IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Sucheng Li>
<Feb 17>

Outline

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

Executive Summary

- Summary of methodologies:
  - Data collection through API and Web Scraping
  - Data wrangling
  - Data Analysis with SQL, Visualizations, and Folium
  - Machine Learning for prediction
- Summary of all results
  - Exploratory results
  - Interactive results from Dash
  - Predictive results form Machine Learning

Introduction

- Project background and context
  - Falcon 9 rocket is a launched by Space X on several launch sites. The major cost saving is from reusing of the first stage. Thus, if the first stage will land would have significant influences on the overall launching cost. By exploring data from different sources with various tools, this project aims to explain the correlation between all the recorded factors and the likelihood of launch success. A machine pipeline is built in the end.
- Problems you want to find answers
  - What are the factors (features) that impact landing?
  - What is the correlation between the features and success rate of landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - API
    - https://api.spacexdata.com/v4
  - Web Scraping
    - https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- Perform data wrangling
  - A True/False labels were added to the data set to indicate the success result of landing.
  - Data was filtered and cleaned
  - Missing values were handled with mean

Methodology

Executive Summary

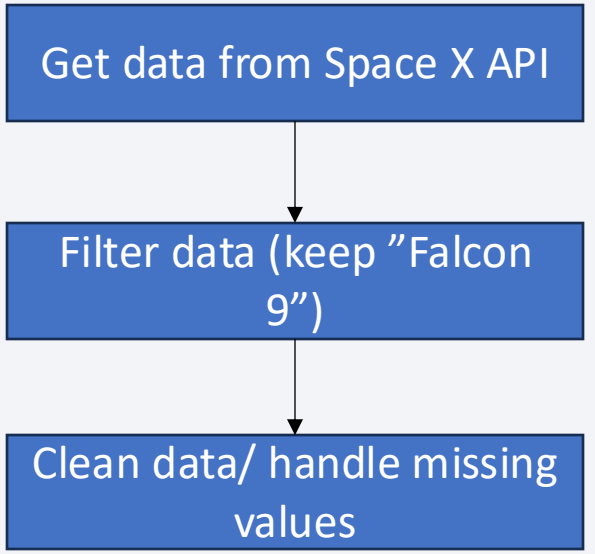
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Data was normalized and then divided into training set and testing set.
  - Evaluated by different Machine Learning Algorithms, the overall accuracy was recorded.

Data Collection

- Describe how data sets were collected.
  - Data sets are collected from API and Web scraping.
- You need to present your data collection process use key phrases and flowcharts

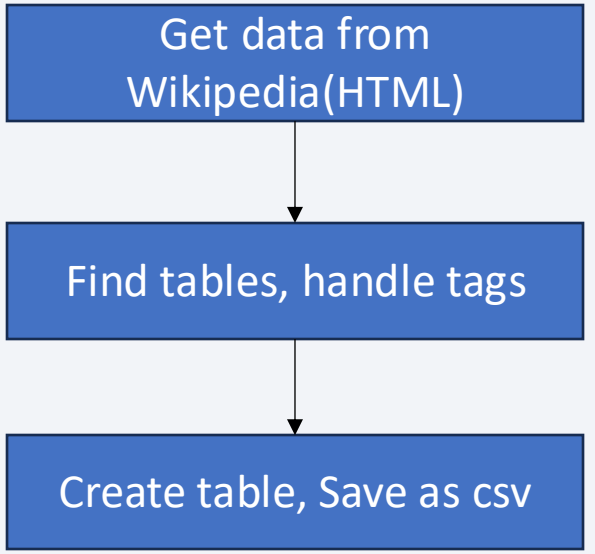
Data Collection – SpaceX API

- GET data from the Space X API → Filter data → clean data and handle missing values
- GitHub URL: https://github.com/kkgob/IBM_ds_final/blob/main/parts_before_dash/spacex-data-collection-api.ipynb



Data Collection - Scraping

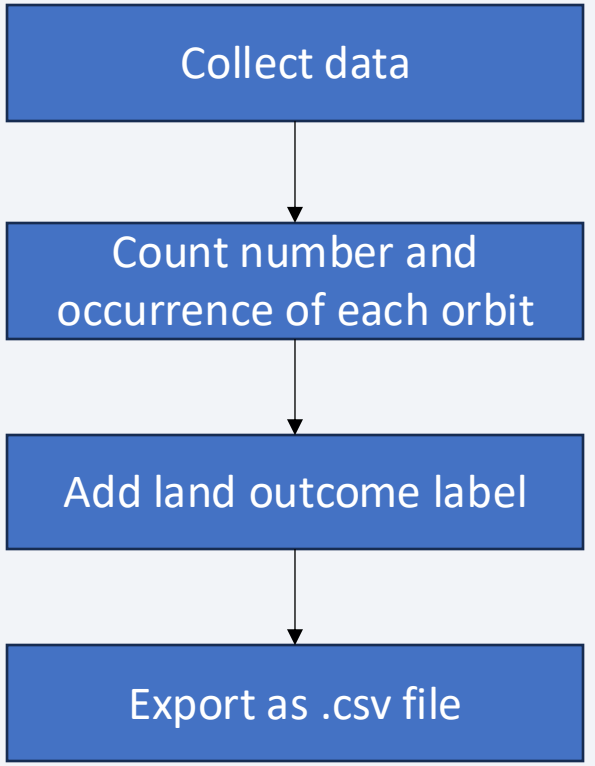
- Web scraping Wikipedia → find HTML table header → create table and save as .csv



- GitHub URL:
- https://github.com/kkgob/IBM_ds_final/blob/main/parts_before_dash/webscraping.ipynb

Data Wrangling

- Collect data → Count the number and occurrence of each orbit → add land outcome label → export as .csv
- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL: https://github.com/kkgob/IBM_ds_final/blob/main/parts_before_dash/Data%20wrangling.ipynb



EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts:
  - Flight number vs Launch → check the flight number distribution in the launch sites
  - Payload mass vs Launch Site → check the payload distribution in the launch sites
  - Success rate by orbit type → check which orbits have the highest success rate
  - Flight number vs Orbit type → check if there is any relationships between flight number and orbit type
  - Payload mass vs Orbit type → check if there is any relationships between payload mass and orbit type
  - Launch success rate over years → investigate the historical data
- GitHub URL: https://github.com/kkgob/IBM_ds_final/blob/main/parts_before_dash/edadataviz.ipynb

EDA with SQL

GitHub URL:

https://github.com/kkgob/IBM_ds_final/blob/main/parts_before_dash/eda-sql-coursera_sqllite.ipynb

- Using bullet point format, summarize the SQL queries you performed
  - Find the names of the unique launch sites
  - Find 5 records where launch sites begin with `CCA`
  - Calculate the total payload carried by boosters from NASA
  - Calculate the average payload mass carried by booster version F9 v1.1
  - Find the dates of the first successful landing outcome on ground pad
  - List the names of the booster which have carried the maximum payload mass
  - List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
  - Calculate the total number of successful and failure mission outcomes
  - List the names of the booster which have carried the maximum payload mass
  - List the failed landing outcomes in drone ship, their booster versions, and launch site names for 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

Build an Interactive Map with Folium

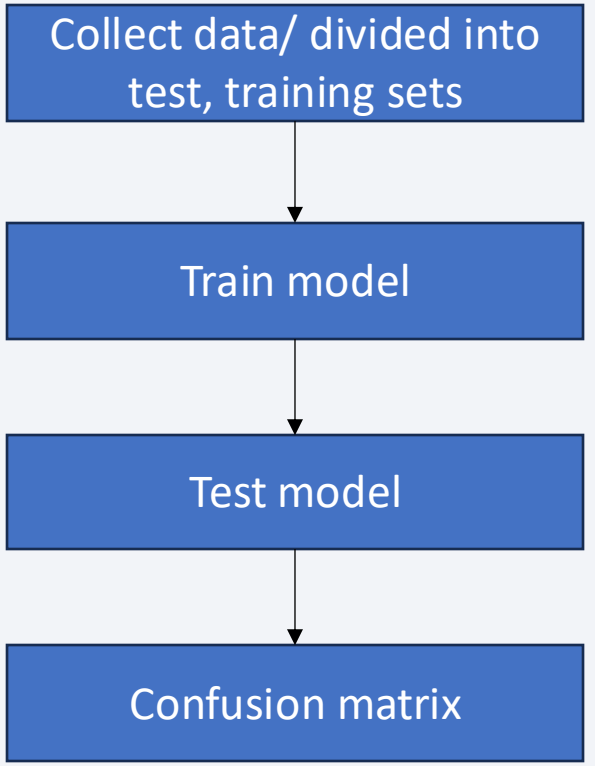
- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
  - All launch sites in US → show all the locations of the launch sites
  - Launch sites with circles and labels → show the cluster of launch sites on the map
  - Distance to the highway → A example of the distance from one launch site to one highway
- GitHub URL:
- https://github.com/kkgob/IBM_ds_final/blob/main/parts_before_dash/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
  - Pie chart → visualize the data for both all sites and individual sites
  - Scatter Plot → visualize the data for both all sites and individual sites
  - Payload range → adjust the filter for payloads
- Add the GitHub URL:
- https://github.com/kkgob/IBM_ds_final/tree/main/dash

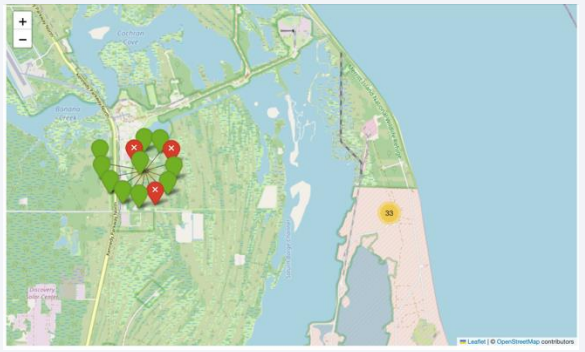
Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL:
- https://github.com/kkgob/IBM_ds_final/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

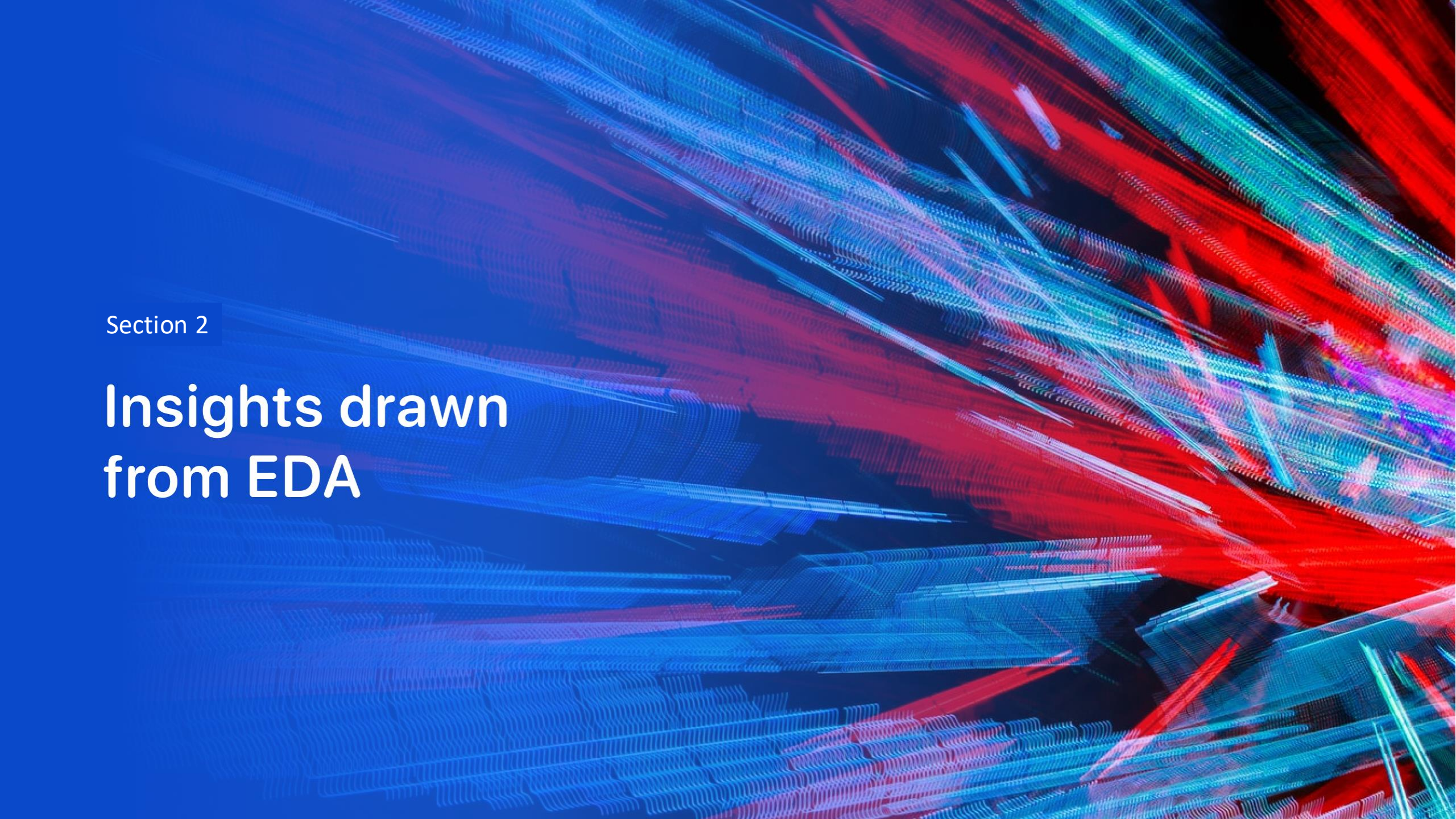


Results

- Exploratory data analysis results
  - The larger the flight amount at a launch site, the greater the success rate at a launch site.
  - **Higher payloads generally have a higher success rate**
  - The success rate is increasing over the years
- Interactive analytics demo in screenshots

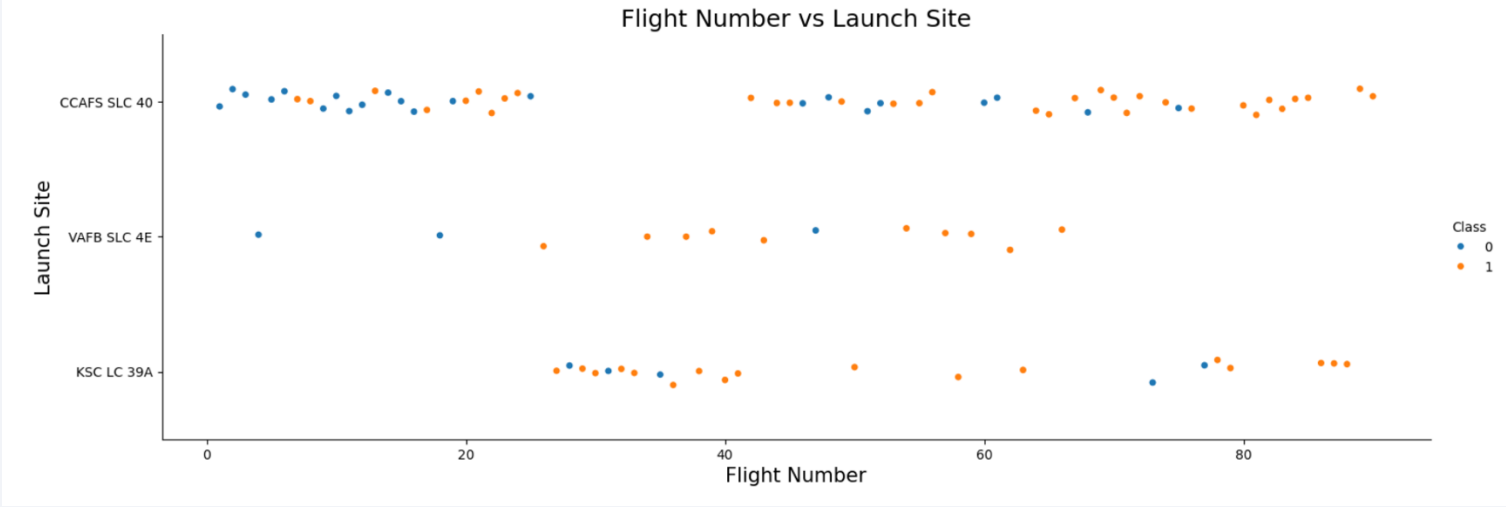


- Predictive analysis results
  - The **Decision Tree model** achieved the highest classification accuracy (**89%**), making it the most effective at predicting landing outcomes.


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site



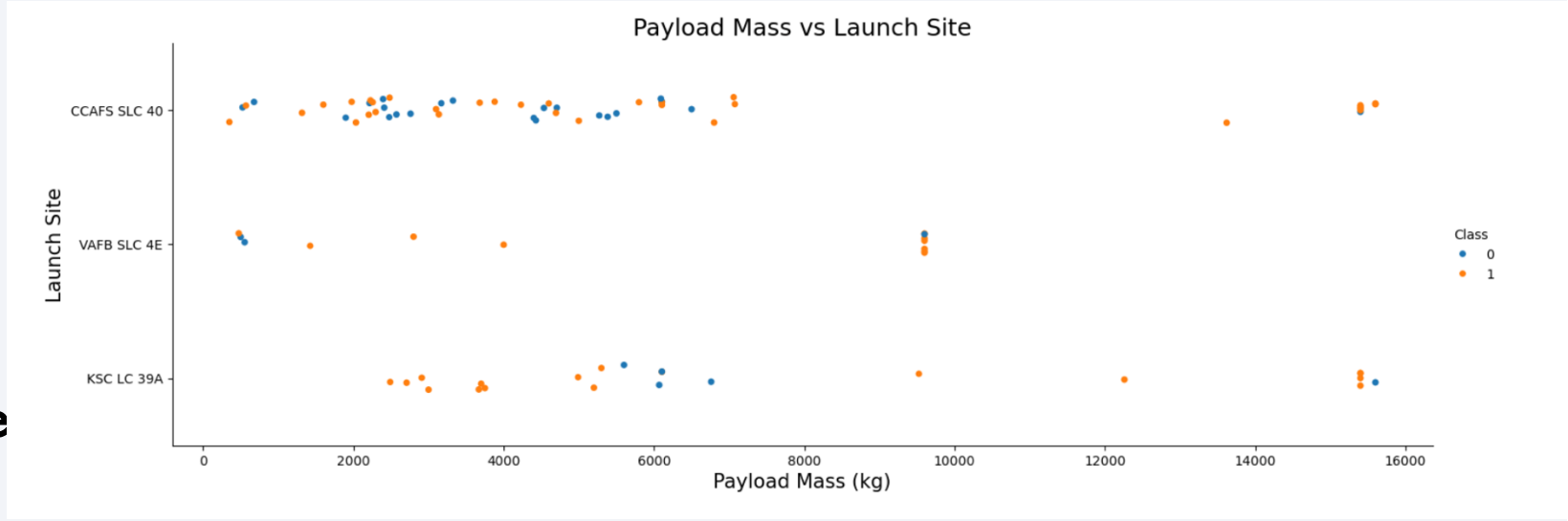
- Explanations:
- The larger the flight amount at a launch site, the greater the success rate at a launch site.

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

- Explanations:

- **Higher payloads generally have a higher success rate**

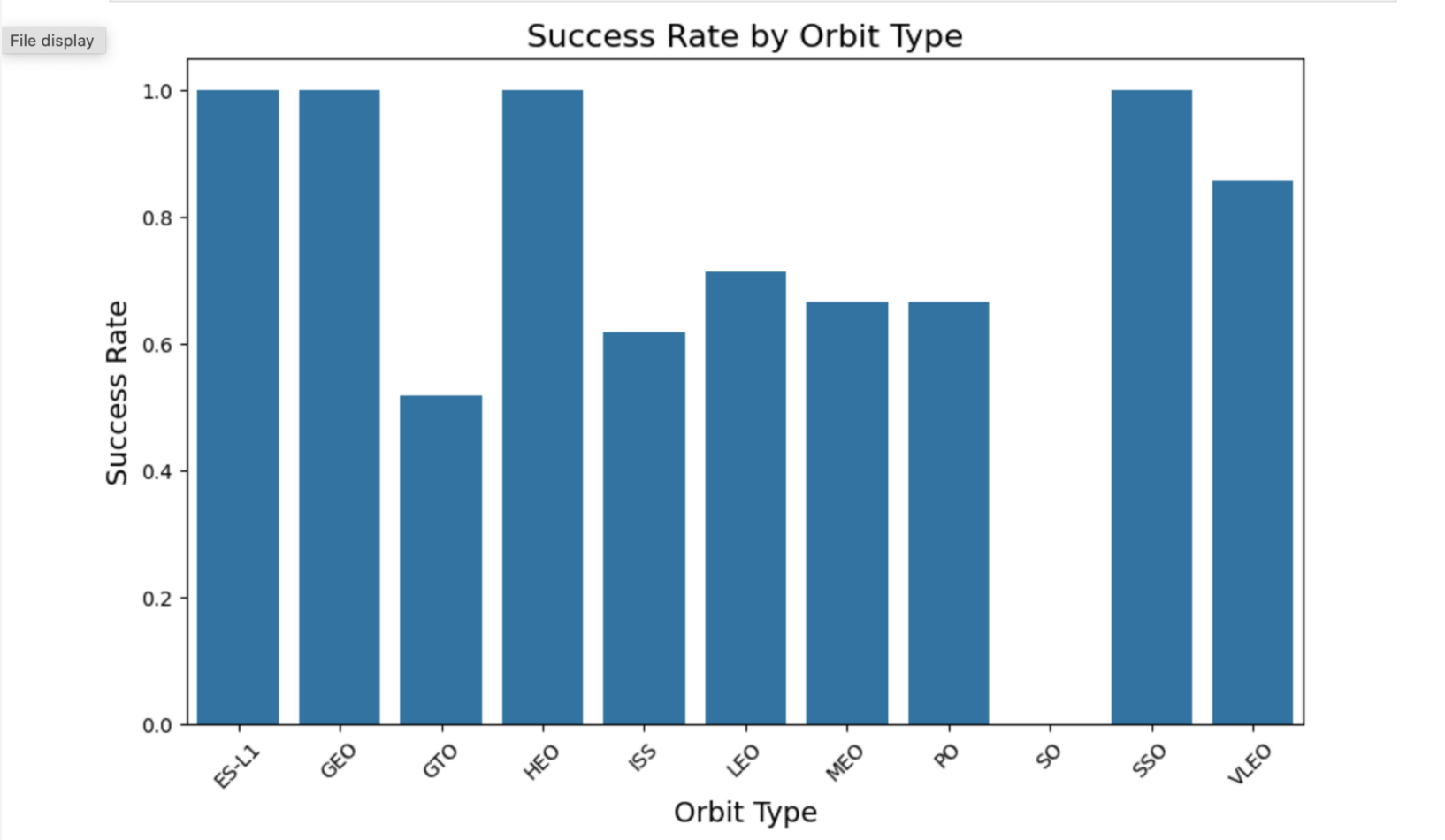


Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

- Explanations:

- **GTO has the lowest success rate**, likely due to its demanding orbital insertion process.
- **SSO, GEO, and ES-L1 are the most reliable orbits** with a 100% success rate.
- **LEO and ISS missions have moderate success rates**, indicating challenges in launching to these orbits but also improvements over time.
- **Overall, SpaceX has achieved high reliability for most orbits**, with improvements seen in high-demand orbits like LEO and ISS.

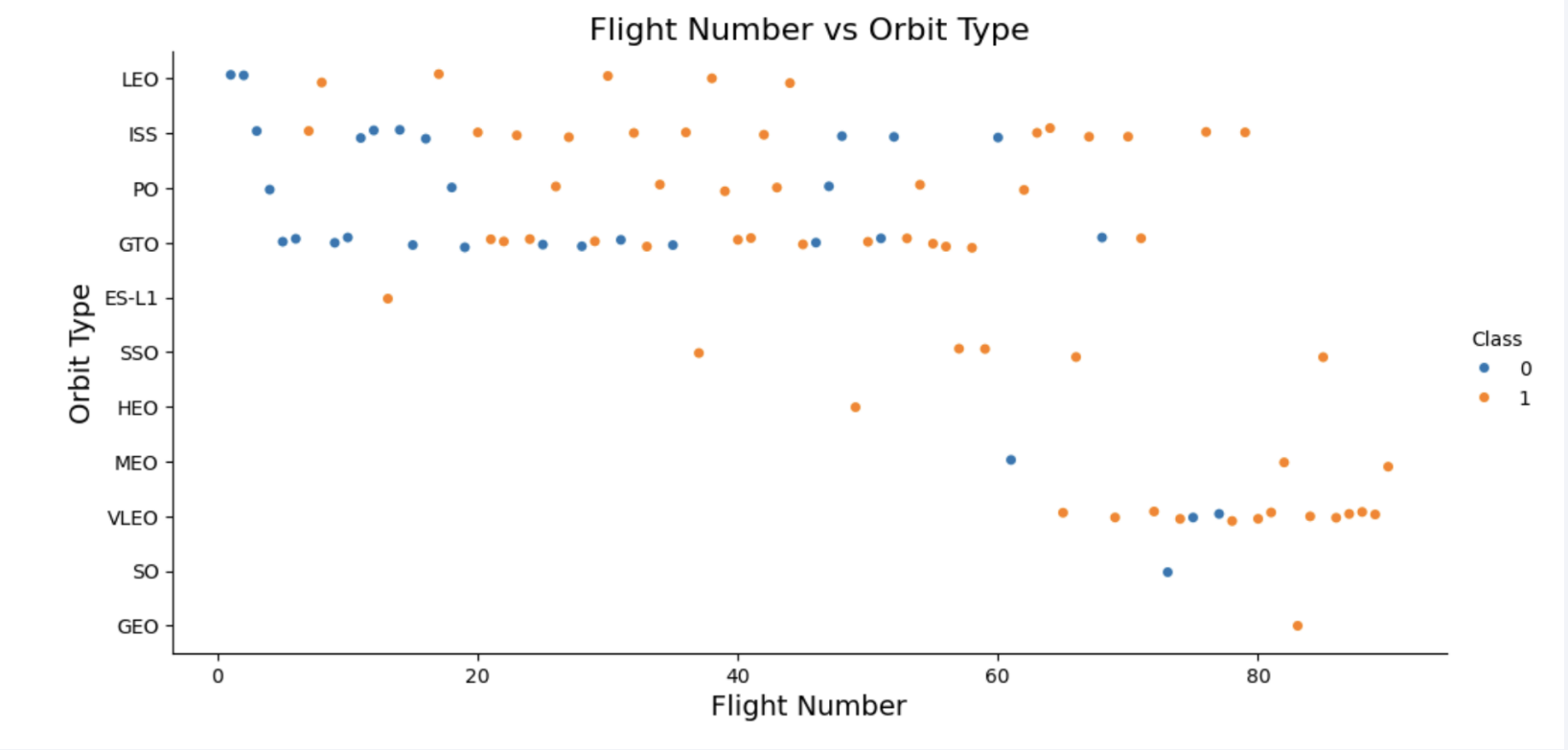


Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

- Explanations:

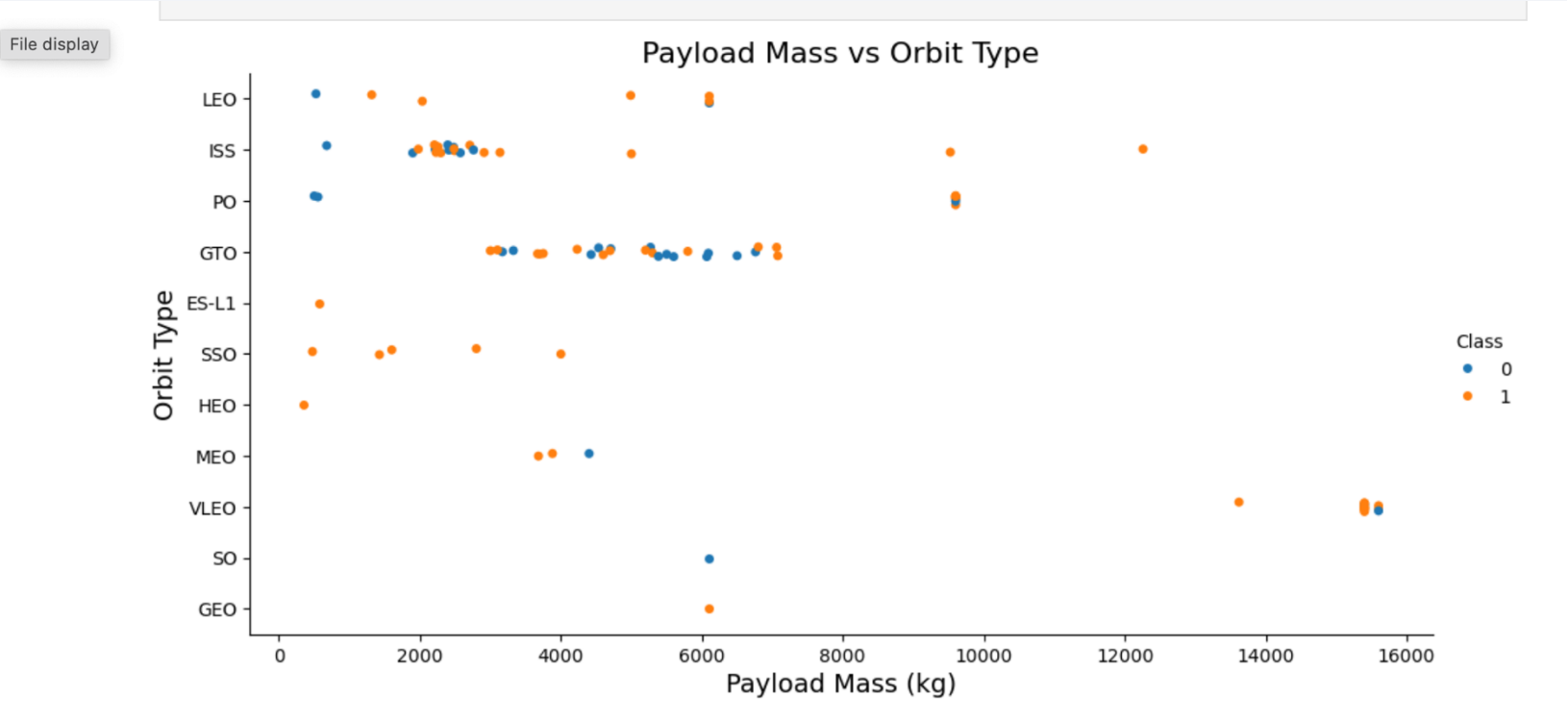
- You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.



Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

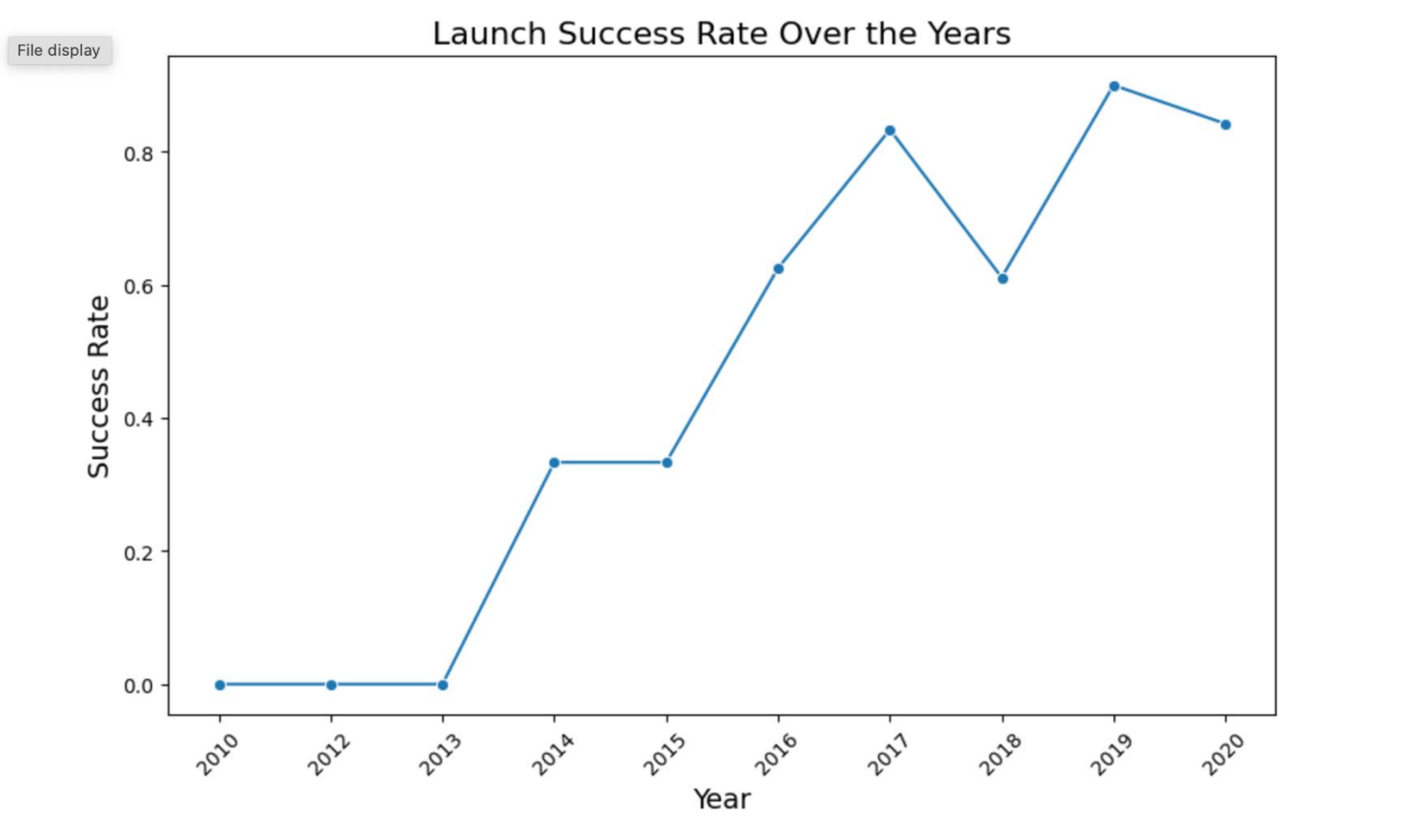
- Explanations:
- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.



Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Explanations:
- The success rate is increasing over the years.



All Launch Site Names

- Find the names of the unique launch sites
- Explanation: 5 sites are found

```
In [11]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
 * sqlite:///my_data1.db
Done.
```

Out[11]:

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

Launch Site Names Begin with 'CCA'

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

File display

Task 2

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

In [12]:
```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

Out[12]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (p
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 (p
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

File display

Task 3

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

In [13]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_Payload_Mass FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';
```

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

Out[13]:

Total_Payload_Mass
45596

Average Payload Mass by F9 v1.1

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

File display

Task 4

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

In [14]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS Avg_Payload_Mass FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1';
```

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

Out[14]:

Avg_Payload_Mass
2928.4

First Successful Ground Landing Date

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

Task 5

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

Hint:Use min function

In [15]:
```
%sql SELECT MIN(Date) AS First_Successful_Ground_Landing FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (groun
```

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

Out[15]:

First_Successful_Ground_Landing
2015-12-22

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

display

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

In [16]:
```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

Out[16]:

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes

In [17]:
```
%sql SELECT Mission_Outcome, COUNT(*) AS Total_Count FROM SPACEXTABLE GROUP BY Mission_Outcome;
```

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

Out[17]:

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

Boosters Carried Maximum Payload

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

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

```
In [18]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX
```

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

Out[18]:

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

```
(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

File display

Task 9

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [19]:
```
%sql SELECT SUBSTR(Date, 6, 2) AS Month,Landing_Outcome,Booster_Version,Launch_Site FROM SPACEXTABLE WHERE Landin
```

```
on,Launch_Site FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Failure (drone ship)' AND SUBSTR(Date, 1, 4) = '2015';
```

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

Out[19]:

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

- 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 LANDING_OUTCOME, COUNT(*) AS OUTCOME_COUNT FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' AND LANDING_OUTCOME IN ('FAILURE (DRONE SHIP)', 'SUCCESS (GROUND PAD)') GROUP BY LANDING_OUTCOME ORDER BY OUTCOME_COUNT DESC;

File display

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.

In [20]:
```
%sql SELECT Landing_Outcome, COUNT(*) AS Outcome_Count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017
```

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

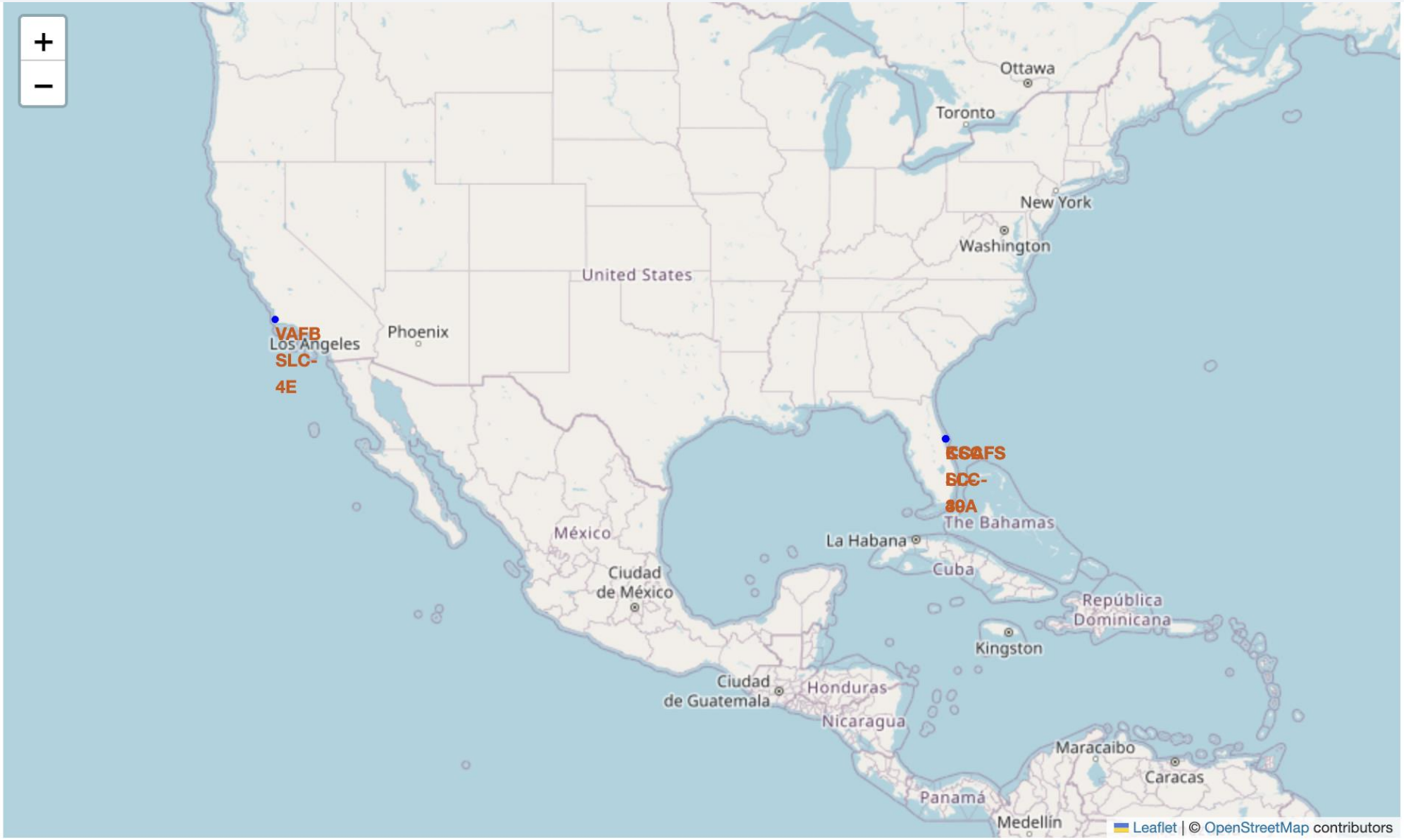
Out[20]:

Landing_Outcome	Outcome_Count
Failure (drone ship)	5
Success (ground pad)	3

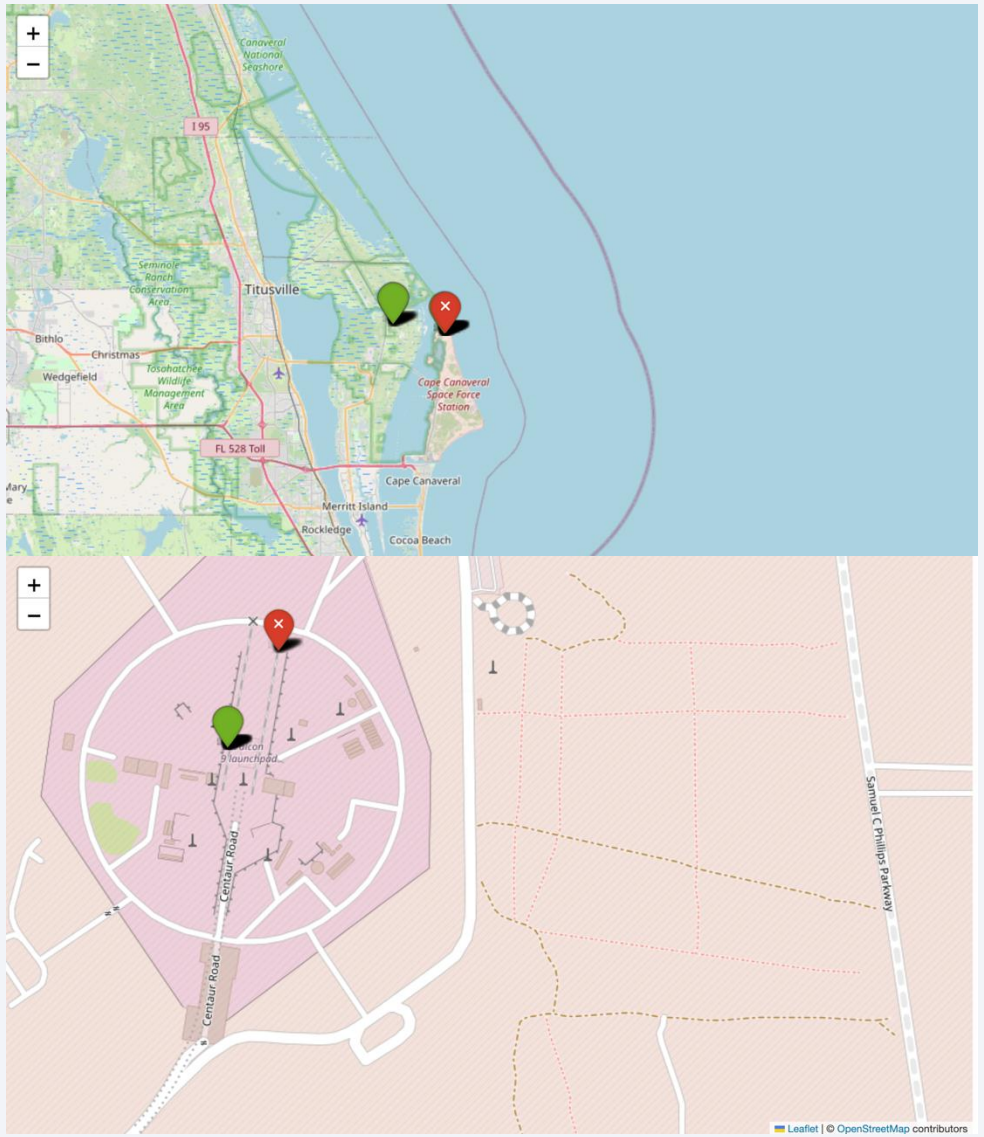
Section 3

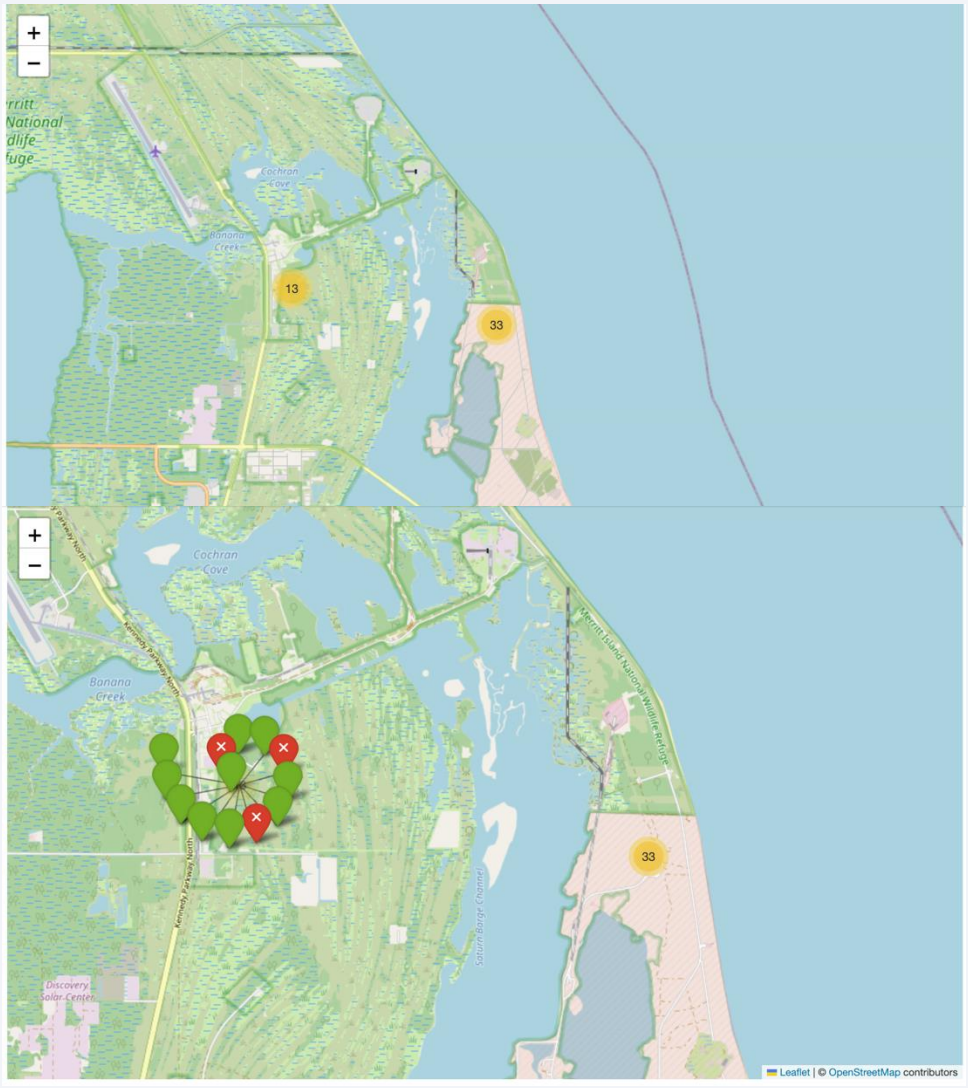
Launch Sites Proximities Analysis

All launch sites in US

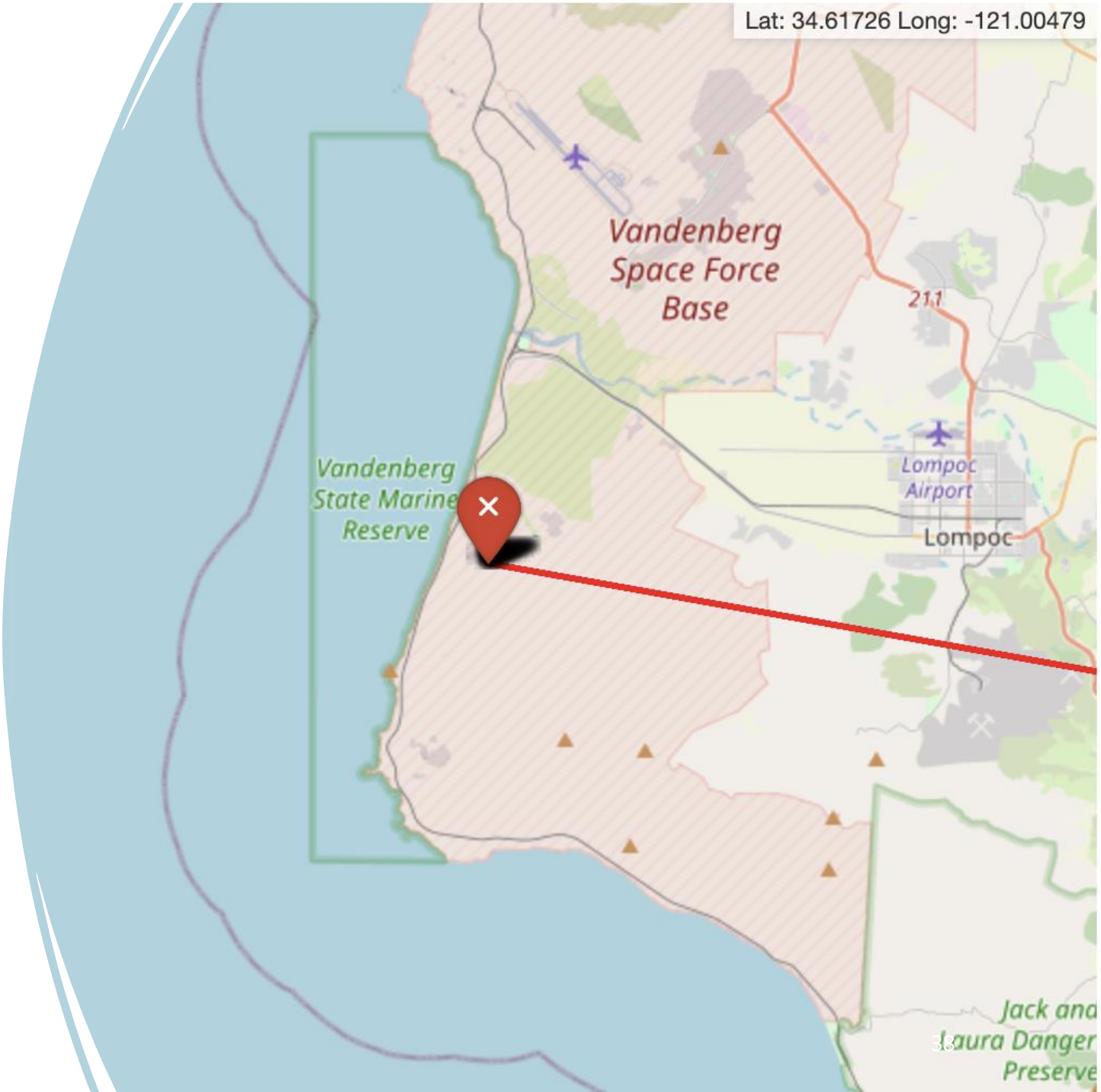


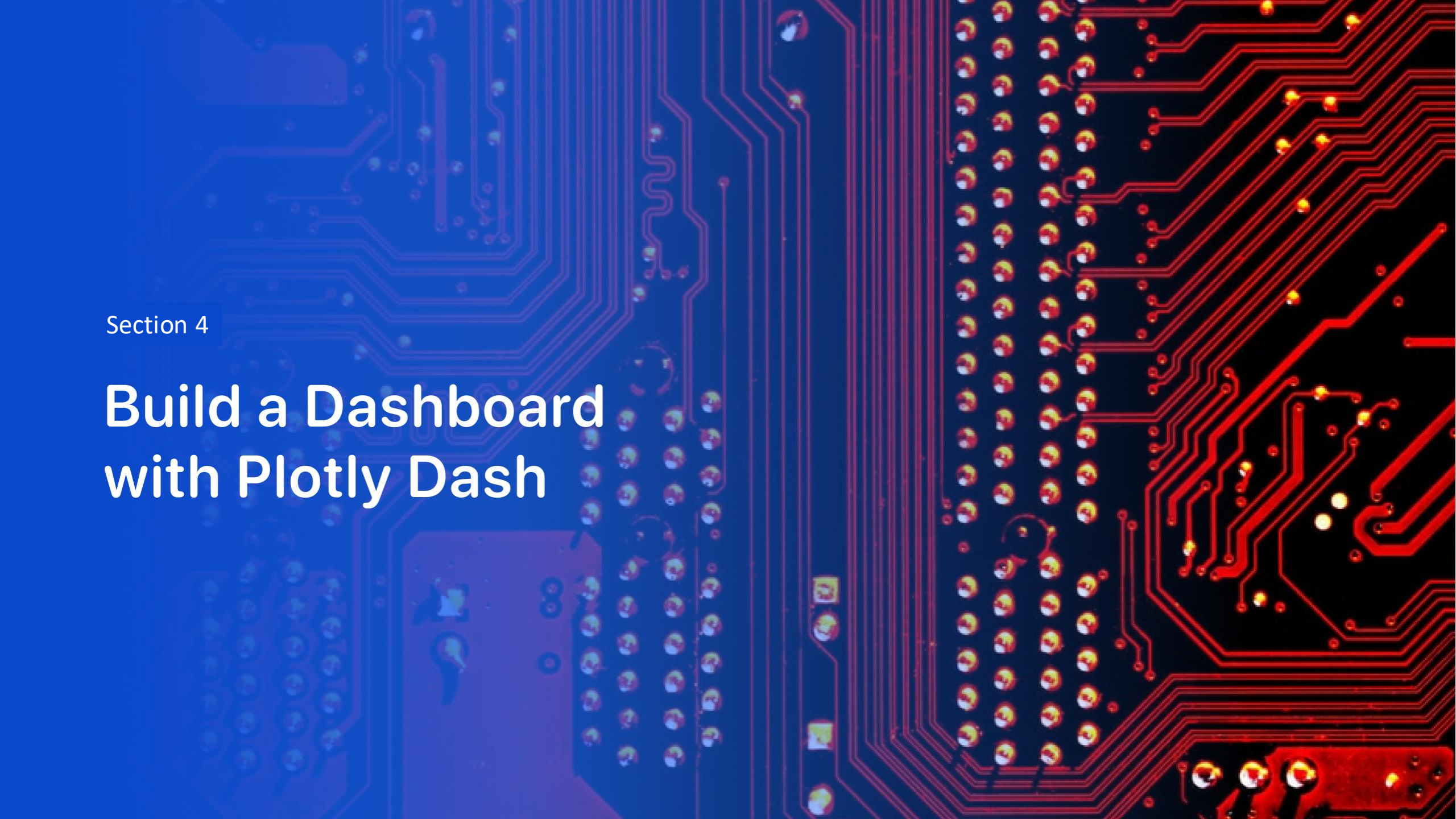
Launch sites with circles and labels



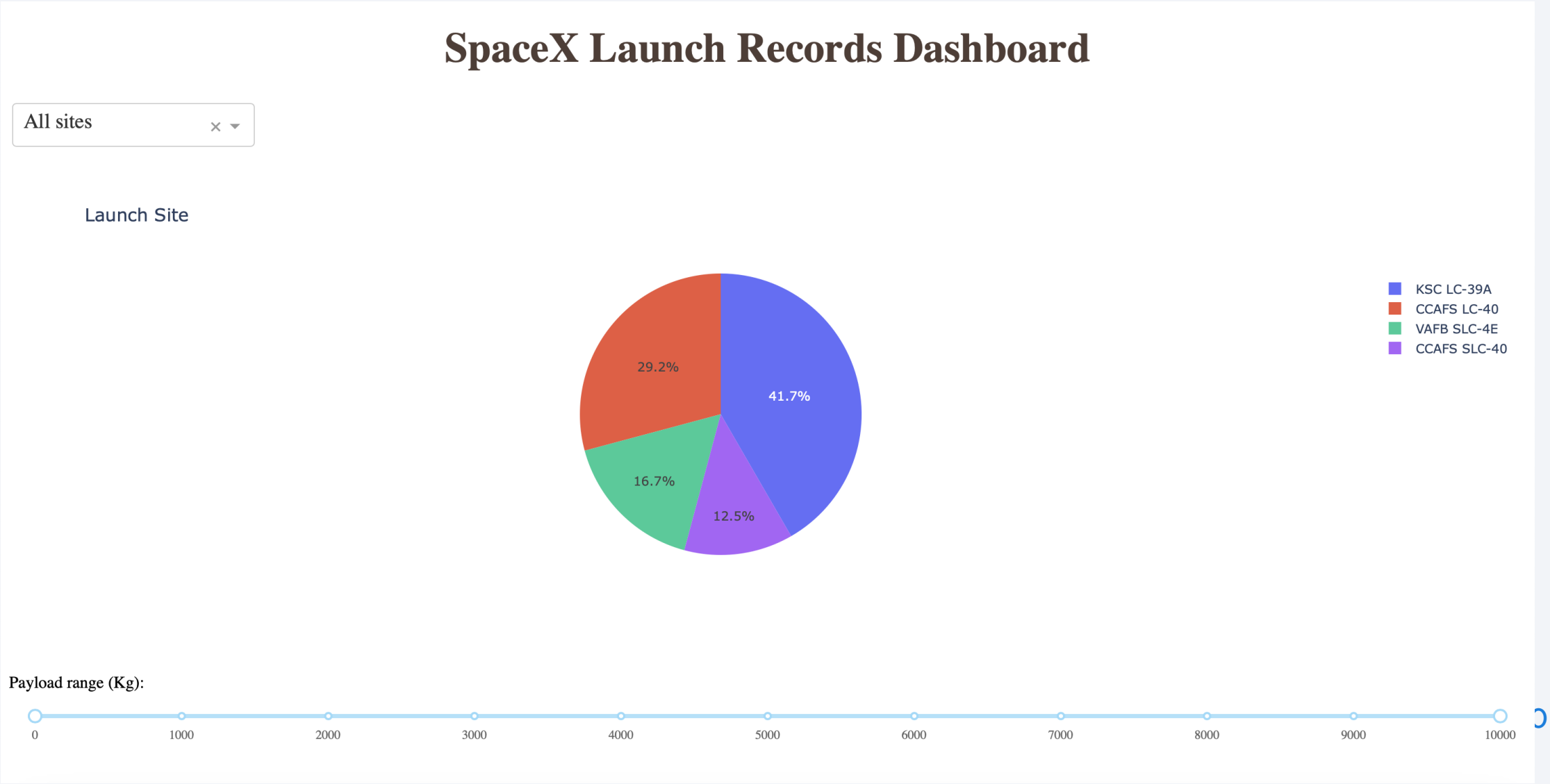


Distance to the highway

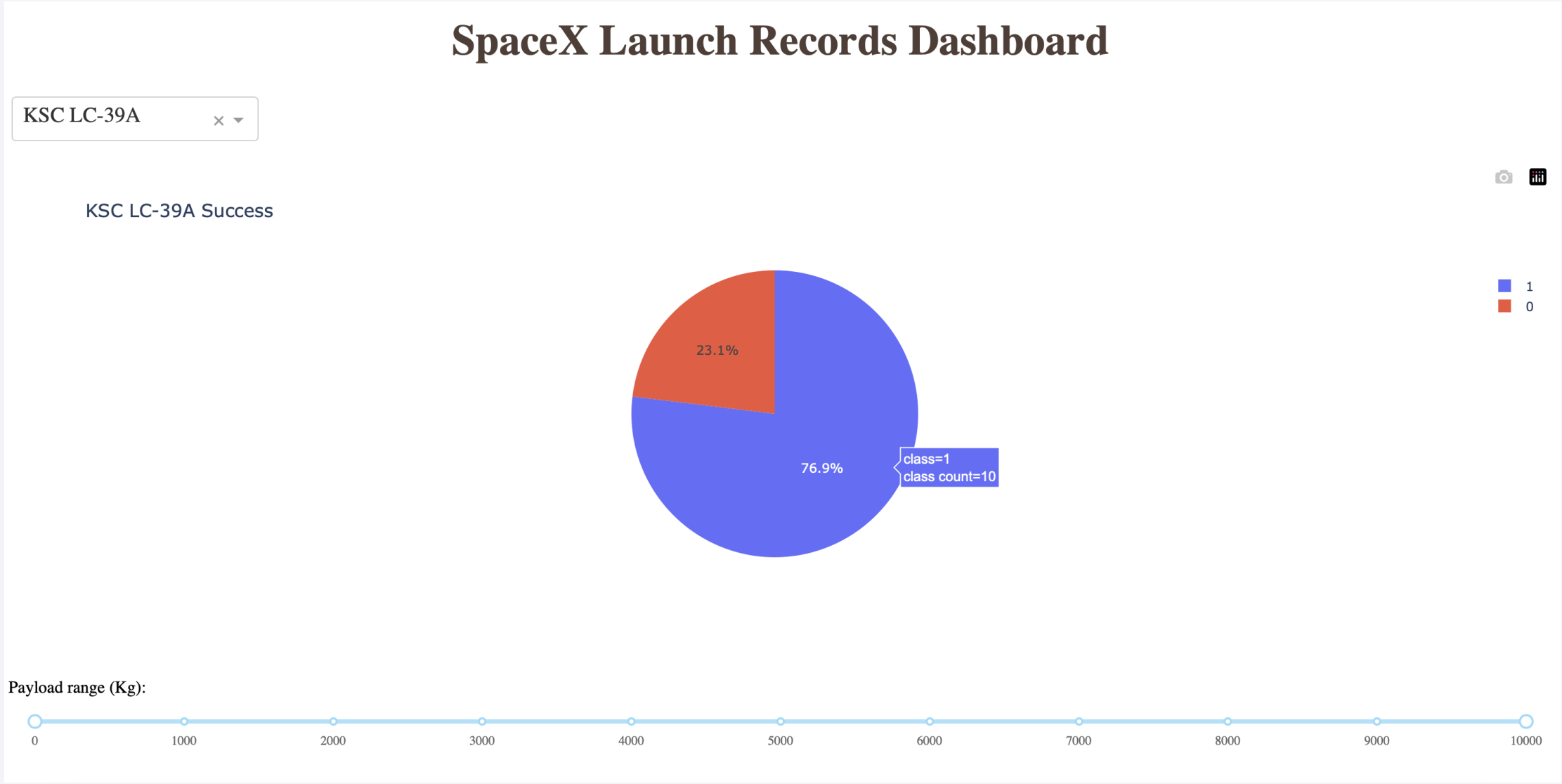



Section 4
Build a Dashboard with Plotly Dash

Dashboard pie chart



Launch site with highest launch success ratio



Payload vs. Launch Outcome scatter plot for all sites 0-10000

Payload vs. Launch Outcome scatter plot for all sites 5000-10000

Payload vs. Launch Outcome scatter plot for all sites 3000-8000

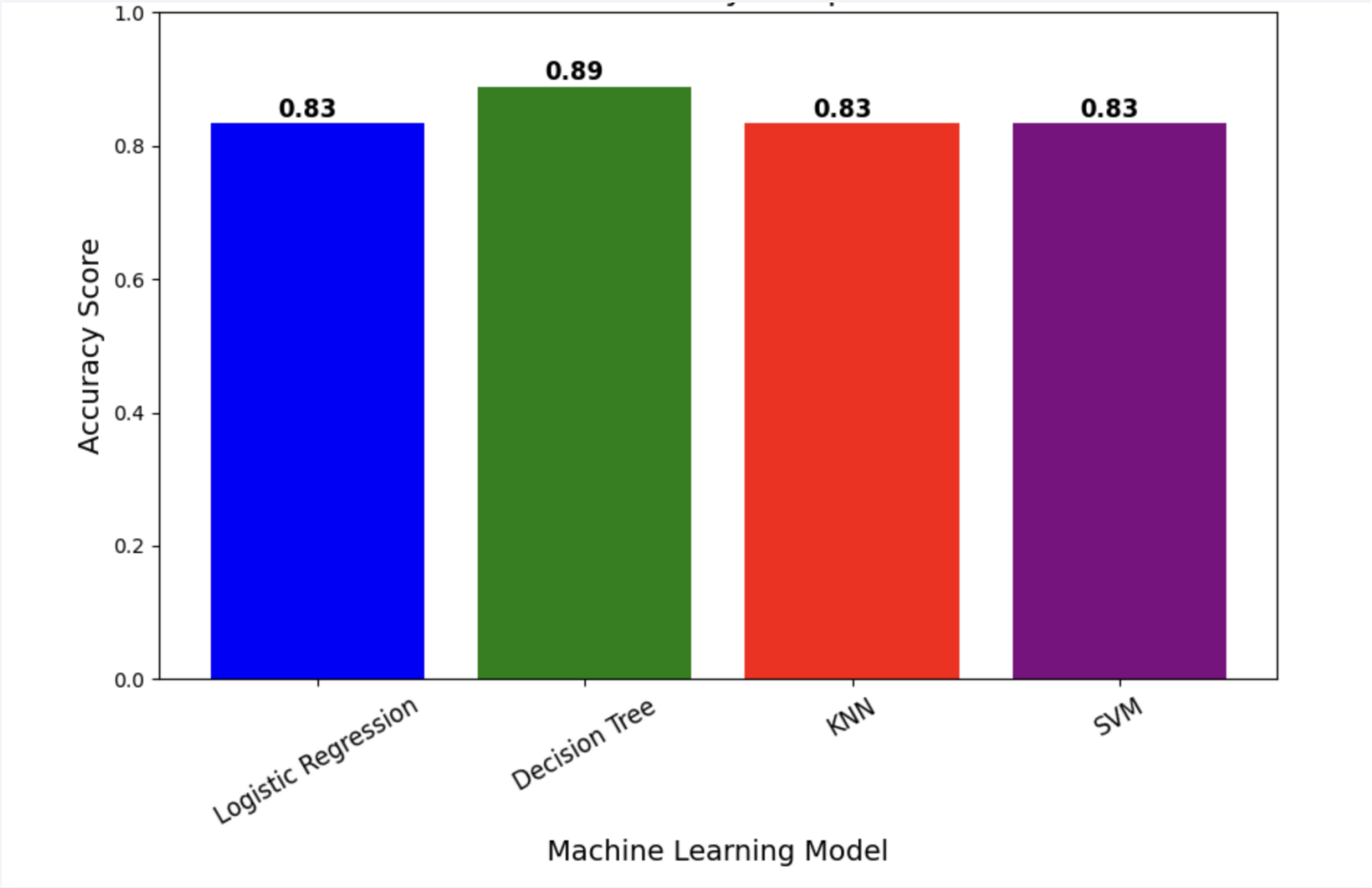
Section 5

Predictive Analysis (Classification)

Classification Accuracy → Decision tree

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

```
model_accuracies = {
    "Logistic Regression": logreg_cv.score(X_test, Y_test),
    "Decision Tree": tree_cv.score(X_test, Y_test),
    "KNN": knn_cv.score(X_test, Y_test),
    "SVM": svm_cv.score(X_test, Y_test)
}

# Extract model names and accuracy values
models = list(model_accuracies.keys())
accuracies = list(model_accuracies.values())

# Create the bar chart
plt.figure(figsize=(10, 6))
bars = plt.bar(models, accuracies, color=['blue', 'green', 'red', 'purple'])

# Add accuracy labels on top of bars
for bar in bars:
    plt.text(bar.get_x() + bar.get_width()/2 - 0.15, bar.get_height() + 0.01,
             f"{bar.get_height():.2f}", fontsize=12, fontweight='bold')

# Chart formatting
plt.xlabel("Machine Learning Model", fontsize=14)
plt.ylabel("Accuracy Score", fontsize=14)
plt.title("Model Accuracy Comparison", fontsize=16)
plt.ylim(0, 1)  # Ensuring the scale stays between 0 and 1
plt.xticks(rotation=30, fontsize=12)  # Rotate x-axis labels for better readability

# Show the plot
plt.show()
```



Confusion Matrix

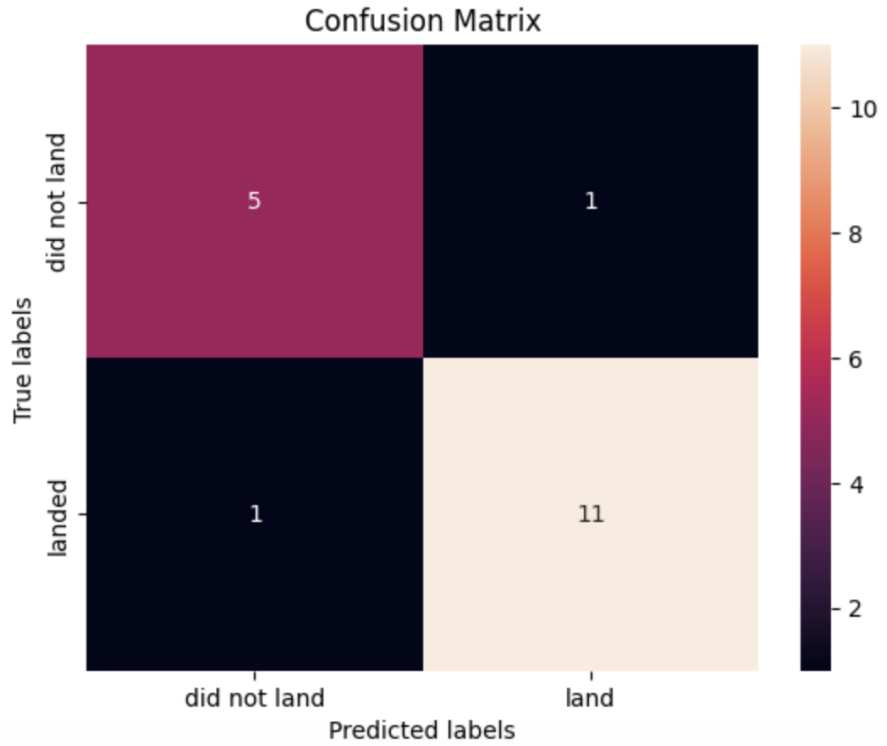
- Explanation:
- **Handles Categorical & Numerical Data Well:**
- The dataset contains categorical data (e.g., booster version category, launch site) and numerical data (e.g., payload mass). Decision Trees are good at handling both types without complex preprocessing.
- **Captures Non-Linear Relationships:**
- Unlike Logistic Regression, which assumes linear relationships, Decision Trees can model complex decision boundaries.
- • **Interpretable & Explainable:**
- Decision Trees provide a **clear structure** that explains why a specific prediction was made, unlike SVM or Neural Networks, which are often **black-box models**.

```
[26]: accuracy=tree_cv.score(X_test, Y_test)
      print("Test Accuracy:", accuracy)
```

```
Test Accuracy: 0.8888888888888888
```

We can plot the confusion matrix

```
[27]: yhat=tree_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- The analysis identified several key factors that **correlate with successful landings**:
- **Payload Mass:** Higher payloads tend to have higher success rates in specific orbits (e.g., **LEO, ISS, and Polar orbits**).
- **Launch Site:** Certain launch sites exhibit **higher success rates**, with **KSC LC 39A and CCAFS SLC 40** being the most reliable.
- **Orbit Type: GTO (Geostationary Transfer Orbit) has the lowest success rate**, while **SSO, GEO, and ES-L1 have 100% success rates**.
- The study confirms that **machine learning models** can predict the success of SpaceX Falcon 9 rocket landings with high accuracy.
- The **Decision Tree model** achieved the highest classification accuracy (**89%**), making it the most effective at predicting landing outcomes.

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!