IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anas ALWohaib
2024/06/18

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Interactive map with folium
  - Dashboard with Plotly Dash
  - Predictive analysis - Classification
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

- Project background and context

  The era of commercial space has arrived, and there are several companies that are making space travel affordable for everyone. Perhaps the most successful of them is SpaceX, and one of the reasons is that their rocket launch is relatively inexpensive.

  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, we will predict if the Falcon 9 first stage will land successfully. 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

- Problems you want to find answers

  Correlations between each rocket variables and successful landing rate

  Conditions to get the best results and ensure the best successful landing rate

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Using Web-Scraping and API
- Perform data wrangling
  - Convert outcomes into labels describing the status of landing
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Finding the best parameter for predictive models (SVM - Classifications models ...)

Data Collection

- Describe how data sets were collected.
  - Datasets were collected from web-Scraping through these processes
    - HTML response > Extract data using Beuatifoul soup > Normalize data to CSV file
  - **Also datasets collected from API through this process**
    - Request data from the API > Return JSON file > Normalize data to CSV file

Data Collection – SpaceX API

- Requesting Data using API

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

response = requests.get(spacex_url)
```

- To JSON File

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

- Applying Functions to clean data

```
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

```
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])
```

```
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])
```

GitHub Link

Data Collection - Scraping

- Requesting data from URL
- Get the data
- Using beautifulsoup

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

```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url).text
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data, 'html5lib')
```


Github Link

Data Wrangling

- Calculating the numbers of each column needed

```
# Apply value_counts() on column LaunchSite
LaunchSite = df['LaunchSite'].value_counts()
LaunchSite
```

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
# Apply value_counts on Orbit column
df.Orbit.value_counts()
```

- Create landing outcome labels depending on Outcome column

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]
landing_class
```

[Github Link]

EDA with Data Visualization

- • Scatter chart:
  - ○ Flight Number vs. Launch Site
  - ○ Payload vs. Launch Site
  - ○ Flight Number vs. Orbit Type
  - ○ Payload vs. Orbit Type
  - ○ A scatter plot shows how much one variable is affected by another. The relationship between two variables is called a correlation. This plot is generally composed of large data bodies.
- • Bar chart:
  - ○ Orbit Type vs. Success Rate
  - ○ A Bar chart makes it easy to compare datasets between multiple groups at a glance. One axis represents a category and the other axis represents a discrete value. The purpose of this chart is to indicate the relationship between the two axes.
- • Line chart:
  - ○ Year vs. Success Rate
  - ○ A Line chart shows data variables and trends very clearly and helps predict the results of data that has not yet been recorded.

GitHub Link

EDA with SQL

- Loading the dataset into the corresponding table in a Db2 database, and executing SQL queries to answer following questions:

  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string 'CCA'
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date when the first successful landing outcome in ground pad was achieved
  - Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster_versions which have carried the maximum payload mass
  - Listing the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - Ranking 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

[Github Link]

Build an Interactive Map with Folium

- Objects created and added to a folium map:
  - Markers that show all launch sites on a map
  - Markers that show the success/failed launches for each site on the map
  - Lines that show the distances between a launch site to its proximities
- • By adding these objects, following geographical patterns about launch sites are found:
  - Are launch sites in close proximity to railways? Yes
  - Are launch sites in close proximity to highways? Yes
  - Are launch sites in close proximity to coastline? Yes
  - Do launch sites keep certain distance away from cities? Yes

Github Link

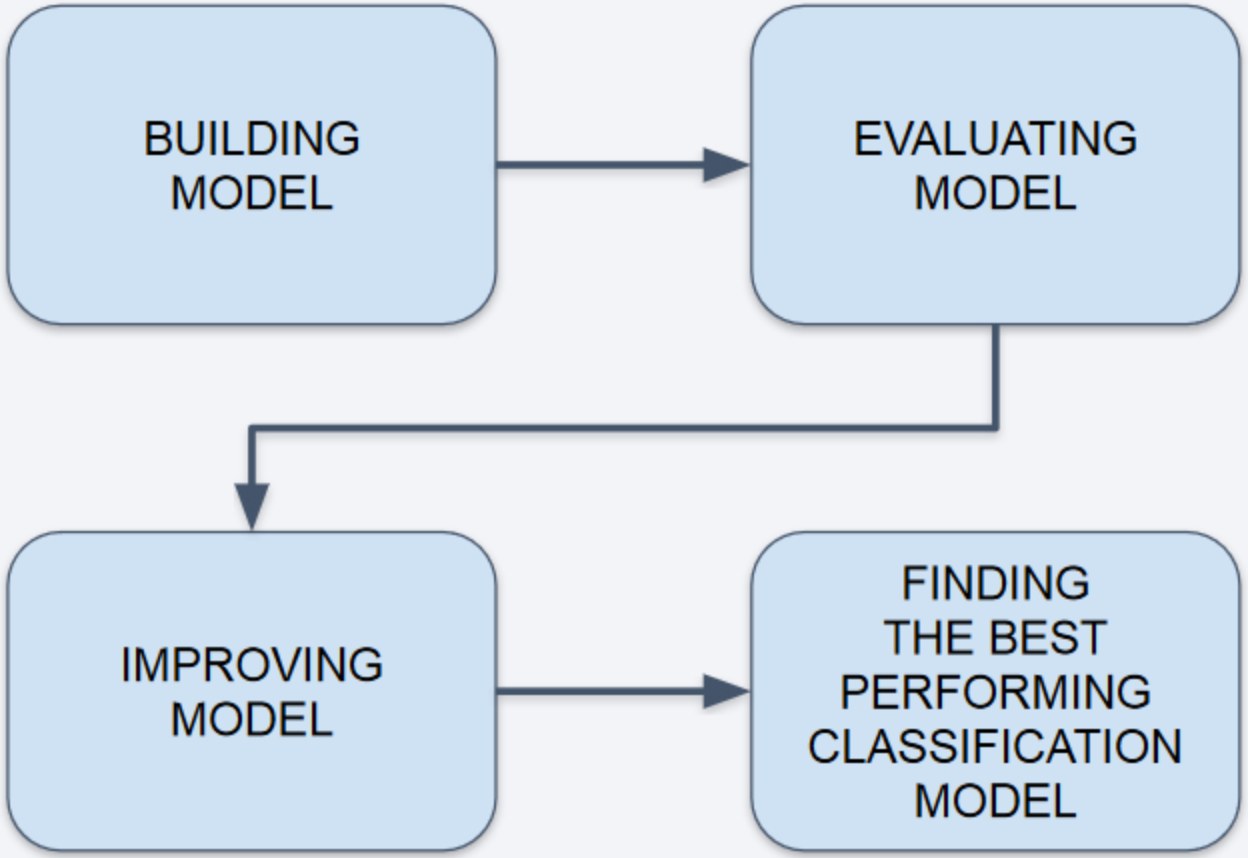
Build a Dashboard with Plotly Dash

- The dashboard application contains a pie chart and a scatter point chart.
  - Pie chart
    - For showing total success launches by sites
    - This chart can be selected to indicate a successful landing distribution across all launch sites or to indicate the success rate of individual launch sites.
  - Scatter chart
    - For showing the relationship between Outcomes and Payload mass (Kg) by different boosters
    - Has 2 inputs: All sites/individual site & Payload mass on a slider between 0 and 10000 kg
- This chart helps determine how success depends on the launch point, payload mass, and booster version categories.

Github Link

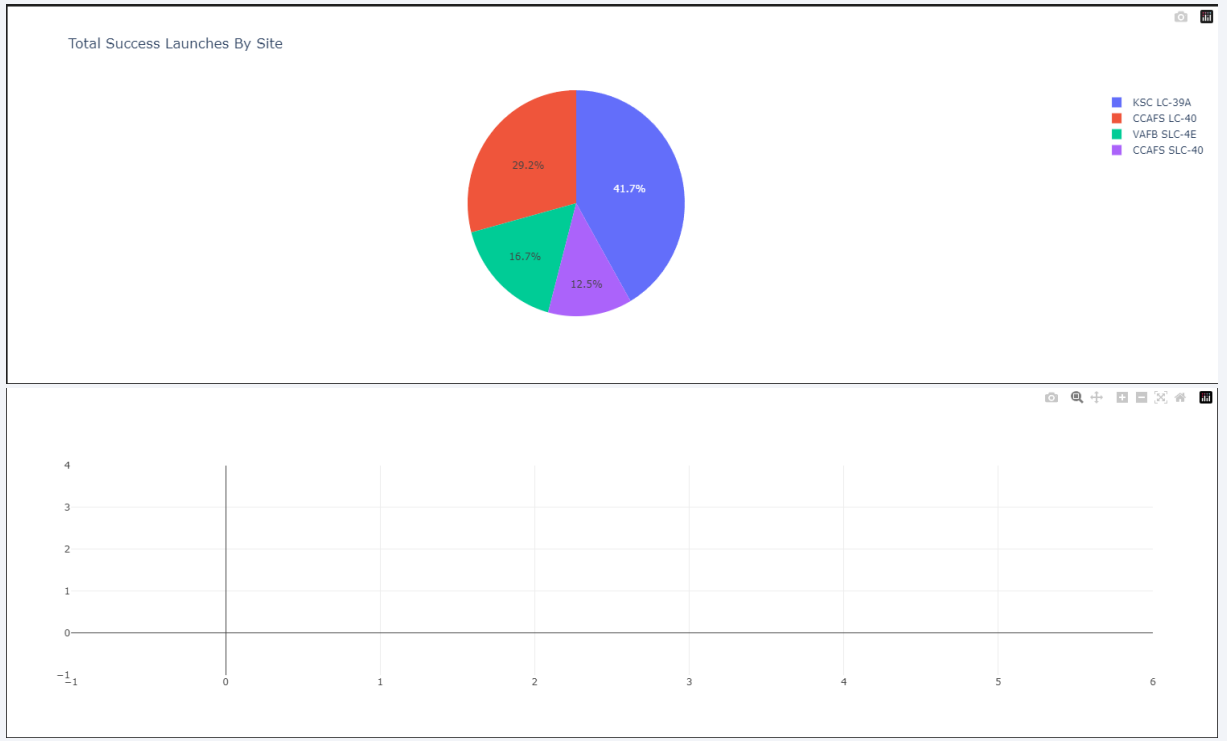
Predictive Analysis (Classification)

- 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

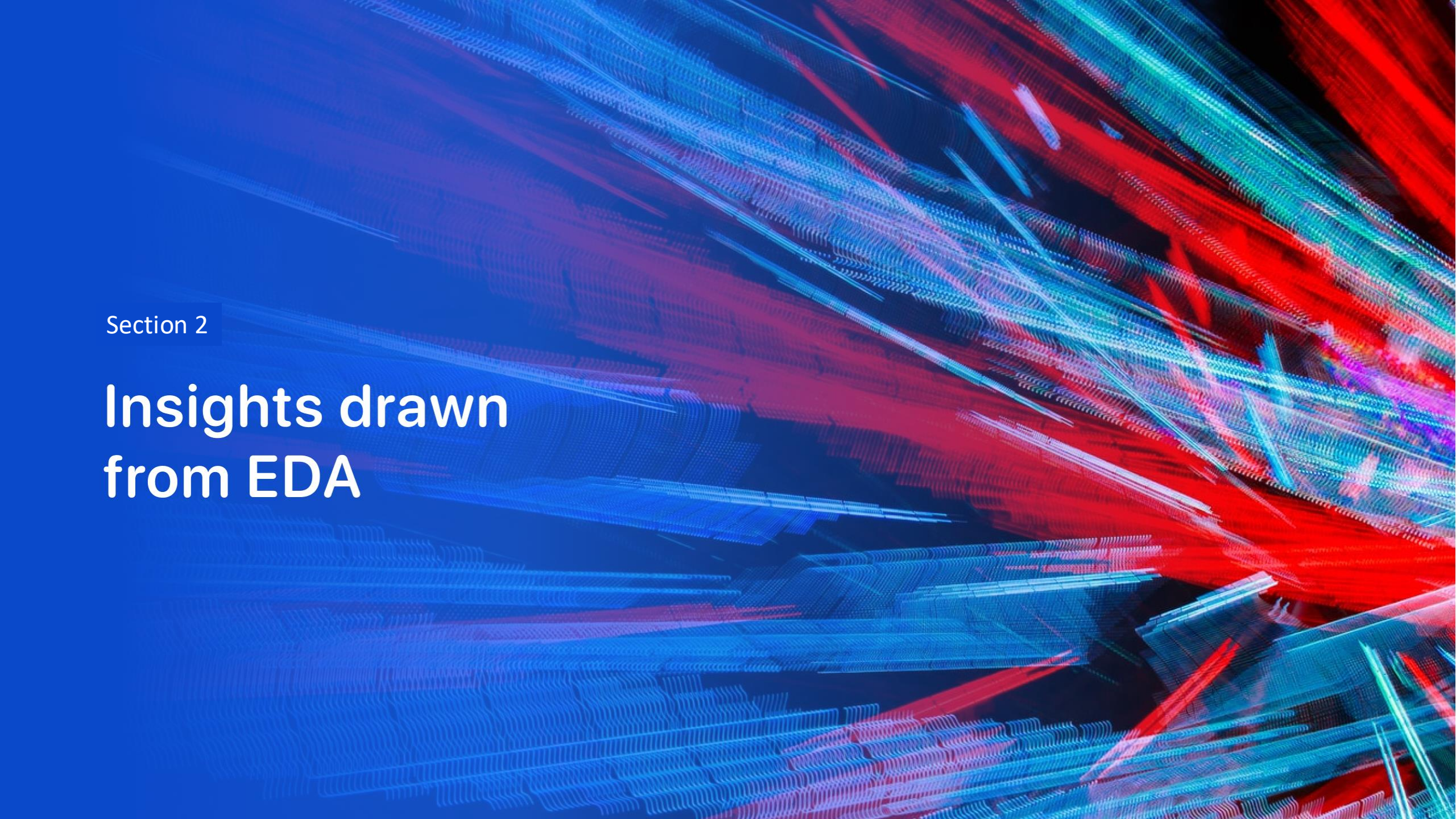


Results

- The left screenshot is a preview of the Dashboardwith Plotly Dash.
- The results of EDA with nvisualization, EDA with SQL, Interactive Map with Folium, and Interactive Dashboardwill be shown in the next slides.
- Comparing the accuracy of the four methods, all returnthe same accuracy of about 83% for test data.



[Github Link]

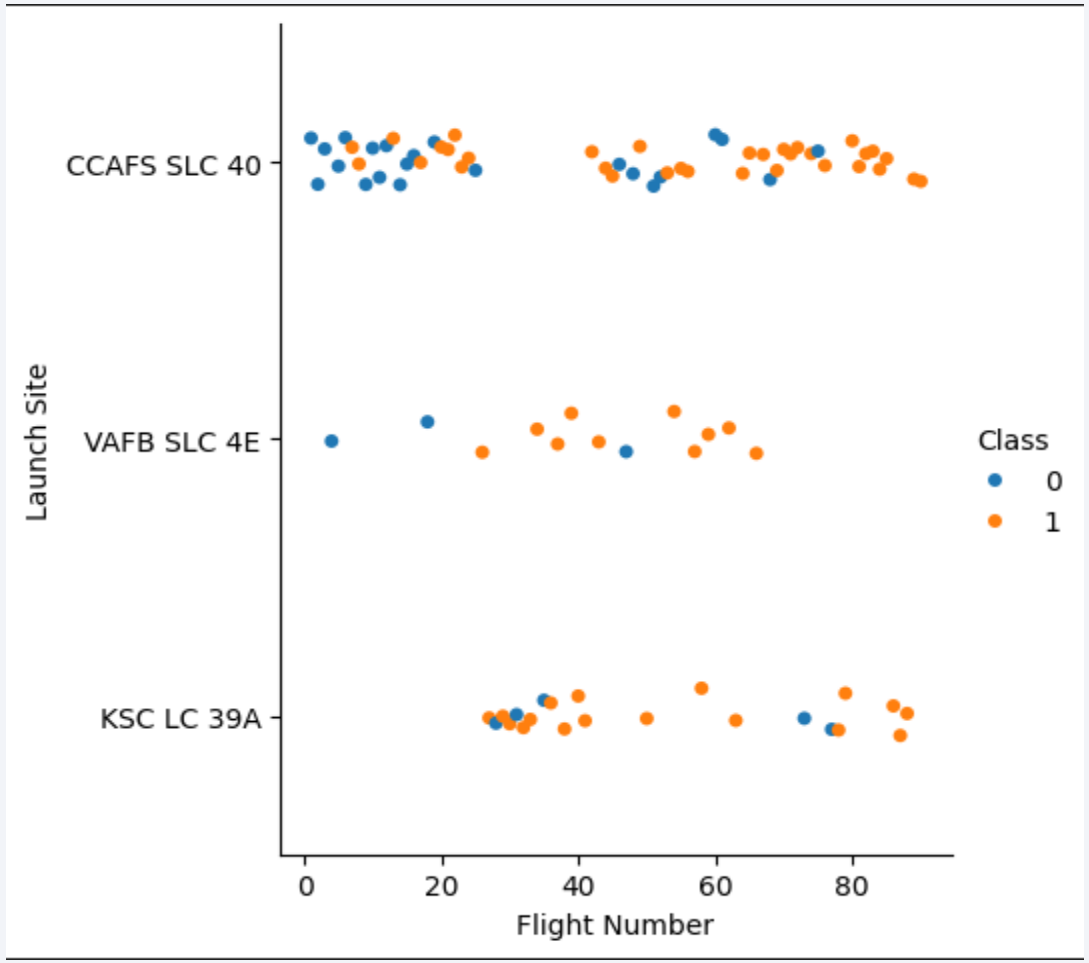

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- Class 0 (blue) represents unsuccessful launch, and Class 1
- (orange) represents successful launch.

- This figure shows that the success rate increased as the number of flights increased.

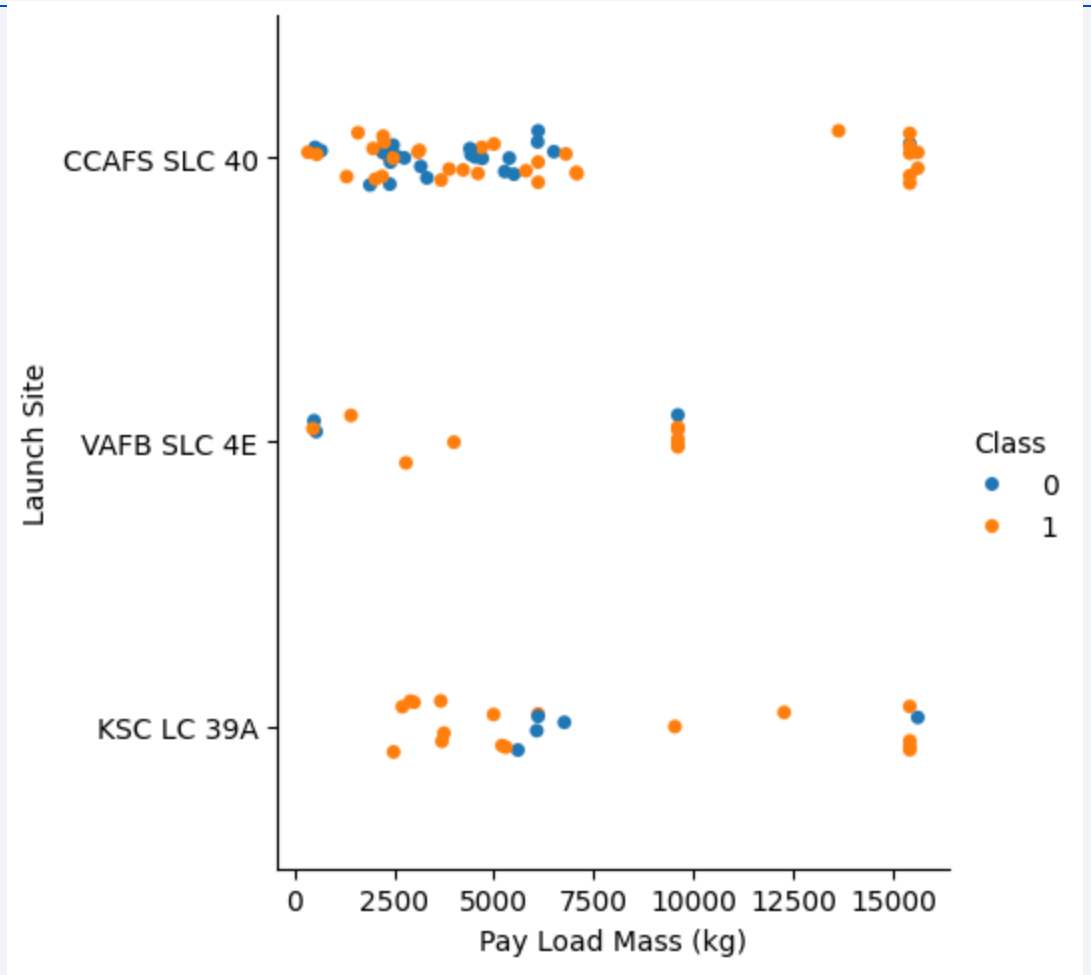
- As the success rate has increased considerably since the 20th flight, this point seems to be a big breakthrough.



Github Link

Payload vs. Launch Site

- At first glance, the larger pay load mass, the higher the rocket's success rate, but it seems difficult to make decisions based on this figure because no clear pattern can be found between successful launch and Pay Load Mass.

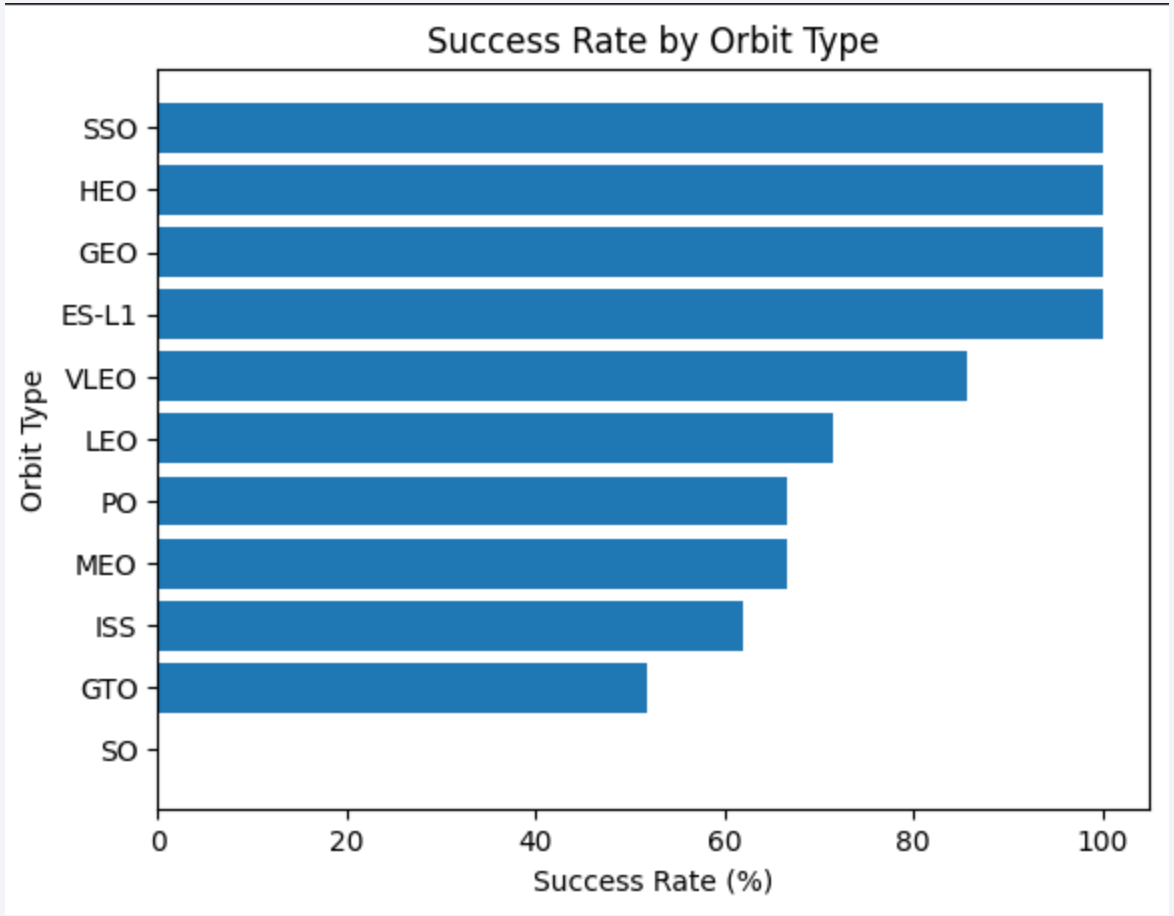


Github Link

Success Rate vs. Orbit Type

- Orbit types SSO, HEO, GEO, and ES-L1 have the highest success rates (100%).

- On the other hand, the success rate of orbit type GTO is only 50%, and it is the lowest except for type SO, which recorded failure in a single attempt.



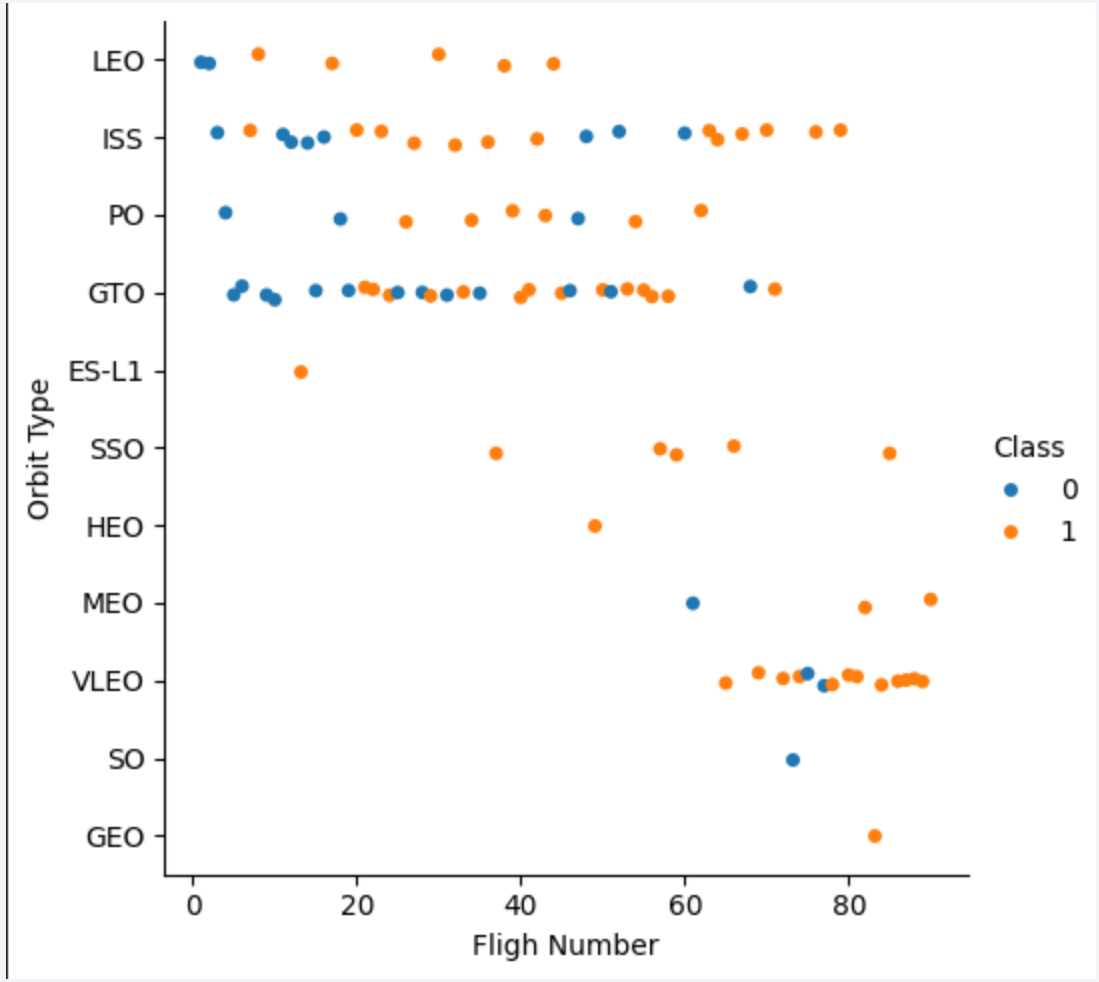
Github Link

Flight Number vs. Orbit Type

- In most cases, the launch outcome seems to be correlated with the flight number.

- On the other hand, in GTO orbit, there seems to be no relationship between flight numbers and success rate.

- SpaceX starts with LEO with a moderate success rate, and it seems that VLEO, which has a high success rate, is used the most in recent launches.

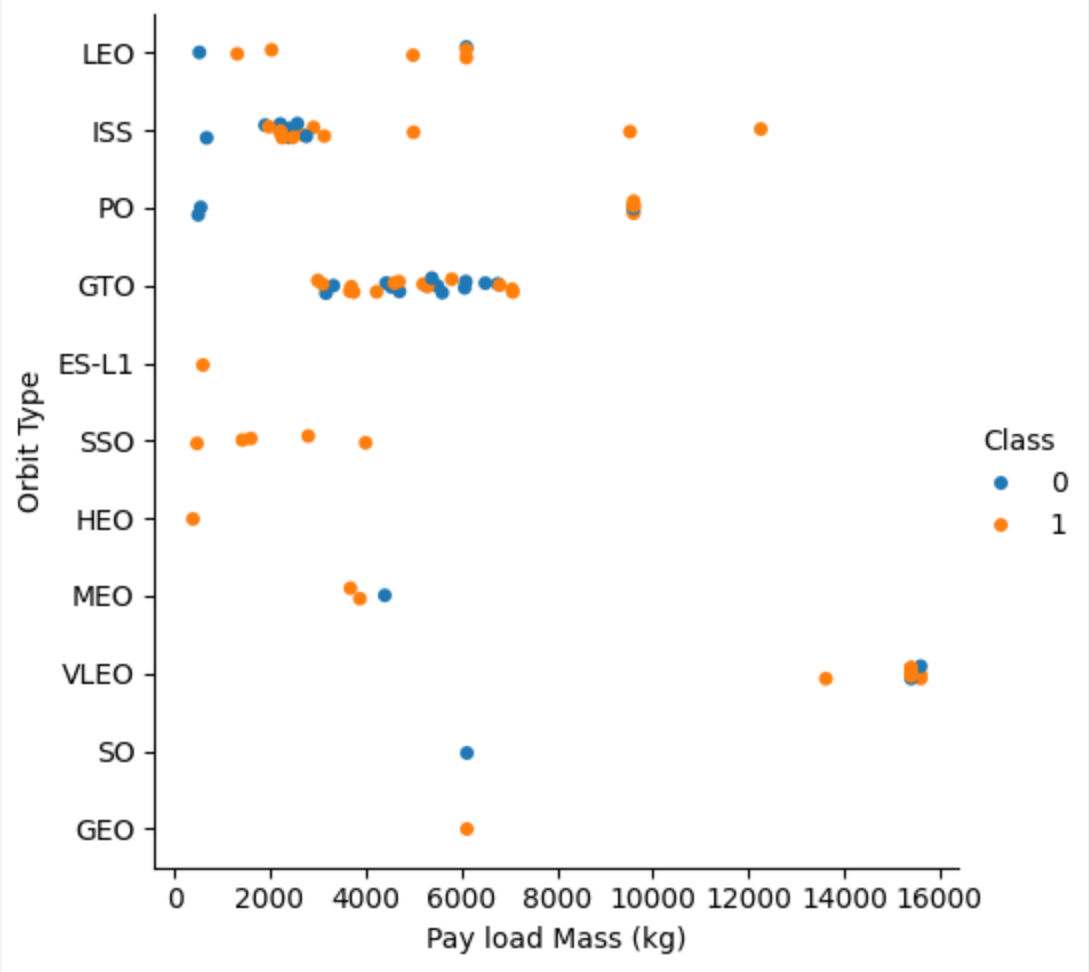


GitHub Link

Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for LEO and ISS.

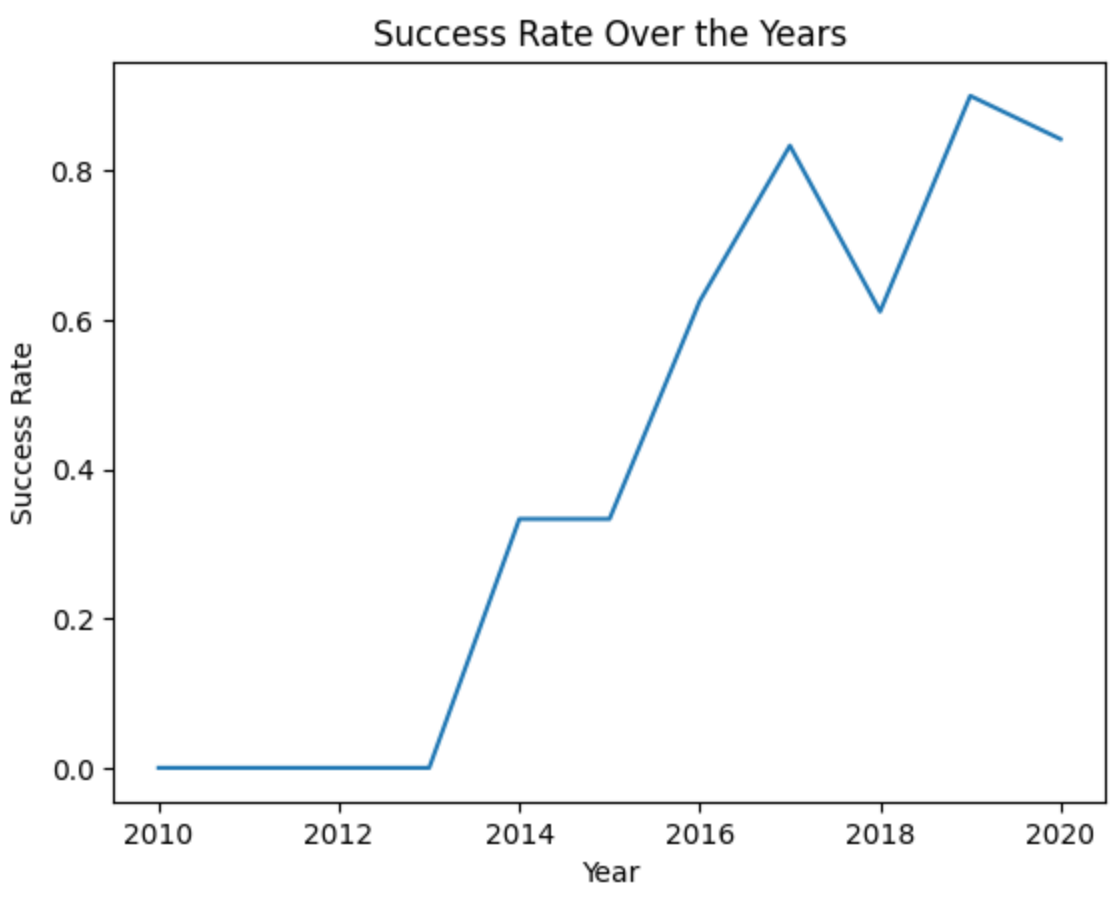
- However, in the case of GTO, it is hard to distinguish between the positive landing rate and the negative landing because they are all gathered together.



Github Link

Launch Success Yearly Trend

- Since 2013, the success rate has continued to increase until 2017.
- The rate decreased slightly in 2018.
- Recently, it has shown a success rate of about 80%.



Github Link

All Launch Site Names

- When the SQL DISTINCT clause is used in the query, only unique values are displayed in the Launch_Site column from the SpaceX table.

- There are four unique launch sites:

CCAFS LC-40, CCAFS SLC-40,

KSC LC-39A, VAFB SLC-4E

```
%sql SELECT distinct Launch_Site     from SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

- Only five records of the SpaceXtable were displayed using LIMIT 5 clause in the query.
- Using the LIKE operator and the percent sign (%) together, the Launch_Site name starting withCAA could be called.

```
%sql select Launch_Site from SPACEXTBL where Launch_Site like 'CCA%' limit 5
```

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

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

Total Payload Mass

- Using the SUM() function to calculate the sum of column PAYLOAD_MASS__KG_.

- In the WHERE clause, filter the dataset to perform calculations only if Customer is NASA (CRS).

```
%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Using the AVG() function to calculate the average value of column PAYLOAD_MASS__KG_.

- In the WHERE clause, filter thedataset to perform calculations only if Booster_version is F9 v1.1.

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL Where Booster_Version = 'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Using the MIN() function to find out the earliest date in the column DATE.

- In the WHERE clause, filter the dataset to perform a search only if Landing__outcome is Success (ground pad).

```
%sql select min(Date) from SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'
✓ 0.0s

 * sqlite:///my_data1.db
Done.

 min(Date)
 2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- In the WHERE clause, filter the dataset to perform a search if Landing__outcome is Success (drone ship).

Using the AND operator to display a record if additional condition PAYLOAD_MASS__KG_ is between 4000 and 6000.

```
%sql select Booster_Version from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Using the COUNT() function to calculate the total number of columns.

- Using the GROUP BY statement, groups rows that have the same values into summary rows to find the total number in each Mission_outcome.

- According to the result, SpaceX seems to have successfully completed nearly 99% of its missions.

```
%sql select count(Mission_Outcome) from SPACEXTBL
```

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

count(Mission_Outcome)
101

```
%%sql
SELECT MISSION_OUTCOME, COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

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

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

Boosters Carried Maximum Payload

- Using a subquery, first, find the maximum value of the payload by using MAX() function, and second, filter the dataset to perform a search if PAYLOAD_MASS__KG_ is the maximum value of the payload.

- According to the result, version F9 B5 B10xx.x

boosters could carried the maximum payload.

```
%sql select 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

- In the WHERE clause, filter the dataset to perform a search if Landing__outcome is Failure (drone ship).
  - Using the AND operator to display a record if additional condition YEAR is 2015.
- In 2015, there were two landing failures on drone ships.

```
%sql select Landing_Outcome,Booster_Version ,Launch_Site from SPACEXTBL WHERE Landing_Outcome = 'Failure (drone ship)' AND strftime('%Y', DATE) = '2015';
```

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

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

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

- In the WHERE clause, filter the dataset to perform a search if the date is between 2010-06-04 and 2017-03-20.

- Using the ORDER BY keyword to sort the records by total number of landing, and using DESC keyword to sort the records in descending order.

- According to the results, the number of successes and failures between 2010-06-04 and 2017-03-20 was similar

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

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

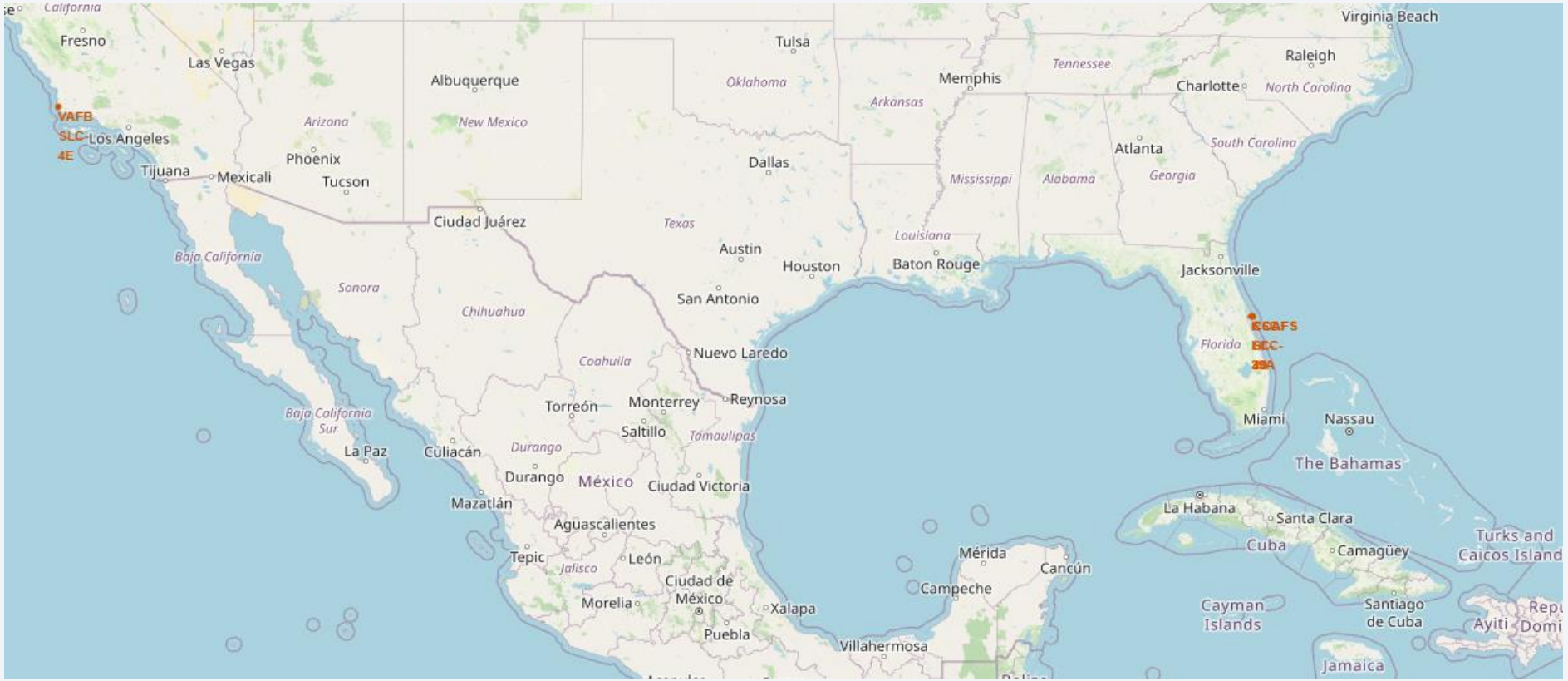
Landing_Outcome	OutcomeCount
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1


Section 3
Launch Sites
Proximities Analysis

All Launch Sites Locations

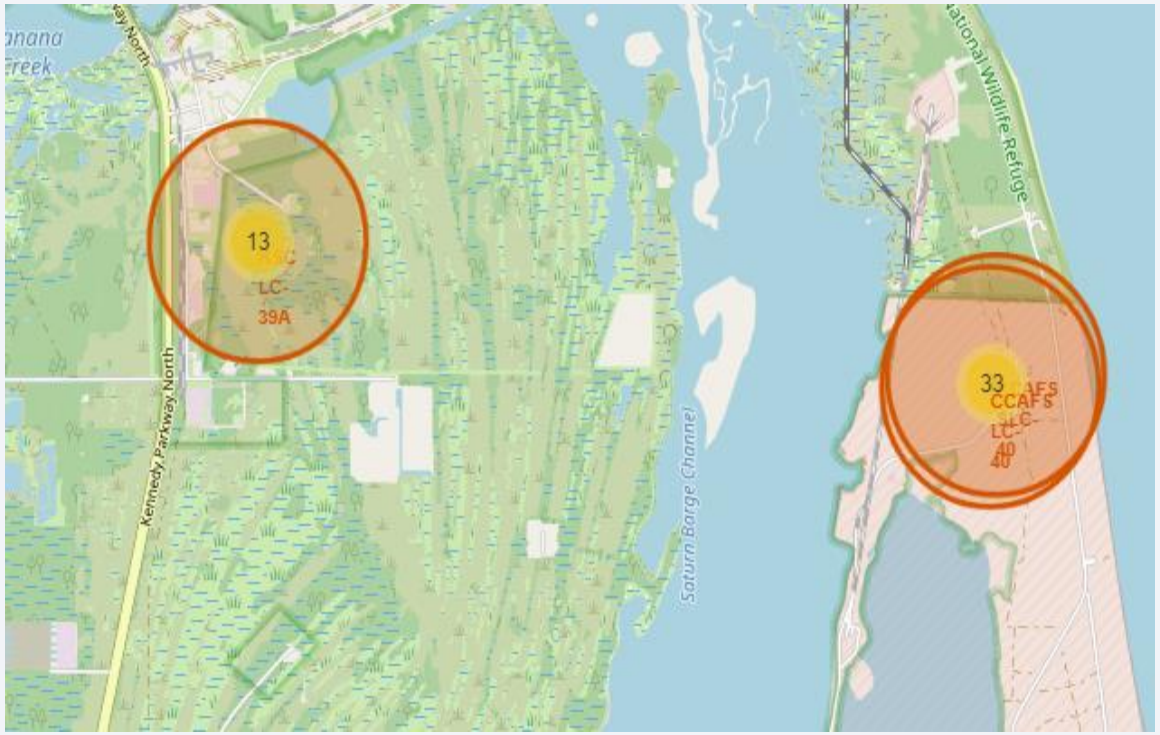
- As can be seen on the map, all launch sites are

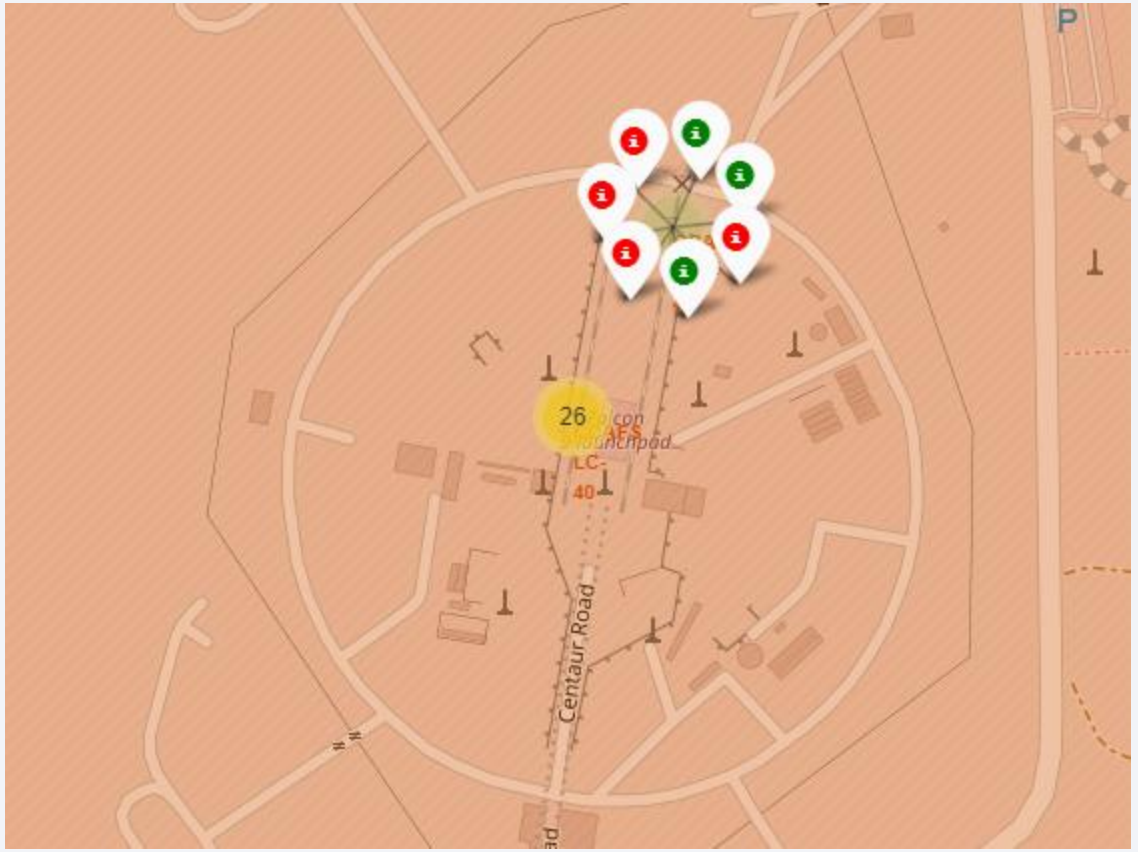
near the coast.



Color-labeled Launch Outcomes

- By clicking on the markerclusters, successful landing (green) or failed landing (red) are displayed.

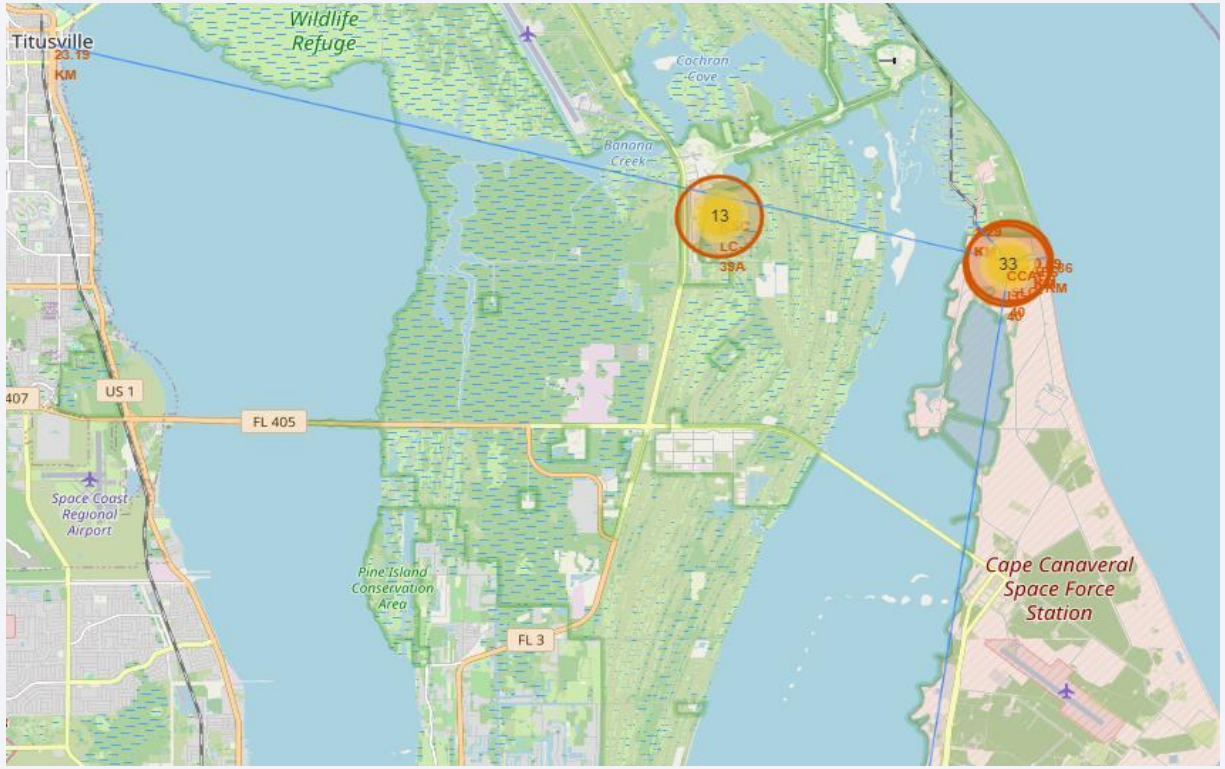


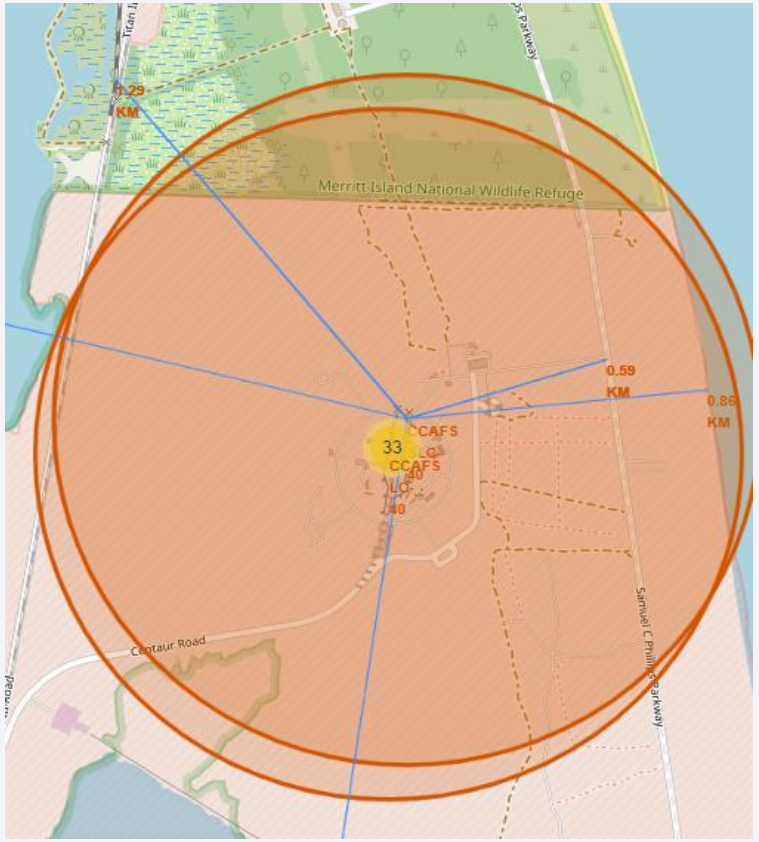


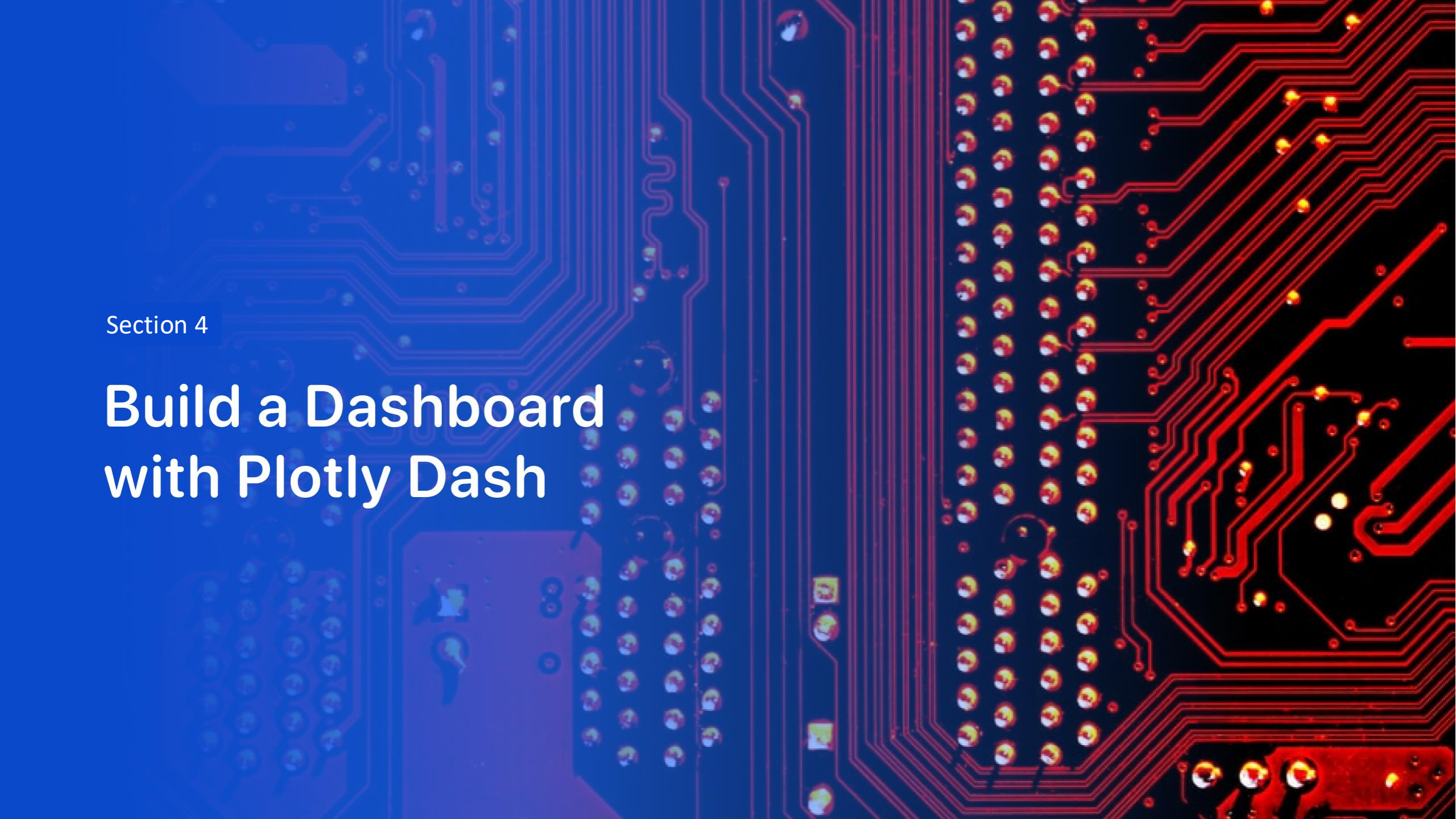
Proximities of Launch Sites

- It can be found that the launch site is close to railways and highways for transportation of equipment or personnel,

and is also close to coastline and relatively far from the cities so that launch failure does not pose a threat.

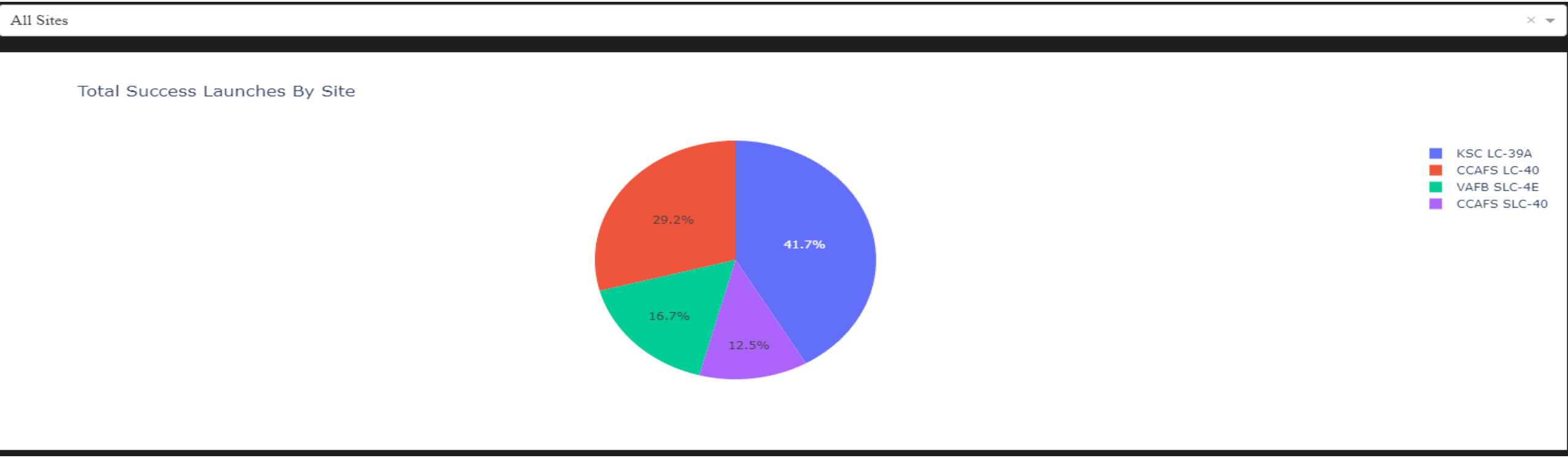





Section 4
Build a Dashboard with Plotly Dash

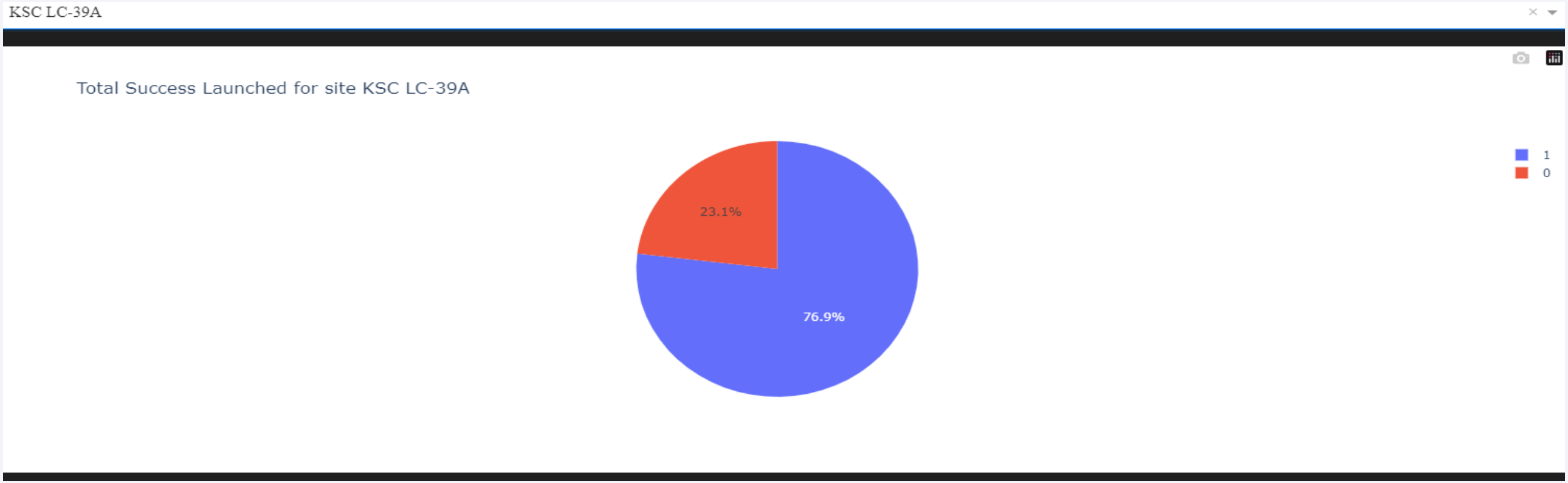
Pie Chart - Total success launches

- We can find here piechart descriping successful launching for KSC LC-39A,CCAFS LC-40, VAFB SLC-4E, CCAFS SLC-40



Highest launch ration

- KSLC-39A has the highest success rate with 10 landing successes (76.9%) and 3 landing failures (23.1%).



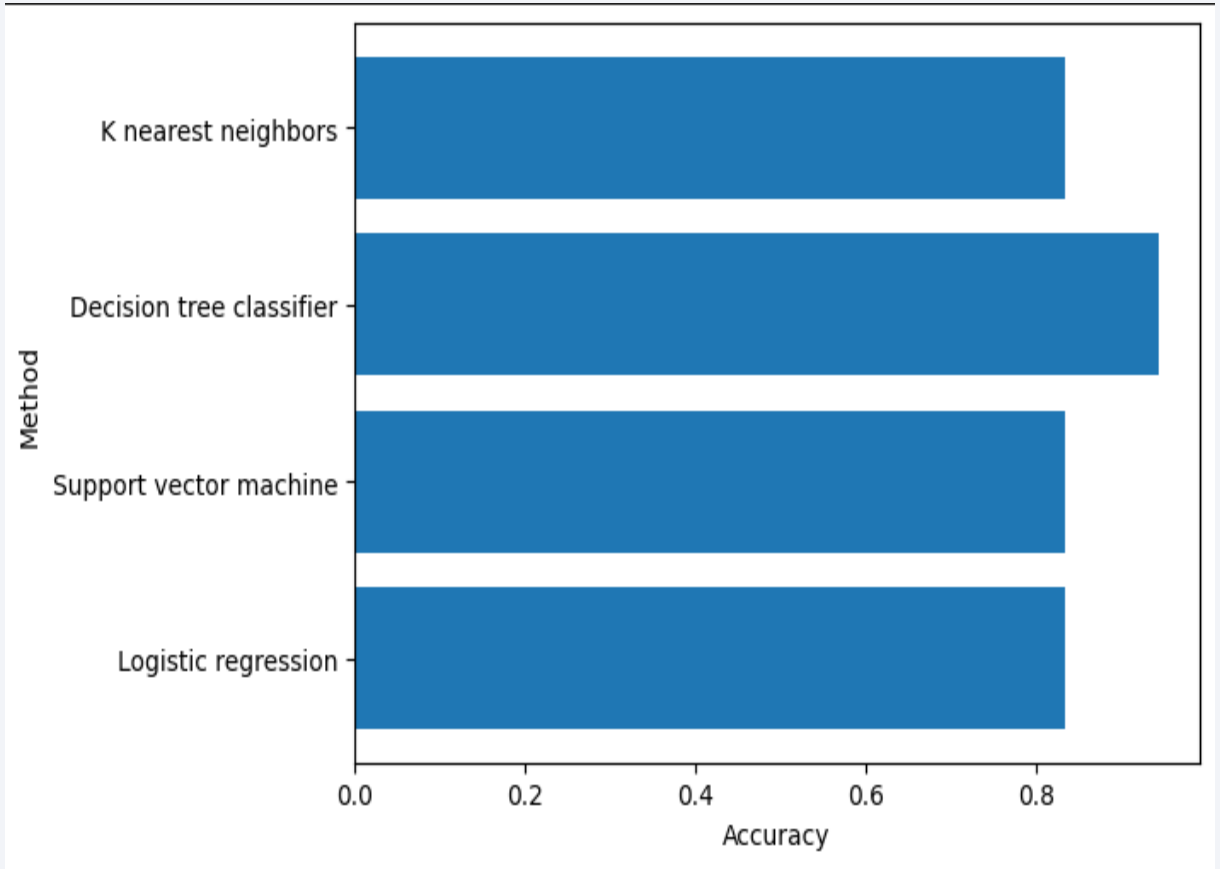

0
2500
5000
7500
10000
0.15
0.1
0.05
0
−0.05
−0.1
0.96
0.97
0.98
0.99
1
1.01
1.02
1.03
1.04

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- We can find here in the barchart the accuracy of each model.
- We can notice that Descision Tree has the best accuracy ~ 0.9%
- We can also notice that the other models having same accuracy rate 0.8%

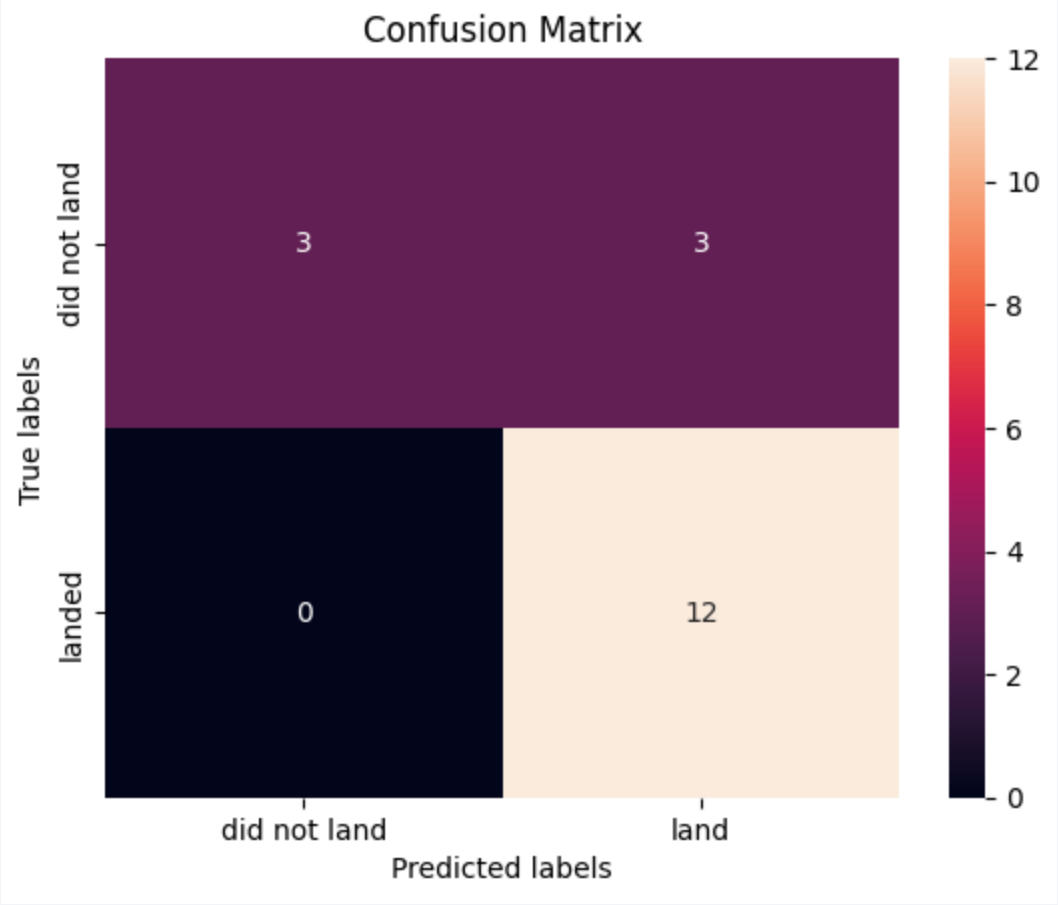


Confusion Matrix

- The confusion matrix is the same for all models because all models performed the same for the test set.

- The models predicted 12 successful landings when the true label was successful and 3 failed landings when the true label was failure. But there nwere also 3 predictions that said successful landings when the true label was failure (false positive).

- Overall, these models predict successful landings.



Conclusions

- As the number of flights increased, the success rate increased, and recently it has exceeded 80%.
- Orbital types SSO, HEO, GEO, and ES-L1 have the highest success rate (100%).
- The launch site is close to railways, highways, and coastline, but far from cities.
- KSLC-39A has the highest number of launch successes and the highest success rate among all sites.
- The launch success rate of low weighted payloads is higher than that of heavy weighted payloads.
- In this dataset, all models have the same accuracy (83.33%) except tree model, but it seems that more data is needed to determine the optimal model due to the small data size.

Appendix

- GitHub Link

- IBM Data Science professional certificate Capastone

Thank you!