IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Brian E. Sosa
05/06/2023

# Outline

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

# Executive Summary

- In this analysis we will collect the data from two different sources: from a SpaceX Rest API and by web scraping a SpaceX Wikipedia entry. Data wrangling, formatting and cleaning is then performed on the collected data. We conduct several SQL queries on the data, and an exploratory data analysis with visualizations is performed. Finally, several machine learning models are trained using the collected data to predict successful Falcon 9 landing outcomes.
- Several results are obtained from the analysis of the data. Visualizations showing descriptive features of the data are presented, such as tables, interactive maps and a dashboard. With regard to the aim of the analysis, we obtained several statistical models to predict landing outcomes of the rocket launches. The accuracy of these models is evaluated in this analysis.

# Introduction

- 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 analysis we will predict if the Falcon 9 first stage will land successfully.
- The aim of this analysis raises several concerns, such as: where do the most successful launches take off?, what features do the successful missions share?, what is the most accurate statistical model to predict Falcon 9 first stage landings?

Section 1

# Methodology

# Methodology

## Executive Summary

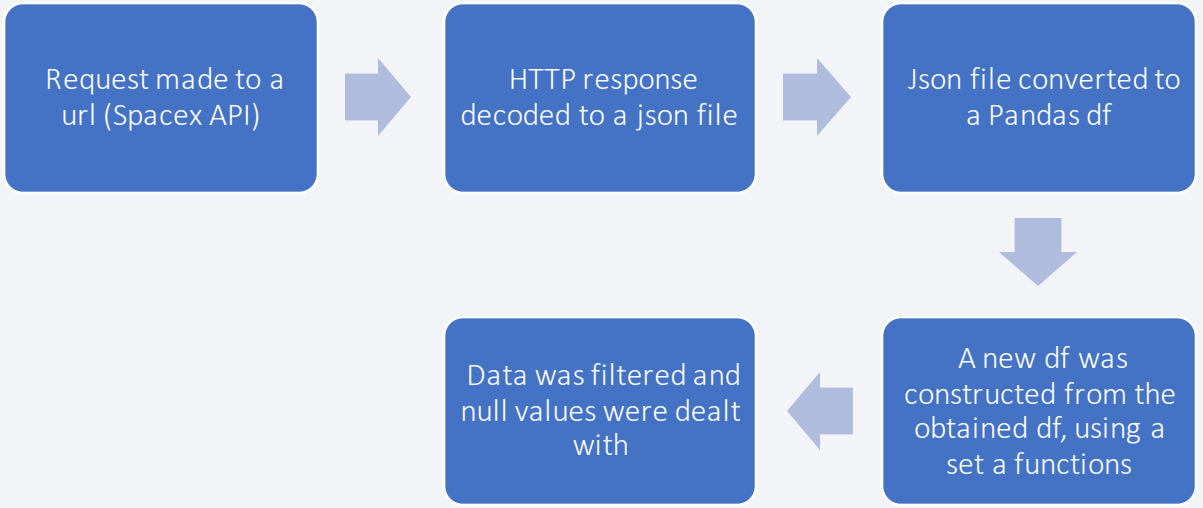
- Data collection methodology:
  - Request made to a SpaceX Rest API
  - Web Scraping from a Spacex Wikipedia entry
- Perform data wrangling
  - Data was filtered, reorganized, cleaned and null values were replaced.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Models were build using Scikit Learn library, then trained on a subset of the data (train). Accuracy evaluation was performed using a different subset of the data (test)

# Data Collection

- Data was collected performing two distinct methodologies:
- Requests made to a Spacex Rest API:
- A json file was obtained from a url that was then transformed into a dataframe using Pandas library.
- This dataframe was subsequently used to create a new dataframe, using functions that were previously defined. Data was filtered and null values were dealt with.
- Web Scraping from SpaceX Wikipedia entry:
- HTTP Method was performed, then a BeautifulSoup object was created from the HTTP response. A dataframe was then constructed using the Soup object.

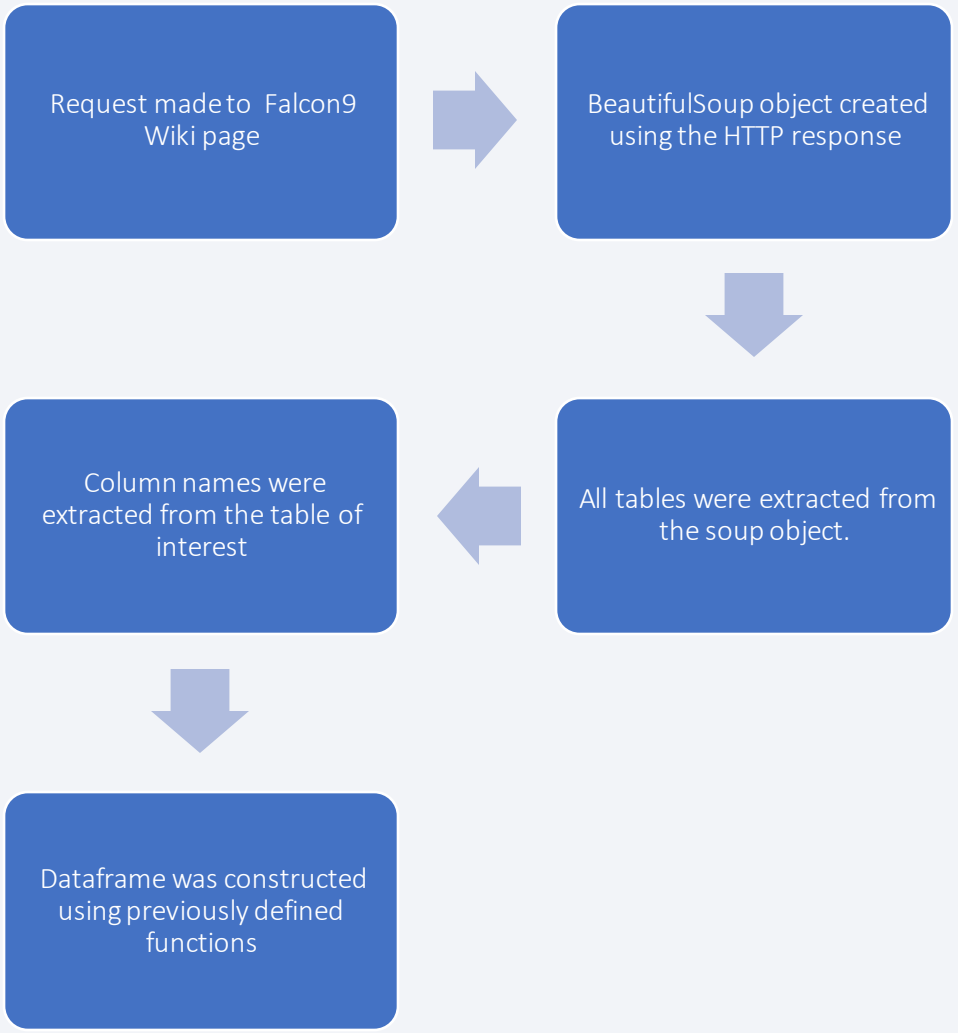
# Data Collection – SpaceX API

- https://github.com/brezsosa/IBM-Data-Science-Professional-Certificate/blob/96cf1be922f2e3cca350c8c4ef794128aa23d3fc/1.%20Collecting%20the%20Data.ipynb



# Data Collection - Scraping

- https://github.com/brezsosa/IBM-Data-Science-Professional-Certificate/blob/96cf1be922f2e3cca350c8c4ef794128aa23d3fc/2.%20Web%20Scraping.ipynb



# Data Wrangling

- A glimpse of the data was obtained using .head method.
- Types of the data values were assessed using dtypes method.
- Number of launches on each site were calculated.
- Number and occurrence of each  destination orbit for every launch were calculated.
- Number and occurrence of each mission outcome were calculated.
- A LandingClass outcome column was created from the Outcome column. LandingClass outcome column assigns a value of '0' to every failed mission outcome, and a value of '1' to every successful one.
- The average of successful mission outcomes was calculated from the LandingClass column.

# EDA with Data Visualization

- Several charts were used:

- Scatter point plots were constructed to visualize the relationship between two variables ("Flight no. Vs PayloadMass", "Flight no. Vs LaunchSite", "Flight no. Vs Orbit Type", and so on)

- Bar plot was constructed to visually check if there is any relationship between success rate and orbit type.

- Line plot was constructed to visualize the average launch success rate by year.

- https://github.com/brezsosa/IBM-Data-Science-Professional-Certificate/blob/96cf1be922f2e3cca350c8c4ef794128aa23d3fc/5.%20EDA%20with%20Visualization.ipynb

# EDA with SQL

- Several queries were performed:

- Unique launch sites in the space mission: %sql SELECT DISTINCT "launch_site" FROM "SPACEXTBL"
- 5 records where launch sites begin with the string 'CCA': %sql SELECT * FROM "SPACEXTBL" WHERE "launch_site" LIKE 'CCA%' LIMIT 5
- Total payload mass carried by boosters launched by NASA (CRS): %sql SELECT SUM("PAYLOAD_MASS__KG_") as Total_Payload_Mass FROM "SPACEXTBL" WHERE "Customer" == 'NASA (CRS)'
- Average payload mass carried by booster version F9 v1.1: %sql SELECT AVG("PAYLOAD_MASS__KG_") as Mean_Payload_Mass FROM "SPACEXTBL" WHERE "Booster_Version" LIKE '%F9 v1.1%'
- Date when the first succesful landing outcome in ground pad was achieved: %sql SELECT MAX(DATE) as First_Succesful_Landing FROM "SPACEXTBL" WHERE Landing_Outcome = 'Success (ground pad)'
- 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" FROM "SPACEXTBL" WHERE Landing_Outcome = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000

# EDA with SQL

- Several queries were performed:

- **Total number of successful and failure mission outcomes**: **%sql** SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Count FROM "SPACEXTBL" GROUP BY "Mission_Outcome"

- **Names of the booster_versions which have carried the maximum payload mass**: %sql SELECT DISTINCT "Booster_Version" FROM "SPACEXTBL" WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM "SPACEXTBL")

- **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", 4, 2) as Month, Landing_Outcome, "Booster_Version", "Launch_Site" FROM "SPACEXTBL" WHERE substr("Date", 7, 4) = '2015' AND "Landing_Outcome" = 'Failure (drone ship)'

- **Count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order**: %sql SELECT Landing_Outcome, COUNT(Landing_Outcome) as Count FROM (SELECT Date, Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success%' limit 8) GROUP BY Landing_Outcome ORDER BY Count DESC

- https://github.com/brezsosa/IBM-Data-Science-Professional-Certificate/blob/96cf1be922f2e3cca350c8c4ef794128aa23d3fc/4.%20EDA%20with%20SQL.ipynb

# Build an Interactive Map with Folium

- Markers were added to an interactive map to tag every launch site.
- A cluster of markers were added to each launch site to mark all successful launches (in green) and all the failed ones (in red).
- Distances from a launch site to its proximities (coastline, railroad, highway, city) were calculated. A marker was added to each proximity and a line was drawn to show the distance from the launch site to each place.
- https://github.com/brezsosa/IBM-Data-Science-Professional-Certificate/blob/96cf1be922f2e3cca350c8c4ef794128aa23d3fc/6.%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb

# Build a Dashboard with Plotly Dash

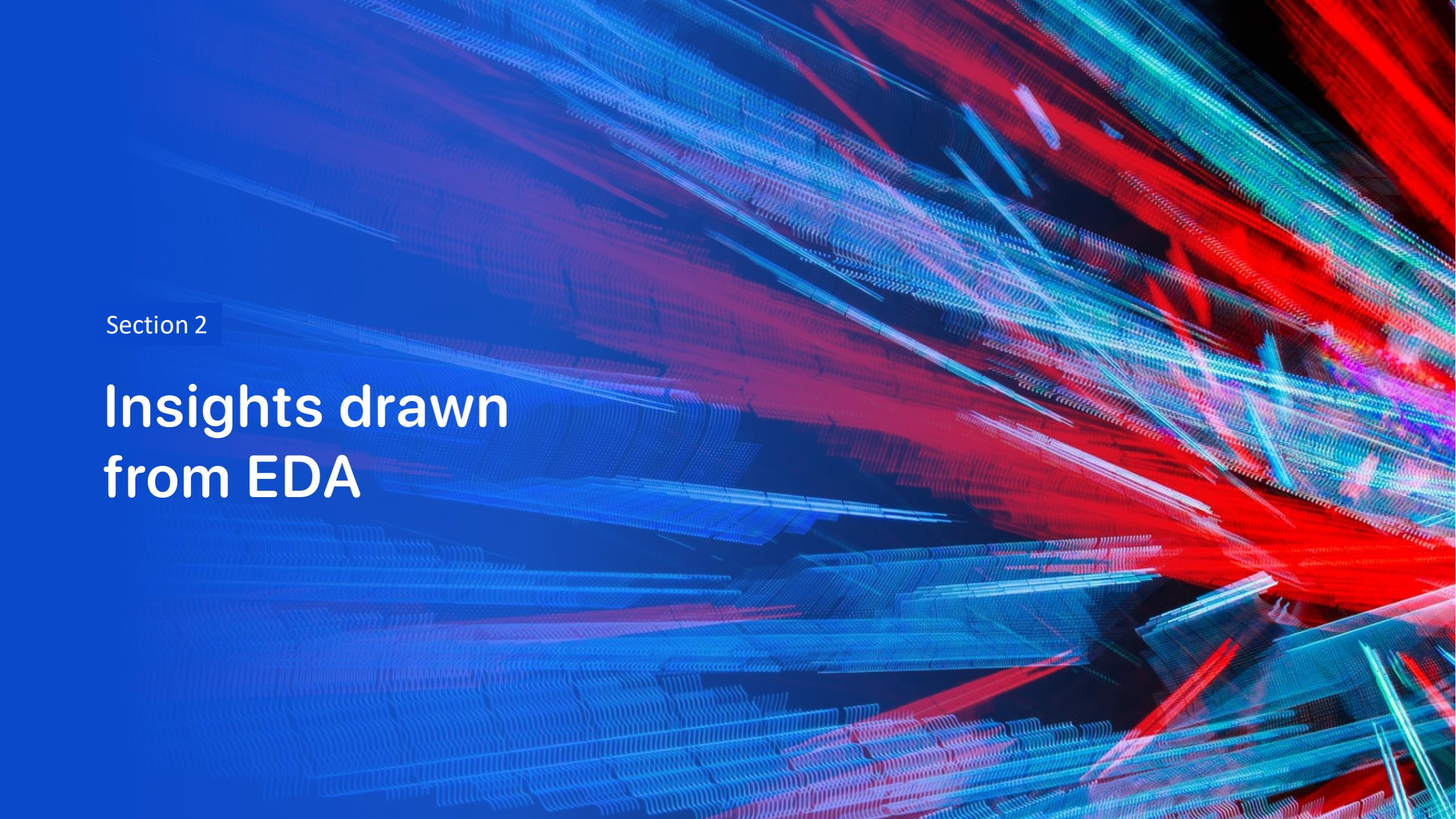
- A pie chart displaying successful/failed launch rates was constructed in a dashboard to show the rates for each individual launch site, and for all the launch sites together.
- A scatter point chart was constructed to show the relationship between PayloadMass and Success rate for each Booster Version. The dashboard permits to show the Success rate for different values of PayloadMass.
- https://github.com/brezsosa/IBM-Data-Science-Professional-Certificate/blob/96cf1be922f2e3cca350c8c4ef794128aa23d3fc/7.%20spacex_dash_app.py

# Predictive Analysis (Classification)

- Several Classification models were constructed using Scikit Learn library: logistic regression, SVM, TreeClassifier and a KNN model. The aim was to correctly predict the successful launch outcomes based on the features previously analyzed (such as Payload Mass, launch site, orbit type, and so on).
- Every model was trained on a subset of the data (training data) and then evaluated using a different subset of the data (test data).
- The accuracy score of each model was assessed using test data. Confusion matrices were constructed for each model as a part of the evaluation.
- The accuracy of the models was plotted in a bar chart to visualize the best performing model (the one with highest accuracy score).
- https://github.com/brezsosa/IBM-Data-Science-Professional-Certificate/blob/96cf1be922f2e3cca350c8c4ef794128aa23d3fc/8.%20Machine%20Learning%20Prediction%20Lab.ipynb

Section 2

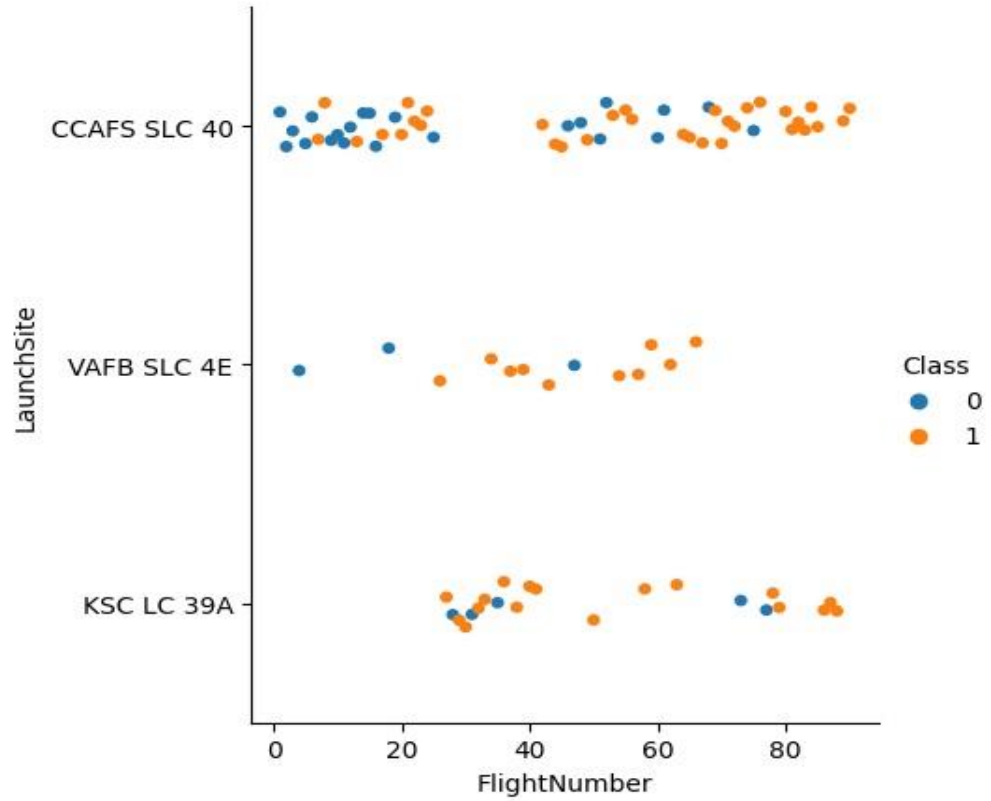
# Insights drawn from EDA



# Flight Number vs. Launch Site

In [4]:
```
# 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 valu
sns.catplot(y = 'LaunchSite', x = 'FlightNumber', hue = 'Class', data = df)
```

Out[4]: <seaborn.axisgrid.FacetGrid at 0x26876c81890>

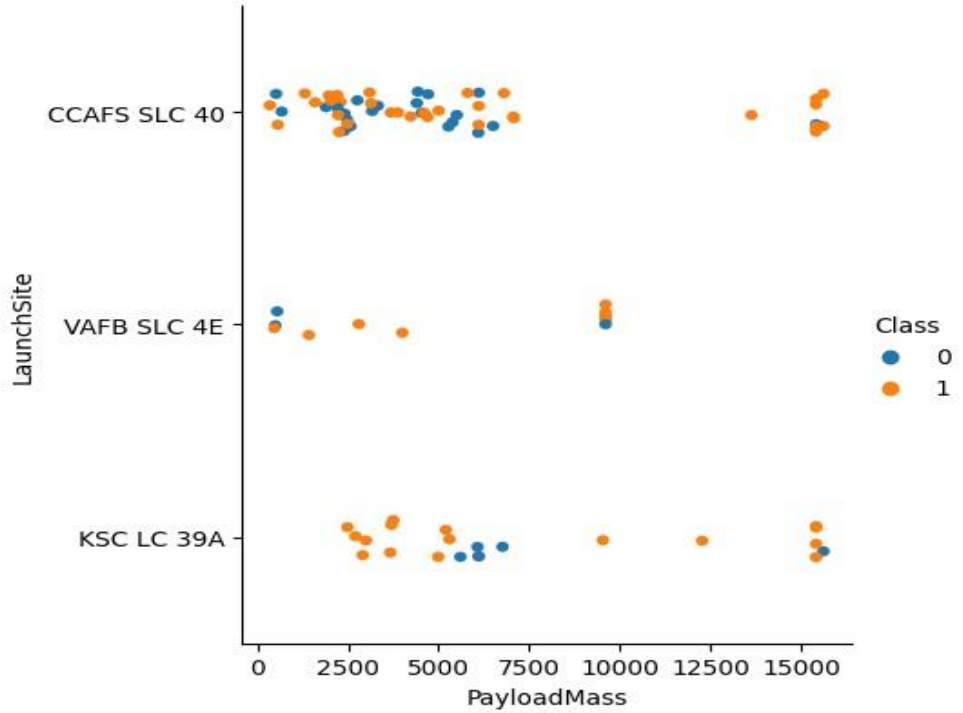


# Payload vs. Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

In [6]:
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class
sns.catplot(x = 'PayloadMass', y = 'LaunchSite', hue = 'Class', data = df)
```

Out[6]: <seaborn.axisgrid.FacetGrid at 0x2687713f810>

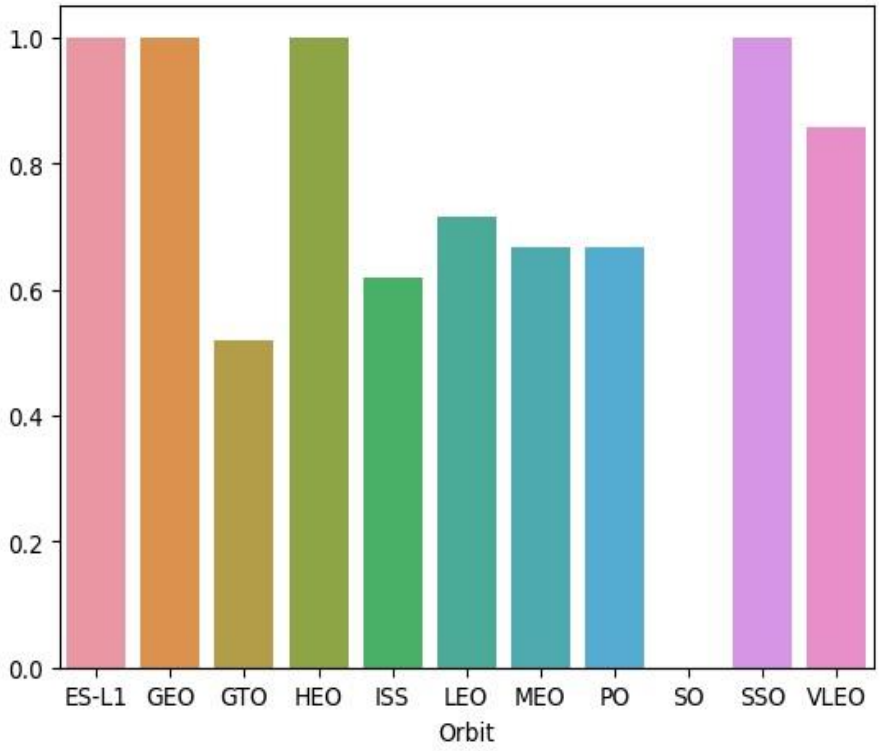



Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

# Success Rate vs. Orbit Type

In [35]:
```
# HINT use groupby method on Orbit column and get the mean of Class column
mean_class = df.groupby('Orbit')['Class'].mean()
sns.barplot(x = mean_class.keys(), y = mean_class.values)
```

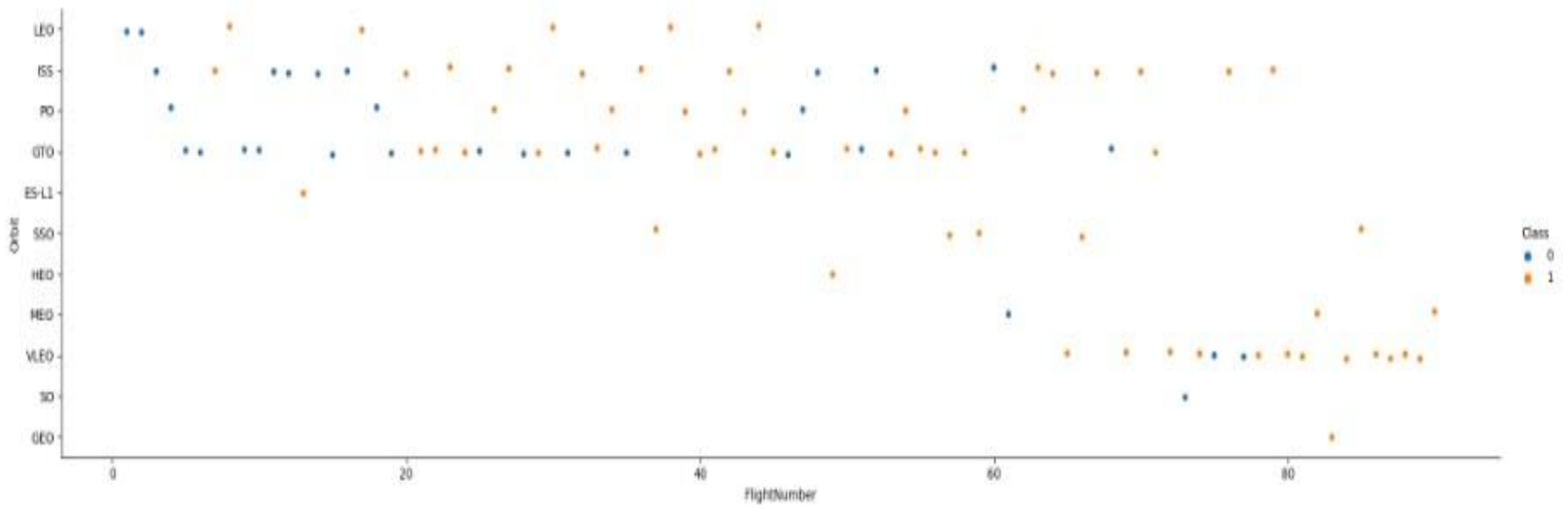
Out[35]: <Axes: xlabel='Orbit'>



# Flight Number vs. Orbit Type

In [41]:
```
# 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(x = 'FlightNumber', y = 'Orbit', hue = 'Class', data = df, aspect = 4)
```

Out[41]: <seaborn.axisgrid.FacetGrid at 0x2687bdc66d0>

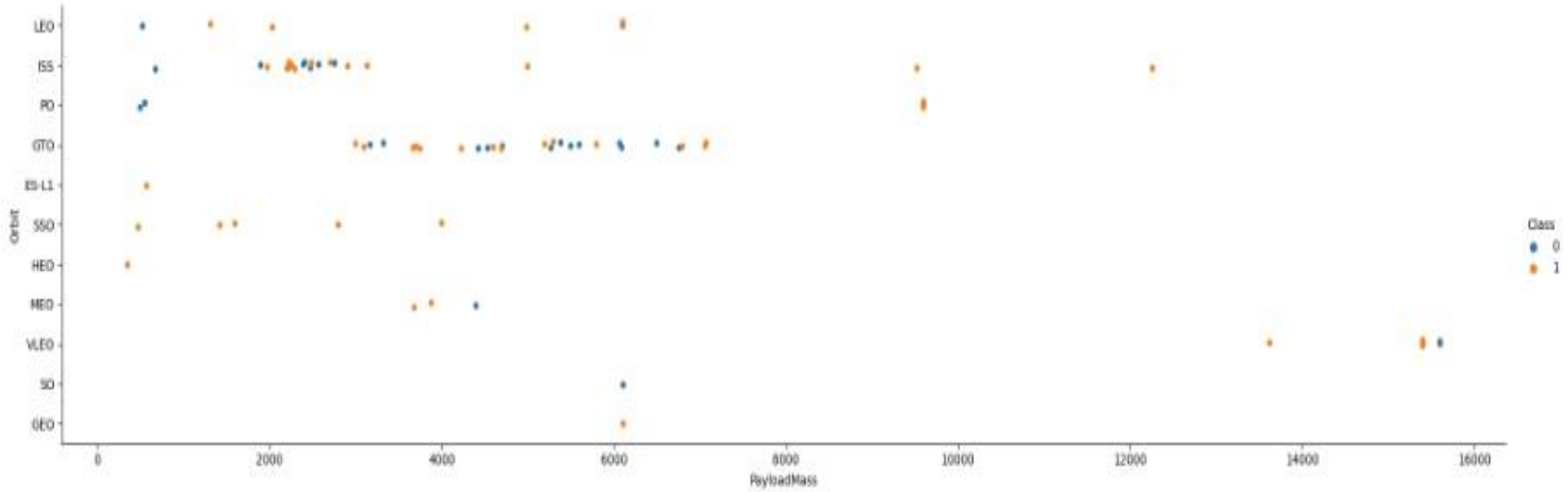



You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

# Payload vs. Orbit Type

In [42]:
```
# 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 = 4)
```

Out[42]: <seaborn.axisgrid.FacetGrid at 0x2687b342dd0>



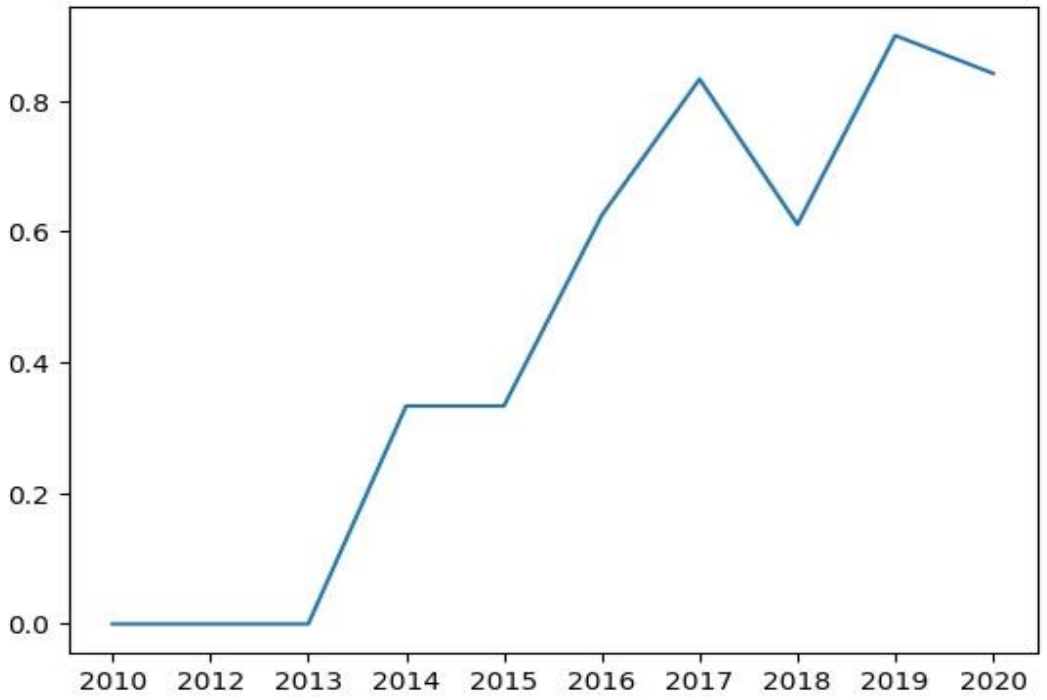

With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

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

In [60]:
```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
mean_success = df.groupby(year)['Class'].mean()
sns.lineplot(x = mean_success.keys(), y = mean_success.values)
```

Out[60]: <Axes: >




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

# All Launch Site Names

## Task 1

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

In [80]:
```
%sql SELECT DISTINCT "launch_site" FROM "SPACEXTBL"
```

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

Out[80]:

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

# Launch Site Names Begin with 'CCA'

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

In [81]:
```
%sql SELECT * FROM "SPACEXTBL" WHERE "launch_site" LIKE 'CCA%' LIMIT 5
```

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

Out[81]:

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 04-06-2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 08-12-2010 | 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) |
| 22-05-2012 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 08-10-2012 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 01-03-2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

# Total Payload Mass

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

In [58]:
```
%sql SELECT SUM("PAYLOAD_MASS__KG_") as Total_Payload_Mass FROM "SPACEXTBL" WHERE "Customer" == 'NASA (CRS)'
```

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

Out[58]:

| Total_Payload_Mass |
|---|
| 45596 |

# Average Payload Mass by F9 v1.1

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

In [51]:
```
%sql SELECT AVG("PAYLOAD_MASS__KG_") as Mean_Payload_Mass FROM "SPACEXTBL" WHERE "Booster_Version" LIKE '%F9 v1.1%'
```

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

Out[51]:

| Mean_Payload_Mass |
|---|
| 2534.6666666666665 |

# First Successful Ground Landing Date

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

*Hint:Use min function*

In [121…

```
%sql SELECT MAX(DATE) as First_Succesful_Landing FROM "SPACEXTBL" WHERE Landing_Outcome = 'Success (ground pad)'
```

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

Out[121…

| First_Succesful_Landing |
| --- |
| 22-12-2015 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

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

In [90]:
```
%sql SELECT DISTINCT "Booster_Version" FROM "SPACEXTBL" WHERE Landing_Outcome = 'Success (drone ship)' AND "PAYLOAD_MASS__K
```

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

Out[90]:

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

# Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [91]:
```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Count FROM "SPACEXTBL" GROUP BY "Mission_Outcome"
```

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

Out[91]:

| Mission_Outcome | Count |
|---|---|
| Failure (in flight) | 1 |
| Success | 98 |
| Success | 1 |
| Success (payload status unclear) | 1 |

# Boosters Carried Maximum Payload

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

In [92]:
```
%sql SELECT DISTINCT "Booster_Version" FROM "SPACEXTBL" WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM "
```

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

Out[92]:

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

**Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.**

In [93]:
```
%sql SELECT substr("Date", 4, 2) as Month, Landing_Outcome, "Booster_Version", "Launch_Site" FROM "SPACEXTBL" WHERE substr('
```

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

Out[93]:

| Month | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 01 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | 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 successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

In [120…

```
%sql SELECT Landing_Outcome, COUNT(Landing_Outcome) as Count FROM (SELECT Date, Landing_Outcome FROM SPACEXTBL WHERE Landing
```

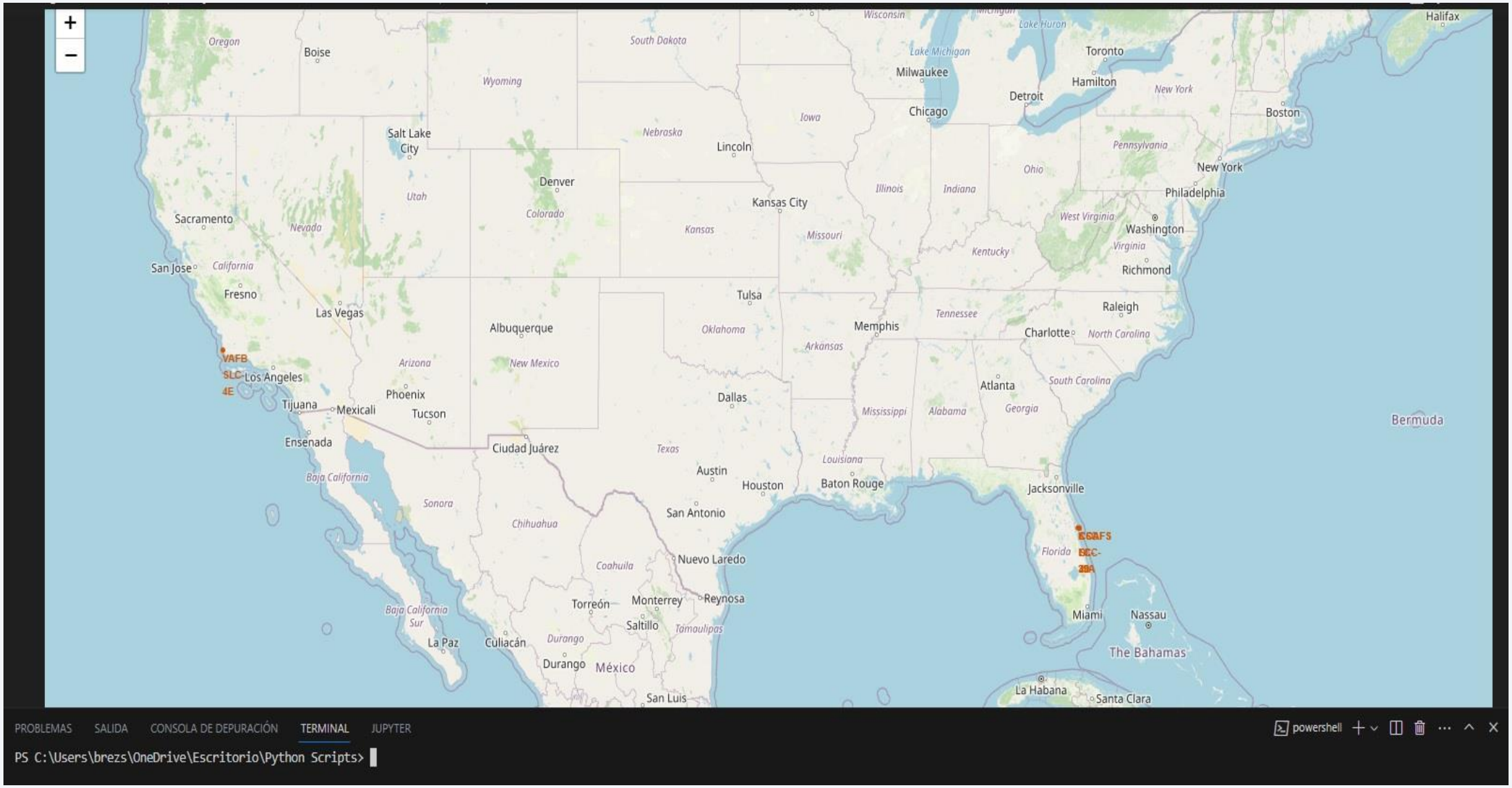
```
* sqlite:///my_data1.db
Done.
```

Out[120…

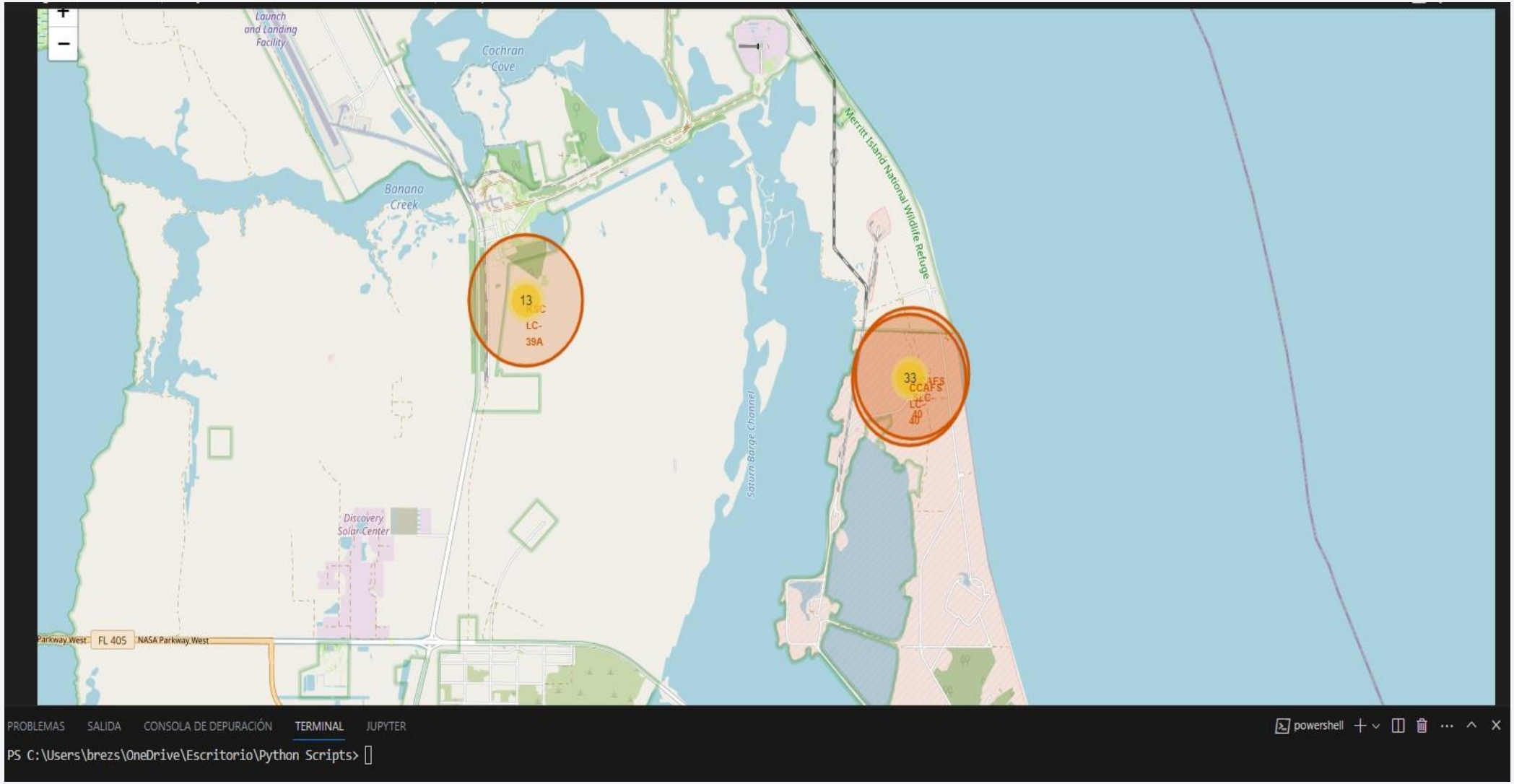
| Landing_Outcome | Count |
|---|---|
| Success (drone ship) | 5 |
| Success (ground pad) | 3 |


Section 3
Launch Sites
Proximities Analysis

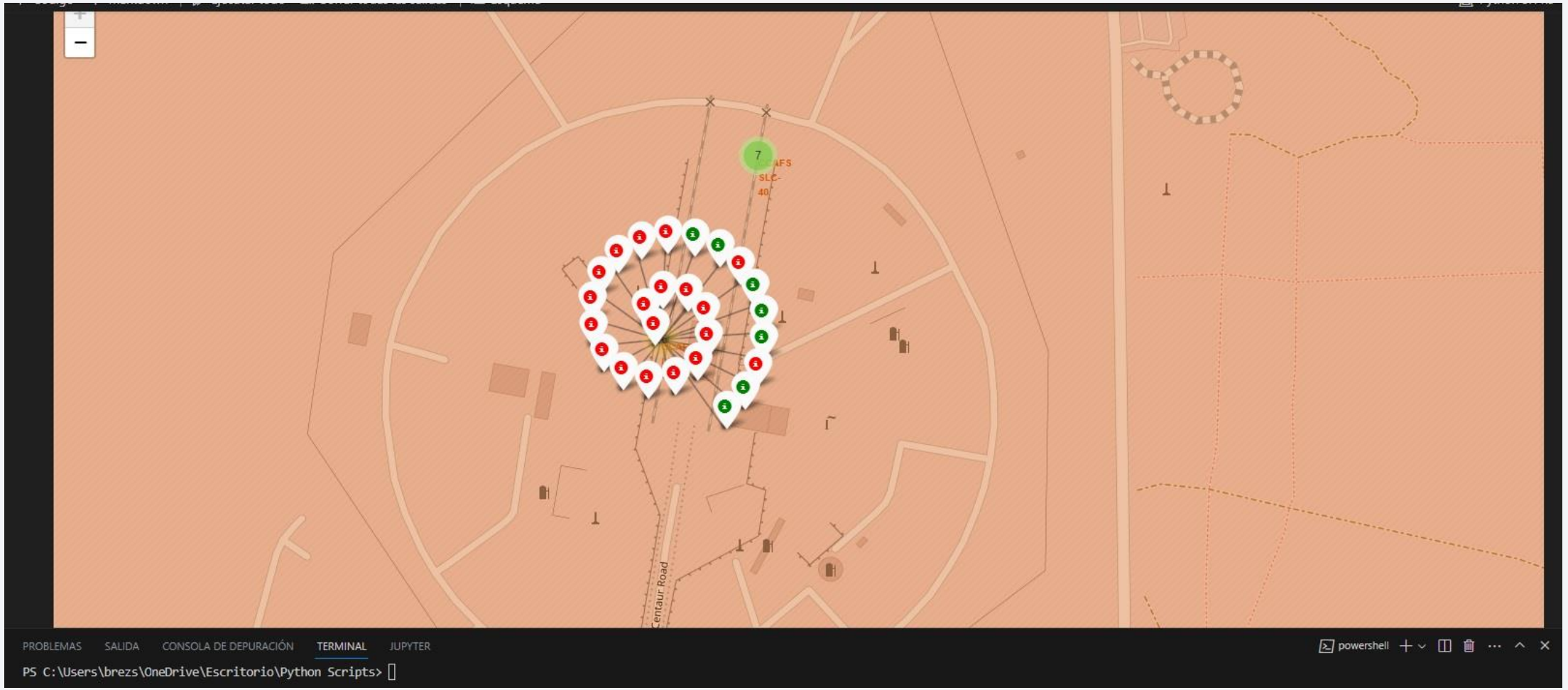
# Launch Sites Location on a Map



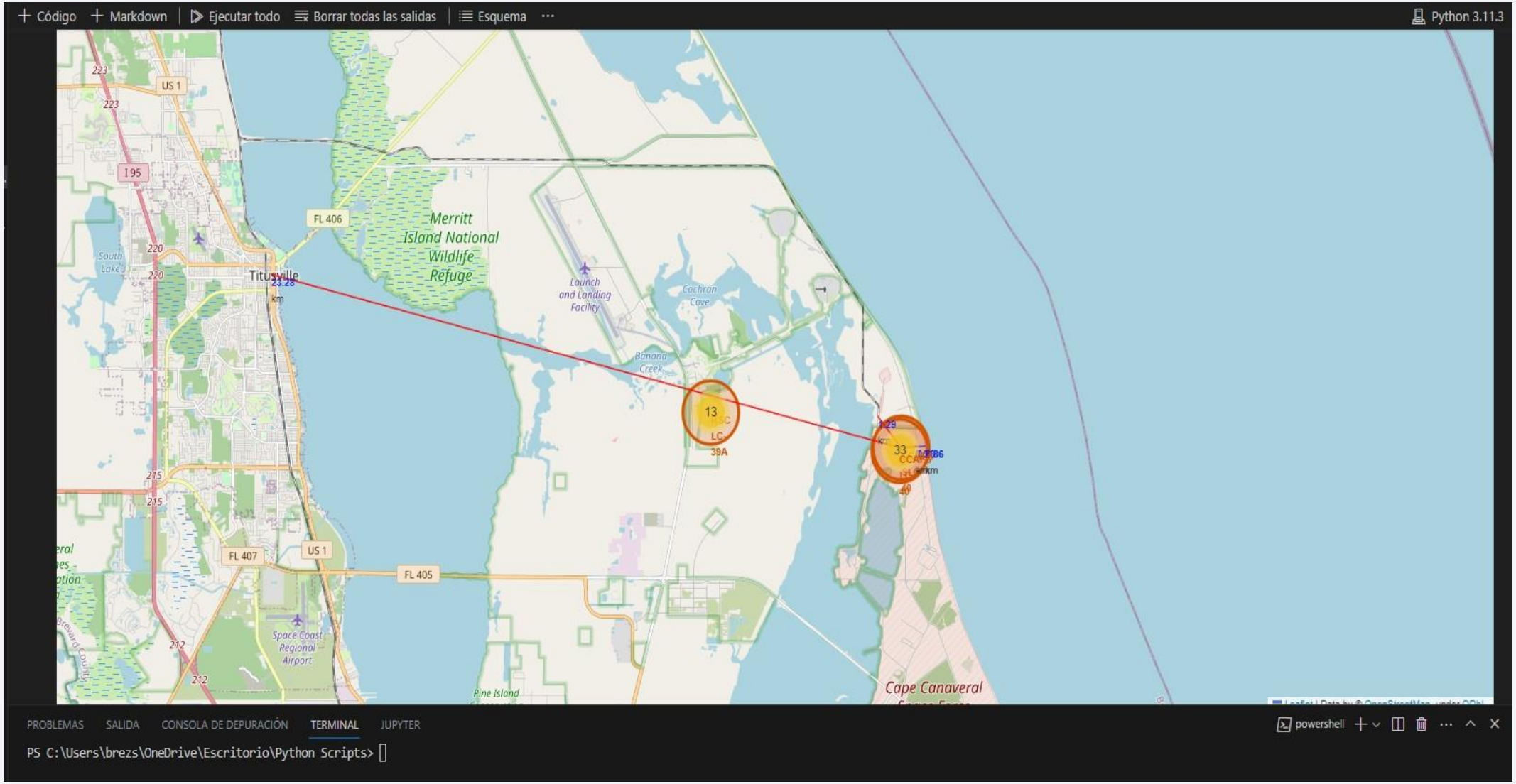
# Launch Outcomes By Launch Site



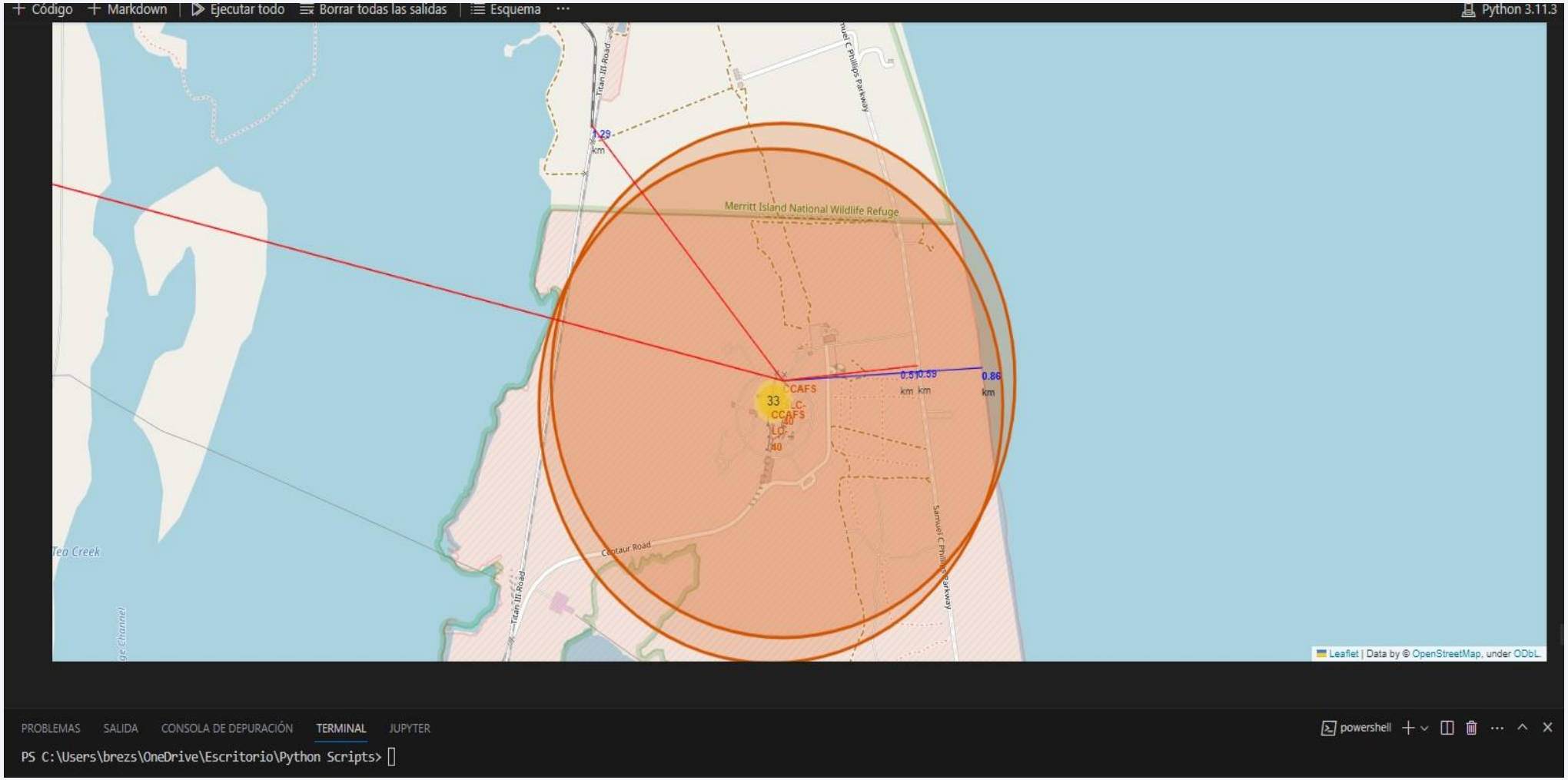
# Launch Outcomes By Launch Site

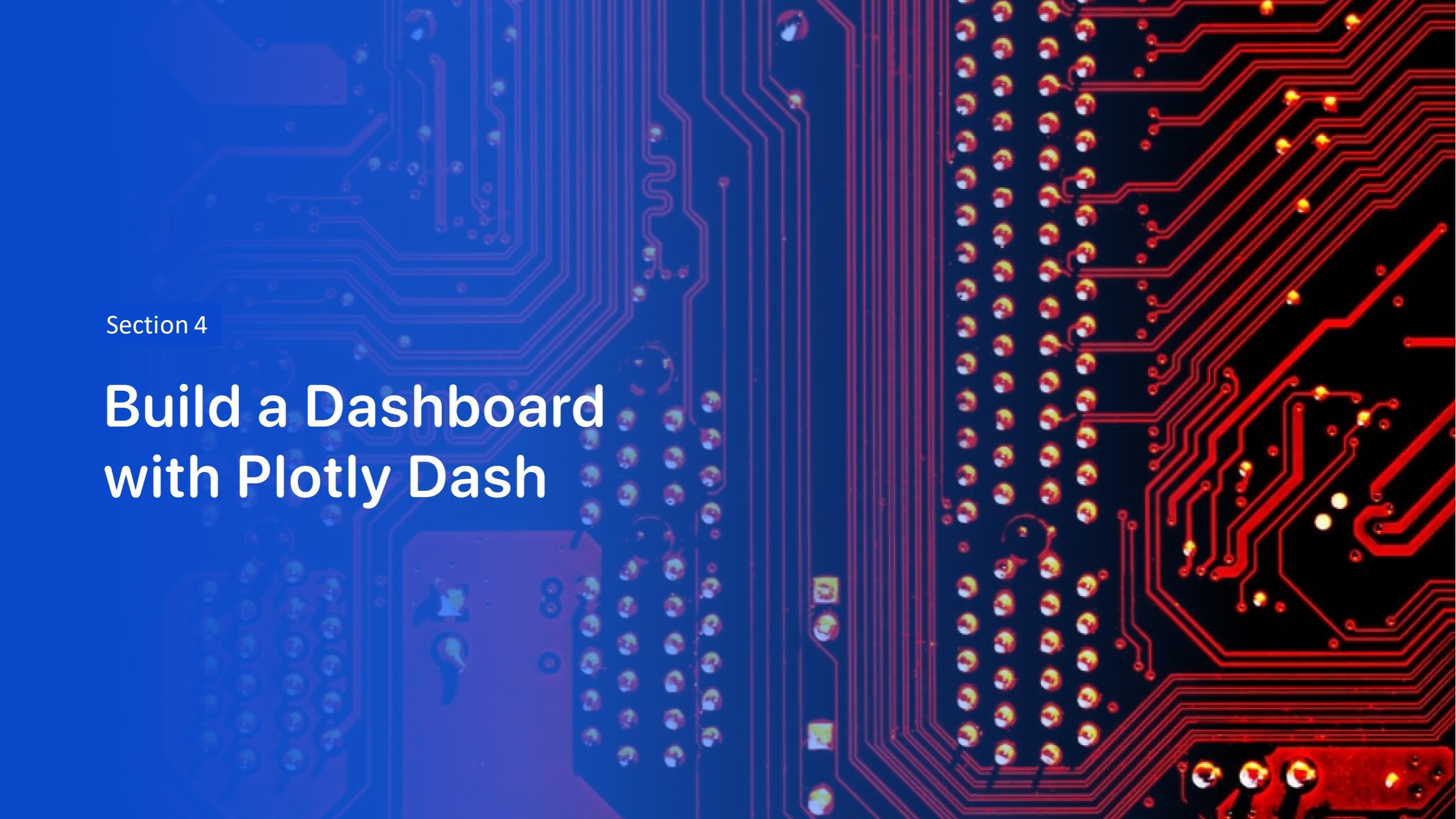


# Distance from Launch Site to Proximities

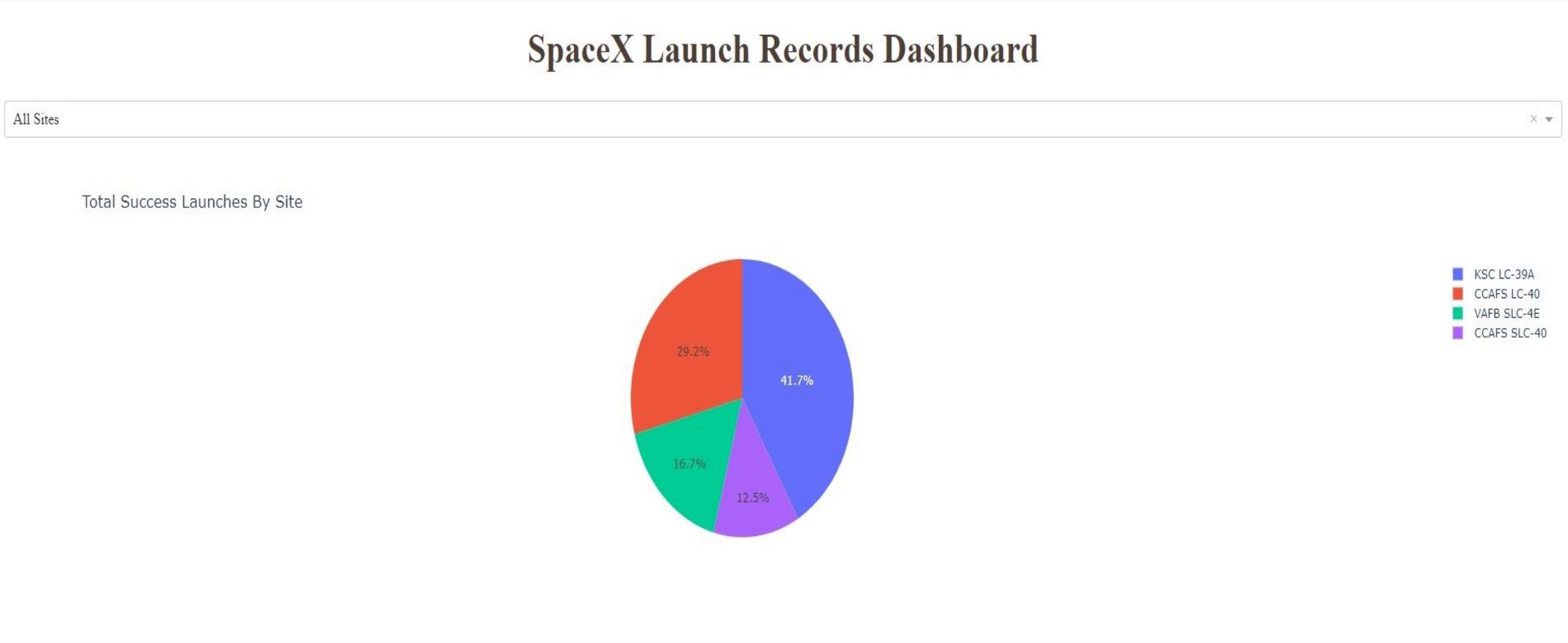


# Distance from Launch Site to Proximities

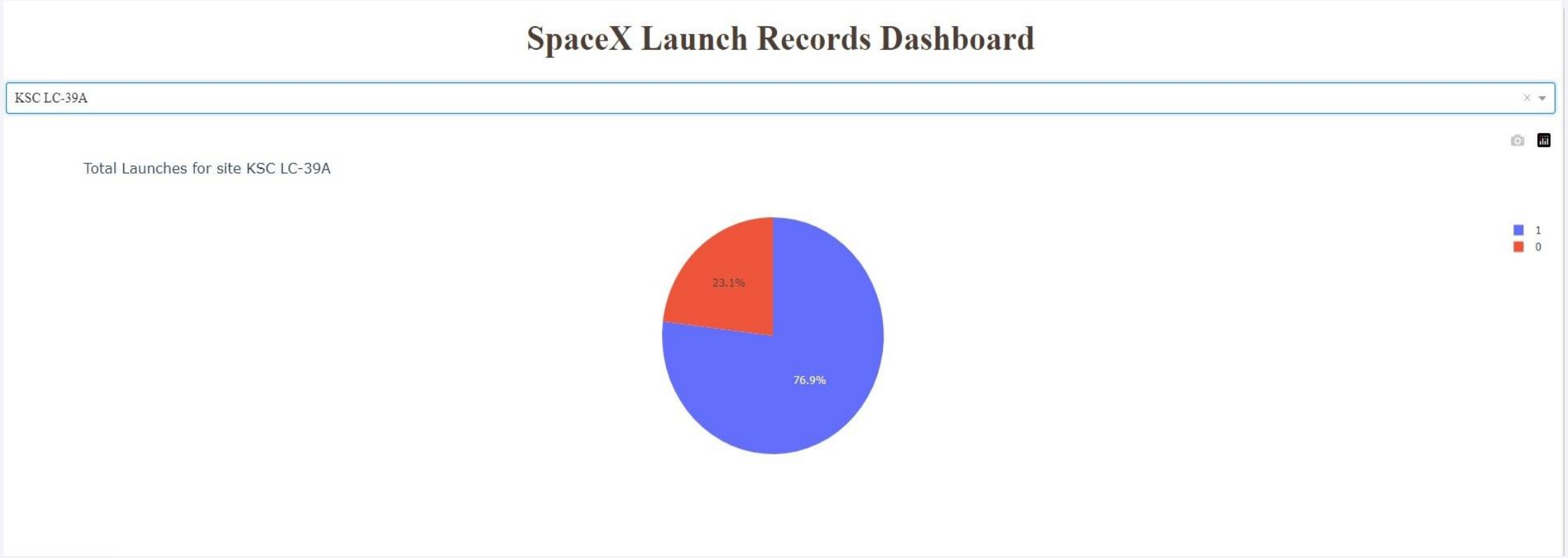



Section 4
Build a Dashboard with Plotly Dash

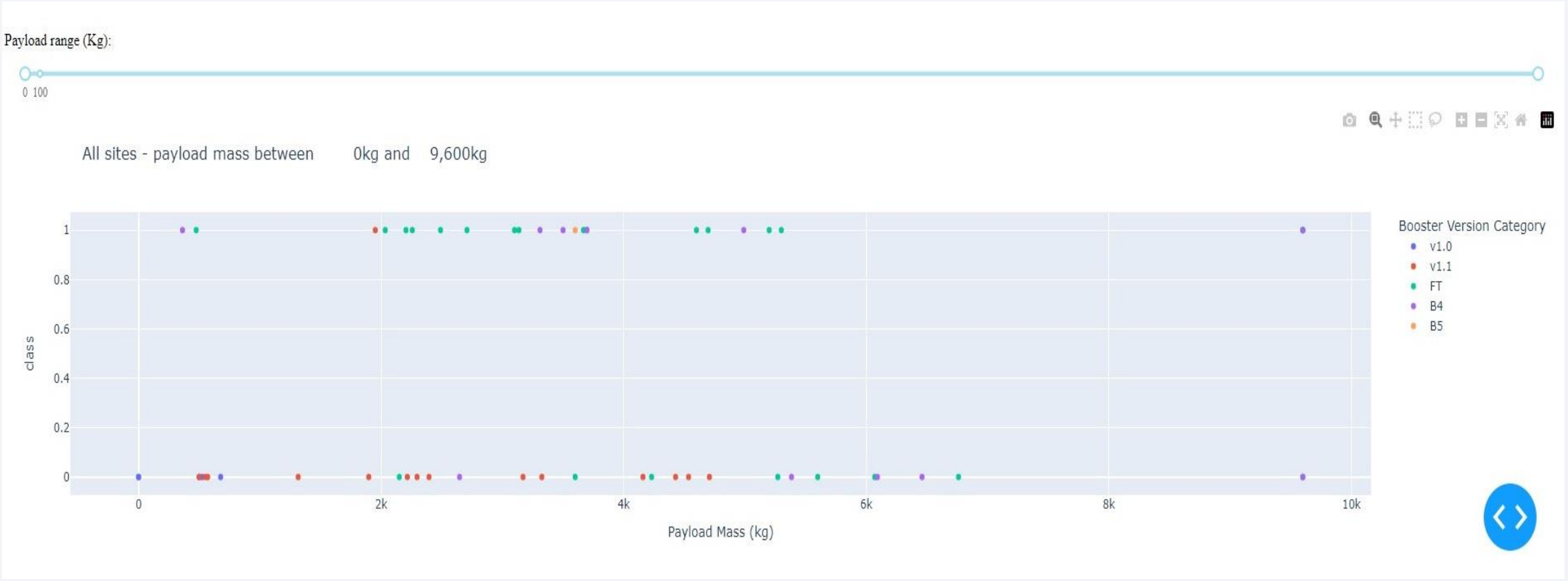
# Total Success Launches By Site



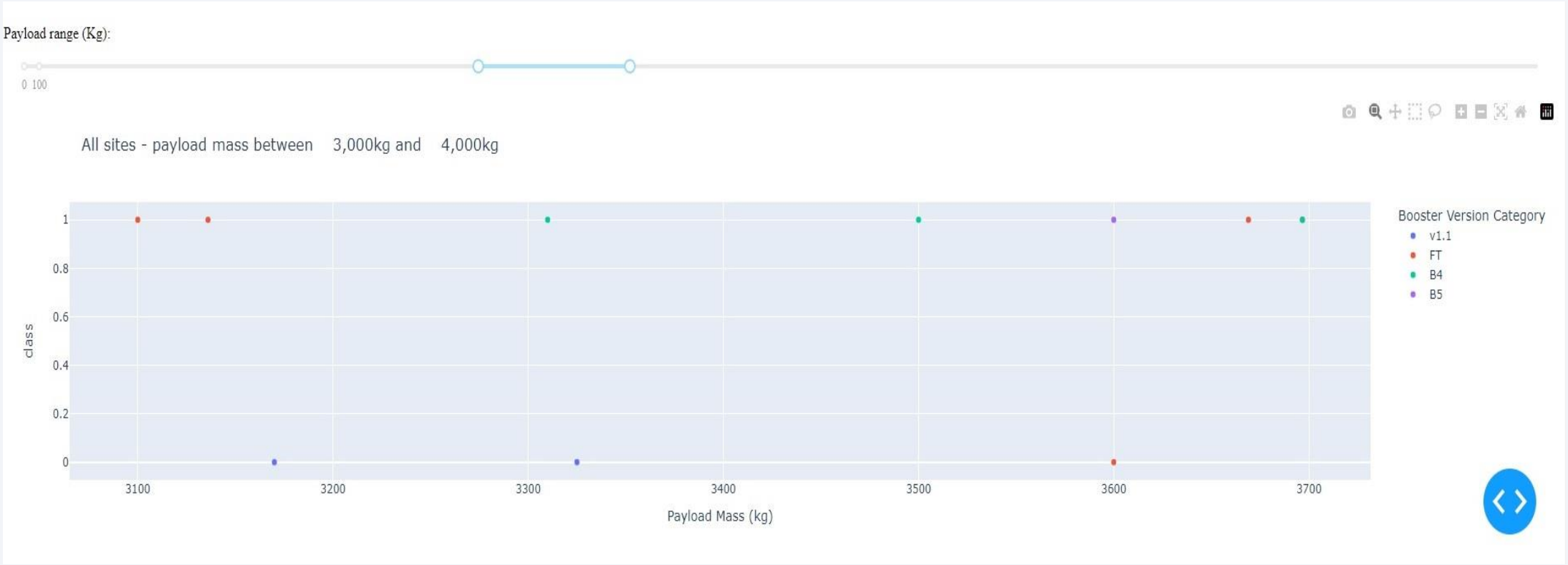
# Launch Site With Highest Launch Success Ratio



# Payload Mass Vs. Launch Outcome

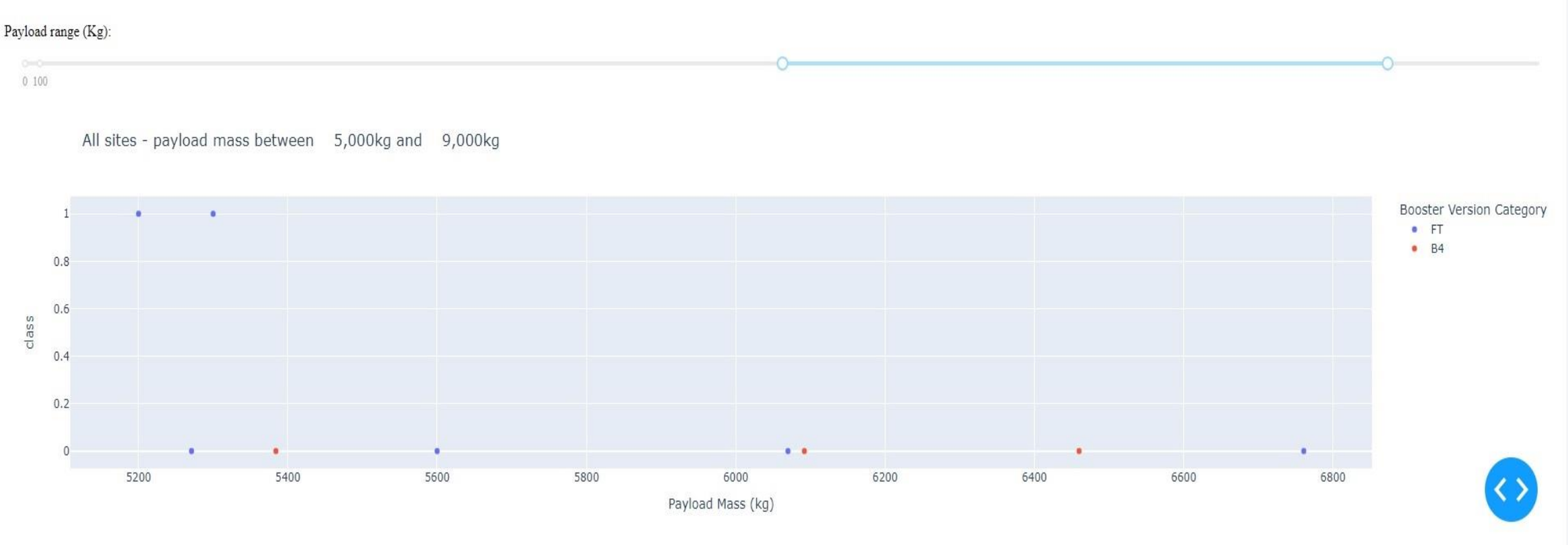


# Payload Mass Vs. Launch Outcome




Payload Mass range with maximum success rate

# Payload Mass Vs. Launch Outcome




Payload Mass range with minimum success rate

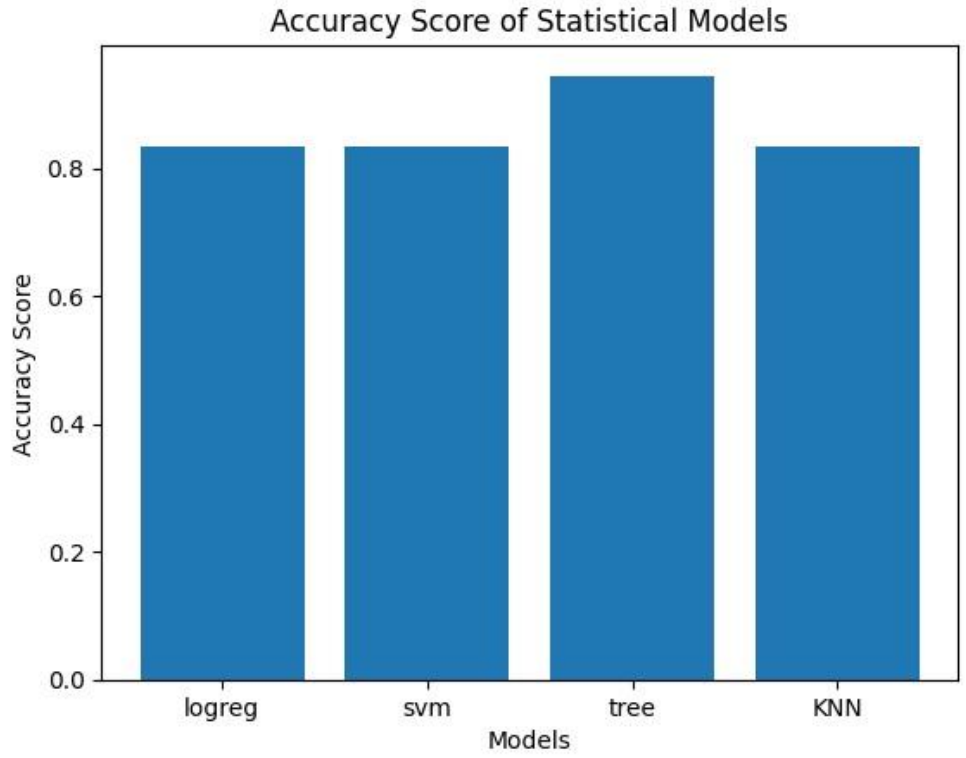
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

In [46]:

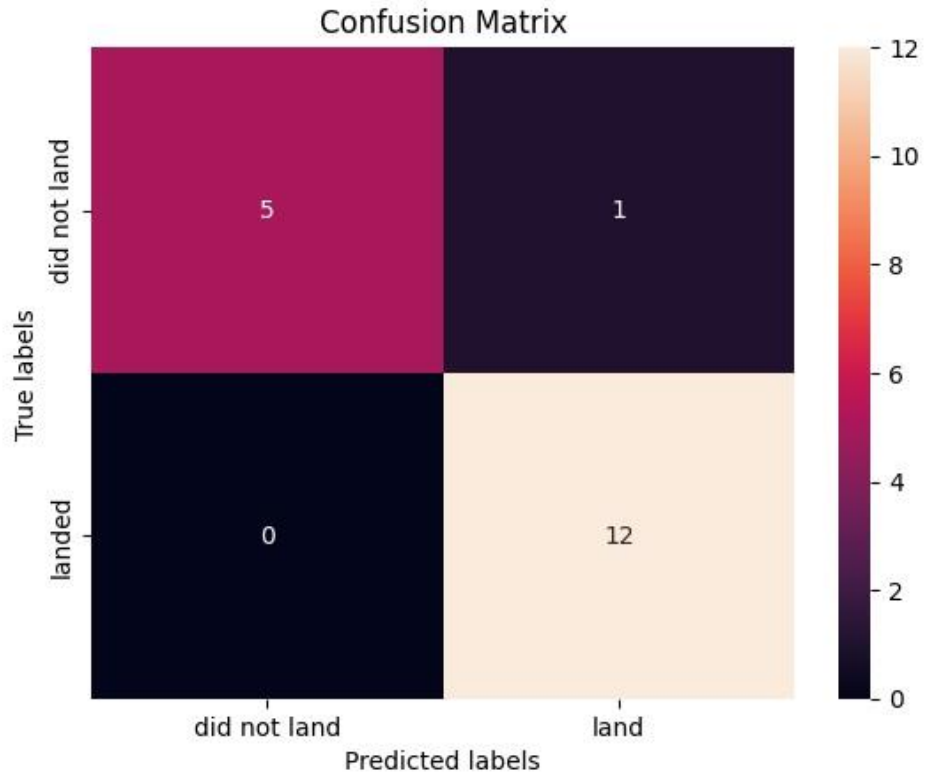
```
dic = {'logreg': logreg_cv.score(X_test, Y_test), 'svm': svm_cv.score(X_test, Y_test), 'tree': tree_cv.score(X_test, Y_te
x = dic.keys()
y= dic.values()
plt.bar(x, y)
plt.title('Accuracy Score of Statistical Models')
plt.xlabel('Models')
plt.ylabel('Accuracy Score')
plt.show()
```




- Tree Classifier shows the highest accuracy in test sample. However, size of the test sample was too small to draw strong conclusions (N = 18).

# Confusion Matrix for best model

```
In [26]: yhat = tree_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```




- True positives = 12 + 5 = 17
- False positives = 0 + 1 = 1
- Precision = 12/ 12+1 = 0.92
- Recall = 12/ 12+0 = 1

# Conclusions

- TreeClassifier seems to be the most accurate model to predict successful mission outcomes. However, test sample sizes were too small (N = 18) to draw strong conclusions regarding the evaluation of the models.
- Launch sites are mostly located near coastlines, but far from cities.
- Payload Mass between 3,000 kg and 4,000 kg seems to have the highest success rate.
- Launch site KSC LC-39A has the highest success rate in launch records.

# Thank you!