IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rodolfo Teixeira
24/07/2023

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data Collection through API and Web Scaping.
  - Data Wrangling.
  - Exploratory Analysis Using SQL, Pandas and Matplotlib.
  - Interactive Visual Analytics and Dashboard
  - Predictive Analysis (Classification)
- **Summary of all results**
  - Exploratory Data Analysis
  - Predictive Analytics Results from supervised machine learning models (classification)

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. Therefore, 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 space X for a rocket launch. The goal of the project is to develop a machine learning model to predict if the first stage will land successfully.

- **Problems you want to find answers**

  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
  - The data was collected using SpaceX API and Wikipedia Web scraping
- **Perform data wrangling**
  - Doe to the presence of categorical features it was needed to implement one-hot encoding. It was also dealt with the missing values of the Payload mass. The values were replaced using the 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**
  - How to build, tune, evaluate classification models

Data Collection

- Data collection was done using get request to the SpaceX API.
- In order to get a pandas dataframe, the response content was decoded as a Json using .json() method and then .json_normalize().
- Then, the data was cleaned, checked for missing values and fill in missing values where necessary.
- In addition, web scraping was performed from Wikipedia for Falcon 9 launch records with BeautifulSoup.
- To get a pandas dataframe from the web page, the launches info was retrieved as an HTML table, then parsed and converted into a pandas dataframe.

Data Collection – SpaceX API

- Start by requesting rocket launch data from SpaceX API with the URL

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

```
response = requests.get(spacex_url)
```

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

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appd
```

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

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

https://github.com/RRFTcap/IBM-Data-Science/blob/main/Lab1_Collecting_the_data.ipynb

Data Collection - Scraping

- Web scrapping was applied to webscrap Falcon 9 launch records with BeautifulSoup
- The table was parsed and converted it into a pandas dataframe.

(See notebook for additional code)

```
response = requests.get(static_url)

soup = BeautifulSoup(response.text, "html.parser")

Extract all column/variable names from the HTML table header

html_tables = soup.findAll("table")

first_launch_table = html_tables[2]

column_names = []

element = first_launch_table.findAll("th")

for th in element:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)

Create a data frame by parsing the launch HTML tables

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']=[]
```

https://github.com/RRFTcap/IBM-Data-Science/blob/main/Lab2_Web_Scaping.ipynb

Data Wrangling

- Exploratory data analysis was performed and determined the training labels.
- The number of launches at each site and the number and occurrence of each orbits were calculated.
- A landing outcome label from outcome column was createad and exported the results to csv.

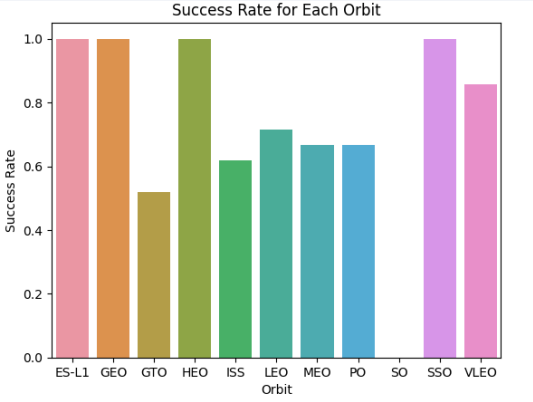
TASK 1: Calculate the number of launches on each site
TASK 2: Calculate the number and occurrence of each orbit
TASK 3: Calculate the number and occurrence of mission outcome per orbit type
TASK 4: Create a landing outcome label from Outcome column

https://github.com/RRFTcap/IBM-Data-Science/blob/main/Lab3_Data_Wrangling.ipynb

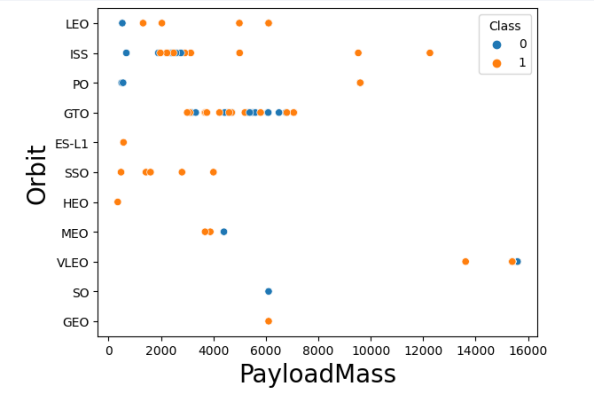
EDA with Data Visualization

- The data was explored by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.

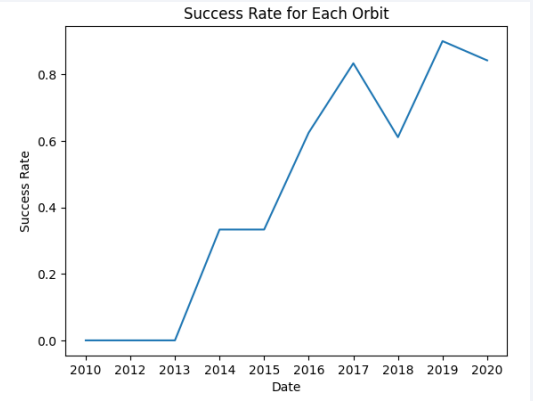
- Bar Chart



- Scatter Plot



- Line Graph



https://github.com/RRFTcap/IBM-Data-Science/blob/main/Lab5_DataViz.ipynb

EDA with SQL

- **SQL queries were executed to gather and understand data from dataset:**

- Displaying 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 achieved.
- 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
- List the records which will display the month names, faiilure landing_ouutcomes in drone ship, booster versions, launch_site for the months in year 2015.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

https://github.com/RRFTcap/IBM-Data-Science/blob/main/Lab4_SQL.ipynb

Build an Interactive Map with Folium

- The launch sites were marked in the folium map.
- Using the color-labeled marker clusters, the launch sites with high success rate were identified.
- The distances between the launch sites and some proximities were calculated and some questions were answered:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.
- These objects are created in order to understand better the problem and the data. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.

https://github.com/RRFTcap/IBM-Data-Science/blob/main/Lab6_Folium.ipynb

Build a Dashboard with Plotly Dash

- **Dashboard with a dropdown, pie chart, rangeslider and scatter plot was developed.**

  - Dropdown allows a user to choose the launch site or all launch sites.
  - Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component.
  - Rangeslider allows a user to select a payload mass in a fixed range.
  - Scatter chart shows the relationship between Success vs Payload Mass.

https://github.com/RRFTcap/IBM-Data-Science/blob/main/Lab7_Plotly_Dash.py

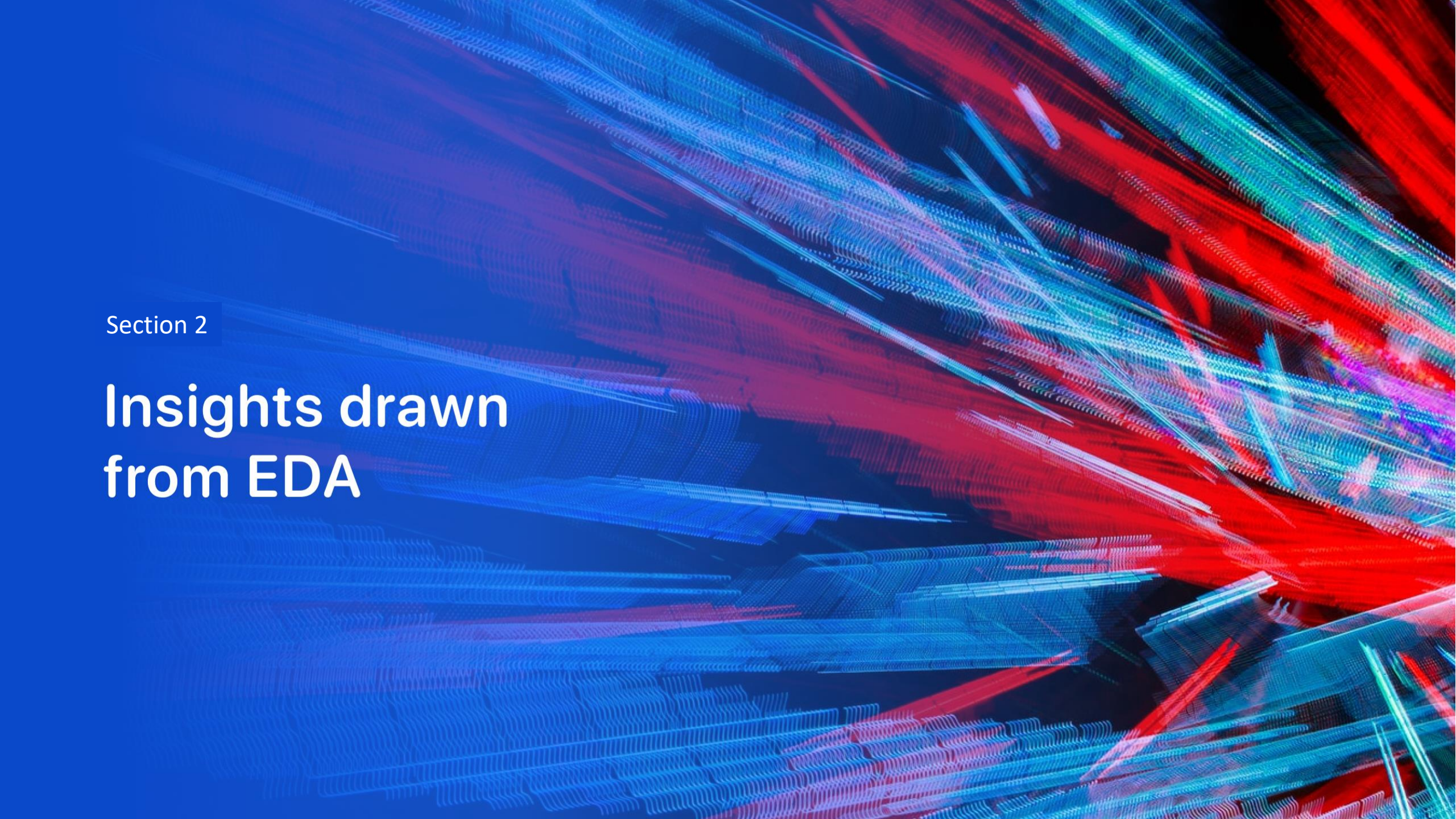
Predictive Analysis (Classification)

- The data was loaded using numpy and pandas, normalized and splitted into training and testing sets.
- Different machine learning models were built and different hyperparameters were tuned using GridSearchCV.
- The models were compared according to their accuracy.
- The model with the best metrics (accuracy and score) was selected.

https://github.com/RRFTcap/IBM-Data-Science/blob/main/Lab8_Classification_Models.ipynb

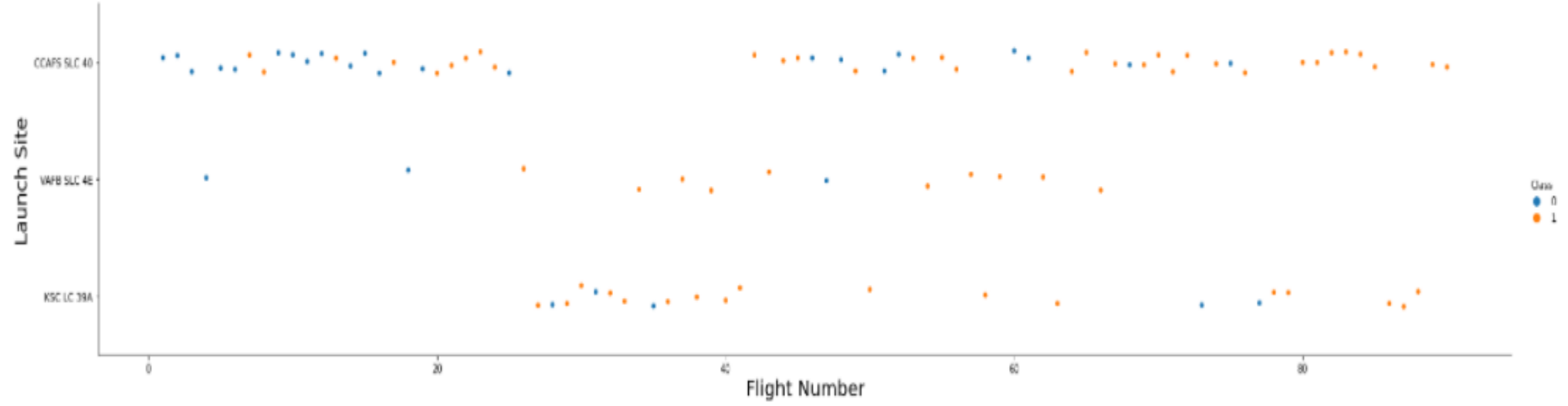
Results

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


Section 2
Insights drawn from EDA

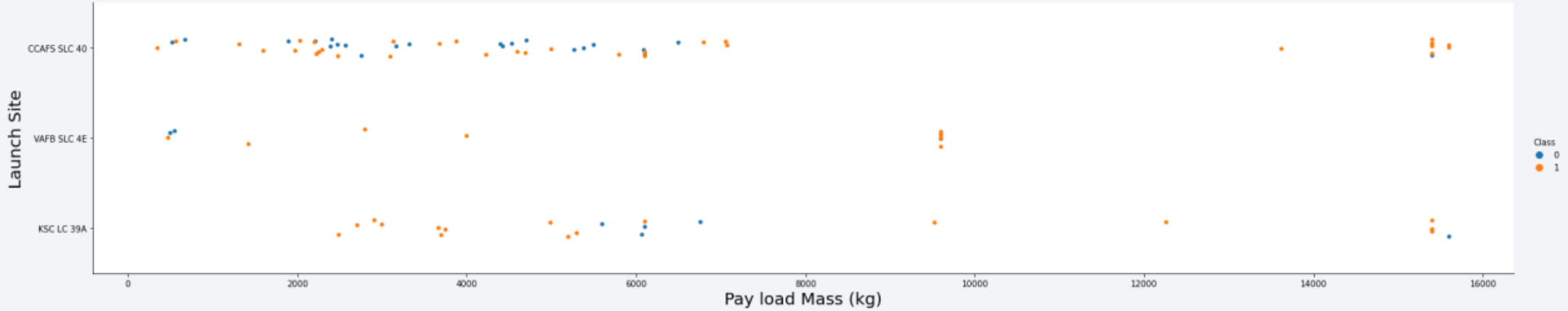
Flight Number vs. Launch Site

- From the plot, it is possible to observe that for each site, increasing the number of flights increases the success rate.



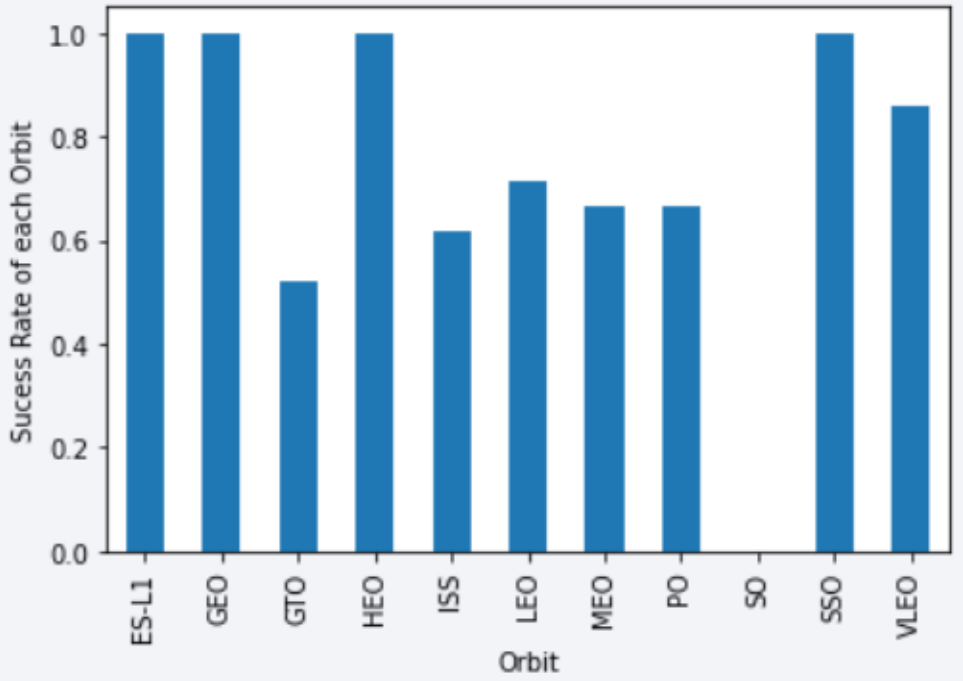
Payload vs. Launch Site

- Across all sites, increasing the Pay Load Mass improves the success of landing.



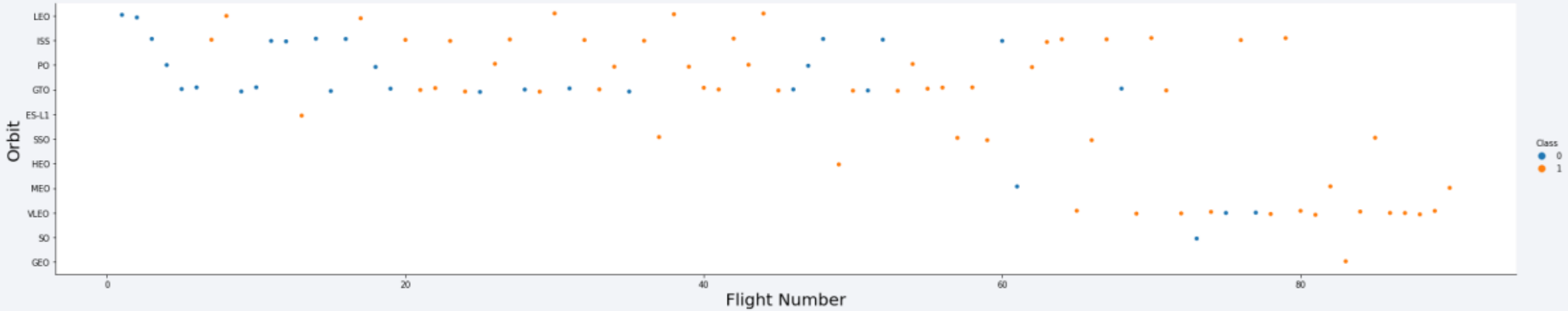
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



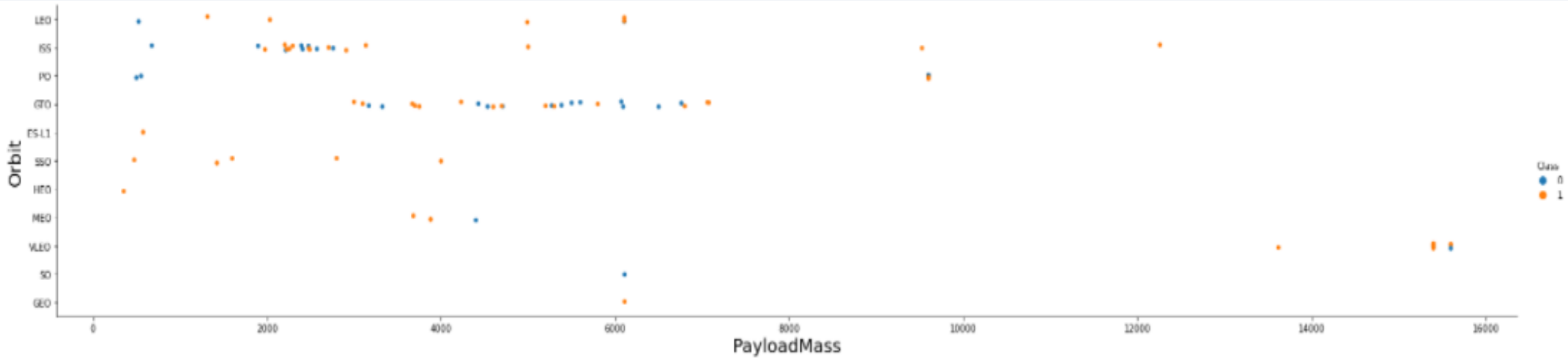
Flight Number vs. Orbit Type

- It is possible to observe that in the LEO orbit, success of landing increases with the number of flights, whereas in the GTO and the ISS orbit, apparently there is no relationship between flight number and the orbit.



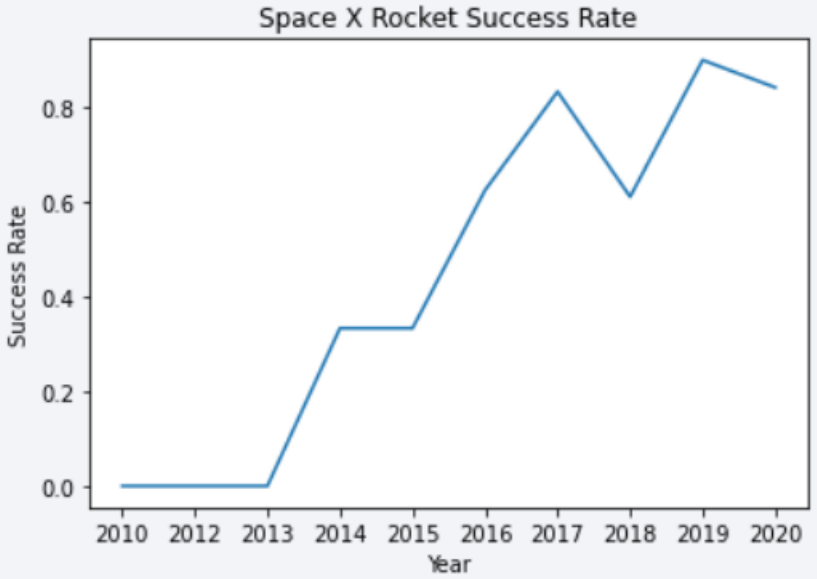
Payload vs. Orbit Type

- It is possible to observer that for the PO, LEO and ISS orbits, the Havier the payload mass have a higher success landing rate.



Launch Success Yearly Trend

- Since 2013, despite the decrease in the year 2018, the SpaceX Rocket Success Rate have been increasing to around 80%.



All Launch Site Names

- The key word DISTINCT was used to show only unique launch sites from the SpaceX data.

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

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
```

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

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

- The WHERE clause followed by LIKE clause filters launch sites that contain the substring “CCA”. LIMIT 5 shows 5 records from filtering.

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

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE '%CCA%' LIMIT 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (paracl
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (paracl
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No att
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No att
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No att

Total Payload Mass

- The query returns the sum of all payload masses where the customer is NASA (CRS).

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer LIKE '%NASA (CRS)%';
```

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

SUM(PAYLOAD_MASS__KG_)
48213.0

Average Payload Mass by F9 v1.1

- The query returns the average of all payload masses where the booster version contains the substring “F9 v1.1”.

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

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

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- With this query, the oldest successful landing was selected. The WHERE clause filters dataset in order to keep only records where landing was successful. With the MIN function, the record with the oldest date was selected. The date format from the column Date needed to be change to YYYY/MM/DD in order to get the earliest successful landing.

```
%%sql
-- Step 1: Create a new temporary column
ALTER TABLE SPACEXTBL ADD COLUMN TempDate TEXT;

-- Step 2: Update values in the temporary column with the new date format
UPDATE SPACEXTBL SET TempDate = SUBSTR(Date, 7, 4) || '/' || SUBSTR(Date, 4, 2) || '/' || SUBSTR(Date, 1, 2);

-- Step 3: Drop the original Date column
ALTER TABLE SPACEXTBL DROP COLUMN Date;

-- Step 4: Rename the temporary column to Date
ALTER TABLE SPACEXTBL RENAME COLUMN TempDate TO Date;
```

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome LIKE '%Success%';
 * sqlite:///my_data1.db
Done.
```

MIN(Date)
2015/12/22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg. The WHERE and AND clauses filter the dataset.

```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome LIKE '%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

- In order to list the total number of successful and failure mission outcomes the UNION key word was used to get the result from both queries.

```
%%sql
SELECT 'Success' AS Mission_Outcome, COUNT(*) AS Total_Count FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Success%'
UNION
SELECT 'Failure' AS Mission_Outcome, COUNT(*) AS Total_Count FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Failure%';
```

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

Mission_Outcome	Total_Count
Failure	1
Success	100

Boosters Carried Maximum Payload

- A subquery was used to filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns the booster version with the heaviest payload mass.

```
%sql SELECT Booster_Version FROM SPACEXTBL Where PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

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

2015 Launch Records

- The query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7, 4) shows year.

```
%%sql
SELECT substr(Date, 4, 2) AS month_name, Booster_Version , Launch_Site , Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome LIKE '%Failure (drone ship)%' AND substr(Date,7,4)='2015';
```

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

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

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

- The query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

```
%%sql

SELECT Landing_Outcome, COUNT(*) AS Outcome_Count
FROM SPACEXTBL
WHERE Date >= '04-06-2010' and Date <= '20-03-2017'
GROUP BY Landing_Outcome
ORDER BY Outcome_Count DESC;
```

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

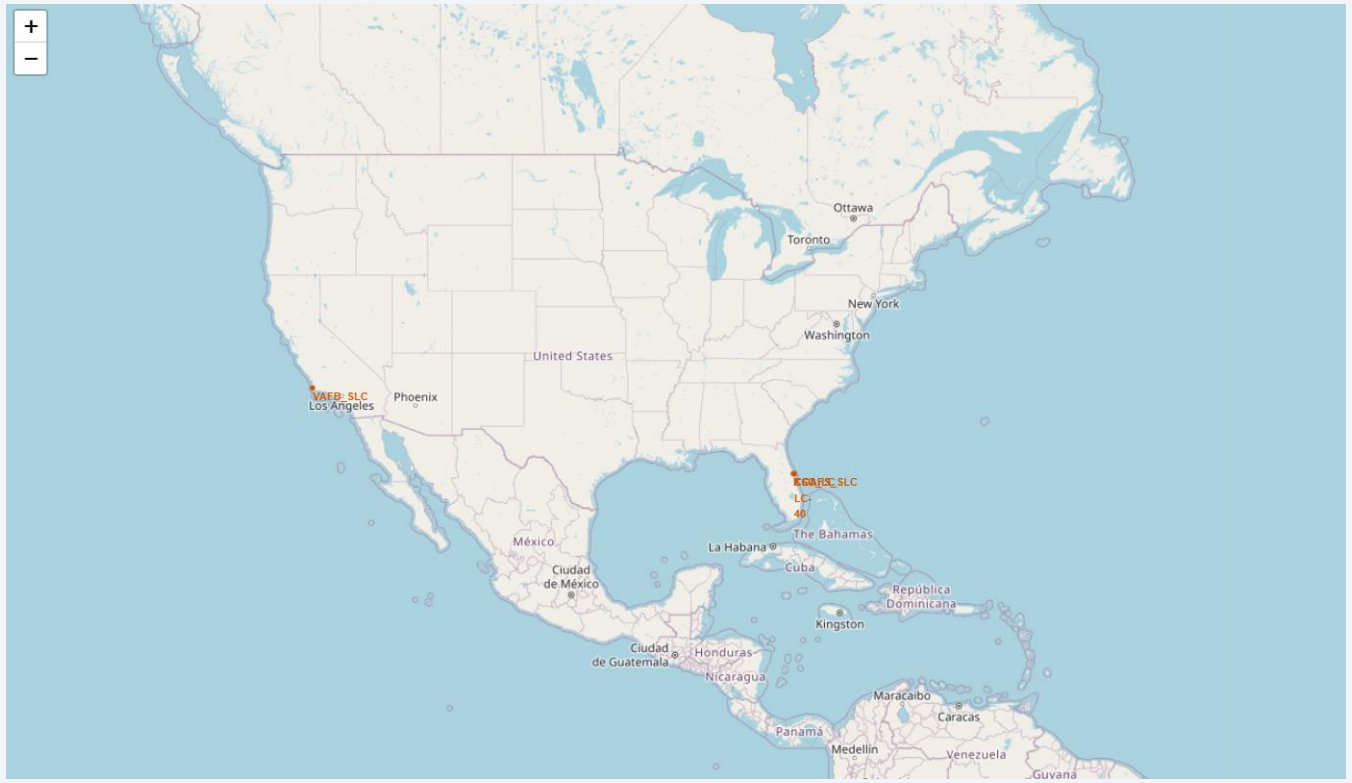
Landing_Outcome	Outcome_Count
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	7
Failure (drone ship)	3
Failure	3
Failure (parachute)	2
Controlled (ocean)	2
No attempt	1

Section 3

Launch Sites Proximities Analysis

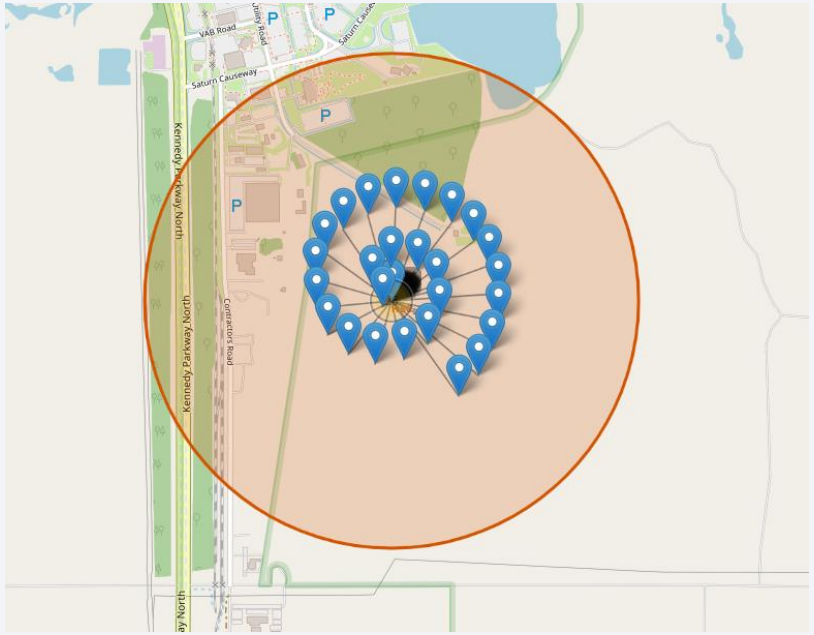
Folium map – Ground stations

- It is possible to see that Space X launch sites are located on the coast of the United States.



Folium map – Color Labeled Markers

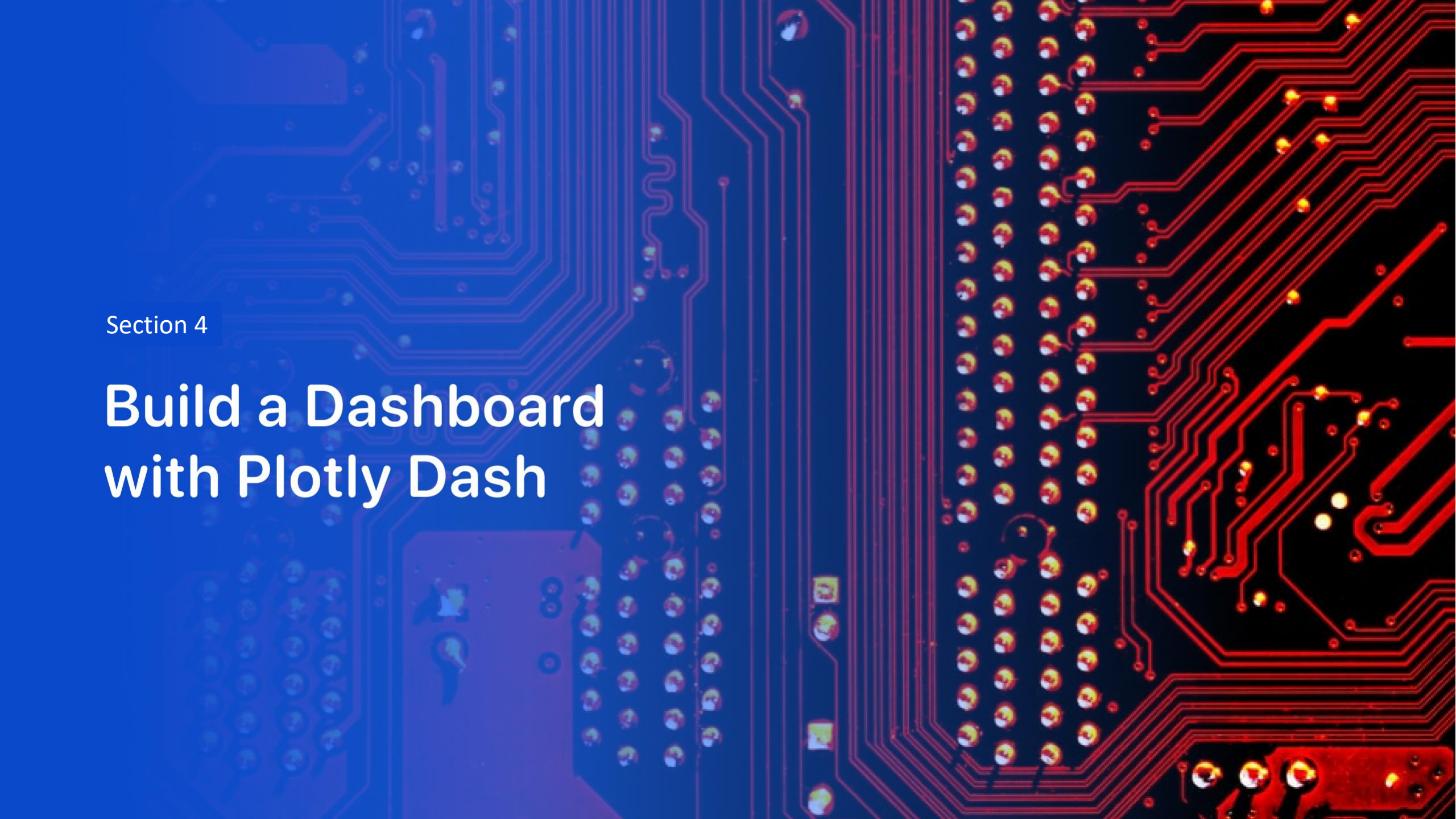
- Despite the code, it was not possible to change the colors of the markers due to outdate libraries. It is possible to notice the markers representing each launch.



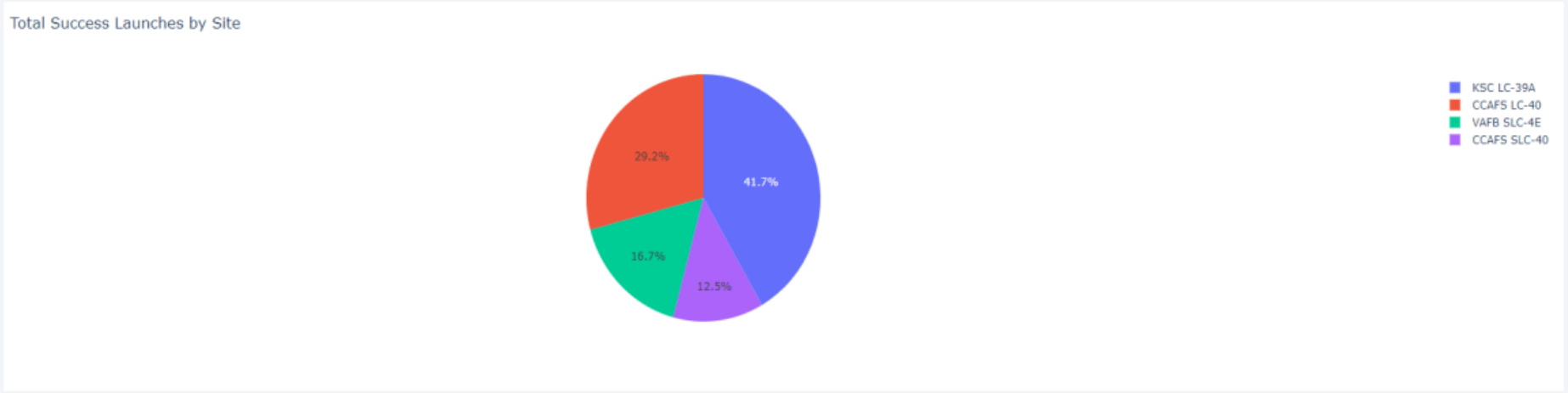
Folium Map – Distances between CCAFS SLC-40 and its proximities

Folium Map – Distances between CCAFS SLC-40 and its proximities

Is CCAFS SLC-40 in close proximity to railways ? Yes
Is CCAFS SLC-40 in close proximity to highways ? Yes
Is CCAFS SLC-40 in close proximity to coastline ? Yes
Do CCAFS SLC-40 keeps certain distance away from cities ? No

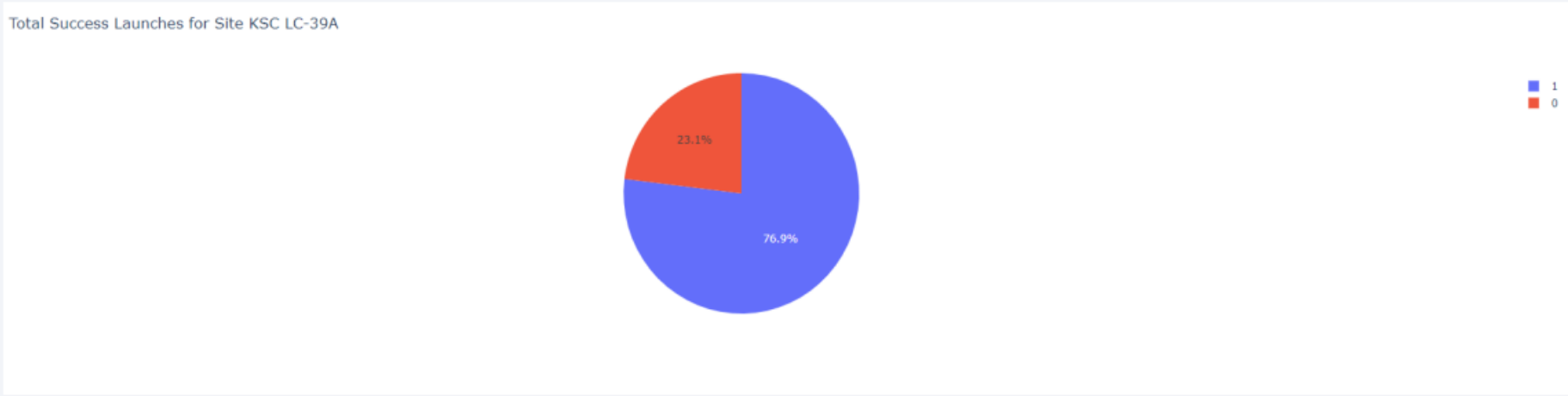

Section 4
Build a Dashboard with Plotly Dash

Dashboard – Total success by Site



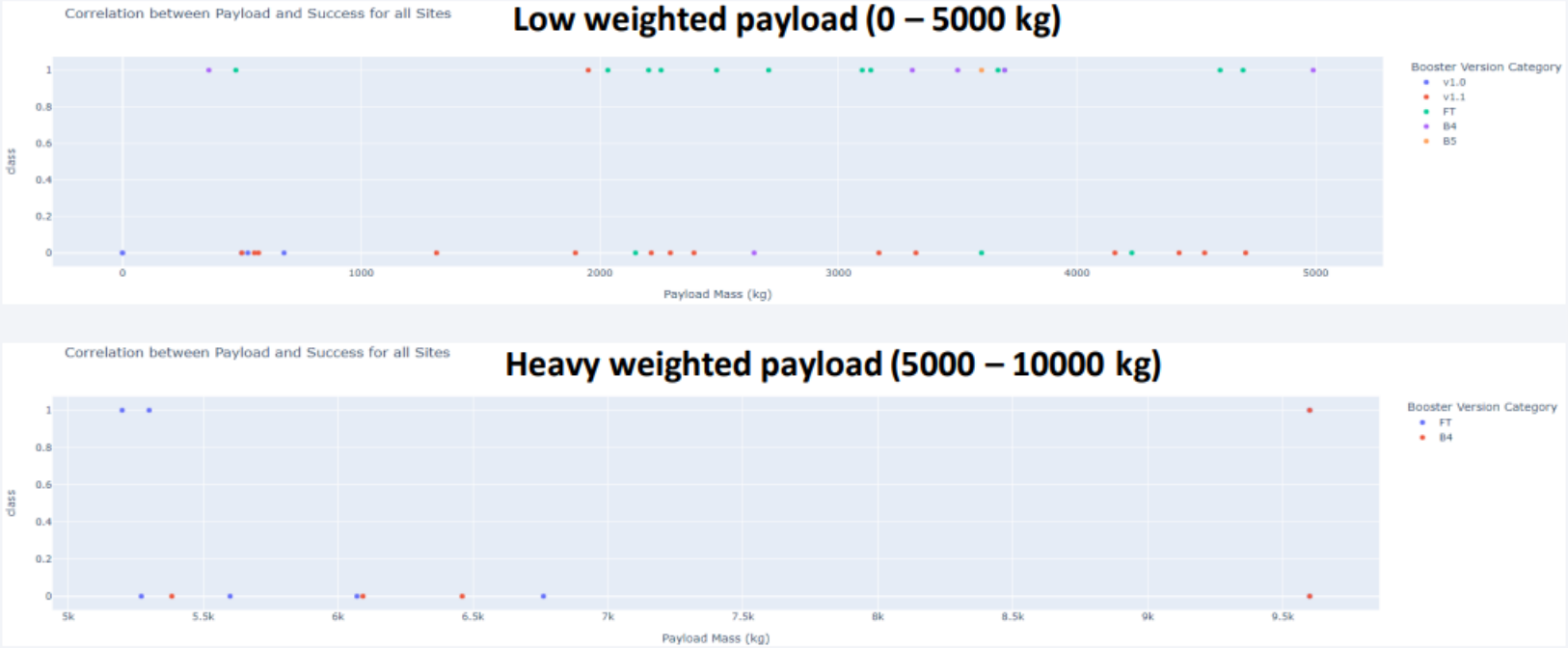
KSC LC-39A has the best success rate of launches.

Dashboard – Total success launches for Site KSC LC-39A



KSC LC-39A has achieved a 76.9% success rate while getting a 23.1% failure rate.

Dashboard – Payload mass vs Outcome for all sites with different payload mass selected



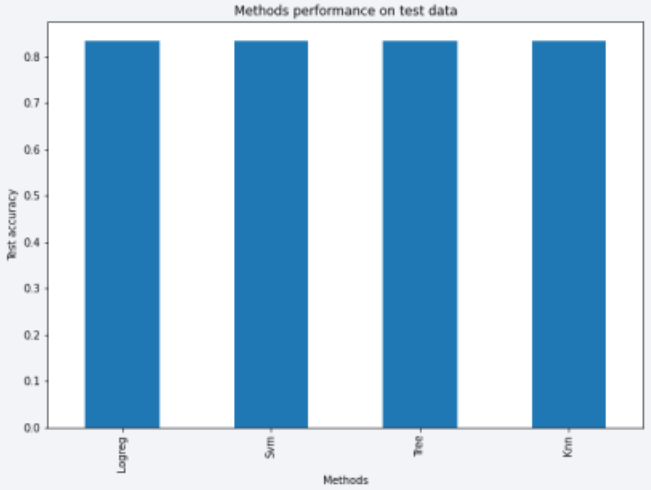
The success rates for heavier payloads is lower than for lighter ones.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All the models had the same accuracy. More data or further tunning of the hyper parameters was needed.



Confusion Matrix

- All the models yield the same Confusion Matrix. The main problem were the false positives.

Conclusions

It is possible to conclude that:

- Increasing the amount of flights at a lunch site increases the success rate of the launch.
- There is a big increase in the launch success rate since 2013.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission.
- KSC LC-39A had the most successful launches of any sites.
- More data or further tunning of the hyper parameters is needed to find a standout model.

Thank you!