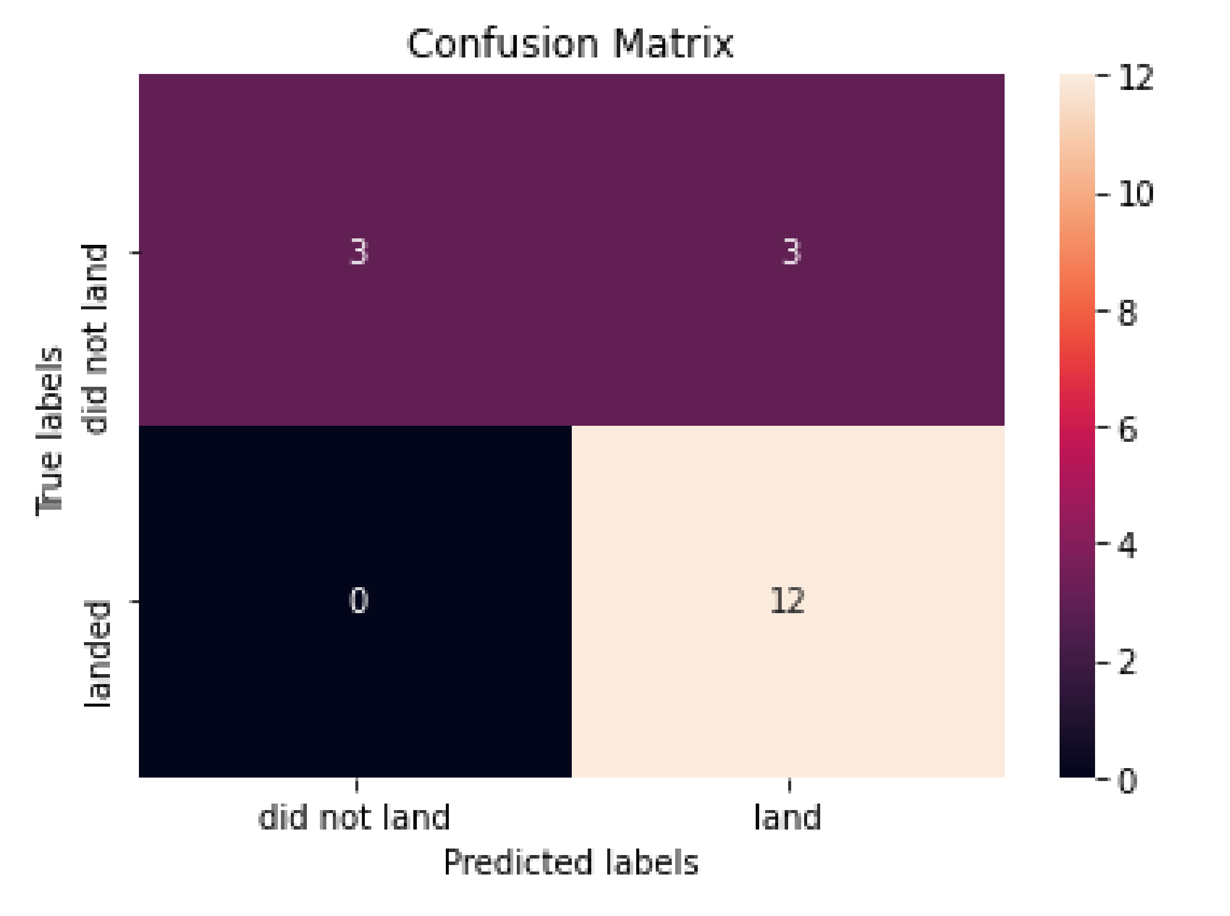
Using code below and the graph, we can say that **Tree method** has the highest accuracy

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRe
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgo
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.9
Best Params is : {'criterion': 'gini', 'max_depth': 10, 'max_features': 'sqr
t', 'min_samples_leaf': 4, 'min_samples_split': 5, 'splitter': 'best'}
{'KNN': 0.8482142857142858, 'Tree': 0.9, 'LogisticRegression': 0.84642857142
85713}
```



Confusion Matrix of Tree Method



We can see the different classes here

Conclusions

- The success rates for SpaceX launches is improving in years
- Light payloads has better successful rate than the heavy payloads
- KSC LC-39A has the most successful launches sites than other
- Orbit HEO,GEO,SSO,ES-L1 has the best Success Rate
- The best for Machine Learning for this dataset is The Tree Classifier
- SpaceX has many attempt, some succesfull and failed. Most failed landing are planned.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!