Winning Space Race with Data Science

Nataliia Lauria
Apr 21, 2024

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

Introduction

- Project background and context
- 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.

•Problems you want to find answers

-What factors determine if the rocket will land successfully?

-The interaction amongst various features that determine the success rate of a successful landing.

-What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Source: The data was collecter from the SpaceX API.
- Perform data wrangling
  - Cleaning the data to remove any inconsistencies or null values.
  - Structuring the data into a format suitable for analysis.
- 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

- SpaceX API
  - API calls were made to SpaceX’s endpoints.
  - Responses from theses calls provided JSON-formatted data.
  - Normalize JSON formatted data
  - Creating a data frame
  - Filter and cleaning the data
- Web Scrapping from Wikipedia
  - Using Python libraries such as BeautifulSoup.
  - The HTML content was parsed and required information extracted into tables

Data Collection – SpaceX API

- https://github.com/natfili01/IBM_data-_project/blob/main/jupyter-labs-spacex-data-collection-api(2).ipynb

API calls were made to SpaceX's endpoints.

```
[8]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
[9]: response.status_code
[9]: 200
```

Creating a data frame

```
[22]: # Create a data from launch_dict
data = pd.DataFrame(launch_dict)
```

Show the summary of the dataframe

```
[23]: # Show the head of the dataframe
data.head()
```

[23]:

FlightNumber	Date	BoosterVersion

Normalize JSON formatted data

```
[10]: # Use json_normalize meethod to convert the json result into a dataframe
response = requests.get (static_json_url)
data = response.json()
df=pd.json_normalize(data)
```

Using the dataframe `data` print the first 5 rows

```
[11]: # Get the head of the dataframe
df.head()
```

[11]:

static_fire_date_utc	static_fire_date_unix	tbd	net	window

Filter and cleaning the data

```
data_falcon91 = data[data['BoosterVersion']=='Falcon 9']
data_falcon91.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass
4	6	2010-06-04	Falcon 9	NaN

```
# Calculate the mean value of PayloadMass column
mean_payload_mass = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].fillna(mean_payload_mass)
data_falcon9['LandingPad'] = data_falcon9['LandingPad'].fillna(mean_payload_mass)
data_falcon9.isnull().sum()
```

Data Collection - Scraping

- Import libraries for fetching webpage and BeautifulSoup for parsing HTML
- Pace the data and create a data frame
- https://github.com/natfili01/IBM_data-_project/blob/main/jupyter-labs-webscraping.ipynb

Import packages

```
[ ]: !pip3 install beautifulsoup4
     !pip3 install requests

[2]: import sys

     import requests
     from bs4 import BeautifulSoup
     import re
     import unicodedata
     import pandas as pd
```

Create a BeautifulSoup object

Create a `BeautifulSoup` object from the HTML `response`

```
[9]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
     data_soup = BeautifulSoup(data.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
[10]: # Use soup.title attribute
      data_soup = BeautifulSoup(data.text, 'html.parser')
      # Extract the title of the webpage
      title = data_soup.title
      title
```

Extract column names

Next, we just need to iterate through the `<th>` elements and apply the provided `extract_column_from_header()` to extract column

```
]: # Apply find_all() function with `th` element on first_launch_tableimport requests
   th_elements = first_launch_table.find_all('th')
   # Iterate each th element and apply the provided extract_column_from_header() to get a column name
   column_names = []
   for th in th_elements:
       column_name = extract_column_from_header(th)
       # Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
       if column_name:
           column_names.append(column_name)
```

Check the extracted column names

```
]: print(column_names)

['Flight No.', 'Date and time ( )', 'Launch site', 'Payload', 'Payload mass', 'Orbit', 'Customer', 'Launch outcome']
```

Create a data frame

```
[51]: df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
      df
```

[51]:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	L
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	
1	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	
2	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	NASA	
3	2	CCAFS	Dragon	525 kg	LEO	NASA	
4	3	CCAFS	Dragon	4,700 kg	LEO	NASA	

Data Wrangling

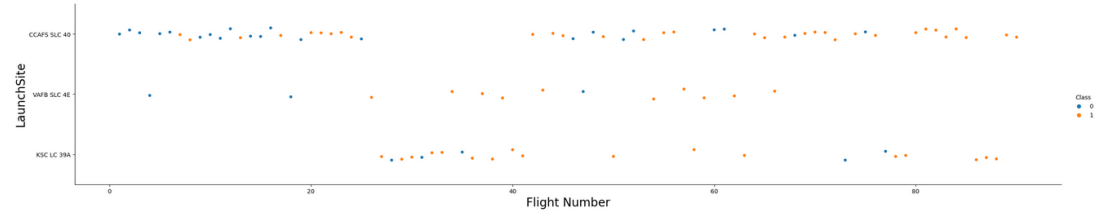
- Perform EDA
- Load Space X dataset, identify and calculate the percentage of the missing values in each attribute, Calculate the number of launches on each site
- Calculate the number and occurrences of mission outcome of the orbits
- Create a landing outcome label from Outcome column

- https://github.com/natfili01/IBM_data-_project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

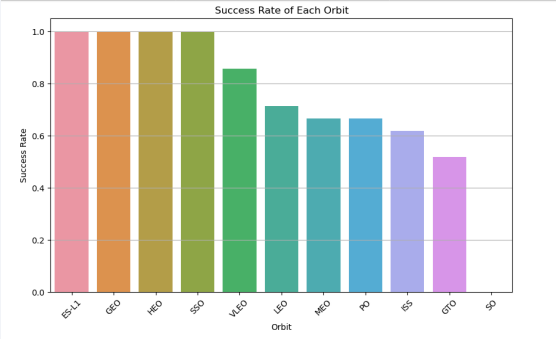
EDA with Data Visualization

Plotted charts provide insights into various aspects of the launch data:

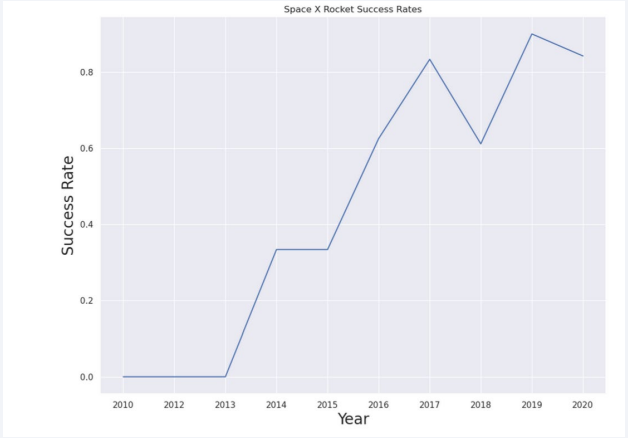
- The relationship between flight number and launch site, which could indicate



- Success rates across different orbit types, helping to identify which orbits have higher success rates.



- The overall trend of launch success over the years, providing insights into the reliability and progress of space launch activities.



- https://github.com/natfili01/IBM_data-_project/blob/main/jupyter-labs-eda-dataviz(1).ipynb

EDA with SQL

- Analyze key metrics and outcomes of space missions.

 - Display unique launch sites in the space mission.

 - List booster versions with the maximum payload mass.

 - Rank landing outcomes between specified dates.

 - Calculate total payload mass carried by NASA (CRS) boosters.-

 - Retrieve records with specific criteria such as payload mass and landing outcomes.

- https://github.com/natfili01/IBM_data-_project/blob/main/jupyter-labs-eda-sql-coursera_sqllite(2).ipynb

Build an Interactive Map with Folium

- Mark all launch sites on a map.
- Mark success/failed launches for each site on the map.
- Calculate distances between launch site and proximities.
- Interactive visual analytics using Folium enables deeper insights into launch site locations.
- Understanding geographical patterns aids in optimizing future launch site selection and operations.
- Importance of Coastal and Airport Proximity
- Coastline: Provides safety buffer for overwater launches, minimizing risks to populated areas.
- Airport: Considerations for airspace management and coordination to ensure safe launch operations.
- https://github.com/natfili01/IBM_data-_project/blob/main/lab_jupyter_launch_site_location(2).ipynb

Build a Dashboard with Plotly Dash

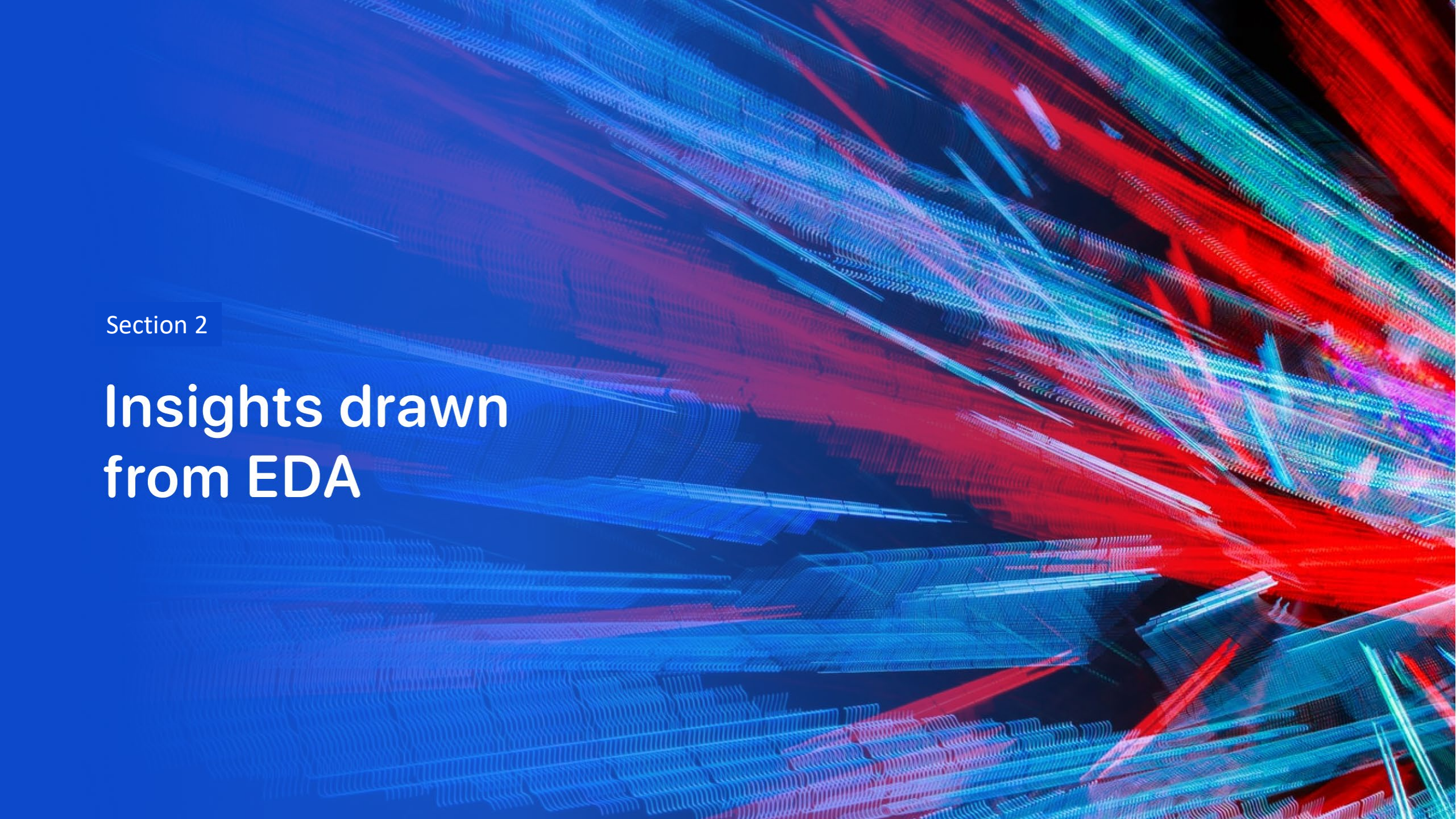
- Plots Included:
- Success Pie Chart:
  - This chart displays the proportion of successful launches at different launch sites or across all sites.
  - Purpose: Provides an immediate visual representation of success rates, making it easy to compare the effectiveness of different launch sites in terms of mission success.
- Success-Payload Scatter Chart:
  - A scatter plot correlating the payload mass with the success of launches for different booster versions.
  - Purpose: Helps to analyze whether heavier payloads have a lower success rate and how different booster versions perform across various payload ranges.
- Reasons for These Choices:
- User Engagement
- Data Exploration
- Visual Appeal and Clarity
- https://github.com/natfili01/IBM_data-_project/blob/main/Dash.ipynb

Predictive Analysis (Classification)

- Collected and cleaned relevant data, then trained multiple classification models such as Decision Tree, SVM, KNN, and Logistic Regression.
- After evaluating ML performance through cross-validation and metrics like accuracy, improved them by tuning hyperparameters and employing feature selection.
- Finally, the model with the highest accuracy was selected as the best performing model for deployment.
- https://github.com/natfili01/IBM_data-_project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.backup(1).ipynb

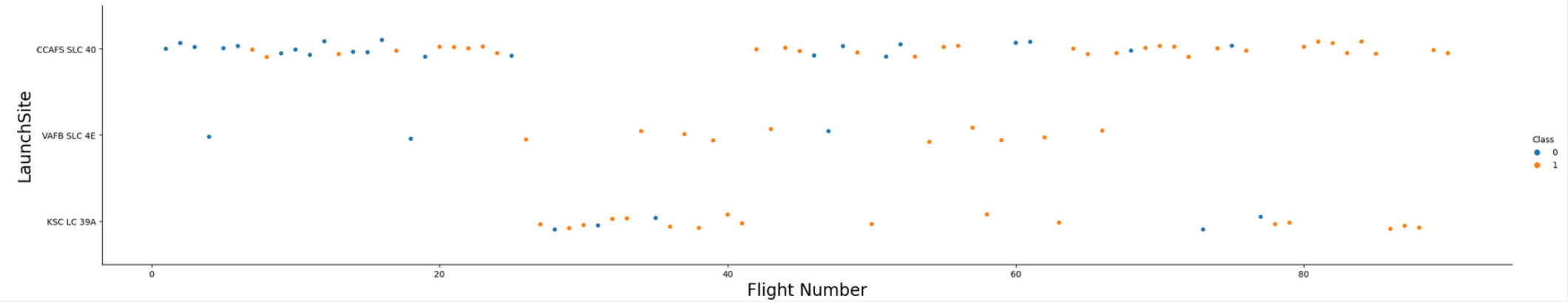
Results

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

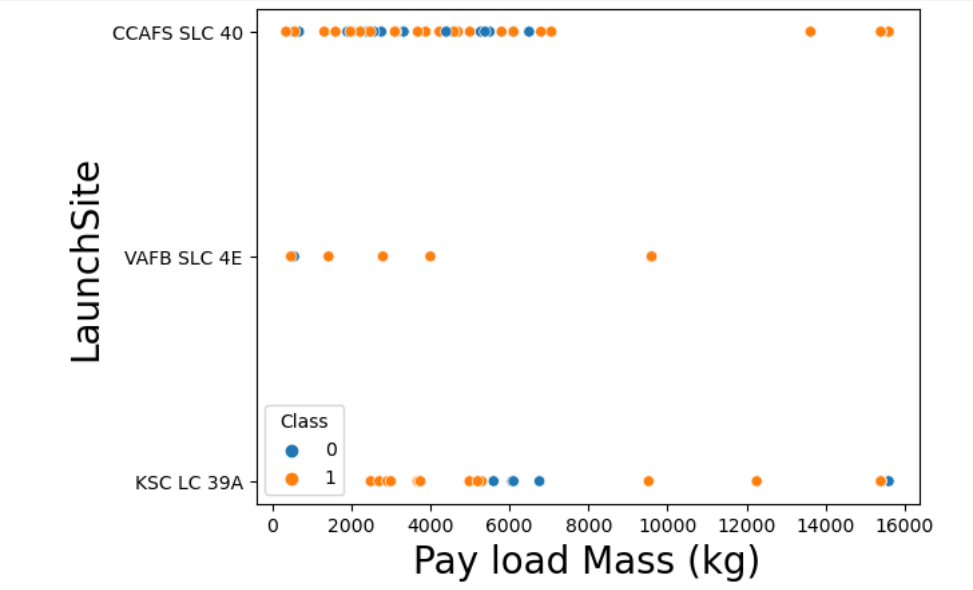

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- For CCAFS SLC 40 the pattern suggests that the likelihood of a successful landing has increased over time or as experience with more flights has been gained. This trend might indicate an improvement in technology, operational procedures, or experience gained with each subsequent flight.



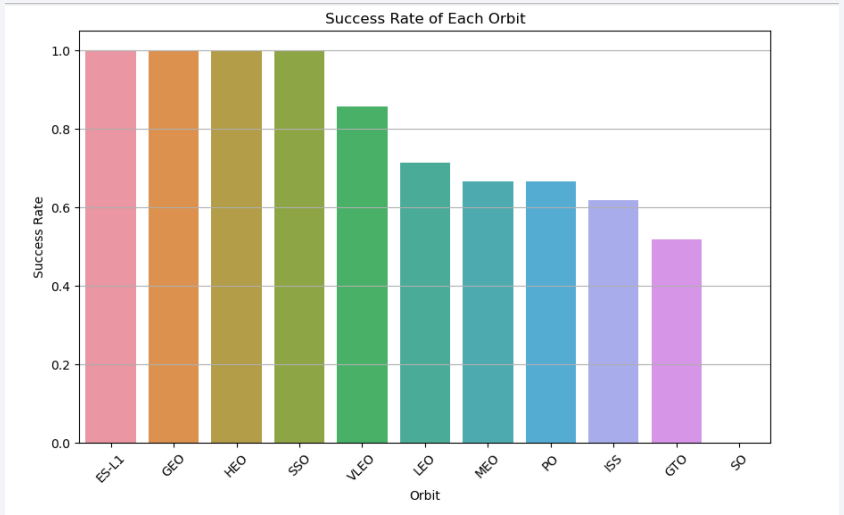
Payload vs. Launch Site



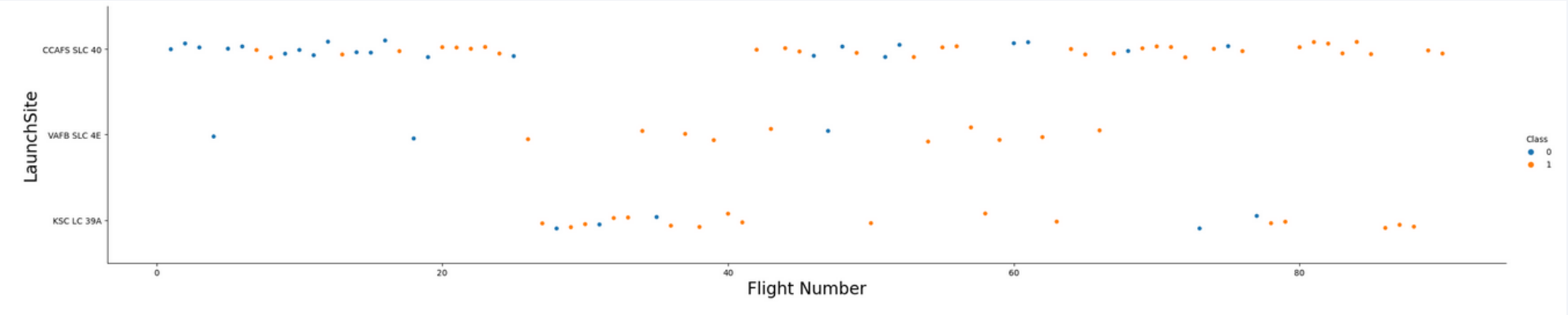
- From the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000). The most succesful were launches form VAFB-SLC launchsite where payload mass wasn't greater that 10000.

Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



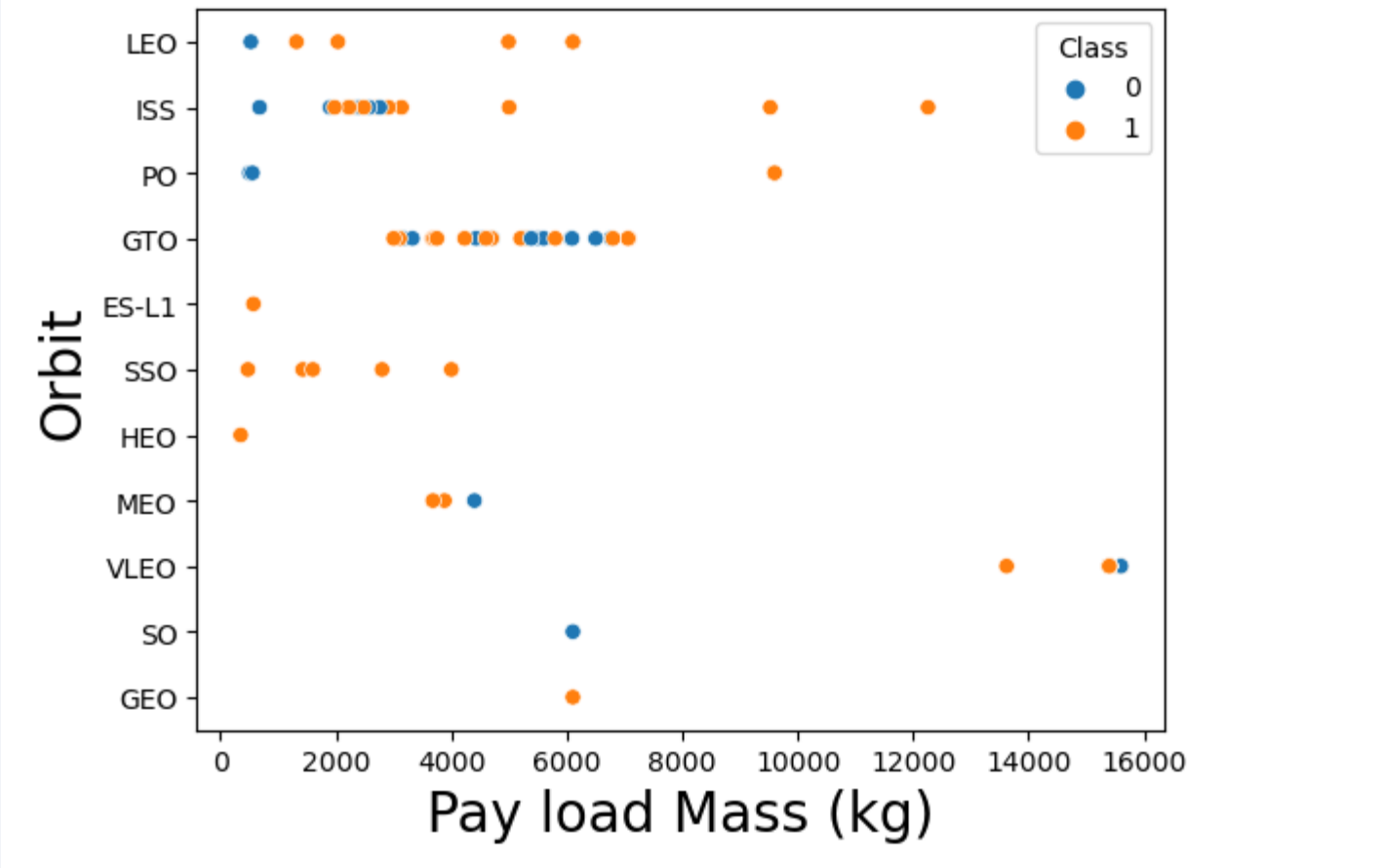
Flight Number vs. Orbit Type



- The plot doesn’t show any relations between flight number and orbit type

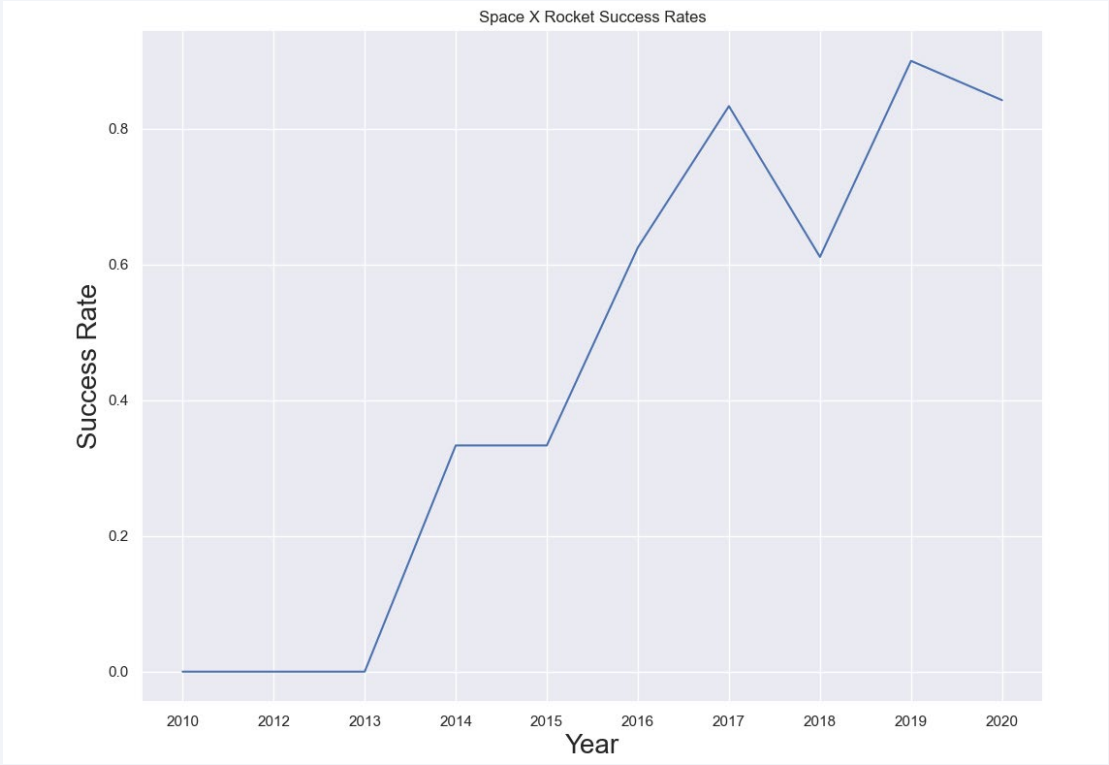
Payload vs. Orbit Type

- With heavy payload the most successful site in ISS, then LEO and Polar.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- Key word DISTINCT used to show only unique launch sites from the SpaceX data.

```
[10]: %sql select distinct Launch_Site from SPACEXTBL
 * sqlite:///my_data1.db
Done.
```

[10]:

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

Launch Site Names Begin with 'CCA'

- The query below used to display 5 records where launch sites begin with `CCA`

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

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

[12]:

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

- The total payload carried by boosters from NASA is 45596. It was calculated using the query below

```
%sql SELECT SUM (PAYLOAD_MASS__KG_) AS Payload_Mass  FROM SPACEXTBL WHERE `Customer`= 'NASA (CRS)';
```

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

Payload_Mass
45596

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1 is 2928.4

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS Avg_Payload_F9  FROM SPACEXTBL WHERE `Booster_Version`= 'F9 v1.1';
 * sqlite:///my_data1.db
Done.
```

Avg_Payload_F9
2928.4

First Successful Ground Landing Date

- Using the query below we calculated when the first successful landing outcome in ground pad was achieved.

```
%sql SELECT MIN(Date) AS First_Success  FROM SPACEXTBL WHERE `Mission_Outcome`= 'Success';
```

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

First_Success
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

- 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 SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND 4000 < PAYLOAD_MASS__KG_ < 6000;
```

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

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes

```
%sql SELECT Mission_Outcome, COUNT(*) AS Total FROM SPACEXTBL GROUP BY Mission_Outcome;
```

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

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

Boosters Carried Maximum Payload

- The names of the booster versions which have carried the maximum payload mass

```
%%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL);
```

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

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

2015 Launch Records

- This records displays the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015

```
%sql SELECT CASE substr(Date, 6, 2)WHEN '01' THEN 'January'WHEN '02' THEN 'February'WHEN '03' THEN 'March'WHEN '04' THEN 'April'WHEN '05' THEN 'May'WHEN '06' THEN 'June'\
```

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

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

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

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

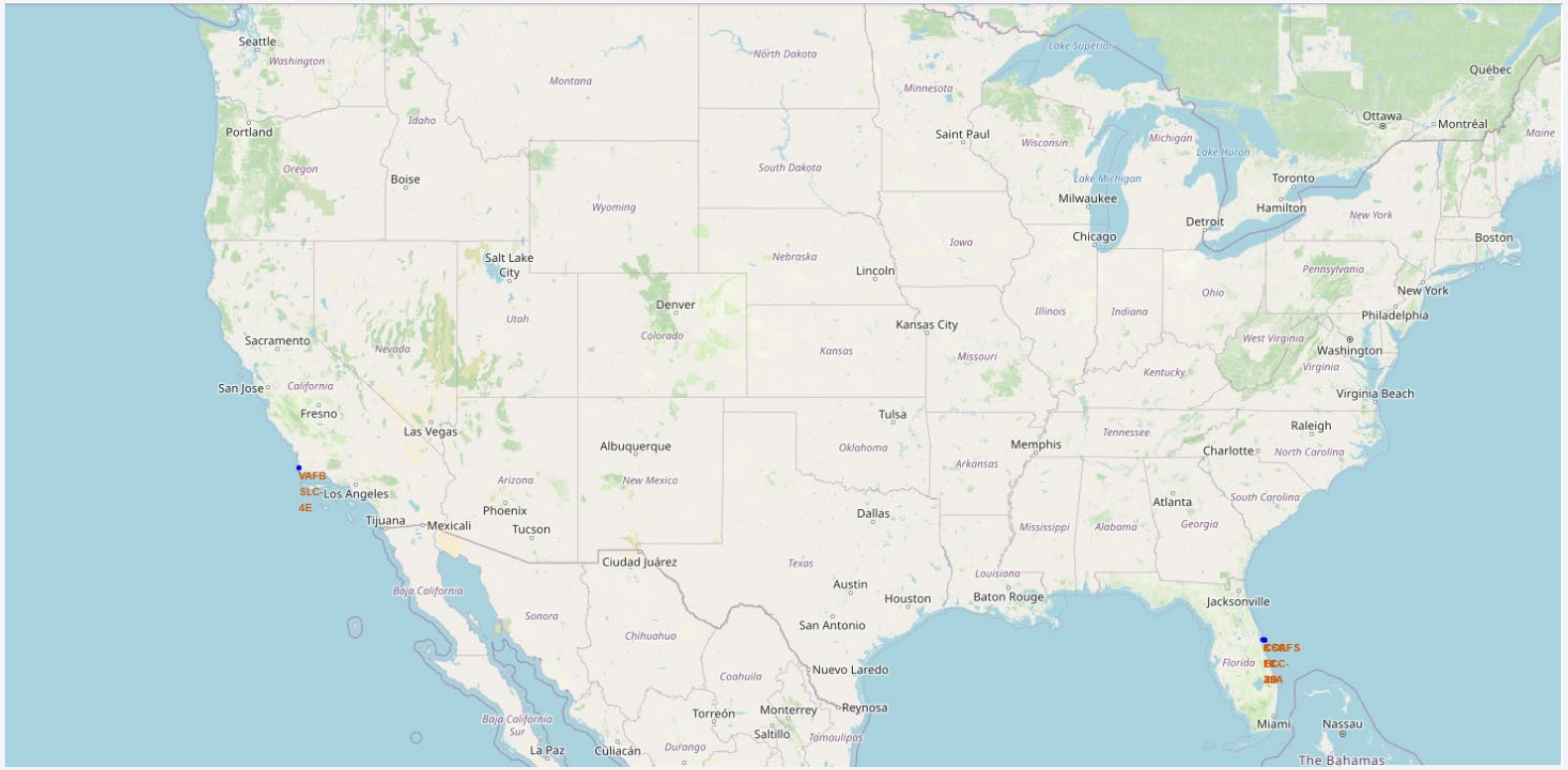
```
 * sqlite:///my_data1.db
Done.
```

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

Section 3

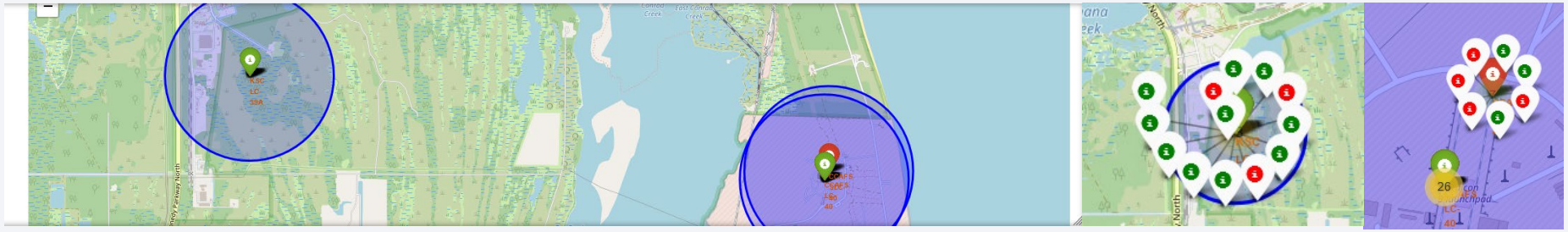
Launch Sites Proximities Analysis

All launch sites global map markers



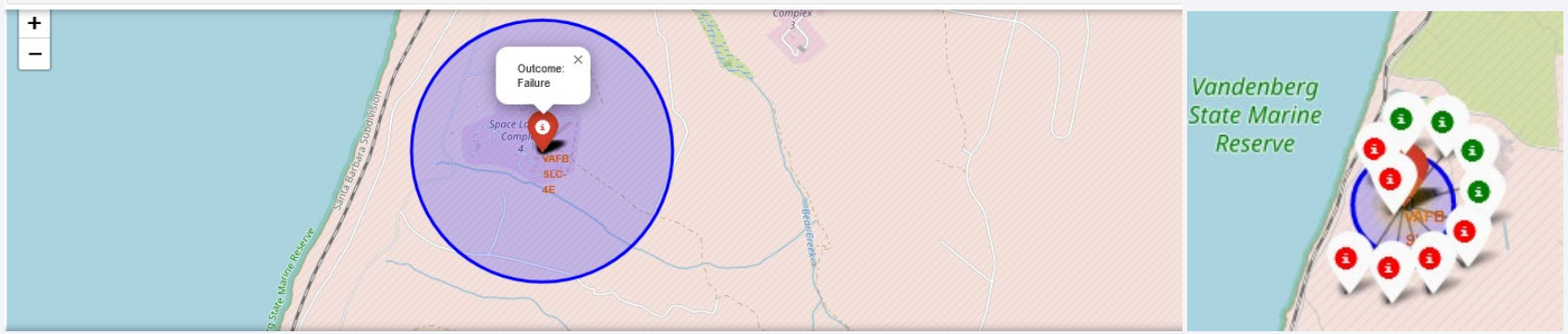
- The map showing SpaceX launch sites with a popup label showing its name.

Markers showing launch sites with color labels



Florida Launch Sites

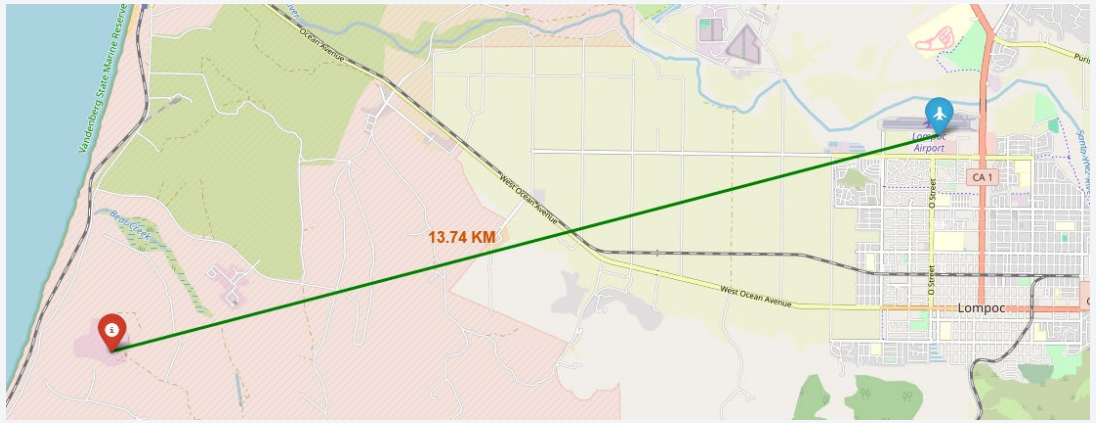
Green Markers shows successful landing
Red Markers shows failed landing

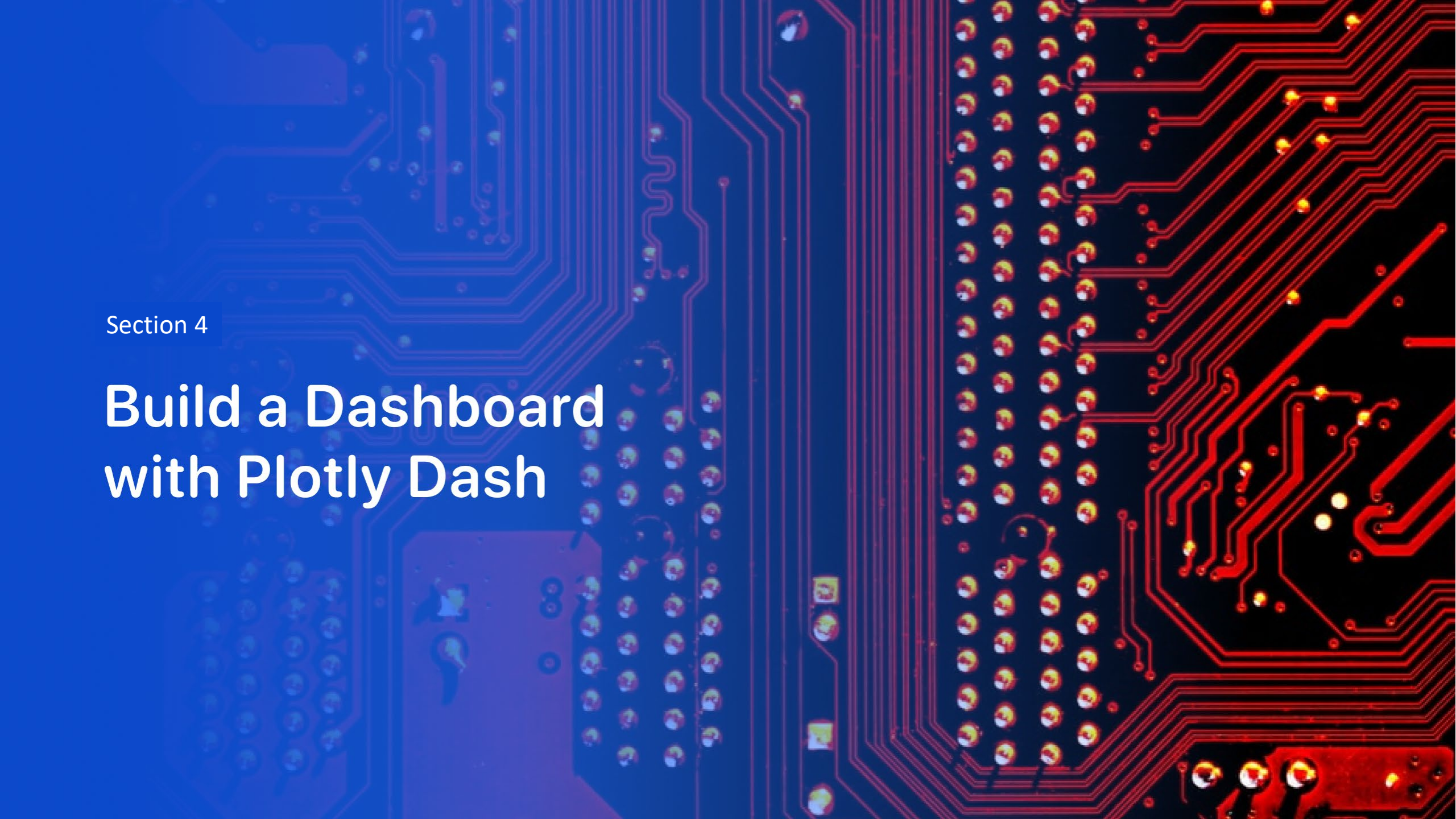


California Launch Sites

Launch Site distance to landmarks

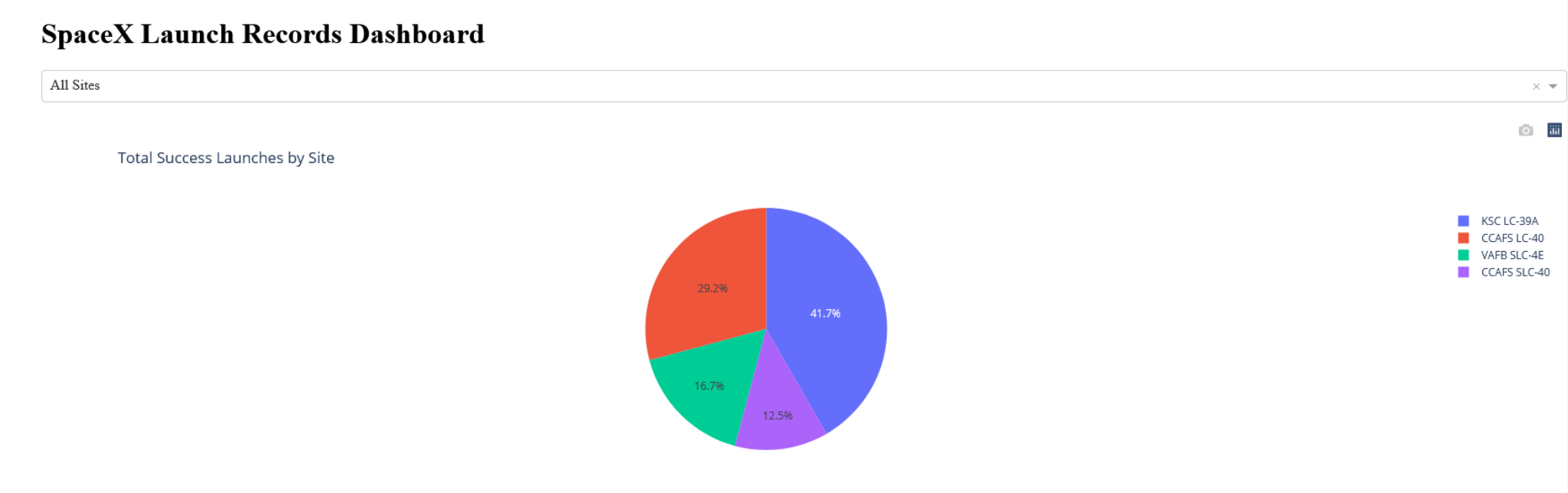
These graphs shows the distance to the closest airports. This knowledge crucial for the launch sites



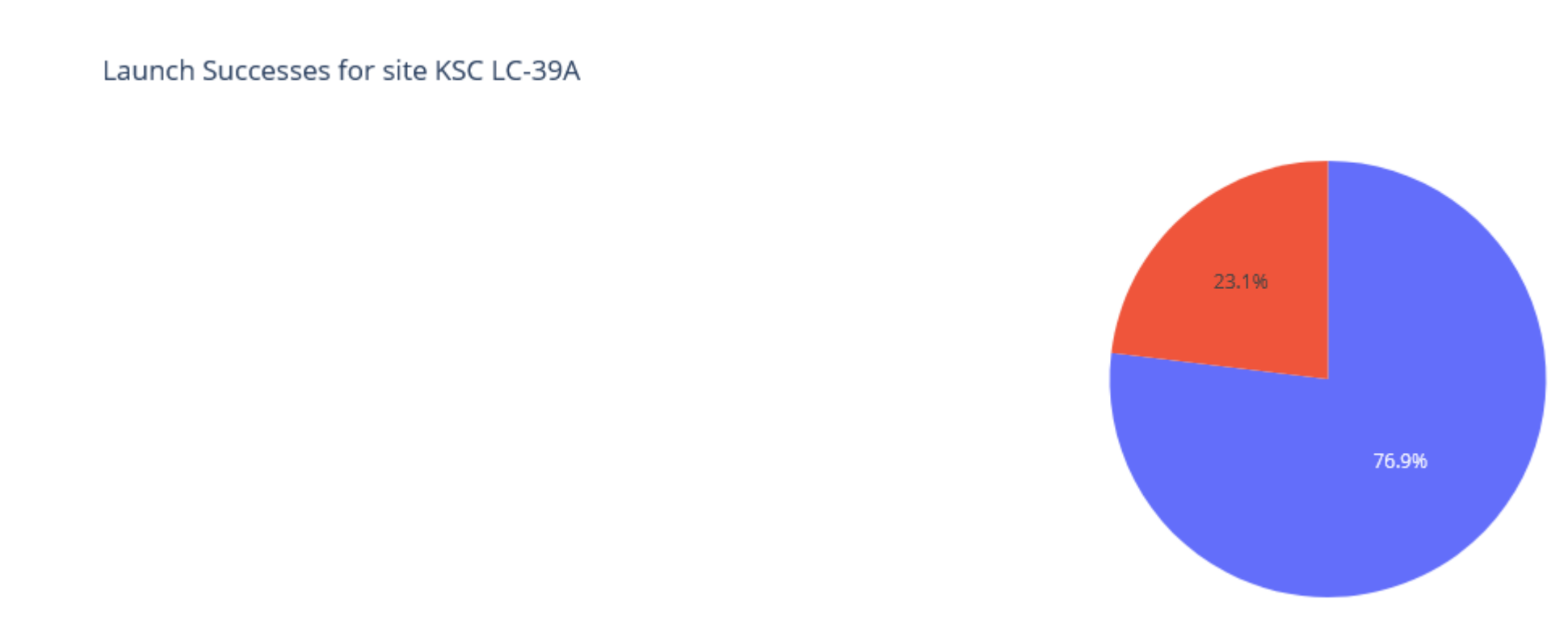

Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site

This chart shows that KSC LC-39A had the most successful launches and CCAPS SLC-40 the least successful.



Pie chart showing the Launch site with the highest launch success ratio



Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

High weight payload

Low weight payload

Section 5

Predictive Analysis (Classification)

Classification Accuracy

The decision tree classifier is the model with the highest classification accuracy

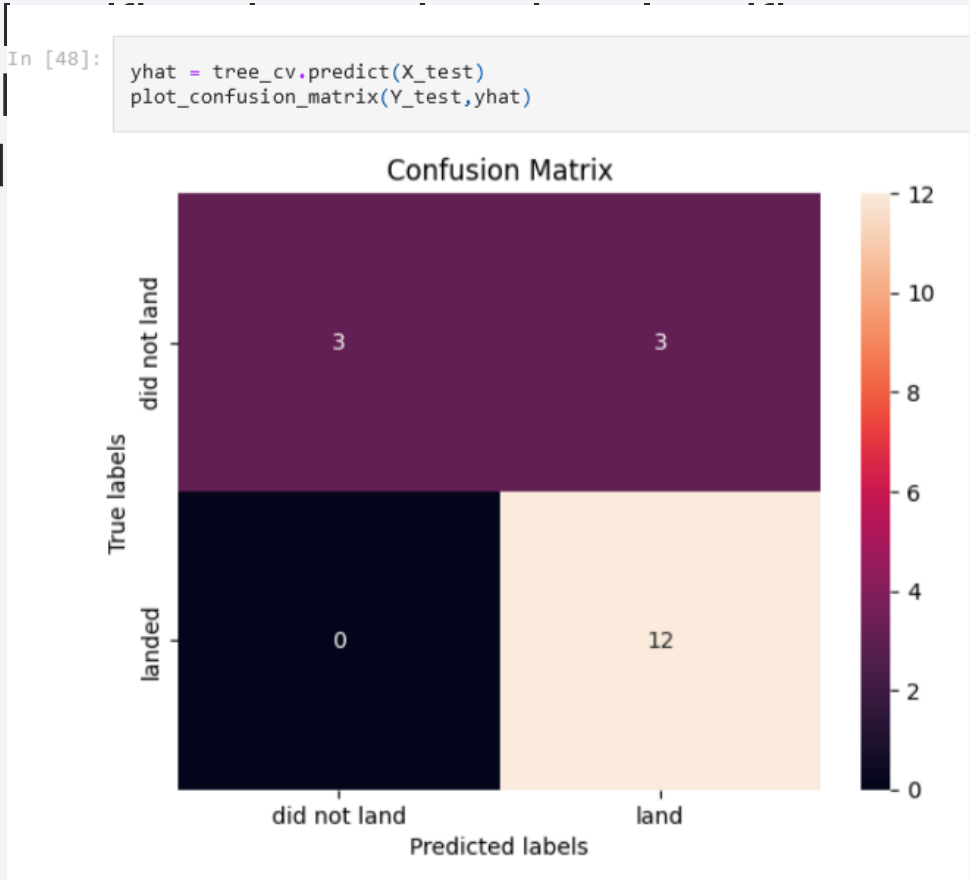
In [56]:
```
best_method = max(accuracy_tree, accuracy_svm, accuracy_knn)
if best_method == accuracy_tree:
    print("Decision Tree performs the best.")
elif best_method == accuracy_svm:
    print("SVM performs the best.")
else:
    print("K Nearest Neighbors performs the best.")
```

```
Decision Tree performs the best.
```

Confusion Matrix

The confusion matrix for the decision tree c
distinguish between the different classes. T
positives .i.e., unsuccessful landing marked
classifier.

```
In [48]:
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!