IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Akash Nigale
22th Nov 2023

Outline

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

Executive Summary

- Data collection, data wrangling, exploratory data analysis, interactive visual analytics and predictive analysis
- Launch site KSC LC-39A using booster version FT with payload less than 6000 kg in orbit ES-L1, GEO, HEO and SSO have the highest rate of success
- Logistic regression, SVM and KNN are the classification models that can best predict the landing outcomes

Introduction

- 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.
- We will predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Request to the SpaceX API
  - Clean the requested data
  - Extract a Falcon 9 launch records HTML table from Wikipedia
  - Parse the table and convert it into a Pandas data frame
- Perform data wrangling
  - Perform exploratory data analysis
  - Determine training labels

Methodology

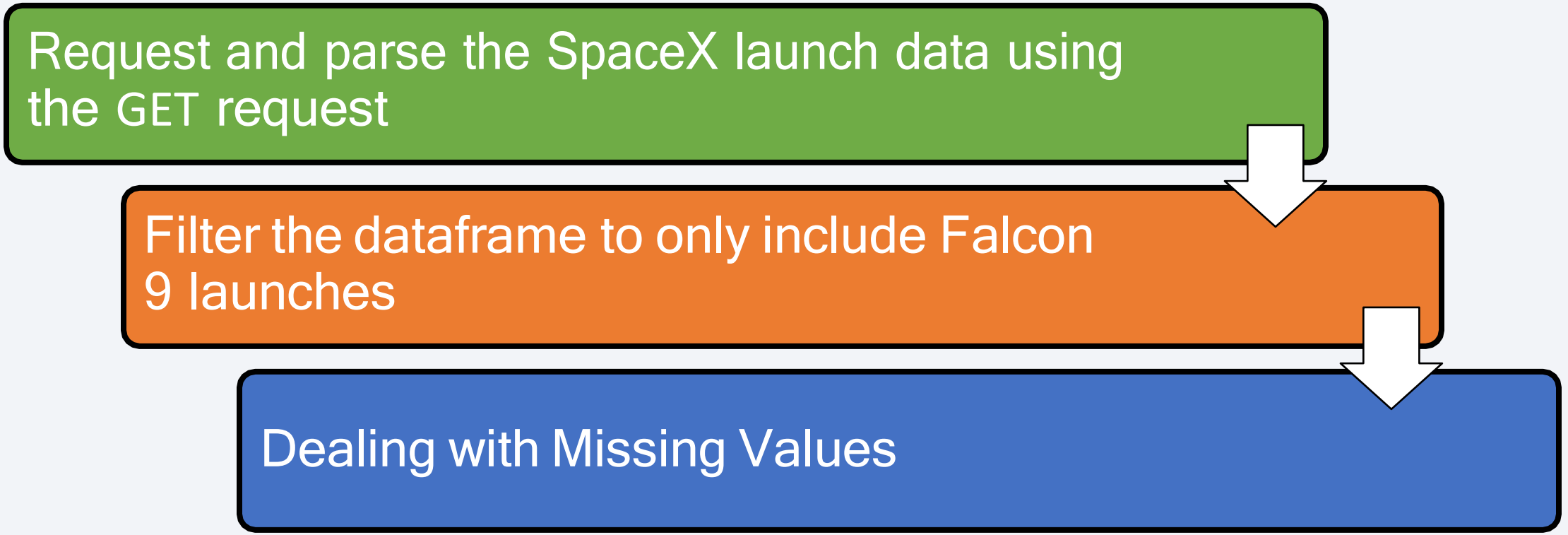
Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Perform exploratory data analysis and determine training labels
  - Create a column for the class
  - Standardize the data
  - Split into training data and test data
  - Find best hyperparameter for SVM, Decision Trees and Logistic Regression
  - Find the method performs best using test data

Data Collection

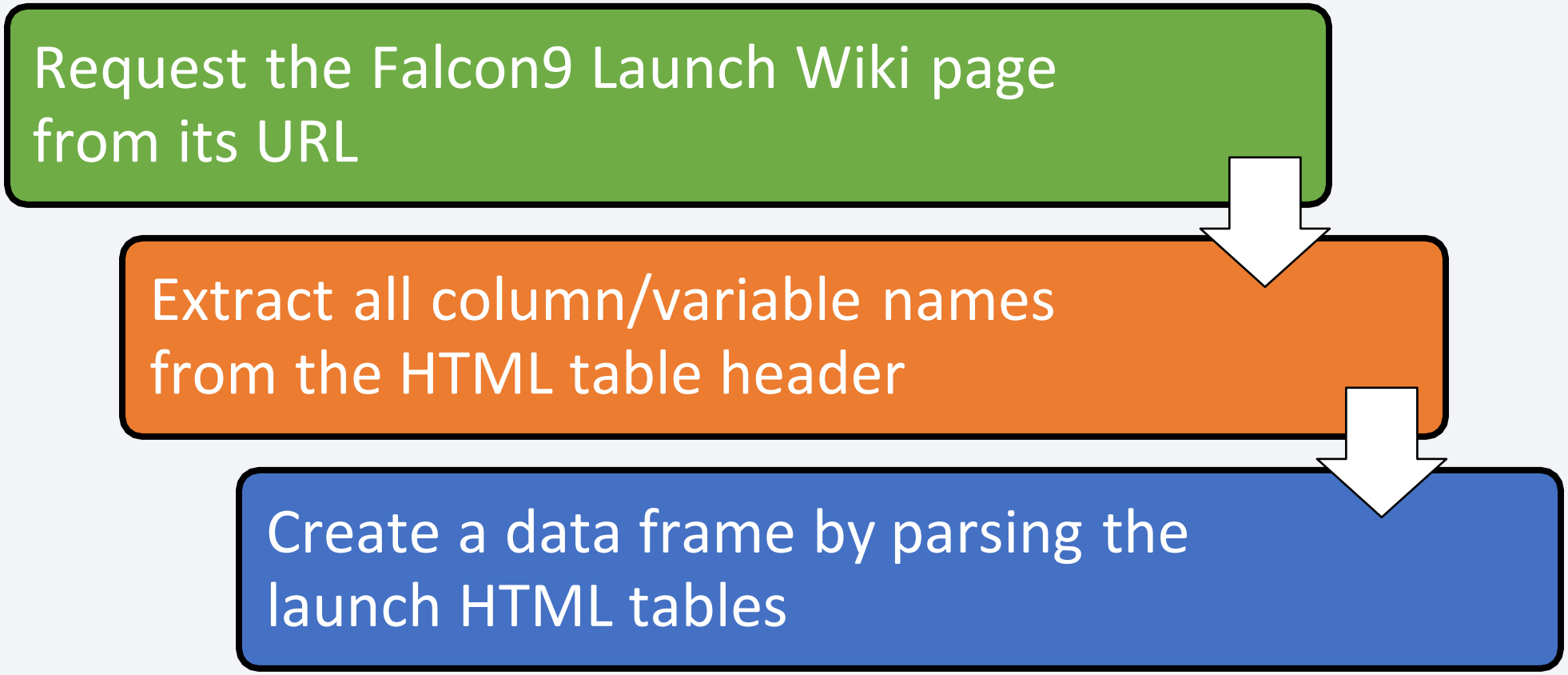
- Collect data and ensure the data is in the correct format from an API
- Web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches
  https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Perform Exploratory Data Analysis (EDA) to find patterns in the data and determine what would be the label for training supervised models.

Data Collection - SpaceX API



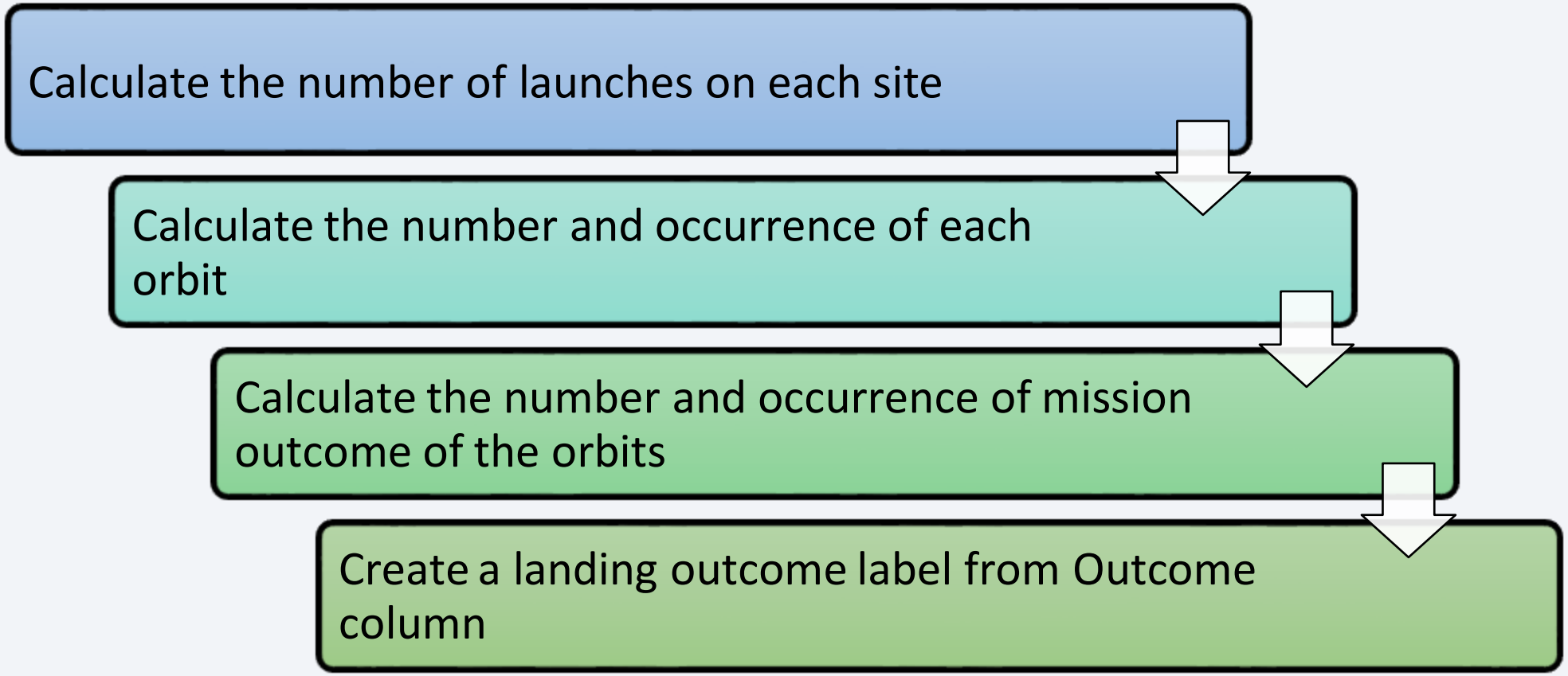
https://github.com/nigale222/COURSERAMATERIALS/blob/ad312deac5c4de8f2f8a0c0f981847d3c1f31623/jupyter-labs-spacex-data-collection-api%20(1).ipynb

Data Collection - Scraping



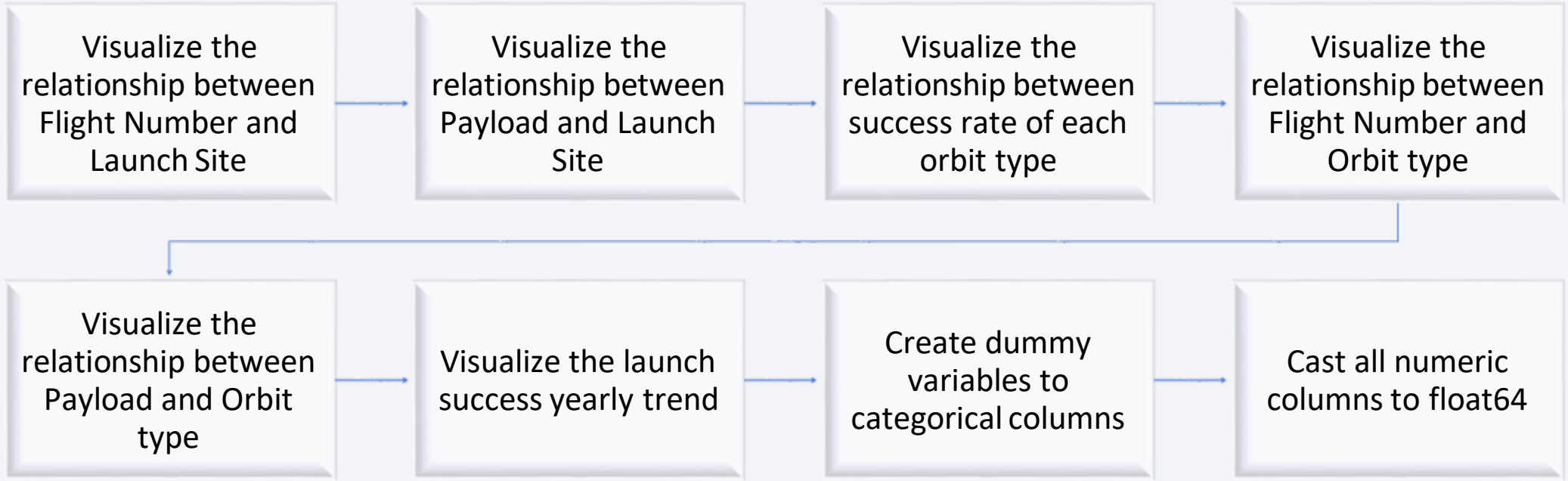
https://github.com/nigale222/COURSERAMATERIALS/blob/ad312deac5c4de8f2f8a0c0f981847d3c1f31623/jupyter-labs-webscraping.ipynb

Data Wrangling



https://github.com/nigale222/COURSERAMATERIALS/blob/bbe639fd59f23bc4c974b67c920b1249f9d5589e/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization



https://github.com/nigale222/COURSERAMATERIALS/blob/2b806992e0b485d17a0d74563130005c6889899a/jupyter-labs-eda-dataviz%20(1).ipynb

EDA with SQL

- 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 successful landing outcome in ground pad was achieved.
- 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 using 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/nigale222/COURSERAMATERIALS/blob/2b806992e0b485d17a0d74563130005c6889899a/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Mark all launch sites on a map

Mark the success/failed launches for each site on the map

Calculate the distances between a launch site to its proximities

https://github.com/nigale222/COURSERAMATERIALS/blob/2b806992e0b485d17a0d74563130005c6889899a/module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Add a launch site drop-down input component to see which launch site has the largest success rate.
- Add a callback function to render success-pie-chart based on selected site dropdown to render a pie chart visualizing launch success counts.
- Add a payload range slider to find if payload is correlated to mission outcome. From a dashboard point of view, we want to be able to easily select different payload range and see if we can identify some visual patterns.
- Add a callback function to render the success-payload-scatter-chart scatter plot to visually observe how payload may be correlated with mission outcomes for selected site. In addition, we want to color-label the Booster version on each scatter point so that we may observe mission outcomes with different boosters.

https://github.com/nigale222/COURSERAMATERIALS/blob/63d0520cb516e9f89565690966a2ffbb4ed9bba3/spacex_dash_app.py

Predictive Analysis (Classification)

Create a NumPy array from the column Class in data, by applying the method to_numpy then assign it to the variable Y to make sure the output is a Pandas series.

Standardize the data in X then reassign it to the variable X using the transform provided.

Use the function train_test_split to split the data X and Y into training and test data. Set the parameter test_size to 0.2 and random_state to 2. The training data and test data should be assigned to the following labels.

X_train, X_test, Y_train, Y_test

https://github.com/nigale222/COURSERAMATERIALS/blob/63d0520cb516e9f89565690966a2ffbb4ed9bba3/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Predictive Analysis (Classification)

Create a logistic regression object then create a GridSearchCV object logreg_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters .

We output the GridSearchCV object for logistic regression. We display the best parameters using the data attribute best_params_and the accuracy on the validation data using the data attribute best_score_

Calculate the accuracy on the test data using the method score and plot confusion matrix.

Repeat the same with Support Vector Machine, Decision Tree Classifier and K-Nearest Neighbours and visualize their accuracy scores to compare the best result

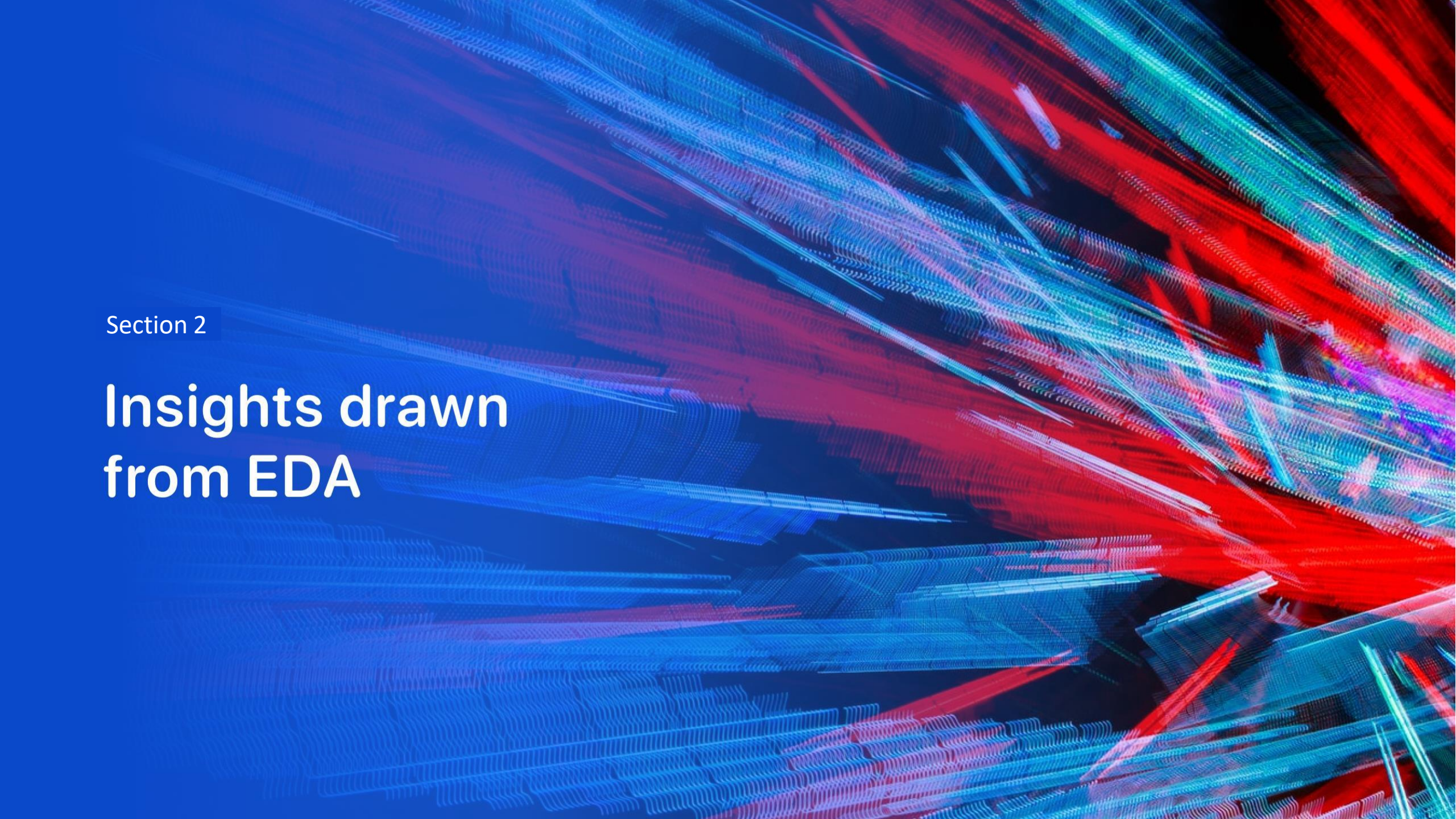
https://github.com/nigale222/COURSERAMATERIALS/blob/63d0520cb516e9f89565690966a2ffbb4ed9bba3/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

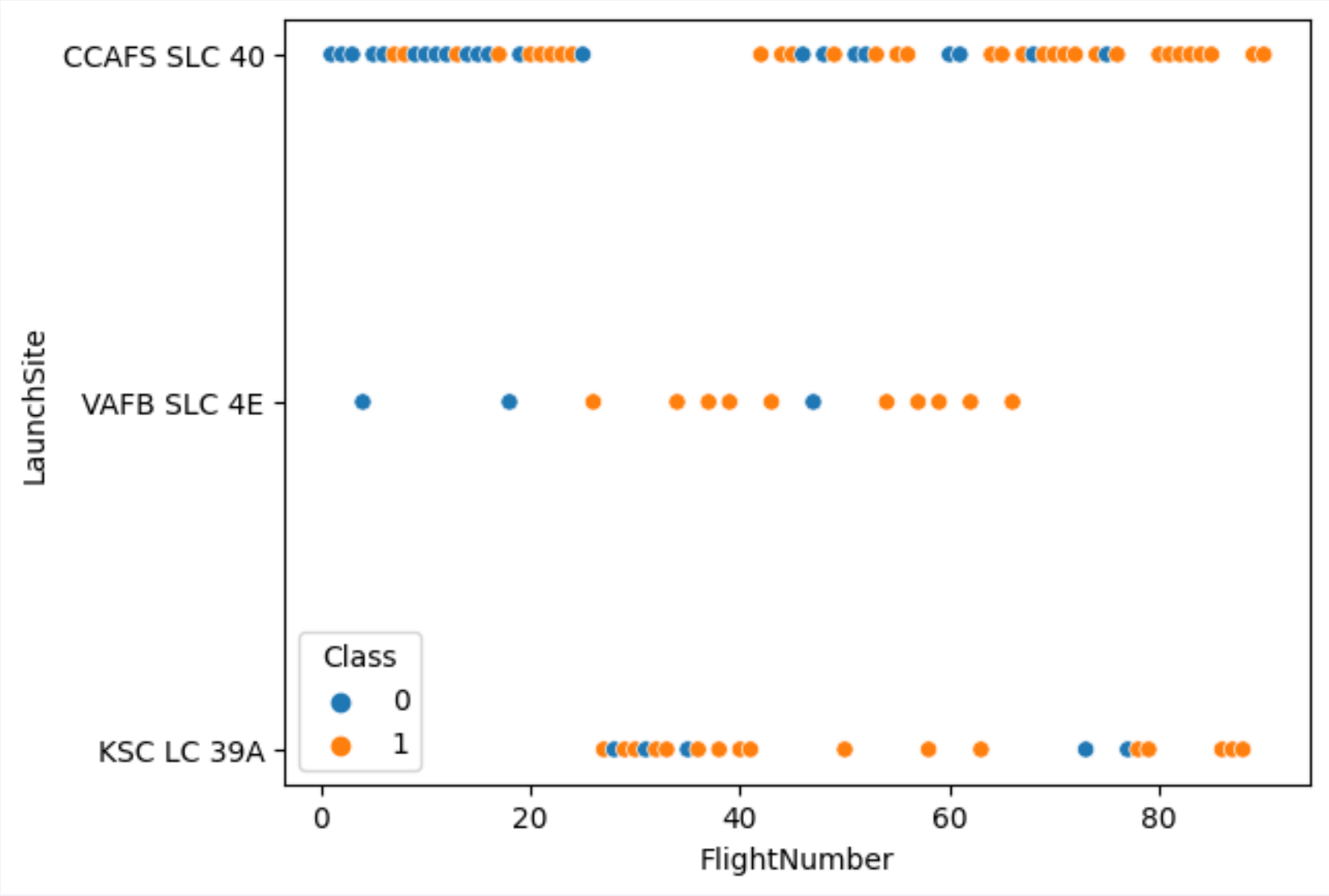
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

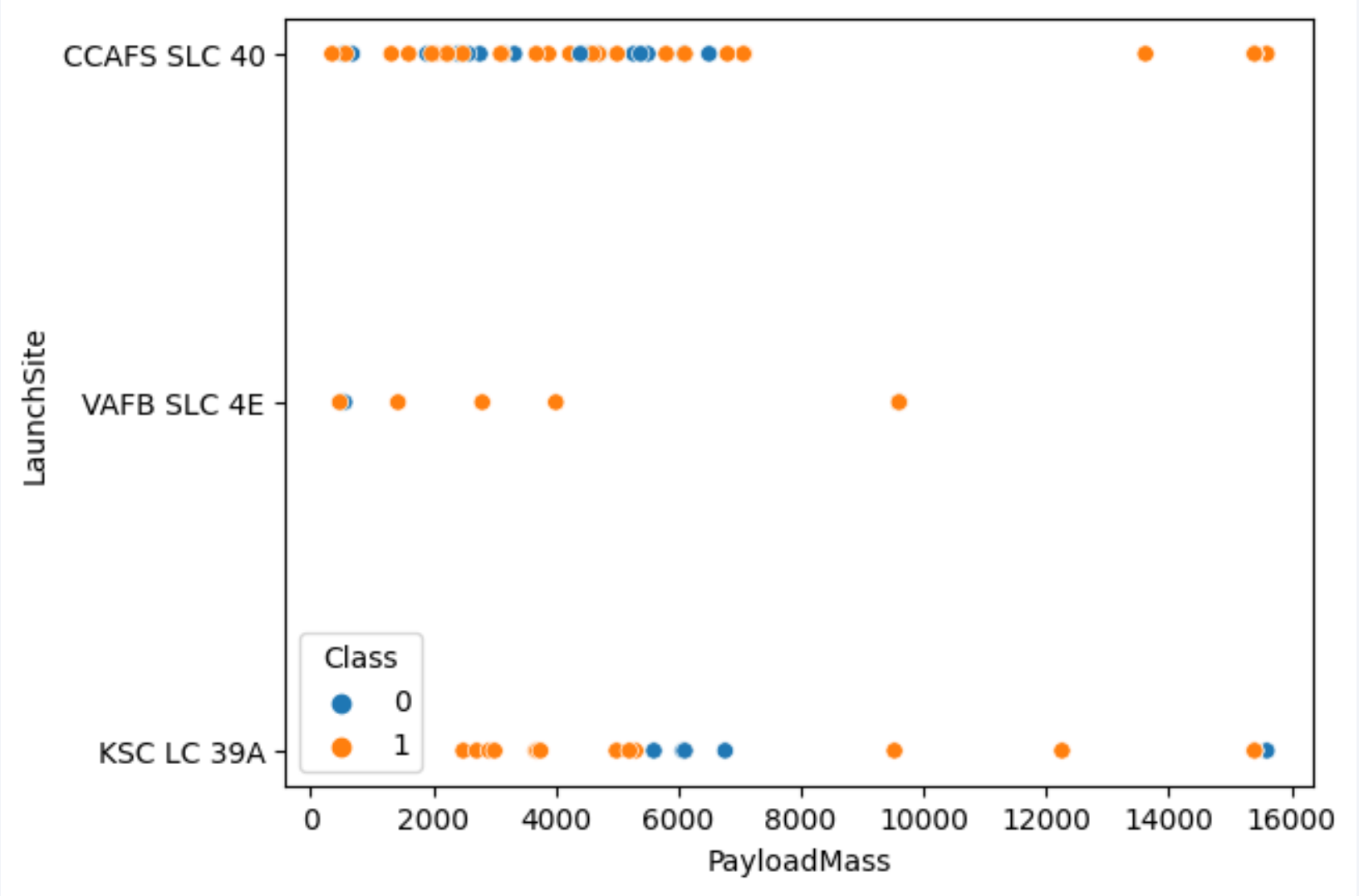


Flight Number vs. Launch Site



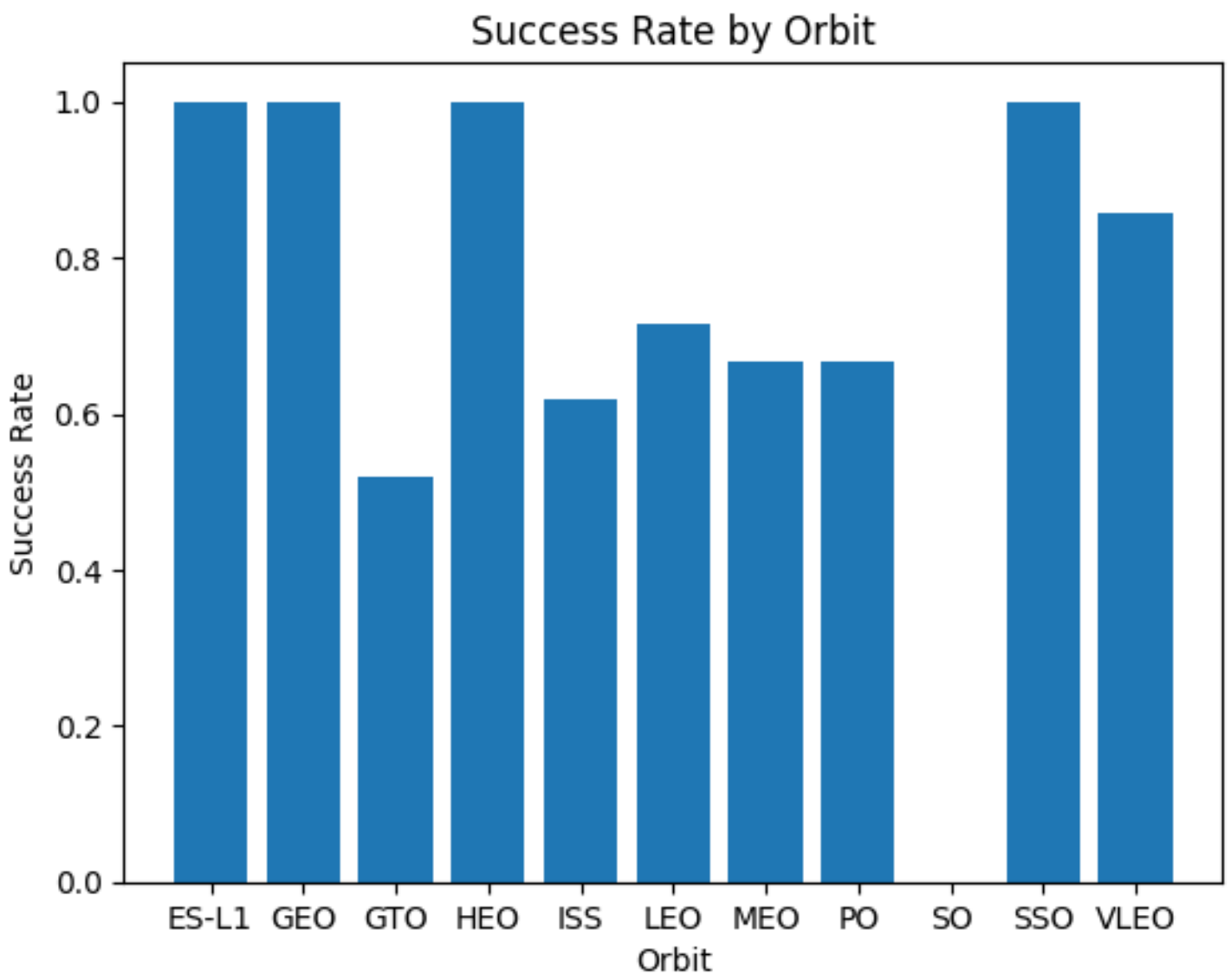
All flight numbers above 80 are successful for CCAFS SLC 40 and KSC LC 39A launch sites.

Payload vs. Launch Site



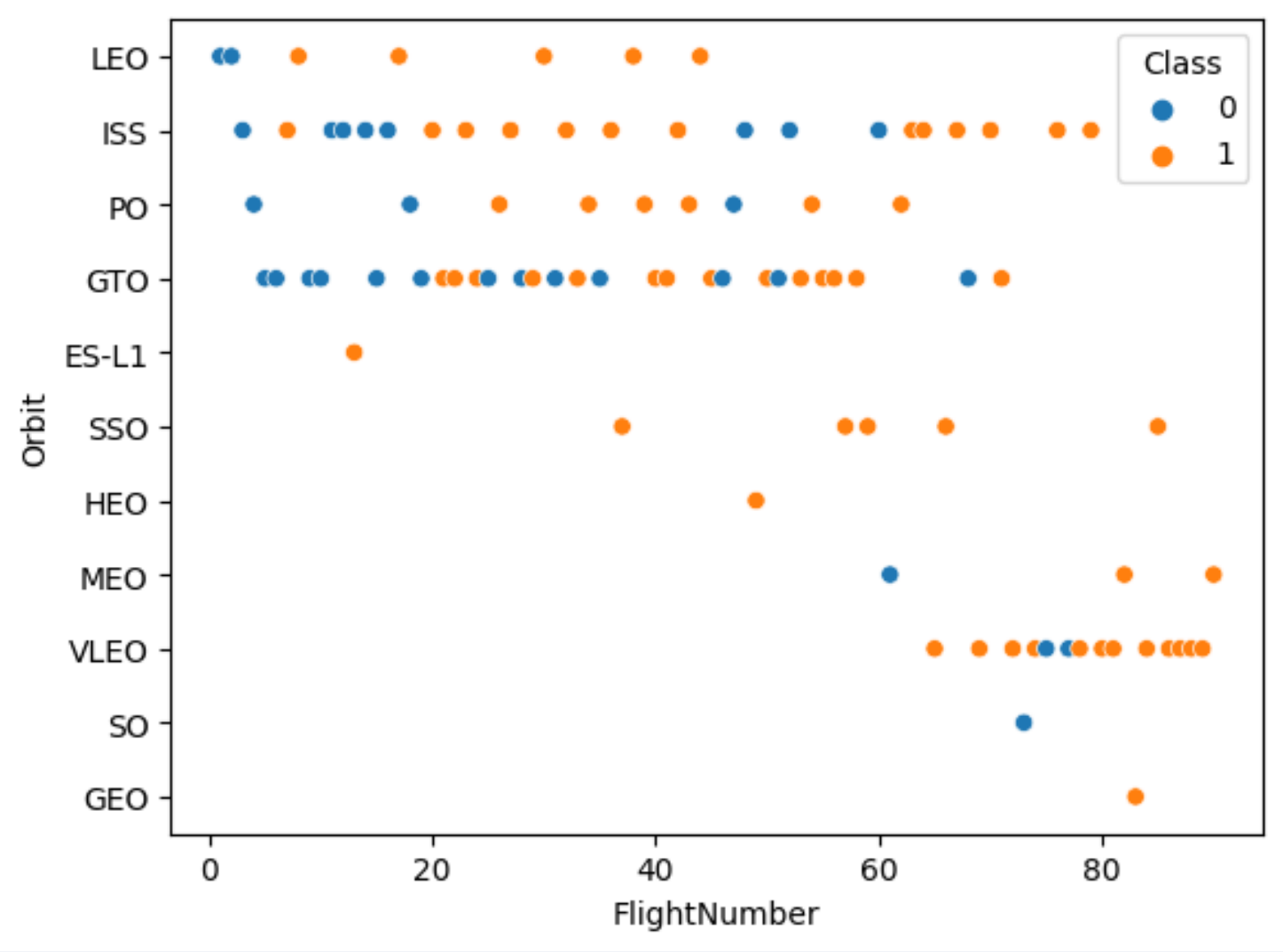
There are no rockets launched for heavy payload mass (*more than 10,000 kg*) from VAFB SLC 4E launch site.

Success Rate vs. Orbit Type



Orbit ES-L1, GEO, HEO and SSO have highest success rate of 100%

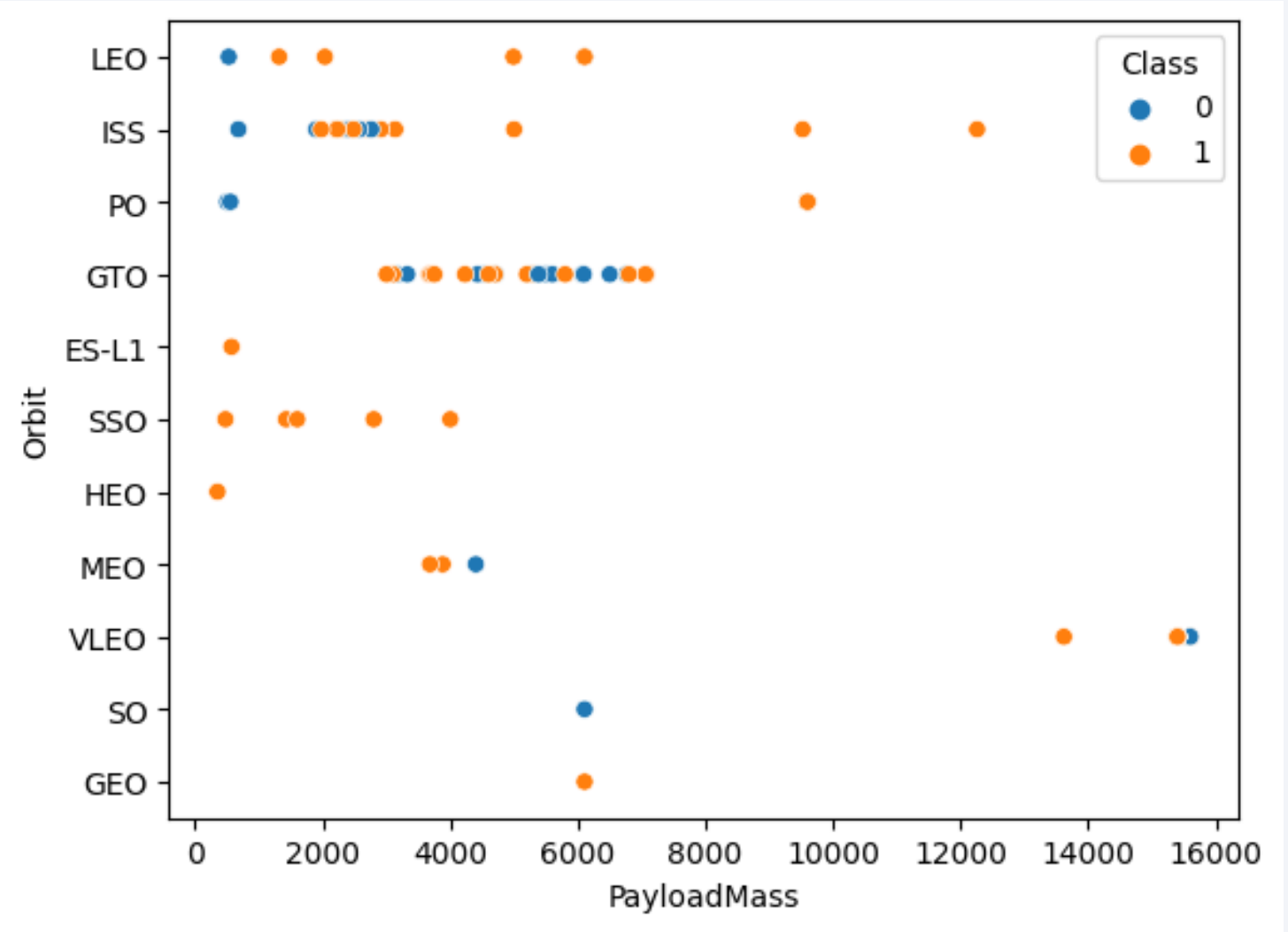
Flight Number vs. Orbit Type



In the LEO orbit, the success rate appears related to the number of flights.

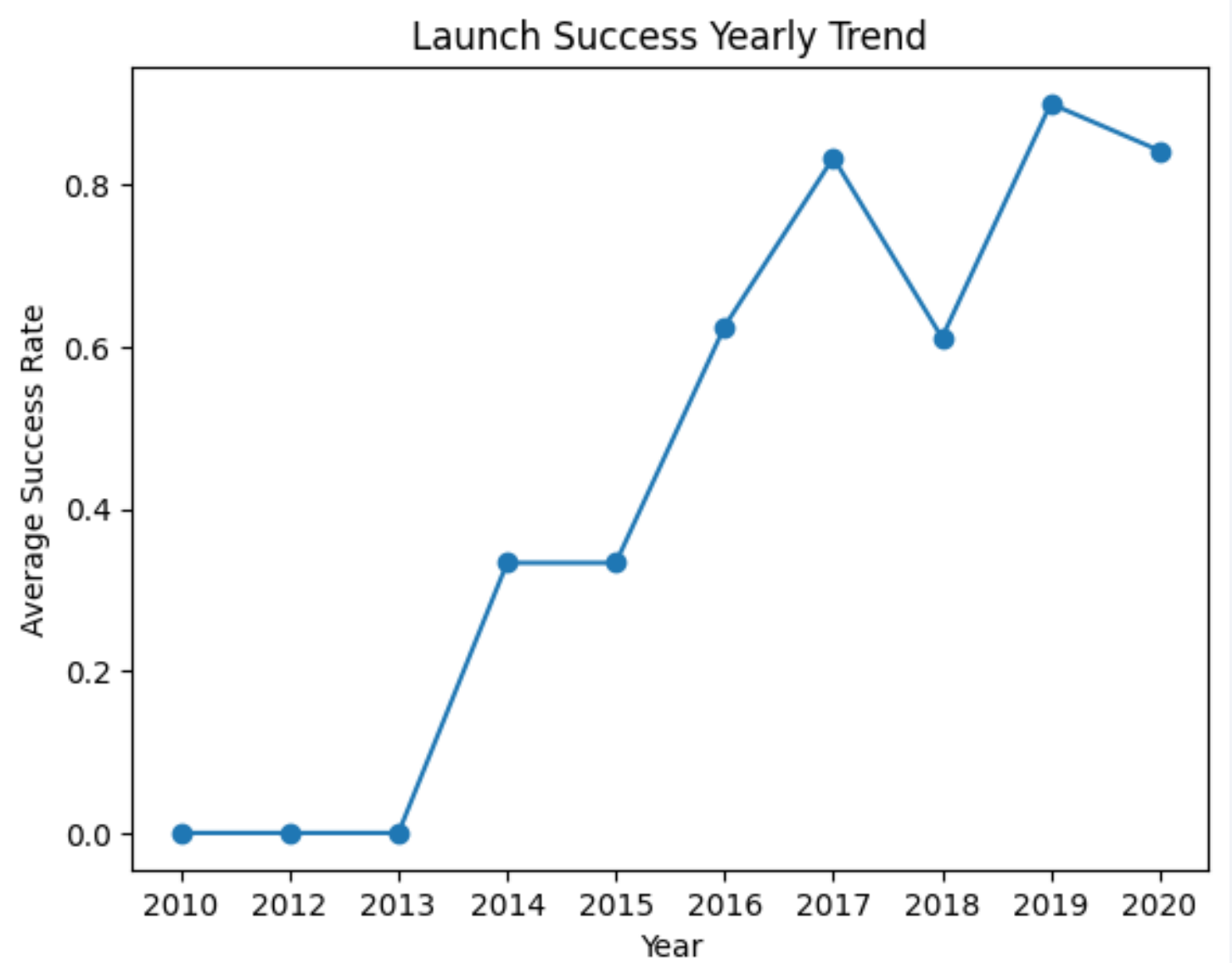
Payload vs. Orbit Type

- ISS and PO have high successful landing rate for heavy payload.
- SSO have high successful landing rate for light payload.



Launch Success Yearly Trend

The success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.



All Launch Site Names

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

There are 4 unique launch sites

Launch Site Names Begin with 'CCA'

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

All launch site names that begin with ‘CCA’ are CCAFS LC-40

Total Payload Mass

SUM(PAYLOAD_MASS__KG_)
45596

Total payload carried by boosters from NASA is 45596 kg

Average Payload Mass by F9 v1.1

AVG(PAYLOAD_MASS__KG_)
2928.4

The average payload mass carried by booster version F9 v1.1 is 2928.4 kg

First Successful Ground Landing Date

MIN(Date)
2015-12-22

The date of the first successful landing outcome on ground pad is 22nd December 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

Here is the list of the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 kg

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

Total Number of Successful and Failure Mission Outcomes

Mission_Outcome	Total
Success	98

The total number of successful mission outcomes is 98

Boosters Carried Maximum Payload

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

The names of the booster which have carried the maximum payload mass are booster version F9 B5

2015 Launch Records

Month	Landing_Outcome	Booster_Version	Launch_Site
October	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Both failed landing outcomes took place in 2015 is drone ship of booster version F9 v1.1 and at CCAFS LC-40 launch site.

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

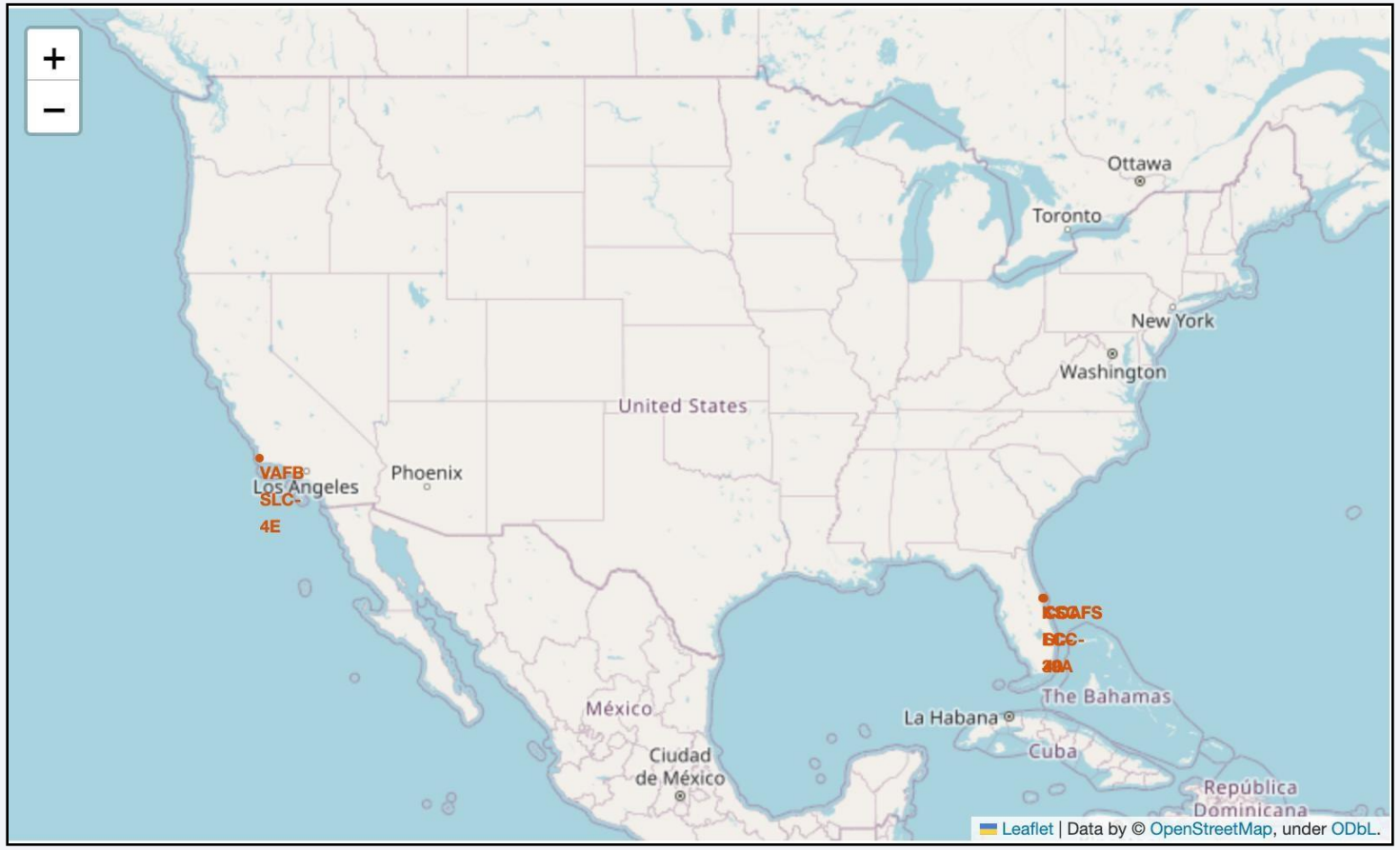
Landing_Outcome	OutcomeCount
Success (ground pad)	5
Failure (drone ship)	5

Equal success and failure outcome count between the date 2010-06-04 and 2017-03-2

Section 3

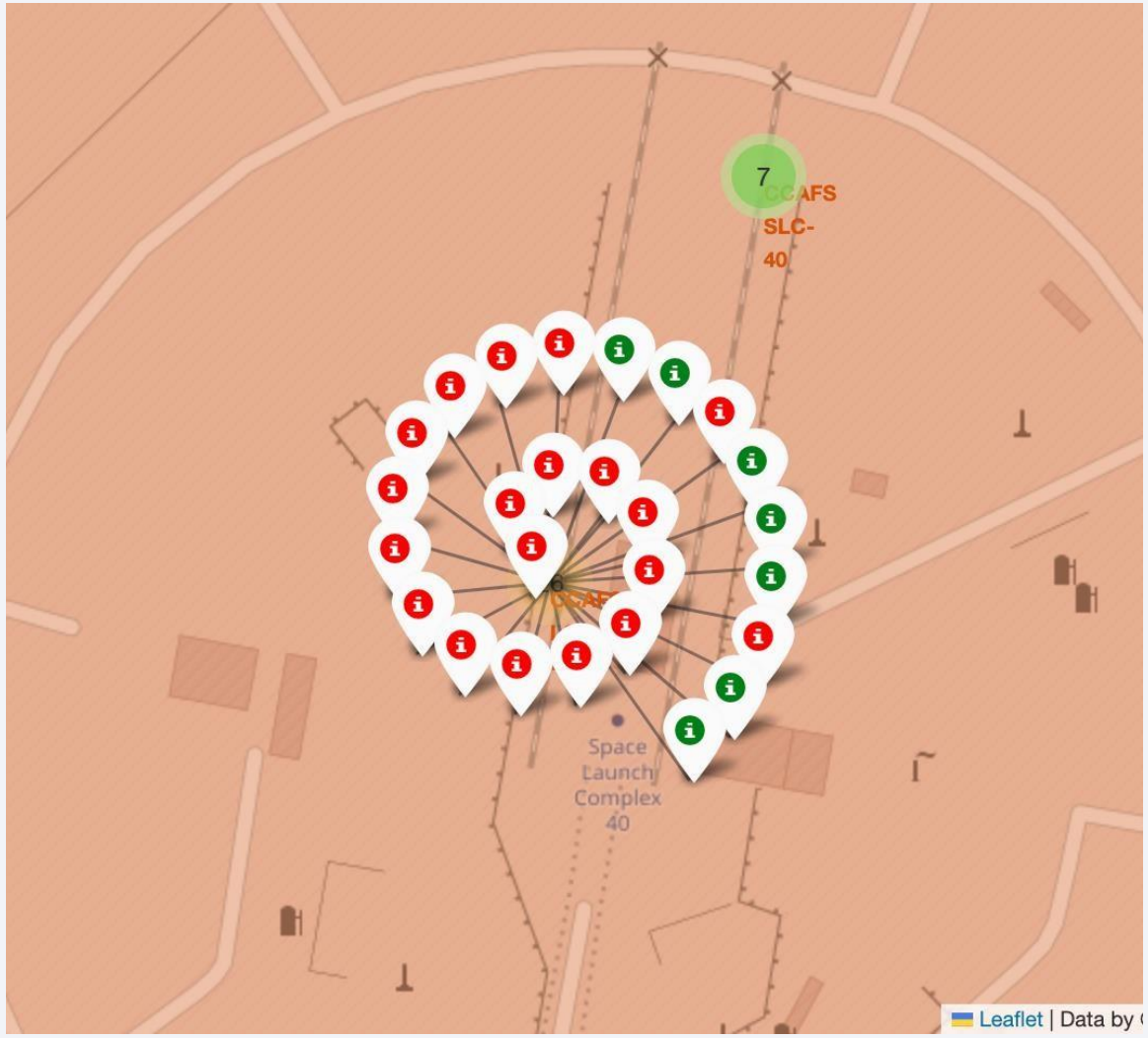
Launch Sites Proximities Analysis

Launch Site Markers



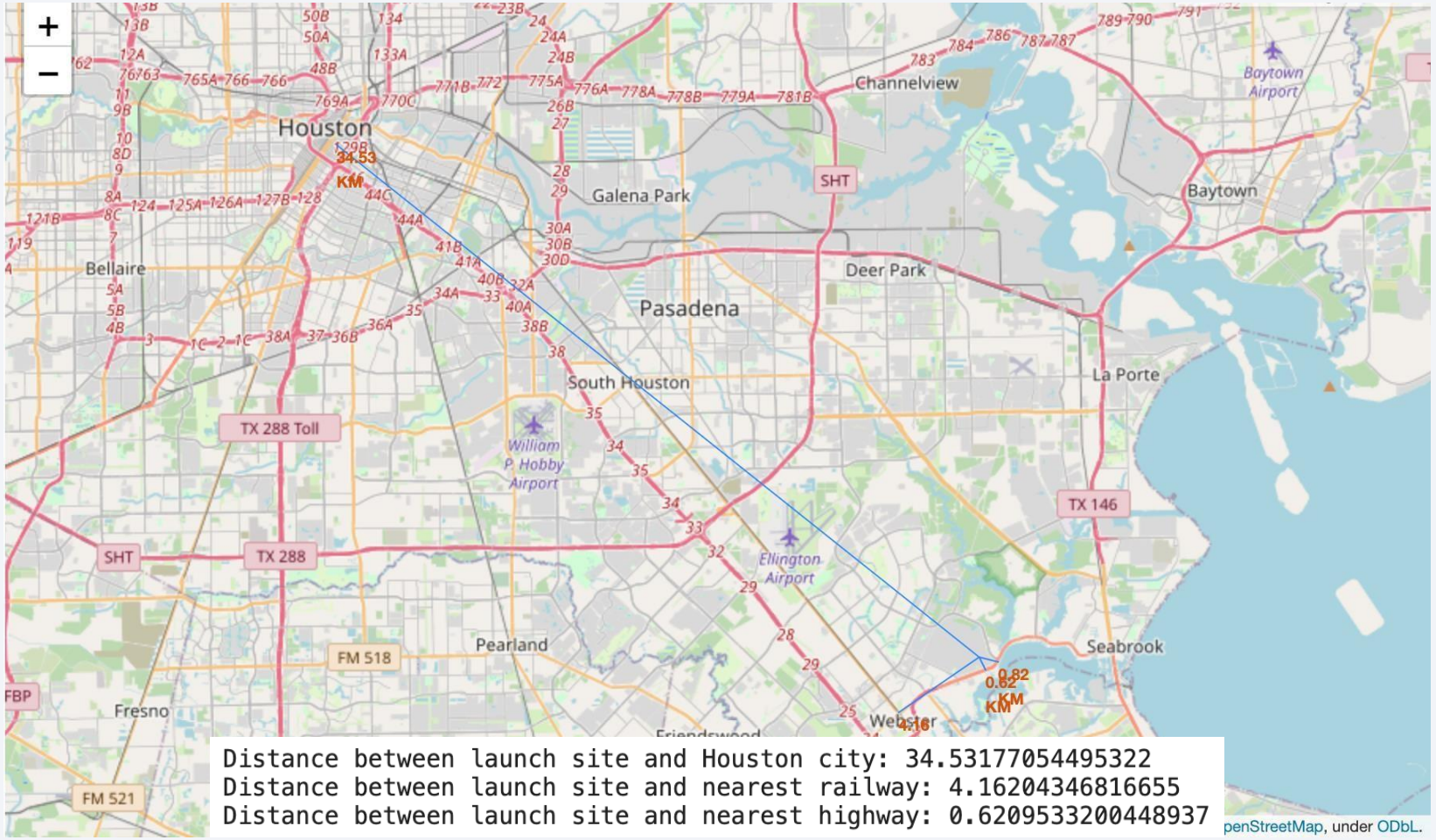
All launch sites are close to the equator line and coast.

Success/failure markers for each launch site

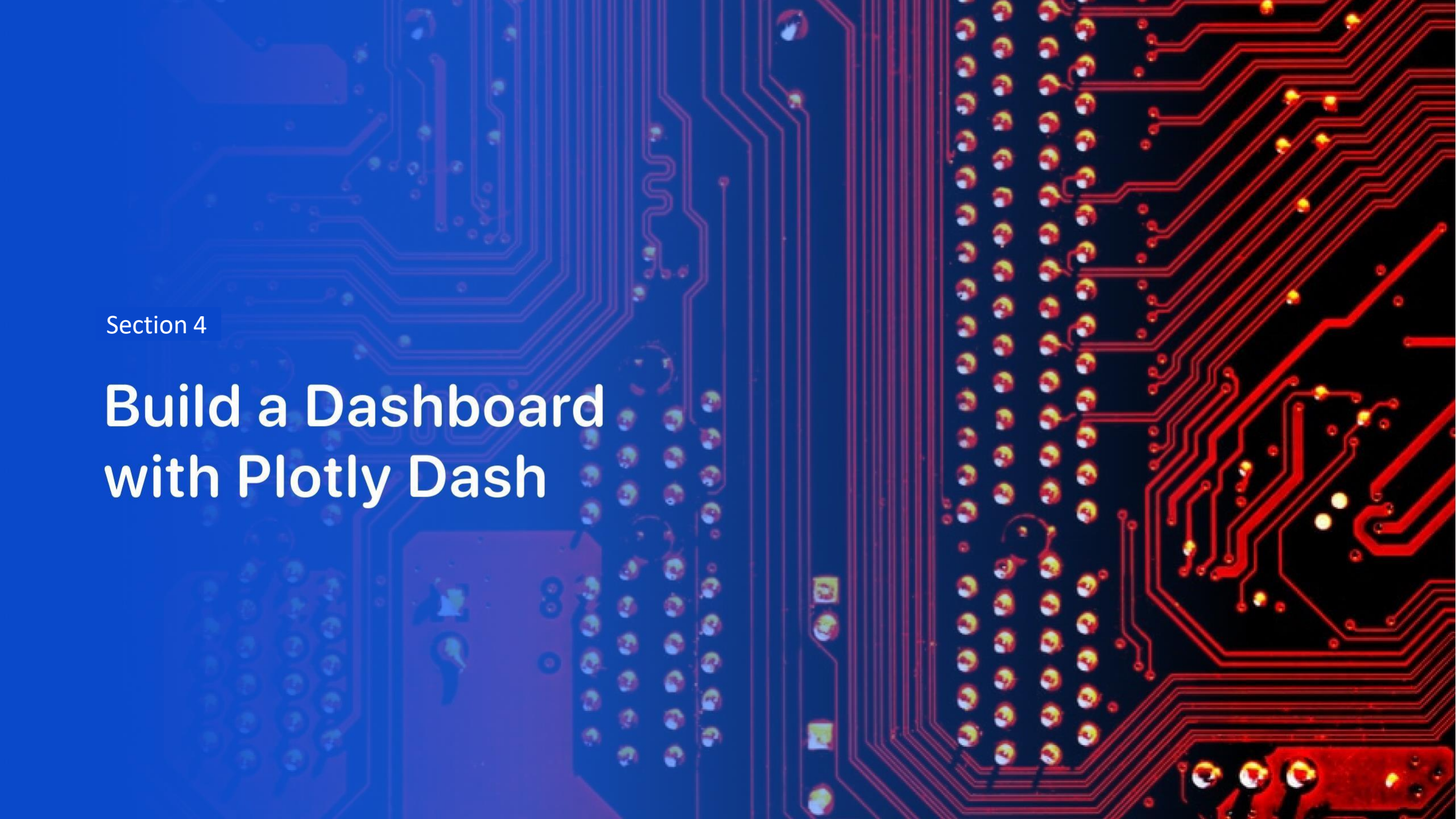


- Created markers for all launch records. If a launch was successful(class=1), then we use a **green** marker and if a launch was failed, we use a **red** marker (class=0)
- **Marker clusters** is used to simplify a map containing many markers having the same coordinate.

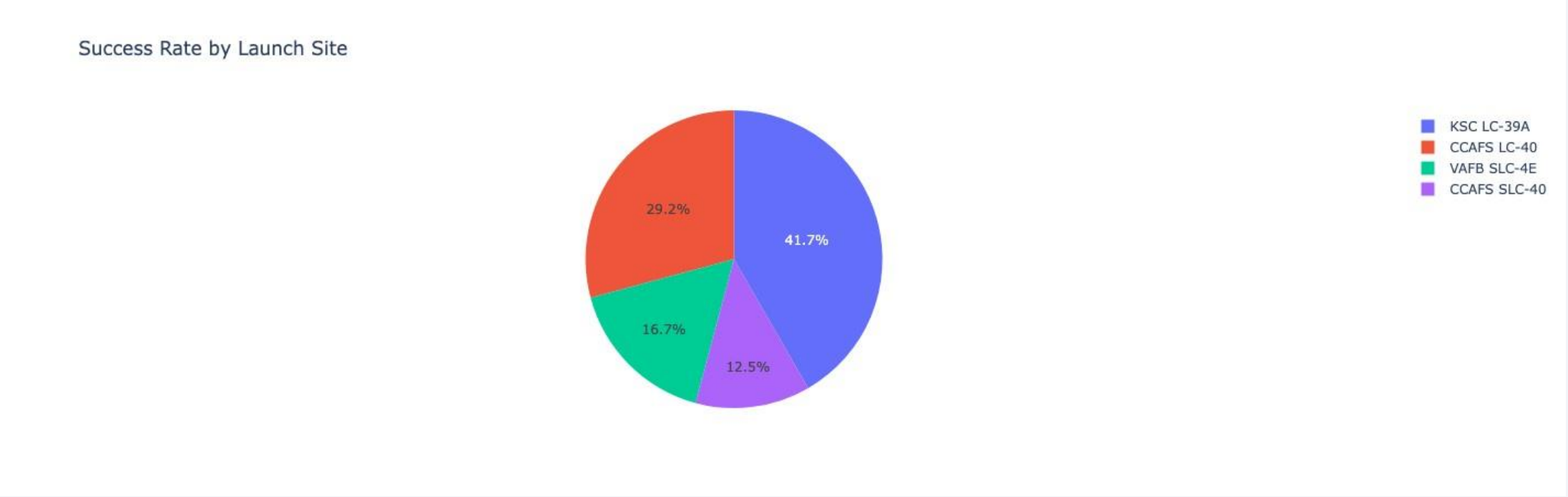
Distances between a launch site to its proximities



- The distance between launch site and nearest highway is the closest
- The distance between launch site and nearest city is the furthest.

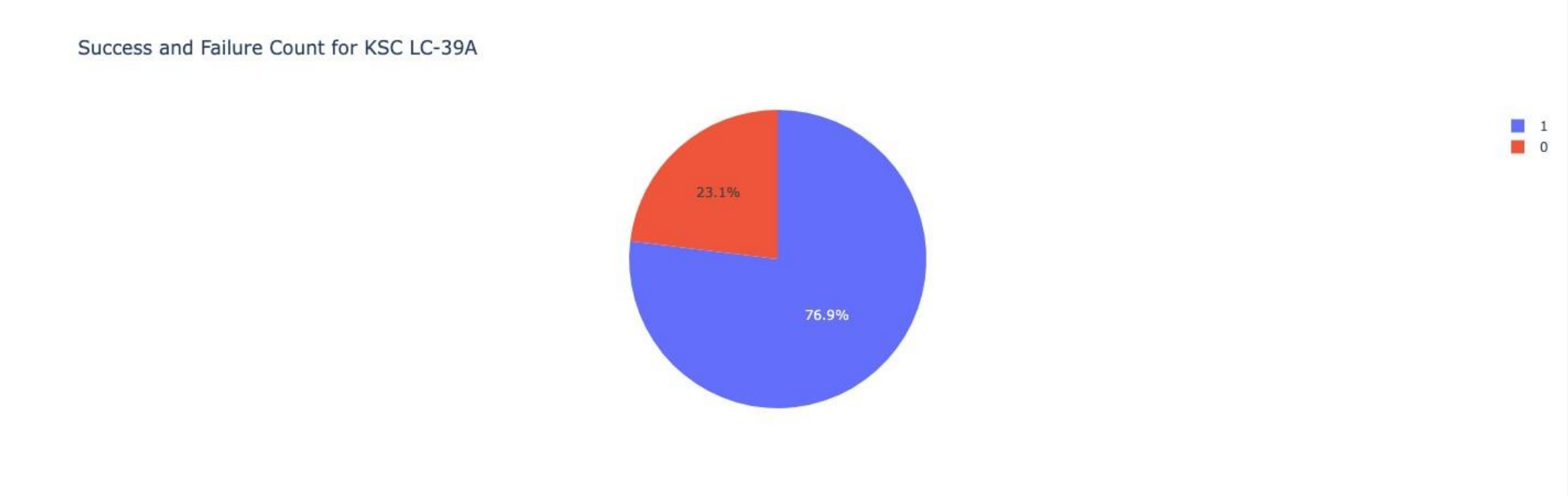

Section 4
Build a Dashboard with Plotly Dash

Success Rate by Launch Site



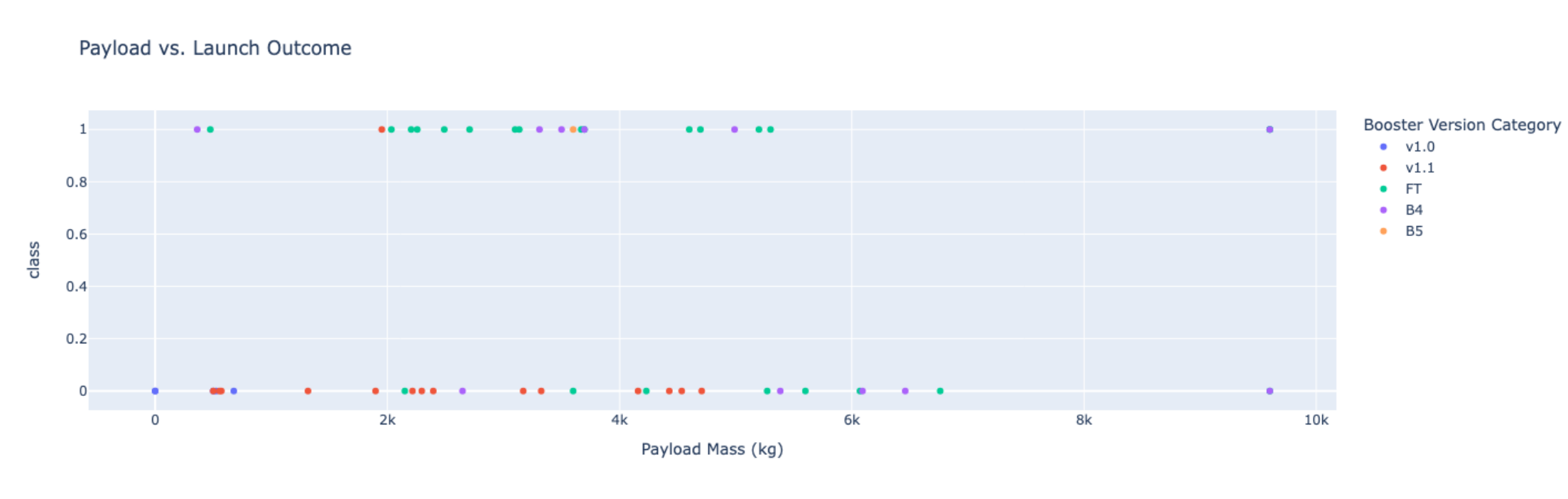
KSC LC-39A has the highest rate of success

Success and Failure Count for KSC LC-39A



KSC LC-39A has the highest success rate compared to other launch sites.

Payload vs Launch Outcome

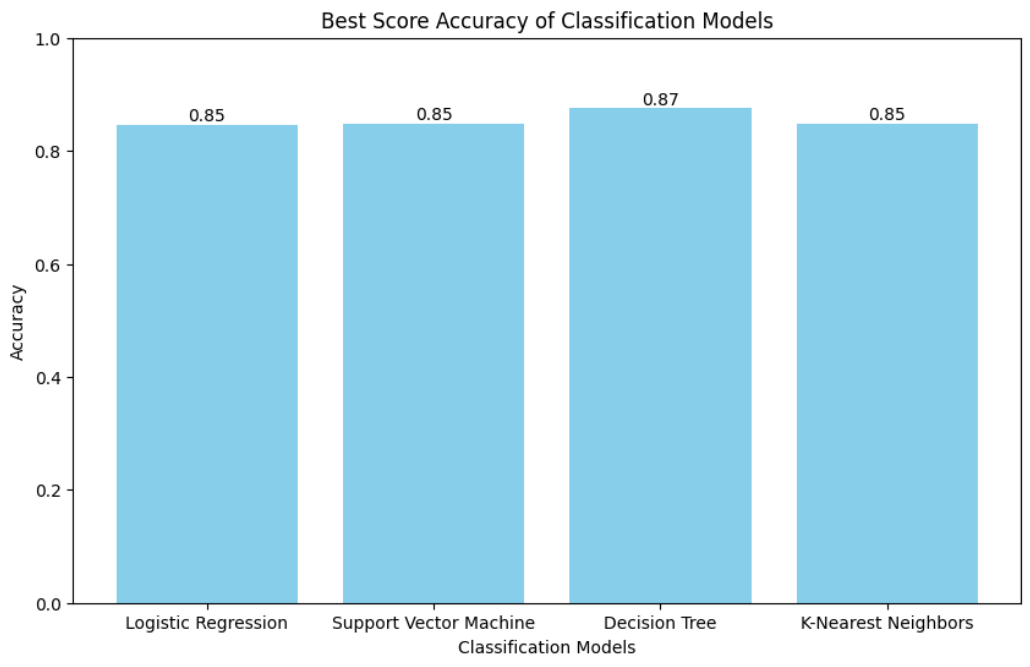


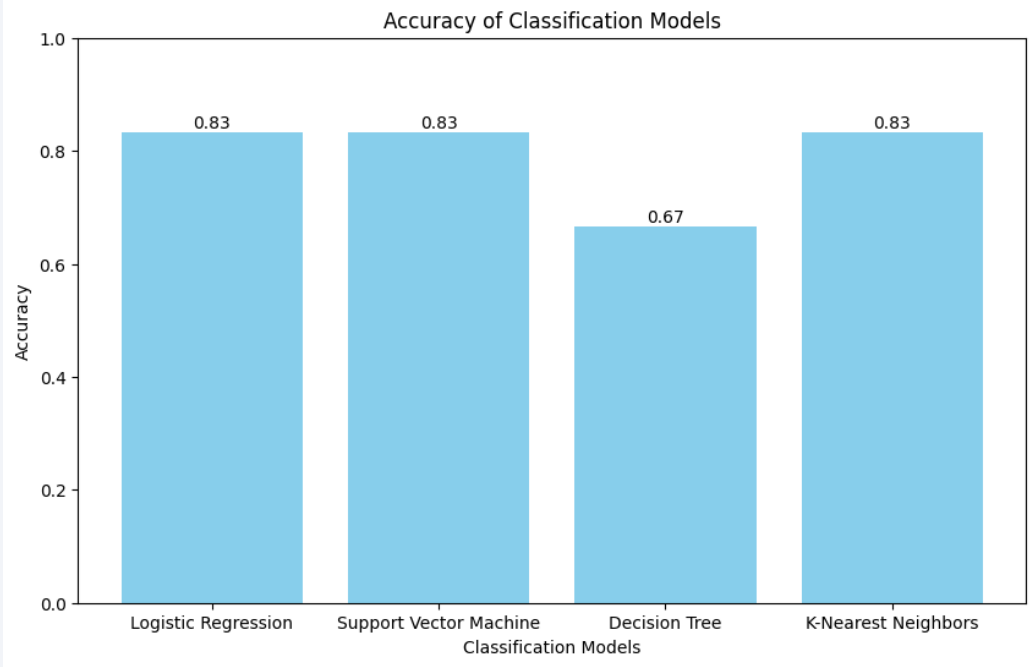
Booster version FT has the largest success rate with payload less than 6000 kg.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

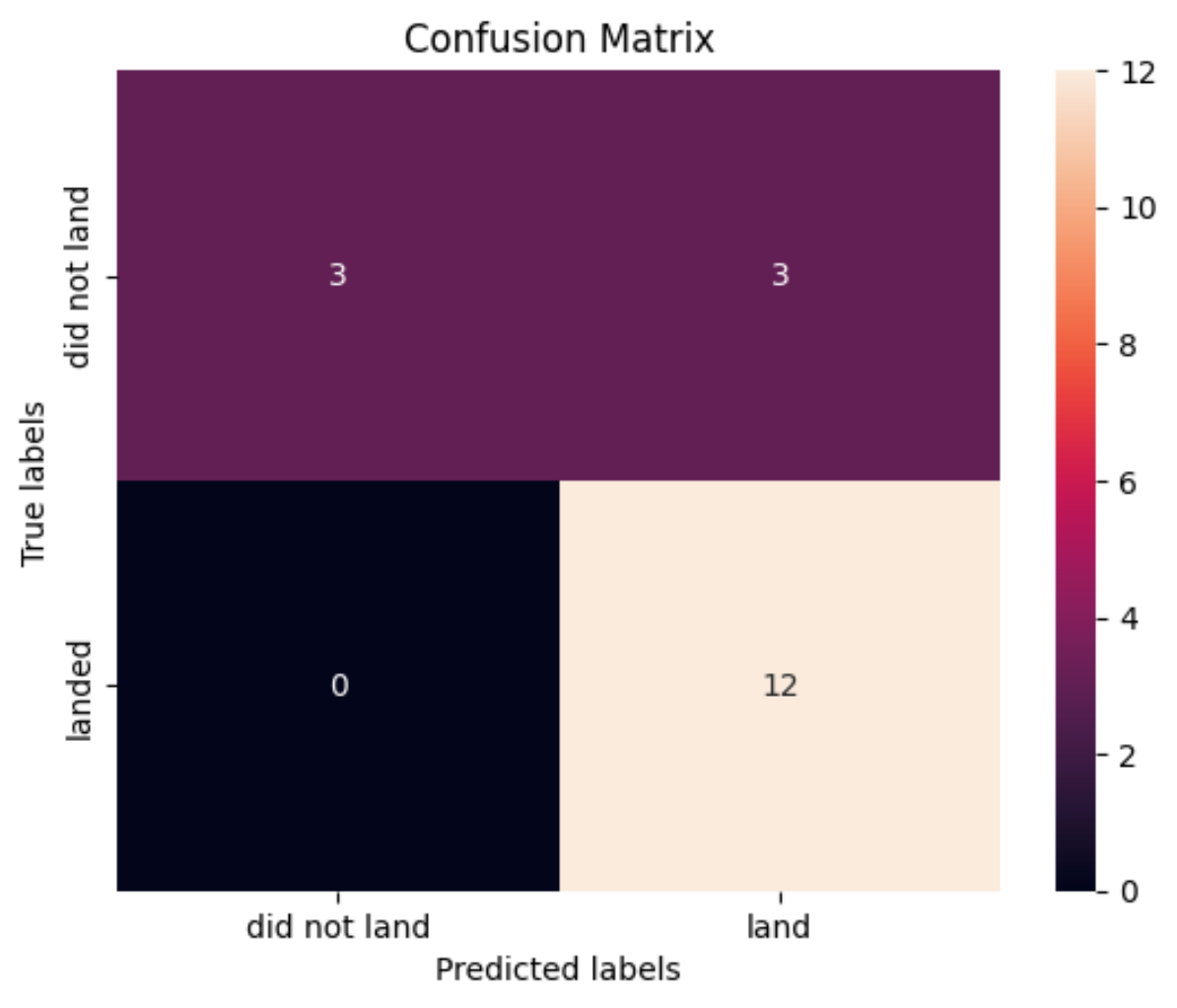




On the left, Decision Tree has shown the greatest best score accuracy of 87% but when it is used to predict with test data (on the right), the accuracy is only 67%.

The other classification models (logistic regression, support vector machine and KNN) are consistently producing high accuracy for both diagrams.

Confusion Matrix



- Although all models accurately predict true positive result, however it has some problems in predicting true negative results.

Conclusions

- Launch Site KSC LC-39A has the highest rate of success
- Booster version FT has the highest success rate with payload less than 6000 kg
- Orbit ES-L1, GEO, HEO and SSO have the highest success rate of 100%
- Logistic regression, SVM and KNN are the classification models that can best predict the landing outcomes

Appendix

```
# Create a marker with distance to a closest city, railway, highway, etc.
# Draw a line between the marker to the launch site

launch_site_lat = 29.55942
launch_site_lon = -95.08291
houston_lat = 29.75246
houston_lon = -95.3627
railway_lat = 29.53799
railway_lon = -95.11817
highway_lat = 29.55438
highway_lon = -95.08015

distance_houston = calculate_distance(launch_site_lat, launch_site_lon, houston_lat, houston_lon)
distance_railway = calculate_distance(launch_site_lat, launch_site_lon, railway_lat, railway_lon)
distance_highway = calculate_distance(launch_site_lat, launch_site_lon, highway_lat, highway_lon)

print("Distance between launch site and Houston city:",distance_houston)
print("Distance between launch site and nearest railway:",distance_railway)
print("Distance between launch site and nearest highway:",distance_highway)
```

```
Distance between launch site and Houston city: 34.53177054495322
Distance between launch site and nearest railway: 4.16204346816655
Distance between launch site and nearest highway: 0.6209533200448937
```

Code snippet of distance between launch site to nearest city, railway and highway

Appendix

```
# List of model names
model_names = ["Logistic Regression", "Support Vector Machine", "Decision Tree", "K-Nearest Neighbors"]

# List of model accuracy scores
model_scores = [lr_score, svm_score, tree_score, knn_score]

# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(model_names, model_scores, color='skyblue')
plt.xlabel('Classification Models')
plt.ylabel('Accuracy')
plt.title('Accuracy of Classification Models')
plt.ylim(0, 1)  # Set the y-axis limit from 0 to 1 (assuming accuracy is in the range [0, 1])

# Display the accuracy scores on top of the bars
for i, score in enumerate(model_scores):
    plt.text(i, score, f'{score:.2f}', ha='center', va='bottom')

plt.show()
```

Code snippet of create a bar chart to visualize accuracy of classification models against test data

Appendix

```
# List of model names
model_names = ["Logistic Regression", "Support Vector Machine", "Decision Tree", "K-Nearest Neighbors"]

# List of model accuracy scores
model_bestscores = [lr_bestscore, svm_bestscore, tree_bestscore, knn_bestscore]

# Create a bar chart
plt.figure(figsize=(10, 6))
plt.bar(model_names, model_bestscores, color='skyblue')
plt.xlabel('Classification Models')
plt.ylabel('Accuracy')
plt.title('Best Score Accuracy of Classification Models')
plt.ylim(0, 1)  # Set the y-axis limit from 0 to 1 (assuming accuracy is in the range [0, 1])

# Display the accuracy scores on top of the bars
for i, score in enumerate(model_bestscores):
    plt.text(i, score, f'{score:.2f}', ha='center', va='bottom')

plt.show()
```

Code snippet of create a bar chart to visualize best score of classification models after hyperparameter tuning

Thank you!