IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

YELAMAN BAIDOL
2024.09.15

# Outline

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

# Executive Summary

- Summary of methodologies
  - Data Collection with API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Visualization
  - Interactive Visual Analysis (Folium, Plotly Dash)
- Summary of all results
  - EDA results
  - Visual analytics
  - Predictive Analysis (SVM, KNN, Logistic Regression, Classification Tree)

# 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.
- Problems you want to find answers
  - To 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


Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

- Data was collected by using GET request method from SpaceX API.
- *json_normalize* method was used to convert the json result into a dataframe.
- The API was used to get information about the launches using the IDs given for each launch.
- A Falcon 9 launch records HTML table are extracted from Wikipedia.
- In order to get HTML table, BeautifulSoup was used.

# Data Collection – SpaceX API

- Data collection with SpaceX API
- https://github.com/YelamanBaidol/IBM-COURSE-10/blob/main/data-collection-api.ipynb

## Task 1: Request and parse the SpaceX launch data using the GET request

+ Code + Markdown

To make the requested JSON results more consistent, we will use the following static response object for this project:

[9]
```python
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```
Python

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

[10]
```python
response.status_code
```
Python

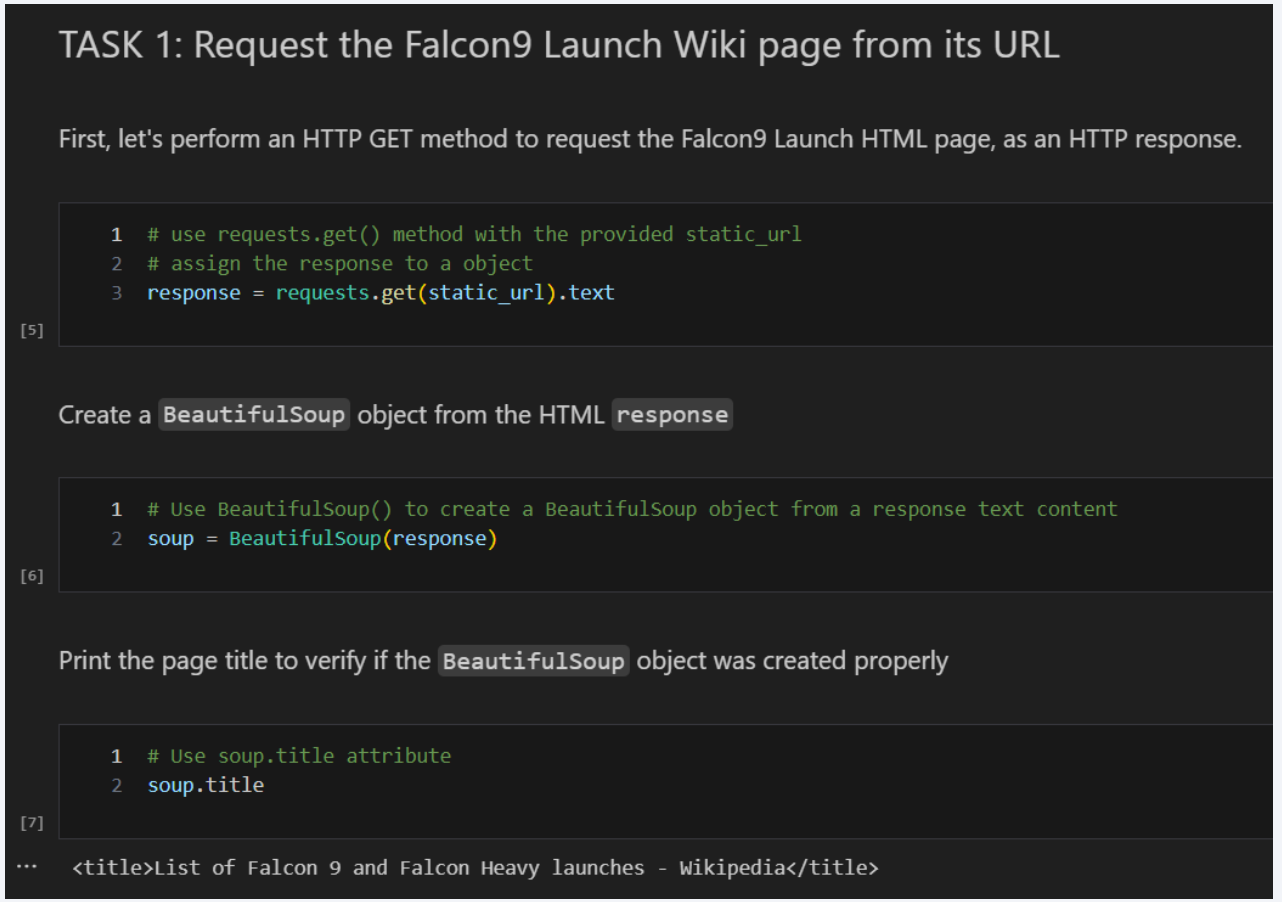
```
200
```

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

[11]
```python
# Use json_normalize meethod to convert the json result into a dataframe
response_json = response.json()
data = pd.json_normalize(response_json)
```
Python

# Data Collection - Scraping

- Web scraping with BeautifulSoup
- https://github.com/YelamanBaidol/IBM-COURSE-10/blob/main/webscraping.ipynb



TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
[5]
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

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

```
[6]
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response)
```

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

```
[7]
# Use soup.title attribute
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

# Data Wrangling

- Number of launches on each site was calculated.
- The number and occurrence of each orbit was determined.
- The number and occurrence of mission outcome of the orbits was calculated.
- https://github.com/YelamanBaidol/IBM-COURSE-10/blob/main/Data%20wrangling.ipynb

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome`

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for out in df['Outcome']:
    if out in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)

# landing_class = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
```

This variable will represent the classification variable that represents the outcome of each launch.
Successfully

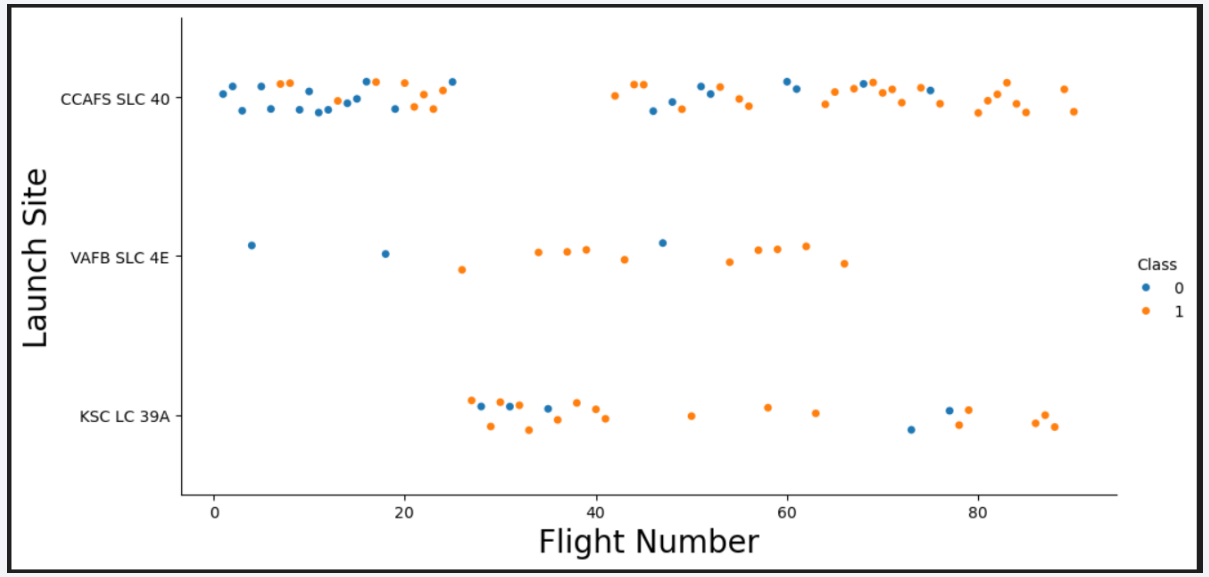
```
df['Class']=landing_class
df[['Class']].head(8)
```

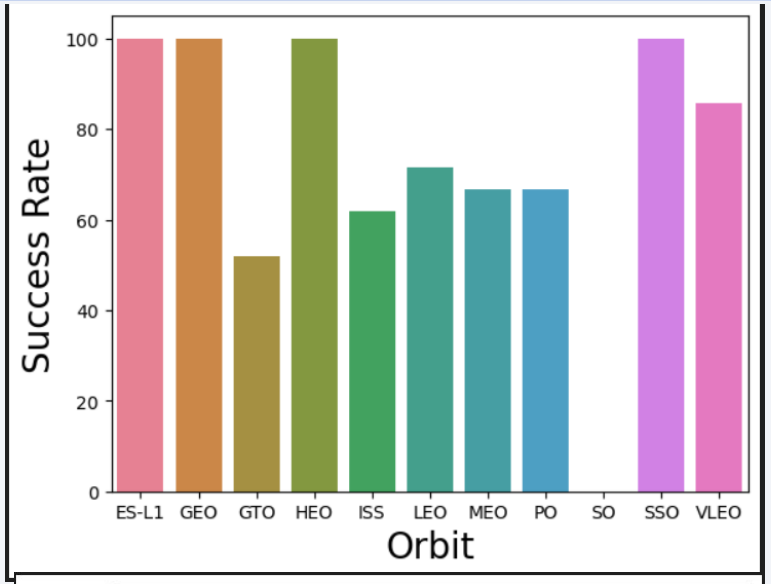
| | Class |
|---|---|
| 0 | 0 |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0 |
| 6 | 1 |
| 7 | 1 |

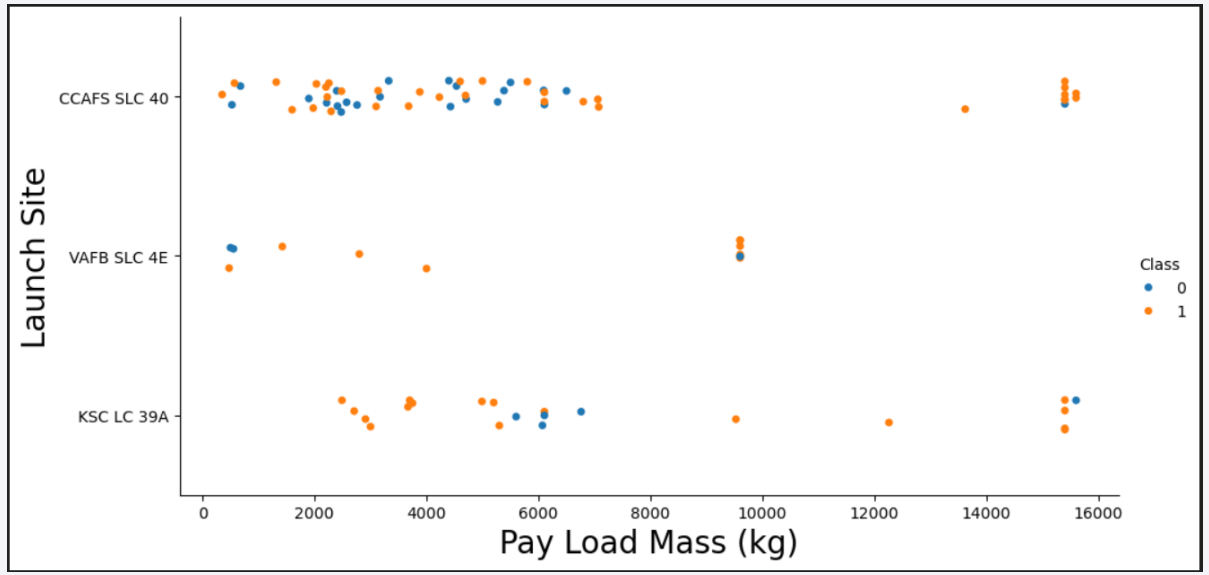
# EDA with Data Visualization

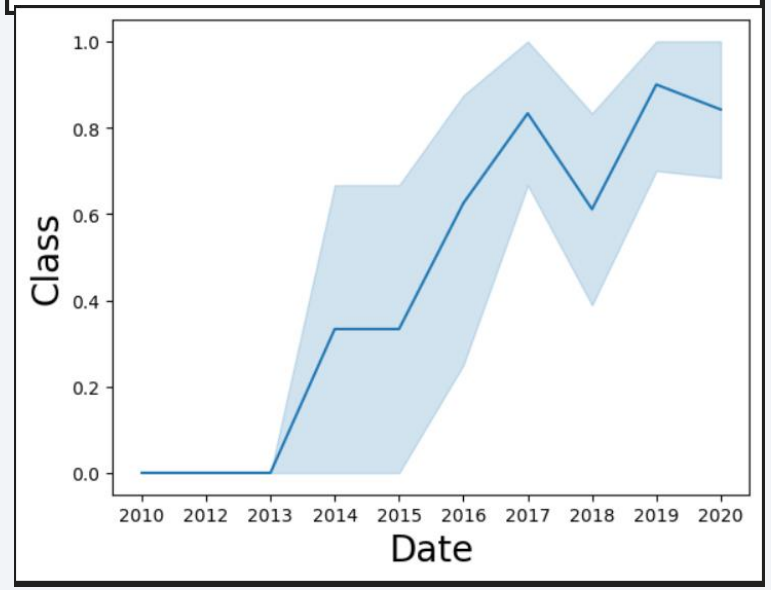
- Exploratory Data Analysis and Feature Engineering were performed using `Pandas` and `Matplotlib`.
- Relationship between
  - Flight number and Launch Site
  - Flight number and Payload Mass
  - Payload Mass and Launch Site
  - Success rate of each orbit type
  - Flight number and Orbit type
  - Payload Mass and Orbit type
  - Launch yearly trend were visualized to analyze the data.
- https://github.com/YelamanBaidol/IBM-COURSE-10/blob/main/edadataviz.ipynb

# EDA with Data Visualization



# EDA with SQL

- SQL queries were performed to understand the data set.
- https://github.com/YelamanBaidol/IBM-COURSE-10/blob/main/eda-sql-coursera_sqllite.ipynb

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

```
1  %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
```

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

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

# EDA with SQL

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

+ Code + Markdown

```
1  %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5
```

Python

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

| 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 |

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

+ Code + Markdown

```
1  %sql SELECT SUM("PAYLOAD_MASS__KG_") as "Total Payload Mass", "Customer" FROM SPACEXTABLE WHERE Customer = "NASA (CRS)"
```

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

| Total Payload Mass | Customer |
|---|---|
| 45596 | NASA (CRS) |

# EDA with SQL

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

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") as "Average Payload Mass", "Booster_Version" FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

*Hint:Use min function*

```
%sql SELECT MIN("Date") FROM SPACEXTABLE WHERE Landing_Outcome = "Success (ground pad)"
```

List 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 SPACEXTABLE WHERE "Landing_Outcome" = "Success (drone ship)" and ("PAYLOAD_MASS__KG_" > 4000 and "PAYLOAD_MASS__KG_" < 6000)
```

List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") FROM SPACEXTABLE GROUP BY "Mission_Outcome"
```

# EDA with SQL

- The count of landing outcomes between the date 2010-06-04 and 2017-03-20

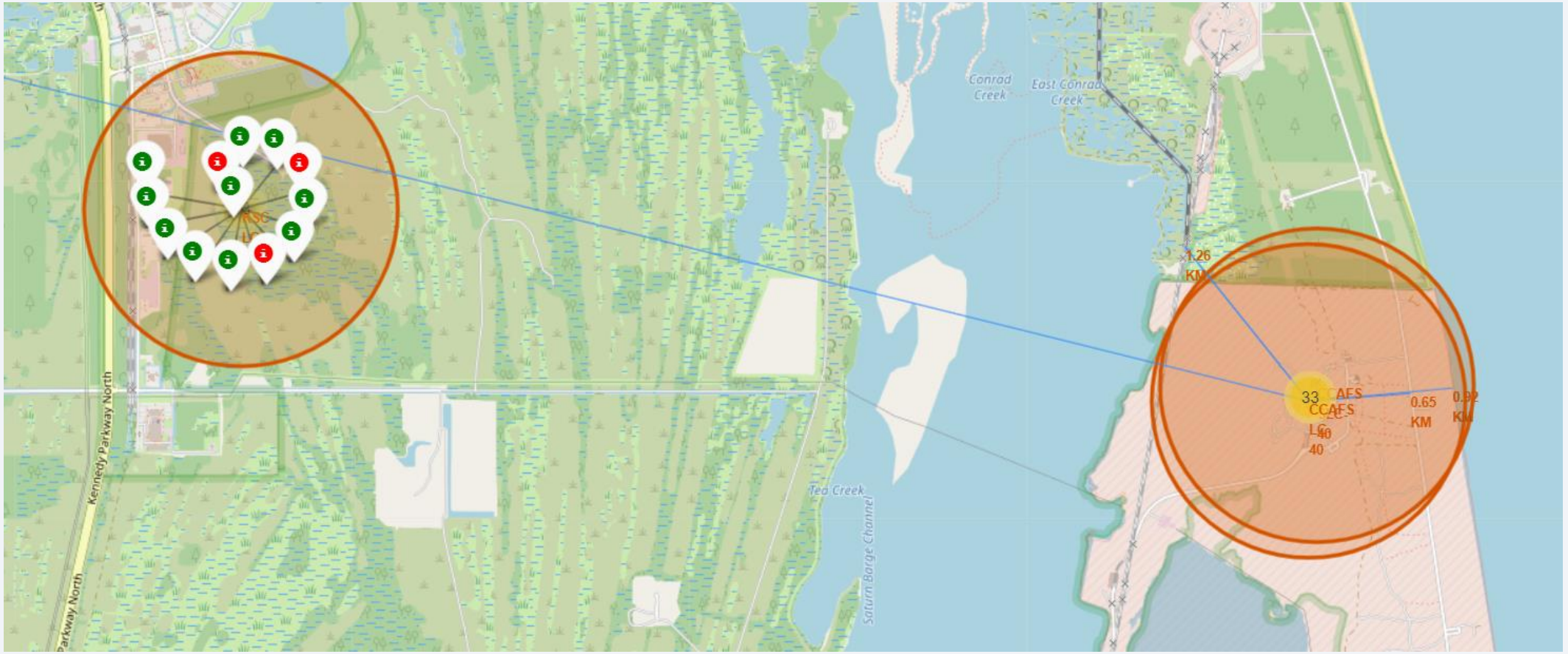
| Date | Landing_Outcome | Count |
|---|---|---|
| 2012-05-22 | No attempt | 10 |
| 2016-04-08 | Success (drone ship) | 5 |
| 2015-01-10 | Failure (drone ship) | 5 |
| 2015-12-22 | Success (ground pad) | 3 |
| 2014-04-18 | Controlled (ocean) | 3 |
| 2013-09-29 | Uncontrolled (ocean) | 2 |
| 2010-06-04 | Failure (parachute) | 2 |
| 2015-06-28 | Precluded (drone ship) | 1 |

# Build an Interactive Map with Folium

- All launch sites were marked on a map.
- The success/failure launches for each site were marked on a map.
- The distances between a launch site to its proximities were calculated.
- https://github.com/YelamanBaidol/IBM-COURSE-10/blob/main/launch_site_location.ipynb

| | Launch Site | Lat | Long |
|---|---|---|---|
| 0 | CCAFS LC-40 | 28.562302 | -80.577356 |
| 1 | CCAFS SLC-40 | 28.563197 | -80.576820 |
| 2 | KSC LC-39A | 28.573255 | -80.646895 |
| 3 | VAFB SLC-4E | 34.632834 | -120.610745 |

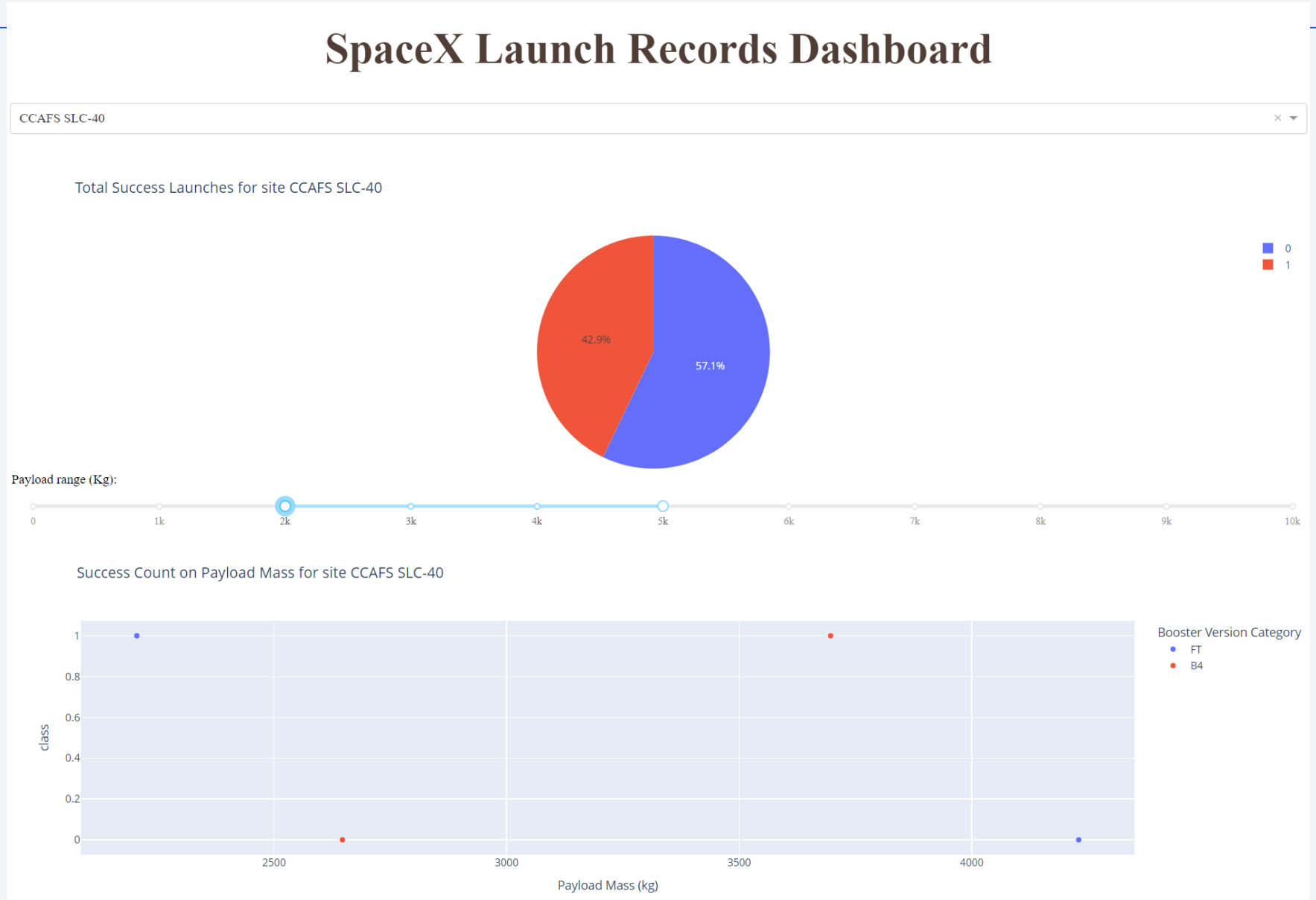
# Build an Interactive Map with Folium



# Build a Dashboard with Plotly Dash

- A Plotly Dash application was built to perform interactive visual analytics on SpaceX launch data in real-time.
- A Launch Site Drop-down Input Component was added.
- Success-pie-chart based on selected site dropdown was added.
- The success-payload-scatter-chart scatter plot was plotted.
- https://github.com/YelamanBaidol/IBM-COURSE-10/blob/main/spacex_dash_app.py

# Build a Dashboard with Plotly Dash



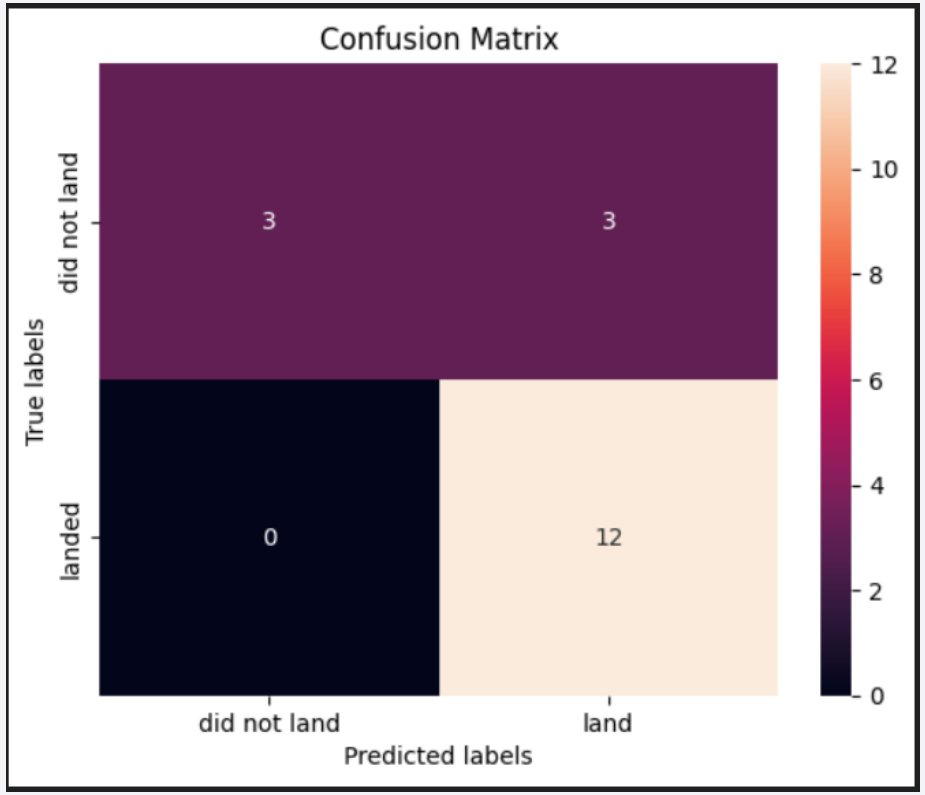
# Predictive Analysis (Classification)

- Support Vector Machine, K Nearest Neighbor, Logistic Regression, Classification tree methods were used.
- GridSearchCV method was used to find best hyperparameters for the methods above.
- Before applying the methods, the data was split into training and test data.
- Then, best hyperparameters were determined and the accuracy scores were calculated.
- https://github.com/YelamanBaidol/IBM-COURSE-10/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

# Predictive Analysis (Classification)

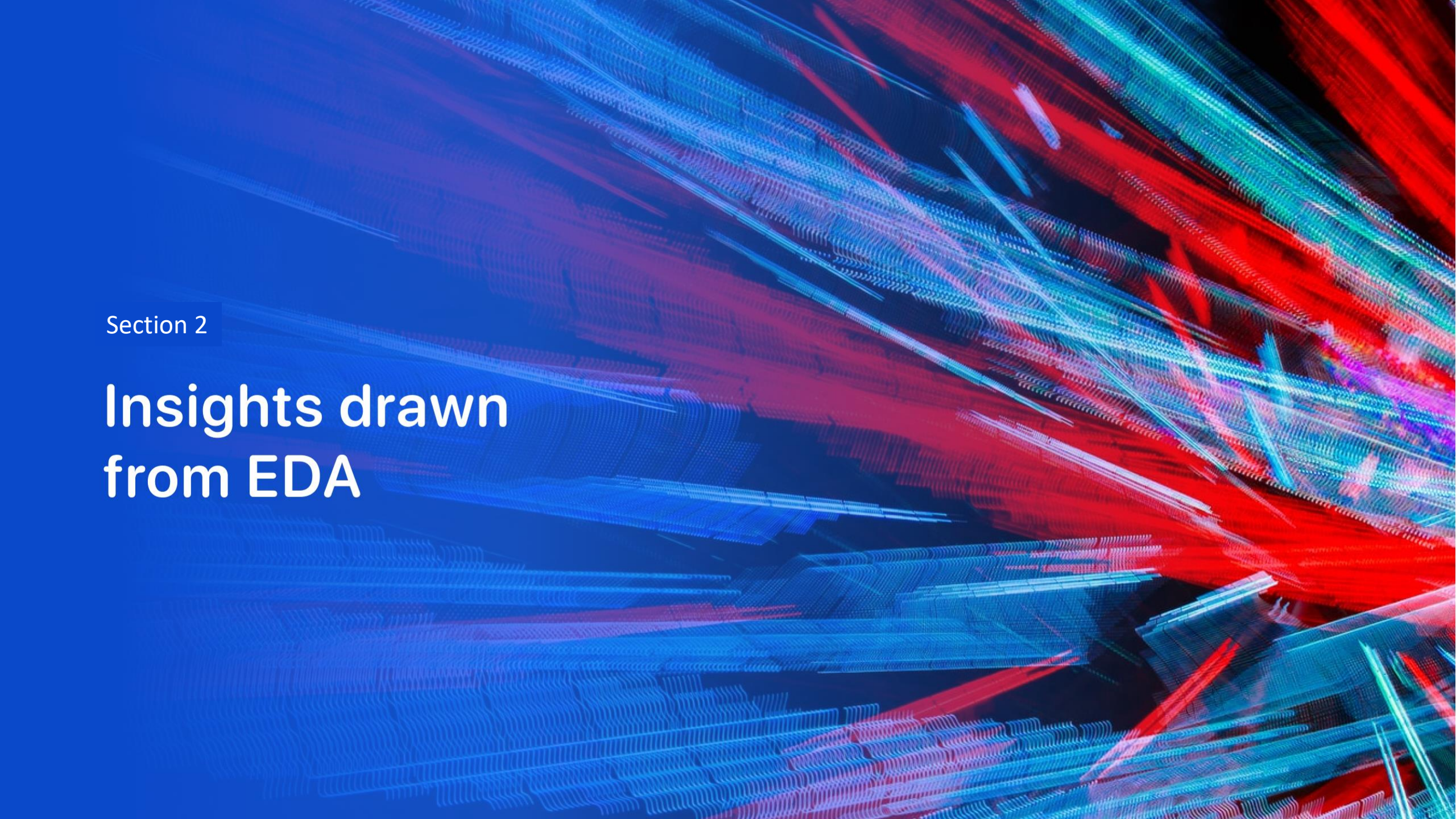
- Confusion matrices were plotted.
- As an example, confusion matrix for classification tree method.
- Test Data Accuracy was shown.

| | Method | Test Data Accuracy |
|---|---|---|
| 0 | Logistic Reg. | 0.833333 |
| 1 | SVM | 0.833333 |
| 2 | Decision Tree | 0.833333 |
| 3 | KNN | 0.833333 |



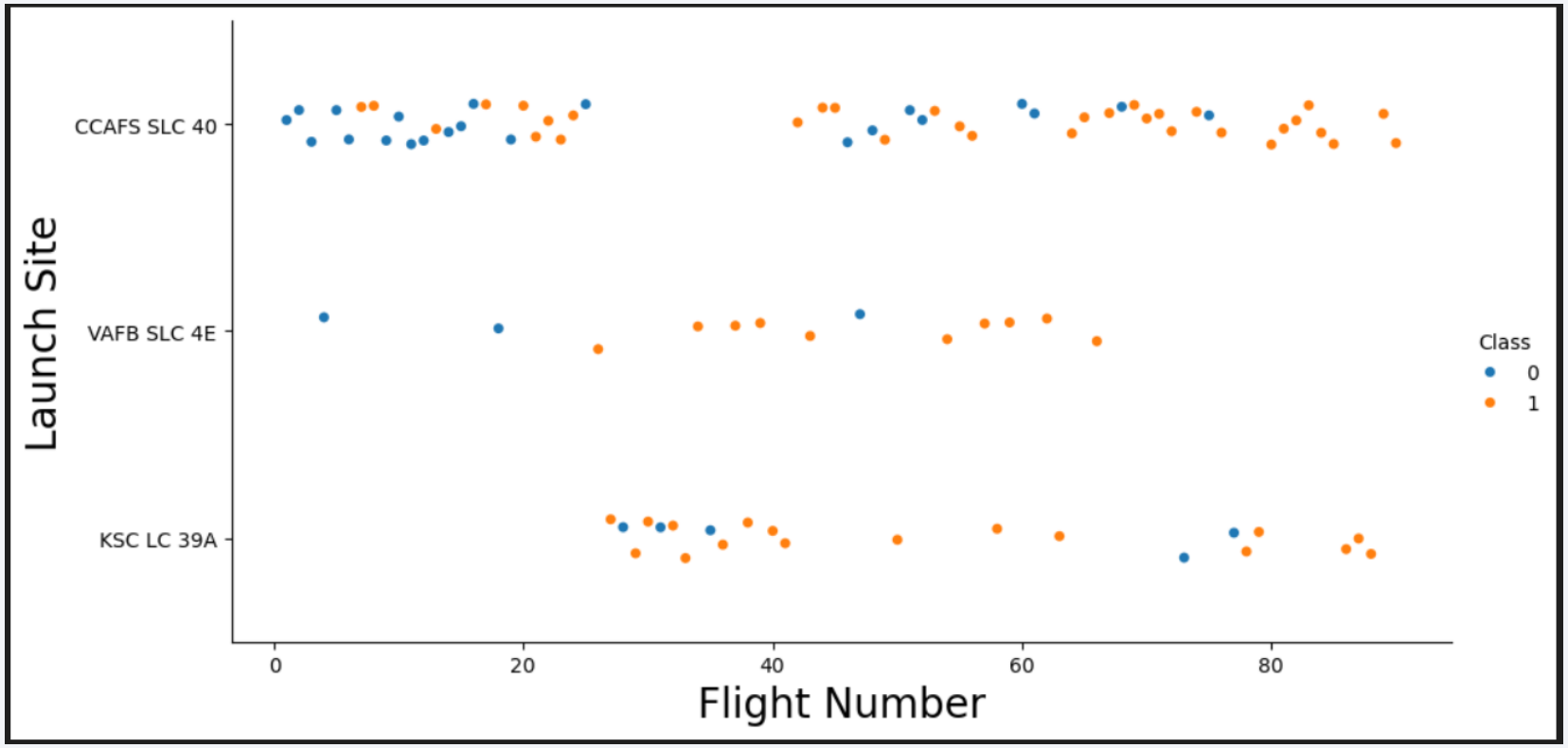
# Results

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


Section 2
Insights drawn from EDA

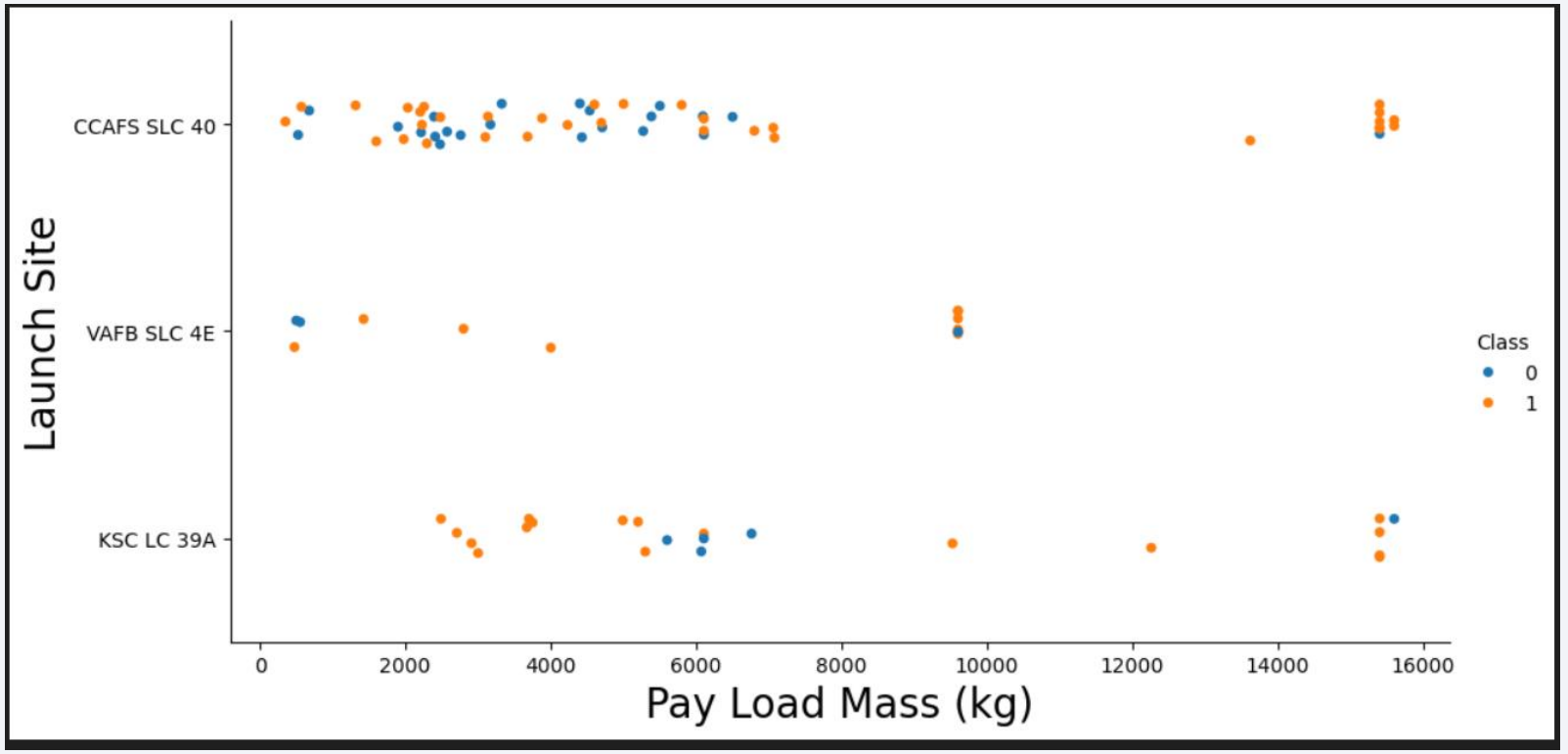
# Flight Number vs. Launch Site

- After flight number 80, for all the launch sites, success rate was 100%. However, flight number between 0 and 20, success rate was too low.



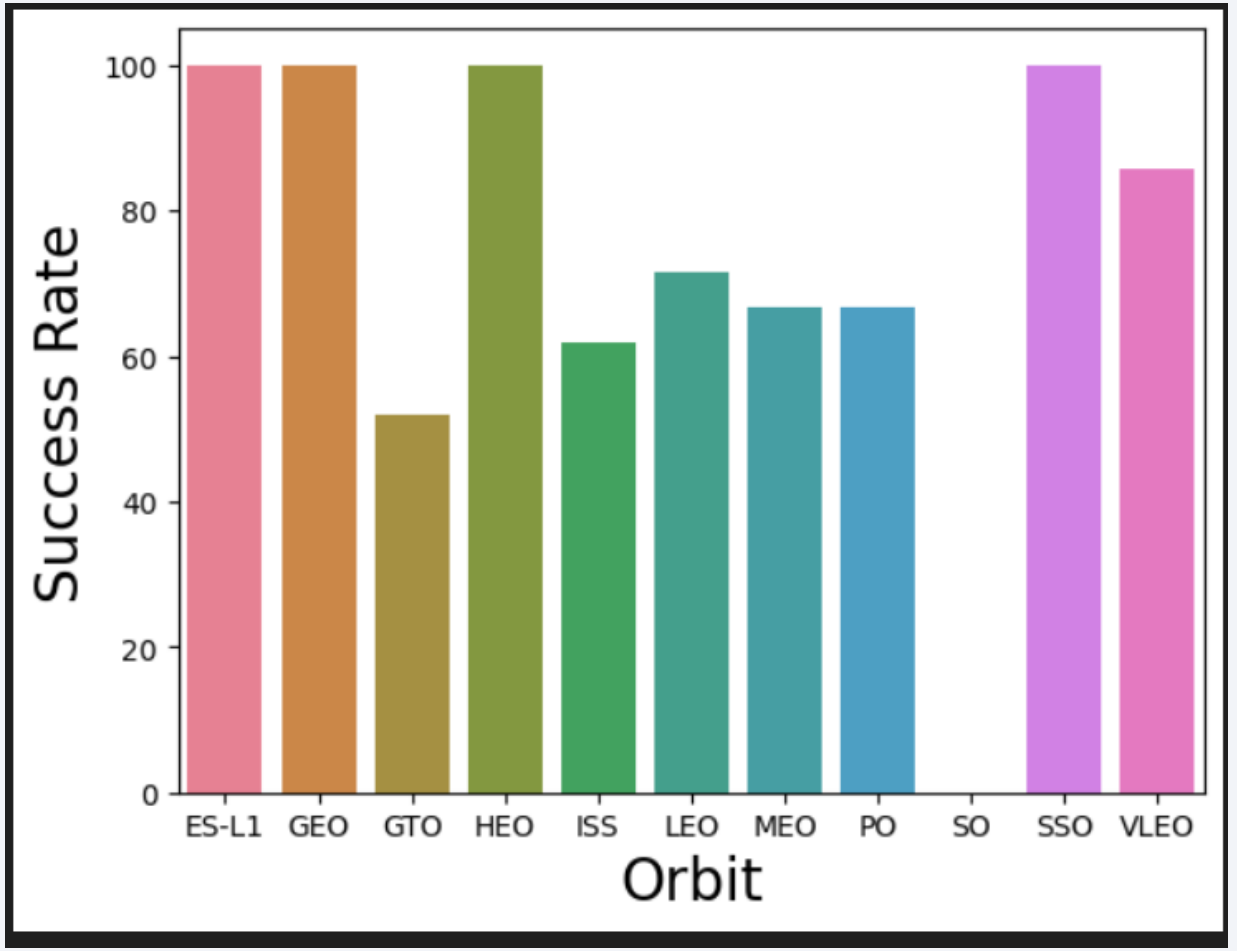
# Payload vs. Launch Site

- From Payload Mass Vs. Launch Site scatter point chart, for the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000).



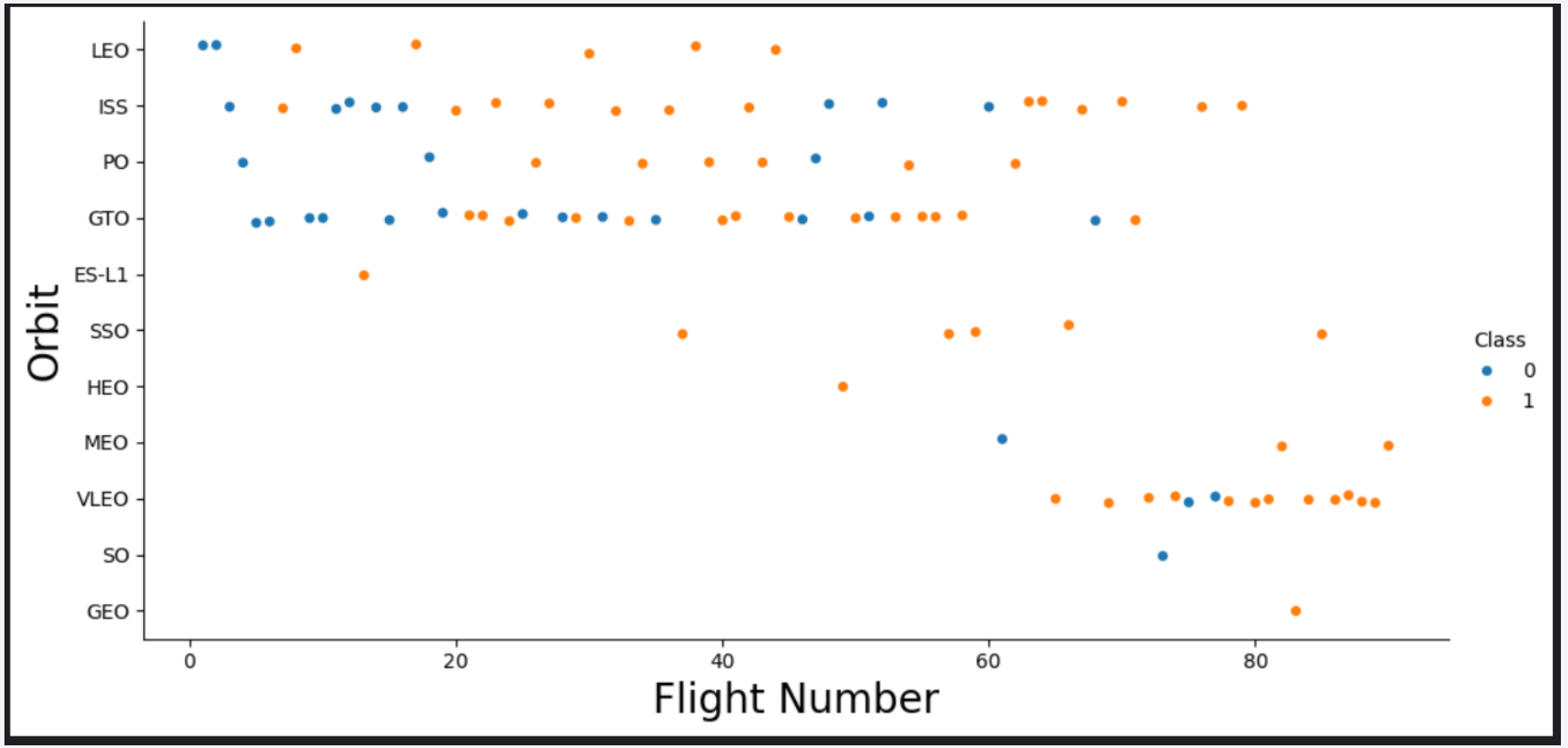
# Success Rate vs. Orbit Type

- From the bar chart, it can be seen that for the orbit type of ES-L1, GEO, HEO and SSO, the success rate was 100%. On the other hand, for SO, it was 100% failure.



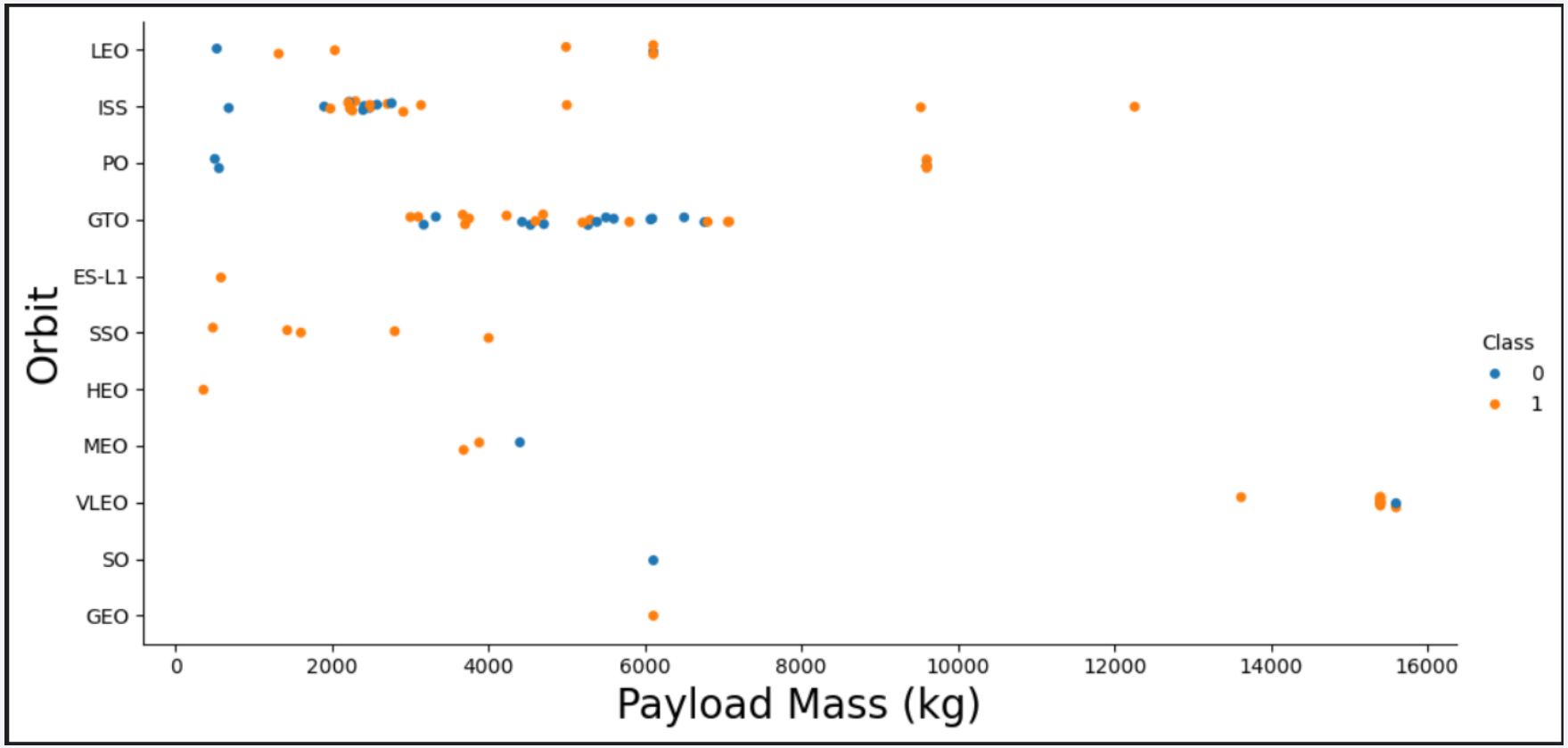
# Flight Number vs. Orbit Type

- In the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO, ISS orbits, there appears to be no relationship between flight number and success.



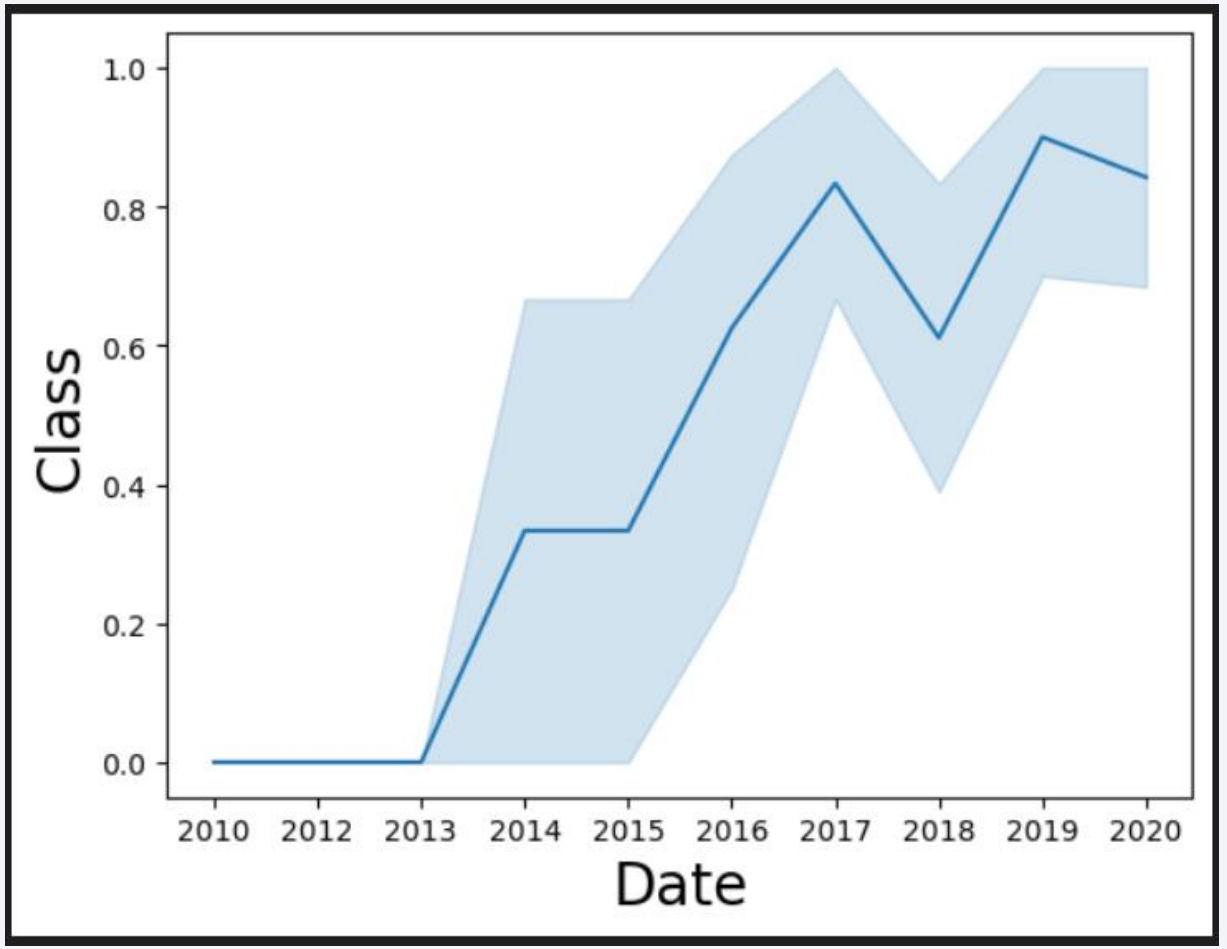
# Payload vs. Orbit Type

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



# Launch Success Yearly Trend

- Success rate kept increasing since 2013 until 2020.



# All Launch Site Names

- The names of the unique launch sites

```
1  %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
```

```
 * 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'

- 5 records where launch sites begin with `CCA`

```
1  %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5
```

Python

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

| 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

```
1  %sql SELECT SUM("PAYLOAD_MASS__KG_") as "Total Payload Mass", "Customer" FROM SPACEXTABLE WHERE Customer = "NASA (CRS)"
```

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

| Total Payload Mass | Customer |
|---|---|
| 45596 | NASA (CRS) |

# Average Payload Mass by F9 v1.1

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

```
1  %sql SELECT AVG("PAYLOAD_MASS__KG_") as "Average Payload Mass", "Booster_Version" FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

| Average Payload Mass | Booster_Version |
|---|---|
| 2534.6666666666665 | F9 v1.1 B1003 |

# First Successful Ground Landing Date

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

```
1  %sql SELECT MIN("Date") FROM SPACEXTABLE WHERE Landing_Outcome = "Success (ground pad)"
```

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

| MIN("Date") |
| --- |
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
1  %sql SELECT "Booster_Version" FROM SPACEXTABLE 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

- The total number of successful and failure mission outcomes

```
1  %sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") FROM SPACEXTABLE GROUP BY "Mission_Outcome"
```

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

| Mission_Outcome | COUNT("Mission_Outcome") |
|---|---|
| Failure (in flight) | 1 |
| Success | 98 |
| Success | 1 |
| Success (payload status unclear) | 1 |

# Boosters Carried Maximum Payload

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

```
1  %sql SELECT DISTINCT "Booster_Version", "PAYLOAD_MASS__KG_" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
```

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

| Booster_Version | PAYLOAD_MASS__KG_ |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1060.3 | 15600 |
| F9 B5 B1049.7 | 15600 |

# 2015 Launch Records

- The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- `%sql SELECT substr(Date, 6,2) as "Month", substr(Date,0,5) as "Year", "Booster_Version", "Launch_Site", "Landing_Outcome" FROM SPACEXTABLE WHERE "Landing_Outcome" = "Failure (drone ship)" and substr(Date,0,5) = "2015"`

| Month | Year | Booster_Version | Launch_Site | Landing_Outcome |
|---|---|---|---|---|
| 01 | 2015 | F9 v1.1 B1012 | CCAFS LC-40 | Failure (drone ship) |
| 04 | 2015 | F9 v1.1 B1015 | CCAFS LC-40 | Failure (drone ship) |

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

- The count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

- `%sql SELECT "Date", "Landing_Outcome", COUNT(*) as "Count" FROM SPACEXTABLE WHERE ("Date" BETWEEN "2010-06-04" and "2017-03-20") GROUP BY "Landing_Outcome" ORDER BY "Count" desc`

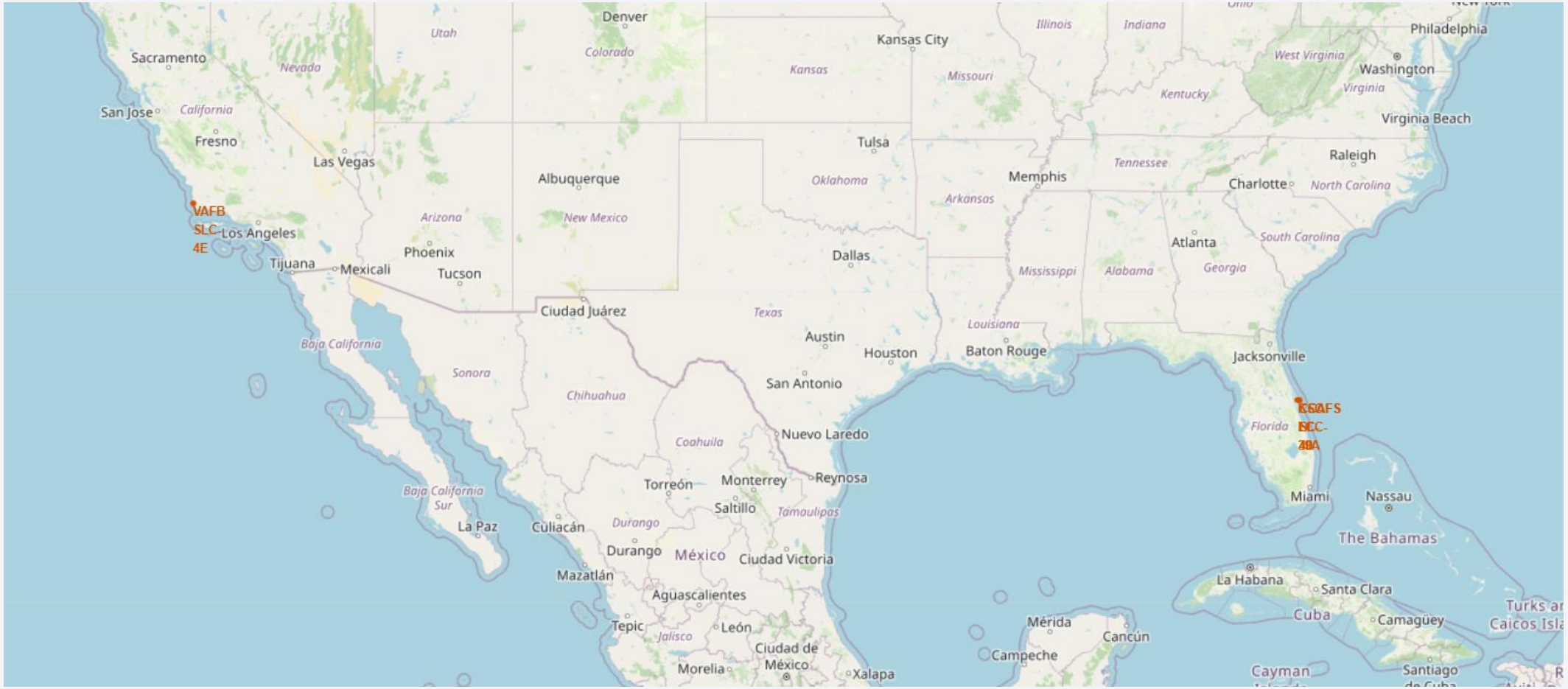
| Date | Landing_Outcome | Count |
|---|---|---|
| 2012-05-22 | No attempt | 10 |
| 2016-04-08 | Success (drone ship) | 5 |
| 2015-01-10 | Failure (drone ship) | 5 |
| 2015-12-22 | Success (ground pad) | 3 |
| 2014-04-18 | Controlled (ocean) | 3 |
| 2013-09-29 | Uncontrolled (ocean) | 2 |
| 2010-06-04 | Failure (parachute) | 2 |
| 2015-06-28 | Precluded (drone ship) | 1 |

Section 3

# Launch Sites Proximities Analysis

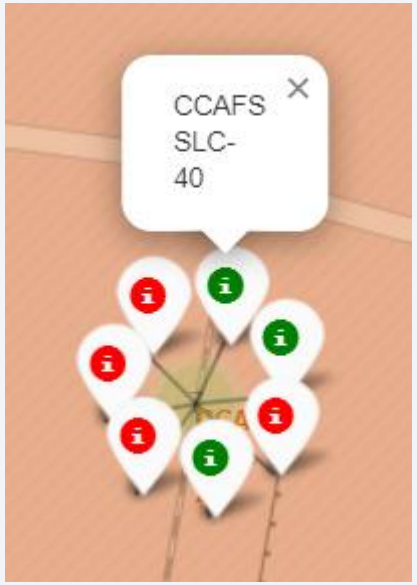
# Markers of all launch sites

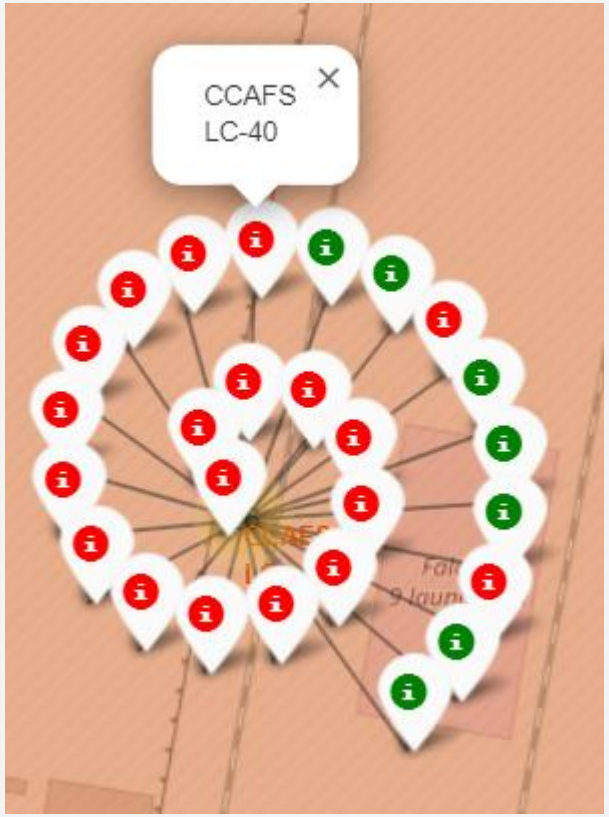
Launch sites are very close to the coast. Three of the sites are on the east part of the US.



# Launch Outcomes on a map

- From the markers, it can be seen that Launch Site KSC LC-39A had the highest success rate.

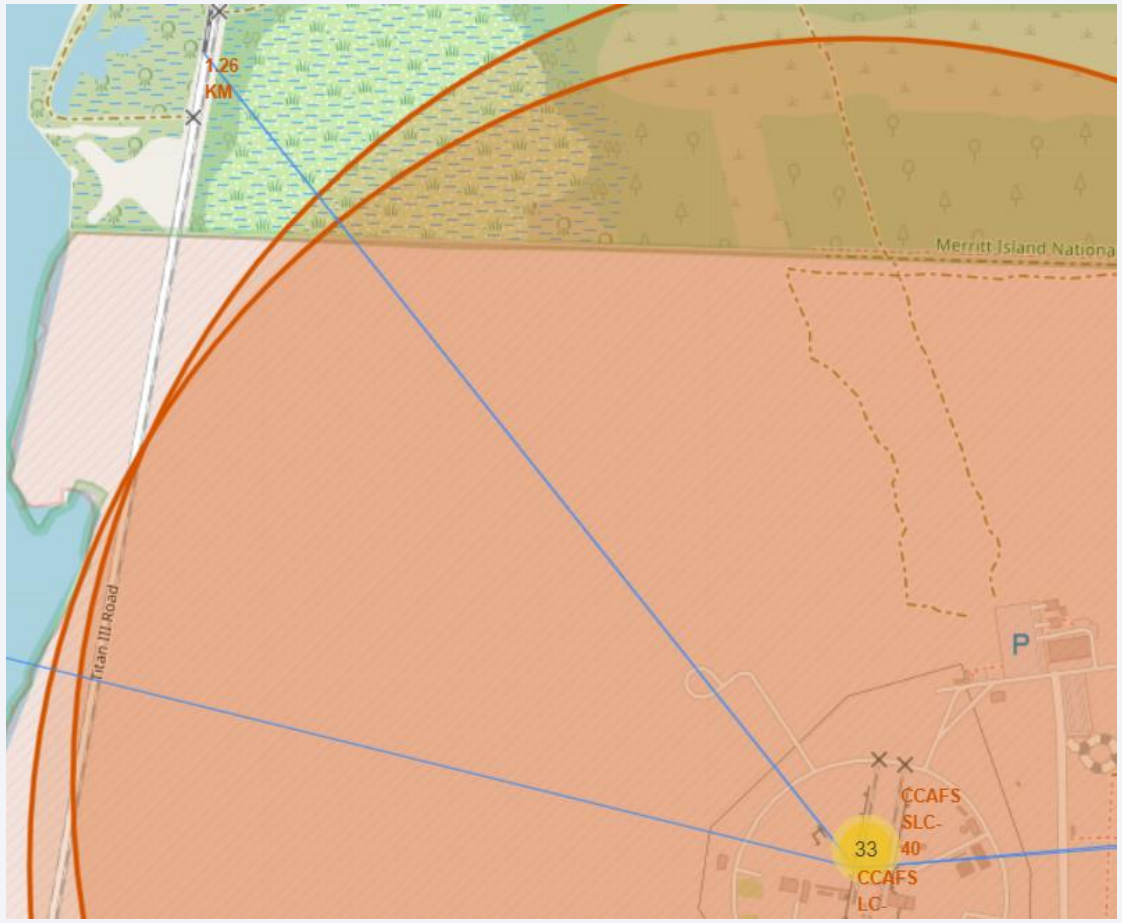



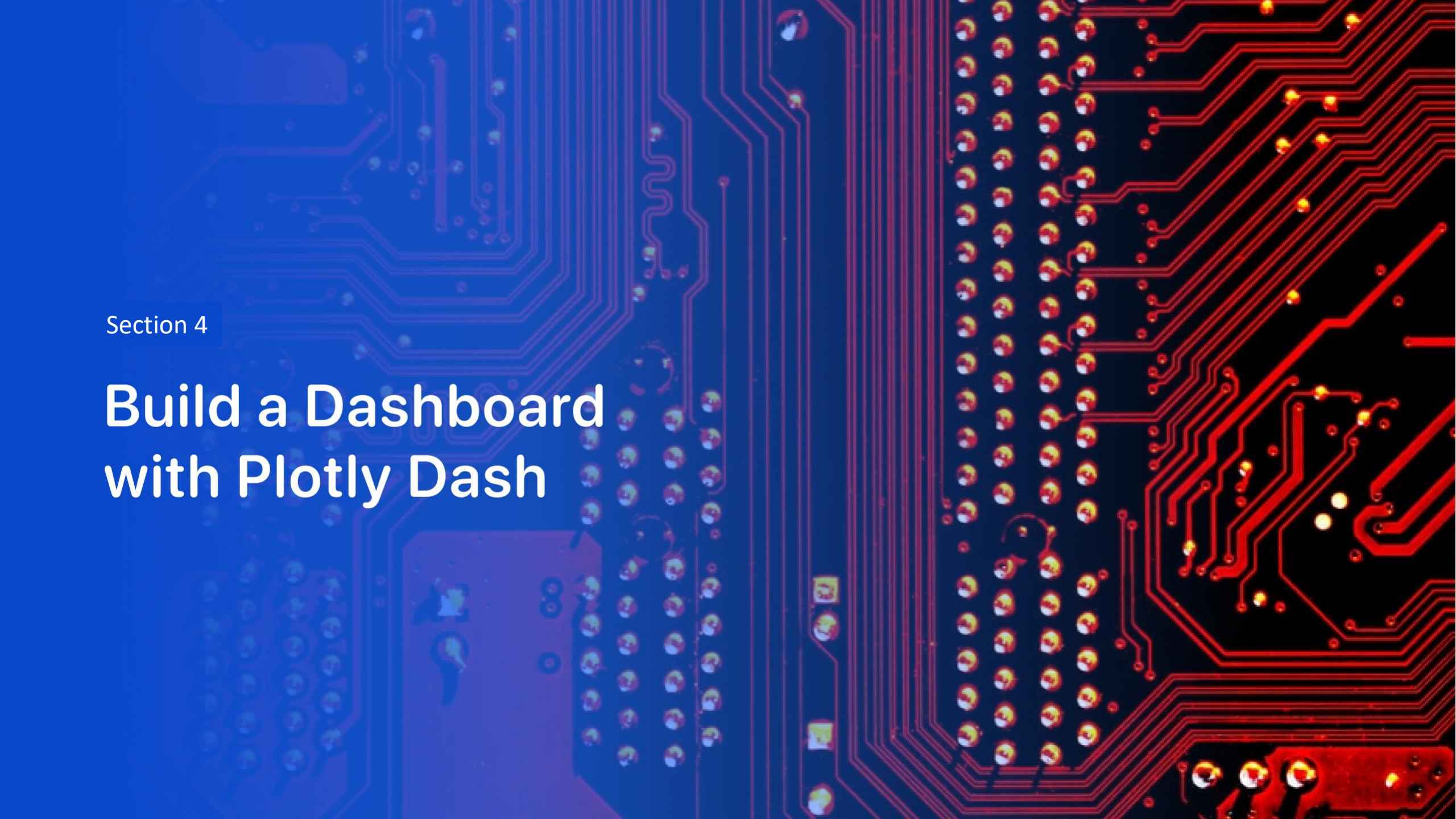

# Launch site CCAFS LC-40 and its proximities

- Launch Site CCAFS LC-40 is in close proximity to railway, highway and coastline.

# Launch site CCAFS LC-40 and its proximities

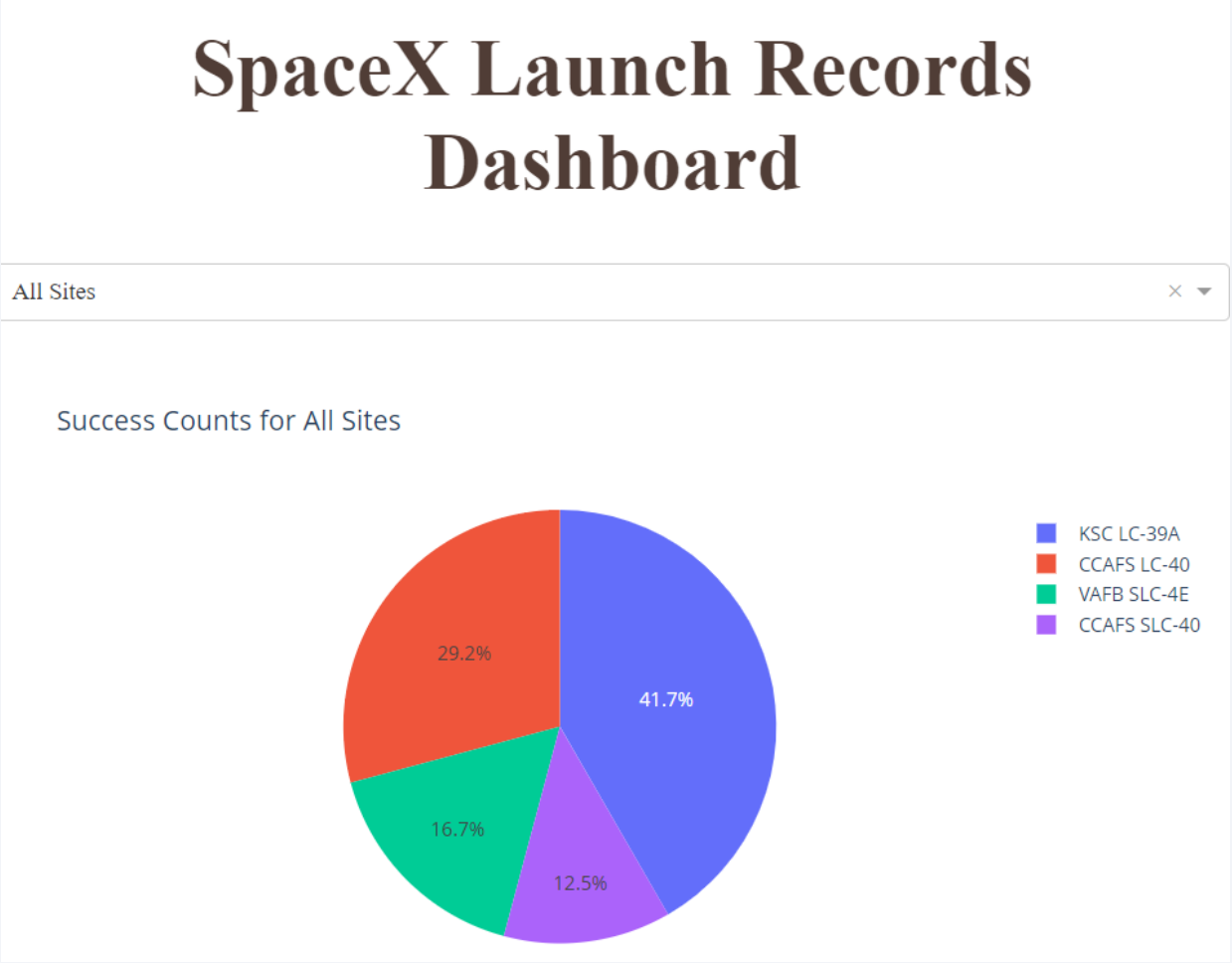
- Launch Site CCAFS LC-40 is in close proximity to railway, highway and coastline.
- But the closest city, Titusville, is 23 km away from the site.





Section 4
Build a Dashboard with Plotly Dash

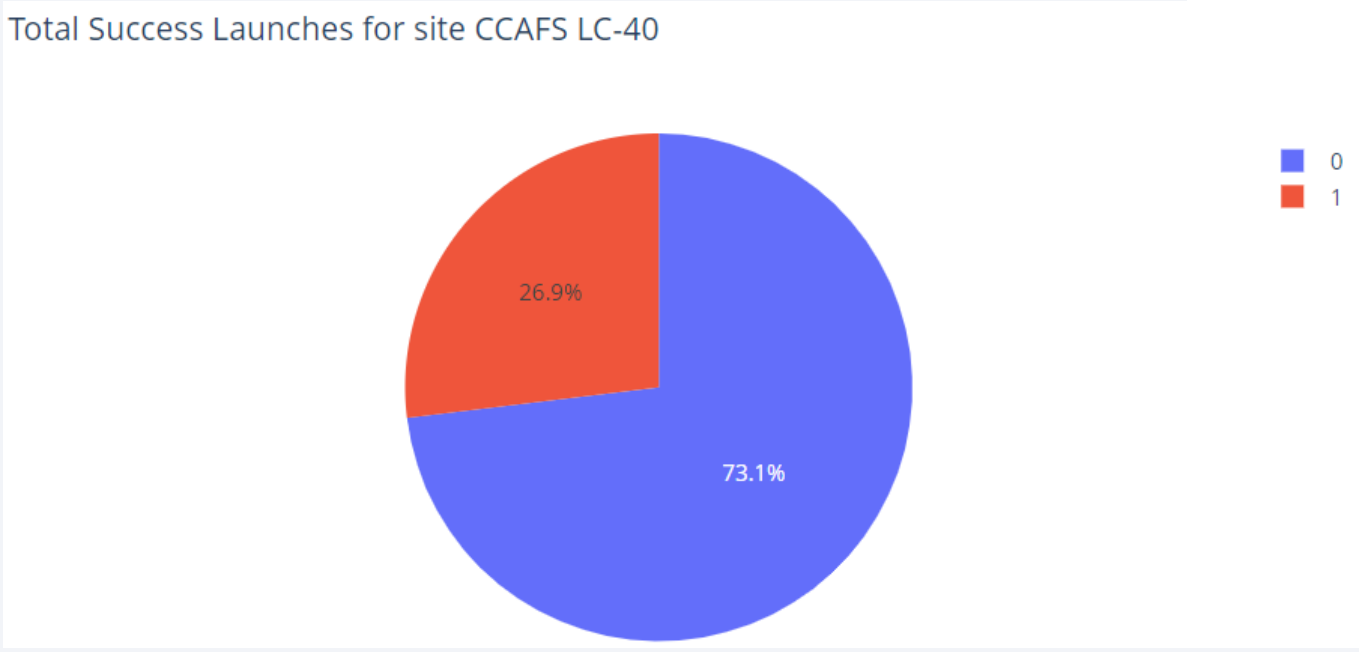
# Success Rate Pie Plot for All Launch Sites

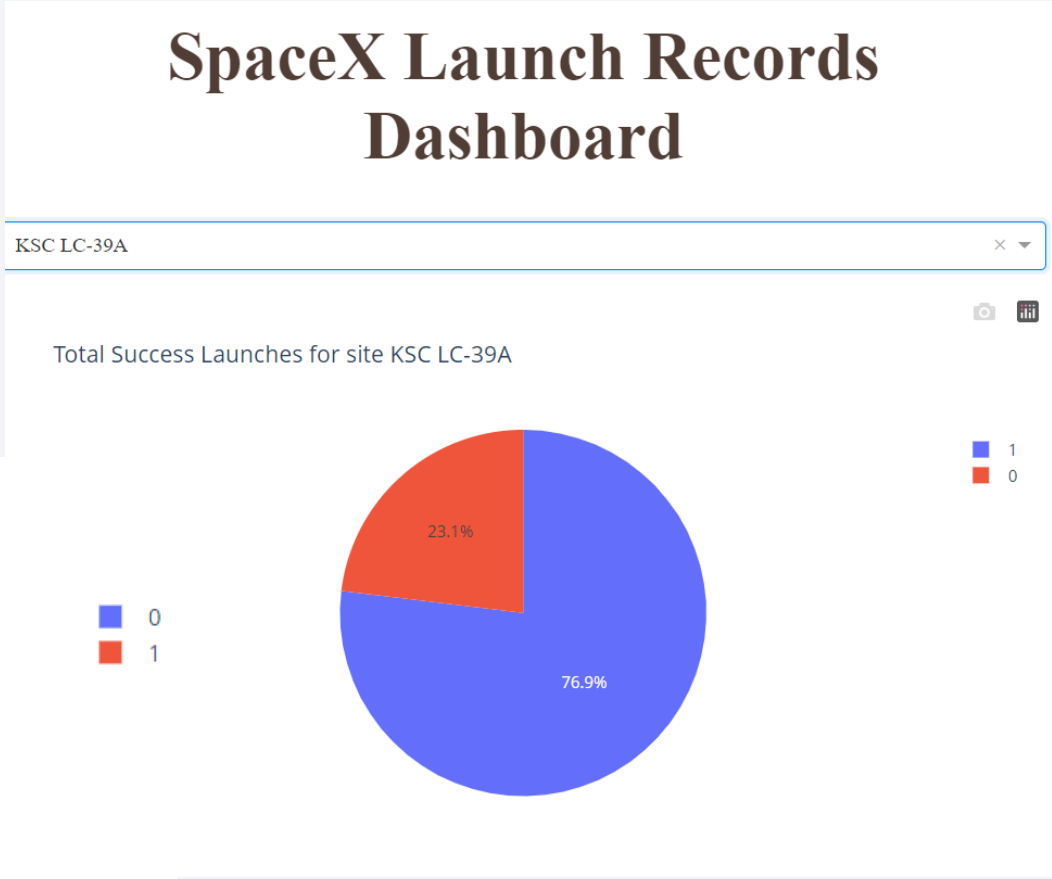
- It can be seen that KSC LC-39A had the highest success rate, while CCAFS SLC-40 had the lowest one.



# Success Rate Pie Plot

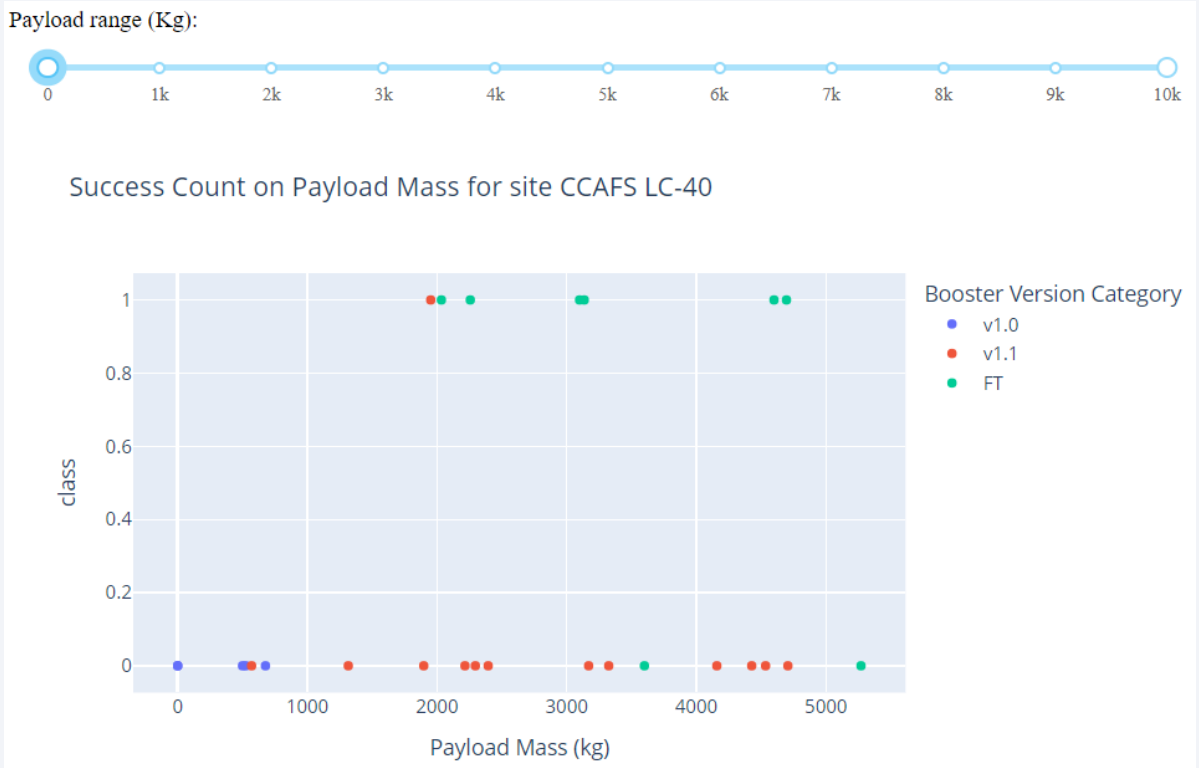
- KSC LC-39A, the successful landing was ~77%. Moreover, CCAFS LC-40 had high success rate of ~73%.

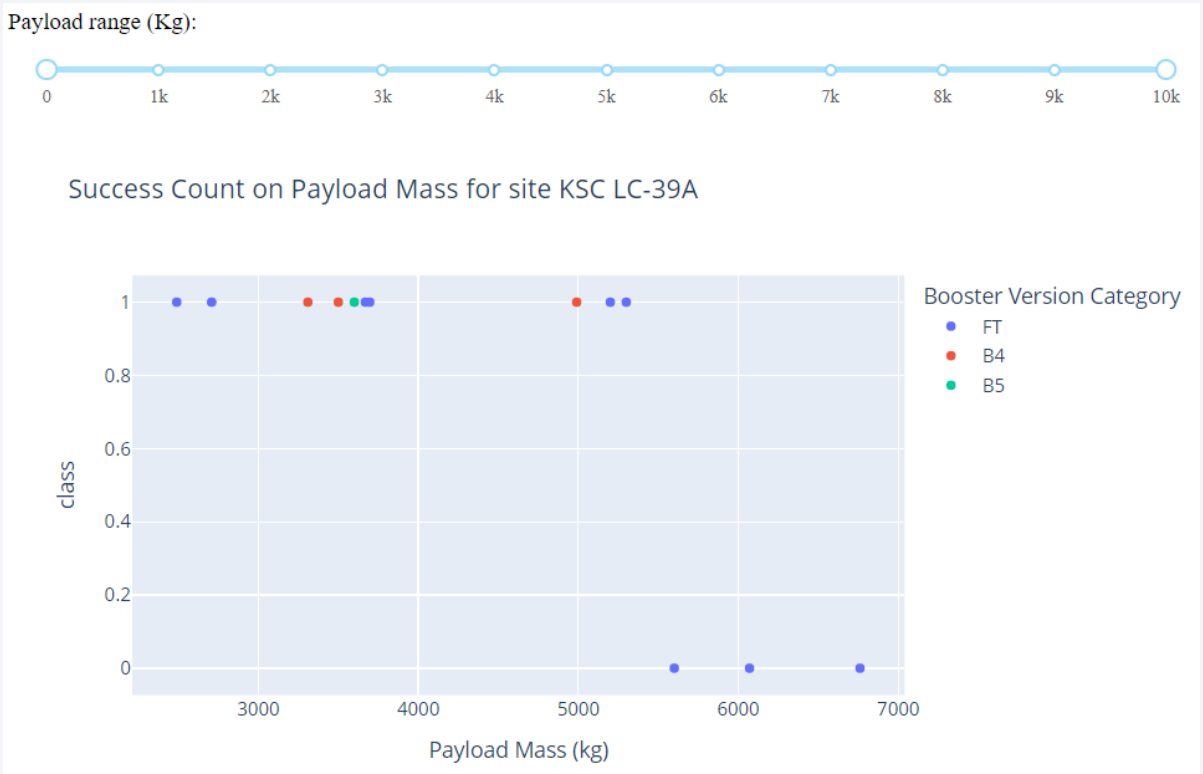


# Relationship between Payload Mass and Class

- Booster Version of FT had the highest success between payload mass of 2000 and 5000 kg. When the payload mass was higher than 5000 kg, it was failure.



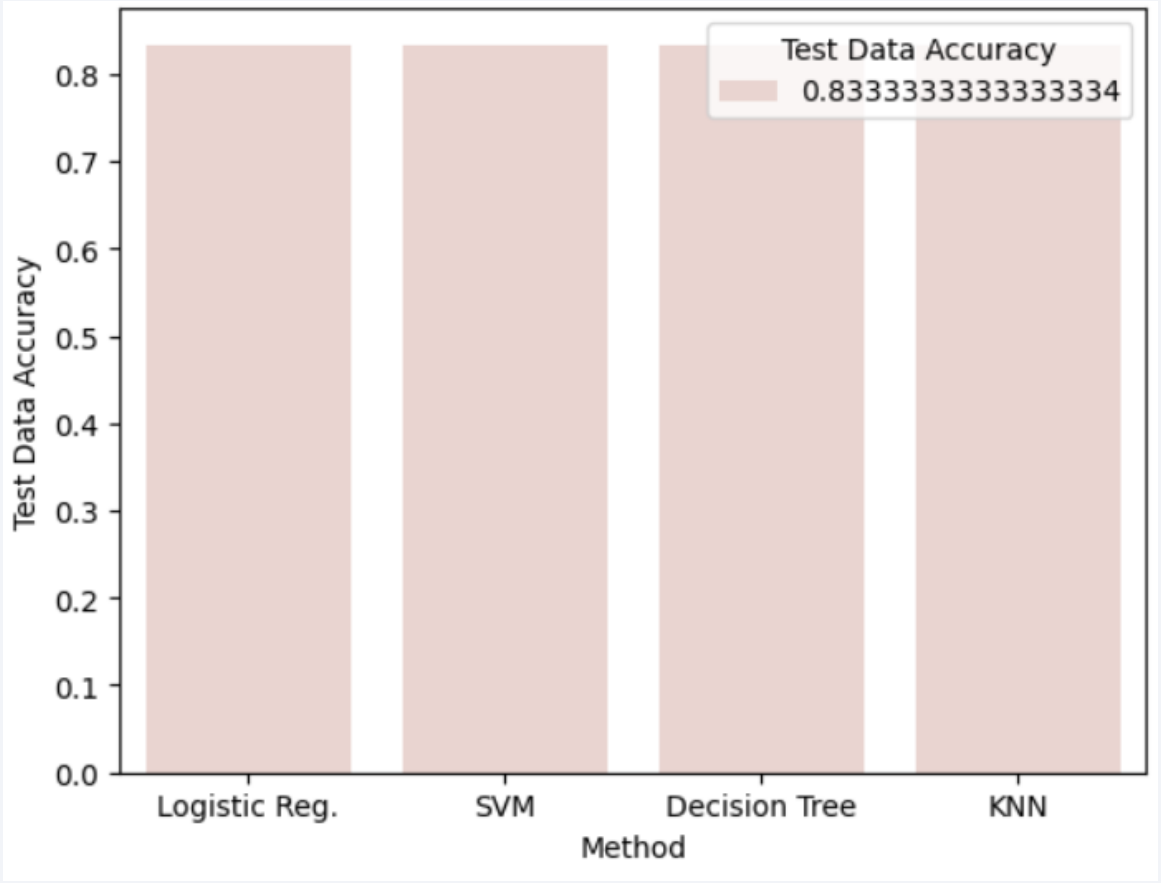



Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

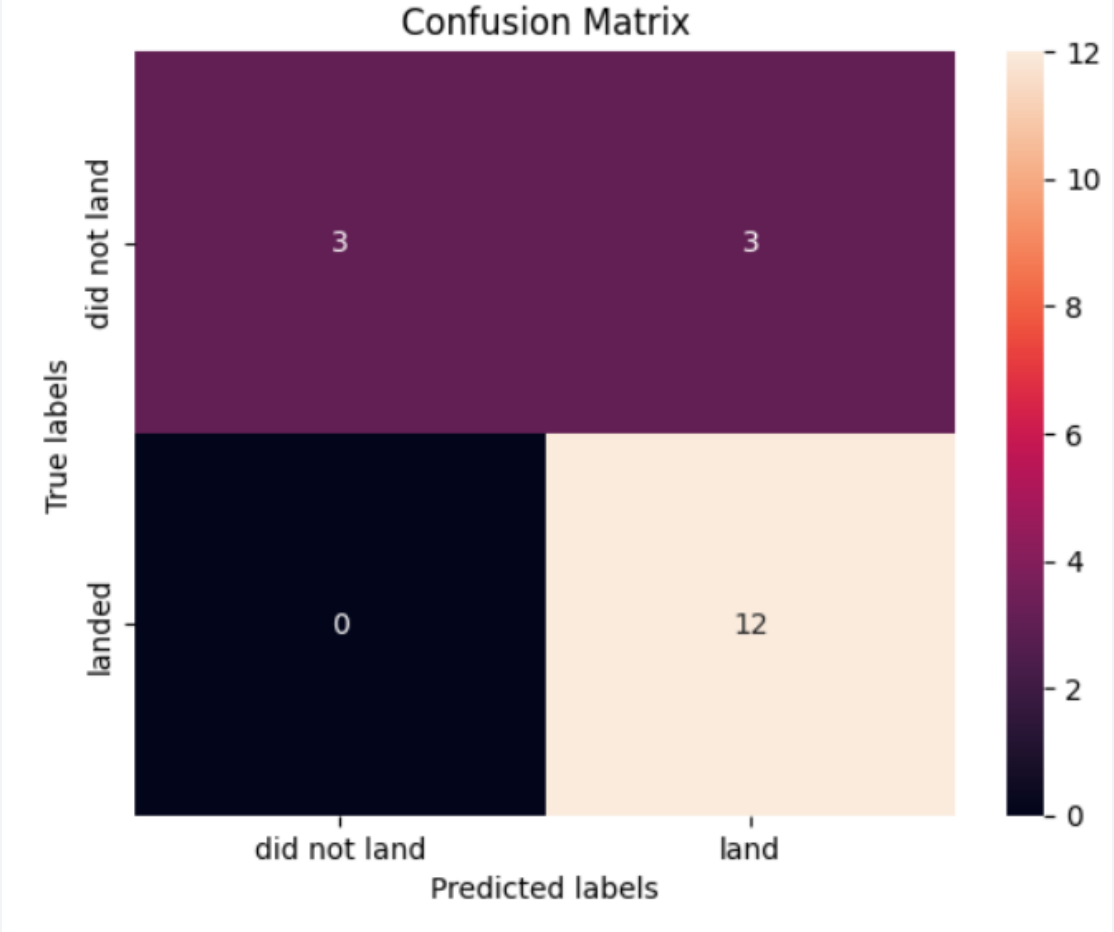
- All the methods had the same test accuracy.




| | Method | Test Data Accuracy |
|---|---|---|
| 0 | Logistic Reg. | 0.833333 |
| 1 | SVM | 0.833333 |
| 2 | Decision Tree | 0.833333 |
| 3 | KNN | 0.833333 |

# Confusion Matrix

- All the models performed the best with accuracy of 83%. All of them had 12 TP, 3 FP, 3 TN, 0 FN.



# Conclusions

- Launch sites had different success rates. For instance, KSC LC-39A had the highest success rate among all sites, while CCAFS SLC-40 had the lowest one. KSC LC-39A, the successful landing was ~77%. Moreover, CCAFS LC-40 had high success rate of ~73%.
- Booster Version of FT had the highest success between payload mass of 2000 and 5000 kg. However, when the payload mass was higher than 5000 kg, it was failure.
- With heavy payloads the successful landing or positive landing rate are more for PO, LEO and ISS.
- In the LEO orbit, success seems to be related to the number of flights. After flight number 7, success rate was 100%.

# Conclusions

- For the orbit type of ES-L1, GEO, HEO and SSO, the success rate was 100%. On the other hand, for SO, it was 100% failure.
- For the VAFB-SLC  launch site there are no rockets launched for  heavy payload mass (greater than 10000).
- After flight number 80, for all the launch sites, success rate was 100%. However, flight number between 0 and 20, success rate was too low.
- In summary, the success rate kept increasing since 2013 until 2020.

# Thank you!