IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mehulkumar Shah
08/13/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  Space X 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 Space X 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 space X for a rocket launch This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - Dropping unnecessary columns which are not playing any role in analysis
  - One hot encoding was applied to categorical features.
- 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

- The data was collected using various methods
- Data collection was done using get request to the SpaceX API
- Next, I decoded the response content as a JSON using json() function call and turn it into a pandas dataframe using json_normalize
- I have the data, checked for missing values and fill in missing values where necessary
- In addition, I have performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup
- The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis

Data Collection – SpaceX API

- I used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebooks is : https://github.com/itismshah/applied-data-science-capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

1. Get request for rocket launch data using API

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

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

2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

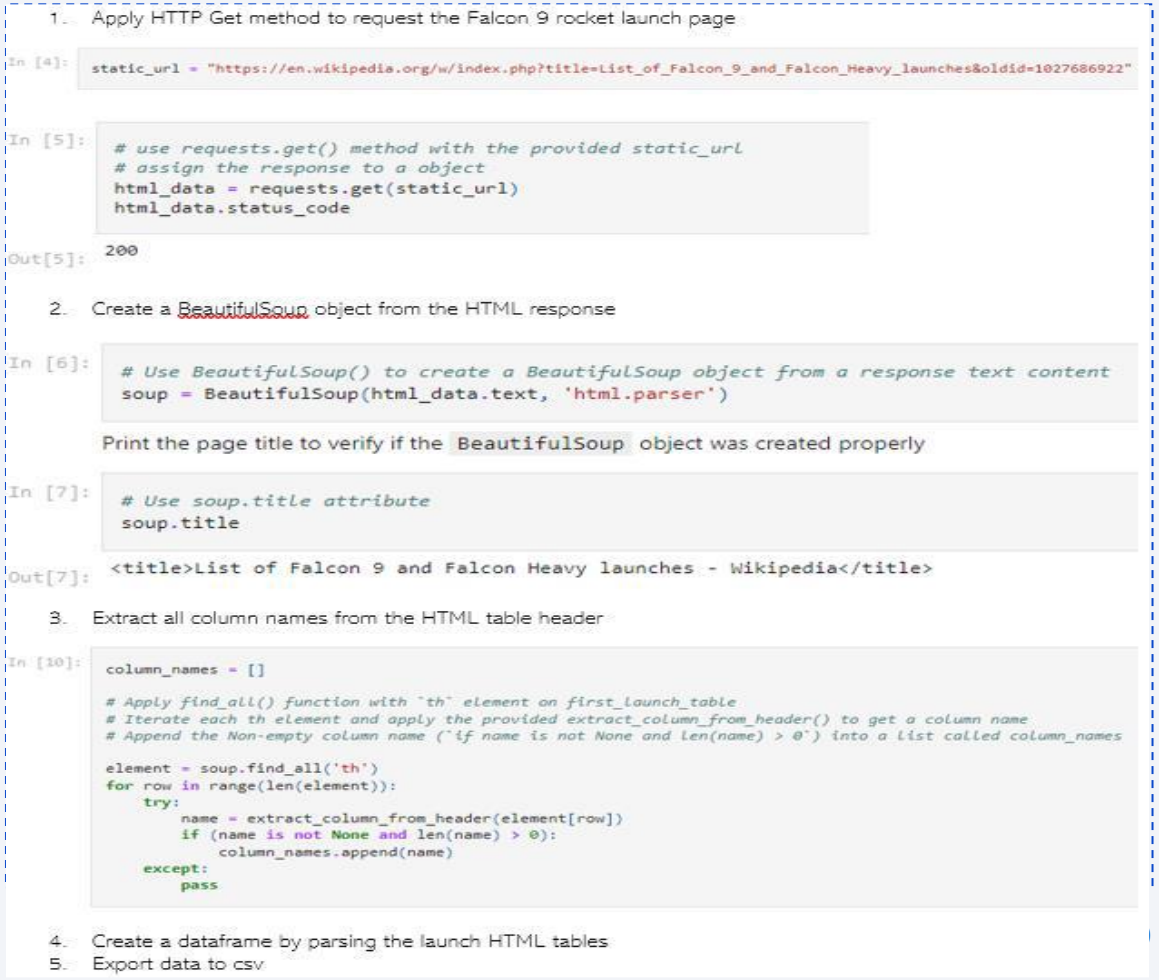
In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```

Data Collection - Scraping

- I have applied web scrapping to web scrap Falcon 9 launch records with BeautifulSoup
- I did parsed the table and converted it into a pandas DataFrame.
- The link to the GitHub is : https://github.com/itismshah/applied-data-science-capstone/blob/main/jupyter-labs-webscraping.ipynb



1. Apply HTTP Get method to request the Falcon 9 rocket launch page

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]:
```
200
```

2. Create a BeautifulSoup object from the HTML response

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

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

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Extract all column names from the HTML table header

In [10]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

Data Wrangling

- In the dataset, there are several cases where the booster did not land successfully.
  - True Ocean, True RTLS, True ASDS means the mission has been successful.
  - False Ocean, False RTLS, False ASDS means the mission was a failure.
- We need to transform string variables into categorical variables where 1 means the mission has been successful and 0 means the mission was a failure.
- The link to the notebook is : https://github.com/itismshah/applied-data-science-capstone/blob/main/jupyter-labs-jupyter-spacex-Data%20wrangling.ipynb

```
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

```
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
SO        1
ES-L1     1
HEO       1
GEO       1
Name: Orbit, dtype: int64
```

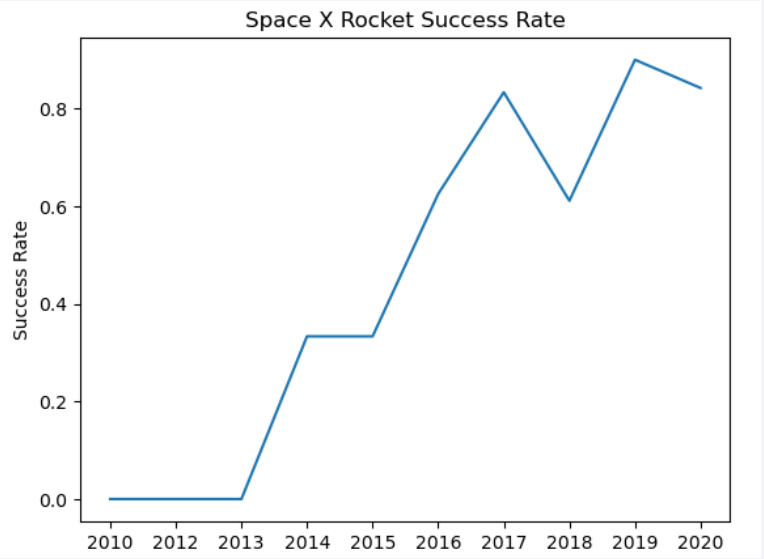
```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

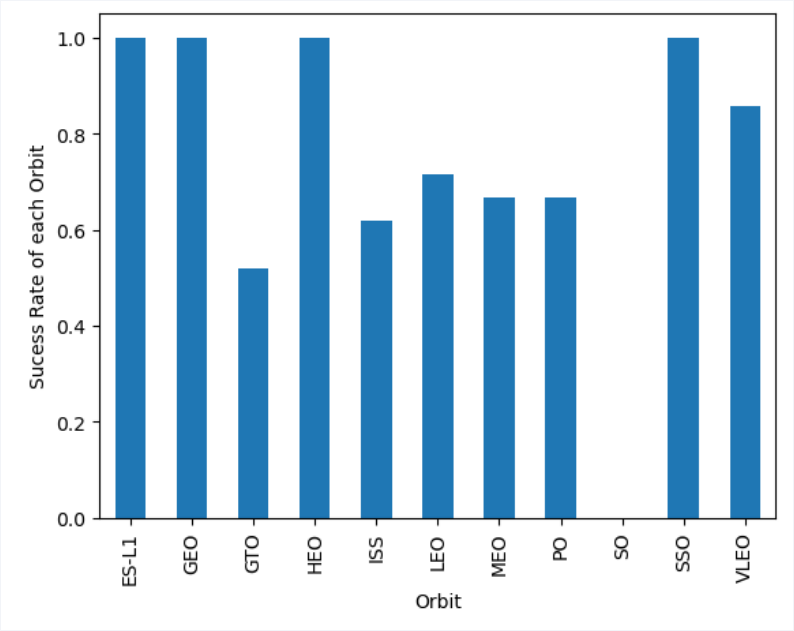
```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

EDA with Data Visualization

- I have explored the data by various combinations as follow:
- Flight number vs Payload Mass
- Flight number vs Launch Site
- Payload vs Launch Site etc.





- The link to the notebook is : https://github.com/itismshah/applied-data-science-capstone/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- I have performed SQL queries to gather and understand data from dataset:
  - Displaying the names of the unique launch sites in the space mission.
  - Display 5 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 records which will display the month names, faiilure landing_ouutcomes in drone ship, booster versions, launch_site forthe months in year 2015.
  - Rank the count of successful landiing_outcomes between the date04-06-2010 and 20-03-2017in descending order.
- The link to the notebook is : https://github.com/itismshah/applied-data-science-capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas
  - Red circle at NASA Johnson Space Center's coordinate with label showing its name(folium.Circle, folium.map.Marker).
  - Red circles at each launch site coordinates with label showing launch site name (folium.Circle, folium.map.Marker, folium.features.DivIcon).
  - The grouping of points in a cluster to display multiple and different information for the same coordinates (folium.plugins.MarkerCluster).
  - Markers to show successful and unsuccessful landings. Greenfor successful landing and Redfor unsuccessful landing. (folium.map.Marker, folium.Icon).
  - Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them. (folium.map.Marker, folium.PolyLine, folium.features.DivIcon)
- These objects are created in order to understand better the problem and the data. I can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.
- The link to the notebook is : https://github.com/itismshah/applied-data-science-capstone/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Dashboard has dropdown,pie chart,rangeslider andscatter plot components
  - Dropdownallows a user to choose the launch site or all launch sites (dash_core_components.Dropdown).
  - Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component(plotly.express.pie).
  - Rangeslider allows a user to select a payload mass in a fixed range (dash_core_components.RangeSlider).
  - Scatter chart showsthe relationship between two variables, in particular Success vs Payload Mass (plotly.express.scatter).

- The link to the notebook is : https://github.com/itismshah/applied-data-science-capstone/blob/main/spacex_dash_app.jpg
- Code https://github.com/itismshah/applied-data-science-capstone/blob/main/spacex_dash_app.py

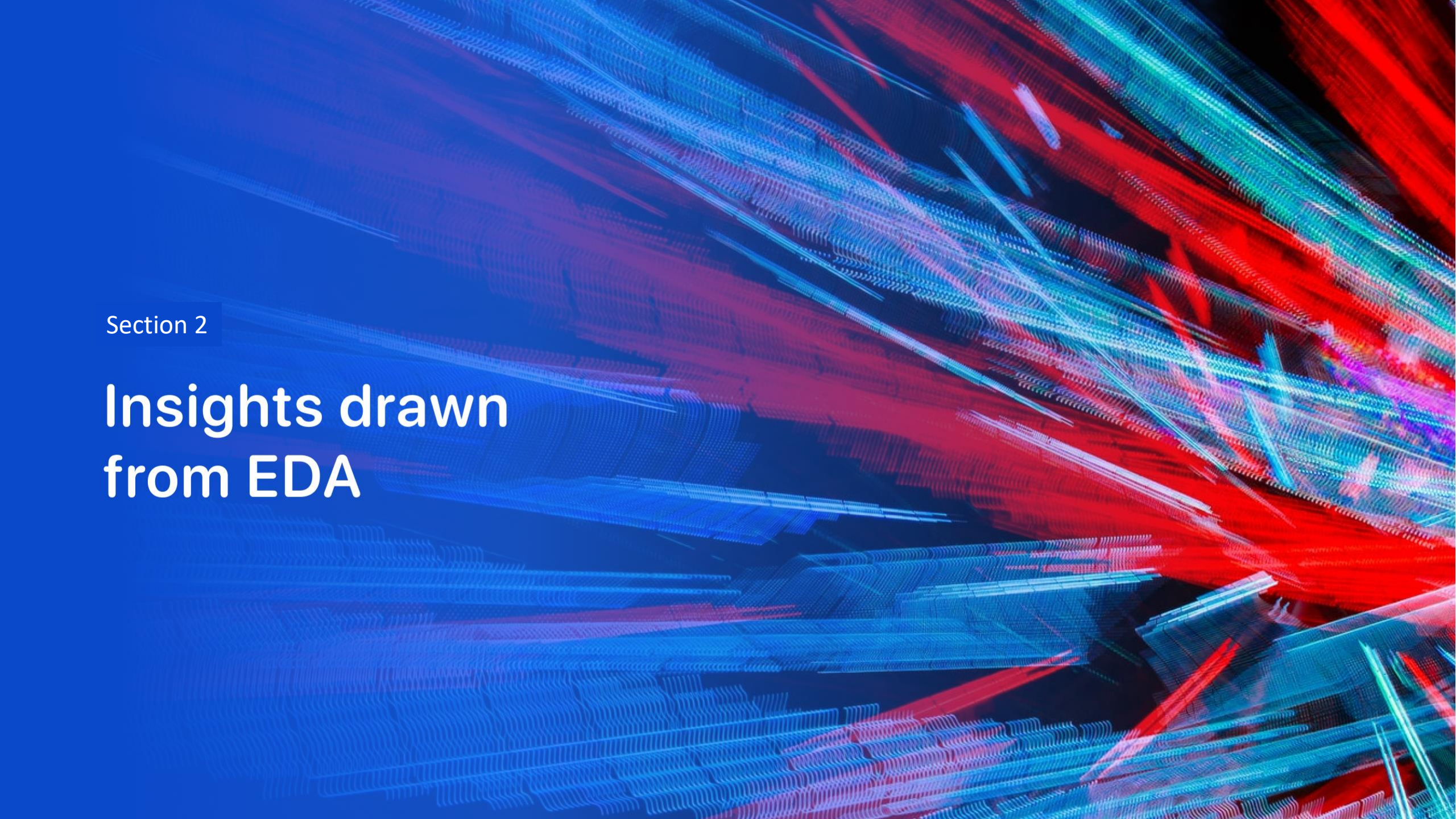
Predictive Analysis (Classification)

- Data preparation
  - Load dataset
  - Normalize data
  - Split data into training and test sets.
- Model preparation
  - Selection of machine learning algorithms
  - Set parameters for each algorithm to GridSearchCV
  - Training GridSearchModel models with training dataset
- Model evaluation
  - Get best hyperparameters for each type of model
  - Compute accuracy for each model with test dataset
  - Plot Confusion Matrix
- Model comparison
  - Comparison of models according to their accuracy
  - The model with the best accuracy will be chosen (see Notebook for result)

- The link to the notebook is: https://github.com/itismshah/applied-data-science-capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

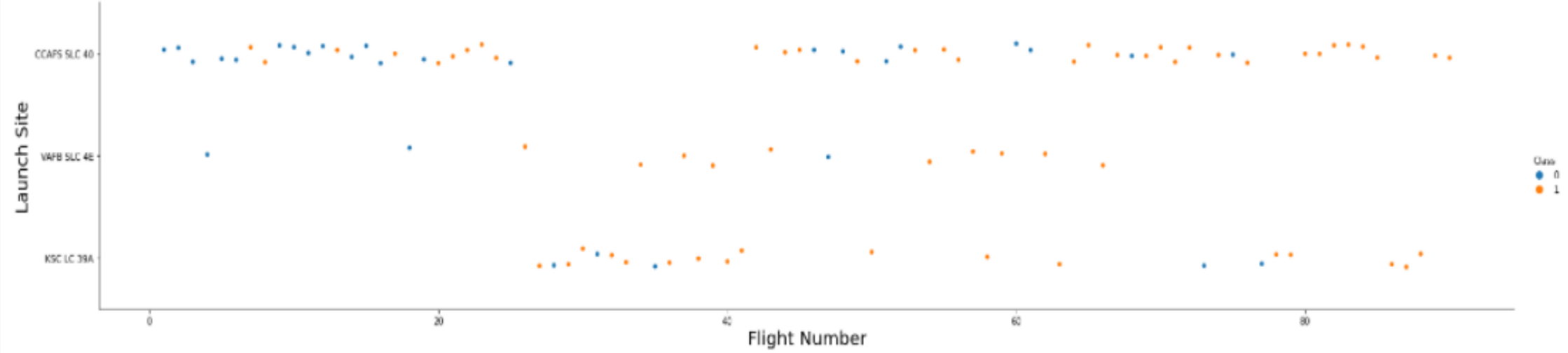
Results

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


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

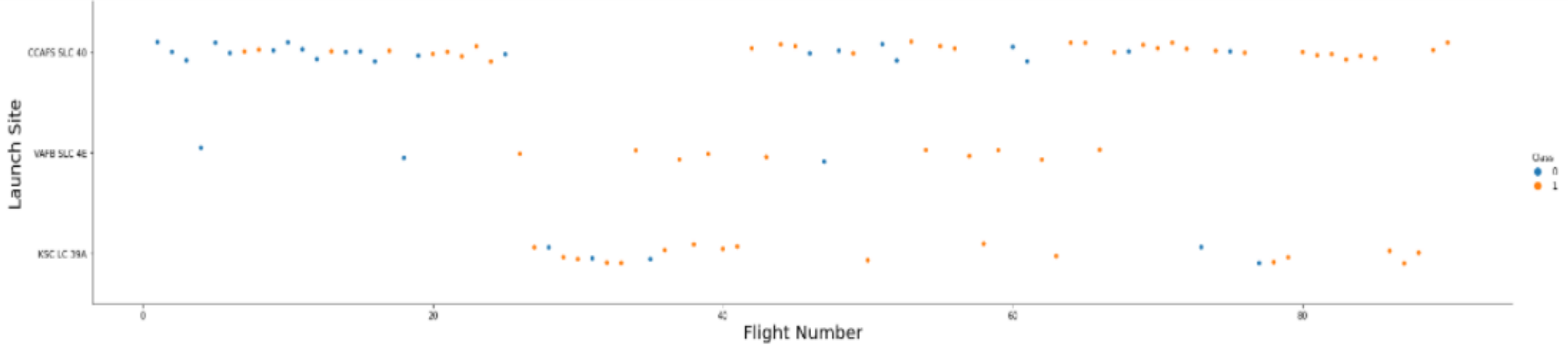
- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



Payload vs. Launch Site

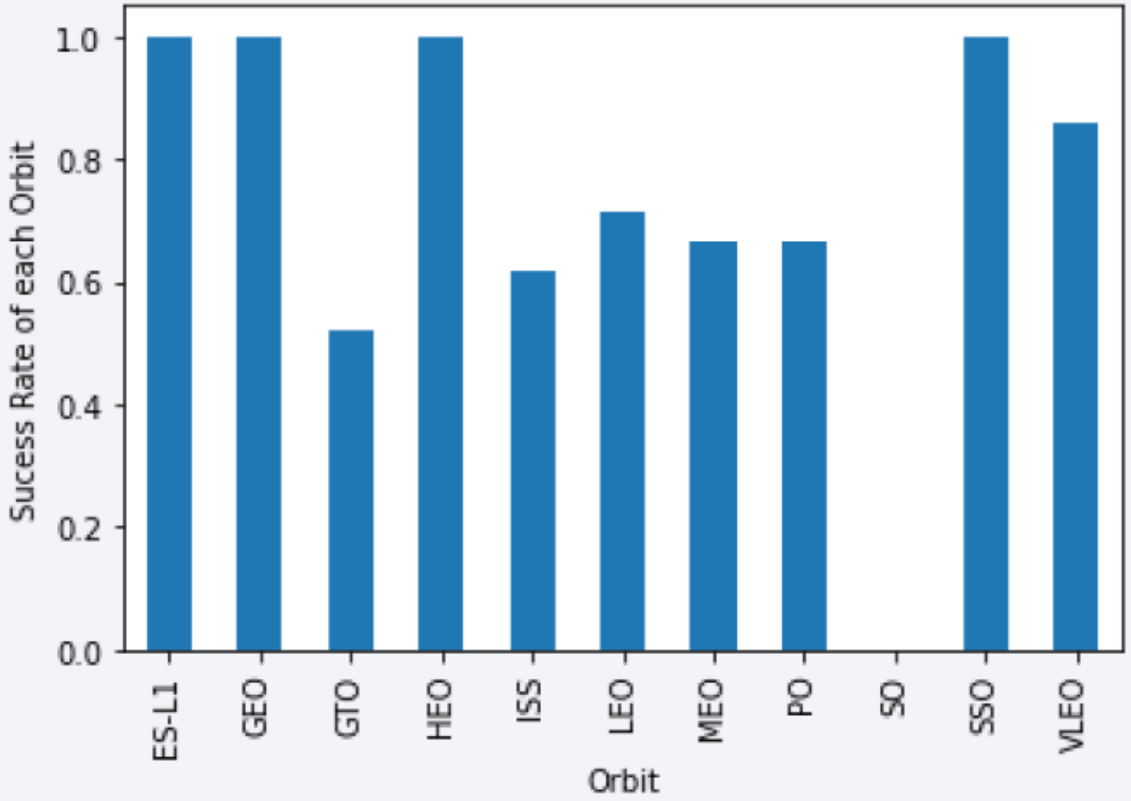
- Show a scatter plot of Payload vs. Launch Site
- Show the screenshot of the scatter plot with explanations

The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



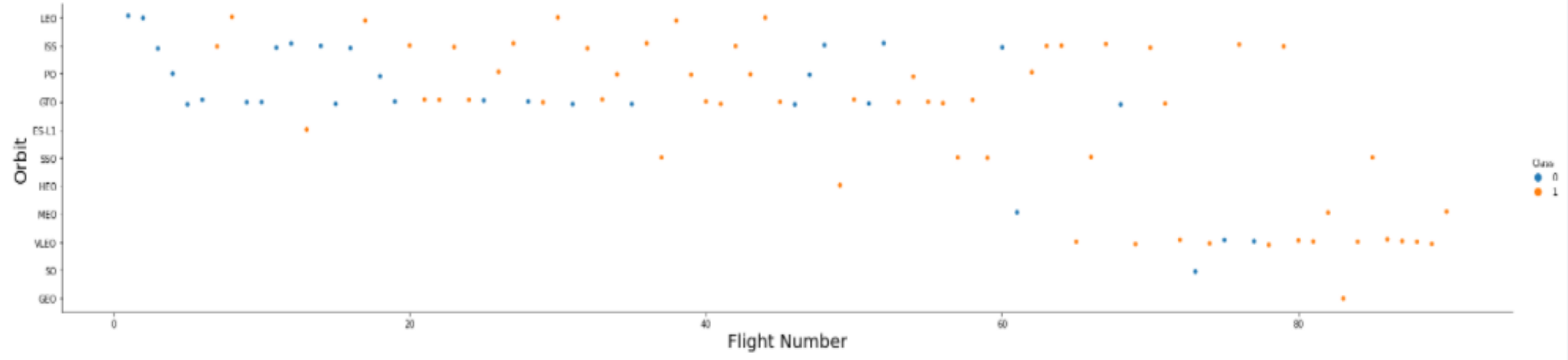
Success Rate vs. Orbit Type

- From the plot, we can see that ES L1, GEO, HEO, SSO, VLEO had the most success rate.
- With this plot, we can see success rate for different orbit types. We note that ES-L1, GEO, HEO, SSO have the best success rate.



Flight Number vs. Orbit Type

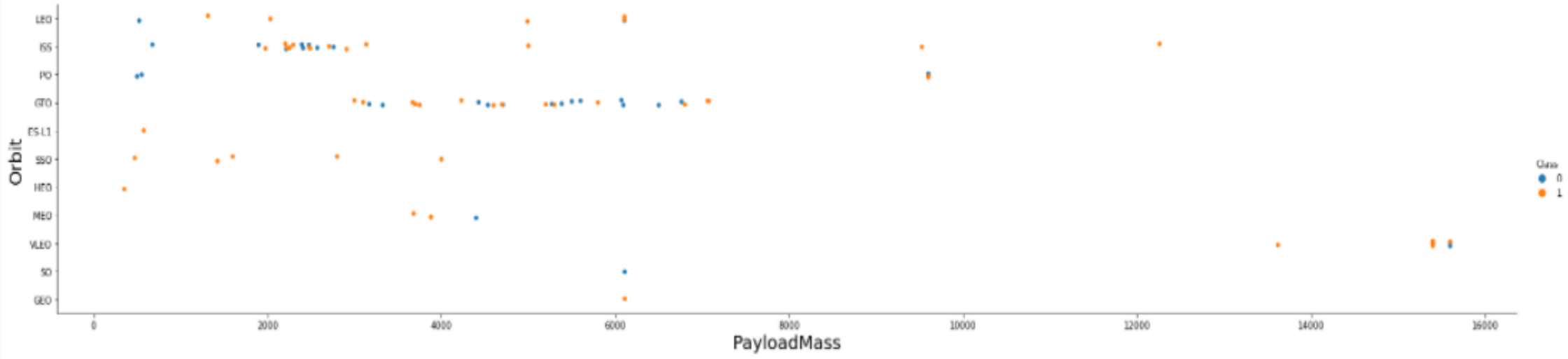
- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.
- We notice that the success rate increases with the number of flights for the LEO orbit. For some orbits like GTO, there is no relation between the success rate and the number of flights. But we can suppose that the high success rate of some orbits like SSO or HEO is due to the knowledge learned during former launches for other orbits.



Payload vs. Orbit Type

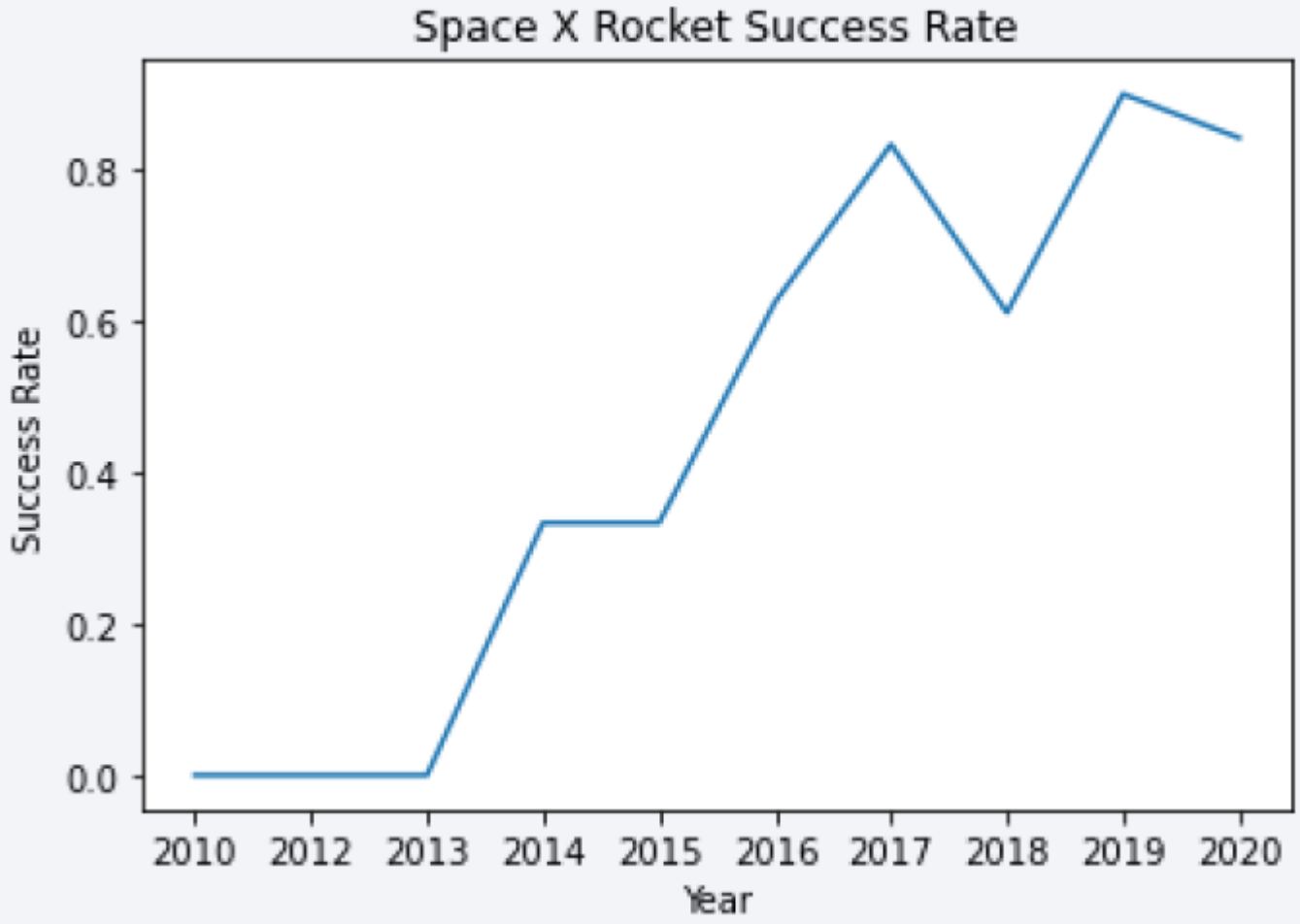
- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.

- The weight of the payloads can have a great influence on the success rate of the launches in certain orbits. For example, heavier payloads improve the success rate for the LEO orbit. Another finding is that decreasing the payload weight for a GTO orbit improves the success of a launch.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

```
In [8]: %sql SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
 * sqlite:///my_data1.db
Done.
```

Out[8]:

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

Launch Site Names Begin with 'CCA'

- We used the query above to display 5 records where launch sites begin with "CCA"

```
%sql SELECT * FROM SPACEXTBL 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_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

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

0]:

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1: This query returns the average of all payload masses where the booster version contains the substring F9 v1.1.

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
```

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

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
  - With this query, we select the oldest successful landing. The WHERE clause filters dataset in order to keep only records where landing was successful. With the MIN function, we select the record with the oldest date.

```
%sql SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing_Outcome" LIKE '%Success%'
```

```
 * 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:
  - This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg. The WHERE and AND clauses filter the dataset.

```
%sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING_OUTCOME" = 'Success (drone ship)' \
AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes:
  - With the first SELECT, we show the subqueries that return results. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission. The WHERE clause followed by LIKE clause filters mission outcome. The COUNT function counts records filtered.

```
%sql SELECT distinct (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS,
\ (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE from SPACEXTBL
```

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

SUCCESS	FAILURE
100	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass:
  - We used a subquery to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass.

```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - This query returns month, booster version, launch site where landing was unsuccessful and landing date took place in2015. Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7, 4) shows year.

```
%sql select substr("Date", 6, 2) as MONTH, substr("Date", 1, 4) AS YEAR, Landing_Outcome,\
Booster_Version, Launch_Site FROM SPACEXTBL WHERE "LANDING_OUTCOME" = 'Failure (drone ship)' AND YEAR = '2015' limit 5
```

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

]:

MONTH	YEAR	Landing_Outcome	Booster_Version	Launch_Site
10	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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
  - This query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNTDESC shows results in decreasing order.

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 "LANDING_OUTCOME", COUNT("LANDING_OUTCOME") as cnt FROM SPACEXTBL \
WHERE "DATE" >= '2010-06-04' and "DATE" <= '2017-03-20' and "LANDING_OUTCOME" like '%Success%' \
GROUP BY "LANDING_OUTCOME" ORDER BY cnt DESC
```

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

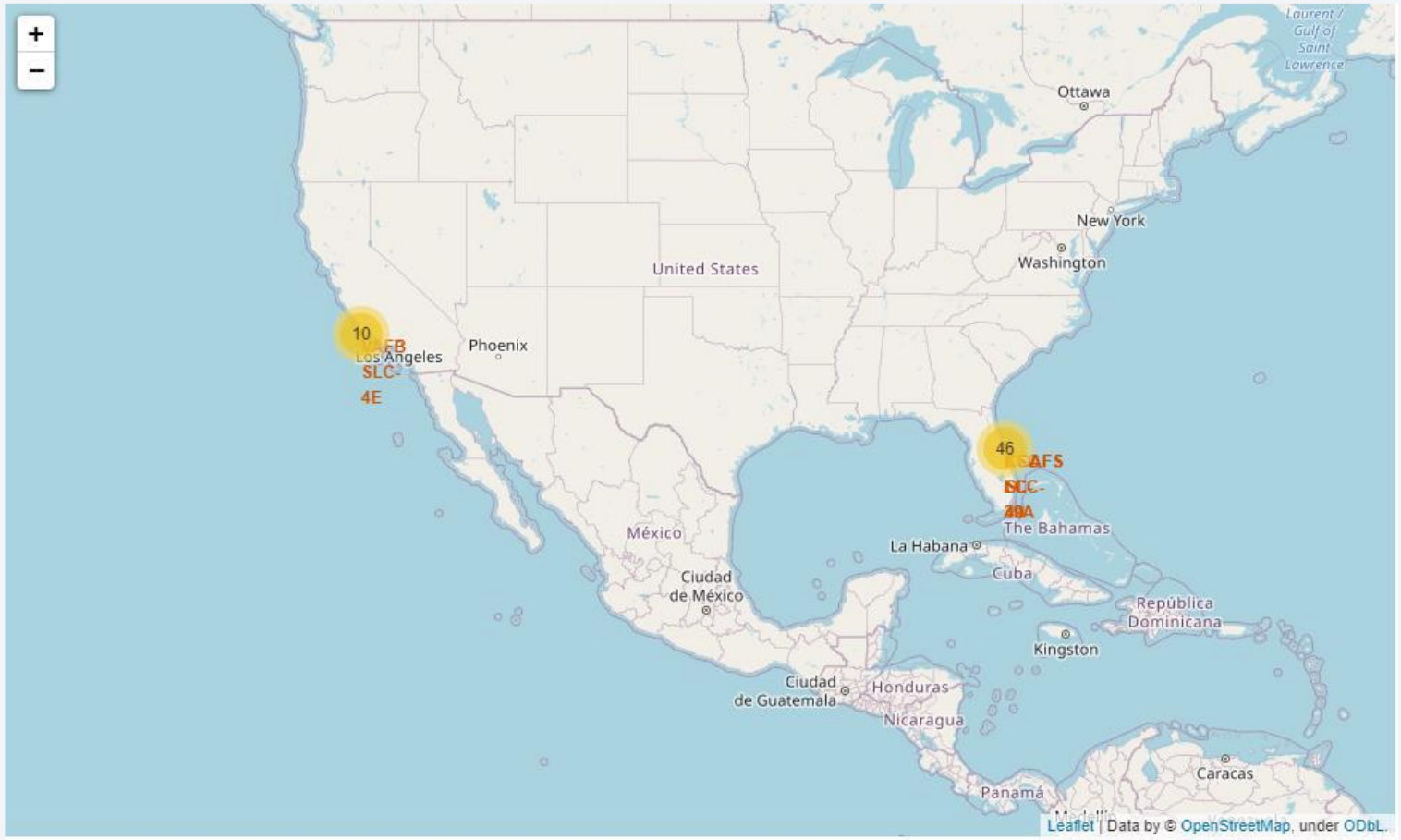
0]:

Landing_Outcome	cnt
Success	20
Success (drone ship)	8
Success (ground pad)	6

Section 3

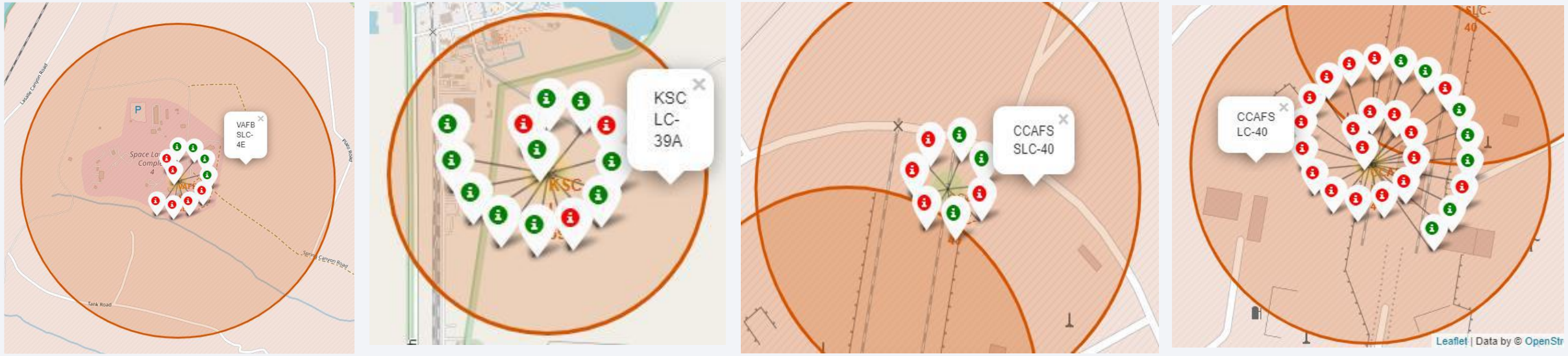
Launch Sites Proximities Analysis

Folium Map - Ground stations



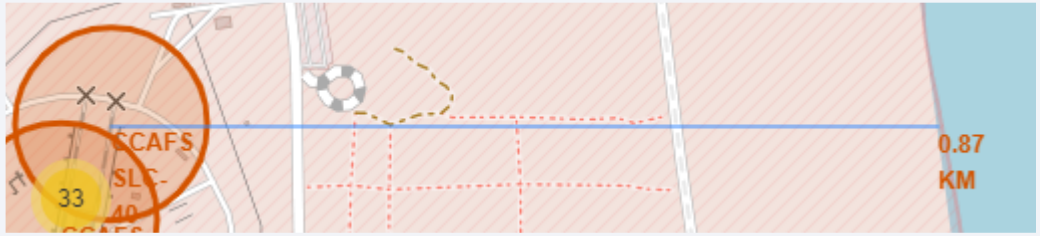
- We see that Space X launch sites are located on the coast of the United States

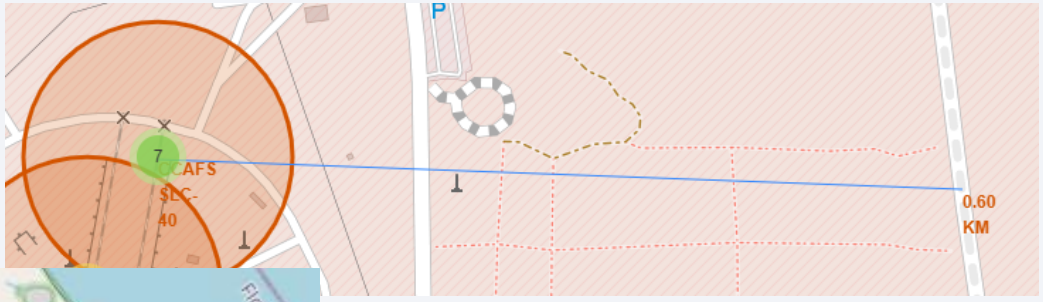
Folium Map - Color Labeled Markers

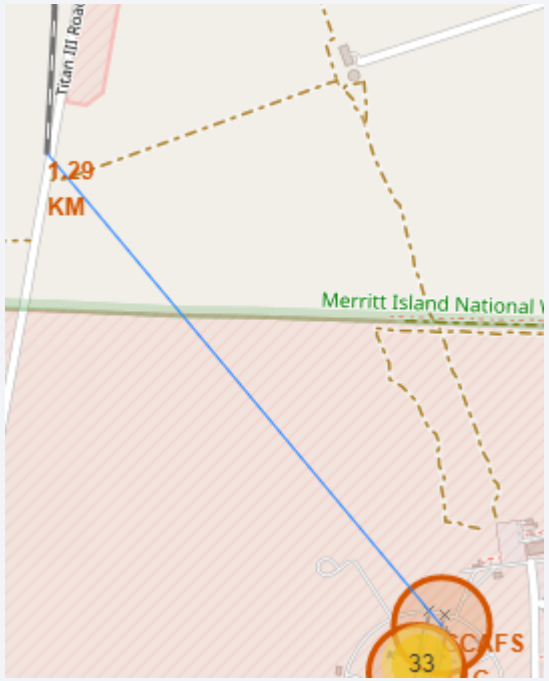


- Green marker represents successful launches. Red marker represents unsuccessful launches. We note that KSC LC 39A has a higher launch success rate.

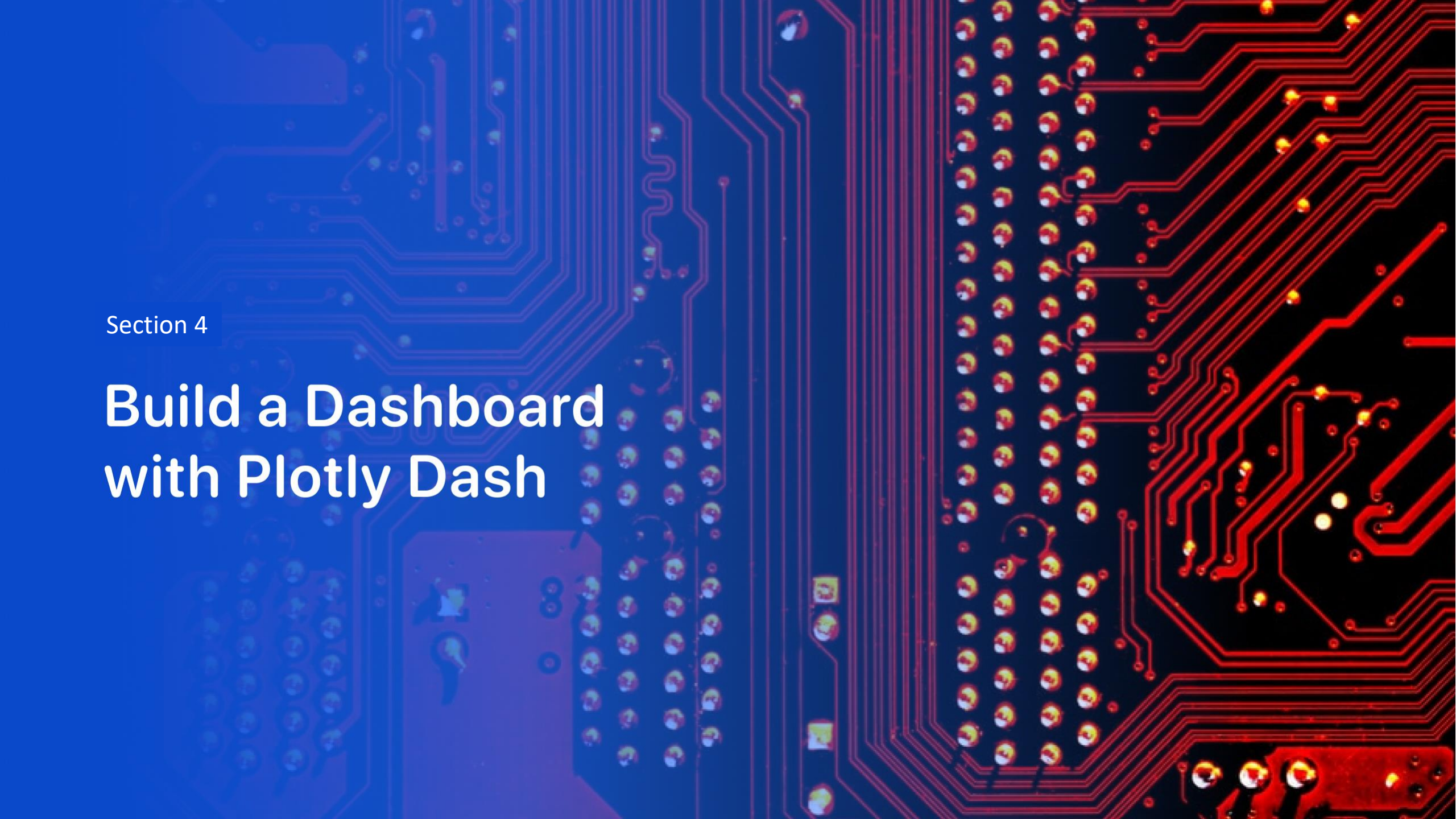
Folium Map - Distances between CCAFS SLC 40 and its proximities



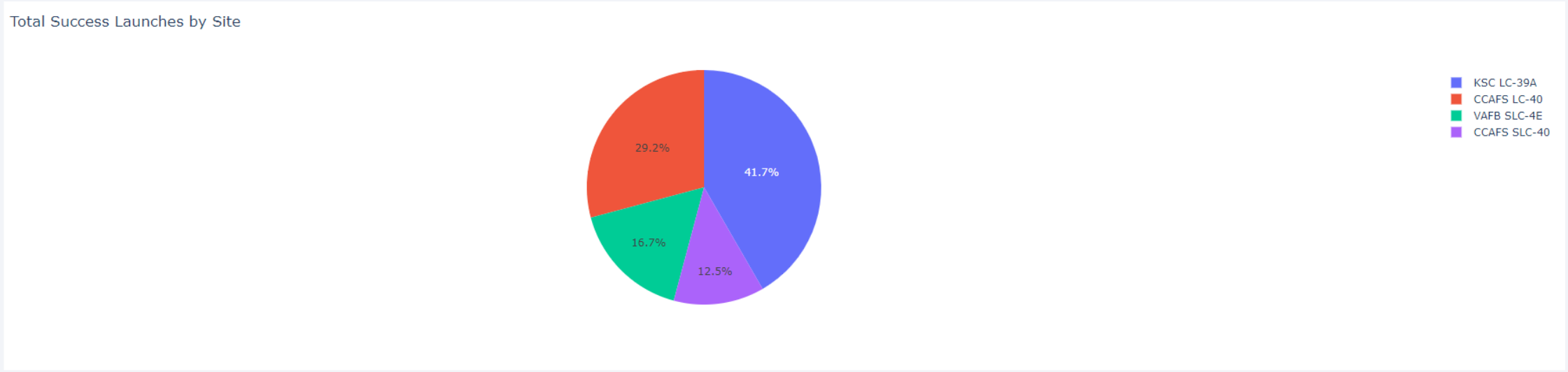




- Is CCAFS SLC-40 in close proximity to railways ? Yes
- Is CCAFS SLC-40in close proximity to highways ? Yes
- Is CCAFS SLC-40in close proximity to coastline ? Yes
- Do CCAFS SLC-40keeps certain distance away from cities ? No

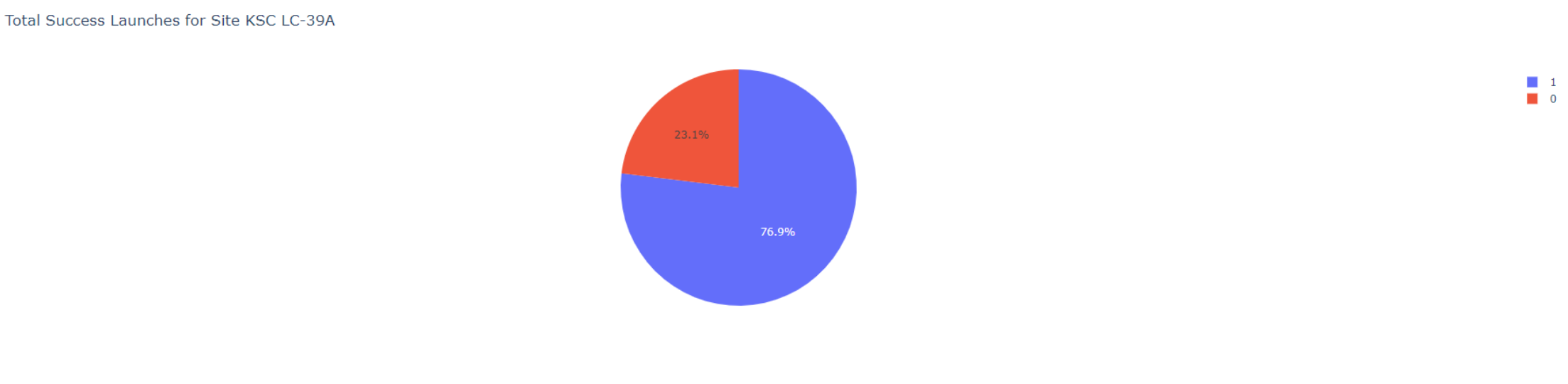

Section 4
Build a Dashboard with Plotly Dash

Dashboard - Total success by Site



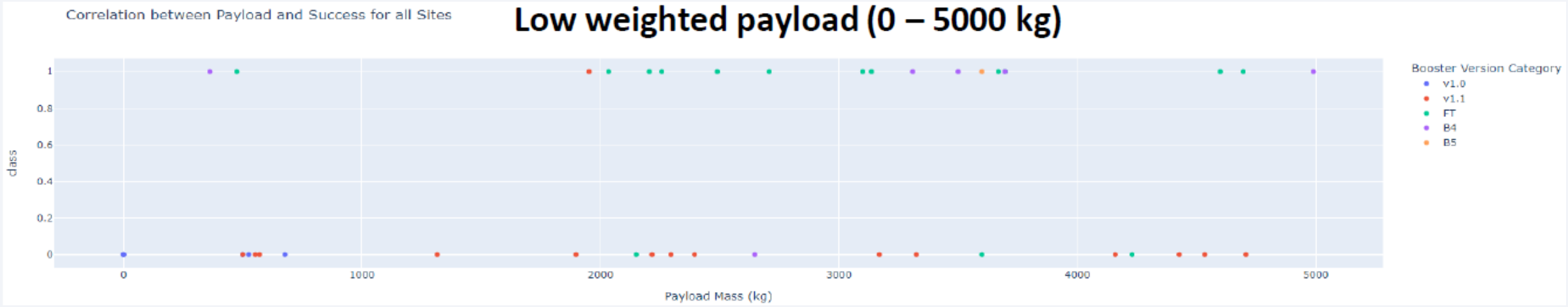
- We see that KSC LC-39A has the best success rate of launches.

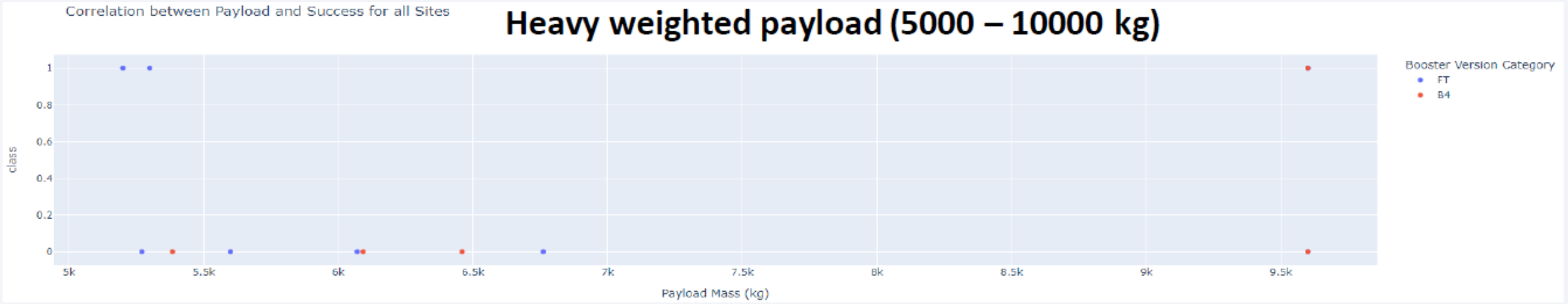
Dashboard - Total success Total success launches for Site KSC LC 39A



- We see that KSC LC-39A has achieved a 76.9%success rate while getting a 23.1% failure rate.

Dashboard - Payload mass vs Outcome for all sites with different payload mass selected





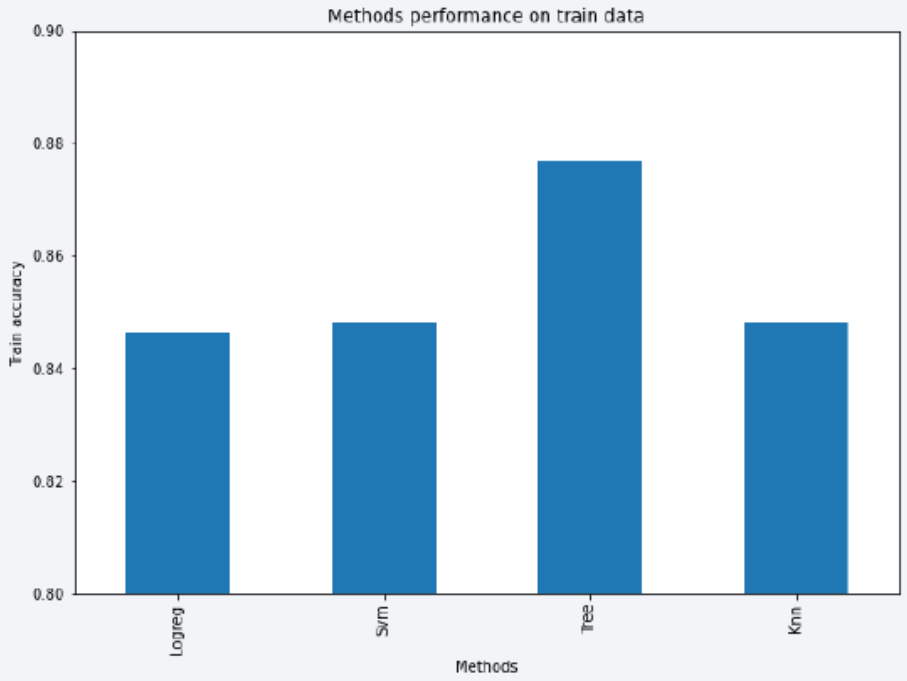
- Low weighted payloads have a better success rate than the heavy weighted payloads.

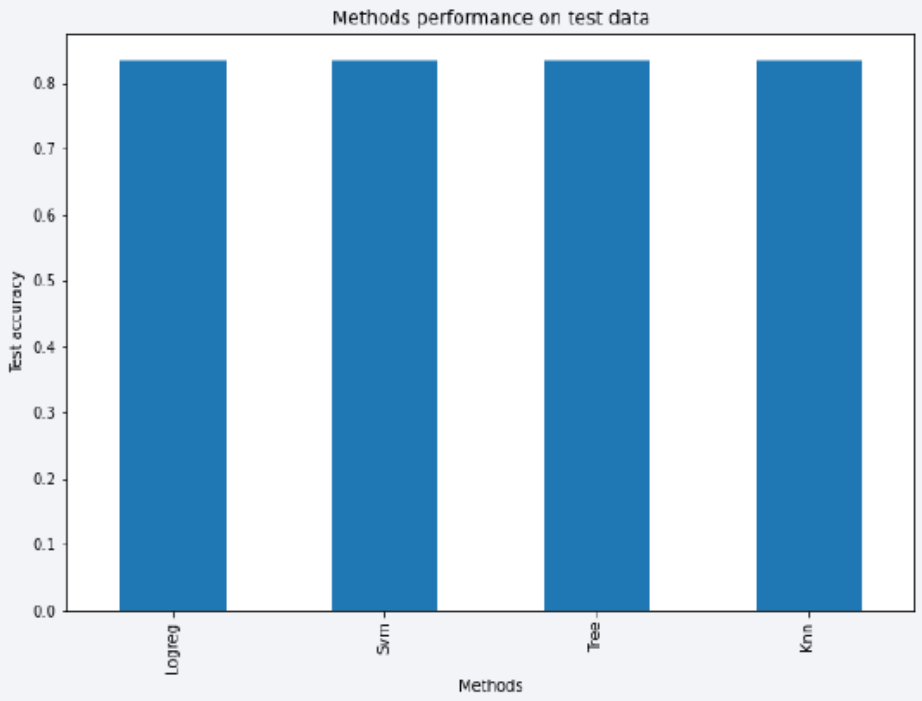
Section 5

Predictive Analysis (Classification)

Classification Accuracy

	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333



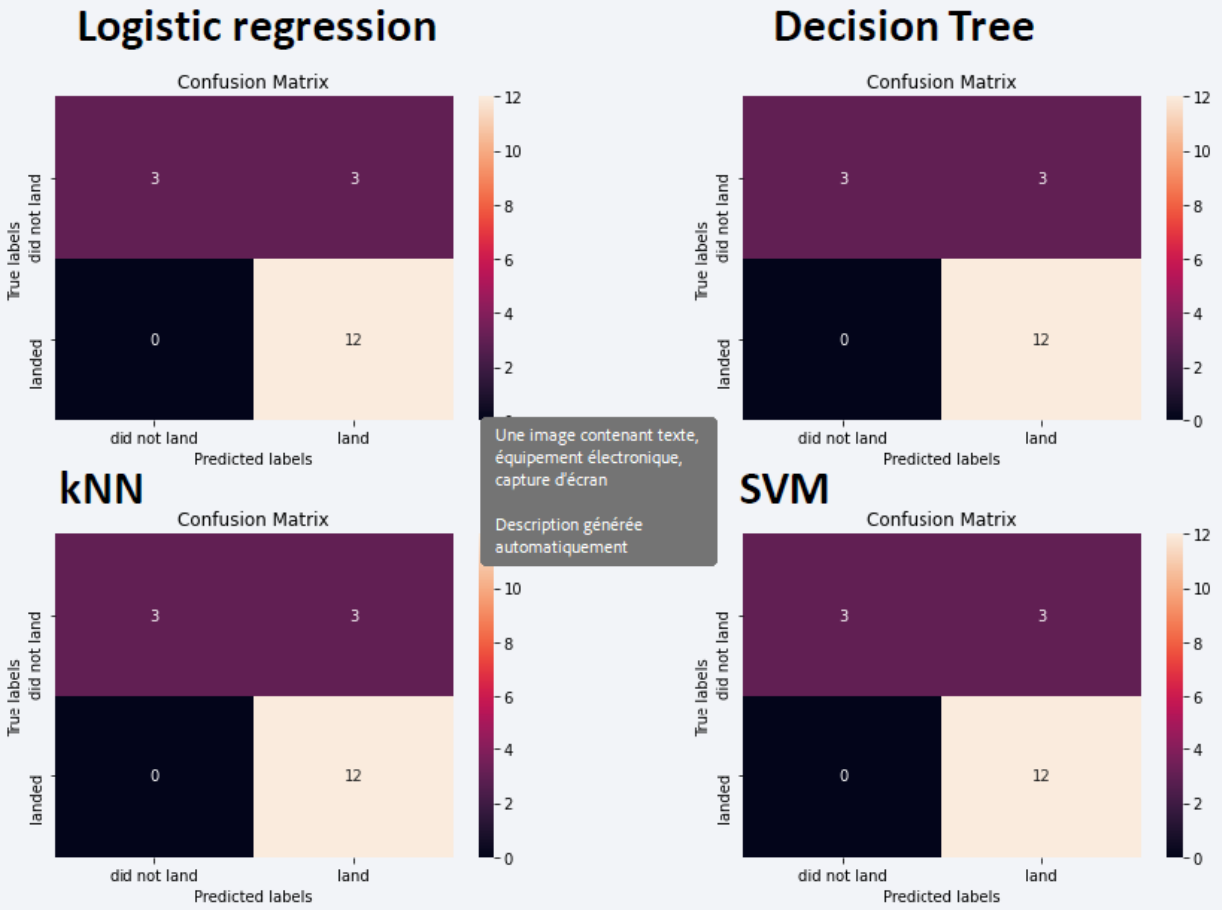


- For accuracy test, all methods performed similar. We could get more test data to decide between them. But if we really need to choose one right now, we would take the decision tree.

Decision tree best parameters

```
tuned hyperparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf':
4, 'min_samples_split': 2, 'splitter': 'random'}
```

Confusion Matrix



As the test accuracy are all equal, the confusion matrices are also identical. The main problem of these models are false positives.

		Actual values 1	Actual values 0
Predicted values	1	TP	FP
	0	FN	TN

Conclusions

- The success of a mission can be explained by several factors such as the launch site, the orbit and especially the number of previous launches. Indeed, we can assume that there has been a gain in knowledge between launches that allowed to go from a launch failure to a success.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1.
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission. Some orbits require a light or heavy payload mass. But generally low weighted payloads perform better than the heavy weighted payloads.
- With the current data, we cannot explain why some launch sites are better than others (KSC LC-39A is the best launch site). To get an answer to this problem, we could obtain atmospheric or other relevant data.
- For this dataset, we choose the Decision Tree Algorithm as the best model even if the test accuracy between all the models used is identical. We choose Decision Tree Algorithm because it has a better train accuracy.

Appendix

- jupyter-labs-spacex-data-collection-api.ipynb
- jupyter-labs-webscraping.ipynb
- jupyter-labs-jupyter-spacex-Data wrangling.ipynb
- jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb
- lab_jupyter_launch_site_location.jupyterlite.ipynb
- SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb
- jupyter-labs-eda-sql-coursera_sqllite.ipynb
- spacex_dash_app.py
- spacex_dash_app.jpg

Thank you!