IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alwin Antony
05/12/2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through SpaceX REST API
  - Data Collection through Web scraping Wikipedia
  - Data Wrangling
  - EDA with SQL
  - EDA with Data Visualization
  - Interactive Launch Site Maps with Folium
  - Interactive Dashboard with Plotly Dash
  - Machine Learning Binary Classification Models
- Summary of all results
  - Exploratory Data Analysis Results
  - Interactive Analytics Screenshots
  - Predictive Analysis Results

Introduction

- Project background and context
  - SpaceX produces rockets like the Falcon 9 which costs 62 million dollars per launch compared165 million dollar rocket launches from it's competition. SpaceX rockets are substantially cheaper due to the Falcon 9's ability to reuse the first stage of the rocket. The goal of this project is to determine whether the first stage will land, which will help us determine the cost of the launch. With data from SpaceX and Wikipedia, we will build a model to predict whether the launch will succeed or not.
- Problems you want to find answers
  - Which factors contribute most to determining whether the launch will succeed or fail?
  - Are there any relation between Launch Sites, Payloads and Orbits which can impact the outcome the launch?
  - Which Orbits and Payload combinations have the most Failures?

Section 1

Methodology

Methodology

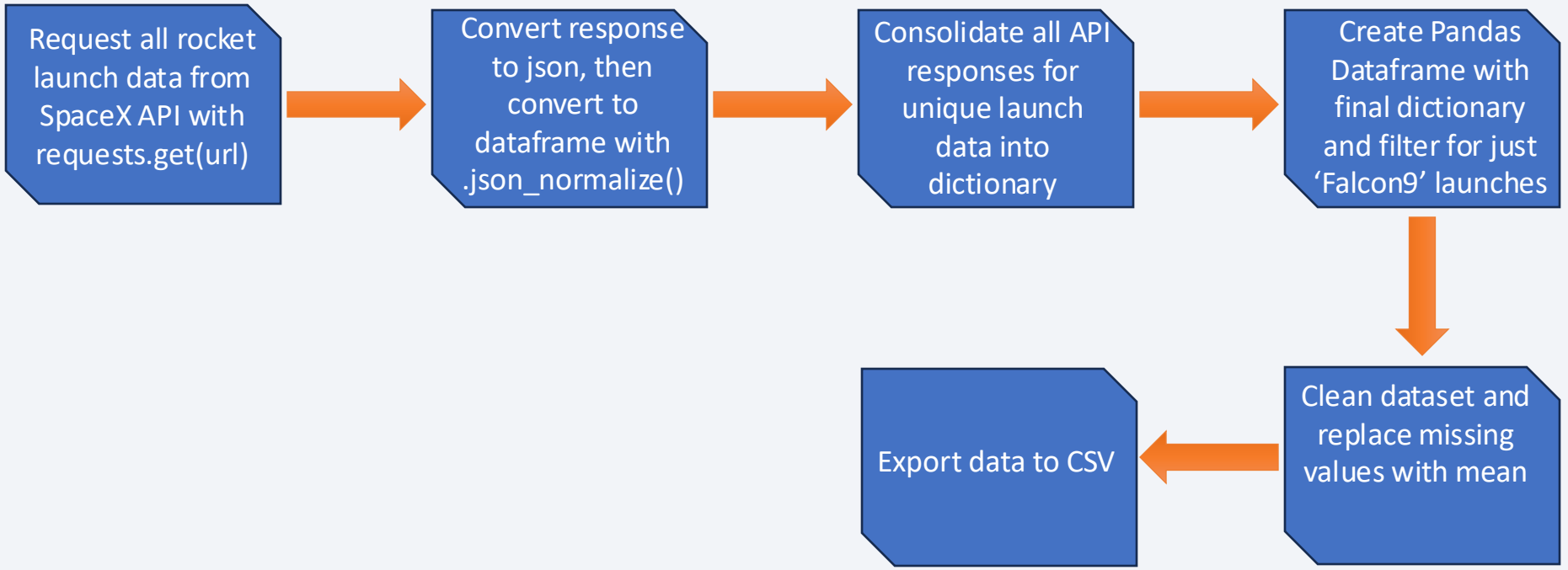
Executive Summary

- Data Collection methodology:
  - Collect data through SpaceX API and Web Scraping from Wikipedia
- Perform data wrangling
  - Handle Missing Values
  - Perform One-hot encoding on 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
  - Build, tune, evaluate classification models to pick the best model for classifying launches

Data Collection

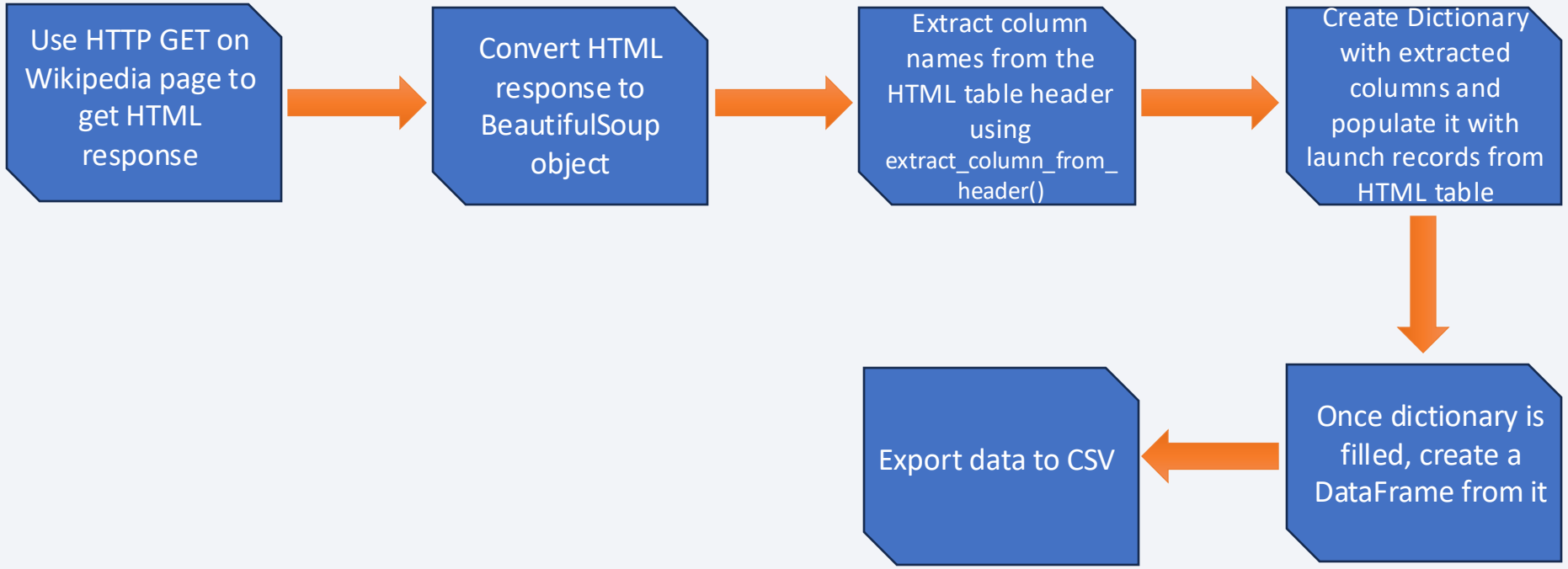
- Data was collected through SpaceX's REST API and the json response was converted into a pandas dataframe
    - Flight Number, Flights, Grid Fins, Reused, Reused Count, Launch Site, Date, Booster Version, Payload Mass, Orbit, Legs, Landing Pad, Block, Serial, Longitude, Latitude, Outcome
- Data was also collected by web scraping Wikipedia tables using BeautifulSoup, the tables contain statuses and other key information about all previous SpaceX launches
    - Flight Number, Date, Time, Launch Site, Customer, Payload, Payload Mass, Orbit, Version Booster, Booster landing, Outcome
- The data was then cleaned and missing values were replaced

Data Collection – SpaceX API



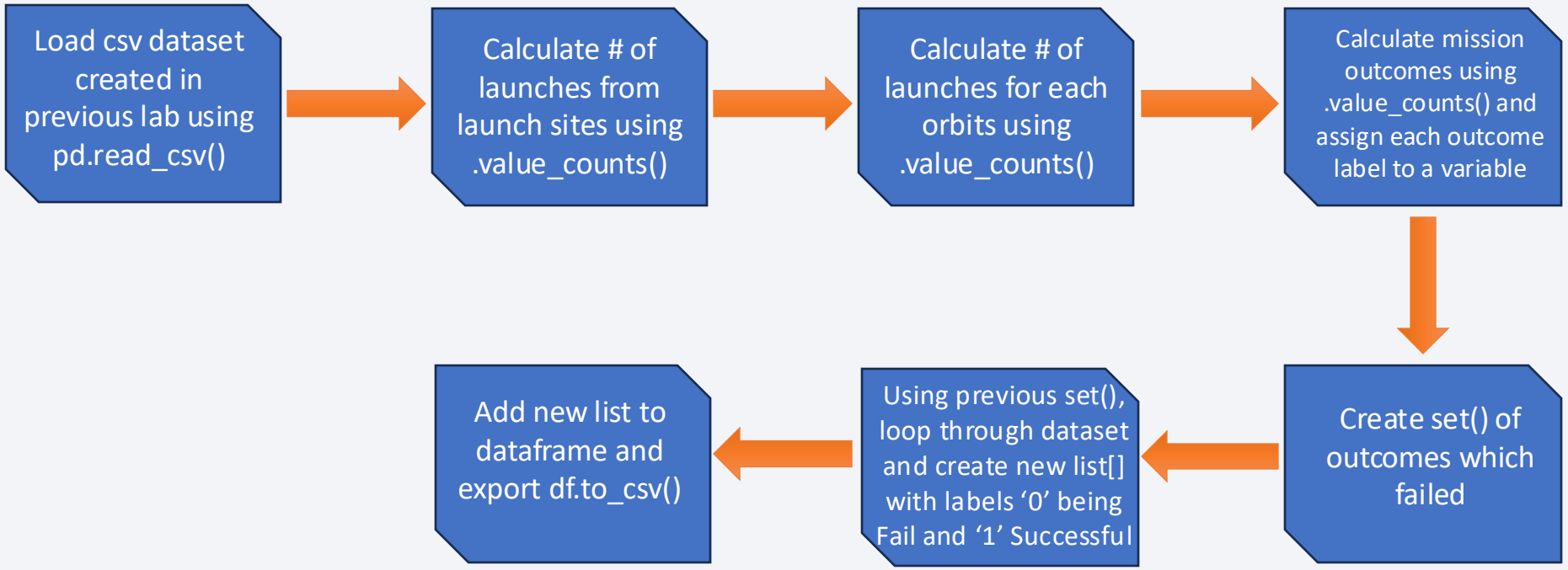
GitHub URL: https://github.com/alwinantony/IBM-DataScience-Captstone/blob/main/1%20-%20Data%20Collection%20API.ipynb

Data Collection – Scraping



GitHub URL: https://github.com/alwinantony/IBM-DataScience-Captstone/blob/main/2%20-%20Webscraping.ipynb

Data Wrangling



GitHub URL: https://github.com/alwinantony/IBM-DataScience-Captstone/blob/main/3%20-%20Data%20Wrangling.ipynb

EDA with Data Visualization

- Charts:
  - Flight Number vs. Payload Mass **(Scatter),** Flight Number vs. Launch Site **(Scatter),** Payload Mass vs. Launch Site **(Scatter)**, Success Rate vs. Orbit Type **(Bar Chart),** Flight Number vs. Orbit Type **(Scatter)**, Payload Mass vs Orbit Type **(Scatter)** and Launch Success Yearly Trend (**Line Chart**)
- Why use these charts:
  - Scatter point plot helps visualize relationship, trends between continuous variables
  - Bar Chart helps compare different categories
  - Line chart displays how continuous independent variables changes over time

GitHub URL: https://github.com/alwinantony/IBM-DataScience-Captstone/blob/main/5%20-%20EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

SQL queries performed:

- 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 all the booster_versions that 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.

GitHub URL: https://github.com/alwinantony/IBM-DataScience-Captstone/blob/main/4%20-%20SQL.ipynb

Build an Interactive Map with Folium

- Added Circle marker to launch sites using latitude and longitude
- Added labels to each markers
- Assigned colors using Marker Clusters indicating launch outcomes in each launch sites.
  - Green for success and Red for Failed launches.
- Calculate distance between launch sites and nearby landmarks, assigned a line from landmark to launch site

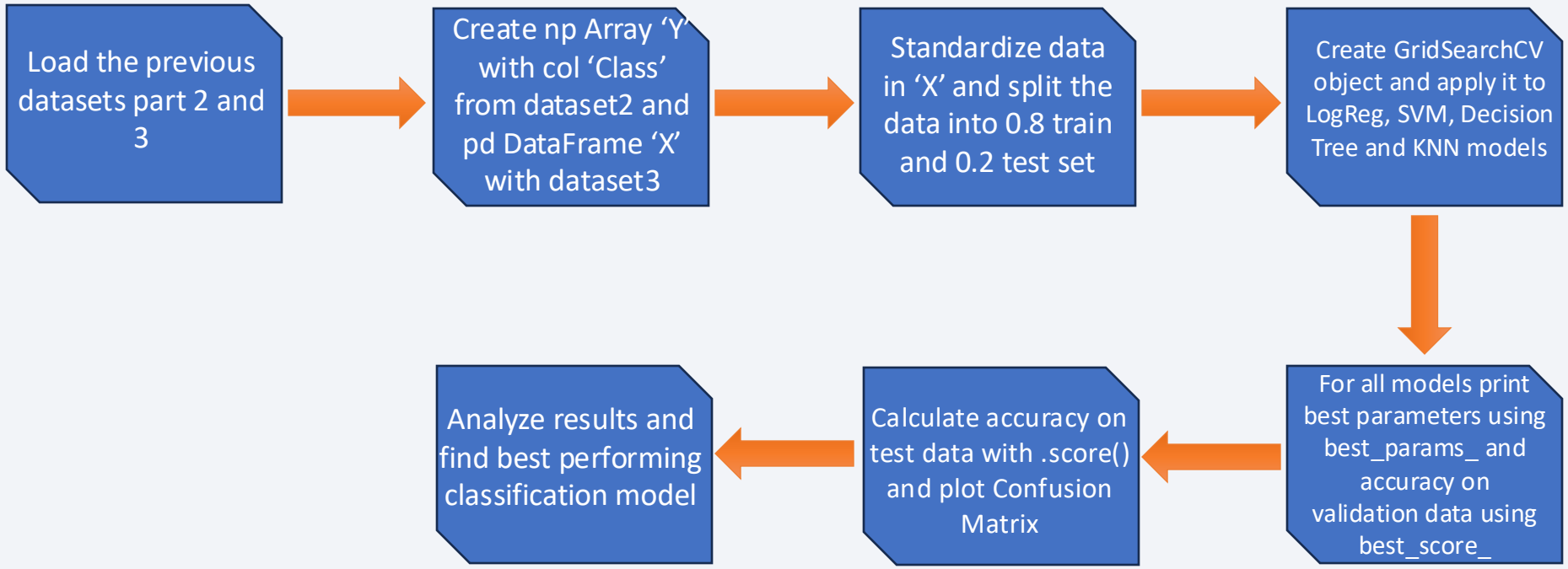
GitHub URL: https://github.com/alwinantony/IBM-DataScience-Captstone/blob/main/6%20-%20Folium%20%20Interactive%20Map.ipynb

Build a Dashboard with Plotly Dash

- Plots/Graphs in dashboard
  - Added dropdown list with options including each launch sites and an ‘All’ option
  - Plotted Pie Chart showing successful launches for all launch sites if ’All’ was selected, else plotted Pie Chart showing successful and failed launches at the selected launch site
  - Added a slider to select range of Payload Mass
  - Plotted Scatter plot showing relation between Payload vs. Outcome for booster versions

GitHub URL: https://github.com/alwinantony/IBM-DataScience-Captstone/blob/main/7%20-%20Dashboard%20with%20Plotly%20Dash.py

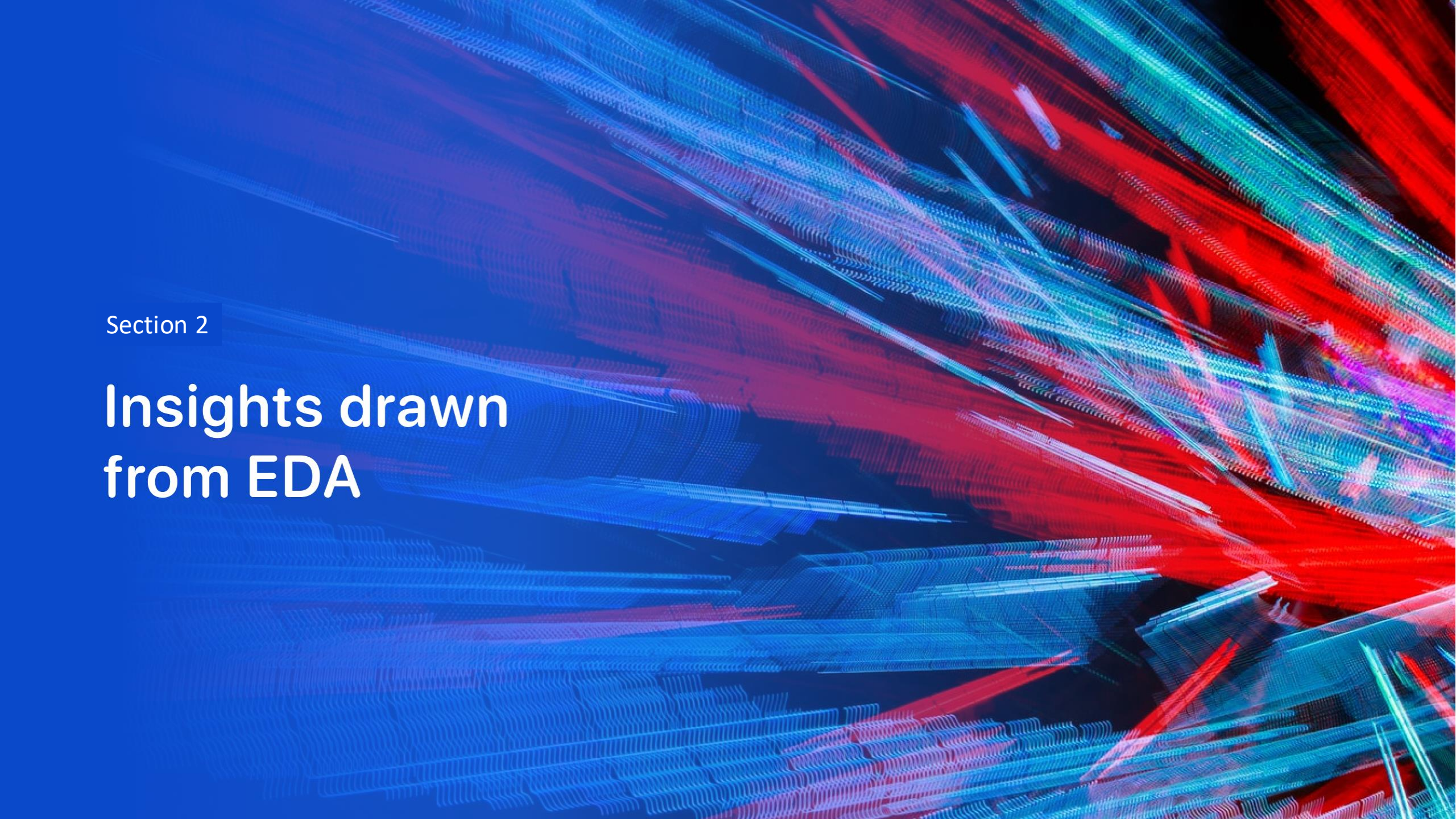
Predictive Analysis (Classification)



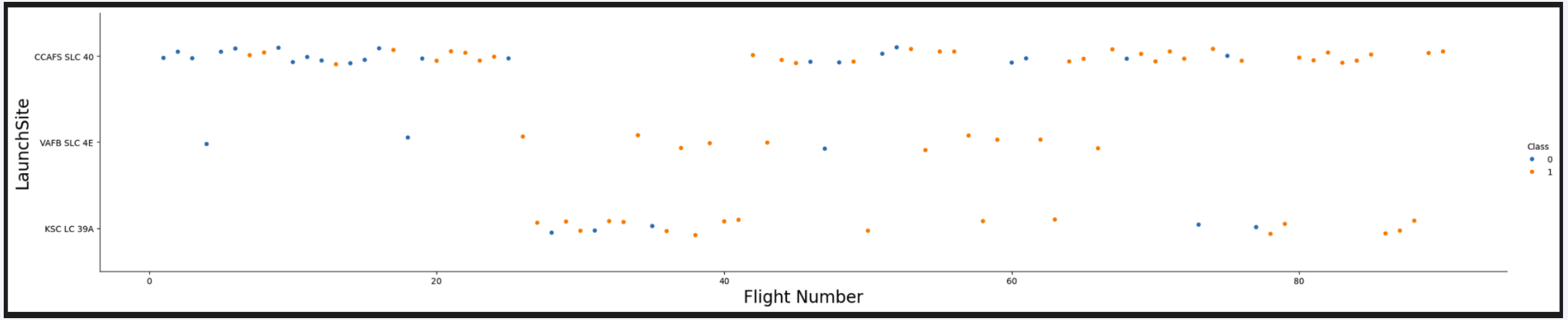
GitHub URL: https://github.com/alwinantony/IBM-DataScience-Captstone/blob/main/8%20-%20Machine%20Learning%20Predictive%20Analysis.ipynb

Results

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


Section 2
Insights drawn from EDA

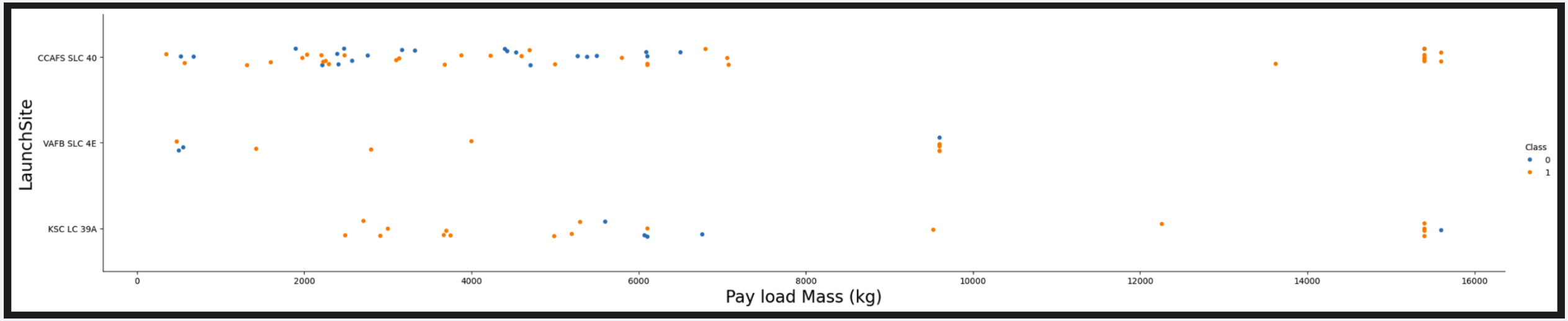
Flight Number vs. Launch Site



Explanation

- We can observe that initial flights irrespective of Launch Site had higher Failures
- Later launches are having higher successes
- CCAFS SLC 40 Launch Site seems to be the more popular launch site
- KSC LC 39A Launch Site has better successful launch ratio compared to other launch sites

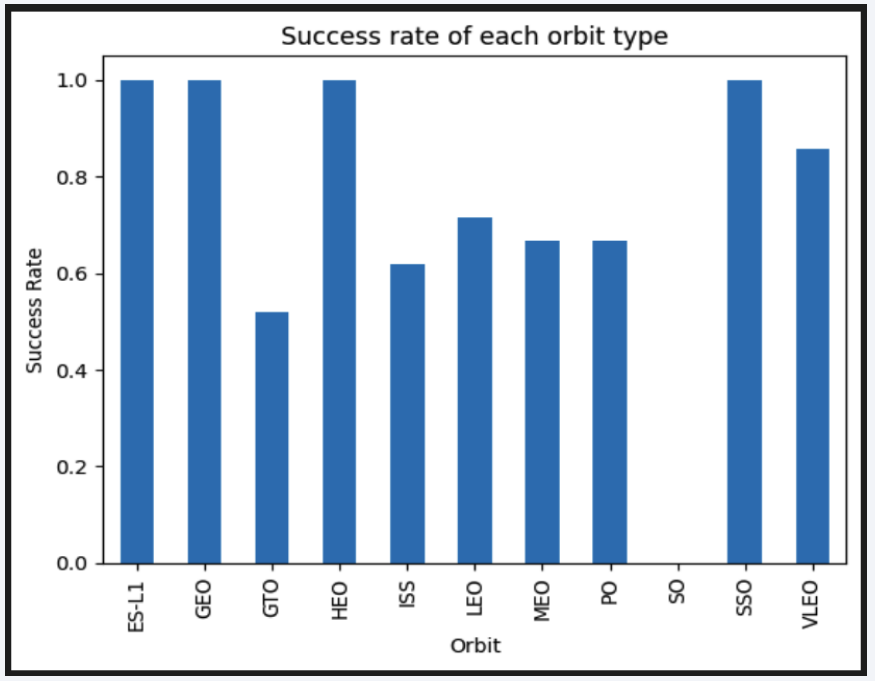
Payload vs. Launch Site



Explanation

- We can observe that Higher Payloads has Higher Success rates
- CCAFS SLC 40 Launch Site has higher Failures for Payloads 0-7000kg while KSC LC 39A has higher Success rates for these Payloads

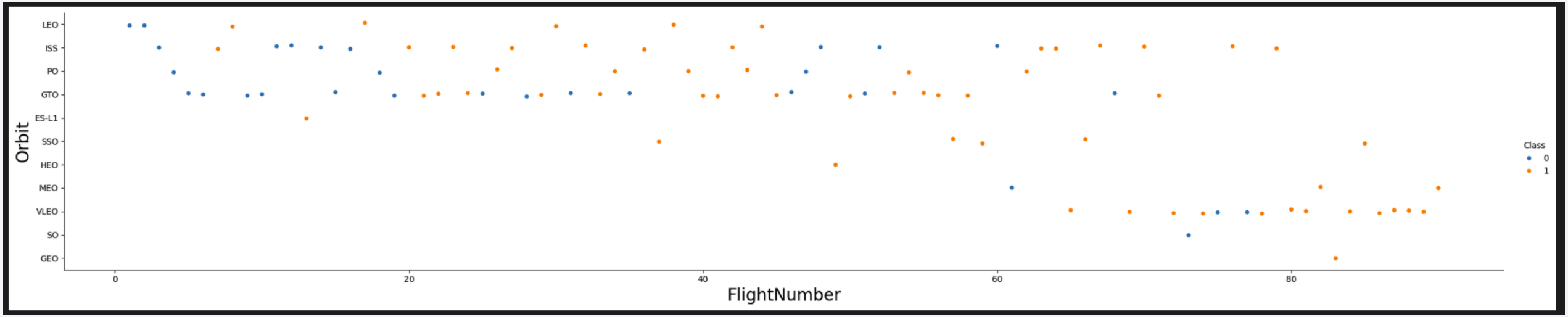
Success Rate vs. Orbit Type



Explanation

- ES-L1, GEO, HEO and SSO Orbits have 100% Success Rate while SO has 0%
- ISS, LEO, MEO, PO has Success Rate of more than 60%
- SO and GTO seems to be the worst orbits

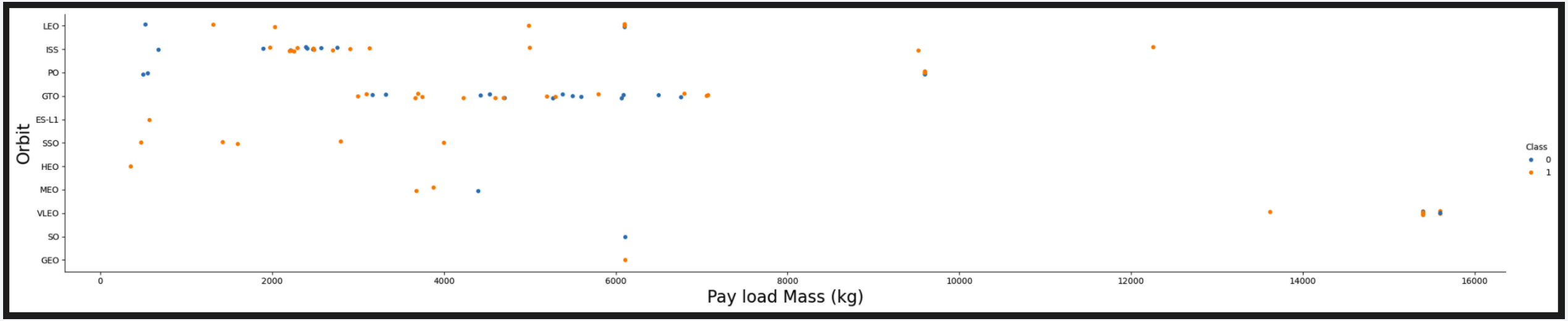
Flight Number vs. Orbit Type



Explanation

- Flight Number and Orbit Types don't seem to produce any interesting trends other than showing that with more flights, Success Rates seems to be increasing

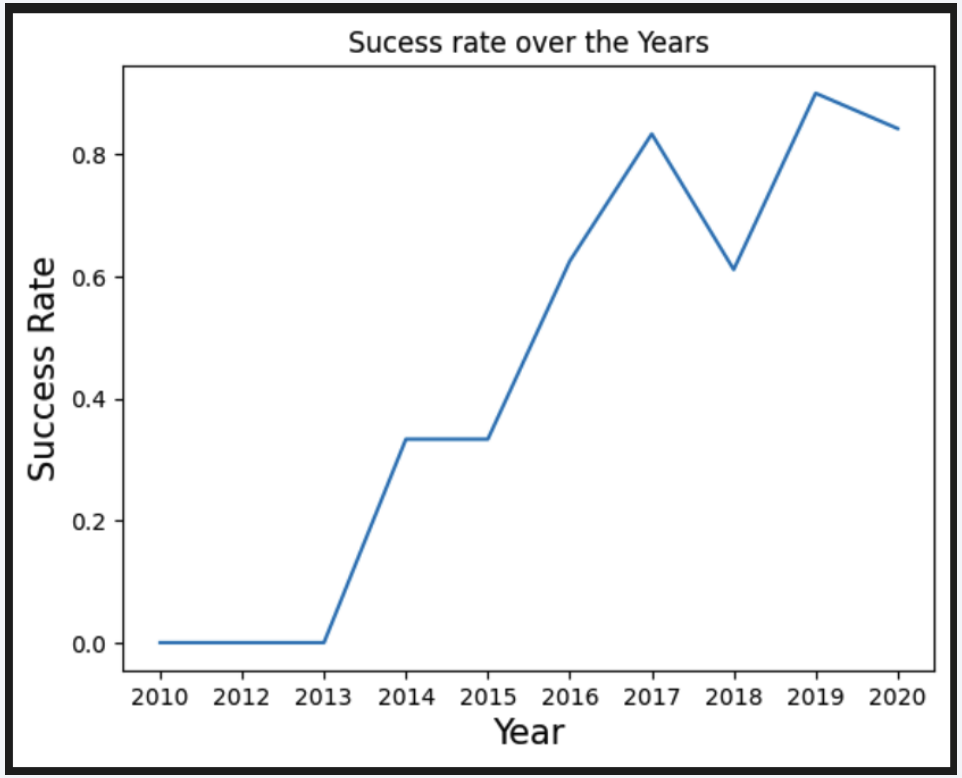
Payload vs. Orbit Type



Explanation

- Mostly all Orbit Types seems to have higher Success Rates with heavier Payloads except for GTO which has more Failures with heavier payloads

Launch Success Yearly Trend



Explanation

- We can observe a positive trend of Success Rate throughout the year
- However, Success Rate declined in the year 2018 and 2020

All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

```
%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

Explanation

- With the DISTINCT keyword, we can get list of unique Launch Sites

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string '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

Explanation

- Using ‘LIKE’ we can query Launch_Sites that start with CCA and with the ’LIMIT’ keyword, we can limit the number of query results returned

Total Payload Mass

Task 3

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as 'Total Payload Mass' FROM SPACEXTABLE WHERE Customer like 'NASA (CRS)%'
```

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

Total Payload Mass
48213

Explanation

- Using ‘LIKE’ we can query Customers that start with ‘NASA (CRS)’ and with the ’SUM’ keyword, we can get total Payload Mass from the result set

Average Payload Mass by F9 v1.1

Task 4

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

```
%sql SELECT ROUND(AVG(PAYLOAD_MASS__KG_), 2) as 'Average Payload Mass' FROM SPACEXTABLE WHERE Booster_Version like 'F9 v1.1%'
```

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

Average Payload Mass
2534.67

Explanation

- Using ‘LIKE’ we can query Booster_Versions that start with ‘F9 v1.1’ and with the ’AVG’ keyword, we can get average Payload Mass from the result set

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql SELECT min(Date) as 'First Successful Landing' FROM SPACEXTABLE WHERE Landing_Outcome like 'Suc%'
```

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

First Successful Landing
2015-12-22

Explanation

- Using ‘LIKE’ we can query Landing_Outcome that start with ‘Suc’ and with the ’MIN’ keyword, we can get the minimum date from the query result set

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

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

Explanation

- We can query Landing_Outcome which equals ‘Success (drone ship)’ and with the ’BETWEEN’ keyword, we can query payloads between 4000 and 6000

Total Number of Successful and Failure Mission Outcomes

```
%%sql SELECT CASE
        WHEN Landing_Outcome LIKE 'Success%' THEN 'Success'
        WHEN Landing_Outcome LIKE 'Failure%' THEN 'Failure' END as Status, COUNT(*) AS TotalCount
    FROM SPACEXTABLE
    WHERE Landing_Outcome like 'Success%' or Landing_Outcome LIKE 'Failure%'
    GROUP BY Status
```

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

Status	TotalCount
Failure	10
Success	61

Explanation

- We can use the CASE keyword to classify all Landing_Outcome starting with 'Success%' as Success and 'Failure%' as Failure then use GROUP BY to group them into 2 groups and COUNT each Success and Failure from the result set

Boosters Carried Maximum Payload

Explanation

- We can query Booster_Version that carried the highest payload mass by using a subquery to query specifically for the maximum Payload from the table

Task 8

List all the booster_versions that have carried the maximum payload mass. Use a subquery.

```
%%sql select Booster_Version
    FROM SPACEXTABLE
    WHERE PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTABLE)
ORDER BY Booster_Version
```

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

Booster_Version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

```
%%sql SELECT substr(Date, 6,2) as Month, Booster_Version, Launch_Site as 'Launch Site'
      FROM SPACEXTABLE
      WHERE Landing_Outcome = 'Failure (drone ship)' AND substr(Date,0,5)='2015'
```

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

Month	Booster_Version	Launch Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

Explanation

- As SQLLITE does not support monthnames, we use substr(Date) to get Month and query by year and Failure on Drone Ships by specifying them in the WHERE clause

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

Task 10

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.

```
%%sql SELECT count(Landing_Outcome) as 'Total Outcomes', Landing_Outcome
      FROM SPACEXTABLE
      WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
      GROUP BY Landing_Outcome
      ORDER BY count(Landing_Outcome) DESC
```

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

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

Explanation

- Using the BETWEEN keyword, we can specify the date range for our query and using 'COUNT' we can get total count of each outcome which we can group together using GROUP BY

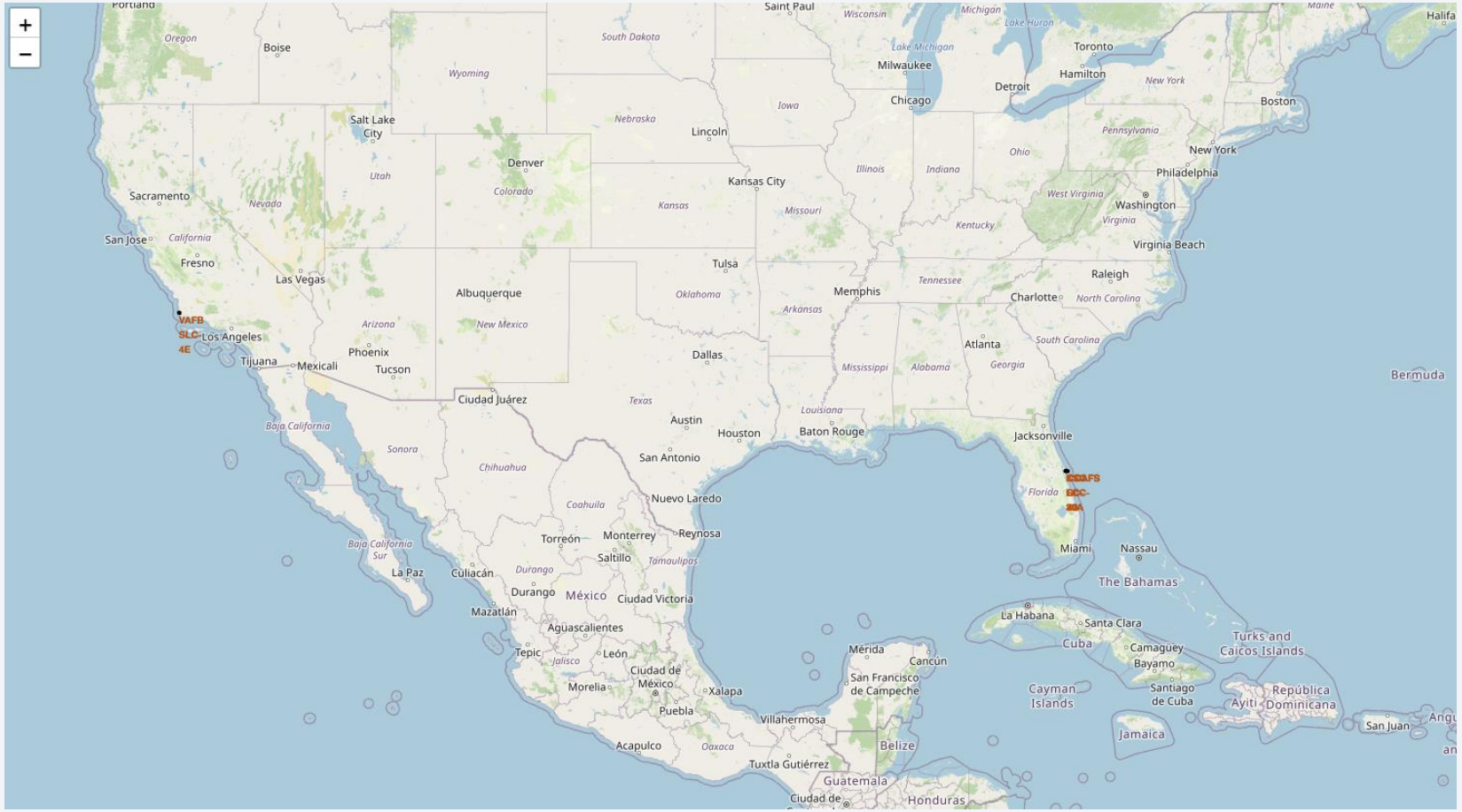
Section 3

Launch Sites Proximities Analysis

Falcon 9 Launch Sites Markers

Explanation

- We can see the launch sites are located in coastal areas and around the equator

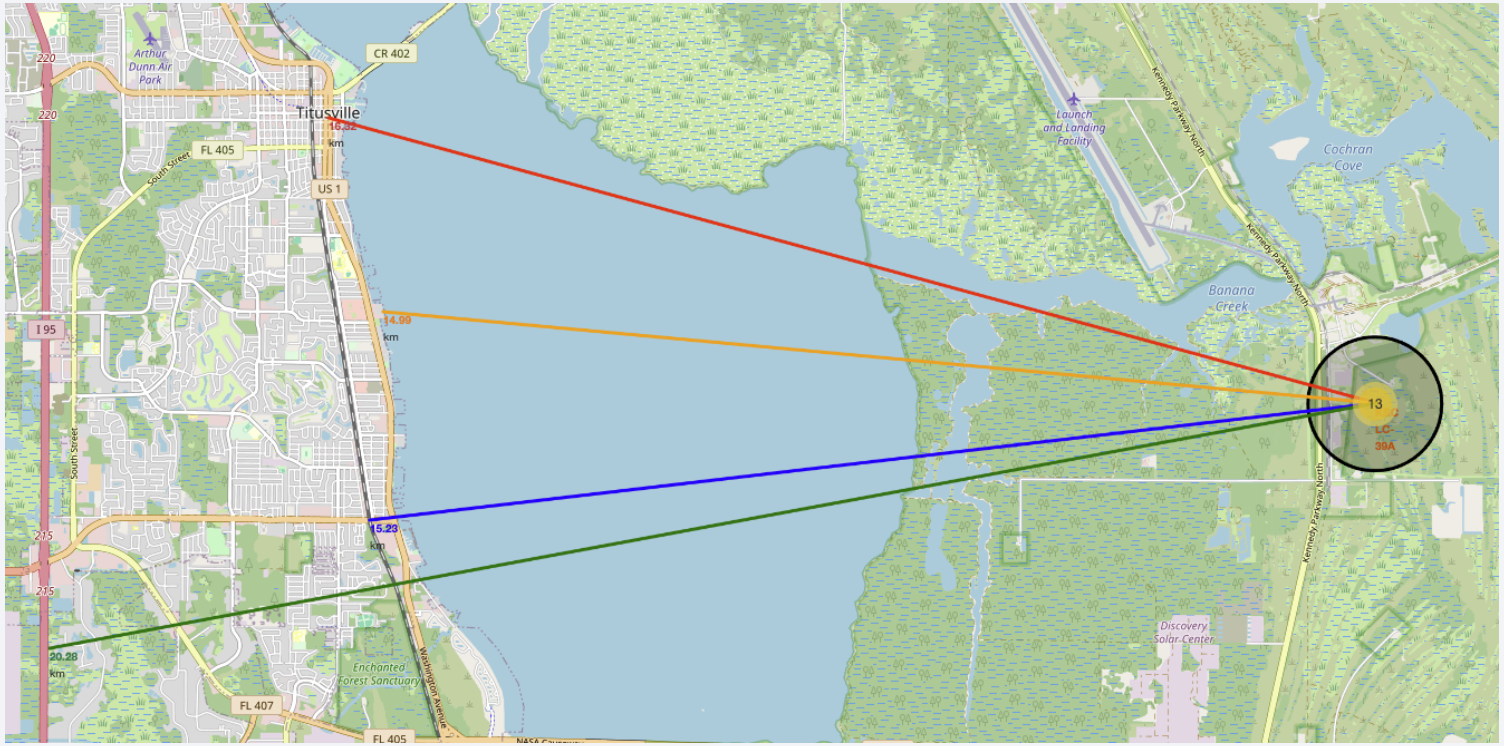


Falcon 9 Launch Outcomes per Launch Sites

Explanation

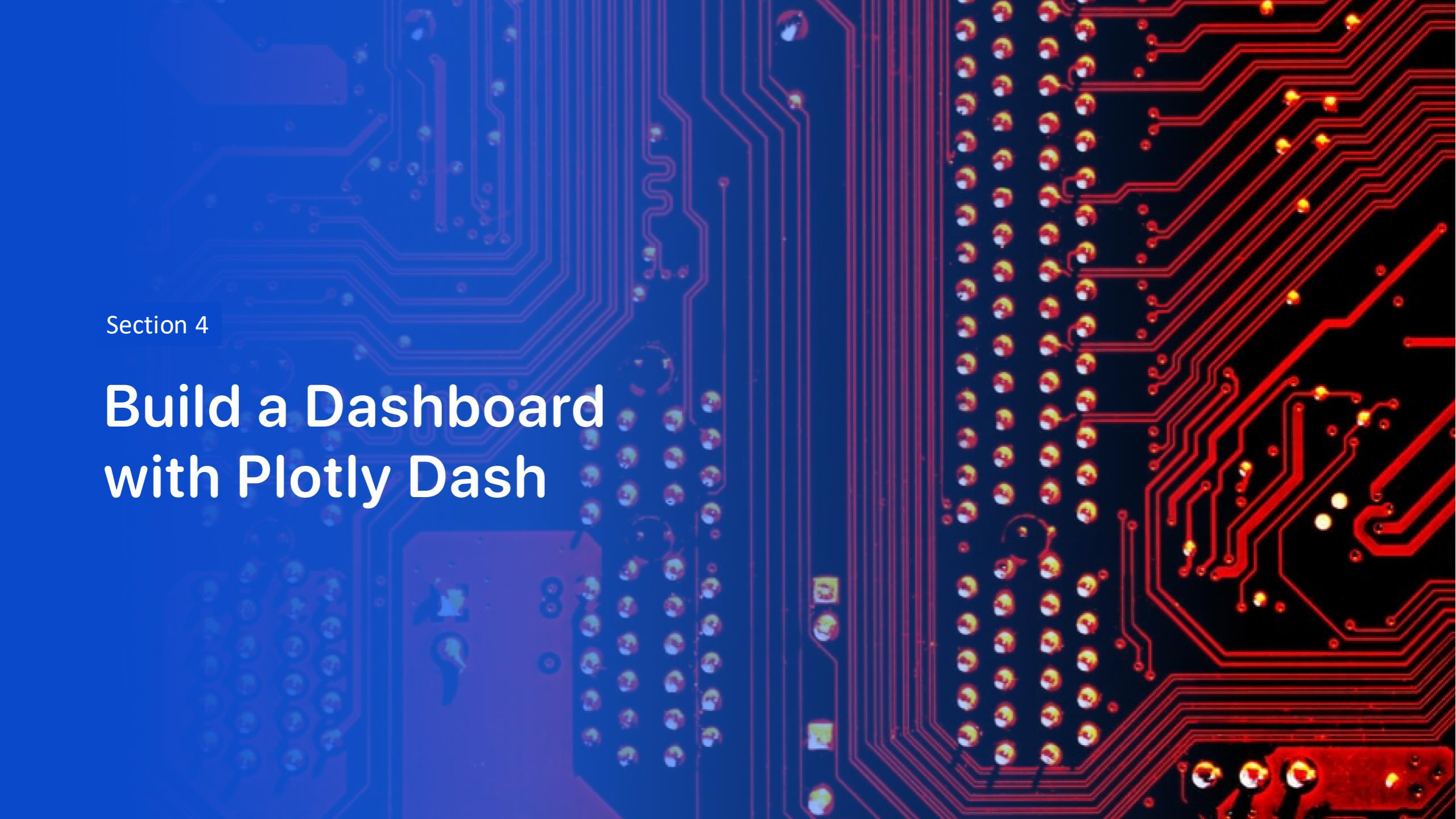
- By clicking on each Launch Site location marker, we can dig in and get colored marker indicating the launch outcome at that Launch Site
  - Red – Failed Launch
  - Green – Successful Launch

Distance from Launch Site to nearby proximities

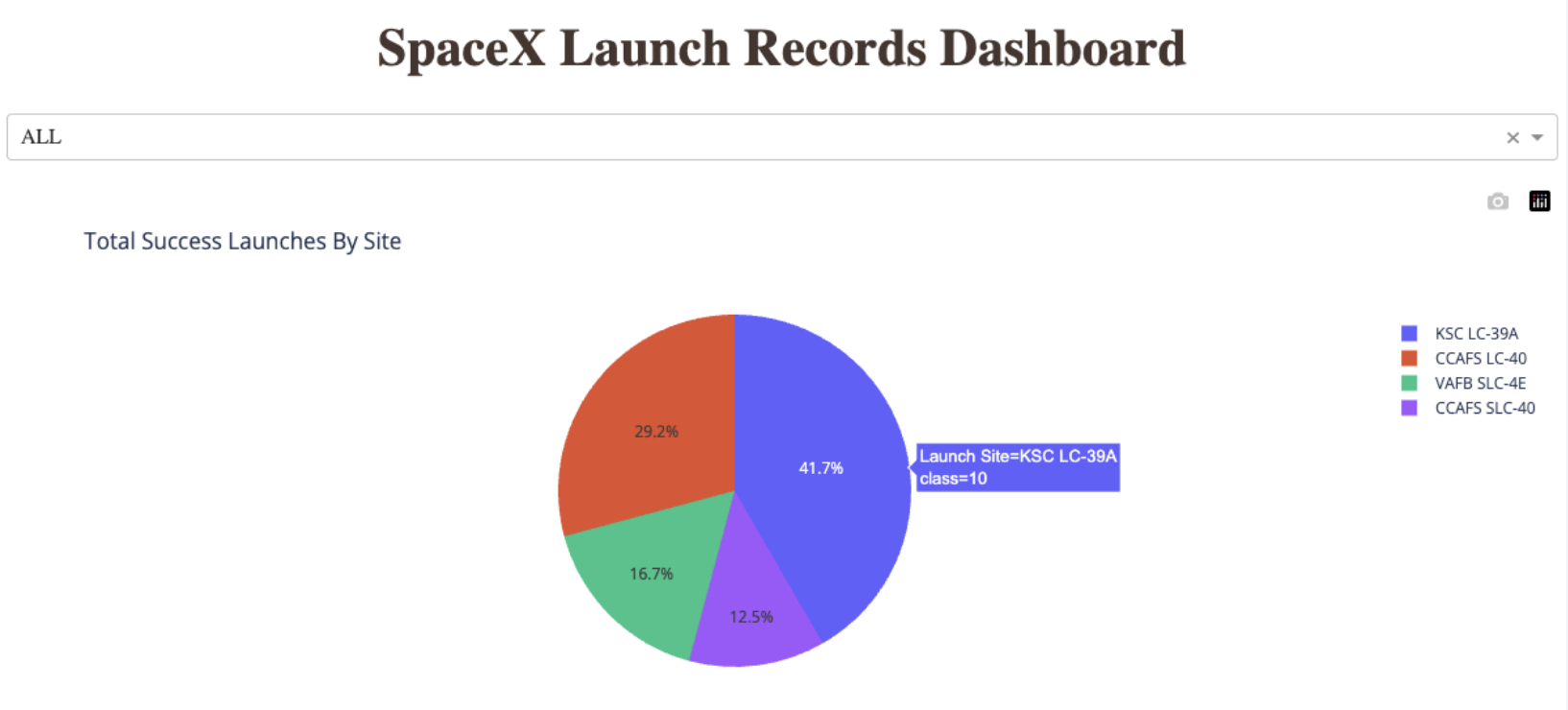


Explanation

- We can use this to determine the distance between the launch site and the nearby populated areas to avoid launch sites which are near these areas. As any failures could lead lot of casualties in these areas.


Section 4
Build a Dashboard with Plotly Dash

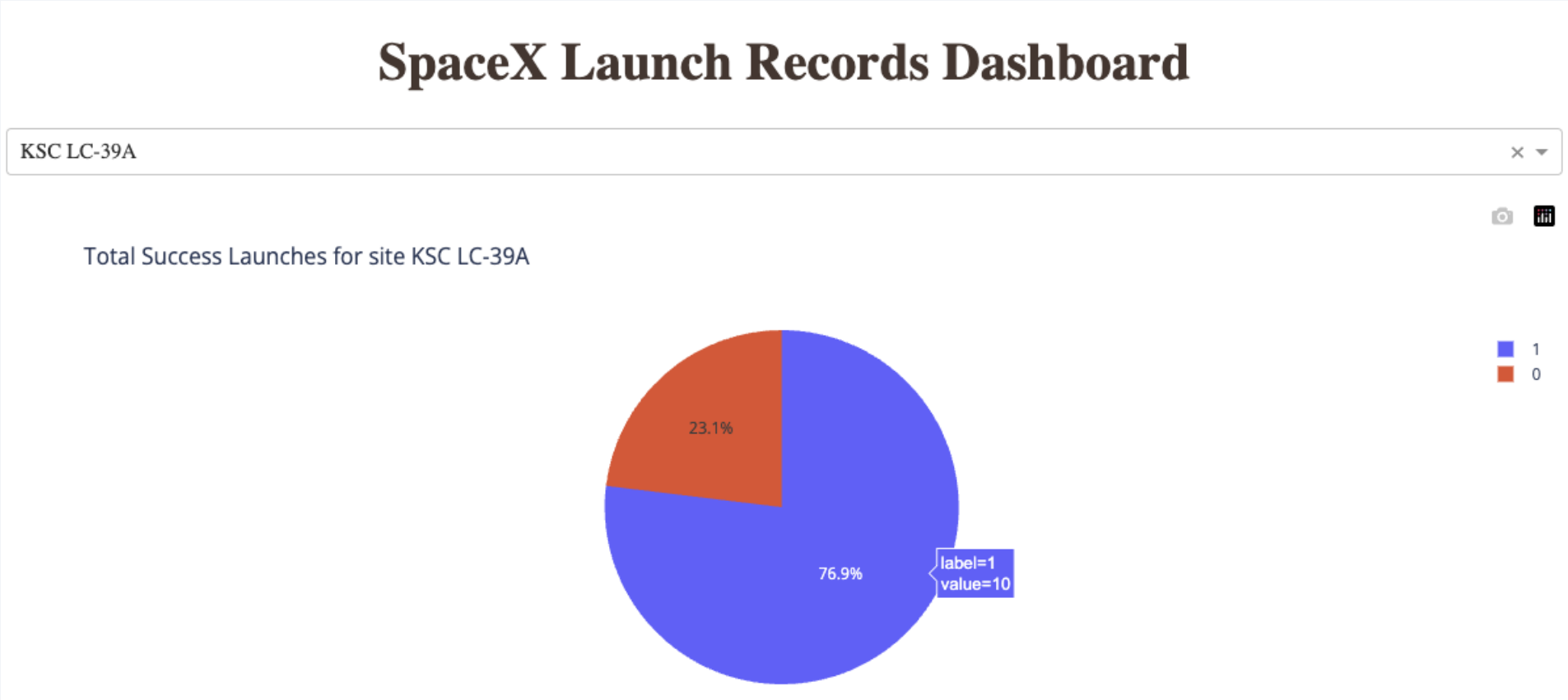
Launch Success Count for all Sites



Explanation

- With this Pie Chart we can see that Launch Site **KSC LC-39A** has the most Success rate

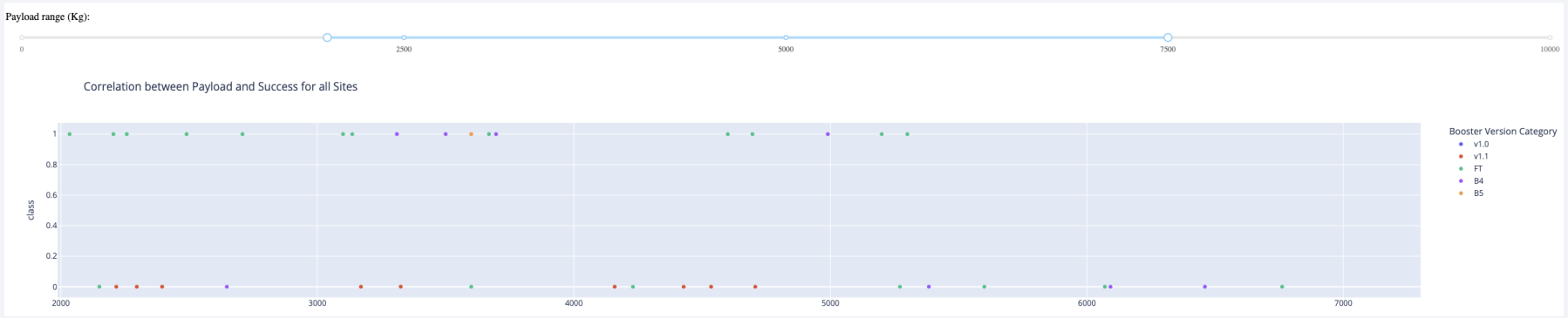
Launch Site with Highest Launch Success Ratio

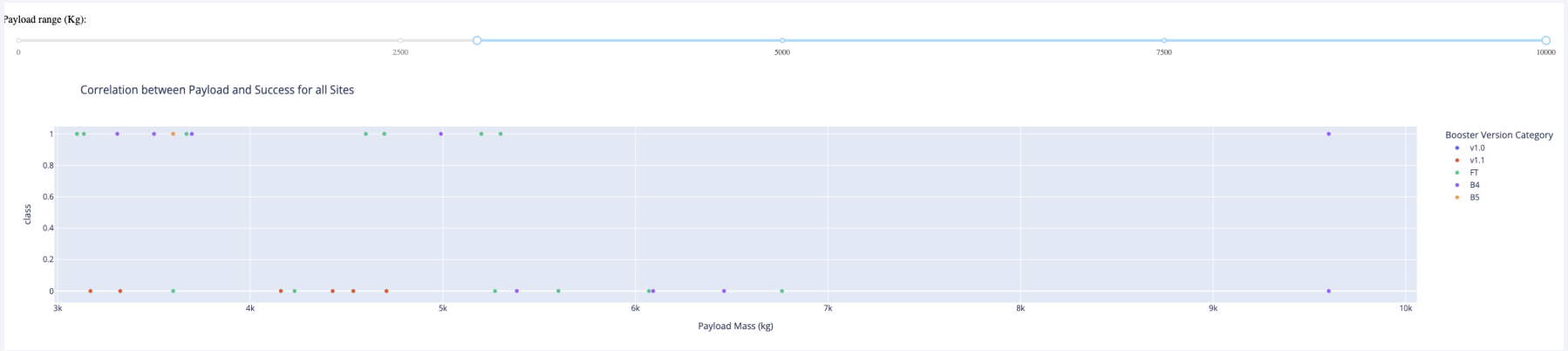


Explanation

- With this Pie Chart we can see that Launch Site **KSC LC-39A** has the most Success rate with 10 Successful landings and 3 Failed landings

Payload vs. Launch Outcome scatter plot for all sites





Explanation

- With this scatter plot we can see certain booster versions have higher success rate with lower payloads instead of higher payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

	LogReg	SVM	Tree	KNN
Jaccard Score	0.833333	0.845070	0.840580	0.819444
F1 Score	0.909091	0.916031	0.913386	0.900763
Accuracy	0.866667	0.877778	0.877778	0.855556
Precision	0.833333	0.845070	0.865672	0.830986
AverageScores	0.860606	0.870987	0.874354	0.851687

Explanation

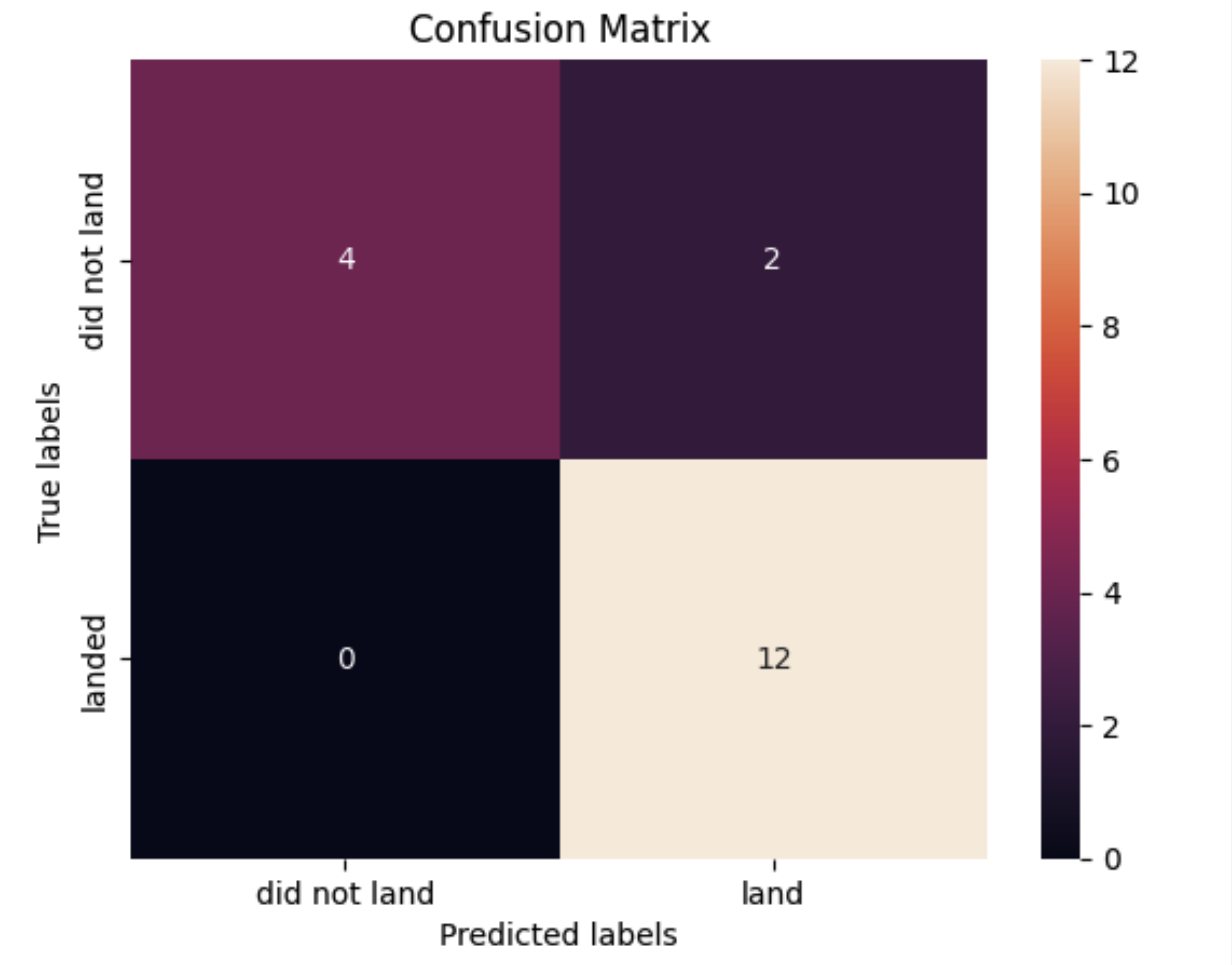
- I chose several scoring metrics to evaluate the binary classification model with the original dataset and took the average of those scores to determine that Decision Tree model is a better classifier for launch outcomes

```
from sklearn.metrics import jaccard_score, f1_score, accuracy_score, recall_score, precision_score

jaccardScores = [
                  jaccard_score(Y, logreg_cv.predict(X), average='binary'),
                  jaccard_score(Y, svm_cv.predict(X), average='binary'),
                  jaccard_score(Y, tree_cv.predict(X), average='binary'),
                  jaccard_score(Y, knn_cv.predict(X), average='binary'),
                 ]

f1Scores = [
             f1_score(Y, logreg_cv.predict(X), average='binary'),
             f1_score(Y, svm_cv.predict(X), average='binary'),
             f1_score(Y, tree_cv.predict(X), average='binary'),
             f1_score(Y, knn_cv.predict(X), average='binary'),
            ]

accuracy = [
             accuracy_score(Y, logreg_cv.predict(X)),
             accuracy_score(Y, svm_cv.predict(X)),
             accuracy_score(Y, tree_cv.predict(X)),
             accuracy_score(Y, knn_cv.predict(X)),
            ]

#recall = [
#             recall_score(Y, logreg_cv.predict(X), average='binary'),
#             recall_score(Y, svm_cv.predict(X), average='binary'),
#             recall_score(Y, tree_cv.predict(X), average='binary'),
#             recall_score(Y, knn_cv.predict(X), average='binary'),
#            ]

precision = [
             precision_score(Y, logreg_cv.predict(X), average='binary'),
             precision_score(Y, svm_cv.predict(X), average='binary'),
             precision_score(Y, tree_cv.predict(X), average='binary'),
             precision_score(Y, knn_cv.predict(X), average='binary'),
            ]
evalScores = pd.DataFrame(np.array([jaccardScores, f1Scores, accuracy, precision]),
                          index=['Jaccard Score', 'F1 Score', 'Accuracy', 'Precision'],
                          columns=['LogReg', 'SVM', 'Tree', 'KNN'])
averageScores = pd.Series(evalScores.mean(axis=0),name='AverageScores')
evalScores = pd.concat([evalScores, averageScores.to_frame().T])
evalScores
```

Confusion Matrix

Explanation

- The Confusion Matrix of the Decision Tree shows:
- TPR is 100% (TP/TP+FN)
- Precision is 86% (TP/TP+FP)
- Accuracy is 88% (Correct/Total Classified)



Conclusions

- Initial flights irrespective of the launch site or payloads had higher Failures, this means SpaceX is doing a good job assessing previous launch failures and rectifying the issues
- KSC LC 39A Launch Site has better successful launches compared to other launch sites
- ES-L1, GEO, HEO and SSO Orbits have 100% Success Rate while SO has 0%
- Launch sites are located around Coastal Areas and Equator
- Launch Site KSC LC-39A has the most Success rate
- Decision Tree proves to be the better Binary Classifier model to predict success rate for Falcon 9 Rocket Launches

Appendix

- GitHub Link to Capstone Project: https://github.com/alwinantony/IBM-DataScience-Captstone/tree/main

Thank you!