DATA SCIENCE CAPSTONE PROJECT - SPACEX

- Minh Duc Le

- 05 September 2021

Outline

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

EXECUTIVE SUMMARY

Summary of methodologies

- Data collection
- Data wrangling
- Exploratory data analysis with SQL
- Exploratory data analysis with visualization
- Interactive map with Folium
- Dashboard with Plotly
- Predictive analysis

Summary of results

- Exploratory data analysis results
- Data visualization results
- Predictive analysis results

INTRODUCTION

- Background

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 solution

What influences if the rocket will land successfully ?

The effect each relationship with certain rocket variables will impact in determining the successful rate of a landing.

What conditions does Spacex have to achive to get the best results and ensure the best rocket success landing rate.

METHODOLOGY

Data collection methodology:

- SpaceX Rest API
- Web scraping from Wikipedia

Performing data wrangling (pre-processing data for Machine Learning algorithms)

- One hot Encoding data fields for Machine Learning and dropping irrelevant columns

Performing exploratory data analysis (EDA) using visualization and SQL

- Plotting: Scatter graphs, Bar graphs to show the relationships between variables to show patterns of data.

Perfoming interactive visual analytics using Folium and Plotly Dash

Performing predictive analysis using classification models

- How to buid, tuning hyperparameter and evaluate classification models

METHODOLOGY - DATA COLLECTION

We worked with the SpaceX launch dataset, which gave us launches data about: rocket used, payload delivered, launch specification, landing specifications and landing outcome.

There are 02 methods we used to gathered the dataset: REST API and Data Scraping from wiki pages.

METHODOLOGY - DATA COLLECTION

REST API

Step 1

- Endpoint: api.spacexdata.com/v4/launches/past
- Python requests module

Step 2

The HTTP response a Json file

Step 3

Using json_normalize function to to restructed json data into a flat table

Data scraping

Step 1

URL: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Step 2

Extracting data using BeautifulSoup library

Step 3

Using json_normalize function to to restructed json data into a flat table

SpaceX API

1. Getting response from API

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

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

```
response.json()
data = pd.json_normalize(response.json())
```

2. Converting Response to Json file for further work

3. Init some feature functions to handle data from the flat table

```
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

From the `launchpad` we would like to know the name of the launch site being used, the logitude, and the latitude.

```
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])
```

[URL Github to see my detailed Notebook]()

SpaceX API

4. Create a dictionary with features as keys and apply custome function to extract data

5. Create a new dataframe with features above and extracted data. Filter the "BoosterVersion" feature with condition "Falcon 1"

In [19]:
```
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}
```

[URL Github to see my detailed Notebook]()

Web Scraping

1. Getting response from HTML file and applying BeautifulSoup library to parse HTML file

```
data = requests.get(static_url).text
html = "data"
```

Create a BeautifulSoup object from the HTML response

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, 'html.parser')
```

```
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
```

2. Extracting tables using "find_all" method and column names (with features as same as API method) for our dataframe from header of table

Url Github to see my detailed Notebook

Web Scraping

3. Extracting and adding data to features by using For loop.

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict["Flight No."].append(flight_number)
            #print(flight_number)
            datatimelist=date_time(row[0])
```

[Url Github to see my detailed Notebook]()

Web Scraping

4. Creating a dictionary with features as keys and converting to DataFrame

```
headings = []
for key,values in dict(launch_dict).items():
    if key not in headings:
        headings.append(key)
    if values is None:
        del launch_dict[key]

def pad_dict_list(dict_list, padel):
    lmax = 0
    for lname in dict_list.keys():
        lmax = max(lmax, len(dict_list[lname]))
    for lname in dict_list.keys():
        ll = len(dict_list[lname])
        if  ll < lmax:
            dict_list[lname] += [padel] * (lmax - ll)
    return dict_list

pad_dict_list(launch_dict,0)

df = pd.DataFrame.from_dict(launch_dict)
df.head(10)
```

[Url Github to see my detailed Notebook]

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 will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

DATA WRANGLING - EXPLORATORY DATA ANALYSIS (EDA)

Data Analysis

Identifying the number of launches on each site

Identifying the number and occurrence of each orbit

Identifying the number and occurrence of each mission outcome per orbit type

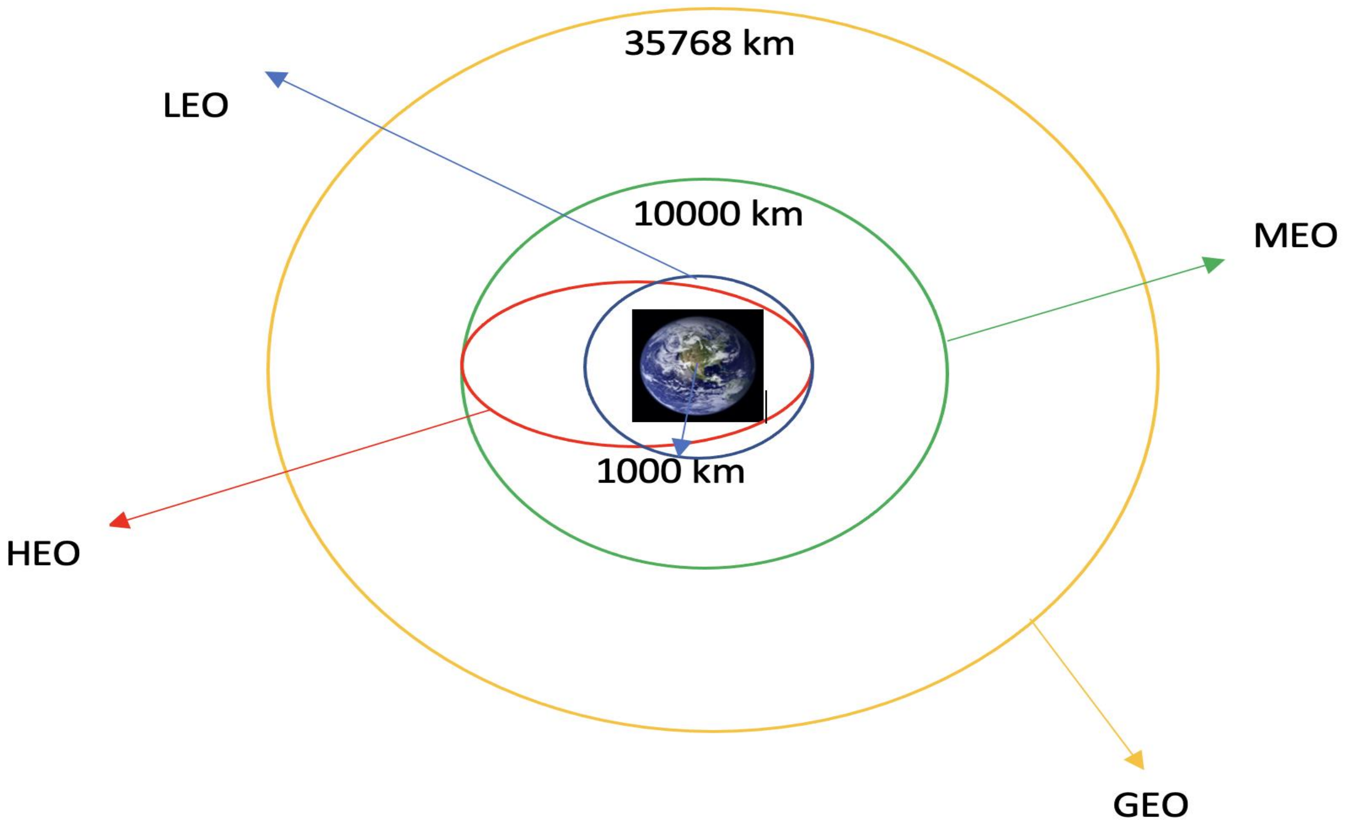
Creating a landing outcome label from Outcome column

[Url Github to see my detailed Notebook]

DATA WRANGLING-EXPLORATORY DATA ANALYSIS (EDA)

Orbits

- LEO
- VLEO
- GTO
- SSO
- ES-L1
- HEO
- ISS
- MEO
- GEO
- PO



EDA WITH SQL

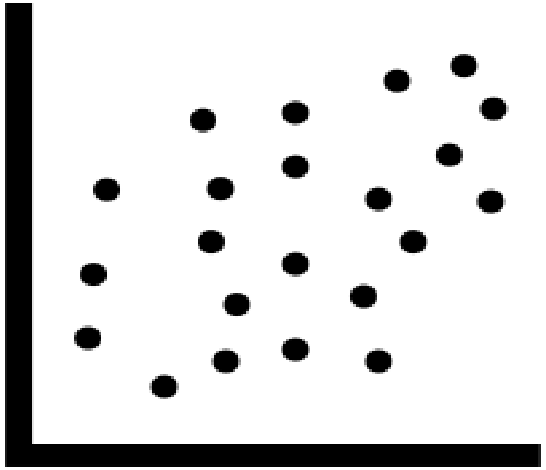
In order to gather and transform data to information, we perform several SQL commands such as:

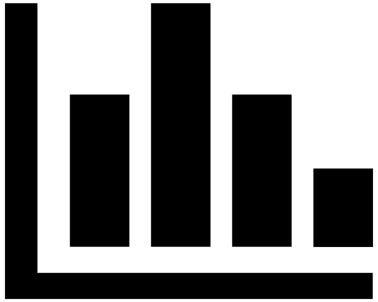
- *Display the names of the unique launch sites in the space mission*
- *Display the total payload mass carried by boosters launched by NASA (CRS)*
- *Display average payload mass carried by booster version F9 v1.1*
- *List the date when the first successful landing outcome in ground pad was achieved.*
- *List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000*
- *List the total number of successful and failure mission outcomes*
- *List the names of the booster_versions which have carried the maximum payload mass. Use a subquery*
- *List the failed landing_outcomes in drone ship, their booster versions, and launch site names for the in year 2015*
- *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*

[Url Github to see my detailed Notebook]()

EDA WITH VISUALIZATION

- Flight Number and Payload Mass
- Flight Number and Launch Site
- Payload and Launch Site
- Orbit type and Flight Number
- Orbit type and Payload





- Orbit type and each Success rate

- Success rate and Yearly trend

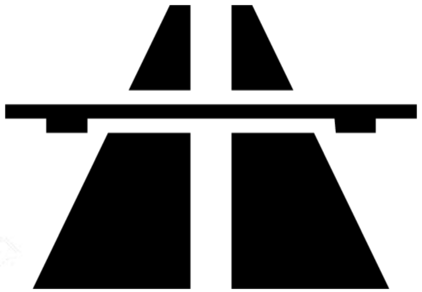
[Url Github to see my detailed Notebook]

INTERACTIVE VISUAL ANALYTICS AND DASHBOARD

- We use Folium library to visualize launch site location
- The goal is to find distance between Launch site locations and:

Railways

Highways



Coastline

City

[Url Github to see my detailed Notebook]

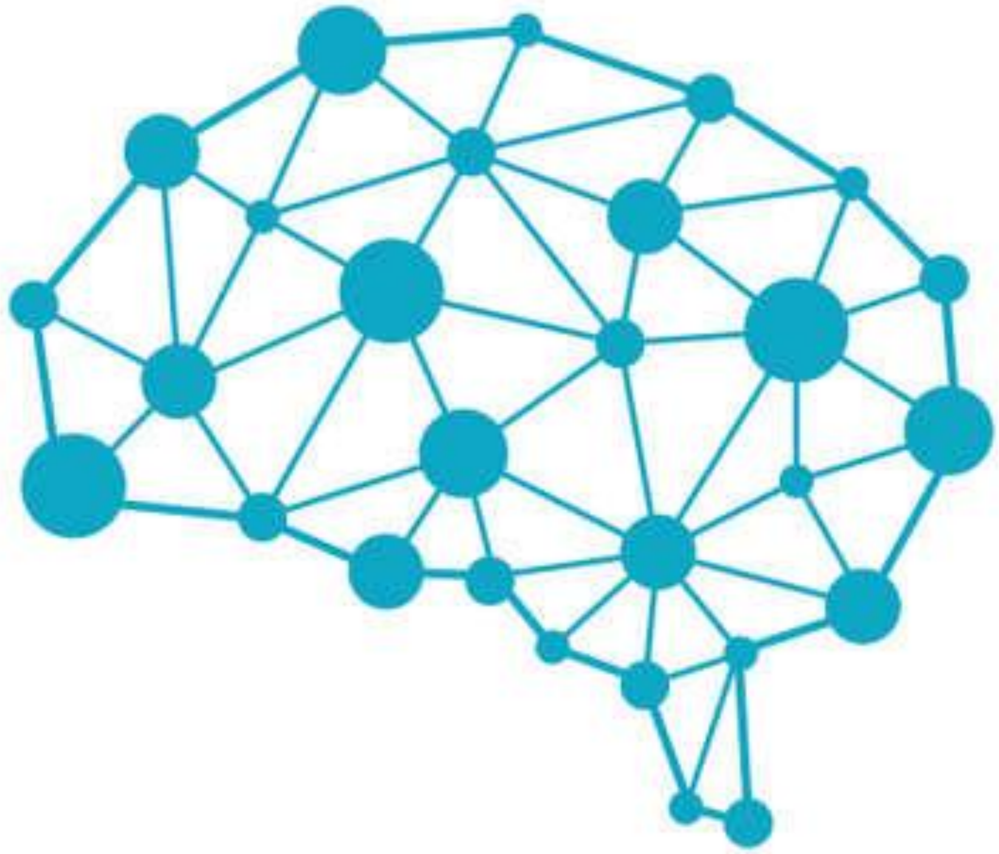
INTERACTIVE DASHBOARD

- The dashboard is built with Plotly library
- The dash combines 2 charts:

To visualize the proportion between launch sites and its success rate

To visualize the Outcome and Payload Mass (kg) for different Booster Version

PREDICTIVE ANALYSIS (CLASSIFICATION)



METHODOLOGY

MODEL DEVELOPMENT

- Normalizing the data by StandardScaler from Scikit-Learn
- Splitting data into train set and test set by train_test_split function
- Developing hypothesis classification models
- Finding the hyperparameter using GridSearchCV
- Fit data to complete models

MODEL EVALUATION

- Applying the models on test set
- Calculating R^2 by using Score method
- Confusion Matrix

THE BEST PERFORMING MODEL

- Comparing (R^2) and Confusion Matrix between models
- The best accuracy model is the final model

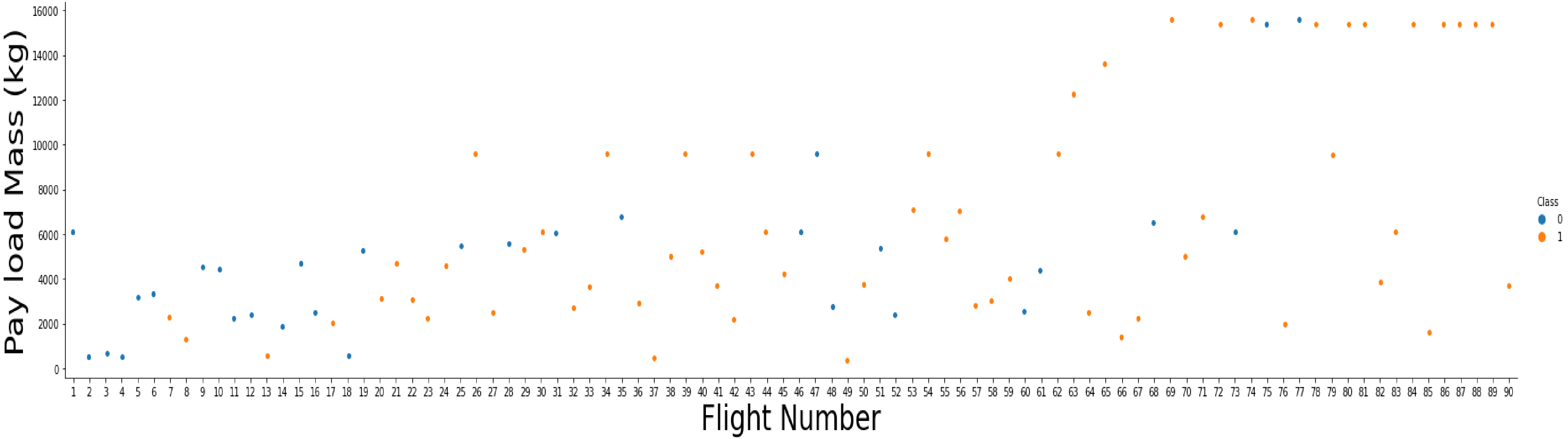
Url Github to see my detailed Notebook

RESULTS

- EXPLORATORY DATA ANALYSIS (EDA)
- INTERACTIVE VISUAL ANALYSIS
- PREDICTIVE ANALYSIS

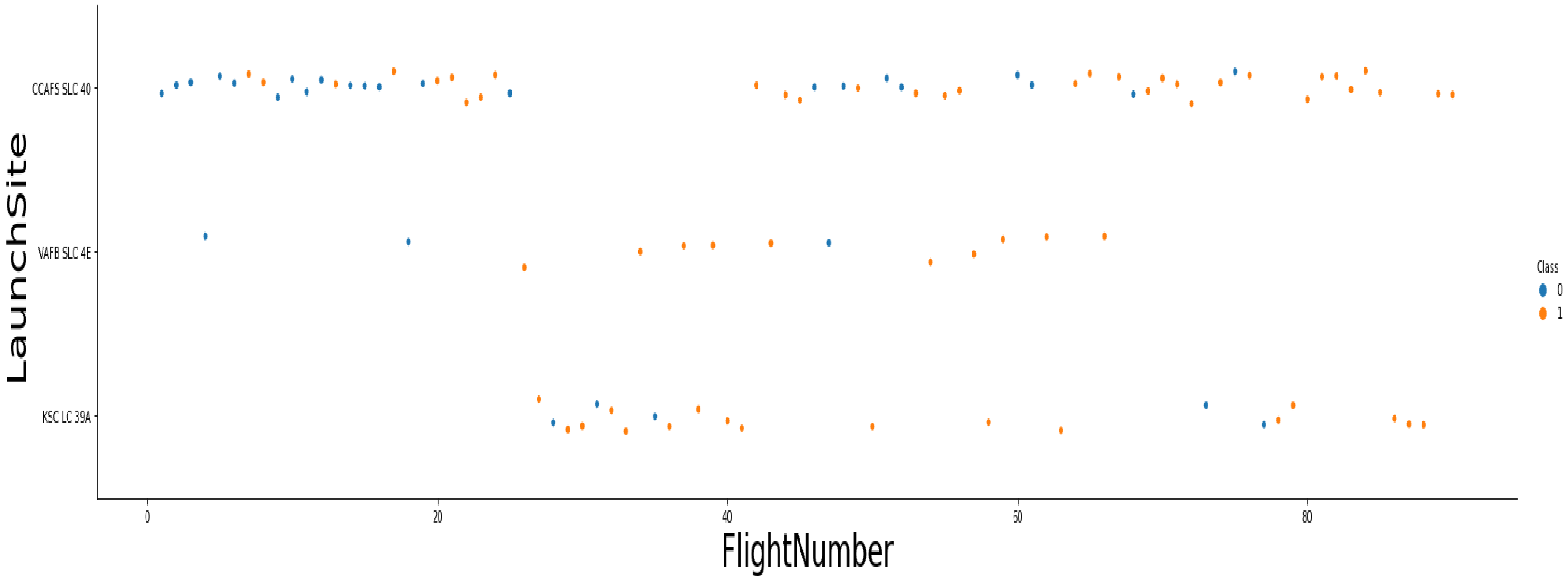
EXPLORATORY DATA ANALYSIS

THE FLIGHT NUMBER VS PAYLOAD MASS (KG)



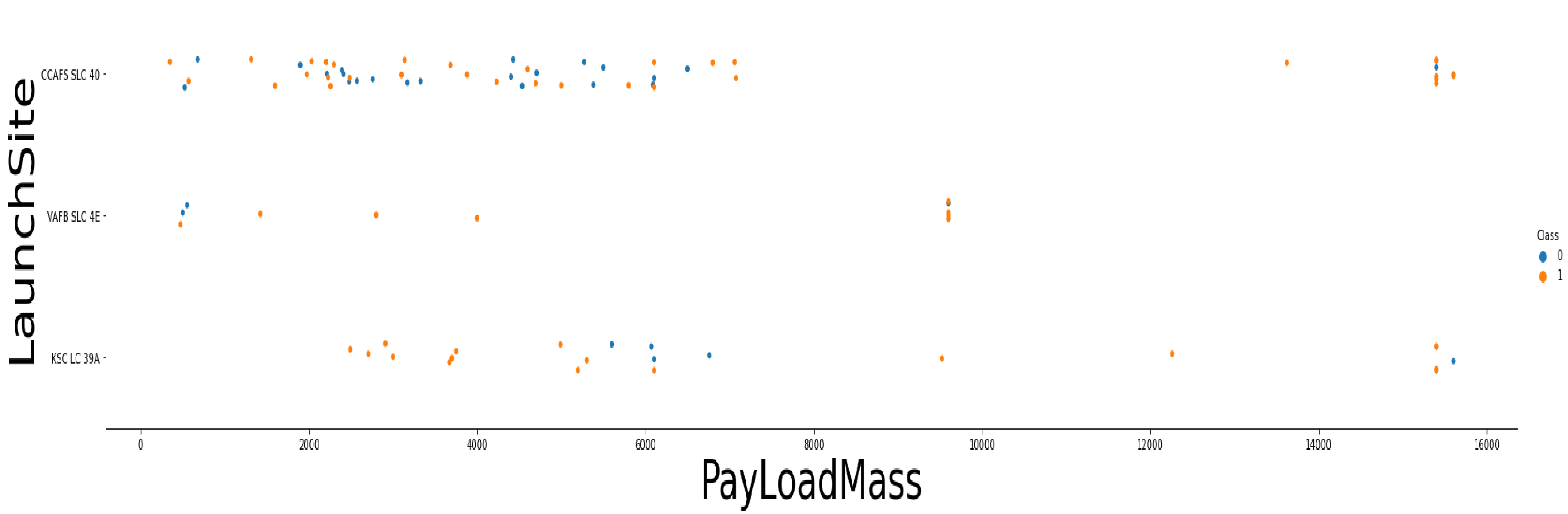
We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important, it seems the more massive the payload, the less likely the first stage will return.

Launch sites vs FlightNumber



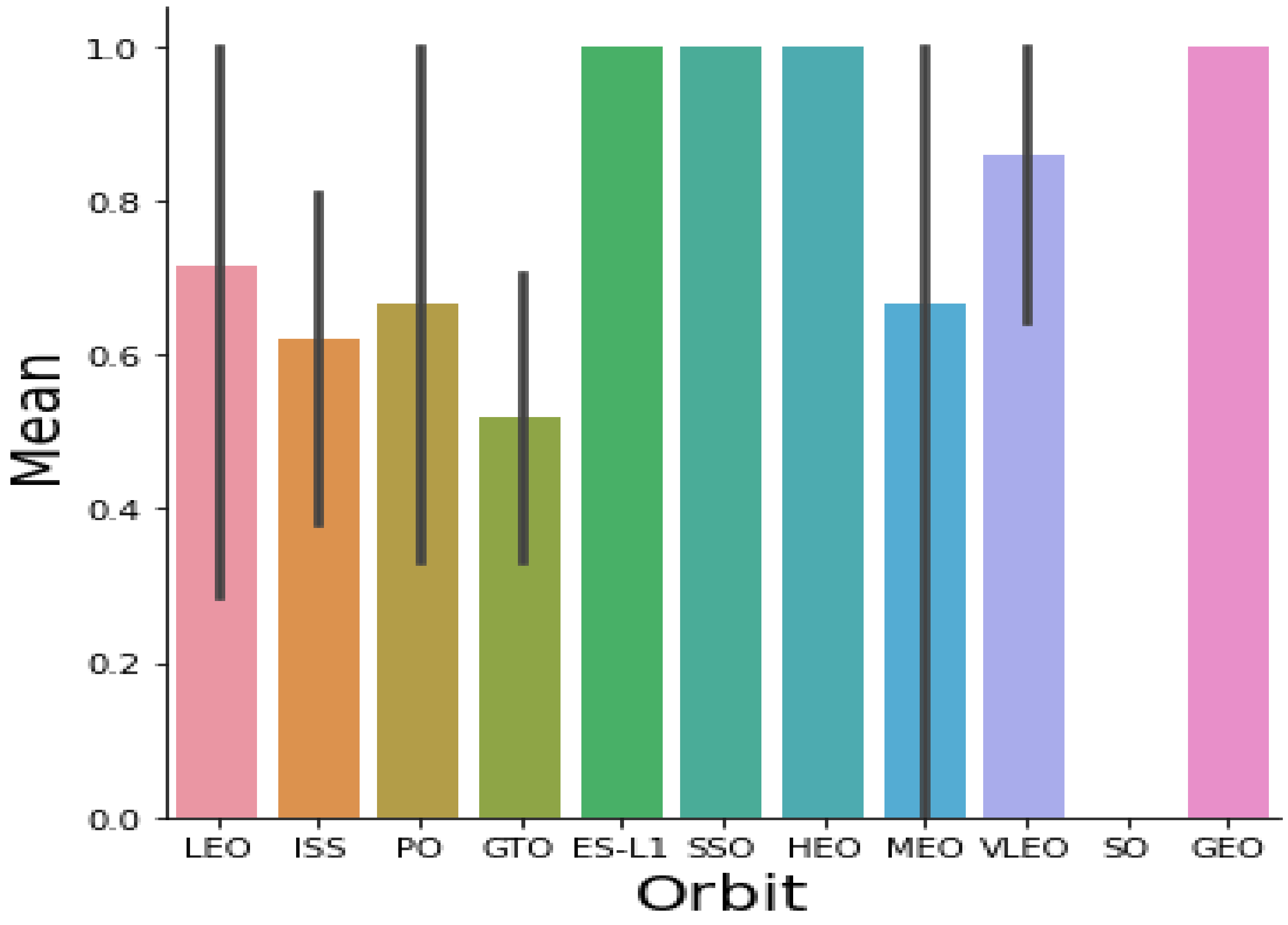
We see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%. Overall, the more number of flight, the greater success rate.

The Payload Mass vs Launchsite



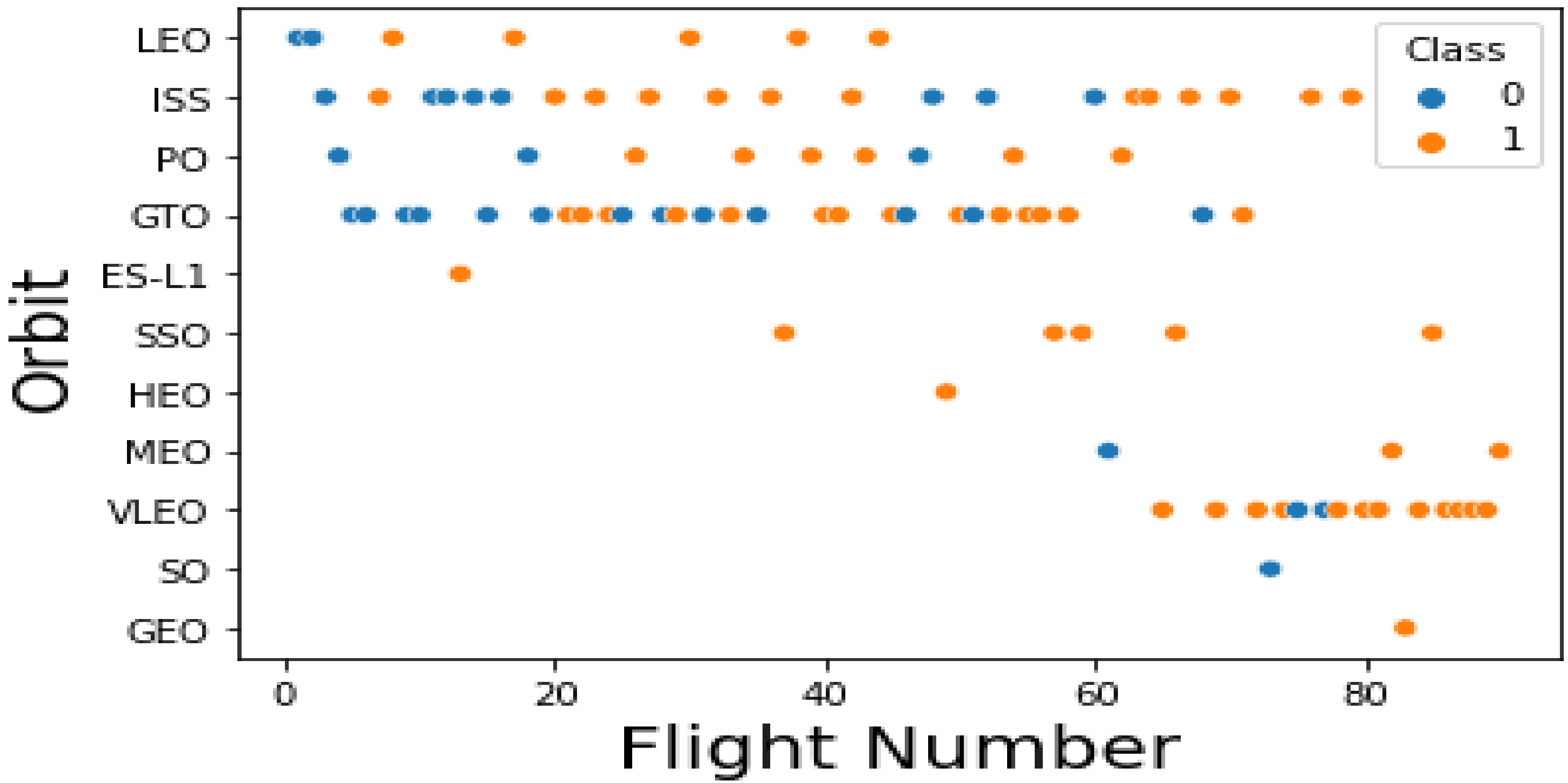
As we can see, the higher Payload Mass (kg), the higher success rate for launch site CCAFS SLC 40. However, this pattern is not quite clear for others launch sites. Therefore, it is hard to say the Payload Mass (kg) has a relationship with Launch sites.

SUCCESS RATE VS ORBIT TYPE



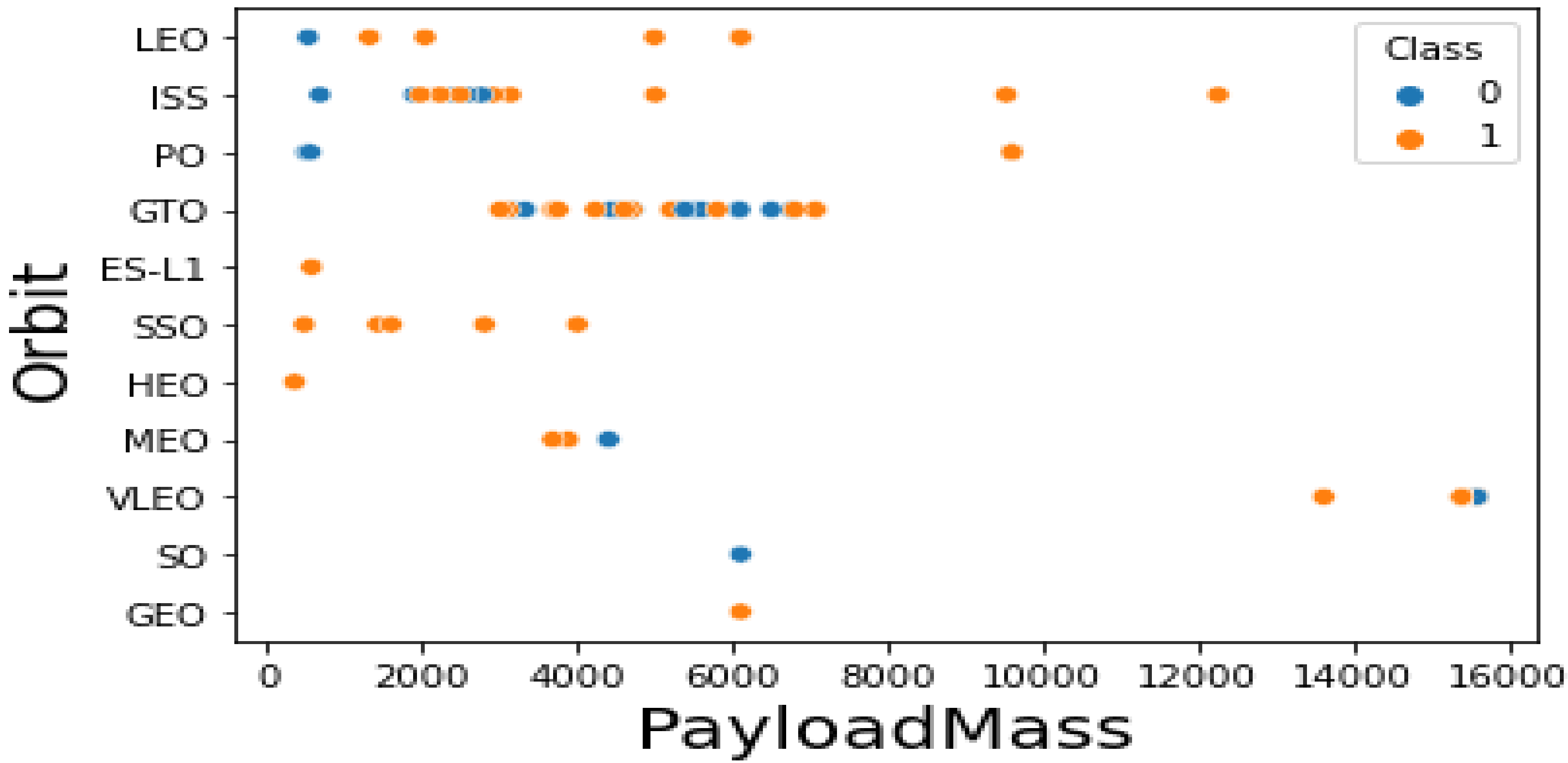
The orbit of ES-L1, SSO, HEO, GEO have the best success rates.

FLIGHT NUMBER VS ORBIT TYPE



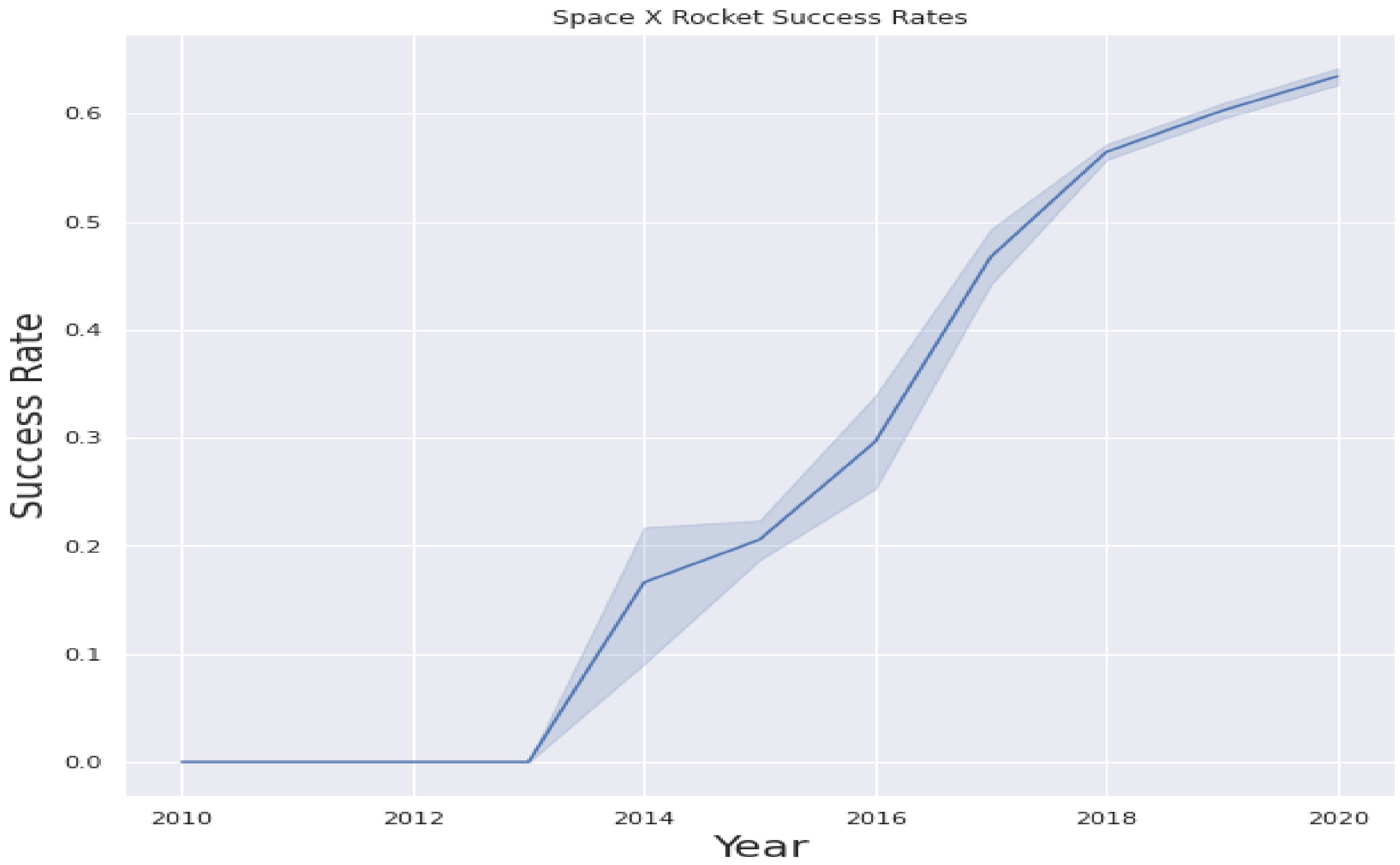
The Success in the Leo orbit appears related to the number of flights, on the other hand, there seems to be no relationship between flight number when in GTO orbit.

ORBIT TYPE VS PAYLOAD MASS (KG)



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

SUCCESS RATE VS YEAR



We can observe that the sucess rate since 2013 kept increasing till 2020

EDA WITH SQL

Display the names of the unique launch sites in the space mission

SQL QUERY

```
select distinct LAUNCH_SITE from
SPACEXTBL
```

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

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

SQL QUERY

select sum (PAYLOAD_MASS__KG_) TotalPayloadMass from SPACEXTBL where CUSTOMER = 'NASA (CRS)'

Out[10]:

totalpayloadmass
45596

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

SQL QUERY

select avg(PAYLOAD_MASS__KG_) AveragePayloadMass from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'

Out[11]:

averagepayloadmass
2928

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

SQL QUERY

select min(DATE) from SPACEXTBL where LANDING__OUTCOME like 'Success (ground pac

Out[12]:

1
2015-12-22

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

SQL QUERY

```
select distinct BOOSTER_VERSION
from SPACEXTBL where
LANDING__OUTCOME = 'Success
(drone ship)' and 4000 <
PAYLOAD_MASS__KG_ < 6000
```

Out[13]:

booster_version
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1
F9 FT B1029.2
F9 FT B1021.1
F9 FT B1023.1
F9 FT B1038.1

List the total number of successful and failure mission outcomes

SQL QUERY

select
MISSION_OUTCOME,
count(MiSSION_OUTCOME
) as COUNT from
SPACEXTBL group by
MISSION_OUTCOME

Out[11]:

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

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

SQL QUERY

select distinct BOOSTER_VERSION, PAYLOAD_MASS__KG_ from SPACEXTBL where (select max(PAYLOAD_MASS__KG_) from SPACEXTBL) order by PAYLOAD_MASS__KG_ DESC

Out[25]:

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

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

SQL QUERY

select DATE, LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE from SPACEXTBL where LANDING__OUTCOME = 'Failure (drone ship)' and year(DATE) = 2015

Out[15]:

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

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

SQL QUERY

select LANDING__OUTCOME, count(LANDING__OUTCOME) as COUNT from SPACEXTBL where DATE between '2010-06-04' and '2017-03-20' group by LANDING__OUTCOME order by COUNT desc

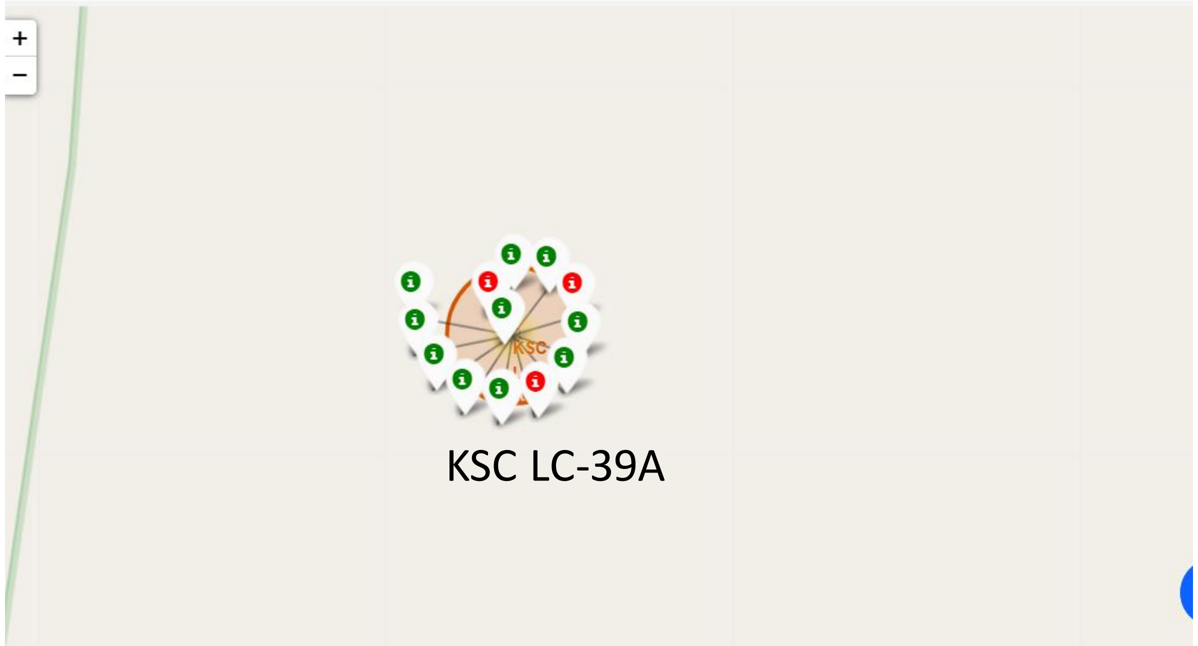
Out[16]:

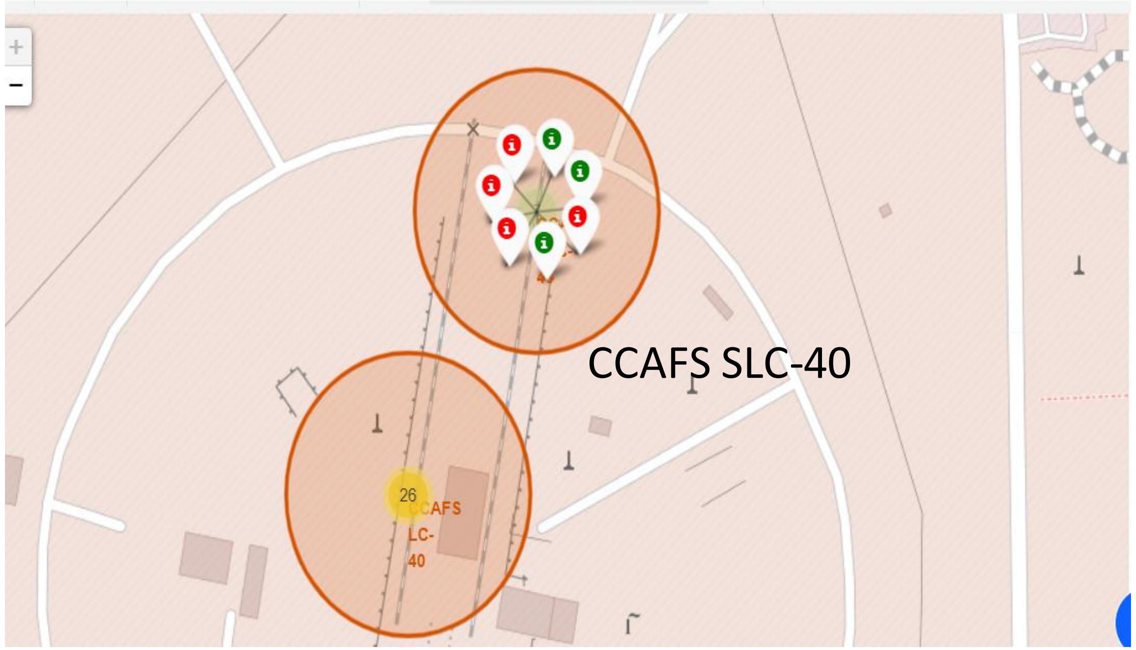
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

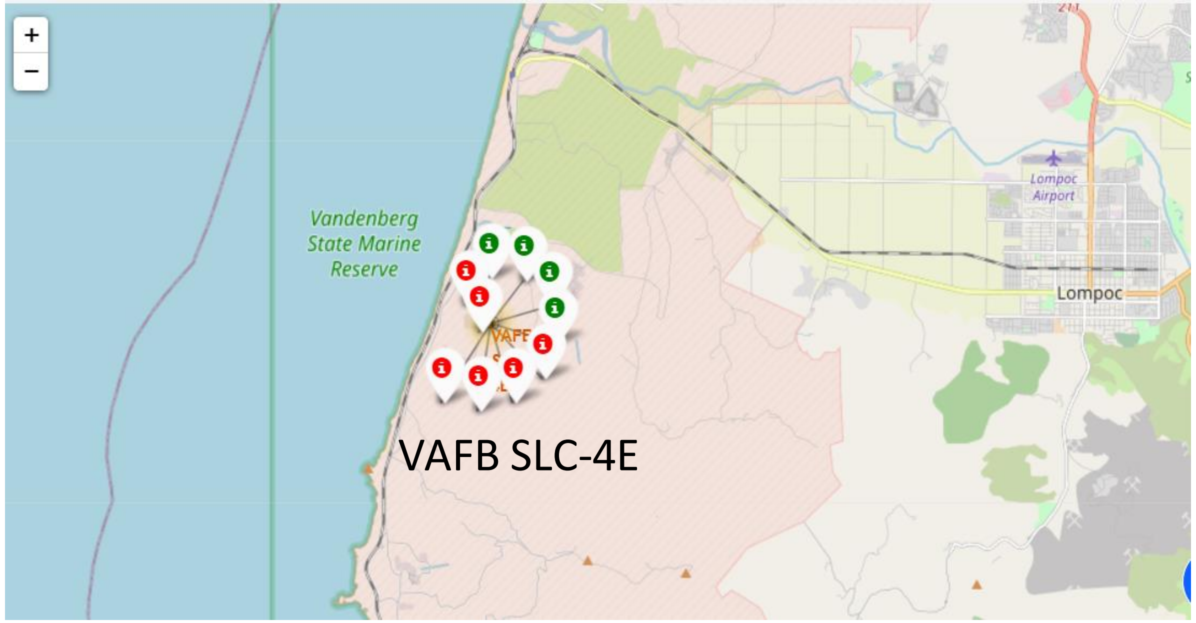
INTERACTIVE MAP

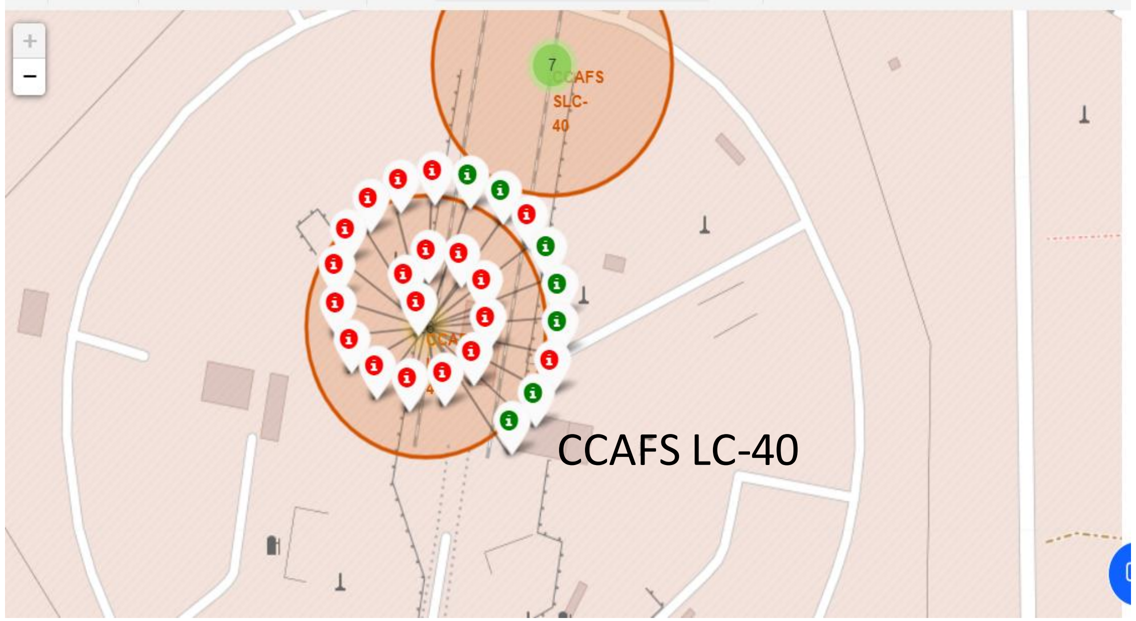
LAUNCH SITES LOCATION ON GLOBAL MAP

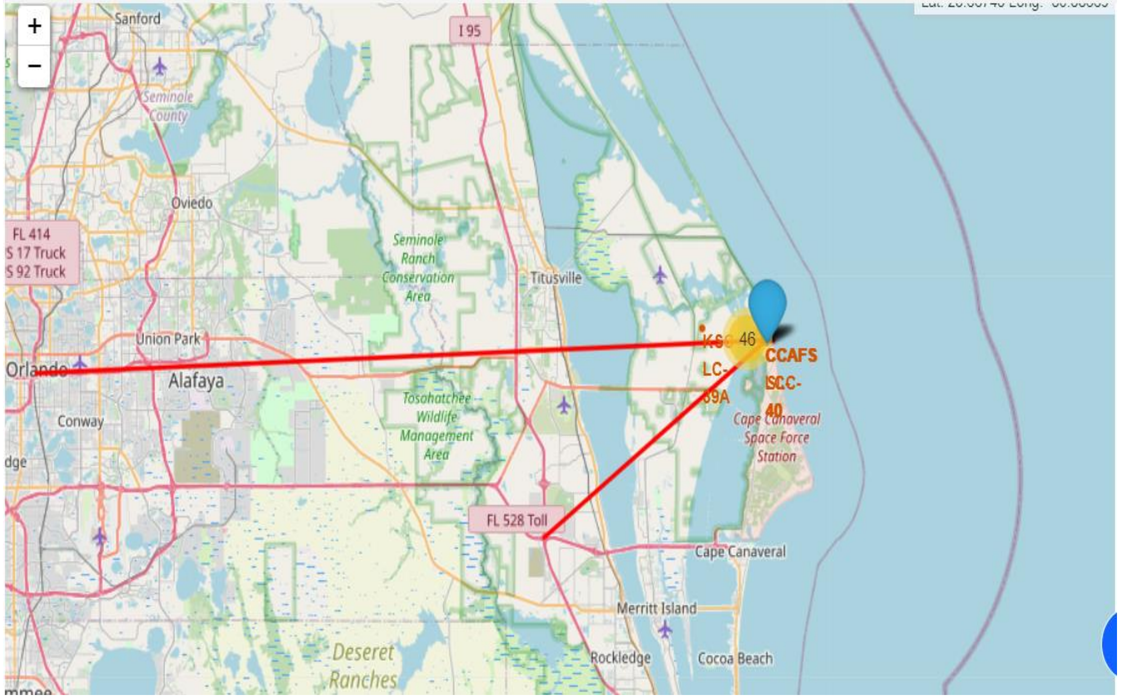
As we can see, all the launch sites are in very close promixity to the coast. Next, we will take a look at launch site one by one and its success/failure. Remember that the success is colored by green, and red for the failure.

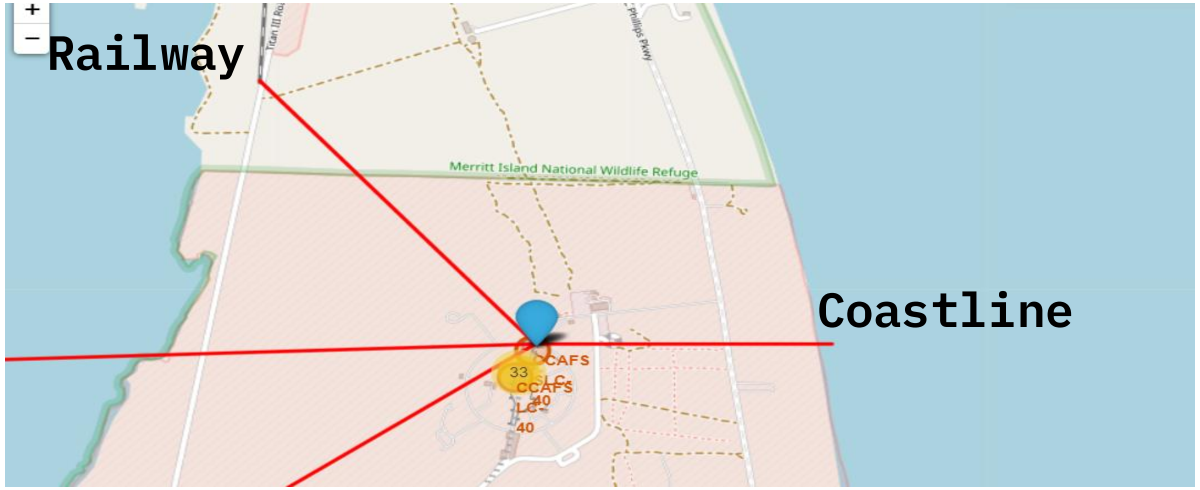

KSC
KSC LC-39A

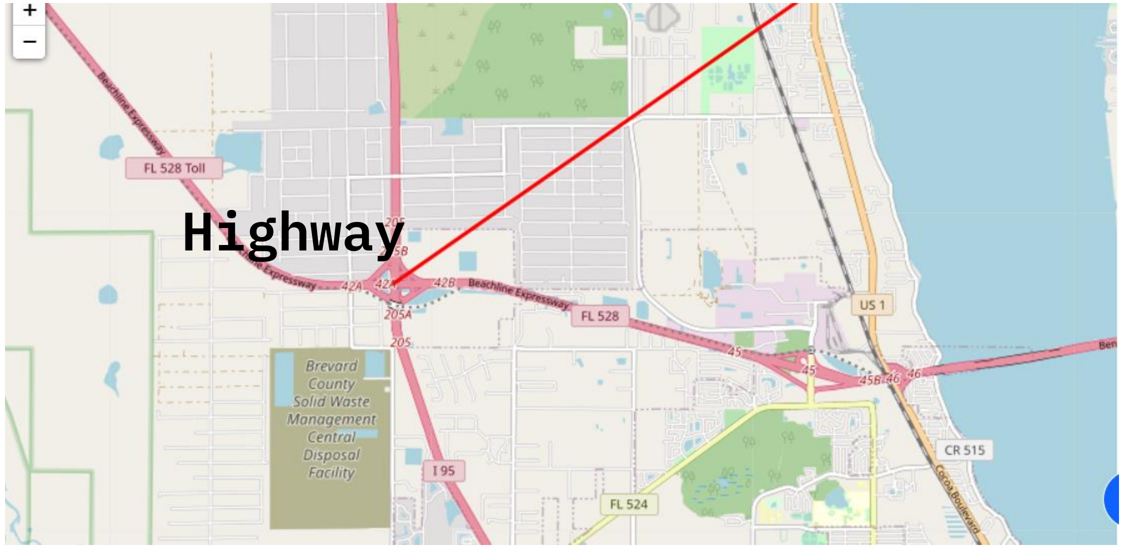

CCAFS SLC-40
26


Vandenberg
State Marine
Reserve
Lompoc
Airport
Lompoc
VAFB SLC-4E


7
CCAFS
SLC-
40
CCAFS LC-40

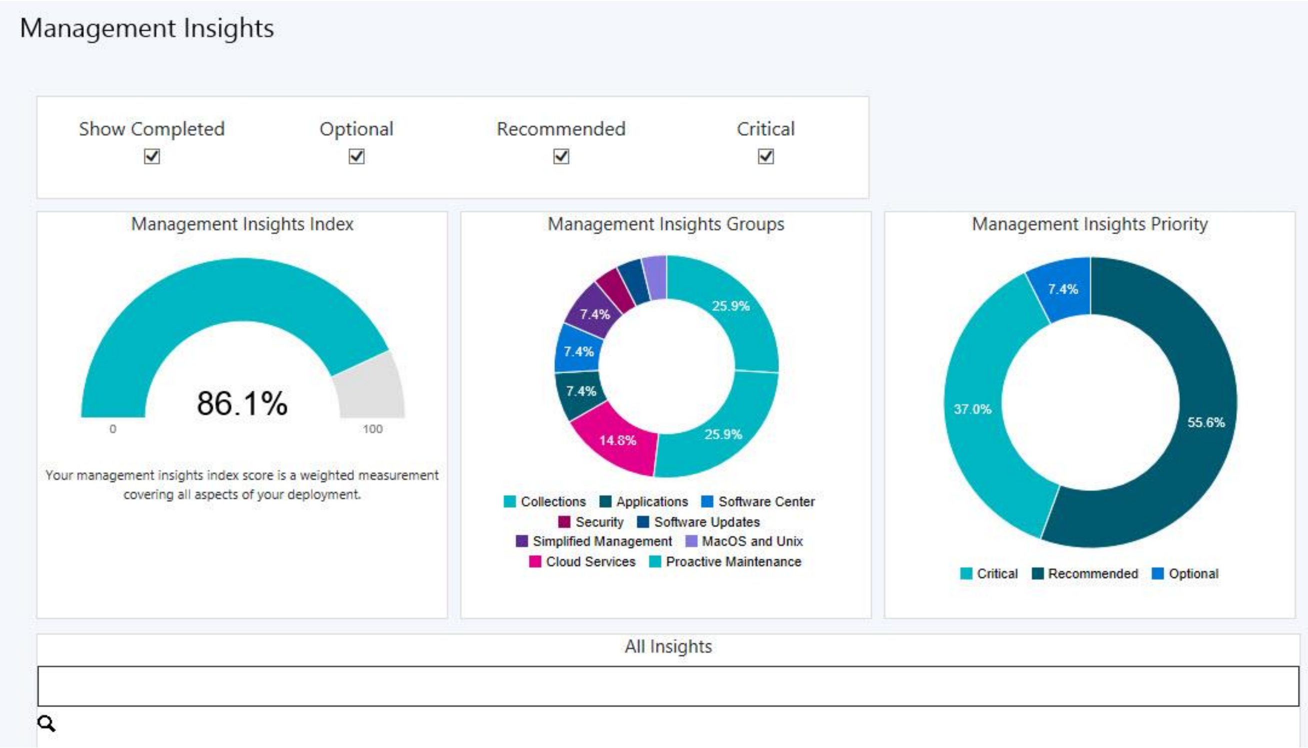




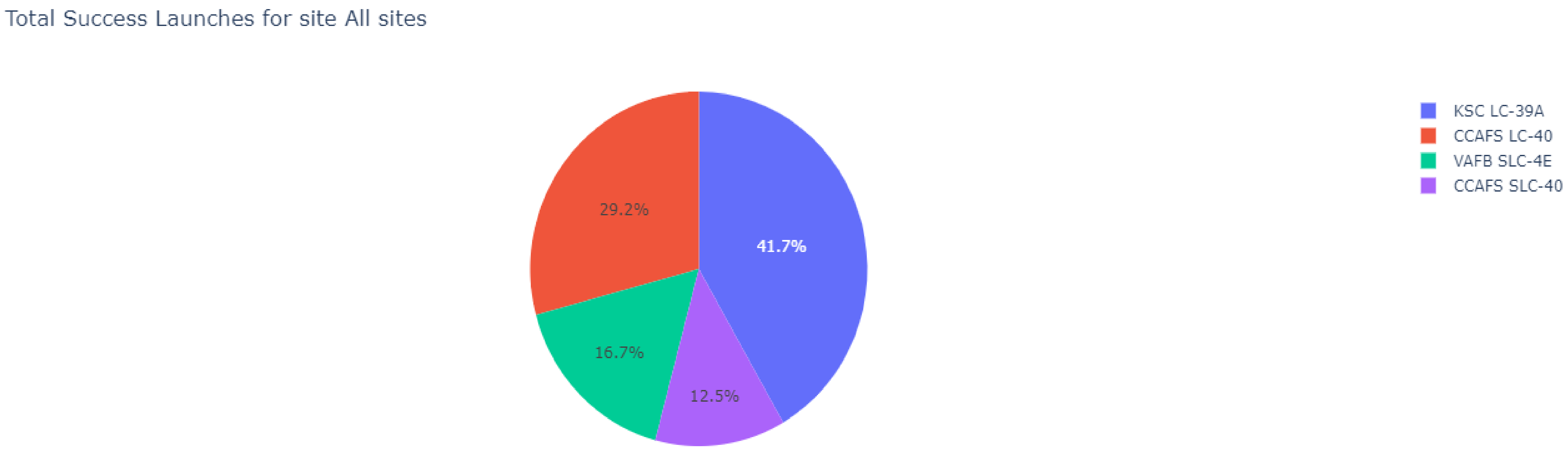


The launch sites are near to railway and coastline. However, they are far from cities and highways.

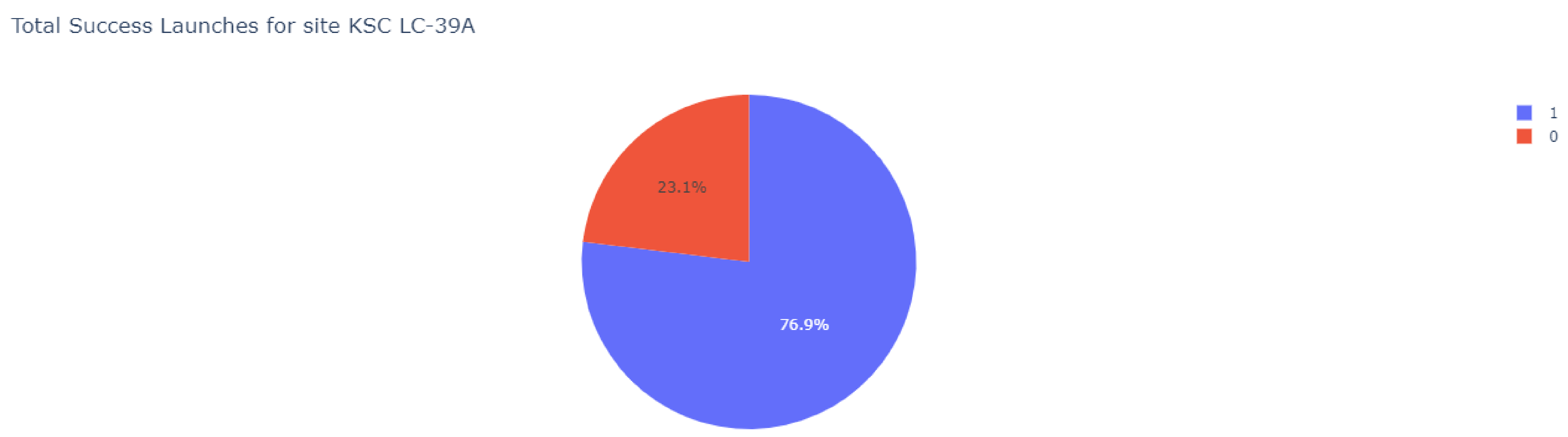
DASHBOARD



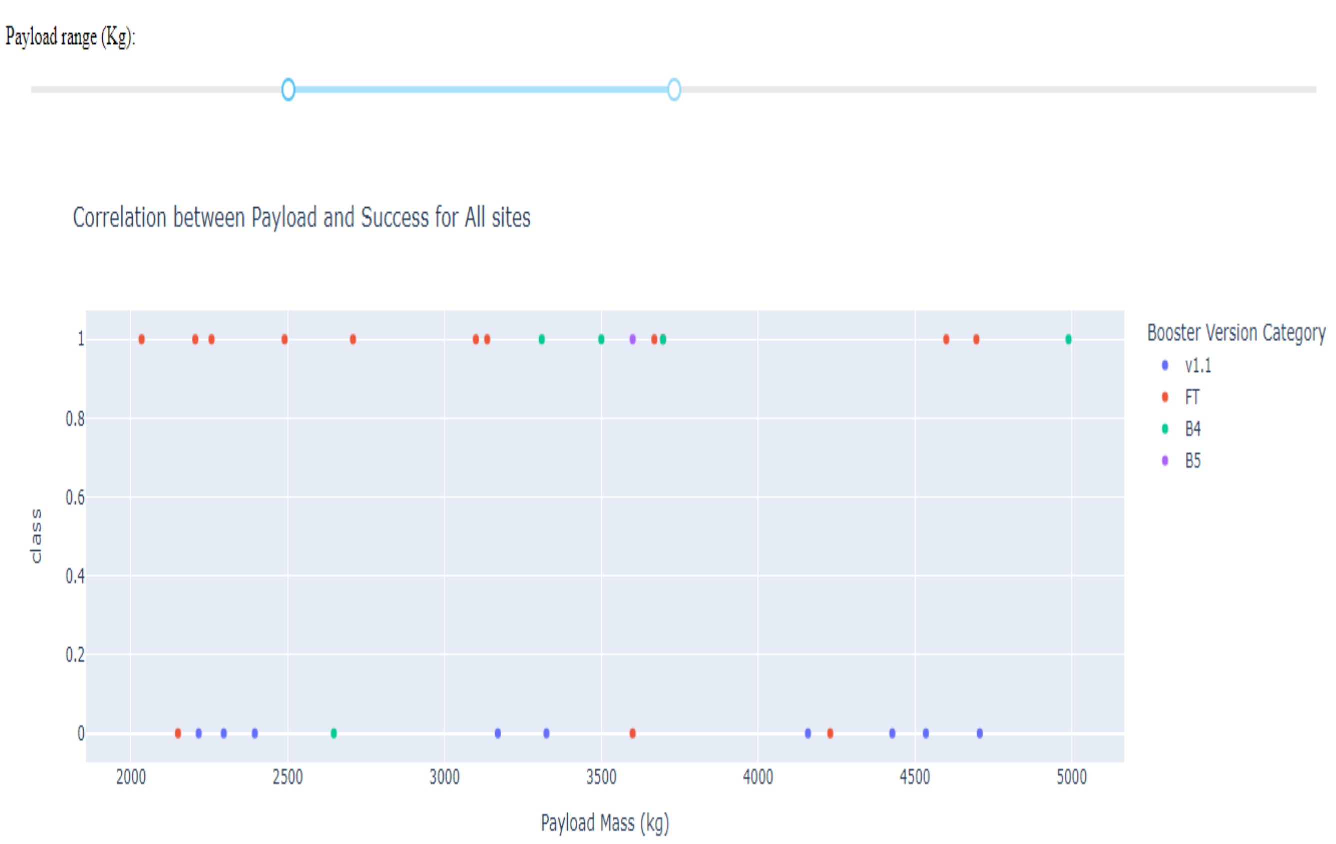
Management Insights
Show Completed
Optional
Recommended
Critical
Management Insights Index
86.1%
0
100
Your management insights index score is a weighted measurement covering all aspects of your deployment.
Management Insights Groups
25.9%
7.4%
7.4%
7.4%
25.9%
14.8%
Collections
Applications
Software Center
Security
Software Updates
Simplified Management
MacOS and Unix
Cloud Services
Proactive Maintenance
Management Insights Priority
7.4%
37.0%
55.6%
Critical
Recommended
Optional
All Insights

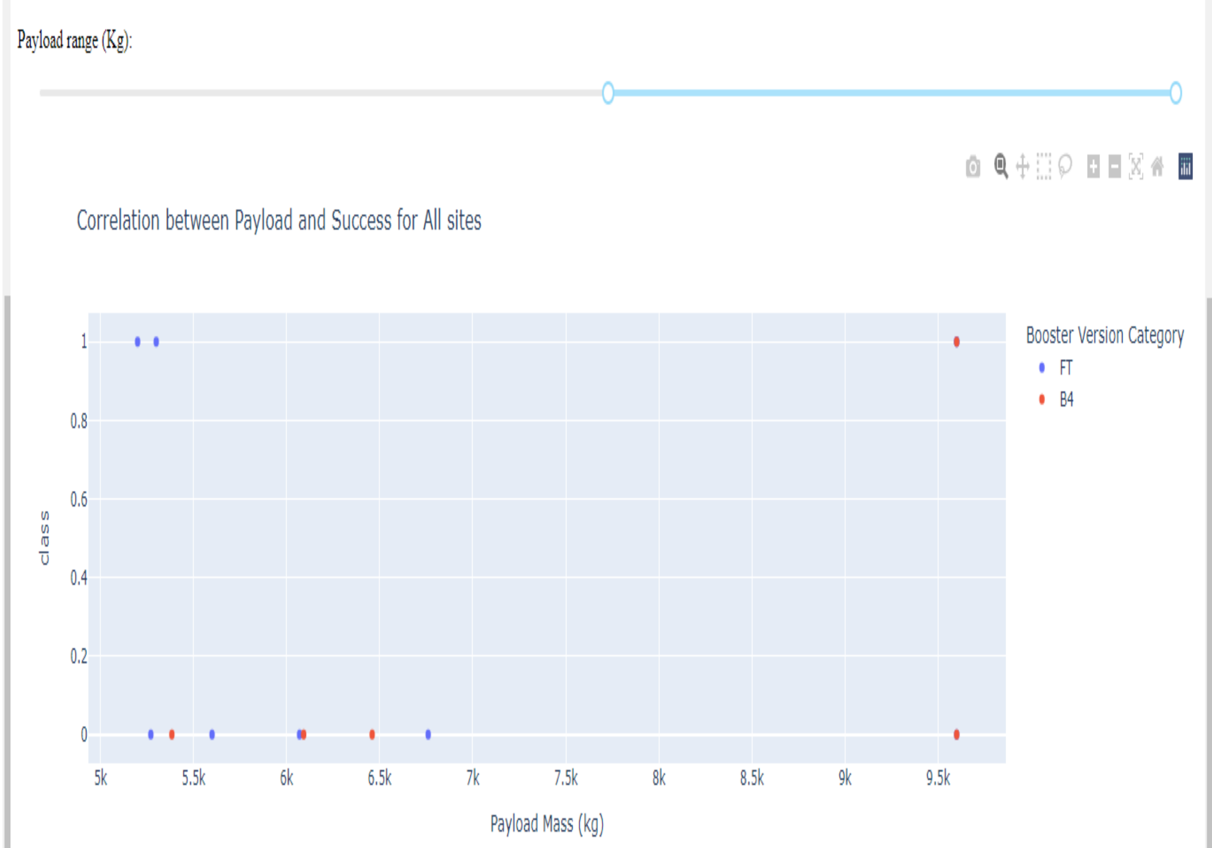


The launch site KSC LC-39A has most of successful launches



The KSC LC-39A has 76.9% success and 23.1% failure



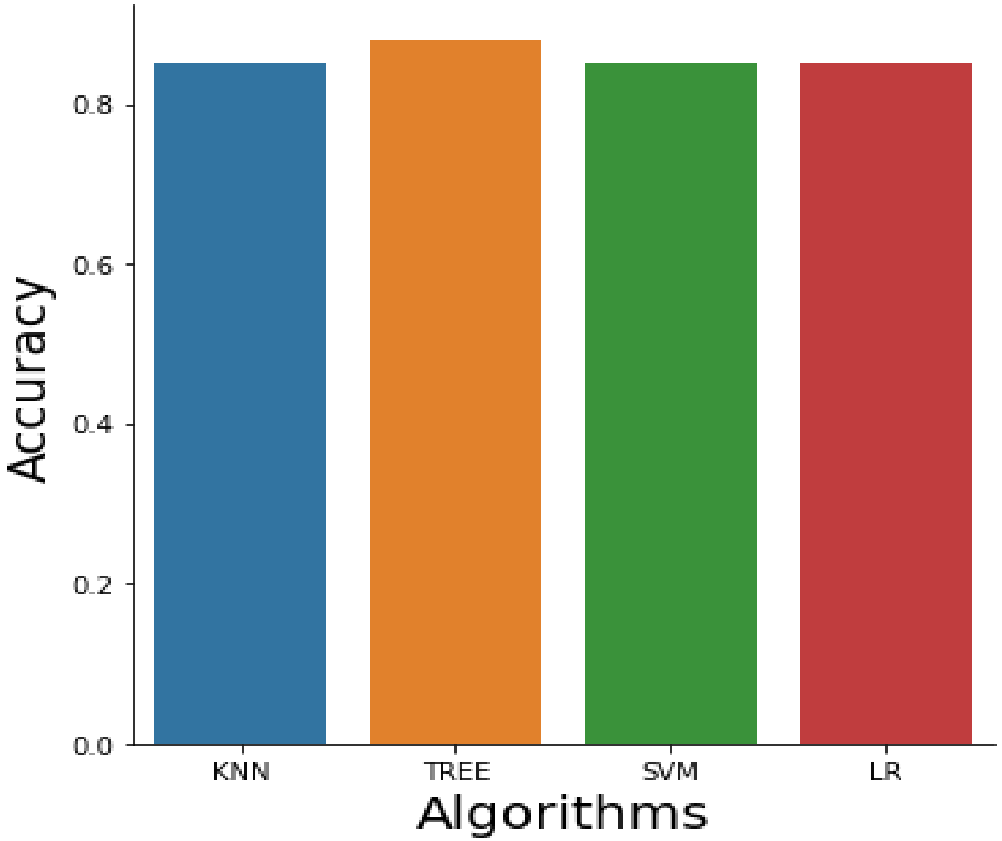


The heavier Payload Mass, the worse success rate. We can see all launch sites perform very well with Payload Mass in range of 2000-5000 kg.

PREDICTIVE ANALYSIS (CLASSIFICATION)

COMPARING ALGORITHM' S ACCURACY

- Using validation set, here we have the accuracy of K-Nearest Neighbor, Support Vector Machine and Logistic Regression are extremely close to 85%.
- The Decision Tree win the best performance with 88% of accuracy.
- The Decision Tree 's R square after using test set is 83%.

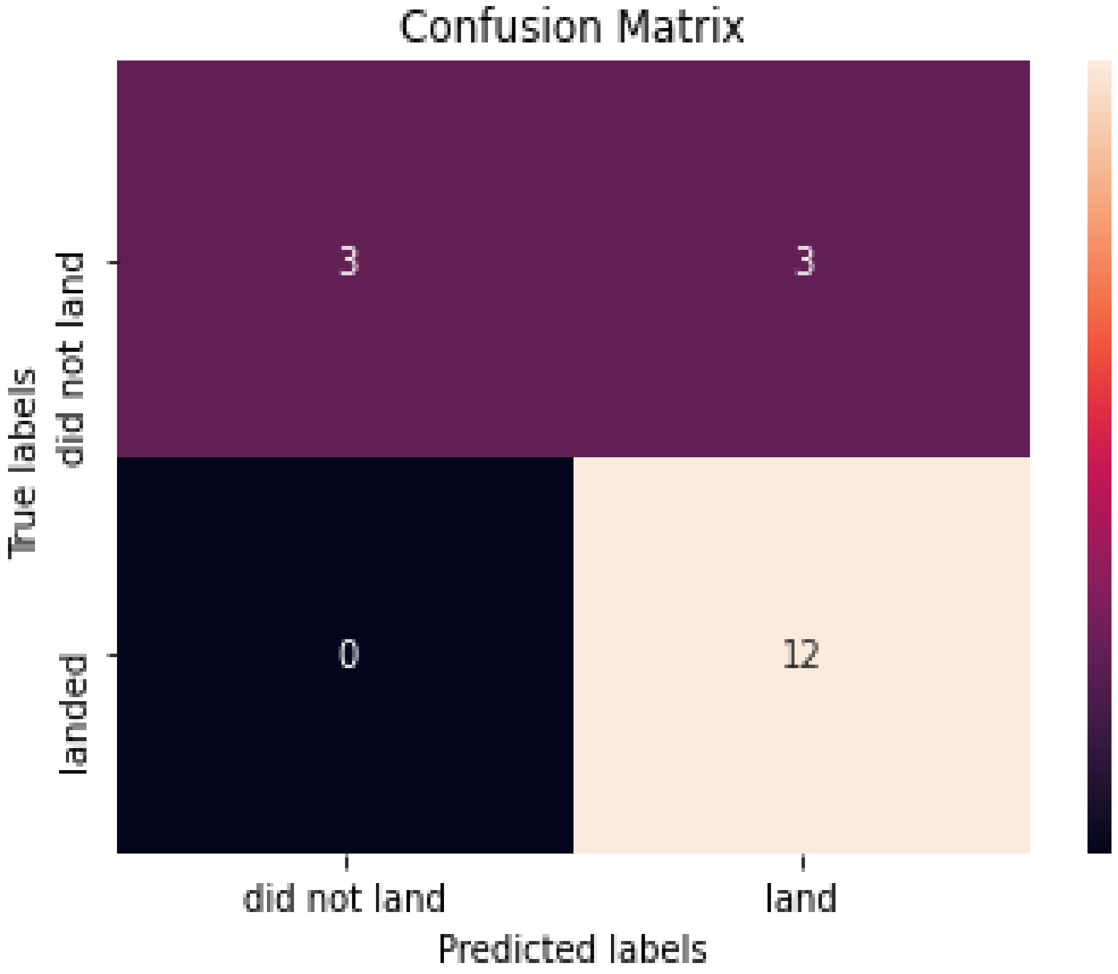



```
Entrée [111]: print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
              print("accuracy :",tree_cv.best_score_)

tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 18, 'max_features': 'sqrt', 'min_sampl
es_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
accuracy : 0.875
```

CONFUSION MATRIX

		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)



We see that Tree Decision can distinguish between the different classes. The major problem is false positives.

CONCLUSION


Morris Charts
Line Chart
Area Chart
Bar Chart
Donut Chart
25%
Donut Chart
Sparkline Charts
Line Chart
Bar Chart
Pie Chart
Easy Pie Charts
25%
50%
75%
100%
Sales
Orders

FINDINGS

The orbit of ES-L1, GEO, HEO SSO has the best successful rate

The success rates for SpaceX launches is directly proportional to yearly line. Since 2013, the success rates are steadily increased.

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

The lower weighted payload is better than the heavier weighted payload for successful rate.

The Tree Classification Algorithm is the best Machine Learning algorithm for this dataset.

Appendix

- HAVERSINE FORMULA
- IBM DB2

HAVERSINE FORMULA

"The **haversine formula** determines the great-circle distance between two points on a sphere given their longitudes and latitudes. " - Wikipedia

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

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

$$d = R \cdot c$$

```
def calculate_distance(lat1, lon1, lat2, lon2):
    # approximate radius of earth in km
    R = 6373.0

    lat1 = radians(lat1)
    lon1 = radians(lon1)
    lat2 = radians(lat2)
    lon2 = radians(lon2)

    dlon = lon2 - lon1
    dlat = lat2 - lat1

    a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))

    distance = R * c
    return distance
```

IBM DB2

- **"Db2** is a family of data management products, including database servers, developed by IBM. They initially supported the relational model, but were extended to support object–relational features and non-relational structures like JSON and XML." - Wikipedia
- The author used DB2 to perform SQL commands such as pull and extract feature's data, stored data,...