IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chloe Wu
1/30/23

Outline

★ Executive Summary

★ Introduction

★ Methodology

★ Results

★ Conclusion

★ Appendix

Executive Summary

★ Summary of methodologies

- Data Collection w/API
- Data Collection w/Web Scraping
- Data Wrangling
- Data Analysis w/SQL
- Data Analysis w/Data Visualization
- Interactive Dashboard creation w/Folium
- Machine Learning Predictions

★ Summary of all results

- Exploratory Data Analysis Results
- Folium GeoData Results
- DashBoard Results
- Predictive Analysis Results

Introduction

We will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

★ Problems to solve

- Identify factors that affect landing outcomes
- Relationships between variables and outcomes
- Find scenario where landing outcome is most successful

Section 1

Methodology

Methodology

Executive Summary

★ Data collection methodology:

- Data Collected from the SpaceX API and web scraped from Wikipedia

★ Perform data wrangling

- Data was processed w/one-hot encoding for categorical features

★ Perform exploratory data analysis (EDA) using visualization and SQL

★ Perform interactive visual analytics using Folium and Plotly Dash

★ Perform predictive analysis using classification models

Data Collection

Data collected would need to be relevant to SpaceX missions and their outcomes.

We first web scraped data from the SpaceX wikipedia page by extracting launch record data and parsing it into a pandas dataframe with BeautifulSoup. We also extracted data from the SpaceX REST API by converting the json file to a pandas dataframe. After getting the data from both sources we cleaned and prepared the data for the next step.

Data Collection – SpaceX API

Use .get() to request the rocket launch API data

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

Convert Json file into pandas dataframe

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

Clean data and fill missing values

```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a si
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

https://github.com/ChloWu/SpaceX-Coursera-Capstone-Project/blob/main/1_Collecting_the%20_data.ipynb

Data Collection – Scraping

Use .get() to request launch data from Wikipedia page url

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url,'html5lib').text
```

Use BeautifulSoup to parse HTML data

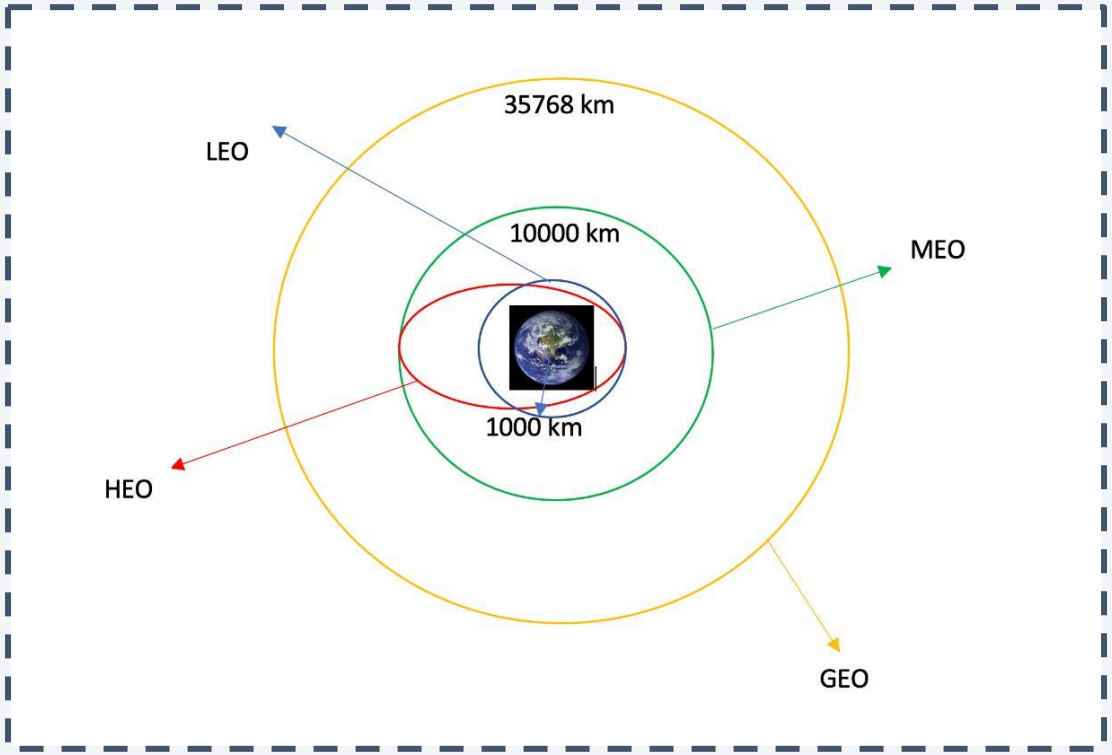
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response)
```

Extract column/variable data

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
```

https://github.com/ChloWu/SpaceX-Coursera-Capstone-Project/blob/main/*1_SpaceX_webscrape_from_wikipedia.ipynb

Data Wrangling



Data will need to be cleaned and processed to make the complex data sets easier to analyze.

First, we identify and calculate missing values from each attribute. We then Find the occurrences of each orbit and mission outcome from which we create a landing class from

https://github.com/ChloWu/SpaceX-Coursera-Capstone-Project/blob/main/2_Data_Wrangling.ipynb

EDA with Data Visualization

We used scatter plots to visualize the relationships between:

- Flight Number and Launch Site
- Payload and Launch Site
- Success Rate and Orbit Type
- Flight Number and Orbit Type
- Payload and Orbit Type

After this we used a line plot to display the yearly trend of launch success.

Scatter plots allow us to correlate relationships between these different attributes.

https://github.com/ChloWu/SpaceX-Coursera-Capstone-Project/blob/main/3_Exploring_and_Preparing_Data.ipynb

EDA with SQL

We preformed many SQL queries to better understand the dataset including...

- Displaying launch site names
- Displaying 5 records that have a launch site name the begins with 'CCA'
- Displaying total payload mass carried by NASA launched boosters
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the data of the first successful groundpad landing
- Listing the names of boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass
- Listing the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch site for the months in year 2015
- Ranking the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

https://github.com/ChloWu/SpaceX-Coursera-Capstone-Project/blob/main/*3_SQL_analysis_notebook.ipynb

Build an Interactive Map with Folium

We visualized the launched sites with an interactive map using longitude and latitude data and assigned red markers to failed landings and green markers to successful ones.

After you plot distance lines to the proximities, you can answer the following questions easily:

- ➔ Are launch sites in close proximity to railways?
- ➔ Are launch sites in close proximity to highways?
- ➔ Are launch sites in close proximity to coastline?
- ➔ Do launch sites keep certain distance away from cities?

https://github.com/ChloWu/SpaceX-Coursera-Capstone-Project/blob/main/*4_Interactive_Visual_%20Analytics_with_Folium_lab.ipynb

Build a Dashboard with Plotly Dash

- We used Plotly Dash to build an interactive dashboard to allow the user to access the data in real-time applications
- We plotted a pie chart on total successful launches to display the success to failure ratio for each site
- We plotted a scatter plot to show correlation between payload and success rate

https://github.com/ChloWu/SpaceX-Coursera-Capstone-Project/blob/main/4_Interactive_Plotly_Dashboard.py

Predictive Analysis (Classification)

Building the model

- Load the dataset into NumPy and Pandas
- Transform and split the data for testing
- Decide what type of Machine Learning to do
- set parameters and algorithms to GridSearchCV and fit the data set

Evaluating and improving the model

- Check accuracy for each model
- Get hyperparameters for each type of algorithm
- Plot the confusion matrix

Finding the best model

- Find the model with the highest accuracy score, this will be the best model

https://github.com/ChloWu/SpaceX-Coursera-Capstone-Project/blob/main/5_Machine_Learning_prediction.ipynb

Results

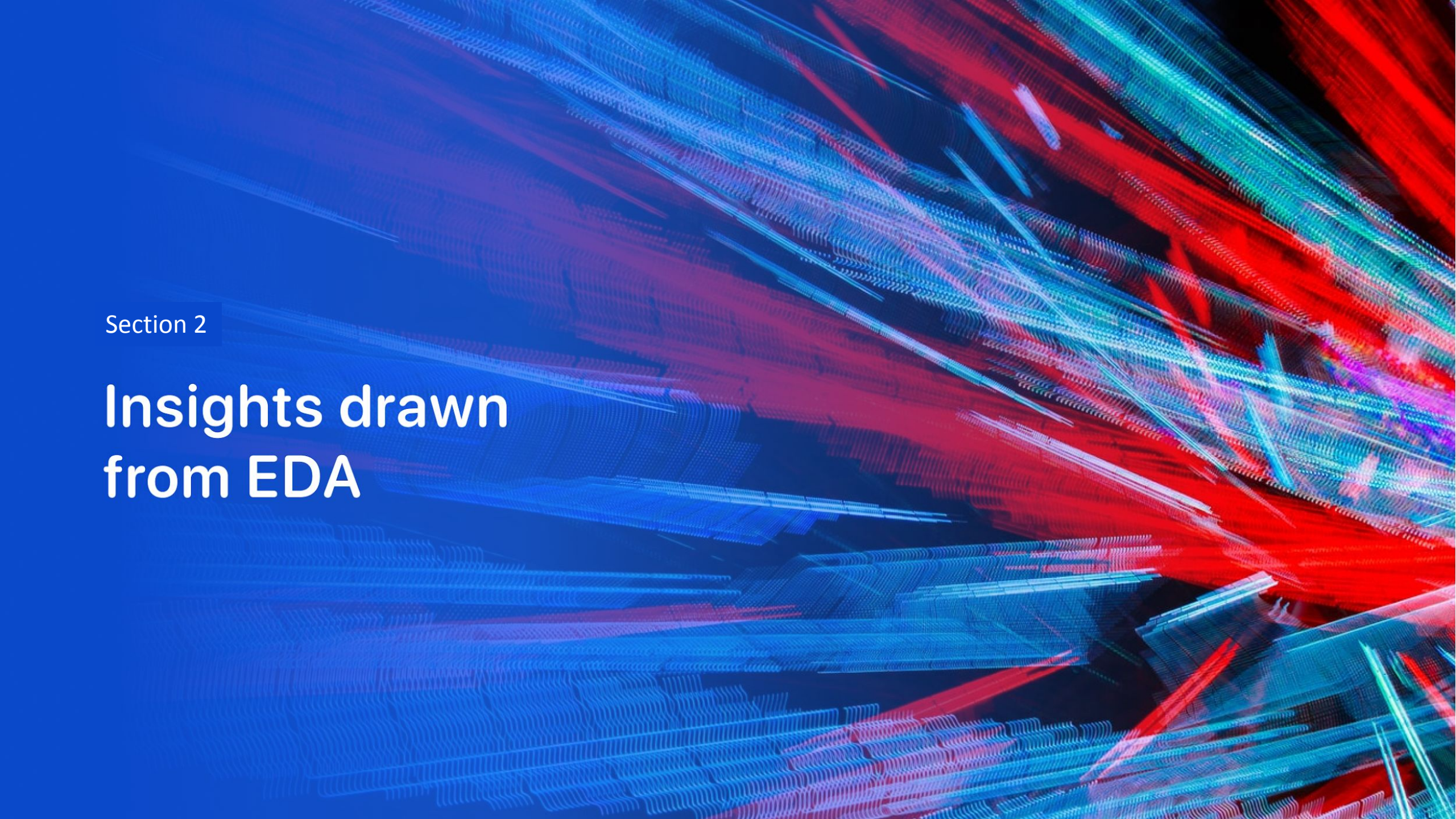
★ Exploratory data analysis results

★ Interactive dashboard and visual analytics results

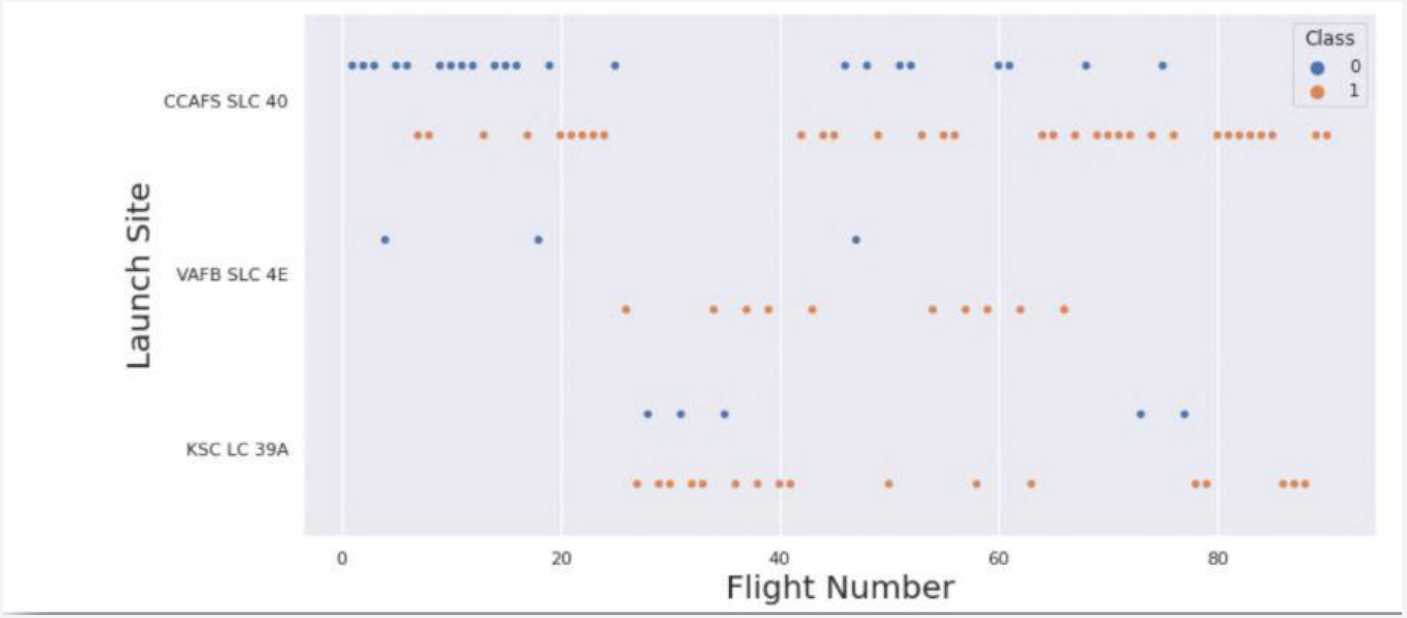
★ Predictive analysis results

Section 2

Insights drawn from EDA



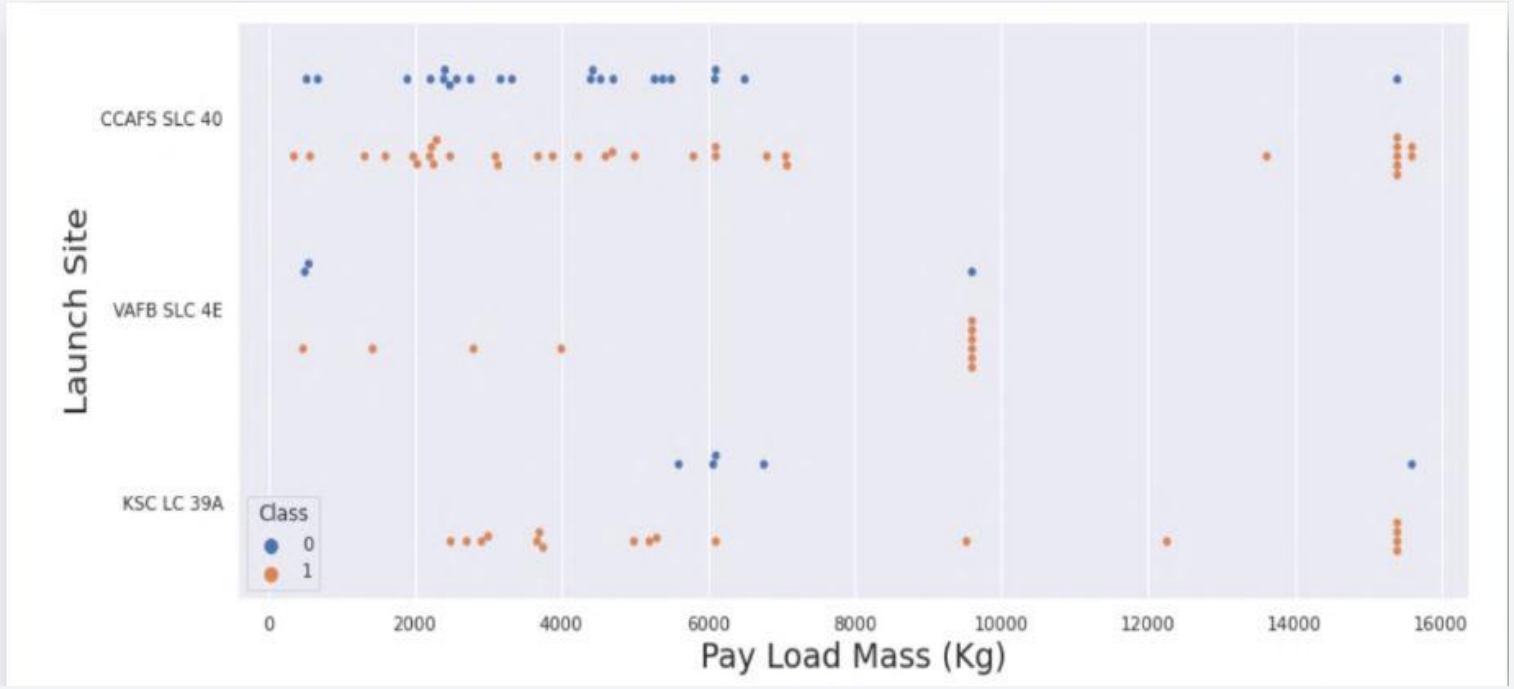
Flight Number vs. Launch Site



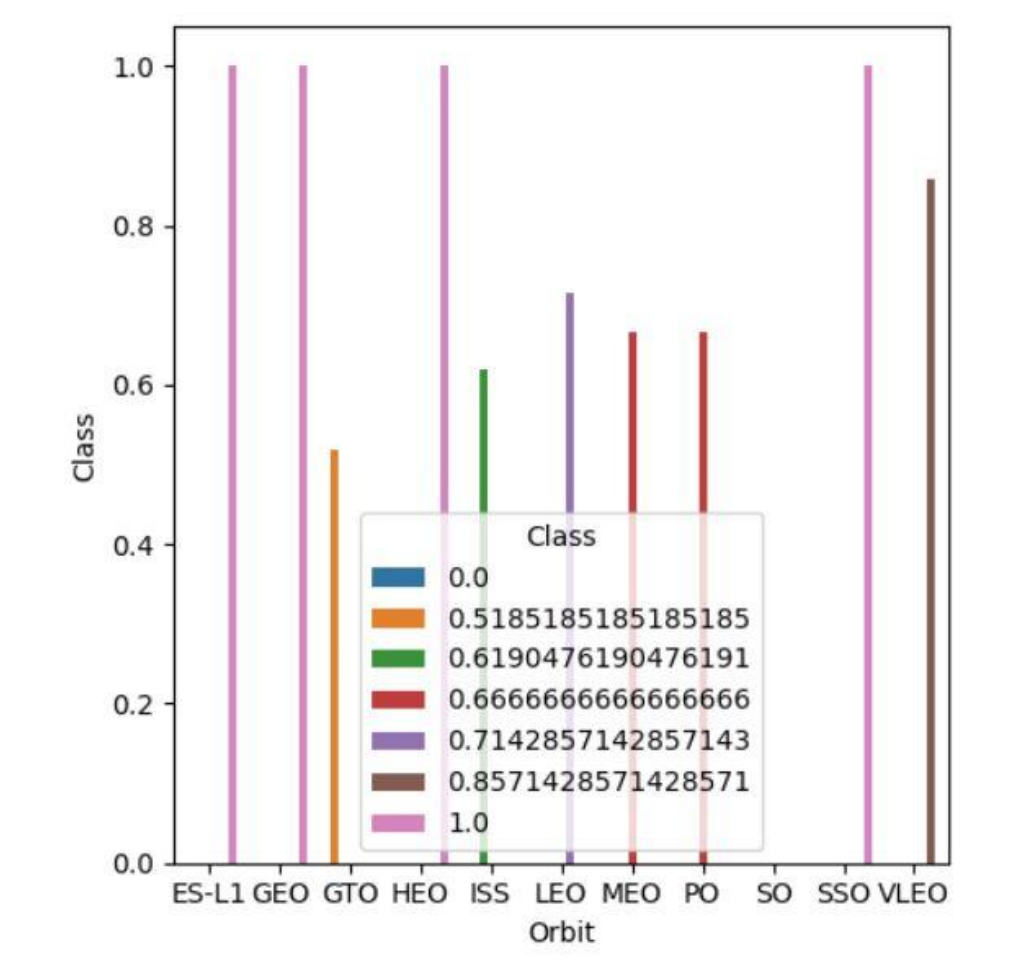
This scatter plot shows the relationship between the number of flights and each launch site. A greater number of flights seem to show more successful attempts.

Payload vs. Launch Site

This scatter plot shows correlation between launch site and payload mass and that most launches after the 7000 kg mark are mostly successful



Success Rate vs. Orbit Type

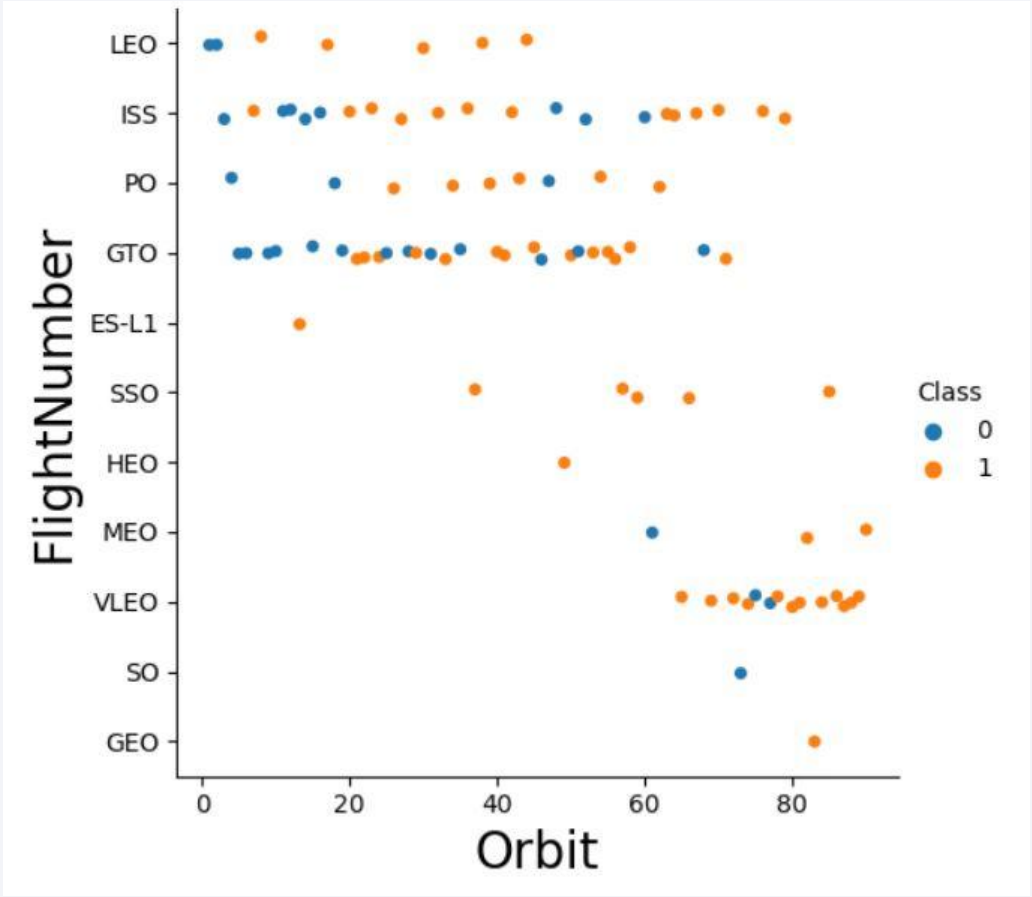


This bar chart visualizes success rate in relation to orbit type. ES-L1, GEO, HEO, and SSO all display a 100% success rate. SO has the lowest success rate of all the launch sites with 0% and GTO has the second lowest success rate of 52%.

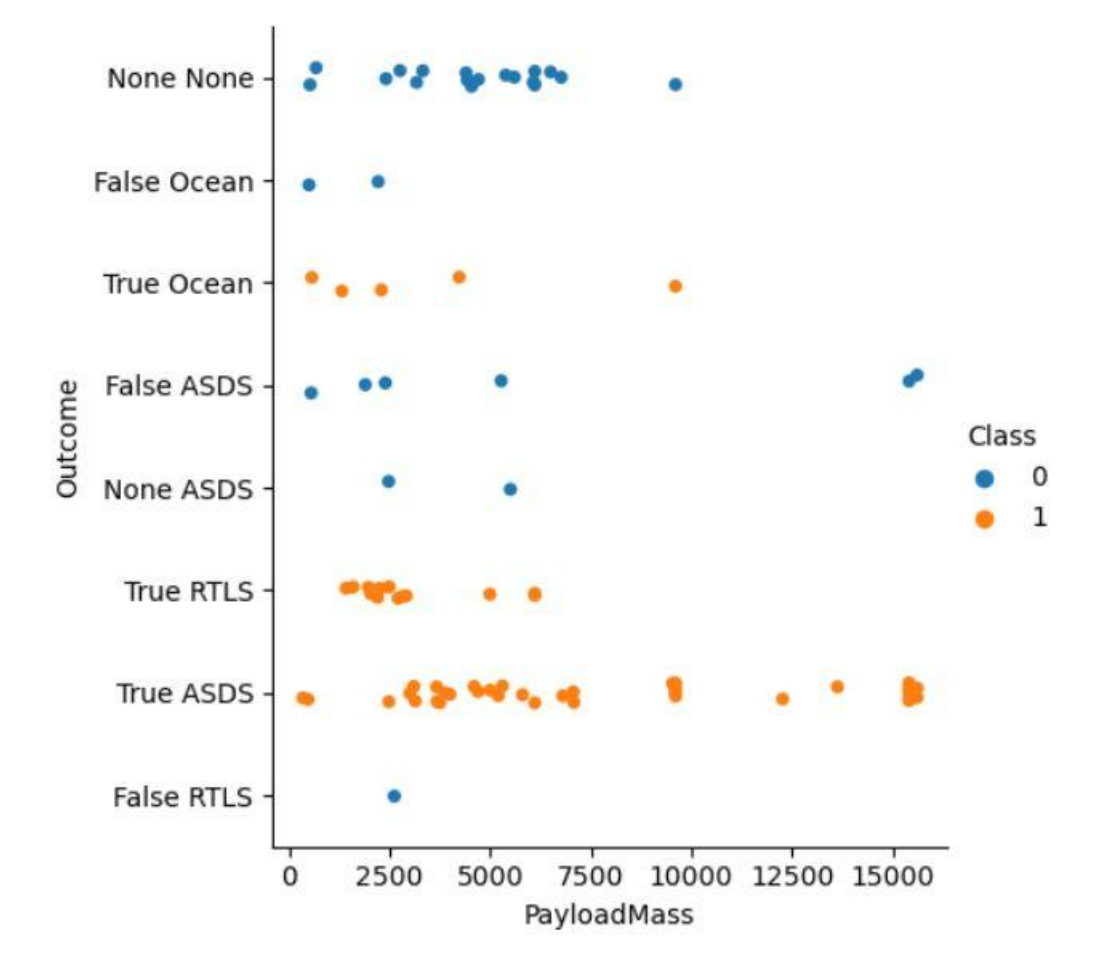
Flight Number vs. Orbit Type

This scatter plot shows that the larger the flight number = the greater the success rate.

However, I have messed up my code and have mixed up the x-axis and y-axis labels

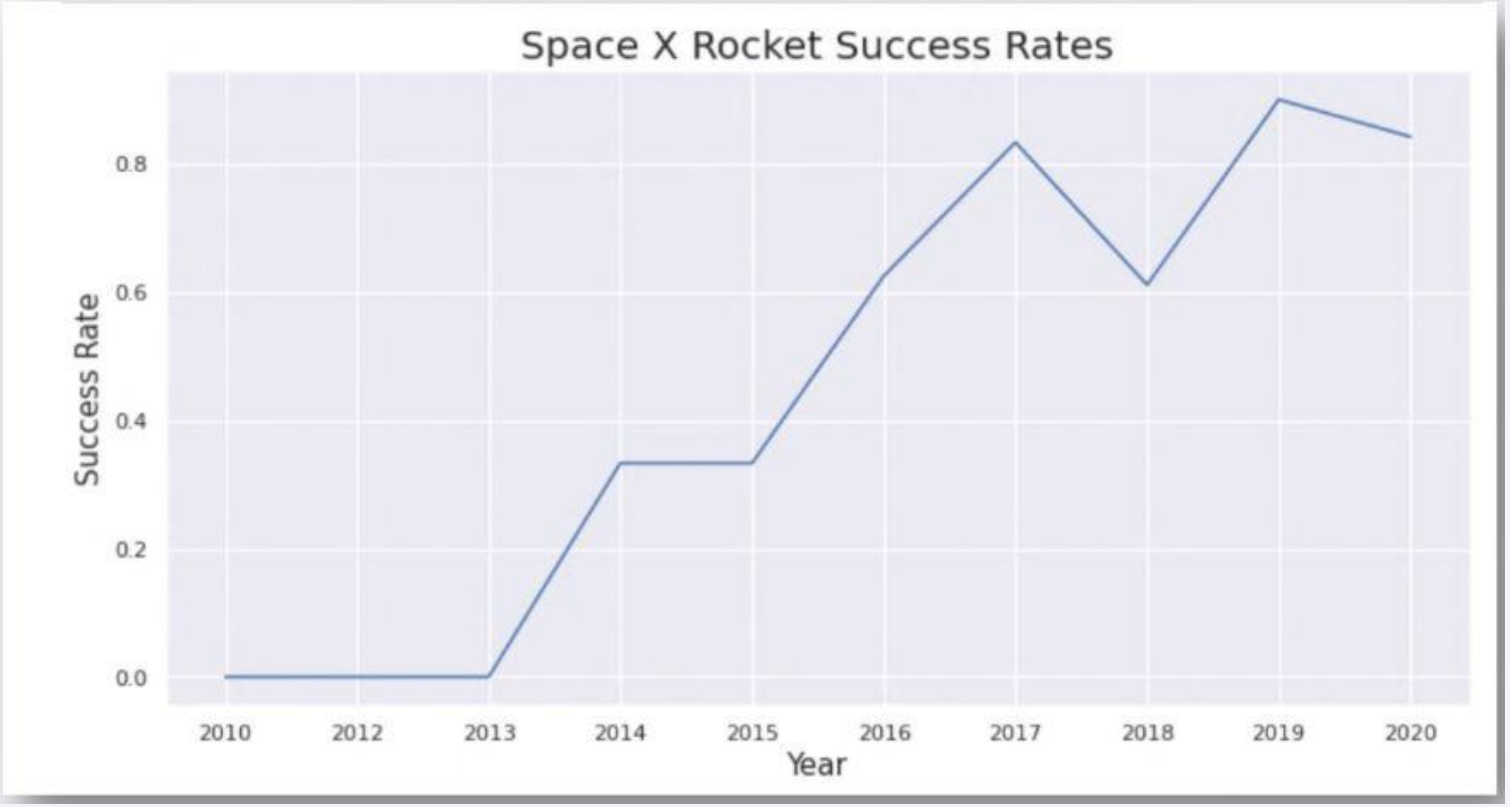


Payload vs. Orbit Type



This scatter plot gives info on the payload and the orbit type by I messed up the labels again

Launch Success Yearly Trend



This line plot showcases an increasing trend in success rate in relation to time in years.

All Launch Site Names

We queried for launch site names with the DISTINCT query to find all the unique site names

```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEX
```

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

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

Launch Site Names Begin with 'CCA'

We found the first five records that have a launch site name starting with the string 'CCA'

```
%sql SELECT * FROM SPACEX 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
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	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 calculated the total payload carried by boosters from NASA. We used sum() to query

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS 'SUM OF PAYLOAD MASS' FROM SPACEX WHERE CUSTOMER LIKE 'NASA (CRS)'
```

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

SUM OF PAYLOAD MASS
45596

Average Payload Mass by F9 v1.1

We found the average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS 'AVERAGE PAYLOAD MASS' FROM SPACEX WHERE BOOSTER_VERSION LIKE 'F9 v1.1%'
```

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

AVERAGE PAYLOAD MASS
2534.6666666666665

First Successful Ground Landing Date

We found the date of the first successful landing outcome on ground pad

```
%sql SELECT MIN(DATE) AS 'FIRST SUCCESSFUL GROUND PAD LANDING' FROM SPACEX WHERE "Landing _Outcome" = "Success (ground pad)";
```

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

FIRST SUCCESSFUL GROUND PAD LANDING
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

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 DISTINCT(BOOSTER_VERSION) AS "BOOSTER VERSION", PAYLOAD_MASS__KG_ AS "PAYLOAD MASS" FROM SPACEX
WHERE  "Landing _Outcome" = "Success (ground pad)"
AND PAYLOAD_MASS__KG_ > 4000
AND PAYLOAD_MASS__KG_ < 6000;
```

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

BOOSTER VERSION	PAYLOAD MASS
F9 FT B1032.1	5300
F9 B4 B1040.1	4990
F9 B4 B1043.1	5000

Total Number of Successful and Failure Mission Outcomes

We calculated the total number of successful and failure mission outcomes

```
%sql SELECT DISTINCT(MISSION_OUTCOME), COUNT(MISSION_OUTCOME) AS OUTCOME FROM SPACEX GROUP BY(MISSION_OUTCOME)
```

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

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

Boosters Carried Maximum Payload

```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEX WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX)
```

```
 * 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

We list the names of the boosters which have carried the maximum payload mass

2015 Launch Records

We listed the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT DATE, BOOSTER_VERSION, LAUNCH_SITE, "Landing _Outcome" FROM SPACEX WHERE "Landing _Outcome" = 'Failure (drone ship)' AND substr(Date,7,4)
```

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

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

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

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") AS LANDING_OUTCOME_COUNT,DATE FROM SPACEX
WHERE substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2) BETWEEN '20100604' and '20170320'
and "Landing _Outcome" LIKE "Success%"
GROUP BY  "Landing _Outcome" order by count("Landing _Outcome") desc
```

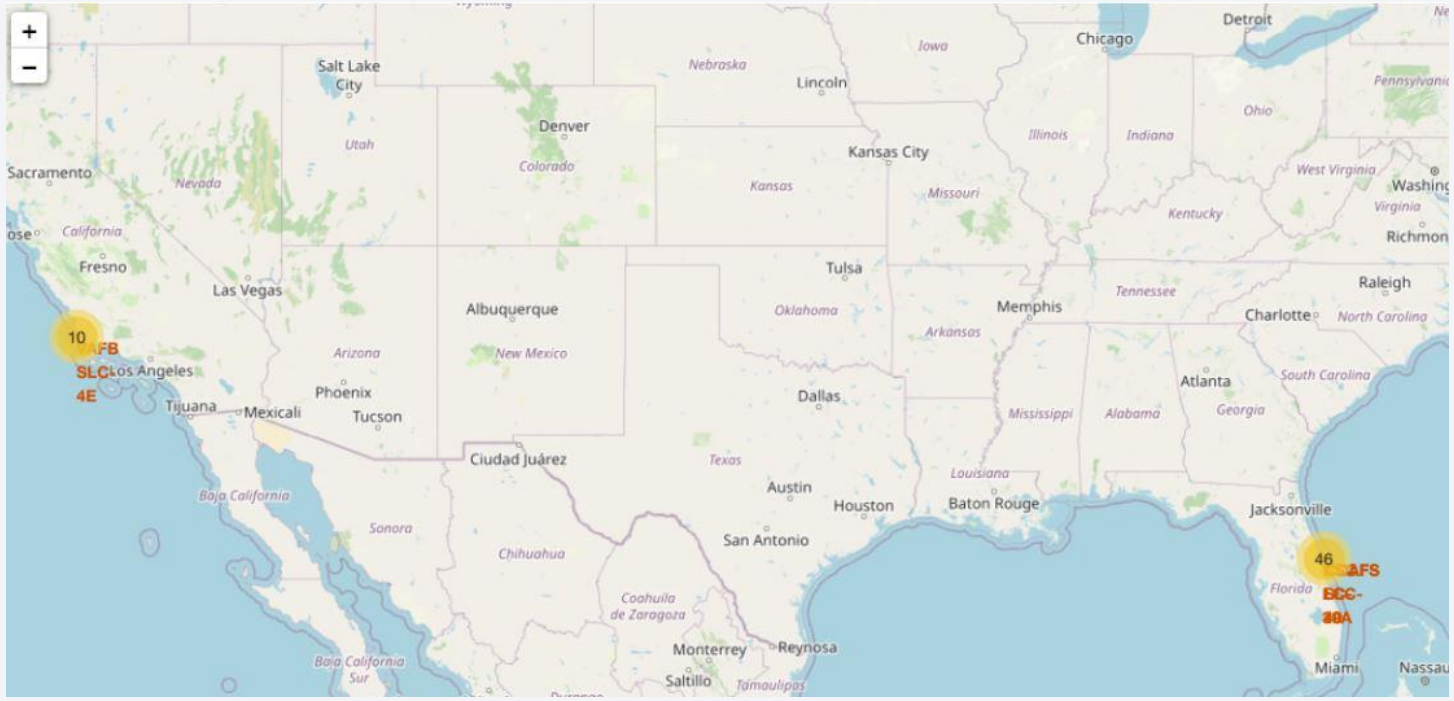
```
 * sqlite:///my_data1.db
Done.
```

Landing _Outcome	LANDING_OUTCOME_COUNT	Date
Success (drone ship)	5	08-04-2016
Success (ground pad)	3	22-12-2015


Section 3
Launch Sites
Proximities Analysis

All Launch Sites

This is a picture of all the launch sites in the U.S. where most are located in California or Florida

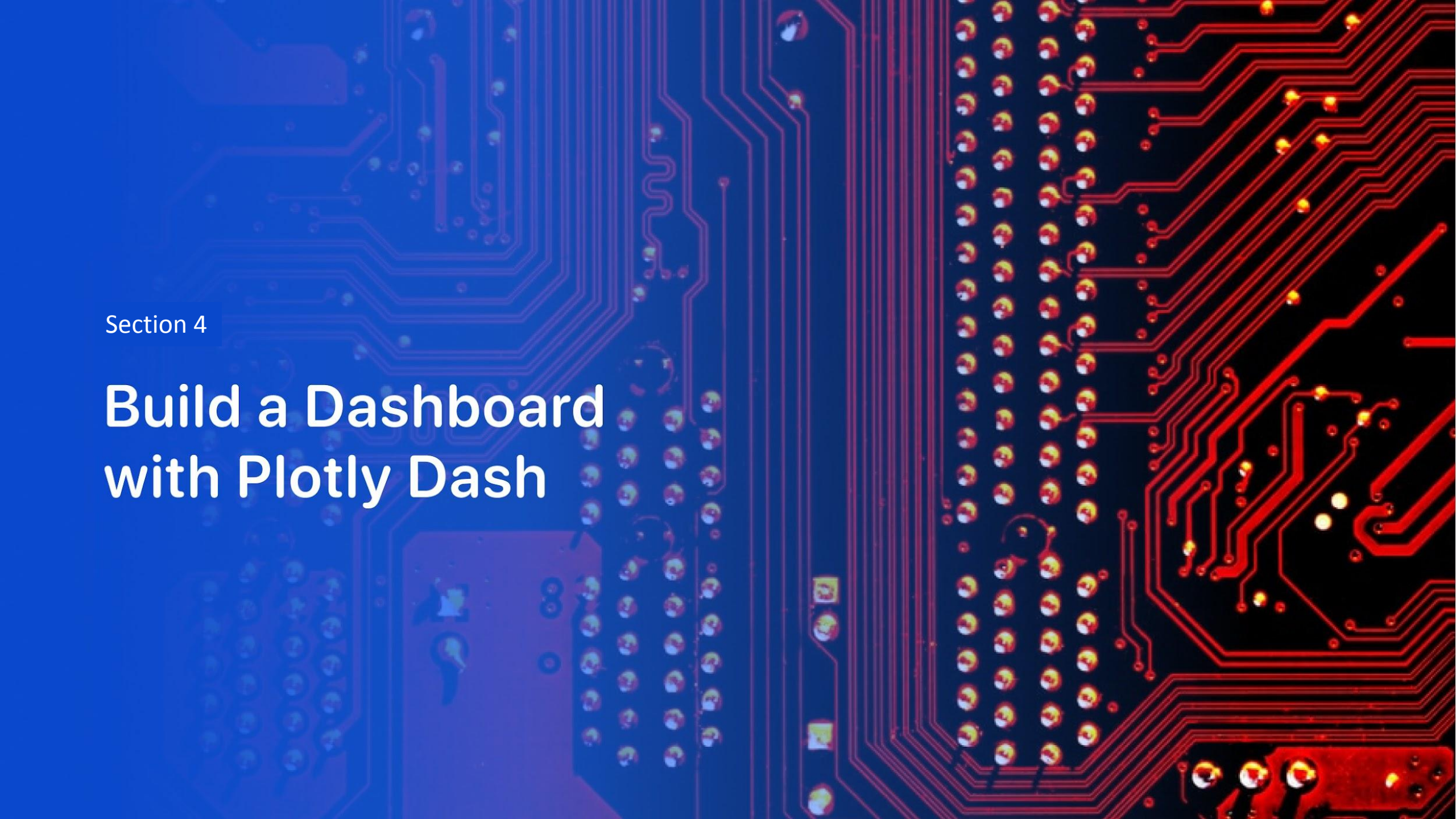


Individual Launch Site Markers

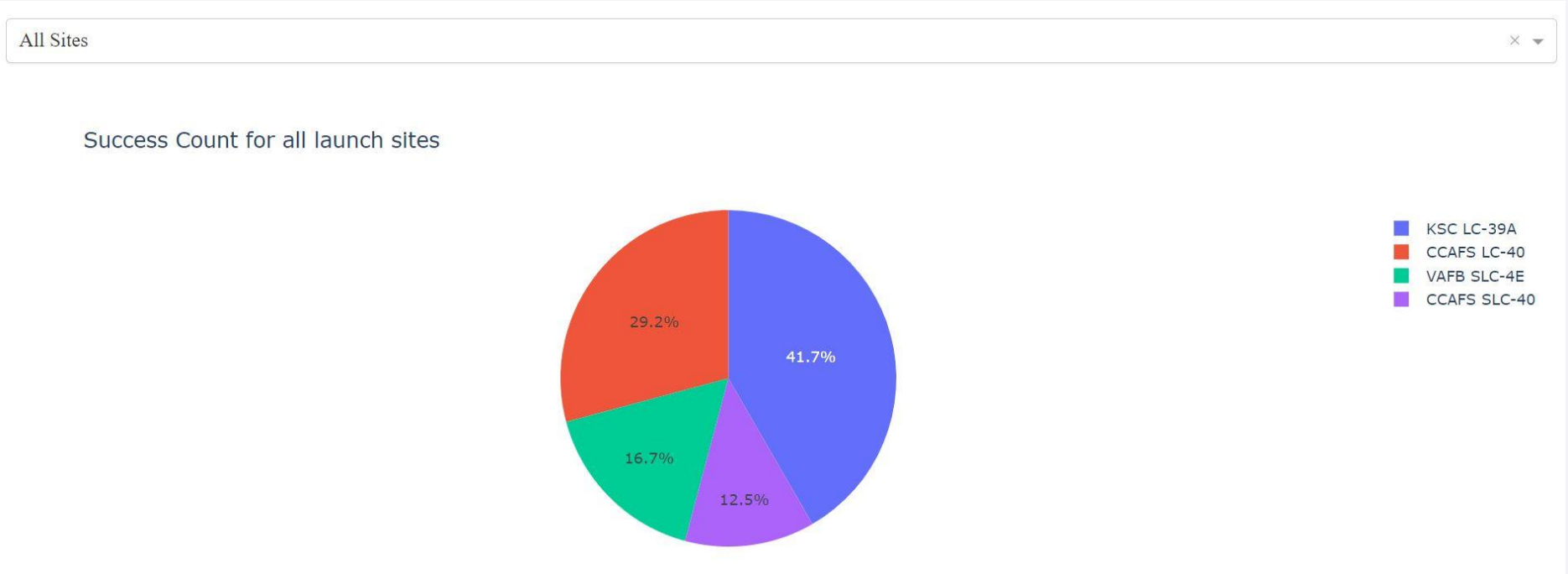
This image is of the launch sites in Florida marked with Green for successful launches and Red for unsuccessful launches

Distance from the Coastline

This image is the updated map that shows each launch site distance from the coastline

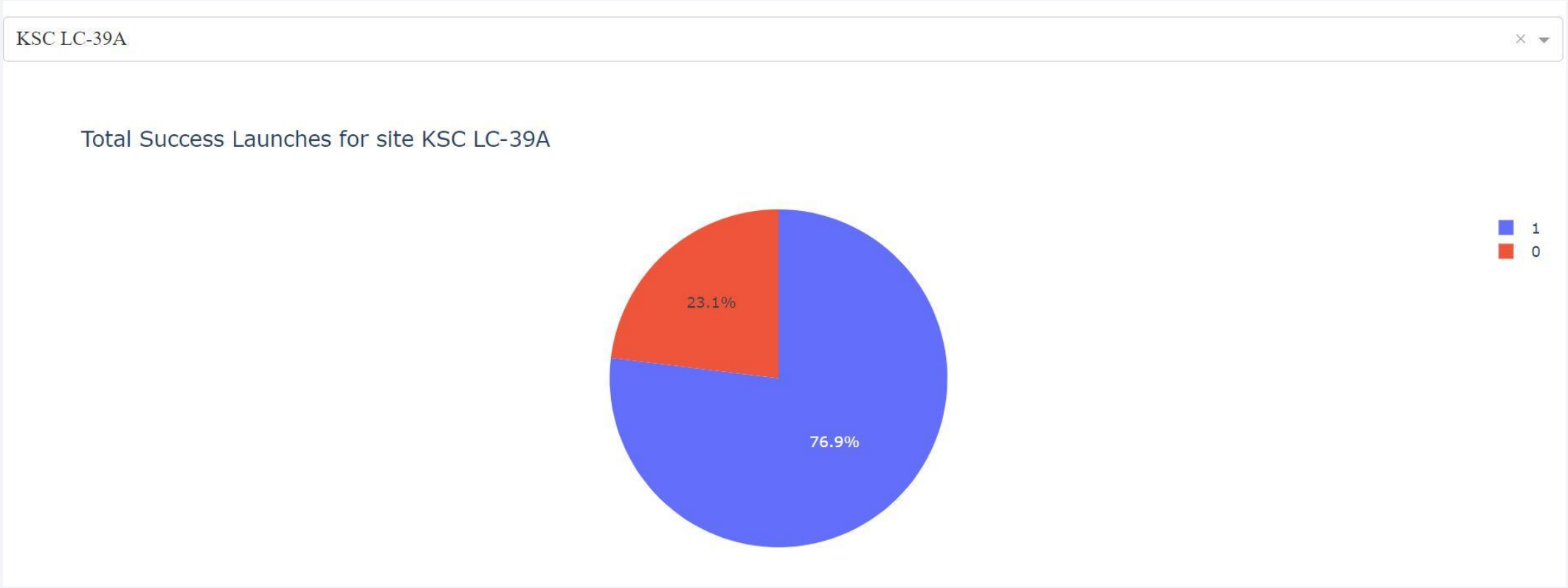

Section 4
Build a Dashboard with Plotly Dash

Pie chart Success Count (All Sites)



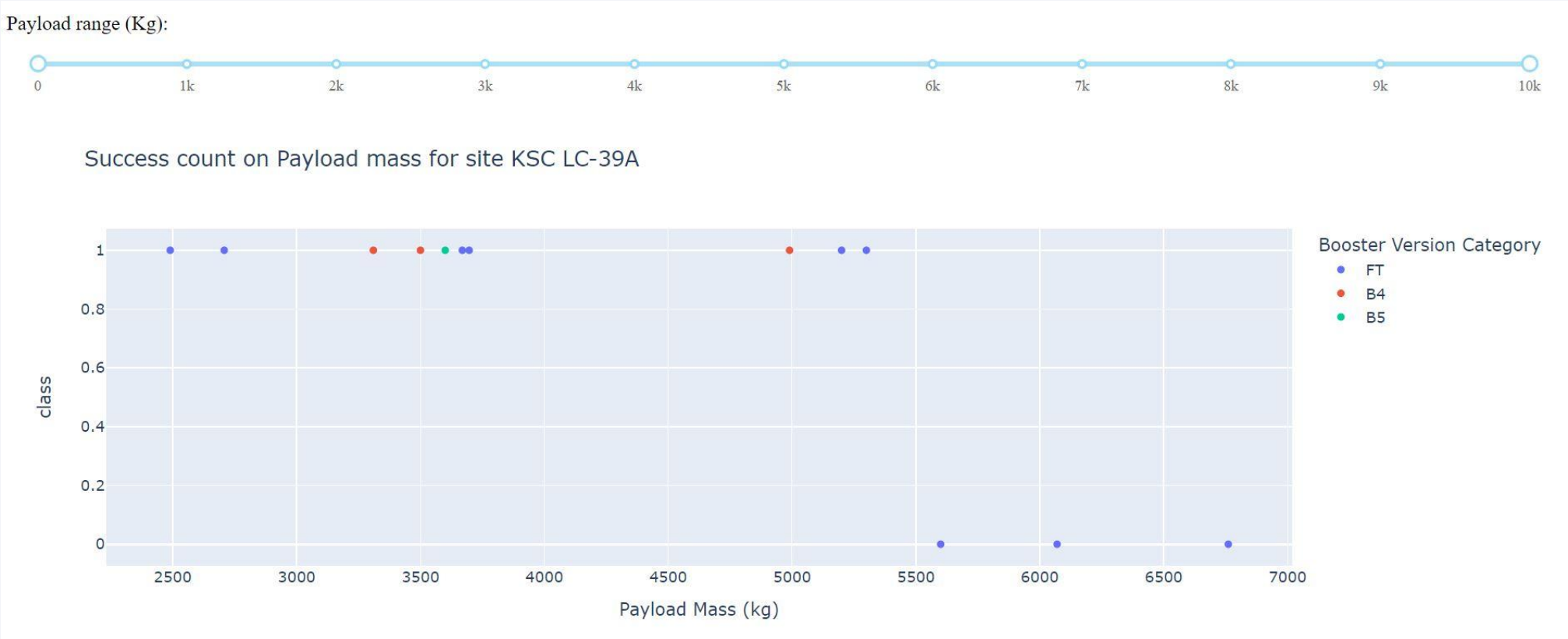
This image displays the success count pie chart on our dashboard. As we can see the KSC LS-39A has the highest success count in all sites.

Success Count Pie chart for KSC LC-39A



This image displayed the success count for KSC LC-39A which is a 10:3 success to failure ratio

Payload to Success Count Scatter Plot



This image shows a scatter plot that displays the outcomes of different payloads with a slider that can change the payload range

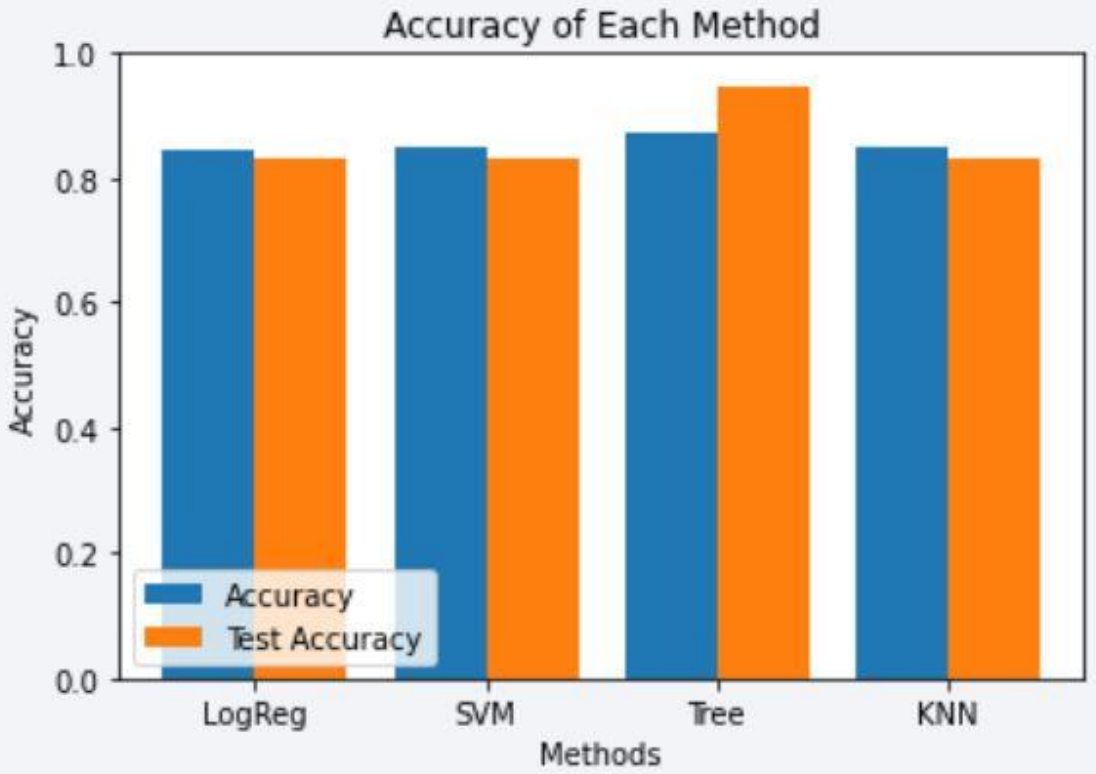
Section 5

Predictive Analysis (Classification)

Classification Accuracy

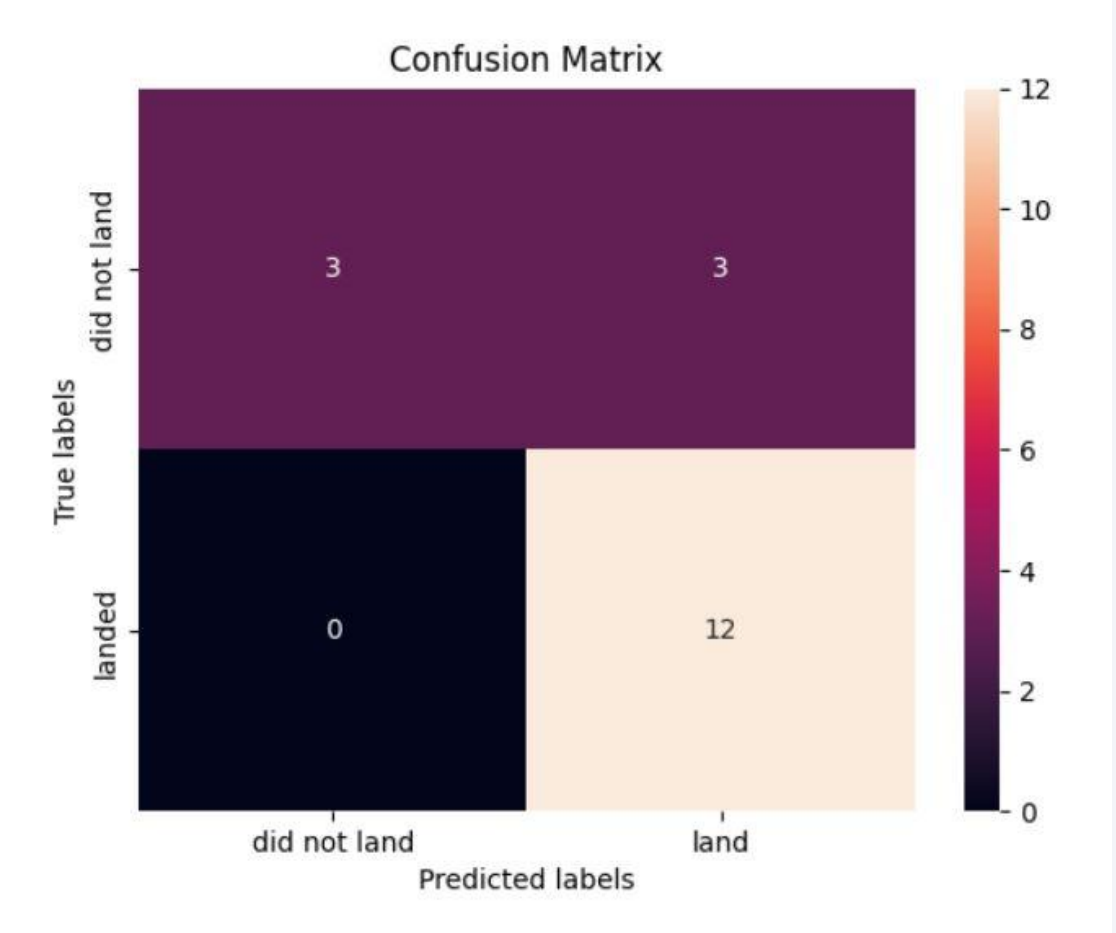
We used 4 classification methods: Logistica regression, SVM, Decision Trees, and KNN.

We can tell from this that the Decision Tree has the highest accuracy.



Confusion Matrix

This is the confusion matrix for the decision tree classifier which had the highest accuracy score



Conclusions

★ The best launch site was KSC LC 39A

★ Launches above 7000 kg are less risky

★ Mission launches are constantly improving with time

★ Decision tree classifier is the best predictive model for this data

Appendix

- Folium notebook didn't show map images so I ran the program again to get the screenshots
  - Plotly application also had to be rerun for the screenshots
- Some of the exploratory data analysis plots were incorrect so I went back in to redo them

Thank you!