IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Zackary Ashworth
07/27/24

Outline

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

Executive Summary

Summary of methodologies

- Models:
  - Created and tested classification models using Sklearn
- Interactive Mediums:
  - Folium maps and Dash web-application
- EDA:
  - Visualized with Seaborn and explored with SQL
- Data Wrangling:
  - Processed using Pandas
- Data Collection:
  - SpaceX API and Web Scraping Wikipedia

Summary of all results:

- Classification of successful recovery at 87% accuracy

- Goals:

. Determine if a SpaceX first stage Rocket will land

Introduction

- Reusing first stage rockets saves a lot of money
- Determining if a spaceX rocket will or will not land will determine how much more or less expensive that launch is
- What features are the biggest influence on a successful recovery?
- What level of accuracy can we expect to predict for any given launch?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Collected using the SpaceX API and web-scraping wikipedia
- Perform data wrangling
  - Processed using Python and Pandas
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Evaluated with Scikit-learn, and several classification models: SVM, Logistic Regression, Decision Tree, and K-nearest neighbors.

Data Collection

Data Sets were collected via SpaceX API and via web-scraping Wikipedia

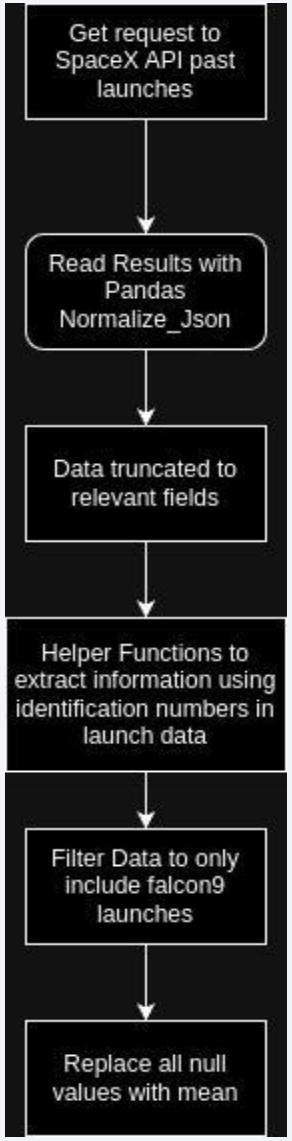
Some data from the SpaceX API needed to be processed separately and added to the dataframe using specially built helper functions.

Data Scraped from Wikipedia had to be processed using the BeautifulSoup library to extract relevant tabular data

Data Collection – SpaceX API

- Data from SpaceXAPI: specifically the v4 past launches
  - "https://api.spacexdata.com/v4/launches/past"

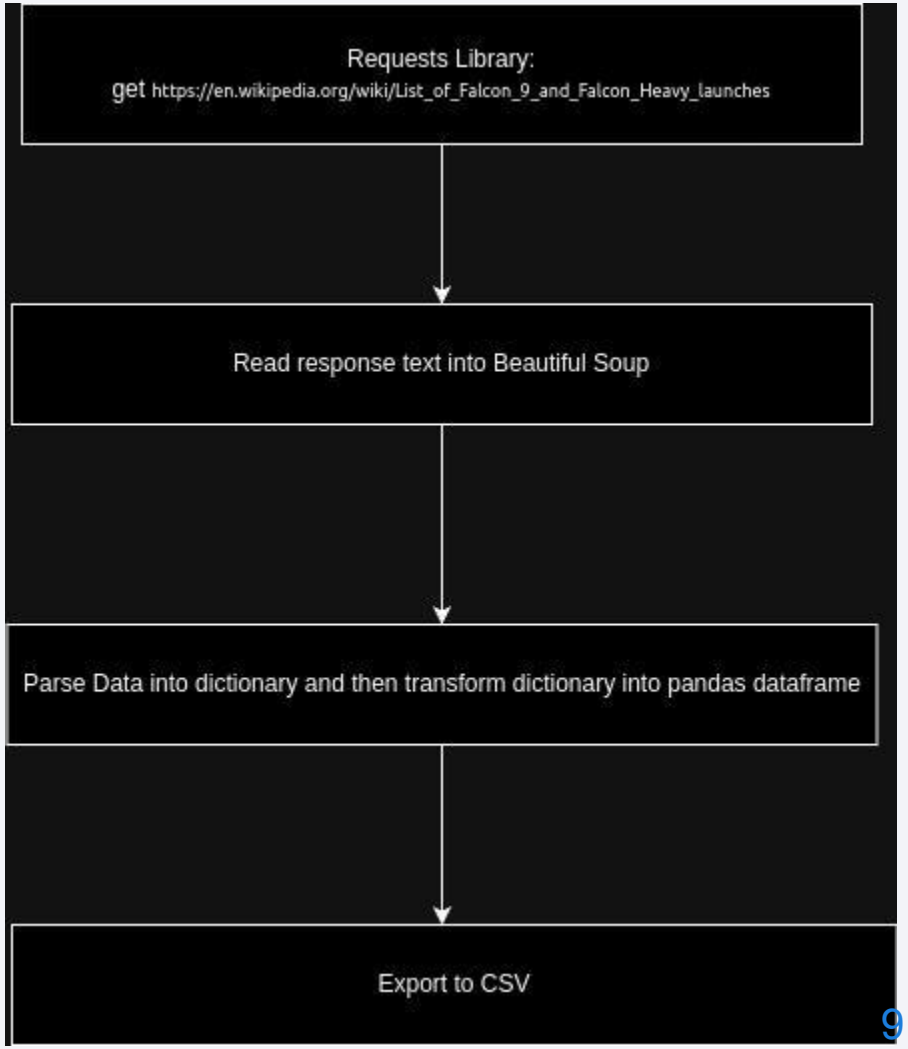
https://github.com/Vocantes/Data_Science_Captstone_IBM/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Data was scraped from wikipedia and then processed using the requests and BeautifulSoup Libraries.
  - ★ https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

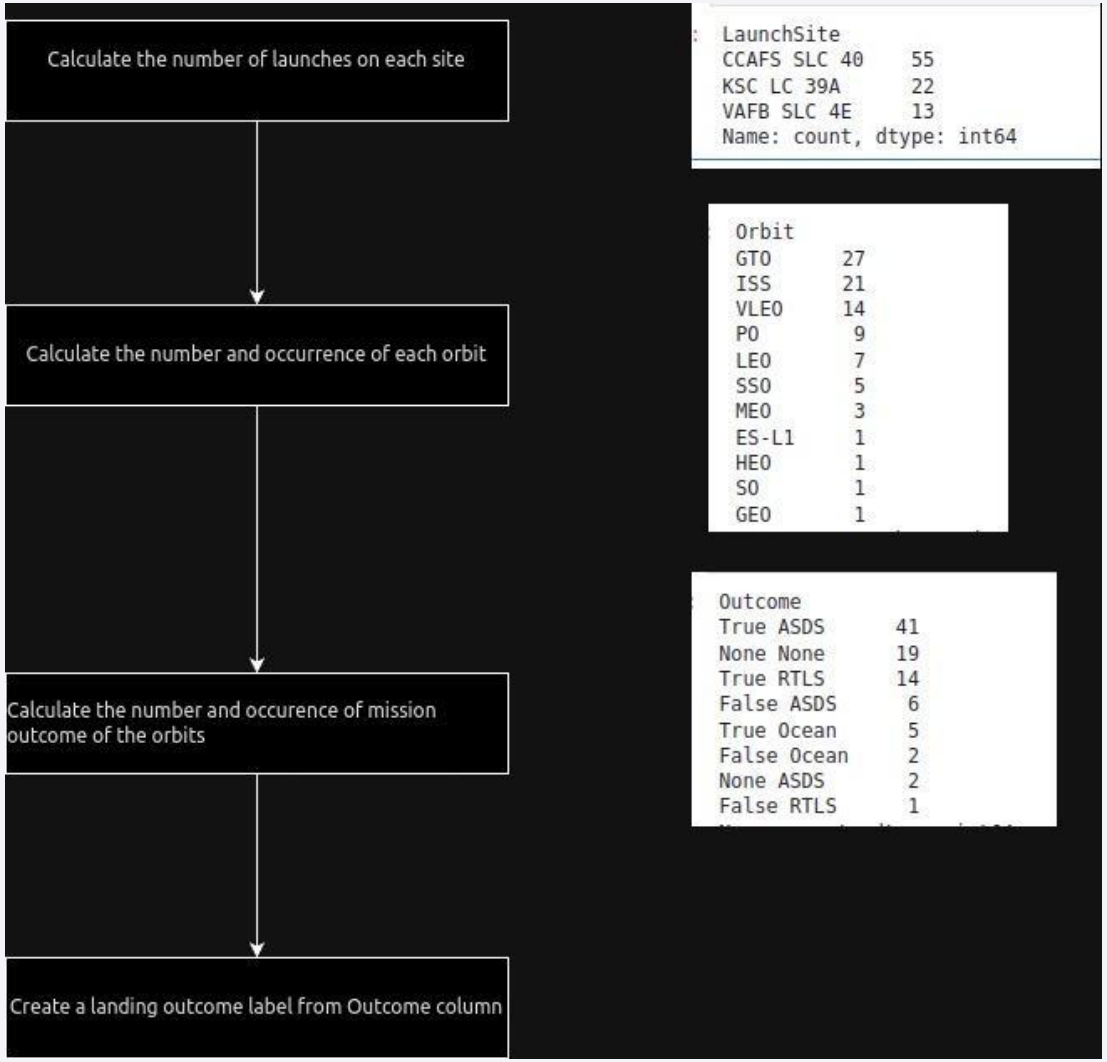
- https://github.com/Vocantes/Data_Science_Captstone_IBM/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

Data was processed by using basic pandas functions and applying filters.

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



EDA with Data Visualization

In the visual EDA of this project several charts were plotted:

-Flight Number and Launch Site

- Payload and Launch Site
- Success Rate of each Orbit Type
- Flight Number and Orbit Type
- Payload Mass and Orbit Type

https://github.com/Vocantes/Data_Science_Captstone_IBM/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

There were several SQL Queries performed on the data:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first succesful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- 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.
- 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

https://github.com/Vocantes/Data_Science_Captstone_IBM/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

The Folium Map had markers added to denote certain locations:

- The launch sites
- The successful and failed launches
- The distance between launch sites and it's proximities, such as:
  - railways
  - highways
  - coastlines

These were added in order to determine if there was a geographic pattern in launch site locations.

https://github.com/Vocantes/Data_Science_Captstone_IBM/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

A few interactive graphs and plots were created using Plotly and Dash:

- ➔ A pie chart of total successes filtered either by all sites or by specific sites
- ➔ A scatter chart plotting payload weight vs successful launches filtered by all sites or specific sites; it is also filtered via range of payload weight

These were chosen to quickly view the effects that specific launch sites and payload weight has on successful launches.

https://github.com/Vocantes/Data_Science_Captstone_IBM/blob/main/spacex_dash_app.py

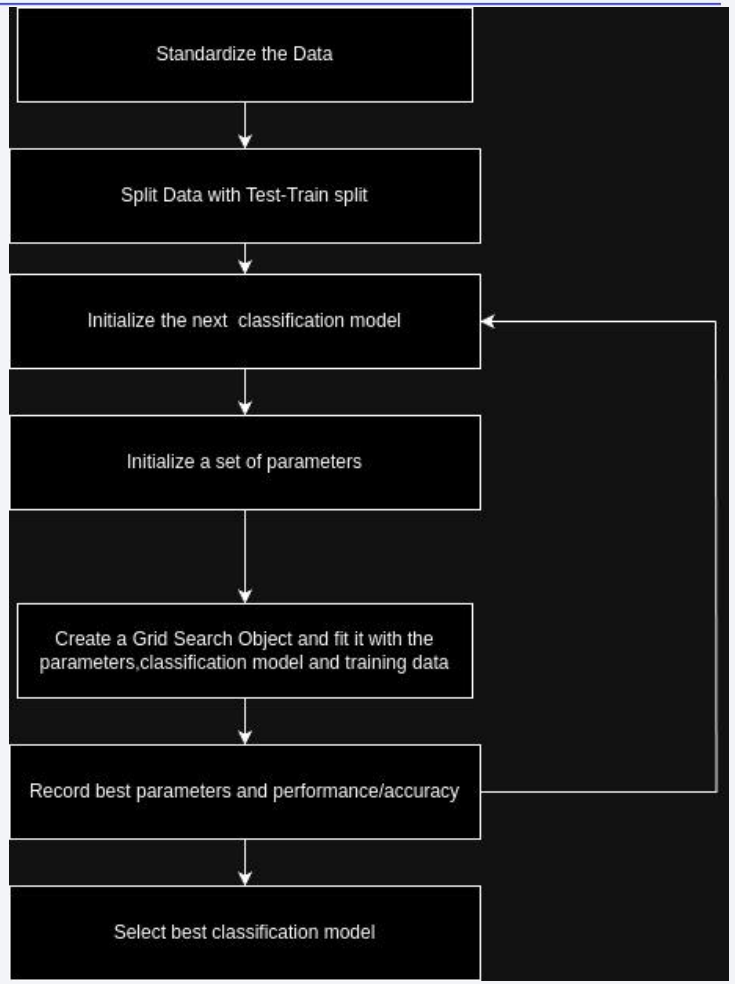
Predictive Analysis (Classification)

Summarize how you built, evaluated, improved, and found the best performing classification model
The classification model was built, evaluated and improved using a test train split and Grid Search on several different classification models:

-Logistic Regression
-Search Vector Machine
- Decision Tree Classifier
-K-nearest neighbors

The best performing classification model appears to be a decision tree classifier with an accuracy score of 87.67%

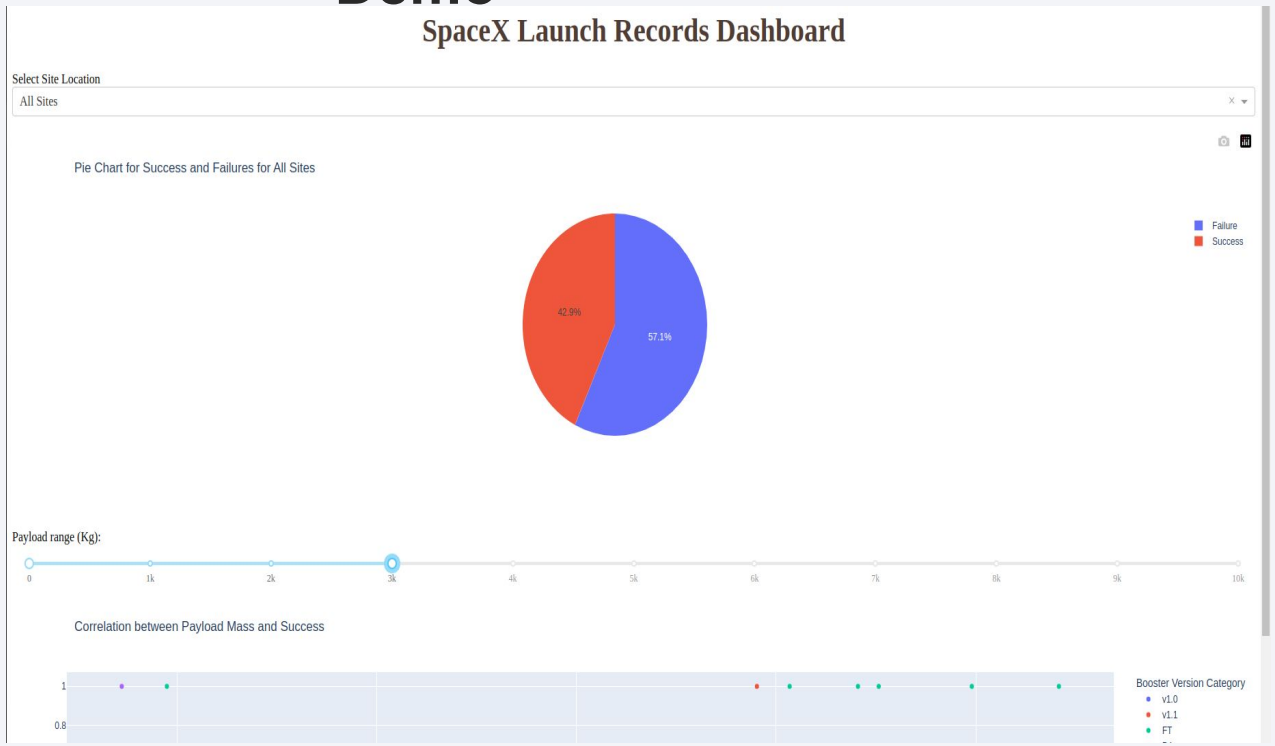
https://github.com/Vocantes/Data_Science_Captstone_IBM/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

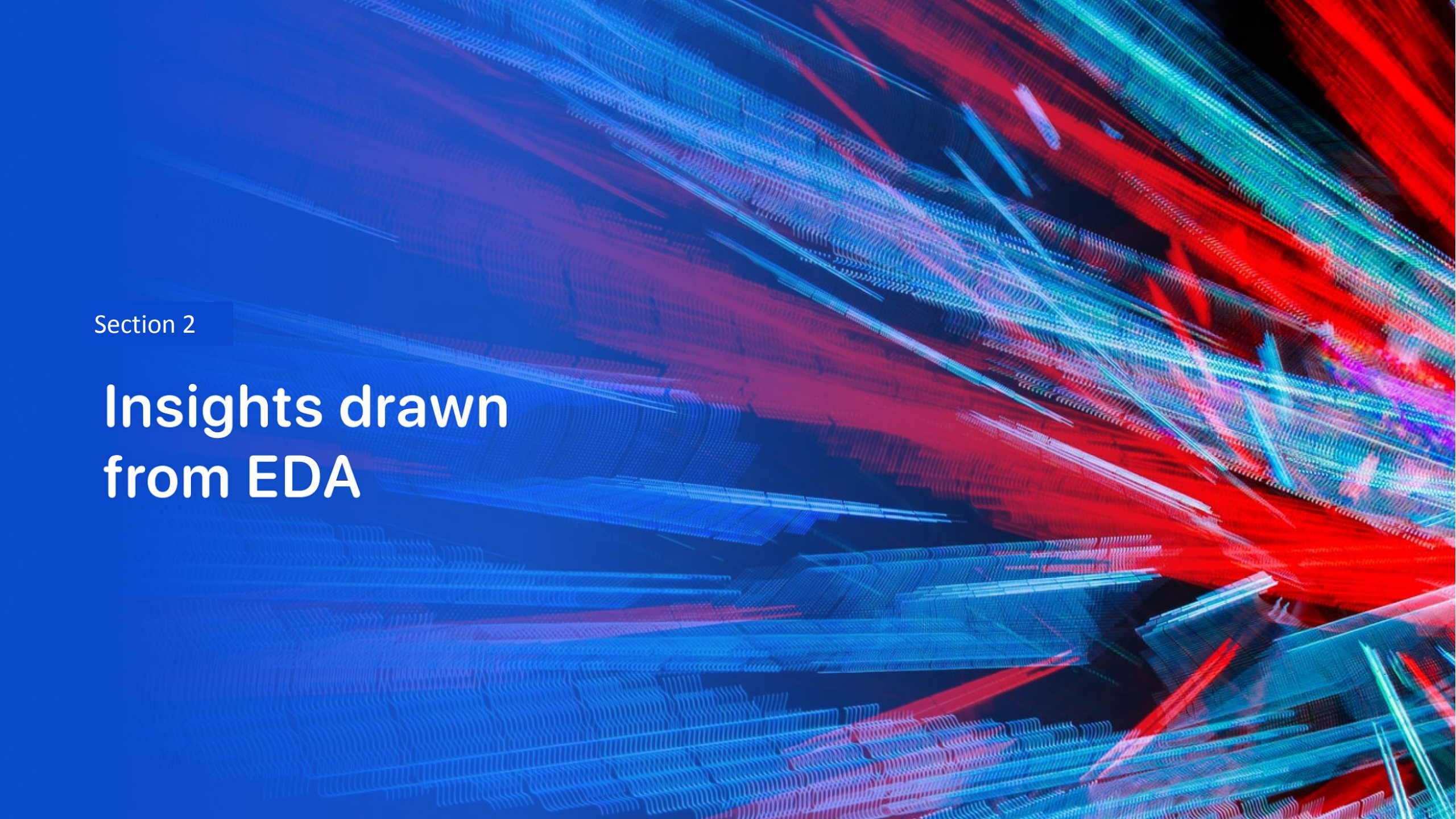


Results

- Exploratory data analysis results
  - Relevant fields:
    - 'FlightNumber', 'PayloadMass', 'Orbit', 'LaunchSite', 'Flights', 'GridFins', 'Reused', 'Legs', 'LandingPad', 'Block', 'ReusedCount', 'Serial'
- Predictive analysis results
  - Decision Tree classification model:
    - 87.67% Accuracy

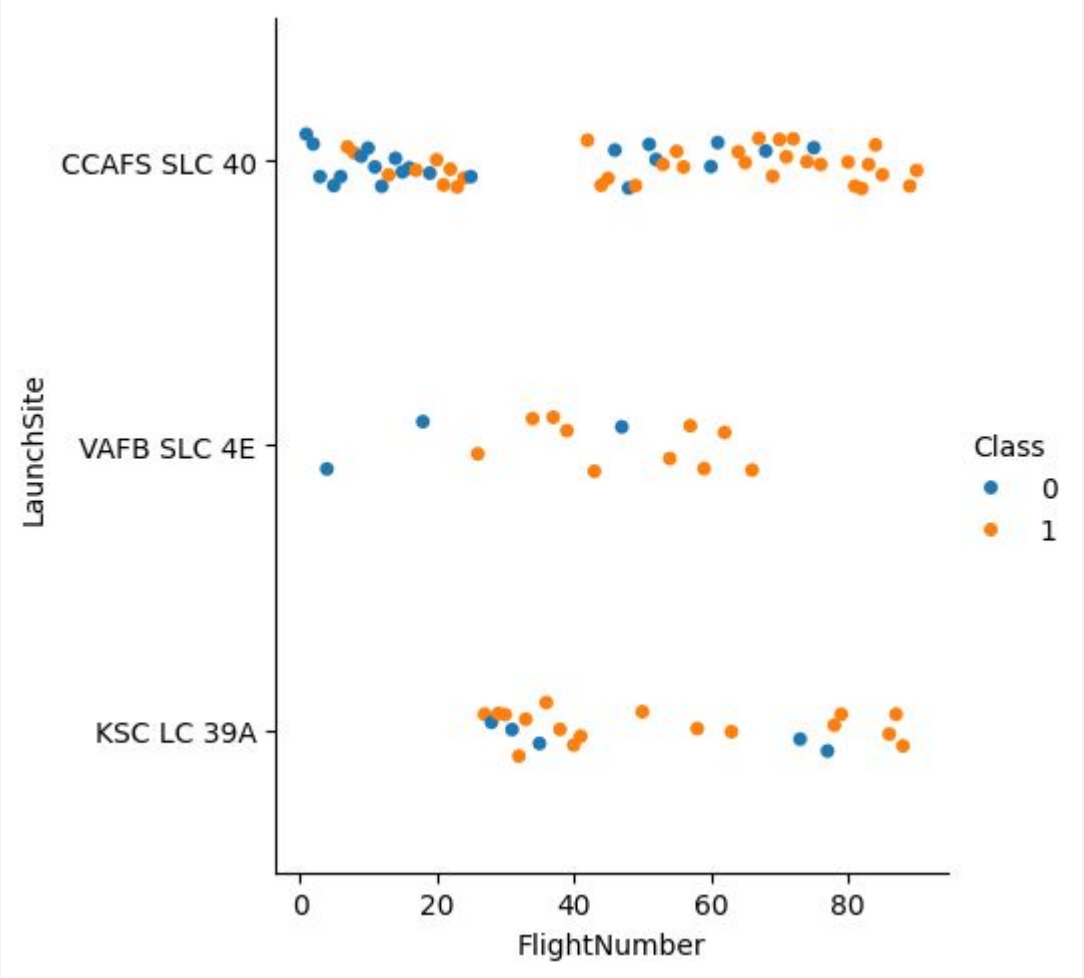
Interactive Analytics Demo




Section 2
Insights drawn from EDA

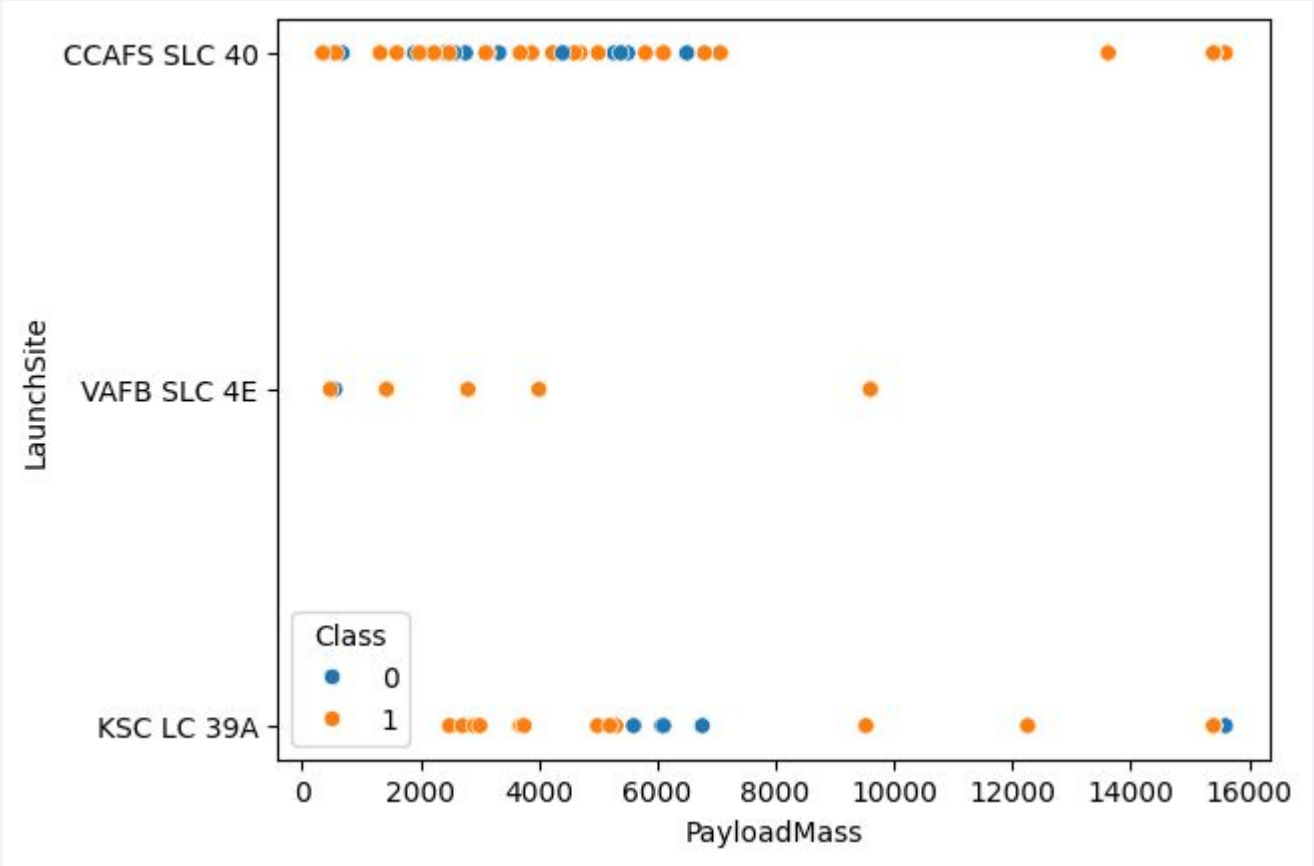
Flight Number vs. Launch Site

- Generally, as the flight number increases the launch site's success rate goes up



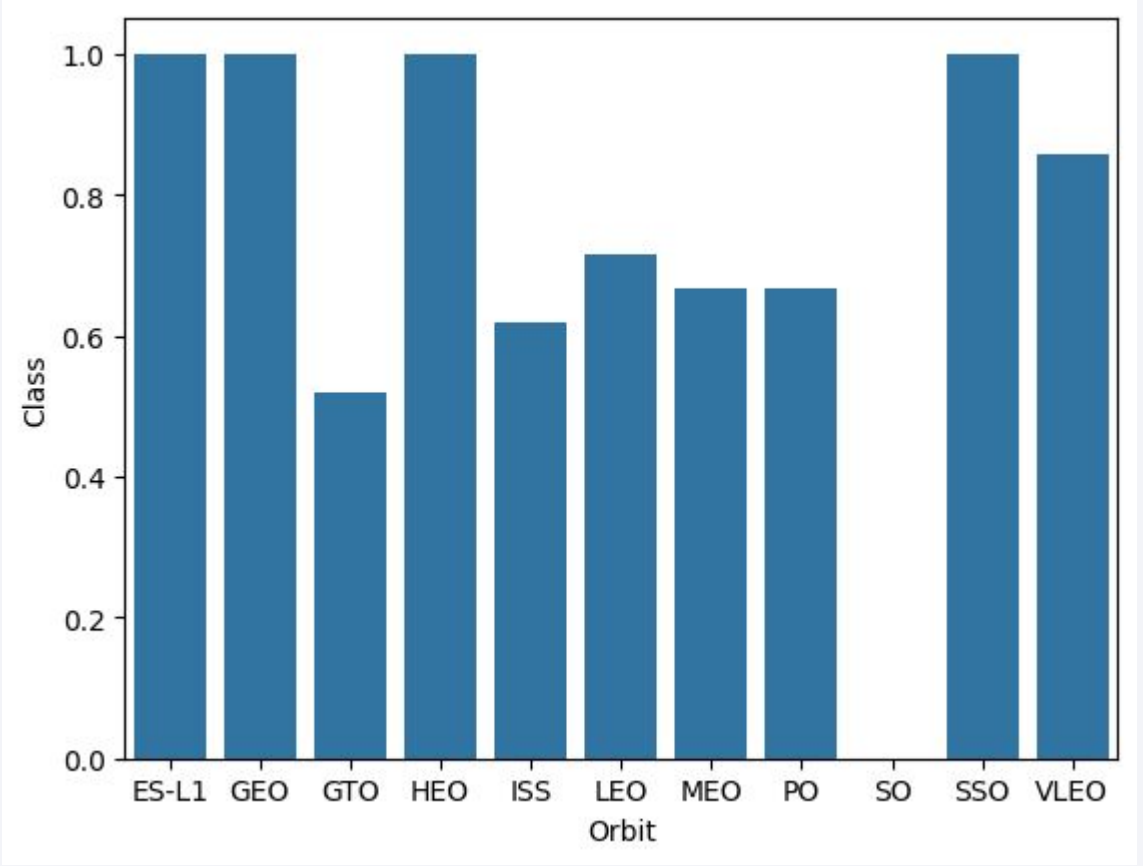
Payload vs. Launch Site

- The launch site VAFB SLC 4E appears to not launch heavy payloads at all and most payloads from all sites are less than 10000



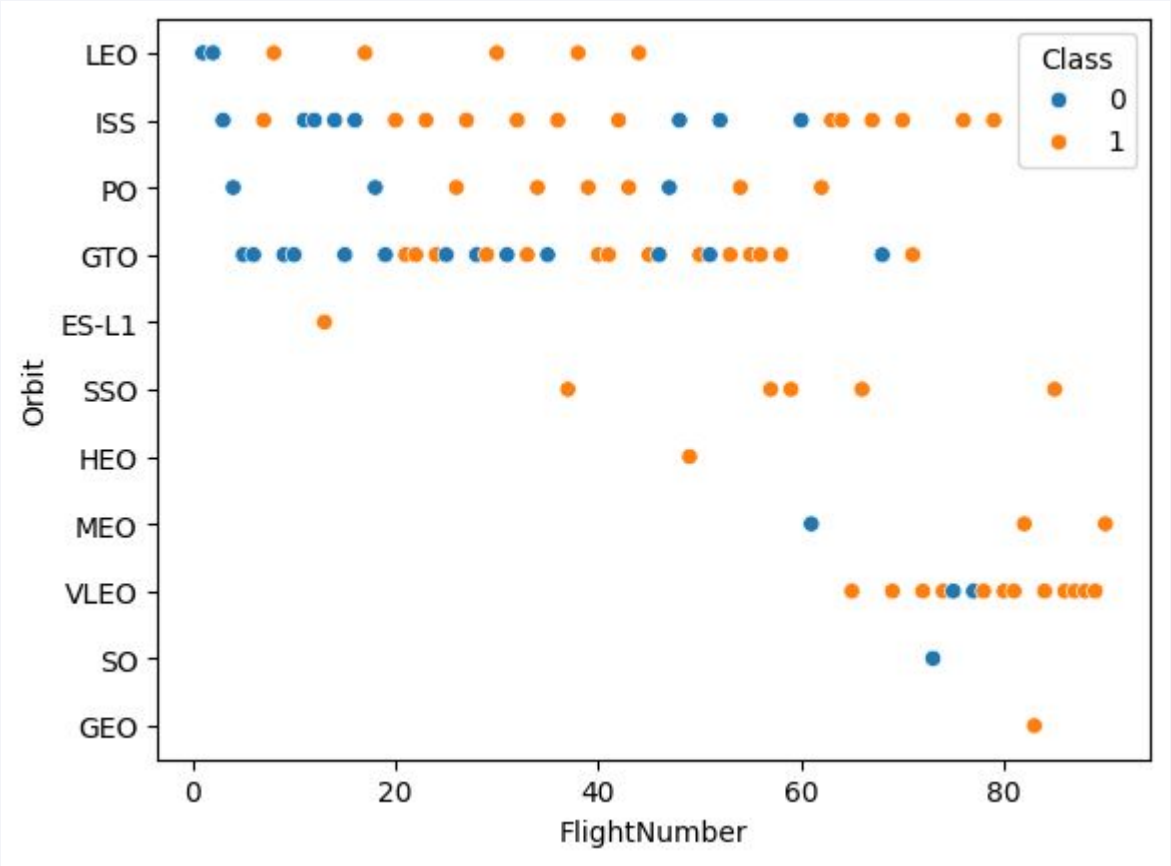
Success Rate vs. Orbit Type

- Certain Orbits tend towards higher success rates and others towards lower.
- There is a lack of data on certain orbits that may make the data appear more favorable than in practice.



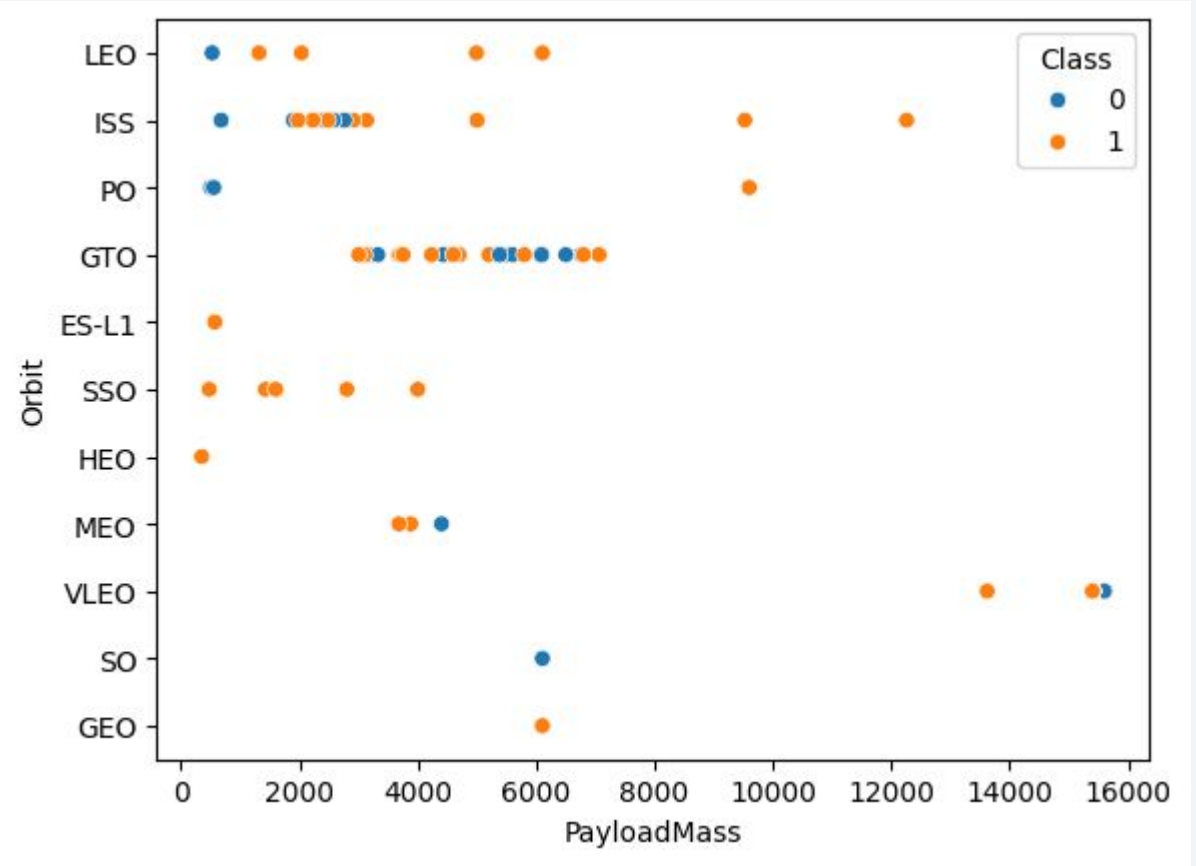
Flight Number vs. Orbit Type

- For some Orbits such as LEO success rates increase as Flight Numbers increase
- For other Orbits the Flight Number appears to have little to no correlation to success rate



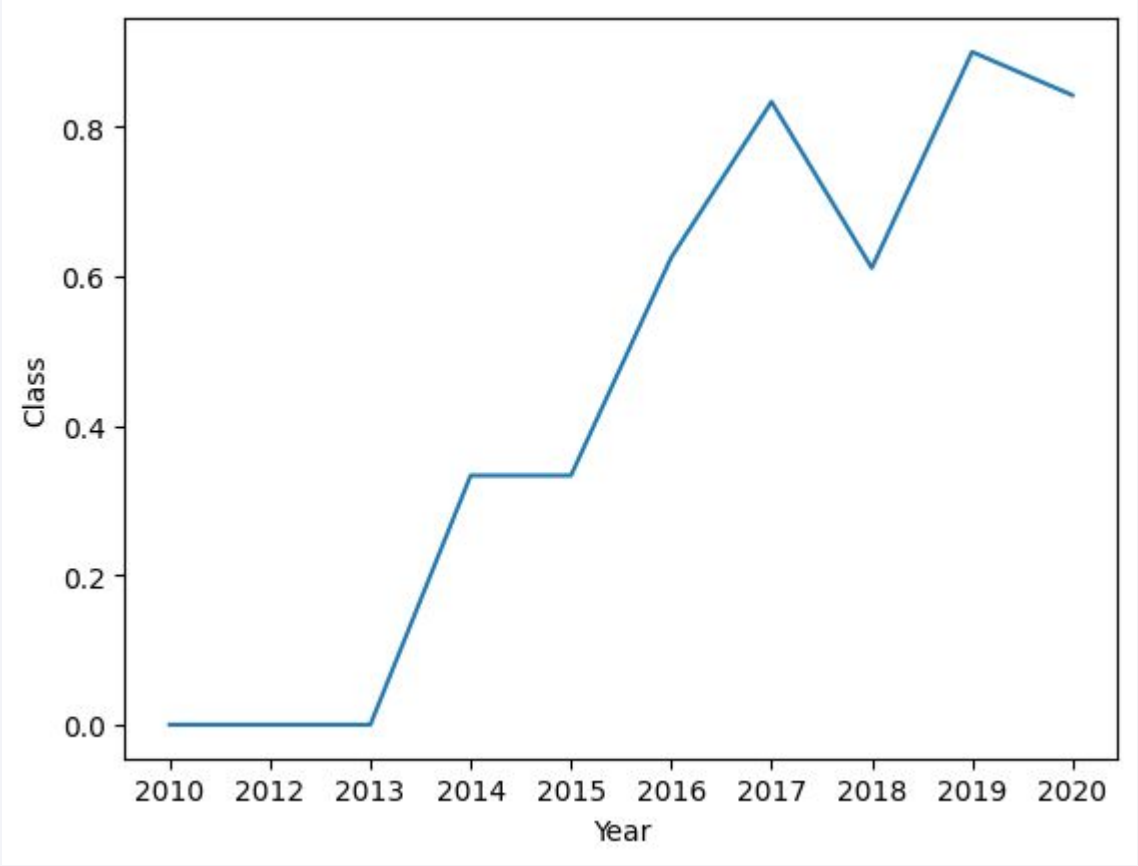
Payload vs. Orbit Type

- The Heavier Payloads tend to only be sent out on certain orbits: ISS, PO, VLEO
- PO and ISS orbits are more likely to succeed for Heavy Payloads



Launch Success Yearly Trend

- As years go by, the success rate of Launches tend to increase with a dip in 2018 and 2020



All Launch Site Names

Query:

“%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE “

Result:

Launch_Site

CCAFS LC-40

VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

This query gets all launch site values and filters them down to only the unique values

Launch Site Names Begin with 'CCA'

Query:
“%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE "CCA%" LIMIT 5;”

This query selects the first 5 rows from the database where launch sites start with CAA

Result:

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

Query:

“%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Customer" LIKE "NASA (CRS)";”

Result:

SUM("PAYLOAD_MASS__KG_")
45596

This query filters the customer to be NASA (CRS) and then sums up all values in the Payload mass KG column.

Average Payload Mass by F9 v1.1

Query:

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE
WHERE "Booster_Version" = "F9 v1.1" GROUP BY "Booster_Version"
```

Result:

AVG("PAYLOAD_MASS__KG_")
2928.4

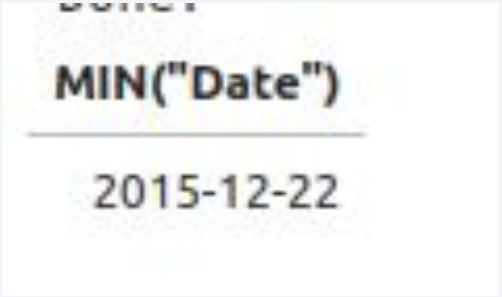
This query filters the results to where booster_version is F9 v1.1 and then Averages the result of the payload mass column.

First Successful Ground Landing Date

Query:

```
%sql SELECT MIN("Date") FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE "Success (ground pad)";
```

Result:



MIN("Date")
2015-12-22

This Query filters the results to where the landing outcome is a successful ground pad and then selects the minimum value from the Date column.

Successful Drone Ship Landing with Payload between 4000 and 6000

Query:

```
“%sql SELECT "Booster_Version" FROM
SPACEXTABLE WHERE "Landing_Outcome" =
"Success (drone ship)"  AND
"PAYLOAD_MASS__KG_" > 4000 AND
"Payload_MASS__KG_" < 6000;”
```

Result:

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

The Query selects Booster versions where the Landing outcome is a successful drone ship and where their payload mass is lower than 6000 and greater than 4000

Total Number of Successful and Failure Mission Outcomes

Query:

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") FROM
SPACEXTABLE GROUP BY "Mission_Outcome";
```

Result:

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

This query selects and groups the mission outcomes into distinct categories and then counts the number of entries in each category.

Boosters Carried Maximum Payload

Query:

```
%sql SELECT DISTINCT("Booster_Version")
FROM SPACEXTABLE WHERE
"PAYLOAD_MASS__KG_" = (SELECT
MAX("PAYLOAD_MASS__KG_") FROM
SPACEXTABLE)
```

This query filters the data to include only results that are equal to a subquery that computes the maximum payload mass. Then it outputs only the Booster_versions that have delivered payloads of that weight

Result

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

2015 Launch Records

Query:

```
%%sql SELECT SUBSTR("Date",6,2) AS "Month",
"Landing_Outcome", "Booster_Version", "Launch_Site"

FROM SPACEXTABLE

WHERE "Landing_Outcome" LIKE "Failure%" and
SUBSTR("Date",0,5) = "2015"

ORDER BY "Month"
```

Result

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

This query gets the month from the date column, the Landing outcome, Booster version and Launch site location from rows where the landing outcome is some form of failure and the year in the date column is 2015.

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

Query:

```
%sql SELECT "Landing_Outcome",
COUNT("Landing_Outcome") AS "Outcome Count" FROM
SPACEXTABLE WHERE DATETIME("Date") >
DATETIME("2010-06-04") AND DATETIME("Date") <
DATETIME("2017-03-20") GROUP BY "Landing_Outcome"
ORDER BY "Outcome Count" DESC;
```

Result

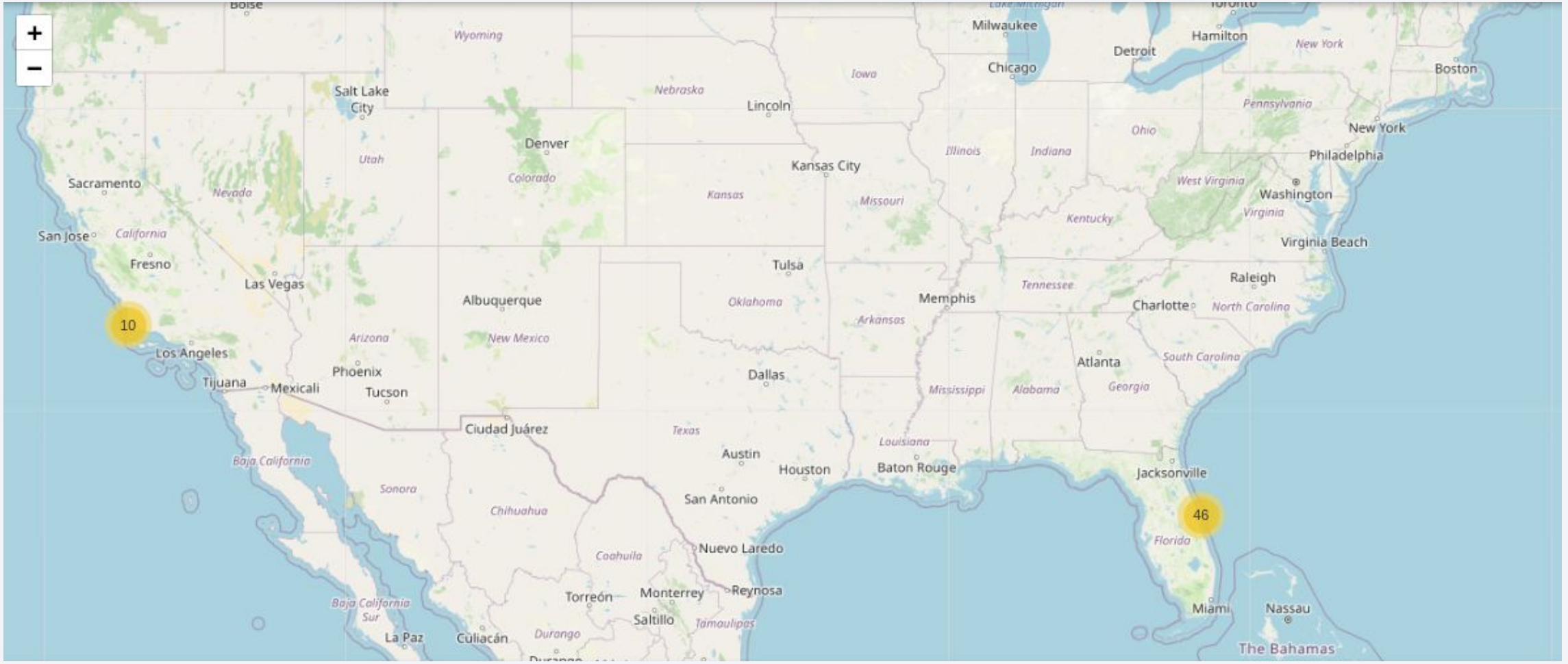
Landing_Outcome	Outcome Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

This query filters the rows to only be between 2010-06-04 and 2017-03-20. It then Groups the results by their Landing Outcome and counts the number of rows in each outcome. Then it orders the results in descending order by their counts.


Section 3
Launch Sites
Proximities Analysis

Folium Map of All Launch Site locations

This Map Shows all the locations that SpaceX launches occur

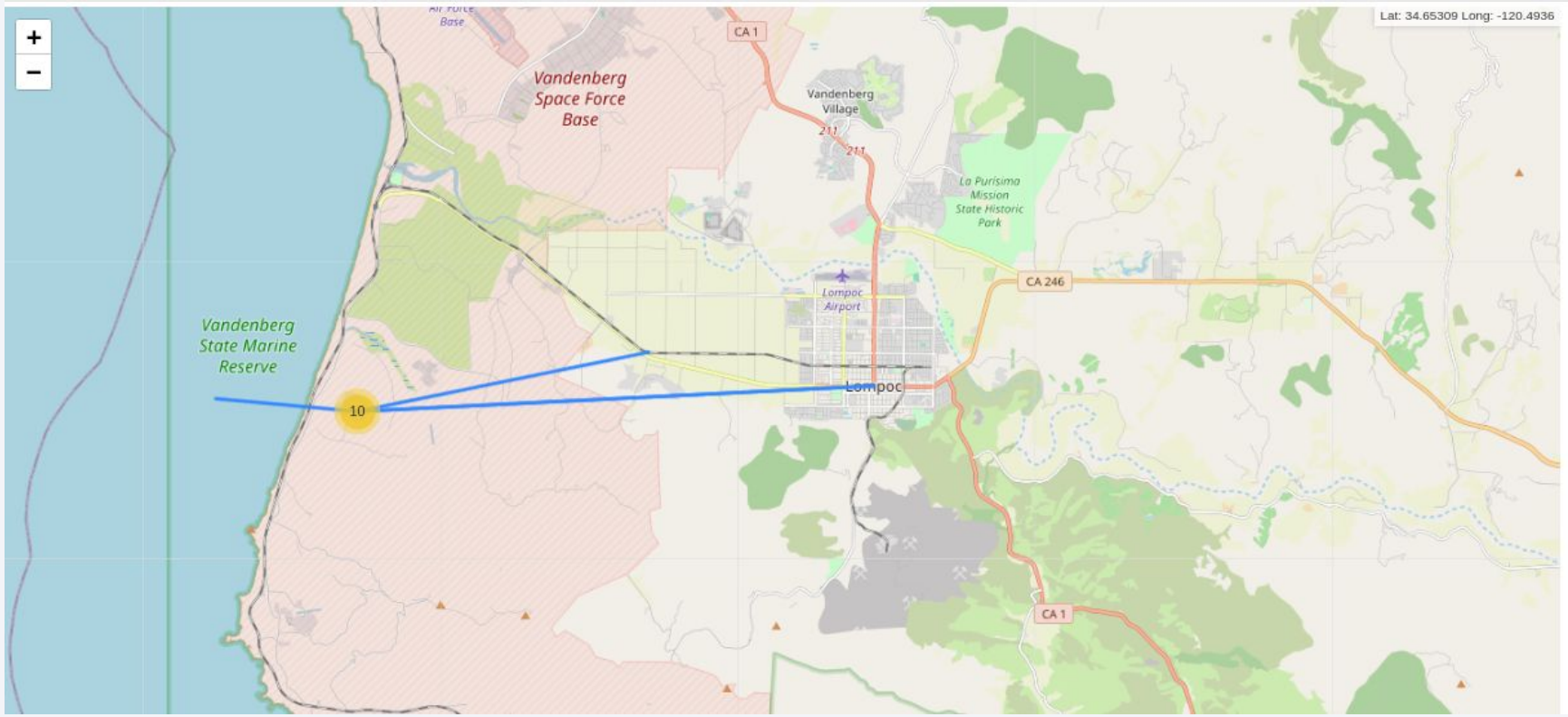


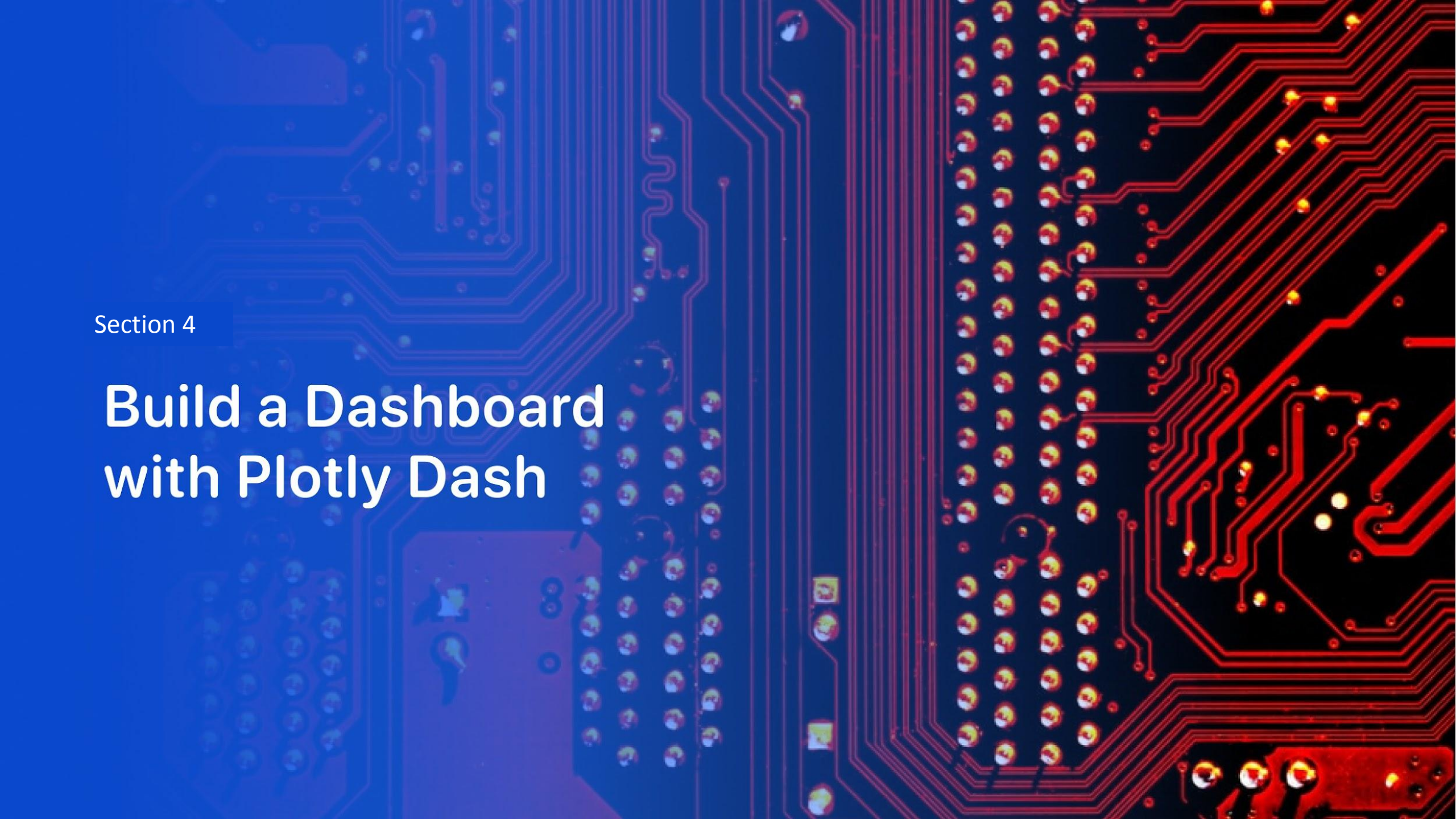
Folium Map of Launch Outcomes

This map color codes the result of each launch from each location. It helps visualize which launch sites tend towards more successful launches.

Folium Map of Proximities

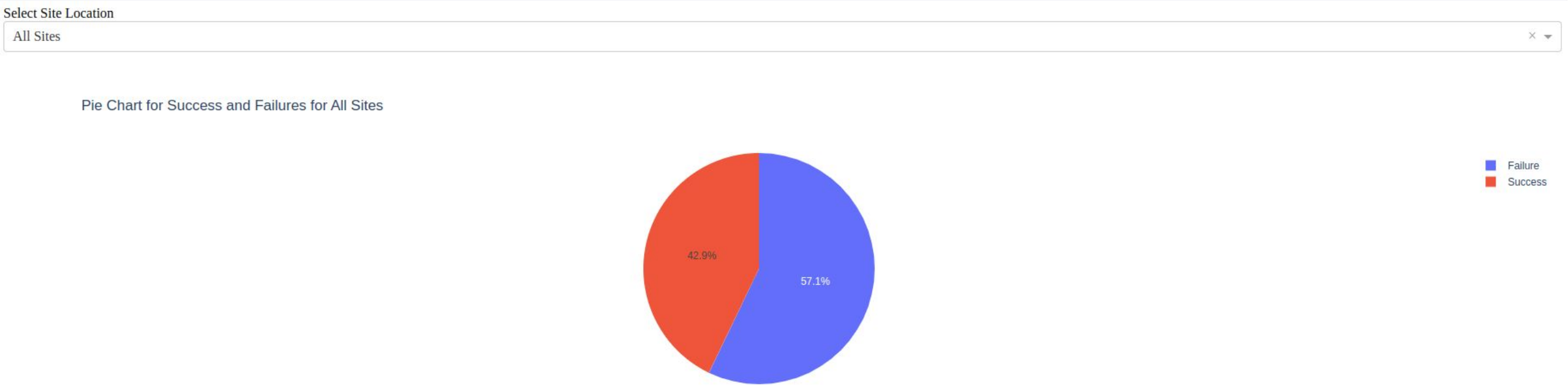
This Folium Map shows the proximity of highways, coastlines and railways near the california launch site. As you can see all of those things are fairly close by




Section 4
Build a Dashboard with Plotly Dash

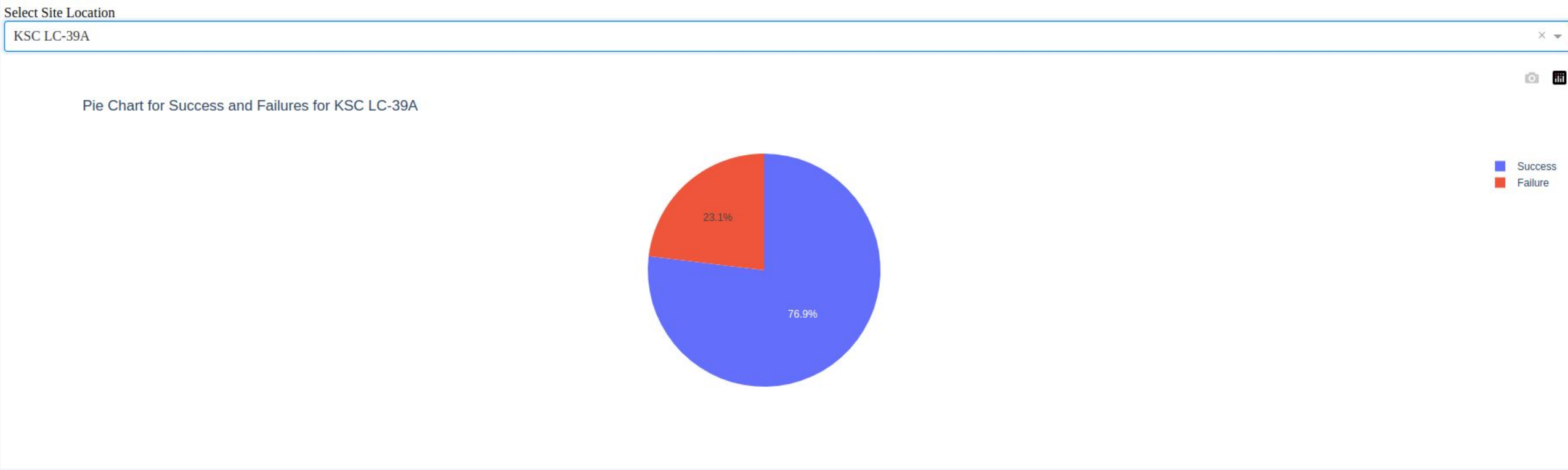
Pie chart of Successful Launches for All Sites

This Pie chart shows that the majority of launches from all sites are failures

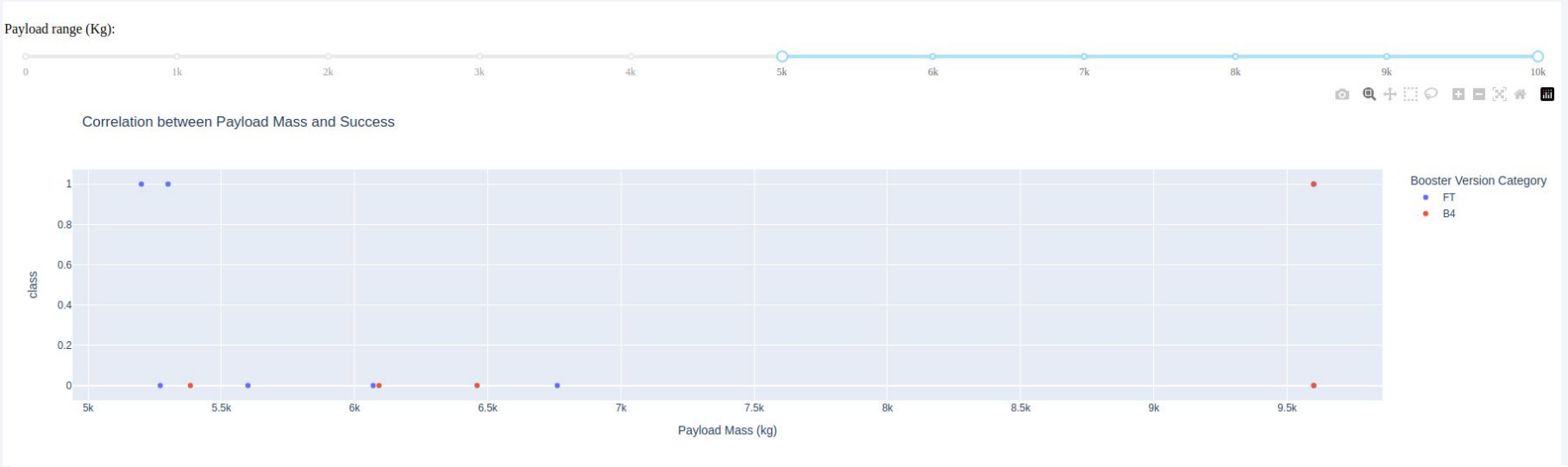


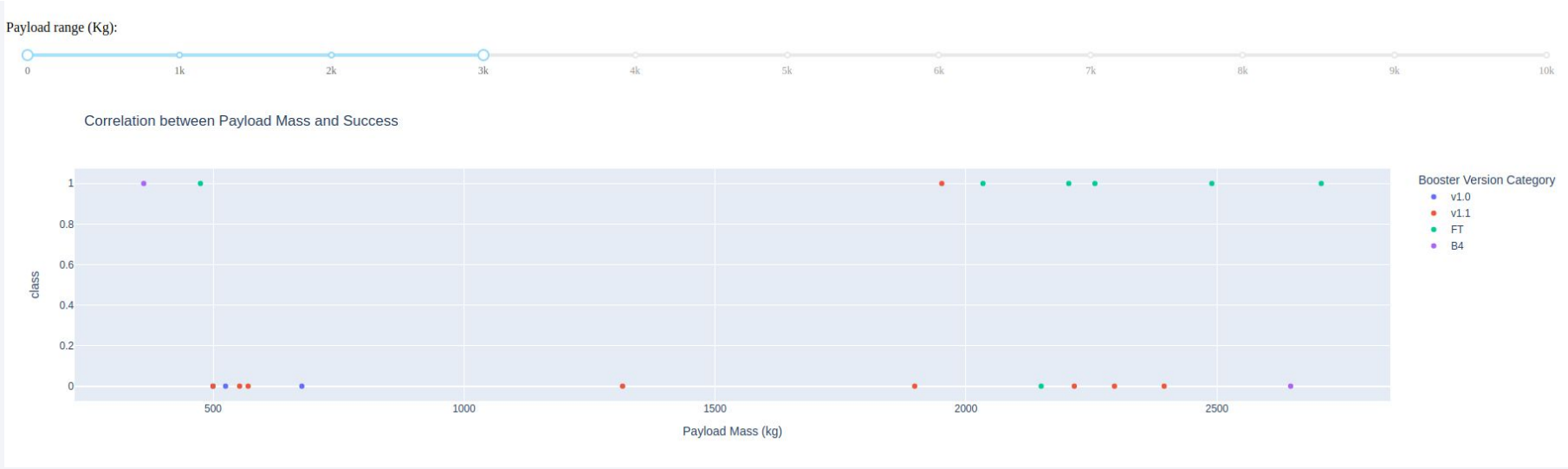
Pie chart of launch site with highest launch success ratio

This pie chart shows that there is a significantly larger amount of success than failures from KSC LC-39A than from other locations or the average.



Payload vs Launch Outcome Scatter Plots





In the high end of payload Mass, the failure rate goes up and only B4 boosters have been used for the highest weight payloads

At the low end FT boosters tend to have the highest success rate and 1.0 boosters have no successes

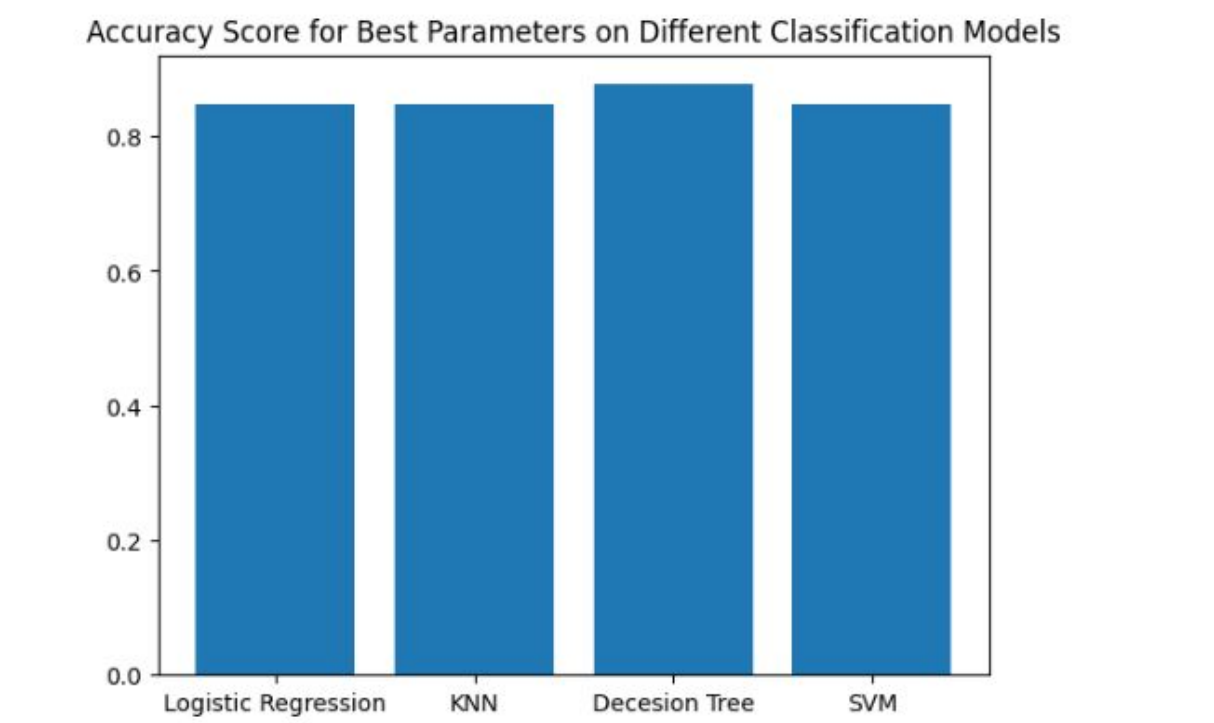
Section 5

Predictive Analysis (Classification)

Classification Accuracy

The Decision Tree has the best Accuracy between all classification models tested at:

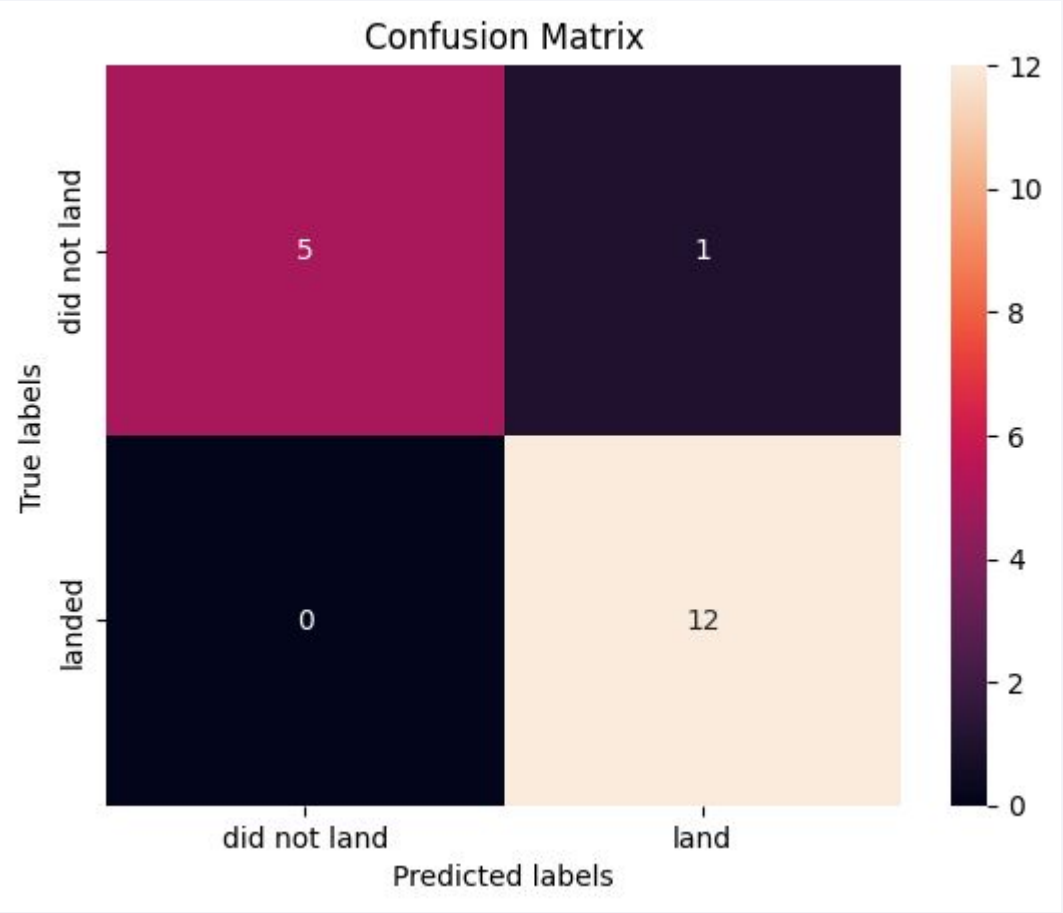
87.67% Accuracy



Confusion Matrix

- The Confusion Matrix of the Decision Tree using the testing data.

- Only one classification was incorrectly classified predicting that it did not land when in fact it did.



Conclusions

- There are several key fields that play a large role in determining whether or not a launch will be successful:
  - 'FlightNumber', 'PayloadMass', 'Orbit', 'LaunchSite', 'Flights', 'GridFins', 'Reused', 'Legs', 'LandingPad', 'Block', 'ReusedCount', 'Serial'
- The best classification model tested was the Decision Tree Classifier with 87.67% accuracy
- SpaceX has been trending towards more successful launches as time goes on

Appendix

```
[40]: spacex_df[(spacex_df["Payload Mass (kg)"] > 500) & (spacex_df["Payload Mass (kg)"] < 3000)]
```

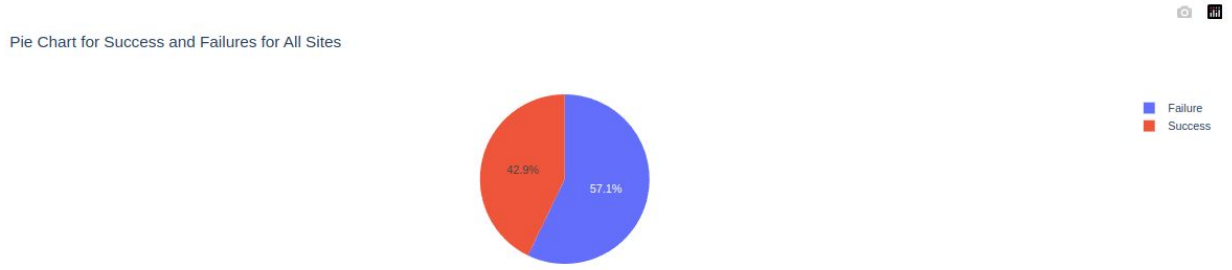
[40]:

	Unnamed: 0	Flight Number	Launch Site	class	Payload Mass (kg)	Booster Version	Booster Version Category
2	2	3	CCAFS LC-40	0	525.0	F9 v1.0 B0005	v1.0
4	4	5	CCAFS LC-40	0	677.0	F9 v1.0 B0007	v1.0
7	7	9	CCAFS LC-40	0	2296.0	F9 v1.1	v1.1
8	8	10	CCAFS LC-40	0	1316.0	F9 v1.1	v1.1
11	11	13	CCAFS LC-40	0	2216.0	F9 v1.1 B1010	v1.1
12	12	14	CCAFS LC-40	0	2395.0	F9 v1.1 B1012	v1.1
13	13	15	CCAFS LC-40	0	570.0	F9 v1.1 B1013	v1.1
15	15	17	CCAFS LC-40	0	1898.0	F9 v1.1 B1015	v1.1
17	17	19	CCAFS LC-40	1	1952.0	F9 v1.1 B1018	v1.1
18	18	20	CCAFS LC-40	1	2034.0	F9 FT B1019	FT
24	24	27	CCAFS LC-40	1	2257.0	F9 FT B1025.1	FT
27	27	21	VAFB SLC-4E	0	553.0	F9 v1.1 B1017	v1.1
33	33	49	VAFB SLC-4E	0	2150.0	F9 FT B1038.2	FT
36	36	30	KSC LC-39A	1	2490.0	F9 FT B1031.1	FT
41	41	35	KSC LC-39A	1	2708.0	F9 FT B1035.1	FT
49	49	45	CCAFS SLC-40	1	2205.0	F9 FT B1035.2	FT
53	53	52	CCAFS SLC-40	0	2647.0	F9 B4 B1039.2	B4

```
[ ]:
```

```
[21]: pie_df = spacex_df["class"].value_counts().to_frame()
```

```
[31]: px.pie(pie_df,values="count",names=["Failure","Success"],title="Pie Chart for Success and Failures for All Sites")
```



Thank you!