Winning Space Race with Data Science

WANG Qi

December 9, 2022

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- **Context**

  SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars, while other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.

- **Problem to be answered**

  Predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology
  - Extract rocket launch data SpaceX API
  - Web scraping from Wikipedia page using BeautifulSoup
- Perform data wrangling
  - StandardScaler for numerical variables, OneHotEncoding for categorical variables, replace missing values by its column's mean value
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Compare four algorithms (KNN, SVM, Decision Tree and Logistic Regression) using GridsearchCV to find the best hyperparameter combination.

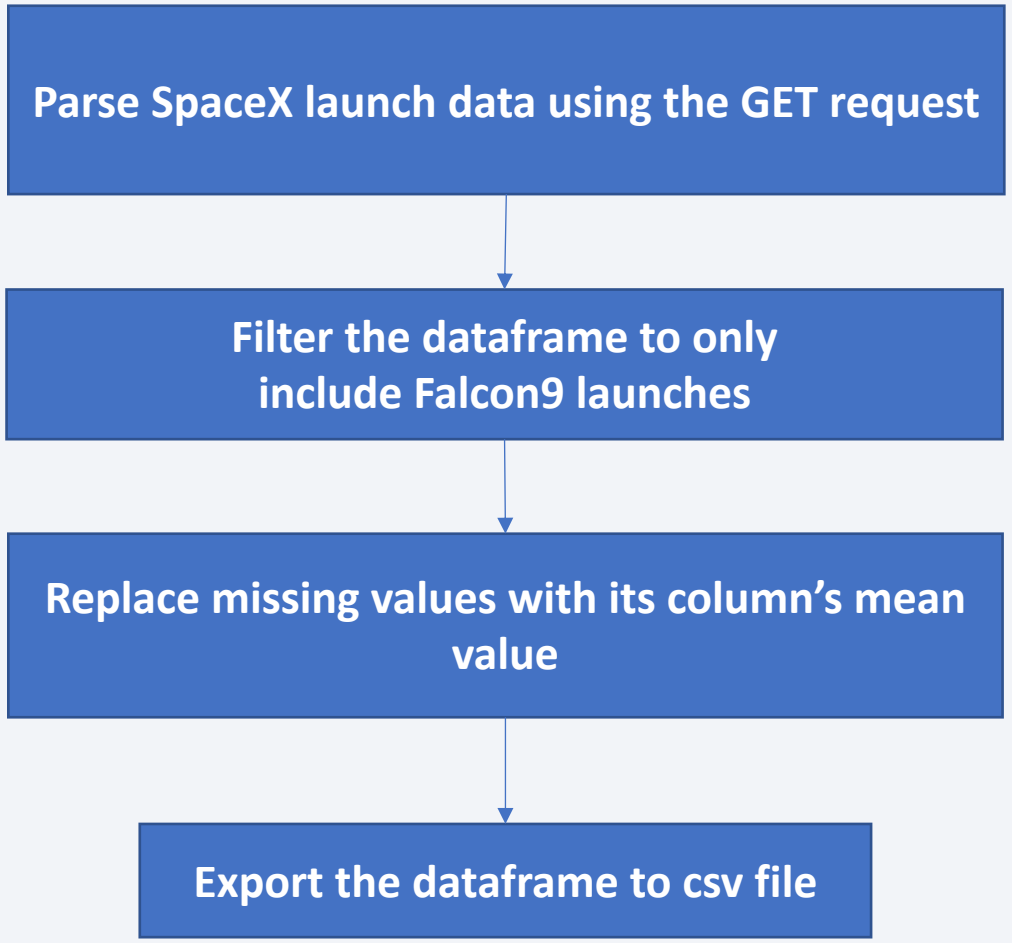
Data Collection

Two datasets were collected in this step

- Rocket launch data using SpaceX API
- Launch records data using web scraping from wikipedia page with BeautifulSoup

Data Collection – SpaceX API

The datasets were collected from SpaceX API using Requests library in Python, which allows us to make HTTP requests which we will use to get data from an API.

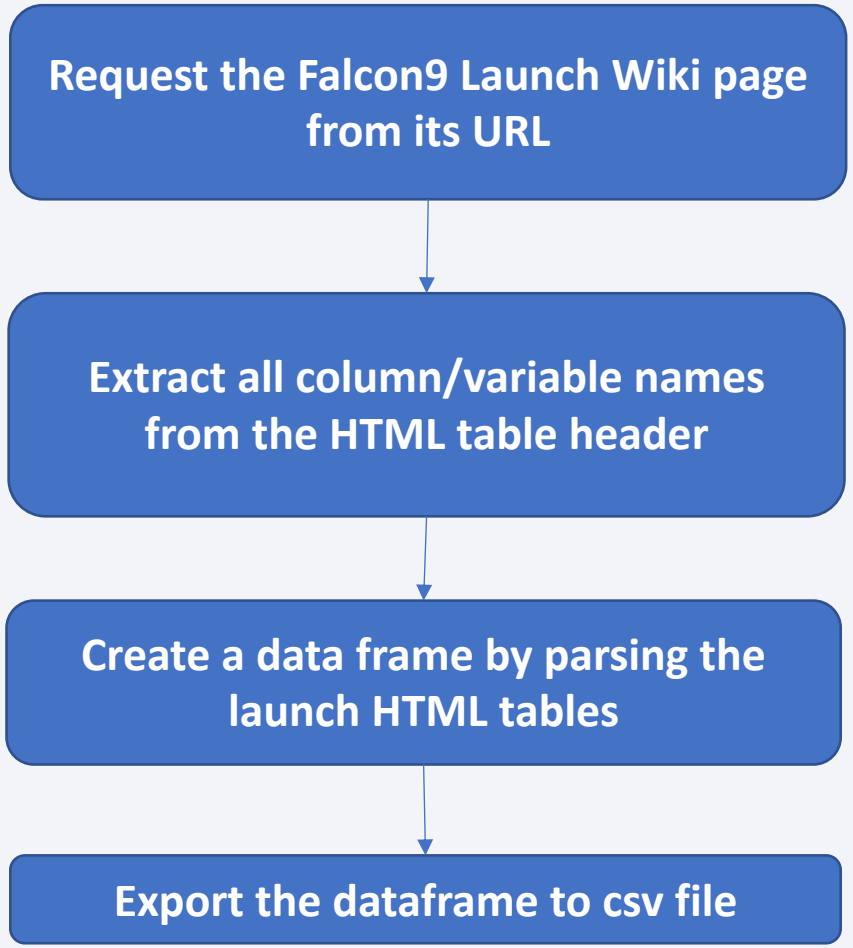


GitHub source code link

Data Collection - Scraping

Web scrap Falcon 9 launch records with BeautifulSoup

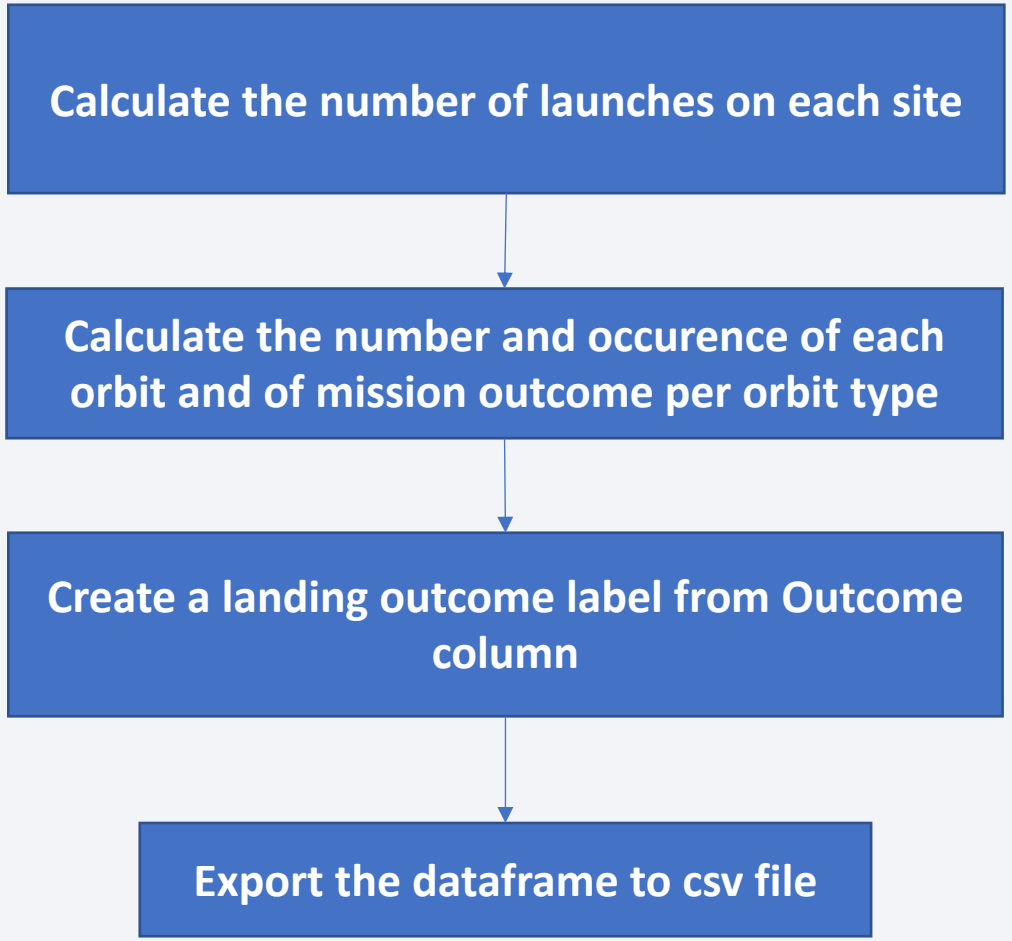
- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame



GitHub source code link

Data Wrangling

In this step, we perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.



GitHub source code link

EDA with Data Visualization

- We visualize the relationship of two chosen variables using scatter plots (for example, Flight Number and Launch Site) to find some patterns of these features if exists.
- We visualize the relationship between success rate of each orbit type using bar plot to compare them in an obvious way.
- We also visualize the launch success yearly trend using line chart.

GitHub source code link

EDA with SQL

- 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 date when the first successful landing outcome in ground pad was acheived.
- 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 total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass using a subquery
- List the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order

GitHub source code link

Build an Interactive Map with Folium

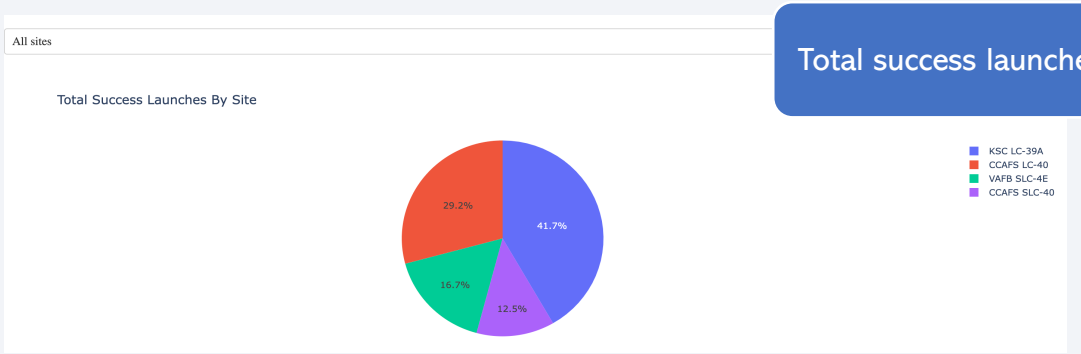
Several methods have been used to create a map with Folium

- Create a folium `Map` object using initial center location
- Add a highlighted circle area with a text label on a specific coordinate with `folium.Circle` and/or `folium.Marker`
- `Makecluster` object is a good way to simplify a map containing many markers having the same coordinate.

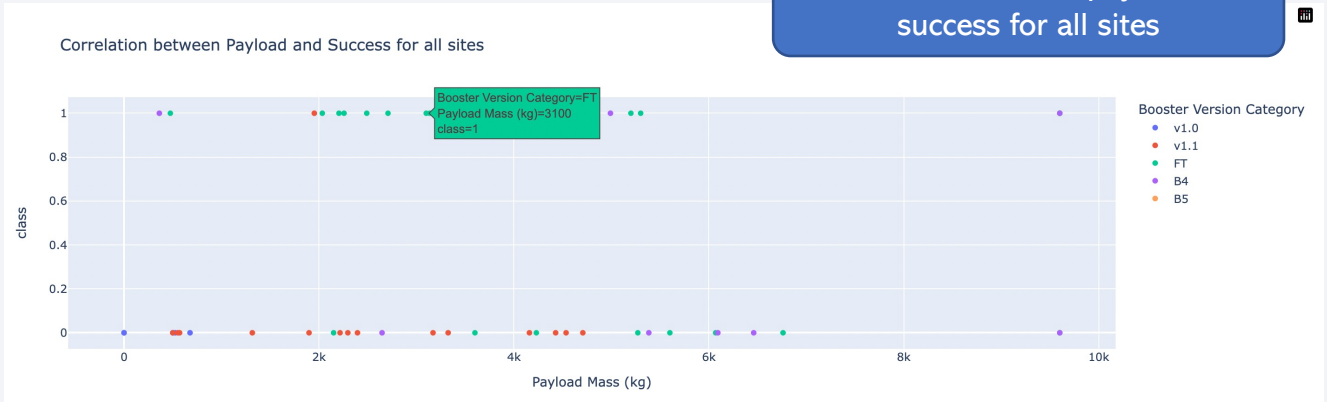
GitHub source code link

Build a Dashboard with Plotly Dash

The dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.



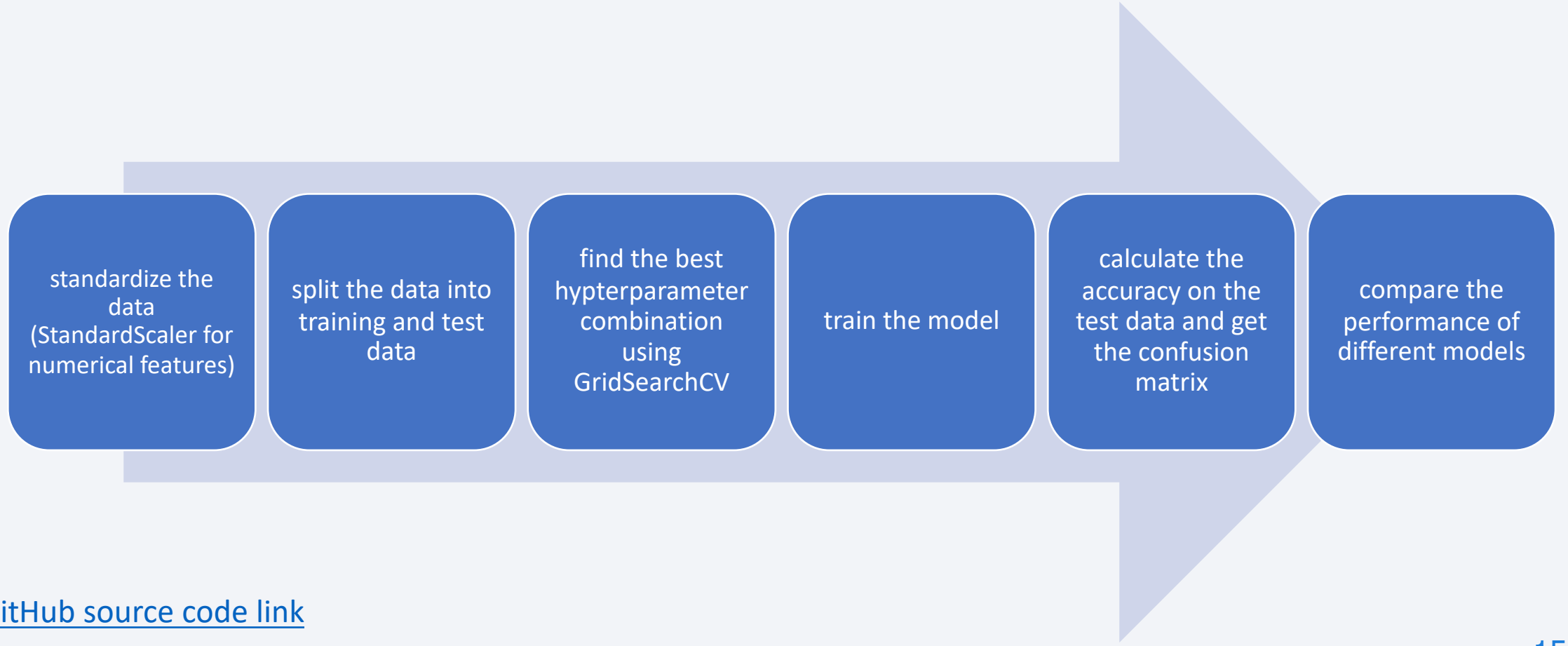
Total success launches by site



Correlation between payload and success for all sites

[GitHub source code link]

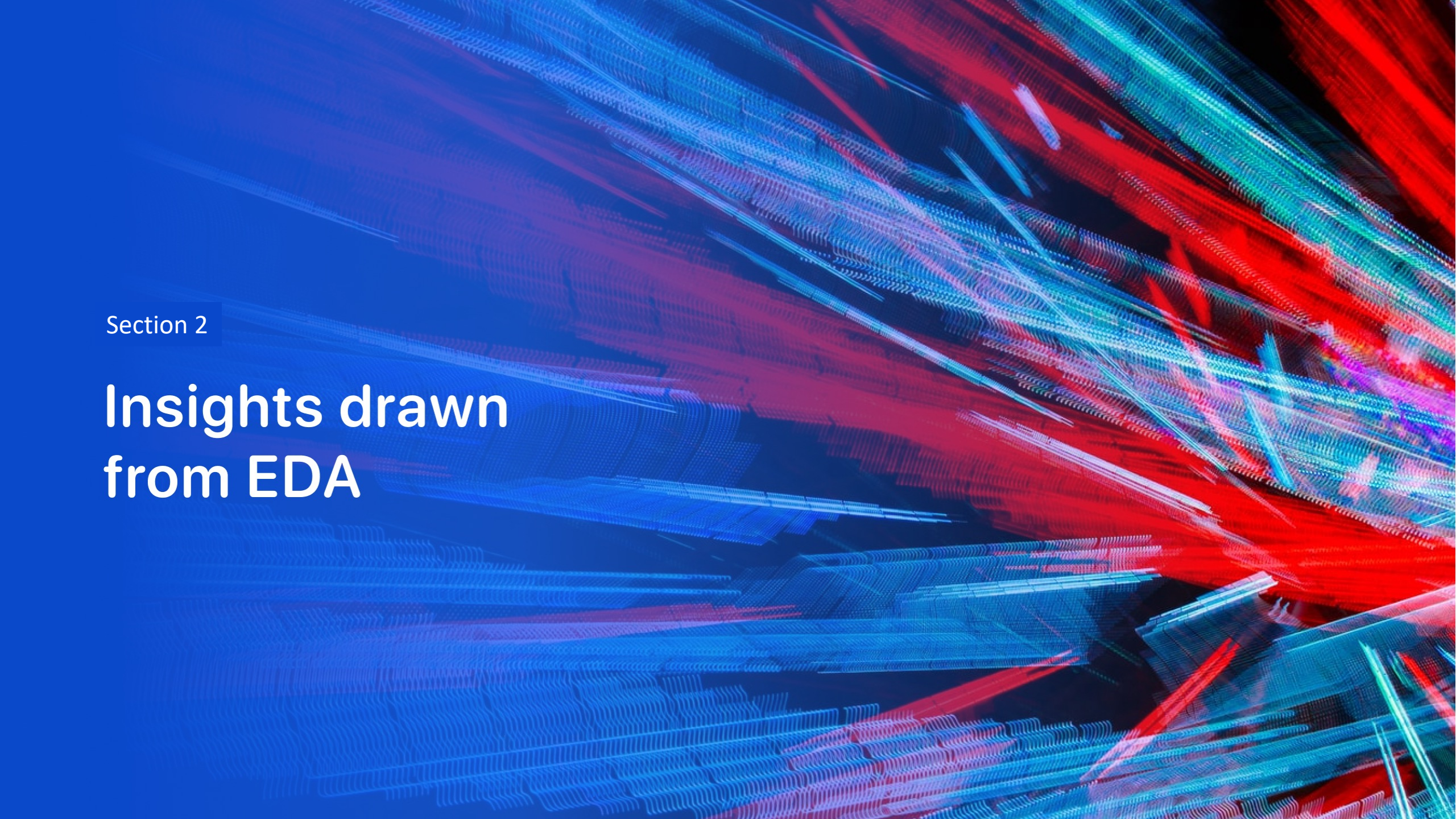
Predictive Analysis (Classification)



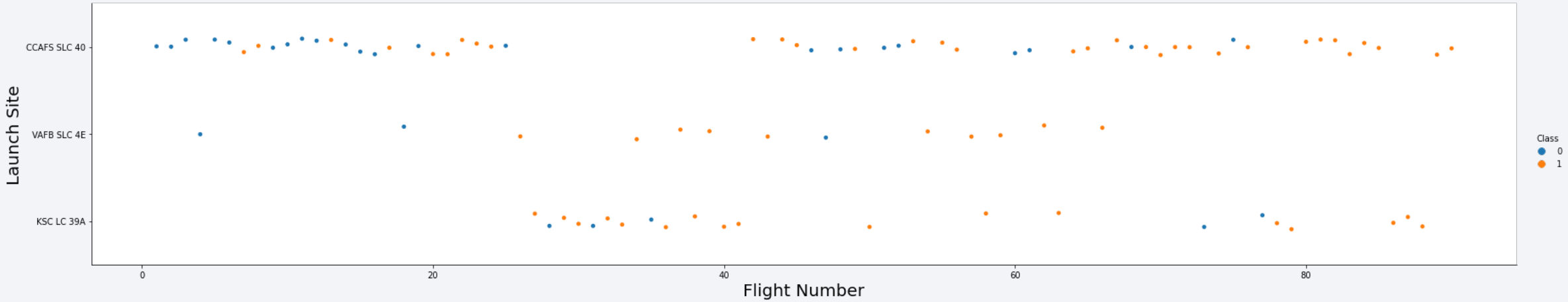
GitHub source code link

Results

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

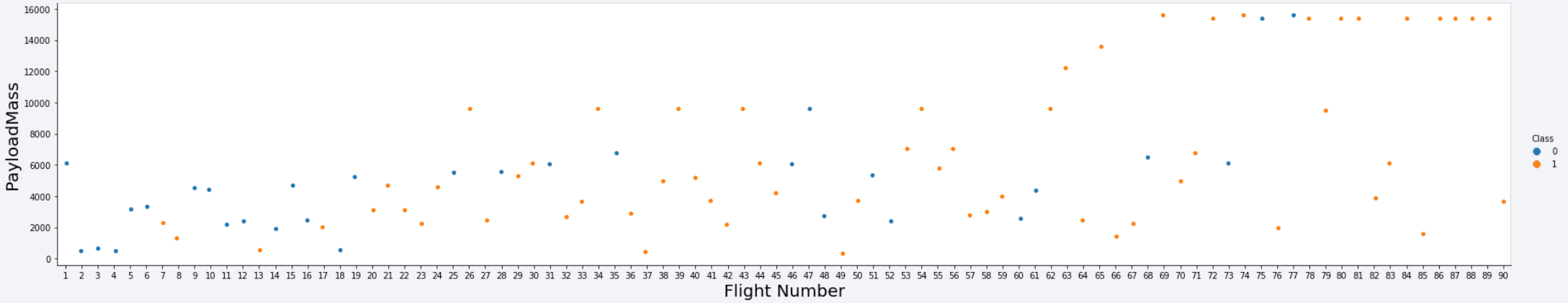

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



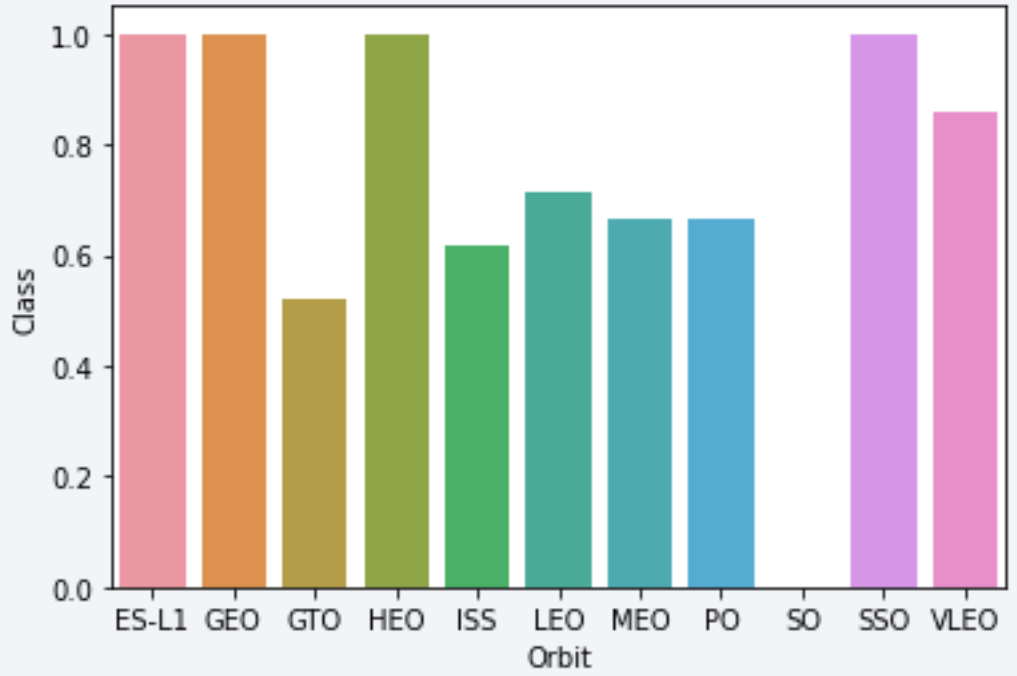
Throughout this plot we see that flight number between 0 and 20 are almost CCAFS LC-40 while the flight number between 20 and 40 are occupied by KSC LC-39A and VAFB SLC 4E. Flight number beyond 40 are almost from CCAFS LC-40 launch site.

Payload vs. Launch Site



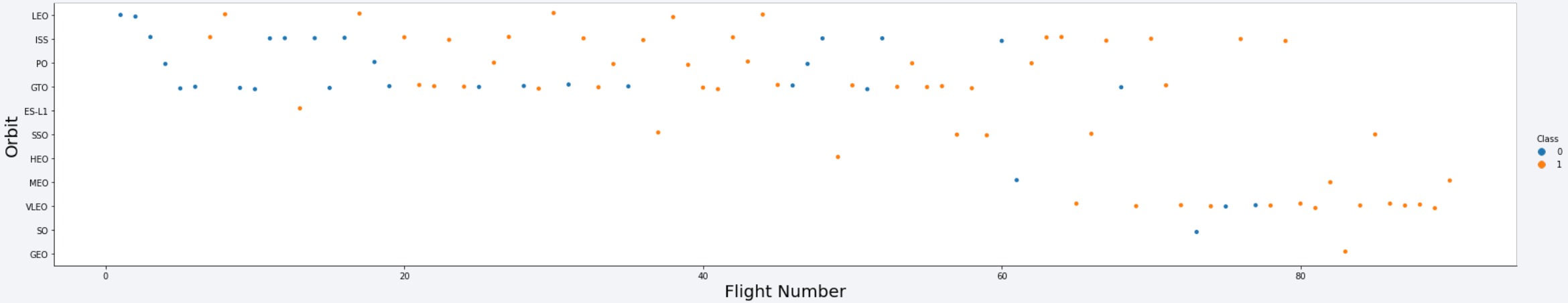
For the VAFB-SLC launchsite there are no rockets launched for heavypayload mass (greater than 10000).

Success Rate vs. Orbit Type



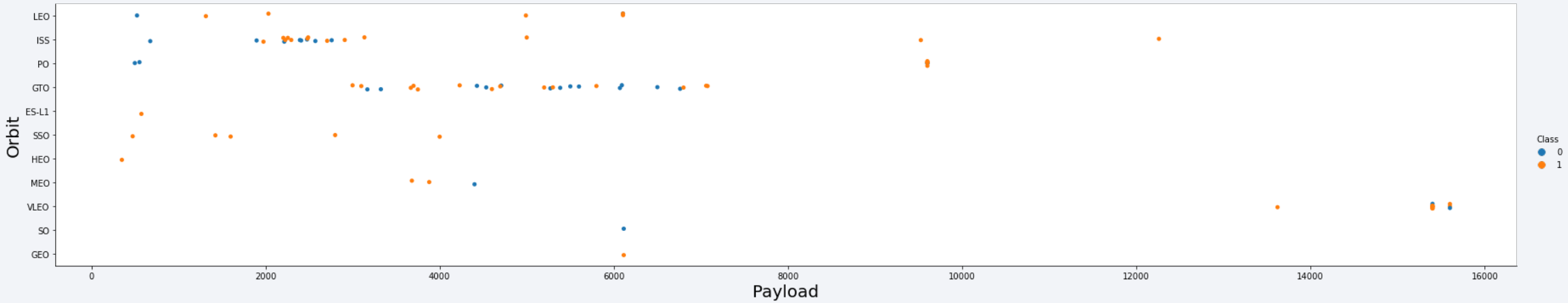
We can observe that the orbit SO doesn't have successful records, other orbits have their successful records, while ES-L1, GEO, HEO and SSO have 100% successful launch records.

Flight Number vs. Orbit Type



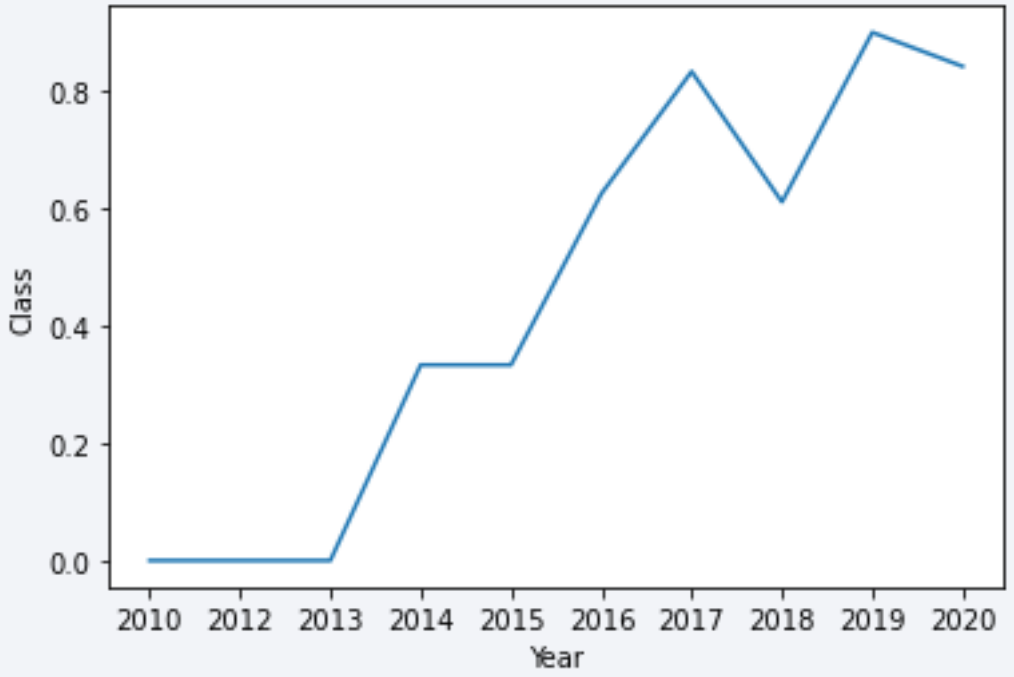
In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



We can observe that the success rate since 2013 kept increasing till 2020.

All Launch Site Names

```
%%sql
SELECT DISTINCT(LAUNCH_SITE)
FROM spacex;
```

```
 * ibm_db_sa://stv34180:***@2d46
Done.
```

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

The function DISTINCT() selects unique name of chose column without any repetition.

Launch Site Names Begin with 'CCA'

In [7]:

```
%%sql
SELECT *
FROM spacex
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT(5);
```

```
 * ibm_db_sa://stv34180:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

Out[7]:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- The function LIMIT() can be used to find limited records with a given number.
- Use % symbol after the string that we want to find can locate all string matches which begin with the string.

Total Payload Mass

```
%%sql
SELECT SUM(payload_mass__kg_) AS "Total_Payload_Mass"
FROM spacex
WHERE customer LIKE 'NASA (CRS)';
```

```
 * ibm_db_sa://stv34180:***@2d46b6b4-cbf6-40eb-bbce-625
Done.
```

Total_Payload_Mass
45596

The function SUM() calculate the sum of payload, whose customer is NASA (CRS), filtered by WHERE clause.

Average Payload Mass by F9 v1.1

```
%%sql
SELECT AVG(payload_mass__kg_) AS "Avg_Payload_Mass"
FROM spacex
WHERE booster_version = 'F9 v1.1';
```

```
 * ibm_db_sa://stv34180:***@2d46b6b4-cbf6-40eb-bbce-6251e
Done.
```

Avg_Payload_Mass
2928

The function AVG() is used to calculate the average of payload whose booster version is F9 v1.1, filtered by WHERE clause.

First Successful Ground Landing Date

```
%%sql

SELECT MIN(DATE) AS "First_successful_date"
FROM spacex
WHERE landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://stv34180:***@2d46b6b4-cbf6-40eb-bbce-6
Done.
```

First_successful_date
2015-12-22

The MIN() function allows us to find the first landing late and the result records of success is filtered by WHERE clause.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT booster_version
FROM spacex
WHERE payload_mass__kg_ BETWEEN 4000 AND 6000 AND landing__outcome = 'Success (drone ship)';
```

```
 * ibm_db_sa://stv34180:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases
Done.
```

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

We select booster version with related conditions using WHERE clause, and the numerical limits can be defined by BETWEEN attribute.

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT
(SELECT COUNT(MISSION_OUTCOME) FROM spacex WHERE MISSION_OUTCOME LIKE '%Success%') AS success,
(SELECT COUNT(MISSION_OUTCOME) FROM spacex WHERE MISSION_OUTCOME LIKE '%Failure%') AS failure
FROM spacex
LIMIT 1;
```

```
 * ibm_db_sa://stv34180:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.ap
Done.
```

success	failure
100	1

Two subqueries are used to pre-select success and failure outcomes respectively, and final result is filtered with a single column displayed by LIMIT() function in order to remove the redundancy of results.

Boosters Carried Maximum Payload

```
%%sql
SELECT BOOSTER_VERSION AS max_booster, payload_mass__kg_ AS payload_mass
FROM spacex
WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM spacex);
```

```
 * ibm_db_sa://stv34180:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08k
Done.
```

max_booster	payload_mass
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

A subquery is used to pre-select the max value of payload. Then we find the booster version which has the same max payload value.

2015 Launch Records

```
%%sql
SELECT date, booster_version, launch_site, landing__outcome
FROM spacex
WHERE year(date) = 2015 AND landing__outcome = 'Failure (drone ship)';
```

```
 * ibm_db_sa://stv34180:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08
Done.
```

DATE	booster_version	launch_site	landing__outcome
2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

The function YEAR() allows to extract the year of a datetime object.

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

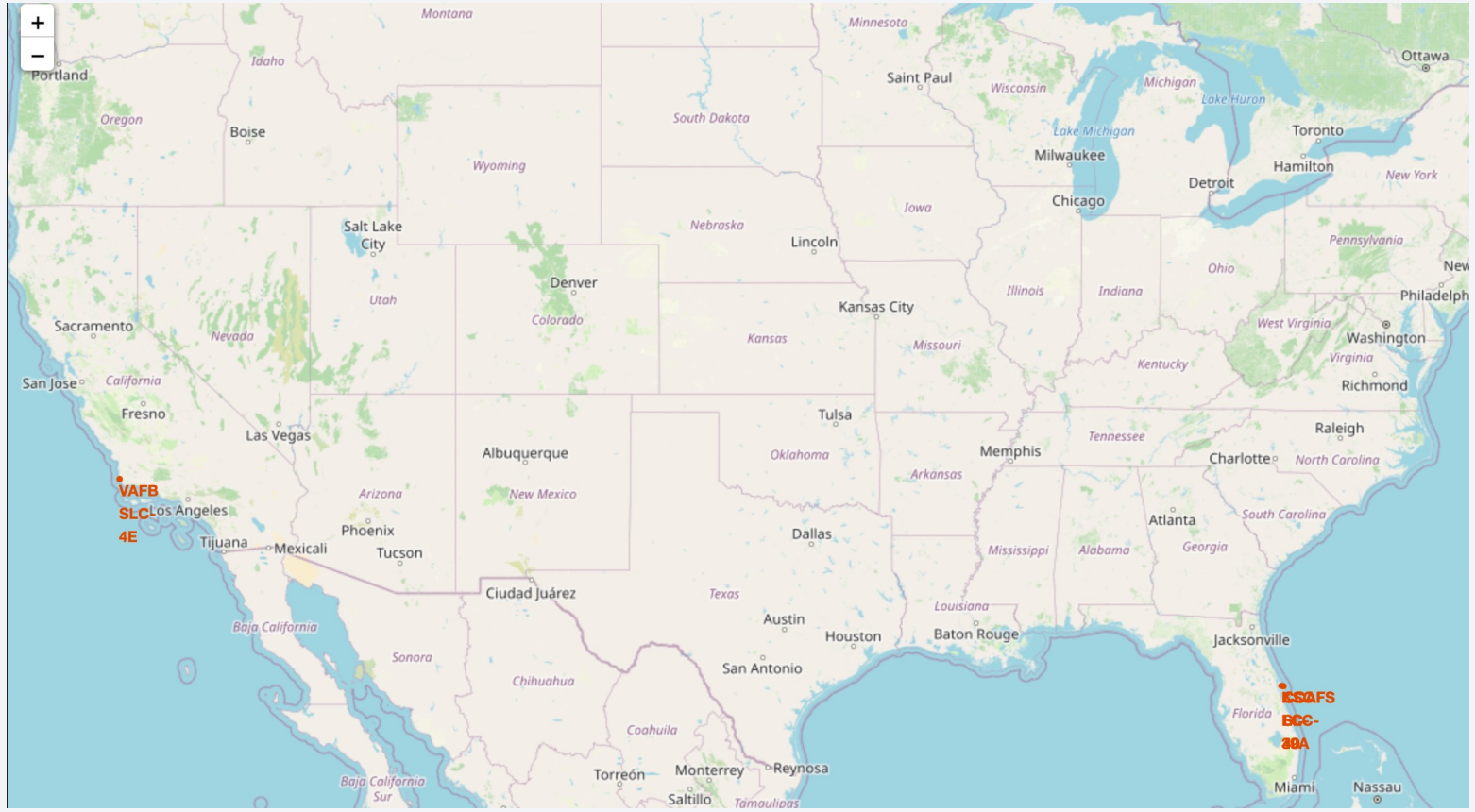
```
%%sql

SELECT date, booster_version, launch_site, landing__outcome
FROM spacex
WHERE date BETWEEN '2010-06-04' AND '2017-03-20'
```

We are able to select a detailed date range using BETWEEN in WHERE clause.

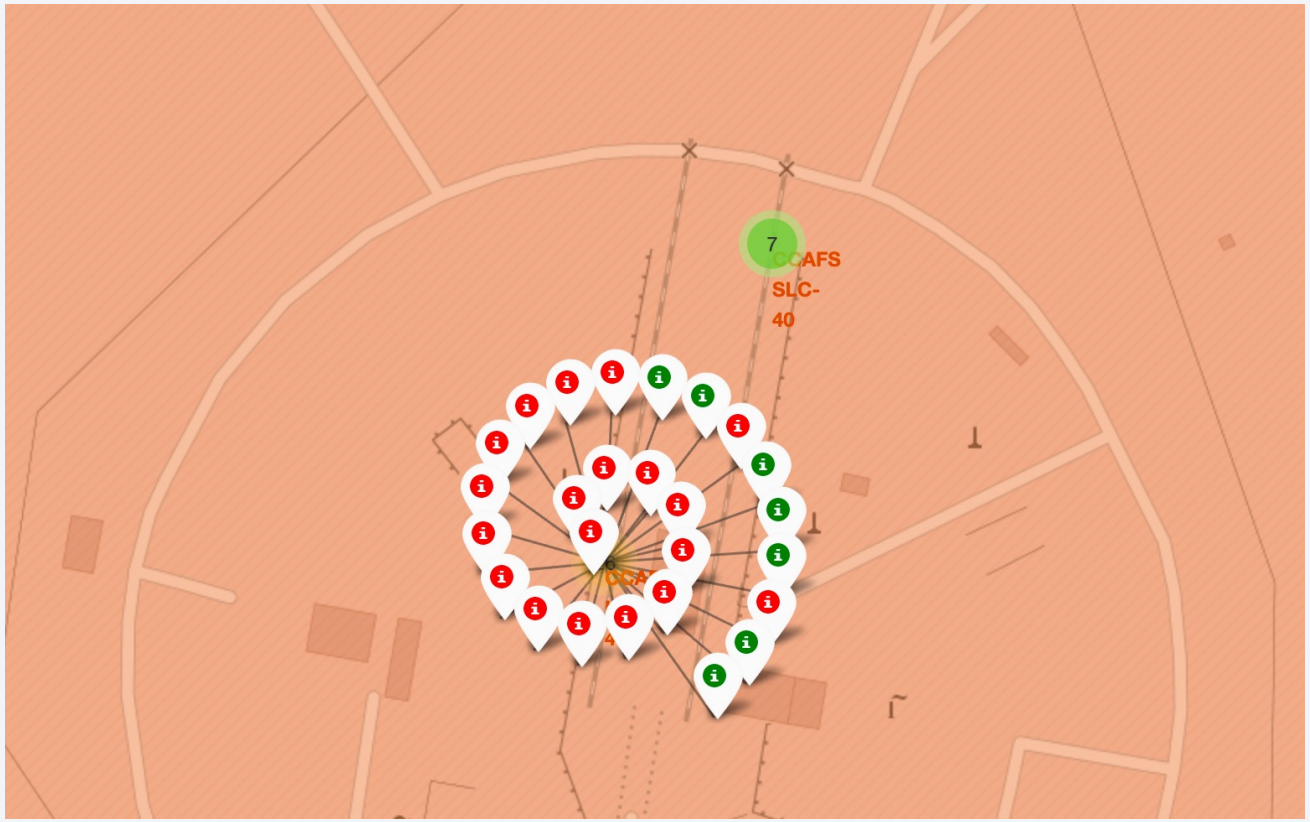

Section 3
Launch Sites
Proximities Analysis

All launch sites on a map



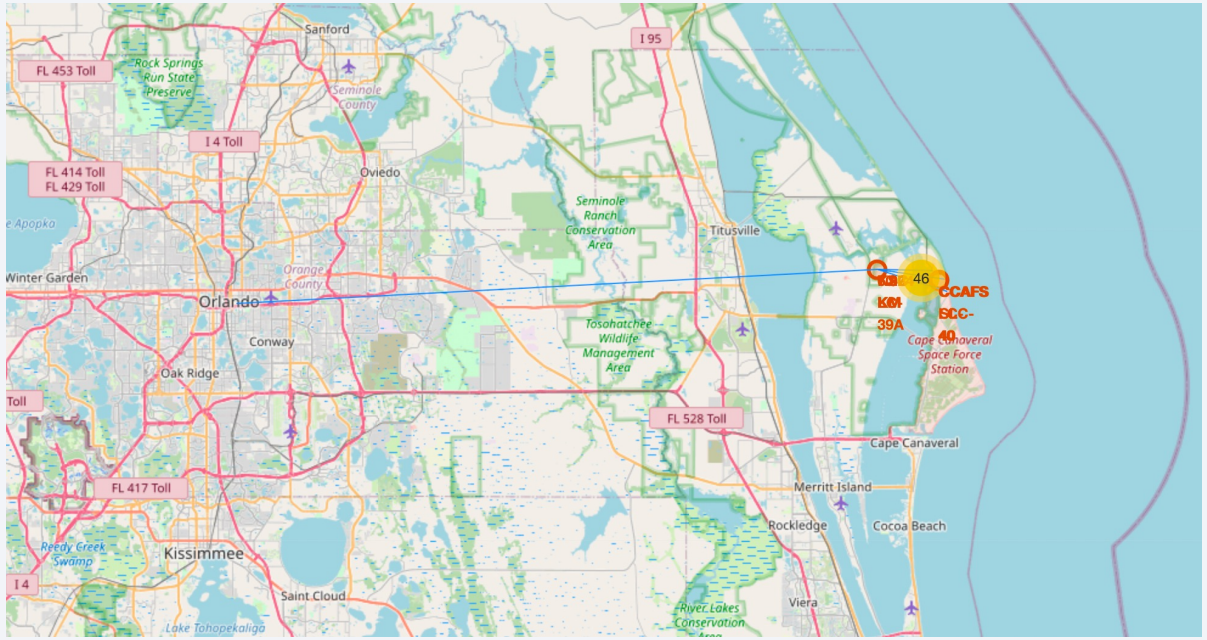
We can see that all launch sites are in very close proximity to the coast and close to the Equator Line.

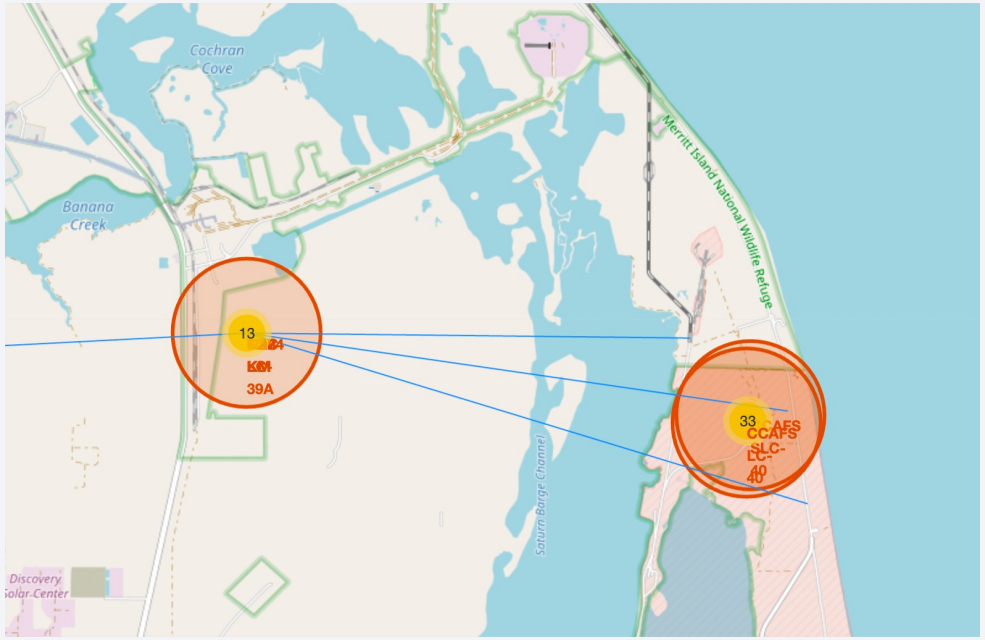
Success and failed launches for each site



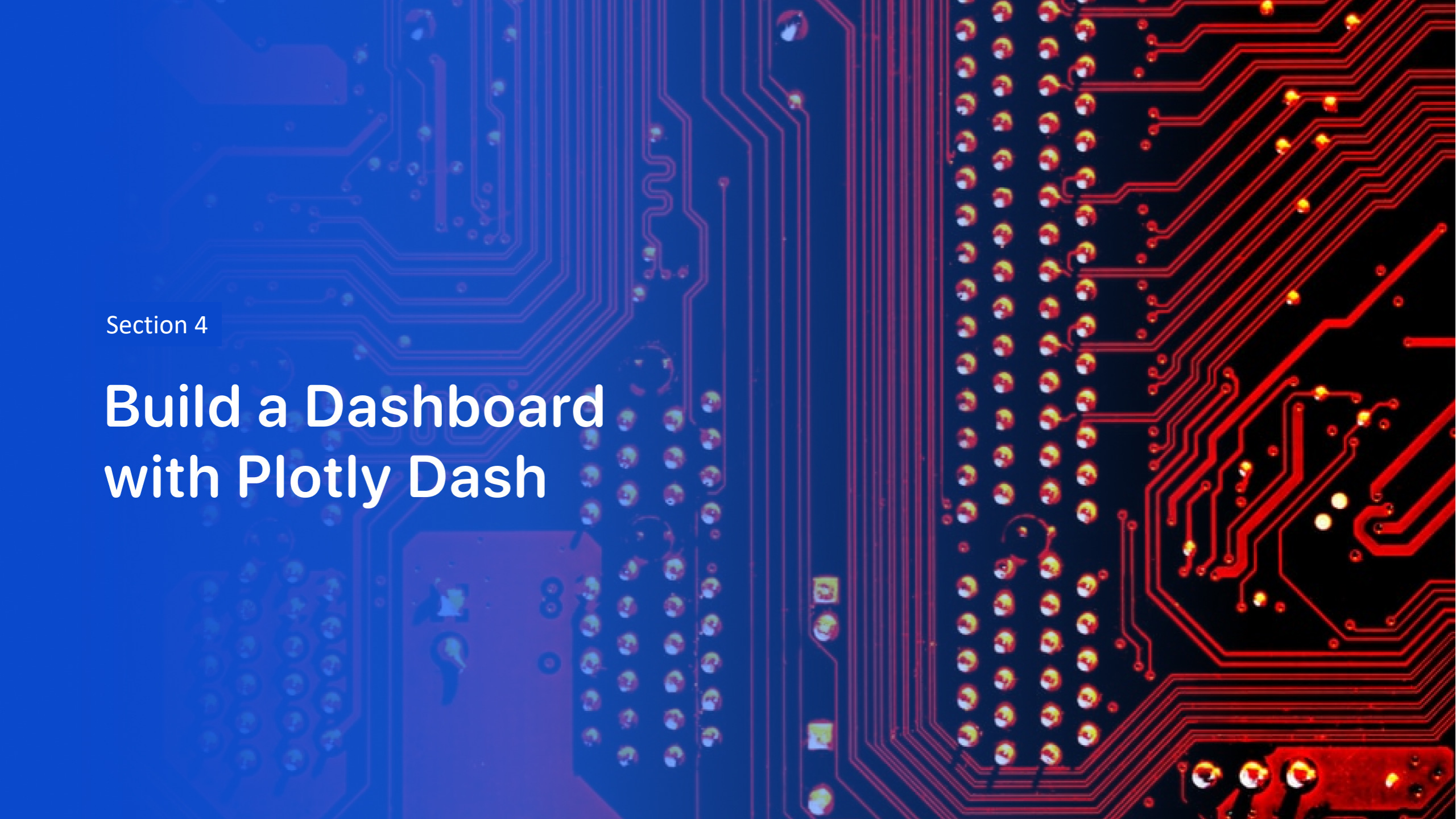
From the color-labeled markers in marker clusters, we can easily identify which launch sites have relatively high success rates.

Distances between a launch site to its proximities

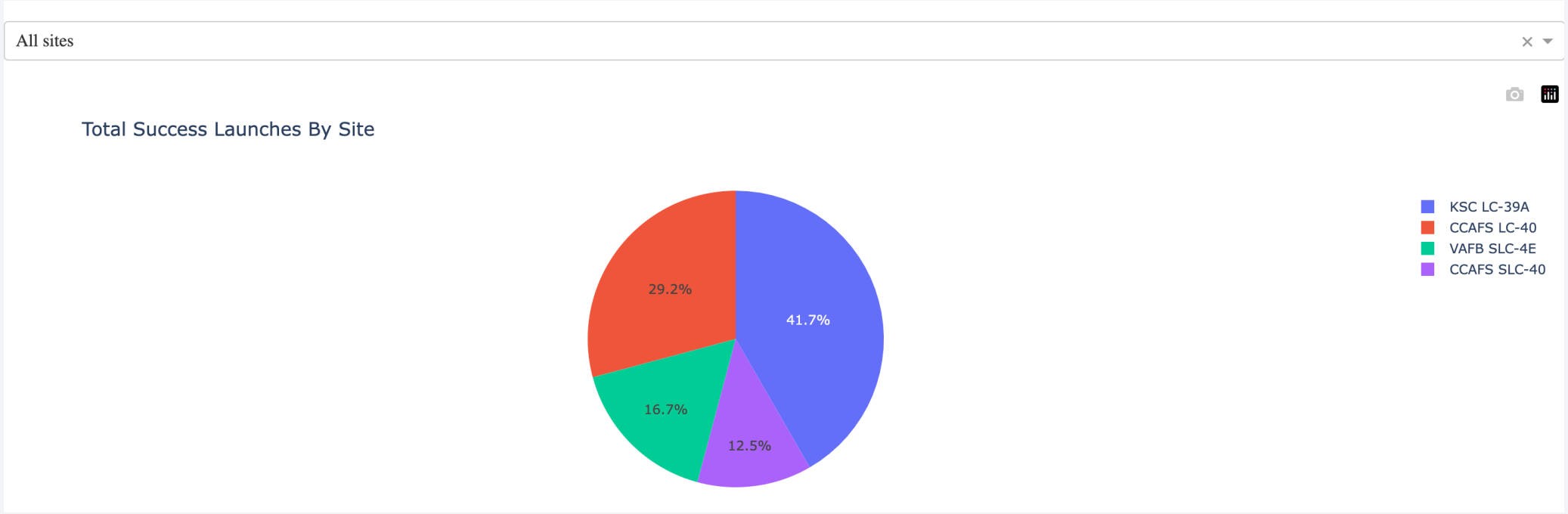




Using launch site KSC LC-39A as example, it is close to railway and highway but far away from city.

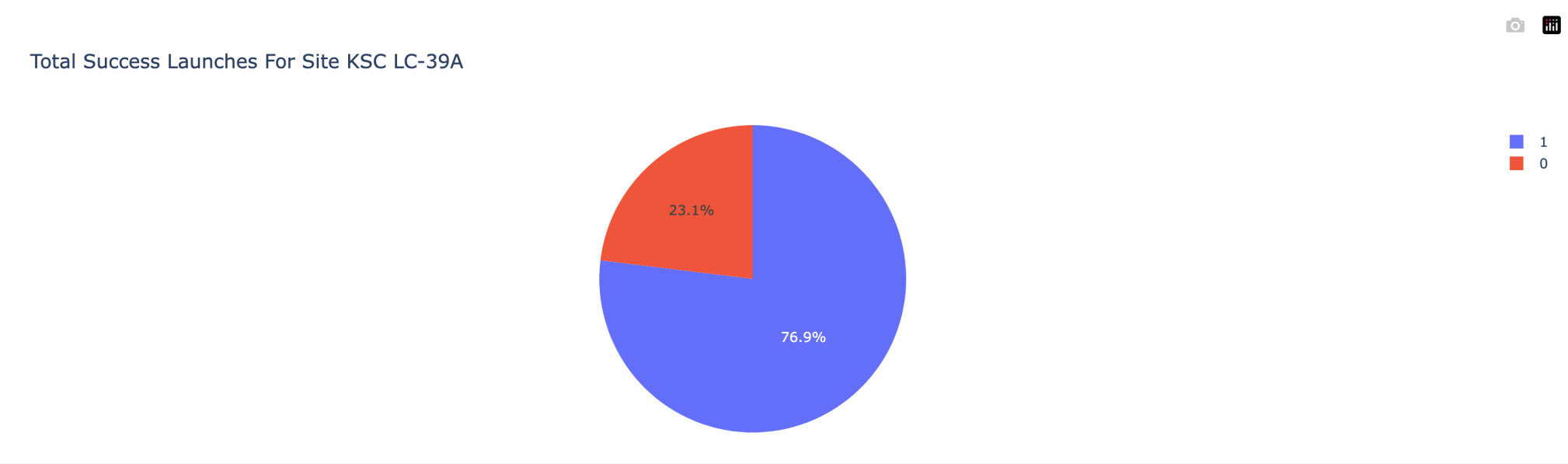

Section 4
Build a Dashboard with Plotly Dash

Total success launches by site



We can see that KSC LC-39A has the most success launches (41.7%) among all of success launches sites., compared to CCAFS SLC-40 with the smallest amount (12.5%).

The most successful launch site



KSC LC-39A is the launch site with highest launch success ratio, i.e., 76.9%.

Correlation between payload and success for all sites

From the scatter plot with different selected payload ranges, we could observe that the booster version v1.1 has more failed launch, while FT tends to have more successful launches.

Section 5

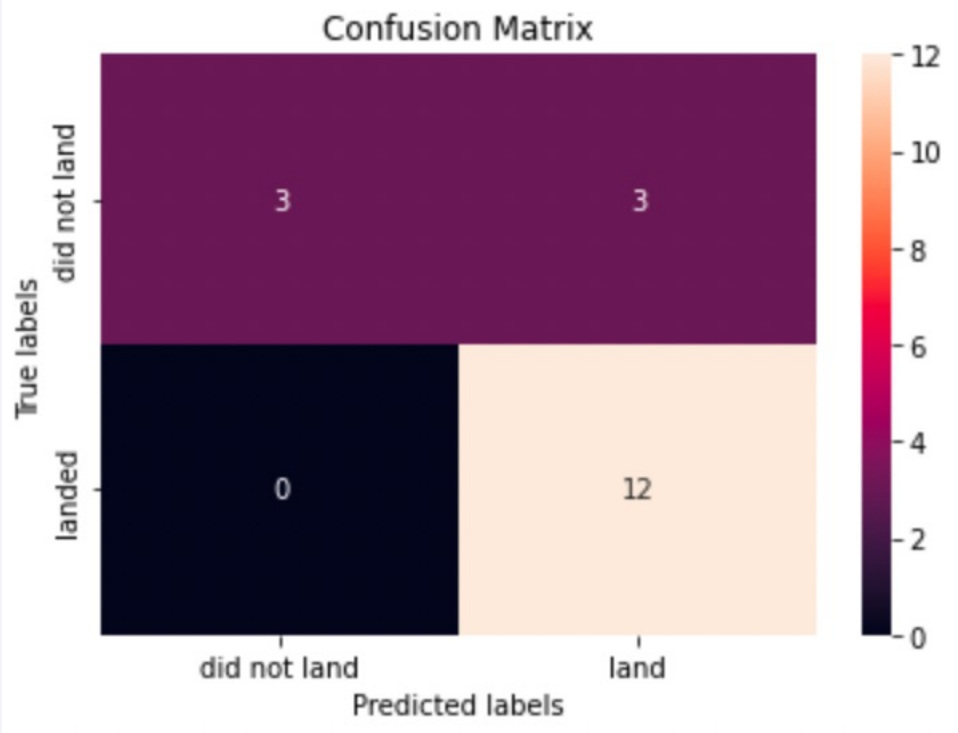
Predictive Analysis (Classification)

Classification Accuracy

KNN, SVM and Logistic Regression have the highest classification accuracy, while the accuracy of Decision Tree is lower.

Confusion Matrix

The figure on the right side shows the confusion matrix of built KNN model, we can see that the model generally predicts well for test data, but it made wrong predictions for some records didn't land successfully (3 out of 6).



Conclusions

- In order to build a robust machine learning model used for prediction, it's significant and necessary to take much time to preprocess and clean data.
- The classification models that we have tested perform well for prediction in a general way. KNN, SVM and Logistic Regression have the highest classification accuracy, while the accuracy of Decision Tree is lower.

Appendix

- All files of this project are available on the GitHub repository.

Thank you!