IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anthony Remichris

June 23, 2023

Outline

EXECUTIVE SUMMARY

INTRODUCTION

METHODOLOGY

RESULTS

CONCLUSION

APPENDIX

Executive Summary

Summary of methodologies

Data Collection

Data Wrangling

EDA with SQL

EDA with Data Visualization

Building an Interactive Map with Folium

Building a Dashboard with Plotly Dash

Predictive Analysis (Classification)

Summary of all results

EDA Results

Interactive Analytics

Predictive Analysis

Introduction

Project background and context

On its website, SpaceX promotes Falcon 9 rocket launches at a price of 62 million dollars. In contrast, other providers charge upwards of 165 million dollars for each launch. A significant portion of the cost savings comes from SpaceX's ability to reuse the first stage of the rocket.

Problems you want to find answers

The project task is to predict if the Falcon 9 first stage rocket will land successfully.

Section 1

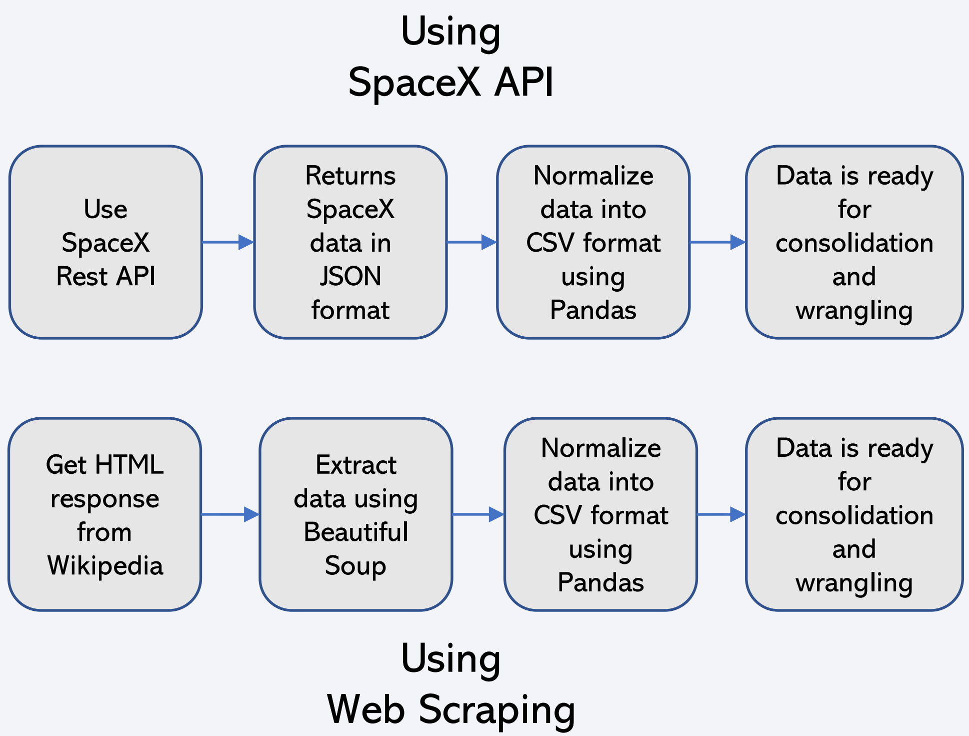
Methodology

Methodology

- Executive Summary
- Data collection methodology:
    - SpaceX Rest API
    - Web Scrapping from Wikipedia
- Perform data wrangling
    - The process involves removing irrelevant columns and handling null values in the data, as well as applying one-hot encoding to categorical variables in preparation for machine learning.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - The construction and assessment of Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), and Decision Tree (DT) models has been done to determine the optimal classifier.

Data Collection

- The following datasets were collected:
  - SpaceX launch data that was extracted from the SpaceX REST API.
  - The API provides us with information on launches, encompassing details about the rocket employed, launch specifications, landing specifications, payload transported, and ultimately, the outcome of the landing.
  - Web scraping Wikipedia using Beautiful Soup is another popular method for obtaining Falcon 9 launch data, serving as a valuable data source.



Data Collection – SpaceX API

- Data collection with SpaceX REST calls is by the side:
- GitHub URL of the completed SpaceX API calls notebook:
  - https://github.com/AnthonyRC7/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/9c8f9c979733139b63e44ee0542da7a90a2b617e/Collecting%20the%20data.ipynb

1. Getting Response from API

```
data = pd.json_normalize(response.json())
```

2. Converting Response to a JSON file

```
spacex_url=https://api.spacexdata.com/v4/launches/past
response = requests.get(spacex_url)
```

3. Applying custom functions to clean data

```
getBoosterVersion(data)          getPayloadData(data)
getLaunchSite(data)              getCoreData(data)
```

4. Constructing our dataset using the cleaned data

```
launch_dict = {'FlightNumber': Listing(data['flight_number']),
'Date': Listing(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}
```

```
main_data = pd.DataFrame(launch_dict)
```

5. Filter data frame and save it in a CSV file

```
data_falcon9 = main_data[main_data["BoosterVersion"] == "Falcon 9"]
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

- Web Scraping from Wikipedia

- GitHub URL of the completed web scraping notebook:
    - https://github.com/AnthonyRC7/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/9c8f9c979733139b63e44ee0542da7a90a2b617e/Web%20scraping%20Falcon%209%20and%20Falcon%20Heavy%20Launches%20Records%20from%20Wikipedia.ipynb

1. Getting Response from HTML and creating a Beautiful Soup Object

```
html_data = requests.get(static_url).text
soup = BeautifulSoup(html_data, "html.parser")
```

2. Finding tables and getting column names

```
tables = soup.find_all("table")
column_names = []
for row in first_launch_table.tbody.tr.find_all("th", {"scope": "col"}):
    column_name = extract_column_from_header(row)
    if column_name:
        column_names.append(column_name)
```

3. Create a data frame by parsing the launch HTML tables

```
launch_dict= dict.fromkeys(column_names)
del launch_dict['Date and time ( )']
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'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

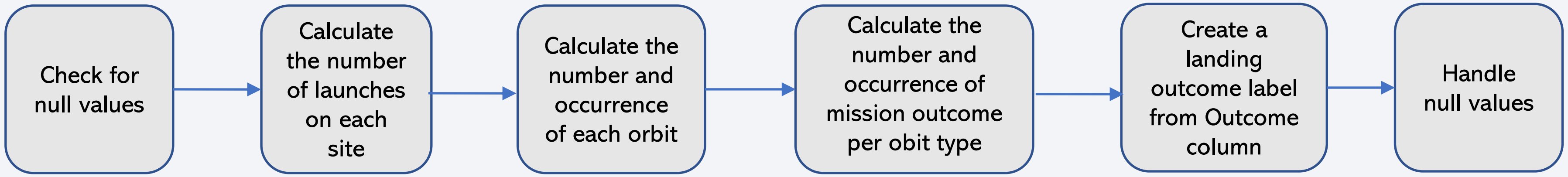
4. Appending data to keys (Code is too long. You can refer to the notebook)

5. Converting dictionary to dataframe and saving the dataframe in a CSV file.

```
df=pd.DataFrame(launch_dict)
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

Exploratory Data Analysis (EDA)



- GitHub URL of the completed data wrangling notebook:
    - https://github.com/AnthonyRC7/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/9c8f9c979733139b63e44ee0542da7a90a2b617e/Data%20Wrangling.ipynb

EDA with Data Visualization

The plotted charts and the reasons for using them:

Scatter chart: The reason for using this type of plot is to explore and visualize the relationship between the variables (Launch Site, Pay Load Mass, Flight Number, etc.)

Bar chart: The reason for using this type of plot is to compare, rank, and visualize the Success rate and Orbit

Line chart: The reason for using this type of plot is to visualize the patterns and Cyclical Behavior of the Success rate over time

The GitHub URL of EDA with data visualization notebook:

https://github.com/AnthonyRC7/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/9c8f9c979733139b63e44ee0542da7a90a2b617e/Exploratory%20Data%20Analysis%20with%20Visualization.ipynb

EDA with SQL

- **SQL queries performed include:**
    - 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. Use a subquery
    - Listing the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
    - Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.
- https://github.com/AnthonyRC7/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/427de0caa4232720cebf498bfa6afbc40052d9c2/Exploratory%20Data%20Analysis%20with%20SQL.ipynb

Build an Interactive Map with Folium

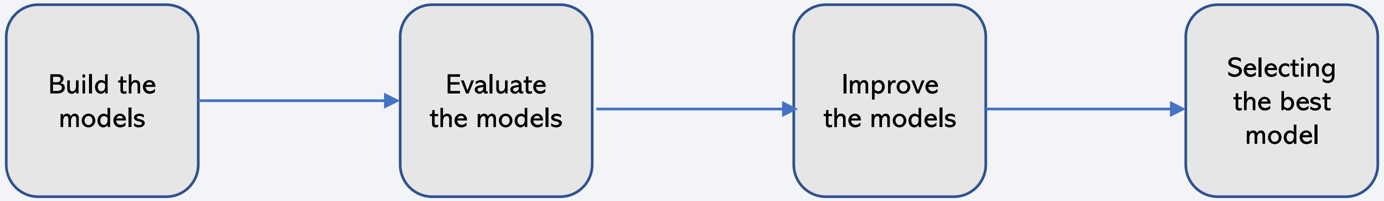
- Map objects that were created and added to the folium map:
  - Map circles have been created to add a highlighted circle area with a text label on a specific coordinate
  - Map markers have been added in order to find an optimal location for building a launch site.
  - Map lines have been added to connect the distance between a launch site and its closest geographic features (city, railway, highway, etc.) with a continuous line segment.
  - Map icons have been added to indicate the number of successful and failed launches in a launch site
- GitHub URL of the completed interactive map with Folium map:
  - https://github.com/AnthonyRC7/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/427de0caa4232720cebf498bfa6afbc40052d9c2/Launch%20Sites%20Locations%20Analysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- The plots/graphs and interactions that have been added to the dashboard include:
    - Drop-down Input Component: This has been added in order to enable users select any launch site data they want to view
    - A Pie chart: This has been added to enable the users compare the sites and their launch success rate.
    - A Scatter chart: A scatter chart has been added to show the correlation between Payload Mass and Success for all sites, and to compare the F9 Booster versions and their launch success rate.
    - Range slider: This has been added to allow the users select a specific range of the Payload Mass with the launch outcomes they want to be displayed on the scatter chart.
- GitHub URL of the completed Plotly Dash lab:
    - https://github.com/AnthonyRC7/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/427de0caa4232720cebf498bfa6afbc40052d9c2/spacex_launch_records_dashboard.py

Predictive Analysis (Classification)

LR, SVM, K-NN, and DT Model Development Process



- Building the models: The models have been built using feature selection and feature engineering.
- Evaluating the models: The models have been evaluated by splitting the data X and Y into training and test data using the train_test_split function
- Improving the models: The models have been improved by creating a GridSearchCV object that takes in a model as parameter in order to find the best parameters for the model.
- Selecting the best model: This has been achieved by comparing the accuracy of the models, their confusion matrix, and their score in order to find the best performing classification model.
- https://github.com/AnthonyRC7/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/035bd93edc6fa30f3875cc9b9dd4dc12c87543fc/Machine%20Learning%20Prediction.ipynb

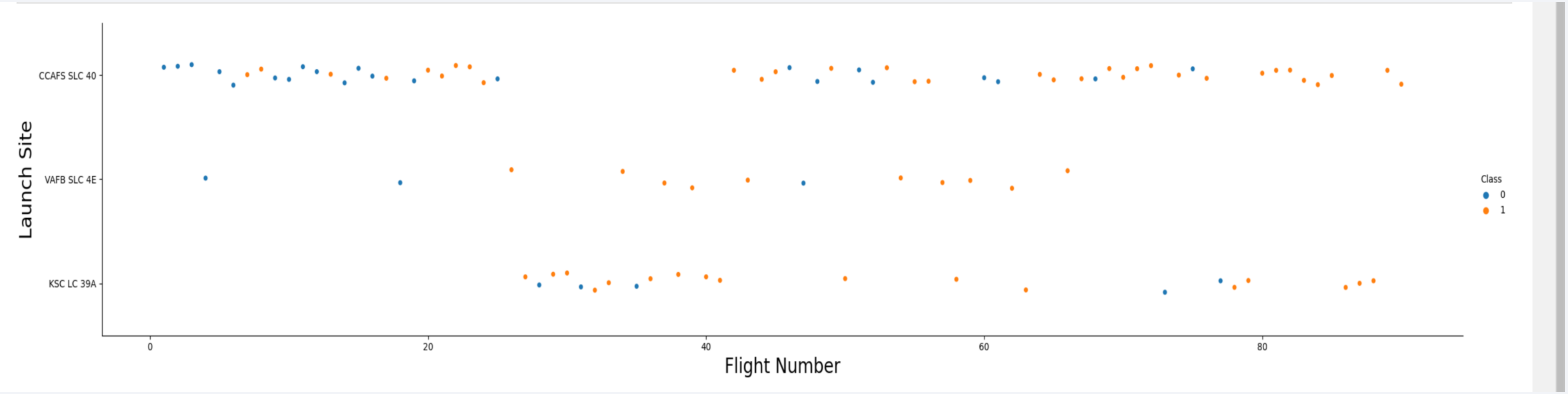
Results

- KSC LC – 39A had the highest launch success rate
- The launches with a low payload mass perform better than the launches with a high payload mass.
- The Orbits with the best success rate are ES-L1, GEO, HEO, and SSO.
- The success rate for SpaceX launches had a significant increase in 2013 and kept increasing till 2020.
- The F9 Booster Version: FT, had the highest launch success rate.
- The Decision Tree model, achieved the highest accuracy score at 94%, while the Logistic Regression, DT and KNN performed the best in terms of the confusion matrix. In terms of overall comparison, the Decision Tree model performed the best.

Section 2

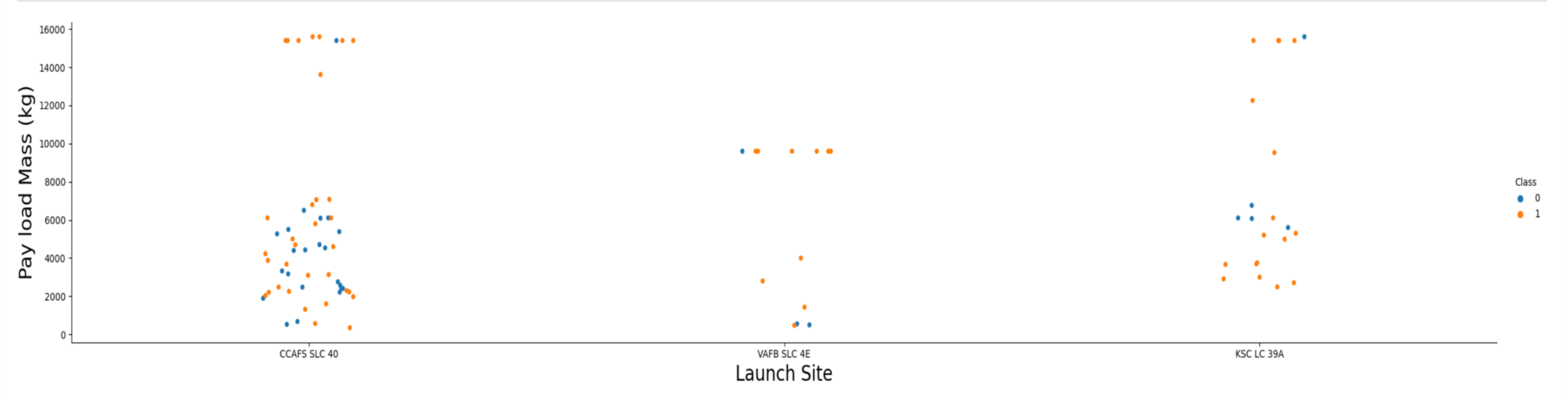
Insights drawn from EDA

Flight Number vs. Launch Site



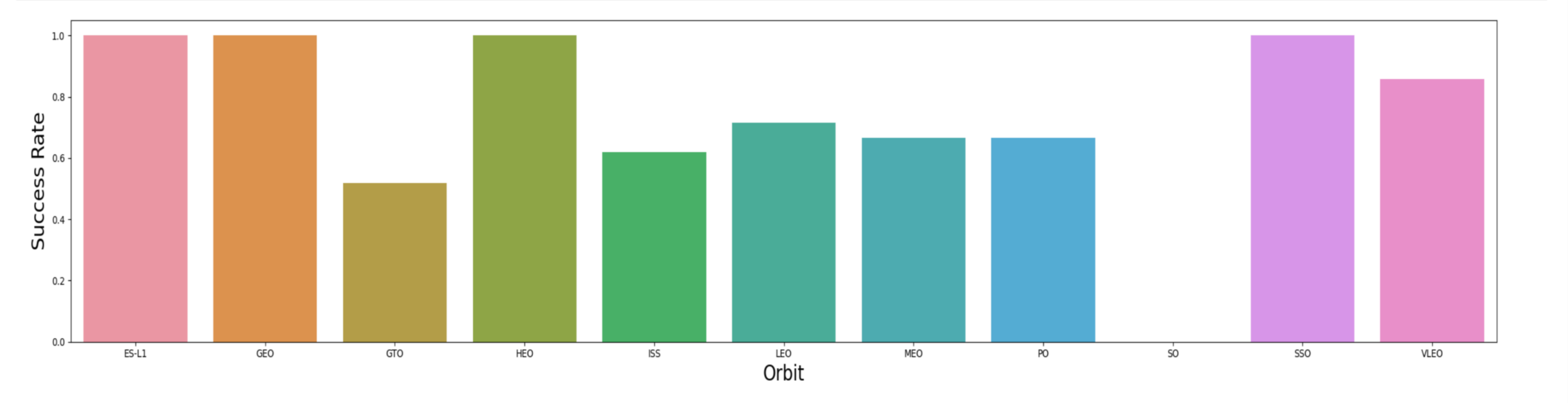
- Launches from the CCAFS SLC 40 site are significantly higher than launches from other launch sites.

Payload vs. Launch Site



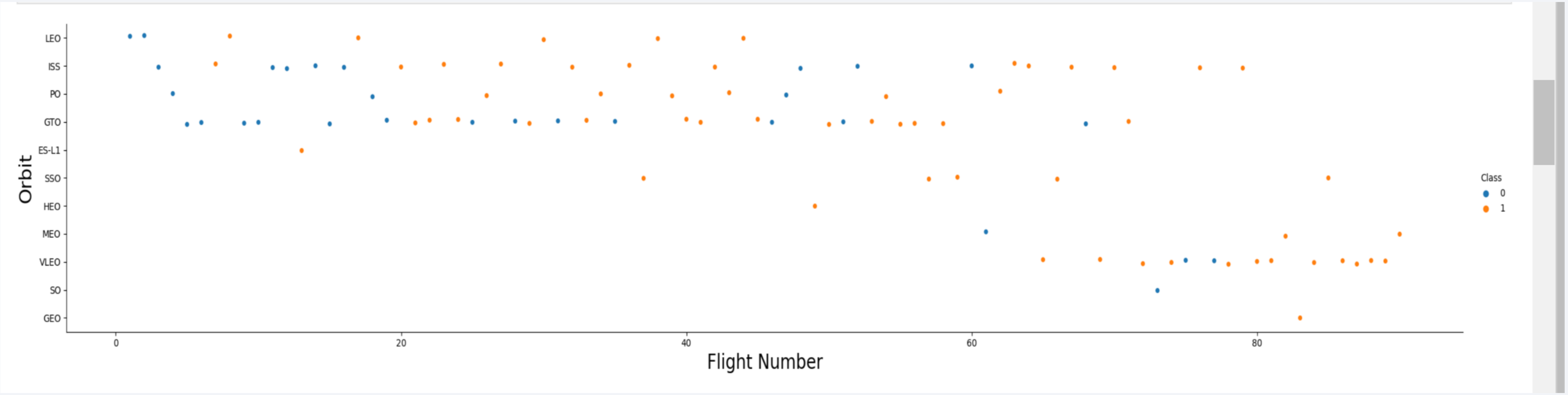
- The majority of Pay Loads with less mass were launched from the CCAFS SLC 40 site. Also, for the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000).

Success Rate vs. Orbit Type



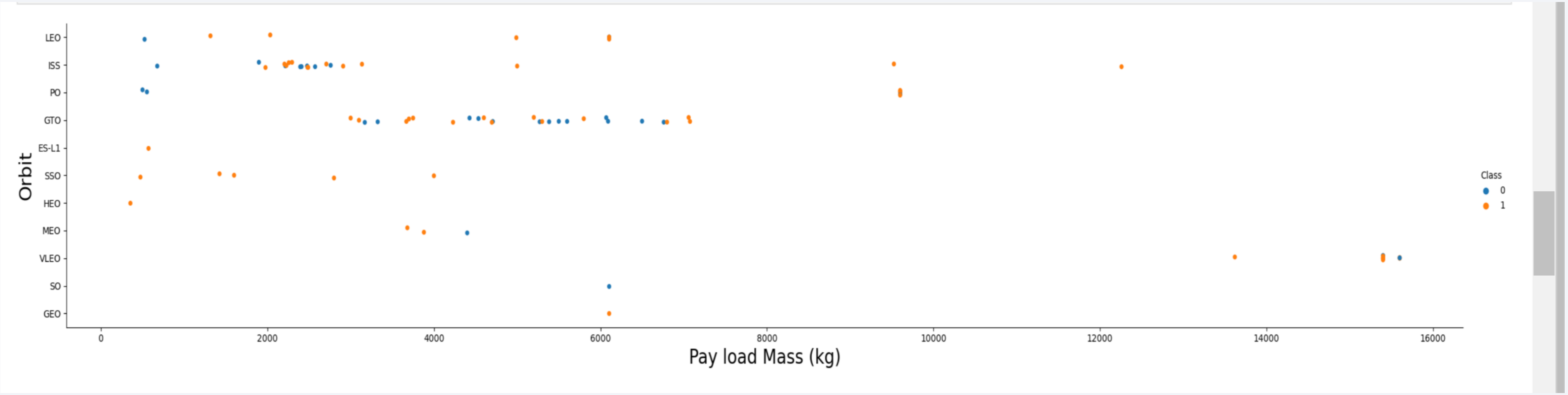
- The Orbits with the best success rate are ES-L1, GEO, HEO, and SSO.

Flight Number vs. Orbit Type



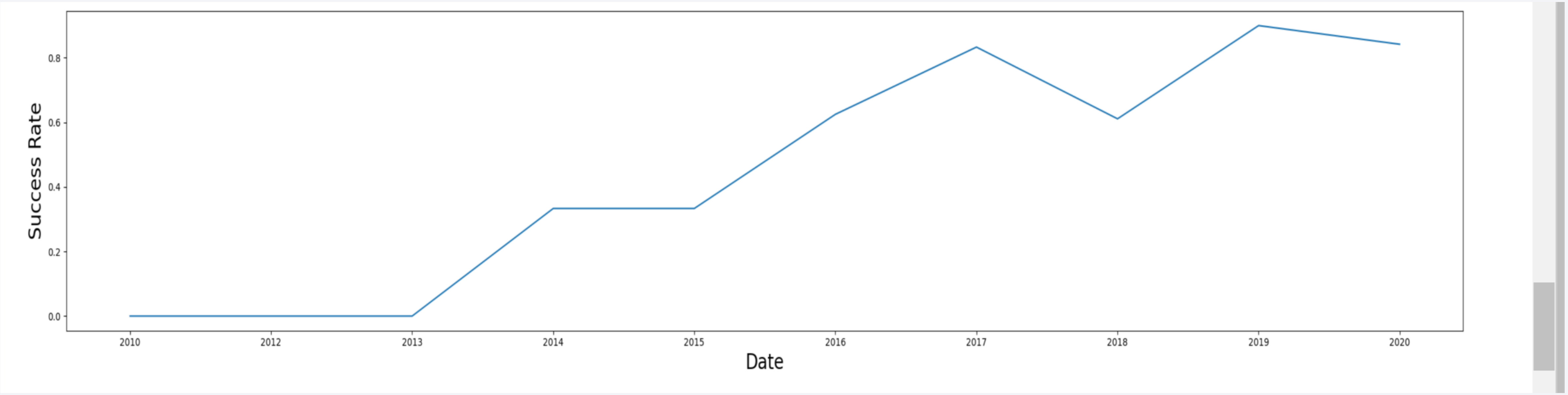
- In the LEO orbit, the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO Orbit. Also, it can be observed that as time went on, there were more launches shifted towards the VLEO Orbit.

Payload vs. Orbit Type



- It can be observed that with heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. However, for GTO it cannot be distinguished as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



- It can be observed that the launch success rate since 2013 kept increasing till 2020

All Launch Site Names

2 hours ago
3 hours ago
21 hours ago
20 hours ago

```
[9]: %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL;
 * sqlite:///my_data1.db
Done.
[9]: ,,,,,,,,,,,,,,,,,,,,,,,
```

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

- The SQL query displays the names of the unique launch sites in the space mission

Launch Site Names Begin with '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_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
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 (parachute)
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 attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

- The SQL query displays 5 records where launch sites begin with the string 'CCA'

Total Payload Mass

```
[6]: %sql SELECT SUM("PAYLOAD_MASS__KG_") AS "TOTAL PAYLOAD MASS" FROM SPACEXTBL WHERE "Customer" = "NASA (CRS)";

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

[6]:

TOTAL PAYLOAD MASS
45596.0

- The SQL query displays the total payload mass carried by boosters launched by NASA (CRS)

Average Payload Mass by F9 v1.1

```
[7]: %sql SELECT AVG("PAYLOAD_MASS__KG_") AS "AVERAGE PAYLOAD MASS" FROM SPACEXTBL \
     WHERE "Booster_Version" = "F9 v1.1";

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

[7]:

AVERAGE PAYLOAD MASS
2928.4

- The SQL query displays the average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

```
[16]: %sql SELECT "Date", "Landing_Outcome" FROM SPACEXTBL \
      WHERE "Landing_Outcome" = "Success (ground pad)" LIMIT 1;

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

[16]:

Date	Landing_Outcome
22/12/2015	Success (ground pad)

- The SQL query displays the date when the first successful landing outcome in ground pad was achieved.

Successful Drone Ship Landing with Payload between 4000 and 6000

[17]:
```
%sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_", "Landing_Outcome" FROM SPACEXTBL \
WHERE "Landing_Outcome" = "Success (drone ship)" \
AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;
```

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

[17]:

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1022	4696.0	Success (drone ship)
F9 FT B1026	4600.0	Success (drone ship)
F9 FT B1021.2	5300.0	Success (drone ship)
F9 FT B1031.2	5200.0	Success (drone ship)

- The SQL query displays the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

```
[18]: %sql SELECT COUNT("Mission_Outcome") AS "Total Number of Successful and Failure Mission Outcomes" \
FROM SPACEXTBL WHERE "Mission_Outcome" LIKE "%Success%" OR "Mission_Outcome" LIKE "Failure%";

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

[18]:

Total Number of Successful and Failure Mission Outcomes
101

- The SQL query displays the total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

```
[19]: %sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM SPACEXTBL \
      WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)

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

[19]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600.0
F9 B5 B1049.4	15600.0
F9 B5 B1051.3	15600.0
F9 B5 B1056.4	15600.0
F9 B5 B1048.5	15600.0
F9 B5 B1051.4	15600.0
F9 B5 B1049.5	15600.0
F9 B5 B1060.2	15600.0
F9 B5 B1058.3	15600.0
F9 B5 B1051.6	15600.0
F9 B5 B1060.3	15600.0
F9 B5 B1049.7	15600.0

- The SQL query displays the names of the booster versions which have carried the maximum payload mass.

2015 Launch Records

```
[36]: %sql SELECT strftime('%m', Date) AS "Month Name", "Booster_Version", "Launch_Site", "Landing_Outcome" \
FROM SPACEXTBL WHERE strftime('%Y', Date) = "2015" AND "Landing_Outcome" = "Failure (drone ship)";

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

[36]:

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)

- The SQL query displays the month names, failure landing outcomes in drone ship, booster versions, launch site for the months in year 2015.

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

11 days ago
minutes ago
14 days ago

[32]:
```
%sql SELECT * FROM SPACEXTBL WHERE "Landing_Outcome" LIKE "%Success%" AND \
(DATE BETWEEN "2010-06-04" AND "2017-03-20") ORDER BY DATE DESC;
# WHERE strftime('%d/%m/%Y', substr(Date, 1, 10)) BETWEEN '04/06/2010' AND '20/03/2017';
```

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

[32]:

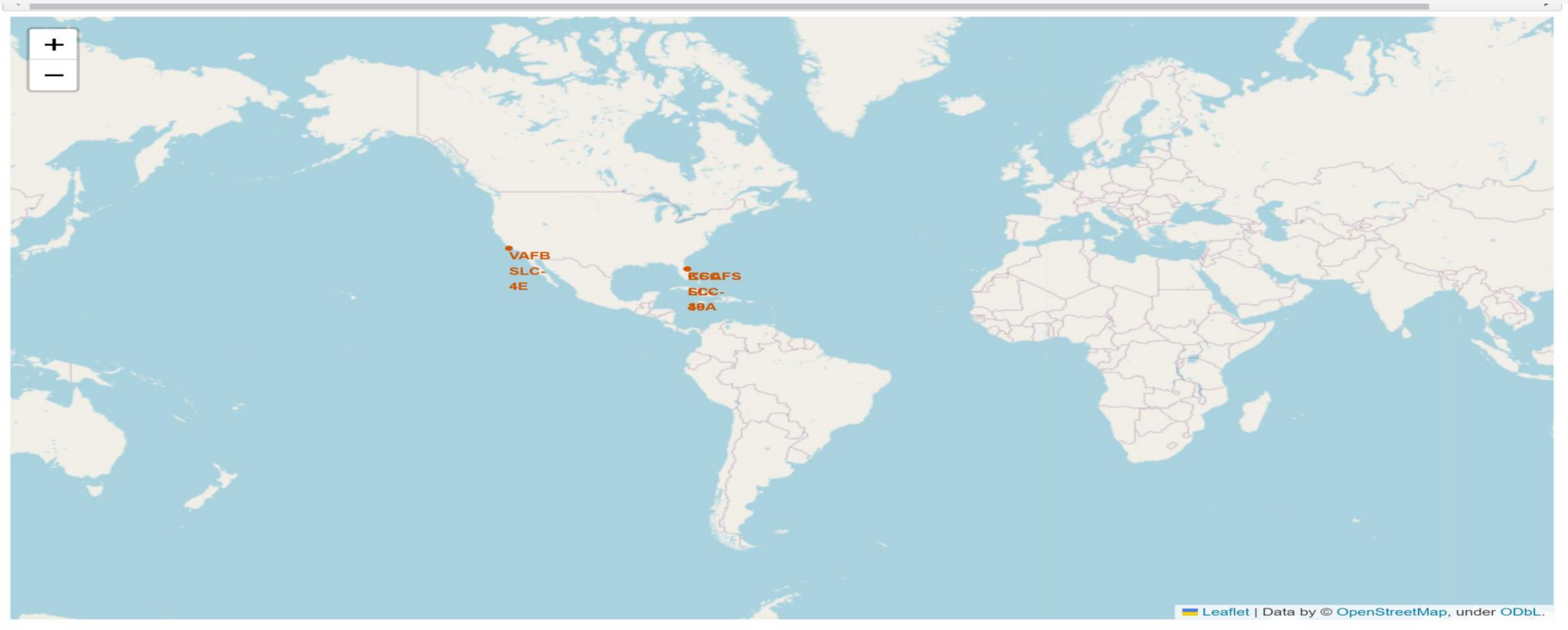
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-03-06	21:07:00	F9 FT B1035.1	KSC LC-39A	SpaceX CRS-11	2708.0	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490.0	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600.0	Polar LEO	Iridium Communications	Success	Success (drone ship)
2017-01-05	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300.0	LEO	NRO	Success	Success (ground pad)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600.0	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-04	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136.0	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257.0	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-06-05	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696.0	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100.0	GTO	Thaicom	Success	Success (drone ship)
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034.0	LEO	Orbcomm	Success	Success (ground pad)

- The SQL query displays the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

Section 3

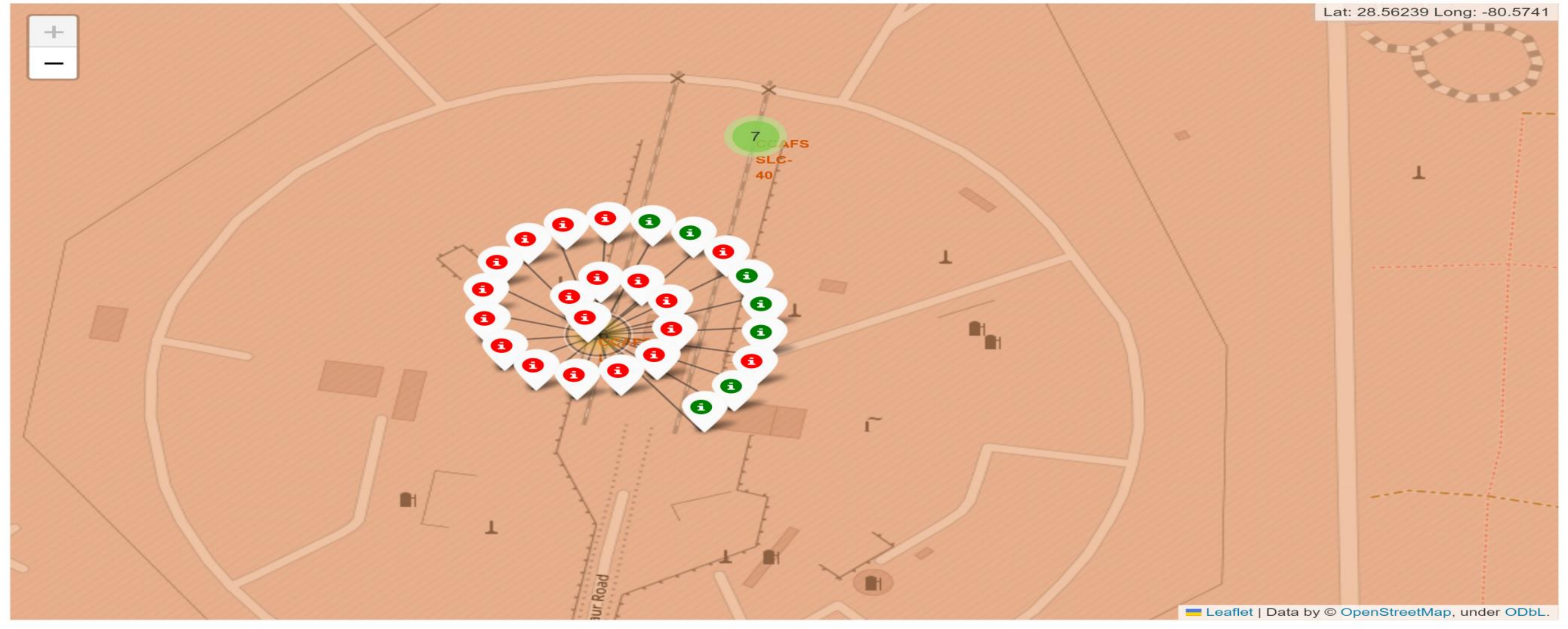
Launch Sites Proximities Analysis

All launch sites marked on a map



- Looking at the map, it can be observed that all launch sites are not in proximity to the Equator line. However, they are in very close proximity to the coast.

Success/failed launches marked on the map



- Looking at the map, from the color-labeled markers in marker clusters, it can be easily identified which launch sites have relatively high success rates. The Green markers represent successful launches, and the red markers represent failed launches.

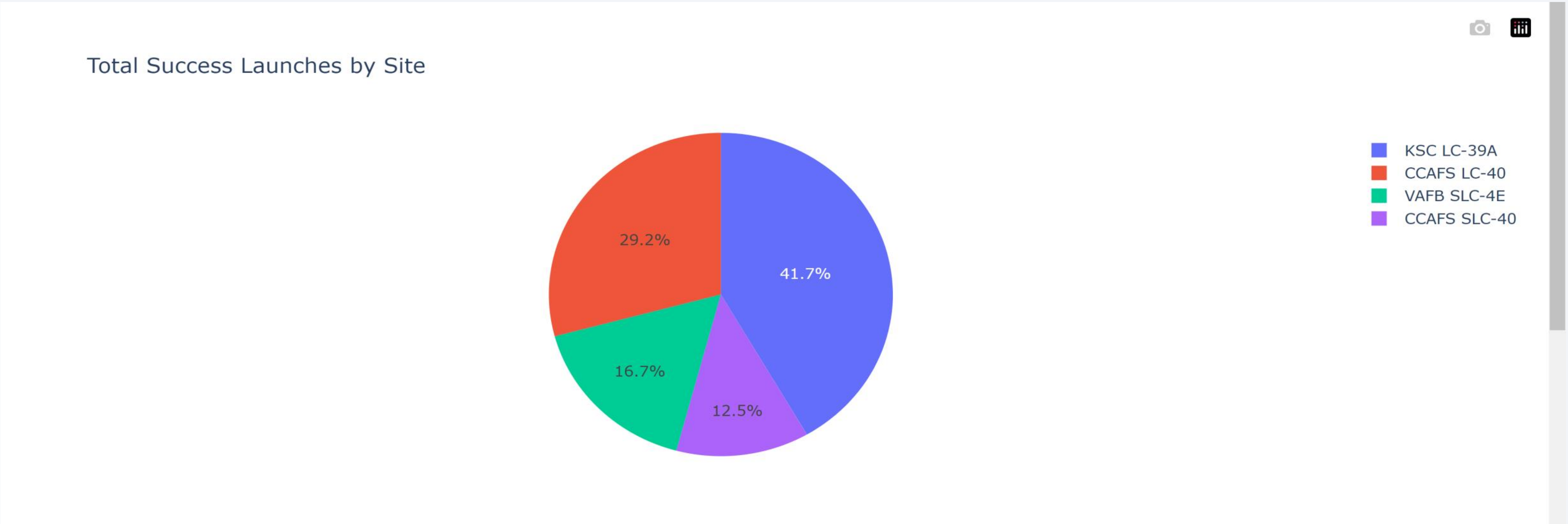
Distance between a launch site to its proximities

- Looking at the map, the distance between a launch site to one of its proximities (coastline) can be observed with the distance displayed and a continuous line that connects the launch site to its proximity (coastline).

Section 4

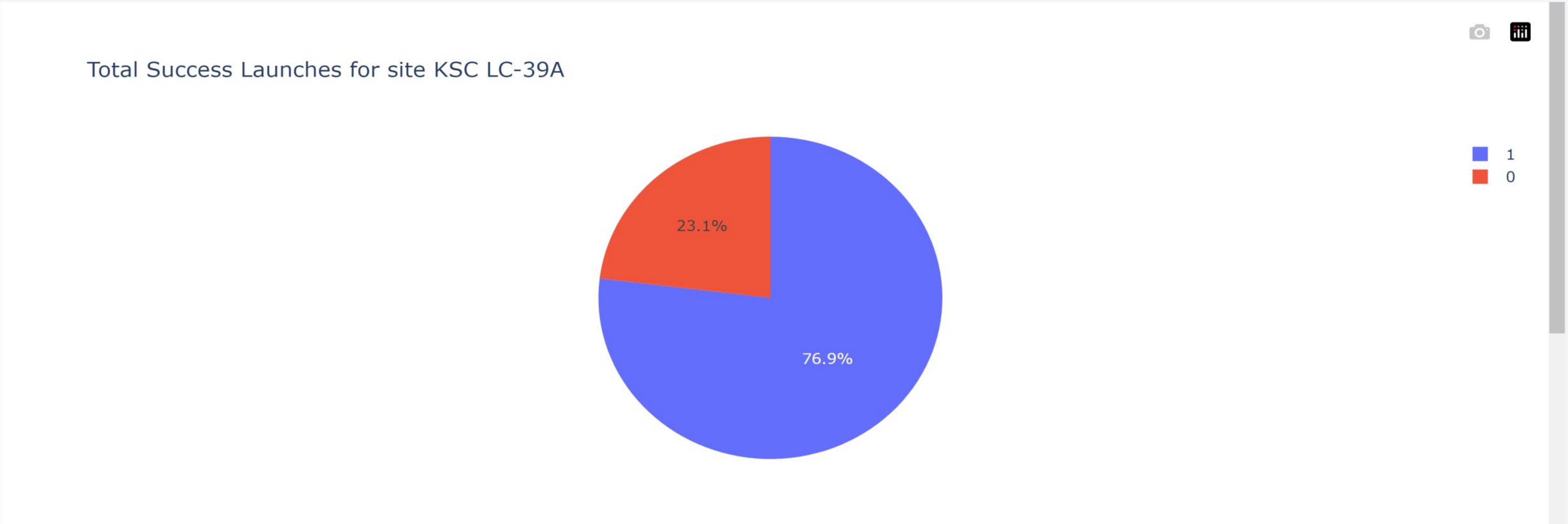
Build a Dashboard with Plotly Dash

Total success launches for all sites



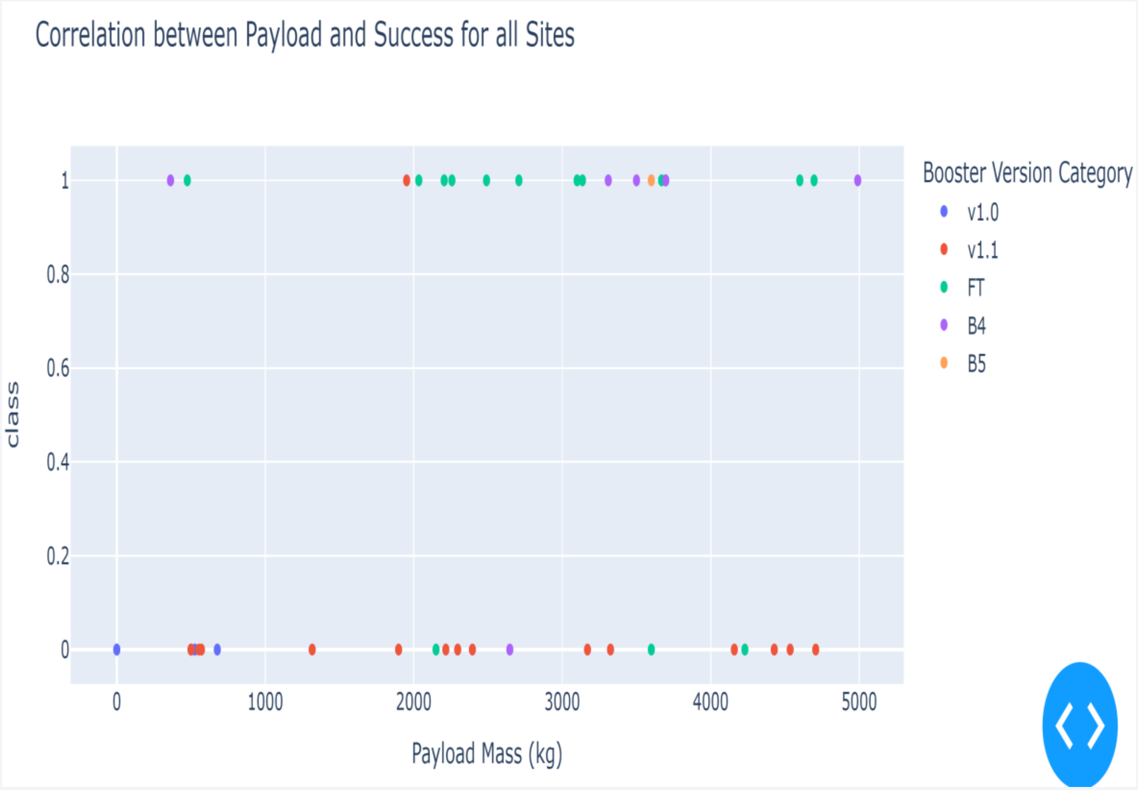
- This pie chart helps us compare the launch success rate for all sites. It can be observed that KSC LC-39A has the best launch success rate.

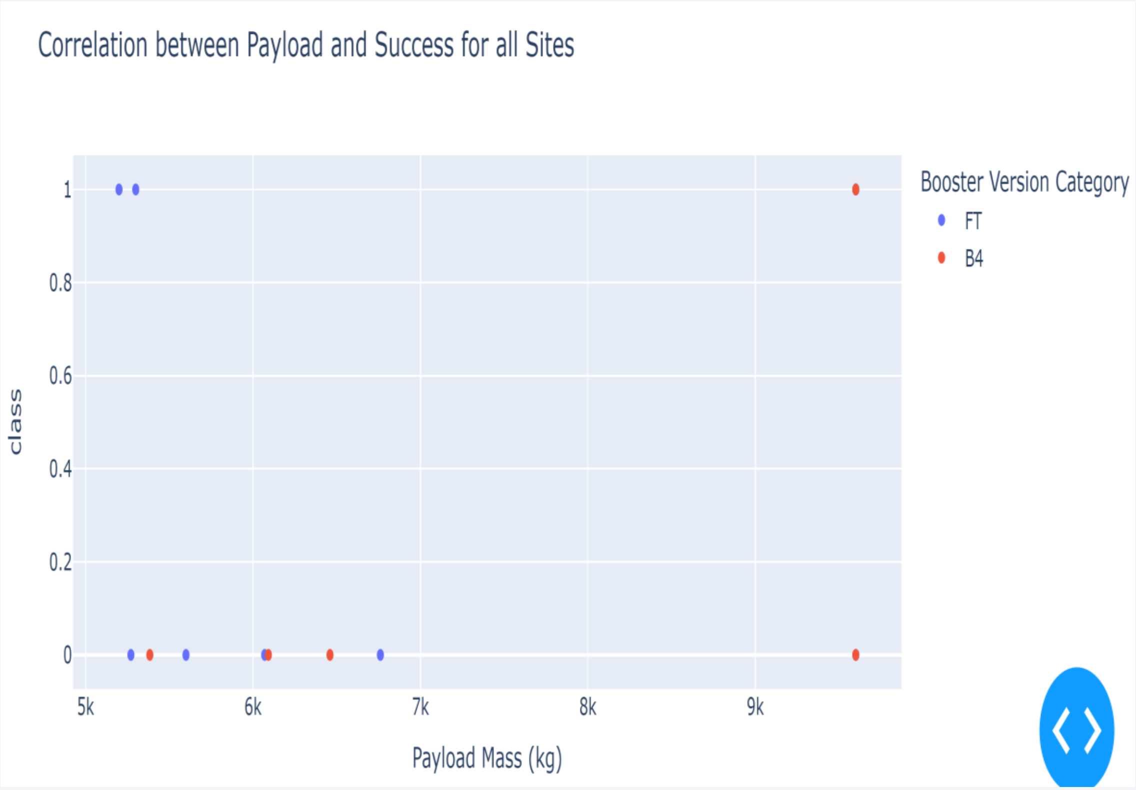
Success rate by site



- This pie chart helps us compare the rate of successful and failed launches for KSC LC-39A launch site. It can be observed that KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate.

Payload vs launch outcome



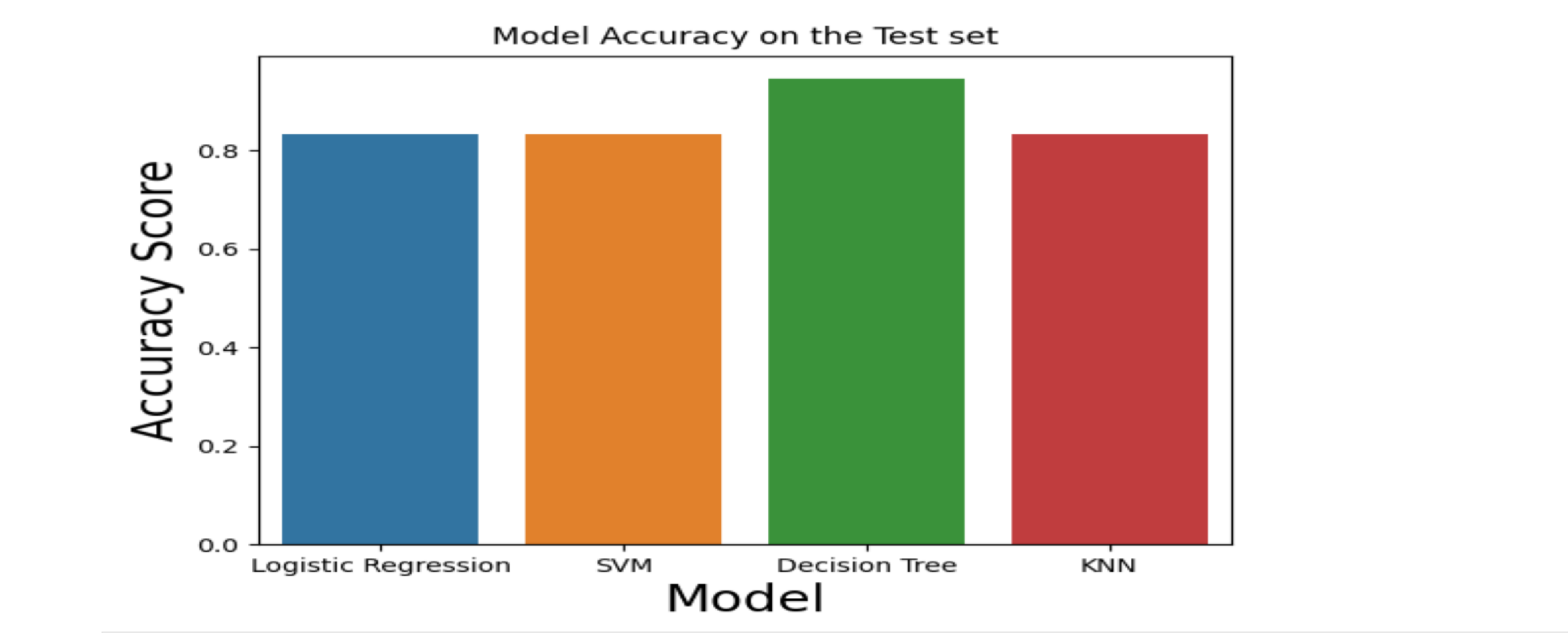


- It can be observed that the success rates for low weighted payloads is higher than the heavy weighted payloads, and the booster version with the highest success rate is FT.

Section 5

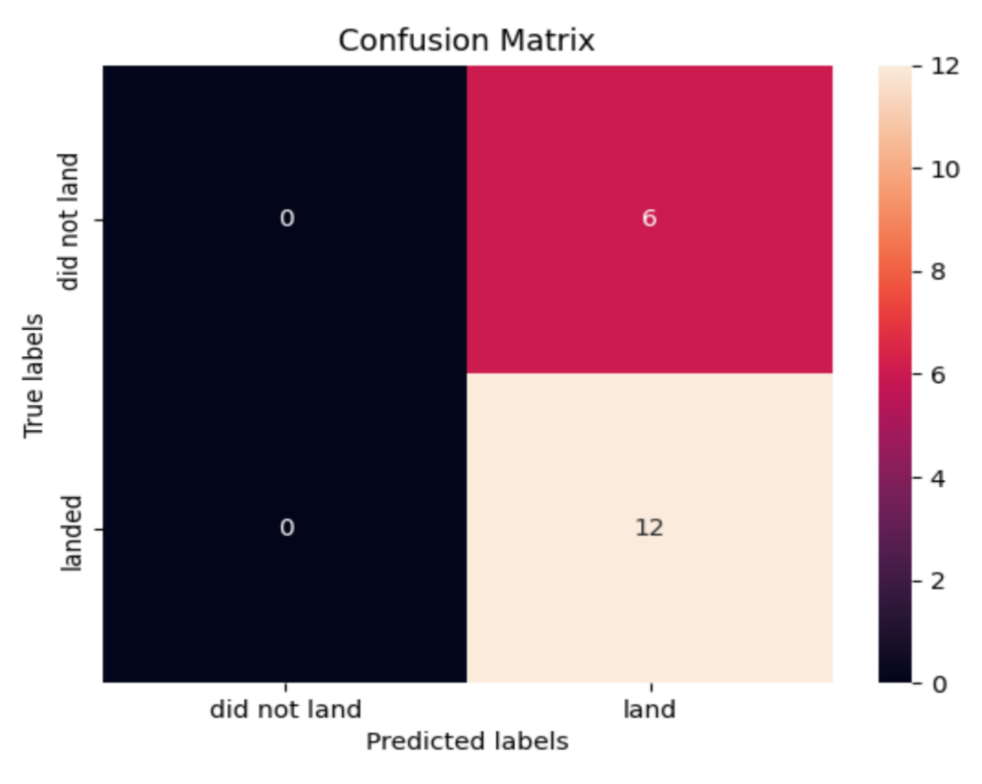
Predictive Analysis (Classification)

Classification Accuracy



- It can be observed that the Decision Tree model has the highest classification accuracy.

Confusion Matrix



The following values were obtained in the confusion matrix for DT:

1. True Positives (TP): 12 This indicates that the Decision Tree model correctly predicted 12 successful landings of the Falcon 9 first stage.

2. True Negatives (TN): 0 It means there were no instances where the model predicted a failed landing, and it was correct.

3. False Positives (FP): 6 This value represents the cases where the model predicted a successful landing, but it was unsuccessful.

4. False Negatives (FN): 0 There were no instances where the model predicted a failed landing, but the landing was successful.

Why the DT model is considered the best based on this confusion matrix:

- High True Positives (TP): The model achieved a high number of true positive predictions, correctly identifying 12 successful landings. This indicates that the model is effective at recognizing successful landings.

- Low False Negatives (FN): The model did not have any false negatives, meaning it did not miss any successful landings. This is a positive aspect as it ensures that the model is not incorrectly labeling successful landings as failed.

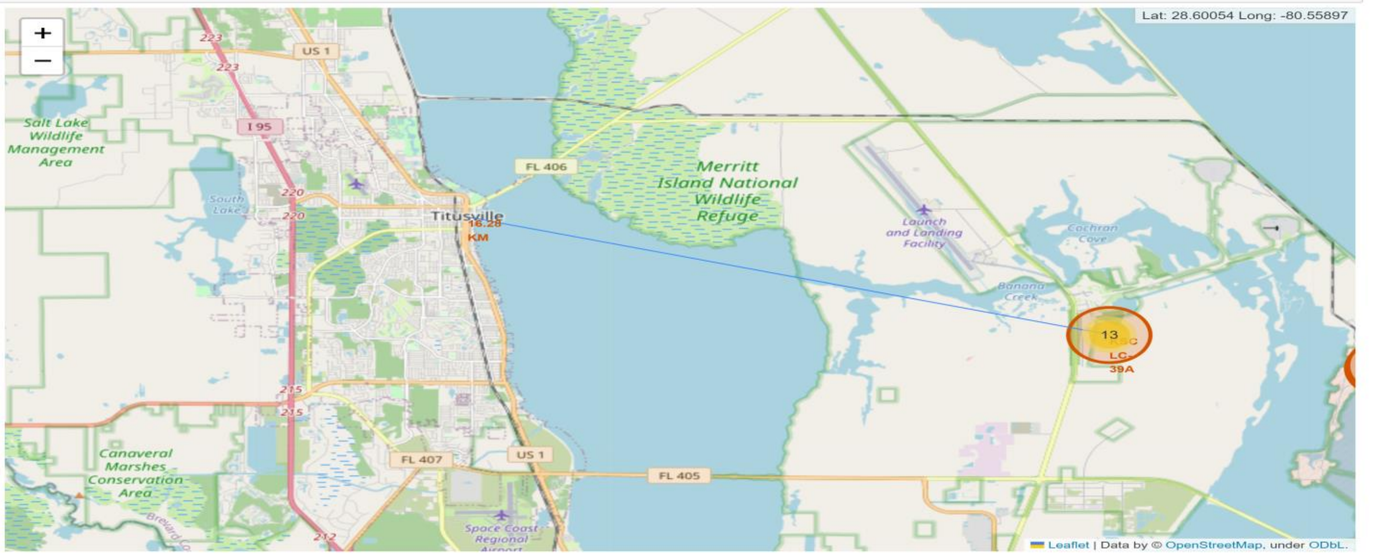
Conclusions

- The launches with a low payload mass perform better than the launches with a high payload mass.
- KSC LC – 39A launch site had the highest launch success rate.
- The Orbits with the best success rate are ES-L1, GEO, HEO, and SSO.
- Over the passing years, SpaceX has witnessed a notable progression in the success rates of its missions. As time has advanced, the company's endeavors have yielded increasingly favorable outcomes. This trend signifies a commendable improvement in the efficiency and effectiveness of SpaceX's operations.
- The F9 Booster Version: FT, had the highest launch success rate.
- The Decision Tree model, achieved the highest accuracy score at 94%, while the Logistic Regression, DT and KNN performed the best in terms of the confusion matrix. In terms of overall comparison, the Decision Tree model performed the best.

Appendix

- The python code to convert the date format from “dd/mm/yyyy” to “yyyy-mm-dd” in the Exploratory Data Analysis (EDA) with SQL lab:

```
import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv")
df["Date"] = pd.to_datetime(df["Date"], format="%d/%m/%Y")
df["Date"] = df["Date"].dt.strftime("%Y-%m-%d")
```



This map shows the distance between KSC LC-39A launch site to its closest city (Titusville). The distance is displayed with a continuous line that connects the launch site to the city.

Thank you!