IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

- Data Science Capstone Project

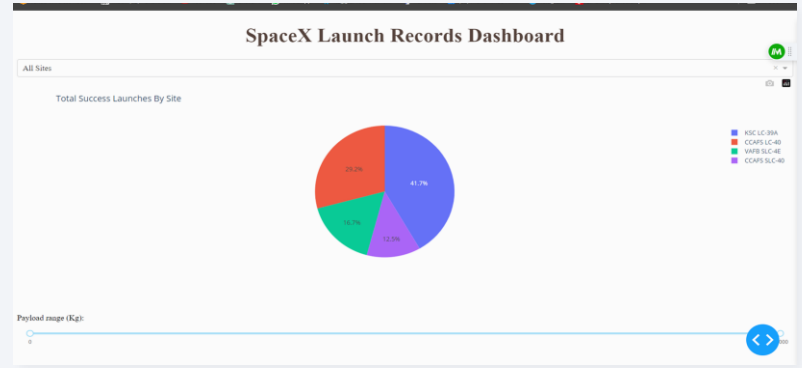
Muhammad Ahsan
18/02/2024

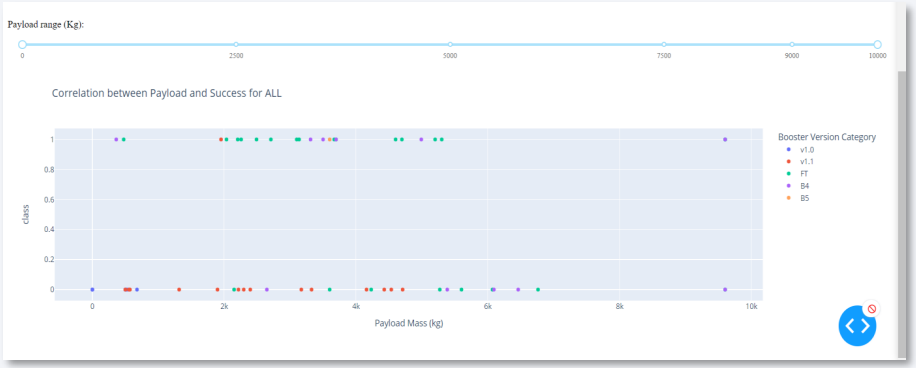
Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a 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:
  - Via SpaceX Rest API
  - Web Scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia
- Perform data wrangling:

Steps:

- Perform EDA and determine the data labels
- Create a binary variable '**class**' that will represent the landing outcome.
- Outcomes converted into 1 for a successful landing and 0 for unsuccessful landing

Methodology

Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
  - Using SQL queries to manipulate and evaluate the SpaceX dataset
  - Using Pandas and Matplotlib to visualize relationships between variables, and determine patterns
- Perform interactive visual analytics using Folium and Plotly Dash
  - Geospatial analytics using Folium
  - Creating an interactive dashboard using Plotly Dash
- Perform predictive analysis using classification models
  - Find best Hyperparameter for SVM, Classification Trees and Logistic Regression

Data Collection

- The data collection process includes a combination of API requests from the SpaceX API and web scraping data from a table in the Wikipedia page of SpaceX, Falcon 9 and Falcon Heavy Launches Records.
  - SpaceX API Data Columns: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude
  - Wikipedia Web Scrape Data Columns: Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

SpaceX API

Web Scraping

Data Collection – SpaceX API

1.Requesting rocket launch data from SpaceX API

```
Click here to ask Blackbox to help you code faster
1 spacex_url="https://api.spacexdata.com/v4/launches/past"
✓ 0.0s

Click here to ask Blackbox to help you code faster
1 response = requests.get(spacex_url)
✓ 1.8s
```

2. Converting Response to a JSON file

```
Click here to ask Blackbox to help you code faster
1 # Use json_normalize meethod to convert the json result into a dataframe
2 data = pd.json_normalize(response.json())
✓ 0.0s
```

3. Using custom functions to clean data

```
Click here to ask Blackbox to help you code faster
1 # Call getBoosterVersion
2 getBoosterVersion(data)

Click here to ask Blackbox to help you code faster
1 # Call getLaunchSite
2 getLaunchSite(data)
✓ 2m 32.4s

Click here to ask Blackbox to help you code faster
1 # Call getPayloadData
2 getPayloadData(data)
2m 52.4s

Click here to ask Blackbox to help you code faster
1 # Call getCoreData
2 getCoreData(data)
2m 39.2s
```

[GitHub URL]

4.Combining the columns into a dictionary to create data frame

```
Click here to ask Blackbox to help you code faster
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}

# Create a data from launch_dict
launch_df = pd.DataFrame(launch_dict)
```

5.Filtering dataframe and exporting to a CSV

```
Click here to ask Blackbox to help you code faster
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_df[launch_df['BoosterVersion']=='Falcon 9']

Click here to ask Blackbox to help you code faster
data_falcon9.to_csv('dataset_part_1.csv', index=False)
✓ 0.1s
```

Data Collection - Scraping

1.Getting response from HTML

```
Click here to ask Blackbox to help you code faster
# use requests.get() method with the provided static_url
response = requests.get(static_url).text
11.0s
```

2.Creating a BeautifulSoup object

```
Click here to ask Blackbox to help you code faster
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html.parser')
```

3.Finding all tables and assigning the result to a list

```
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

4.Extracting column name one by one

```
column_names = []
for row in first_launch_table.find_all('th'):
    # print("Header: " + str(row))
    col_name = extract_column_from_header(row)
    if col_name is not None and len(col_name) > 0:
        column_names.append(col_name)

print("\nColumn Names Extracted: " + str(column_names))
```

5.Creating an empty dictionary with keys

```
Click here to ask Blackbox to help you code faster
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
0.0s
```

6.Filling up the launch_dict with launch records (Too long to put in here, so please refer to the notebook)

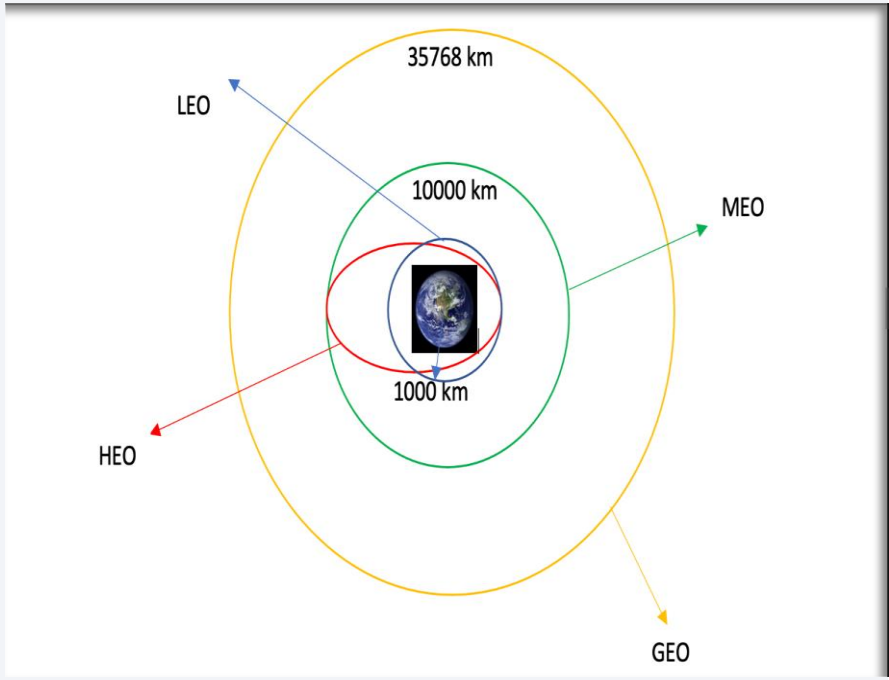
7.Creating a Dataframe and exporting it to a CSV

```
Click here to ask Blackbox to help you code faster
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
0.0s
Click here to ask Blackbox to help you code faster
df.to_csv('spacex_web_scraped.csv', index=False)
```

GitHub URL

Data Wrangling

- There are several cases in which the booster failed to successfully land on the dataset, and sometimes it attempted to land but failed because of accident.
  - True Ocean: the mission result has successfully landed in a specific area of the ocean
  - False Ocean: the mission result has not successfully landed in a specific area of the ocean
  - True RTLS: the mission result successfully landed on the ground pad
  - False RTLS: the mission result has not successfully landed on the ground pad
  - True ASDS: the mission result has successfully landed on the drone ship
  - False ASDS: the mission result has not landed on the drone ship
- Converting these results into training labels:
  - 1 = successful / 0 = failure



GitHub URL

Data Wrangling

1.Calculating the number of launches at each site

```
Click here to ask Blackbox to help you code faster
# Apply value_counts() on column LaunchSite
df["LaunchSite"].value_counts()
0.0s
```

2.Calculating the number and occurrence of each orbit

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

3.Calculating the number and occurrence of mission outcome per orbit type

```
Click here to ask Blackbox to help you code faster
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
```

4.Creating a landing outcome label from Outcome column

```
landing_class = [0 if row in bad_outcomes else 1 for row in outcome]
```

5.Calculating the success rate for every landing in dataset

```
df["Class"].mean()
```

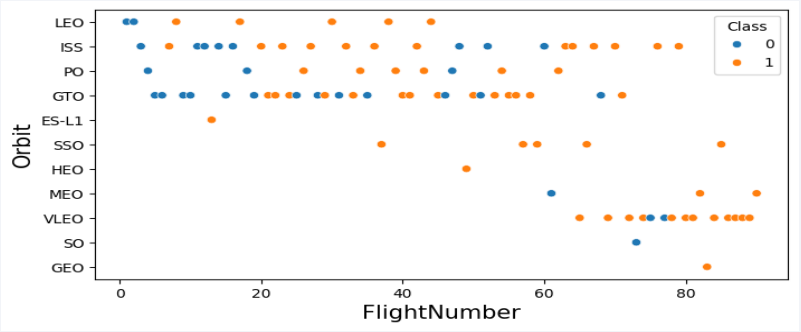
6.Exporting dataset to a CSV

```
Click here to ask Blackbox to help you code faster
df.to_csv("dataset_part_2.csv", index=False)
```

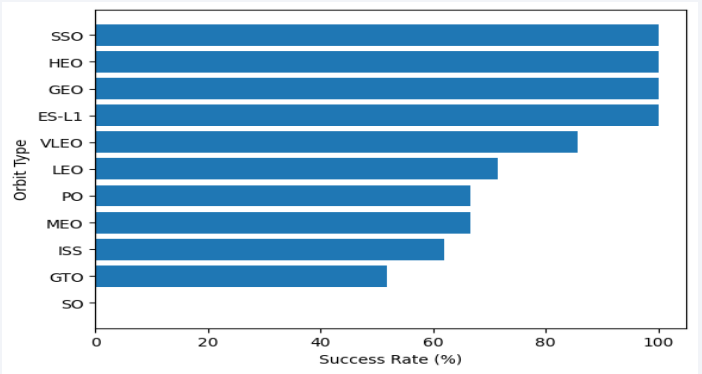
GitHub URL

EDA with Data Visualization

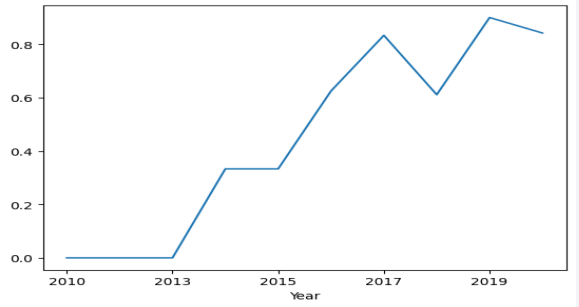
- **Scatter chart:**
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Flight Number vs. Orbit Type
  - Payload vs. Orbit Type
  - A scatter plot shows how much one variable is affected by another. The relationship between two variables is called a correlation. This plot is generally 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 URL

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 URL

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 URL**

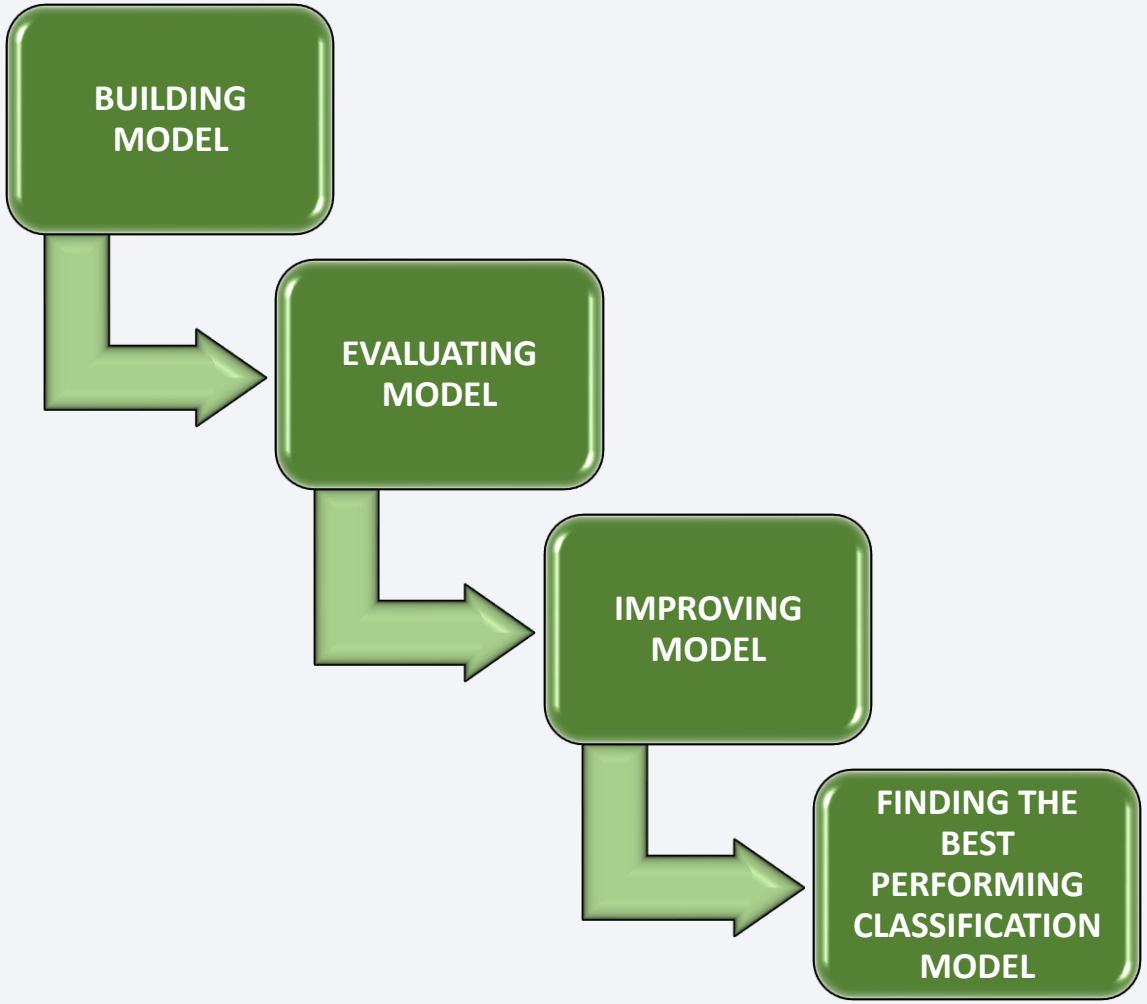
Build a Dashboard with Plotly Dash

- 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 chat*
    - 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 URL**

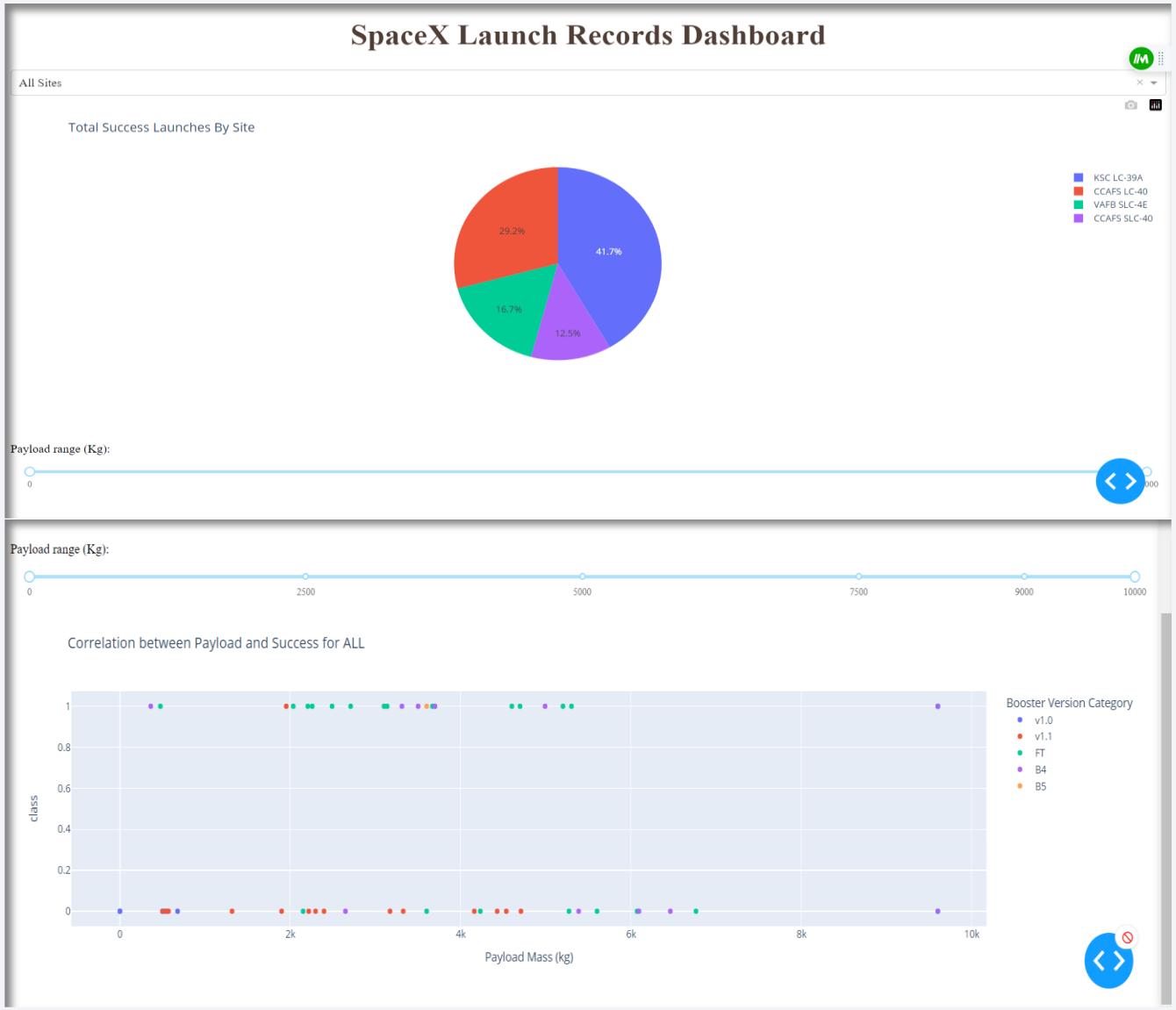
Predictive Analysis (Classification)

- Perform exploratory Data Analysis and determine Training Labels
  - ➢ Create a column for the class o 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

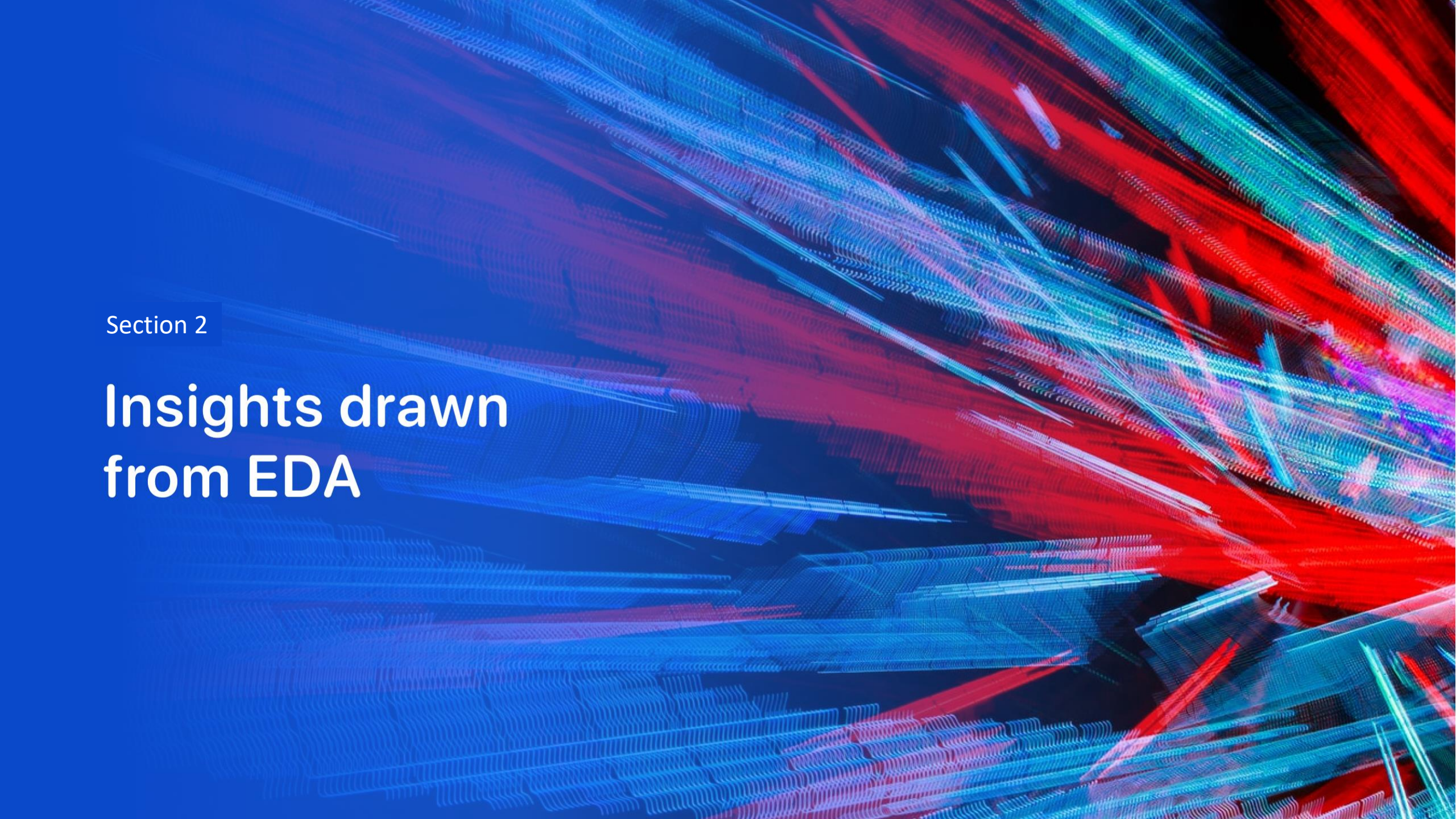


- GitHub URL

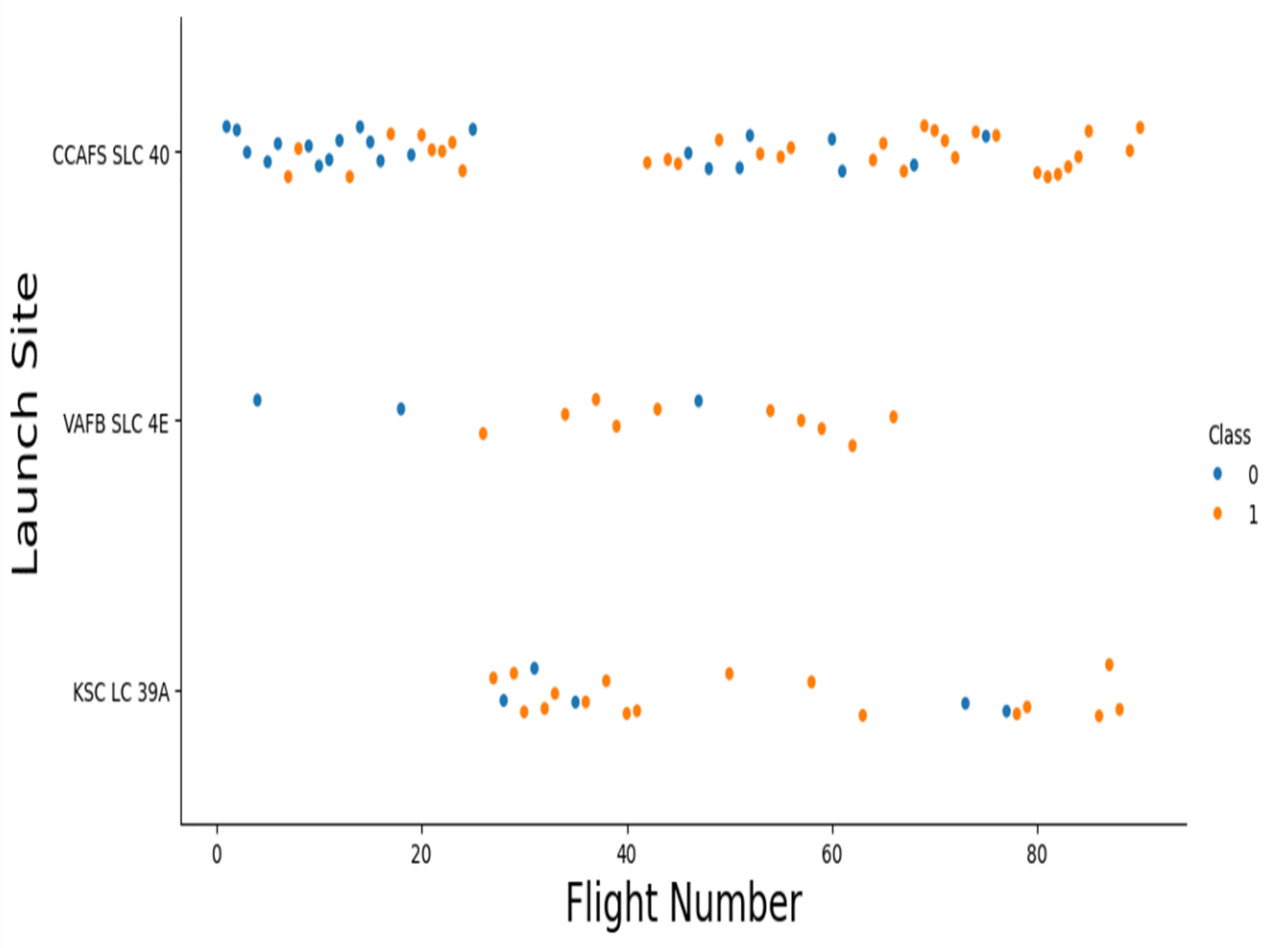
Results



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

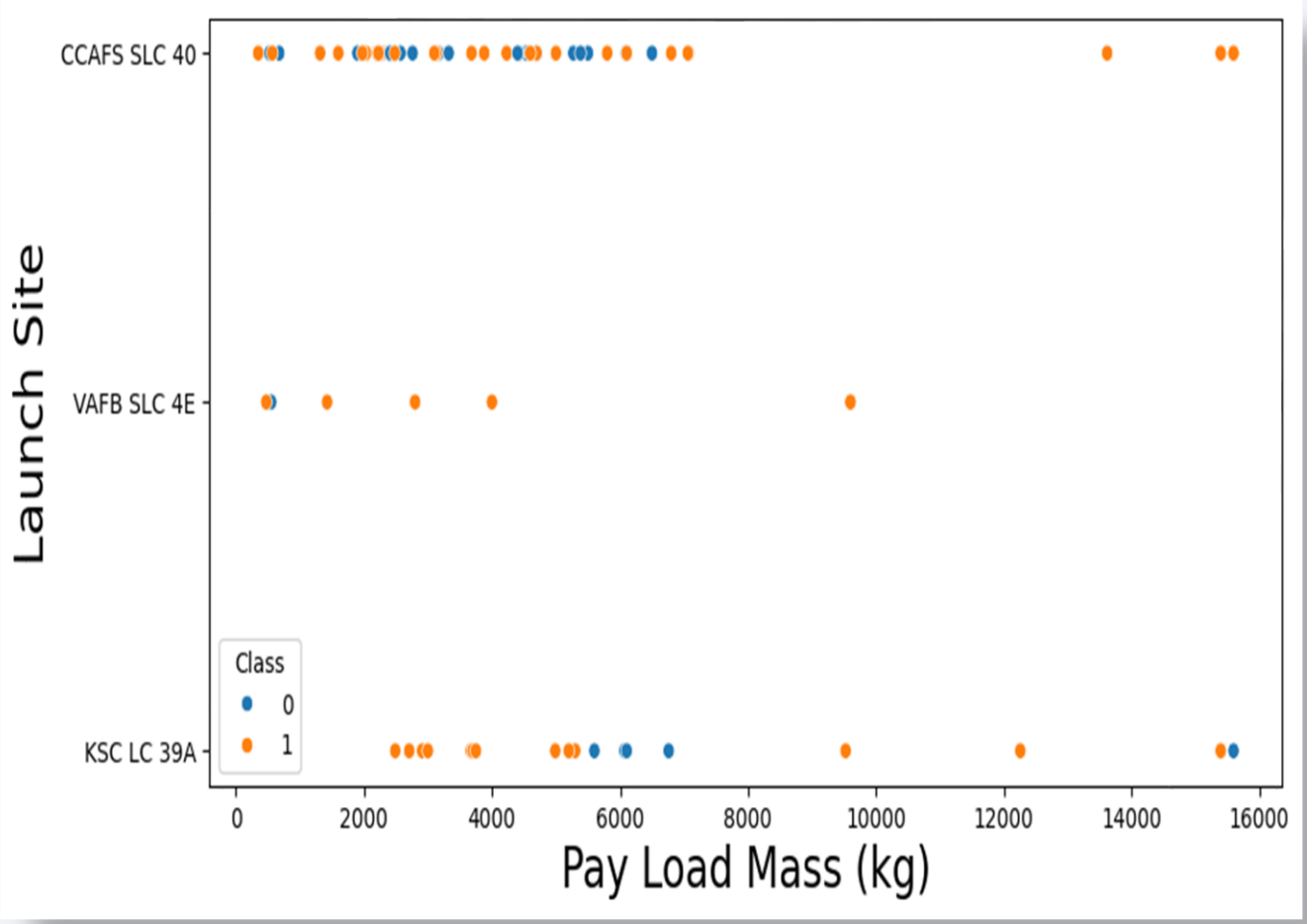

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



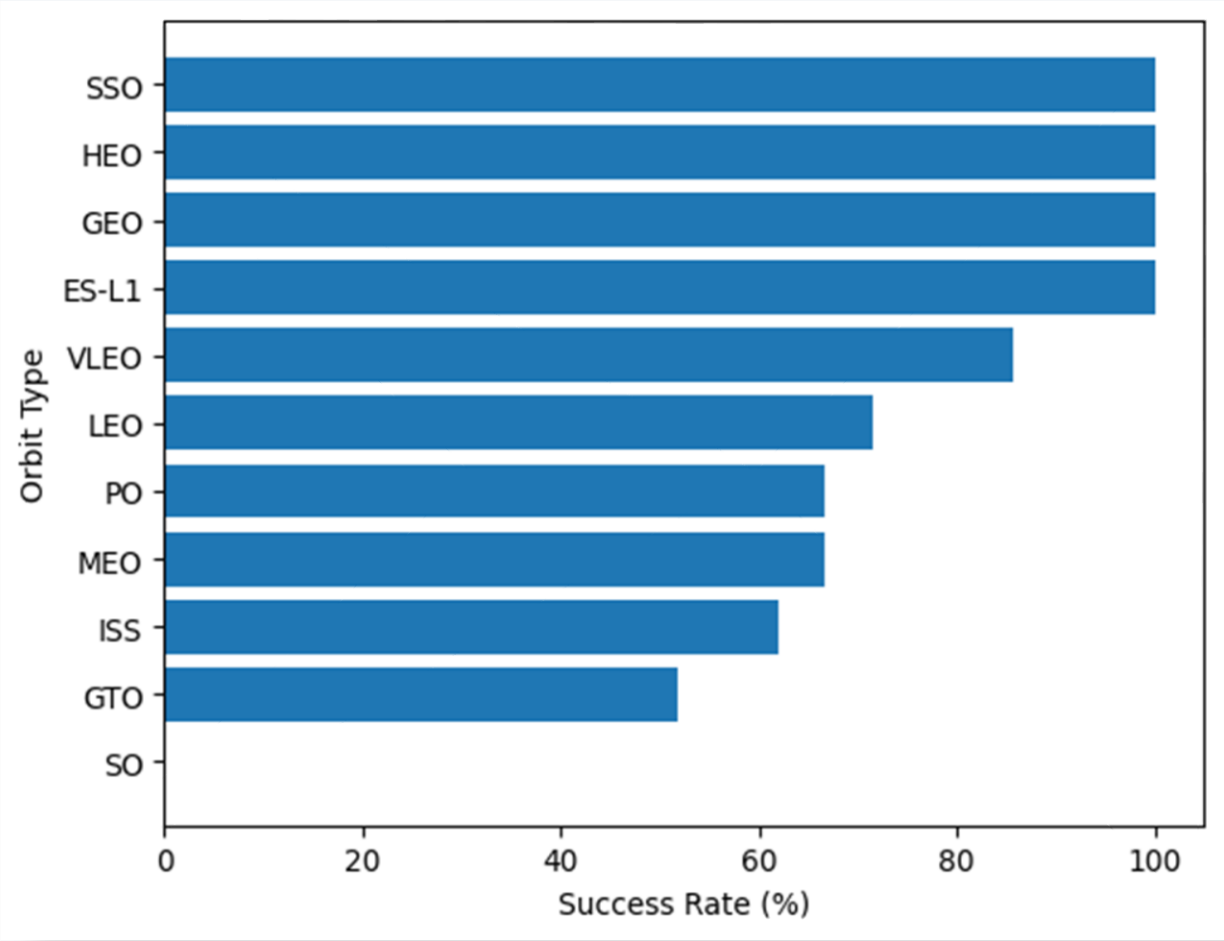
- Class 0 (blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch.
- This figure shows that **the success rate increased as the number of flights increased**.
- As the success rate has increased considerably since the 20th flight, this point seems to be a big breakthrough.

Payload vs. Launch Site



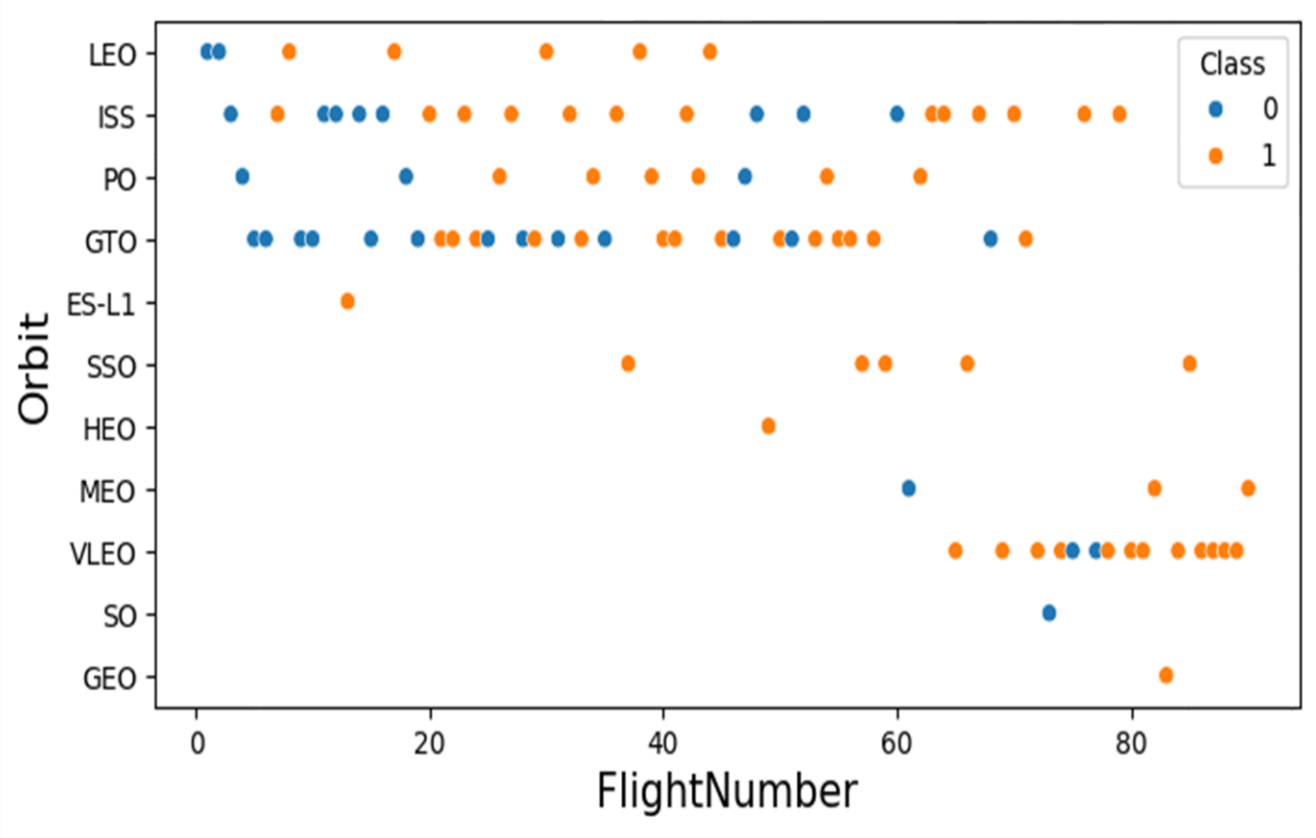
- Class 0 (blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch.
- At first glance, the larger pay load mass, the higher the rocket's success rate, but it seems difficult to make decisions based on this figure because **no clear pattern can be found between successful launch and Pay Load Mass.**

Success Rate vs. Orbit Type



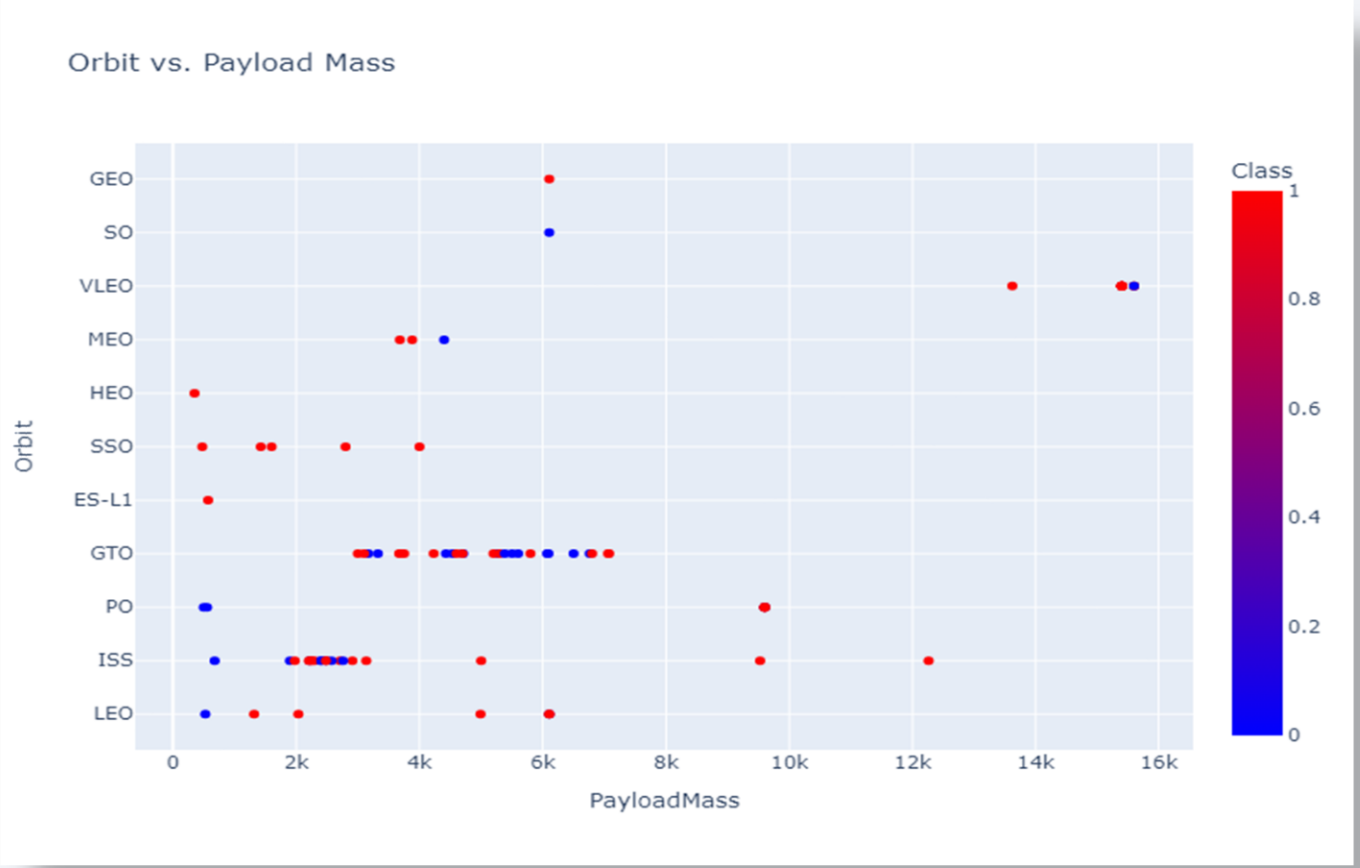
- Orbit types **SSO, HEO, GEO,** and **ES-L1** have **the highest success rates** (100%).
- On the other hand, the success rate of orbit type **GTO** is only 50%, and it is the **lowest** except for type SO, which recorded failure in a single attempt.

Flight Number vs. Orbit Type



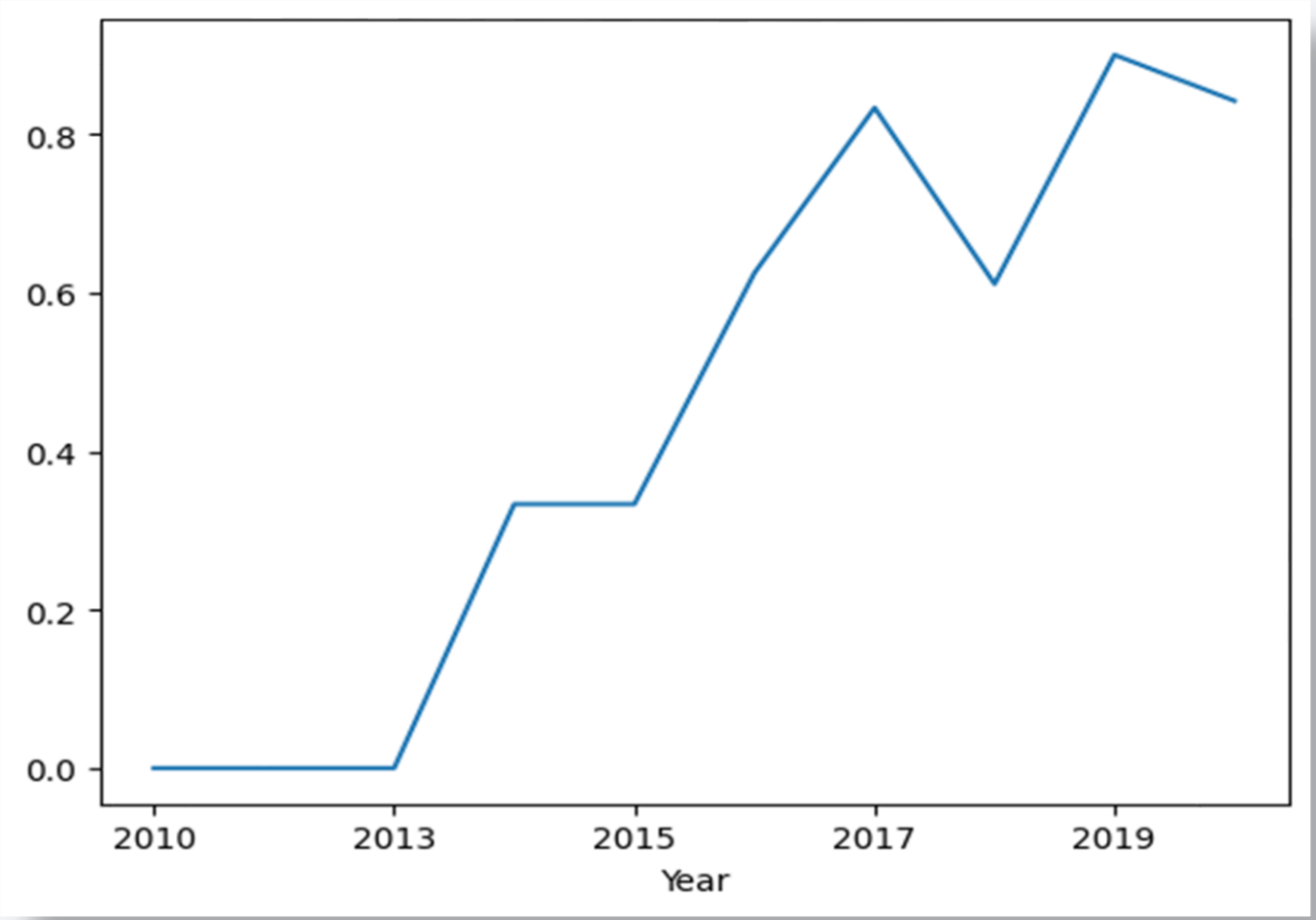
- Class 0 (blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch.
- In most cases, the launch outcome seems to be correlated with the flight number.
- On the other hand, in **GTO** orbit, there seems to be **no** relationship between flight numbers and success rate.
- 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.

Payload vs. Orbit Type



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

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%.

Section 3

EDA with SQL

All Launch Site Names

- **Query**

```
%sql select distinct
Launch_Site from
SPACEXTBL;
```

- **Result**

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

- 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

Launch Site Names Begin with 'CCA'

- **Query**

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

- **Result**

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

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

Total Payload Mass

- **Query**

```
%sql select sum(PAYLOAD_MASS__KG_)
from SPACEXTBL where Customer =
'NASA (CRS)';
```

- **Result**

sum(PAYLOAD_MASS__KG_)
45596

- 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).

Average Payload Mass by F9 v1.1

- **Query**

```
%sql select avg(PAYLOAD_MASS__KG_)
from SPACEXTBL where
Booster_Version = 'F9 v1.1';
```

- **Result**

avg(PAYLOAD_MASS__KG_)
2928.4

- Using the AVG() function to calculate the average value of column PAYLOAD_MASS__KG_.
- In the WHERE clause, filter the dataset to perform calculations only if Booster_version is F9 v1.1.

First Successful Ground Landing Date

- **Query**

```
%sql select min(Date) from
SPACEXTBL where Landing_Outcome =
'Success (ground pad)';
```

- **Result**

min(Date)
2015-12-22

- 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).

Successful Drone Ship Landing with Payload between 4000 and 6000

- **Query**

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

- **Result**

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

- 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.

Total Number of Successful and Failure Mission Outcomes

- **Query**

```
%sql select Mission_Outcome, count(*) as
count from SPACEXTBL group by
Mission_Outcome;
```

- **Result**

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

- 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.**

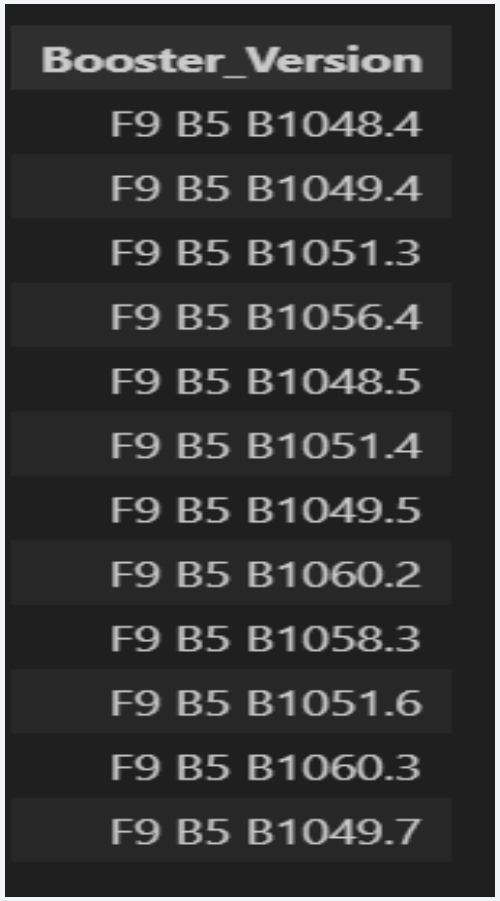
Boosters Carried Maximum Payload

- **Query**

```
%sql select Booster_Version from SPACEXTBL where
PAYLOAD_MASS__KG_ = (select
max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

- 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.

- **Result**



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

- **Query**

```
%sql select substr(Date, 6, 2) as month,
Landing_Outcome, Booster_Version, Launch_Site
from SPACEXTBL where substr(Date, 0, 5) = '2015'
and Landing_Outcome = 'Failure (drone ship)';
```

- **Result**

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

- 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.

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

- **Query**

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

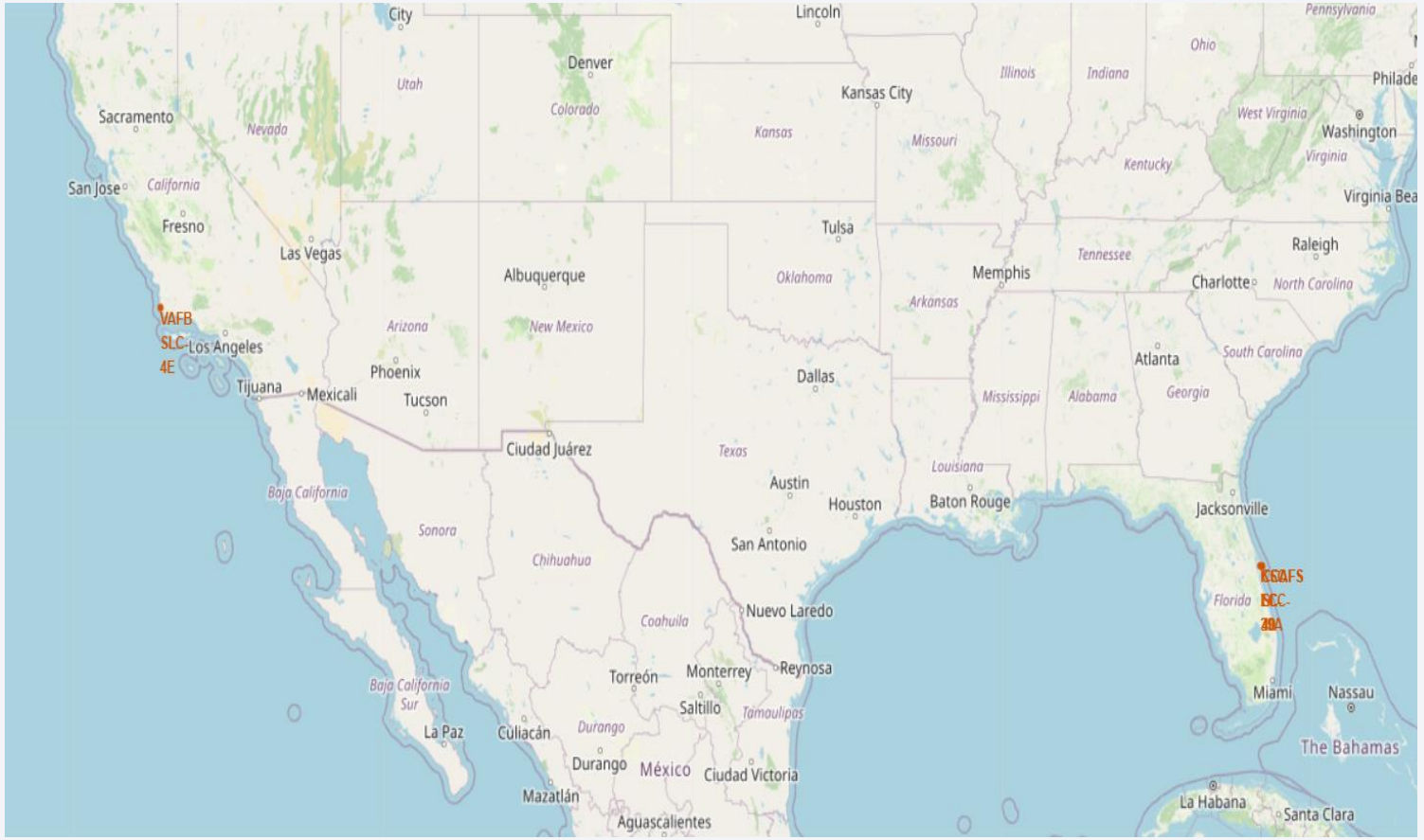
- **Result**

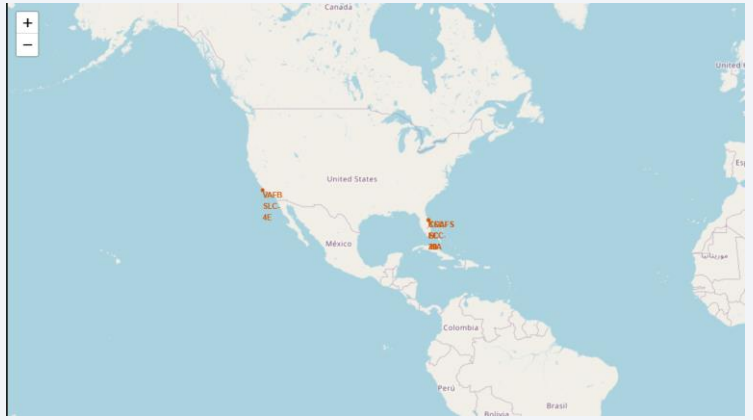
Landing_Outcome	total_number
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- 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.


Section 4
Launch Sites
Proximities Analysis

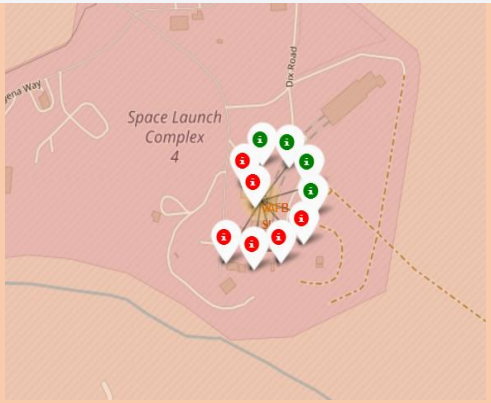
All Launch Sites' Location





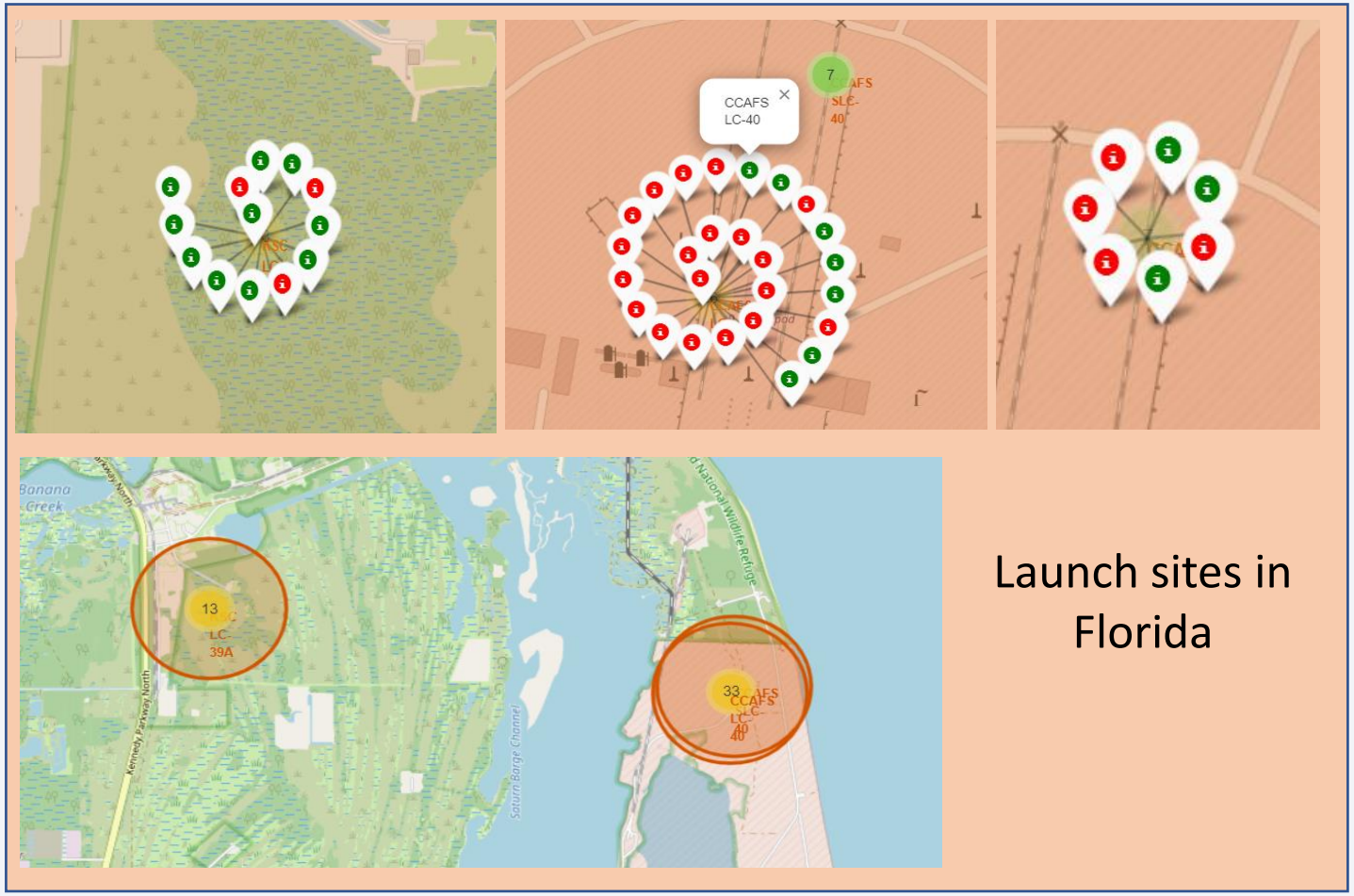
- The left map shows all SpaceX launch sites, and the right map also shows that all launch sites are in the United States.
- As can be seen on the map, all launch sites are near the coast.

Color-Labeled Launch Outcomes



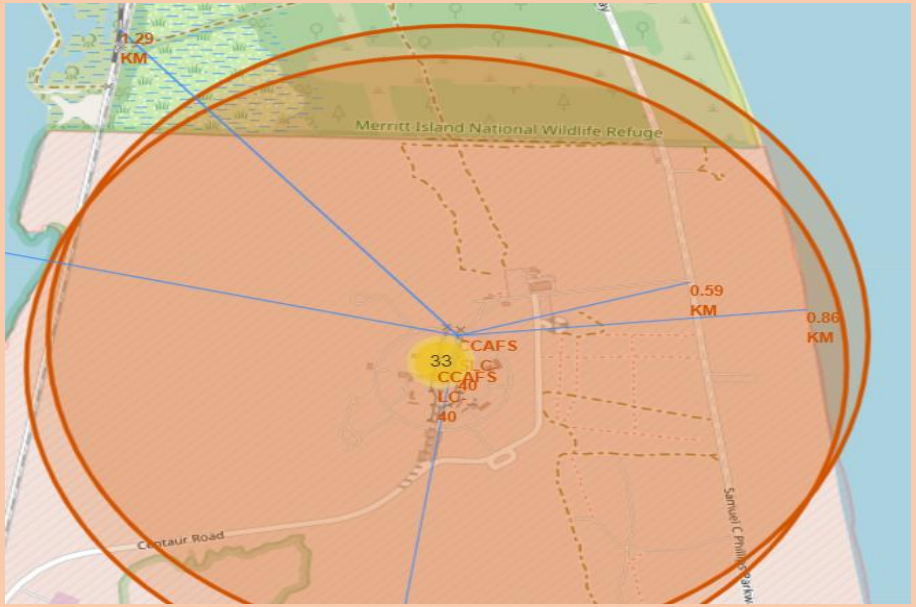
Launch site in California

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

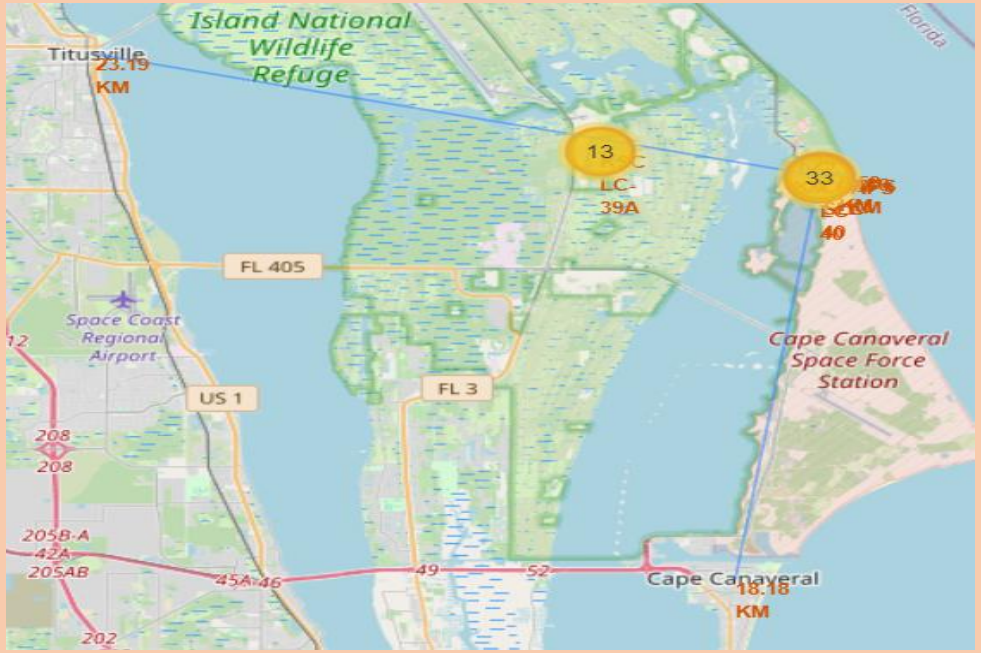


Launch sites in Florida

Proximities of Launch Sites

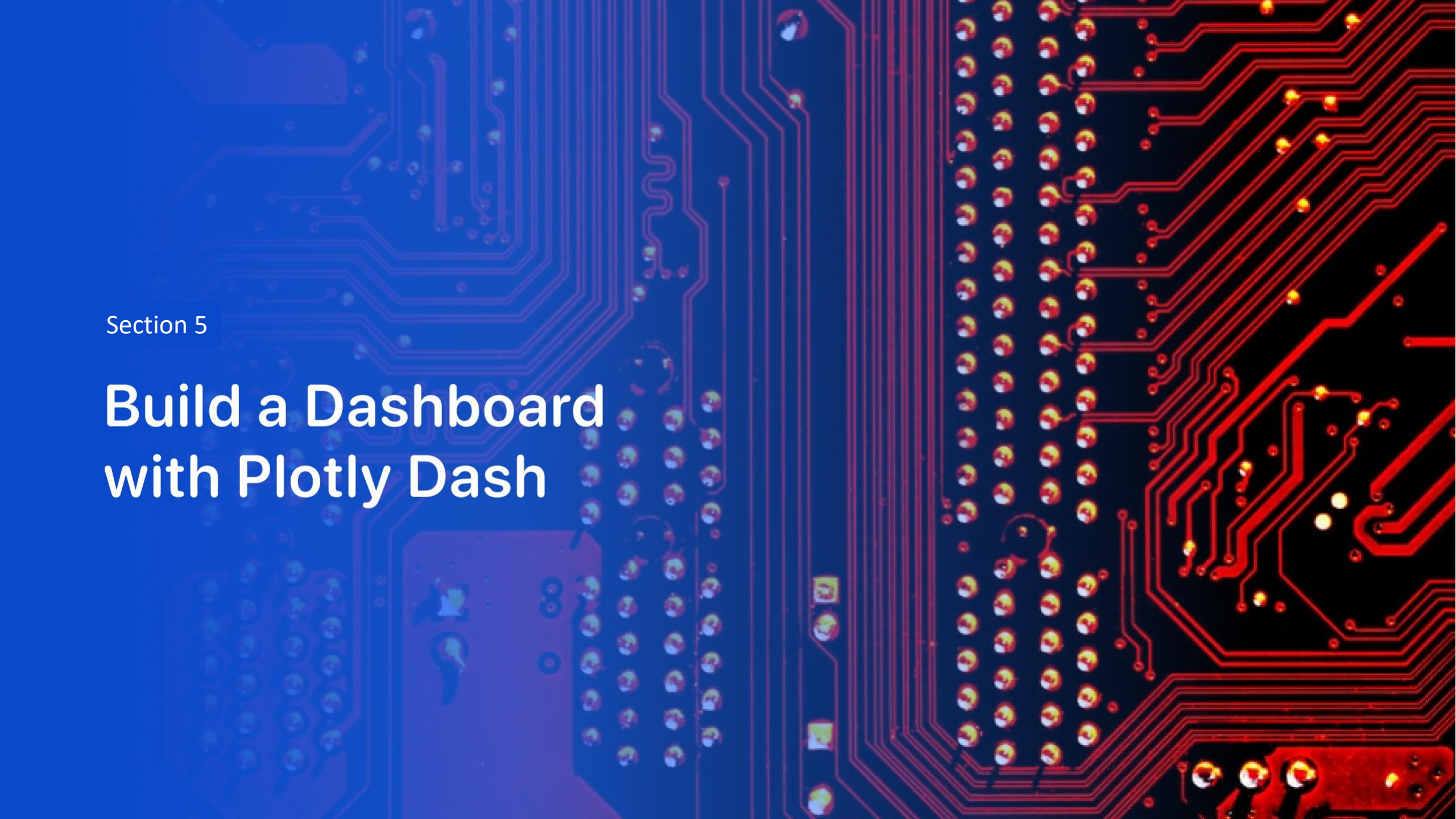


Are launch sites in close proximity to railways? **Yes**
Are launch sites in close proximity to highways? **Yes**
Are launch sites in close proximity to coastline? **Yes**

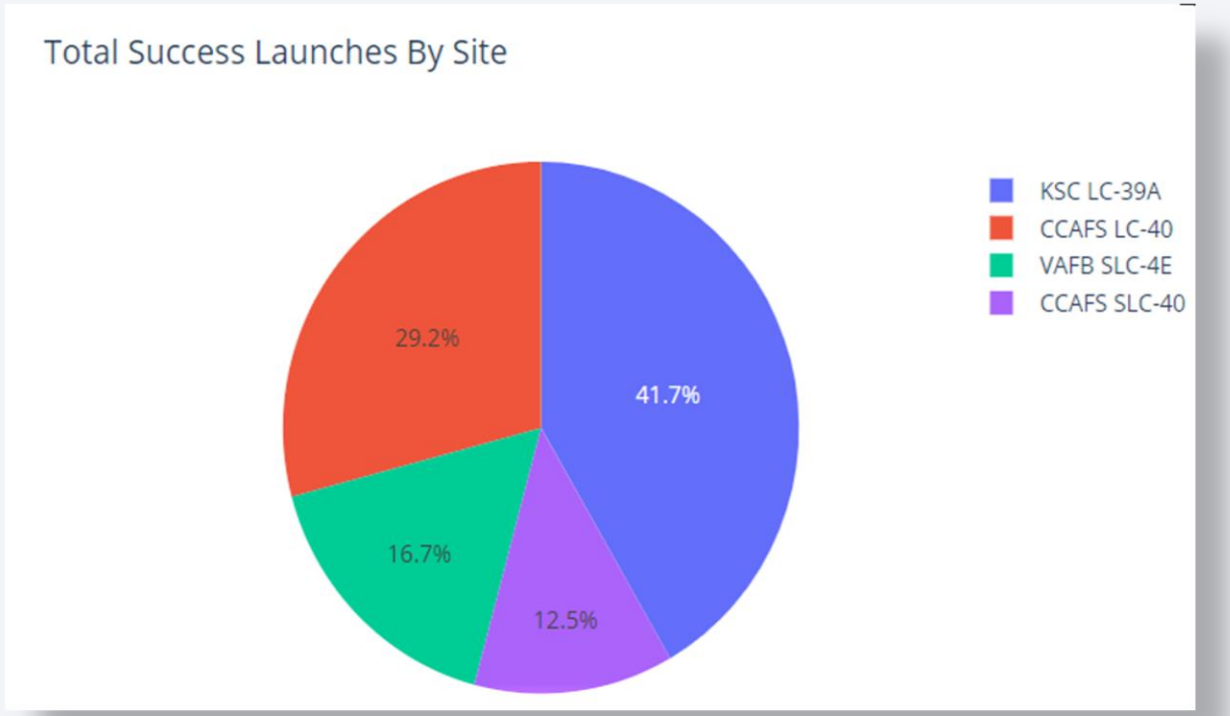


Do launch sites keep certain distance away from cities? **Yes**

- 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.

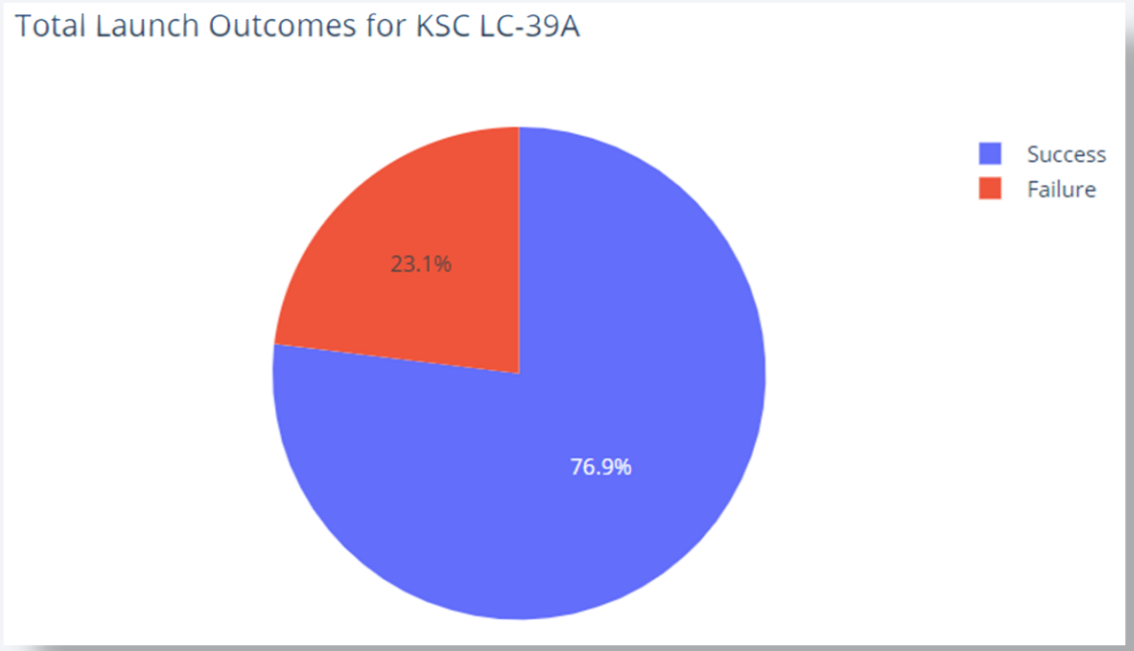

Section 5
Build a Dashboard with Plotly Dash

Total Success Launches By all sites



- KSLC-39A records the most launch success among all sites.
- The VAFB SLC-4E has the fewest launch success, possibly because
  - the data sample is small, or
  - because it is the only site located in California, so the launch difficulty on the west coast may be higher than on the east coast.

Launch Site with Highest Launch Success Ratio



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

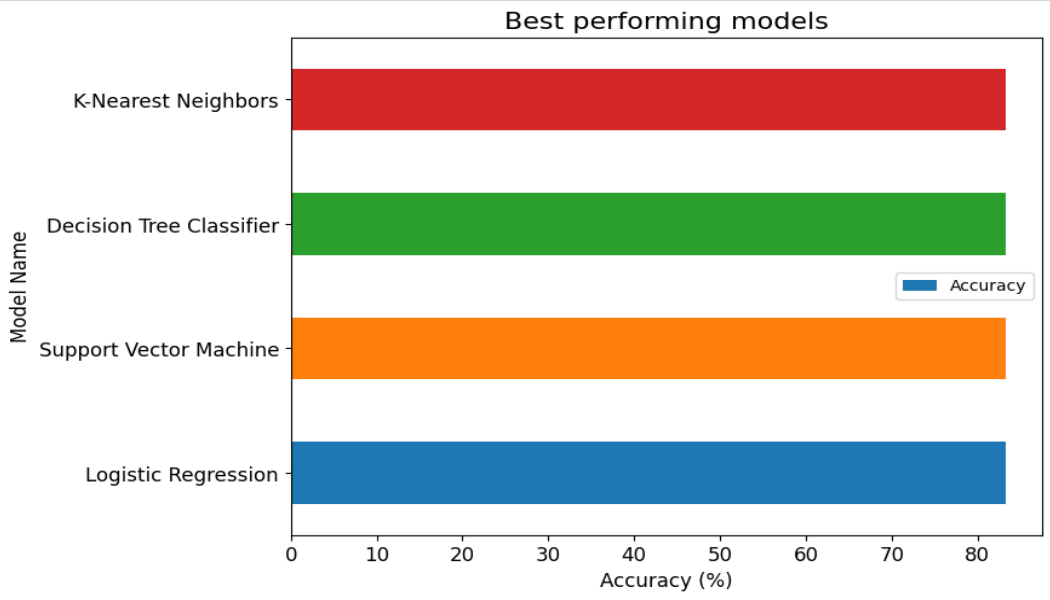
Payload vs. Launch Outcome Scatter Plot for all sites

- These figures show that **the launch success rate** (class 1) **for low weighted payloads(0-5000 kg) is higher** than that of heavy weighted payloads(5000-10000 kg).

Section 6

Predictive Analysis (Classification)

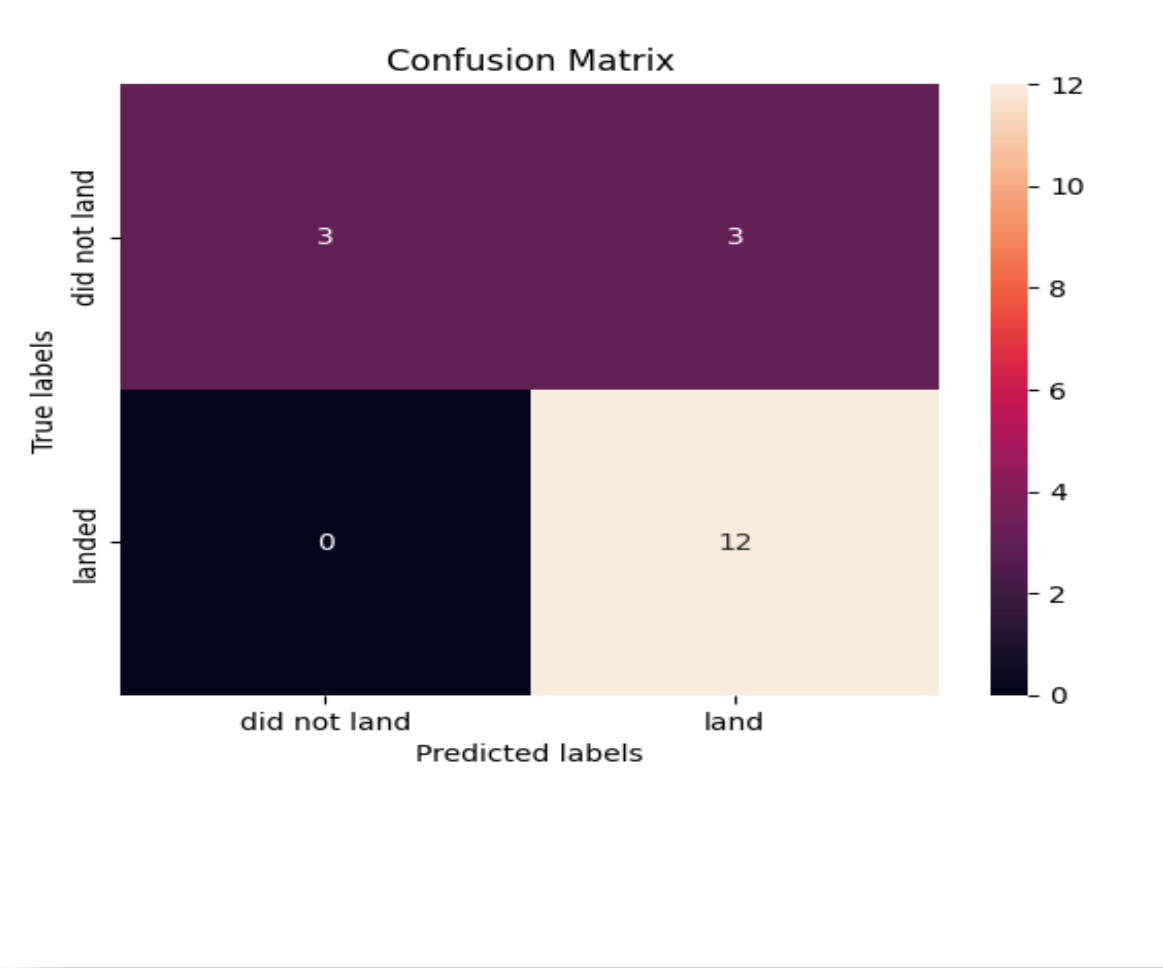
Classification Accuracy



	Model	Accuracy
0	Logistic Regression	83.333333
1	Support Vector Machine	83.333333
2	Decision Tree Classifier	83.333333
3	K-Nearest Neighbors	83.333333

- In the test set, **the accuracy of all models** was virtually the **same at 83.33%.**
- It should be noted that the test size was small at 18.
- Therefore, more data is needed to determine the optimal model.

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 were 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%), but it seems that more data is needed to determine the optimal model due to the small data size.

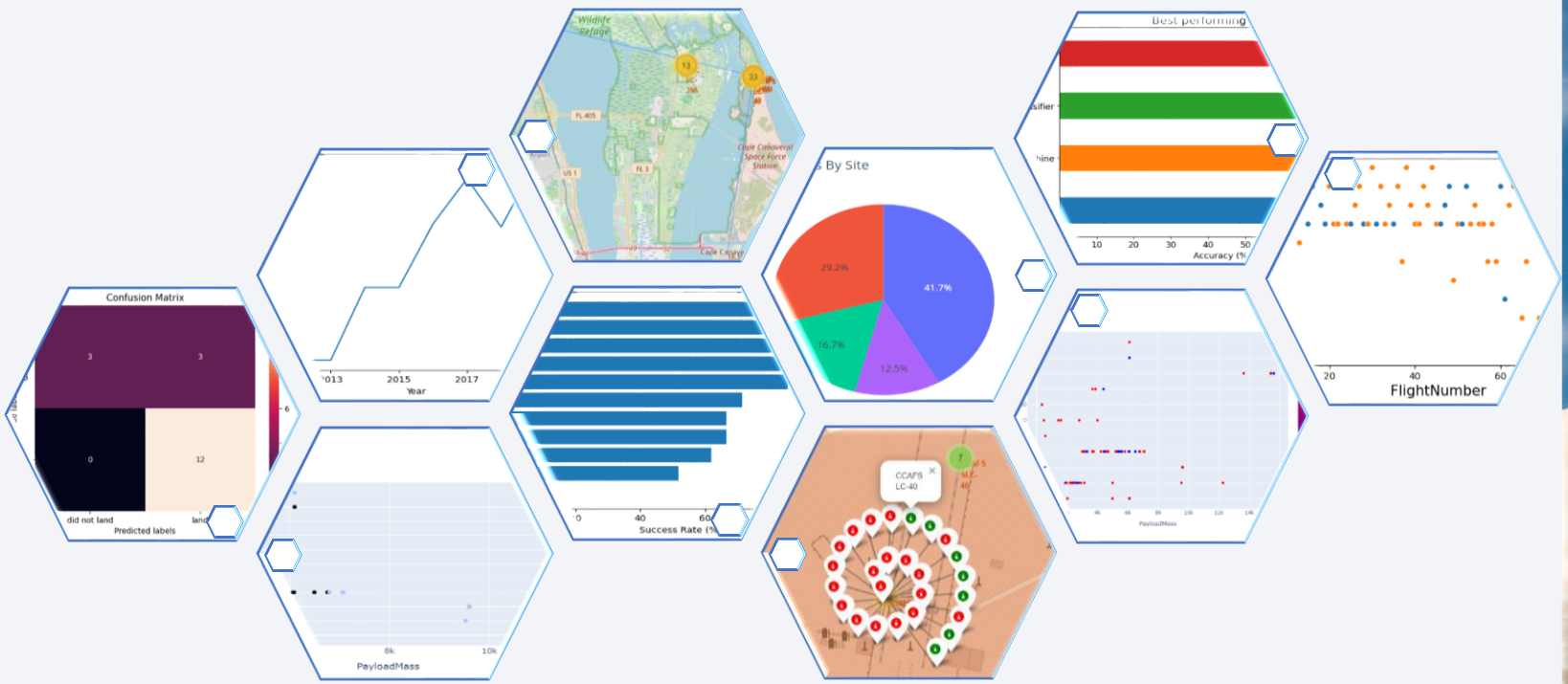
Appendix

Appendix: Key Findings

1. **Flight Success Rate**:
    1. As the number of flights increased, the success rate also improved.
    2. Recently, the success rate has exceeded 80%.
2. **Orbital Types and Success Rate**:
    1. Orbital types SSO (Sun-Synchronous Orbit), HEO (Highly Elliptical Orbit), GEO (Geostationary Orbit), and ES-L1 (Earth-Sun Lagrange Point 1) have a perfect success rate of 100%.
3. **Launch Site Location**:
    1. The launch site is strategically positioned:
        1. Close to railways, highways, and coastline for logistical convenience.
        2. Far from densely populated cities to minimize risks.
4. **Site-Specific Success**:
    1. KSLC-39A stands out:
        1. Highest number of launch successes.
        2. Highest success rate among all launch sites.
5. **Payload Weight and Success**:
    1. Low-weight payloads have a higher launch success rate compared to heavy-weight payloads.
6. **Model Accuracy**:
    1. All models in the dataset exhibit the same accuracy of 83.33%.
    2. However, due to the small data size, further data collection is necessary to determine the optimal model.

Appendix

- **GitHub URL**



Thank you!