IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ben Cottrell
September 24th 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Collection: the SpaceX launch was collected and scraped from the SpaceX API and Wiki
  - Data was wrangled, cleaned and then stored within the Python library Pandas.
  - Exploratory data analysis (EDA) and visualizations executed with SQL queries, Pandas graphs, Seaborn graphs, data manipulations, mapping using Folium and using Dash to build a dashboard using graphs generated with Plotly Express
- Summary of all results
  - Exploratory data analysis showed a vast array of insights about mission success. Most importantly that success increased with continued test flights, launch site KSC LC-39A was the most consistent site for successful missions. Launch site VAFB SLC-4E is the only site with successful max payloads and the orbit does not impact the successful recovery of the first stage
    - The launch site KSC LC39A had no successful landings where the payload was over 5300kg
  - Predictive analysis revealed that most algorithms performed similarly in determining whether a launch was successful in landing at its first stage.
    - The Decision Tree algorithm achieved 83.3% accuracy on test data and a top accuracy of 87.32%

Introduction

- We as prospective company Space Y are looking to understand our rival competitor SpaceX
  - They are able to save millions of dollars compared to other competitors which is due to their rocket's first stage recovery ability
  - We must build a mission capable rocket system in order to beat the SpaceX price point
    - Using their data we need to circumvent years of research, development and testing
- By answering the following questions, we can then begin our development of a cheap, high capacity rocket that can take on SpaceX and reduce their control of the market
- Questions to be answered:
  - What factors affect the successful recovery of the rocket's first stage?
  - Can we predict whether a mission will be successful?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

As referenced in the executive summary, the data was collected via two methods:

SpaceX API

1. Identify API calls to collect the raw and get the URL from https.//api.spacexdata.com
2. Import requests, json and Pandas libraries
3. Using the requests library, use the get() method to request the raw data in json format
4. From the json library, load the request data using the loads() method
5. In Pandas, use the json_normalize() method to load the json file into a Pandas dataframe
6. If necessary, repeat steps 3-5 if the dataframe is too large in order to get specific subsets of the data

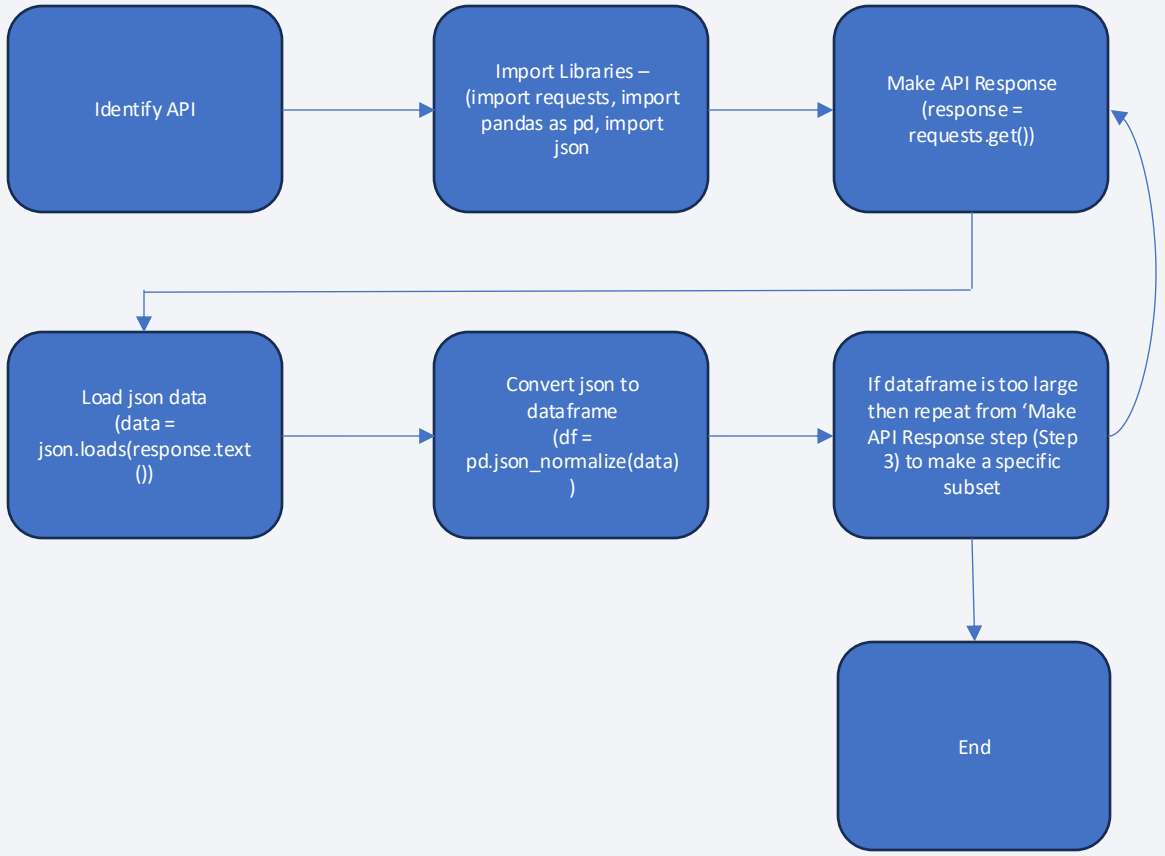
Web Scraping

1. Identify dataset on the internet and get URL from https://en.Wikipedia.org
2. Import requests, BeautifulSoup and Pandas libraries
3. Using the requests library, use the get() method on the URL
4. Store the BeautifulSoup parser built with the requested data
5. Using the find_all('table') method we isolate the tables into a list and then select the desired data with slicing
6. Use the find_all('th') method to gather the column names and then iterate through gathering all information for 'tr' and 'td', the rows and elements. These were then stored in a dictionary
7. All the gathered data was then stored in a Pandas dataframe

Data Collection – SpaceX API

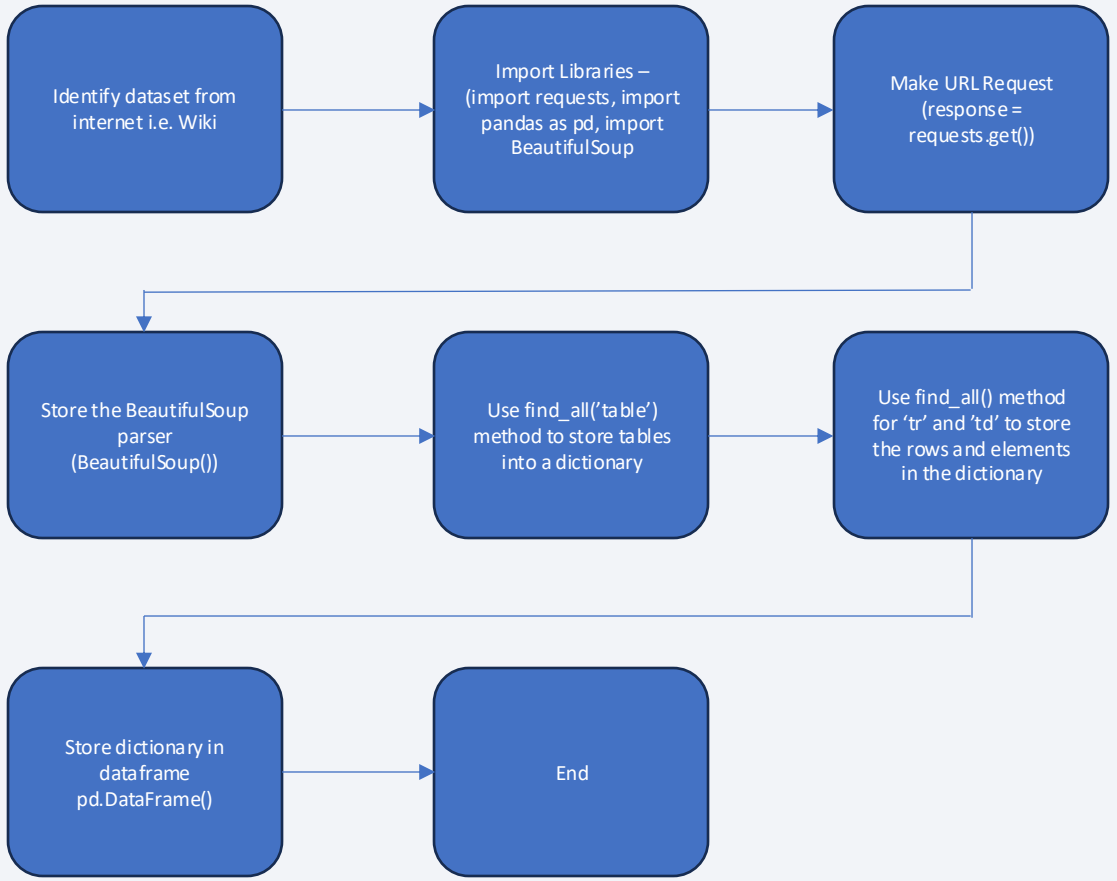
- API calling process to acquire the dataset from the SpaceX REST API

- https://github.com/bcottrell93/DataScienceCapstoneProject/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



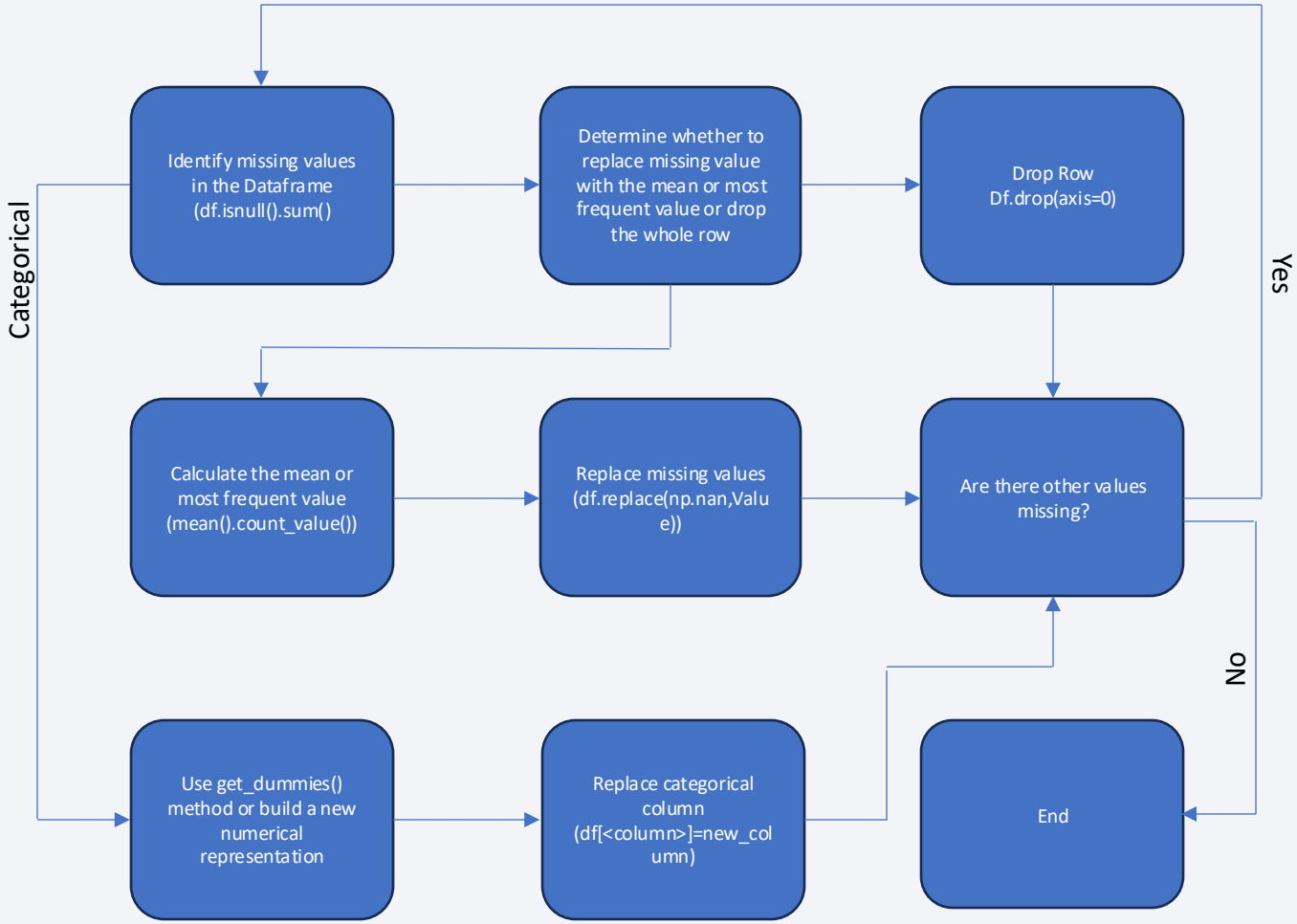
Data Collection - Scraping

- Web scraping process to acquire the dataset from the SpaceX Wikipedia page
- https://github.com/bcottrell93/DataScienceCapstoneProject/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Data wrangling is the process of creating a dataset that is effective and straightforward to query and manipulate for exploratory data analysis
- This process includes the replacement of empty/null values and categorical values with numerical values, making the process of EDA much simpler
- https://github.com/bcottrell93/DataScienceCapstoneProject/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- The graphs were chosen to better understand the Spacex rocket launch success as affected by different factors, including with respect to time, which would reflect their continued research and development
    - Flight Number vs Launch Site | Payload Mass | Orbit
    - Payload Mass vs Launch Site | Orbit
    - Orbit vs Success bar chart
    - Success Rate vs Years
- https://github.com/bcottrell93/DataScienceCapstoneProject/blob/main/edadataviz.ipynb

EDA with SQL

- SQL queries used in the form of: SELECT VALUES FROM DATABASE WHERE CONDITIONS
  - Queries were used to identify successful and failed launch dates, frequency of landings that failed at the first stage, launch locations, payload sizes and boosters
  - Example query: SQL query to get the total number of successful and failed mission outcomes
    - SELECT COUNT(Mission_Outcome), (SELECT COUNT(Mission_Outcome) FROM SPACEXTABLE WHERE Mission_Outcome LIKE "Failure%") AS Failure FROM SPACEXTABLE WHERE Mission_Outcome = "Success";
- https://github.com/bcottrell93/DataScienceCapstoneProject/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

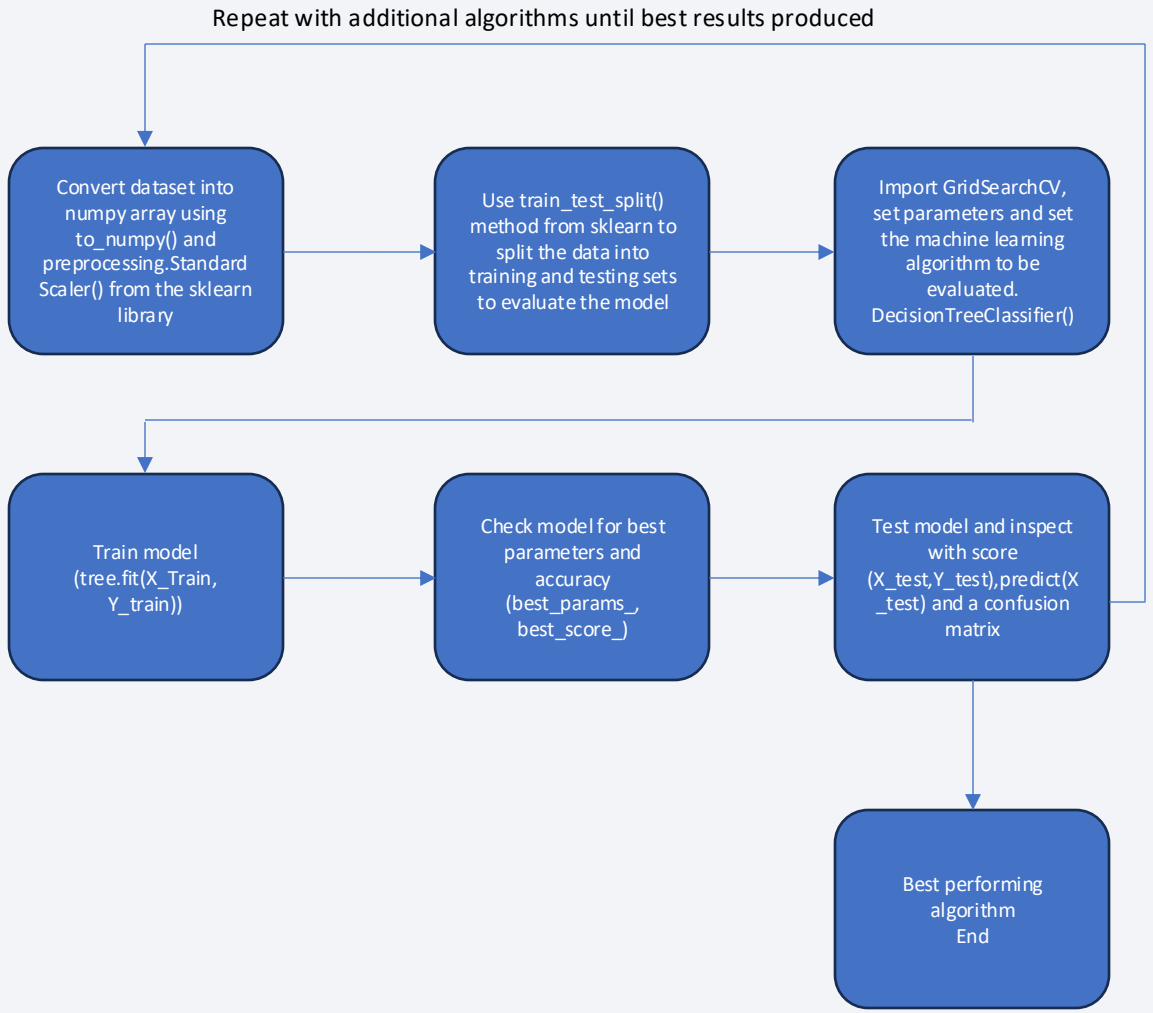
- Created circles around launch sites and marked them on the map
- Marked each site with colored mark for successful and failed launches using a markercluster
- Added mouse position for estimating longitude and latitude
- Drew lines marking the closest ocean, highway, train track and city to the VAFB SLC-4E site
- The objects were added in order to understand the positioning of the SpaceX launch sites and see how these locations might affect the success of rocket landings
- https://github.com/bcottrell93/DataScienceCapstoneProject/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- The dashboard contains two graphs:
  - The first is a pie chart controlled by the user selecting from a dropdown menu that contains the launch sites. The second graph is controlled by the user input of both the launch site dropdown and the payload range via a slider.
  - The pie chart shows success of each site, and the individual pie chart shows the success and failure rate of the selected launch site
  - The scatter plot shows the success and failure of all, or a selected site, with respect to the payload size.
- These plots on the dashboard allow a much more granular level of understanding in regard to how each site and different sized payloads impacted a mission's success
- https://github.com/bcottrell93/DataScienceCapstoneProject/blob/main/spacex_dash_app.py

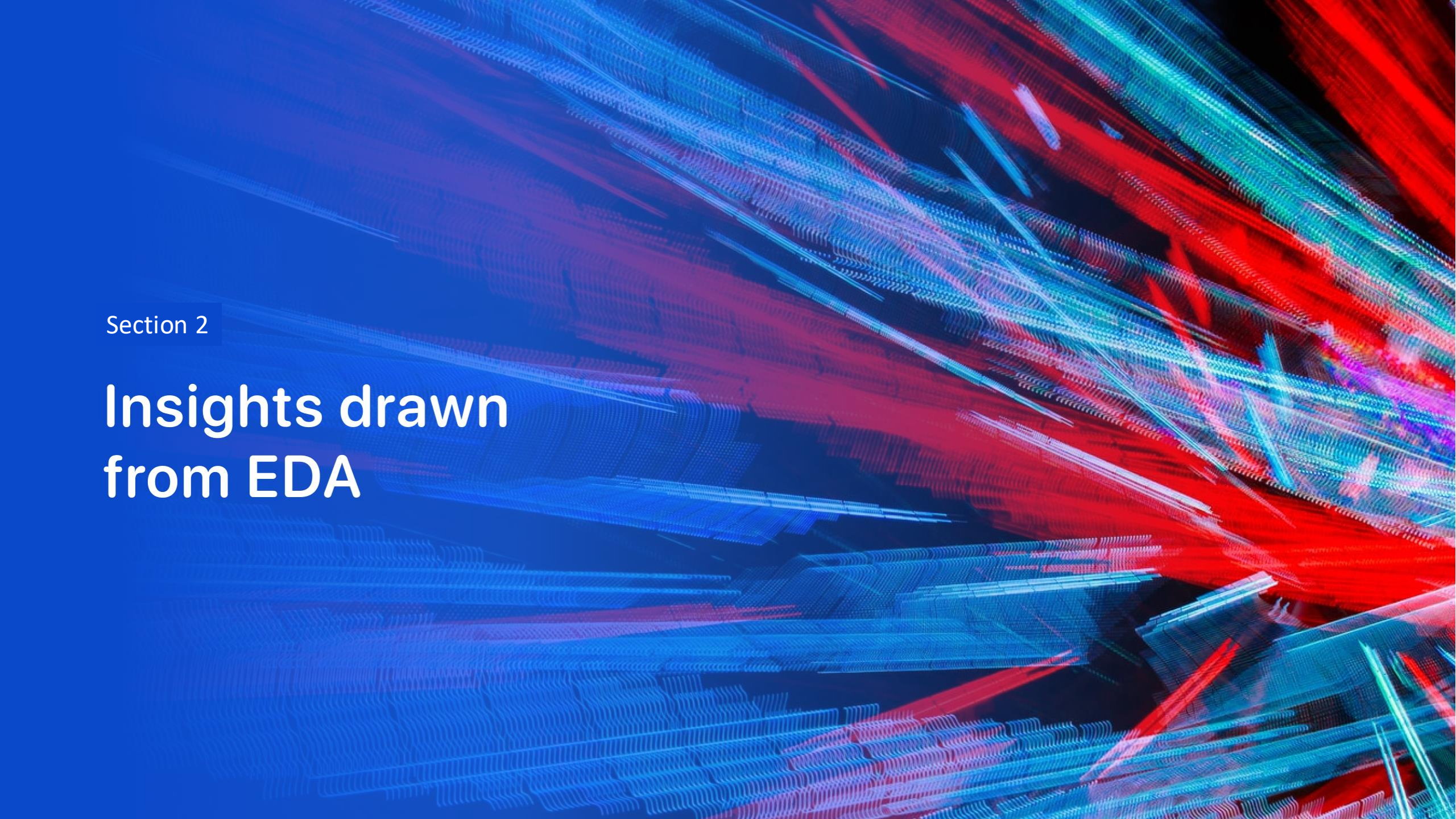
Predictive Analysis (Classification)

- Through iteration the best classification model was developed
- Using GridSearchCV allowed for the training of the models to find the best parameters for the fit
- The accuracy score was used to separate the models and identify the best performing algorithm
- https://github.com/bcottrell93/DataScienceCapstoneProject/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

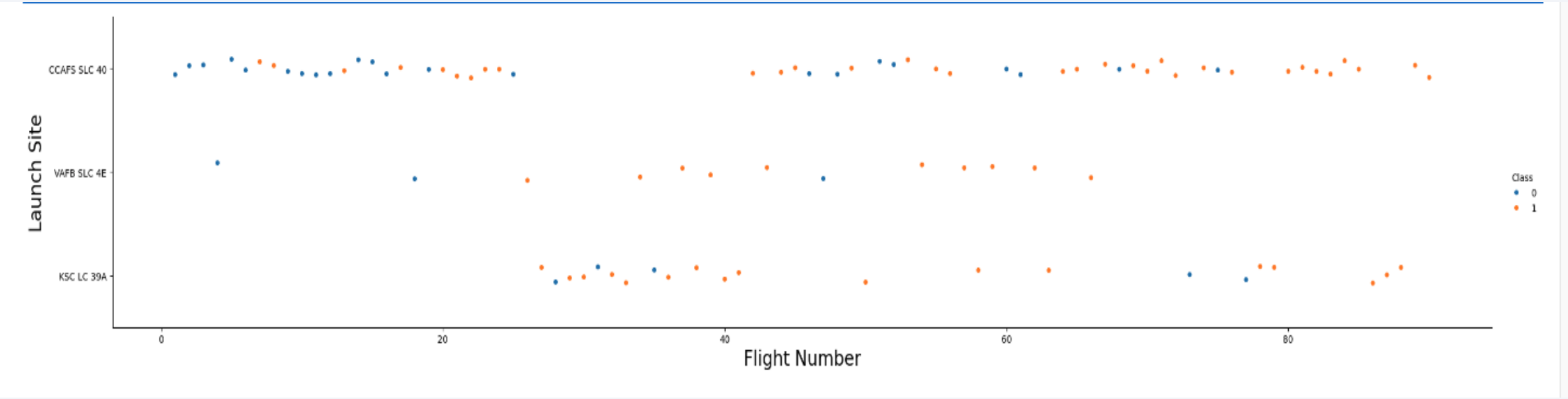


Results

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

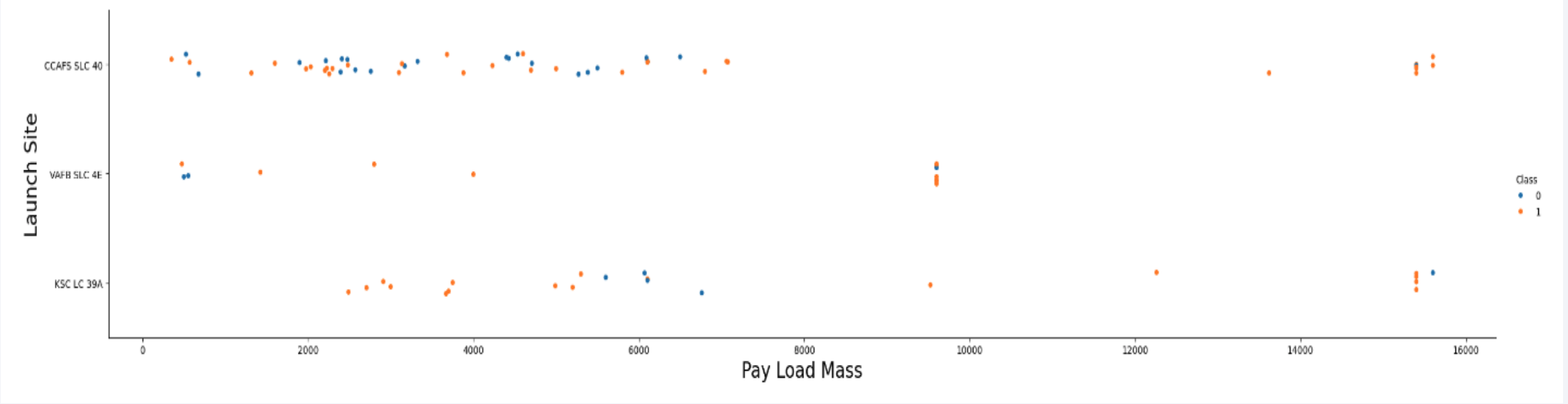

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



There are no major variations between the frequency of success at the different sites. For CCAFS SLC 40 we see a halt of use between launches 25-40 and at the same point we then start to see the KSC LC 39A site be used. As the number of launches increases, we see an increasing trend of successful launches

Payload vs. Launch Site

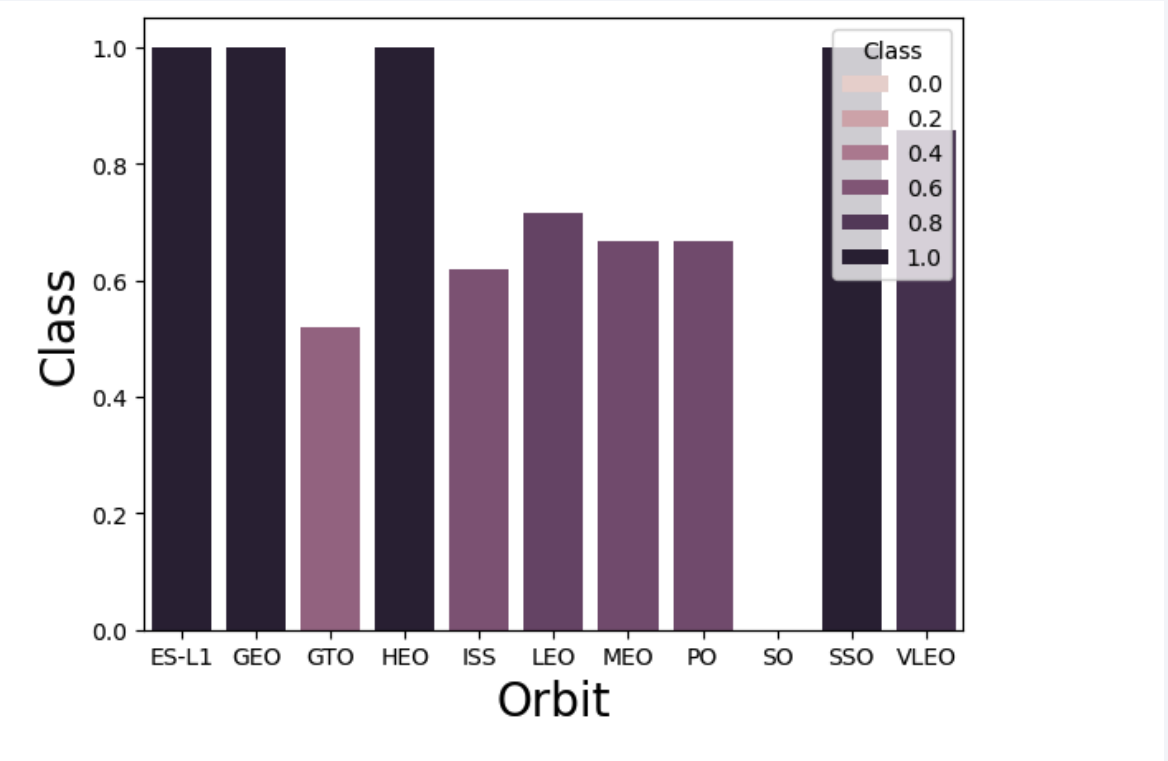


Payload mass is pretty evenly distributed across each of the launch sites, however VAFB SLC 4E does not handle any payloads over 10000kg. The larger payloads do appear to have a high success rate on both CCAFS SLC 40 and KSC LC 39A sites.

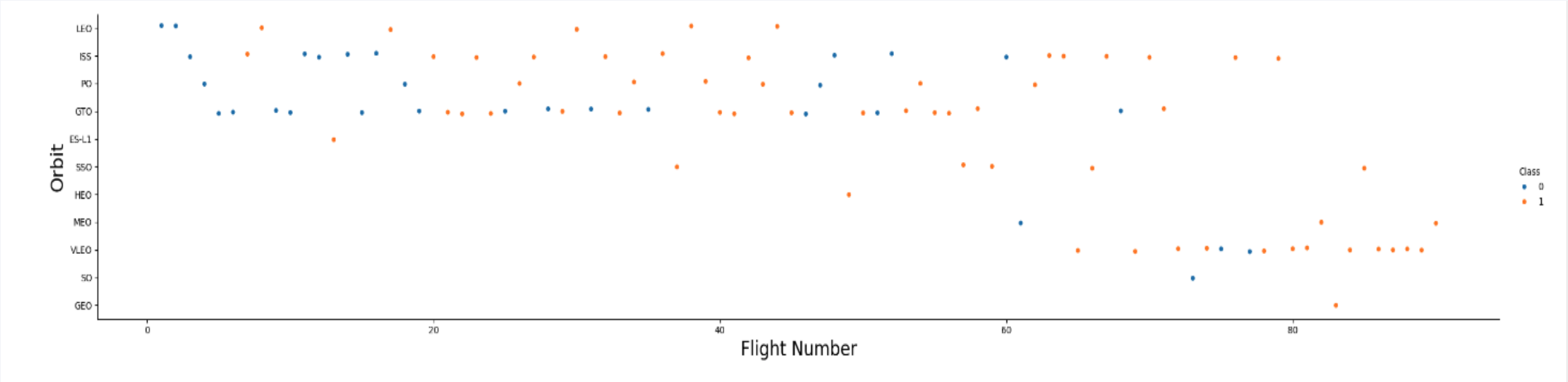
Success Rate vs. Orbit Type

We see a range of success rates in the graph; however, this is a very misleading representation of the data. If we look at ES-L1 we see a 100% success, SO has a 0% success rate and GTO has had 27 missions with a 40-50% success rate.

It is worth noting that VLEO has a 90% success rate from 14 missions which is significant.



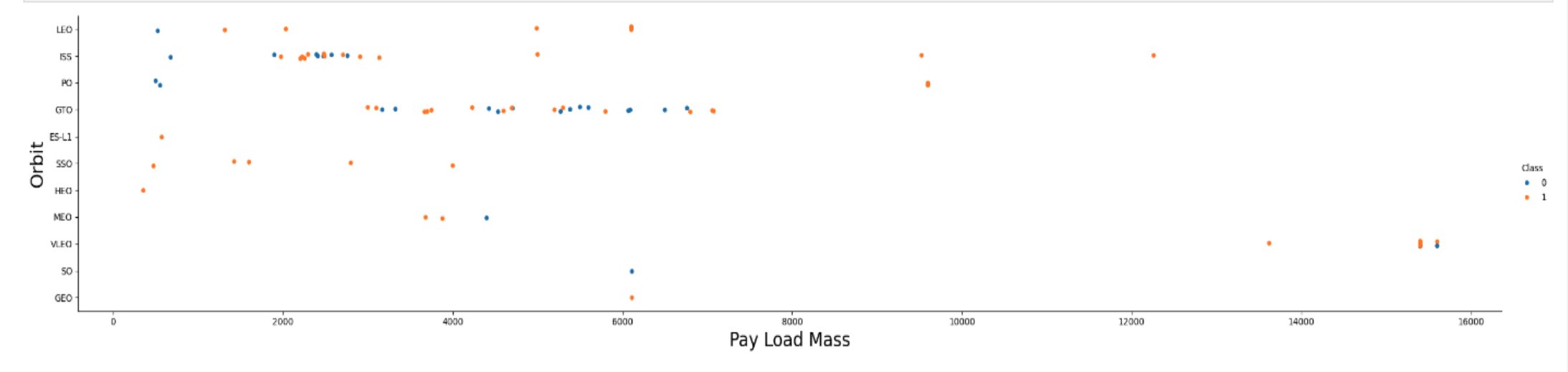
Flight Number vs. Orbit Type



There doesn't appear to be much of a relationship between orbit and flight number or success.

For VLEO, the beginning of launches is after flight number 60 which would indicate it was later in the research. These launches also have a high success rate.

Payload vs. Orbit Type



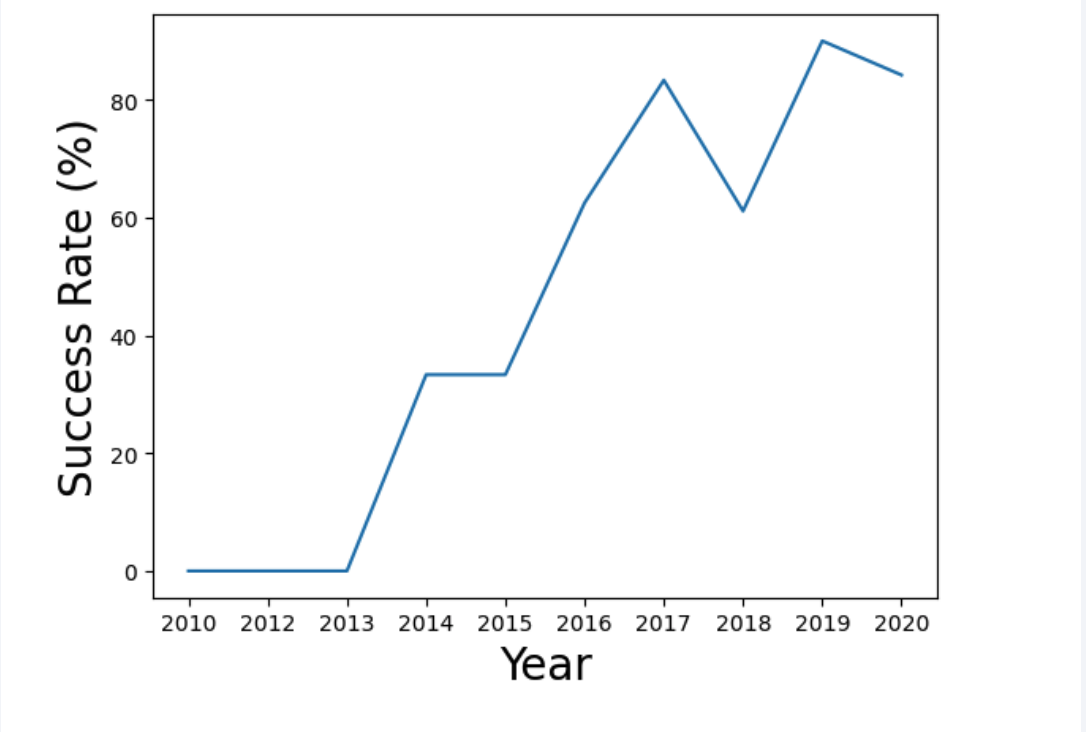
The scatter plot shows that the heavier the payload, the higher the successful landing rate. The only orbits with large payloads are ISS, PO and VLEO.

Success with smaller payloads show a lack of consistency, however it should be noted that SSO orbit types show a 100% success rate

Launch Success Yearly Trend

The graph shows that the success rate has progressively increased over time. This is backed up by the flight numbers.

As research and testing continues the success rate will continue to increase, and currently stands at over 80%



All Launch Site Names

```
[10]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

[10]:

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

Here we use the DISTINCT statement on the LAUNCH_SITE field to show the four 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_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

Using the LIKE statement on the LAUNCH_SITE field, we can find all rows with a launch site beginning with ‘CCA’. A LIMIT function is then used to only print 5 rows

Total Payload Mass

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

```
[12]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE CUSTOMER = 'NASA (CRS)'
 * sqlite:///my_data1.db
Done.
[12]: SUM(PAYLOAD_MASS__KG_)
                       45596
```

By using the SUM() function on the PAYLOAD_MASS__KG_ field and a WHERE clause on CUSTOMER to filter on 'NASA', we see that the total payload mass that has been carried by boosters launched by NASA is 45,596kg

Average Payload Mass by F9 v1.1

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

```
[13]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE BOOSTER_VERSION = 'F9 v1.1'
 * sqlite:///my_data1.db
Done.
```

[13]:

AVG(PAYLOAD_MASS__KG_)
2928.4

Using the AVG() function on the PAYLOAD_MASS__KG_ field and a WHERE clause on the BOOSTER_VERSION field enable us to determine the average payload mass of the F9 v.1.1 rocket booster.

First Successful Ground Landing Date

```
[14]: %sql SELECT Date AS "First Successful Ground Landing" FROM SPACEXTABLE WHERE Date = (SELECT MIN(DATE) FROM SPACEXTABLE WHERE Landing_Outcome = "Success (ground pad)");
 * sqlite:///my_data1.db
Done.
[14]:
```

First Successful Ground Landing
2015-12-22

The query above uses a MIN function to show the first successful ground landing date, which was in December 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
[15]: %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE (Landing_Outcome = "Success (drone ship)") AND (PAYLOAD_MASS__KG_ >= 4000) AND (PAYLOAD_MASS__KG_ <= 6000);
 * sqlite:///my_data1.db
Done.
```

[15]:

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

The query above shows that there were only four successful ‘drone ship’ landings when the payload mass was between 4000-6000kg.

To get this data, the query uses a BETWEEN statement on the PAYLOAD_MASS__KG_ field and a WHERE clause on Landing_Outcome to filter on ‘drone ship’

Total Number of Successful and Failure Mission Outcomes

```
[16]: %sql SELECT COUNT(Mission_Outcome), (SELECT COUNT(Mission_Outcome) FROM SPACEXTABLE WHERE Mission_Outcome LIKE "Failure%") AS Failure FROM SPACEXTABLE WHERE Mission_Outcome = "Su
 * sqlite:///my_data1.db
Done.
```

[16]:

COUNT(Mission_Outcome)	Failure
98	1

Using the COUNT() statement we find the number of successes and failures from the Mission_Outcome column. It is important to note that this refers to the overall mission and not just the recovery of the first stage of the rocket.

Boosters Carried Maximum Payload

```
[17]: %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
 * sqlite:///my_data1.db
Done.
```

[17]:

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

Here a subquery is used to find the boosters that were able to carry the maximum payload, along with the maximum value of 15,600kg in order to ensure that the query had ran correctly

2015 Launch Records

```
[18]: %sql SELECT substr(Date, 6,2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE (substr(Date, 0,5) = '2015') AND (Landing_Outcome='Failure (drone shi
```

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

[18]:

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

Using SUBSTR() to isolate the month and year of the launches, we see that only two missions that had failed drone ship landings in January and April 2015

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

```
%sql SELECT Landing_Outcome AS Outcomes, COUNT(Landing_Outcome) AS Count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY COUNT
```

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

Outcomes	Count
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

Using the COUNT() and BETWEEN statements, we are able to see the number of launches and their outcomes between 2010-06-04 and 2017-03-20. We see that a third of the missions were classed as “No attempt” which indicates a lack of confidence in attempting first stage landings consistently

Section 3

Launch Sites Proximities Analysis

Launch Site Map

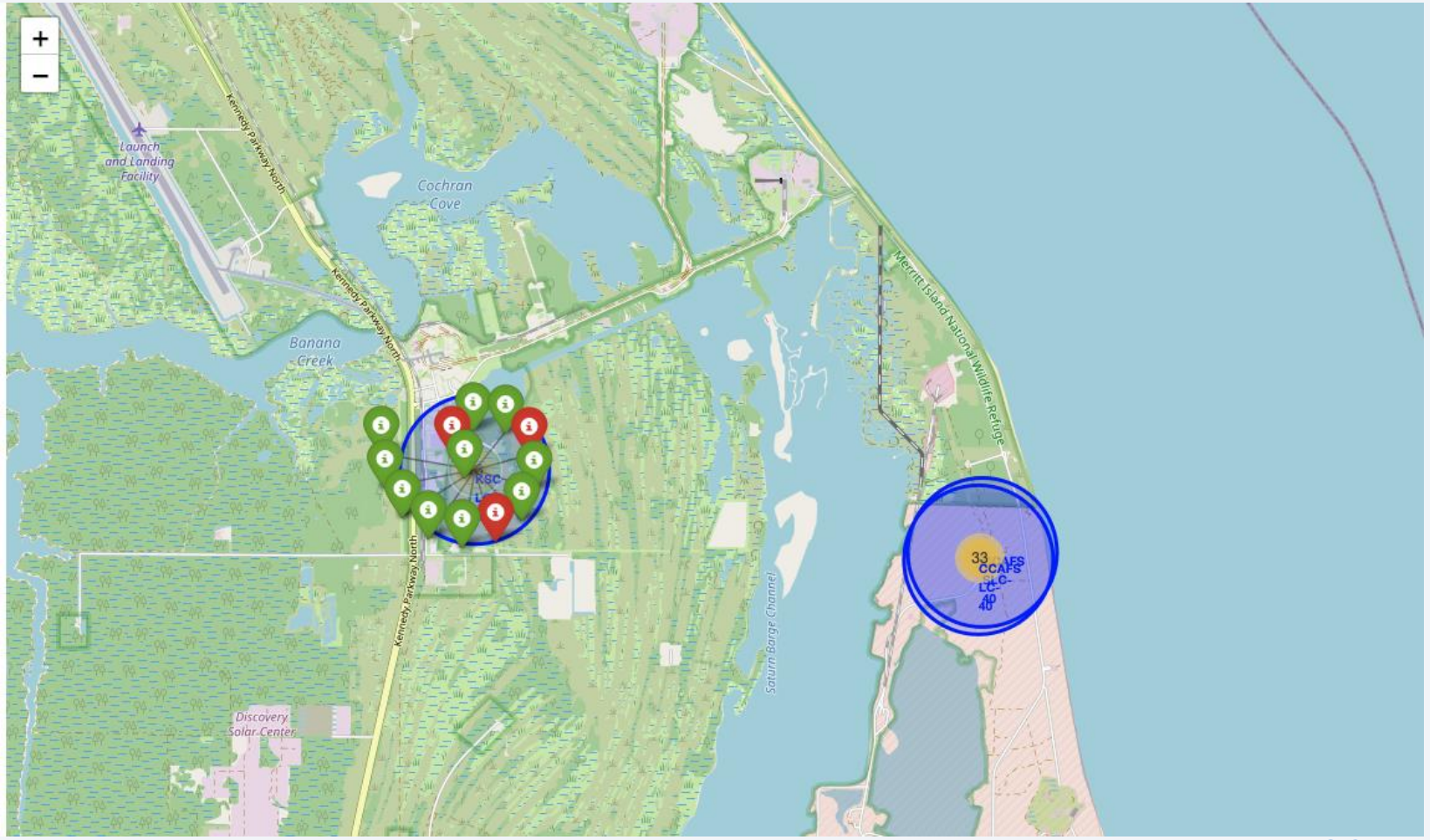
Using the Folium library, we have marked out the four different sites for SpaceX's rocket program.

Three of the sites are on the east coast of the US, whilst the other is on the west coast

Launch Outcomes at Sites

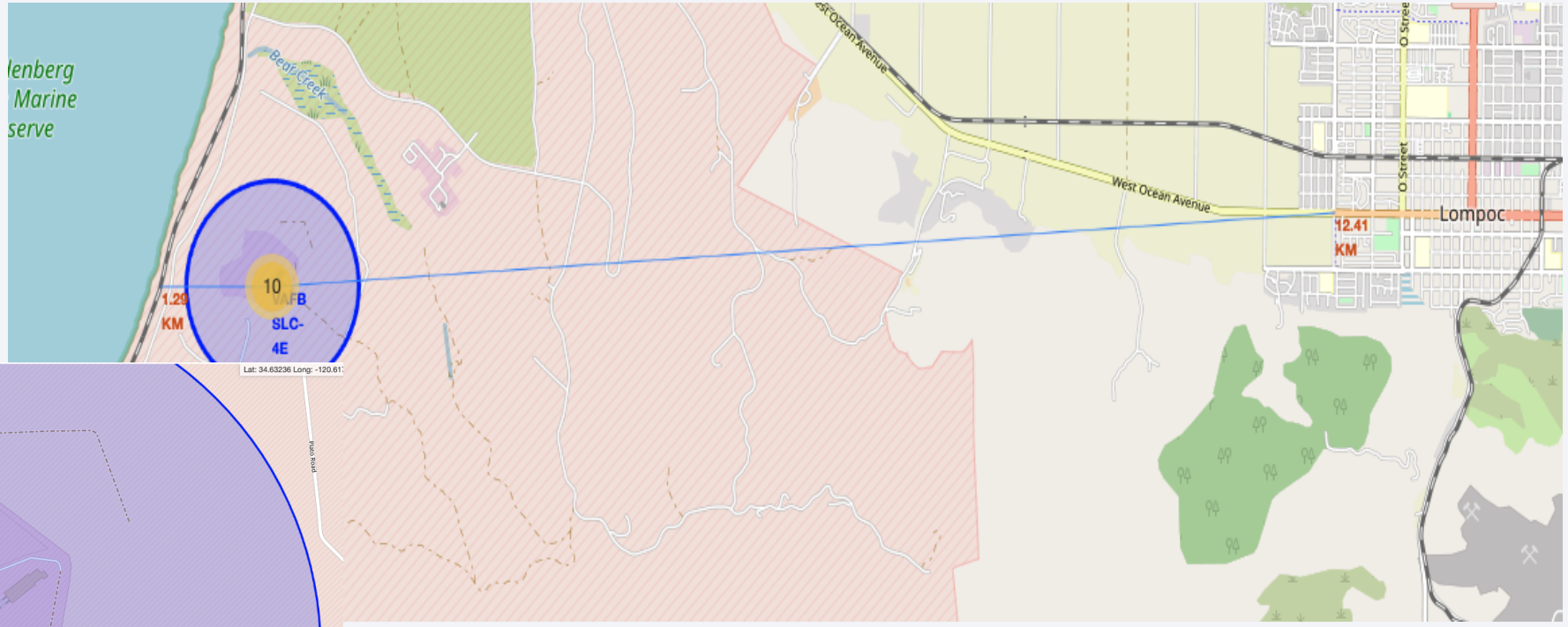
Using the marker_cluster function, we can place all of the launches at each site. This is indicated by the number inside each of the launch site circle marks.

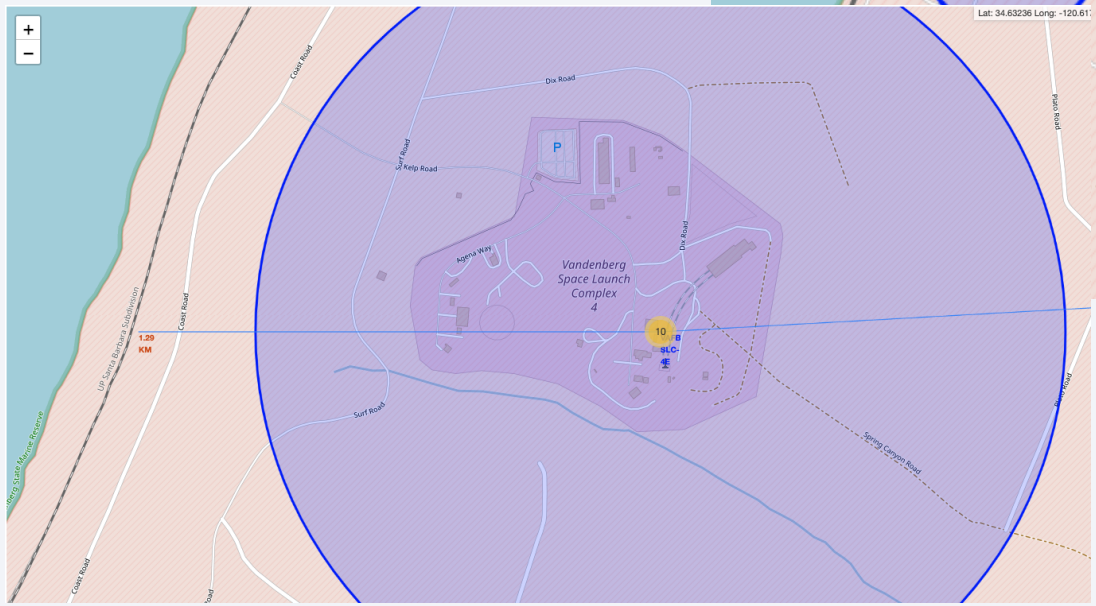
When clicked on, all of the single color marked launch data points are displayed per site. Green markers indicate successful missions, while red markers indicate failed missions.



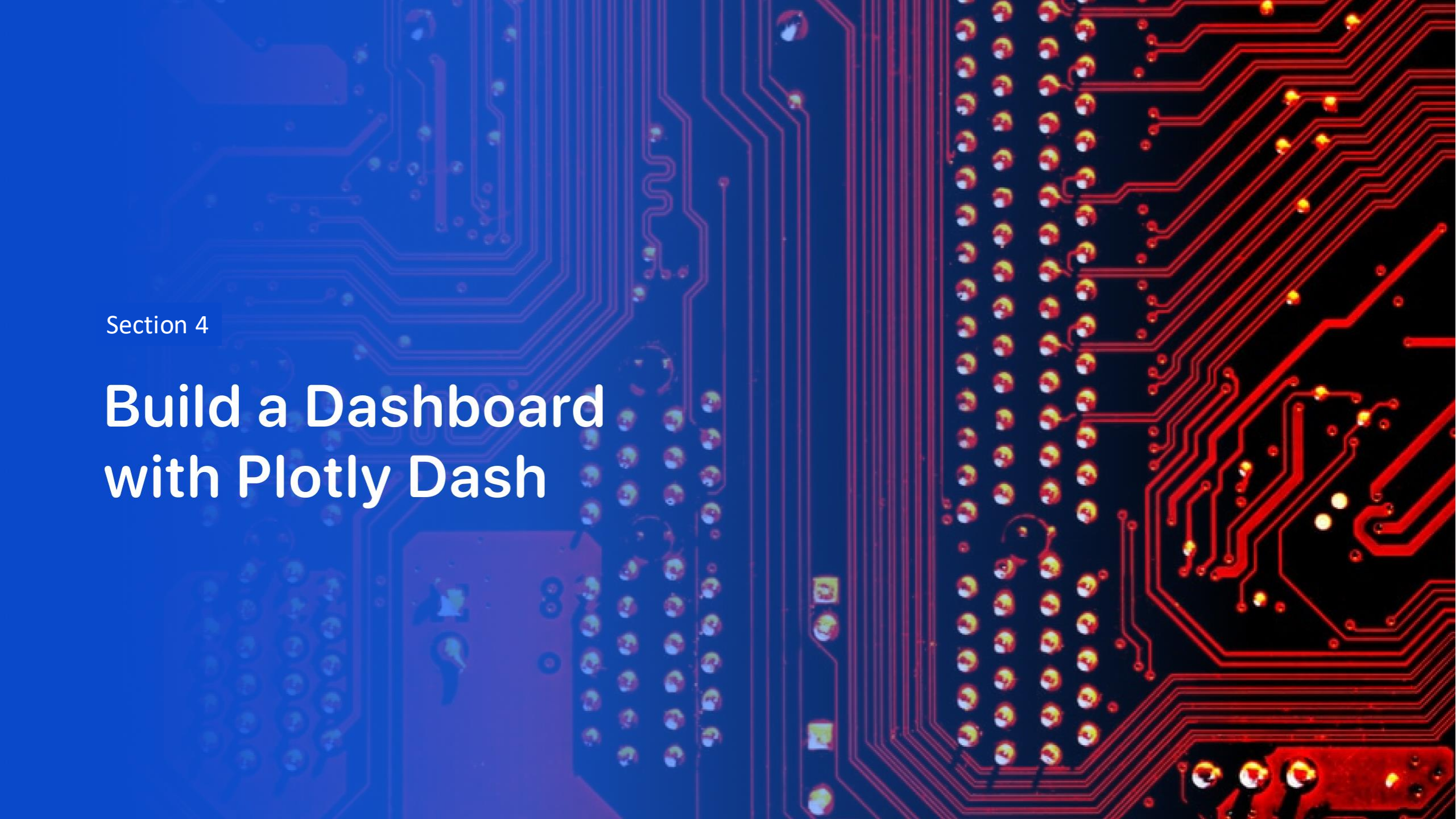
Proximity of Landmarks to Launch Site VAFB SLC-4E

This map shows the closest railroad, coastline, highway and city distances marked out for the VAFB SLC-4E launch site, which is the single west coast launch site

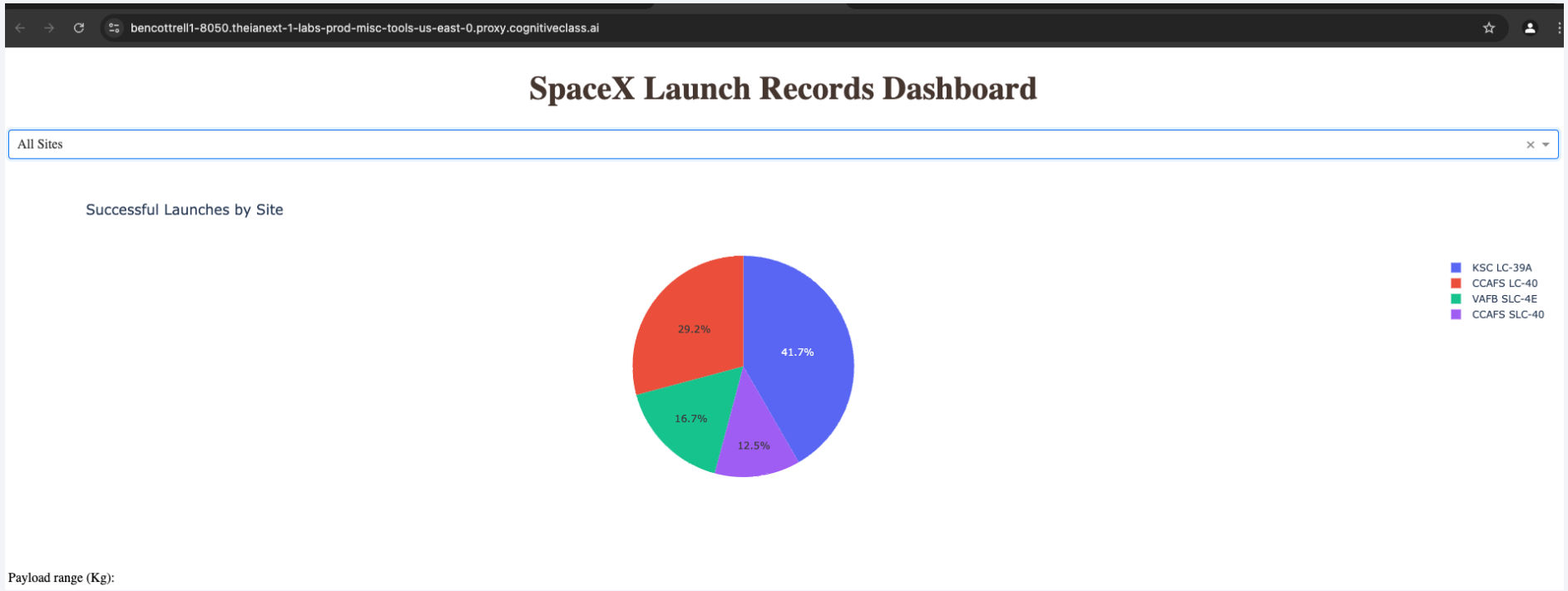




2nd map is zoomed in to show the nearest railroad and the distance to the coastline

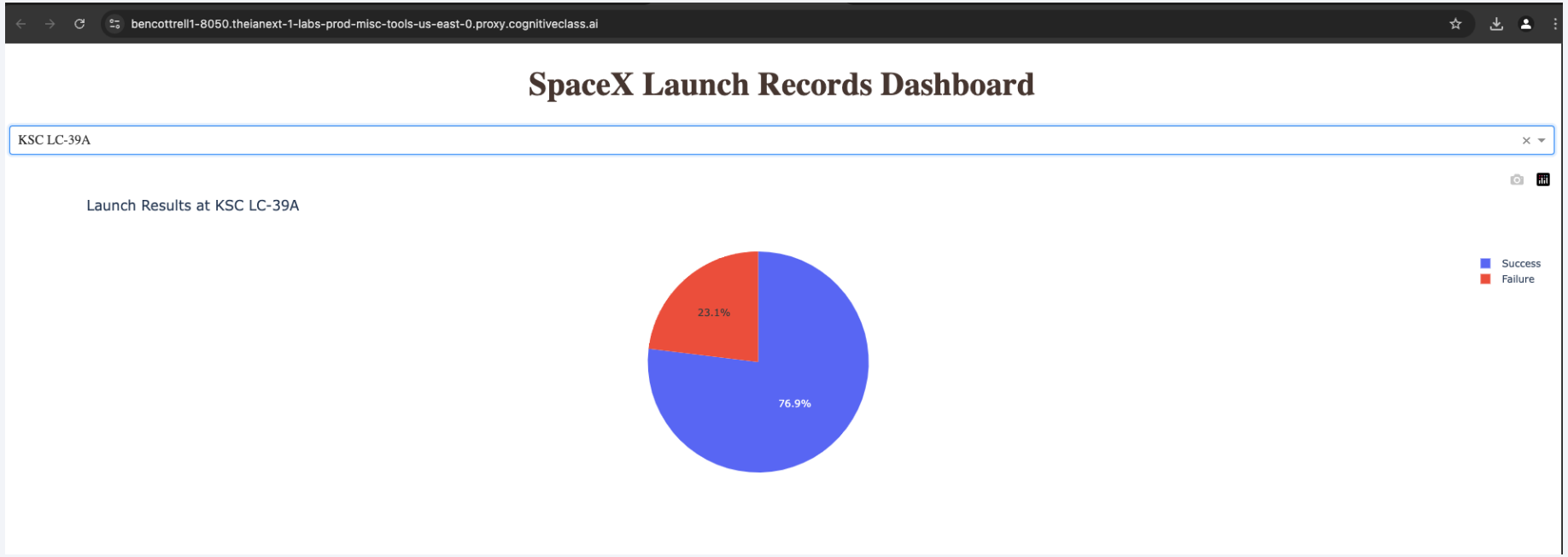

Section 4
Build a Dashboard with Plotly Dash

Successful SpaceX Launch Missions by Site



In the pie chart above we are observing the ratios of the successful missions by all launch sites. The site with the most successful launches is the KSC LC-39A site at 41.7%.

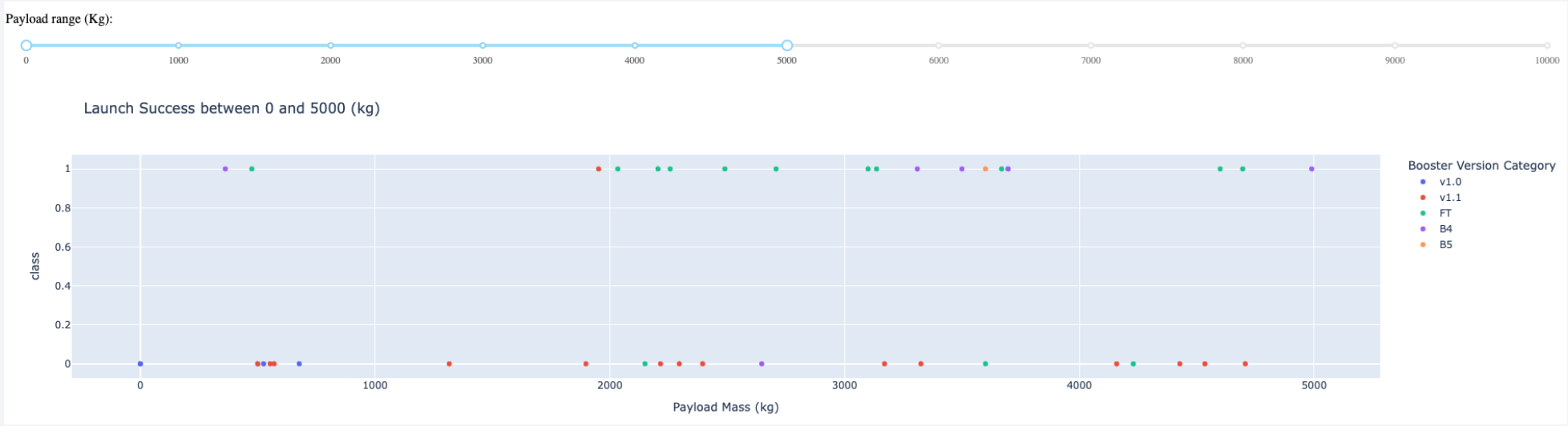
Site with Highest Launch Success



As indicated by the 'All Sites' pie chart, the KSC LC-39A site also had the highest individual success rate, with only 23.1% of it's launches resulting in failure.

It is worth noting however that this site only had 13 launches, whereas CCAFS LC-40 had 26 launches, possibly indicating that the CCAFS LC-40 site was used more frequently in the early research phase of the program

Success vs Payload Mass and Booster Type



When observing payloads below 5000kg, nearly all the failed launches are boosters v1.0 and v1.1

When observing payloads over 5000kg, we only see that two boosters are used (FT and B4) and both have low success rates

Section 5

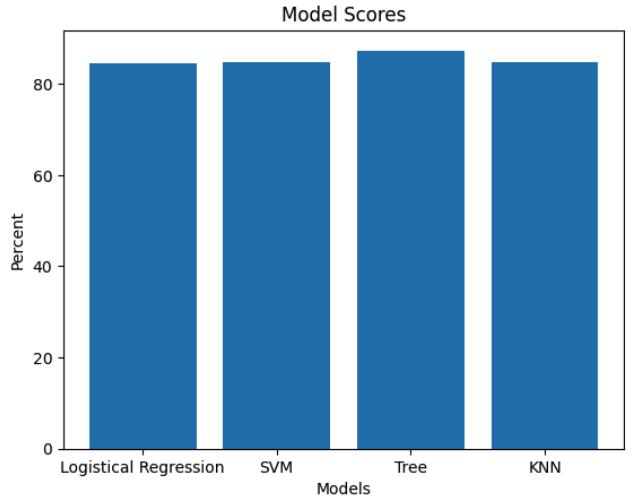
Predictive Analysis (Classification)

Classification Accuracy

```
[59]: scores = [(logreg_bestscore*100), (svm_bestscore*100), (tree_bestscore*100), (knn_bestscore*100)]
models = ['Logistical Regression', 'SVM', 'Tree', 'KNN']
print(scores)
print(models)

plt.bar(models, scores)
plt.xlabel('Models')
plt.ylabel('Percent')
plt.title('Model Scores')
plt.show()
```

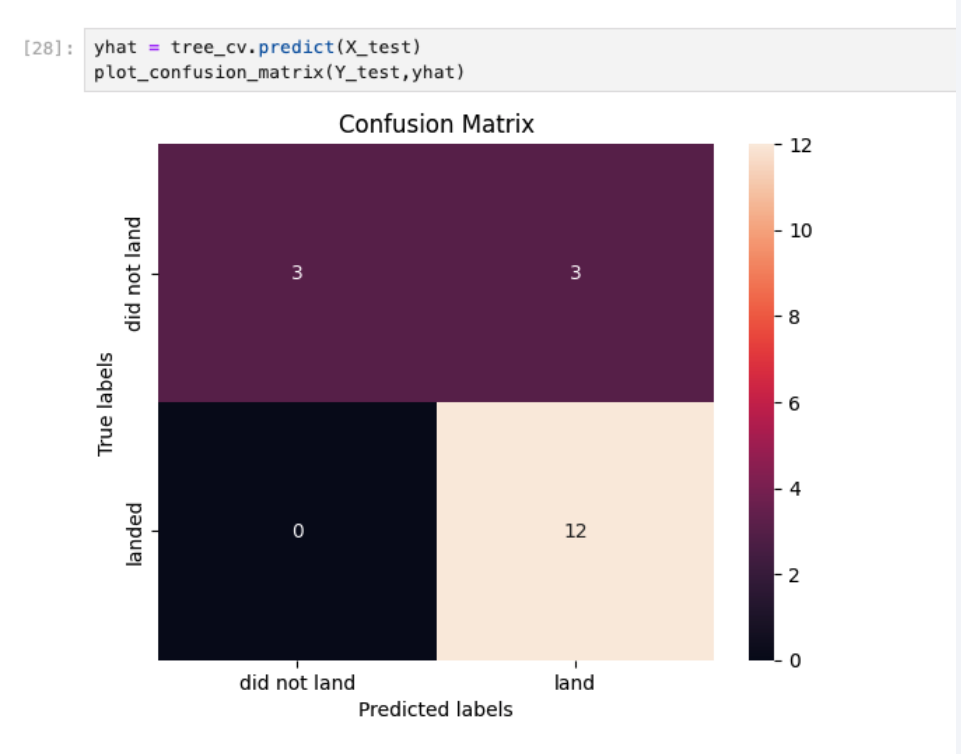
```
[84.64285714285712, 84.82142857142856, 87.32142857142857, 84.82142857142858]
['Logistical Regression', 'SVM', 'Tree', 'KNN']
```



After evaluation all classification models, such as Logistical Regression, Support Vector Machine (SVM), Decision Tree and K-Nearest Neighbour (KNN) we see that the Decision tree algorithm was the most accurate at %87.32.

Confusion Matrix

The Decision Tree algorithm does a good job in classifying the test set. The confusion matrix shows that it only misinterpreted three data points, which put out three false-positive predictions



Conclusions

- The VLEO orbit missions have high success rates.
- Progressive research and time have seen higher successful returns with all missions.
- All launch sites are on the southern edge of the country and close to major cities, supply lines such as highways and train tracks and coasts.
- Heavy payload missions continue to struggle to land successfully.
- Understanding the differences between the v1.0 and v1.1 rockets and FT, B4, and B5 rockets will give us a better understanding of how to have successful missions as their success rates are higher.
- The Decision Tree is the best classification model to use to predict launch outcomes, with 87.3% accuracy

Appendix

- https://github.com/bcottrell93/DataScienceCapstoneProject/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

Thank you!