IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Daniel Ladage
6/19/2023

Outline

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

Executive Summary

- Methodology summary:
  - First we gathered the data and then organized it,
  - Next we visualized the data into interactive maps, charts, and tables
  - Finally we ran several machine learning models to get the best predictive model possible.
- Summary of results:
  - Re-using the 1st stage of a rocket saves the most money
  - The location KSC LC-39a has the highest success average for launches
  - The K nearest Neighbor model accurately predicted the launch data at 100%

Space Y: Who we are and what we are about!

- We are a Space Y, a start-up space travel company looking make space travel more affordable!

- With funding from Billionaire industrialist Allon Musk We are looking to compete with Space X in this seemingly untapped market.

- Goals for this phase of Space Y growth:
  - Determine the price of each launch.
  - Determine if the first stage would be reused

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - We gathered data via web-scraping information off of websites in the public domain and the SpaceX Rest AP
- Perform data wrangling
  - We converted the raw data into tables and created a classification of successful and unsuccessful launches
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We then trained several different models to test the data and came up with the best predictive model

Data Collection

- Data was collected via two popular methods Web scraping and the SpaceX Rest API
- Rest API
  - Started by requesting rocket launch data from SpaceX API
  - Request and parse the SpaceX launch data using the GET request
  - Filter the data frame to only include Falcon 9 launches
  - Deal with Missing Values
- Web scraping
  - Retrieve information
  - Extracting column and variables from html code
  - Create a data frame by parsing

Data Collection – SpaceX API

Request and parse the SpaceX launch data

- `data = pd.json_normalize(response.json())`
- Use API to get information about the launches using the IDs given for each launch
- Finally construct the dataset using the data obtained and combining it to a dictionary

Filter the dataframe

- `data_falcon9 = data.loc[data['BoosterVersion'] != "Falcon 1"]`
- `data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))`
- Wrangle data to find missing values

Deal with Missing Values

- *Calculate the mean value of PayloadMass column*
- `data_falcon9['PayloadMass'].mean()`
- *Replace the np.nan values with its mean value*
- `data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, 6124)`

- https://github.com/dannybravo599/capstone-project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Retrieve information

- data= requests.get(static_url)
- Create a beautiful soup with soup = BeautifulSoup(data.content, "html.parser")

Extracting column and variables from html code

- `html_tables = soup.find_all('table')`
- iterate through the `<th>` elements and apply the provided `extract_column_from_header()` to extract column name

Create a data frame by parsing

- create an empty dictionary with keys from the extracted column names
- fill up the `launch_dict` with launch records extracted from table rows
- create the dataframe

- https://github.com/dannybravo599/capstone-project/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- Data was processed through several SQL queries and various Data Graphics

- Including:
  - Line charts
  - Bar graphs
  - Scatterplots

- Related git hub links
  - https://github.com/dannybravo599/capstone-project/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb
  - https://github.com/dannybravo599/capstone-project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with Data Visualization

- Charts that were used
- Scatter plots
  - Payload Mass vs Flight Number vs Class (used to determine relationship between weight and a successful recovery of the first stage)
  - Launch Site vs Flight Number vs Class (used to determine relationship between a launch sites success)
  - Orbit vs Flight Number vs Class (used to determine if desired orbit affected success)
- Line Chart
  - Year vs success rate (used to determine if the odds of success increased as time continues)

    - https://github.com/dannybravo599/capstone-project/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Display the names of the unique launch sites in the space mission : **%sql** SELECT DISTINCT Launch_Site FROM SPACEXTBL;
- Display 5 records where launch sites begin with the string 'CCA : **%sql** SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'c%' LIMIT 5;
- Display the total payload mass carried by boosters launched by NASA (CRS): **%sql** SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'
- Display average payload mass carried by booster version F9 v1.1: **%sql** SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1'
- List the date when the first succesful landing outcome in ground pad was acheived.: **%sql** SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success (ground pad)'
- 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 LIKE 'Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000
- List the total number of successful and failure mission outcomes: **%sql** SELECT COUNT(Launch_Site) FROM SPACEXTBL GROUP BY Mission_Outcome
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery: **%sql** SELECT booster_version from SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)FROM SPACEXTBL)
- 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 substr(DATE,4,2), LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Failure (drone ship)' and substr(Date,7,4)
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order: **%sql** select distinct landing_outcome, date from SPACEXTBL where landing_outcome like 'S%' and date between '04-06-2010' and '20-03-2017' limit 8

https://github.com/dannybravo599/capstone-project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Markers and lines on Folium maps
  - Launch sites (to determine where geographically were each launch occurred)
  - Line to coast line (to help determine if having a coast line near would lead to better results)
  - Marker clusters (to help see the success rate of launch locations)
  - Lines to rail roads and cities (to help determine if being next to a rail road is helpful and how far away from major cities are the launches)

Link to GitHub:

- https://github.com/dannybravo599/capstone-project/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

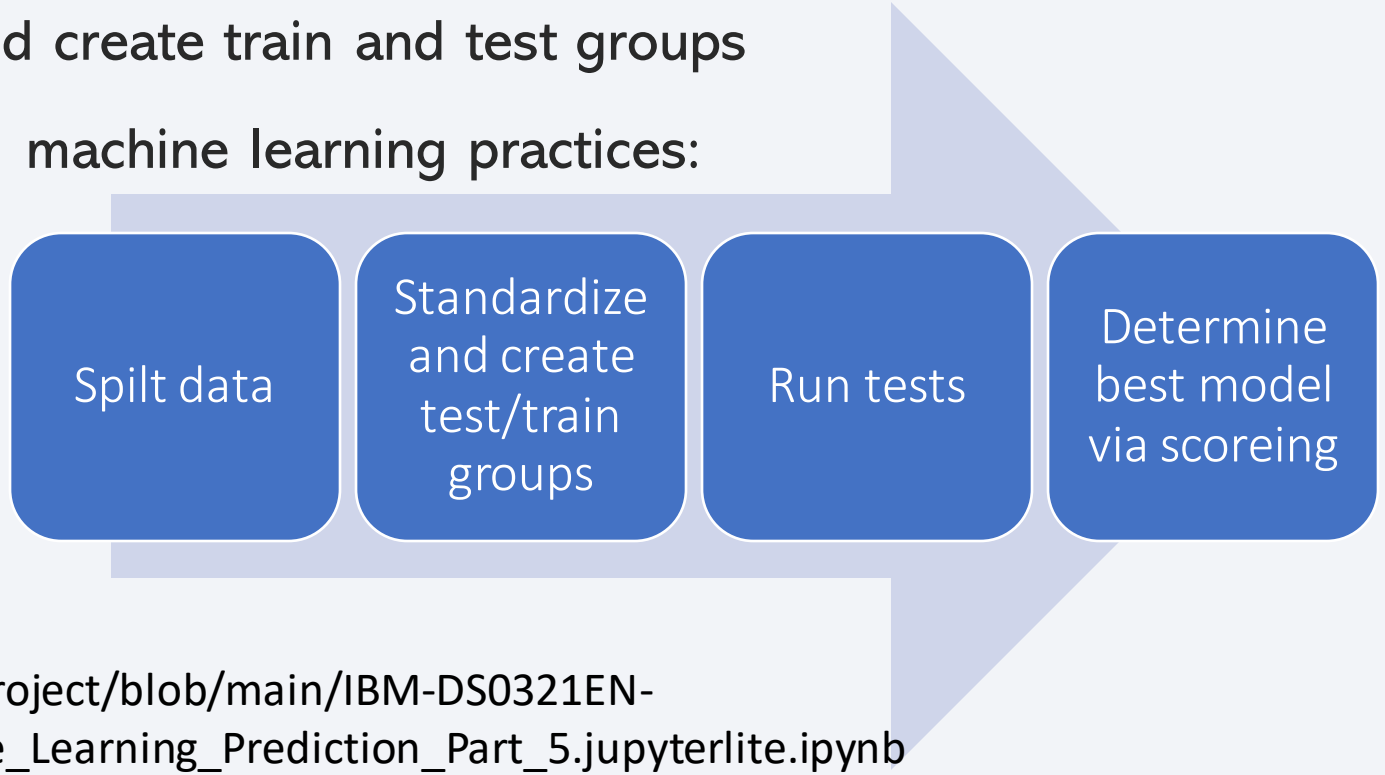
Build a Dashboard with Plotly Dash

- For dash A pie chart titled "Success count for Launch Sites", and a scatter plot titled "Success count on payload mass for all sites" were created for each site.
- The web page allows you to select a launch site, or see all the sites together, and you can see the charts listed, also by hovering over the chart you can see more in depth info on each chart.
- It was important to add those charts to help illustrate which sites had the most success in launches and to help visualize the relationship that payload mass has on the success rate.

- Git hub for dash code:
  - https://github.com/dannybravo599/capstone-project/blob/main/dash%20code

Predictive Analysis (Classification)

- To perform the predictive analysis we had to first split the data into sets
- Next was to standardize the data and create train and test groups
- Then we ran through some standard machine learning practices:
  - Logistic regression
  - Support vector machine
  - Decision tree
  - K nearest neighbor



https://github.com/dannybravo599/capstone-project/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

Eda resulted in a high correlation of mass and desired orbit in the success of retrieving a first stage , we were also able to determine the best locations to launch from and the best strategies to recover said first stage.

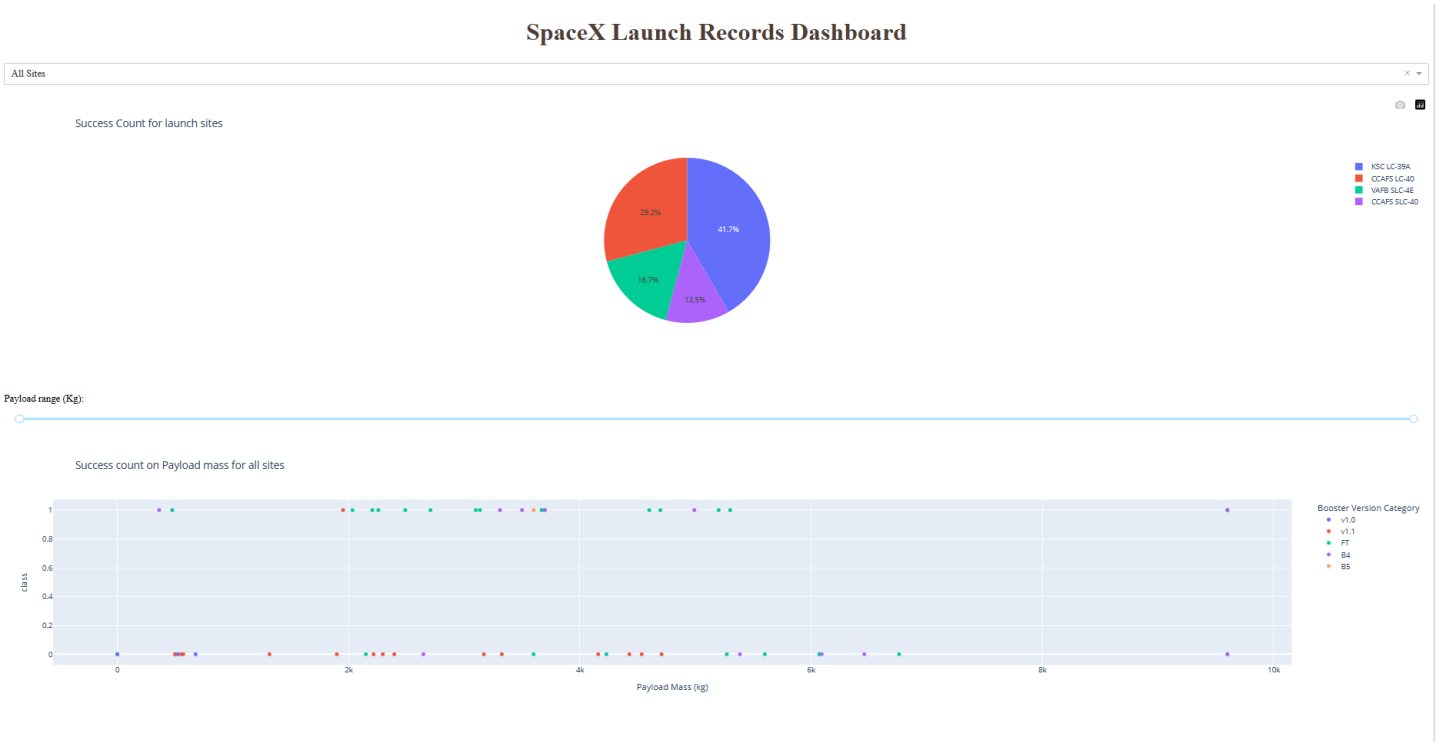
Predictive analysis results showed that the k nearest neighbor was the best model for predicting the outcome of the class variable with the accuracy of 1.0

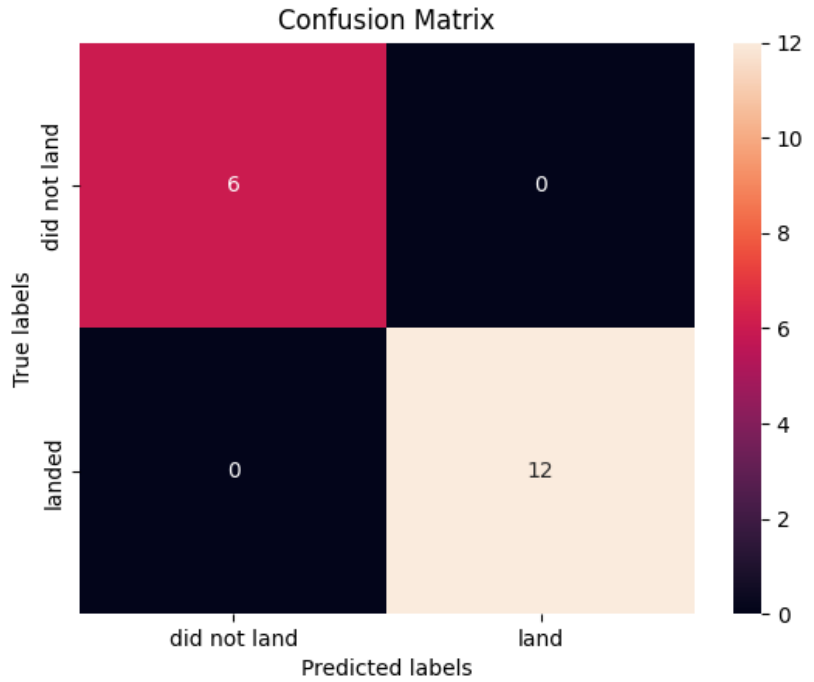
In [41]:
```
knn_cv.score(X_test, Y_test)
```

Out[41]: 1.0

We can plot the confusion matrix

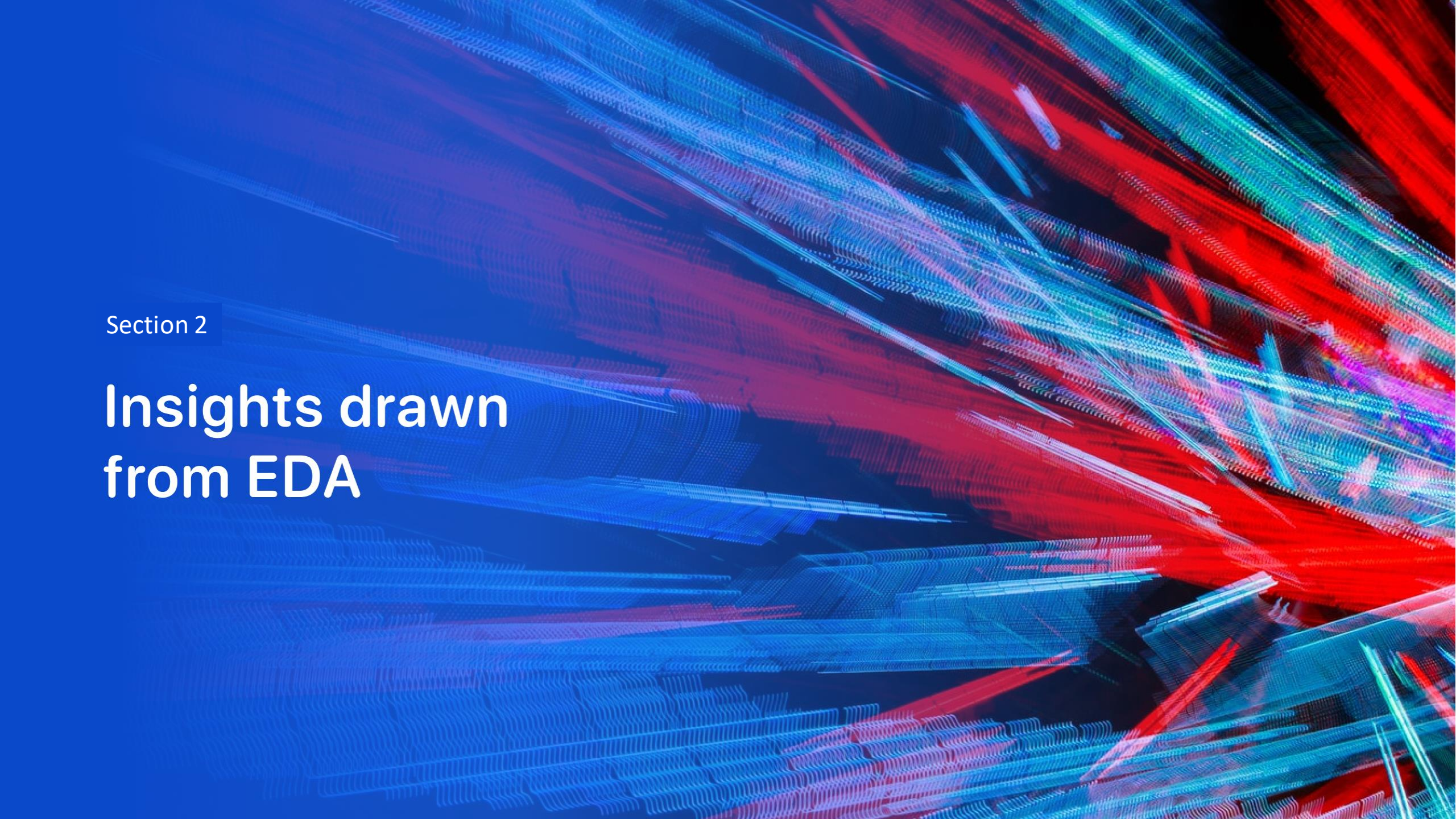
In [42]:
```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```





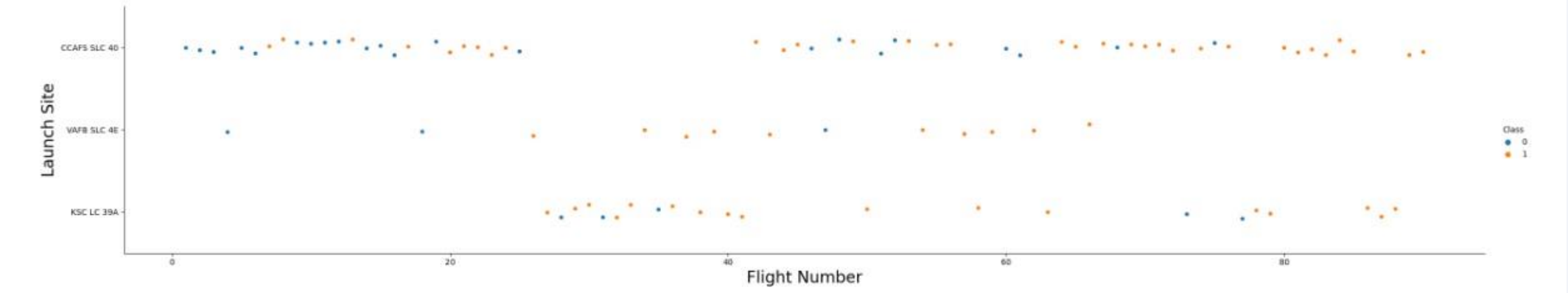
Section 2

Insights drawn from EDA



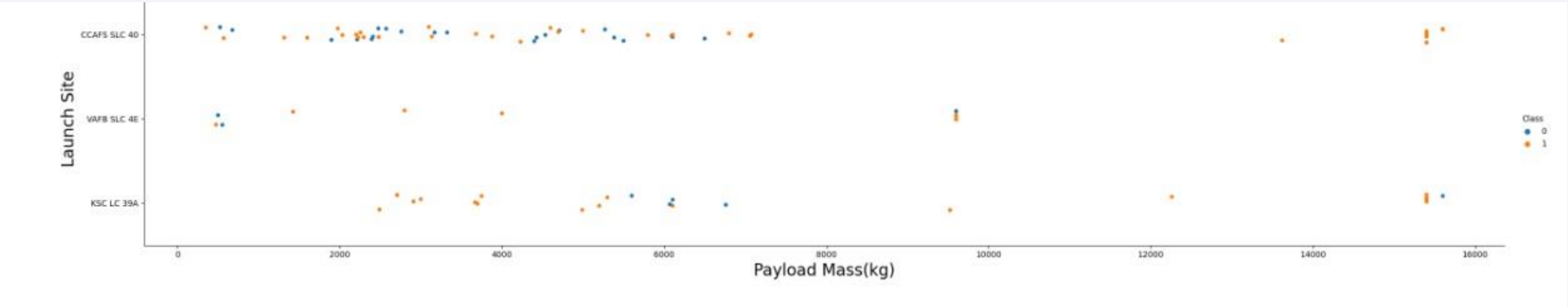
Flight Number vs. Launch Site

- sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
- plt.xlabel("Flight Number",fontsize=20)
- plt.ylabel("Launch Site",fontsize=20)



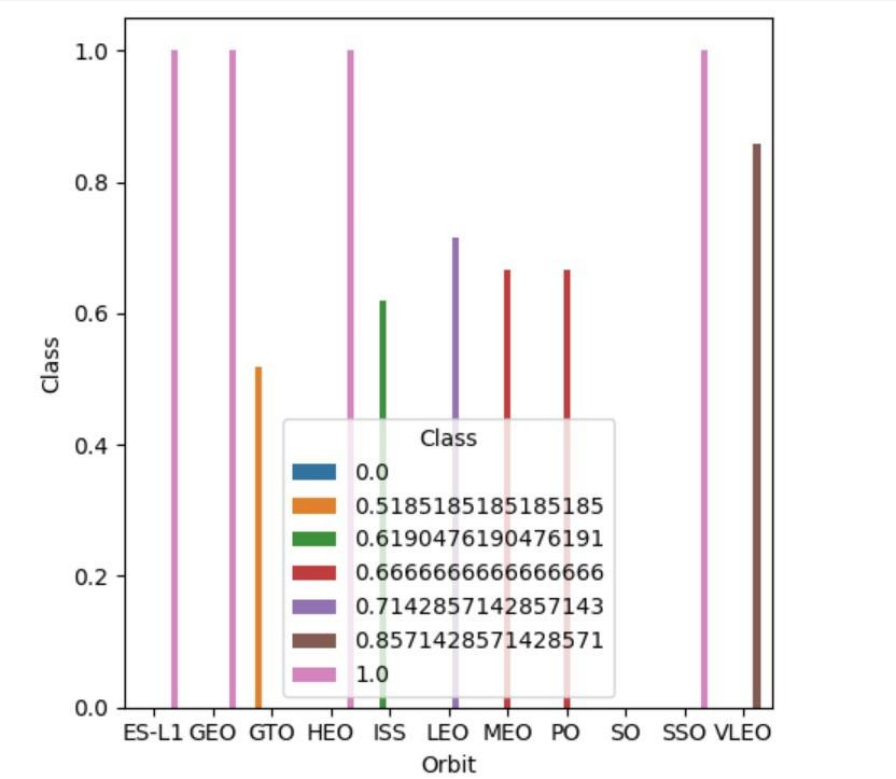
- Explanation: As More launches were attempted at each location the success rate of capture increased. This can be due to an increase of technology and just plain and simple trail and error.

Payload vs. Launch Site

- sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
- plt.xlabel("Payload Mass(kg)",fontsize=20)
- plt.ylabel("Launch Site",fontsize=20)
- 

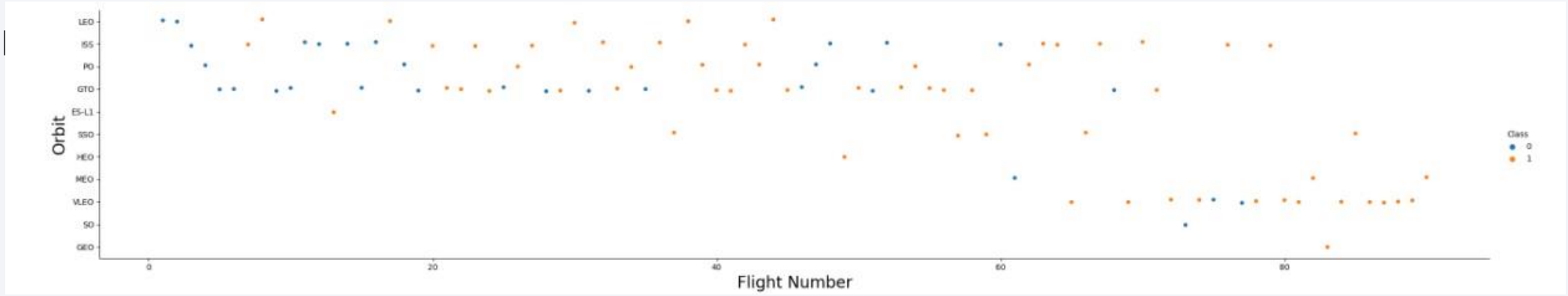
- Explanation: The result show that the higher the payload the more successful the recovery of the first stage is

Success Rate vs. Orbit Type



- Explanation: we can see with this chart that SSO, HEO, GEO, and ES-I1 have the highest average success rate.

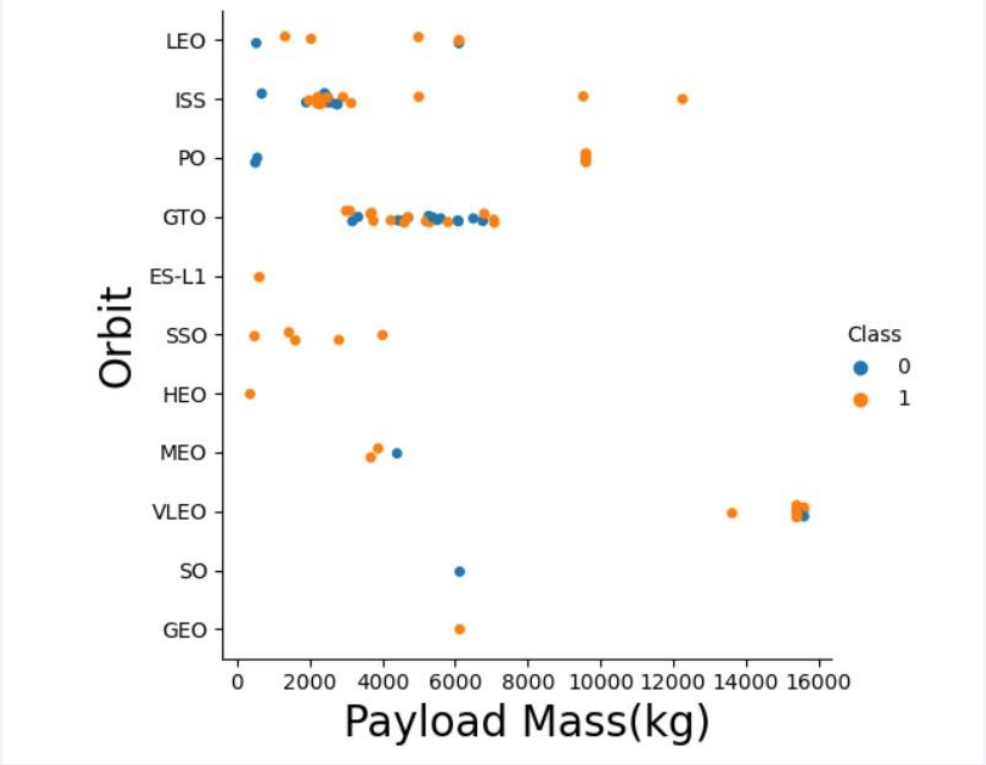
Flight Number vs. Orbit Type

- sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
- plt.xlabel("Flight Number",fontsize=20)
- plt.ylabel("Orbit",fontsize=20)
- 

- Explanation: in 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

- sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df)
- plt.xlabel("Payload Mass(kg)",fontsize=20)
- plt.ylabel("Orbit",fontsize=20)
- plt.show()

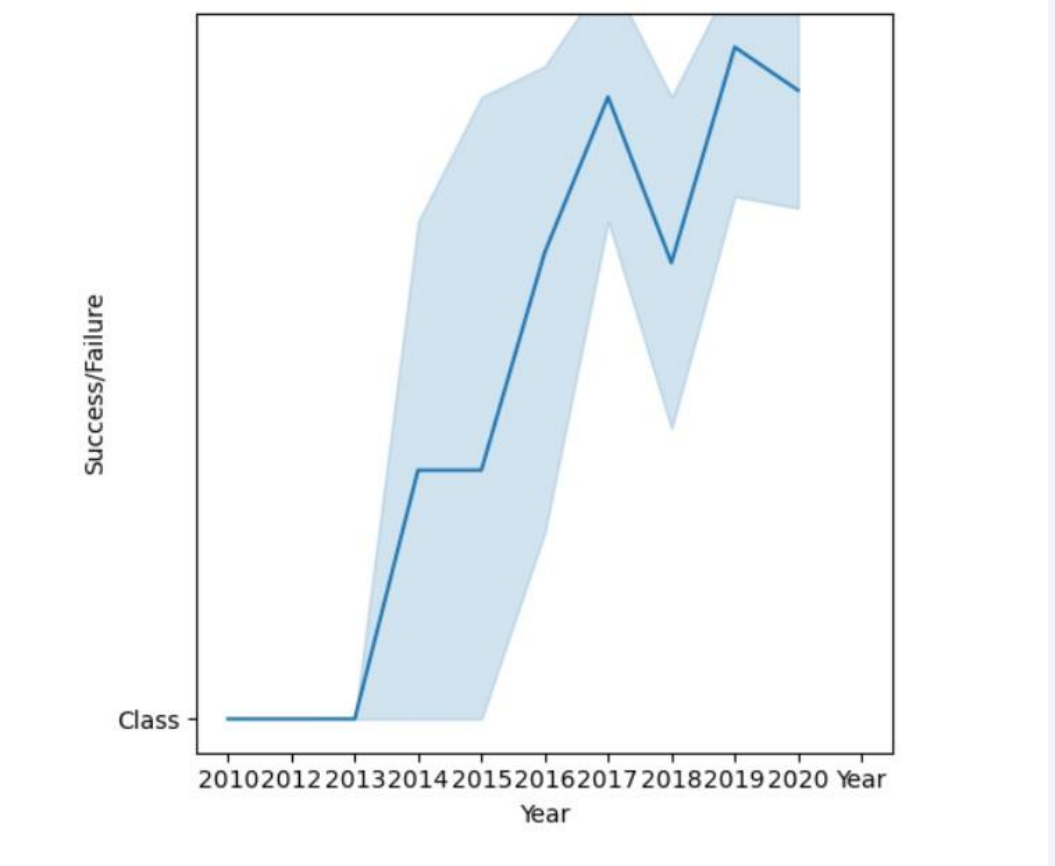


- Explanation: With heaver payloads, successful landings are more for polar, LEO and ISS but there is not enough info to come to a conclusion on GTO

Launch Success Yearly Trend

- plt.plot(["Year"],["Class"])
- plt.xlabel("Year")
- plt.ylabel("Success/Failure")
- plt.show()

- Explanation: As years pass the success rate increases, this is most likely due to improvements of technology



All Launch Site Names

- Find the names of the unique launch sites
- Results: these are the five sites, in the data they are abbreviated but they represent real world launch sites such as nasa.

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- This shows that the mission outcomes for NASA location are generally successful

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (paracl
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (paracl
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No att
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No att
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No att

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- This is the total payload mass carried by boosters launched by NASA

SUM(PAYLOAD_MASS__KG_)
45596.0

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- This is the average payload mass carried by the falchon 9 v1.1 booster

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- I used the min function to select the first date that there was a successful landing on ground pad
- MIN(Date) 01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- By looking for these boosters we can find which one will give us the best odds of landing on a drone ship.

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- By showing the mission outcome we can see the success rate for the mission outcome heavily leans to what the desired outcome that is initially predicted

COUNT(Launch_Site)
0
1
98
1
1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

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

Out[45]:

substr(DATE,4,2)	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

- By finding the failures and comparing them to the success we can determine what may have caused the failure in the first place.

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

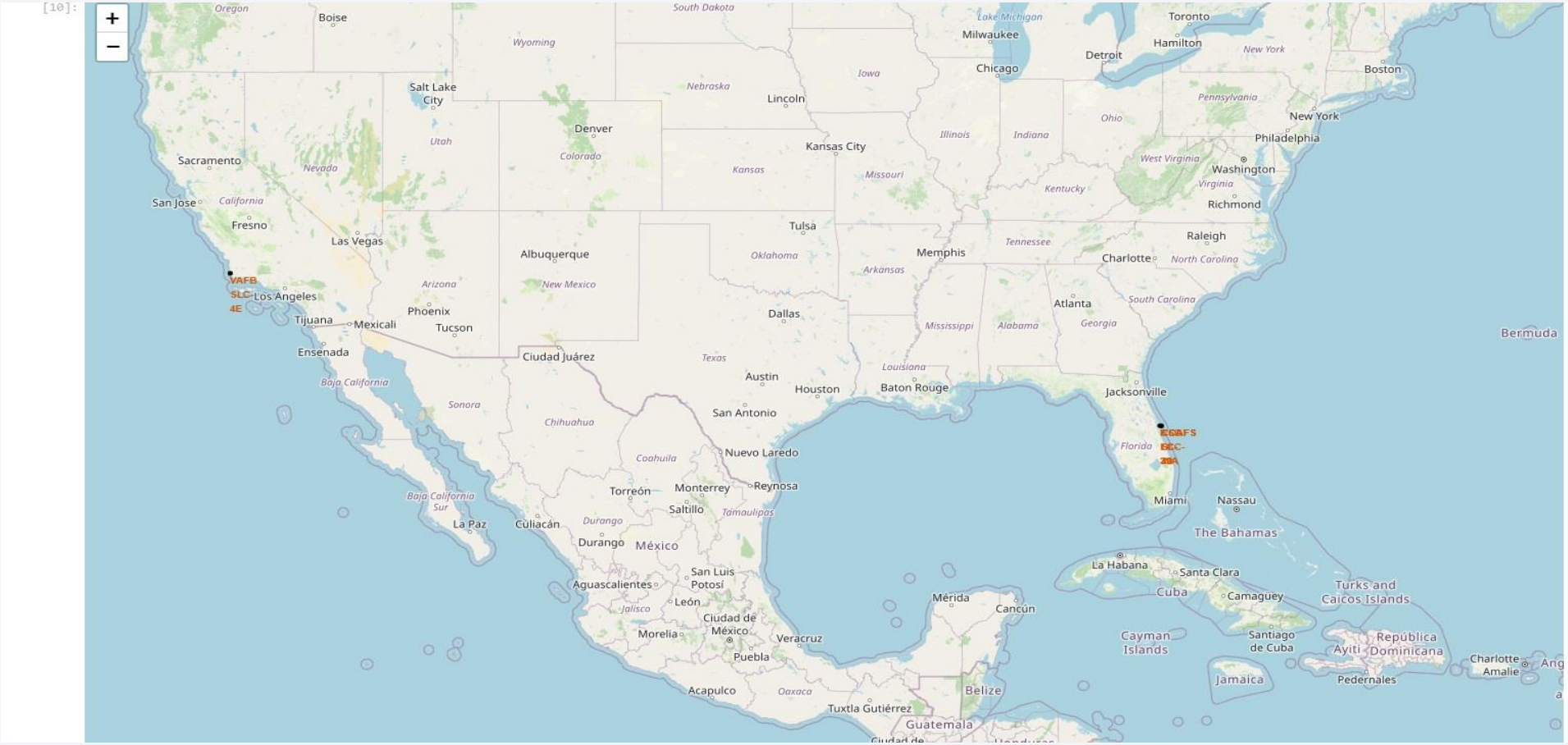
- Ground pad landings and drone ship landings were predominantly successful from the given time range!

Landing_Outcome	Date
Success (drone ship)	04/08/2016
Success (drone ship)	05/06/2016
Success (ground pad)	18/07/2016
Success (drone ship)	14/08/2016
Success (drone ship)	14/01/2017
Success (ground pad)	19/02/2017
Success (ground pad)	05/01/2017
Success (ground pad)	06/03/2017


Section 3
Launch Sites
Proximities Analysis

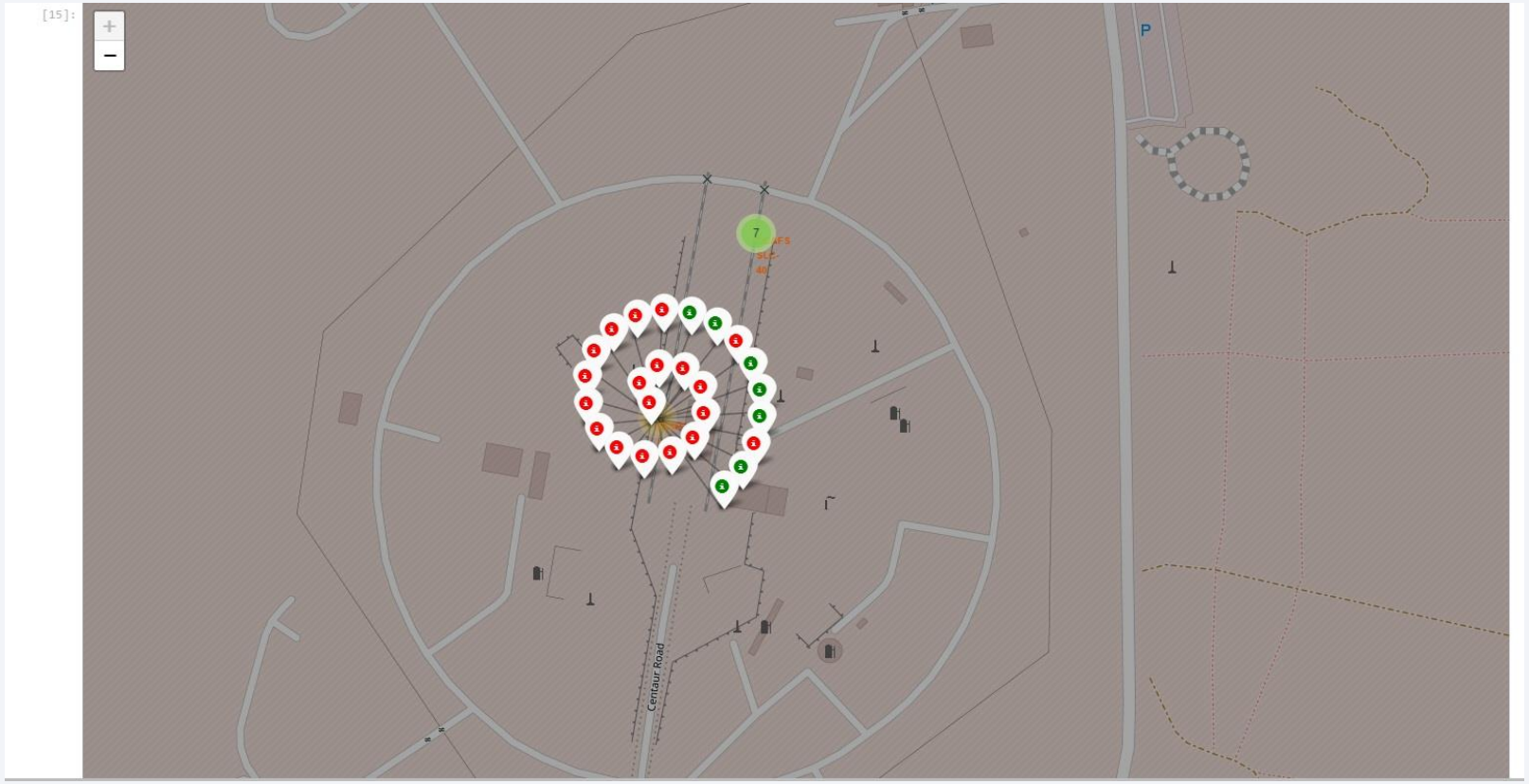
Launch site Locations

[10]:



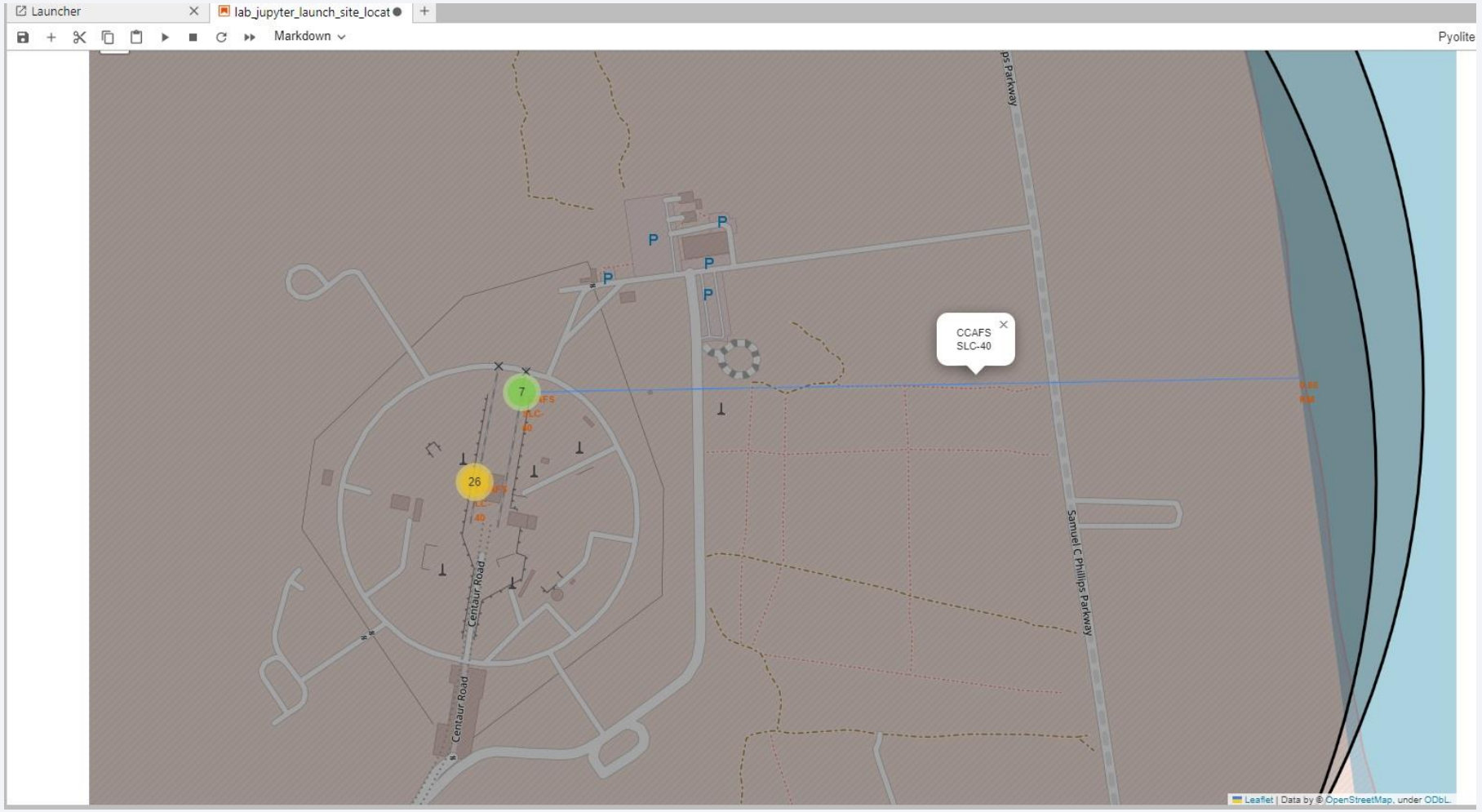
- As we can determine from the map that all launch sites are fairly close to the equator and are along the coast. This is most likely due to prevent the launch crashing into a city upon failure

Launch outcomes

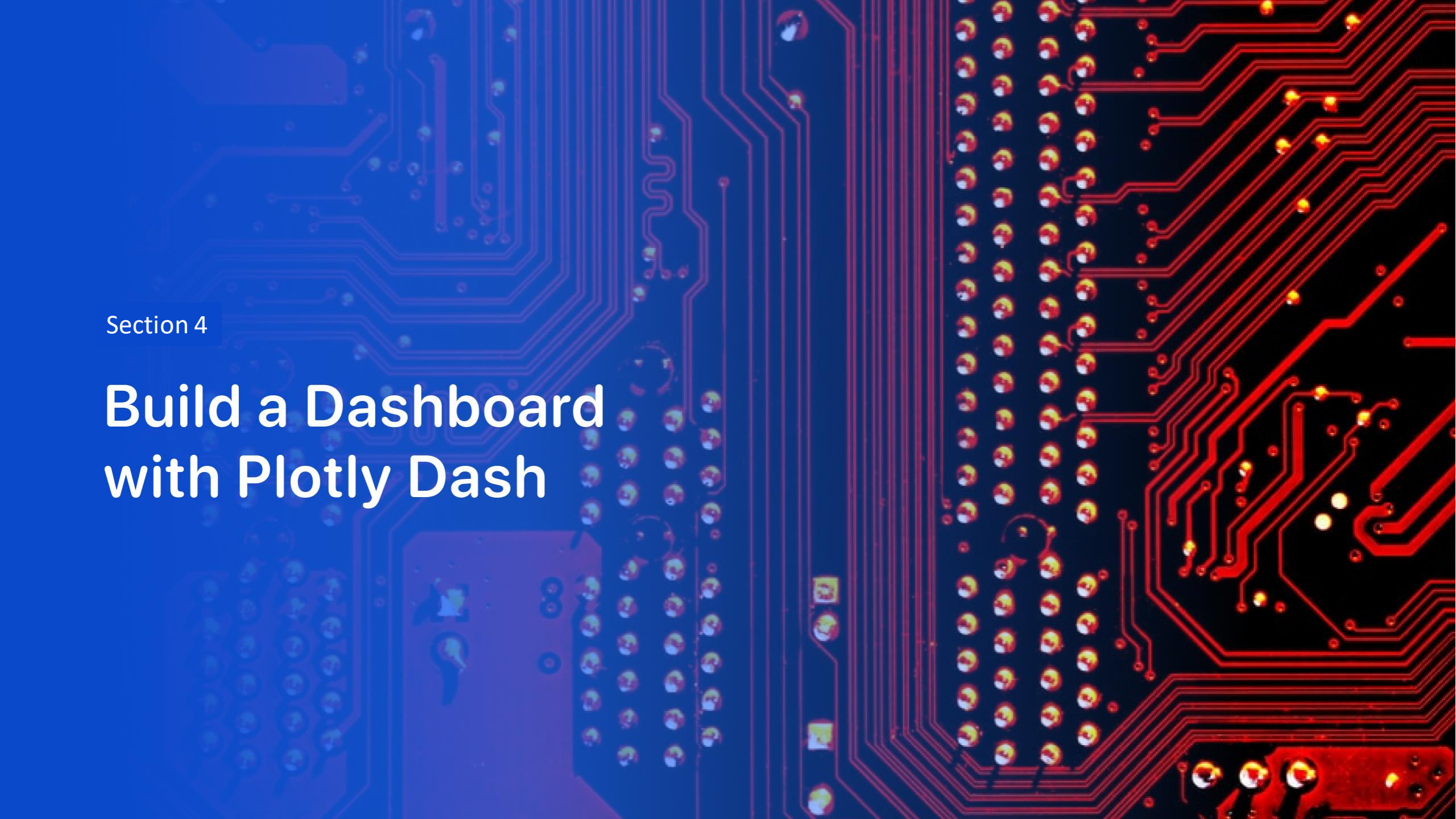


- These markers represent the cluster of launches from this specific site, red meaning failure green meaning success, also the green 7 is another site and if we were to select it on the map it would show the 7 seven results of the launches on that site.

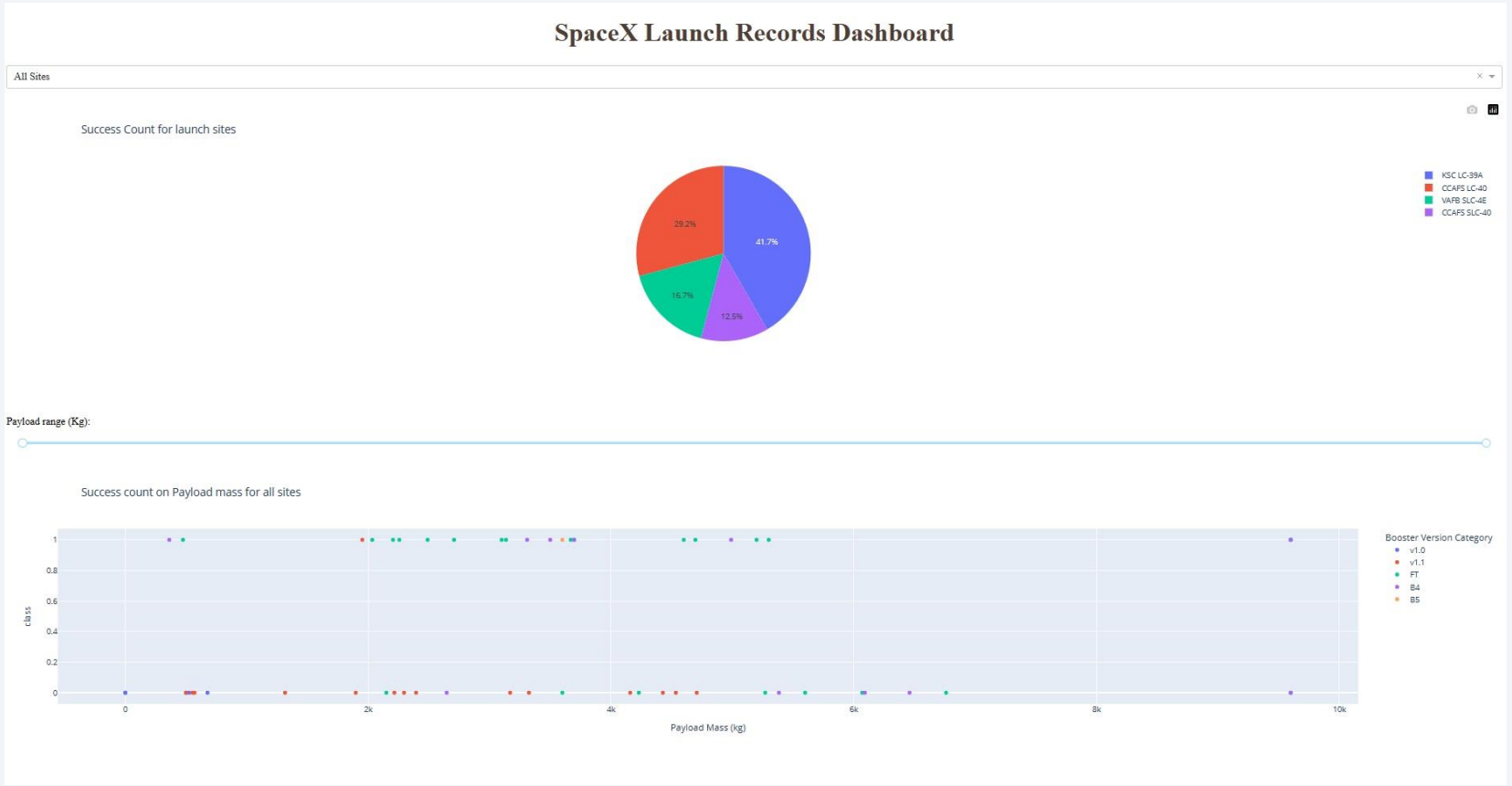
Proximities to launch site



- With this line it shows how close the coastline is to the launch site. Being able to illustrate the distance to key locations near a launch site can lead to greater Insite on what it takes to have a successful mission outcome!

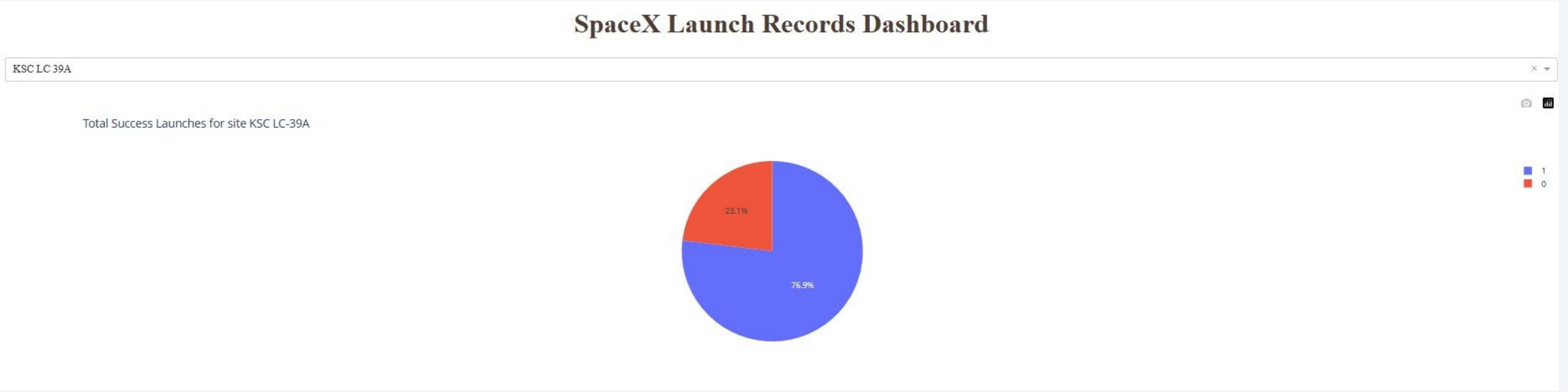

Section 4
Build a Dashboard with Plotly Dash

SpaceX Launch Dashboard (ALL Sites)



- From this pie chart we can see that KSC LC-39a has the most successful launches out of all locations.

Highest Launch Rate Success



- This pie chart shows that location KSC LC 39A has the highest average success rate at 76.9%.

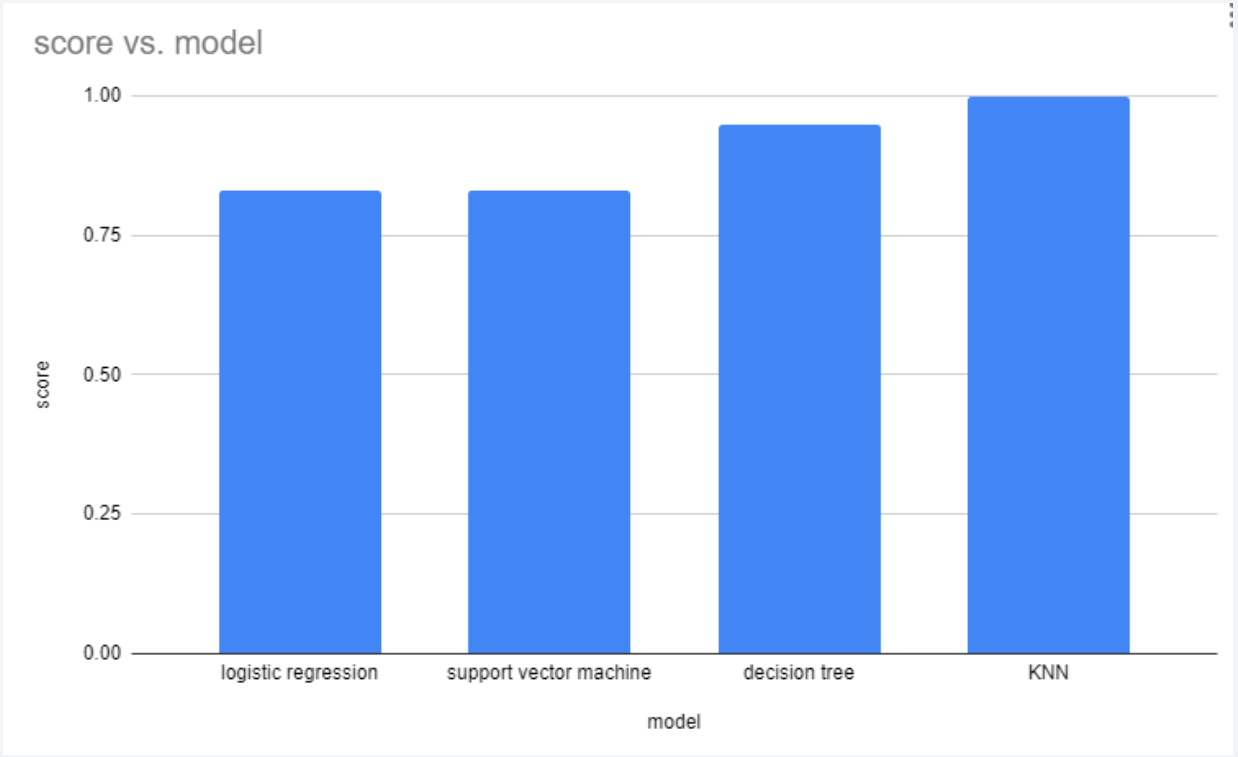
Payload vs Launch Outcome scatter plots (all sites)

These plots show that for the payload range of 0kg - 5000kgs the best booster to use would be ft booster version

Section 5

Predictive Analysis (Classification)

Classification Accuracy



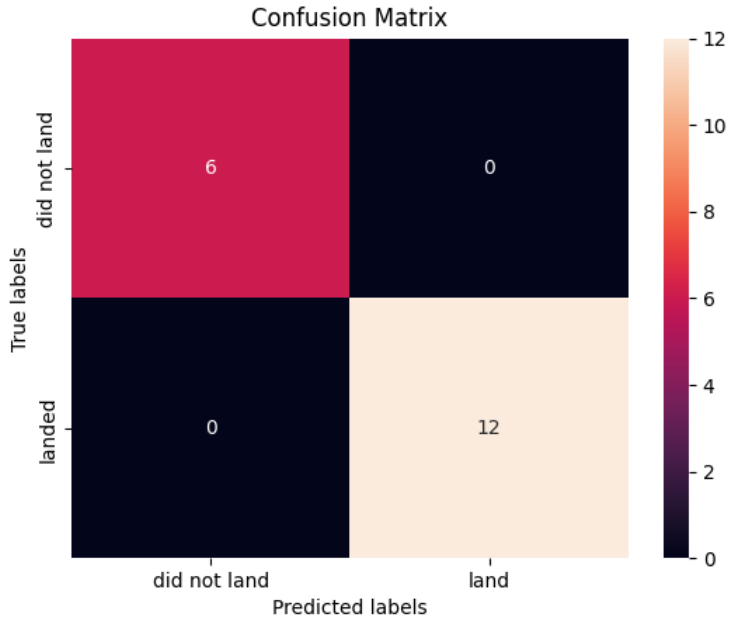
- This bar graph shows while decision tree has a great 95 percent accuracy K Nearest Neighbor calculated with 100 percent accuracy. This makes KNN the best model.

Confusion Matrix

```
In [41]: knn_cv.score(X_test, Y_test)
Out[41]: 1.0
```

We can plot the confusion matrix

```
In [42]: yhat = knn_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```



- This model accurately predicted all of the correct outcomes, making it the best training method

Conclusions

Overall thanks to the information gathered throughout this assignment we can gather the following:

- When it comes to saving money on a launch one of the biggest cost savers is being able to reuse the first stage of a rocket. With the falcon 9, ft booster model being the most likely to be recovered successfully
- We can also tell by the information given to use on the interactive maps, the location KSC LC-39a is going to be the best location to launch from with the highest success rate.
- Thanks to machine learning we can now test different variable without launching an actual rocket to better the odds of success. Such as changing the payload or the distance into orbit!
- As for Recovering the first stage, since the launch site is near the coastline, an ocean remote drone ship may provide the best method of recovery.

Appendix

- Git hub link to all code used in project:
- https://github.com/dannybravo599/capstone-project/tree/main

Thank you!