IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Zeyad Shorman
Jan 01, 2024

Outline

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

Executive Summary

- **Summary of methodologies**

    1- Complete the Data Collection API.

    2- Complete the Data Collection with Web Scraping.

    3- Data Wrangling.

    4- Complete the EDA with SQL.

    5- EDA with Visualization Lab.

    6- Interactive Visual Analytics with Folium.

    7- Build an Interactive Dashboard with Ploty Dash.

    8- Complete the Machine Learning Prediction.

- **Summary of all results**

    1- Exploratory Data Analysis.

    2- Visual Analytics and Dashboards.

Introduction

- Project background and context
  - 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.
- Problems you want to find answers
  - you will collect and make sure the data is in the correct format from an API. we will predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

- Describe how data sets were collected.

- Data was first collected using SpaceX API by making a get request to the SpaceX API. By defining a series of helper functions that will help us use the API to extract information using identification numbers in the launch data.

- From the rocket column we would like to get the booster name. From the launchpad we would like to know the name of the launch site being used, the logitude, and the latitude. From the payload we would like to know the mass of the payload and the orbit that it is going to. From cores we would like to get the outcome of the landing, the type of the landing, number of flights with that core, whether gridfins were used, whether the core is reused, whether legs were used, the landing pad used, the block of the core which is a number used to separate version of cores, the number of times this specific core has been reused, and the serial of the core.

- performing web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled ***List of Falcon 9 and Falcon Heavy launches***

Data Collection – SpaceX API

- Data was first collected using SpaceX API by making a get request to the SpaceX API. By defining a series of helper functions that will help us use the API to extract information using identification numbers in the launch data.

- Here is the GitHub URL of the completed SpaceX API.
- https://github.com/Zeyad-Shorman/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/1%20-%20SpaceX%20%20Falcon%209%20first%20stage%20Landing%20Prediction.ipynb

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[63]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

```
[13]: static_json_url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

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

```
[64]: response.status_code
[64]: 200
```

```
[14]: response.status_code
[14]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
[65]: # Use json_normalize meethod to convert the json result into a dataframe
response_json = response.json()
data = pd.json_normalize(response_json)
```

Using the dataframe `data` print the first 5 rows

```
[66]: # Get the head of the dataframe
data.head()
```

Data Collection - Scraping

- Performing web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled ”List of Falcon 9 and Falcon Heavy launches”. Web scrap Falcon 9 launch records
  with BeautifulSoup: 1-Extract a Falcon 9 launch records HTML table from Wikipedia.
  2-Parse the table and convert it into a Pandas data frame.

- Here is the GitHub URL for Web Scraping. https://github.com/Zeyad-Shorman/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/2%20-%20Complete%20the%20Data%20Collection%20with%20Web%20Scraping%20lab.ipynb

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

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

response = requests.get(static_url)
```

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

In [7]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, 'html.parser')
```

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

In [8]:
```
# Use soup.title attribute
soup.title
```

Out[8]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

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

In [9]:
```
# 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')
```

Starting from the third table is our target table contains the actual launch records.

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

Data Wrangling

- Perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.

- The dataset includes various instances of unsuccessful booster landings, categorized by outcomes like True Ocean (successful ocean landing), False Ocean (unsuccessful ocean landing), True RTLS (successful ground pad landing), False RTLS (unsuccessful ground pad landing), True ASDS (successful drone ship landing), and False ASDS (unsuccessful drone ship landing).

- Here is GitHub URL for Data Wrangling:https://github.com/Zeyad-Shorman/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/3%20-%20Data%20Wrangling.ipynb

TASK 3: Calculate the number and occurence of mission outcome of the orbits

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.

In [15]:
```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

Out[15]:
```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

`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 to a drone ship `False ASDS` means the mission outcome was unsuccessfully landed to a drone ship. `None ASDS` and `None None` these represent a failure to land.

In [16]:
```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

We create a set of outcomes where the second stage did not land successfully:

In [17]:
```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

Out[17]:
```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

EDA with Data Visualization

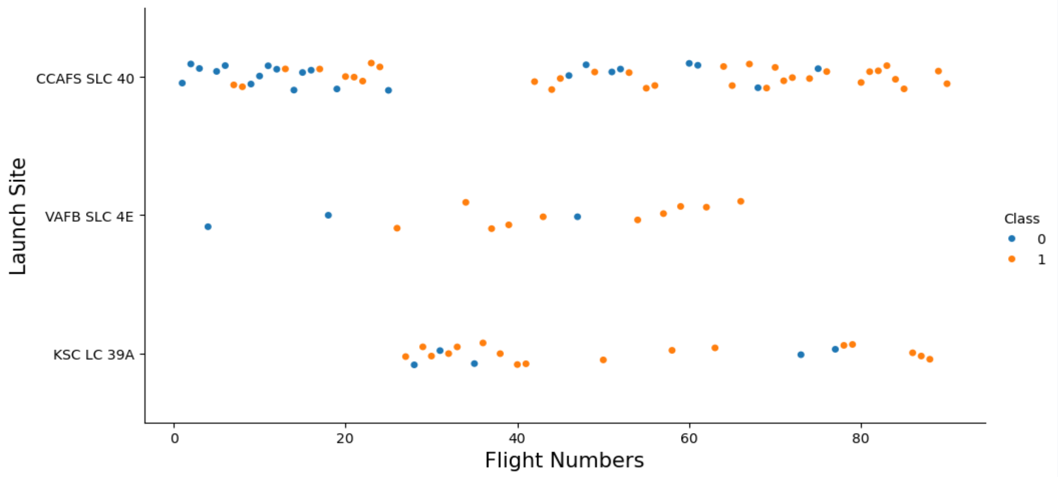
- Predicted if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is due to the fact that SpaceX can reuse the first stage.

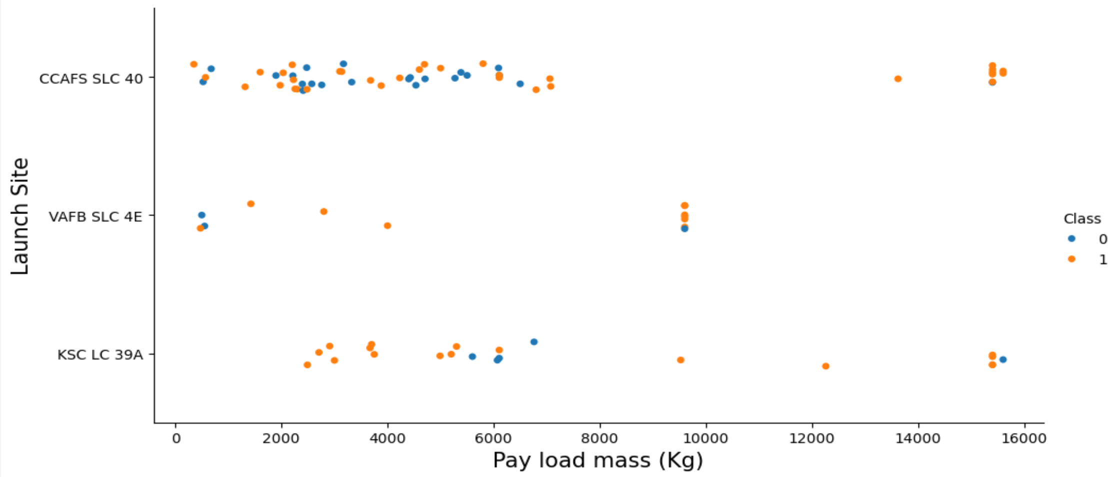
- Perform exploratory Data Analysis and Feature Engineering using Pandas and Matplotli:
  - Exploratory Data Analysis
  - Preparing Data Feature Engineering

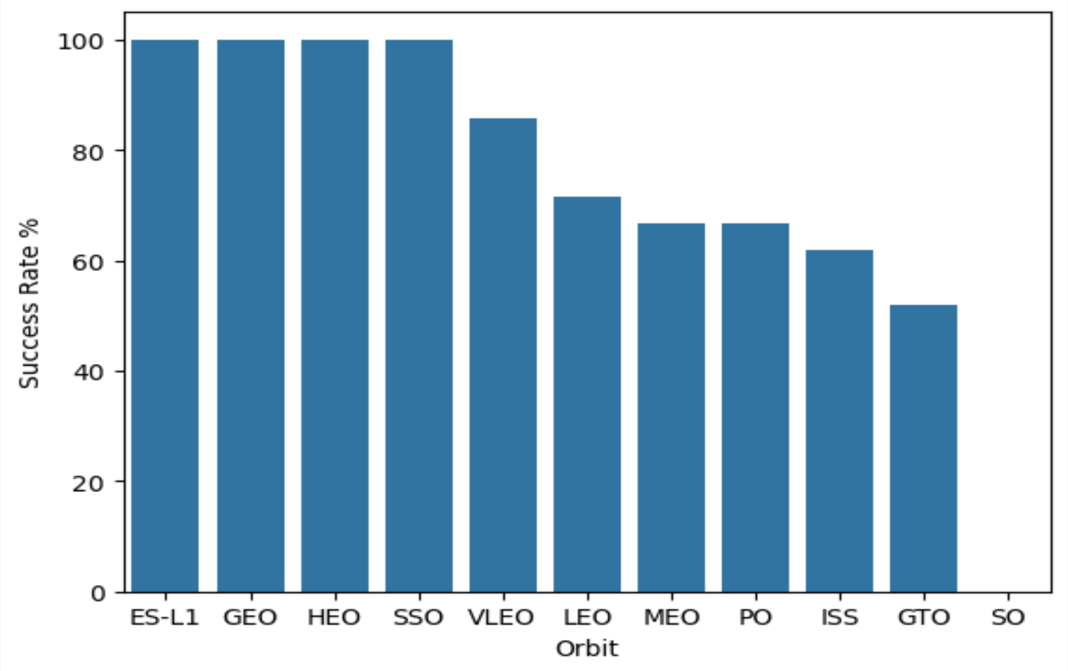
- Here is GitHub URL for Data Visualization:

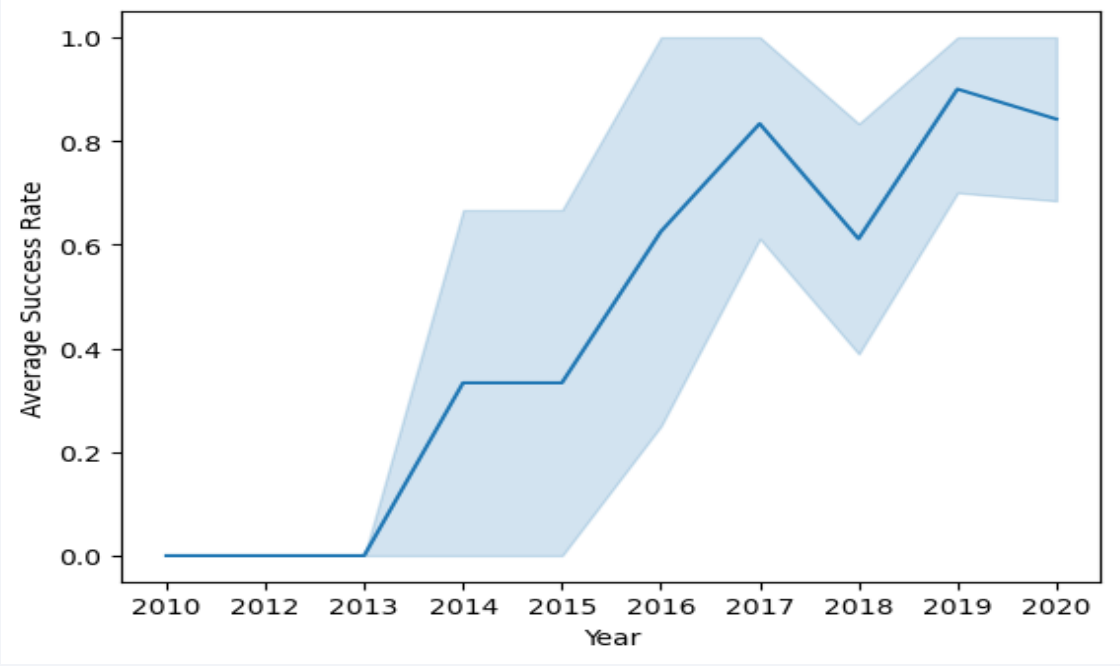
https://github.com/Zeyad-Shorman/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/5%20-%20EDA%20with%20Visualization%20Lab.ipynb

EDA with Data Visualization (Charts...)









EDA with SQL

- EDA SQL Queries:

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

```
%sql select DISTINCT launch_site as "Launch_Sites" from SPACEXTABLE
```

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

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

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

```
%sql select sum(PAYLOAD_MASS__KG_) as 'Total Payload Mass (Kg)',Customer  from SPACEXTABLE where Customer ='NASA (CRS)'
```

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

```
%sql select avg(PAYLOAD_MASS__KG_) as 'Average Payload Mass', Booster_Version from SPACEXTABLE where Booster_Version like 'F9 v1.1%'
```

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

```
%sql select min(Date) from SPACEXTABLE where Landing_Outcome  = "Success (ground pad)";
```

EDA with SQL (Cont….)

- EDA SQL Queries:

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

```
%sql select DISTINCT Booster_Version as 'Names of the Boosters', Payload, PAYLOAD_MASS__KG_ as'Payload Mass (Kg)' from SPACEXTABLE
where Landing_Outcome = "Success (drone ship)" and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

- List the total number of successful and failure mission outcomes

```
%sql select Mission_Outcome, count(Mission_Outcome) as 'Total' from SPACEXTABLE group by "Mission_Outcome";
```

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

```
%sql select Booster_Version,PAYLOAD_MASS__KG_ from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015

```
%sql select substr(Date,6,2) as 'Month', substr(Date,0,5) as 'Year', Booster_Version, Launch_Site, Mission_Outcome,Landing_Outcome from SPACEXTABLE
where substr(Date,0,5) = '2015' and Landing_Outcome = 'Failure (drone ship)'
```

- 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 * from SPACEXTABLE where Landing_Outcome like 'Success%' and (date between '2010-06-04' and '2017-03-20') ORDER BY Date DESC
```

- GitHub URL for Completed EDA with SQL.

https://github.com/Zeyad-Shorman/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/4%20-%20Complete%20the%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Generate a Folium Map marking all launch sites and use map elements such as markers, circles, and lines to denote the success or failure of launches at each site.

- Established a launch outcome set where Failure is represented as (0) and Success as (1).

- Here is GitHub URL of the completed interactive map with Folium map, as an external reference and peer-review purpose:

- https://github.com/Zeyad-Shorman/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/6%20%20Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

- build a Plotly Dash application for users to perform interactive visual analytics on SpaceX launch data in real-time.

  - Add a Launch Site Drop-down Input Component

  - Add a callback function to render success-pie-chart based on selected site dropdown

  - Add a Range Slider to Select Payload

  - Add a callback function to render the success-payload-scatter-chart scatter plot

- Here is my GitHub URL of the completed Plotly Dash lab, as an external reference and peer-review purpose
- \- https://github.com/Zeyad-Shorman/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/7%20%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.ipynb.py

Predictive Analysis (Classification)

- Summarize of how i built, evaluated, improved, and found the best performing classification model
- After Loading the data I Perform exploratory Data Analysis and determine Training Labels:

  - Create a NumPy array from the column Class in data, by applying the method to_numpy() then assign it to the variable Y

  - Standardize the data in X then reassign it to the variable X

  - Use the function train_test_split to split the data X and Y into training and test data. Set the parameter test_size to 0.2 and random_state to 2.

Predictive Analysis (Classification) Cont ….

- Create a logistic regression object then create a GridSearchCV to find the best parameters from the dictionary parameters.

- Create a support vector machine object then create a GridSearchCV to find the best parameters from the dictionary parameters.

- Create a decision tree classifier object then create a GridSearchCV to find the best parameters from the dictionary parameters.

- finally i used the method score to fine the accuracy on the test data for each model

- Here is my GitHub URL of the completed predictive analysis lab, as an external reference and peer-review purpose

  https://github.com/Zeyad-Shorman/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/main/8%20-%20Complete%20the%20Machine%20Learning%20Prediction%20lab.ipynb

Results

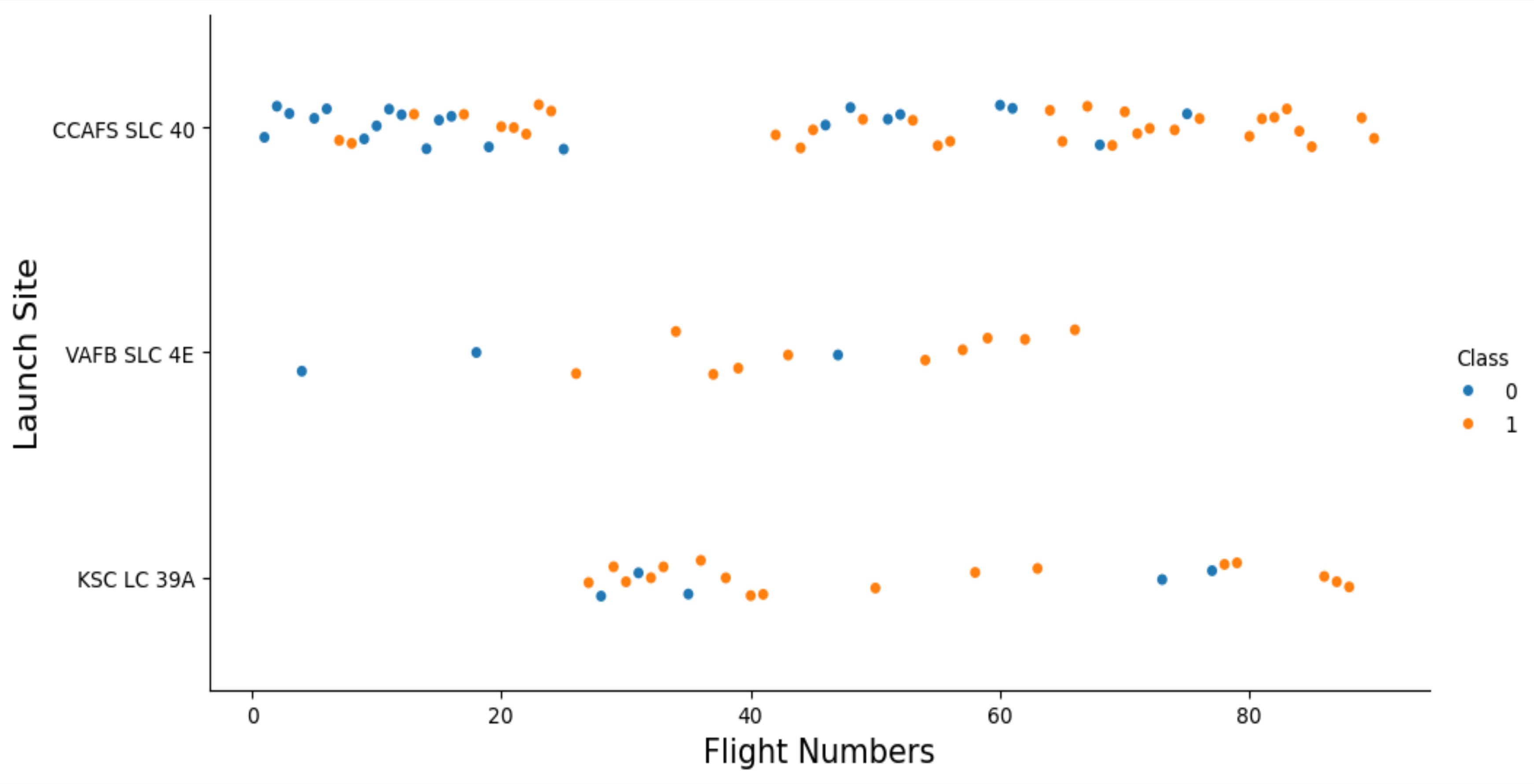
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

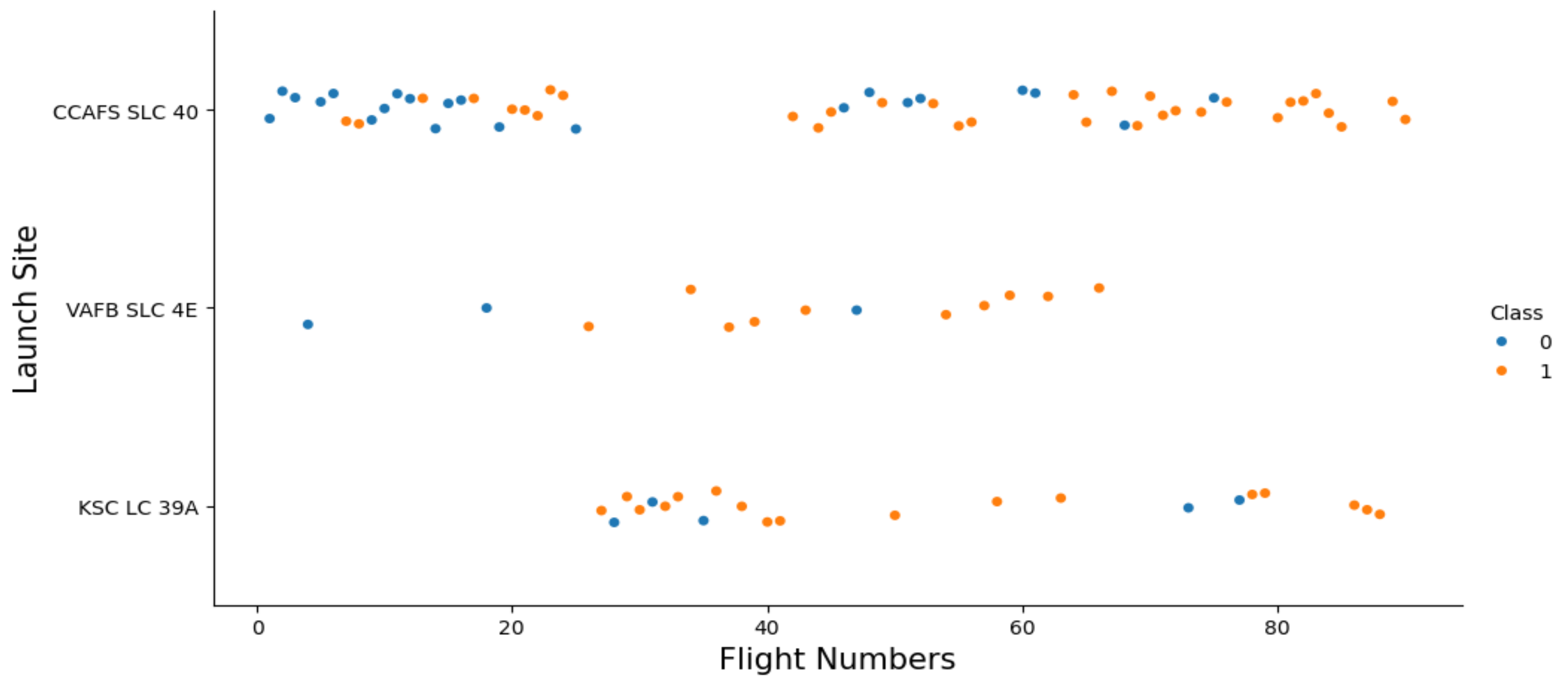
Flight Number vs. Launch Site

Show a scatter plot of Flight Number vs. Launch Site



Flight Number vs. Launch Site (Cont…)

Show the screenshot of the scatter plot with explanations

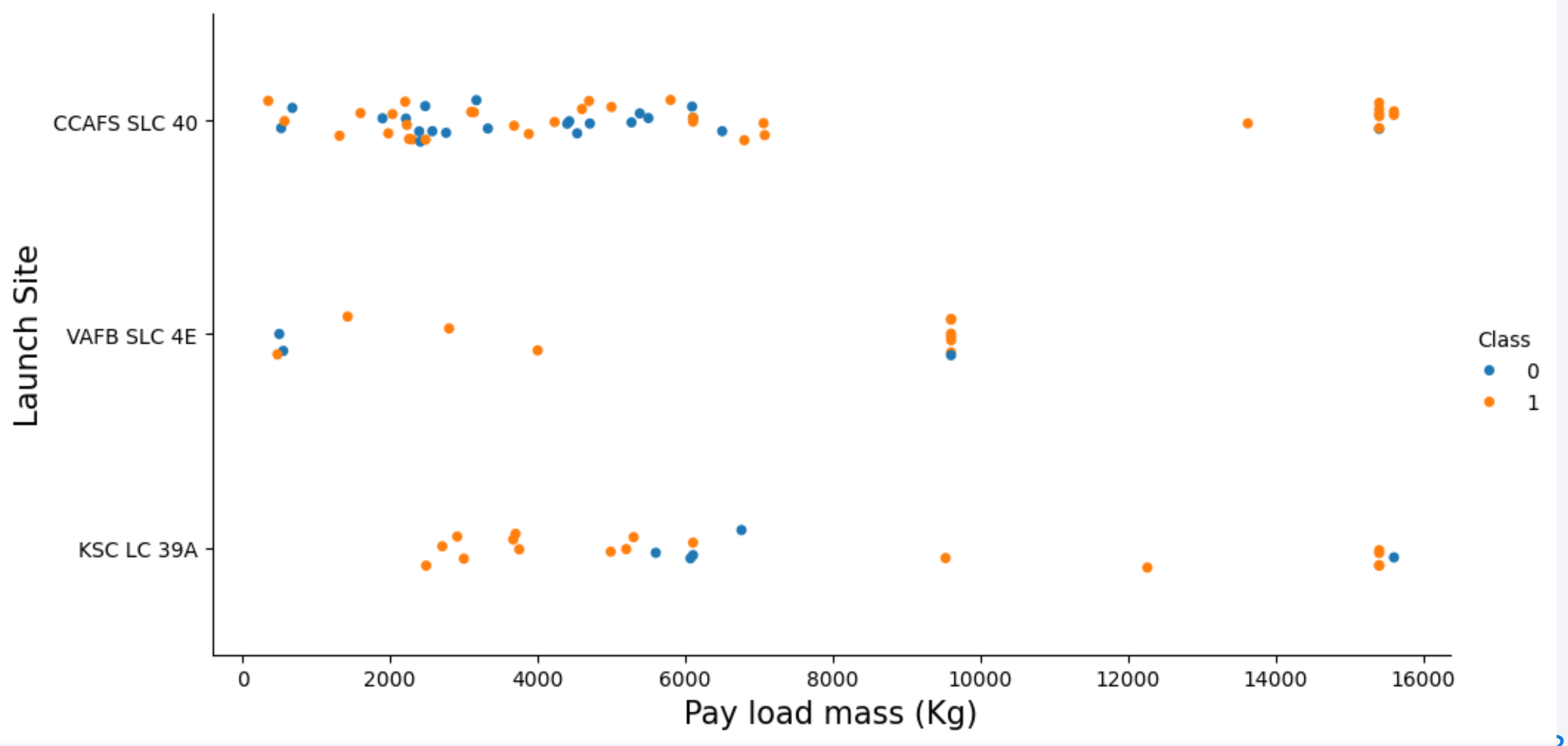


Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

We can explain that whenb the flights number increases the success rate increases. We can see in the chart that (CCAFS SLC 40 Launch site) all last flights were success and same as (VAFB SLC 4E) and (KSC LC 39A) both success rate were 100% after 50 flights and 80 flights

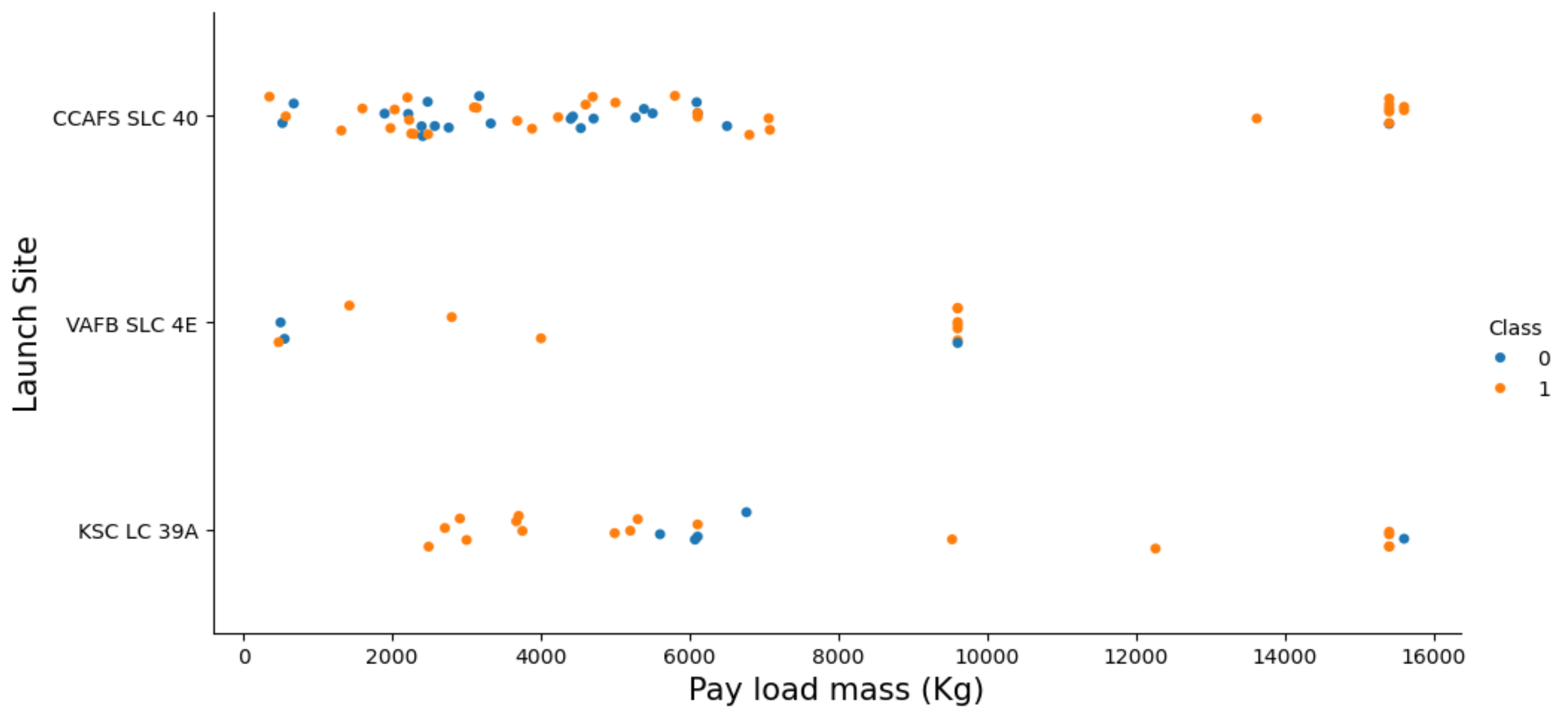
Payload vs. Launch Site

Show a scatter plot of Payload vs. Launch Site



Payload vs. Launch Site (Cont…)

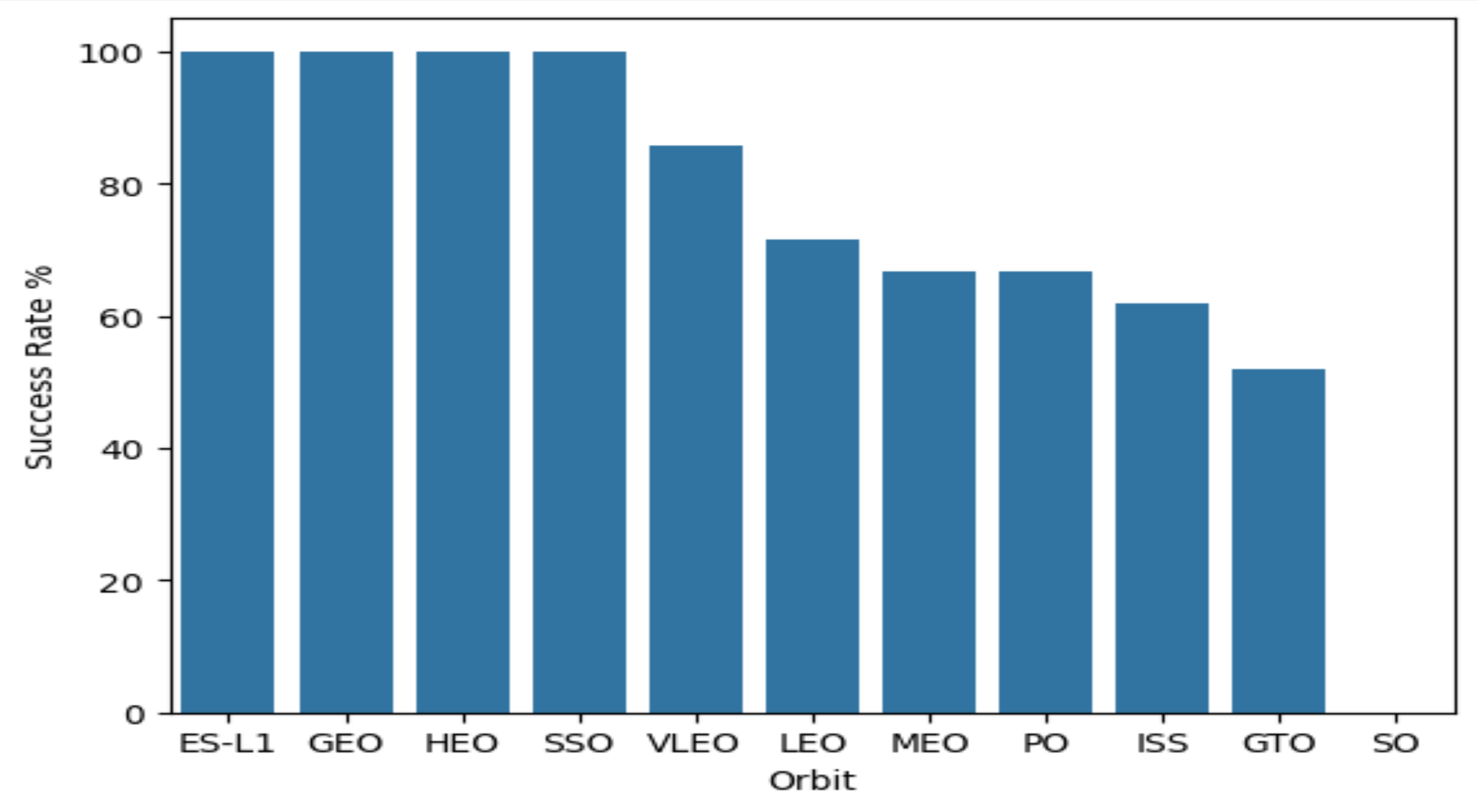
Show the screenshot of the scatter plot with explanations



Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

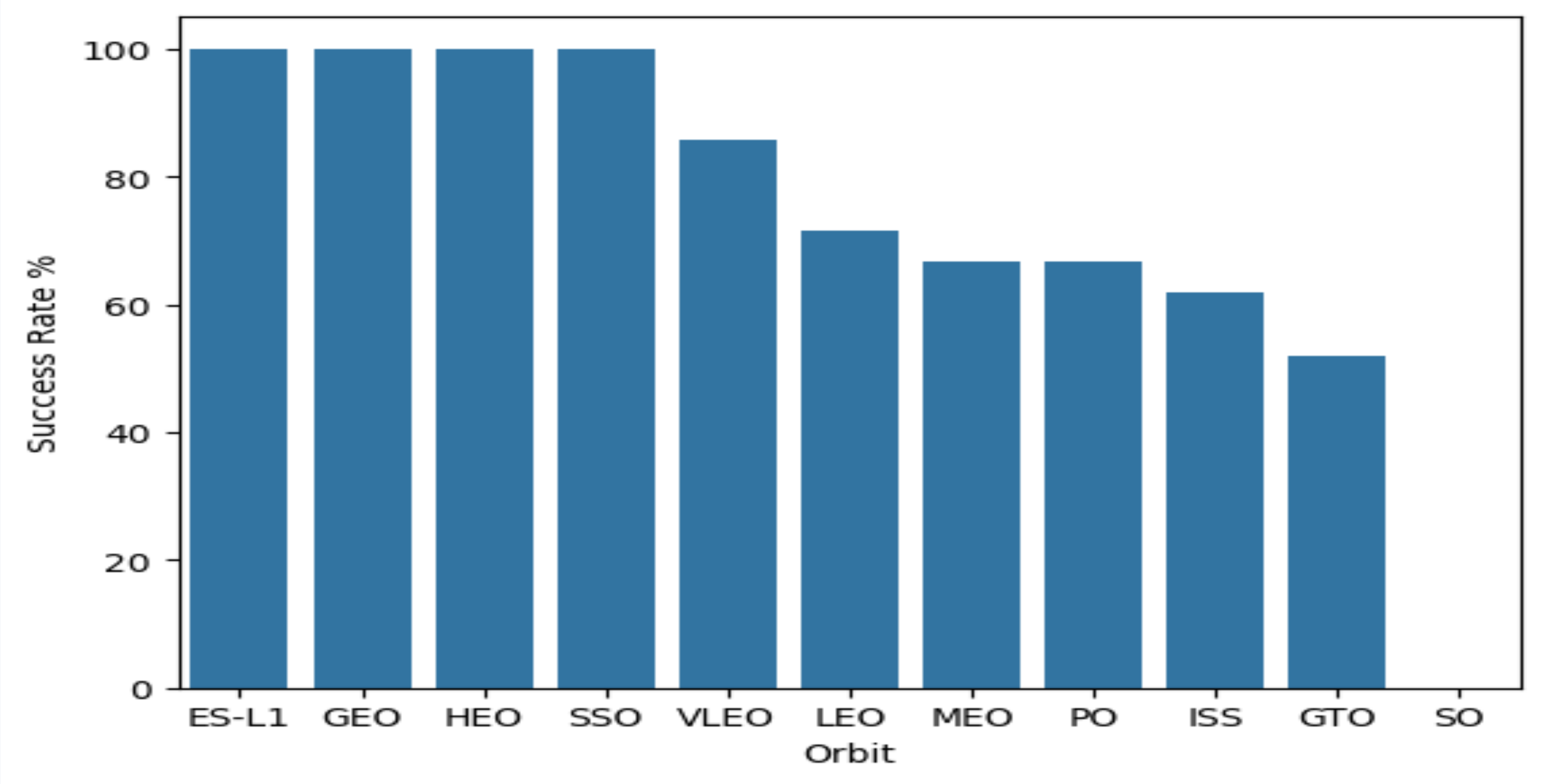
Success Rate vs. Orbit Type

Show a bar chart for the success rate of each orbit type



Success Rate vs. Orbit Type (Cont…)

Show the screenshot of the scatter plot with explanations

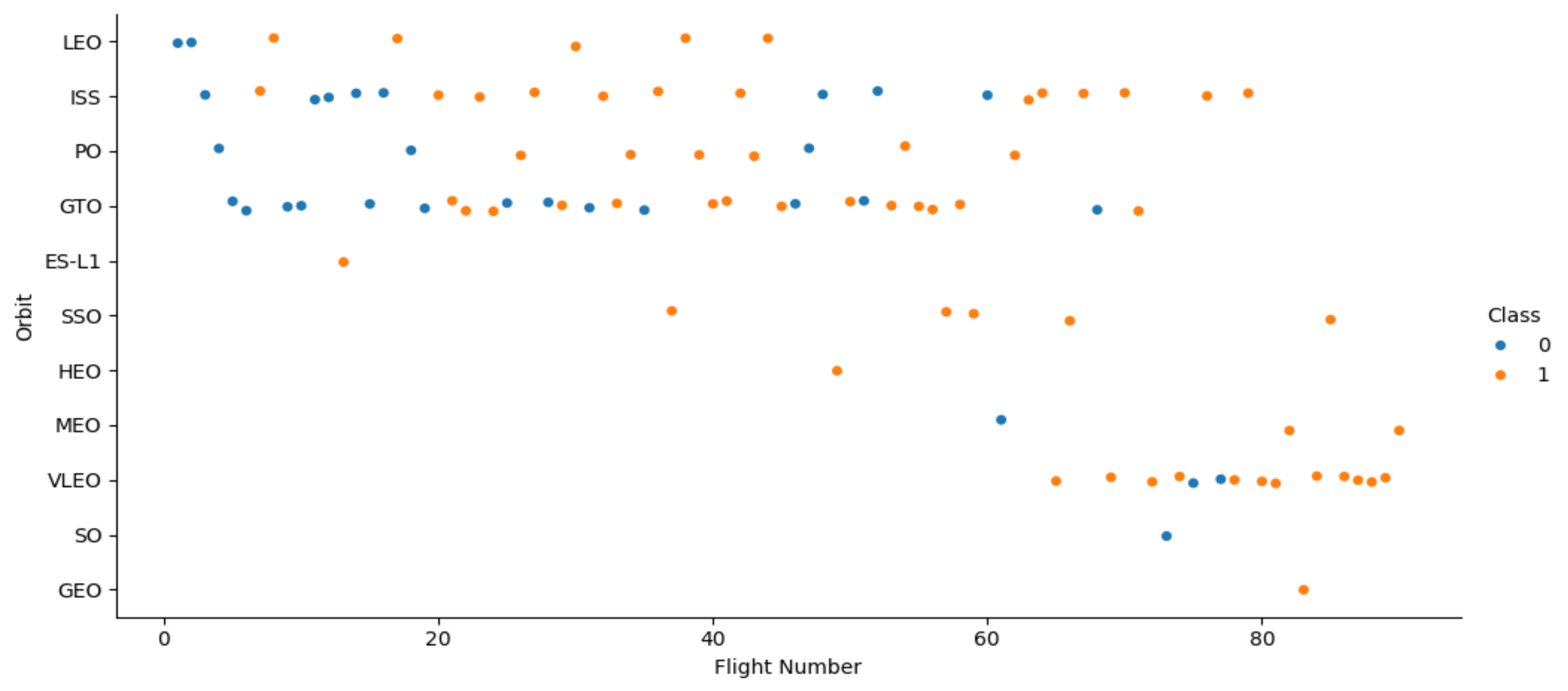


Analyze the ploted bar chart try to find which orbits have high sucess rate.

Based on the Chart. ES-L1,GEO,HEO and SSO are the orbits who have a high success rate with 100% success rate for each

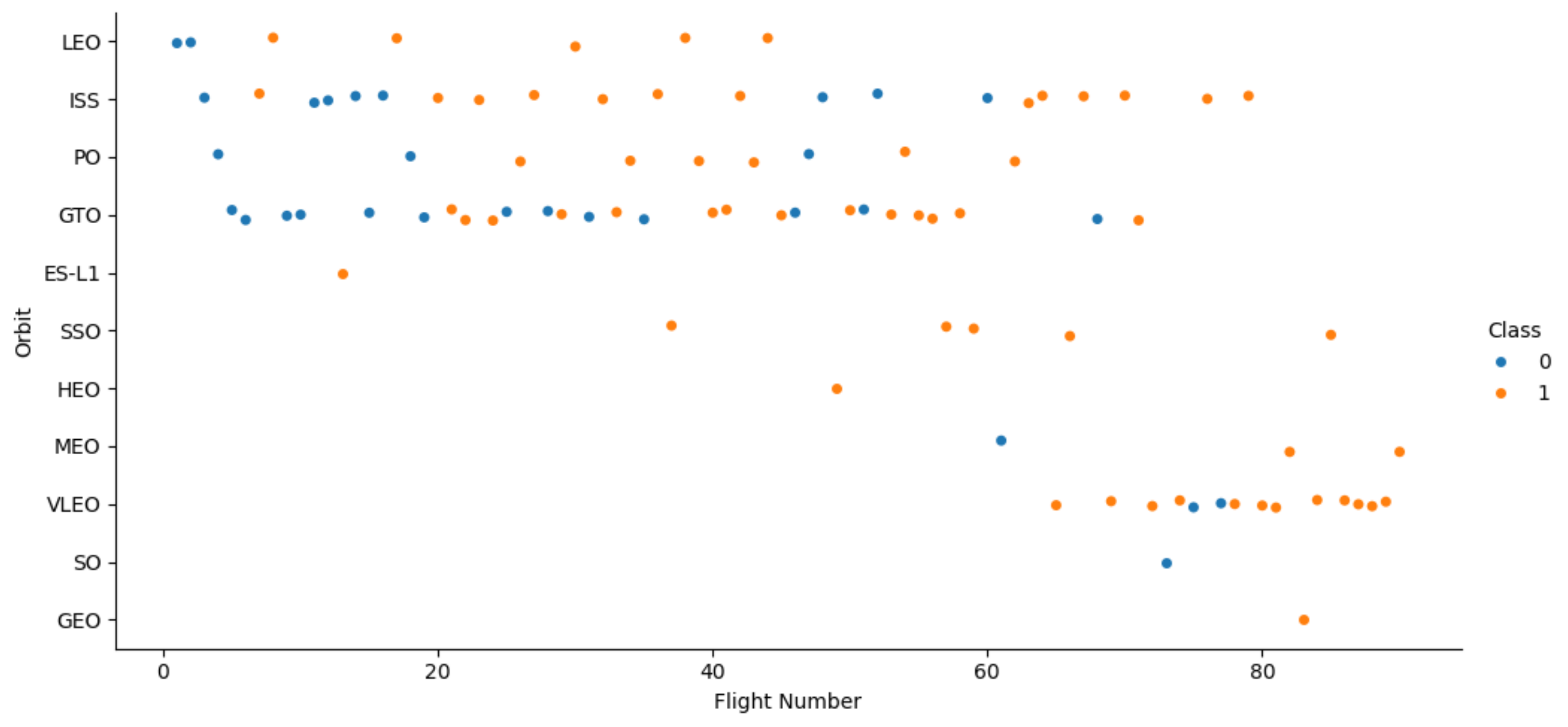
Flight Number vs. Orbit Type

Show a scatter point of Flight number vs. Orbit type



Flight Number vs. Orbit Type (Cont…)

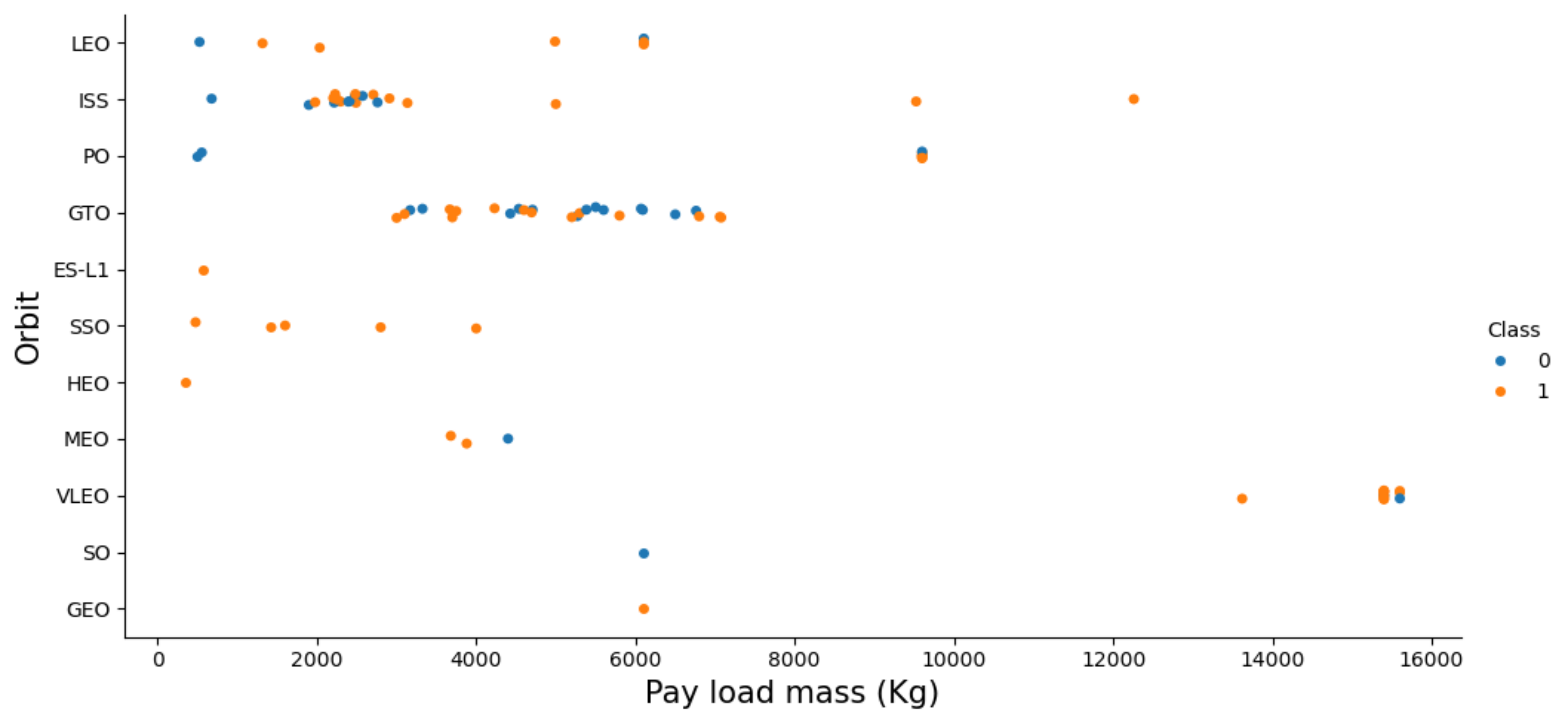
Show the screenshot of the scatter plot with explanations



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

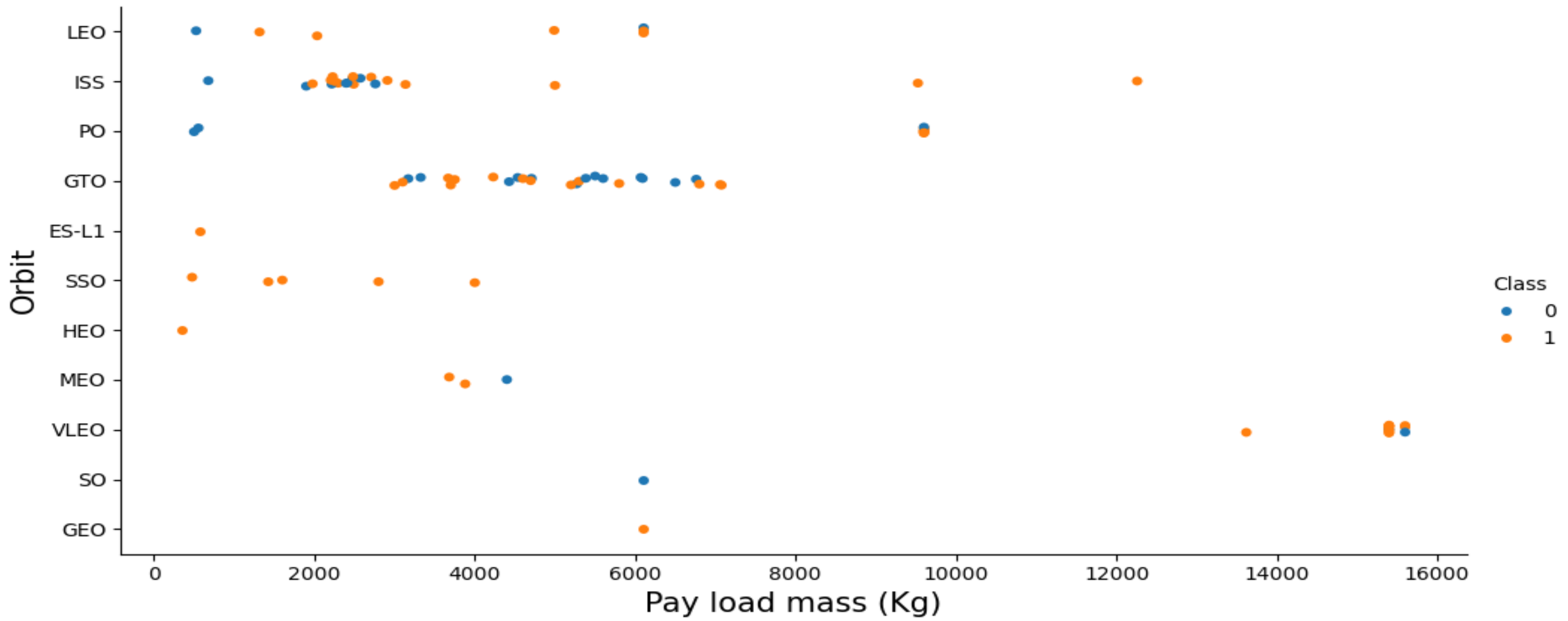
Payload vs. Orbit Type

Show a scatter point of payload vs. orbit type



Payload vs. Orbit Type (Cont…)

Show the screenshot of the scatter plot with explanations

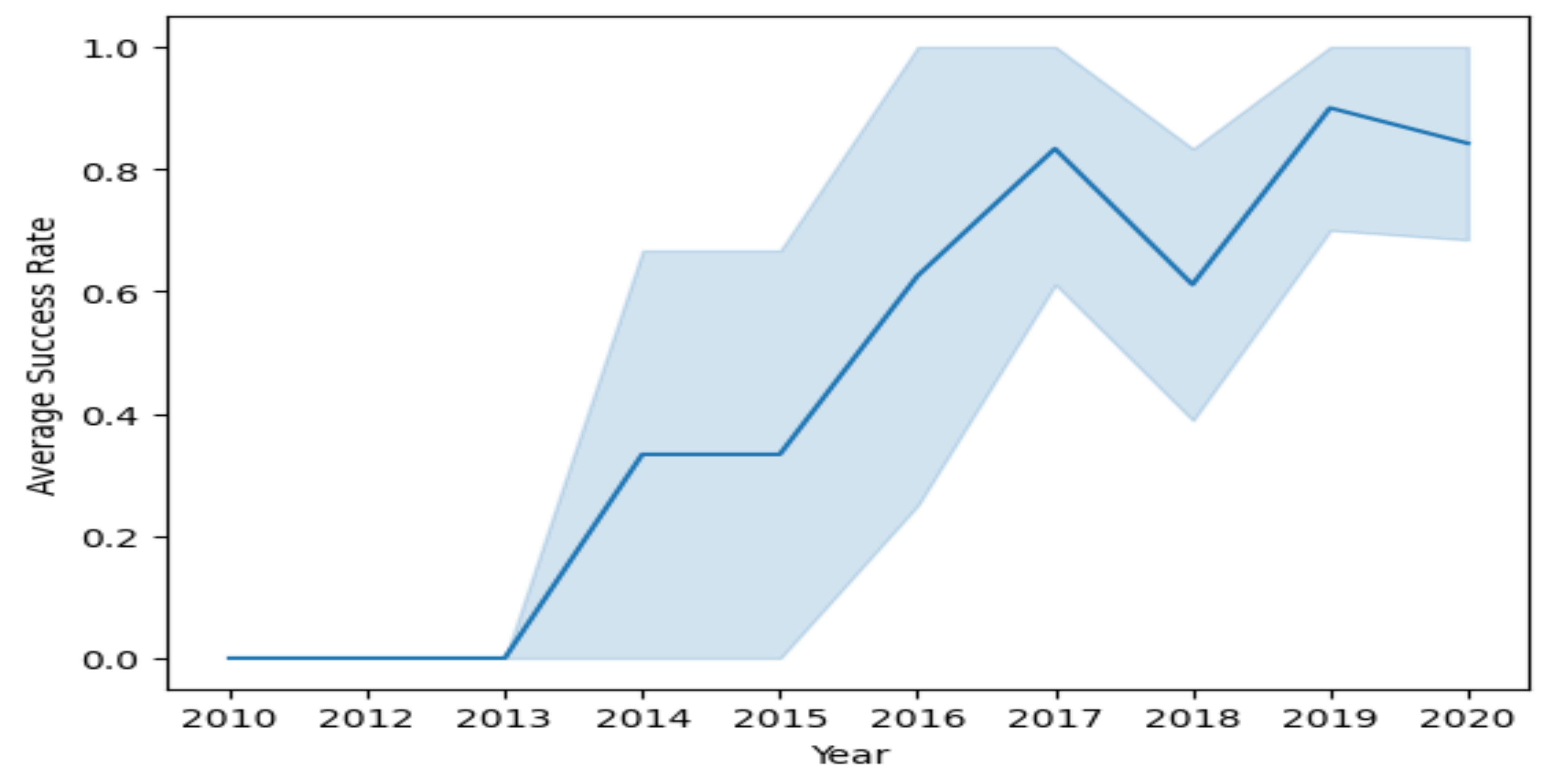


With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

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

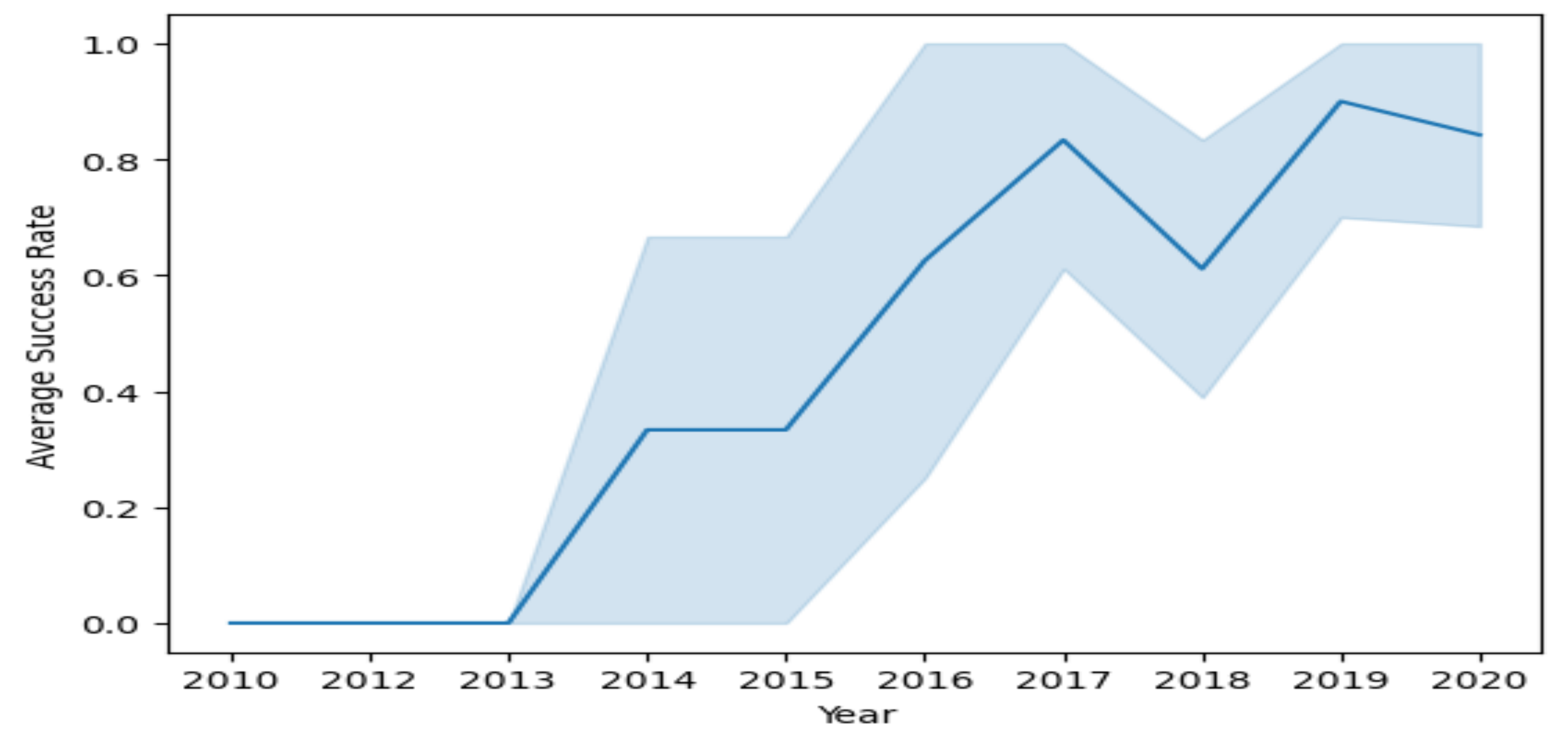
Launch Success Yearly Trend

Show a line chart of yearly average success rate



Launch Success Yearly Trend (Cont…)

Show the screenshot of the scatter plot with explanations



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

All Launch Site Names

- Find the names of the unique launch sites

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

- Present your query result with a short explanation here

```
%sql select DISTINCT launch_site as "Launch_Sites" from SPACEXTABLE
```

```
 * sqlite:///my_data1.db
Done.
```

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

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

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

```
* sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (p
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 (p
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

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

```
]: %sql select sum(PAYLOAD_MASS__KG_) as 'Total Payload Mass (Kg)',Customer  from SPACEXTABLE where Customer ='NASA
```

```
* sqlite:///my_data1.db
Done.
```

Total Payload Mass (Kg)	Customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

```
%sql select avg(PAYLOAD_MASS__KG_) as 'Average Payload Mass', Booster_Version from SPACEXTABLE where Booster_Version like 'F9 v1.1%'
```

Average Payload Mass	Booster_Version
2534.6666666666665	F9 v1.1 B1003

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

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

Hint:Use min function

```
%sql select min(Date) from SPACEXTABLE where Landing_Outcome  = "Success (ground pad)";
```

```
 * sqlite:///my_data1.db
Done.
```

min(Date)
2015-12-22

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

Names of the Boosters	Payload	Payload Mass (Kg)
F9 FT B1022	JCSAT-14	4696
F9 FT B1026	JCSAT-16	4600
F9 FT B1021.2	SES-10	5300
F9 FT B1031.2	SES-11 / EchoStar 105	5200

- Present your query result with a short explanation here

```
%sql select DISTINCT Booster_Version as 'Names of the Boosters', Payload, PAYLOAD_MASS__KG_ as'Payload Mass (Kg)' from SPACEXTABLE
where Landing_Outcome = "Success (drone ship)" and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

Mission_Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- Present your query result with a short explanation here

```
%sql select Mission_Outcome, count(Mission_Outcome) as 'Total' from SPACEXTABLE group by "Mission_Outcome";
```

Boosters Carried Maximum Payload

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

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

- Present your query result with a short explanation here

```
%sql select Booster_Version,PAYLOAD_MASS__KG_ from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

2015 Launch Records

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

Month	Year	Booster_Version	Launch_Site	Mission_Outcome	Landing_Outcome
01	2015	F9 v1.1 B1012	CCAFS LC-40	Success	Failure (drone ship)
04	2015	F9 v1.1 B1015	CCAFS LC-40	Success	Failure (drone ship)

- Present your query result with a short explanation here

```
%sql select substr(Date,6,2) as 'Month', substr(Date,0,5) as 'Year', Booster_Version, Launch_Site, Mission_Outcome,Landing_Outcome from SPACEXTABLE
where substr(Date,0,5) = '2015' and Landing_Outcome = 'Failure (drone ship)'
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

- Present your query result with a short explanation here

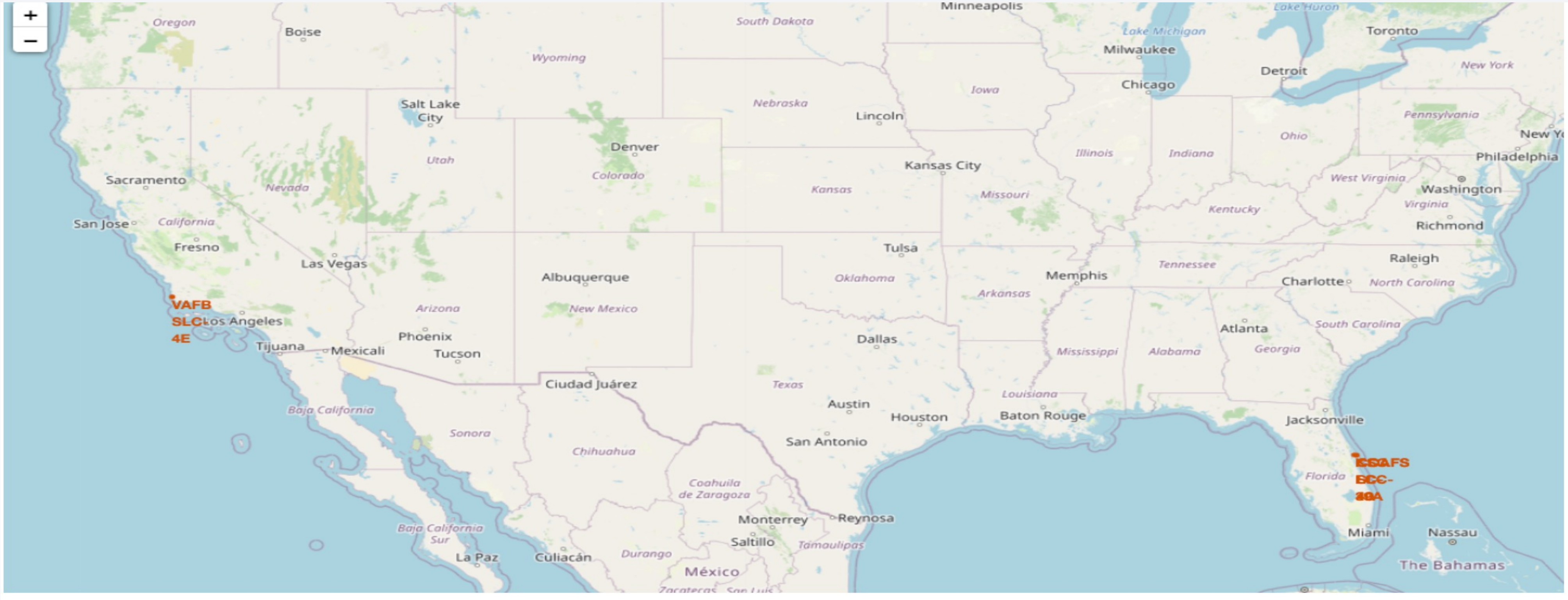
```
%sql select * from SPACEXTABLE where Landing_Outcome like 'Success%' and (date between '2010-06-04' and '2017-03-20') ORDER BY Date DESC
```

Section 3

Launch Sites Proximities Analysis

all launch sites marked on a map

- Explore the generated folium map and make a proper screenshot to include all launch sites’ location markers on a global map



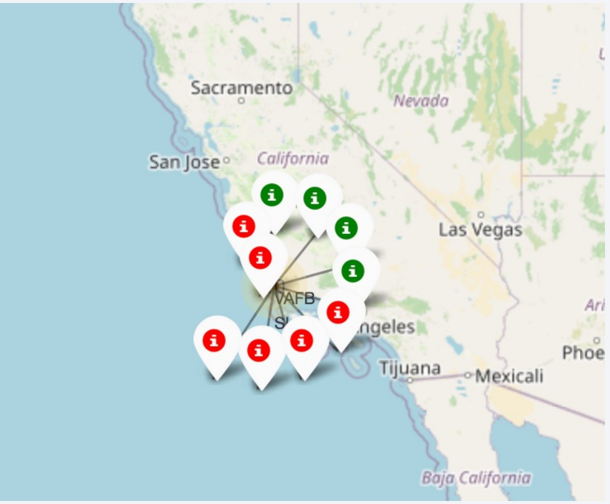
all launch sites marked on a map(cont …)

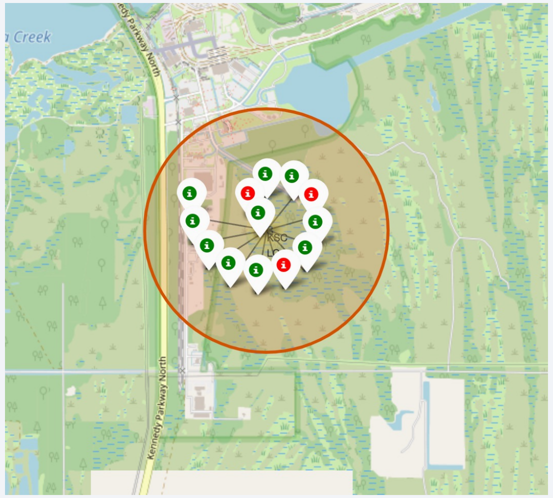
- Explain the important elements and findings on the screenshot

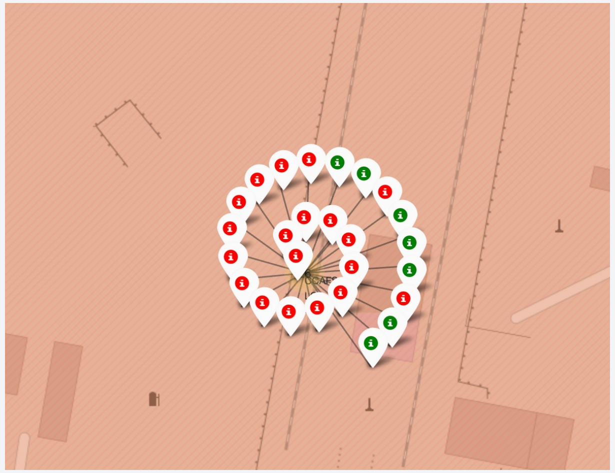
	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

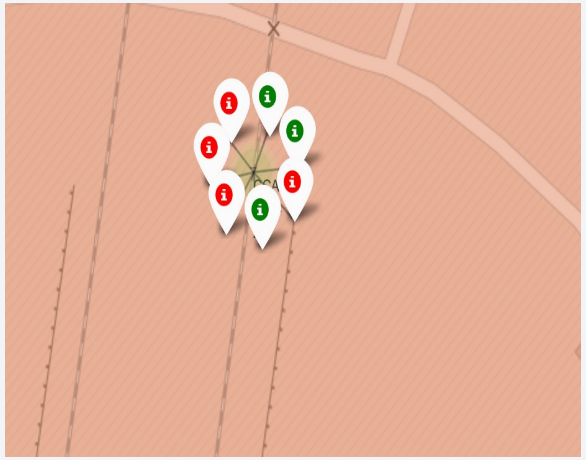
Success/failed launches for each site on the map

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map









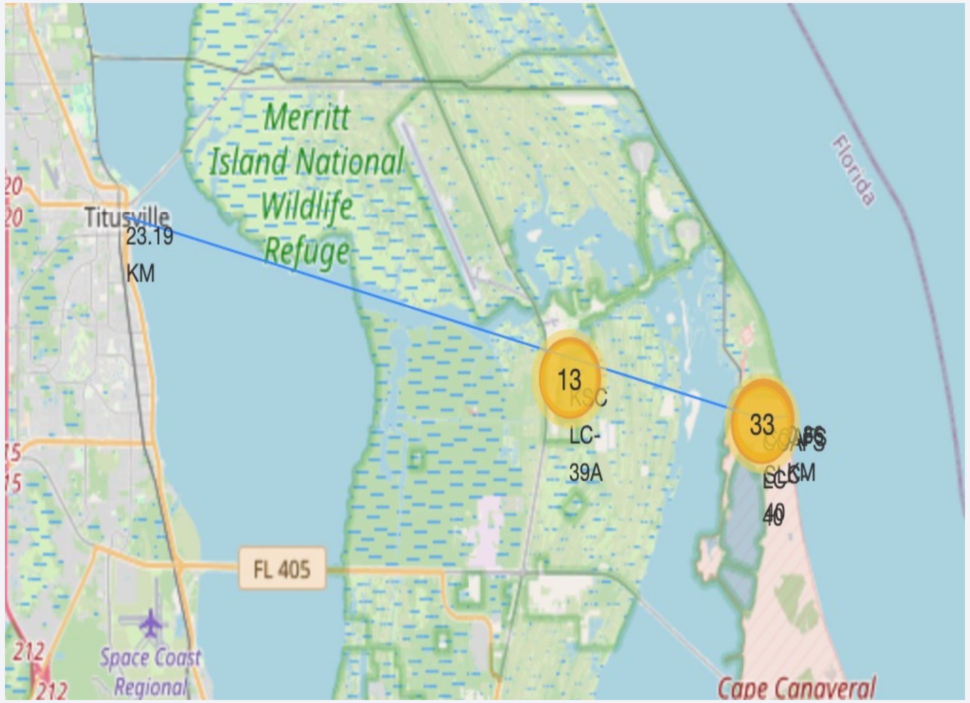
Success/failed launches for each site on the map(cont...)

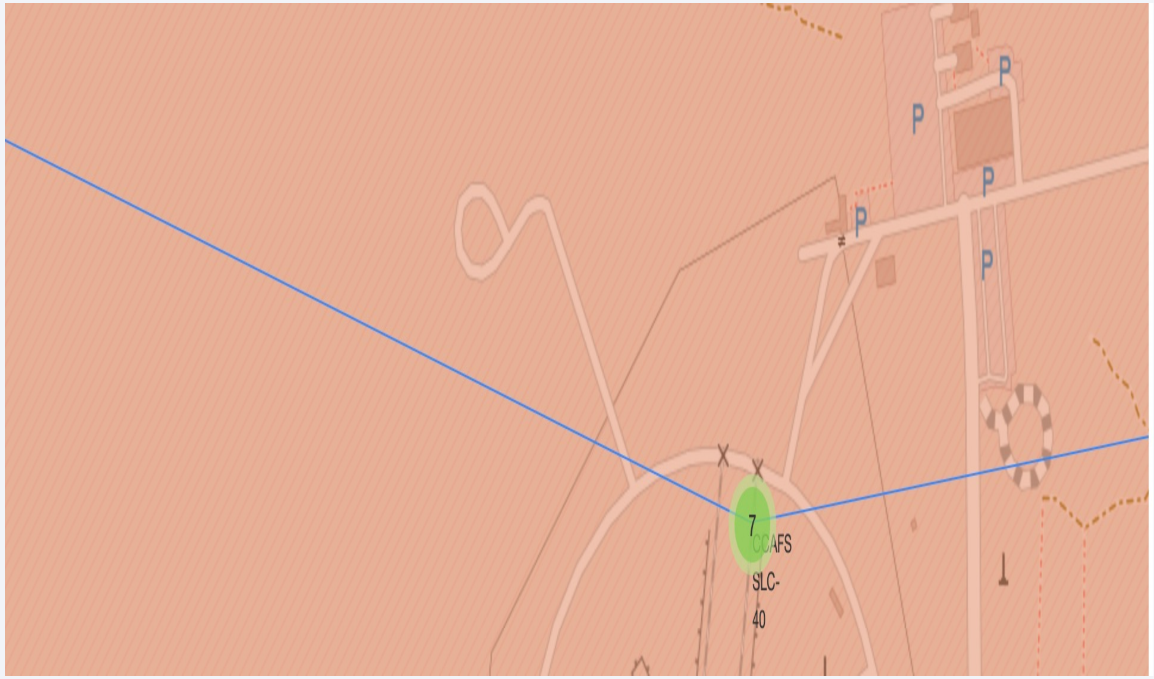
Explain the important elements and findings on the screenshot

	Launch Site	Lat	Long	class	marker_color
46	KSC LC-39A	28.573255	-80.646895	1	green
47	KSC LC-39A	28.573255	-80.646895	1	green
48	KSC LC-39A	28.573255	-80.646895	1	green
49	CCAFS SLC-40	28.563197	-80.576820	1	green
50	CCAFS SLC-40	28.563197	-80.576820	1	green
51	CCAFS SLC-40	28.563197	-80.576820	0	red
52	CCAFS SLC-40	28.563197	-80.576820	0	red
53	CCAFS SLC-40	28.563197	-80.576820	0	red
54	CCAFS SLC-40	28.563197	-80.576820	1	green
55	CCAFS SLC-40	28.563197	-80.576820	0	red

Calculate the distances between a launch site to its proximities

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed





Thank you!