IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jian Yu
02.08.2023

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

  Space X 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 Space X 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 space X for a rocket launch.

  In this capstone, we will predict if the Falcon9 first stage will land successfully.

- Problems you want to find answers

  - The factors determine whether a rocket can successfully land.
  - The interaction amongst features that determine the success rate of a successful landing.
  - What operating conditions are requited to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection
- 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 Collection

- Make a get request to the SpaceX API;

response = resquests.get (url)

response.json()

- Web scraping Falcon9 launch records with BeautifulSoup
- Clean the requested data

Wrangling data using an API

Sampling data

Dealing with Nulls

Data Collection – SpaceX API

- We made a get request to the SpaceX API to collect data and did some basic data wrangling and formatting.

- The GitHub URL of the completed SpaceX API calls notebook is:

https://github.com/meeteryu/Applied-data-science-capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
[6]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
[7]: response = requests.get(spacex_url)
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
[11]: # Use json_normalize meethod to convert the json result into a dataframe

import pandas as pd
from pandas.io.json import json_normalize

json_result = response.json()
-------
# Convert JSON result to DataFrame
df = pd.json_normalize(json_result)
# Print the resulting DataFrame
print(df)
```

Task 3: Dealing with Missing Values

Calculate below the mean for the `PayloadMass` using the `.mean()`. Then use the mean and the `.replace()` function to replace `np.nan` values in the data with the mean you calculated.

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

# Replace np.nan values with the calculated mean
data_falcon9['PayloadMass'].replace(np.nan, mean_payload_mass, inplace=True)

# Print the modified dataframe
print(data_falcon9)
```

Data Collection - Scraping

- Web scrap Falcon 9 launch records with BeautifulSoup; Parse the table and convert it into a pandas dataframe

- The GitHub URL of the completed web scraping notebook is:

https://github.com/meeteryu/Applied-data-science-capstone/blob/main/jupyter-labs-webscraping.ipynb

EDA with Data Visualization

- Explore the data to see how the FlightNumber, Payload variables and launch site, success rate of each orbit type, Flight Number and obit would affect the launch outcome.
- The GitHub URL of completed EDA with data visualization notebook is:

https://github.com/meeteryu/Applied-data-science-capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Load the dataset into the corresponding table in a Db2 database.
- Execute SQL queries to answer the following questions:

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

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

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

  List the total number of successful and failure mission outcomes

  List the failed landing outcomes in drone ship, their booster version and launch sites.

  - The GitHub URL of completed EDA with SQL notebook is

  https://github.com/meeteryu/Applied-data-science-capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Mark all launch sites on the folium map;
- Mark the success/failed launches for each site on the map;
- Calculate the distances between a launch site to its proximities. Answer the questions:

  Are launch sites in close proximities to railways, highways and coastlines?

  Do launch sites keep certain distance away from cities?

- The GitHub URL of completed interactive map with Folium map notebook is

https://github.com/meeteryu/Applied-data-science-capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Add a Launch Site Drop-down Input Component;

  Add a callback function to render success-pie-chart based on selected site dropdown;

  Add a Range Slider to Select Payload;

  Add a callback function to render the success_payload_scatter_chart scatter plot;

- Plot a pie chart to show the total successful launches count for all sites;

  Plot a scatter chart to show the correlation between payload and launch success (for different booster version).

- The GitHub URL of completed Plotly Dash notebook is

https://github.com/meeteryu/Applied-data-science-capstone/blob/main/spacex_dash_app.py

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

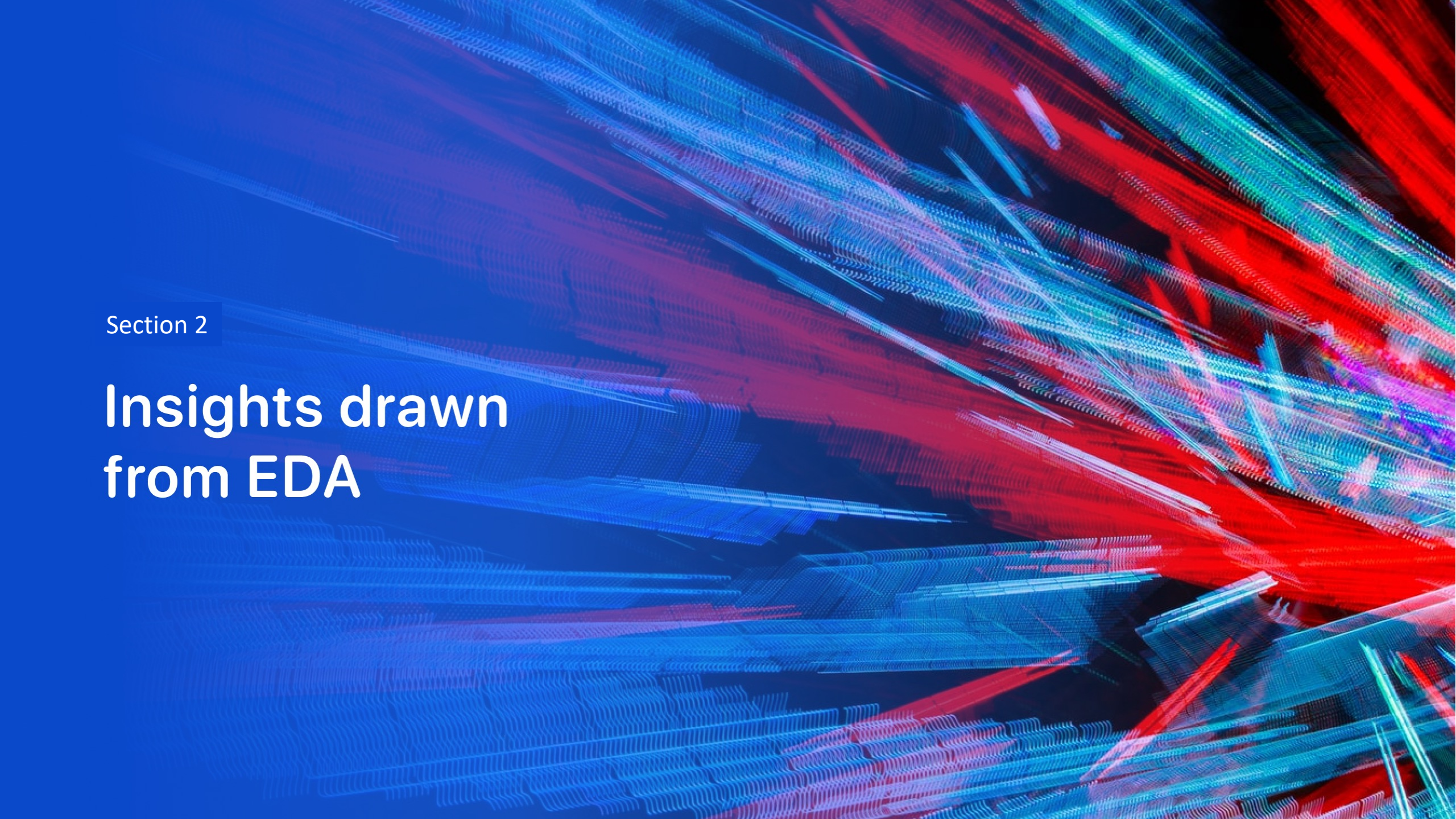
  find the method performs best using test data.

- The GitHub URL of completed predictive analysis notebook is

https://github.com/meeteryu/Applied-data-science-capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

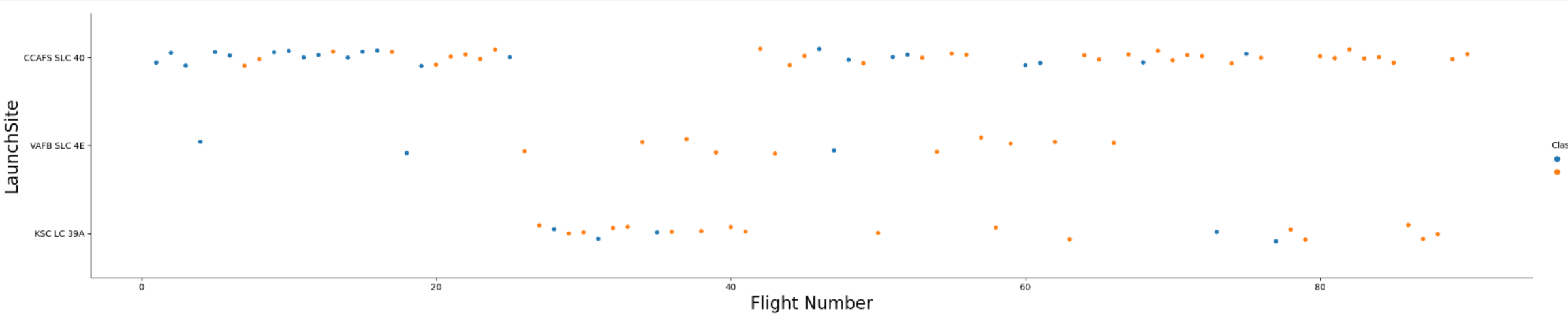
Results

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


Section 2
Insights drawn from EDA

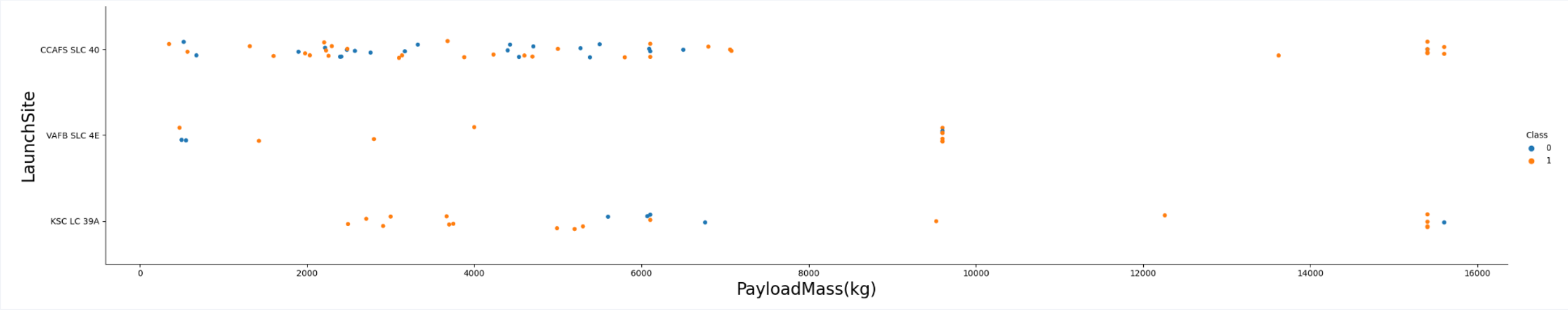
Flight Number vs. Launch Site

- As shown in the plot, the success of launches exhibited improvement over flight number.



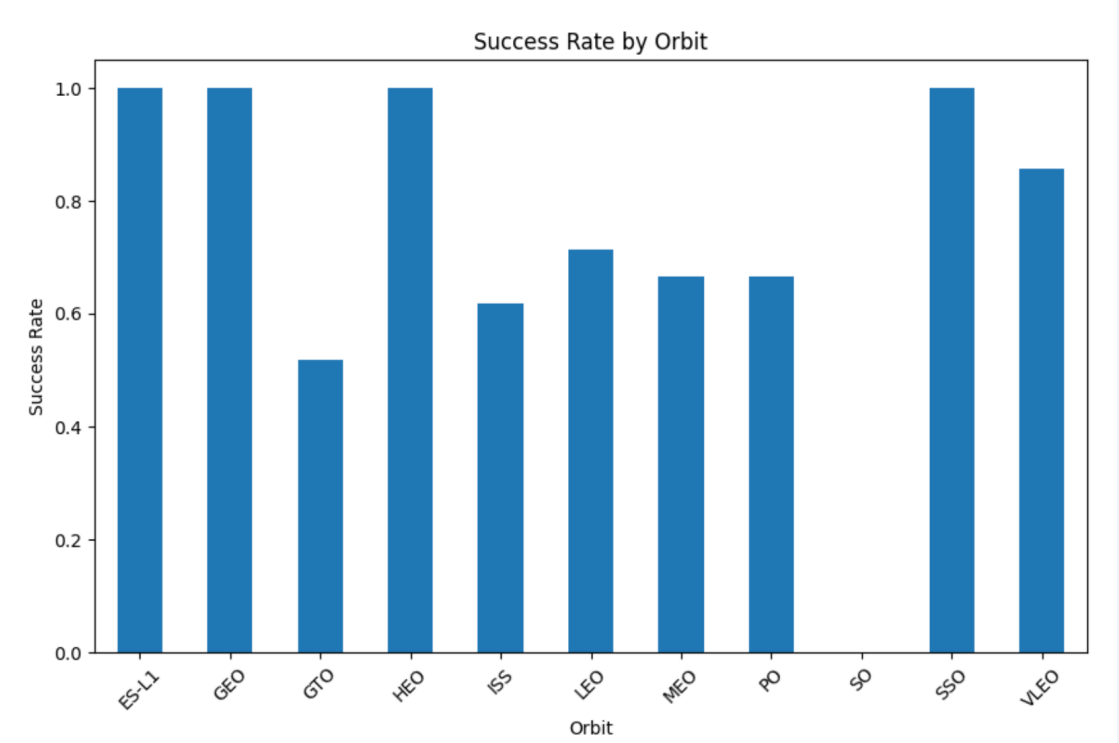
Payload vs. Launch Site

- For VAFB SLC-4E launch site, there are no rockets launched for heavy payload mass (<10000 kg)
- The greater payload mass, the higher success rate at CCAFS SLC-40 launch site.



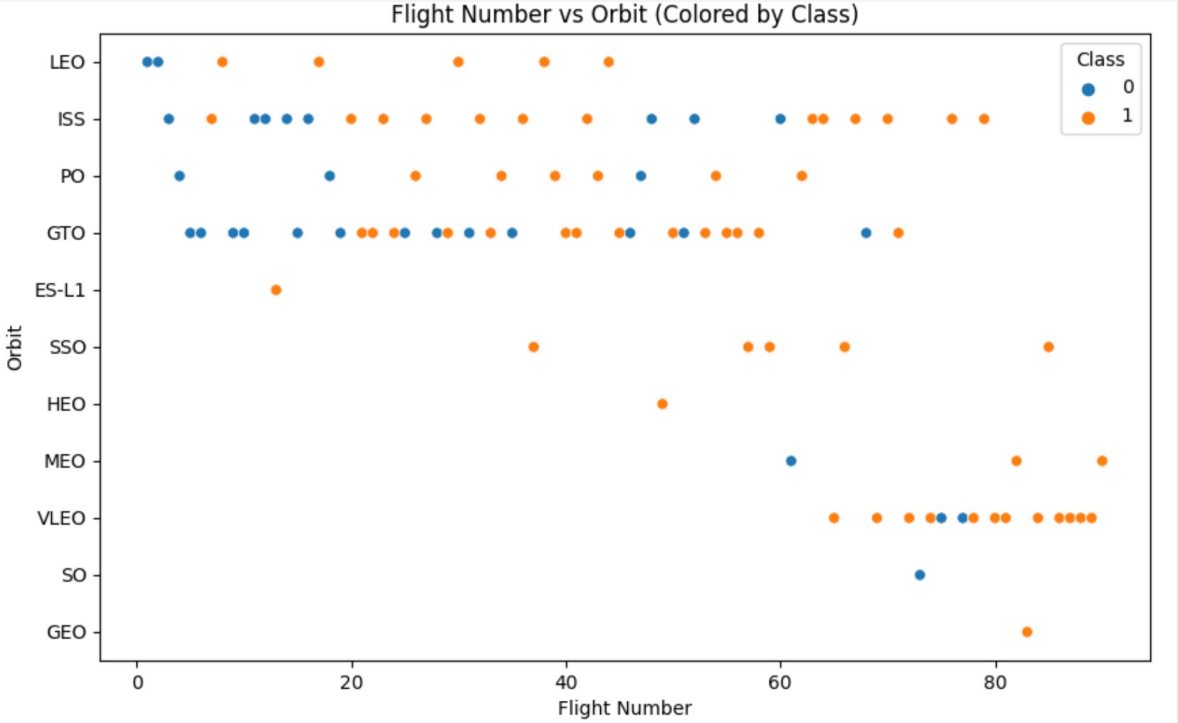
Success Rate vs. Orbit Type

- Orbits ES-L1, GEO, HEO, and SSO have high success rate (100%)



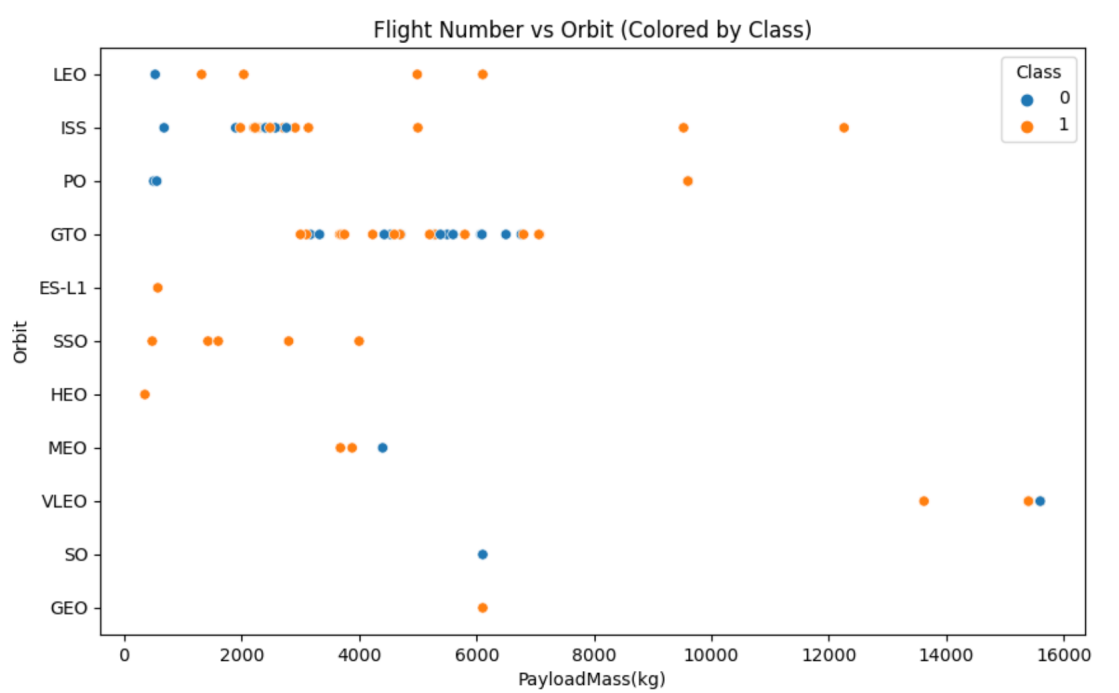
Flight Number vs. Orbit Type

- In the LEO orbit, the success appears related to the number of flight number
- There seems to be no relationship between flight number in the GTO orbit



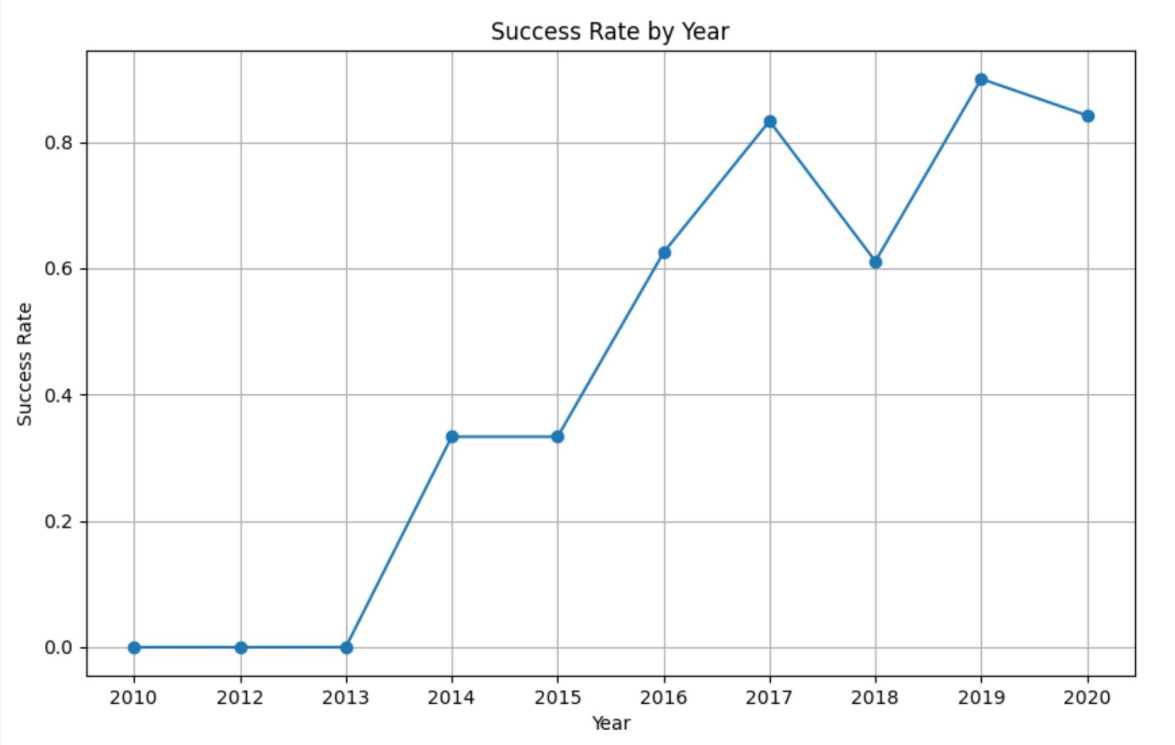
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Orbit LEO, PO and ISS
- As for GTO, it is hard to distinguish the trend



Launch Success Yearly Trend

- The success rate since 2013 keep increasing till 2020



All Launch Site Names

- Use DISTINCT( ) to find unique launch sites from the SPACEXTBL.

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

```
[31]: %sql SELECT DISTINCT(launch_site) FROM SPACEXTBL;
```

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

[31]:

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

Launch Site Names Begin with 'CCA'

- Use the following query to display 5 records where launch sites begin with `CCA`

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

[42]:
```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

[42]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_C
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (pa
2010-08-12	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 (pa
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No

Total Payload Mass

- Use the following query to calculate the total payload carried by boosters from NASA(CRS)
- The result is 45596

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

```
[47]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS Total_PayloadMass FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)';
 * sqlite:///my_data1.db
Done.
```

[47]:

Total_PayloadMass
45596

Average Payload Mass by F9 v1.1

- Use the following query to calculate the average payload mass carried by booster version F9 v1.1
- The result is 2928.4

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

```
[51]: %sql SELECT AVG(PAYLOAD_MASS__KG_) AS Avg_PayloadMass FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';
```

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

[51]:

Avg_PayloadMass
2928.4

First Successful Ground Landing Date

- Use the following query to find the dates of the first successful landing outcome on ground pad
- The date is 2015-12-22

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

Hint:Use min function

```
[55]: %sql SELECT MIN(Date) AS FirstSuccessfull_landing_date FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success (ground
 * sqlite:///my_data1.db
Done.
```

[55]:

FirstSuccessfull_landing_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Use the WHERE clause to find the boosters which have successfully landed on drone ship and use AND condition to narrow down the successful landing with payload mass greater than 4000 but less than 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

```
[60]: %sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ >
```

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

[60]:

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

Total Number of Successful and Failure Mission Outcomes

- Use the following query to calculate the total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

```
[64]: %sql SELECT COUNT(Mission_Outcome) AS Mission_Outcome FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
 * sqlite:///my_data1.db
Done.
```

[64]:

Mission_Outcome
1
98
1
1

Boosters Carried Maximum Payload

- Use the following query to list the names of the booster which have carried the maximum payload mass
- use a subquery in the WHERE clause and the MAX() function to find the booster that carried the maximum payload

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

```
[77]: %sql SELECT Booster_Version AS Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__K
```

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

[77]:

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

- Use the following query, including the WHERE clause, LIKE, AND and BETWEEN conditions, to list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

[79]:
```
%sql SELECT Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Failure (drc
```

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

[79]:

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

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

- Use the WHERE clause to filter the landing outcomes between 2010-06-04 and 2017-03-20;
- Use the GROUP BY clause to group the landing outcomes;
- Use the ORDER BY to order result in descending order.

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.

```
[90]: %sql SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-2
```

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

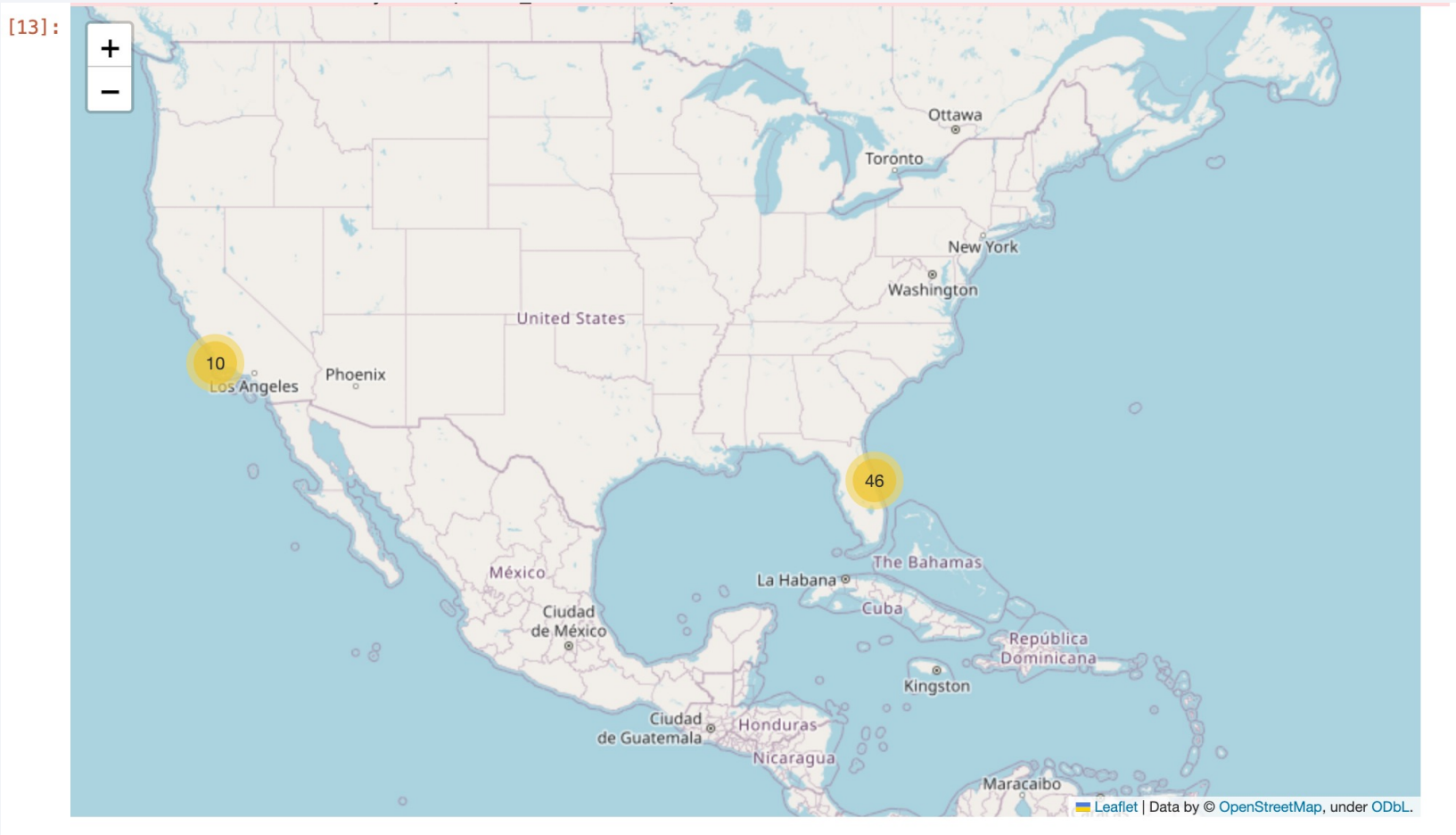
[90]:

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


Section 3
Launch Sites
Proximities Analysis

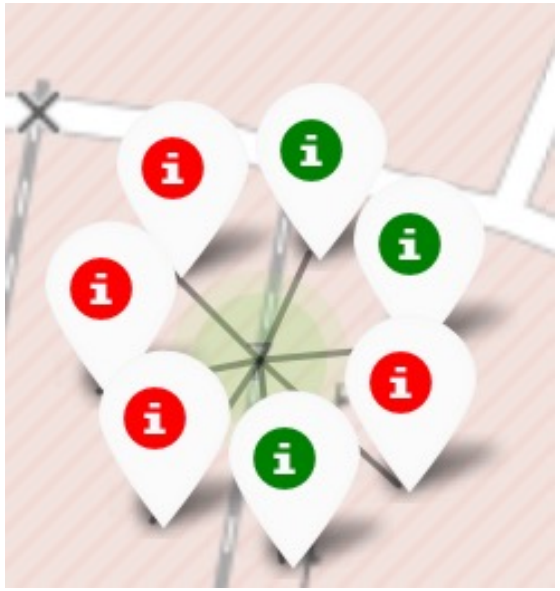
All launch sites marked on a global map

- All launch sites are located around the coasts Florida and California.

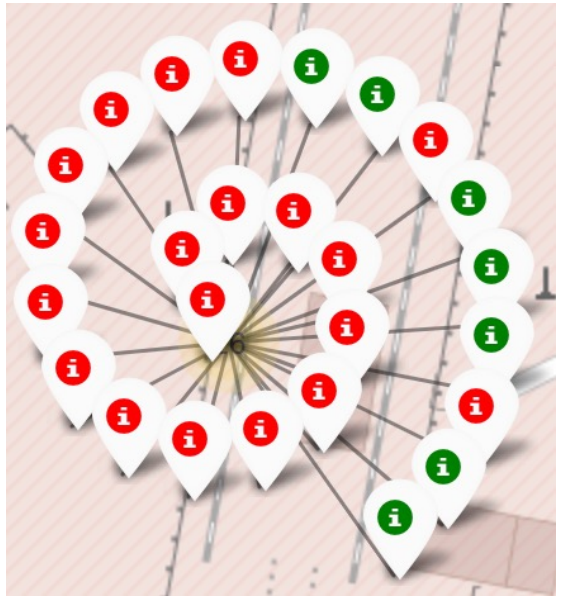


Launch sites with color-labeled markers

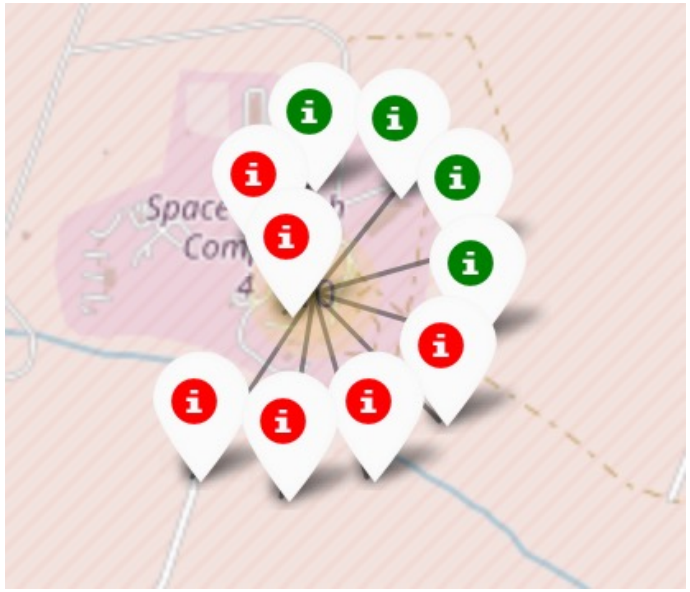
CCAFS SLC-40



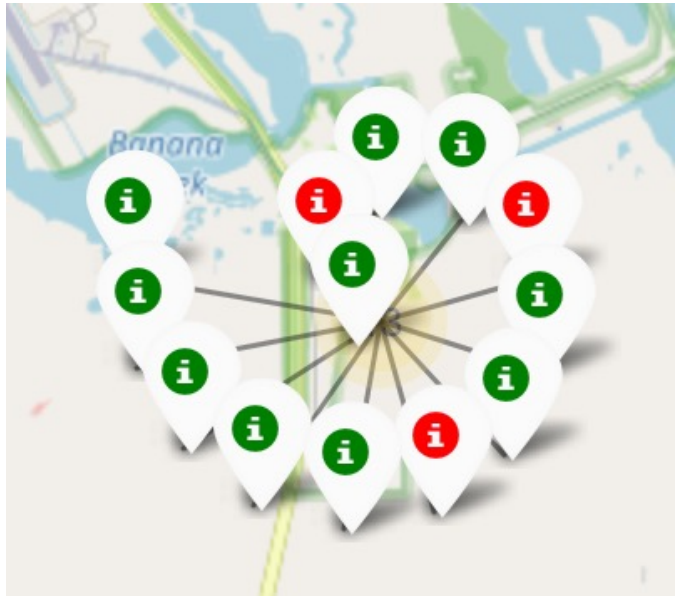
CCAFS LC-40



VAFB SLC-4E



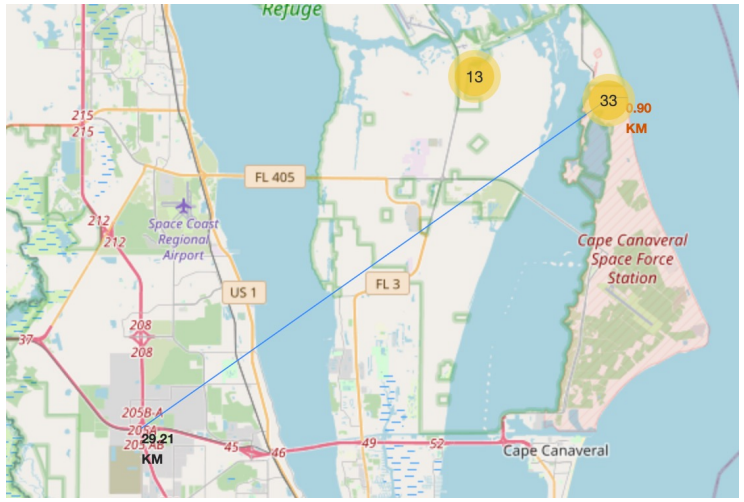
KSC LC-39A



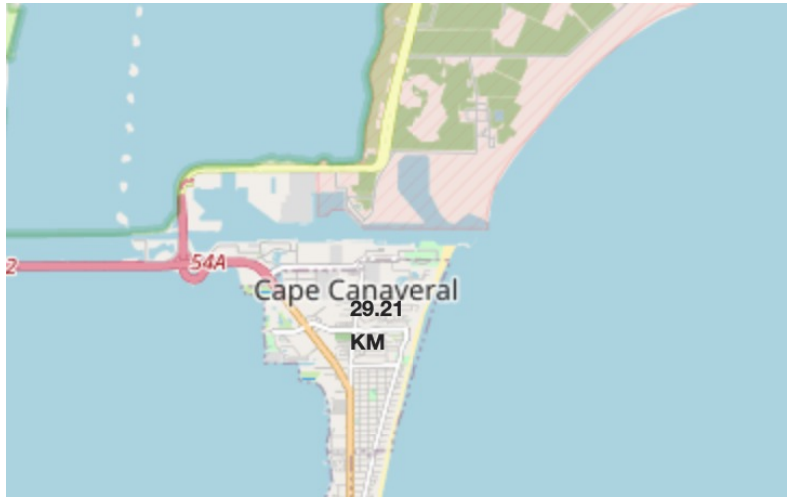
- Green markers indicate successful launch outcome;
- Red markers indicate failed launch outcome.

Launch Site distance to landmarks

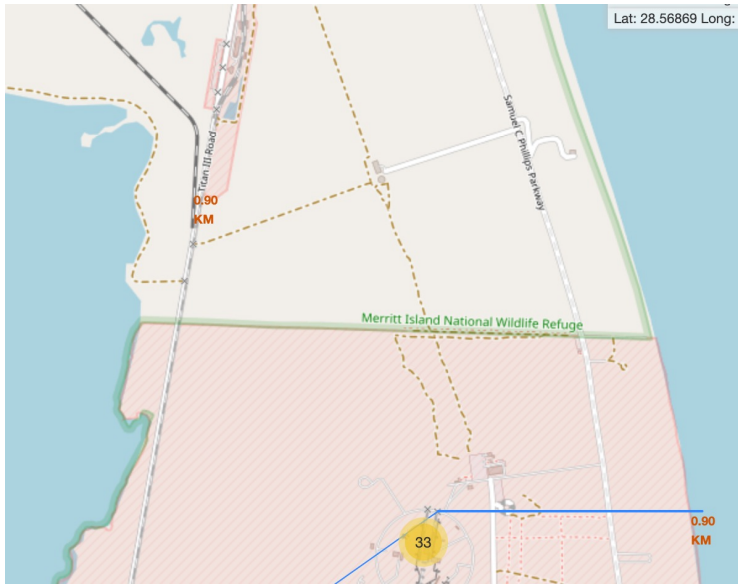
distance to coastline



distance to highway

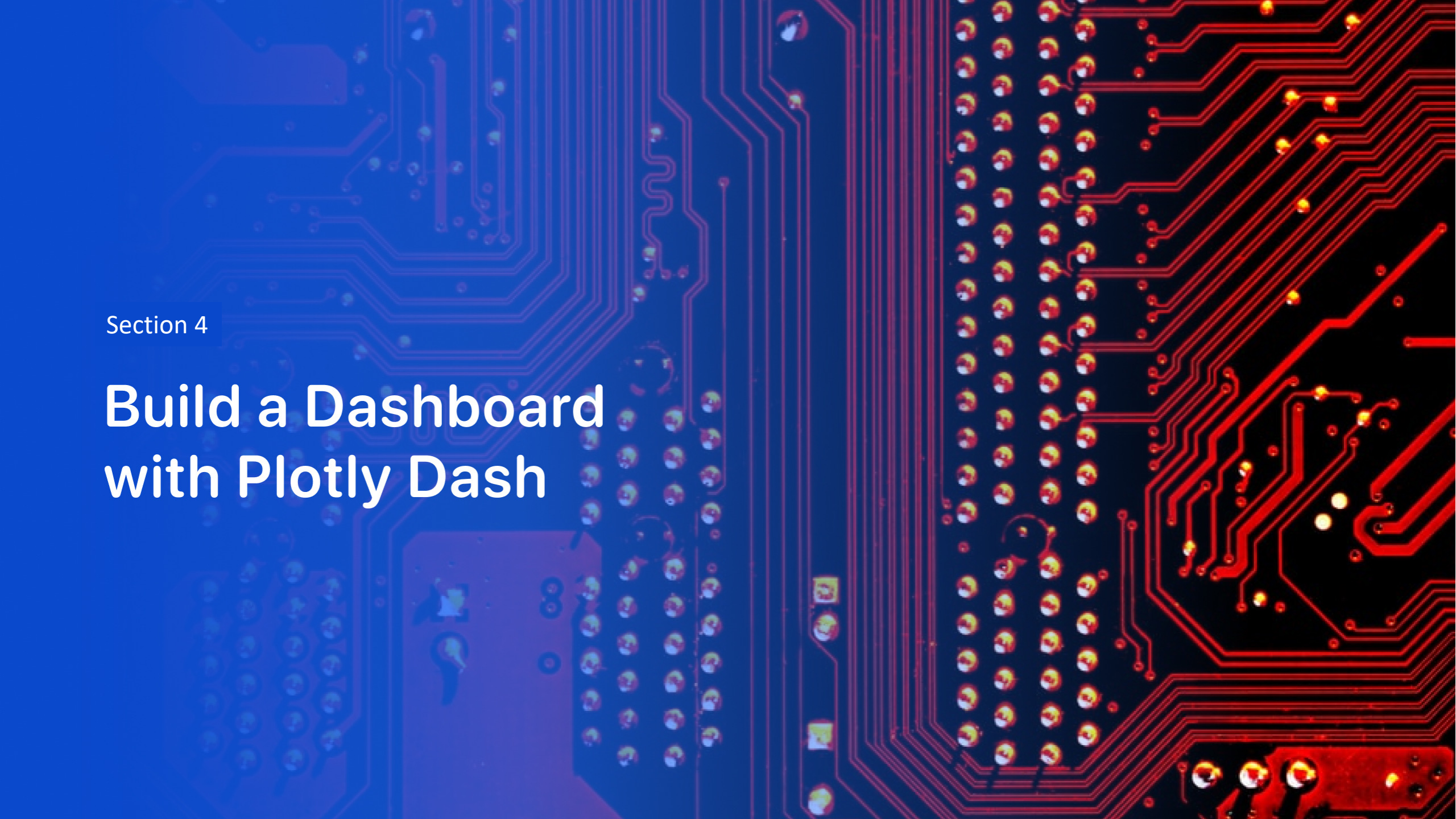


distance to City



distance to railway

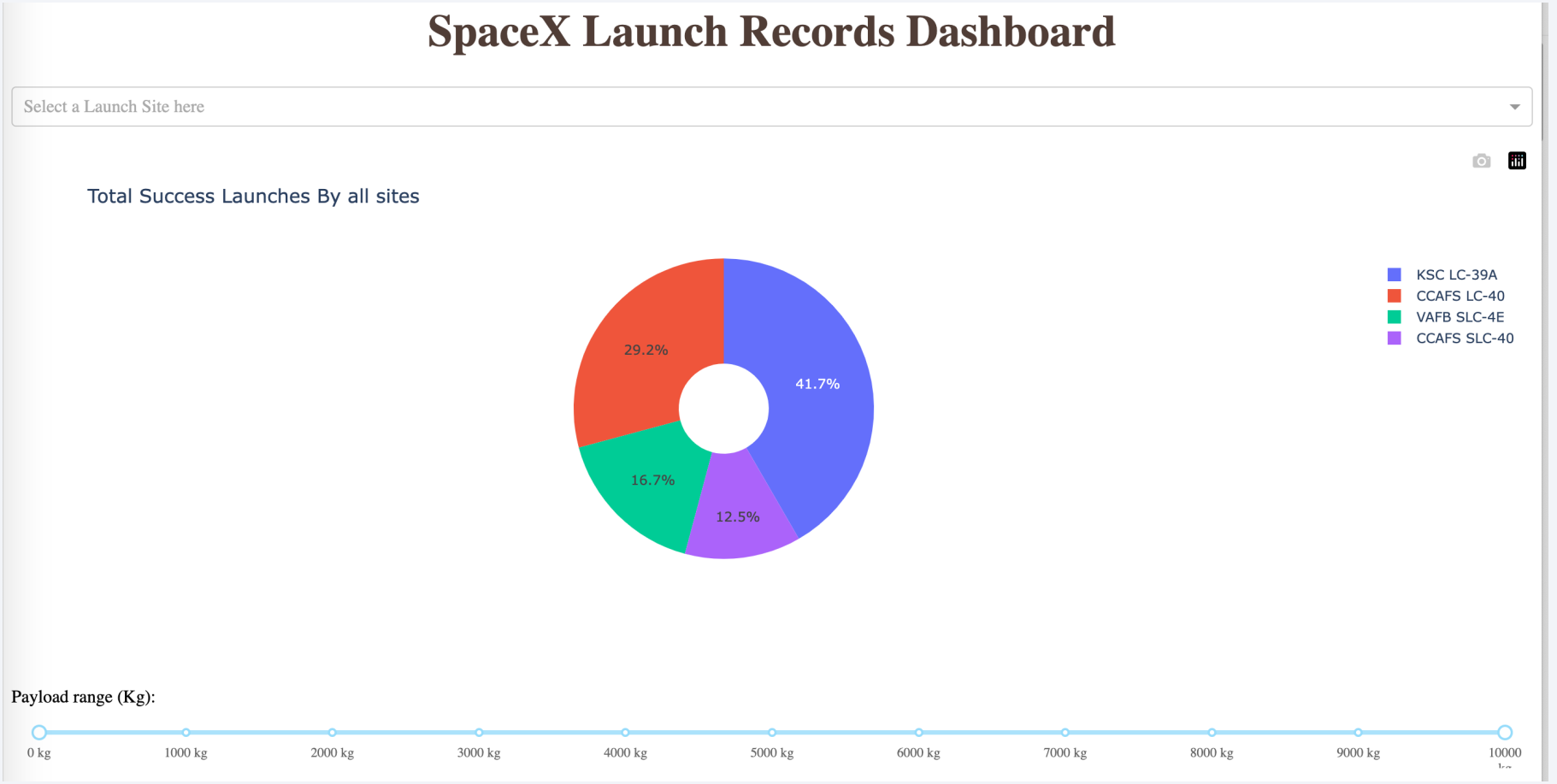
All launch sites are not close proximity to railways.
All launch sites are not close proximity to highways.
All launch sites are close proximity to coastline.
All launch sites keep certain distance away from cities.


Section 4
Build a Dashboard with Plotly Dash

Pie chart showing total success launches by all sites

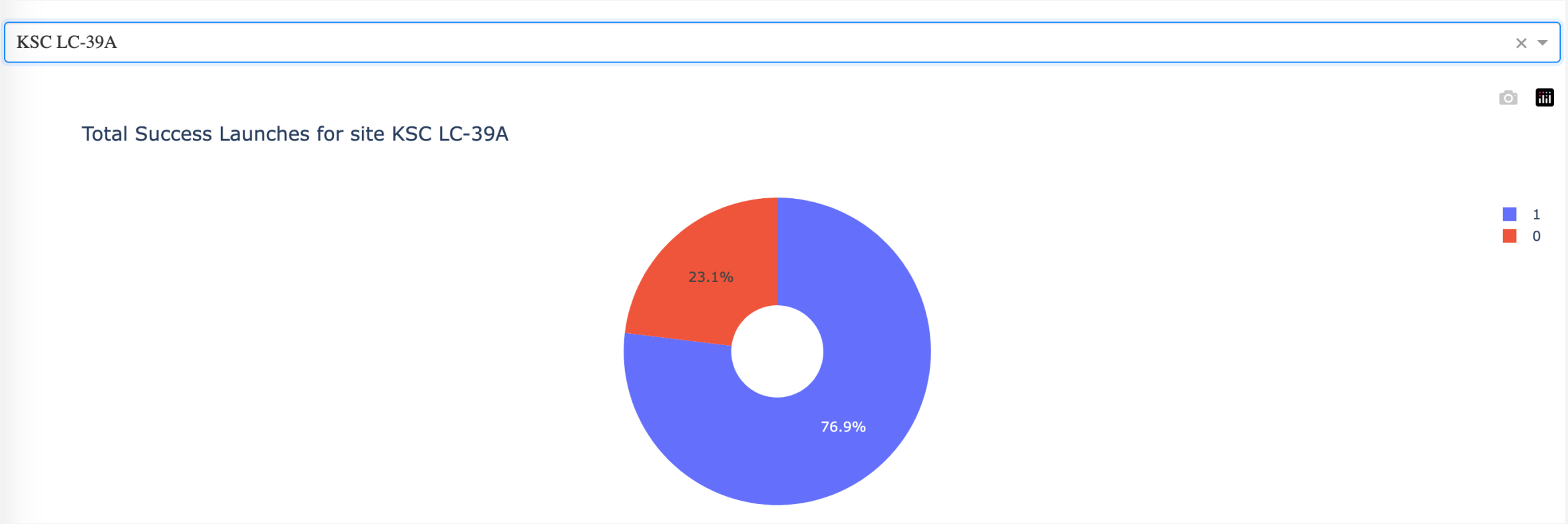
- Success rate:

KSC LC-39A: 41.7% (highest); CCAFS LC-40: 29.2%; VAFB SLC-4E: 16.7%; CCAFS SLC-40: 12.5%



Pie chart showing total success launches for site KSC LC-39A

- Success rate: 76.9%; Failure rate: 23.1%



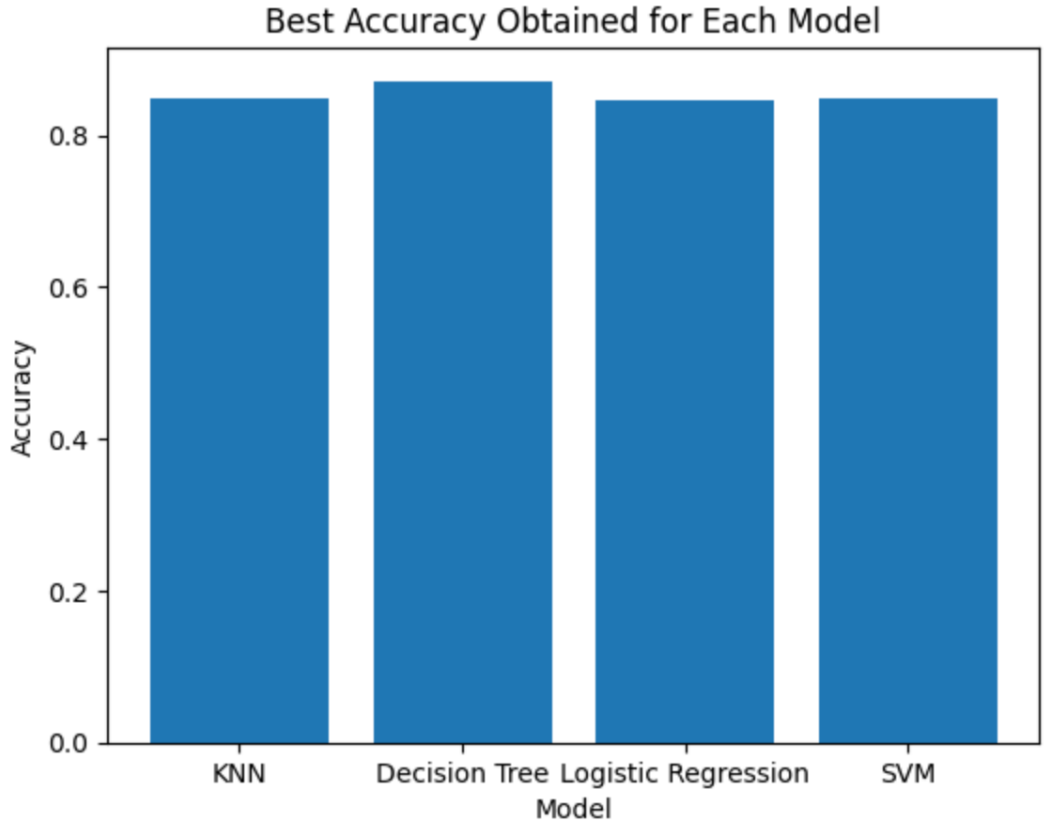
Payload vs. Launch Outcome scatter plots for all sites, with different payload selected in the range slider

- Payload in the range of 2k-7k kg and 9k-10k kg, the success rate is higher.

Section 5

Predictive Analysis (Classification)

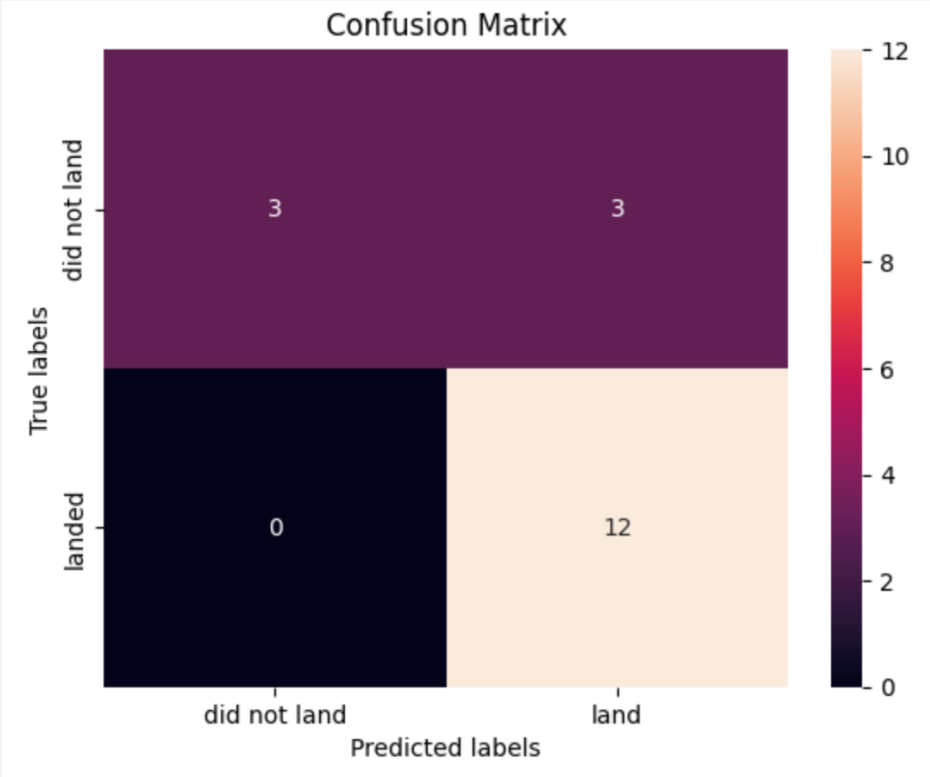
Classification Accuracy




```
Best model is DecisionTree with a score of 0.8714285714285713
Best params is : {'criterion': 'gini', 'max_depth': 6, 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitter': 'bes
t'}
```

Confusion Matrix

- The decision tree classifier's confusion matrix indicates its ability to differentiate between classes. However, a significant issue arises with false positives, where unsuccessful landings are incorrectly identified as successful landings by the classifier.



Conclusions

- The successful launch rate keep increasing after 2013
- Orbits ES-L1, GEO, HEO, and SSO have high success rate (100%).
- All launch sites are not close proximity to railways, highways.
- All launch sites are close proximity to coastline and keep certain distance away from cities.
- KSC LC-39A had the most successful launch outcome (41.7%).
- The best machine learning algorithm is the DecisionTree classifier.

Thank you!