IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rebecca Smith
February 2, 2024

Outline

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

Executive Summary

Summary of Methodologies

- Data Collection
- Data Wrangling
- EDA with data visulization
- EDA with SQL
- Building Interactive maps with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis - Classification

Summary of Results

- EDA results
- Interactive results
- Predictive analysis results

Introduction

- Background and Context:
  - SpaceX advertises Falcon 9 rocket launches on its website, they are stating that their costs is much cheaper than other providers and the reason for this is that they reuse the first stage booster
- Problem to be answered:
  - Accurately predict if the first stage booster of the Falcon 9 lands successfully

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected from a SpaceX Rest API
- Perform data wrangling
  - The data was analyzed and cleaned to make sure the dataset was clear of null values and irrelevant columns for the research at hand
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Some test include:
    - Logistic Regression
    - Support Vector Machines
    - Decision Tree Classifier
    - K-Nearest Neighbors

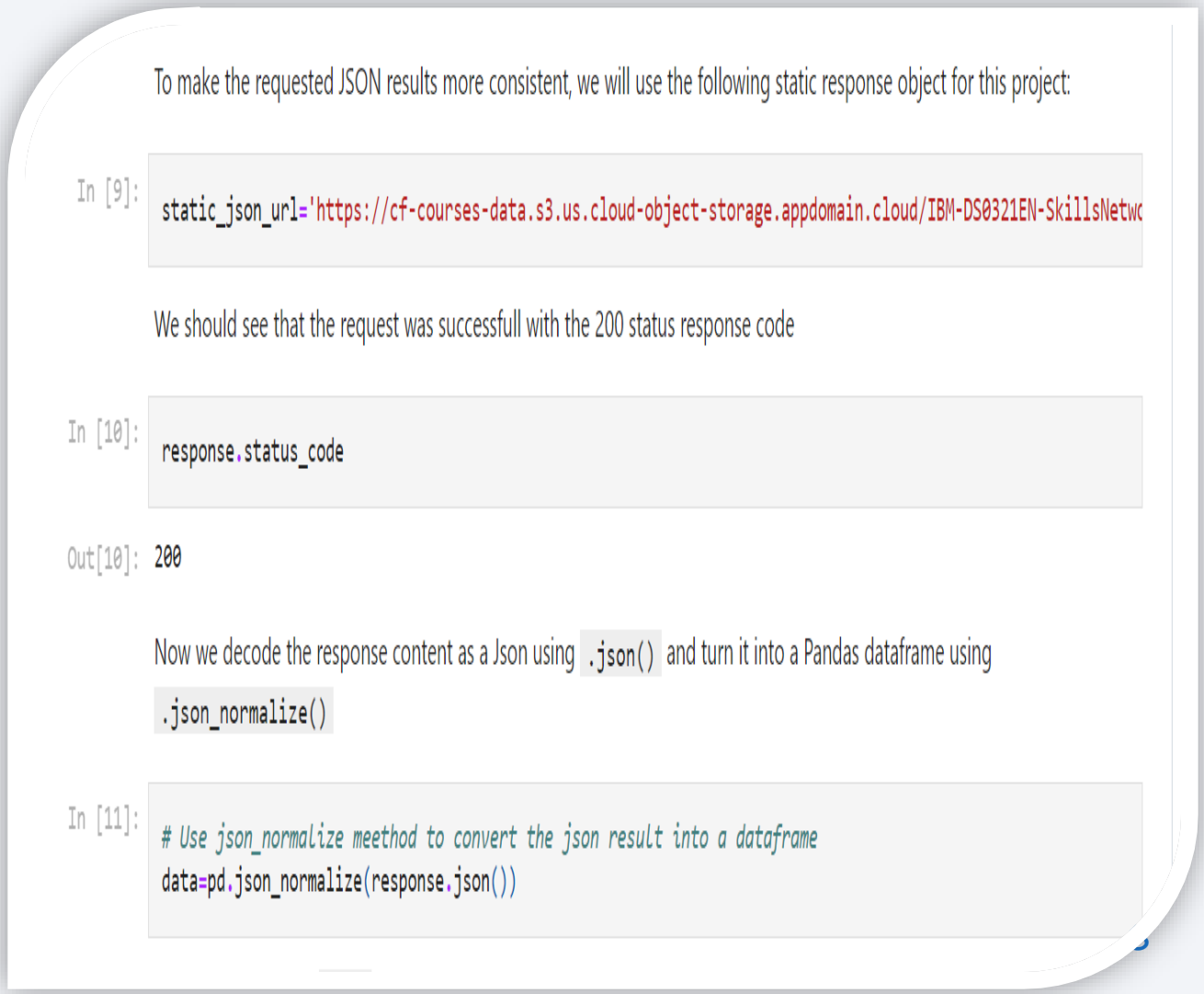
Data Collection

- The data was collected using the following methods:
  - Data was collected using get request to the SpaceX API
  - Following that we received the data as a .json() function and we turned it into a dataframe by normalizing the data using .json_normalize()
  - The data was then cleaned and removed the missing values.
  - Falcon 9 launch data was also collected using webscraping on Wikipedia

Data Collection – SpaceX API

- Data was collected using an API and then cleaned and formatted to be able to be used easier.

- GitHub
  URL: https://github.com/mathmanatee16/Applied-Data-Science-Capstone/blob/main/Data%20Collection%20API.ipynb



To make the requested JSON results more consistent, we will use the following static response object for this project:

In [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwo
```

We should see that the request was successfull with the 200 status response code

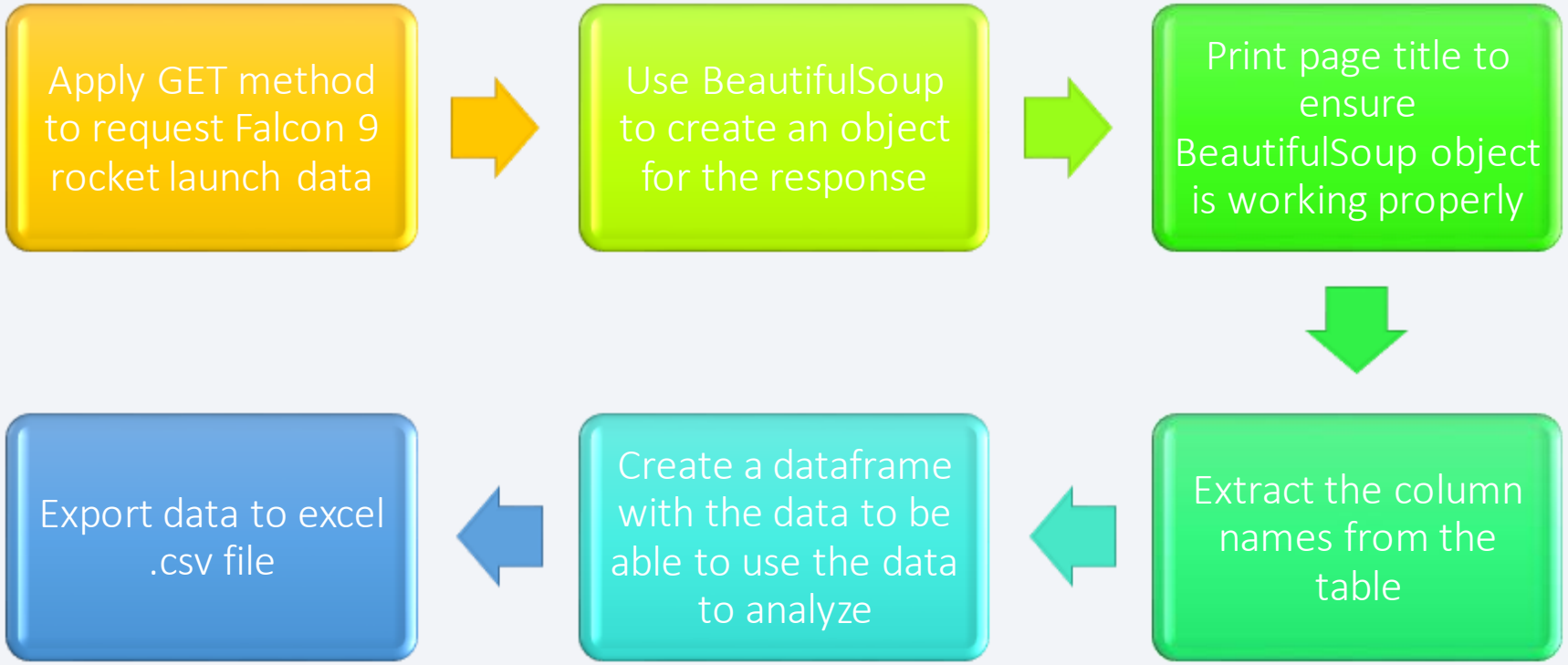
In [10]:
```
response.status_code
```

Out[10]: 200

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

In [11]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data=pd.json_normalize(response.json())
```

Data Collection - Scraping

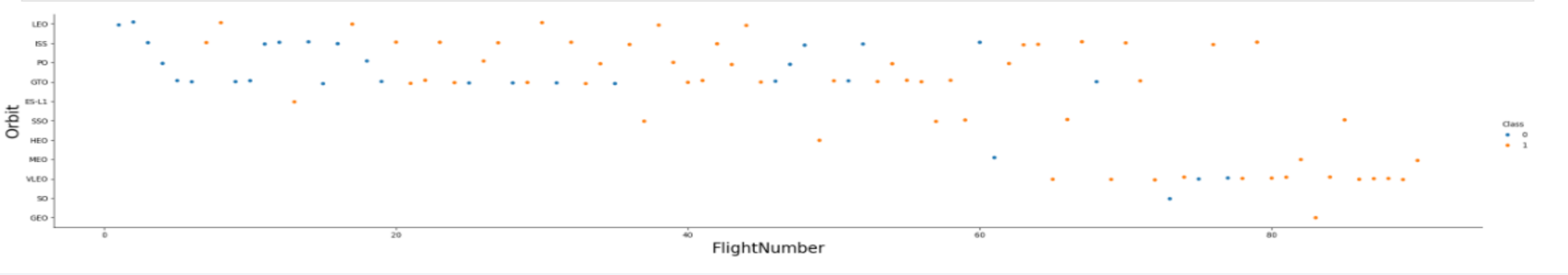


- GitHub URL: https://github.com/mathmanatee16/Applied-Data-Science-Capstone/blob/main/Webscraping.ipynb

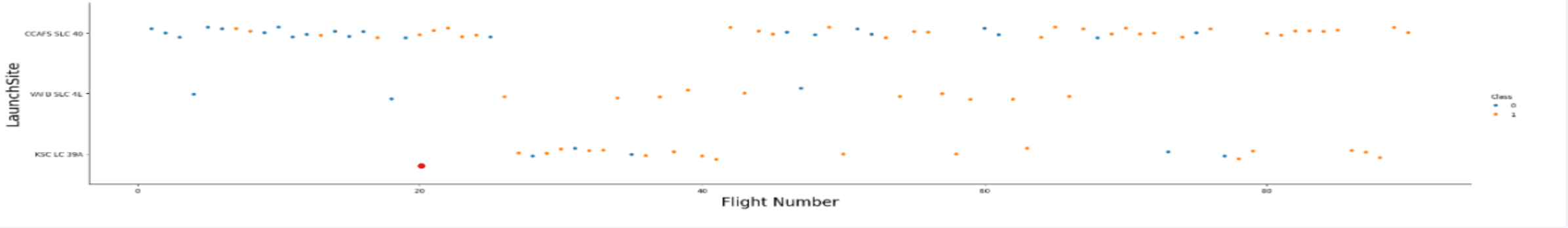
Data Wrangling

- Exploratory analysis of the data
  - This was used to be able to create training labels
  - Also, a calculation was done to find the number of launches that happened at each site
  - A landing outcome label was created and added to a column for the outcome and then the results were placed into a .csv file
- GitHub URL: https://github.com/mathmanatee16/Applied-Data-Science-Capstone/blob/main/Data%20Wrangling.ipynb

EDA with Data Visualization



- We used many different plots to see the relationship of the data
- GitHub URL: https://github.com/mathmanatee16/Applied-Data-Science-Capstone/blob/main/EDA%20with%20Data%20Visualization.ipynb.jupyterlite.ipynb



EDA with SQL

- SQL Queries Performed:
  - Unique launch sites in the data
  - Total payload mass carried by boosters launched by Nasa
  - Average payload mass carried by booster version F9 v1.1
  - Total number of successful and failed missions
  - Failed landing outcomes in drone ship, their booster version and launch site
- GitHub URL: https://github.com/mathmanatee16/Applied-Data-Science-Capstone/blob/main/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Folium Map
  - Marked Launch Sites
  - Added markers and circles and lines to show success or failures of launches
  - Launch outcomes assigned 0 or 1
    - 0 – Failure
    - 1 – Success
  - By marking the successes and failures at each launch site it helps to better and more quickly identify what happened at each site
- GitHub URL: https://github.com/mathmanatee16/Applied-Data-Science-Capstone/blob/main/Interactive%20Map%20with%20Folium.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Interactive dashboards were created by using Plotly Dash
- Pie charts were used to show the total number of launches at different sites
- Scatter plots were used to show the relationship between the outcome and the payload mass for each of the different booster versions

Predictive Analysis (Classification)

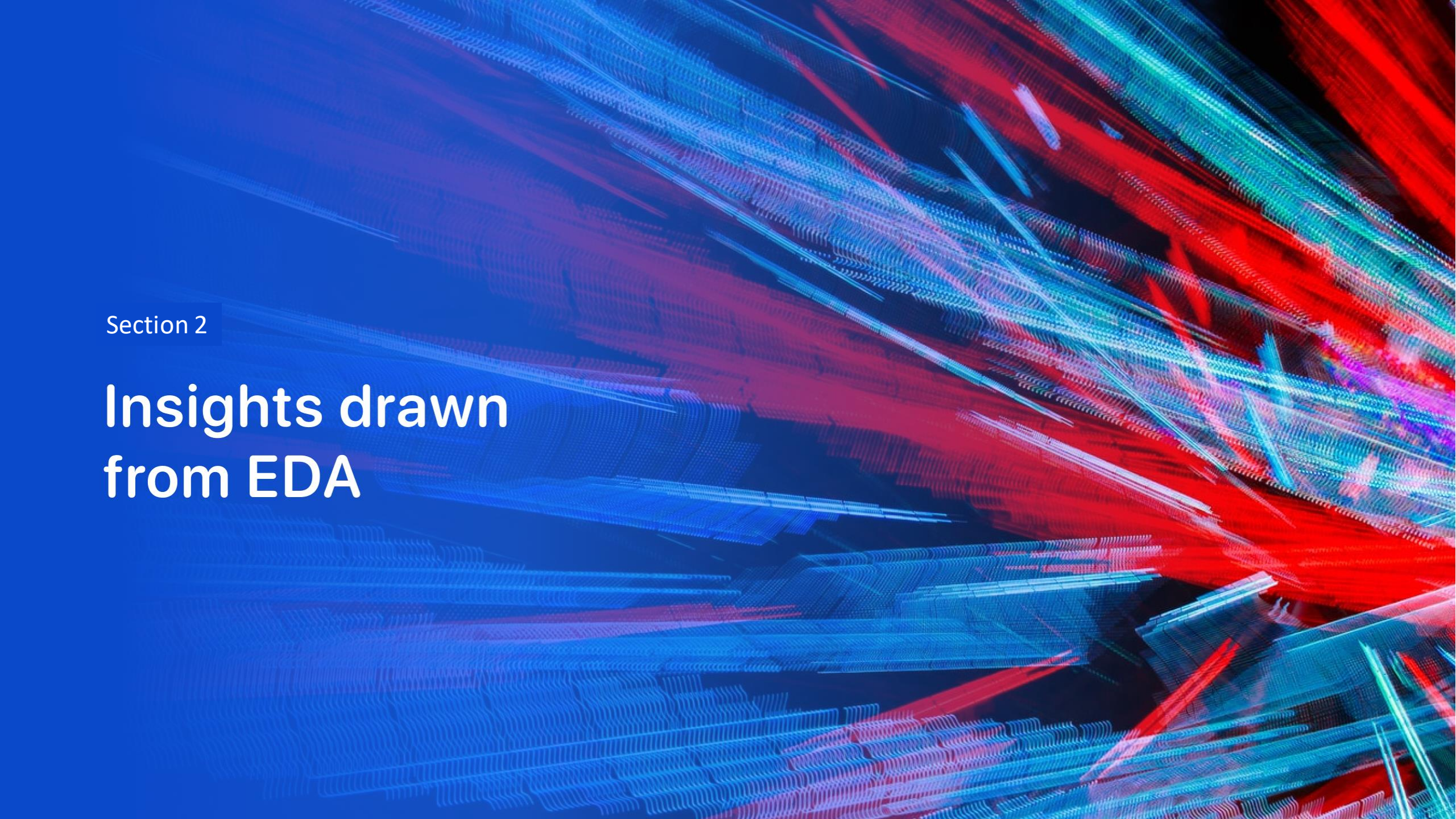
- Step 1: Load in the data using numpy and pandas
- Step 2: Transform the data
- Step 3: Split the data into training data and testing data using train_test_split()
- Step 4: Use different machine learning approaches and tune the hyperparameters to find the model that gives us the most accurate result when using the test data. While doing this we used GridSearchCV to visualized the confusion matrix on how the data is being represented.
- Step 5: The model that has the best accuracy is then the best performing model and that is the one that is going to be used.
- GitHub URL: https://github.com/mathmanatee16/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine_Learning_Prediction.jupyterlite.ipynb

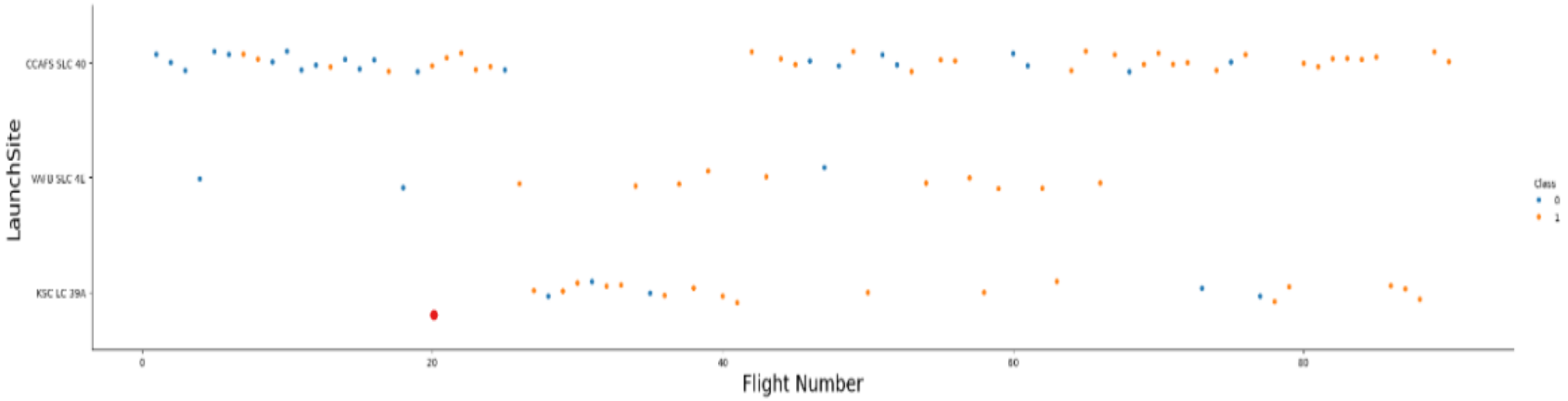
Results

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

Section 2

Insights drawn from EDA



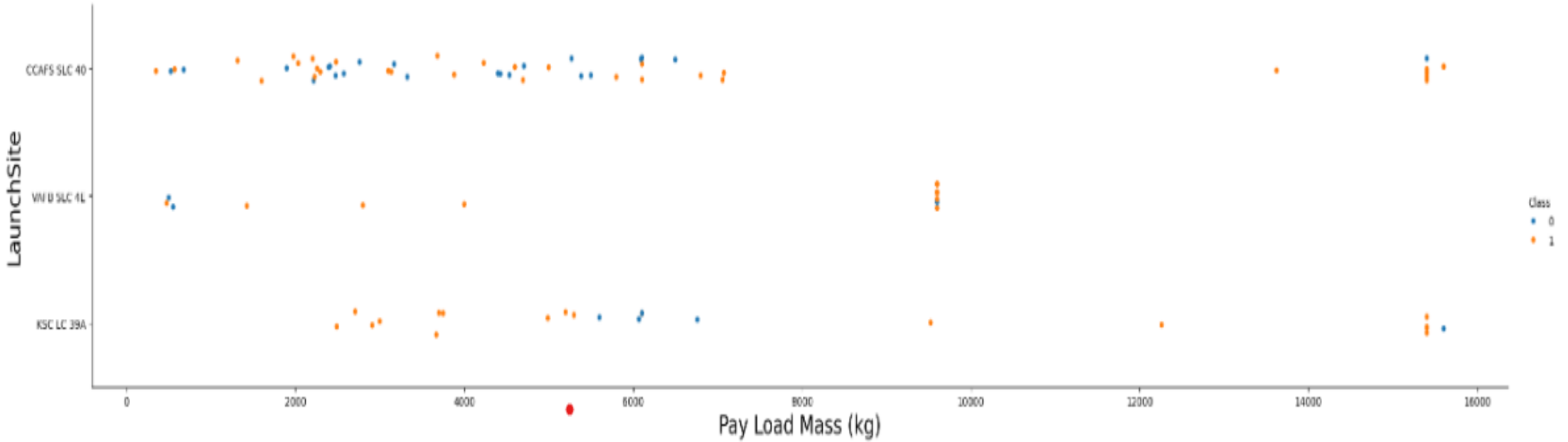


Flight Number vs. Launch Site

- From this plot we can see that the more flights happening at a launch site the better the success rate is.

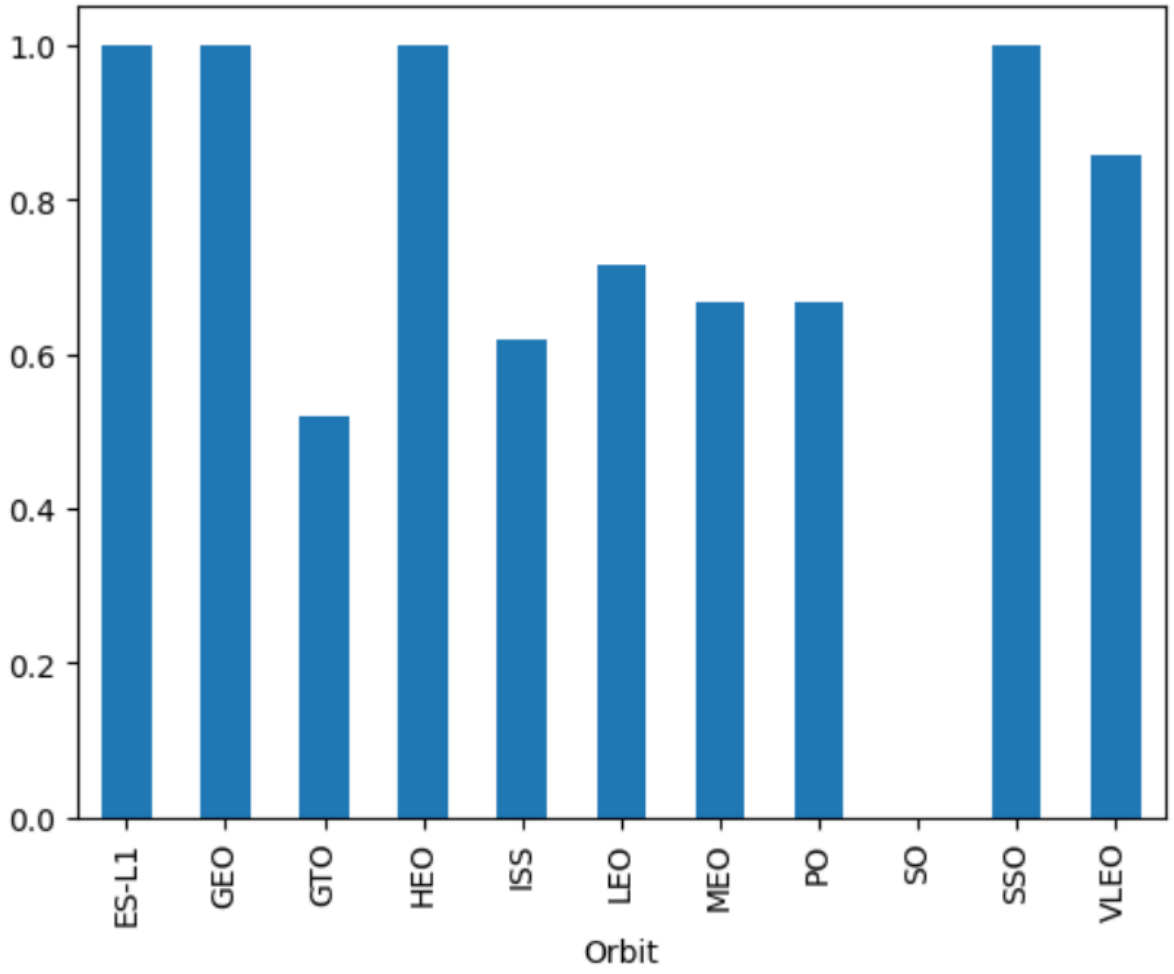
Payload vs. Launch Site

- We can see from this plot that the launch site of CCAFS SLC 40 has more flights and is having more successful launches.



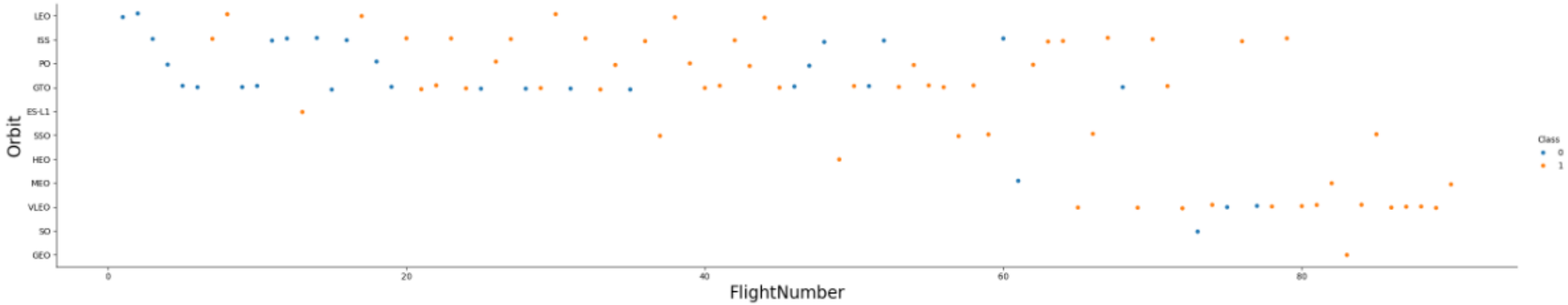
Success Rate vs. Orbit Type

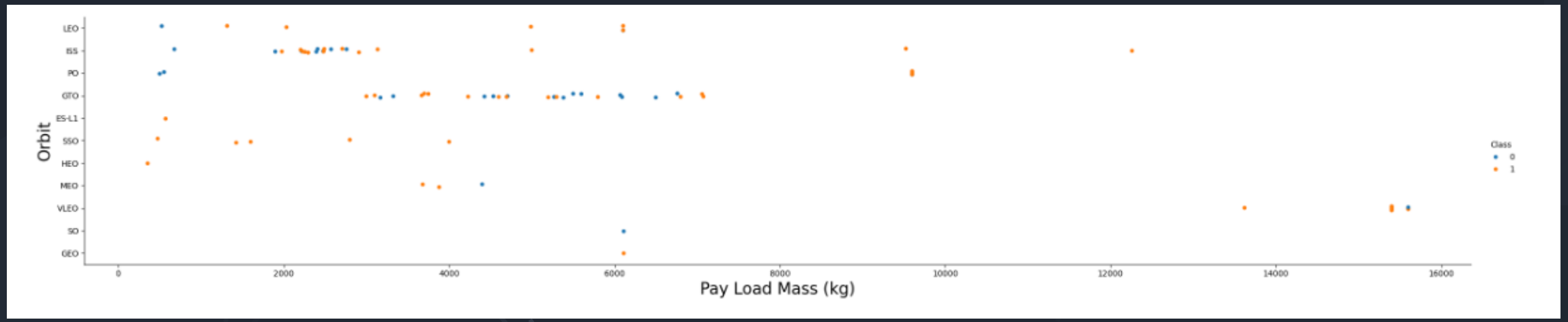
- ES-L1, GEO, HEO, and SSO have the highest success rate
- While SO has the highest failure rate



Flight Number vs. Orbit Type

- From this plot we can see the relationship between Orbit type and Flight number.
- This helps us to better see if a specific orbit type is getting better over time based on the flight number



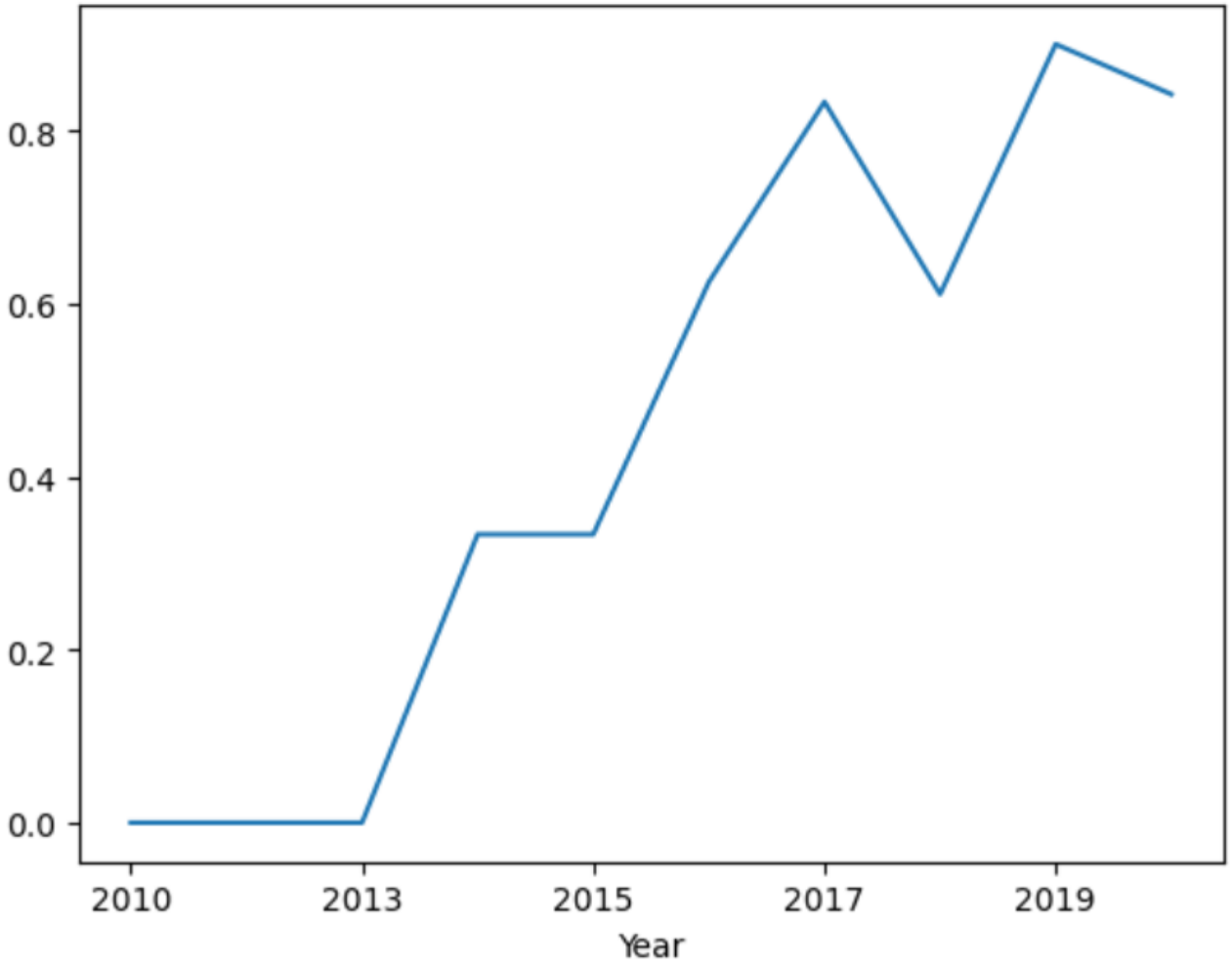


Payload vs. Orbit Type

- We can see the spread of payload mass and from this we can see that the larger payload mass is really just being used in the LEO and ISS orbit.

Launch Success Yearly Trend

- From this line chart we can see that over time the launch success has been getting increasingly better. One thing to note is that in 2018 there was a dip which tells me something new might have been tried and it did not work according to plan and you can see that the successes are growing again so that problem has been fixed



All Launch Site Names

- This query is pulling in each of the different launch sites by using DISTINCT which means it is only going to pull in new launch sites and no duplicates

```
[8]: %sql SELECT DISTINCT(launch_site) FROM SPACEXTBL;
 * sqlite:///my_data1.db
Done.
```

[8]:

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

Launch Site Names Being with 'CCA'

- This is showing where we are limiting the number of records that come back and they need to have 'CCA'

```
[10]: %sql SELECT * FROM SPACEXTBL WHERE launch_site LIKE 'CCA%' LIMIT 5;
 * sqlite:///my_data1.db
Done.
```

[10]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- We are able to calculate the total Payload Mass by SUM all of the Payload Masses and then we use the WHERE clause to say we only want to sum the payload masses for the customers that meet a specific name

```
[12]: %sql SELECT SUM(payload_mass__kg_) AS Total_Payload_Mass FROM SPACEXTBL WHERE customer='NASA (CRS)';
 * sqlite:///my_data1.db
Done.
```

[12]:

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- We use the AVG function to calculate the average
- We use the WHERE function to limit the selection to something specific

```
[13]: %sql SELECT AVG(payload_mass__kg_) AS Avg_Payload_Mass FROM SPACEXTBL WHERE booster_version = 'F9 v1.1';
 * sqlite:///my_data1.db
Done.
```

[13]:

Avg_Payload_Mass
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- We use MIN to pull the earliest date
- We use WHERE to choose the specific landing outcome

```
[14]: %sql SELECT MIN(DATE) AS First_Successful_Landing FROM SPACEXTBL WHERE landing_outcome = 'Success (ground pad
 * sqlite:///my_data1.db
Done.
```

[14]:

First_Successful_Landing
2015-12-22

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
- WHERE to pull in successful landed on drone ship
- BETWEEN to pull masses greater than 4000 and less than 6000

```
[16]: %sql SELECT booster_version, payload_mass__kg_, landing_outcome FROM SPACEXTBL \
          WHERE landing_outcome='Success (drone ship)' AND (payload_mass__kg_ BETWEEN 4000 AND 6000) ;
 * sqlite:///my_data1.db
Done.
```

[16]:

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

- Calculate the total number of successful and failure mission outcomes

```
[17]: %sql SELECT mission_outcome, COUNT(mission_outcome) AS Total FROM SPACEXTBL GROUP BY mission_outcome;
 * sqlite:///my_data1.db
Done.
```

[17]:

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
[18]: SELECT DISTINCT(booster_version), (SELECT MAX(payload_mass__kg_) AS "Maximum_Payload_Mass" FROM SPACEXTBL)
```

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

[18]:

Booster_Version	(SELECT MAX(payload_mass__kg_) AS "Maximum_Payload_Mass" FROM SPACEXTBL)
F9 v1.0 B0003	15600
F9 v1.0 B0004	15600
F9 v1.0 B0005	15600
F9 v1.0 B0006	15600
F9 v1.0 B0007	15600
F9 v1.1 B1003	15600

2015 Launch Records

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

```
[21]: %sql SELECT landing_outcome, booster_version, launch_site, DATE FROM SPACEXTBL\
      WHERE landing_outcome LIKE '%Failure (drone ship)%' AND substr(Date,0,5)='2015';
```

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

[21]:

Landing_Outcome	Booster_Version	Launch_Site	Date
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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.
- The query uses 4 key elements and they are COUNT, WHERE, BETWEEN, and GROUP BY.

```
[27]: %sql SELECT landing_outcome, COUNT(landing_outcome) AS 'Total' FROM SPACEXTBL\
      WHERE (DATE BETWEEN '2010-06-04' AND '2017-03-20') GROUP BY (landing_outcome);
 * sqlite:///my_data1.db
Done.
```

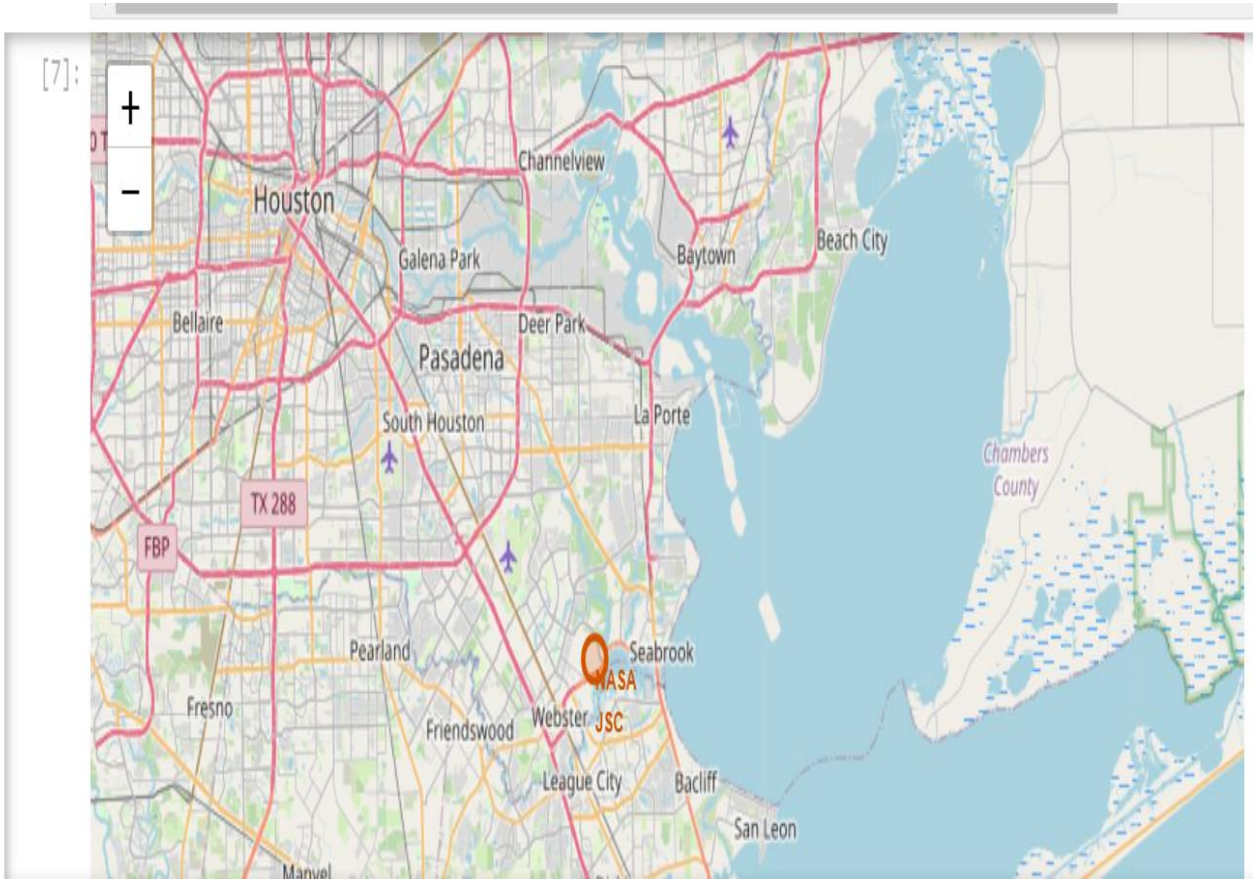
[27]:

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


Section 3
Launch Sites
Proximities Analysis

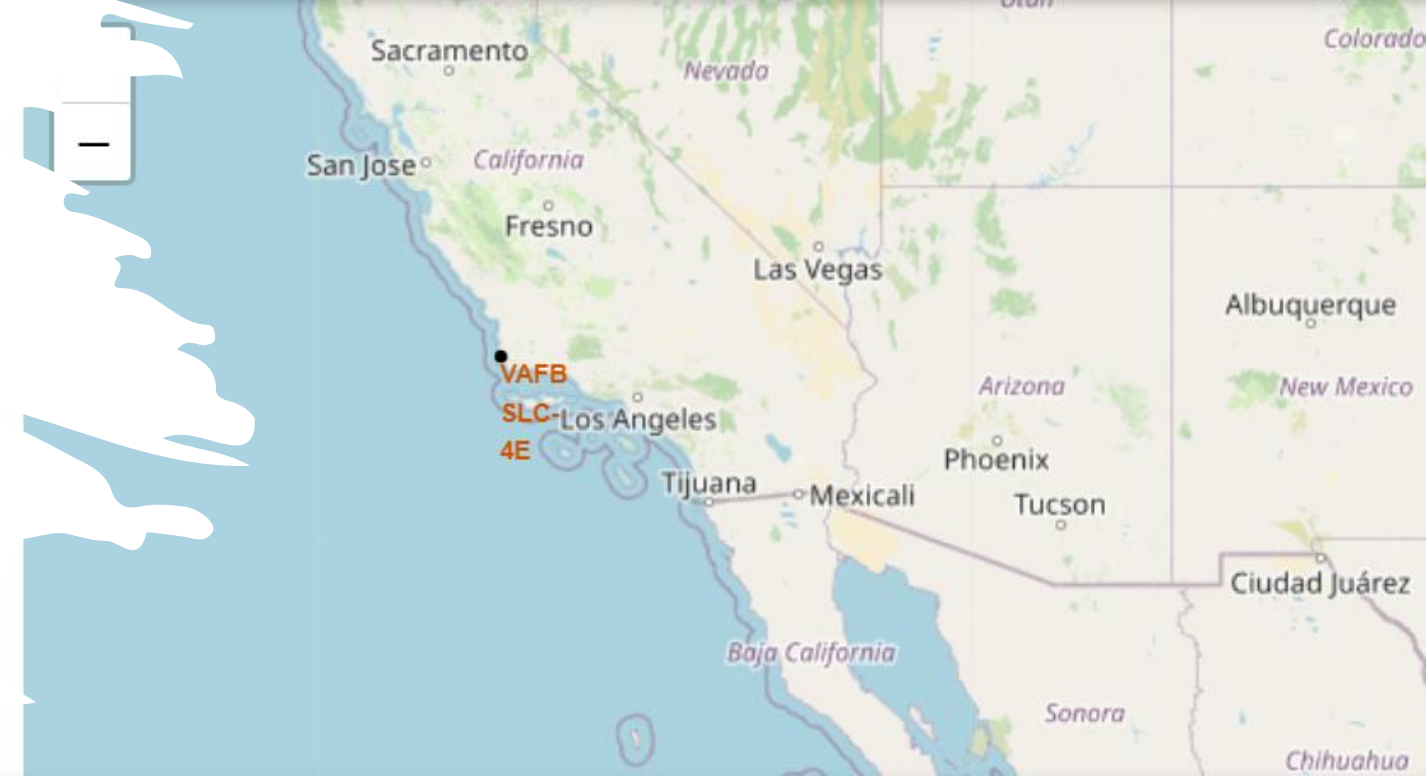
Nasa Johnson Space Center

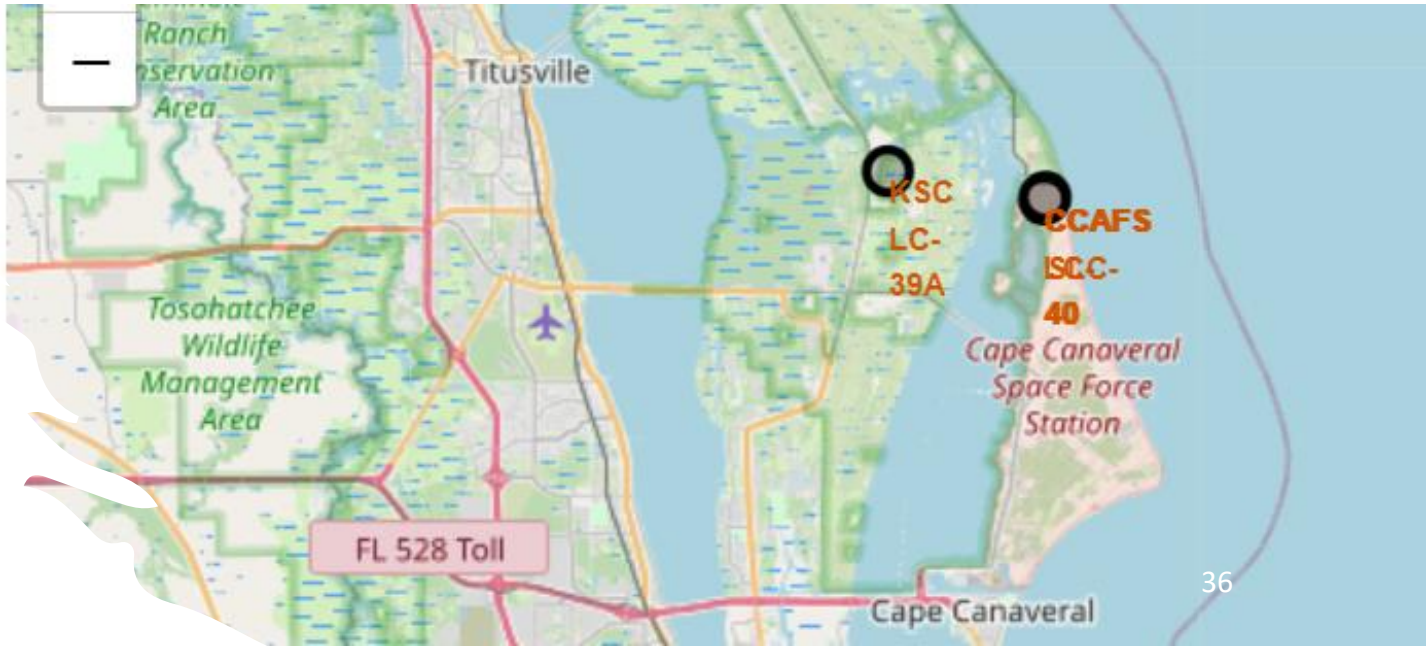
This map is showing the location with a marker of Nasa's Johnson Space Center



Launch Site Locations

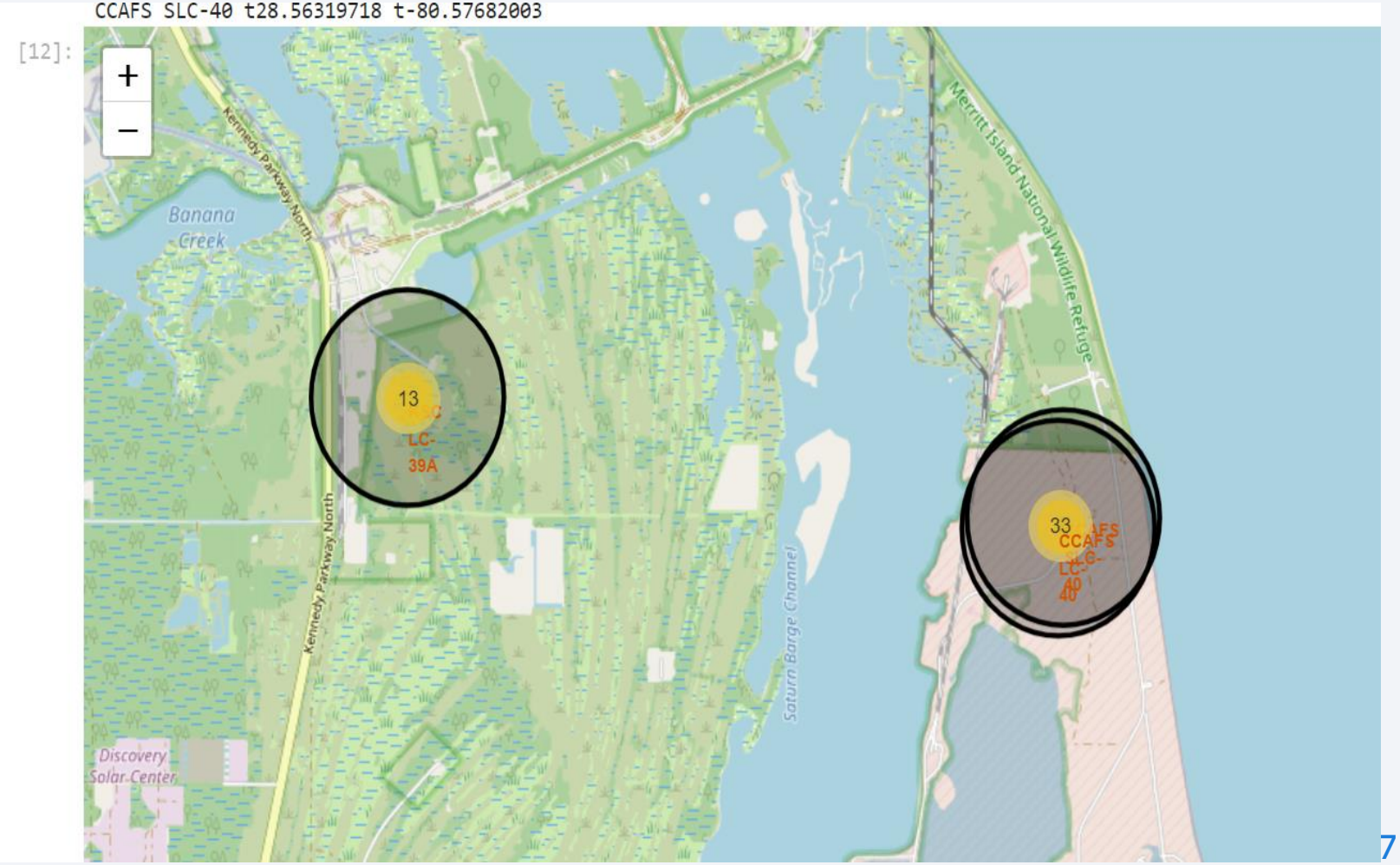
- This map is showing where the different launch sites from the data set are located.





Launch Numbers by Launch Site

This map is showing the number of launches at each of the different launch sites



Section 5

Predictive Analysis (Classification)

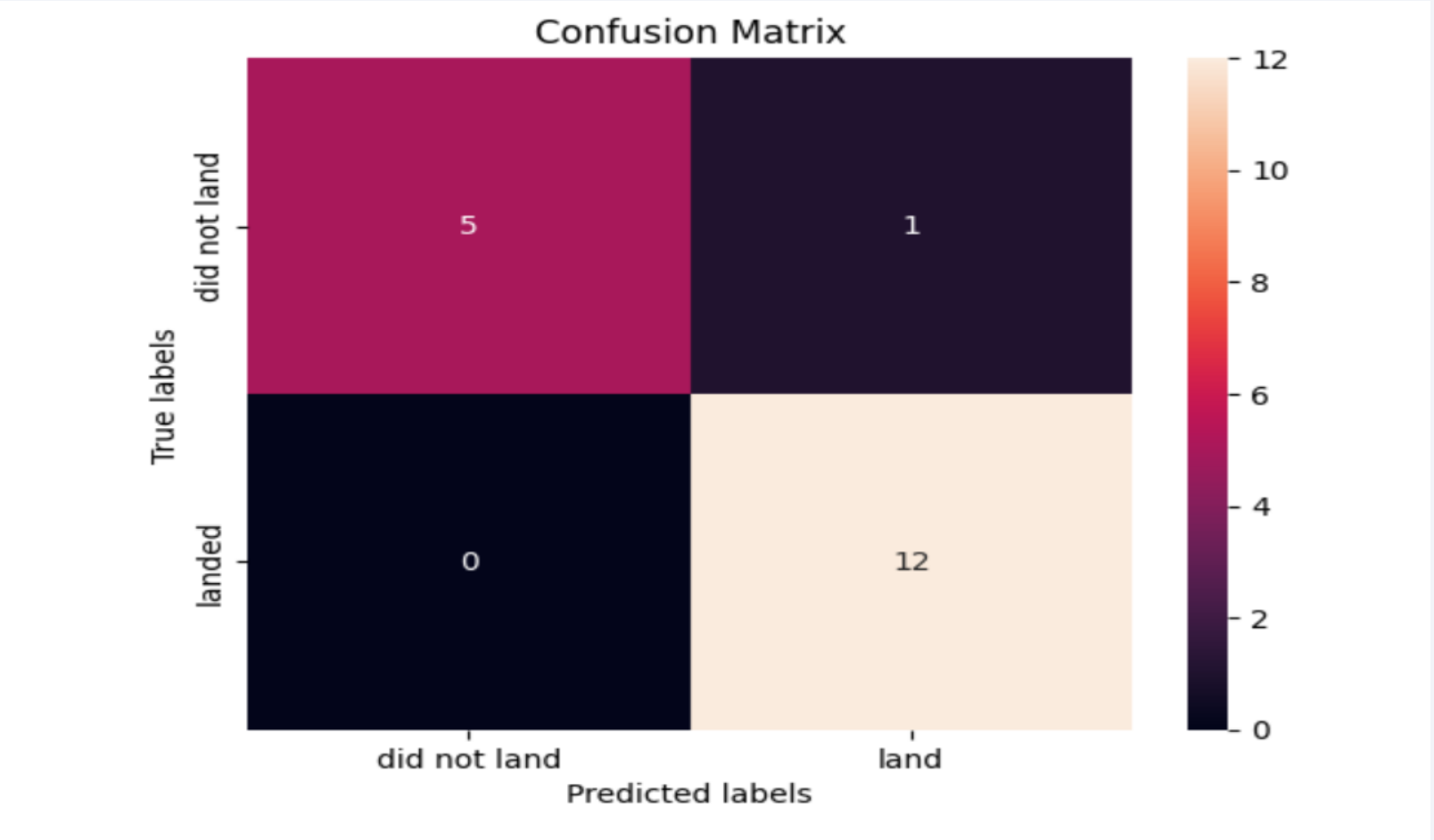
Classification Accuracy

```
Model Accuracy TestAccuracy
LogReg t t0.84643 t t0.83333
SVM t t0.84821 t t0.83333
Tree t t0.88571 t t0.94444
KNN t t0.84821 t t0.83333
```

- The Tree model has the best accuracy and test accuracy out of the models tested

Confusion Matrix

- This confusion matrix is for the tree. This is the most accurate because we have minimized the amount of false positives



Conclusions

- Each model performs slightly different than the one before it
- The Tree does the best at accurately predicting the outcome of whether the land will be successful
- While 88% is a pretty good accuracy there is still room to tune hyperparameters to try to get a better result

Appendix

- GitHub
  URL: https://github.com/mathmanatee16/Applied-Data-Science-Capstone/tree/main

Thank you!