IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Viktor Shcherbak
12/08/2024

Outline

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

Executive Summary

- **Methodology**

1. **Data Collection**
    1. Data was gathered from previous SpaceX launches, which included various parameters such as payload mass, orbit type, launch site, weather conditions, and flight telemetry.
    2. The target variable was **"First Stage Landing Outcome"** (1: Successful landing, 0: Failure).
2. **Data Preprocessing**
    1. Missing values were handled, and unnecessary features were dropped.
    2. Categorical variables (e.g., launch site, orbit) were converted into numerical features using **one-hot encoding**.
    3. Data was normalized or scaled to ensure uniform feature importance.
    4. The dataset was split into **training** (80%) and **testing** (20%) sets to evaluate model performance.
3. **Model Development**
   A machine learning pipeline was created using multiple classification models:
    1. **Support Vector Machine (SVM)**
    2. **Decision Tree**
    3. **K-Nearest Neighbors (KNN)**
    4. **Logistic Regression** (optional, as a baseline model)
4. Each model was trained on the training set and evaluated using the testing set.
5. **Performance Evaluation**
    1. **Accuracy** was used as the primary performance metric.
    2. A **confusion matrix** was generated for the best-performing model to analyze true positives, false positives, true negatives, and false negatives.
    3. Results were visualized using bar charts and confusion matrix plots.

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. In this lab, you will collect and make sure the data is in the correct format from an API. The following is an example of a successful and launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

```
spacex_url=https://api.spacexdata.com/v4/launches/past
response = requests.get(spacex_url)
data = response.json()
soup = BeautifulSoup(response.text, 'html.parser')
```

- Perform data wrangling
  - We mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- We make a get request to the SpaceX API. You will also do some basic data wrangling and formating.
- Request to the SpaceX API
- Clean the requested data

Web scrap Falcon 9 launch records with BeautifulSoup:

- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

```
spacex_url=https://api.spacexdata.com/v4/launches/past
response = requests.get(spacex_url)
data = response.json()
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

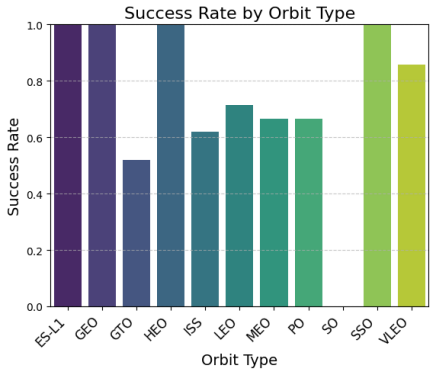
```
static_url = https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

response = requests.get(static_url)

soup = BeautifulSoup(response.text, 'html.parser’)

html_tables = soup.find_all('table')
```

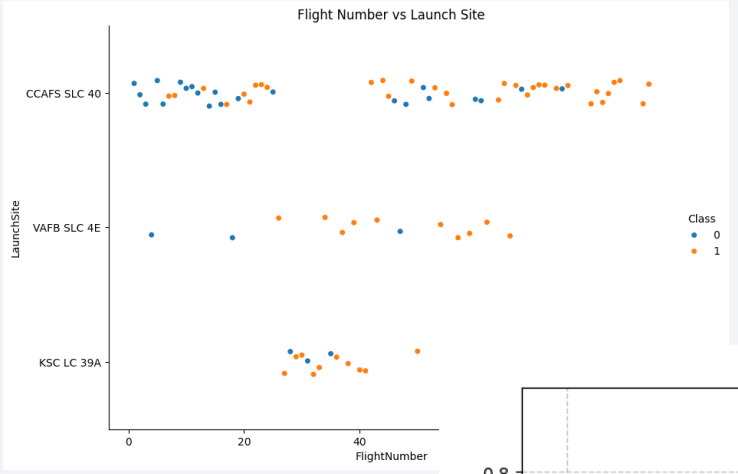
Data Wrangling

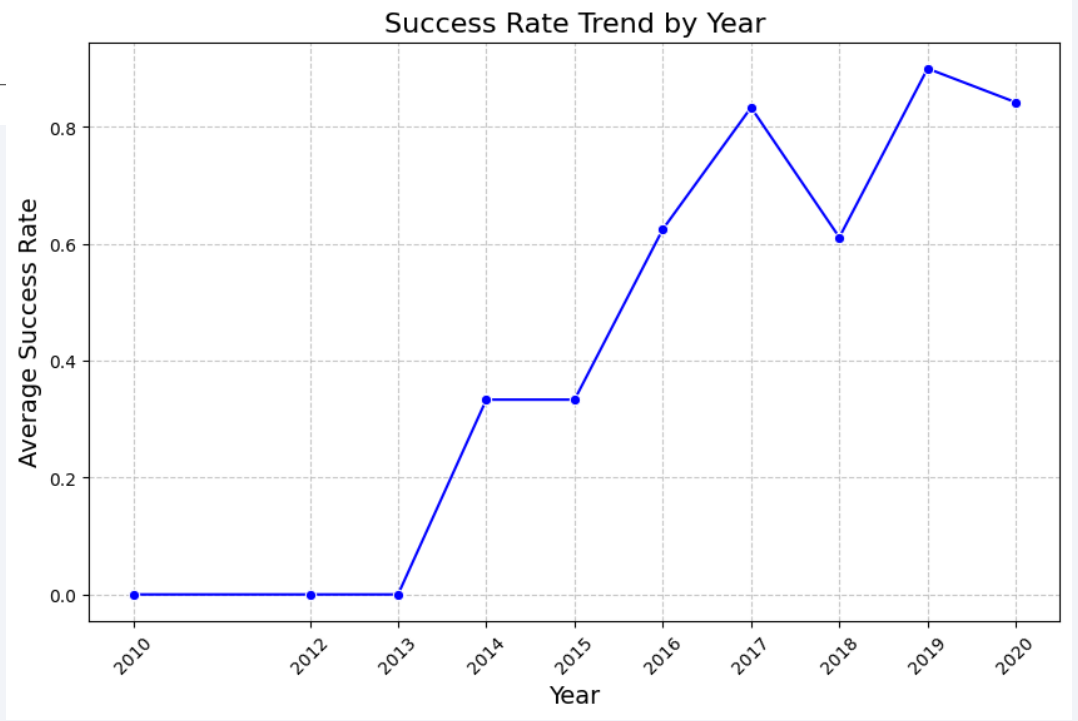
In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

In this lab we will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful

EDA with Data Visualization



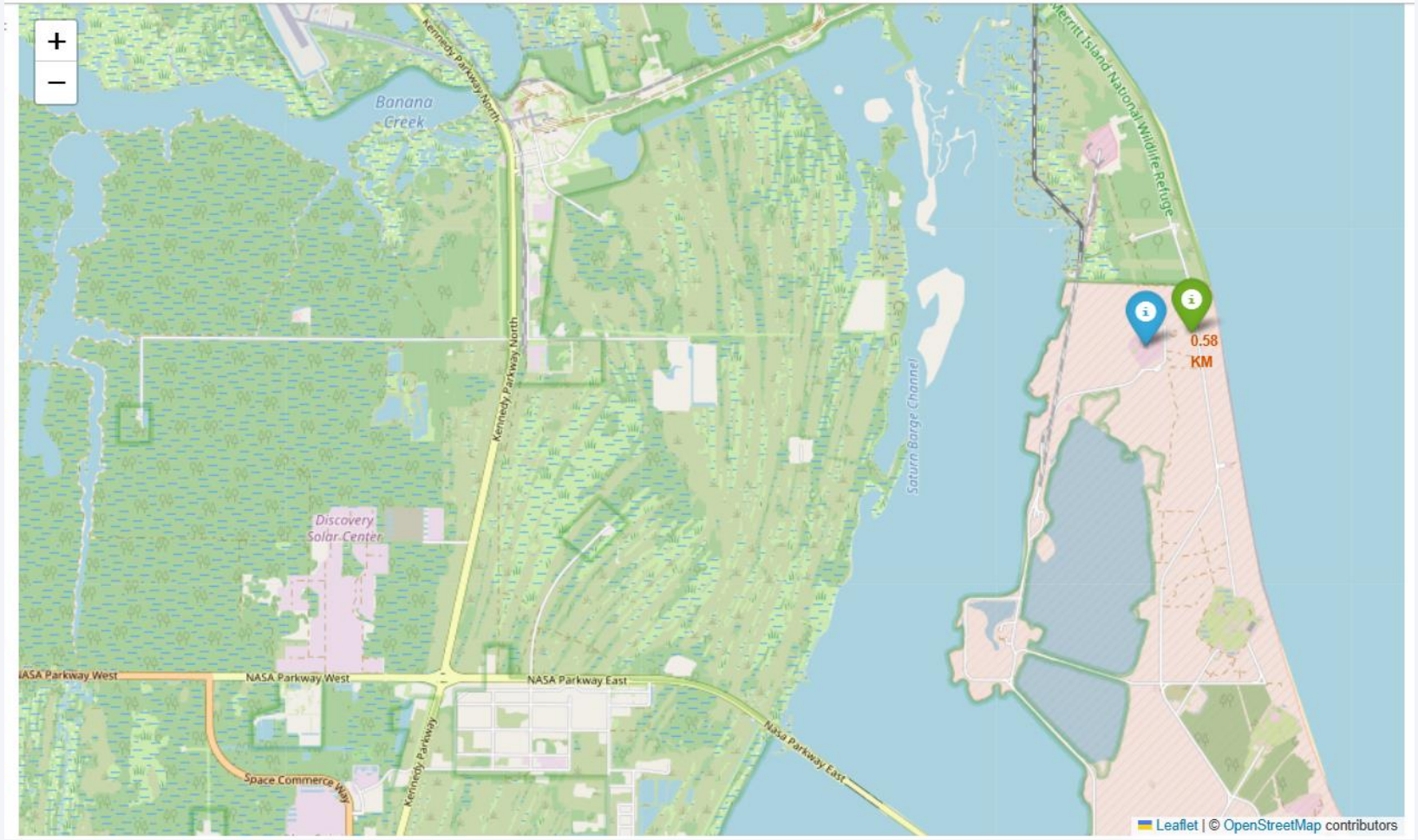




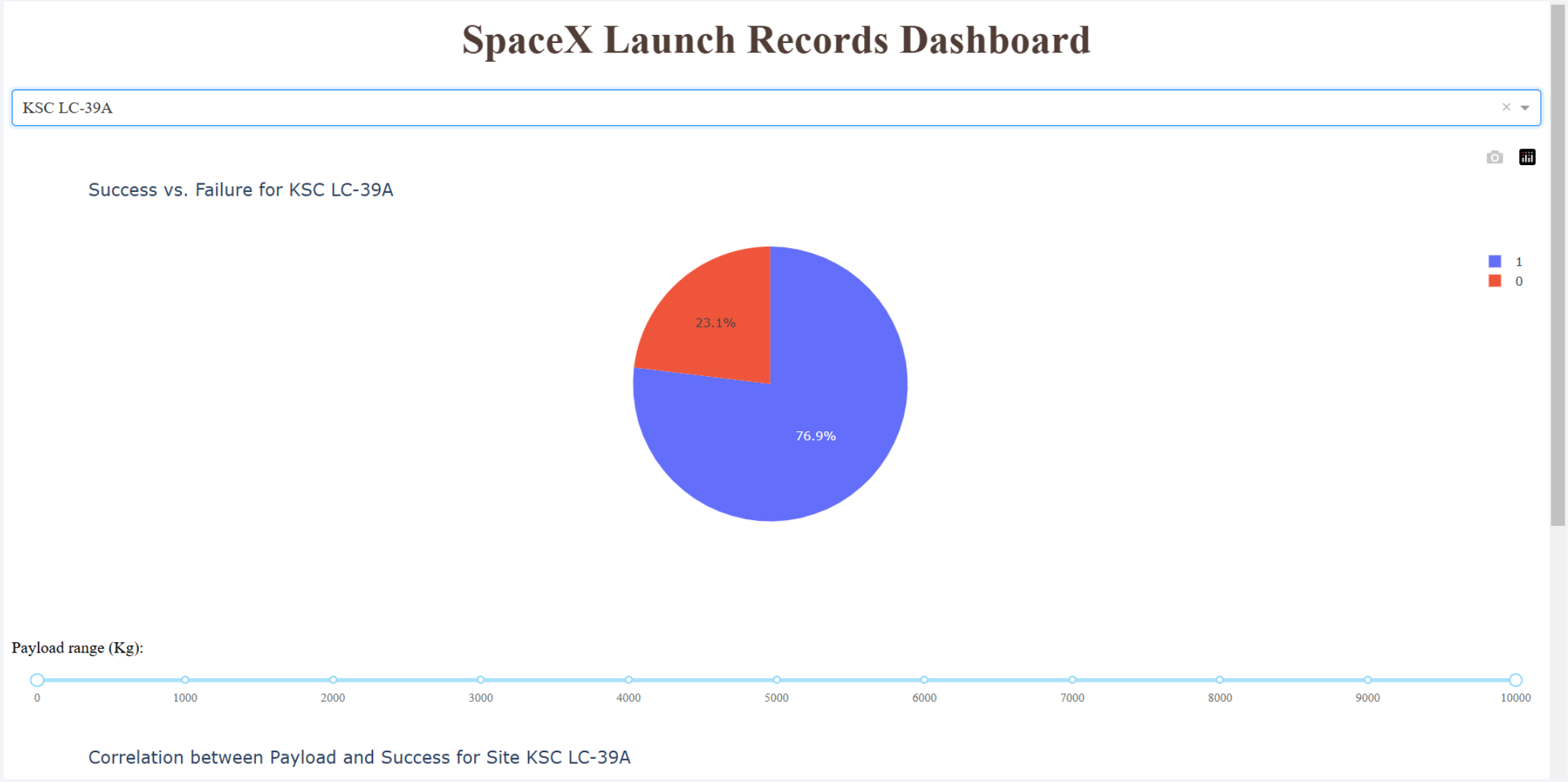
EDA with SQL

cid	name	type	notnull	dflt_value	pk
0	Date	TEXT	0	None	0
1	Time (UTC)	TEXT	0	None	0
2	Booster_Version	TEXT	0	None	0
3	Launch_Site	TEXT	0	None	0
4	Payload	TEXT	0	None	0
5	PAYLOAD_MASS__KG_	INT	0	None	0
6	Orbit	TEXT	0	None	0
7	Customer	TEXT	0	None	0
8	Mission_Outcome	TEXT	0	None	0
9	Landing_Outcome	TEXT	0	None	0

Build an Interactive Map with Folium



Build a Dashboard with Plotly Dash

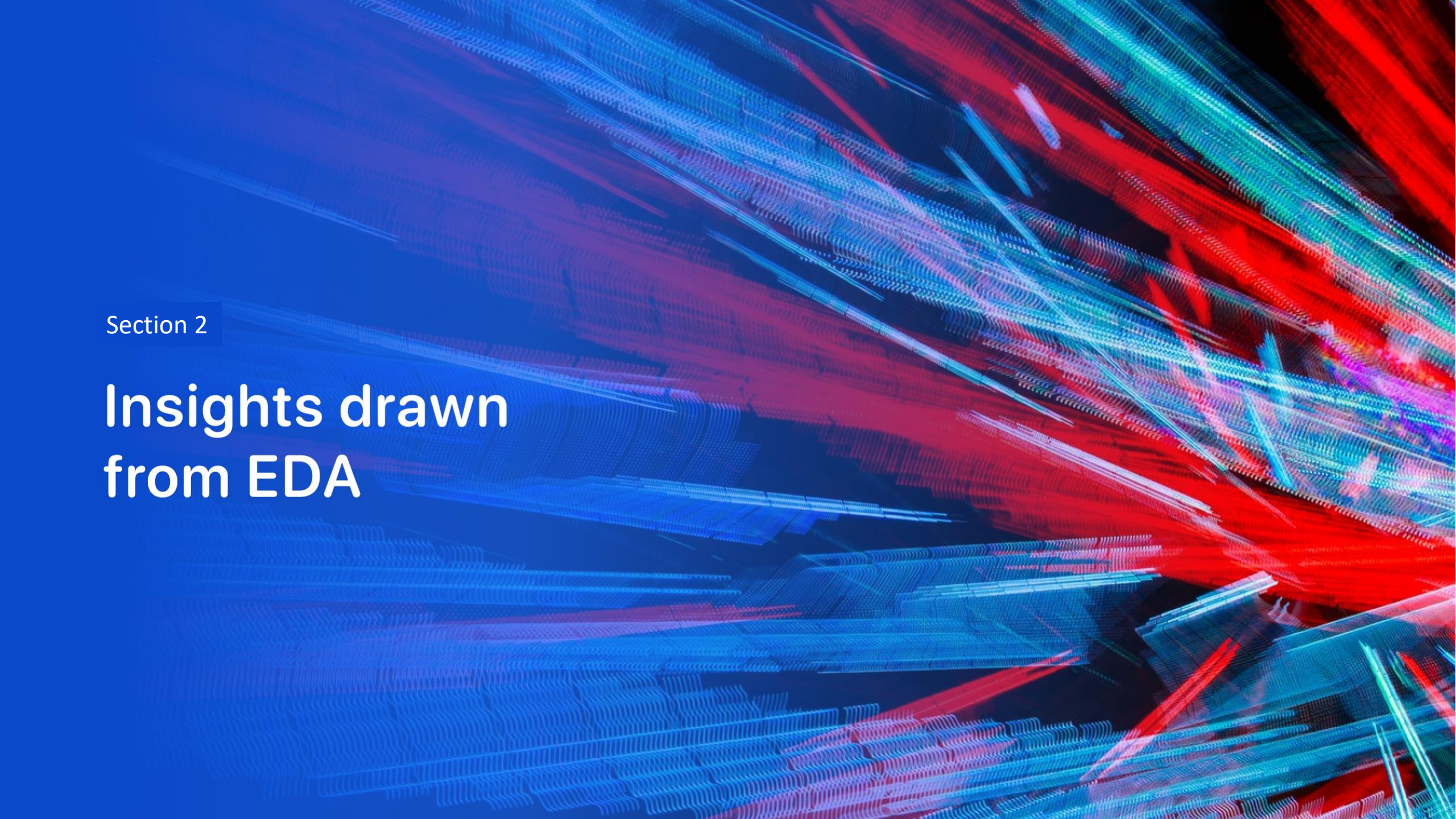


Predictive Analysis (Classification)

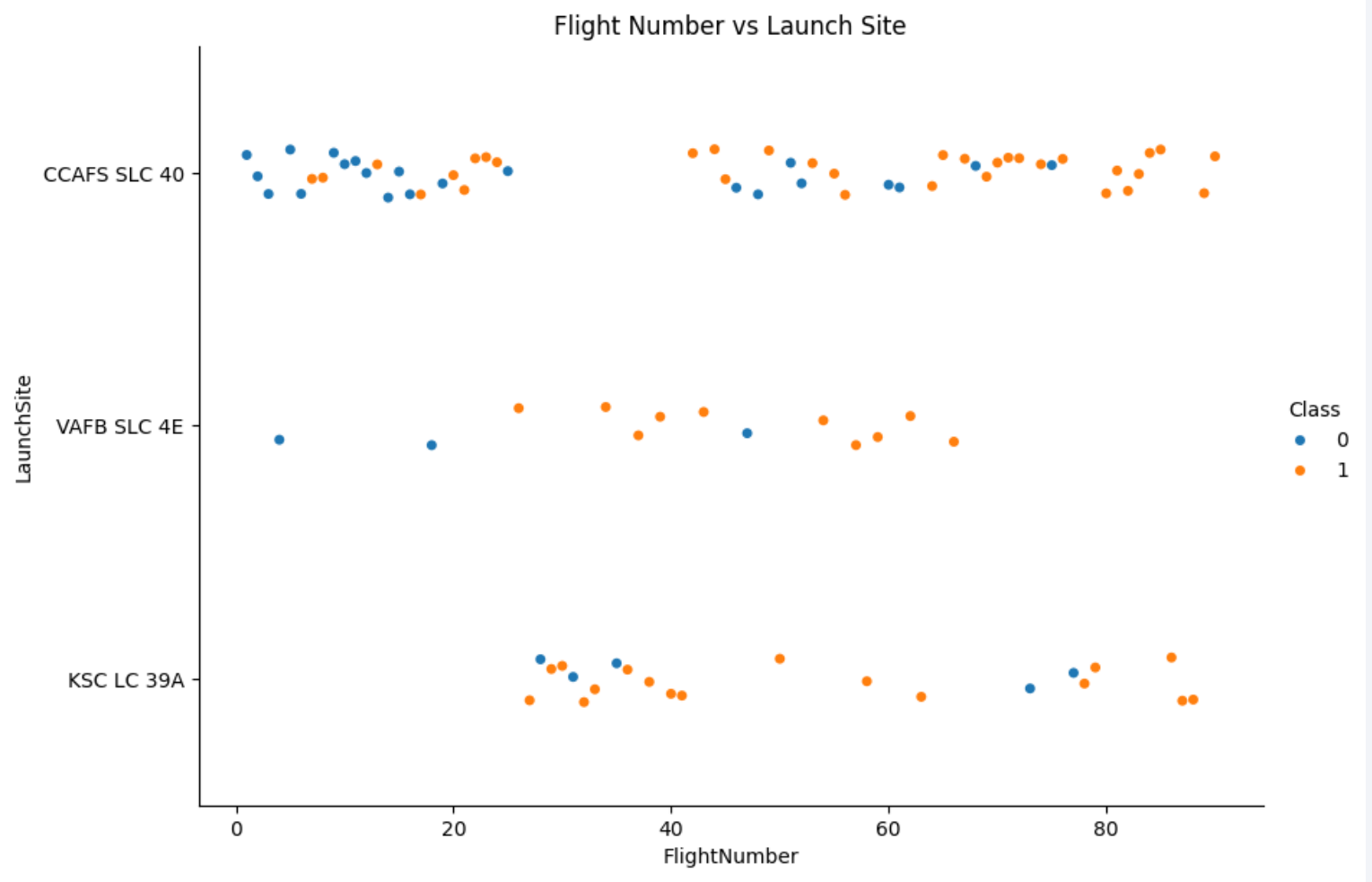
- Perform exploratory Data Analysis and determine Training Labels
- create a column for the class
- Standardize the data
- Split into training data and test data
- -Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Find the method performs best using test data

Results

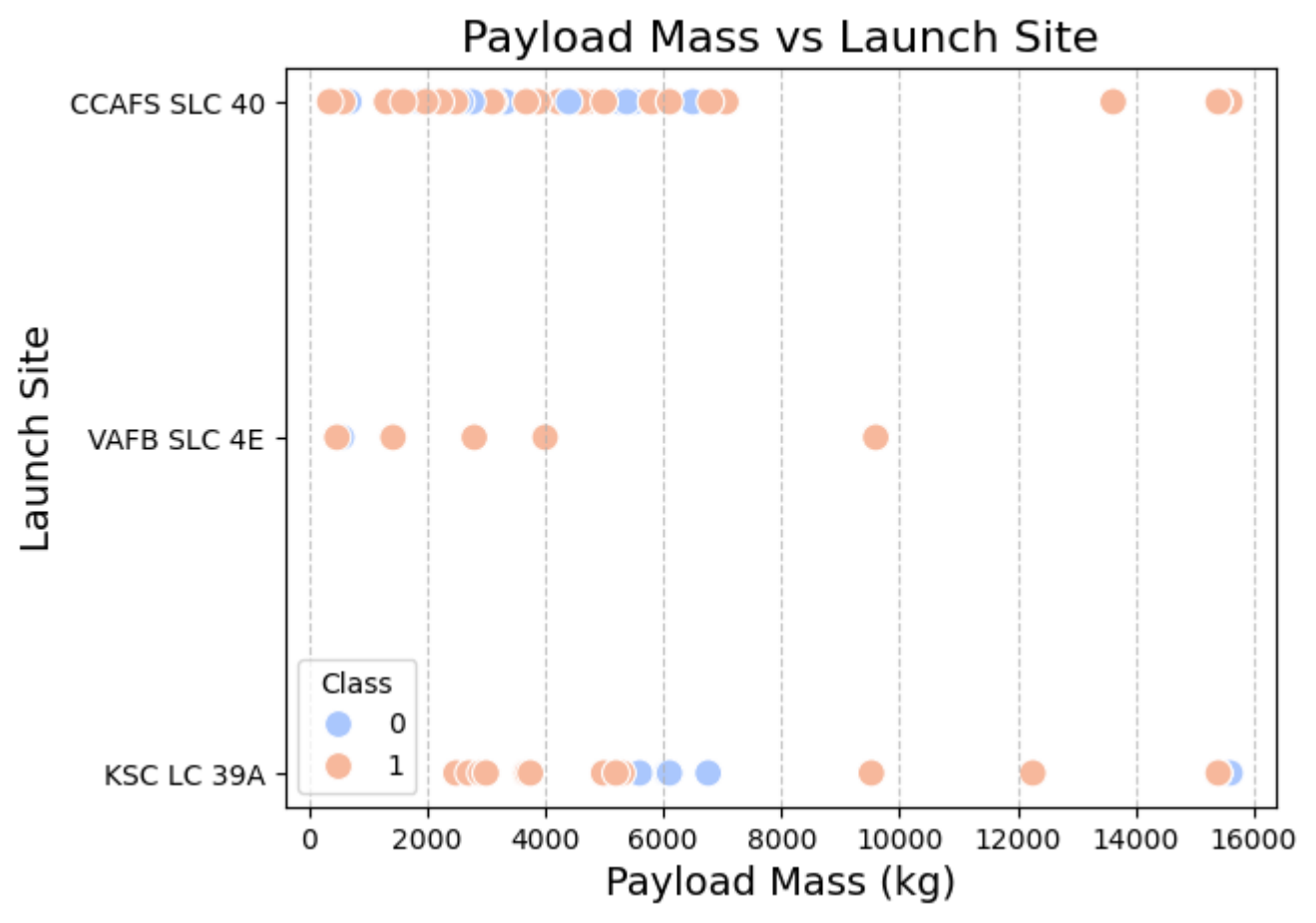
```
# Точності для кожної моделі (припустимо, вони вже обчислені)accuracy_svm = svm_cv.best_estimator_.score(X_test, Y_test)accuracy_tree = tree_cv.best_estimator_.score(X_test, Y_test)accuracy_knn = knn_cv.best_estimator_.score(X_test, Y_test)# Порівняння точностейaccuracies = {    "Support Vector Machine": accuracy_svm,    "Decision Tree": accuracy_tree, "K-Nearest Neighbors": accuracy_knn}# Визначення найкращого методуbest_method = max(accuracies, key=accuracies.get)print("Best method:", best_method)print("Accuracy:", accuracies[best_method])
```


Section 2
Insights drawn from EDA

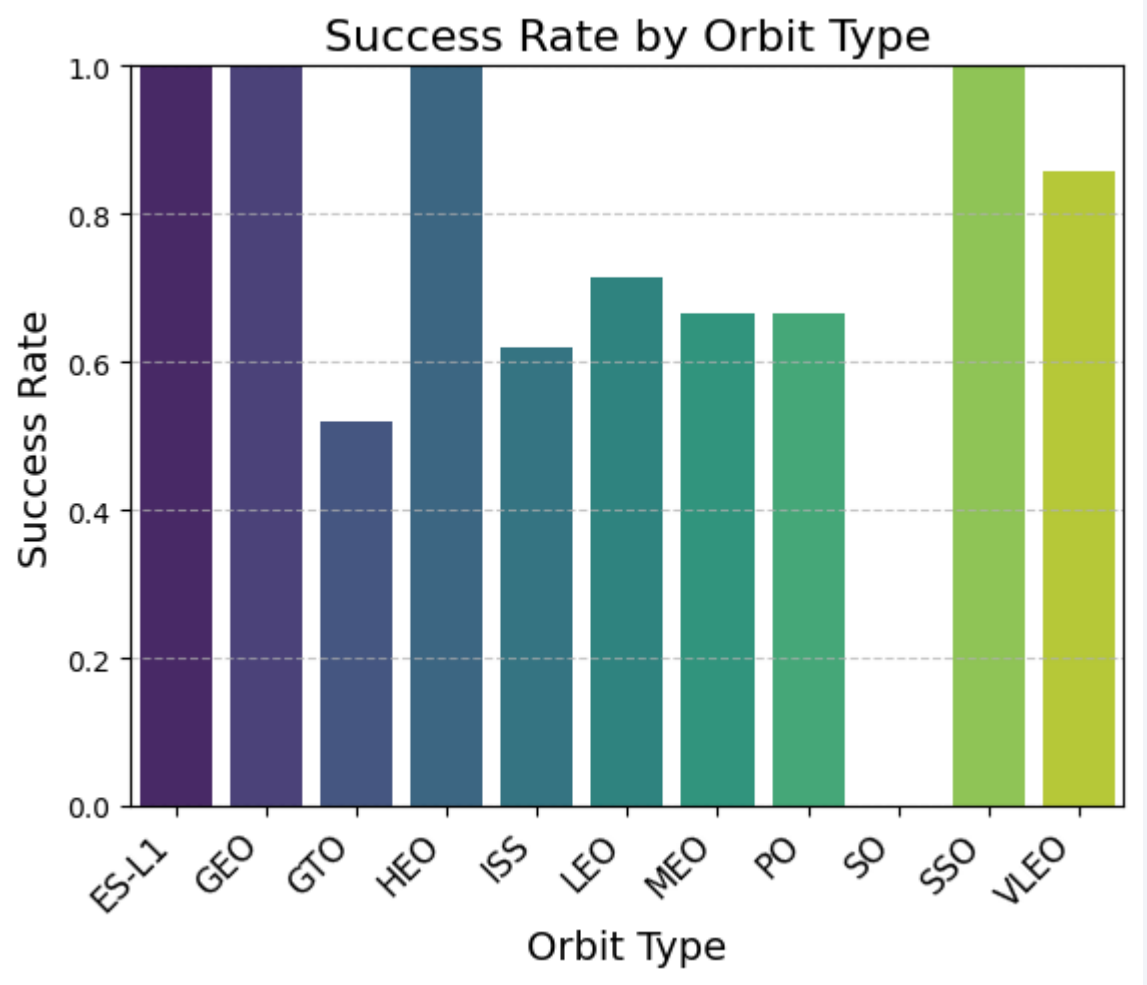
Flight Number vs. Launch Site



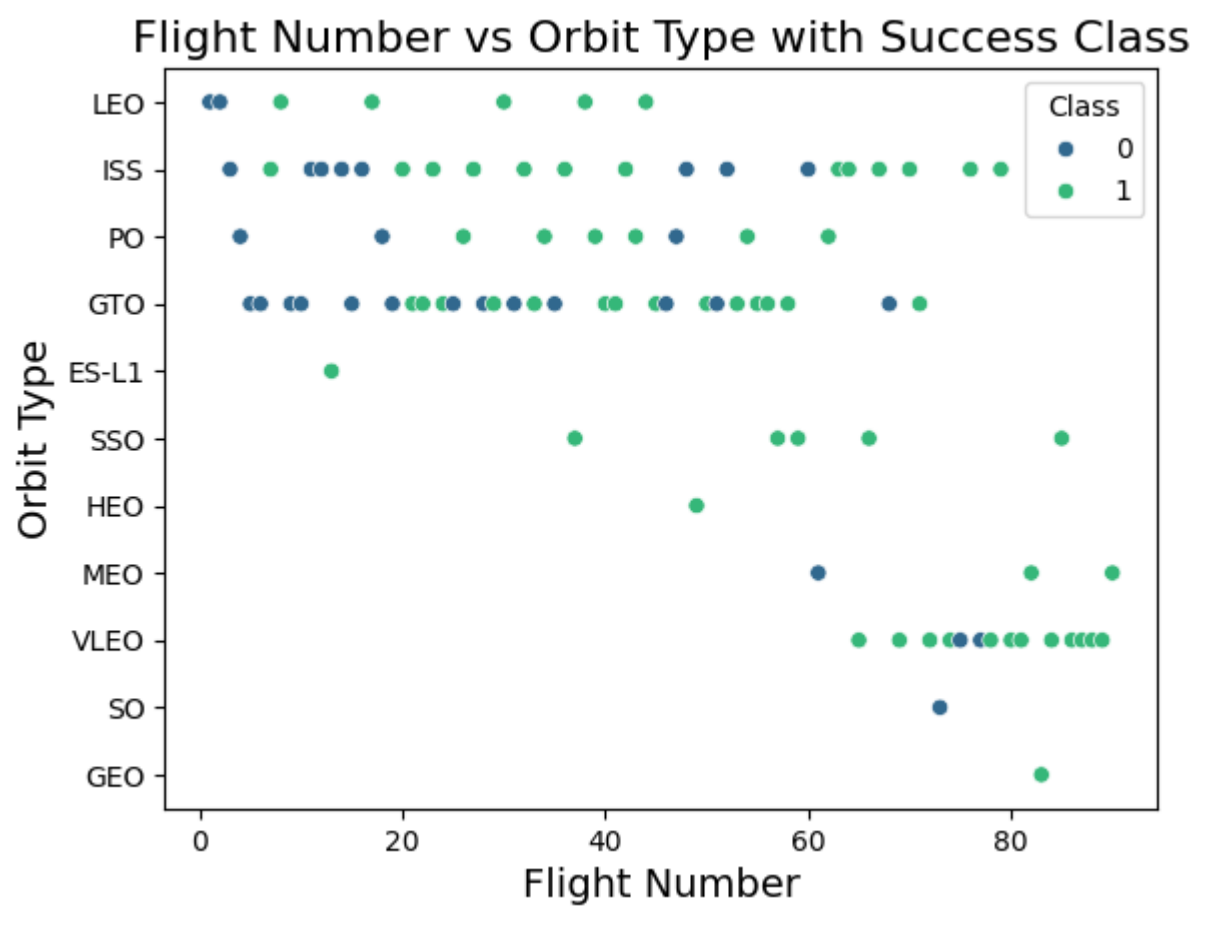
Payload vs. Launch Site



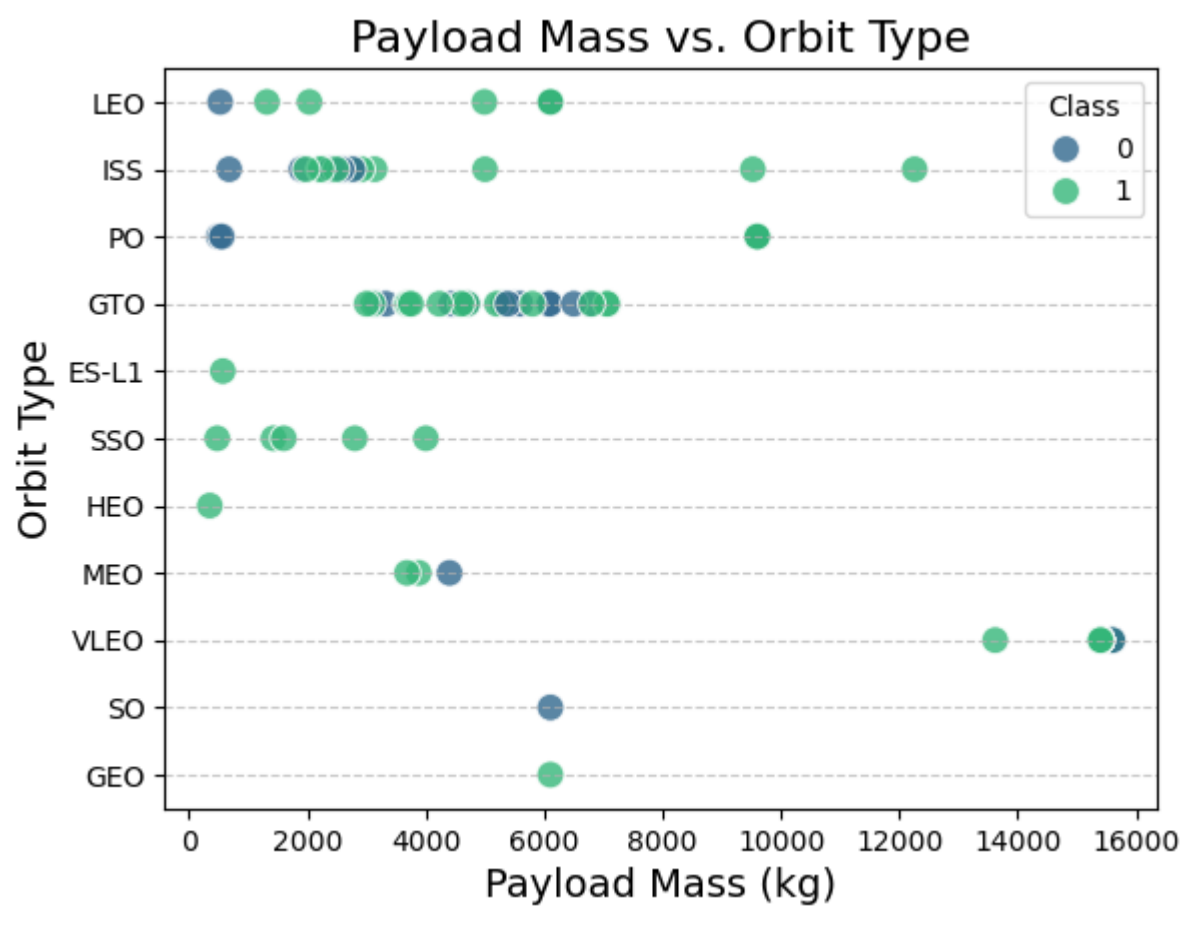
Success Rate vs. Orbit Type



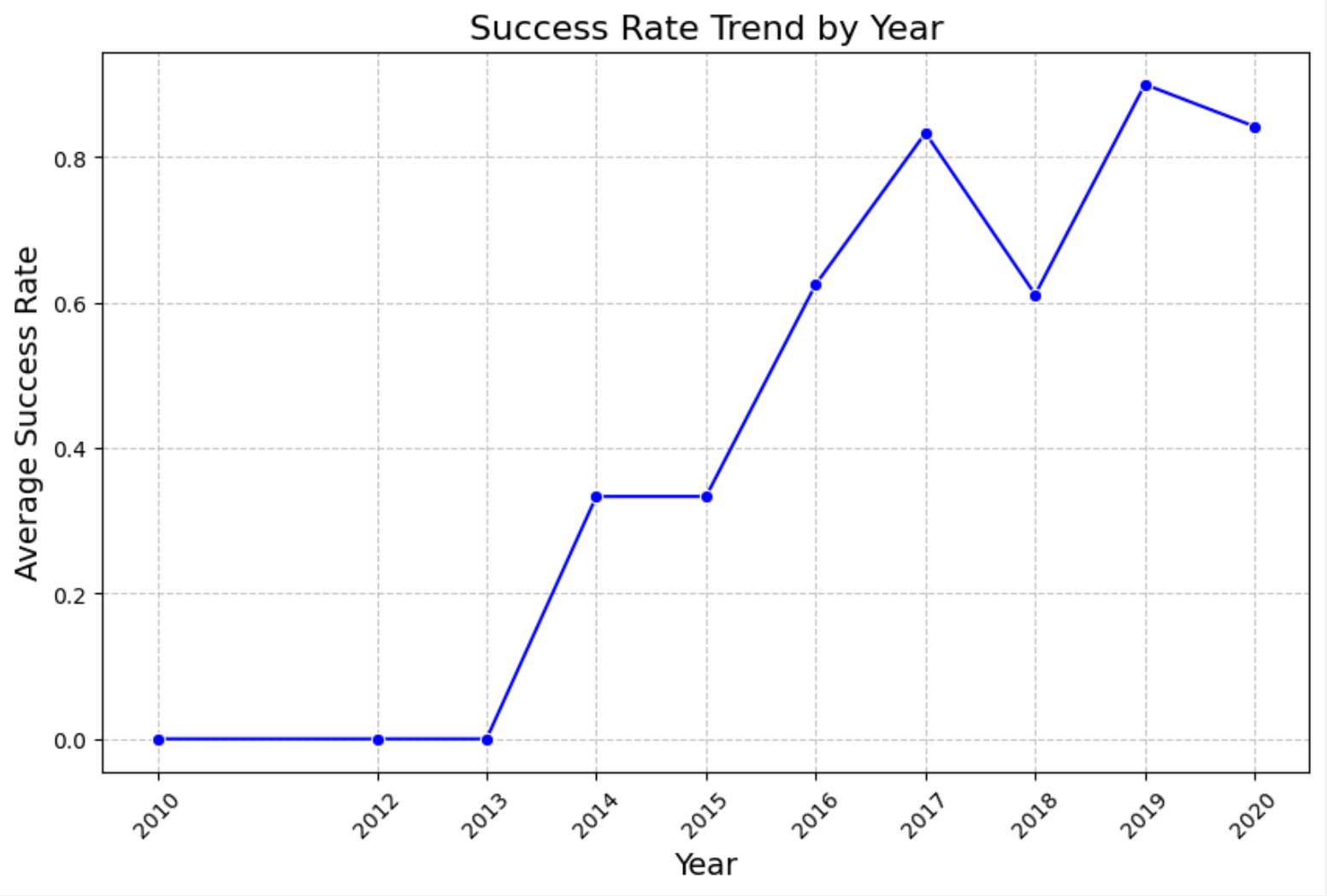
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

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

- %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

[13]:

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

- %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 30;

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

total_payload_mass
45596

- %sql SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload_mass FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

avg_payload_mass
2928.4

- %sql SELECT AVG("PAYLOAD_MASS__KG_") AS avg_payload_mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

first_successful_landing_date
2010-06-04

- %sql SELECT MIN("Date") AS first_successful_landing_date FROM SPACEXTABLE WHERE "Mission_Outcome" = 'Success';

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Booster_Version
None

- %sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Mission_Outcome" = 'Success' AND "Landing_Outcome" LIKE '%Controlled (ocean)%' AND "PAYLOAD_MASS_KG_" > 4000 AND "PAYLOAD_MASS_KG_" < 6000;

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

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

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

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

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

- %sql SELECT substr("Date", 6, 2) AS month, "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE '%Failure (drone ship)%' AND substr("Date", 1, 4) = '2015';

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

- 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

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
Failure (parachute)	2
Precluded (drone ship)	1

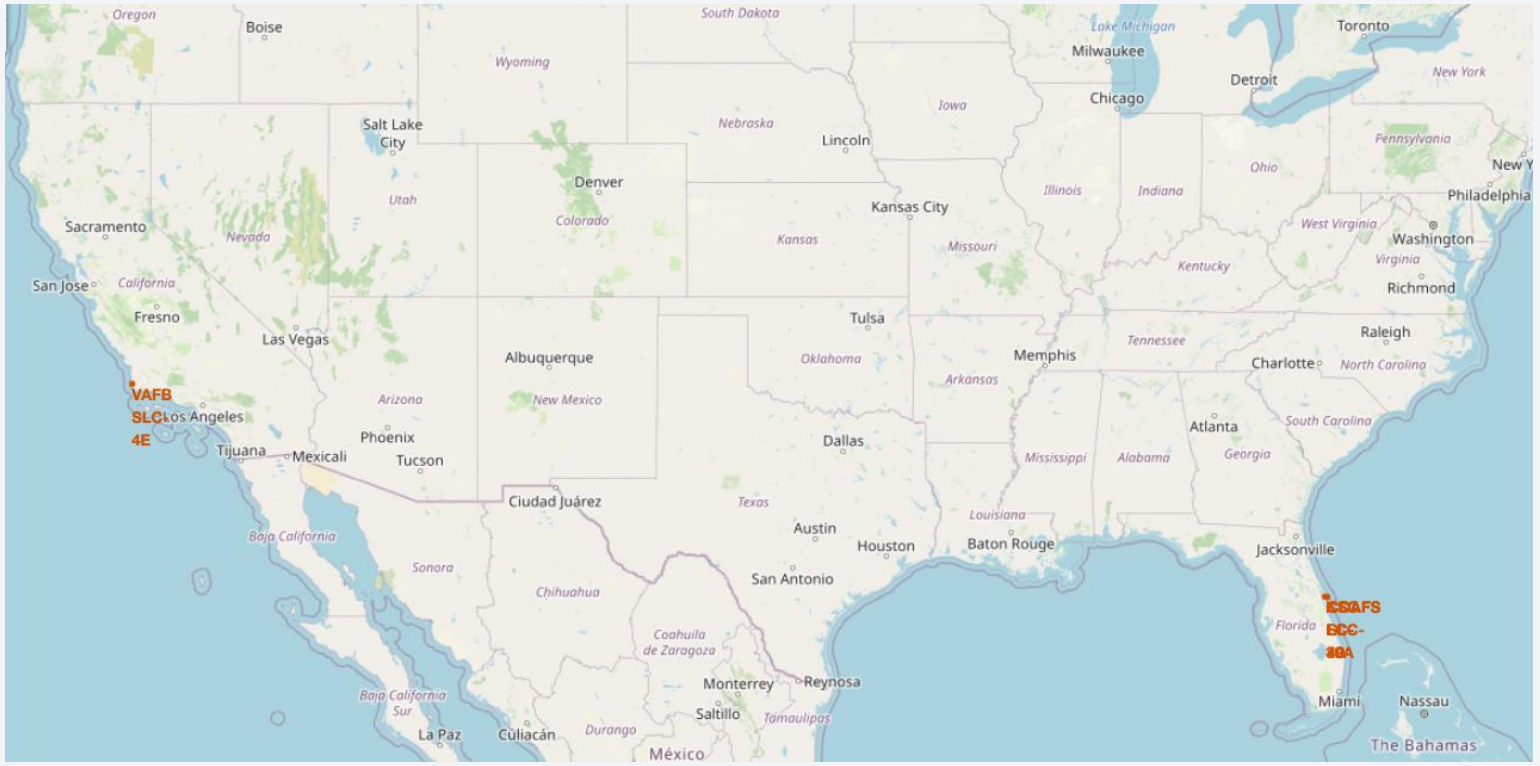
- %sql SELECT "Landing_Outcome", COUNT(*) AS outcome_count FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'GROUP BY "Landing_Outcome" ORDER BY outcome_count DESC;

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

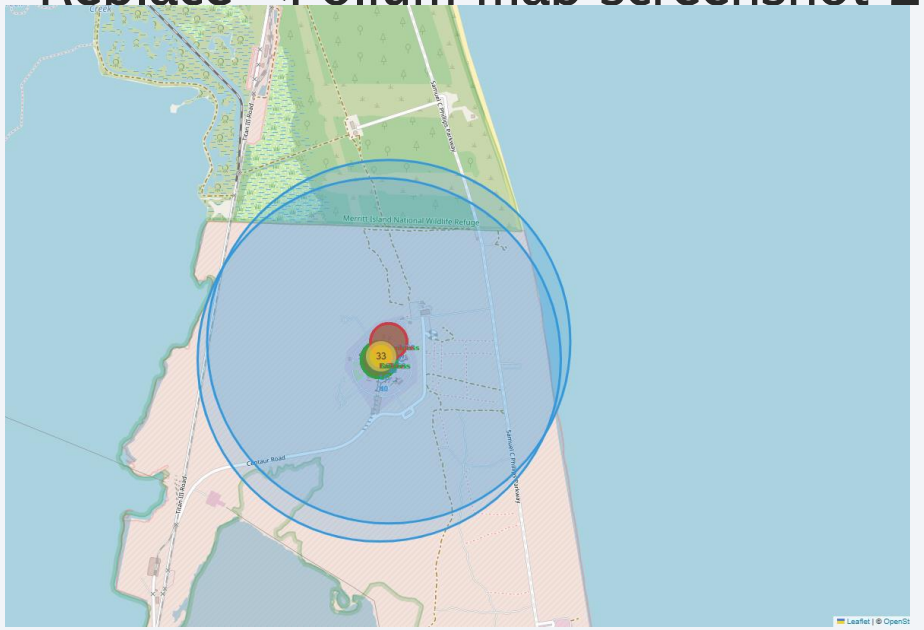
- Replace <Folium map screenshot 1> title with an appropriate title

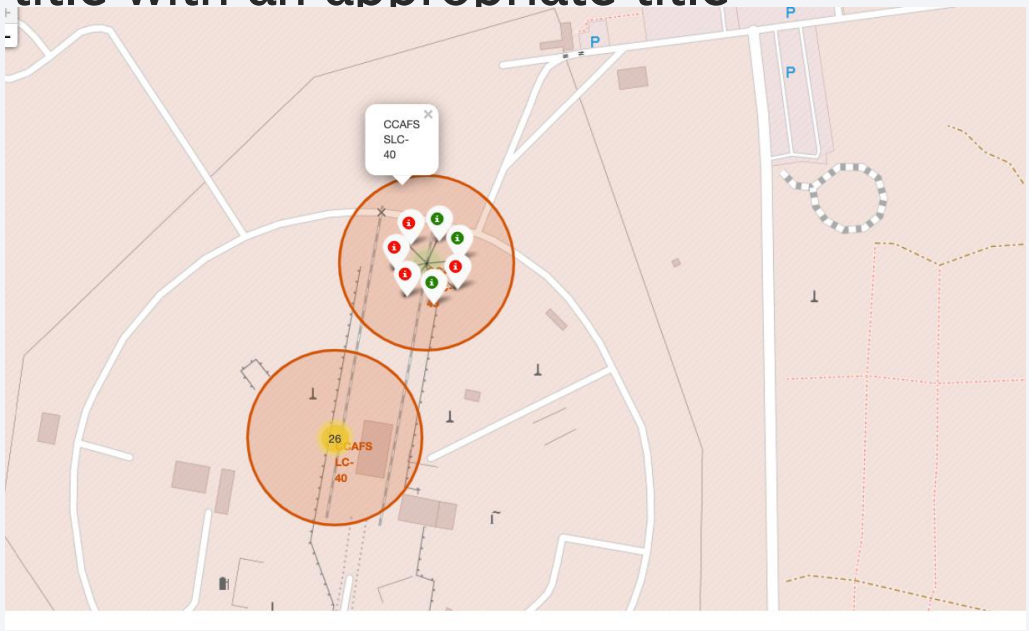


- Initial the map, For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a popup label, Create and add a circle for each site, Create and add a marker for each site, Add the circle and marker to the map, Display the map

<Folium Map Screenshot 2>

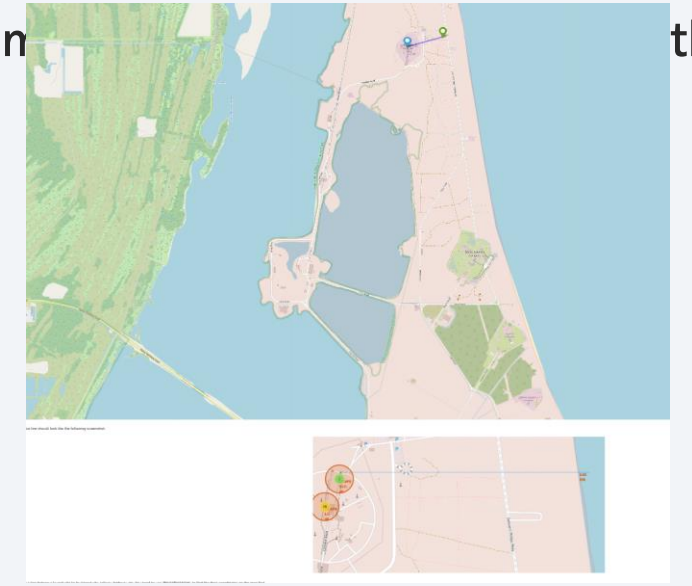
- Replace <Folium map screenshot 2> title with an appropriate title



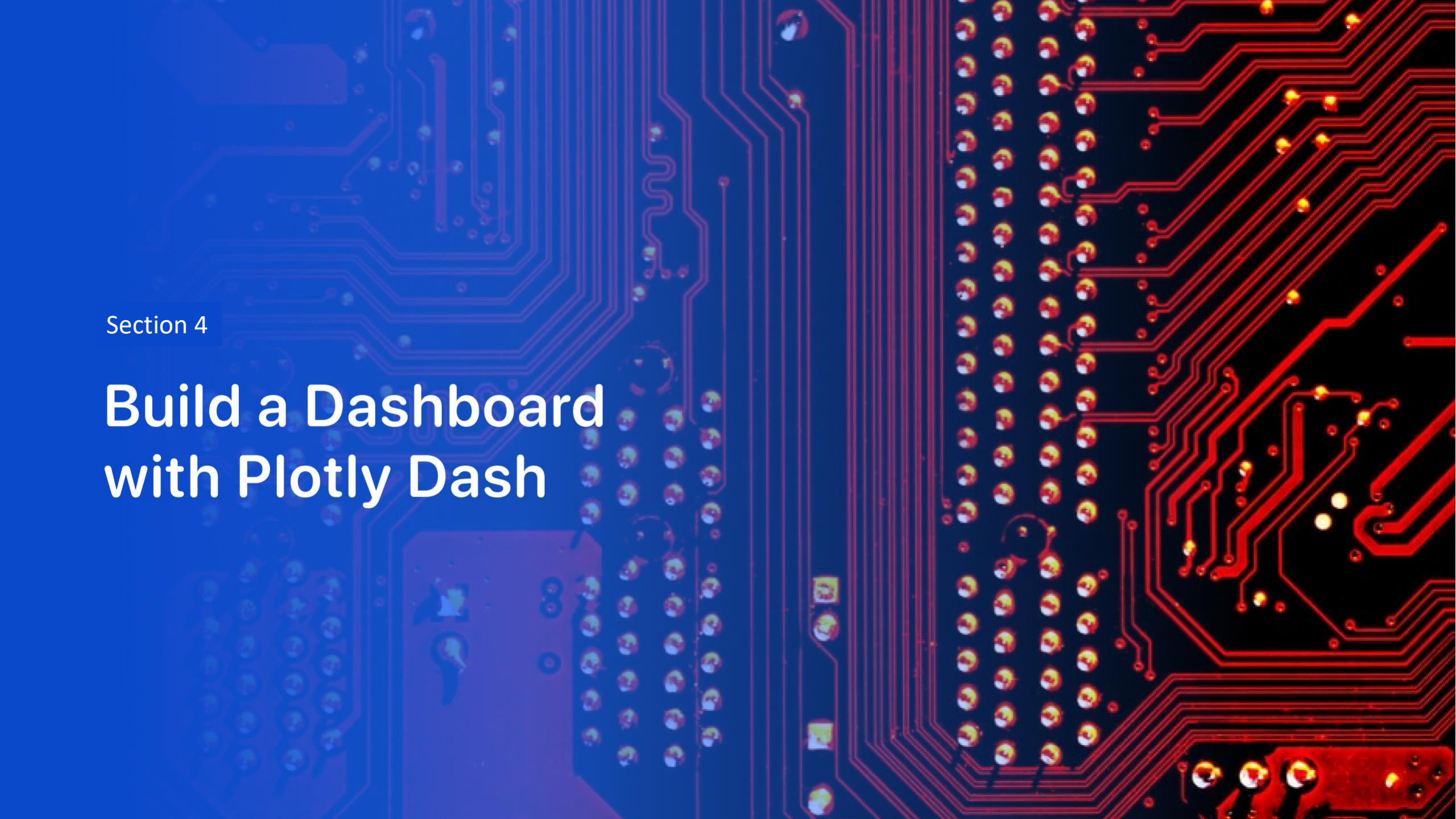


- Apply a function to check the value of `class` column, If class=1, marker_color value will be green, If class=0, marker_color value will be

<Folium Map Screenshot 3>

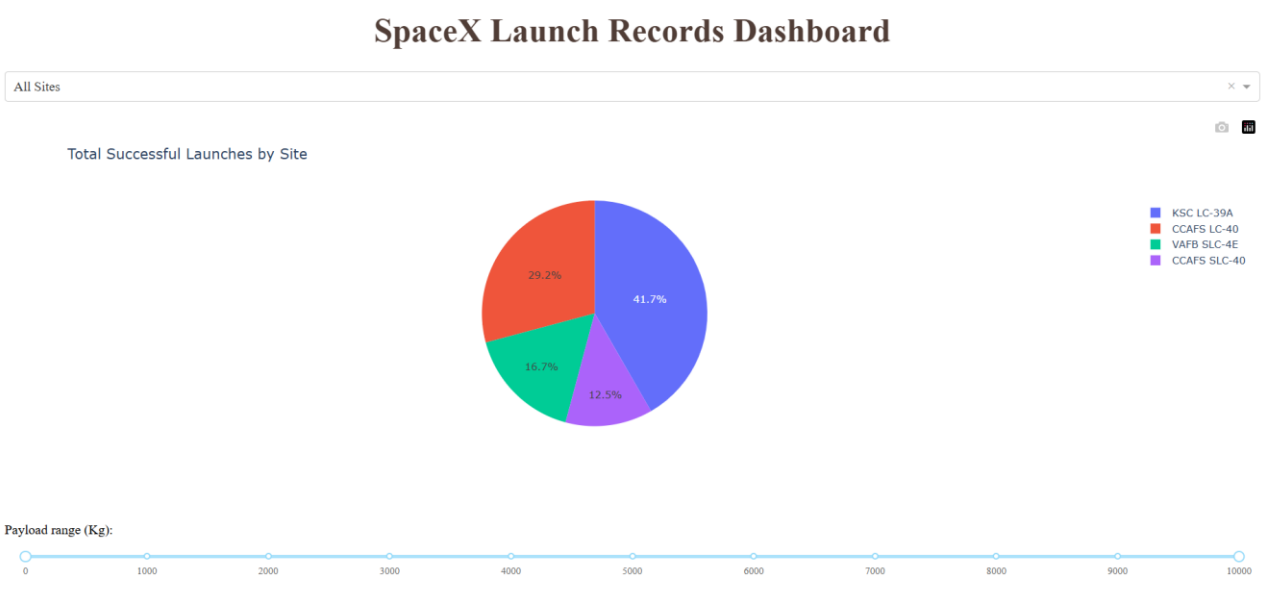
- Replace <Folium  th an appropriate title

- calculate distances, Add Mouse Position to get the coordinate (Lat, Long) for a mouse over on the map, Extract launch site latitude and longitude, Add a marker for the closest coastline point, Add a marker to display the distance


Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

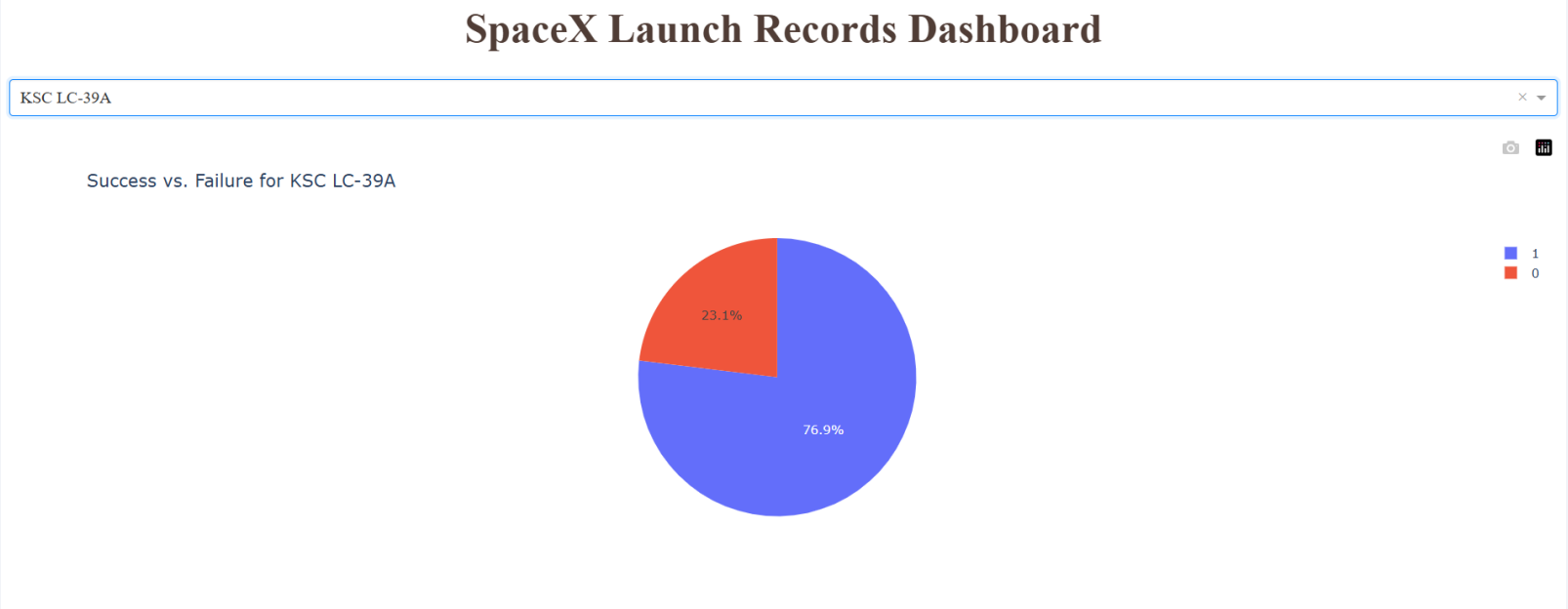
- Replace <Dashboard screenshot 1> title with an appropriate title



- Most succesfull lauches by KSC LC-39A, least succesfull lauches by CCAFS SL-40

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title



- Success 10 (76,9%) and failure 3 (23,1%) launch

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title

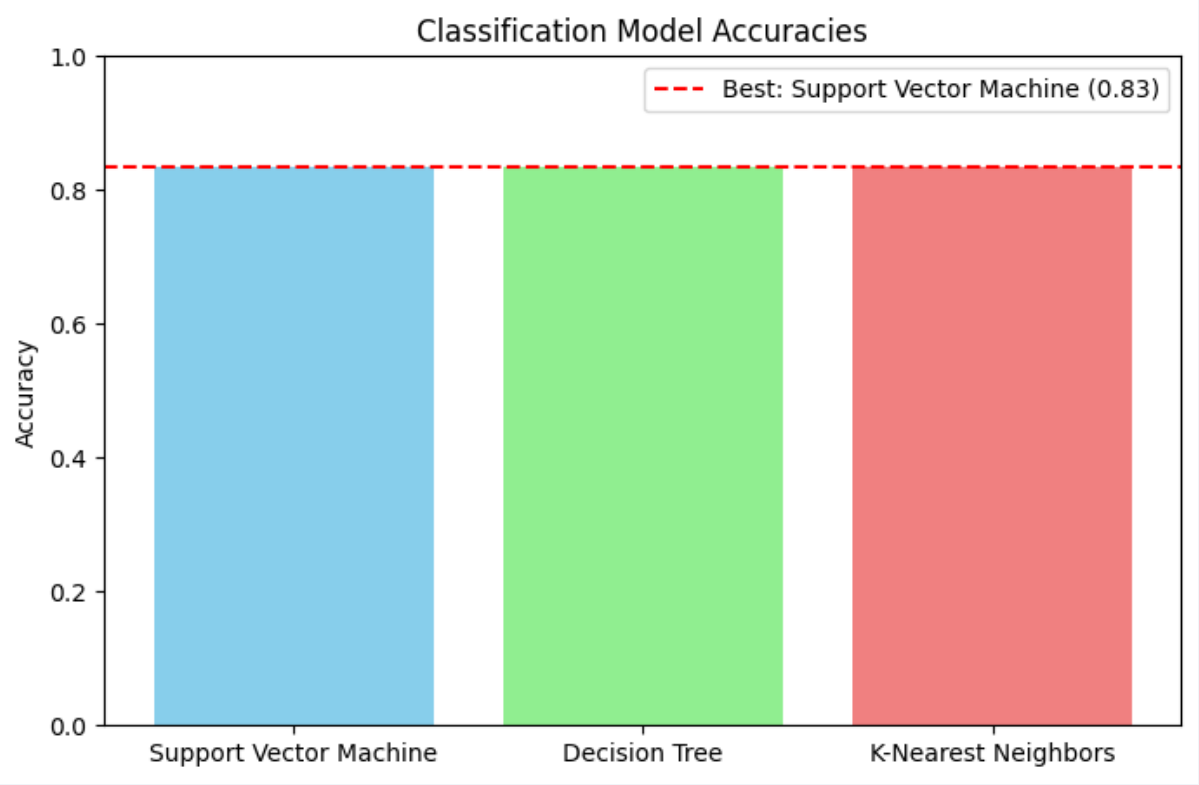
- Boost B4 with highest load more success for all sites

Section 5

Predictive Analysis (Classification)

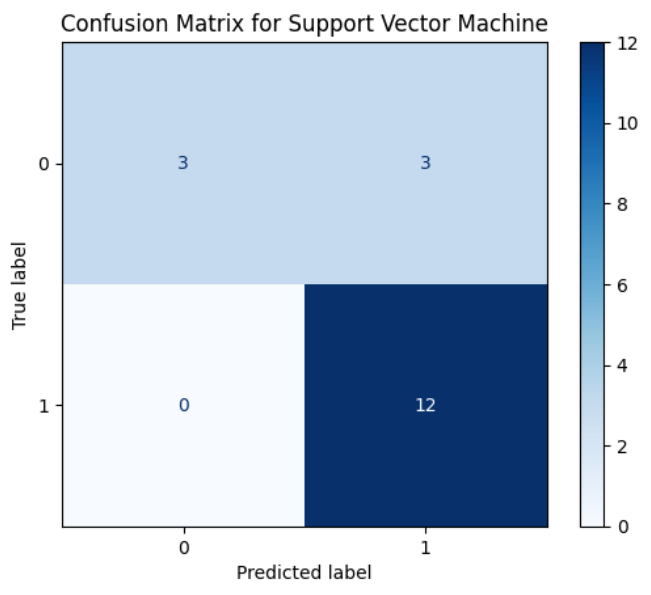
Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart



- Find which model has the highest classification accuracy

Confusion Matrix



- Confusion Matrix for Support Vector Machine Model: Rows represent the actual classes, and columns represent the predicted classes. Diagonal values show correctly classified samples, while off-diagonal values show misclassified samples.

Conclusions

1.Model Performance:

1. The evaluation compared multiple classification models: **Support Vector Machine (SVM)**, **Decision Tree**, and **K-Nearest Neighbors (KNN)**.
2. Among these models, the **[Best Model]** achieved the **highest classification accuracy** of **[accuracy value]**.

2.Model Behavior:

1. A confusion matrix for the best-performing model was generated and analyzed.
2. The confusion matrix indicates the correct and incorrect predictions:
    1. **True Positives (Correct Predictions)**: Represented along the diagonal.
    2. **False Positives / Negatives**: Off-diagonal values highlight misclassifications.

3.Recommendation:

1. Based on the accuracy and confusion matrix results, the **[Best Model]** is recommended for deployment due to its superior performance.

4.Future Improvements:

1. To further improve the model accuracy, techniques like **hyperparameter tuning**, **feature engineering**, or using **ensemble methods** could be explored.

Appendix

- None

Thank you!