IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Arjun Virk

December 10th, 2023

Outline

- Executive Summary: Slide #3
- Introduction: Slide #4
- Methodology: Slides #5 - #15
- Results: Slides #16 - #44
- Conclusion: Slide #45
- Appendix: Slide #46

Executive Summary

- In this presentation, we will start with discussing the methodology for assessing this SpaceX data and finding meaningful results. Upon Data collection, we highlight the processing which we pre-process the data (Webscraping and Data Wrangling). Following this, we detail the process through which we perform Exploratory Data Analysis (using various plots as well as SQL queries). Then we will describe using a flow chart how a folium map and Dash Web application were created to further evaluate the data set. Finally, we discuss the predictive analysis that was done to determine the best classification model for this dataset
- All of the plots and SQL queries (as well as their results) are shown from Exploratory Data Analysis. In addition, 3 screenshots of the Folium map as well as screenshots from the built Dash Web application are also shown. Finally, a comparison of the accuracy of various classification models as part of a predictive analysis is shown, as well as a confusion matrix to showcase the classification performance.

Introduction

- This project involves looking at the launch data records from a space company (Space X). The data was previously collected and the goal of this project is to process the data (webscraping, data wrangling), create a series of visuals and queries (plots, SQL commands), create interactive visuals to compare different launch sites (Folium and Dash) and ultimately create predictive models using machine learning techniques.
- We wish to explore the impact of various factors on launch success at four different launch sites, particularly at the 1st stage. These factors include orbit type, payload mass and improvement of flight design over time (i.e. the flight numbers). We evaluate the relationship of these factors with launch success at different locations in the hopes of making machine learning models that can predict the outcome of a launch (whether it's a success or failure) based purely on these factors.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was pulled from a URL and eventually converted into a pandas dataframe after webscraping
- Perform data wrangling
  - Data was processed by using the value_counts() method to create a “outcome” label for each launch
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - After standardizing and splitting the data into training and test sets, 4 classification models were trained using the training data. GridSearchCV was used to find the best parameters for each model, after which the accuracy score was found using the test data.

Data Collection

- Data was collected from the following URL Document:
  - "https://api.spacexdata.com/v4/launches/past"

- The next several slides shows the breakdown of the the data collection processing using rest API and webscraping

Data Collection – SpaceX API

SpaceX Data is pulled from a URL using the request.get() method.

The response from the website was then normalized as a JSON object using the json_normalize() function. A subset of the useful data was then taken and the data was passed through several prebuilt functions to get the relevant columns (i.e Booster Version, Launch Site, payload etc.)

The dataframe was filtered to only include Falcon 9 data as well as replace the missing values of the payload mass column with the average value

URL to Github: https://github.com/ASVirk-SBU/ArjunVirk_DataScience_Capstone/blob/main/DataCollection.ipynb
File name: DataCollection.ipynb

Data Collection - Scraping

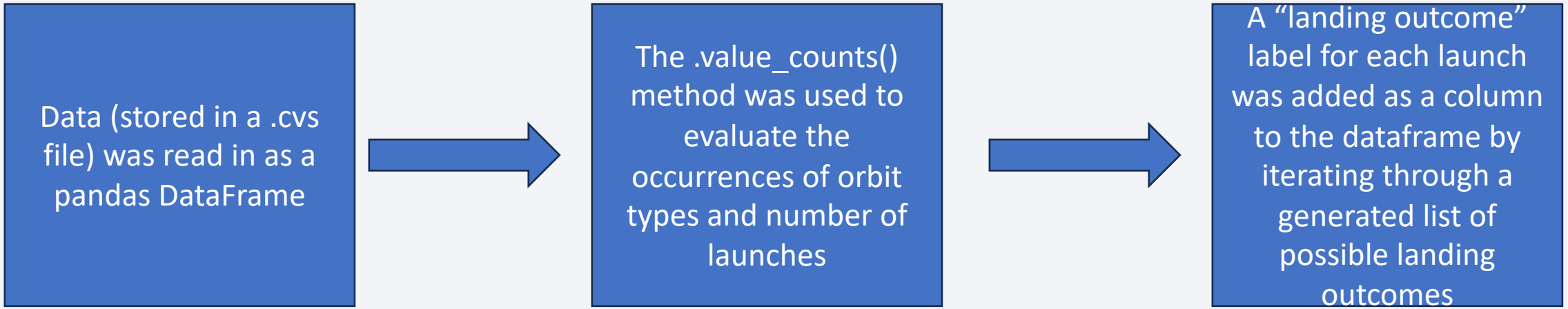
Data is pulled from a URL using the request.get() method.

The data from the response is parsed through using a BeautifulSoup Object. Each column of relevant data from the parsed text was extracted

The extracted columns (in the form of HTML tables) were used to create a pandas dataframe, which can be easily analyzed.

URL to Github: https://github.com/ASVirk-SBU/ArjunVirk_DataScience_Capstone/blob/main/Webscraping.ipynb
File name: Webscraping.ipynb

Data Wrangling



URL to Github: https://github.com/ASVirk-SBU/ArjunVirk_DataScience_Capstone/blob/main/DataWrangling.ipynb
File name: DataWrangling.ipynb

EDA with Data Visualization

- Scatter plot to shown relationship between Flight Number vs. Launch Site
- Scatter plot to shown relationship between Payload vs. Launch Site
- Bar plot to show relationship between Success Rate vs. Orbit Type
- Scatter plot to shown relationship between Flight Number vs. Orbit Type
- Scatter plot to shown relationship between Payload vs. Orbit Type
- Line plot to show Launch Success Yearly Trend

URL to Github: https://github.com/ASVirk-SBU/ArjunVirk_DataScience_Capstone/blob/main/EDA_DataVisualization.ipynb
File name: EDA_DataVisualization.ipynb

EDA with SQL

- Determine all unique launch site names
- Query all launch records that begin with “CCA”
- Find total payload mass from all launches
- Find the average payload mass for the F9 v1.1 Booster version
- Determine the date for the First Successful Ground Landing
- Find the booster version for all successful drone ship landing with Payload between 4000 and 6000 kg
- Determine total number of successful and failure mission outcomes
- Which boosters carried the maximum payload
- 2015 launch records that ended in failure
- Rank the landing outcomes between 2010-06-04 and 2017-03-20 in descending order

URL to Github: https://github.com/ASVirk-SBU/ArjunVirk_DataScience_Capstone/blob/main/EDA_SQL.ipynb
File name: EDA_SQL.ipynb

Build an Interactive Map with Folium

Data (stored in a .cvs file) was read in as a pandas DataFrame and an map object was initialized that showed a map of the USA

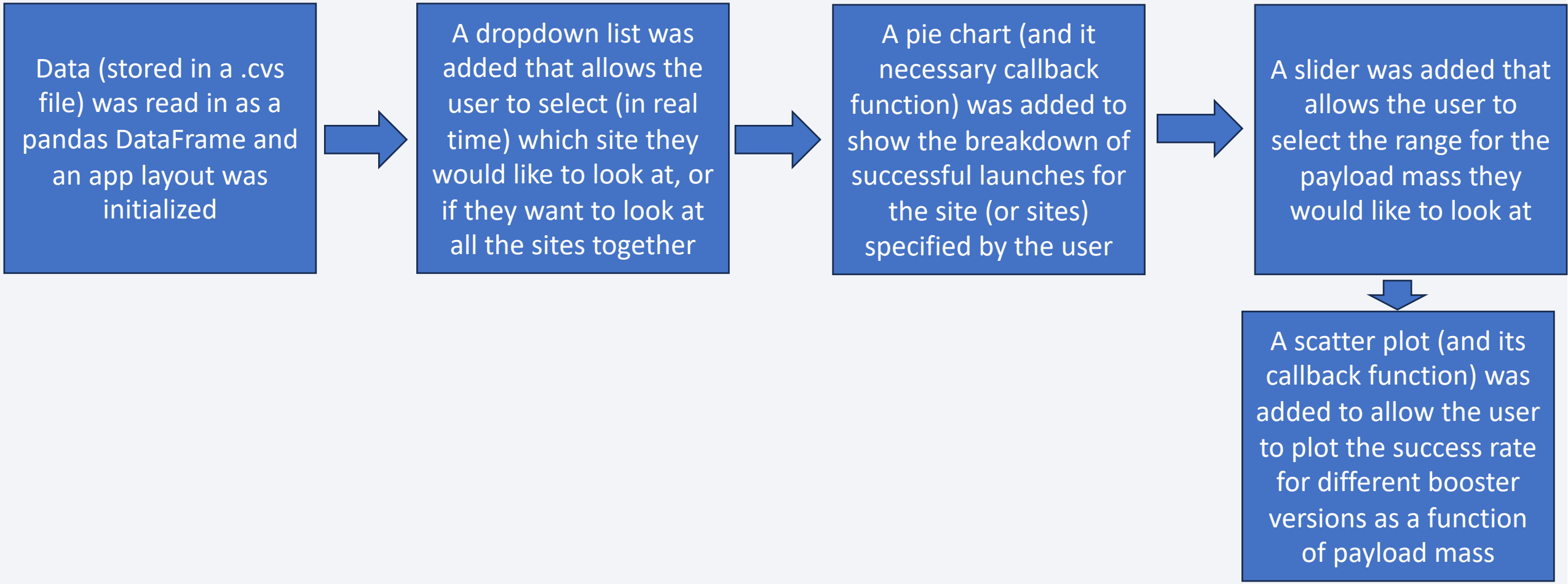
Markers were added that showed the location of the 4 launch sites in the USA on the same map.

A marker cluster object was created that allowed us to specify points at each launch site that were either a success or failure

Additional markers were added to locate various landmarks (including the coastline, roads, highways, and towns) as well as the distance there points are from the launch site, for safety reasons (i.e. population safety, methods of transportation etc.)

URL to Github: https://github.com/ASVirk-SBU/ArjunVirk_DataScience_Capstone/blob/main/LaunchSiteFolium.ipynb
File name: LaunchSiteFolium.ipynb

Build a Dashboard with Plotly Dash



URL to Github: https://github.com/ASVirk-SBU/ArjunVirk_DataScience_Capstone/blob/main/SpaceX_DashApplication.ipynb
File name: SpaceX_DashApplication.ipynb

Predictive Analysis (Classification)

Dataframe was loaded in and standardized using the StandardScaler() function

X and Y data was split into training and test data using the train_test_slit() function

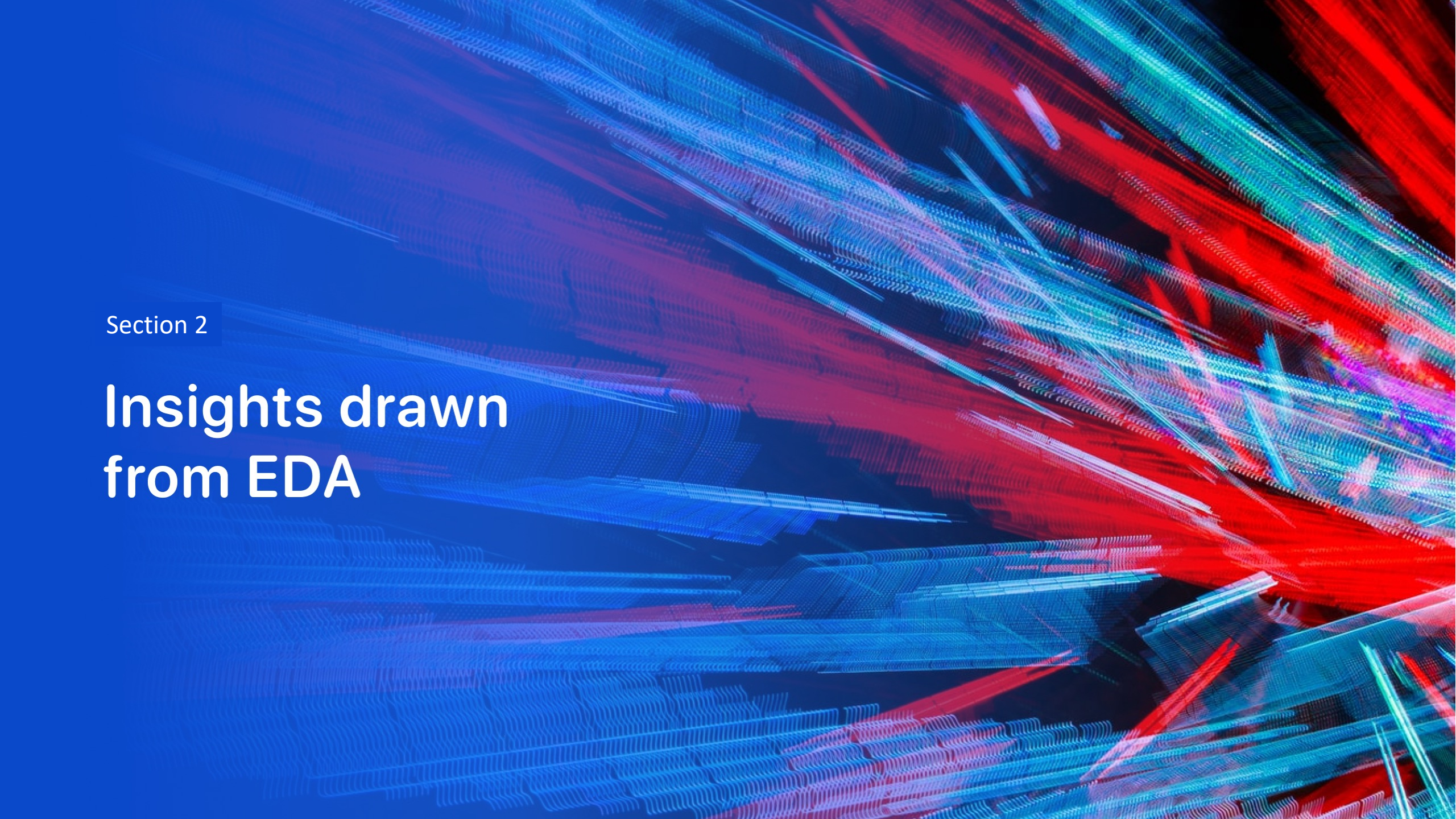
4 classification models were trained using the training data (LogRegression, SVM, Decision Tree and KNN). The best Hyperparameters for each model was found using a Grid Search with 10-fold cross validations

After training and determining the best parameters, the accuracy score of each model was determined by using the test data as well as plotting a confusion matrix for each model. These metrics were compared between each model

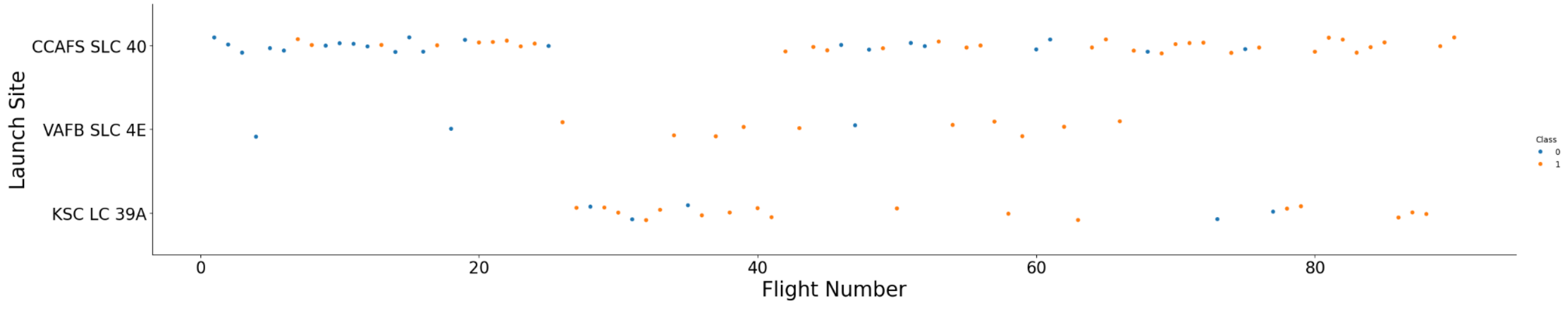
URL to Github: https://github.com/ASVirk-SBU/ArjunVirk_DataScience_Capstone/blob/main/MachineLearningPredictions.ipynb
File name: MachineLearningPredictions.ipynb

Results

- Exploratory data analysis results (Plots in Matplotlib and SQL Queries)
- Interactive analytics demo in screenshots (Folium Map and Dash Application)
- Predictive analysis results (Bar chart to compare accuracy scores and confusion matrix)

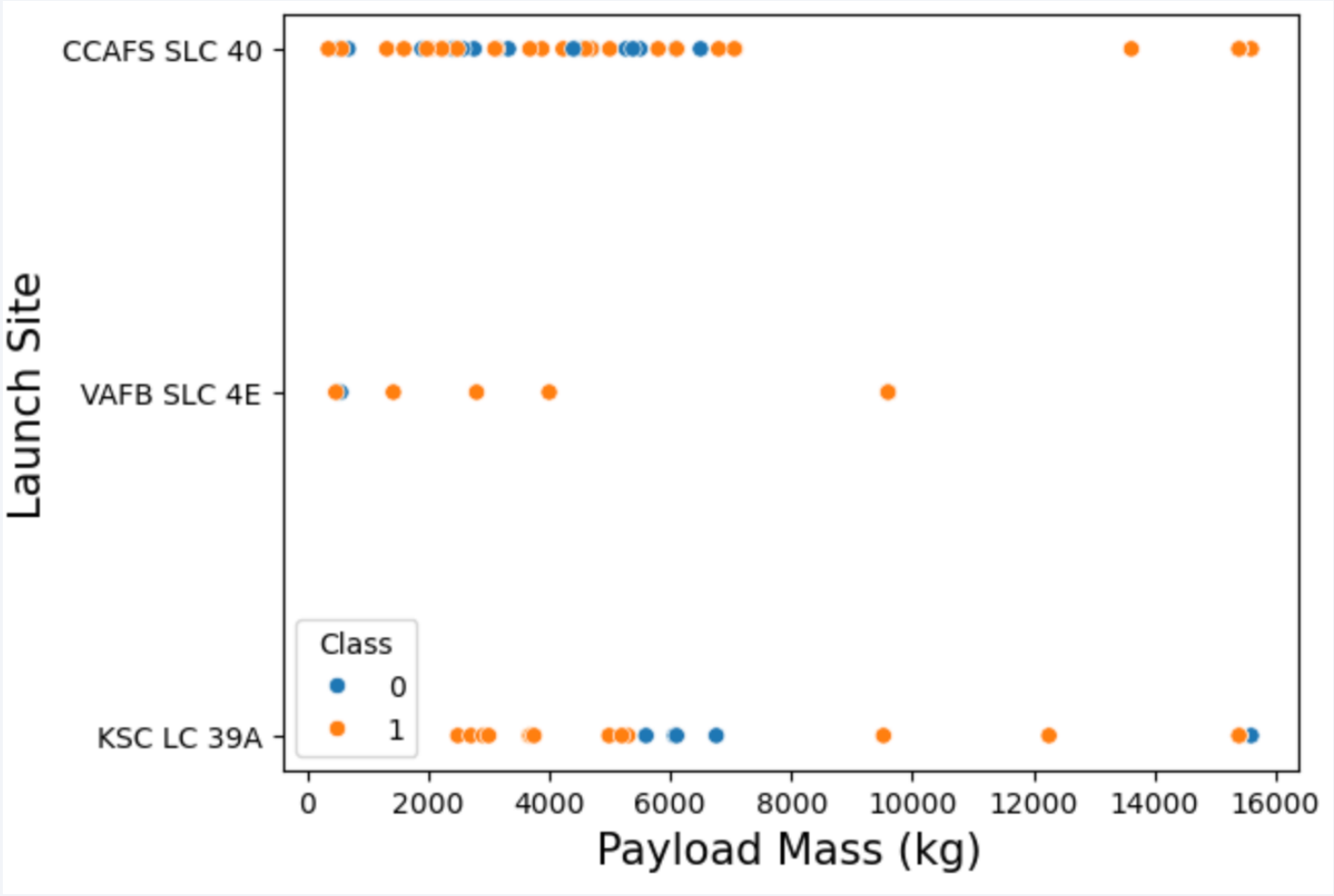

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



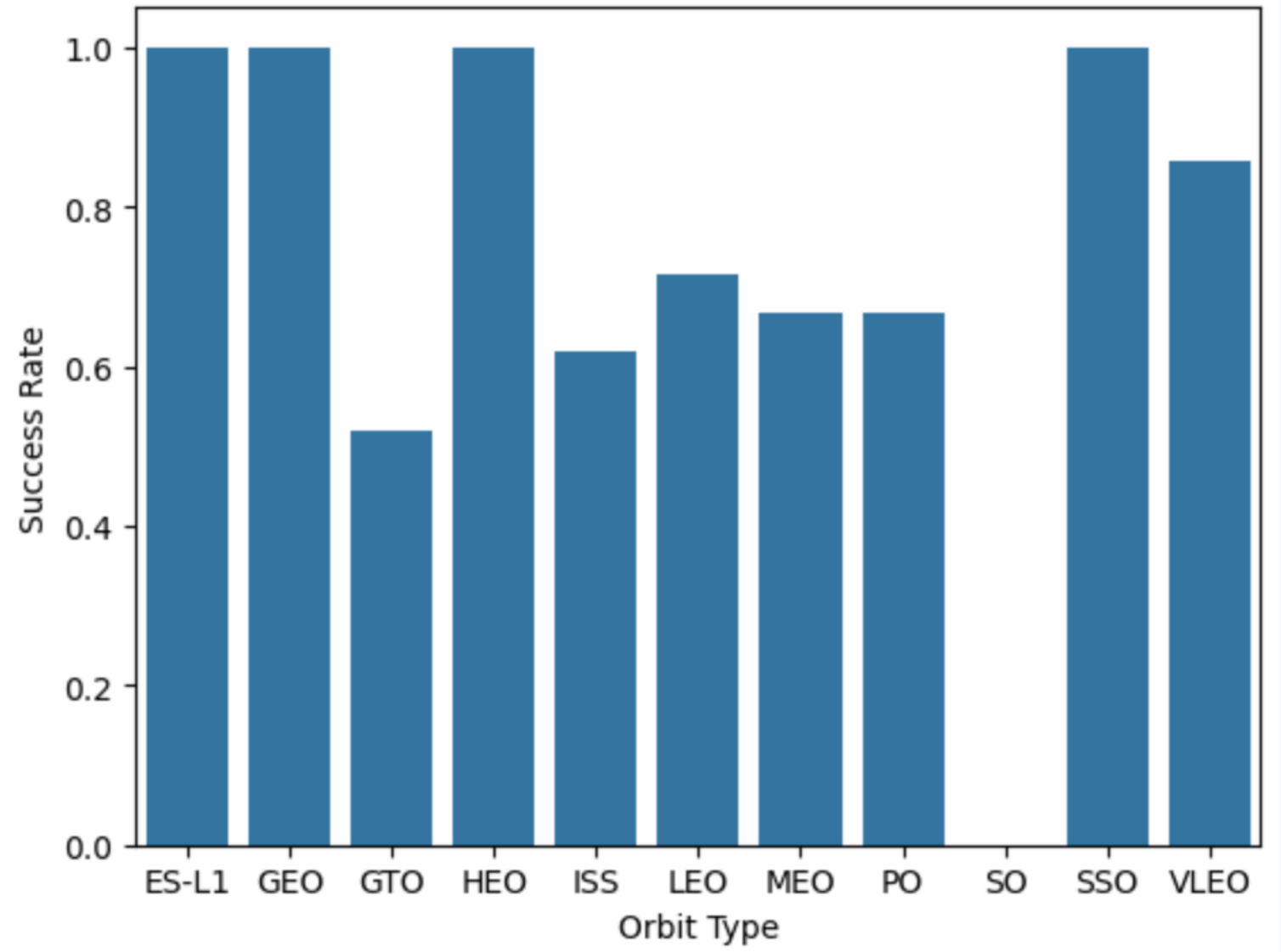
- CCAFS SLC 40 seems that have the most flights here
- Overall, it seems that for higher flight numbers, the success for the flight as the VAFB SLC 4E site have increased
- A clear relationship between flight number and success rate cannot be seen as KSC LC 39A

Payload vs. Launch Site



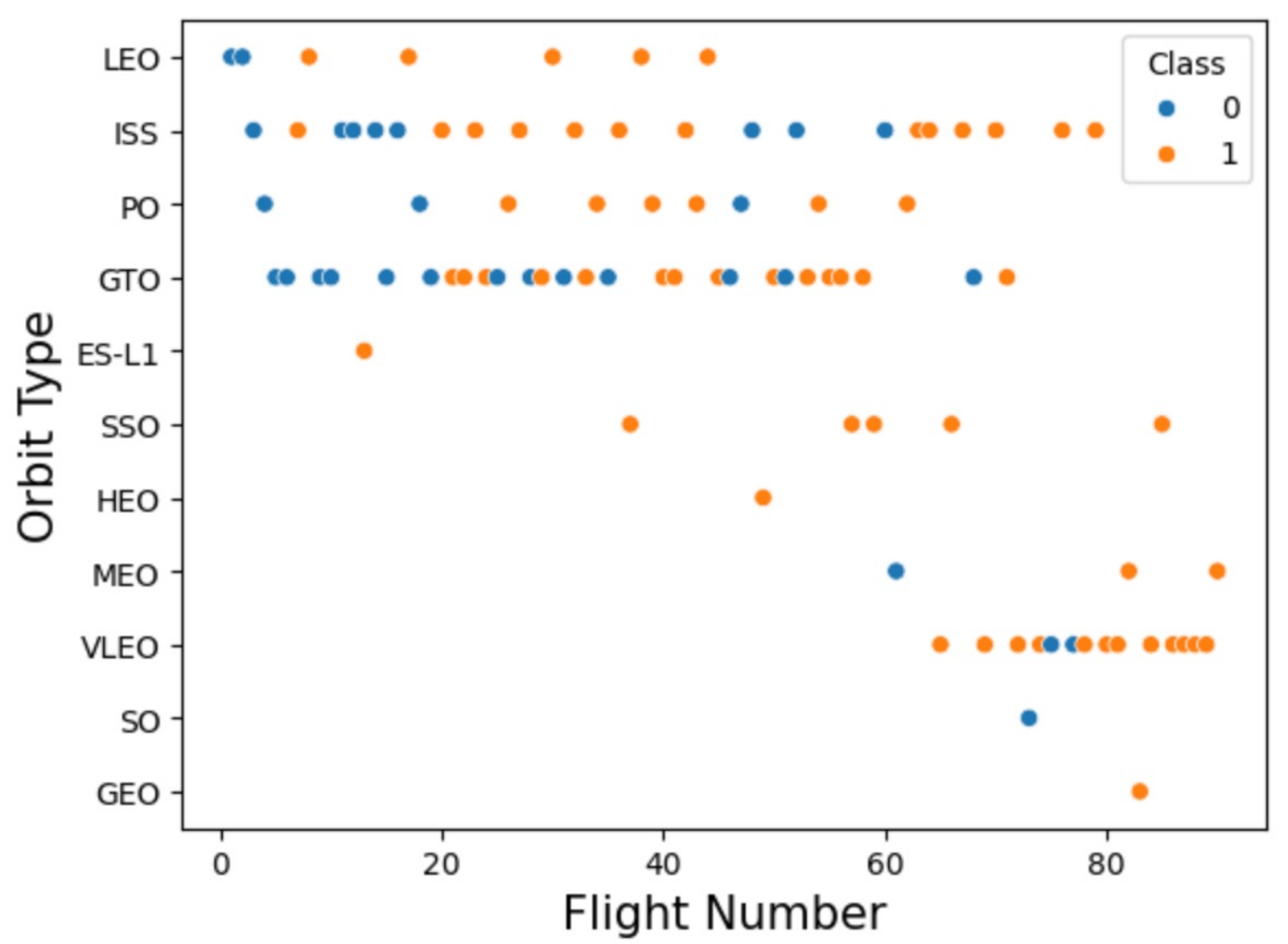
- At CCAFS SLC 40, there does not seem to be a clear relationship between payload mass and success rate, but launches with very high payloads (>10000) appear to be successful
- At VAFB SLC 4E, various payloads seem to provide success launches
- At KSC LC 39A, launches with mass payloads around 6000 kg appear to be unsuccessful

Success Rate vs. Orbit Type



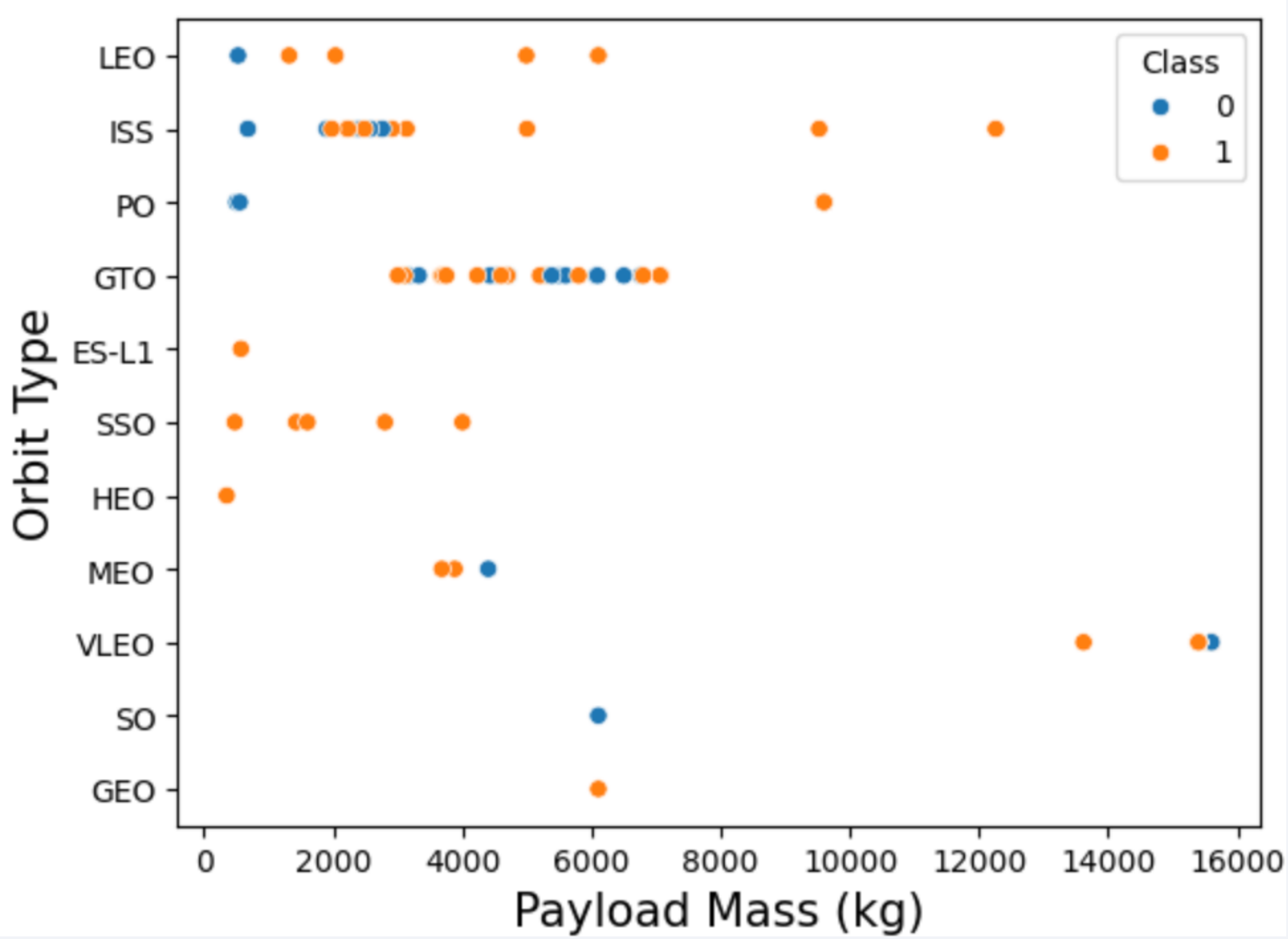
- THE ES-L1, GEO, HEO and SSO orbit types seem to have the highest average success rate
- MEO and PO have similar success rates
- SO orbit types seems to be the least successful (0%)

Flight Number vs. Orbit Type



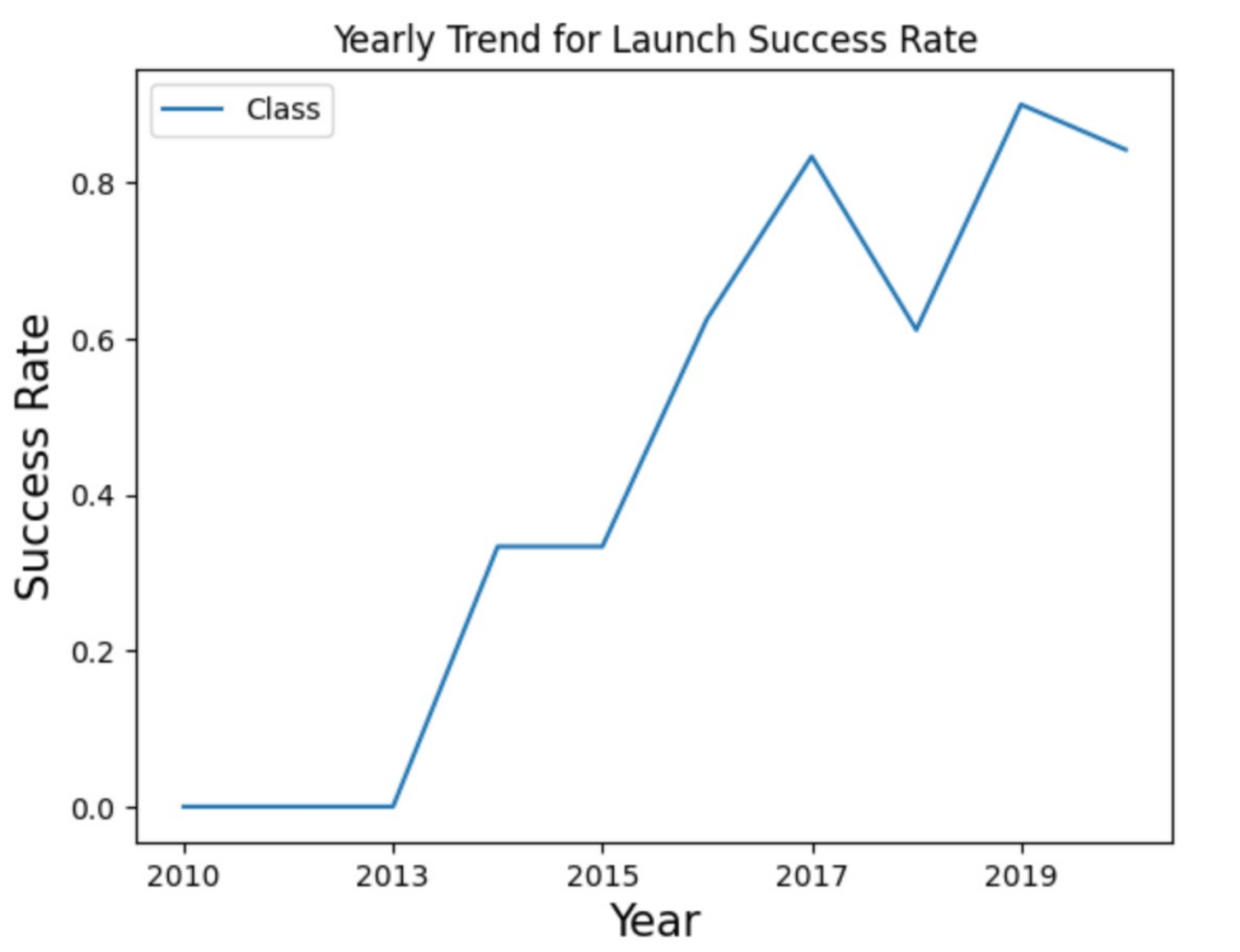
- LEO Flights seem be more successful at higher flight numbers
- All SSO flights have been successful
- GTO flights seem to be mixed between failure and success.
- Flights with VLEO orbit type seem to happen at higher flight numbers

Payload vs. Orbit Type



- For LEO Orbit types, it seems that higher payload masses yield successful launches
- For the GTO orbit type, it is unclear what the impact of payload mass on the success is
- SSO orbit types seems to be successful for relatively low payload masses (less then 4000)

Launch Success Yearly Trend



- As we can see, prior to 2013, there were no successful launches
- After 2013, however, we see a generally linear increase in success rate up until 2017.
- After a drop in success rate in 2018, the average success rate for successful launches continues to increase.

All Launch Site Names

```
%%sql
SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
```

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

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

- Using the DISTINCT clause in SQL, we identify a total of 4 unique launch sites

Launch Site Names Begin with '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_Outcon
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachut
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 (parachut
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attem
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attem
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attem

- Using the LIKE operator to generalize the search for launch site names that begin with “CCA” and using the LIMIT clause to only provide 5 results, we see the following:

Total Payload Mass

```
%%sql

SELECT SUM("PAYLOAD_MASS__KG_") AS "Total Payload Mass (kg)"
FROM SPACEXTABLE WHERE ("CUSTOMER" = "NASA (CRS)")
```

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

Total Payload Mass (kg)
45596

- Using the SUM aggregate function on the Payload mass column, we determine a total payload mass of 45,596 kg

Average Payload Mass by F9 v1.1

```
%%sql

SELECT AVG("PAYLOAD_MASS__KG_") AS "Average Payload Mass for F9 v1.1 (kg)"
FROM SPACEXTABLE WHERE "Booster_Version" LIKE "%F9 v1.1%"
```

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

Average Payload Mass for F9 v1.1 (kg)
2534.6666666666665

- Using the AVG aggregate function on the Payload mass column and limiting the results to only the launches that have the F9 v1.1 Booster version, we determine an payload mass of 2532.6 kg

First Successful Ground Landing Date

```
%%sql

SELECT MIN("Date") AS "Date for 1st succesful landing outcome in ground pad"
FROM SPACEXTABLE WHERE ("Landing_Outcome" = "Success (ground pad)")
```

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

Date for 1st succesful landing outcome in ground pad
2015-12-22

- Using the MIN function on the date column, the date for the 1st successful landing outcome on a ground pass was on December 22nd, 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql

SELECT "Booster_Version" FROM SPACEXTABLE
WHERE ("Landing_Outcome" = "Success (drone ship)")
AND ("PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000)
```

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

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

- By looking at the booster version column and limiting the landing outcomes to successes only and the payload mass to be between 4000 and 6000 kg, we discover 4 possible Booster versions, which are all F9.

Total Number of Successful and Failure Mission Outcomes

```
%%sql

SELECT "Mission_Outcome", COUNT("Mission_Outcome")
FROM SPACEXTABLE GROUP BY "Mission_Outcome"
```

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

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

- There are a total of 101 successful missions, and 1 Failure

Boosters Carried Maximum Payload

```
%%sql

SELECT "Booster_Version" FROM SPACEXTABLE WHERE
"PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_")
                       FROM SPACEXTABLE)
```

```
 * 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

- It seems that 12 F9 Boosters carried the maximum payload, as determined by the subquery used in SQL

2015 Launch Records

```
%%sql

SELECT SUBSTR("Date", 6, 2) AS "Month", "Landing_Outcome",
"Booster_Version", "Launch_Site"  FROM SPACEXTABLE
WHERE ("Landing_Outcome" LIKE "%Failure%") AND (SUBSTR("Date", 0, 5) = '2015')
```

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

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

- There were 2 landing failures in 2015, both from the same launch site and with similar Booster versions

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

```
%%sql

SELECT "Landing_Outcome", COUNT("Landing_Outcome") FROM SPACEXTABLE
WHERE "DATE" BETWEEN ("2010-06-04") AND ("2017-03-20")
GROUP BY "Landing_Outcome" ORDER BY COUNT("Landing_Outcome") DESC
```

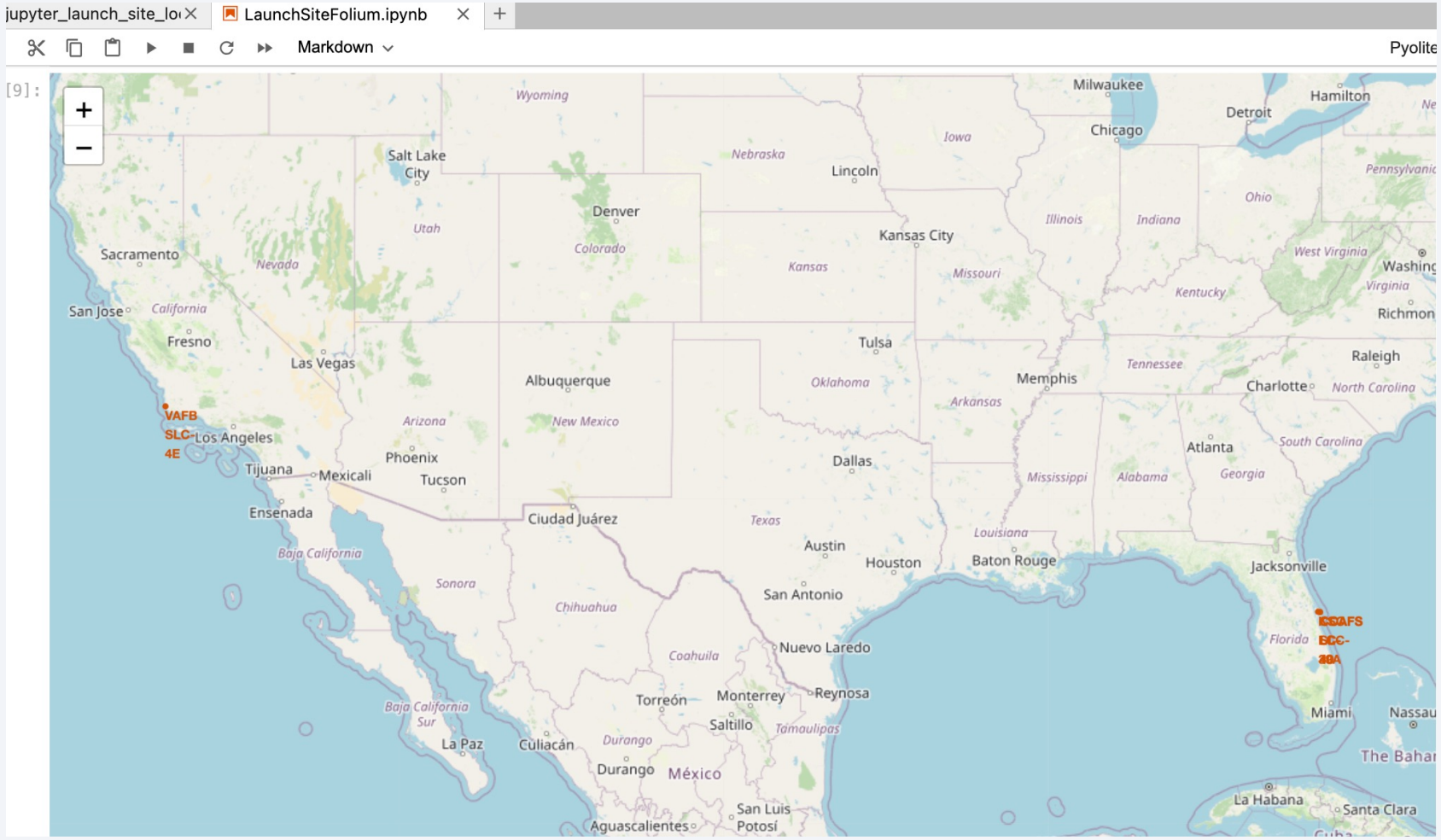
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	COUNT("Landing_Outcome")
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- Between the specified time period in the title of this slide, most of the land outcomes were actually not even attempted. There were 8 total successes, 7 failures, 1 precluded, 3 controlled and 2 uncontrolled

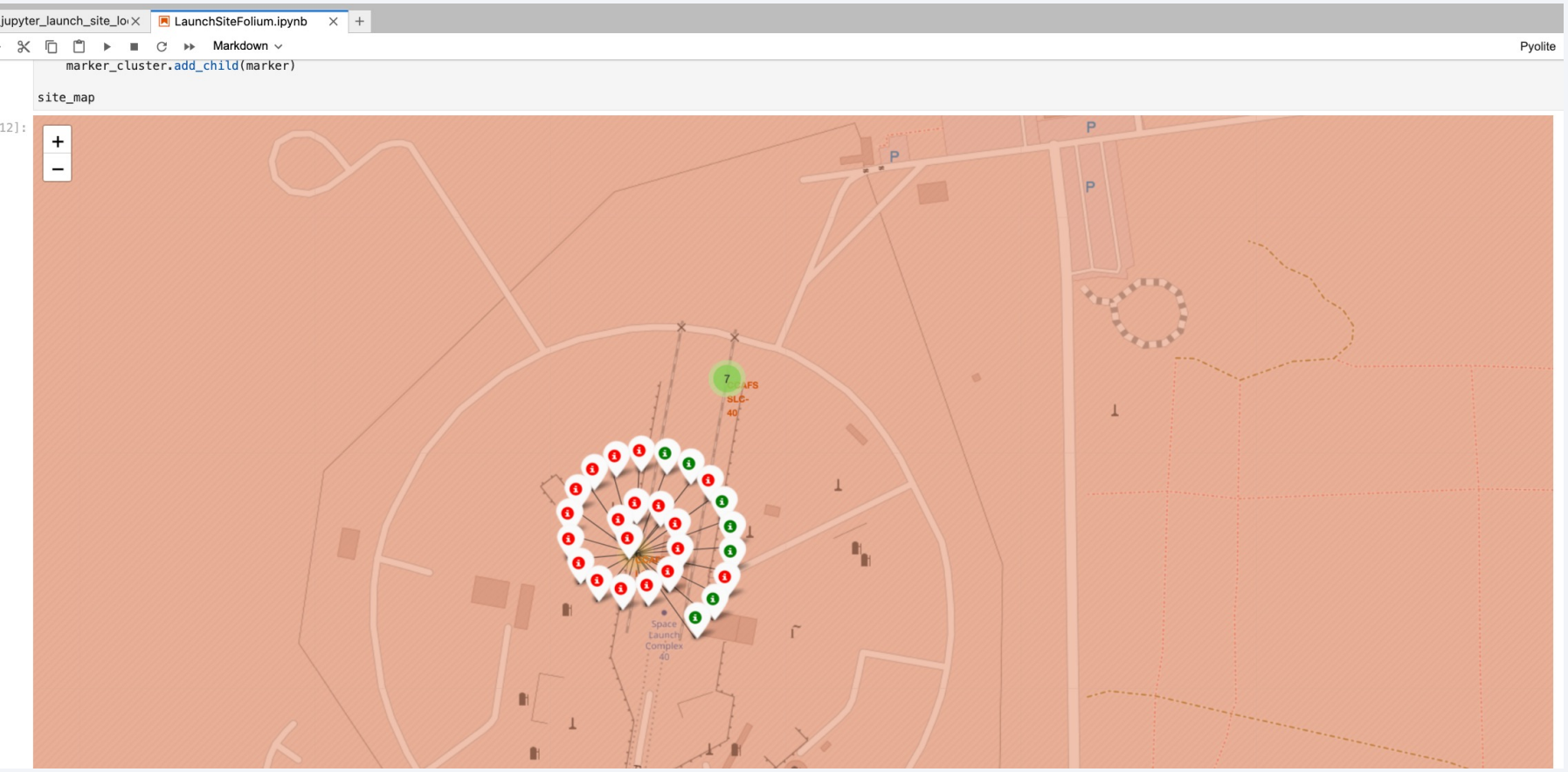

Section 3
Launch Sites
Proximities Analysis

Folium Map with locations of 4 SPACEX Launch Sites



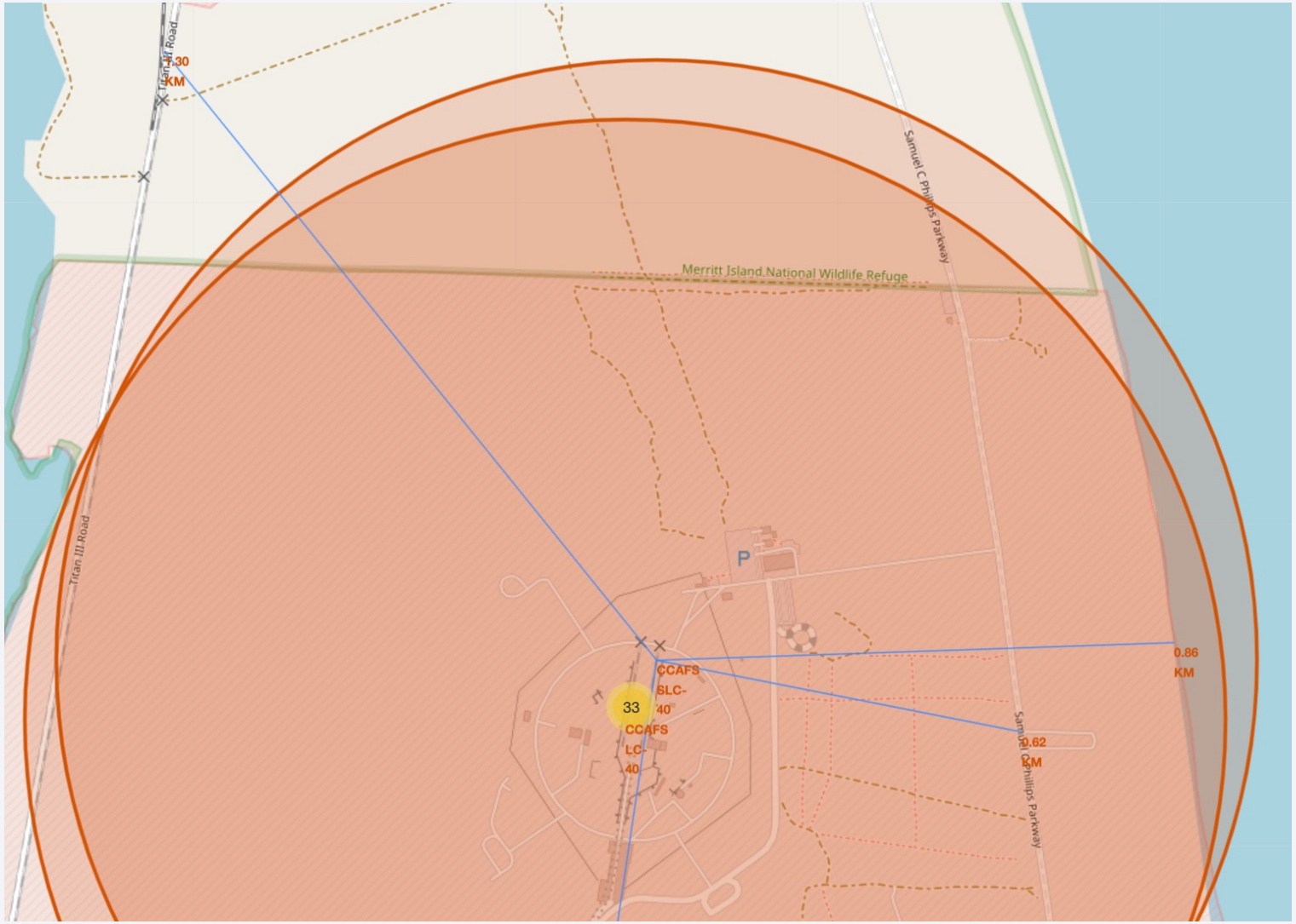
- 3 of the Launch Sites are located in Florida, with 2 of them on the same island
- The other remaining base is located in California

Location of Successful Launches

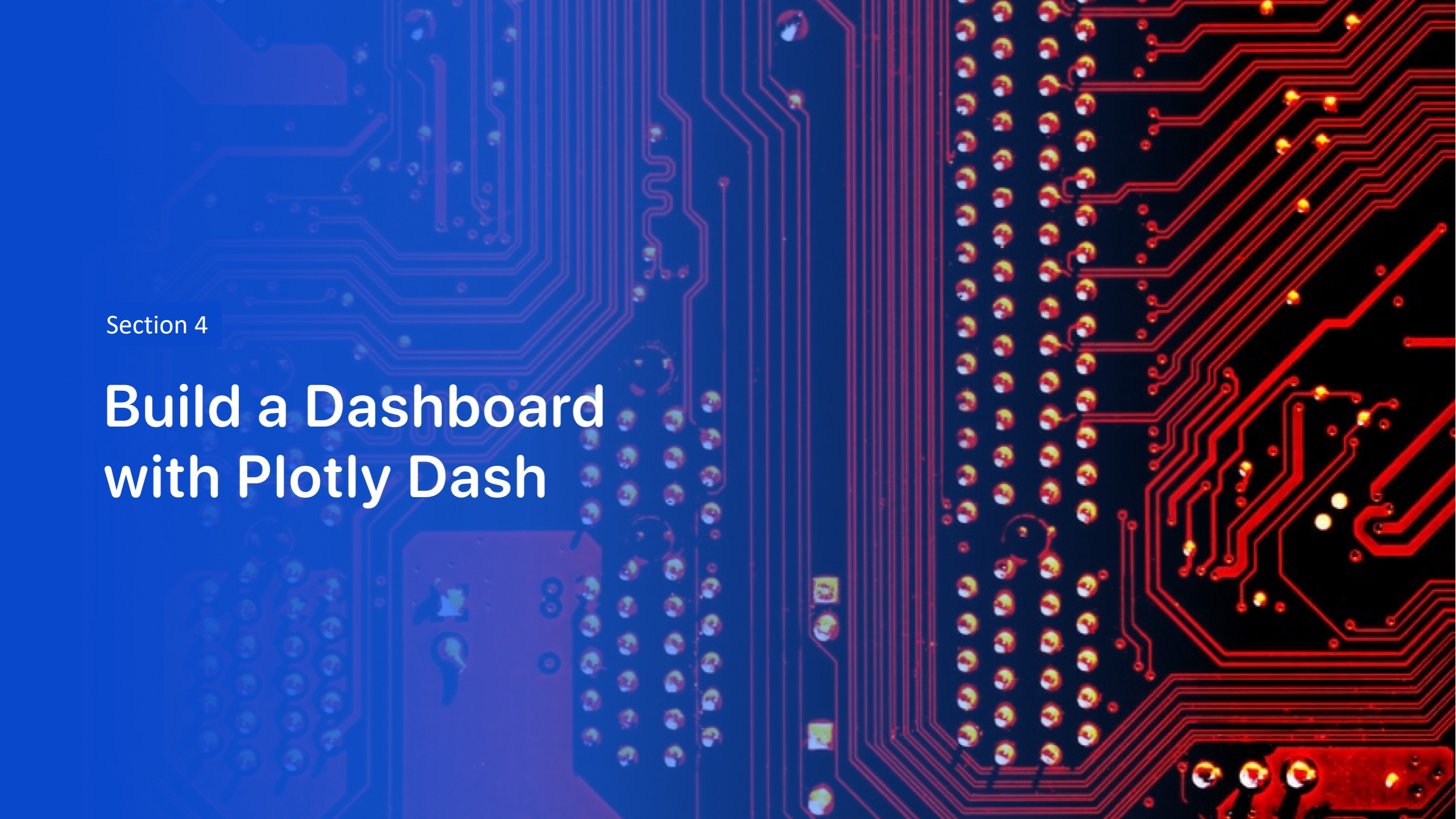


- Adding markers using a marker cluster object, we can distinguish between successful, and failure launches at each launch site, an example of which is shown here at CCAFS LC-40 (which has more failures then successful in almost a radial-like pattern)

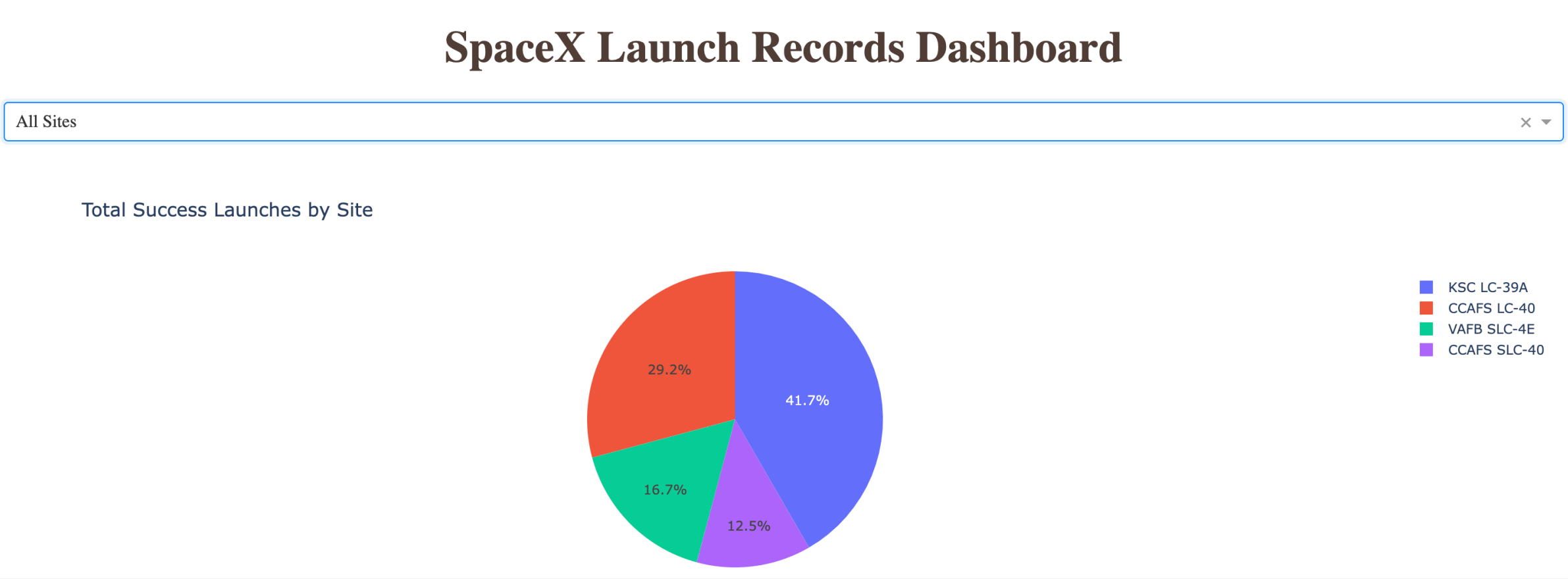
Marking distance between Launch Site and various Landmarkers



- We have identified several major land markers (road, parkway and town), as well as the distance they are from the launch site
- It seem that the parkway and (rail)road are within 2 km from the launch site, for ease of transportation.
- However, towns are located far away from the launch site for safety reasons. The closest town is 18 km away from the launch site

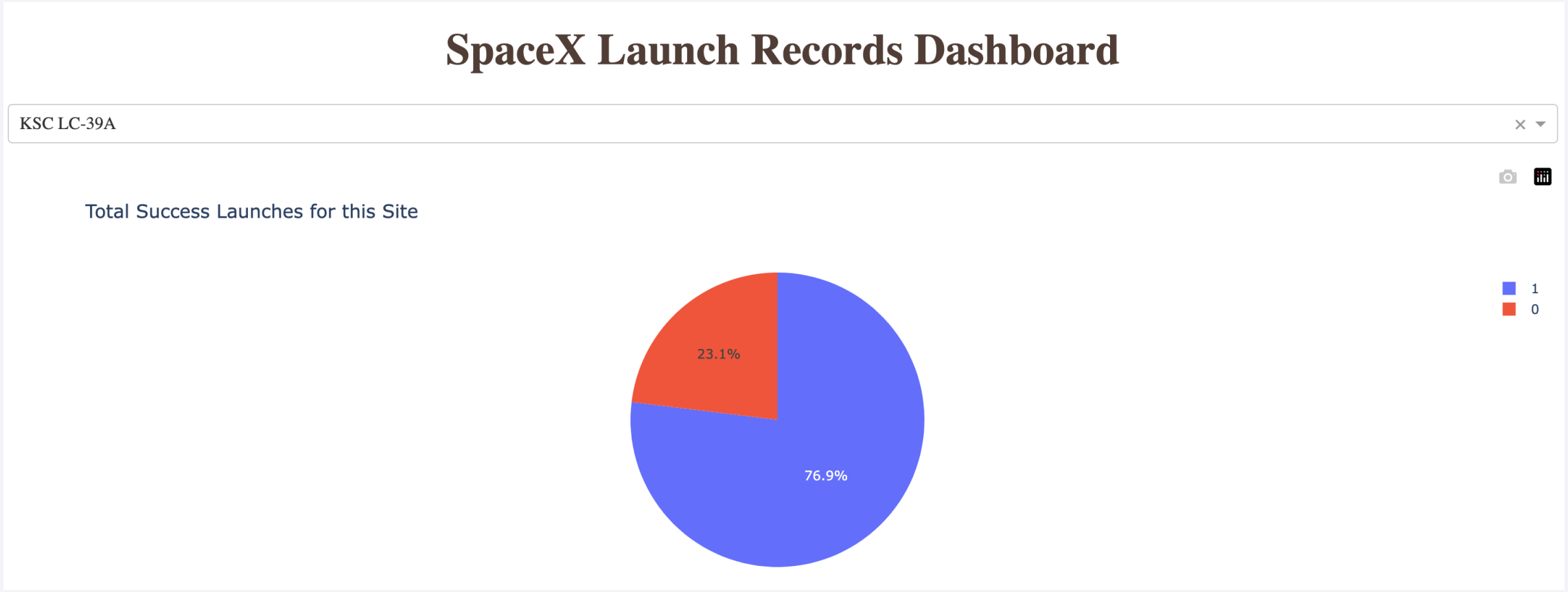

Section 4
Build a Dashboard with Plotly Dash

Breakdown of Successful Launches at each site



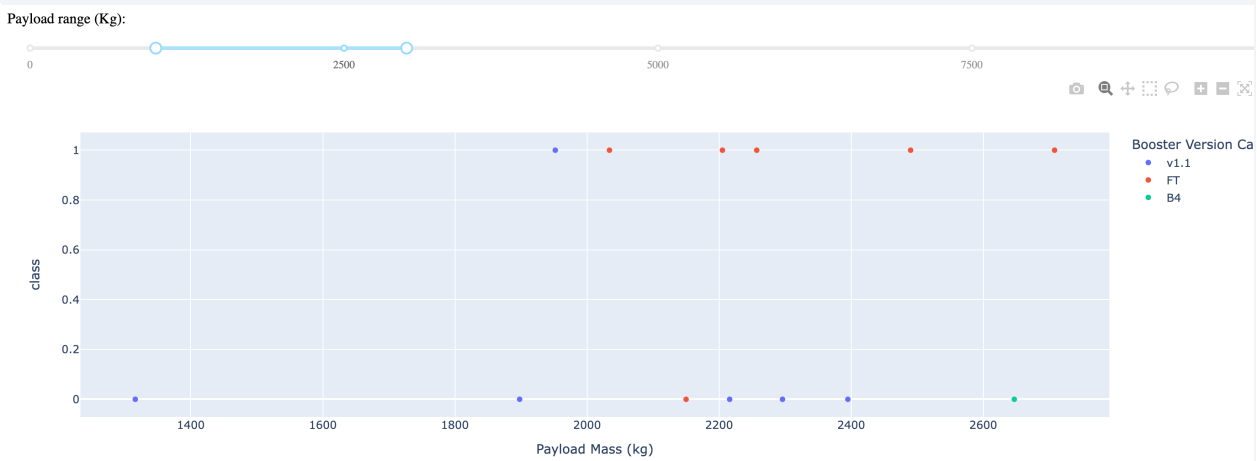
- It seems that KSC LC- 39A has the most successes of the 4 sites, and CCAFS SLC-40 has the least

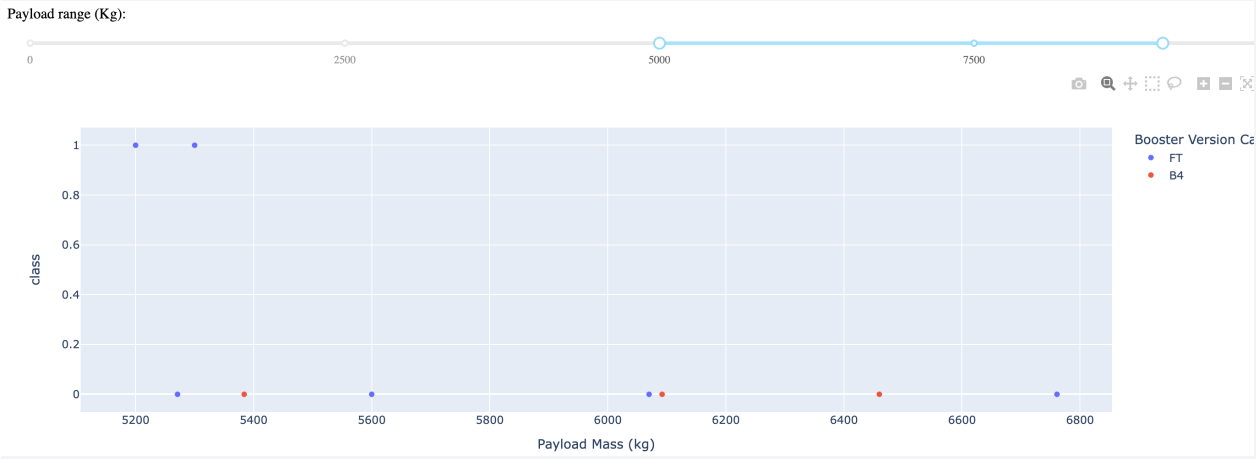
Breakdown of Flight Successes at KSC LC-39A



- Over 75% of the flights at this site were successful

Impact of Payload Mass on Success for different Booster Versions



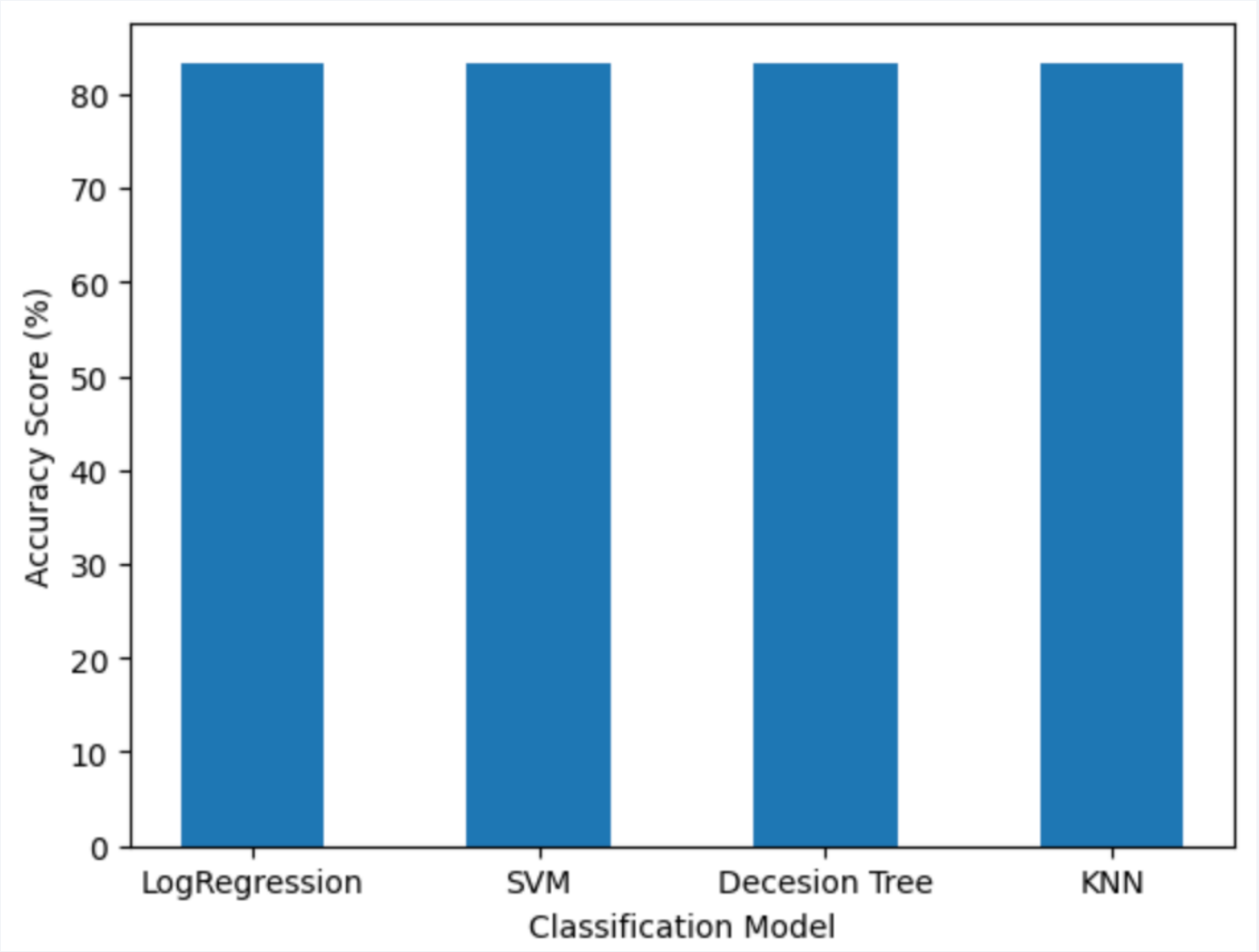


- It seems that the FT Booster version seems to yield the most success launches
- Almost all successful launches seem to occur at Payload masses of less then 5300 kg

Section 5

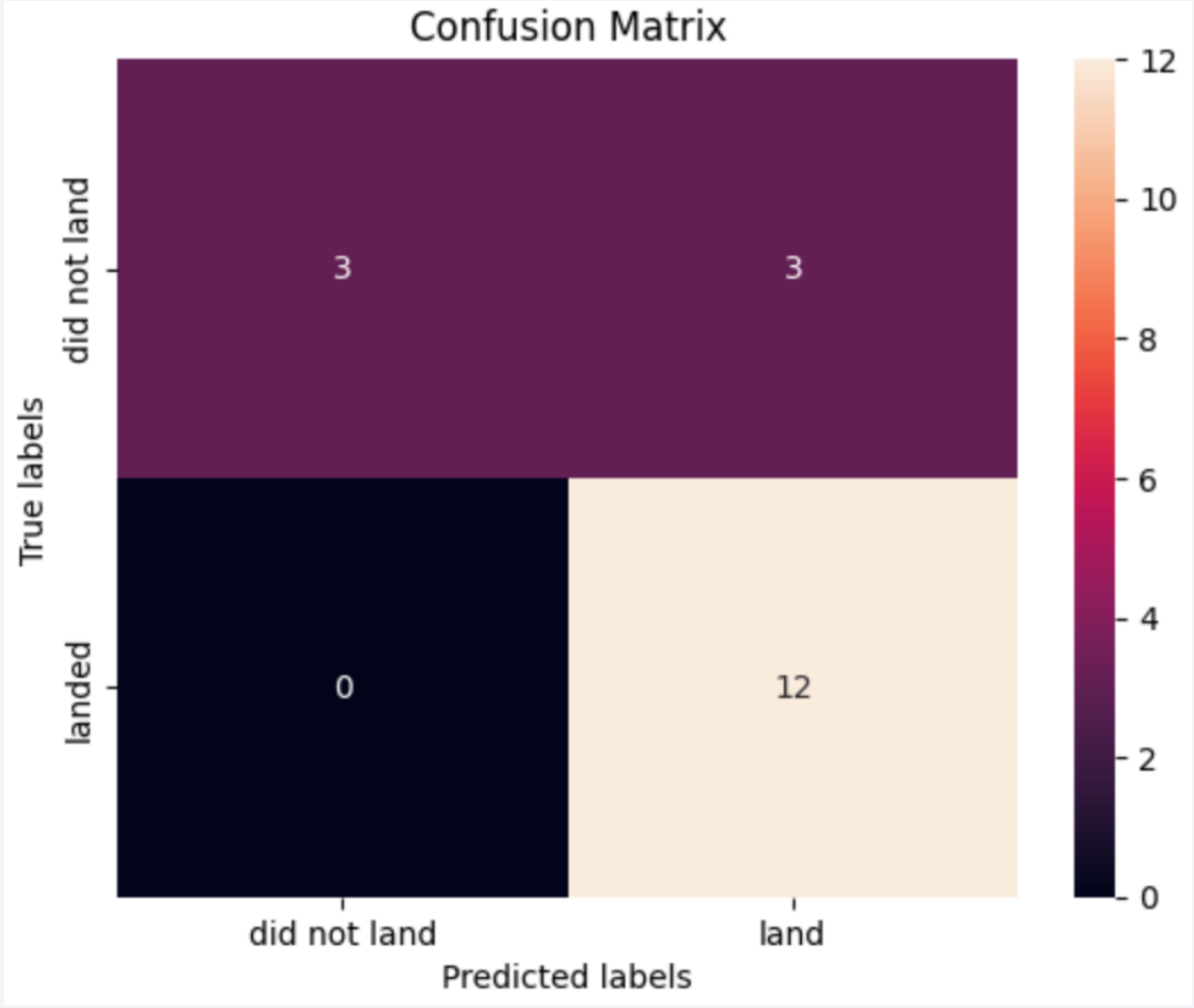
Predictive Analysis (Classification)

Classification Accuracy



- All 4 models were trained with the same data and used the same number of folds in a grid-search cross validations
- All 4 models yield the same accuracy score on the test data of 83%

Confusion Matrix



- Given that all 4 models have similar accuracy measures, they all had the same confusion matrix.
- There were 12 True Positives, 3 True Negatives, 3 False Positives, and 0 False Negatives

Conclusions

- Since 2013, the rate for success launches have generally increased
- The success rate of a launch does seem to be related to the orbit type.
- Using a Folium map, we determine that launch site are generally located close to the coastline (water) and roads and parkways (less then 2 km), likely for ease of transportation. However, launch sites are far away from any major towns/cities for the safety of the general population
- The dash application show that KSC LC 39A has the overall highest success rate of the 4 launch sites. Also, successful launches are typically associated with payload masses of less then 5000 kg.
- All 4 classification models (Logistic Regression, SVM, Decision Trees and KNN) as perform with the same amount of accuracy (83.33%) on the test data.

Appendix

- All of the relevant codes used to obtain the results shown in the presentation can be found at the github URL given throughout the methodology section of the slides.
- The original template for those codes were provided by **Coursera** as part of the IBM Data Science Professional Certificate Program, and the templates were adjusted by the author of this presentation (**Arjun S. Virk**) to satisfy the requirements and answer the questions provided by the program.
- Some snippets of the completed codes are also shown here:

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`:

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []

for i in list(df['Outcome']):
    if i in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

```
[0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1
1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

TASK 6

Create a support vector machine object then create a `GridSearchCV` object `svm_cv` with cv - 10. Fit the object to find the best parameters from the dictionary `parameters`.

```
[17]: parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
                    'C': np.logspace(-3, 3, 5),
                    'gamma':np.logspace(-3, 3, 5)}
      svm = SVC()

[18]: svm_cv = GridSearchCV(svm, parameters, cv = 10)
      svm_cv.fit(X_train, Y_train)

[18]: ▸ GridSearchCV
      ▸ estimator: SVC
          ▸ SVC

[19]: print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
      print("accuracy :",svm_cv.best_score_)

      tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
      accuracy : 0.8482142857142856
```

```
#-----------------------------------------------------------------------------------------------------------------------
parkway_lat = 28.56212
parkway_long = -80.5706
distance_parkway = calculate_distance(launch_lat, launch_long, parkway_lat, parkway_long)

parkway_marker = folium.Marker([parkway_lat,parkway_long], icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0),
        html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % "{:10.2f} KM".format(distance_parkway),))

site_map.add_child(parkway_marker)

parkway_coordinates = [[launch_lat, launch_long],[parkway_lat, parkway_long]]
parkway_lines=folium.PolyLine(locations=parkway_coordinates, weight=1)
site_map.add_child(parkway_lines)
#-----------------------------------------------------------------------------------------------------------------------
road_lat = 28.57231
road_long = -80.58522
distance_road = calculate_distance(launch_lat, launch_long, road_lat, road_long)

road_marker = folium.Marker([road_lat,road_long], icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0),
        html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % "{:10.2f} KM".format(distance_road),))

site_map.add_child(road_marker)

road_coordinates = [[launch_lat, launch_long],[road_lat, road_long]]
road_lines=folium.PolyLine(locations=road_coordinates, weight=1)
site_map.add_child(road_lines)
#-----------------------------------------------------------------------------------------------------------------------
cape_lat = 28.40076
cape_long = -80.60411
distance_cape = calculate_distance(launch_lat, launch_long, cape_lat, cape_long)

cape_marker = folium.Marker([cape_lat,cape_long], icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0),
        html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % "{:10.2f} KM".format(distance_cape),))

site_map.add_child(cape_marker)

cape_coordinates = [[launch_lat, launch_long],[cape_lat, cape_long]]
cape_lines=folium.PolyLine(locations=cape_coordinates, weight=1)
site_map.add_child(cape_lines)
#-----------------------------------------------------------------------------------------------------------------------
```

Thank you!