IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Zhaoyu Bai
Feb. 17th 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result from Machine Learning Lab

Introduction

- **Space Exploration Technologies Corp.,** commonly known as SpaceX, is an American aerospace company headquartered at the Starbase development site near Brownsville, Texas. Since its founding, the company has made multiple attempts to successfully re-land spacecraft.
- **The goal of this study** is to identify the key factors that most significantly influence landing outcomes, providing guidance for future attempts and ultimately leading to the commercialization of space travel.
- **Problems to be answered:**
  - Identifying the factors that influence the landing outcome
  - The relationship between each variable and how it affects the outcome
  - The optimal condition for a successful landing and what is the take away for the future attempts

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX REST API and web Scrapping from the Wikipedia
- Perform data wrangling
  - Data was processed using 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
  - Built several Machine Learning Models, including Logistic Regression, SVM, Decision Tree, KNN, etc.

Data Collection

- Data was collected using SpaceX REST API and web Scrapping from the Wikipedia

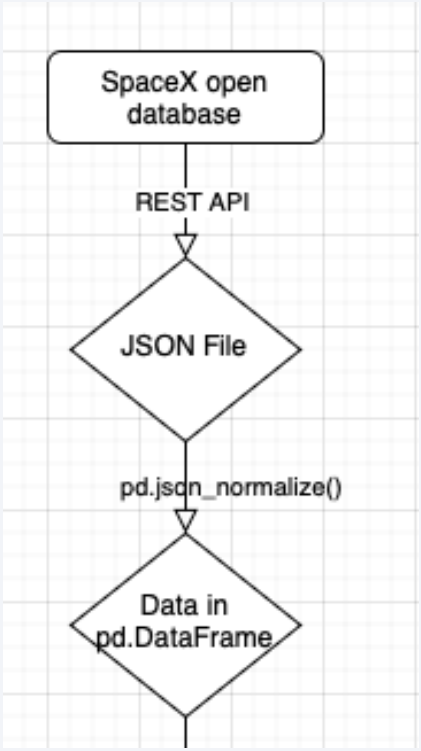
- For REST API, by building a get request. One can encode the response content as .Json file and can future transform it to a pandas data frame using json_normalize().
  - We cleaned the data, checked for missing values, transformed the data to the correct format, cooked needed features, etc.
- For web scraping, we used BeautifulSoup to extract the launch records as an HTML table. We parse the table and convert it into a pandas data frame.
  - As before we cleaned the data and cooked features for further analysis.

Data Collection – SpaceX API

- See details in:
  https://github.com/ZBaiY/Applied_Data_Science_Capstone.git

- File name:

  jupyter-labs-spacex-data-collection-api.ipynb

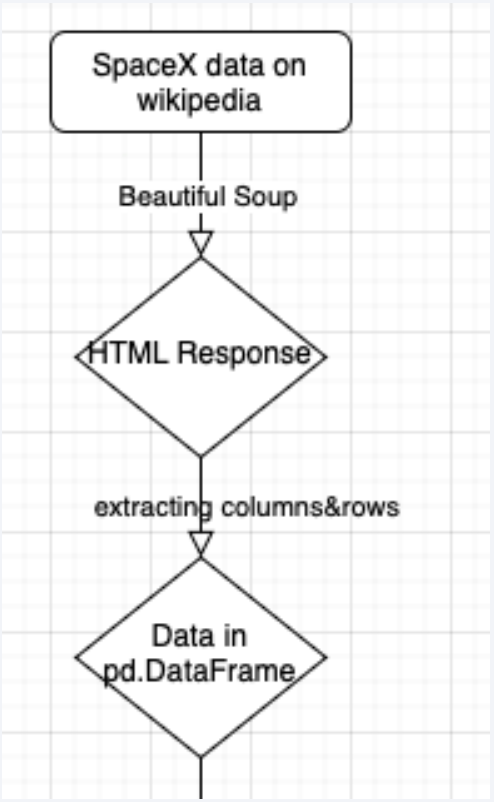


Data Collection - Scraping

- See details in:
  https://github.com/ZBaiY/Applied_Data_Science_Capstone.git

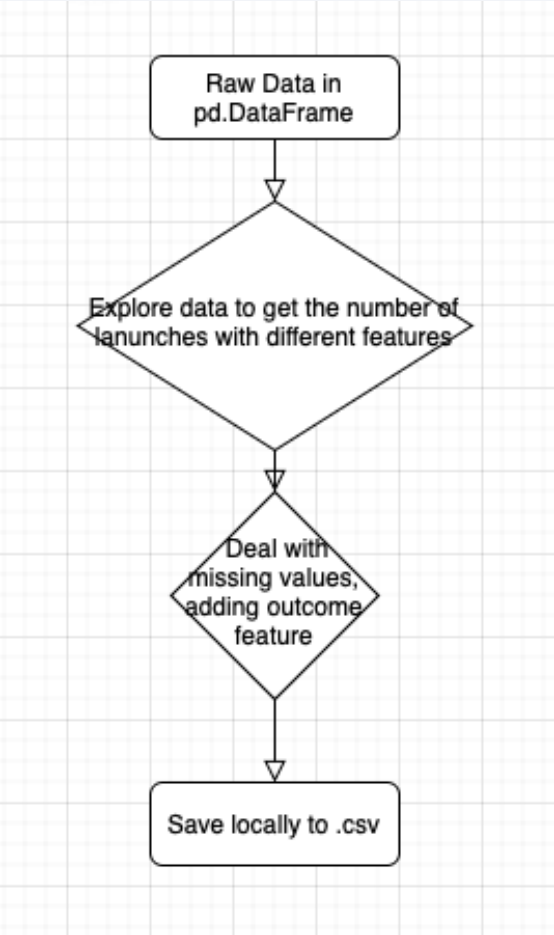
- File name:

  jupyter-labs-webscraping.ipynb



Data Wrangling

- The purpose is to clean the data and prepare it for exploratory Data Analysis (EDA)
- First calculate the number of launches on each site and then calculate the number and occurrence of mission outcome per orbit type
- Remove the missing data or fill them with proper value.
- Then create landing outcome feature, as numbers will make it easier for future analysis.
- Last but not least, we save the result to .csv file



- See details in: https://github.com/ZBaiY/Applied_Data_Science_Capstone.git
- File name:

labs-jupyter-spacex-Data wrangling.ipynb

EDA with Data Visualization

- We first started by using a scatter graph to find the relations between the following features (with hue):
  - Payload and flight numbers
  - Flight Number and Launch Site
  - Payload and Launch Site
  - Flight Number and Orbit Type
  - Payload and Orbit Type
- See details in: https://github.com/ZBaiY/Applied_Data_Science_Capstone.git
- File name:

  edadataviz.ipynb
- The scatter plot visualizes and reveals quickly the relationship between two variables.

EDA with Data Visualization

- Once we got a rough relation between variables from the scatter plots, we used line chart and bar chart for further analysis
  - Plotting bar graph is the most obvious way to see relations between attributes, in this case, we studied which orbits have the highest probability to success.
  - For timely trends, using a line chart is the most optimal, so we used line chart to see the launch success yearly trend
- In the end, we created dummy variables to the categorical columns for feature engineering

  - See details in: https://github.com/ZBaiY/Applied_Data_Science_Capstone.git File name: edadataviz.ipynb

EDA with SQL

- Using SQL, we performed many queries to get a better understanding of the dataset:
  - Displaying the names of launch sites
  - Displaying 5 records where launch sites begin with the string 'CCA'
  - Displaying the total payload mass carried by booster launched by NASA (CRS)
  - Displaying the average payload mass carried by booster version F9 v1.1.
  - Listing the date when the first successful landing outcome in ground pad was achieved
  - Listing the names of the 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 failed landing_outcomes in drone ship, their booster versions, and launch sites names for in year 2015
  - Ranking the count of landing outcomes or success between the date 2010-06-04 and 2017-03-20 in descending order

- See details in: https://github.com/ZBaiY/Applied_Data_Science_Capstone.git File name: jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- In order to visualize the launch data into an interactive map, we used the module Folium to generate maps and pin each launch site.
- We assigned the dataframe outcome features to classes 0(failure), and 1(success). We mark them with Red and Green markers respectively.
- In the end we used the Harversine's formula to calculate the distance of the launch sites to various landmarks to answer:
  - How close the launch sites with railways, highways and coastlines
  - How close the launch sites with nearby cities?

- See details in: https://github.com/ZBaiY/Applied_Data_Science_Capstone.git File name: lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- In order to allow the users to interactively explore the data on a dashboard. We used the Plotly module.
- We plotted pie charts showing the total launches from different sites
- We plotted a scatter graph showing the relation between outcome and payload mass for different booster versions.

- See details in: https://github.com/ZBaiY/Applied_Data_Science_Capstone.git File name: spacex_dash_app.py

Predictive Analysis (Classification)

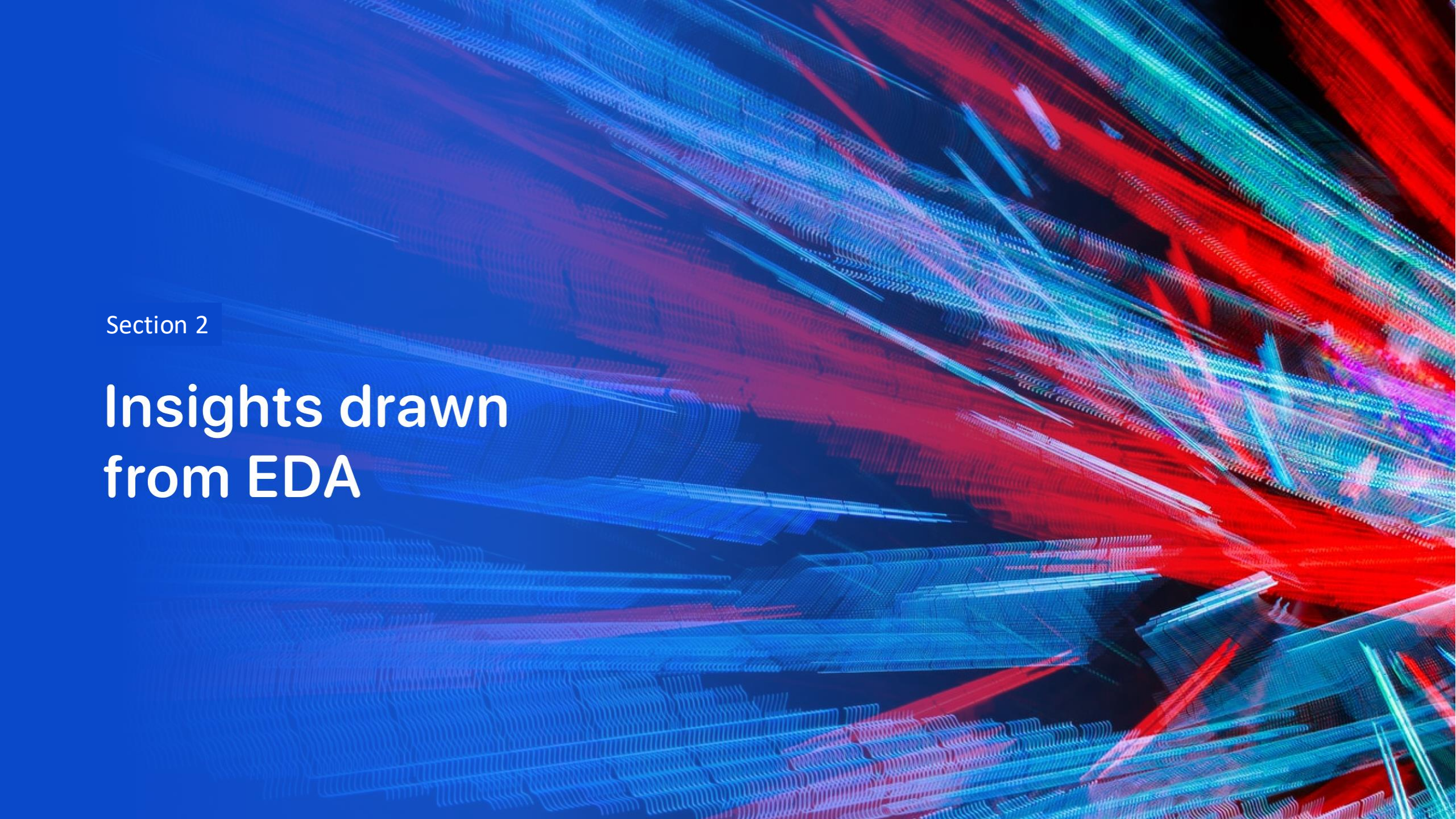
- We built several Machine Learning Models, including Logistic Regression, SVM, Decision Tree, KNN, etc.
  - First, we load the dataset into NumPy and Pandas
  - Then, we transform the data and split it into training and test datasets.
  - Build and train models using sci-kit learn and we explore hyperparameters to GridSearchCV and find the optimal model.
- We evaluate the models using different scales, such as R-2 and confusion matrix, etc.
- With Feature engineering and algorithm tuning, we improved the model.
- In the end, we select the model with the best accuracy score.

- See details in: https://github.com/ZBaiY/Applied_Data_Science_Capstone.git File name: SpaceX_Machine Learning Prediction_Part_5.ipynb

Results

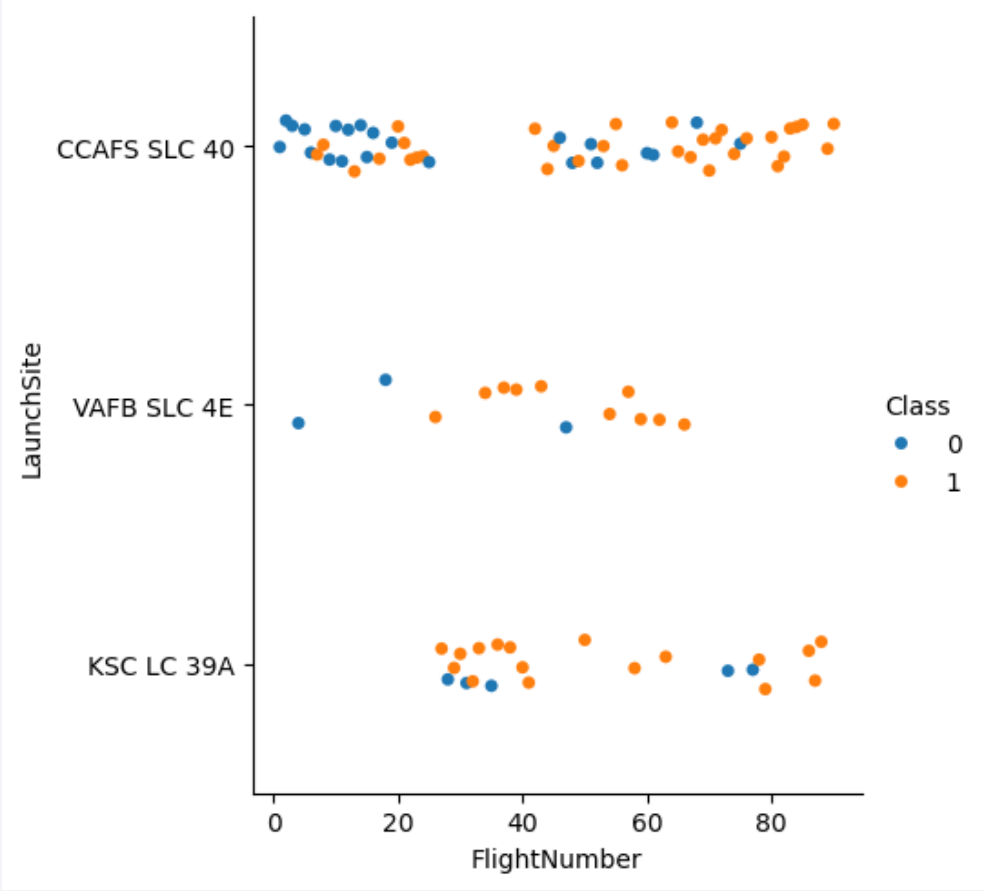
We categorize the result into three parts:

- Data Exploration results
- Interactive analytics
- Predictive analysis results


Section 2
Insights drawn from EDA

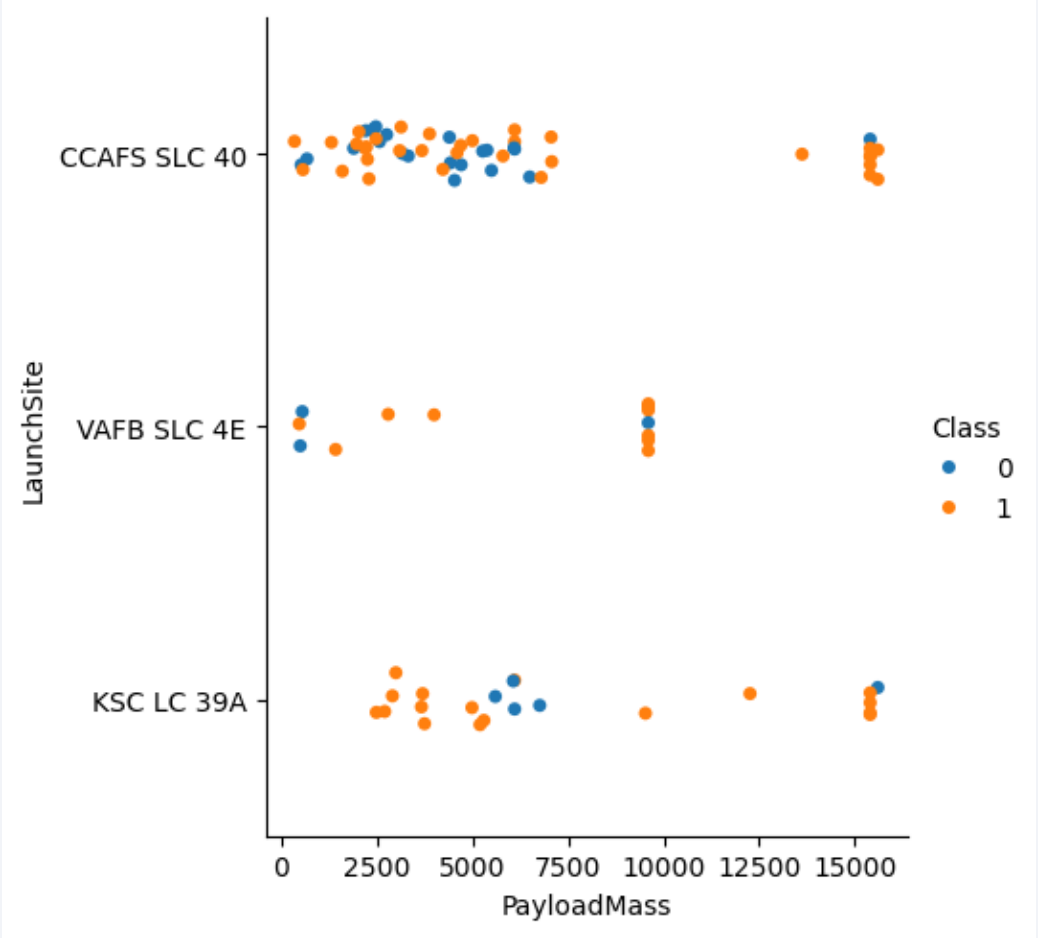
Flight Number vs. Launch Site

- The scatter plot shows that the larger the flight number of the launch site is, the greater the success rate will be.



Payload vs. Launch Site

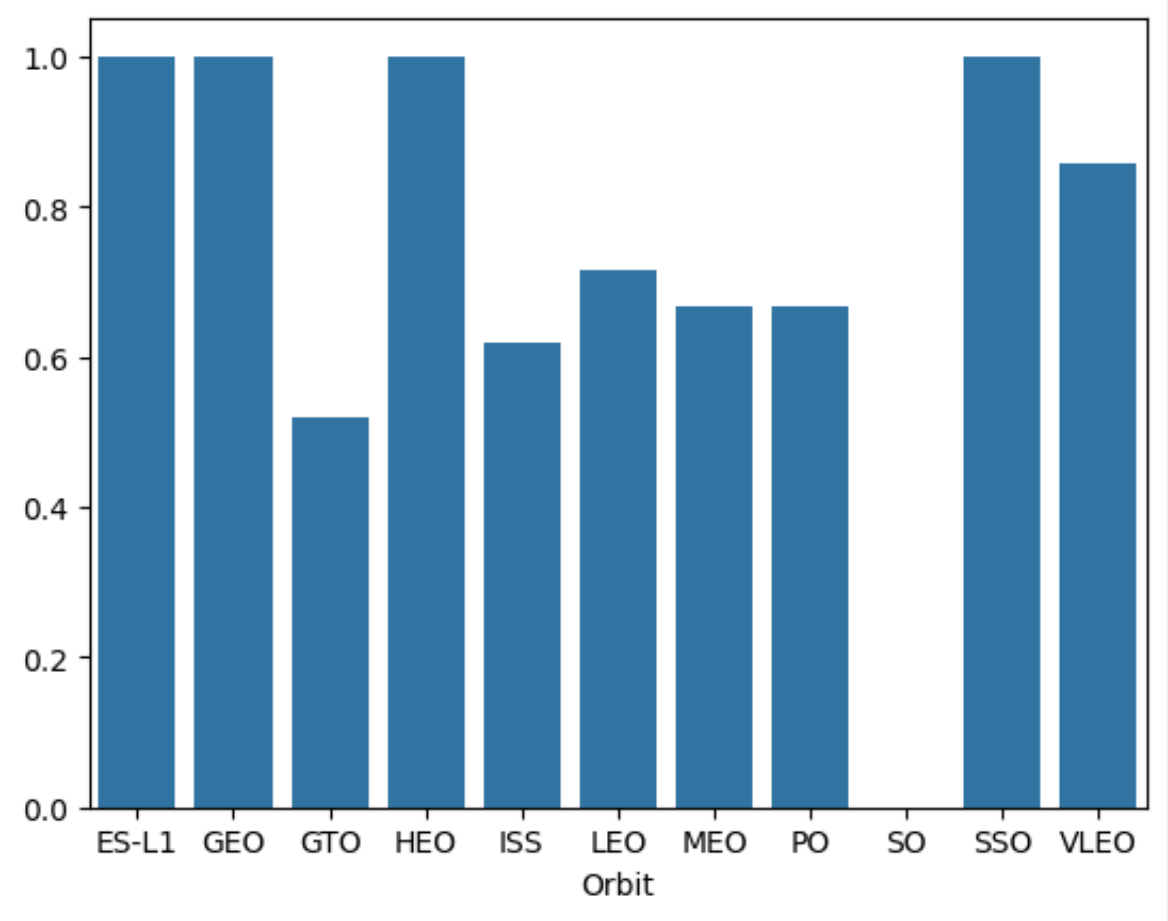
- Once the payload mass reaches 7000kg, the success rate increases.
- Launch site dependency is less clear.



Success Rate vs. Orbit Type

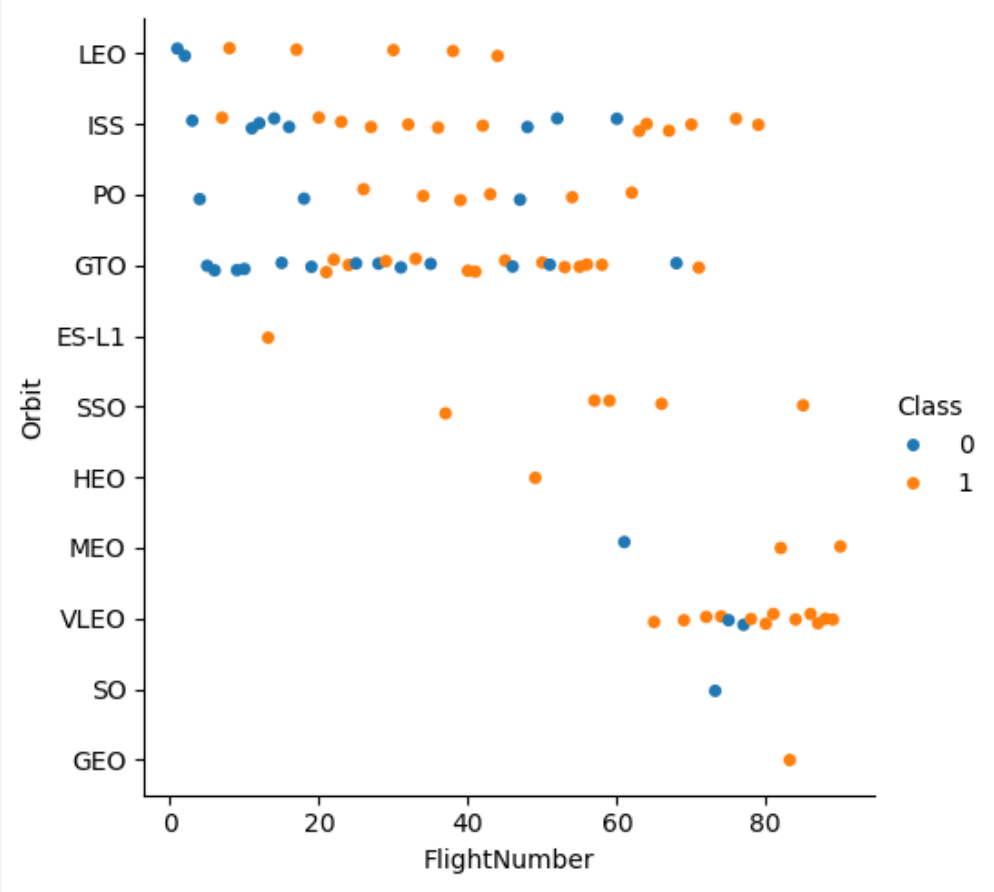
- Some orbit has 100% success rates such as SSO, HEO, GEO, and ES-L1 while SO does not have a successful mission

- Nevertheless, some orbit needs more data to have meaningful implications.



Flight Number vs. Orbit Type

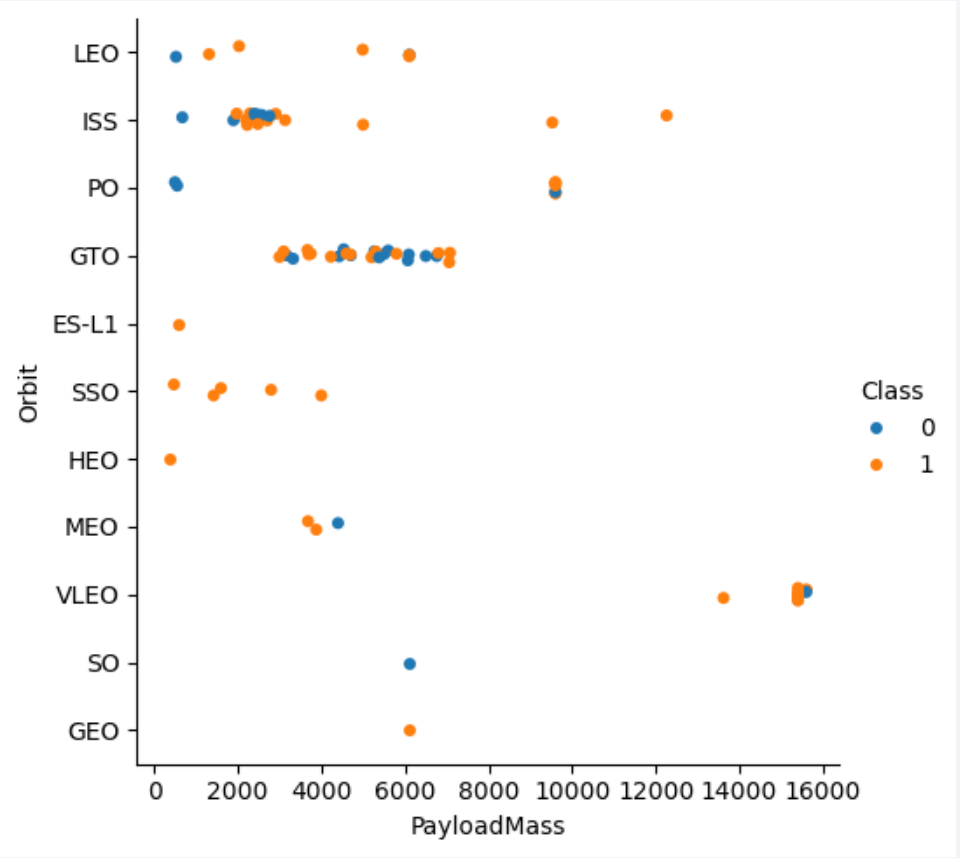
- In general, successful mission are with higher Flight numbers.
- Some orbits has only one data point, the statistic is not meaningful at this point.s



Payload vs. Orbit Type

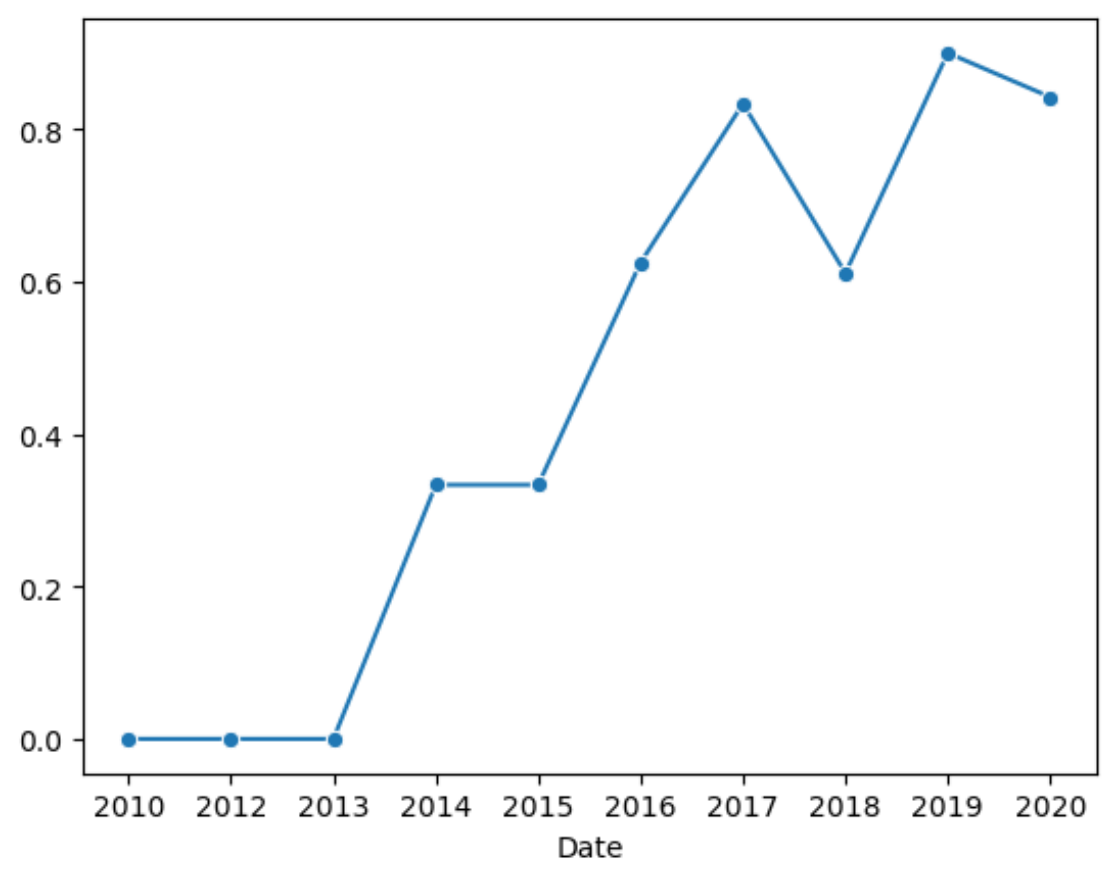
- Heavier payload performs better with LEO, ISS and PO orbit. However, negative on MEO and VLEO

- As before, SO, GEO, HEO only have one point and they do not have meaningful implications.



Launch Success Yearly Trend

- The success rate is increasing as time goes by implicating SpaceX is getting more experiences



All Launch Site Names

- DISTINCT keyword shows the unique launch sites

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

```
 * 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'

- LIMIT keyword selects the first few rows, LIKE keyword finds the records we want

Display 5 records where launch sites begin with the string 'CCA'

```
%sql SELECT * FROM SPACEXTABLE where Launch_Site LIKE 'CCA%' LIMIT 5
```

Python

```
 * 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

Total Payload Mass

- SUM function calculated the summation and WHERE keywords selects rows we need

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

```
%sql SELECT SUM(Payload_Mass__KG_) AS TotalPayload From SPACEXTABLE Where Customer = 'NASA (CRS)';
```

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

TotalPayload
45596

Average Payload Mass by F9 v1.1

- AVG function calculates the average, and WHERE select the rows we want

```
%sql SELECT AVG(Payload_Mass__KG_) AS AvgPayload From SPACEXTABLE Where Booster_Version = 'F9 v1.1';
```

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

AvgPayload
2928.4

First Successful Ground Landing Date

- MIN function finds the minimum, and WHERE selects the rows we want

```
%sql SELECT MIN(Date) AS frst FROM SPACEXTABLE WHere Landing_Outcome = 'Success (ground pad)'
```

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

frst
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- WHERE to filter the successfully landed on a drone ship
- AND condition to determine the landing with the desired payload

```
%sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE Where Landing_Outcome LIKE 'Success (drone%' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_< 6000
```

Python

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

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

Total Number of Successful and Failure Mission Outcomes

- GROUP BY to group mission_outcome column

List the total number of successful and failure mission outcomes

```
%sql SELECT Mission_Outcome, COUNT(*) AS Total_Count FROM SPACEXTABLE GROUP BY Mission_Outcome;
```

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

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

Boosters Carried Maximum Payload

- Using parentheses to enclose a subquery.

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql SELECT booster_version as maxpayload FROM SPACEXTABLE WHERE (PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX
```

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

maxpayload
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

- We combined keywords like WHERE and AND.

```
%%sql SELECT
    substr(Date, 6, 2) AS month,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Failure (drone ship)'
  AND substr(Date, 0, 5) = '2015';
```

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

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

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

- DESC mean descending
- We combined WHERE, BETWEEN, AND, for the condition
- GROUP BY to group the column

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

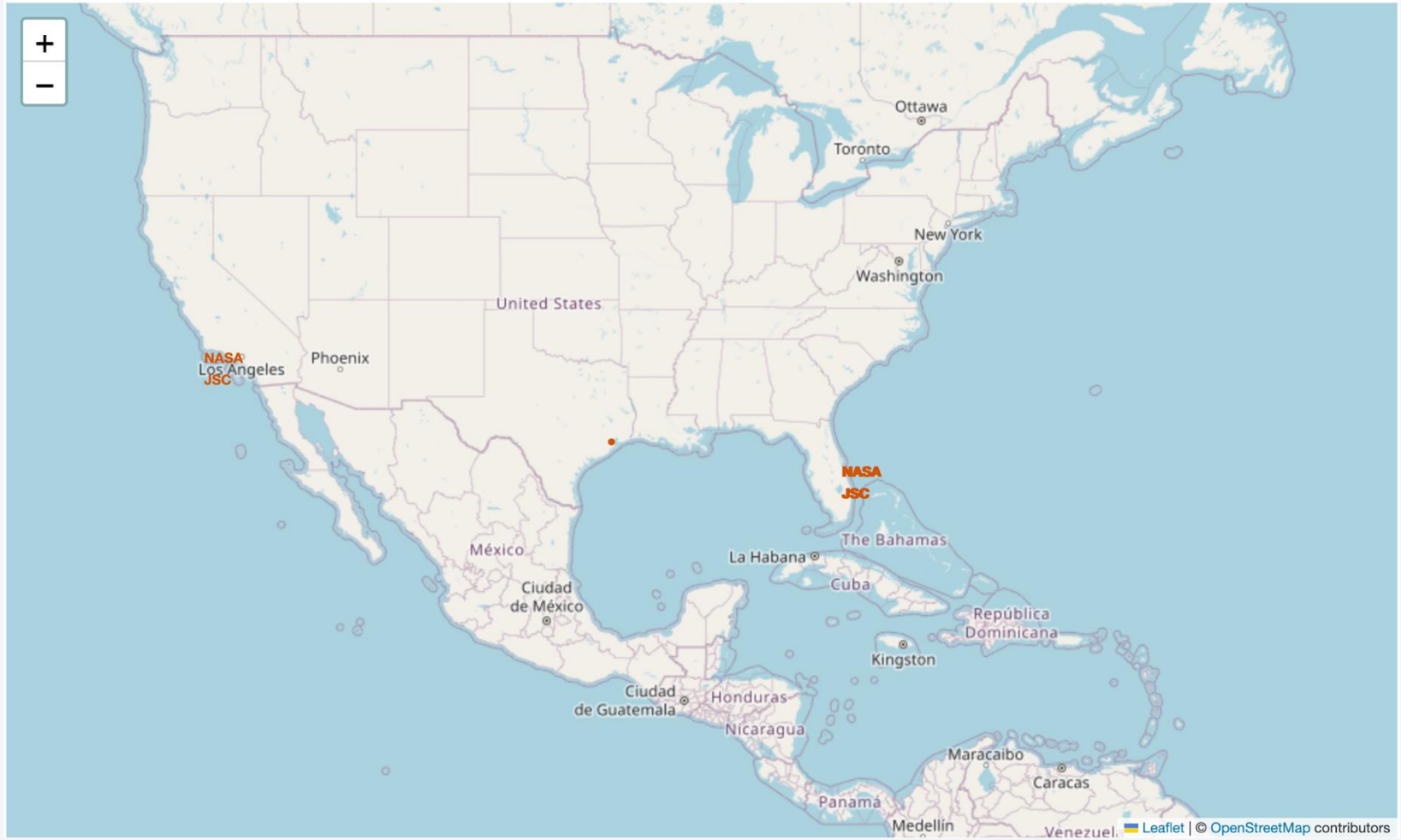
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	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

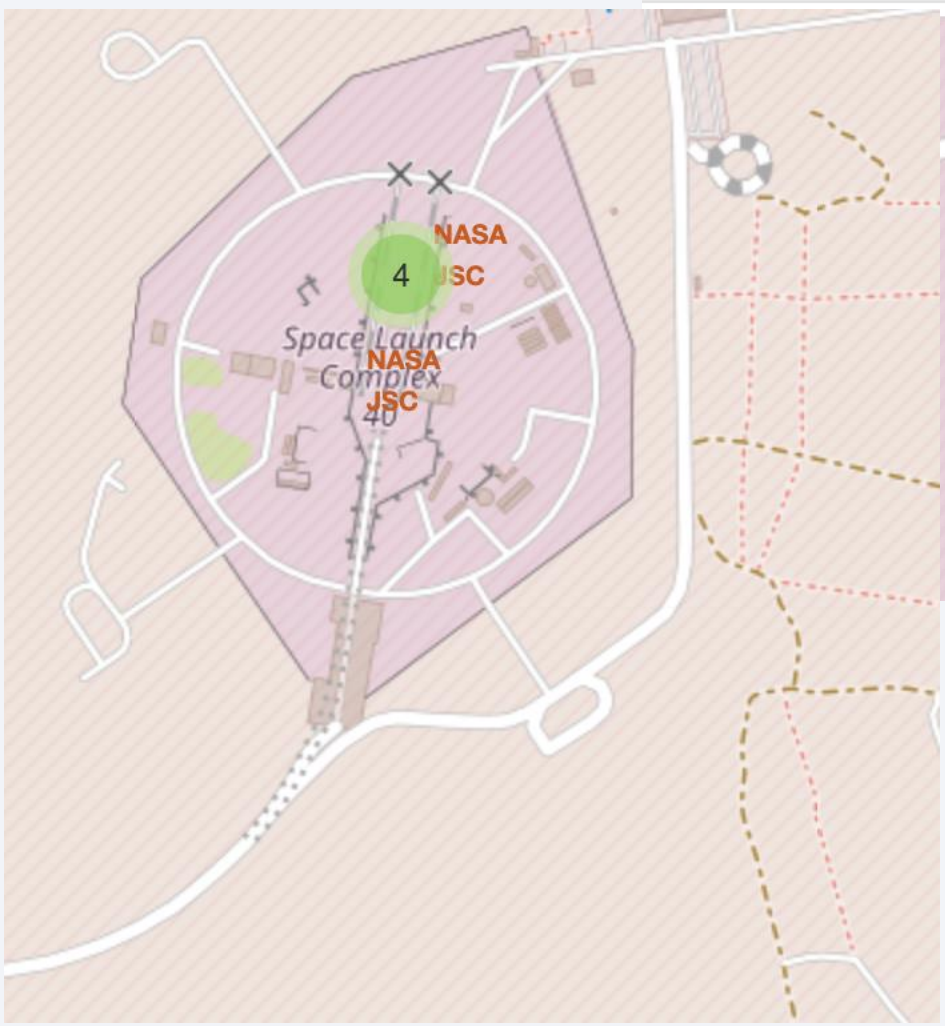
Section 3

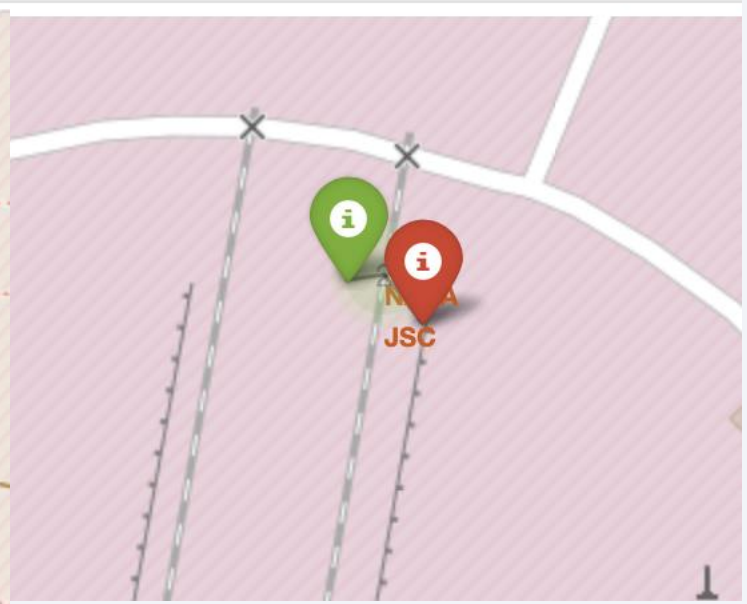
Launch Sites Proximities Analysis

Location of all the Launch Sites



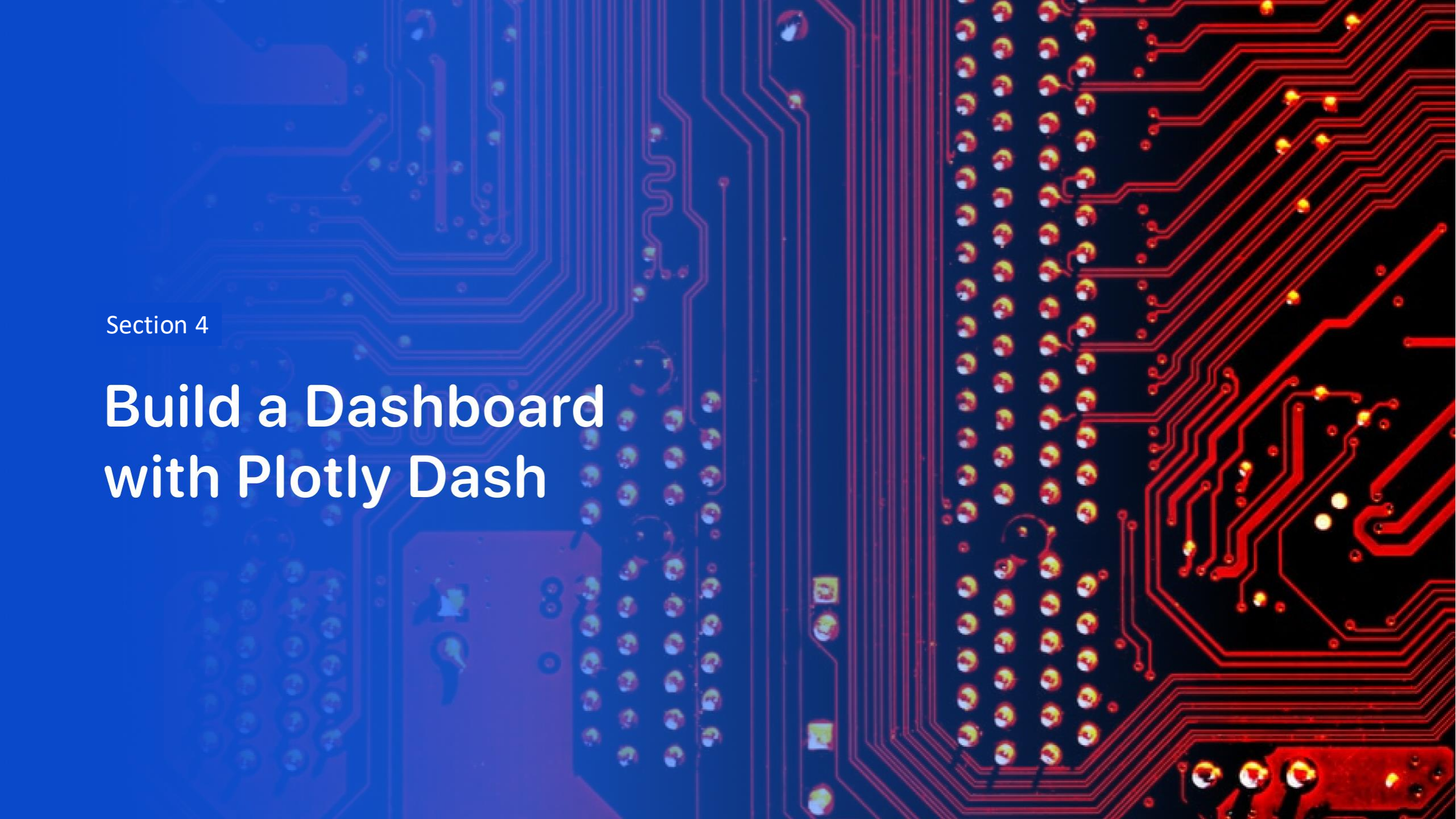
Markers showing launch sites with color labels



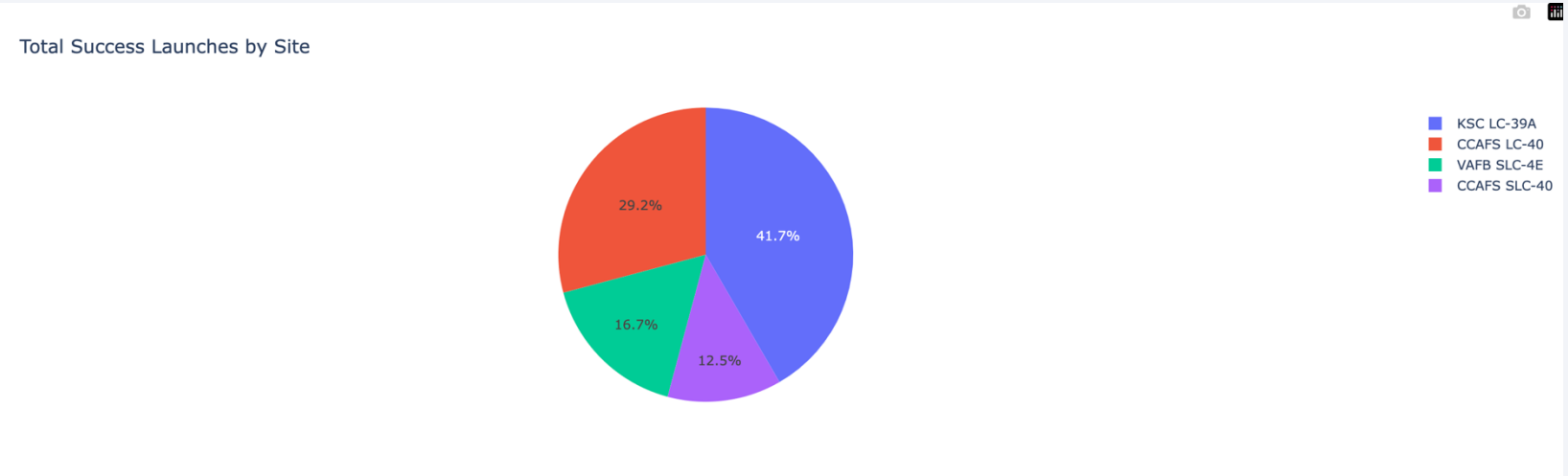


<Folium Map Screenshot 3>

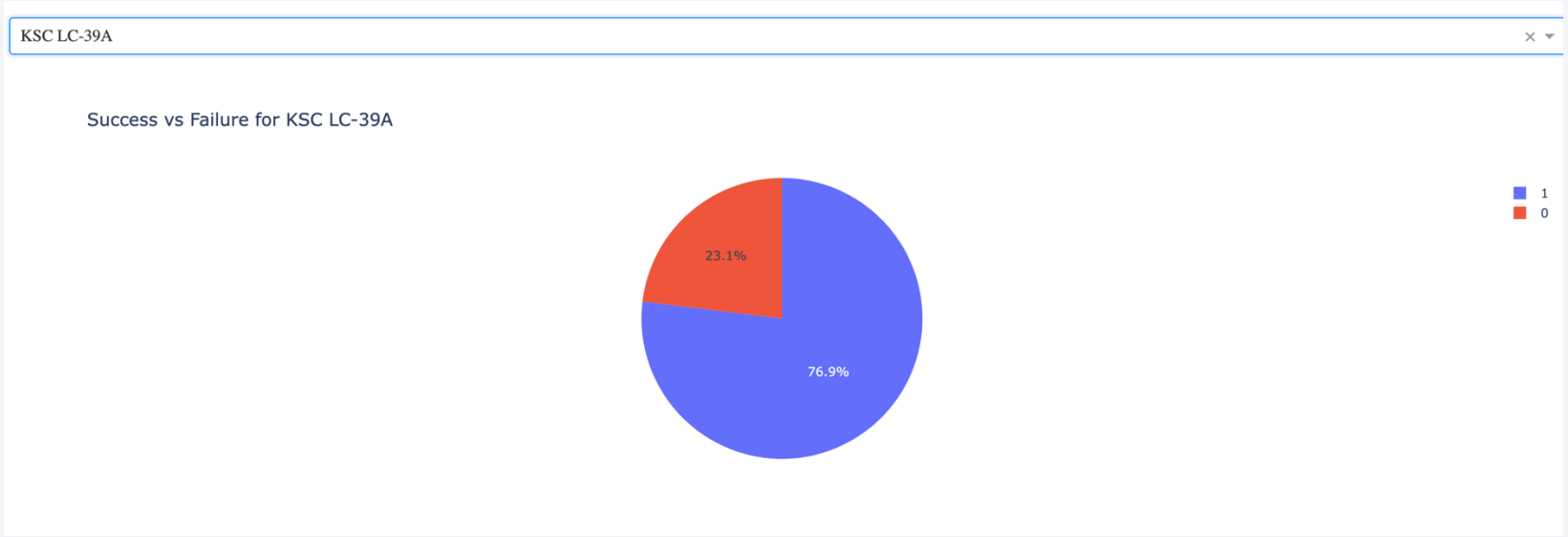
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

The success percentage by each sites.

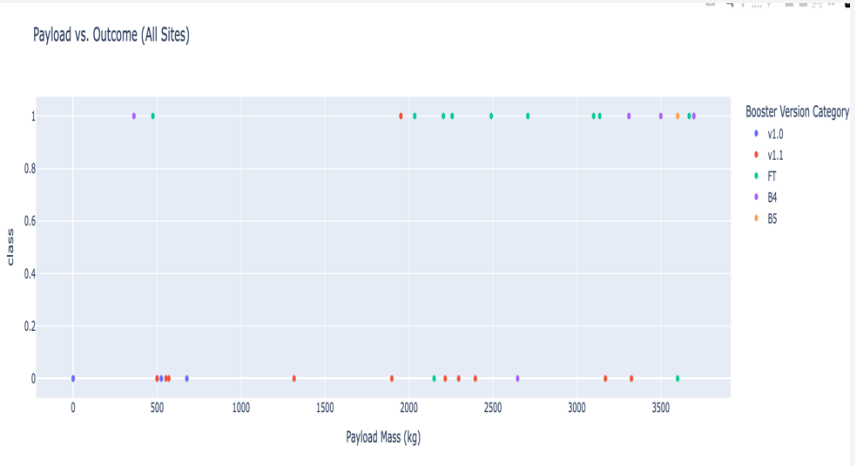


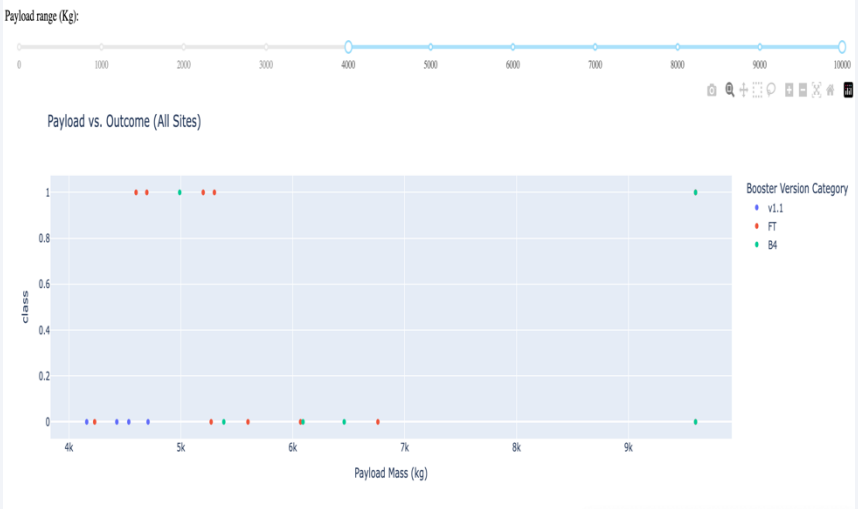
The highest launch-success ratio: KSC LC-39A



Payload vs Launch Outcome Scatter Plot

- The success rate for low weighted is higher than heavy





Section 5

Predictive Analysis (Classification)

Classification Accuracy

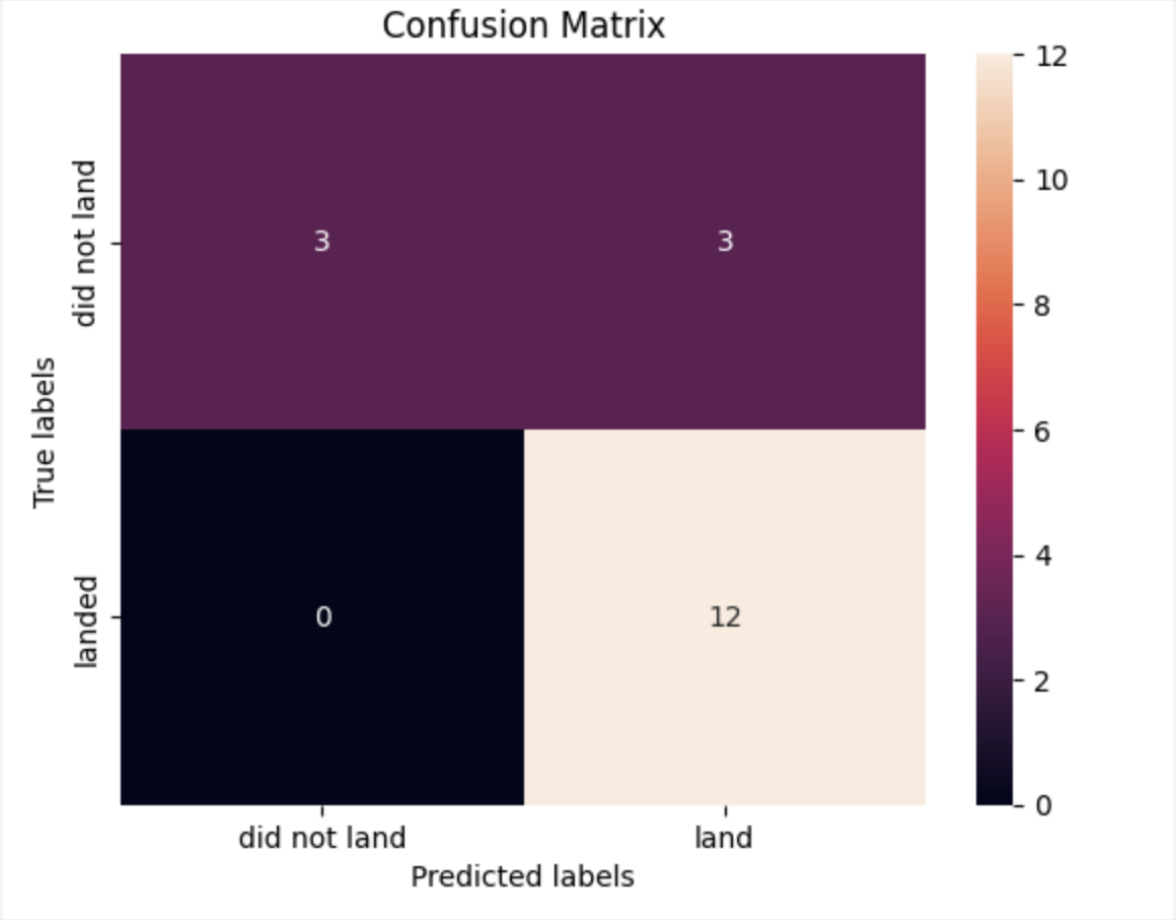
- The best model is the Decision Tree Model with the highest classification accuracy

```
print("knn hpyerparameters :(best parameters) ",knn_cv.best_params_)
print("accuracy :",knn_cv.best_score_)
print("tree hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
print("svm hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
print("logreg hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

```
knn hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
accuracy : 0.8482142857142858
tree hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'm
in_samples_split': 10, 'splitter': 'random'}
accuracy : 0.875
svm hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
logreg hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713
```

Confusion Matrix

- The classifier can distinguish between different classes.
- However, the major problem is the false positive. i.e. the unsuccessful landing being predicted as successful



Conclusions

- The decision tree is the best prediction model
- Low-weighted payloads(below 4000kg) performed better
- As time goes by, SpaceX is getting better and better. One can believe that in the future they can manage to do successful launches consistently
- SSO orbit have be best successful rate with more than one data points
- KSC LC-39A has the most successful launches of any sites

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!