IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Dao Hung>
<2022 March 06>

Outline

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

Executive Summary

- Summary of methodologies
  - ✓ Data collection
  - ✓ Data wrangling
  - ✓ EDA with data visualization
  - ✓ EDA with SQL
  - ✓ Building an interactive map with Folium
  - ✓ Building a Dashboard with Plotly Dash
  - ✓ Predictive analysis (Classification)
- Summary of all results
  - ✓ Exploratory data analysis results
  - ✓ Interactive analytics demo in screenshots
  - ✓ Predictive analysis results

Introduction

- Project background and context
  - ✓ 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. In this lab, you will collect and make sure the data is in the correct format from an API. The following is an example of a successful and launch.
- Common Problems that needed solving
  - ✓ What influences if the rocket will land successfully?
  - ✓ The factors that will impact in determining the success rate of a successful landing.
  - ✓ What conditions does SpaceX have to achieve to ensure the best rocket success landing rate.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - (Web Scrapping) from WikipediaDescribe how data was collected
- Perform data wrangling
  - Hot Encoding data fields for Machine Learning and dropping irrelevant columns
- 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 following datasets was collected by

- SpaceX launch data that is gathered from the SpaceX REST API.
- This API will provice data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- This data will be used to predict whether SpaceX will attempt to land a rocket or not.
- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/.
- Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using BeautifulSoup.

Data Collection – SpaceX API

- A request object was created using the space x api's end point The response object was converted to a dataframe using pandas.json_normalize(response.json())

- Github URL

1. Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)
```

2. Converting to a .json file

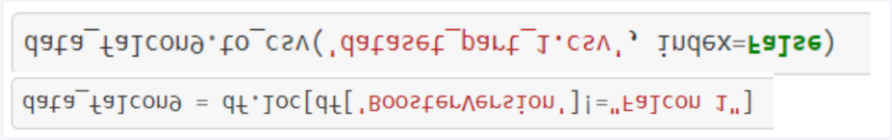
```
response.status_code

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

3. Clean data

4. Dictionary/ datafram

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Filter data and export



Data Collection - Scraping

- Some of the essential data was collected from Wikipedia using web scrapping with the help of beautiful soup framework.

1. Creating BeautifulSoup Object

```
soup = BeautifulSoup(page.text, 'html.parser')
```

2. Finding tables

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

- Github URL

3. Column names / dictionary

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

4. Appending data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumer
   # get table row
    for rows in table.find_all("
        #check to see if first t
```

5. Converting to datafram and export

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

Data Wrangling

- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean.
  - One hot encoding was done for some categorical variables in order to feed them to the ML algorithm after performing feature scaling.
  - The class variable had the values {'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'} These values were converted to 0 and {'True ASDS', 'True Ocean', 'True RTLS'} to 1 representing successful landing of stage 1. Additionally, success rate was found out to be: 66%
  - Github URL:

EDA with Data Visualization

- Scatter Graphs being drawn:
  - Flight Number VS. Payload Mass
  - Flight Number VS. Launch Site
  - Payload VS. Launch Site
  - Orbit VS. Flight Number
  - Payload VS. Orbit Type
  - Orbit VS. Payload Mass
- Bar Graph being drawn:
  - Mean vs. Orbit
- Line Graph being drawn:
  - Success Rate vs. Year

Github URL:

EDA with SQL

- **Performed SQL queries to gather information about the dataset:**

• Displaying the names of the unique launch sites in the space mission
• Displaying 5 records where launch sites begin with the string 'KSC’
• Displaying the total payload mass carried by boosters launched by NASA (CRS)
• Displaying average payload mass carried by booster version F9 v1.1
• Listing the date where the successful landing outcome in drone ship was achieved.
• Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000
but less than 6000
• Listing the total number of successful and failure mission outcomes
• Listing the names of the booster_versions which have carried the maximum payload mass.
• Listing the records which will display the month names, successful landing_outcomes in ground pad ,booster
versions, launch_site for the months in year 2017
• Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending

order.

- Github URL:

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, i.e., the initial position of rocket trajectories. 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.
  - **We assigned the dataframe launch_outcomes(failures, successes) to *classes 0 and 1*** with Green and Red markers on the map in a MarkerCluster()
  - **Using Haversine's formula we calculated the distance** from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. **Lines** are drawn on the map to measure distance to landmarks

Github URL:

Predictive Analysis (Classification)

- 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. In this lab, you will create a machine learning pipeline to predict if the first stage will land given the data from the preceding labs.

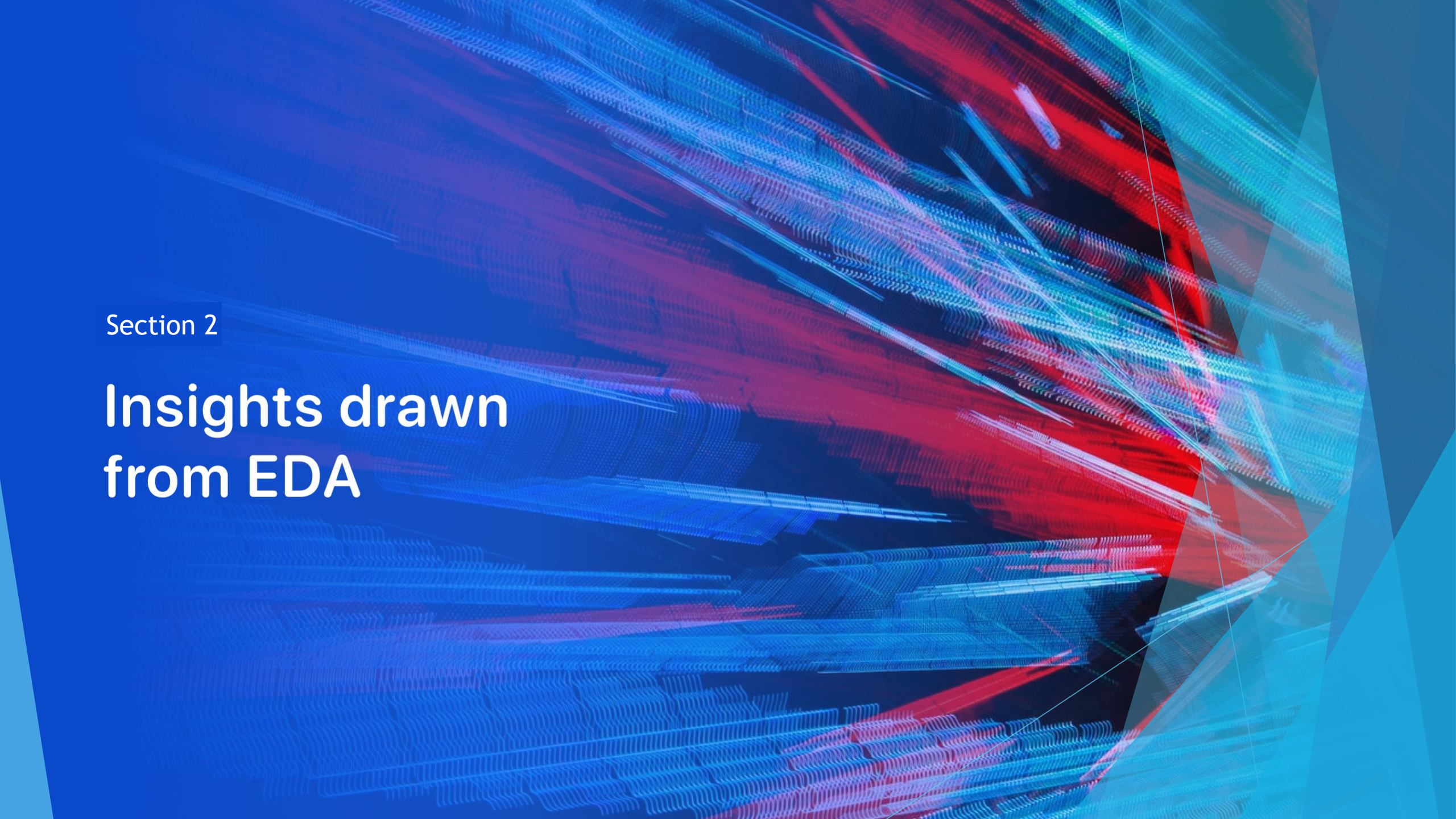
Github URL:

Results

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

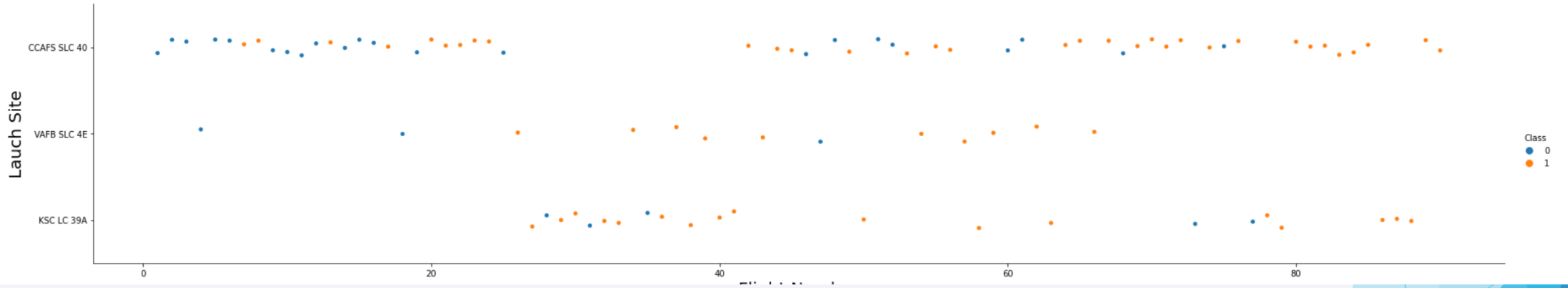
```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test, Y_test))
```

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.8888888888888888
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

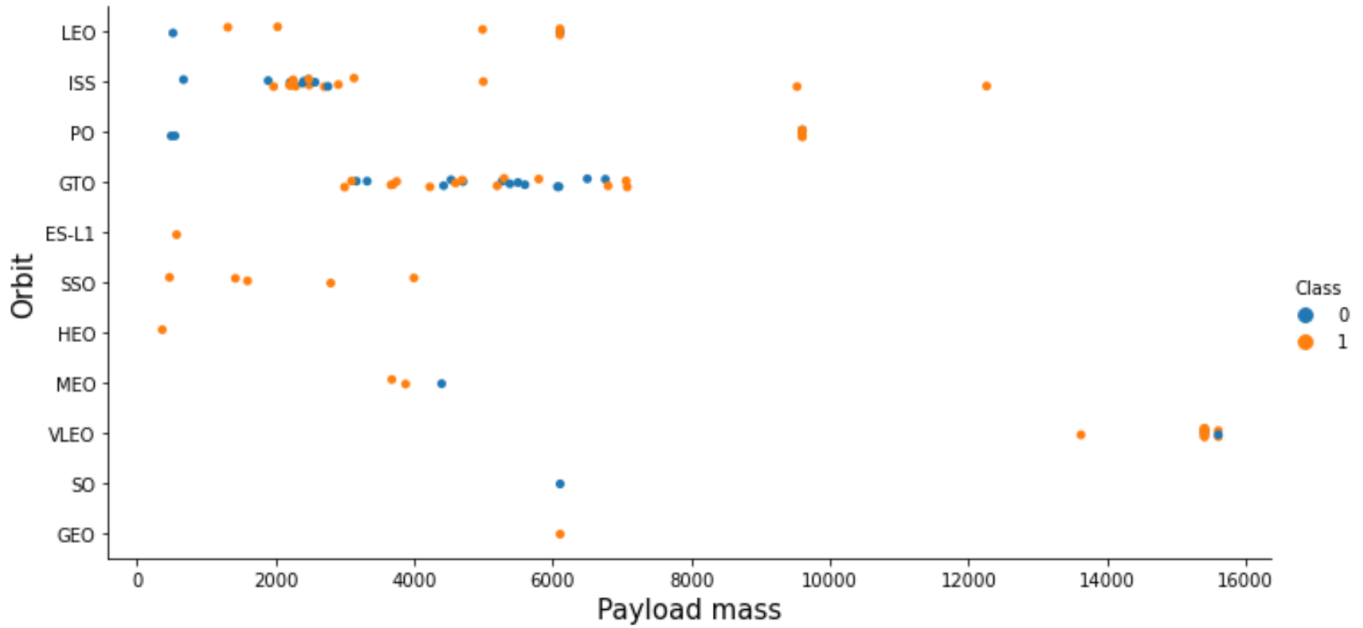
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Lauch Site",fontsize=20)
plt.show()
```



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

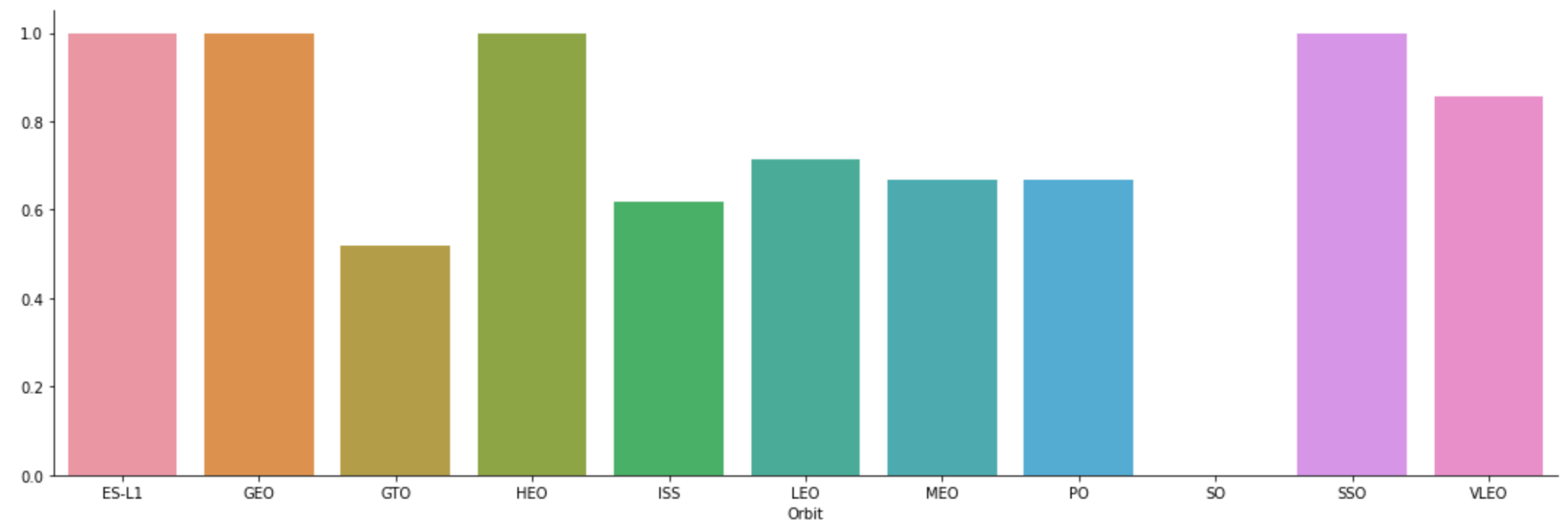
Payload vs. Launch Site

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(x="PayloadMass",y="Orbit",hue='Class' ,data=df,aspect=2)
plt.xlabel("Payload mass", fontsize=15)
plt.ylabel("Orbit", fontsize=15)
plt.show()
```



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

Success Rate vs. Orbit Type

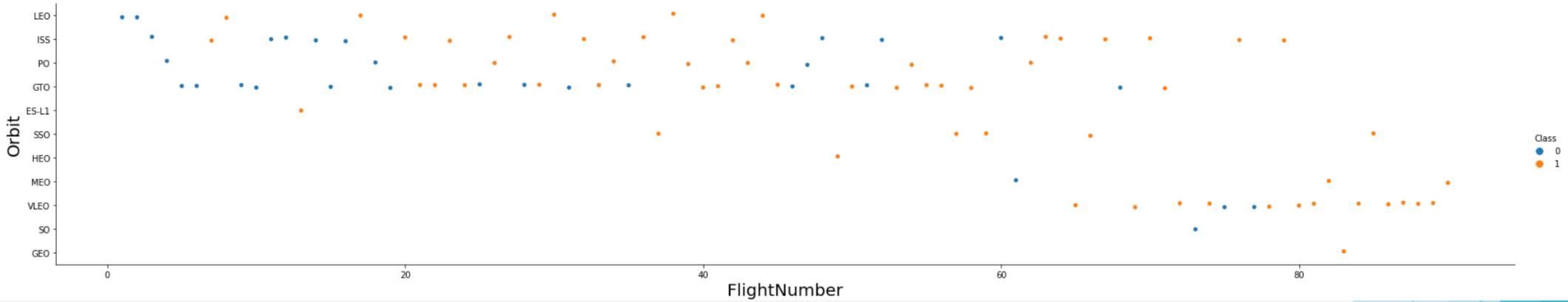


Analyze the ploted bar chart try to find which orbits have high sucess rate.

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

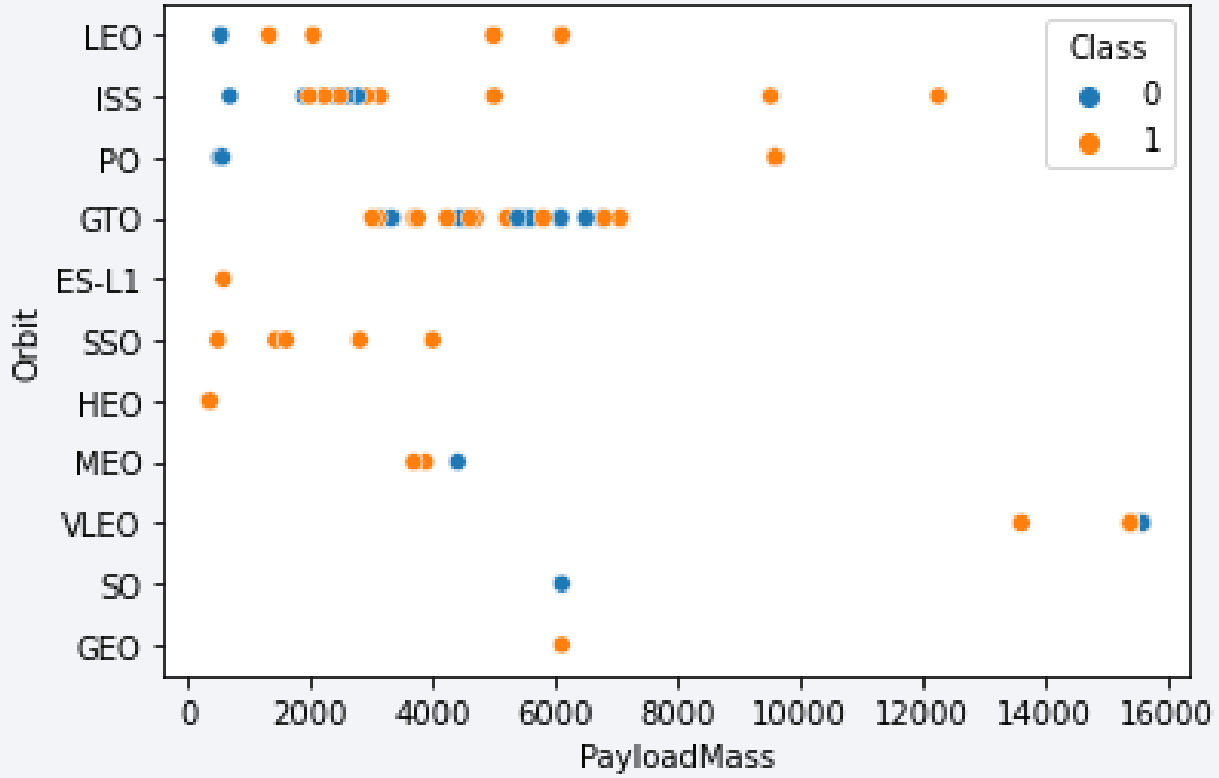
Flight Number vs. Orbit Type

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



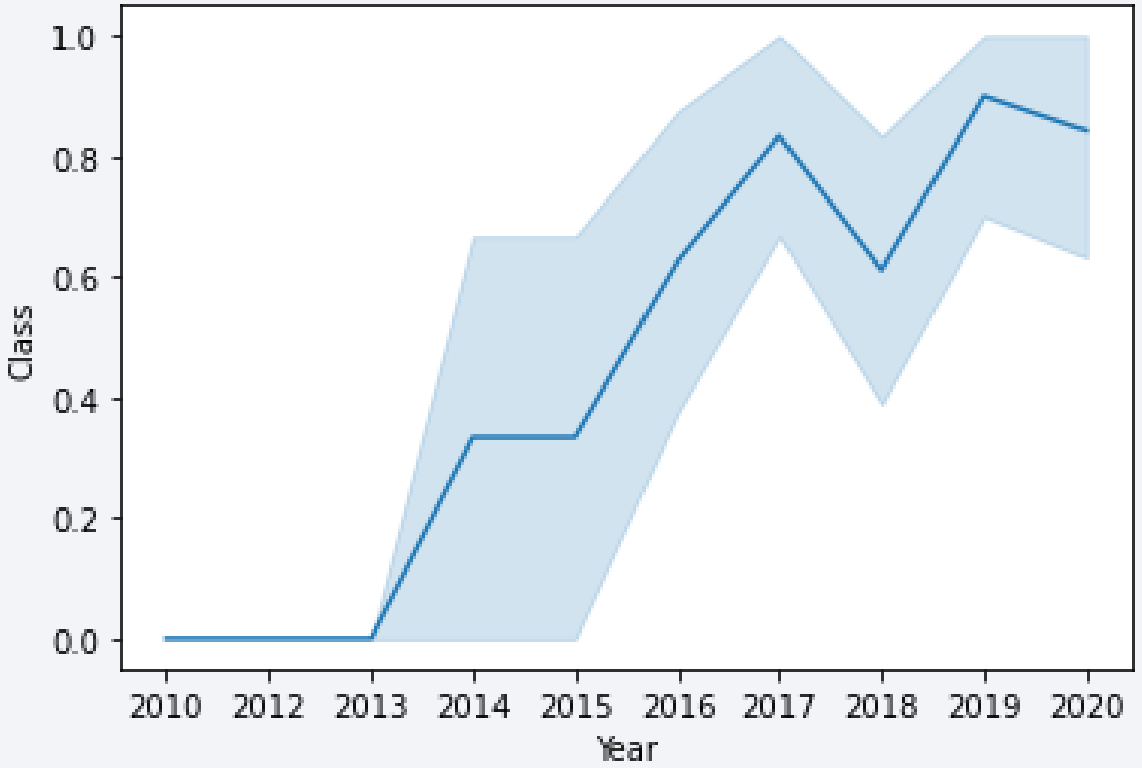
- The LEO orbit the Success appears related to the number of flights
- On the other hand, there seems to be no relationship between flight number when in GTO orbit

Payload vs. Orbit Type



Heavy payloads have a negative influence on GTO orbits and positive on ISS and LEO orbits.

Launch Success Yearly Trend



The success rate since 2013 kept increasing till 2020

1. All Launch Site Names

Using the word ***DISTINCT*** in the query means that it will only show Unique values in the ***Launch_Site*** column from ***tblSpaceX***

```
%sql select Unique(LAUNCH_SITE) from SPACEXTBL;
```

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

2. Launch Site Names Begin with 'CCA'

Using the word ***TOP 5*** in the query means that it will only show 5 records from ***tblSpaceX*** and ***LIKE*** keyword has a wild card with the words ***'KSC%'*** the percentage in the end suggests that the Launch_Site name must start with KSC.

Task 2

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

```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

3. Total Payload Mass

Using the function ***sum*** summates the total in the column ***PAYLOAD_MASS_KG_***

Task 3

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

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

payloadmass
619967

4. Average Payload Mass by F9 v1.1

Using the function ***AVG*** works out the average in the column ***PAYLOAD_MASS_KG***

Task 4

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

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

payloadmass
6138

5. First Successful Ground Landing Date

Using the function ***MIN*** works out the minimum date in the column ***Date***
The ***WHERE*** clause filters the dataset to only perform calculations on ***Landing_Outcome Success (drone ship)***

Task 5

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

Hint:Use min function

```
%sql select min(DATE) from SPACEXTBL;
```

1
2010-06-04

6. Successful Drone Ship Landing with Payload between 4000 and 6000

Selecting only ***Booster_Version***
The ***WHERE*** clause filters the dataset to ***Landing_Outcome = Success (drone ship)***

Task 6

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 BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN
```

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

7. Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

missionoutcomes
1
99
1

8. Boosters Carried Maximum Payload

Using the word ***DISTINCT*** in the query means that it will only show Unique values in the ***Booster_Version*** column from ***tblSpaceX***

Task 8

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

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) fr
```

boosterversion
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

9. 2015 Launch Records

a much more complex query as I had my ***Date*** fields in SQL Server stored as ***NVARCHAR*** the ***MONTH*** function returns name
month. The function ***CONVERT*** converts ***NVARCHAR*** to ***Date***.

Task 9

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 MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where EXTRACT(YEAR FROM DATE)='2015
```

1	mission_outcome	booster_version	launch_site
1	Success	F9 v1.1 B1012	CCAFS LC-40
2	Success	F9 v1.1 B1013	CCAFS LC-40
3	Success	F9 v1.1 B1014	CCAFS LC-40
4	Success	F9 v1.1 B1015	CCAFS LC-40
4	Success	F9 v1.1 B1016	CCAFS LC-40
6	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

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

unction ***COUNT*** counts records in column
WHERE filters data

Task 10

Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

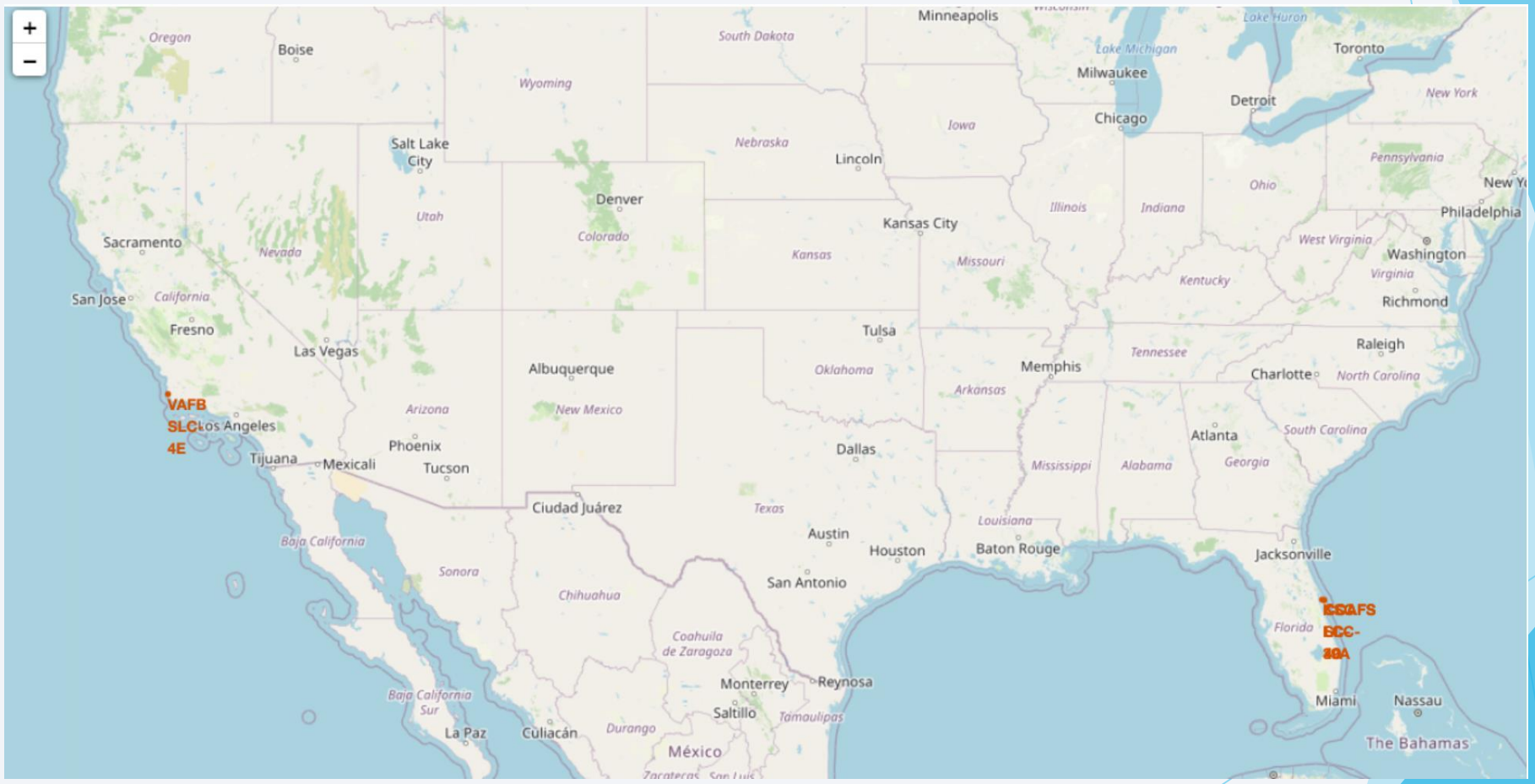
```
%sql SELECT LANDING__OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

landing__outcome
No attempt
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)
Success (drone ship)
Success (drone ship)
Success (drone ship)
Failure (drone ship)
Failure (drone ship)

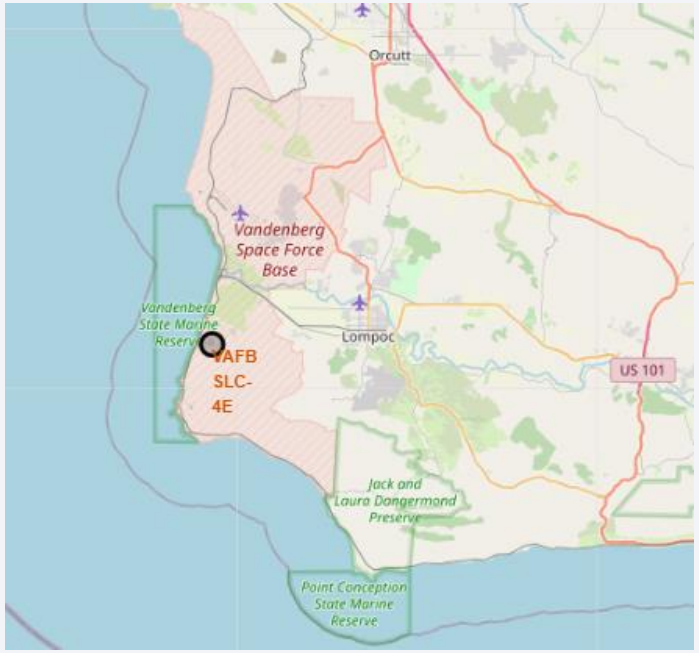
Section 3

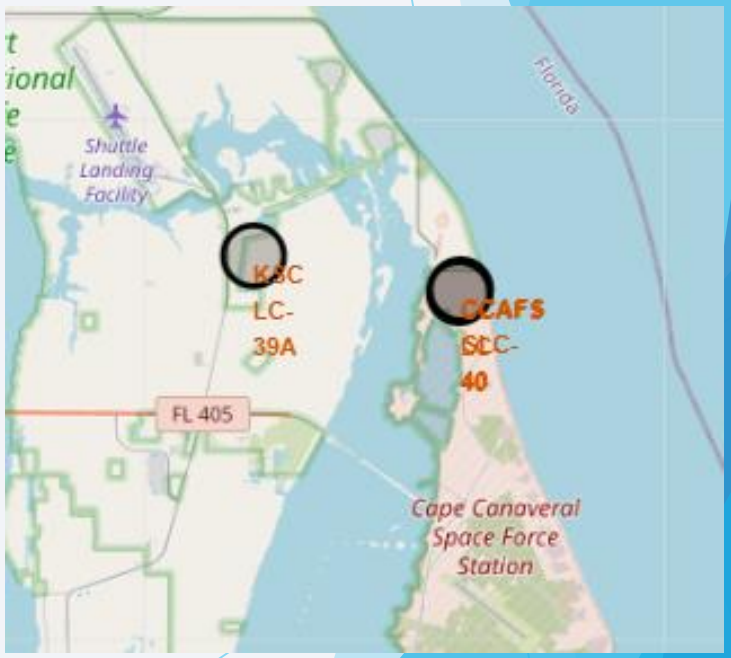
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

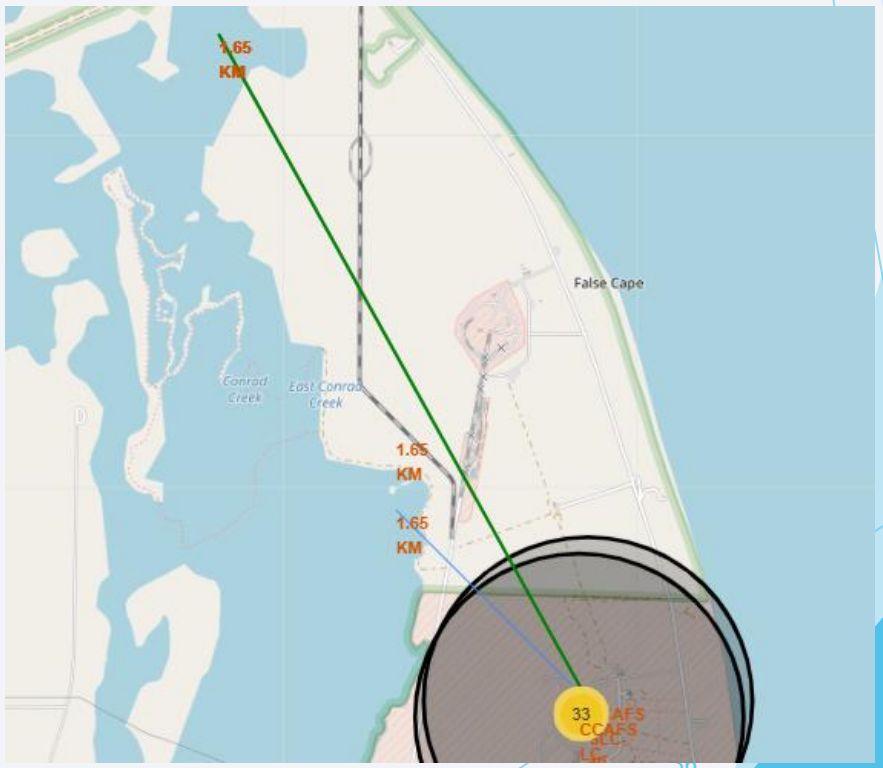


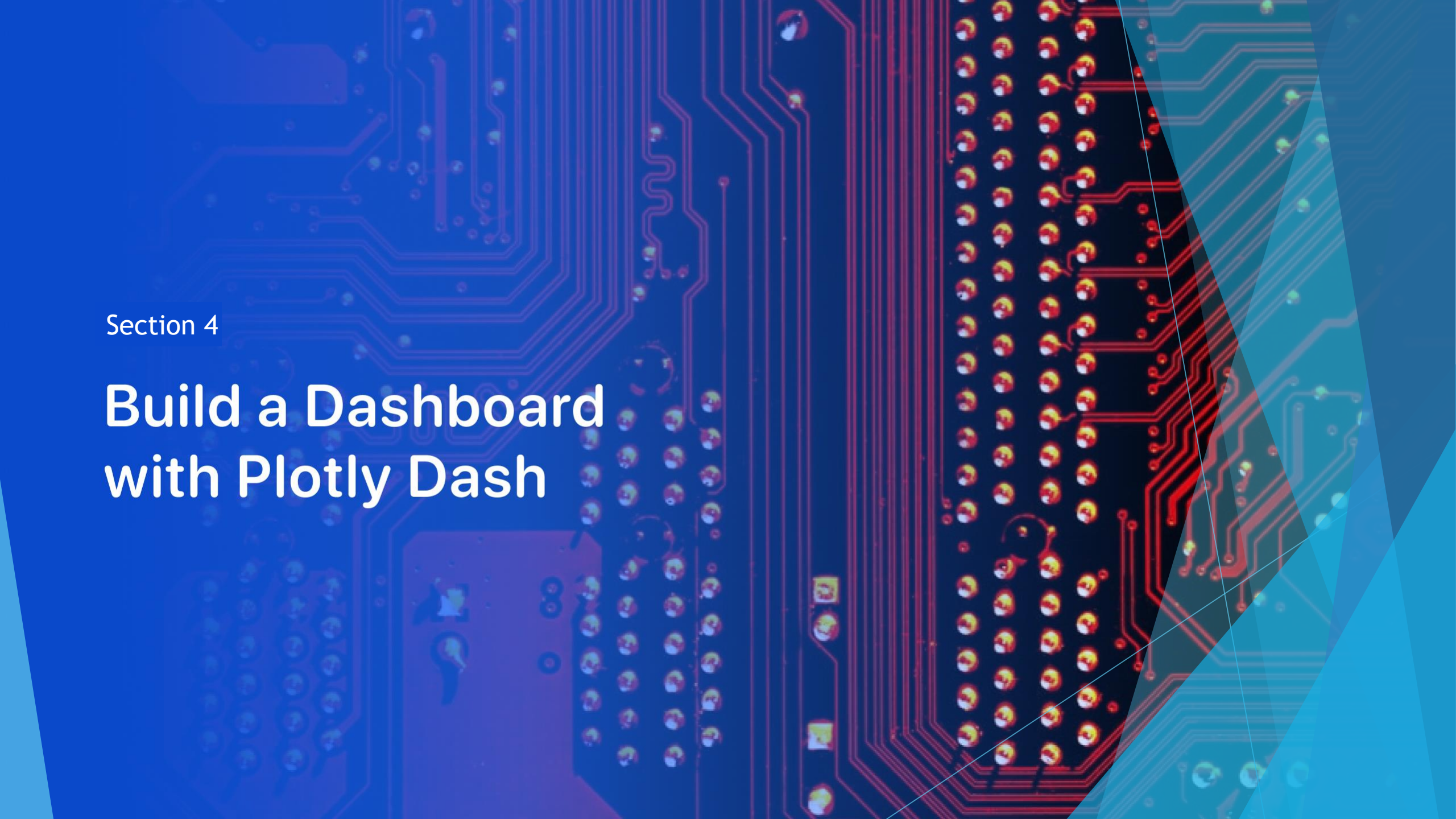
<Folium Map Screenshot 2>



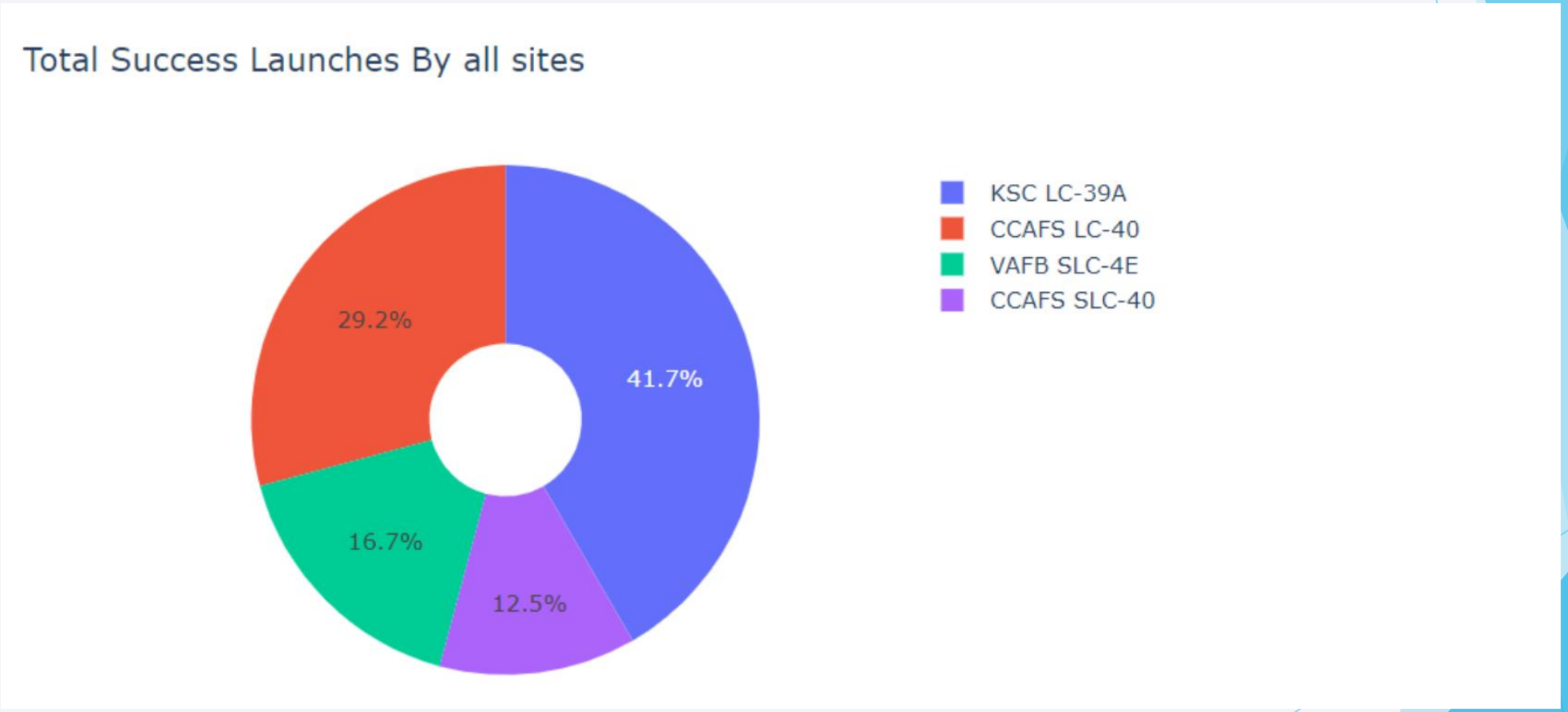


<Folium Map Screenshot 3>

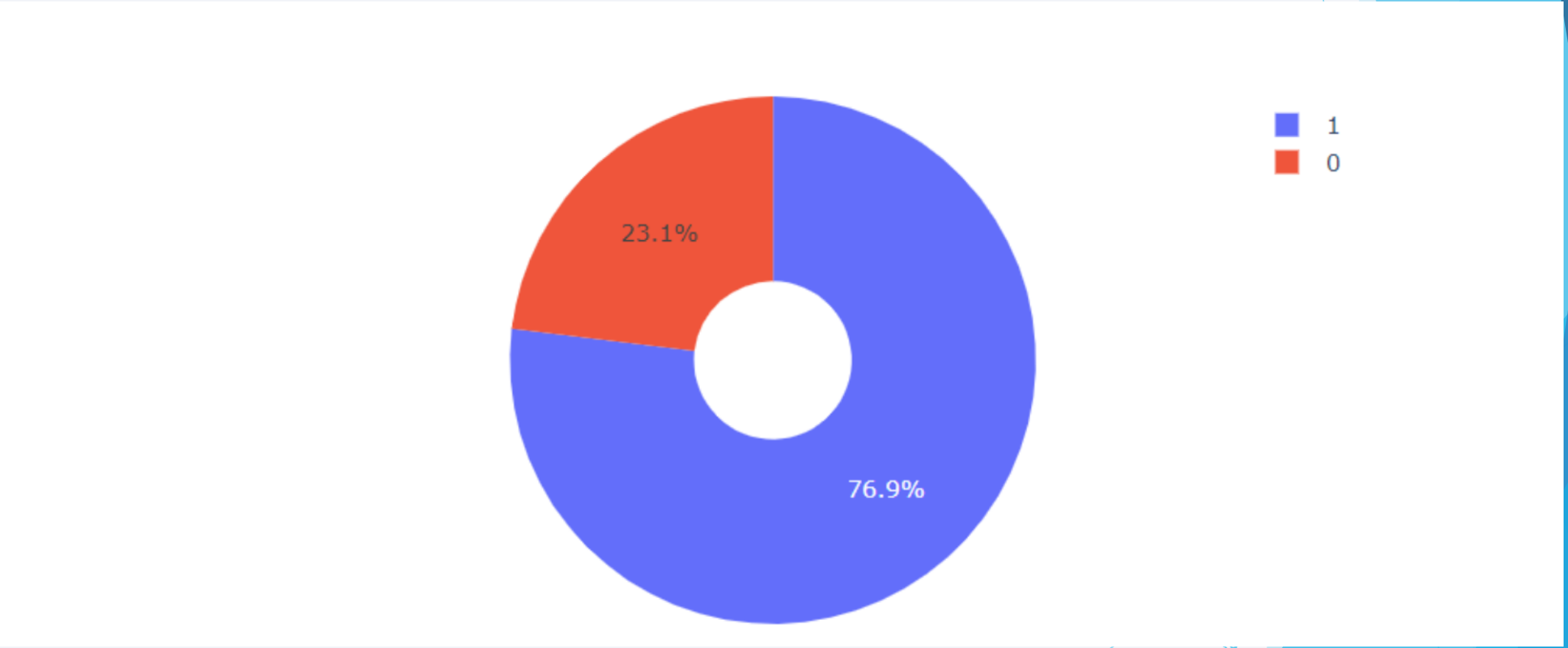



Section 4
Build a Dashboard with Plotly Dash

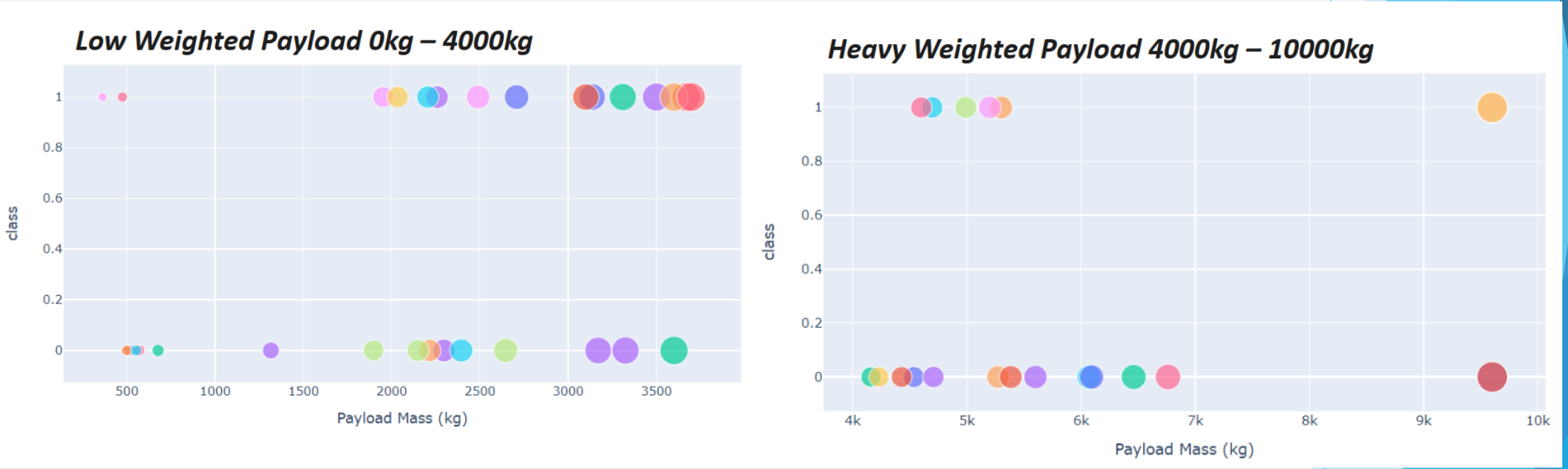
<Dashboard Screenshot 1>



<Dashboard Screenshot 2>



<Dashboard Screenshot 3>



Section 5

Predictive Analysis (Classification)

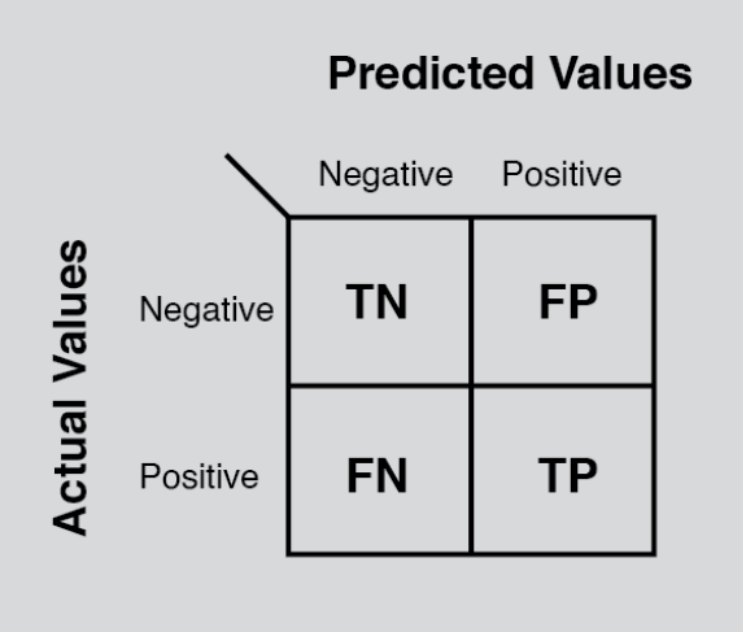
Classification Accuracy

```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test, Y_test))
```

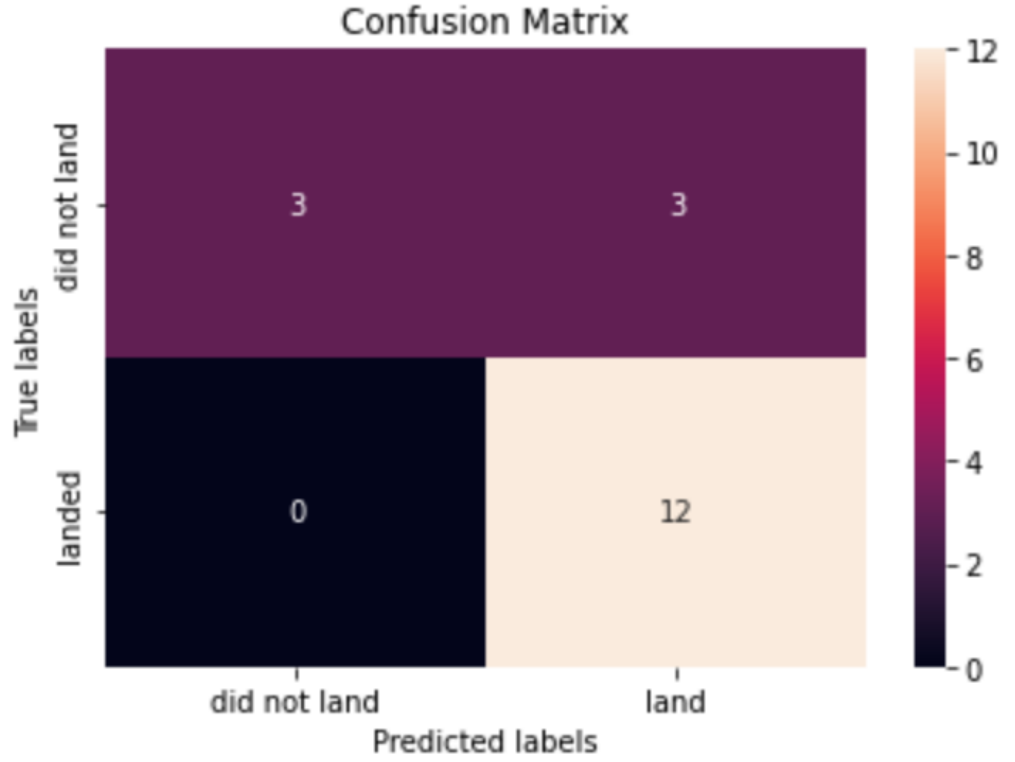
```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.8888888888888888
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```

Confusion Matrix

Examining the confusion matrix from knn model, the classes are distinguished pretty clear.




```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- The Tree Classifier Algorithm is the best for Machine Learning for this dataset
- Low weighted payloads perform better than the heavier payloads
- SpaceX is leading the space race, and there are some major improvements in the rate of success of landing of falcon 9 stage one.
- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Thank you!