Winning Space Race with Data Science

Qingyi Li
05/18/2022

GitHub

Outline

Executive Summary

Introduction

Methodology

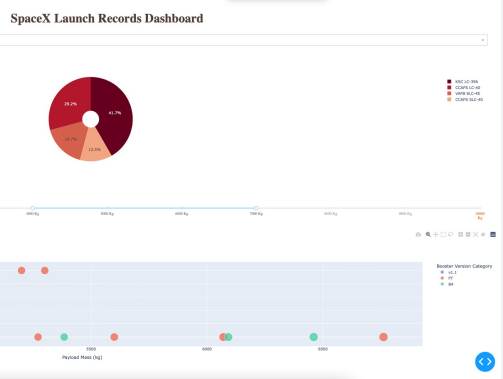
Results

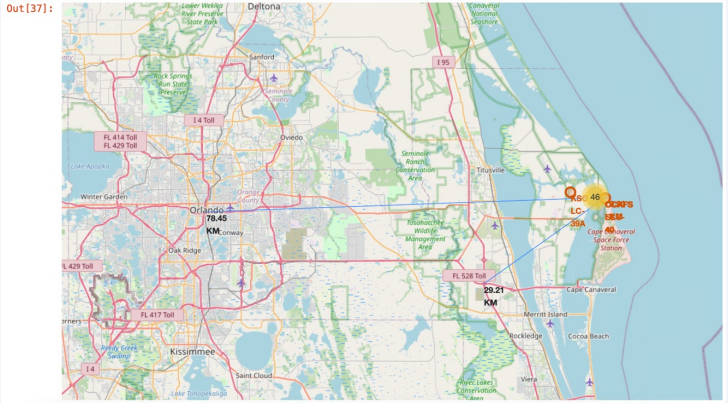
Conclusion

Executive Summary

- Summary of methodologies
  - Data collection using API & Web Scraping
    - SpaceX Rest API
    - Web scraping from Wikipedia
  - Data Wrangling
    - Data preprocessing and transformation for machine learning
  - Exploratory Data Analysis
    - Using SQL
    - Catplot, scatter plots and bar plots
    - Data visualization using Folium and interactive plots with Dash
  - Predictive Analysis

- Summary of all results
  - Exploratory data analysis results
    - Shows as interactive plots with Dash
    - Predictive analysis results





Introduction

Background

In this capstone, we will predict 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.

Research Questions:

1. What are the relationships between flight numbers and successful launch rate at each launch site?
2. What are the relationships between orbits and successful launch rate at each launch site?
3. What are the relationships payload mass and successful launch rate?
4. What machine learning model show the best performance using SpaceX dataset?

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- SpaceX Rest API
- Data collection by web scraping from Wikipedia

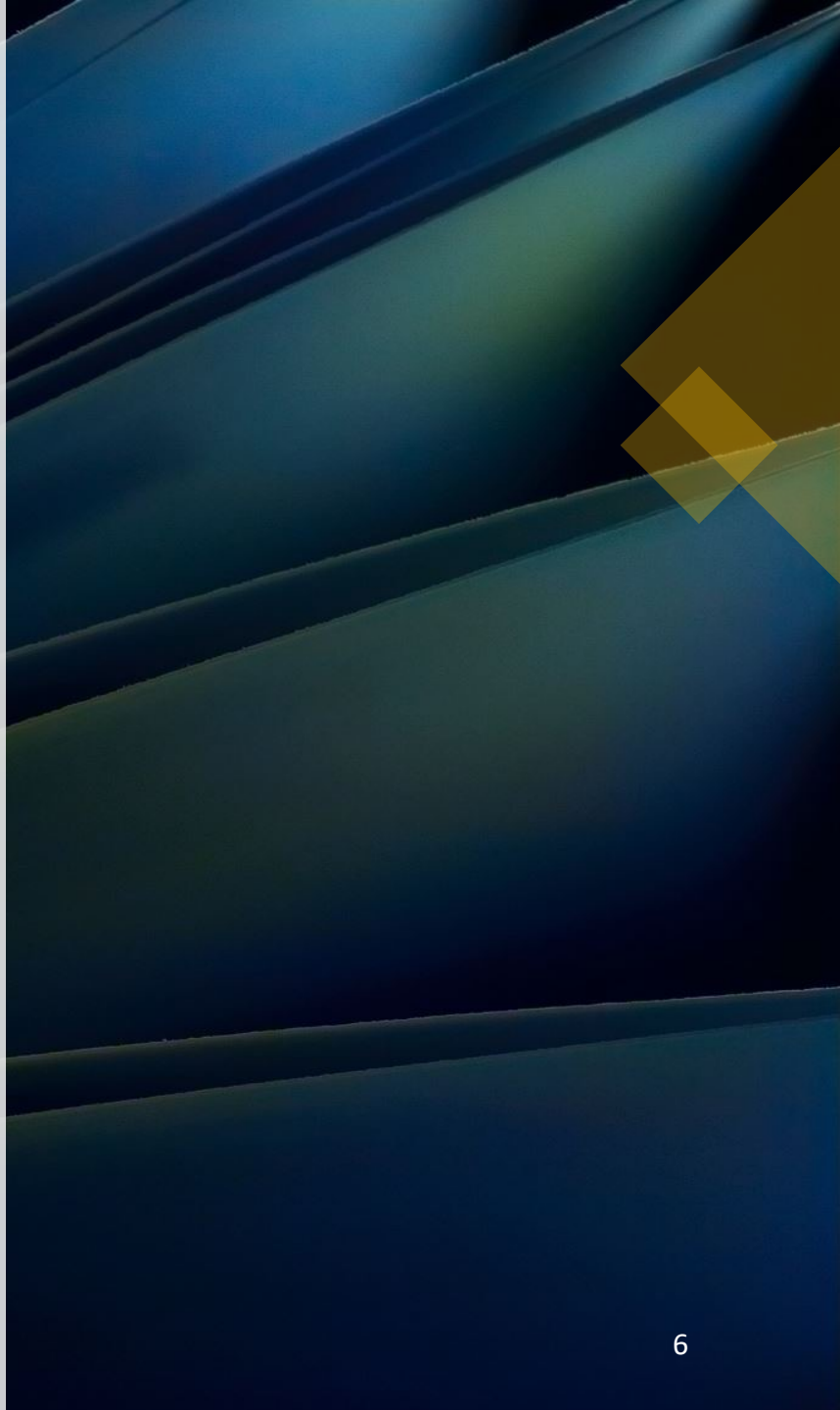
Perform data wrangling

- On-hot Encoding for categorical features
- Feature selection: selecting features in the dataset that we want to apply for data analysis

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

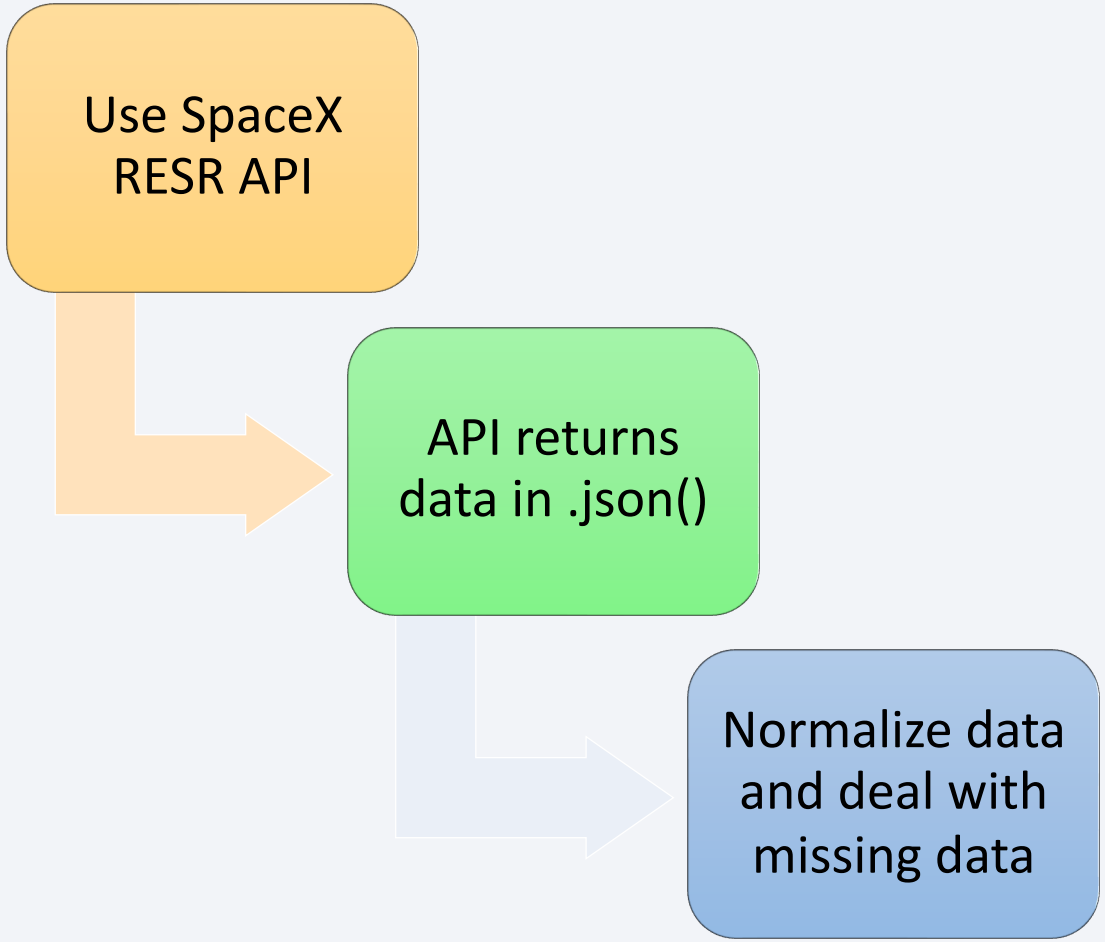
Perform predictive analysis using classification models



Data Collection

- **The SpaceX Launch dataset was collected by applying REST API**
  - Request and parse the SpaceX launch site data using the ”Get’ request
  - Filter the data frame to only include “Falcon 9”
  - Dealing with missing data
- **Web scraping SpaceX launches records from Wikipedia**
  - Request the Falcon 9 wiki page from its URL
  - Extract all variables names from the HTML table header
  - Create a data frame by parsing the launch HTML tables

Data Collection – SpaceX API




```
In [9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.a
```

We should see that the request was successfull with the 200 status response code

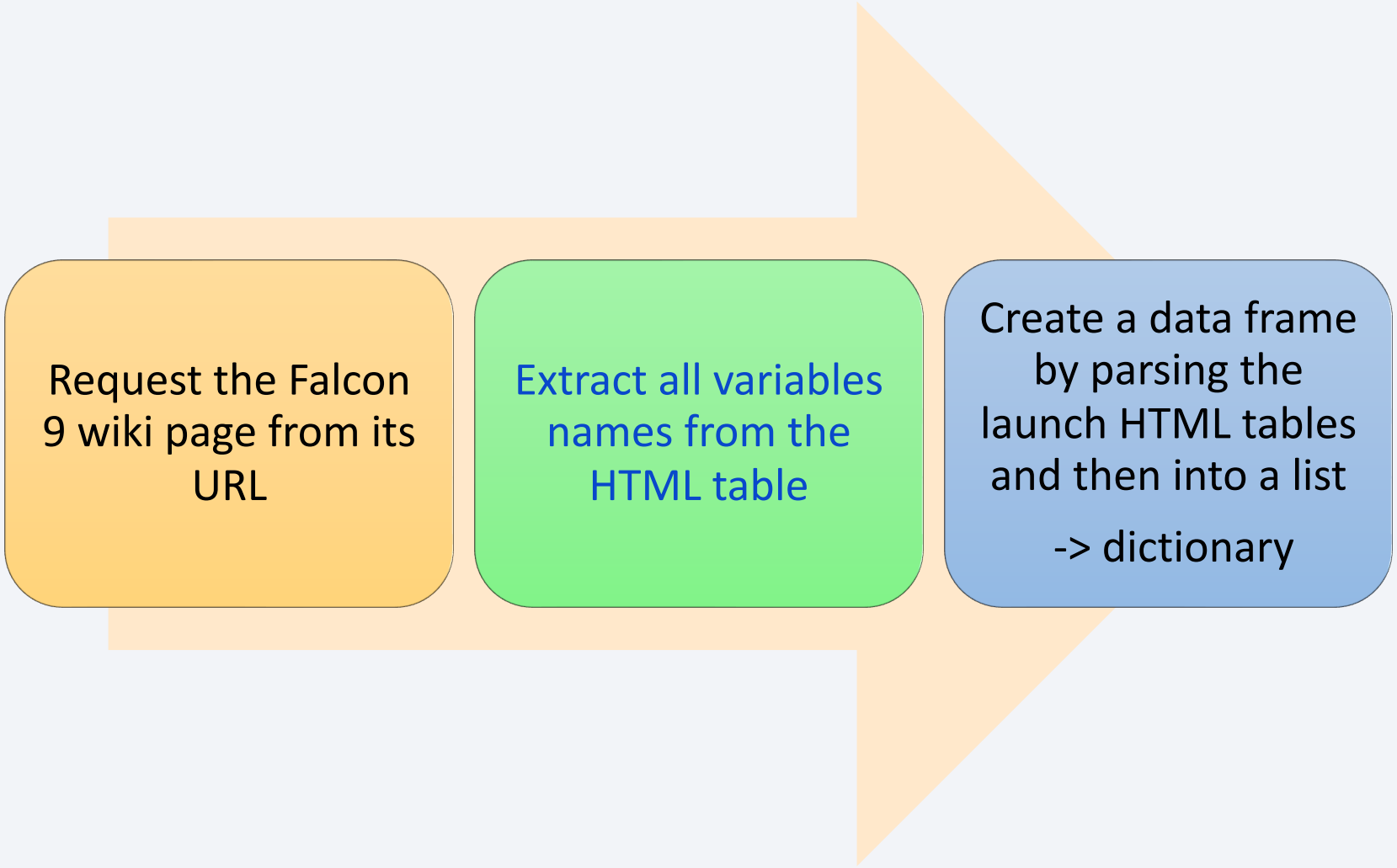
```
In [10]: response.status_code
Out[10]: 200
```

```
# Use json_normalize meethod to convert the json result into a dataframe
json_list = requests.get(static_json_url).json()
data = pd.json_normalize(json_list)
data.head()
```

```
# Calculate the mean value of PayloadMass column
avg_payload_mass = data_falcon9['PayloadMass'].astype('float').mean(a
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, avg_payload_mass, inplace
```

GitHub

Data Collection - Scraping



GitHub

Data Collection - Scraping

- Request the Falcon 9 wiki page from its URL
- Extract all variables names from the HTML table header
- Create a data frame by parsing the launch HTML tables

HTTP response.

```
In [6]: # use requests.get() method with the provided static_url
        # assign the response to a object
        df = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

```
In [7]: # Use BeautifulSoup() to create a BeautifulSoup object from a respons
        soup = BeautifulSoup(df, 'html5lib')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
In [8]: # Use soup.title attribute
        tag_title=soup.title
        tag_string_tag_title = tag_title.string
        tag_string_tag_title
Out[8]: 'List of Falcon 9 and Falcon Heavy launches - Wikipedia'
```

```
In [10]: # Let's print the third table and check its content
         first_launch_table = html_tables[2]
         print(first_launch_table)
```

```
In [ ]: extracted_row = 0
        #Extract each table
        for table_number,table in enumerate(soup.find_all('table',"wikitable
            # get table row
            for rows in table.find_all("tr"):
                #check to see if first table heading is as number correspondi
                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 ke
                    #print(flight_number)
                    datatimelist=date_time(row[0])
```

```
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']=[]
```

```
In [ ]: df=pd.DataFrame(launch_dict)
```

```
In [ ]: df.to_csv('spacex_web_scraped.csv', index=False)
```

GitHub

Data Wrangling

- Introduction
  - The purpose of data wrangling lab is to perform Exploratory Data Analysis (EDA) and to determine training labels.
  - There are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident
  - we will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

	Mission Outcome
True Ocean	successfully landed to a specific region of the ocean
False Ocean	unsuccessfully landed to a specific region of the ocean
True RTLS	successfully landed to a ground pad
False RTLS	unsuccessfully landed to a ground pad
True ASDS	successfully landed on a drone ship
False ASDS	unsuccessfully landed on a drone ship

GitHub

Data Wrangling Flowchart

Calculate the number of launches on each site

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

Calculate the number and occurrence of each orbit

```
In [6]: # Apply value_counts on Orbit column
        df['Orbit'].value_counts()
```

Calculate the number and occurrence of mission outcome per orbit type

```
In [7]: # landing_outcomes = values on Outcome column
        landing_outcomes = df['Outcome'].value_counts()
        landing_outcomes
```

```
In [9]: bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
        bad_outcomes
```

Create a landing outcome label from Outcome column

```
In [10]: # landing_class = 0 if bad_outcome
         # landing_class = 1 otherwise
         df['Class']=df['Outcome'].apply(lambda landing_class: 0 if landing_cl
         df[['Class']].head(5)
```

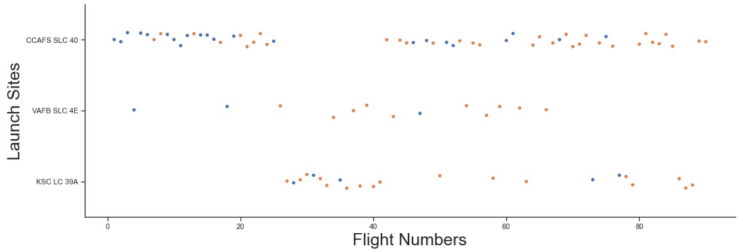
```
In [13]: df["Class"].mean()
Out[13]: 0.6666666666666666
```

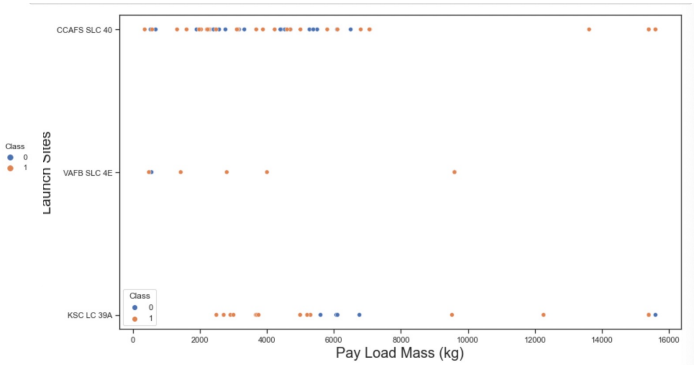
GitHub

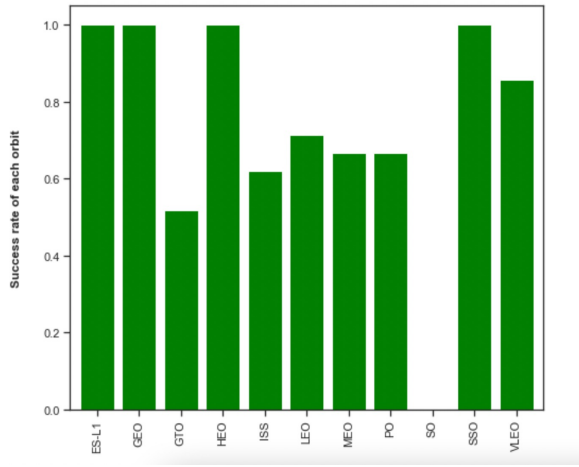
EDA with Data Visualization

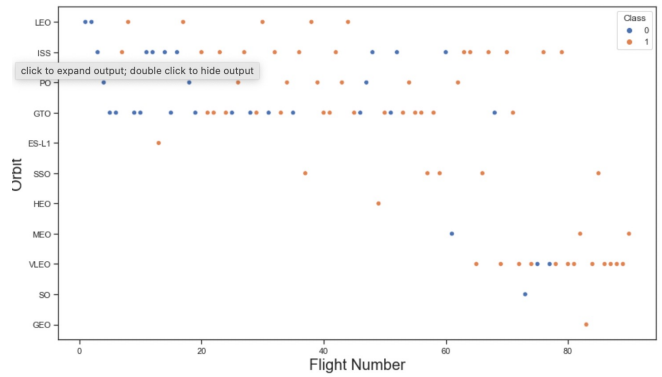
- Scatter Plots
  - Visualize the relationship between Flight Number and Launch Site
  - Visualize the relationship between Payload and Launch Site
  - Visualize the relationship between success rate of each orbit type
  - Visualize the relationship between Flight Number and Orbit type
  - Visualize the relationship between Payload and Orbit type
  - Visualize the launch success yearly trend

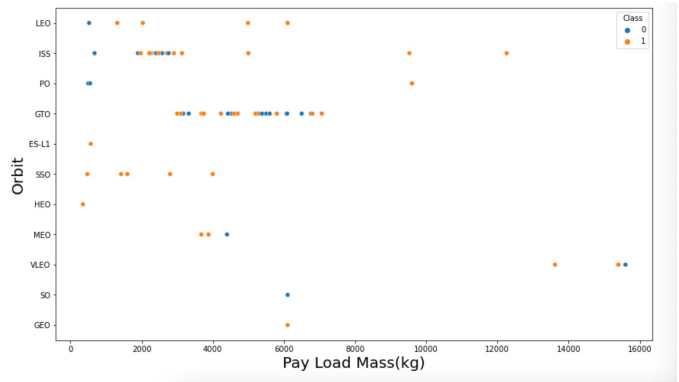
GitHub

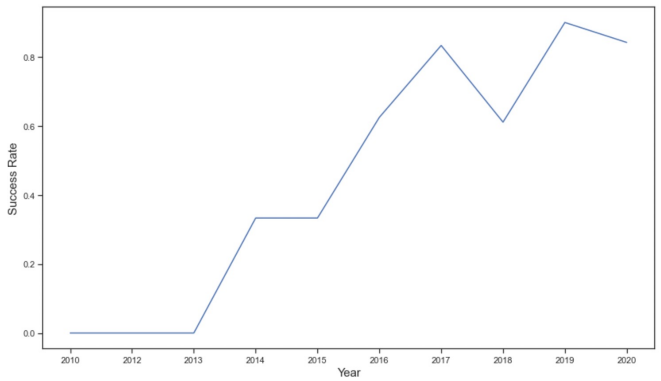












EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 4 records where launch sites begin with the string “CCA”
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List the 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.
- 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

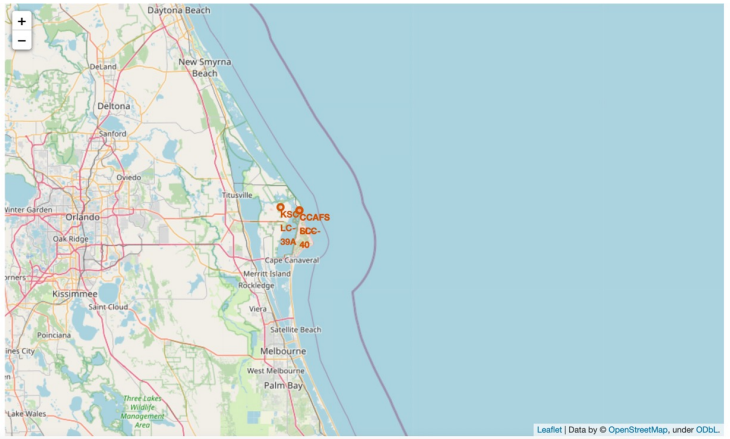
Build an Interactive Map with Folium

- The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site.
- Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations.
- Our purpose is to be able to find geographical patterns about launch site.
- Doing so, we first need to mark all launch site on a map. Then, we need to mark the success/fail launches for each site on the map. Finally, we will calculate the distance between a launch site to it proximities.

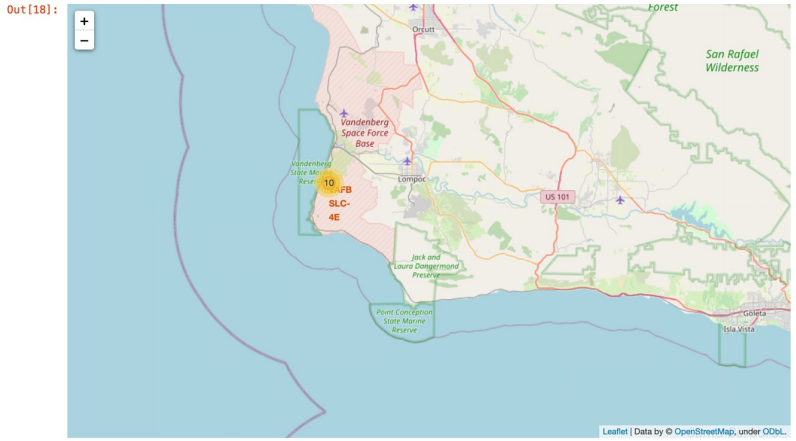
GitHub

Build an Interactive Map with Folium

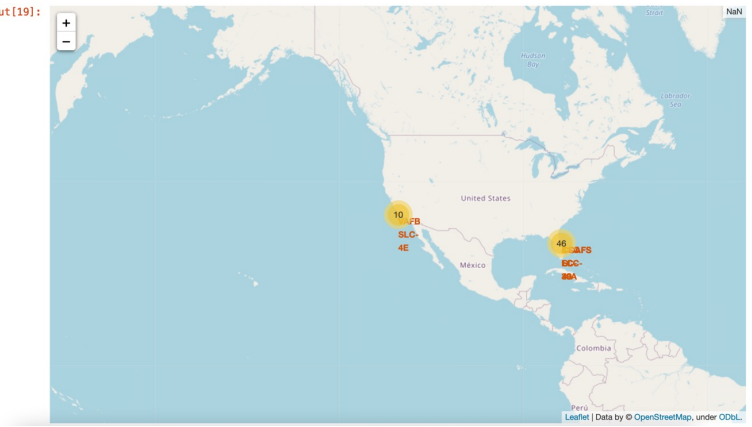
Out[12]:



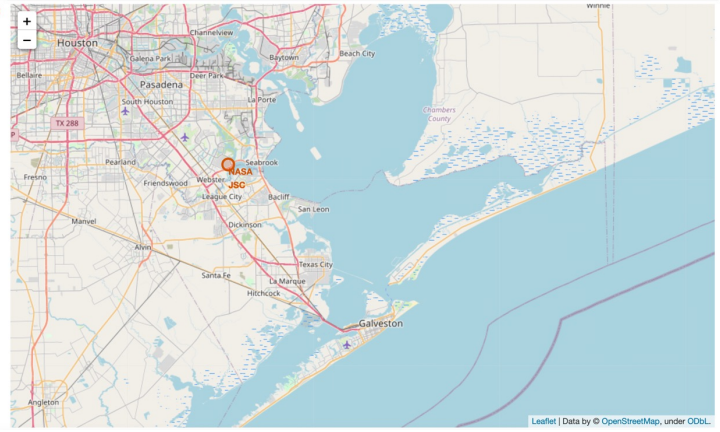
Out[18]:



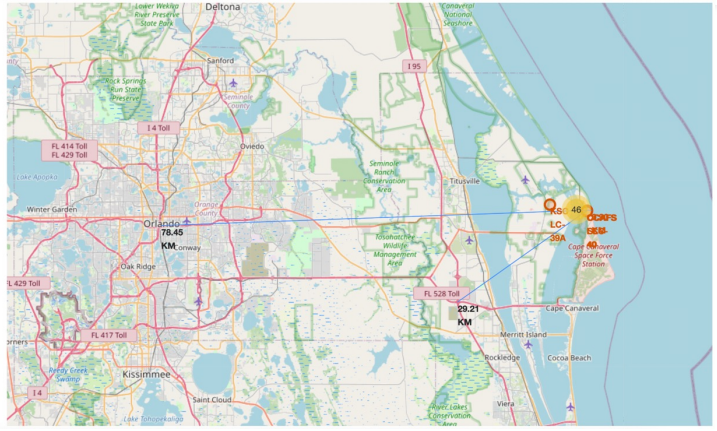
Out[19]:



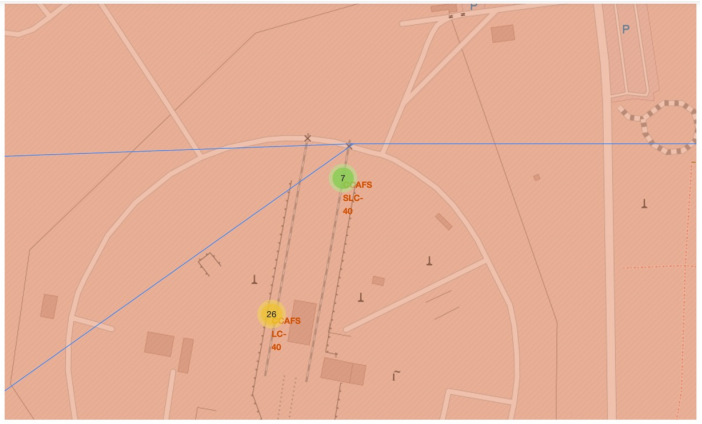
Out[27]:



Out[37]:



Out[37]:



GitHub

Build a Dashboard with Plotly Dash

Plotly Dash can help users perform interactive visual analytics on SpaceX launch data in real-time.

In the visual analysis using the Plotly Dash, we are able to obtain insights on:

- The launch site has the largest successful launches.
- The launch site has the highest launch success rate.
- The payload range(s) gas the highest launch success rate.
- Payload range(s) has the lowest launch success rate.
- The Falcon9 Booster version has the highest launch success rate.

GitHub

Predictive Analysis (Classification)

Perform exploratory Data Analysis and determine Training Labels

Create a column for the class

Standardize the data

Split into training data and test data

Find best Hyperparameter for SVM, Classification Trees and Logistic Regression

Find the method performs best using test data

GitHub

Predictive Analysis (Classification) Continued

GitHub

Building Model

- loading dataset to Numpy and Pandas
- Standardized data
- Train Test Set Split
- Choose machine learning algorithms
- GridSearch CV to predefine hyperparameters and fit the model on our training set

Model Evaluation

- Accuracy Check
- Tune hyperparameters for each machine learning algorithm
- Plot confusion matrix

Model Improvement

- Feature Engineering
- Algorithms tuning

Best Hyperparameter

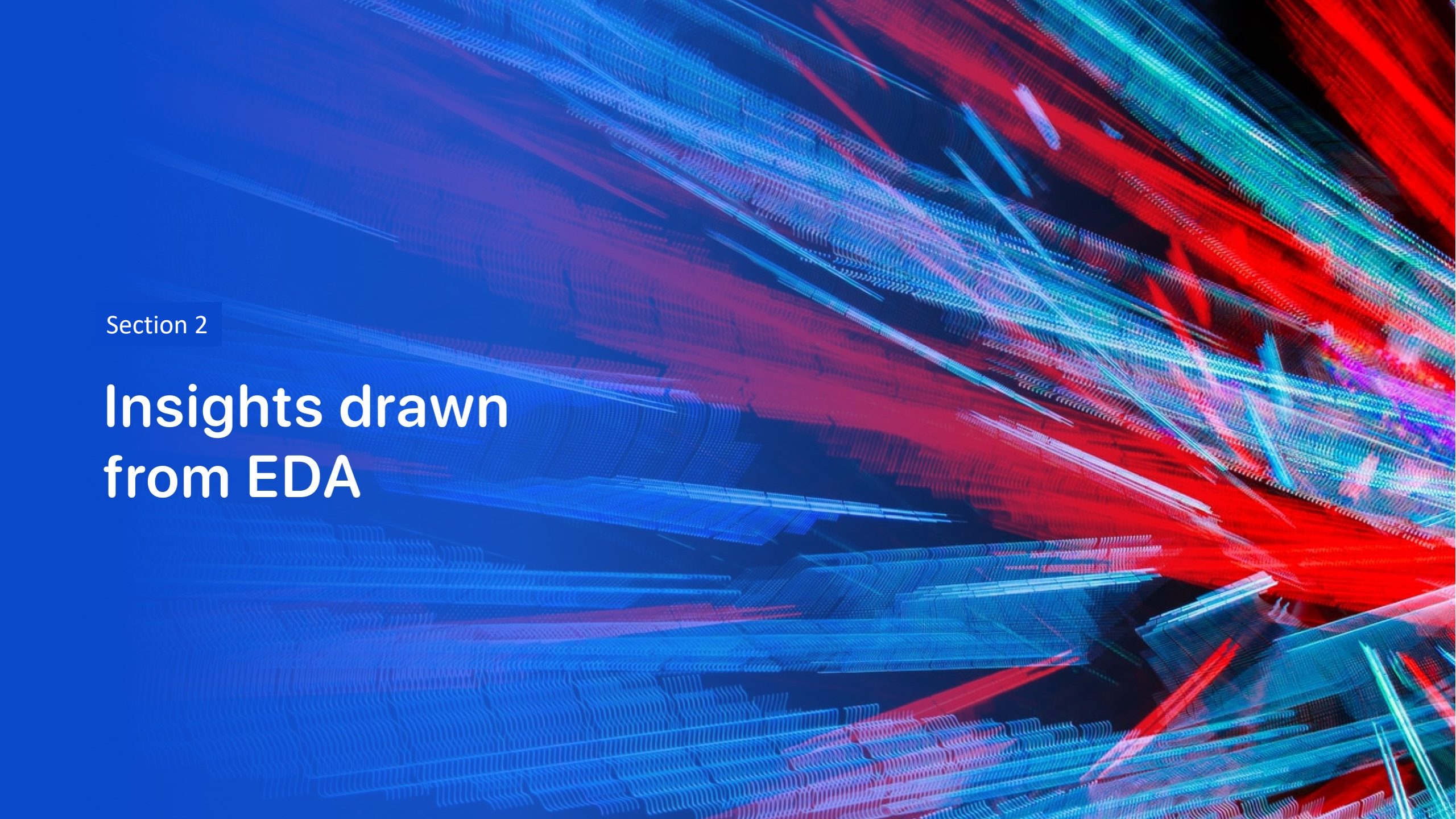
- Find the best accuracy score that shows the best model

Results

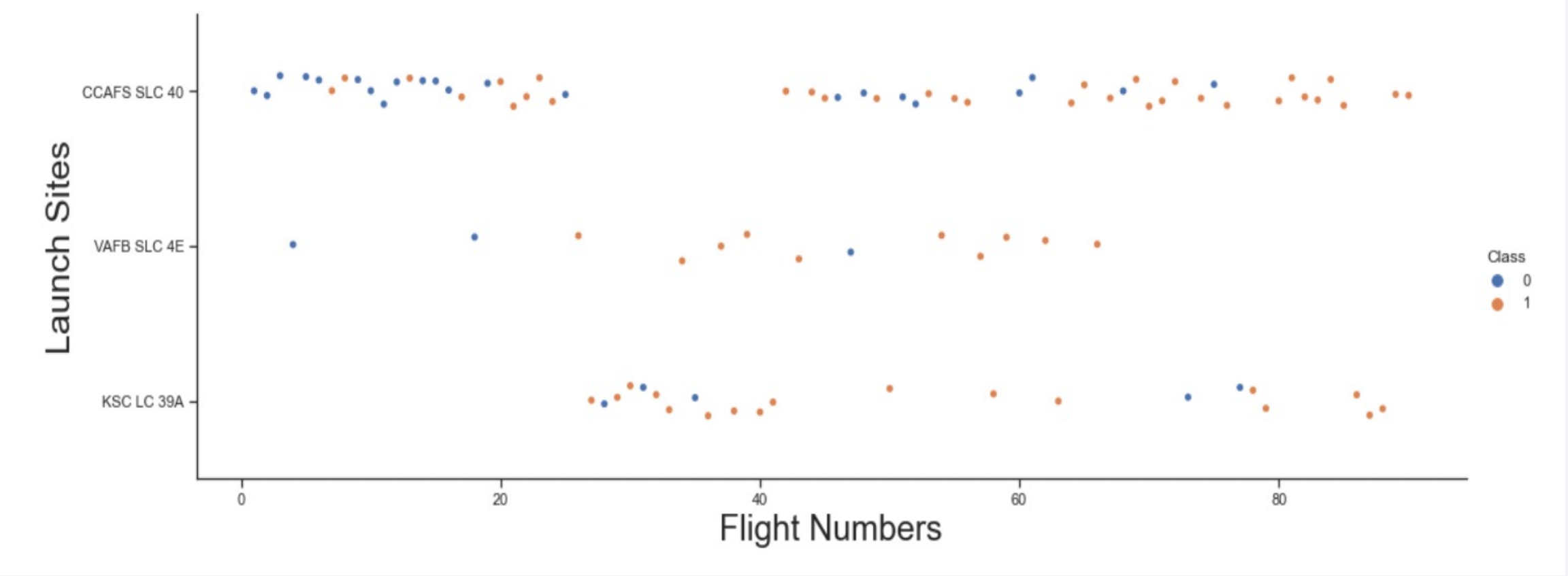
Exploratory data analysis results

Interactive analytics demo in screenshots

Predictive analysis results

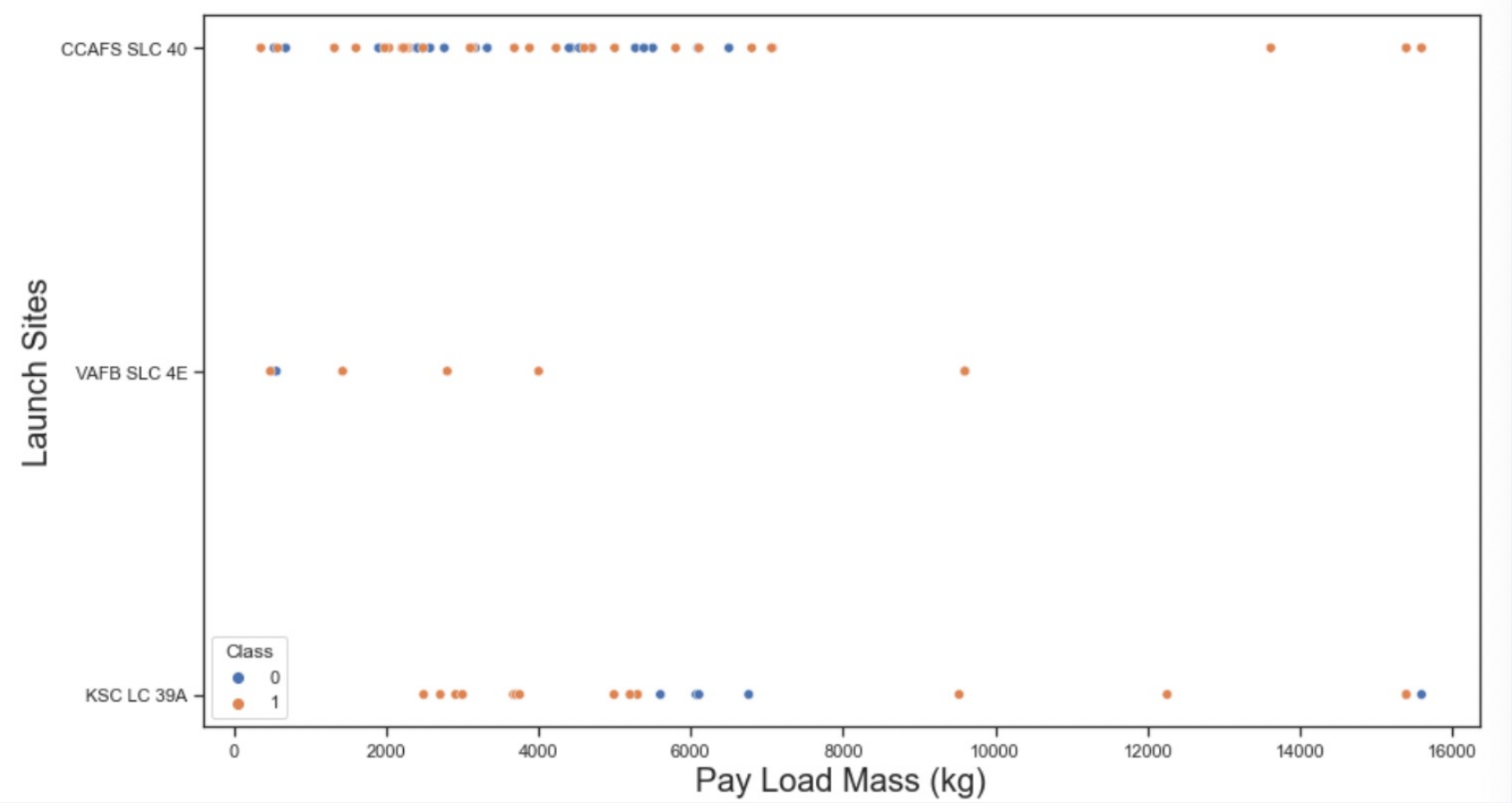

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



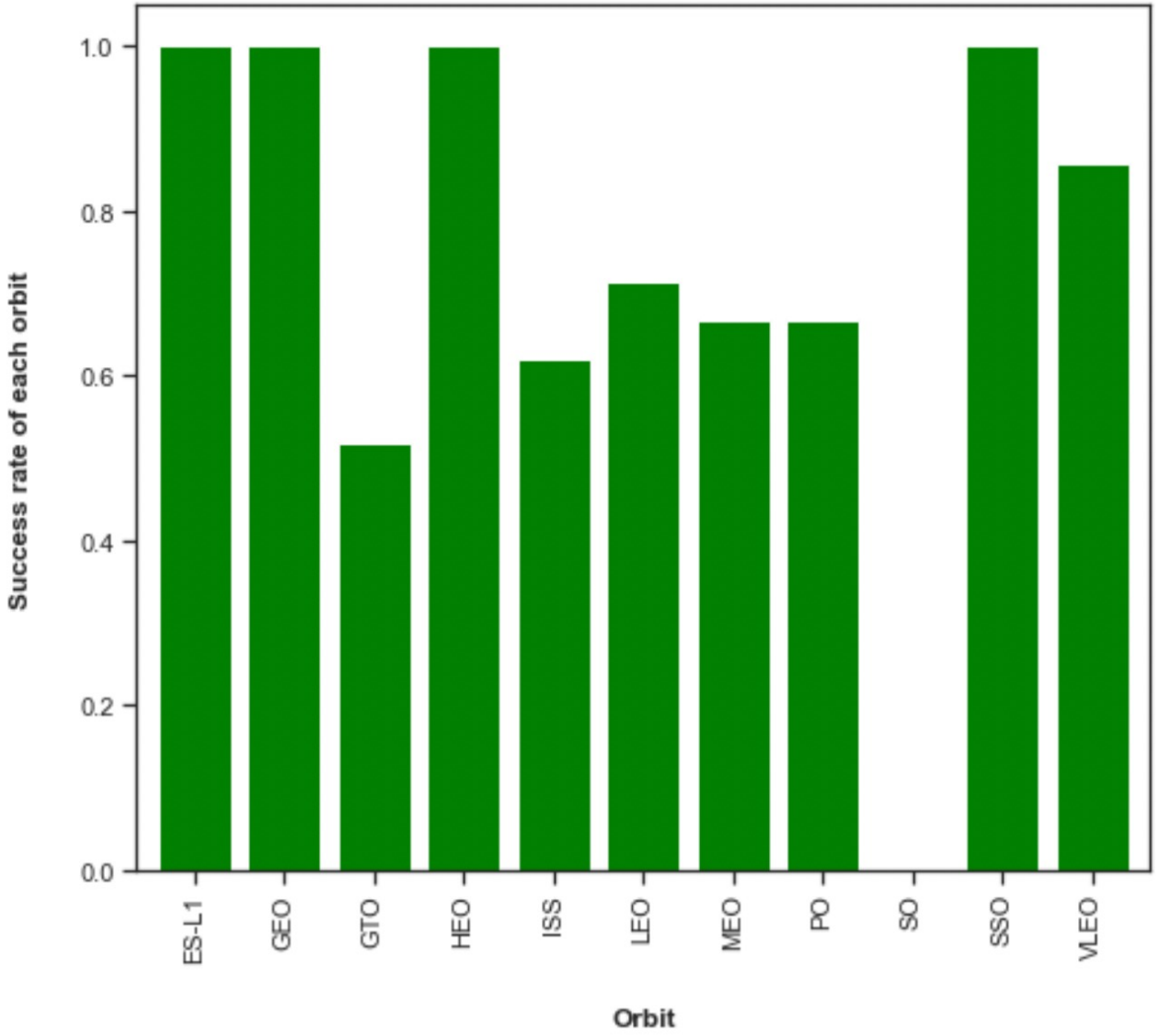
The more flights at a launch site the higher success rate at the launch site.

Payload vs. Launch Site



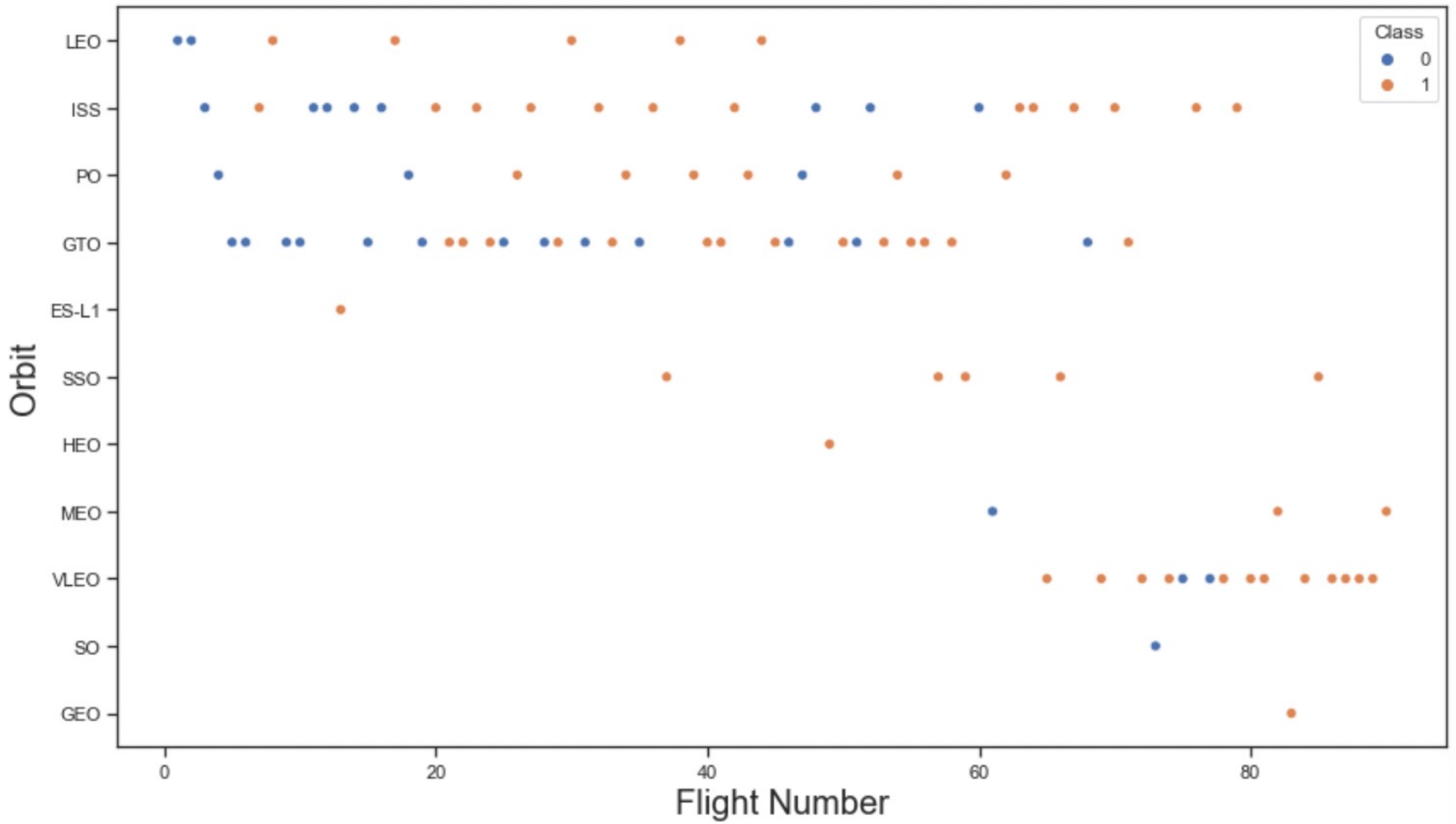
As shown in the Figure (left), the pattern between payload mass and launch site is not very clear. We cannot tell the pattern based on this graph.

Success Rate vs. Orbit Type



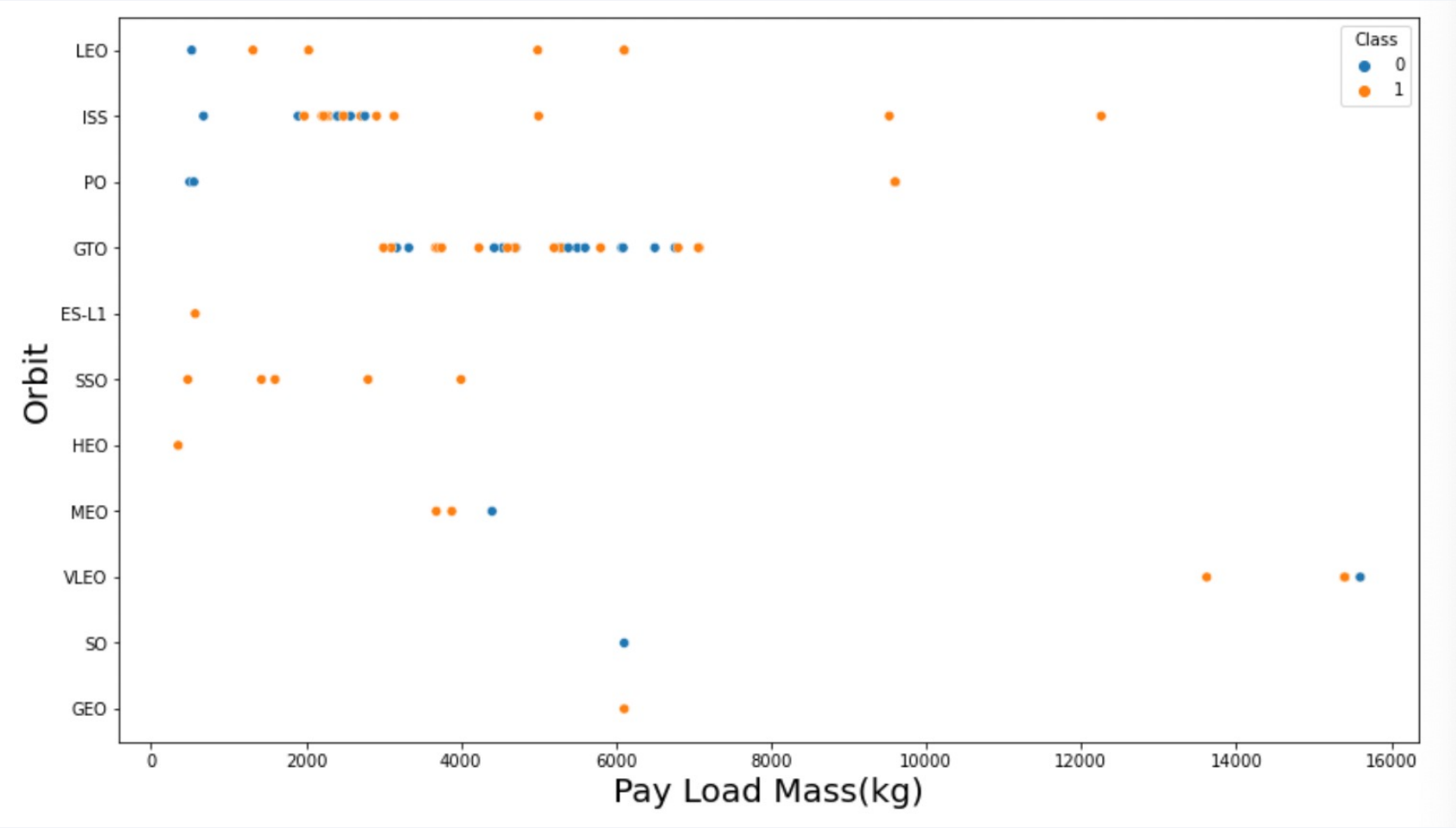
The orbits ”ES-L1”, “GEO”, “HEO”, and “SSO” show the highest launch success rate among all orbits.

Flight Number vs. Orbit Type



As shown in the Figure, there is no clear pattern between flight numbers and orbit type.

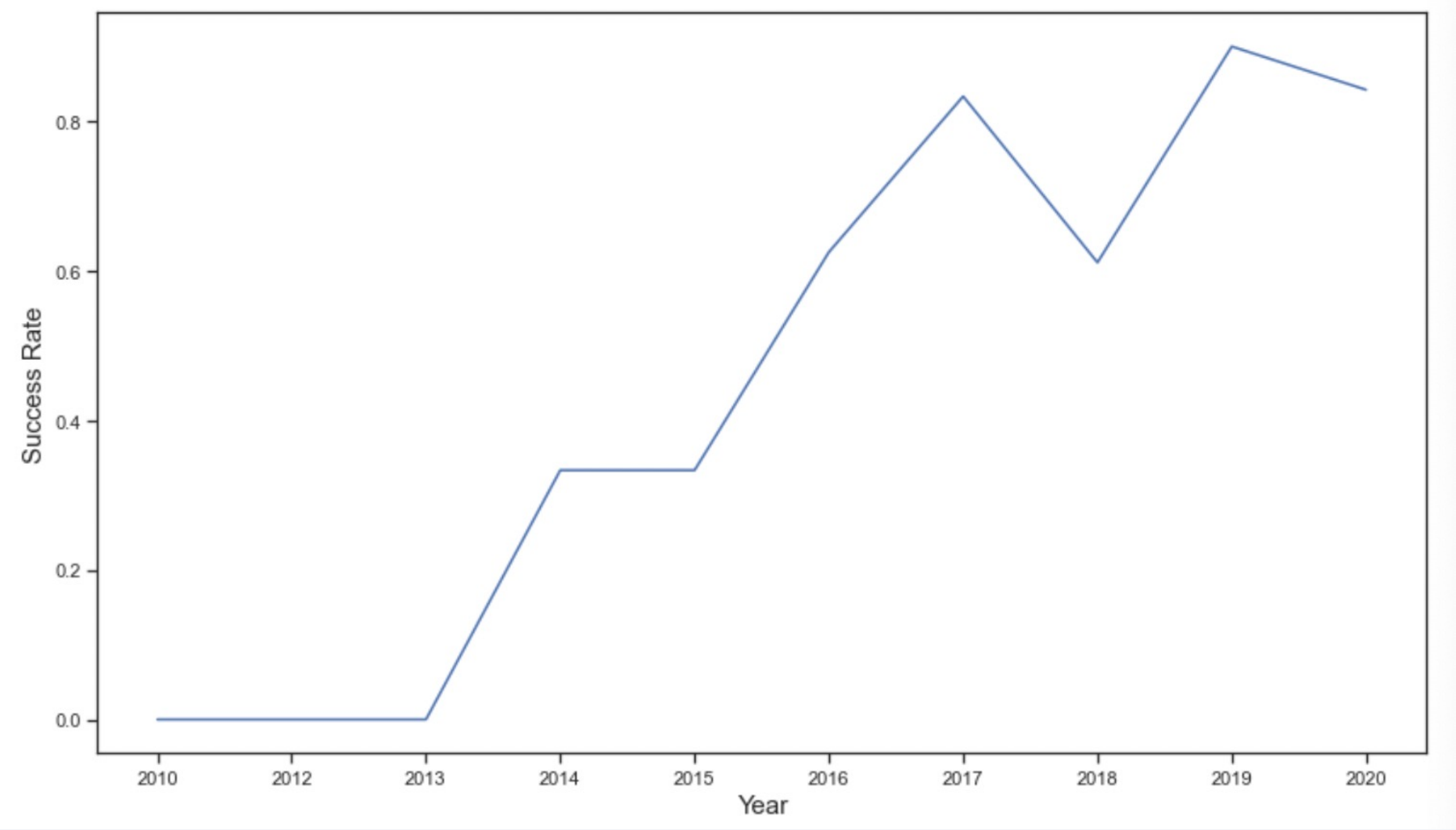
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

However, for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



As shown in the Figure, the successful launch rate since 2013 kept increasing till 2020.

All Launch Site Names

```
In [5]: %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

Description:

We will show all the launch site by applying the DISTINCT function that we pull all unique values for launch sites from SpaceX dataset

Out[5]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

```
In [6]: %sql SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

- Present your query result with a short explanation here

Out[6]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
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
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

We use “LIMIT 5” function to show the 5 records from SpaceX dataset and apply condition function using “LIKE” to get wild card “CCS%”.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
In [7]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)"
        FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)';
```

- Present your query result with a short explanation here

Out[7]:

Total Payload Mass by NASA (CRS)
45596

  - We applied the function “SUM” to display the total payload mass carried by boosters launched by NASA and clause filter “WHERE” to filter the data by name with “NASA (CRS)”

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

```
In [8]: %sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster V
        WHERE BOOSTER_VERSION = 'F9 v1.1';
```

- Present your query result with a short explanation here

Out[8]:

Average Payload Mass by Booster Version F9 v1.1
2928

  - We used the function “AVG” to display average payload mass carried by booster version F9 v1.1 and ”WHERE” to filter the data with booster version with “F9 v1.1”.

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

```
In [9]: %sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad"
        WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

- Present your query result with a short explanation here

Out[9]:

First Succesful Landing Outcome in Ground Pad
2015-12-22

  - We applied the “MIN” function and “WHERE” function to filter on Launching_Outcome with “Success (ground pad)” to list the SpaceX date when the first successful landing outcome in ground pad was achieved.

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
In [10]: %sql SELECT BOOSTER_VERSION FROM SPACEX WHERE
         LANDING__OUTCOME = 'Success (drone ship)' \
         AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

- Present your query result with a short explanation here

Out[10]:

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

We first applied the “SELECT” function to list show column “booster_version” and applied “WHERE” function to filter on both “Launching_Outcome” and “Payload_Mass_KG to list the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
In [14]: %sql SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0
         AS "Successful Mission", \
             sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end)
         AS "Failure Mission" \
         FROM SPACEX;
```

- Present your query result with a short explanation here

Out[14]:

Successful Mission	Failure Mission
100	1

To list the total number of successful and failure mission outcomes, we used “SUM” functions to calculate both “%Success” and “%Failure” under the ”SECLECT” function.

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
In [15]: %sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried
         WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

- Present your query result with a short explanation here

Out[15]:

Booster Versions which carried the Maximum Payload Mass
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

We used a subquery “MAX” to list the names of the booster_versions which have carried the maximum payload mass.

2015 Launch Records

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

```
In [19]: %sql SELECT {fn MONTHNAME(DATE)} as "Month",
         BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE year(DATE) = '2015' AND \
         LANDING__OUTCOME = 'Failure (drone ship)';
```

- Present your query result with a short explanation here

Out[19]:

Month	booster_version	launch_site
January	F9 v1.1 B1012	CCAFS LC-40
April	F9 v1.1 B1015	CCAFS LC-40

First, we need to select the booster version and launch sites and filter on “2015-%” and “AND \
LANDING__OUTCOME = 'Failure (drone ship)’;”

Second, we need to select the booster version, launch sites, and month. Then, filter on “2015-%” and “AND \
LANDING__OUTCOME = 'Failure (drone ship)’;”
Third,

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

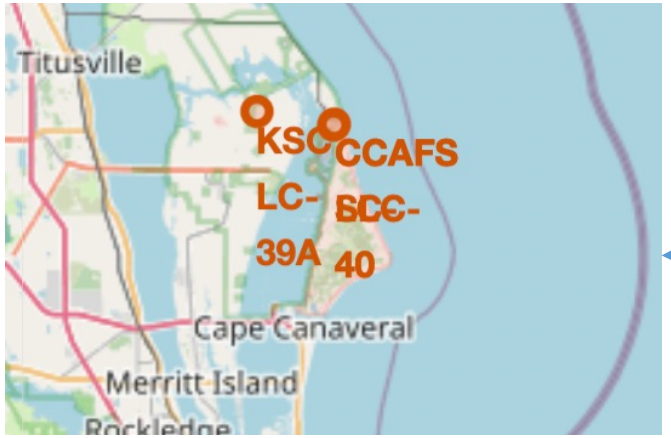
```
%sql SELECT COUNT(LANDING__OUTCOME)
AS "Rank success count between 2010-06-04 and 2017-03-20"
FROM SPACEX \
WHERE LANDING__OUTCOME LIKE '%Success%'
AND DATE > '2010-06-04' AND DATE < '2017-03-20' ;
```

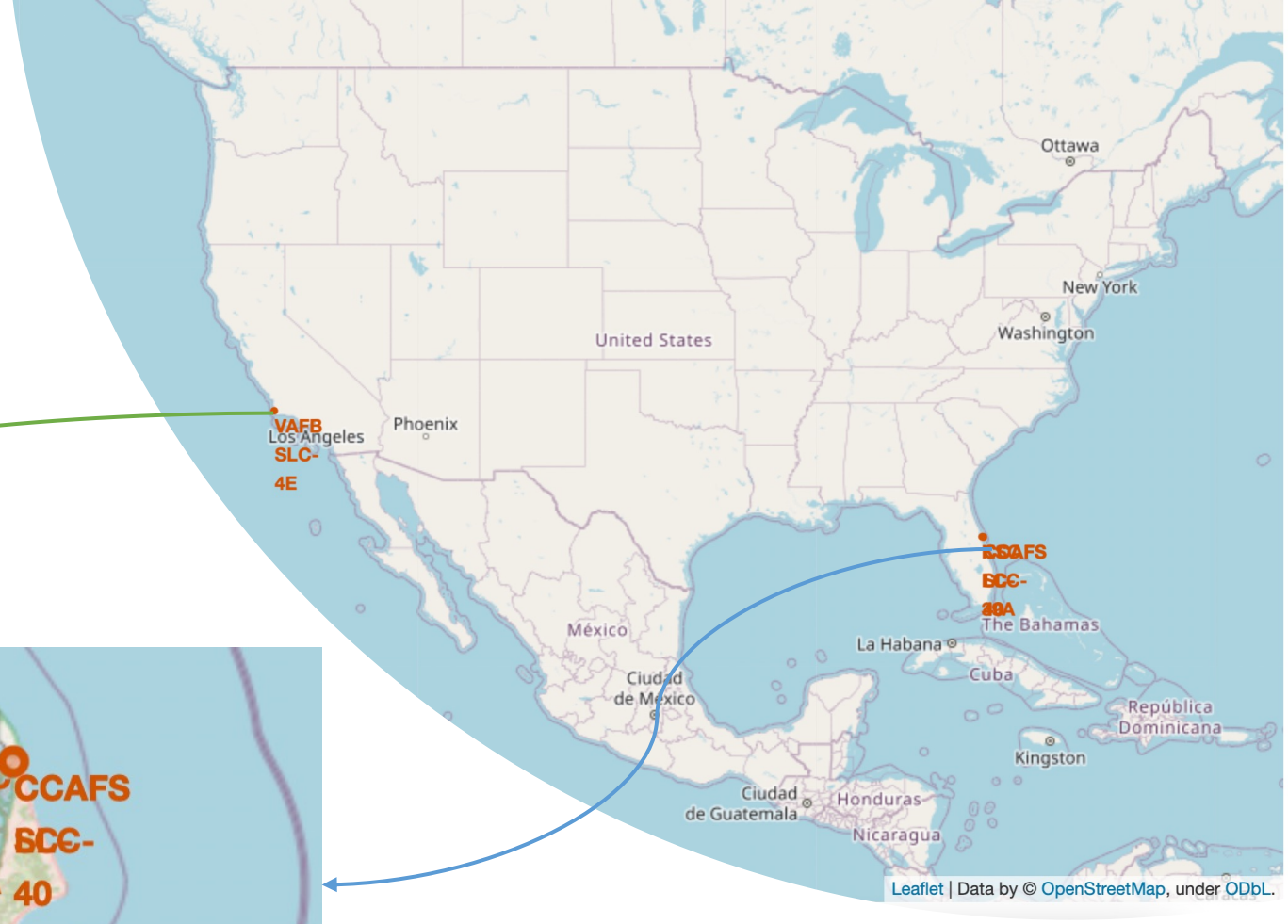
- Present your query result with a short explanation here

Rank success count between 2010-06-04 and 2017-03-20
8


Section 3
Launch Sites
Proximities Analysis

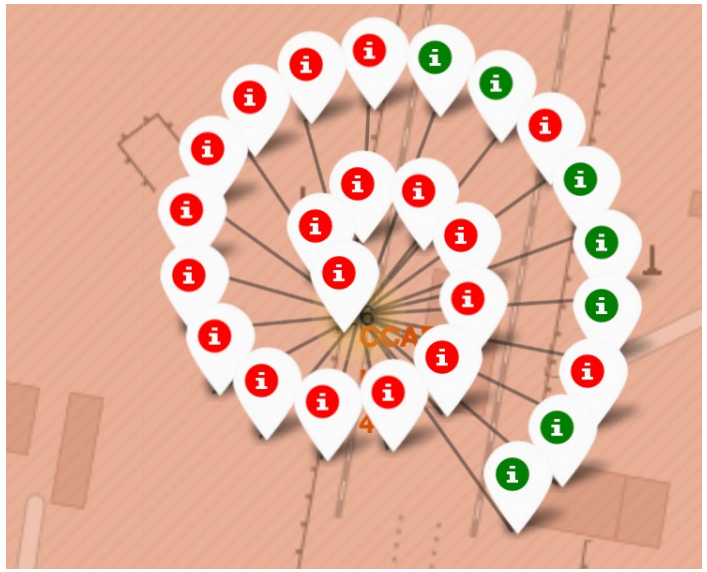
- All SpaceX Launch Sites location in the west coastline (California) and east coastline (Florida)

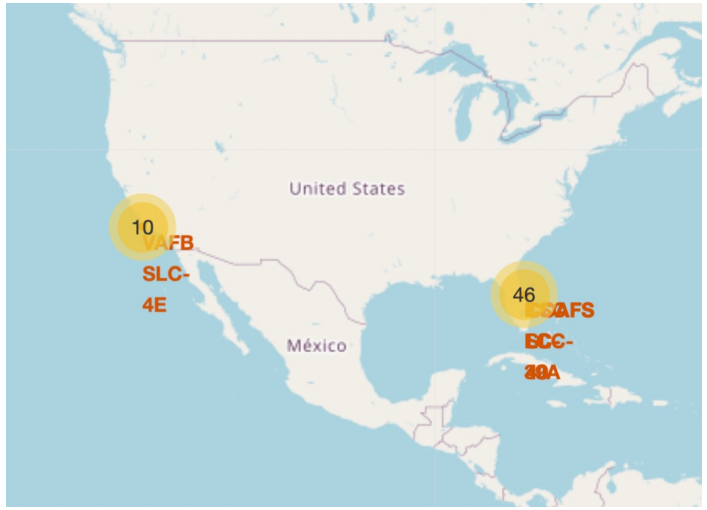


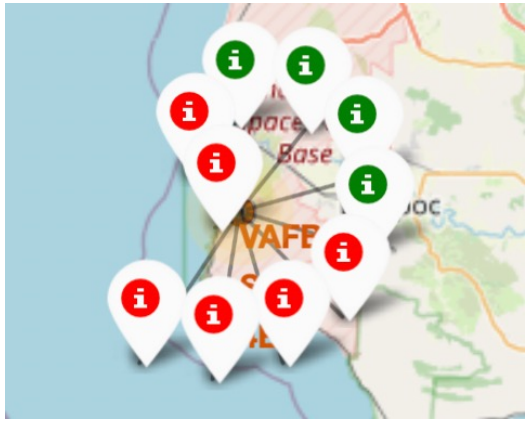


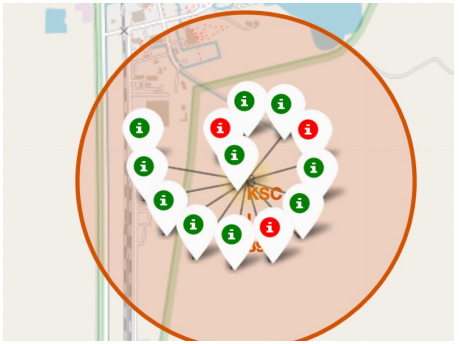
All Launch Site in the U.S.

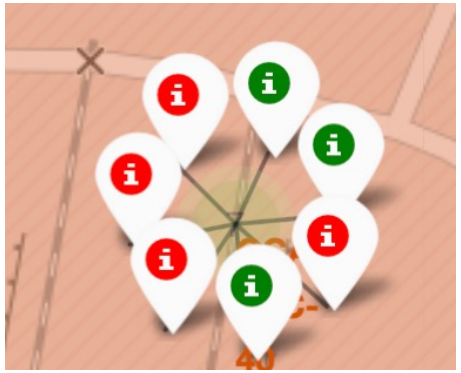
Colored Label Markers on Launch Sites





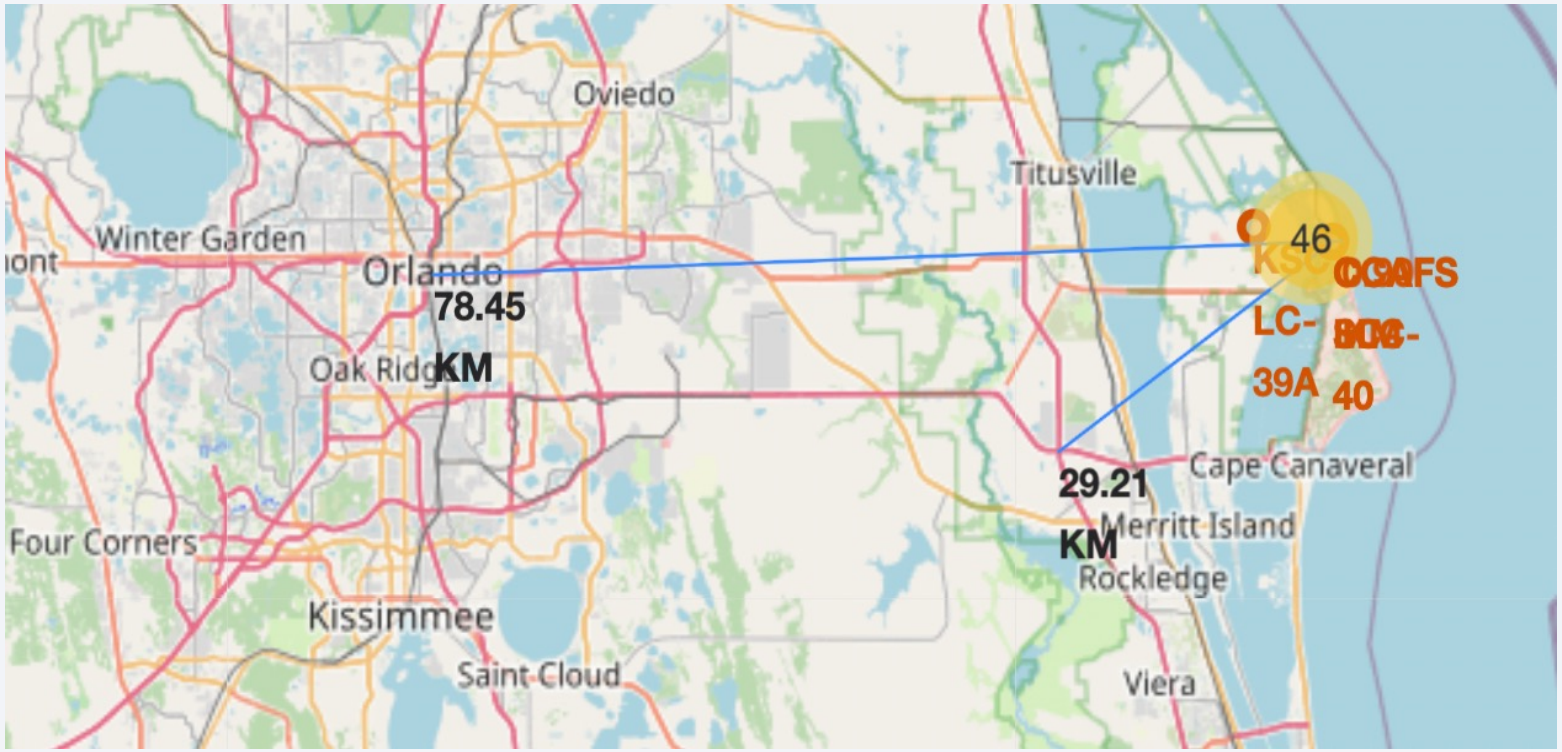






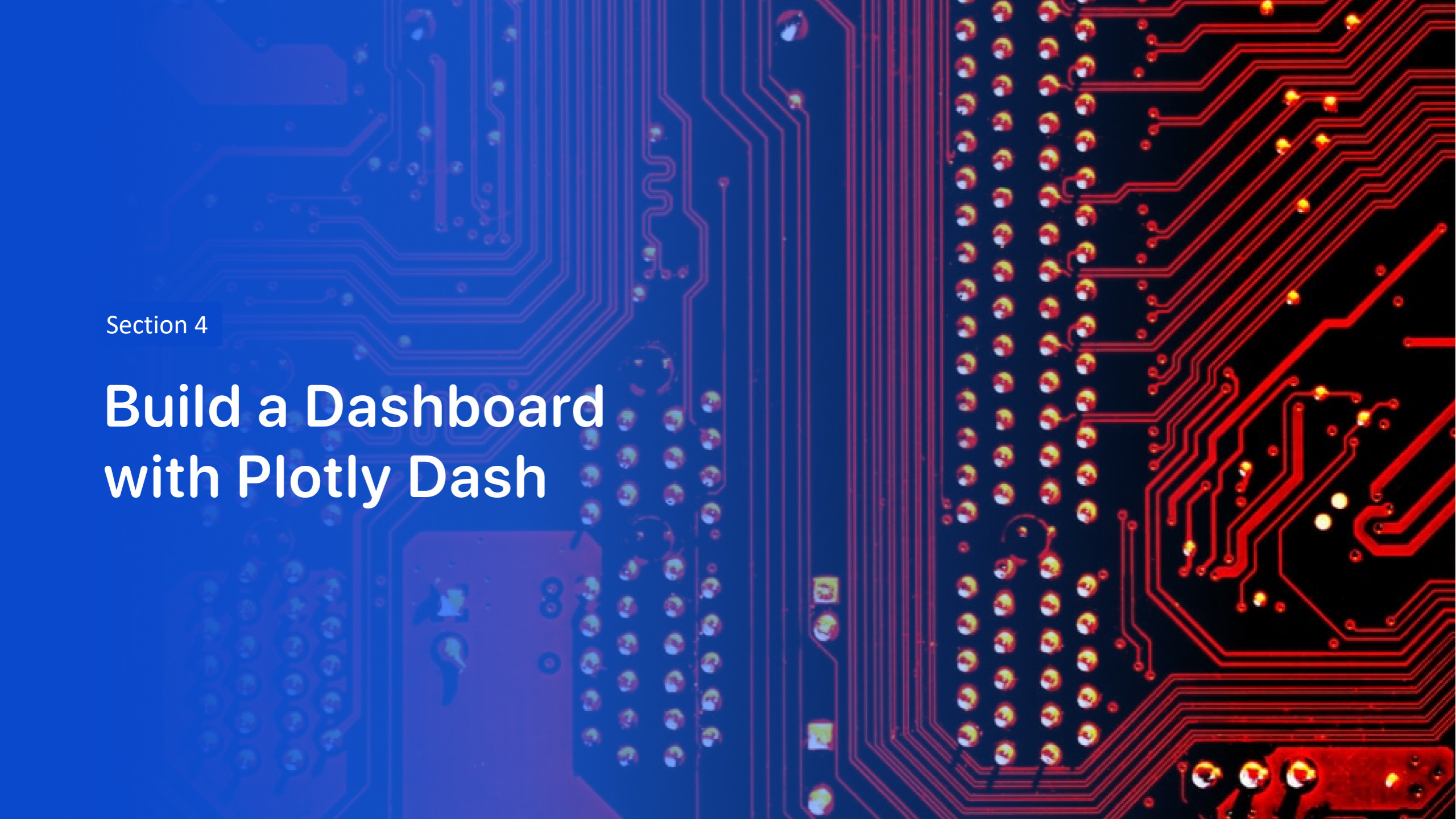
Red markers show launch failures whereas Green markers show successful launches.

Distance between SpaceX Launch Sites and Landmarks in Florida

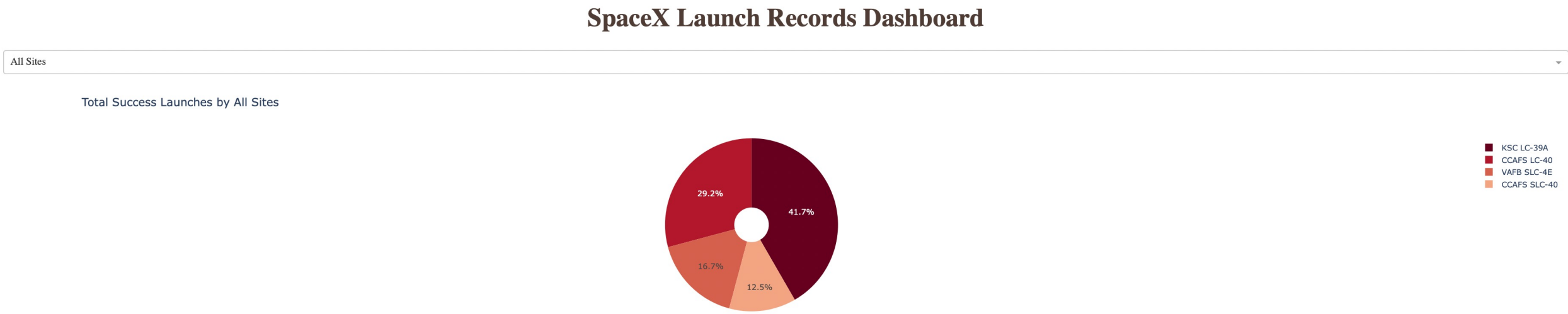


After you plot distance lines to the proximities, the Folium map shows:

- Space X launch sites are not close to railways.
- SpaceX launch sites are not close to highways.
- SpaceX launch sites are close coastline.
- SpaceX launch sites keep certain distance away from cities.

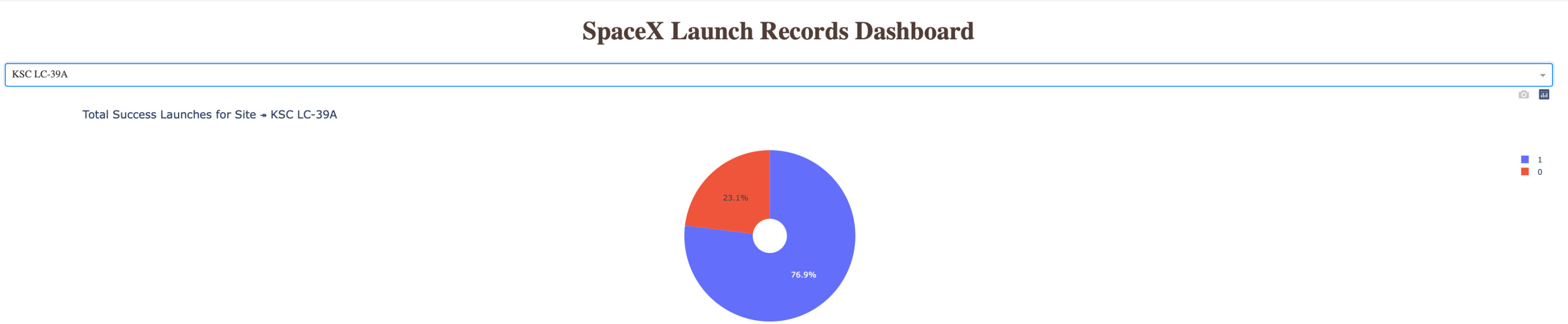

Section 4
Build a Dashboard with Plotly Dash

Successful Launch Rates in All Launch Sites



The pie chart illustrates the total success launches by all sites. KSC LC-39A shows the highest rate whereas CCAFS SLC-40 shows the lowest success rate.

KSC LC-39A with the Highest Success Rate among All Launch Sites



As shown in the above pie chart, KSC LC-39A demonstrates the highest successful rate with 76.9% among all the SpaceX launch sites.

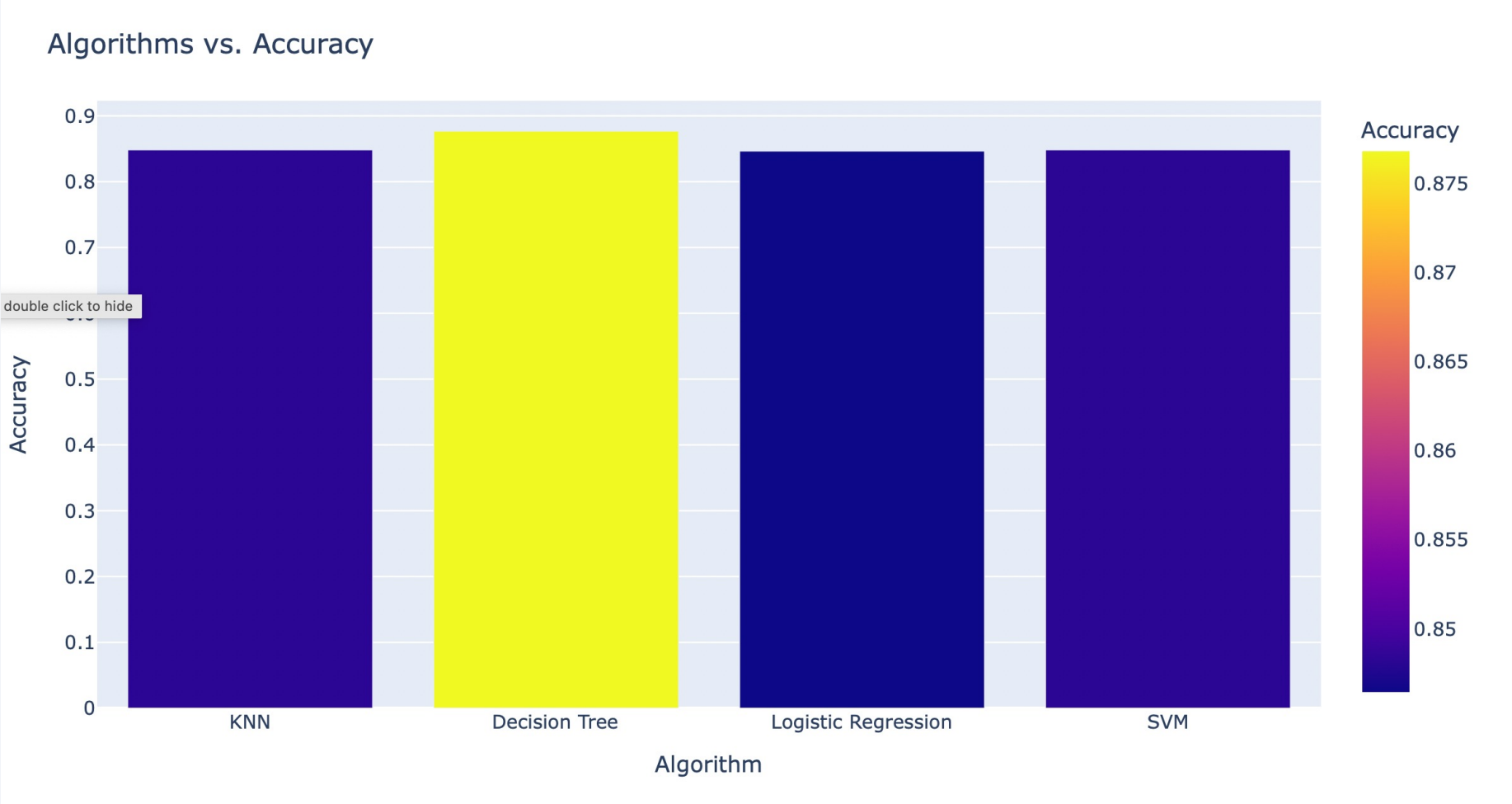
Payload Mass vs. Launch Outcomes for All Launch Sites

As the left scatter plot shown, there are more successful launch outcomes in lighter weights than those with heavier weights.

Section 5

Predictive Analysis (Classification)

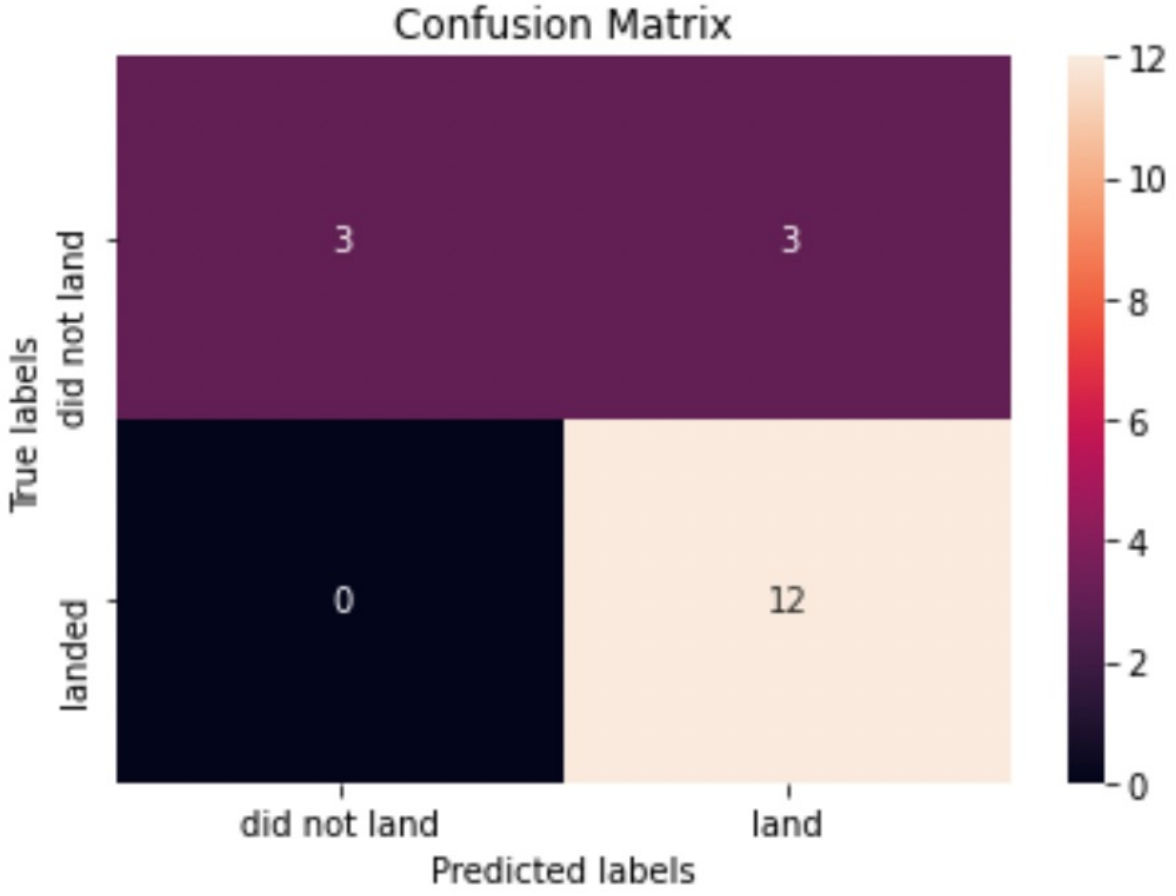
Classification Accuracy



As shown in the bar chart (left), Decision Tree model illustrates the highest classification accuracy.

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation



The left graph is the confusion matrix of our best performing model ----Decision Tree. However, in our SpaceX dataset, the confusion matrices for all four models are the same.

Conclusions

Important Conclusions

- The more flights at a launch site the higher success rate at the launch site.
- The orbits ”ES-L1”, “GEO”, “HEO”, and “SSO” show the highest launch success rate among all orbits.
- The successful launch rate since 2013 kept increasing till 2020 in SpaceX.
- The total success launches by all sites. KSC LC-39A shows the highest rate whereas CCAFS SLC-40 shows the lowest success rate. Particularly, KSC LC-39A demonstrates the highest successful rate with 76.9% among all the SpaceX launch sites.
- SpaceX had more successful launch outcomes in lighter weights than those with heavier weights.
- The Decision Tree is the best predictive model using SpaceX dataset.

Thank you!