IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Arushi Kapoor
16th April 2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- The objective of the capstone project is to determine the price of each launch. This is done by gathering information about Space X and creating dashboards.
- It is also found whether SpaceX will reuse the first stage. Instead of using rocket science to determine if the first stage will land successfully, we have trained a machine learning model and used public information to predict if SpaceX will reuse the first stage.

Introduction

- Using the Falcon9 data we are trying to draw conclusions if first stage will be re-used and hence determine the cost of the launch.
- Stages of the capstone project:
  - Collecting the data - API and Web Scrapping
  - Data Wrangling - EDA to find some patterns in that data.
  - Exploratory analysis with SQL
  - Exploratory analysis with Pandas and Matplotlib
  - Interactive Visual Analytics
  - Predictive Analysis

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data is collected via API and Web Scrapping
- Perform data wrangling
  - EDA to find some patterns in the data
- 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 was trained on several models like Logistic Regression, SVM, Decision Tree and KNN to see which gives best accuracy
  - Hyperparameters for the models was selected by Grid Search approach.

Data Collection: API

1. **Request and parse the SpaceX launch data using the GET request**
   - Decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()
   - Use the API again to get information about the launches using the IDs given for each launch. Specifically we will be using columns rocket, payloads, launchpad, and cores.
2. **Filter the dataframe to only include `Falcon 9` launches**
   - remove the Falcon 1 launches keeping only the Falcon 9 launches. Filter the data dataframe using the BoosterVersion column to only keep the Falcon 9 launches. Save the filtered data to a new dataframe called data_falcon9.

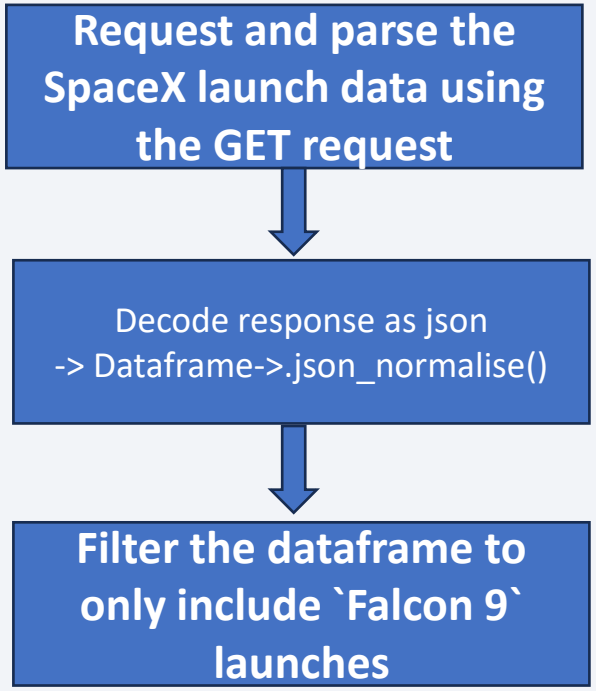
Data Collection: Web Scrapping

Web scrap Falcon 9 launch records with BeautifulSoup:

- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame

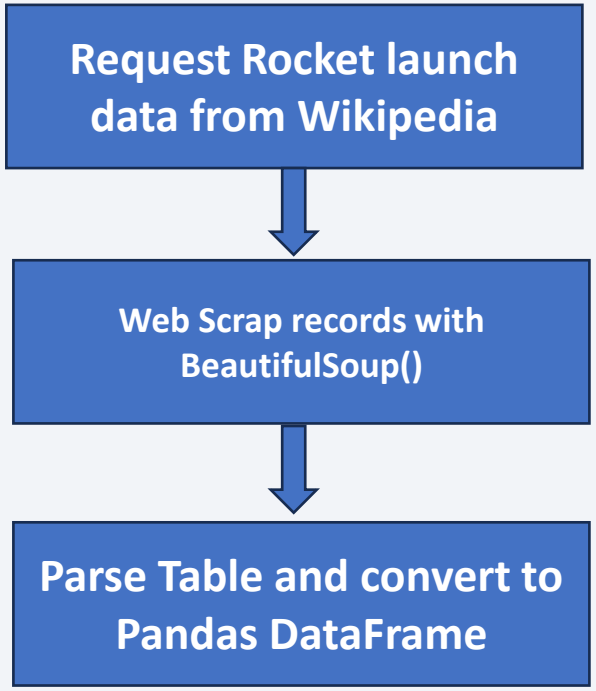
Data Collection – SpaceX API

- Link to Notebook:
  - Click here



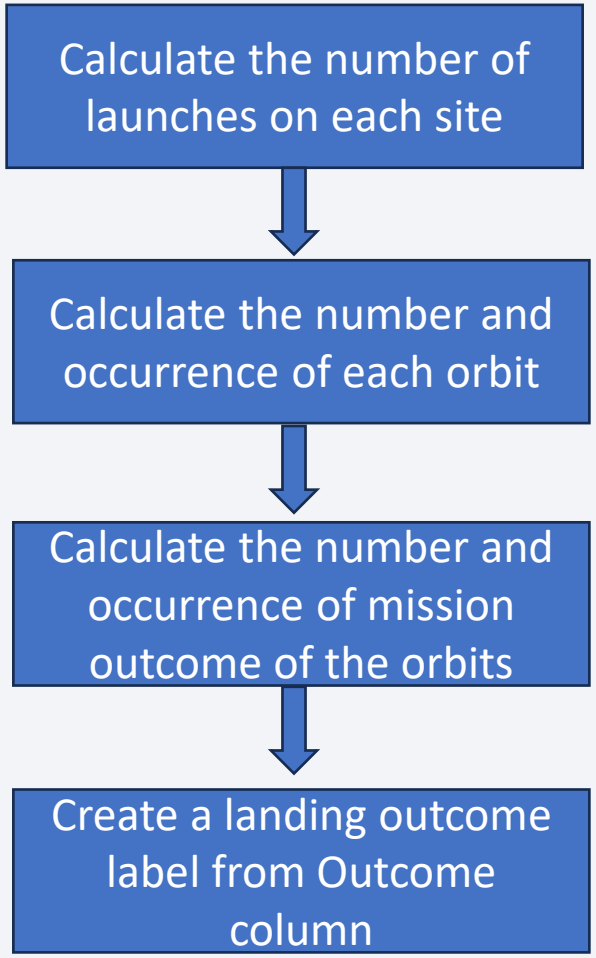
Data Collection - Scraping

- Link to notebook
  - Click here



Data Wrangling

Performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.



Github Link

- Click here

EDA with Data Visualization

- Charts Plotted:

    1. Scatter Plot- *Visualize the relationship between Flight Number and Launch Site, Visualize the relationship between Payload and Launch Site* etc

    2. Bar Chart - *Visualize the relationship between success rate of each orbit type*

    3. Line chart- *Visualize the launch success yearly trend*

Github link – Click here

EDA with SQL

- SQL queries carried out to do the following:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - 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.

- Github link – click here

Build an Interactive Map with Folium

- Folium.Marker and folium.circle are added for each launch site on map
- Success/failed launches for each site on the map are marked
- Folium.Marker and folium.circle are added for each launch result in spacex_df
- Distances between a launch site to its proximities is calculated
- Github link – click here

Build a Dashboard with Plotly Dash

- Pie chart shows success count for all launch sites
  - KSC LC-39A has the most number of successful launches
  - CCAFS SLC 40 has the least number of successful launches
- Scatter plot shows success count on Payload masses for all sites
- Github link – click here

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 – Grid Search
- Find the method performs best using test data
- Github link – click here

Results

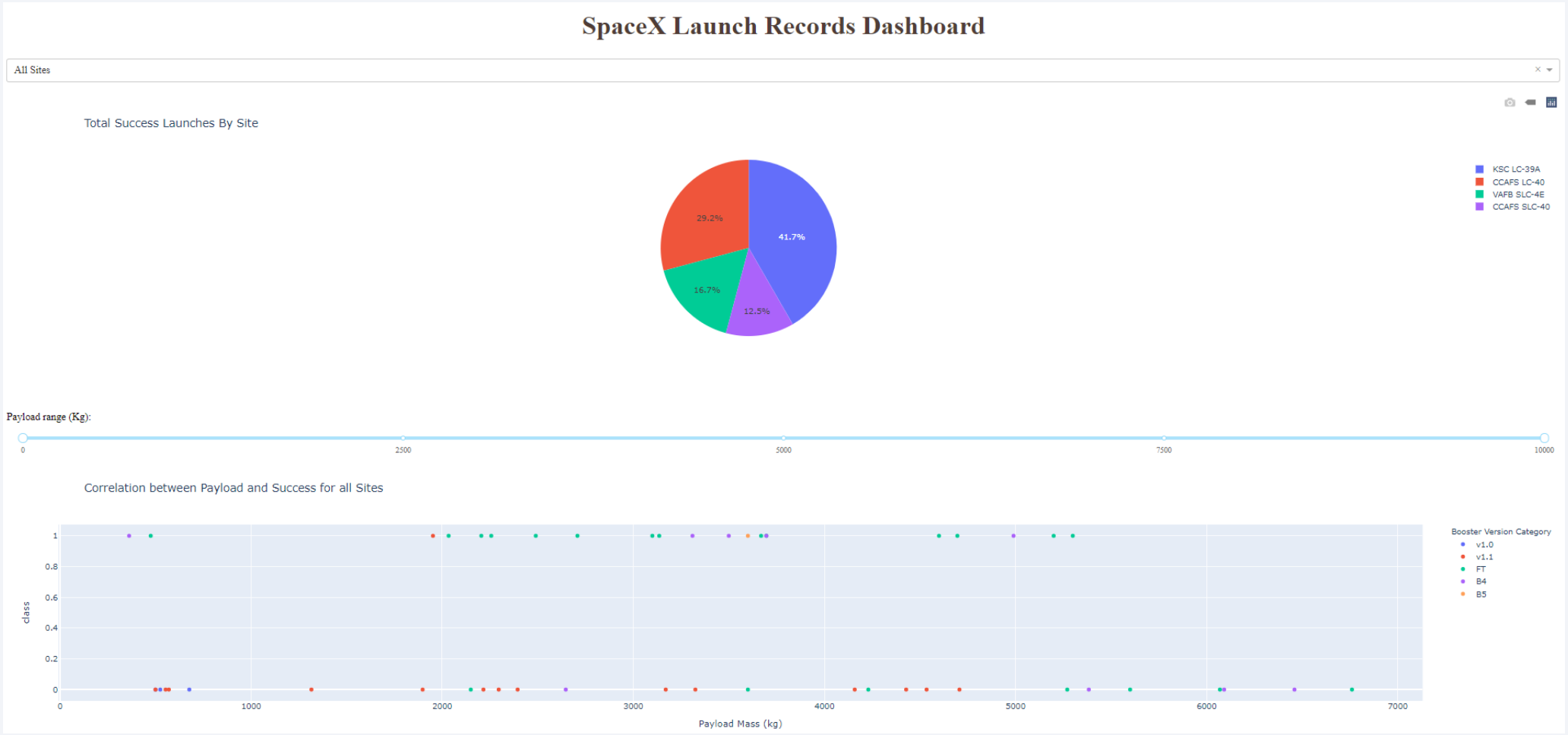
- All models performed equally well with accuracy 83.3%

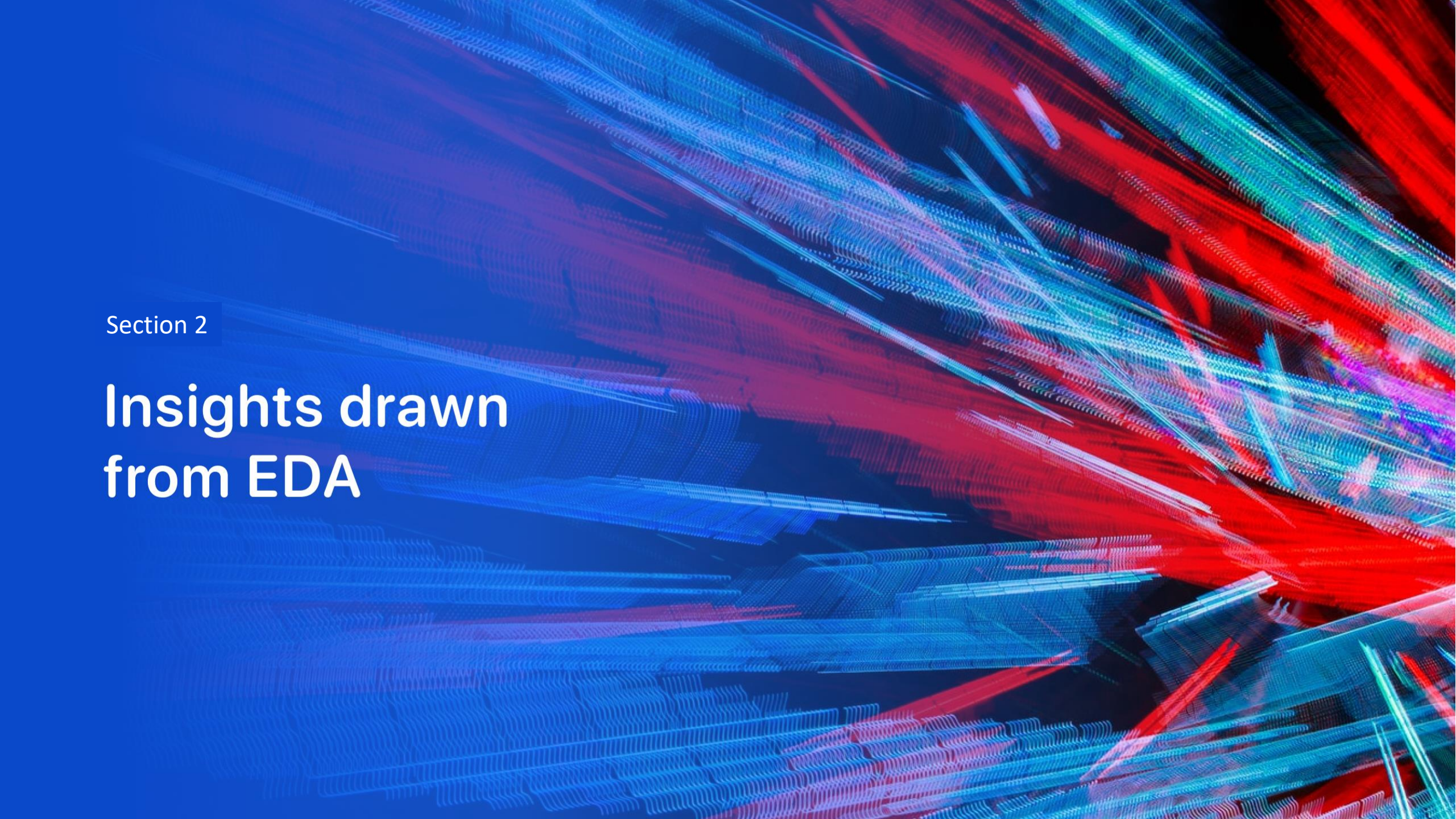
Out[41]:

	0
Method	Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

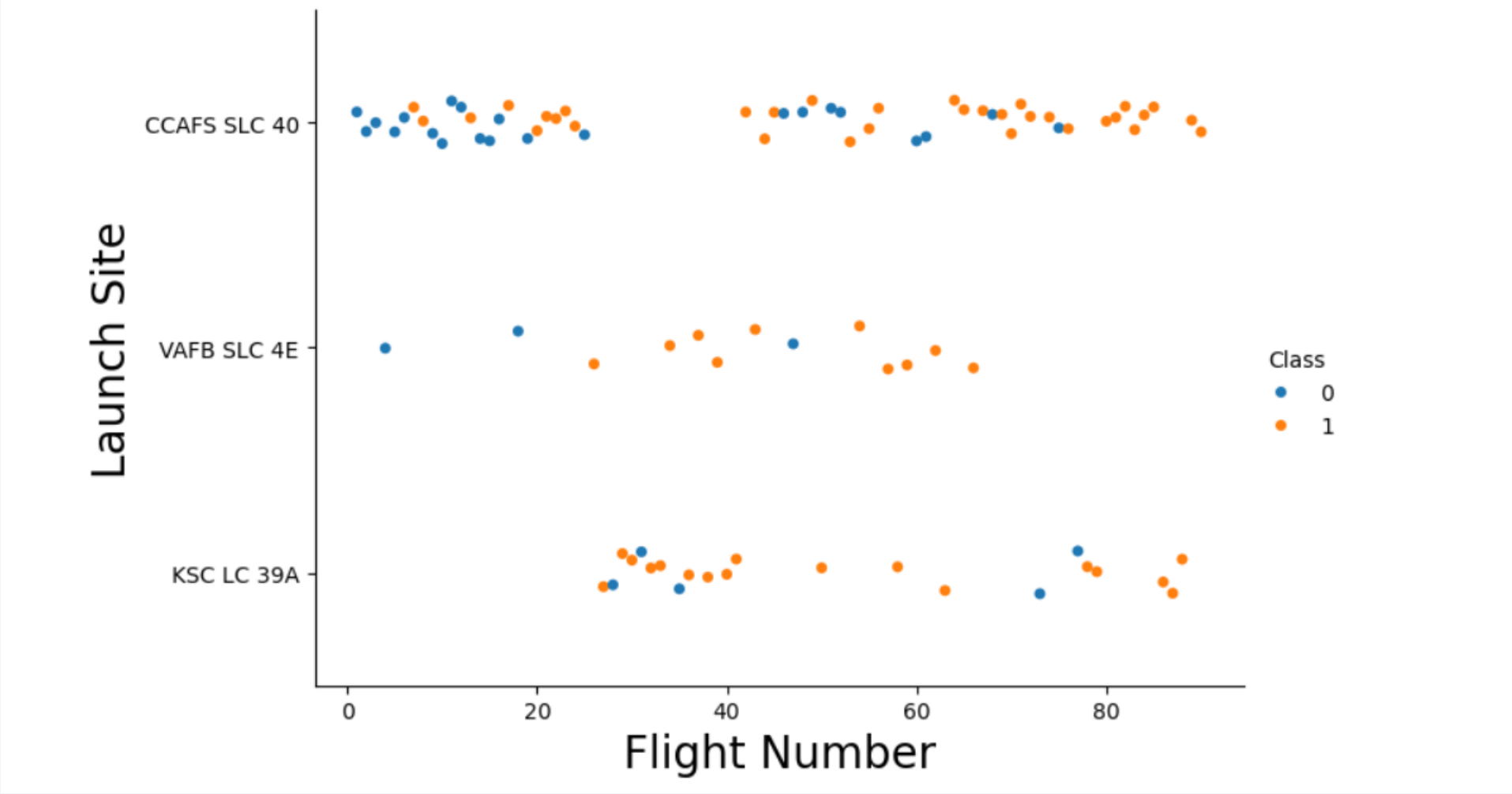
Results

- Dashboard

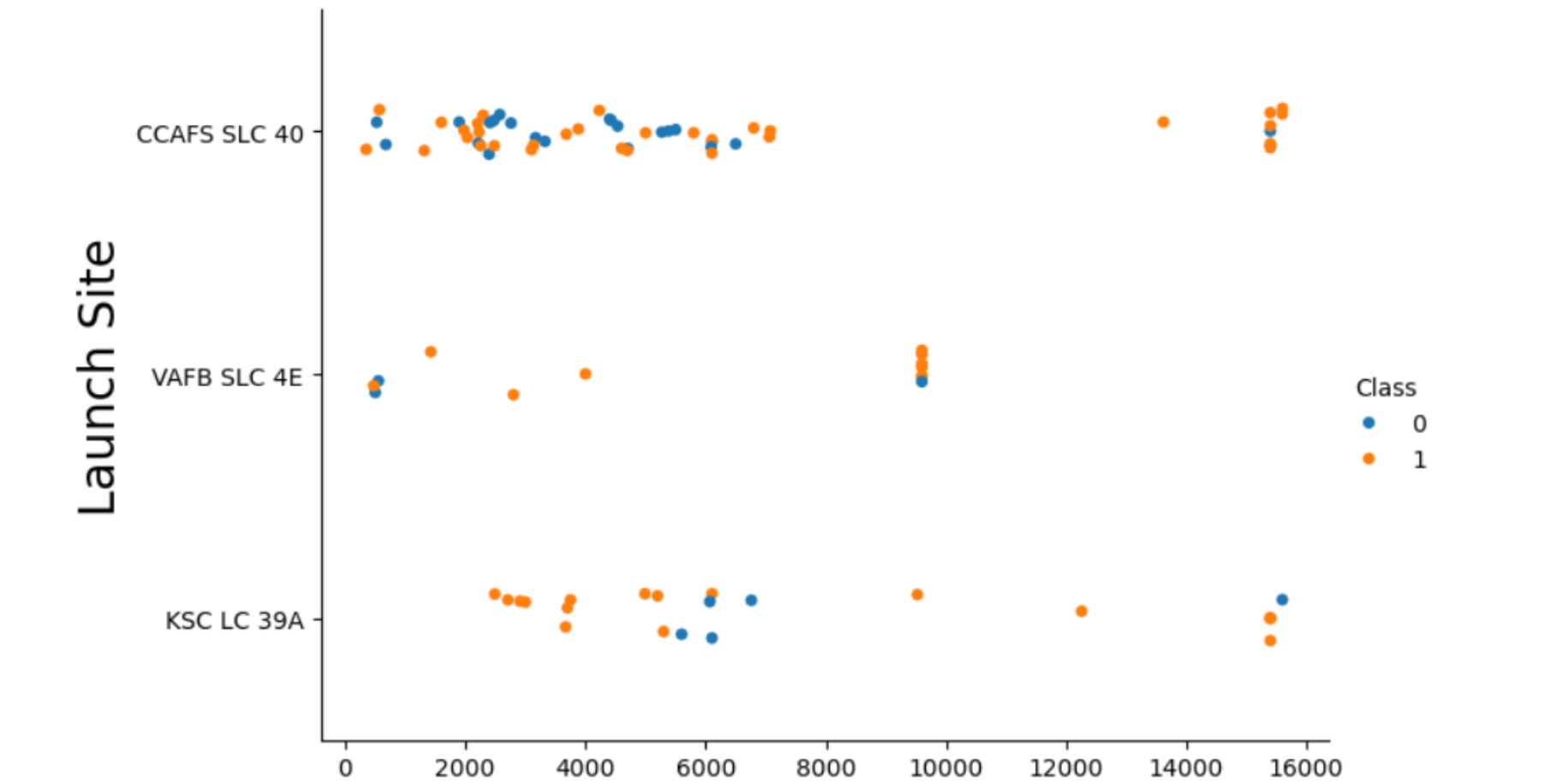



Section 2
Insights drawn from EDA

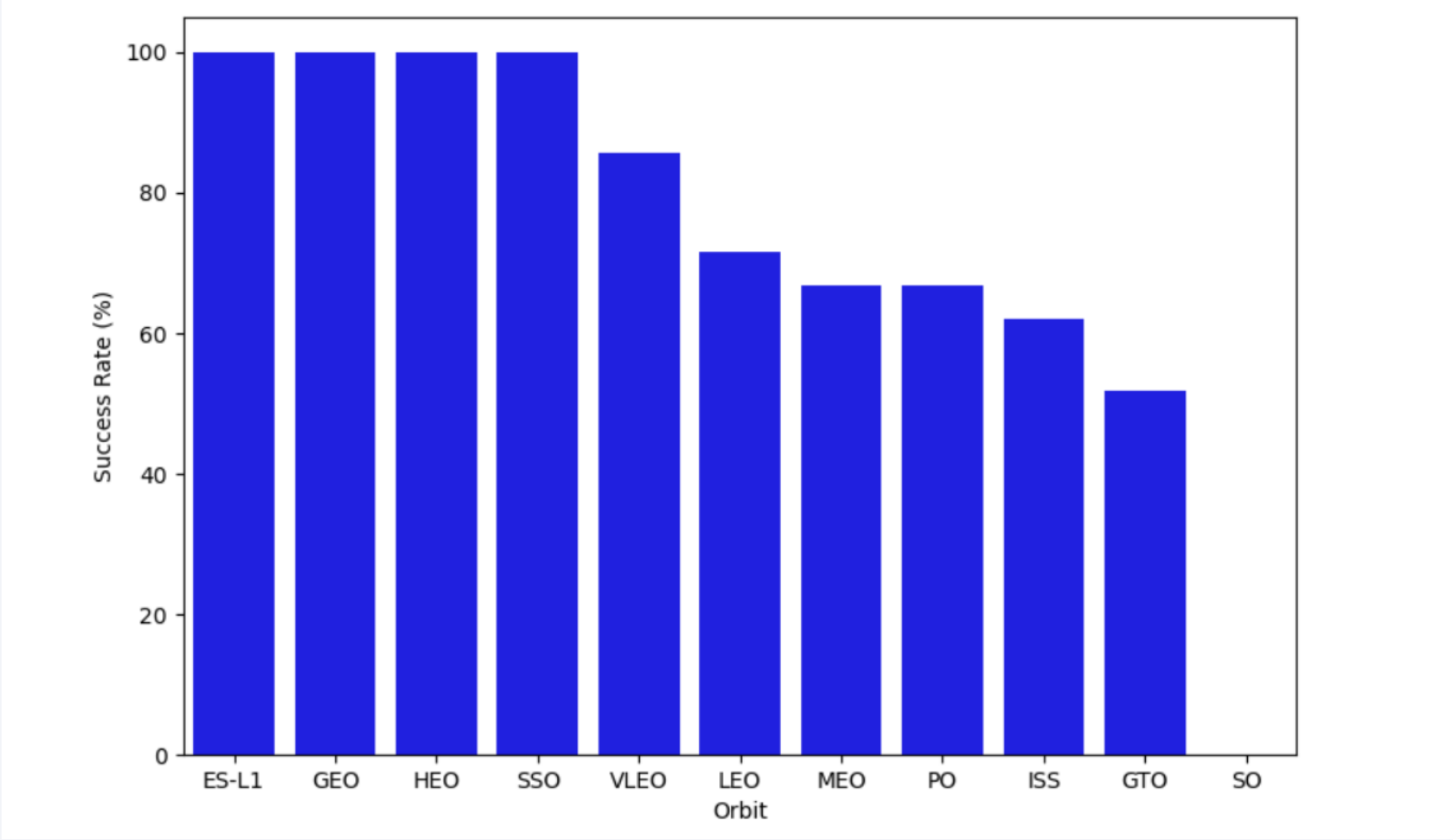
Flight Number vs. Launch Site



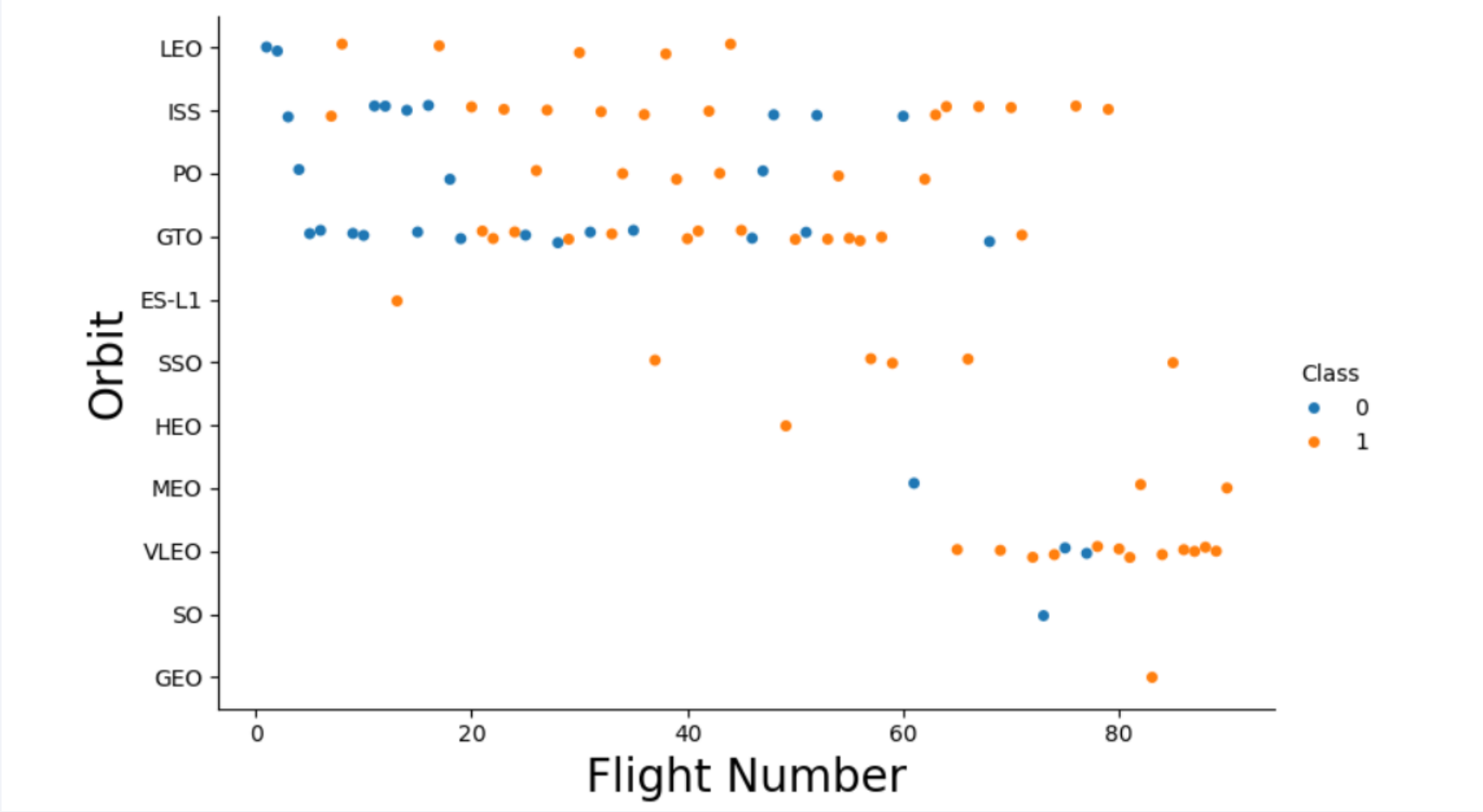
Payload vs. Launch Site



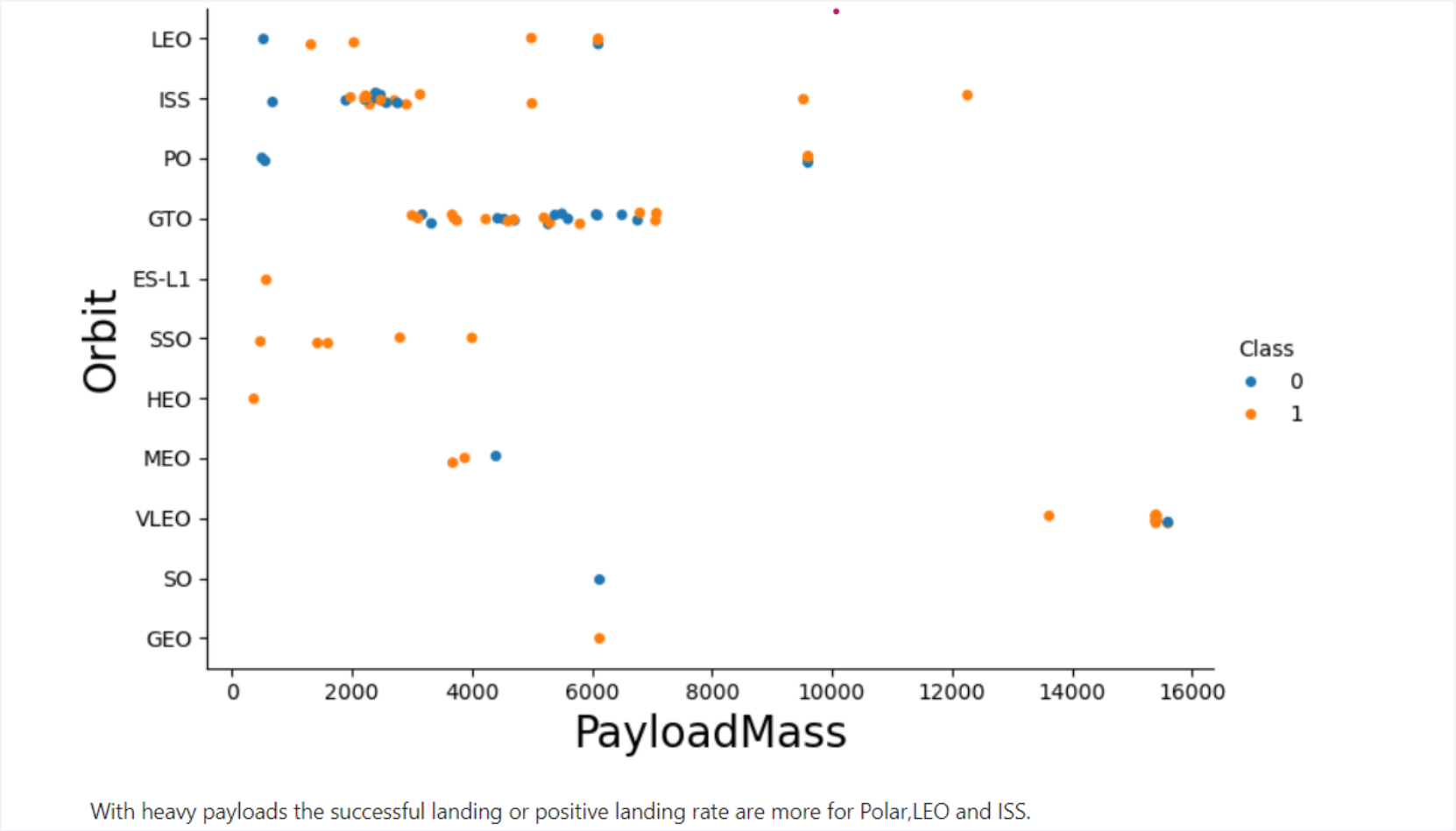
Success Rate vs. Orbit Type



Flight Number vs. Orbit Type

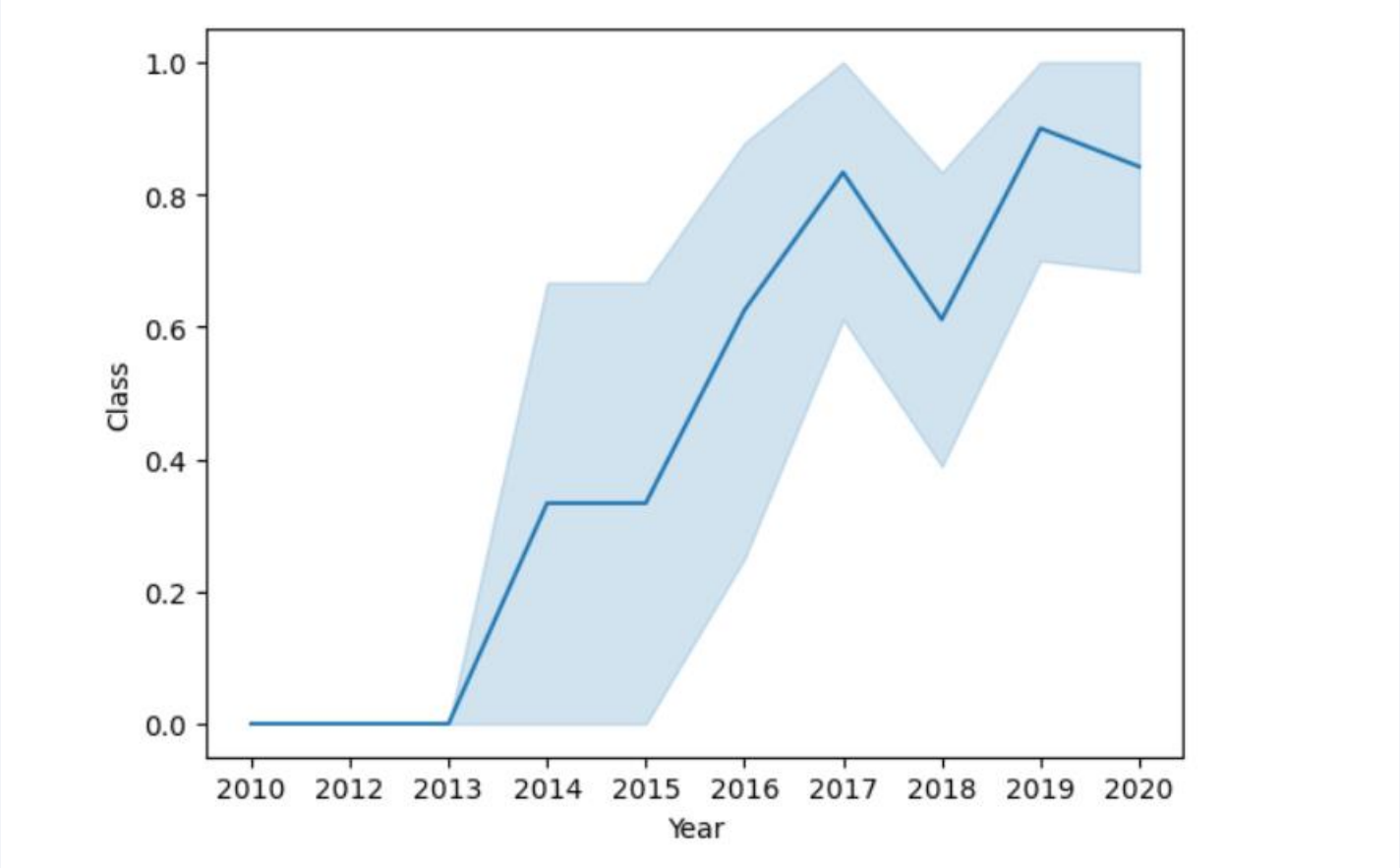


Payload vs. Orbit Type



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

Launch Success Yearly Trend



All Launch Site Names

In [8]:
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

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

Out[8]:

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

Launch Site Names Begin with 'CCA'

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

In [45]:
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Out[45]:

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

Task 3

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

```
[2]: %sql SELECT SUM(PAYLOAD_MASS__KG_),CUSTOMER FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'
```

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

[2]:

SUM(PAYLOAD_MASS__KG_)	Customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

Task 4

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

```
3]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

3]:

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

```
%sql SELECT MIN(Date) FROM SPACEXTBL 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

- %sql SELECT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing_Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ <6000

Out[24]:

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
27]: %sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

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

27]:

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

Boosters Carried Maximum Payload

- %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)

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

- %sql SELECT substr(Date, 6, 2) AS Month,"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Outcome", "Landing _Outcome" FROM SPACEXTBL WHERE substr(Date,0,5)='2015' AND "Landing _Outcome" = 'Failure (drone ship)';

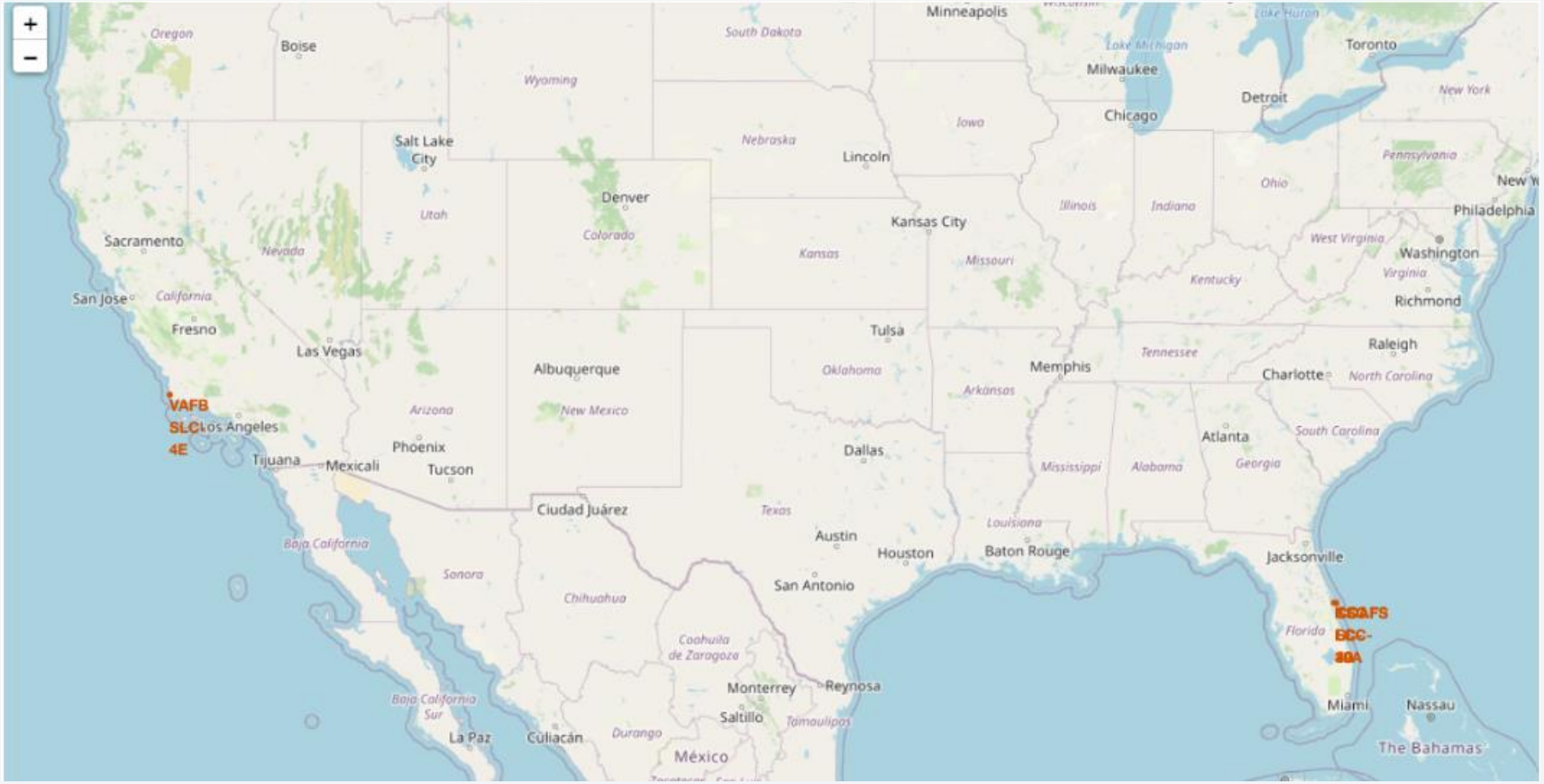
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- %sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '2010-06-04' AND '2017-03-20') ORDER BY Date DESC;

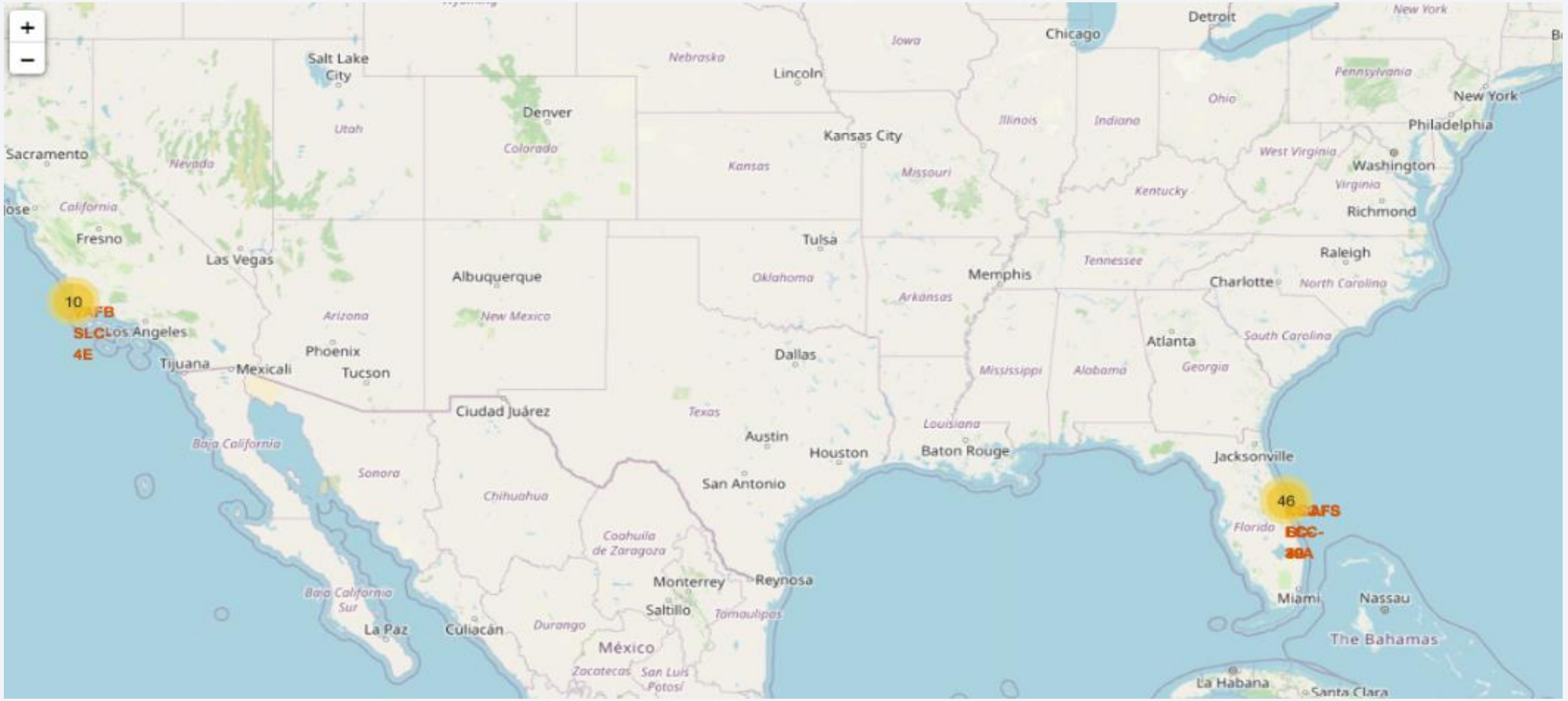
Section 3

Launch Sites Proximities Analysis

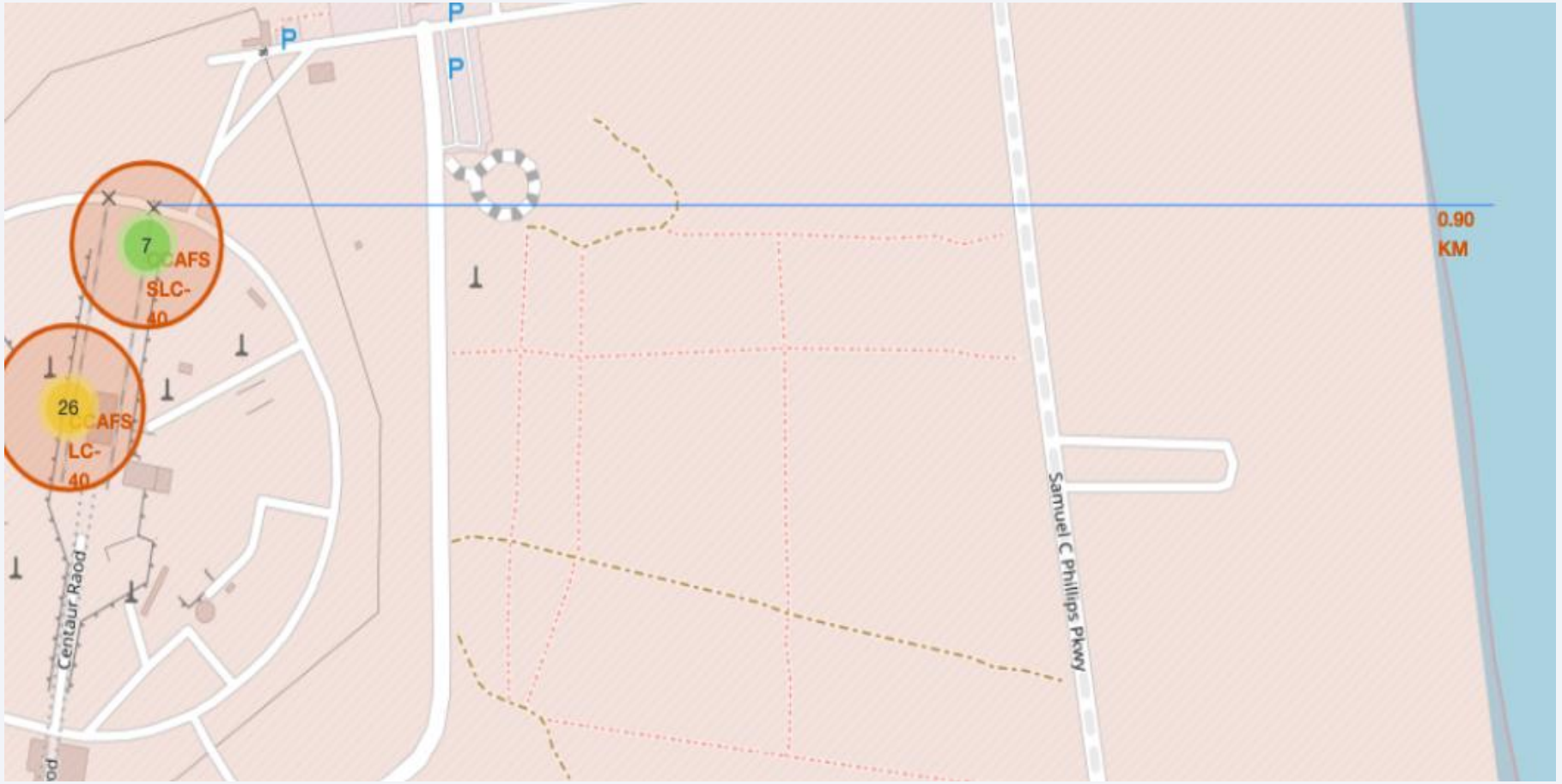
Folium Map: map with marked launch sites

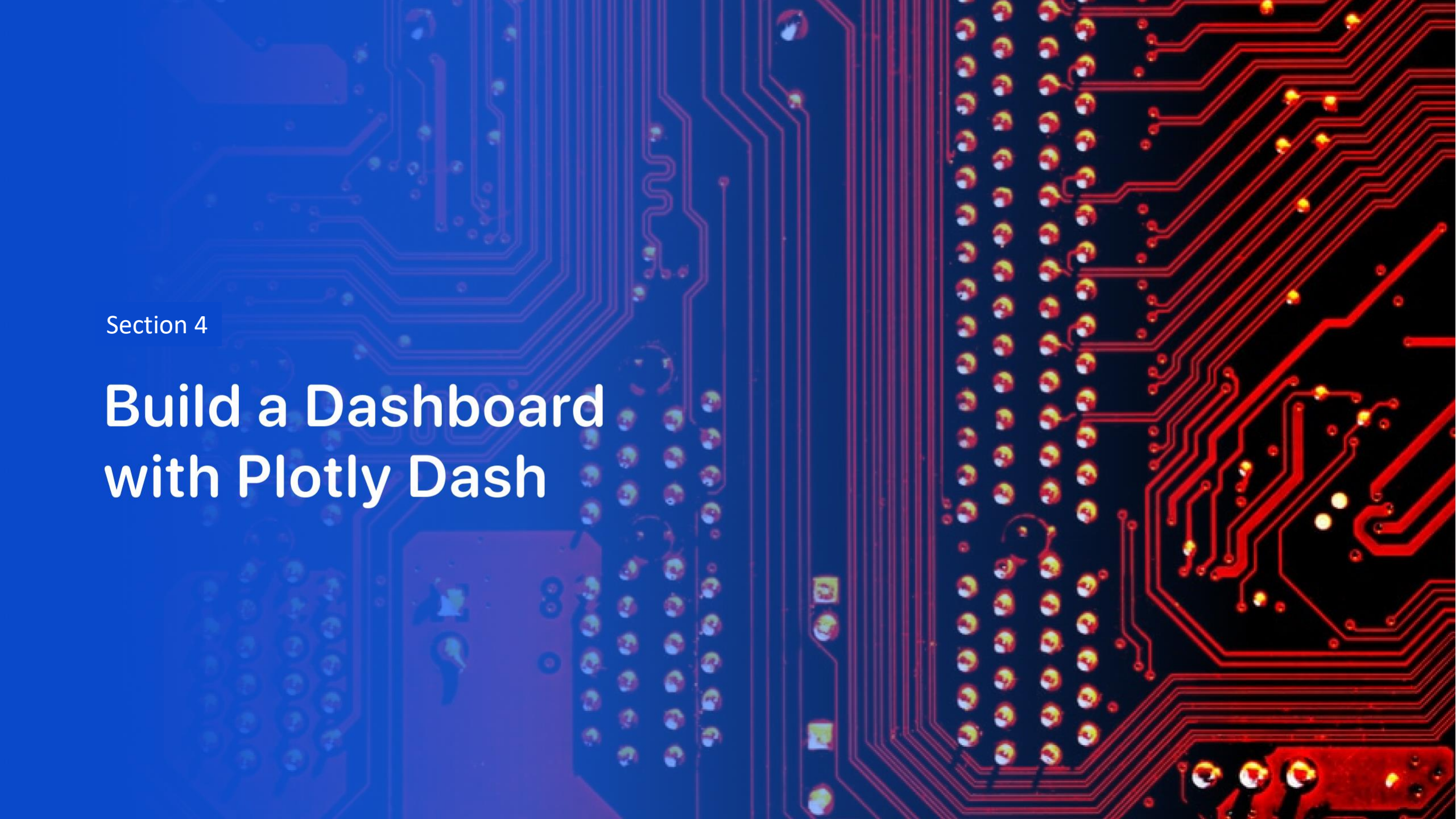


Folium Map: Mark the success/failed launches for each site on the map

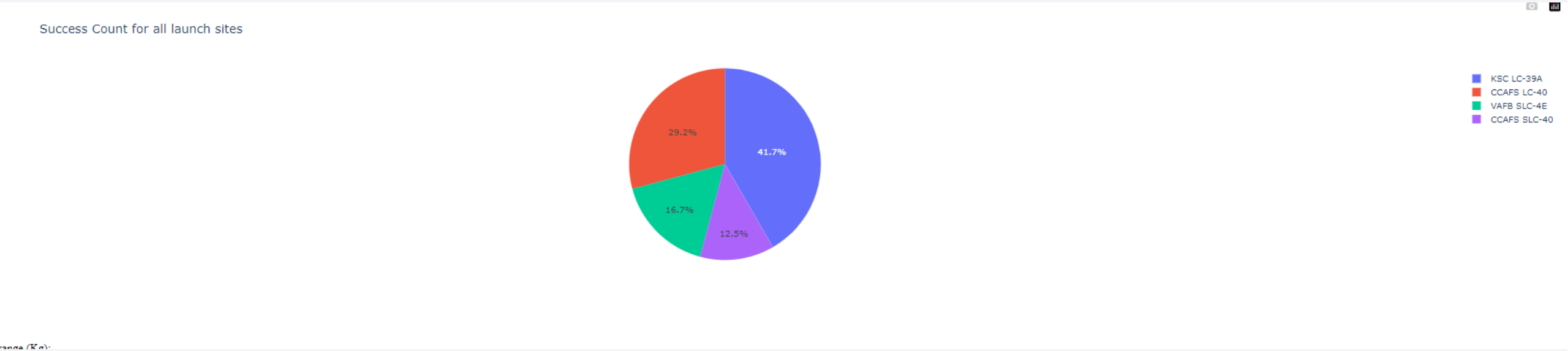


Folium Map: Calculate the distances between a launch site to its proximities

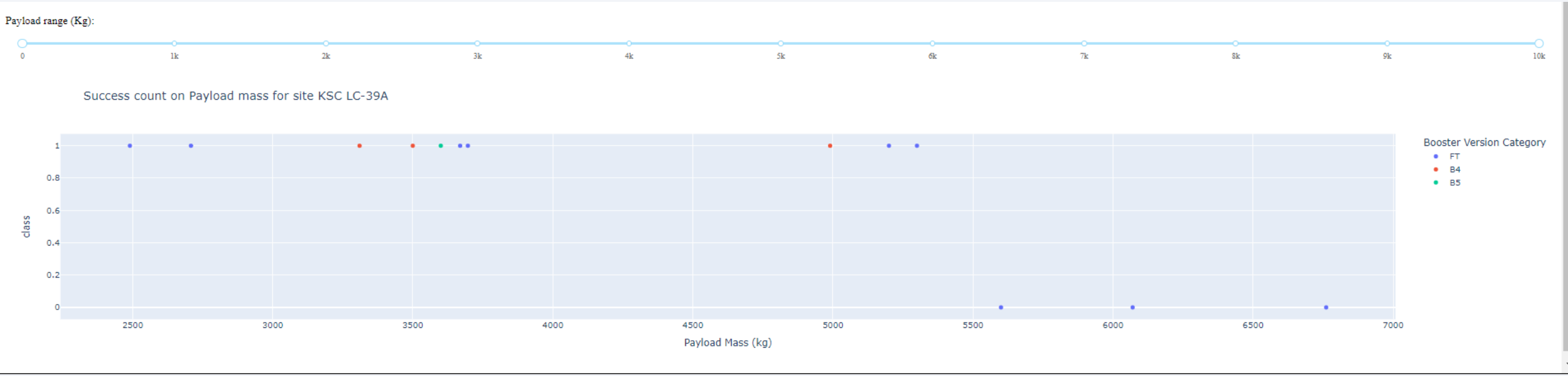



Section 4
Build a Dashboard with Plotly Dash

Dashboard: Pie chart



Dashboard


Payload range (Kg):
0
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Success count on Payload mass for site KSC LC-39A
class
1
0.8
0.6
0.4
0.2
0
2500
3000
3500
4000
4500
5000
5500
6000
6500
7000
Payload Mass (kg)
Booster Version Category
FT
B4
B5

Section 5

Predictive Analysis (Classification)

Classification Accuracy

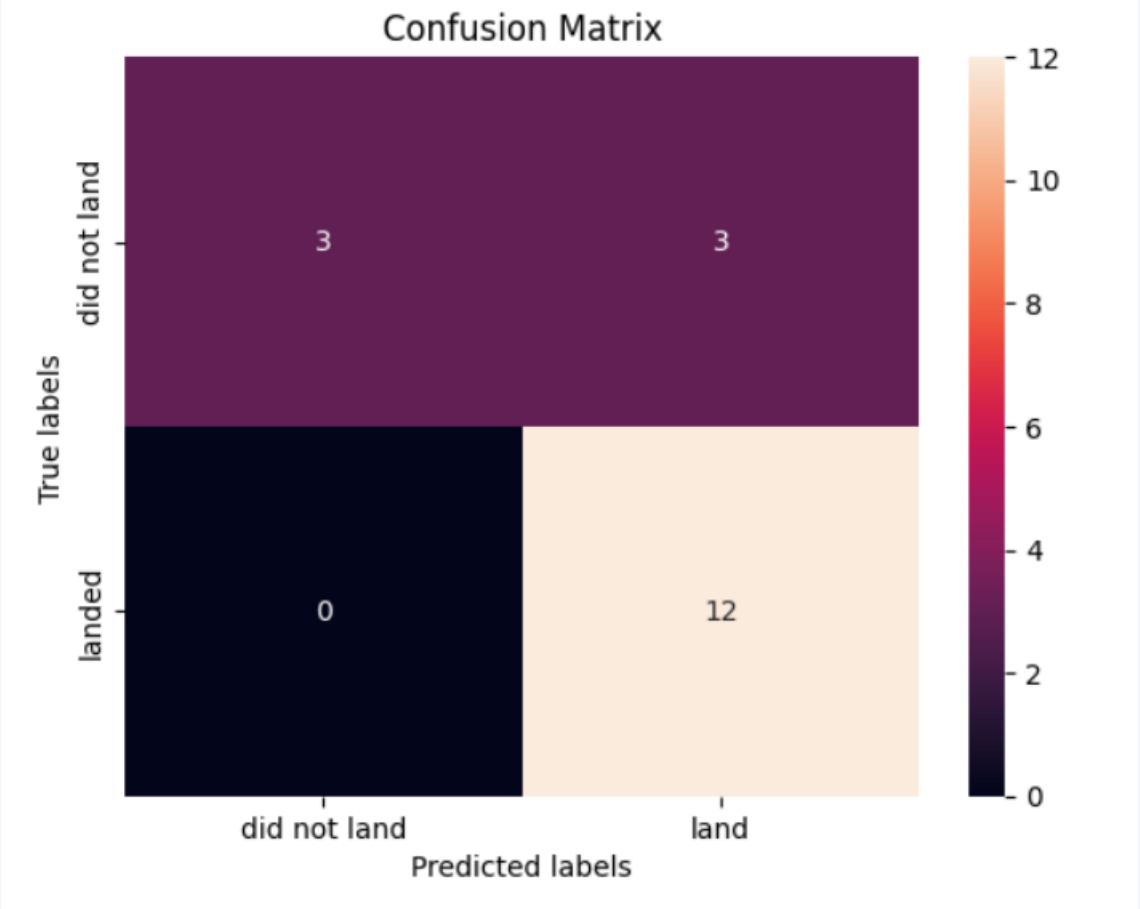
- All models performed equally well with an accuracy of 83.3%

Out[41]:

	0
Method	Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Confusion Matrix

- All models performed equally well with an accuracy of 83.3%



Conclusions

- Data collection with API and Web Scrapping
- EDA with SQL
- EDA with data visualization
- Data trained on several models – Logistic Regression, KNN, Decision Trees, SVM
- Tunning of Hyperparameter using Grid Search
- Accuracy: 83.3% for all the models

Thank you!