IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mitchell Lee
9/10/2024

Outline

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

Executive Summary

- Methodology:
  - In this project we gathered data from the SPACEX launch data API
  - We also used web scraping from a related Wiki page using beautiful soup to gather more launch data
  - We performed exploratory analysis using SQL and Python
  - Used plotly and folium to generate visualizations and dashboards to assist
  - Performed predictive analysis using ML algorithms
- Summary of all results
  - We wanted to create a machine learning model to accurately predict whether or not the Falcon 9 Spaceship would successfully land

Introduction

- SpaceX advertises the Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers can cause upward of 165 million dollars each.
- Most of this cost is saved because SpaceX can reuse the first stage of the rocket

Problems we are looking to find answers to:

- What factors are related to a successful launch
- Does the type of orbit affect launch
- Does the size of the payload affect the outcome of the rocket launch
- Does the location and its proximity to certain geographical locations make a difference

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The Space X REST API was used to collect data
- Perform data wrangling
  - Python BeautifulSoup package was used to perform data wrangling on the relevant Space X launch data wiki
- 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

- Datasets were collected using a static version of launch data from the Space X REST API
- Response was sent as a JSON file which was decoded to a pandas dataframe.
- Once the response was received we created a new dataframe to model specific features we were interested in and used this information for a new dataset filtered for only Falcon 9 launches.

```
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

Note: The Mean of the dataset's payload mass was used to estimate null values

Data Collection – SpaceX API

- SpaceX REST calls flowchart as pictured in figure 1
- Source code can be found at:

https://github.com/upurpie/SpaceXCourseraFinalProject/blob/main/1%20-%20jupyter-labs-spacex-data-collection-api.ipynb

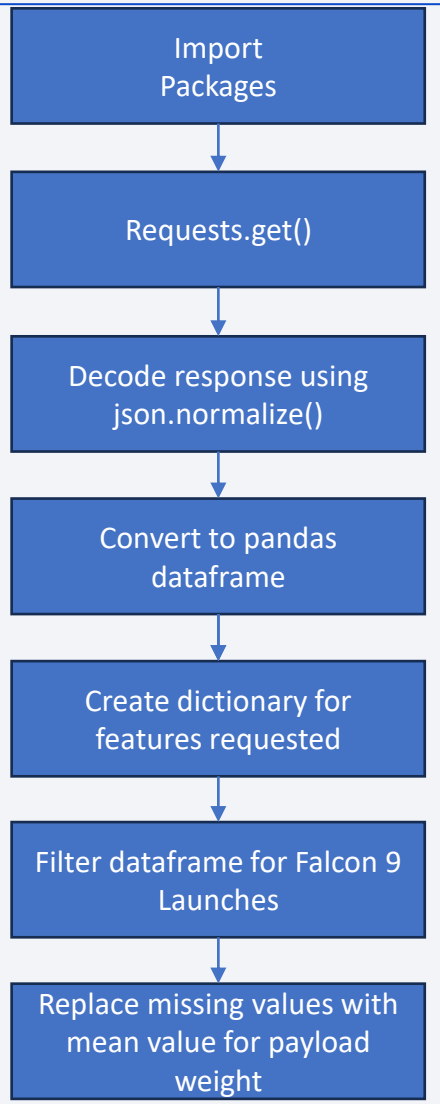


Figure 1. Flow chart for SpaceX API process

Data Collection - Scraping

- Web Scraping flowchart as pictured in figure 2
- Source code can be found at:

https://github.com/upurpie/SpaceXCourseraFinalProject/blob/main/2-%20jupyter-labs-webscraping.ipynb

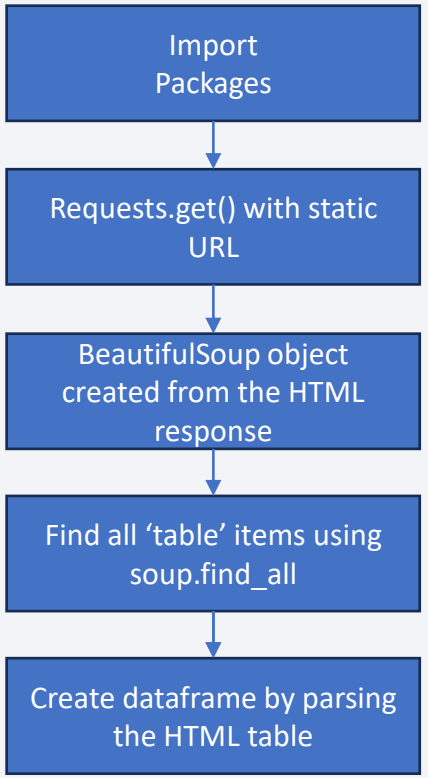


Figure 2. Flowchart for webscraping

Data Wrangling

- Once the datasets were created and processed we used data wrangling to convert the data to a usable output by:
  - Analyzing the data types
  - Gathering value counts on outcomes and orbit types
  - Assigning a Class to each row depending on outcome
- https://github.com/upurpie/SpaceXCourseraFinalProject/blob/main/3%20-%20labs-jupyter-spacex-Data%20wrangling.ipynb

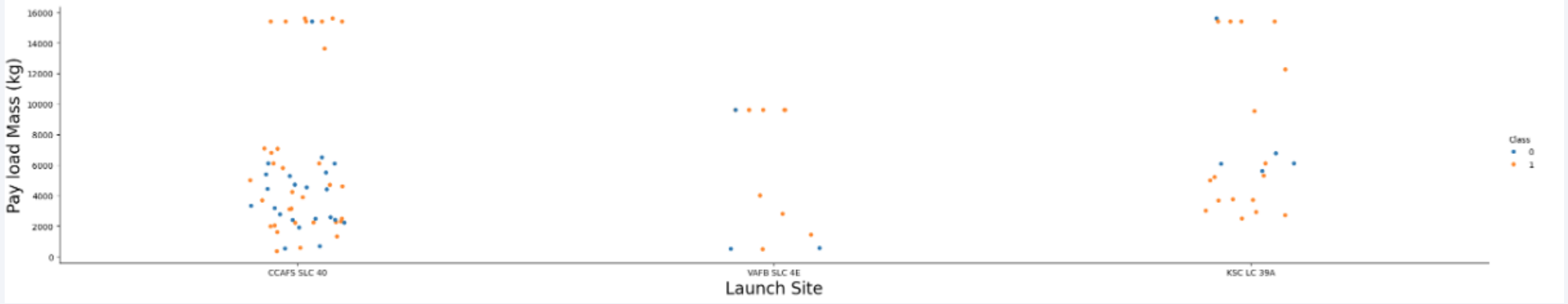
EDA with Data Visualization

- Using Matplotlib, we created visualizations for exploratory data analysis
- https://github.com/upurpie/SpaceXCourseraFinalProject/blob/main/5%20-%20data%20visualization.ipynb

EDA with Data Visualization

- Visualized the relationship between Payload Mass and Launch Site to see the relationship. No rockets were launched for heavy payloads in certain sites

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue
sns.catplot(y="PayloadMass", x="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Launch Site",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```

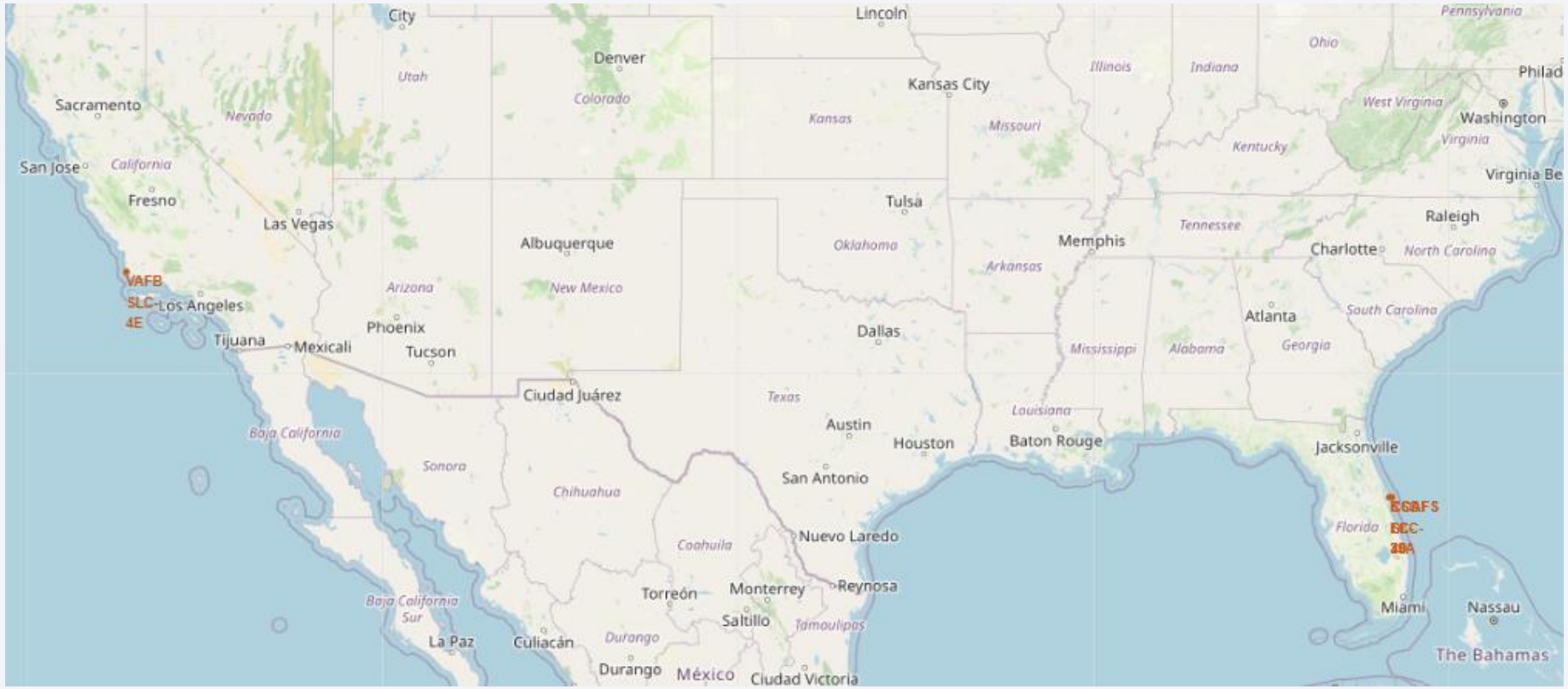


EDA with SQL

- SQL queries performed for EDA
- https://github.com/upurpie/SpaceXCourseraFinalProject/blob/main/4%20-%20jupyter-labs-eda-sql-coursera_sqllite.ipynb
  - Names of the unique launch sites in the space mission
  - 5 Records where launch sites begin with the string CCA
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - Date when first successful landing outcome in ground pad was achieved
  - Names of the boosters which have success in drone ship and have payload mass between 4000 and 6000
  - Total number of successful and failure missions
  - Names of the booster versions which have carried the maximum payload mass
  - Month names and failures in 2015
  - Rank of count of landing outcomes between 2010 and 2017 in descending order

Build an Interactive Map with Folium

- Created a folium map plotting all launch stations. Showing they are all in proximity to the coast



Build an Interactive Map with Folium

- Added markers to the map to show number of flights from each location

Build an Interactive Map with Folium

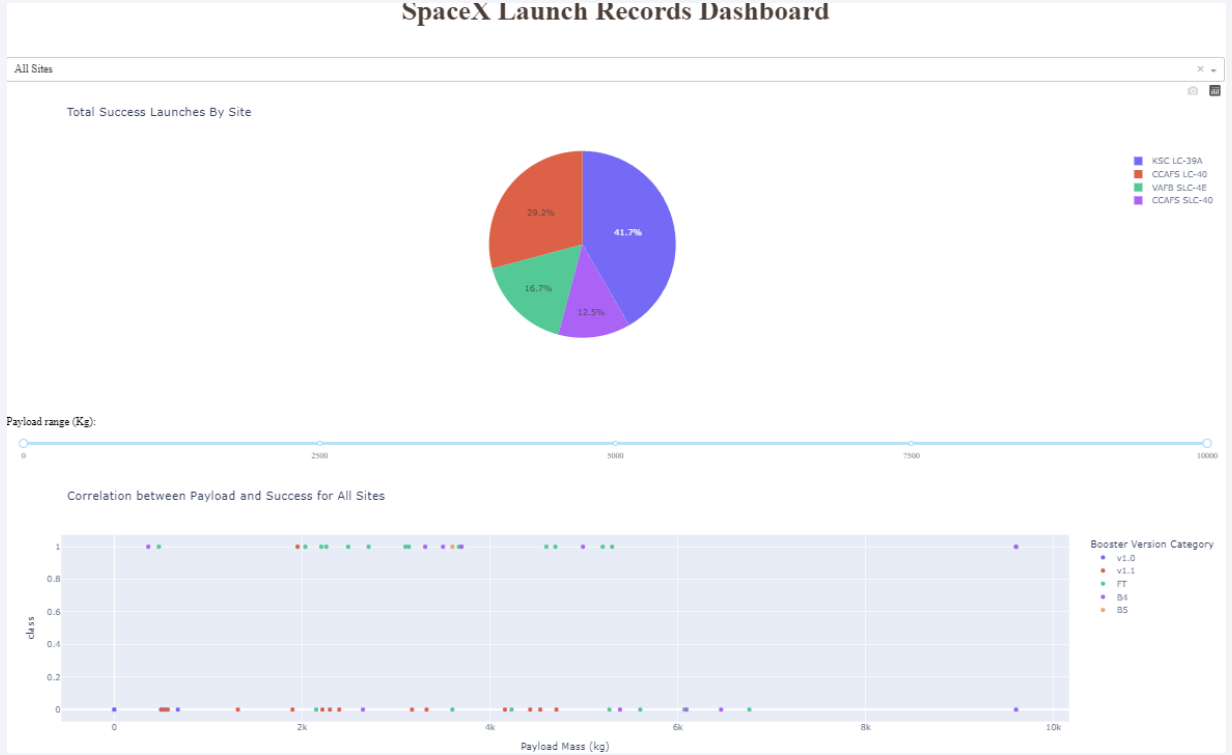
- Added labels to the map to show number of successful and failed flights in a site

Build an Interactive Map with Folium

- Used a polyline to display distance from the site and the coast and other landmarks

Build a Dashboard with Plotly Dash

- Created a dashboard on Dash using Plotly to visualize data from different sites with different payload ranges



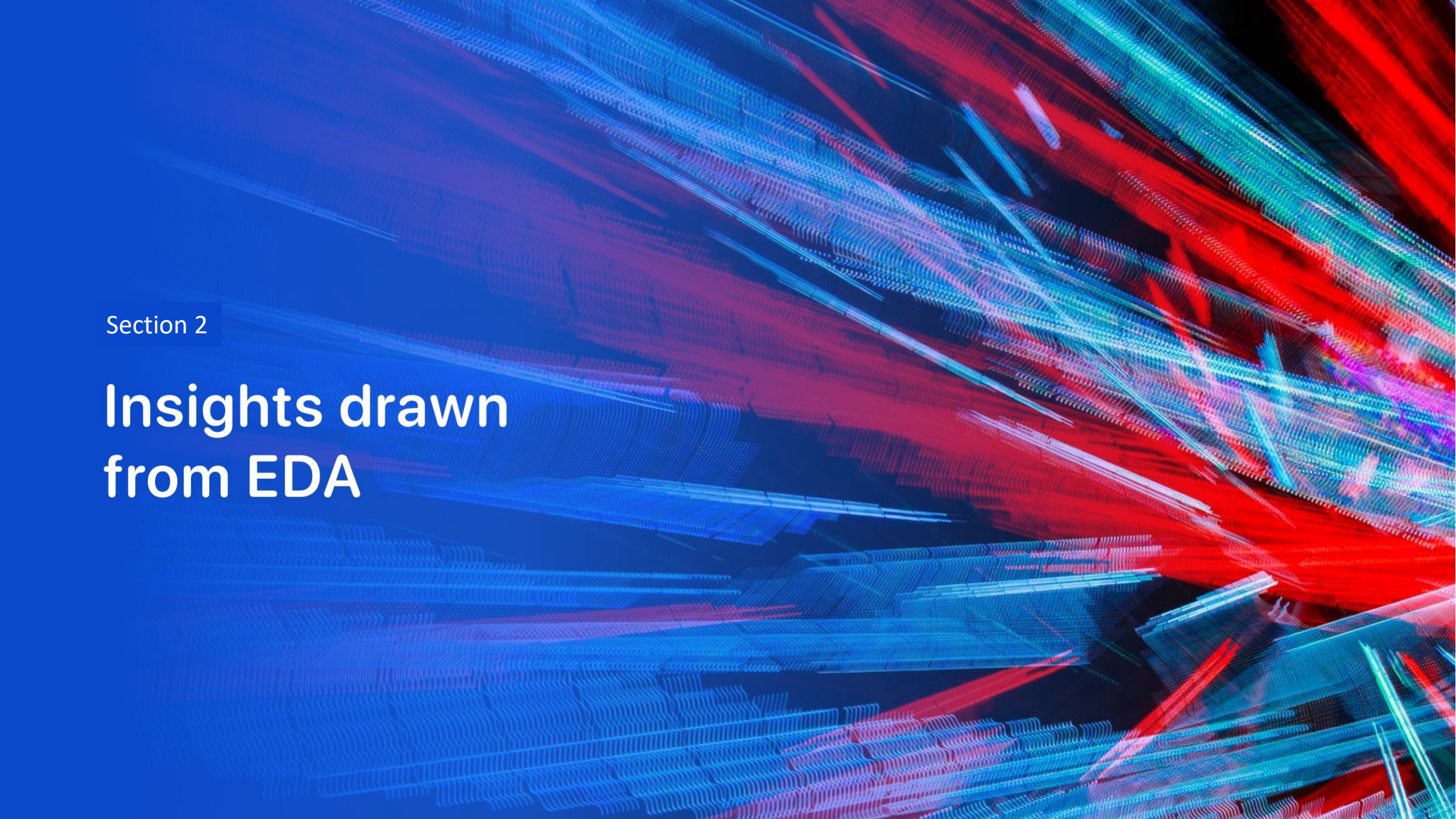
- https://github.com/upurpie/SpaceXCourseraFinalProject/blob/main/7%20-%20spacex_dash_app.py

Predictive Analysis (Classification)

- Using sklearn we split our data into training and test data
- We created a model to classify whether or not a flight would be successful using the following machine learning algorithms:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - K Nearest Neighbours
- https://github.com/upurpie/SpaceXCourseraFinalProject/blob/main/8%20-%20SpaceX_Machine%20Learning%20Prediction.ipynb

Results

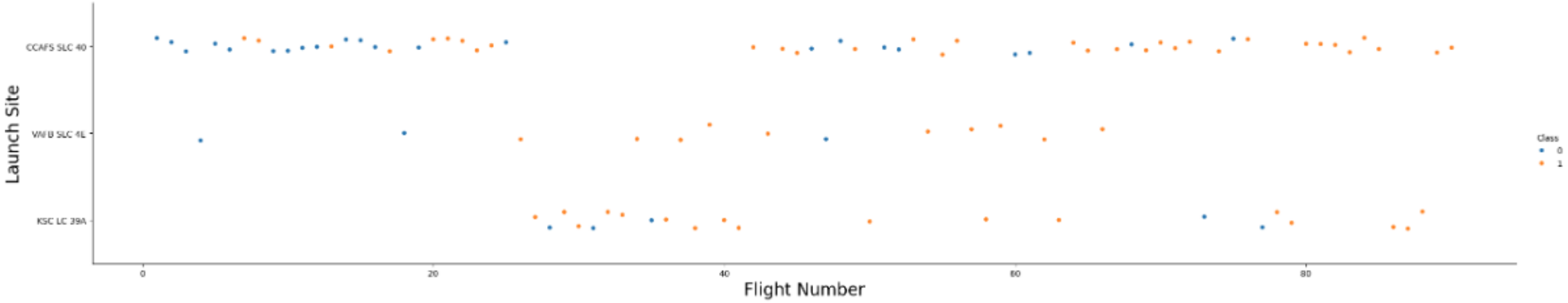
- Exploratory data analysis results
  - With the above steps, we were able to infer that the success of a mission was dependent on the payload weight and the type of orbit being performed
- Interactive analytics demo in scr
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs Launch Site

- Created a scatter point chart to show the relationship between the launch site and the flight number. Showing that we increased flight number we had more success. There were also more launches from site CCAFS SLC 40 than others

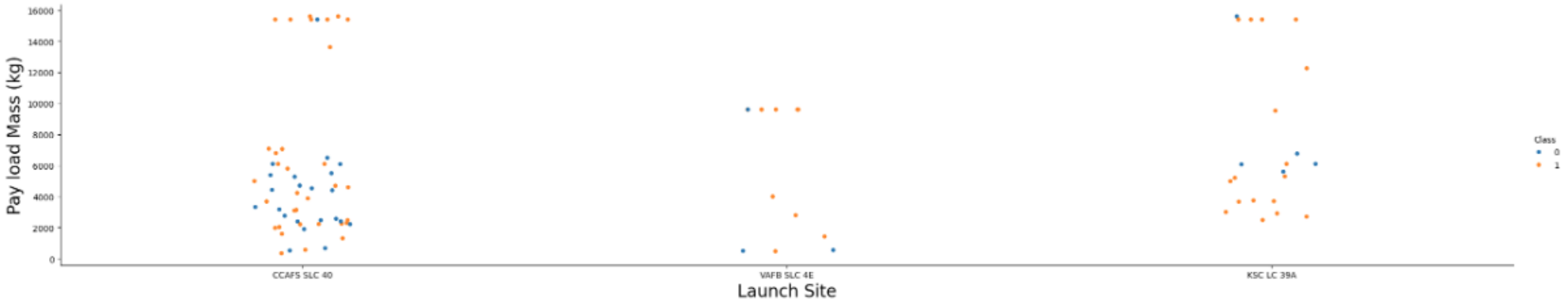
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Payload vs Launch Site

- Visualized the relationship between Payload Mass and Launch Site to see the relationship. No rockets were launched for heavy payloads in certain sites

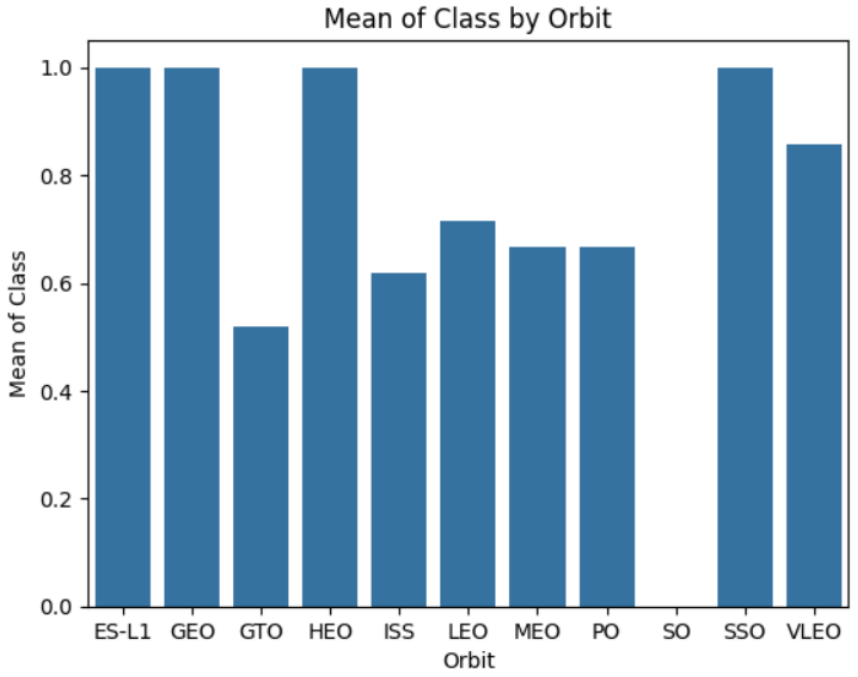
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue
sns.catplot(y="PayloadMass", x="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Launch Site",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```



Success Rate vs Orbit Type

- Plotted the mean of each class (success) based on the orbit.
- Showing the best performing:
  - ES-L1
  - GEO
  - HEO
  - SSO
- Worst performing orbit types:
  - SO

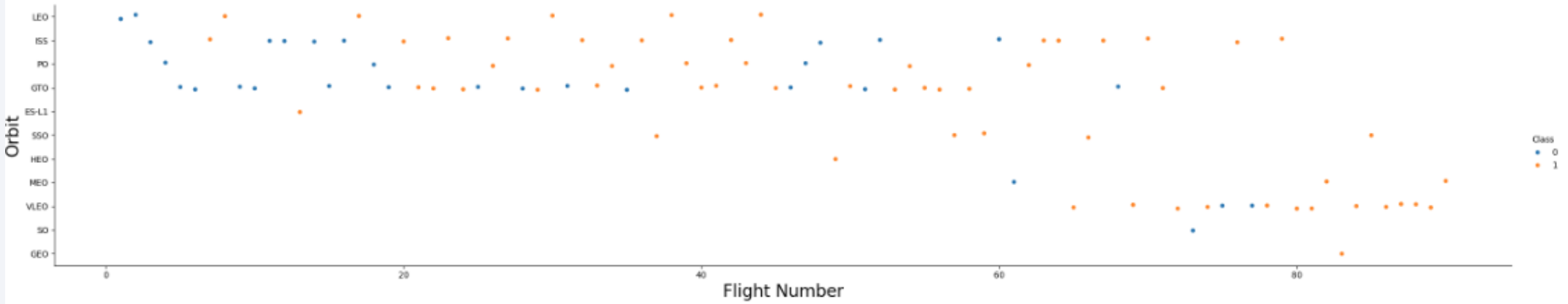
```
# HINT use groupby method on Orbit column and get the mean of Class column
orbit_mean = df.groupby('Orbit')['Class'].mean().reset_index()
sns.barplot(x='Orbit',y='Class', data=orbit_mean)
plt.xlabel('Orbit')
plt.ylabel('Mean of Class')
plt.title('Mean of Class by Orbit')
plt.show()
```



Flight Number vs. Orbit Type

- Visualized relationship between flight number and orbit type. Showing that for some orbit types, the success rate depends on the number of flights performed

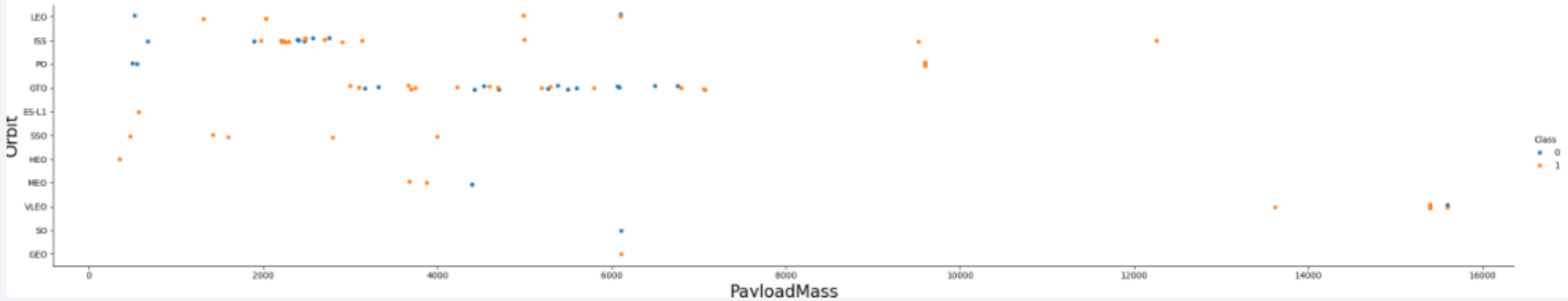
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(x='FlightNumber',y='Orbit',hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Payload vs. Orbit Type

- Plotted the payload mass vs Orbit on a scatterpoint chart to show that some types of orbits have more successful positive landing rates with heavier payloads. However, it is difficult to infer for some

```
# Plot a scatter point chart with x axis to be Payload Mass 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 = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Launch Success Yearly Trend

- Plotted a line chart to see the success rate per year showing in general an increase in launch success

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
sns.lineplot(x='Date',y='Class', data=df)
plt.xlabel('Year', fontsize=20)
plt.ylabel('Average Success Rate', fontsize=20)
plt.title('Launch Success Yearly Trend')
plt.show()
```

All Launch Site Names

- A SQL query was performed to identify the unique launch sites as listed below. It shows us our 4 sites

```
%sql select "Launch_Site" from SPACEXTABLE group by "Launch_Site"
```

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

Launch Site Names Begin with 'CCA'

- We queried to find 5 records where launch sites begin with `CCA which gave us this table

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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 (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	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 used a sum function to calculate the total payload carried by boosters from NASA

```
%sql select SUM("PAYLOAD_MASS__KG_") from SPACEXTABLE
```

SUM("PAYLOAD_MASS__KG_")
619967

Average Payload Mass by F9 v1.1

- Using an average function we calculated the average payload mass carried by booster version F9 v1.1

```
%sql select avg("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Booster_Version" like "F9 v1.1%"
```

avg("PAYLOAD_MASS__KG_")
2534.6666666666665

First Successful Ground Landing Date

- Using a min function we found the date of the first successful landing outcome on ground pad

```
%sql select min("Date") from SPACEXTABLE where "Landing_Outcome" like "%ground pad%"
```

min("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Using the below query we listed the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
%sql select "Booster_Version", "PAYLOAD_MASS__KG_", "Landing_Outcome" from SPACEXTABLE where "Landing_Outcome" = "Success (drone ship)"
and "PAYLOAD_MASS__KG_" between 4000 AND 6000
```

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- Using a count function we listed the total number of successful and failure mission outcomes

```
%sql select "Landing_Outcome", count("Landing_Outcome") from SPACEXTABLE group by "Landing_Outcome"
```

Landing_Outcome	count("Landing_Outcome")
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

- Using a subquery to find the max payload mass we found the names of the booster which have carried the maximum payload mass

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

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

2015 Launch Records

- Using some filters we found the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 with some additional data

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

substr(Date, 6,2)	substr(Date,0,5)='2015'	Landing_Outcome	Booster_Version	Launch_Site
01	1	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	1	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Using ORDER BY we ranked 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") from SPACEXTABLE group by "Landing_Outcome" order by count("Landing_Outcome") desc
```

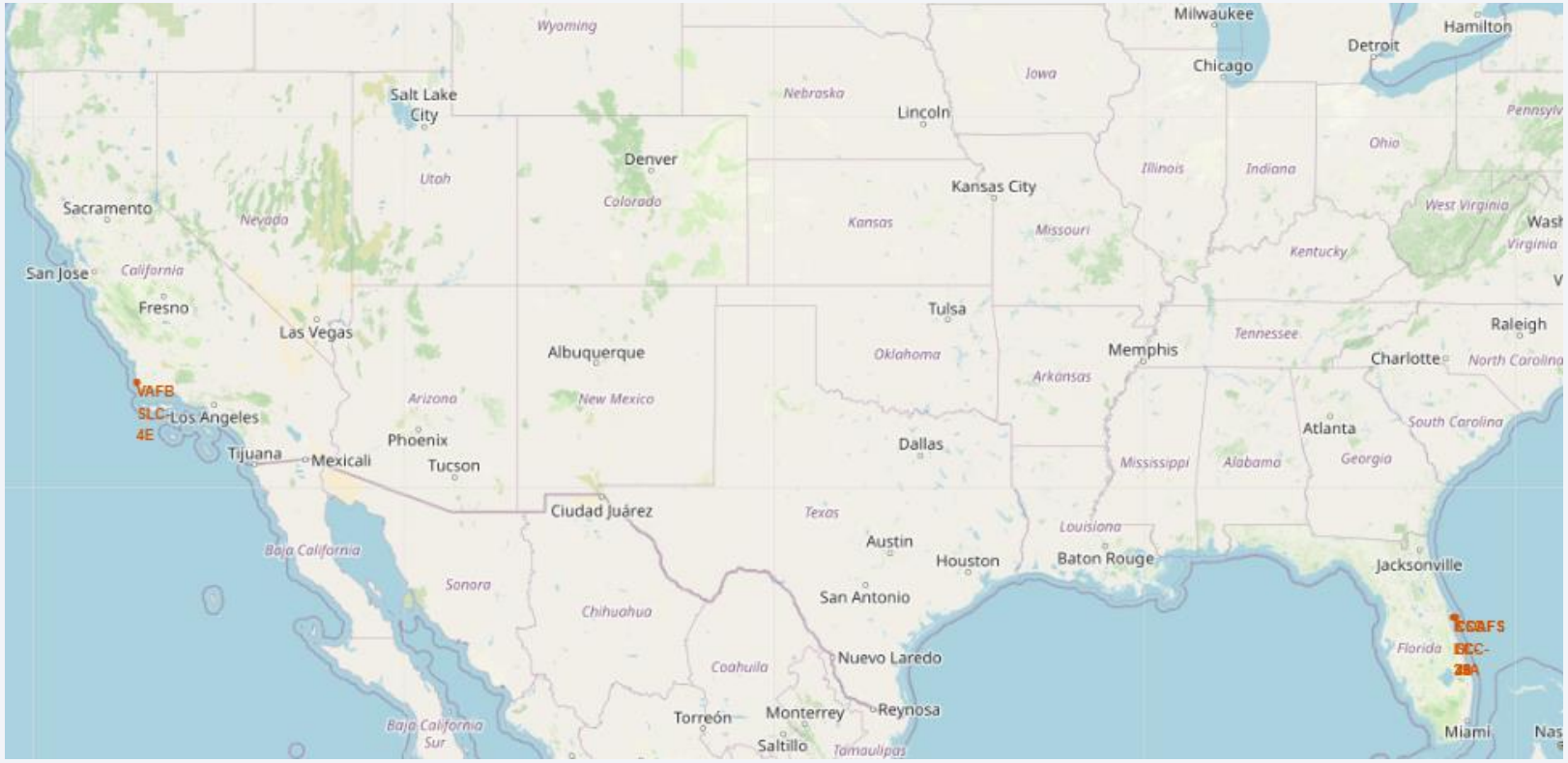
Landing_Outcome	count("Landing_Outcome")
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1

Section 3

Launch Sites Proximities Analysis

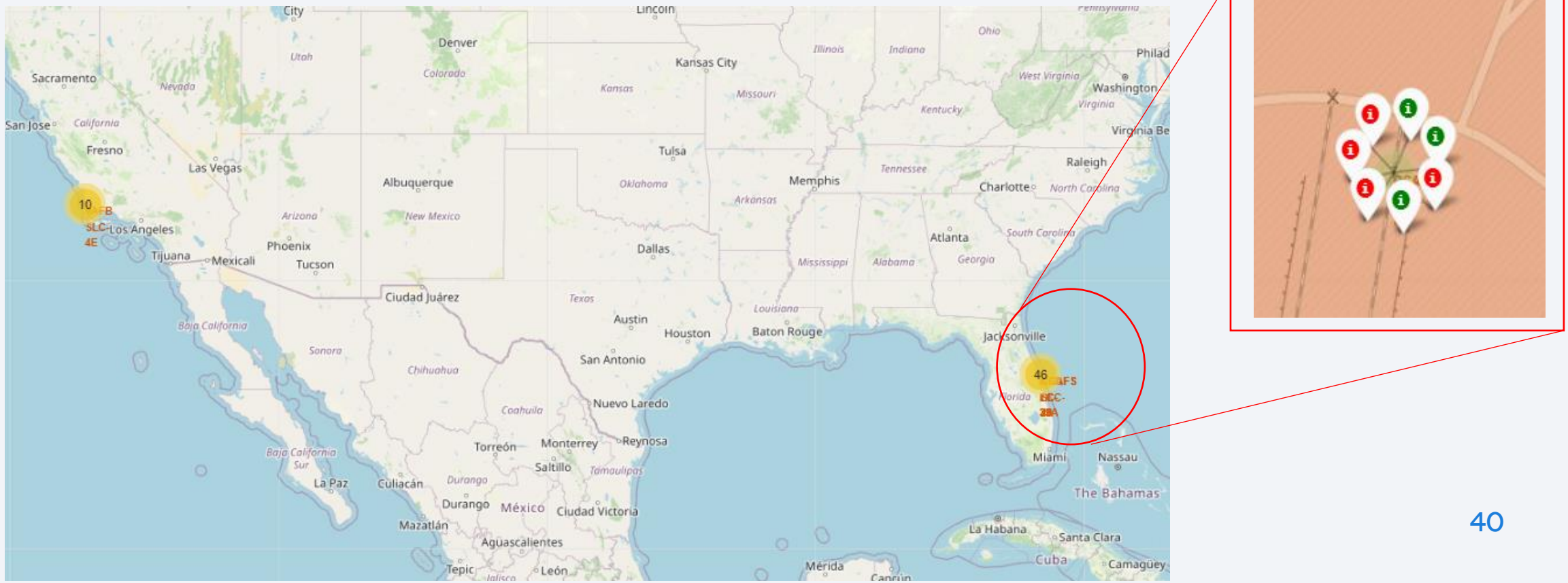
Interactive Launch Map

- Initially we created a folium map to plot the launch sites to see their general location on a map showing that they are all by the coast



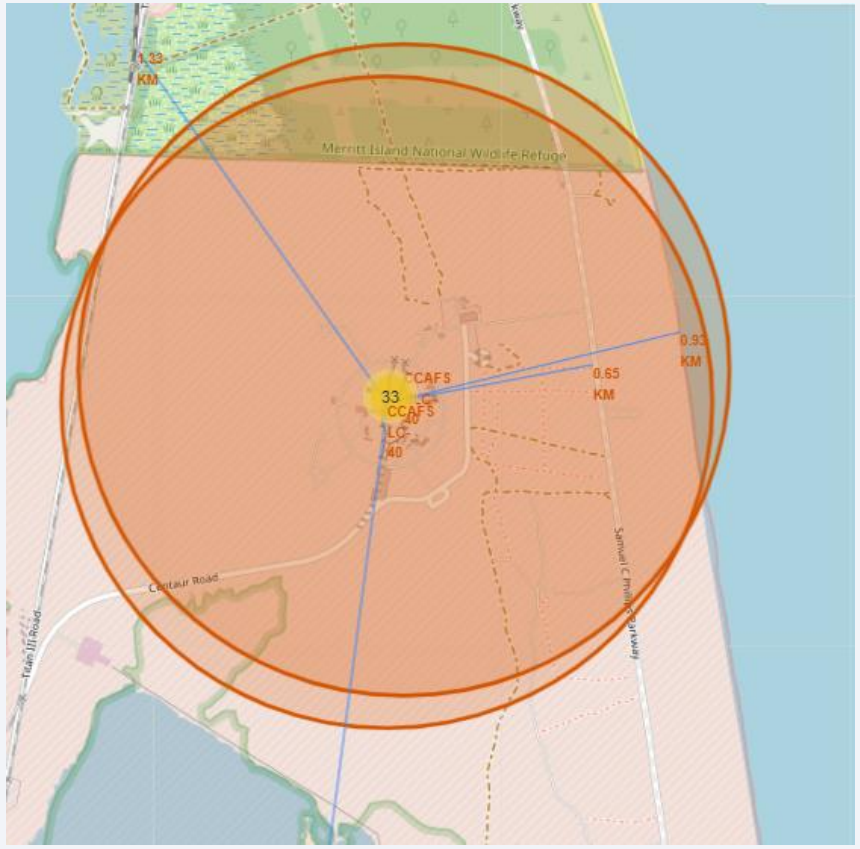
Interactive Launch Map

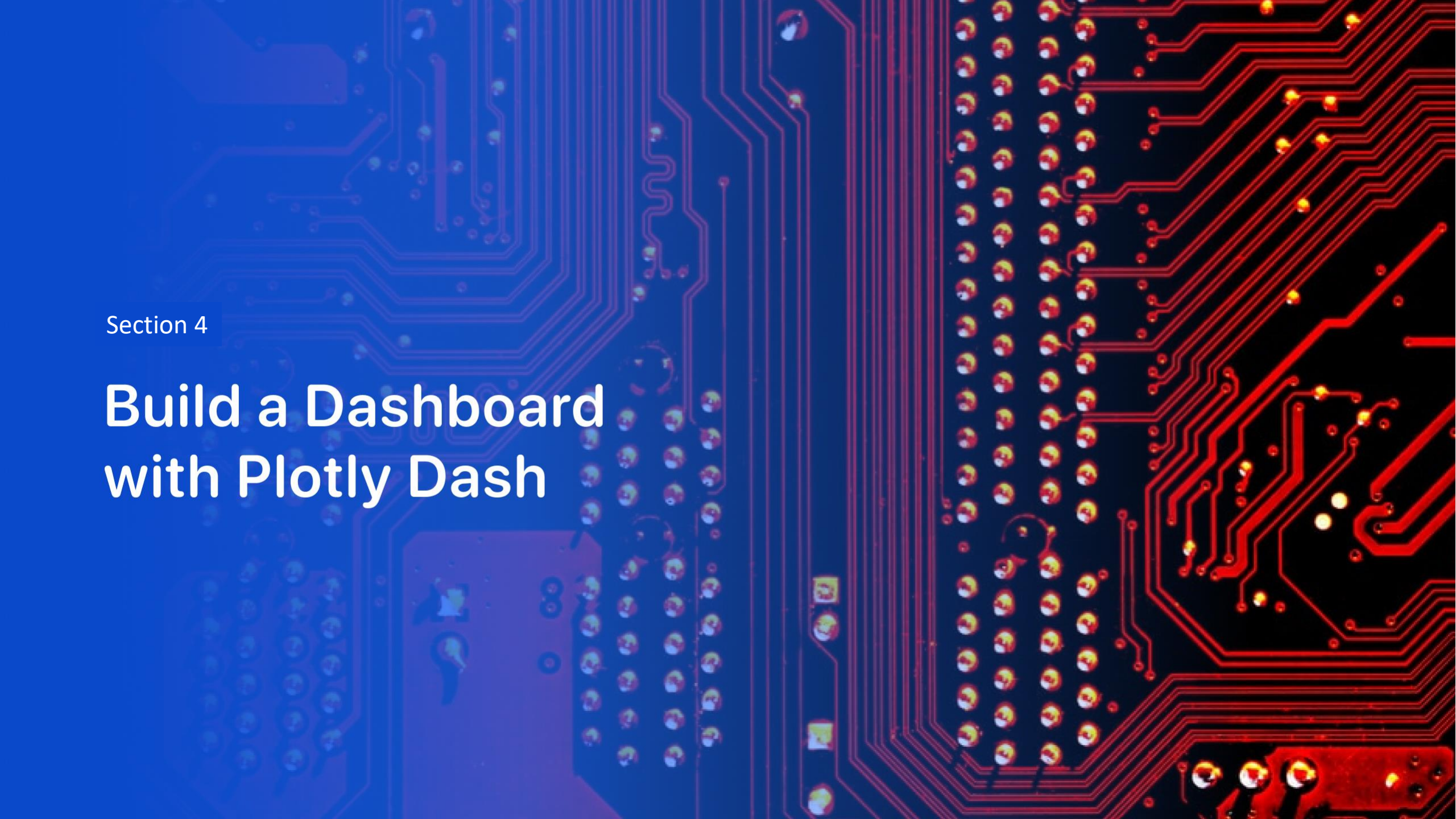
- Using Folium we added marks to the interactive map to plot each launch and their launch sites and their results showing that most flights happen in the east coast



<Folium Map Screenshot 3>

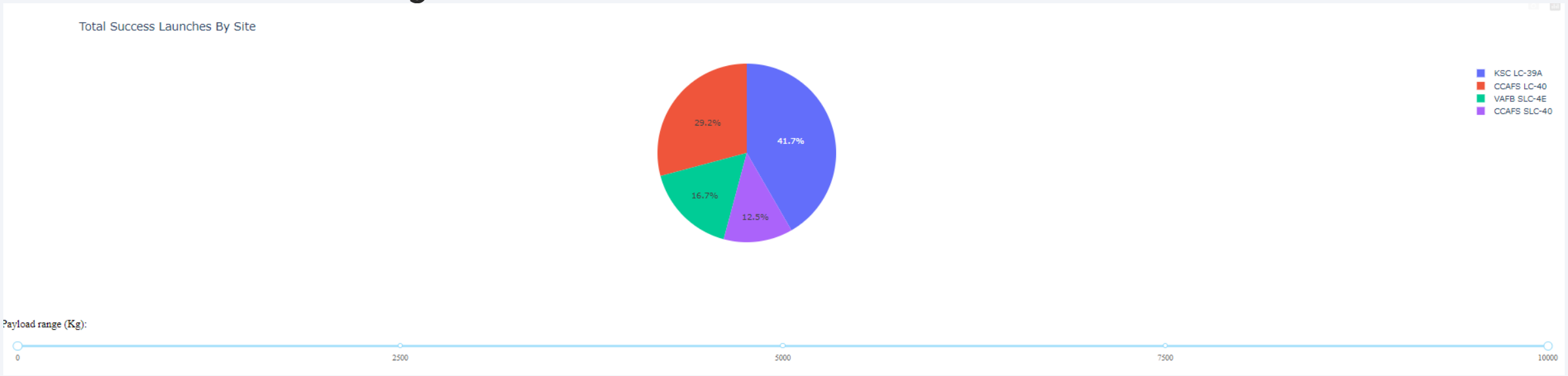
- Lastly we analyzed the proximity of the site to certain geographical features and found that they are all in close proximity to highways, coastlines, and railways, however, far from cities. This could be to allow the launch to land in the water safely and to bring in supplies while not disrupting the city.




Section 4
Build a Dashboard with Plotly Dash

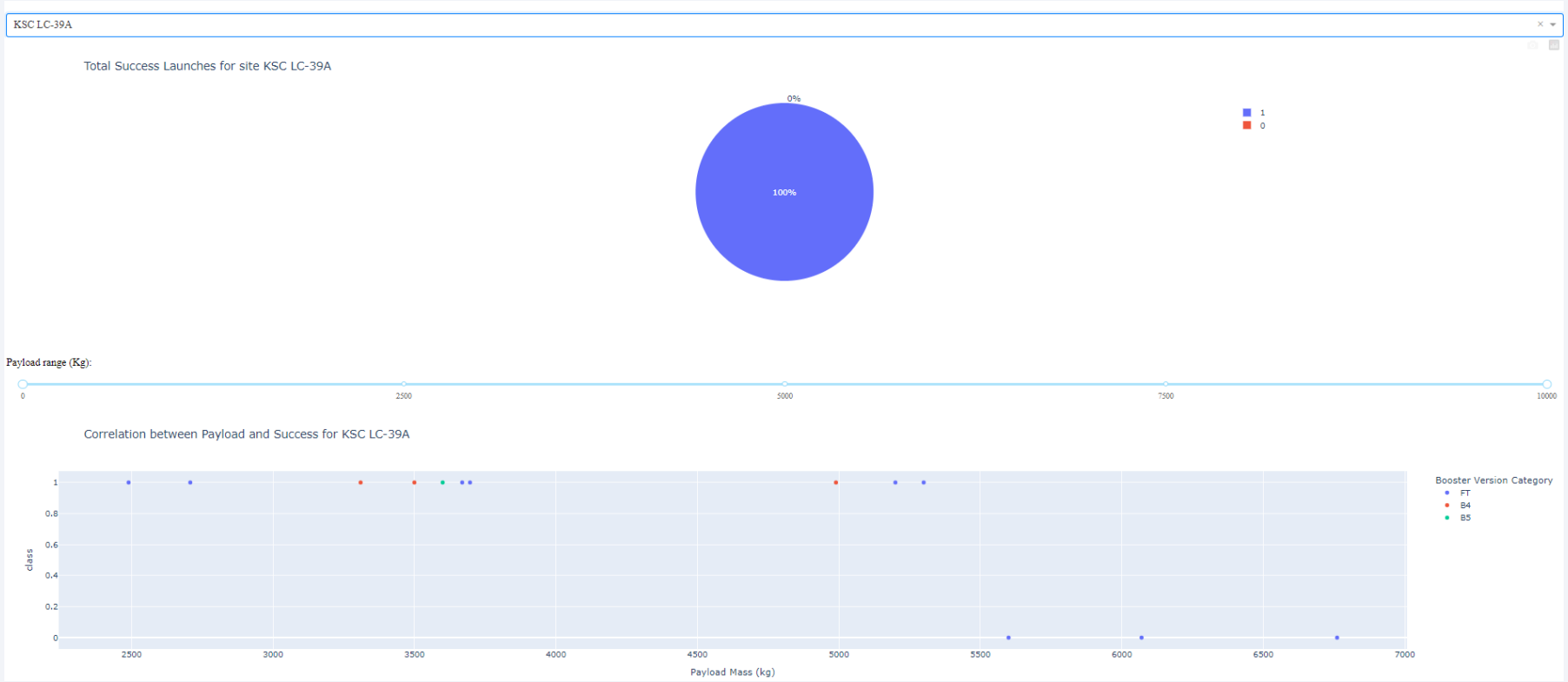
Interactive Dashboard with Dash

- We created a highly interactive dashboard showing charts for different sites and the success per site and payload range. This pie chart showed that KSC LC-39A had the highest success launches



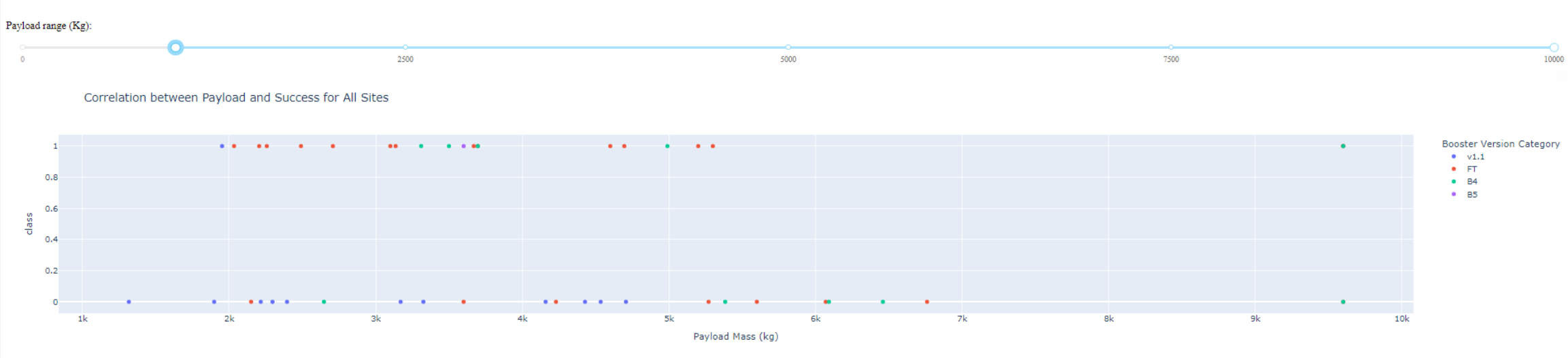
Interactive Dashboard with Dash

- When drilling down to the launches for KSC LC-39A you can see that they performed well, however for higher payload masses, the data shows that it did not seem to perform well



Interactive Dashboard with Dash

- Looking at the correlation between mass and sites, it seems that there are less flights, and higher mass flights mostly happen with booster version B4 and FT

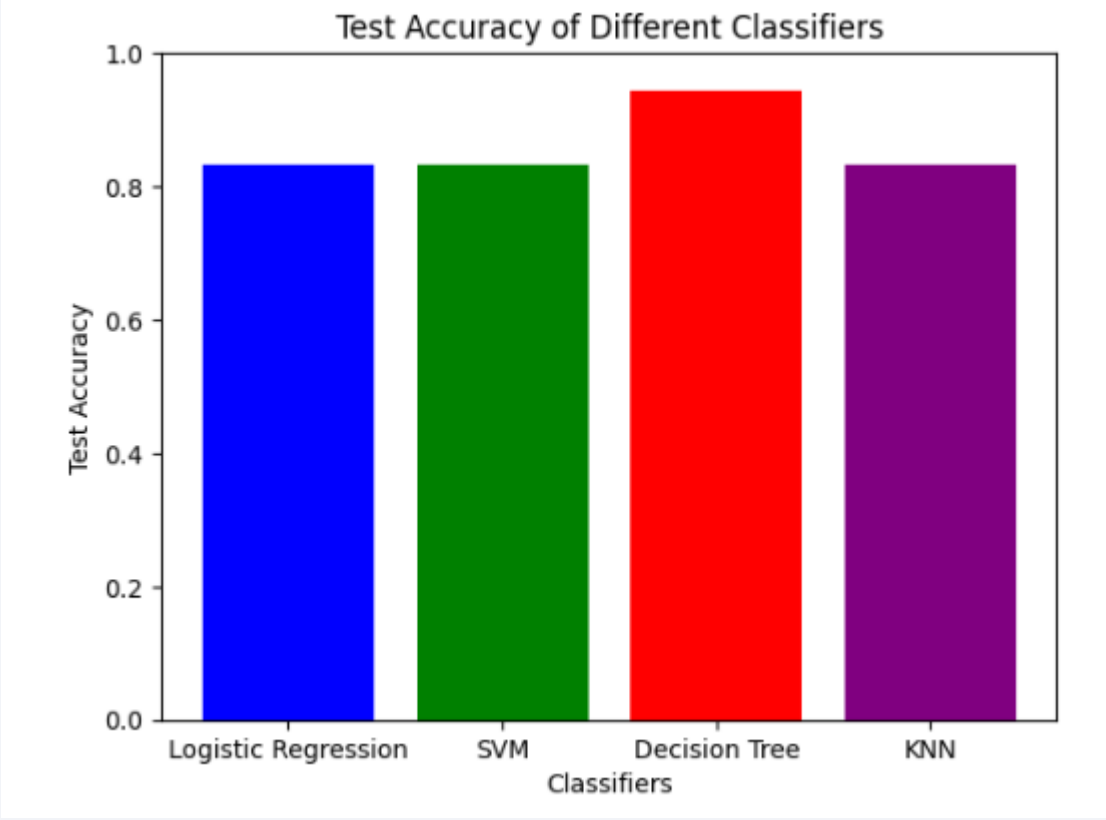


Section 5

Predictive Analysis (Classification)

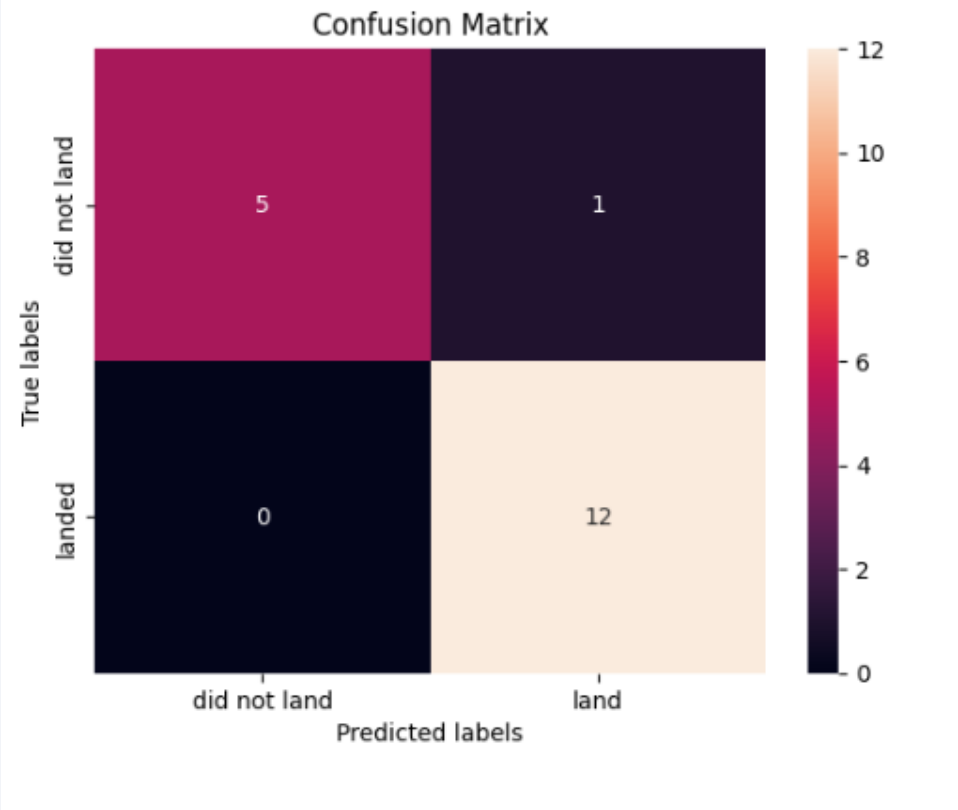
Classification Accuracy

- Using Matplotlib we can visualize which ML model had the greatest accuracy. In this case it is the decision tree algorithm with the parameters
- Best Decision Tree Parameters: {'criterion': 'gini', 'max_depth': 2, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'splitter': 'best'}



Confusion Matrix

- The confusion matrix shows that the model accurately predicted most of the test samples with only 1 false positive



Conclusions

- The more tests that are performed (higher flight number) the greater the chance of success, as the rockets are further developed and tested.
- The mass of the payload is significant and different versions should be used dependent on the mass.
- Each launch site should be situated close to the coast for safety and far from cities'
- The decision tree classifier had the best performance compared to other models making it the best predictor for an ML model

Thank you!