IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

SIVARAMAN VISHAL RAMANATHAN

19 DECEMBER 2023

Outline

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

Executive Summary

➢ Summary of methodologies :

- ✓ Data Collection through API.
- ✓ Data Collection with Web Scraping.
- ✓ Data Wrangling.
- ✓ Exploratory Data Analysis with SQL.
- ✓ Exploratory Data Analysis with Data Visualization.
- ✓ Interactive Visual Analytics with Folium.
- ✓ Machine Learning Prediction.

➢ Summary of all results :

- ✓ Exploratory Data Analysis result.
- ✓ Interactive analytics screenshots.
- ✓ Predictive Analytics result.

Introduction

➢ Project background and context :

- ✓ SpaceX spent around 62 million on Falcon 9 rocket launches.
- ✓ The fact that SpaceX can land and reuse the rocket's first stage accounts for a considerable amount of the savings as compared to other providers, which normally charge more than 165 million.
- ✓ If we are able to predict whether the first stage will land, we can use this information to estimate the cost of a launch and determine whether another company should compete with SpaceX for a rocket launch.
- ✓ Ultimately, this research will be able to predict whether or not the Space X Falcon 9 first stage will land successfully.

➢ Problems you want to find answers :

- ✓ What elements affect the rocket's likelihood of a successful landing
- ✓ The way in which different elements interact to determine the likelihood of a successful landing
- ✓ What operational conditions must be met in order to guarantee the success of the landing programme

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - **Utilising the spaceX API and web scraping from Wikipedia, data was gathered.**
- Perform data wrangling
    - **We used one-hot encoding on the categorical features.**
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - **Building the optimal model for Machine Learning.**

Data Collection

➢ **Describe how data sets were collected :**

✓ Utilising a receive call to the SpaceX API, data was gathered.

✓ Next, we used the.json() function call to decode the response content as JSON and the.json_normalize() method to convert it into a pandas dataframe.

✓ After that, we cleaned the data, looked for any missing values, and, if needed, filled them in.

✓ Additionally, we used BeautifulSoup to perform web scraping of Wikipedia to find launch records for the Falcon 9.

✓ The aim was to retrieve the launch records in the form of an HTML table, parse the information, and then transform it into a pandas dataframe for upcoming examination.

Data Collection – SpaceX API

- ✓ The SpaceX API's get request was utilised to gather data, clean the requested data, and do some simple data wrangling and formatting.

- ✓ GitHub URL : https://github.com/Sivaraman-VishalRamanathan/FINAL_CAPSTON_PROJECT_IBM_DATA_SCIENCE/blob/main/Vishal_Ramanathan_20231218_jupyter-labs-spacex-data-collection-api.ipynb

1. Get request for rocket launch data using API

In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [7]:
```
response = requests.get(spacex_url)
```

2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```

Data Collection - Scraping

- ✓ Using BeautifulSoup, we used web scraping to gather Falcon 9 launch records. The table was parsed, and a pandas dataframe was created.

- ✓ GitHub URL : https://github.com/Sivaraman-VishalRamanathan/FINAL_CAPSTON_PROJECT_IBM_DATA_SCIENCE/blob/main/Vishal_Ramanathan_20231218_jupyter-labs-webscraping.ipynb

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

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

Out[5]:
```
200
```

2. Create a BeautifulSoup object from the HTML response

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

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

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

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Extract all column names from the HTML table header

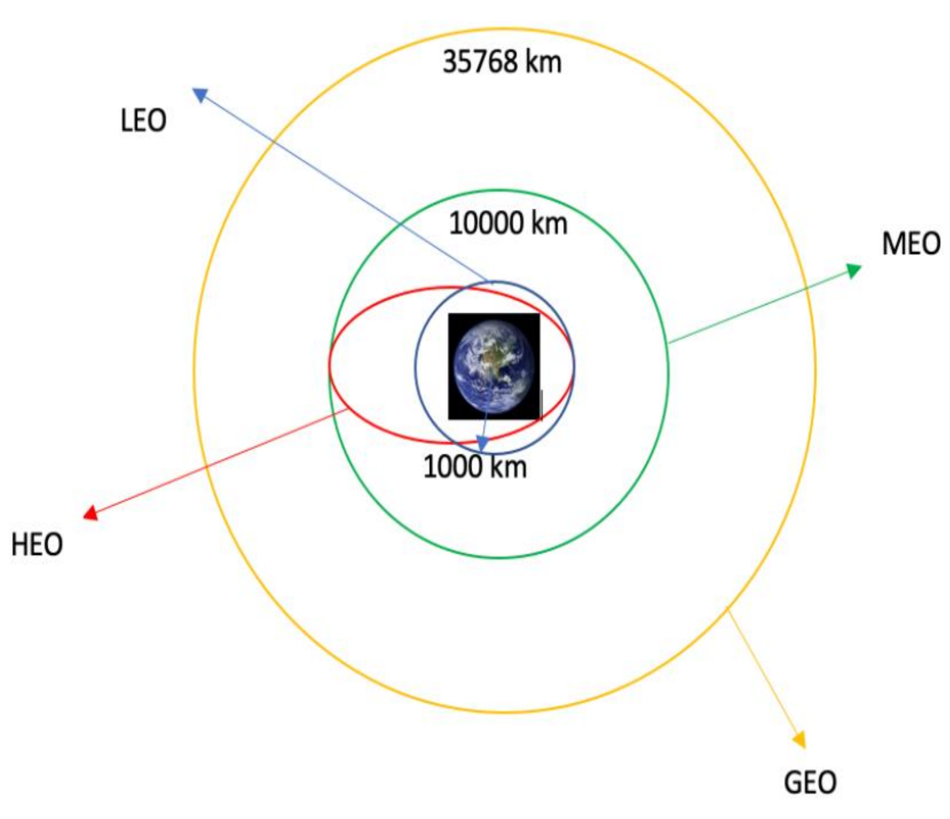
In [10]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

Data Wrangling

- ✓ We identified the training labels by doing an exploratory data analysis.
- ✓ We determined the quantity of launches at every location as well as the quantity and frequency of each orbit.
- ✓ From the outcome column, we generated the landing outcome label and exported the data to CSV.

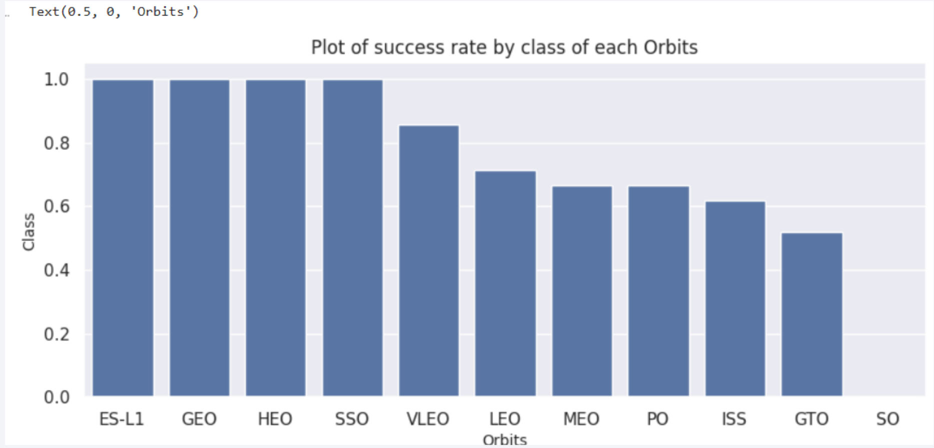
- ✓ GitHub URL : https://github.com/Sivaraman-VishalRamanathan/FINAL_CAPSTON_PROJECT_IBM_DATA_SCIENCE/blob/main/Vishal_Ramanathan_20231218_labs-jupyter-spacex-Data%20wrangling.ipynb

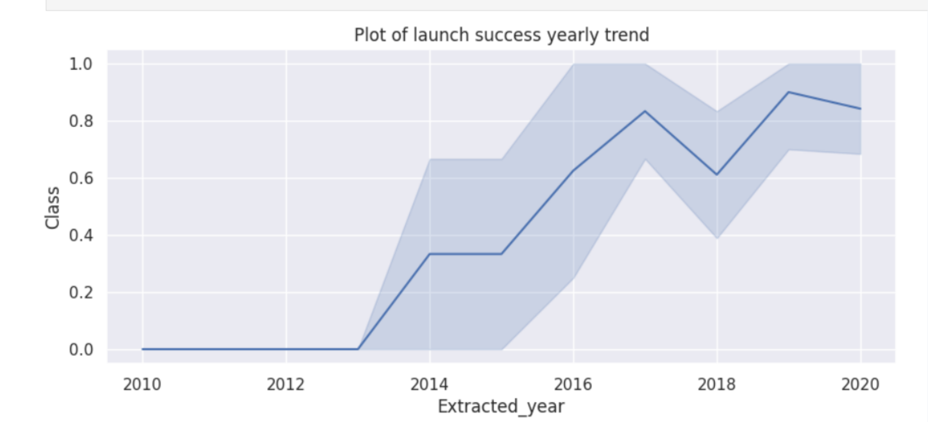


EDA with Data Visualization

- ✓ In order to better understand the data, we plotted the relationships between the flight number vs the launch site, the payload vs the launch site, the success rate of each orbit type, the flight number vs the orbit type, and the annual trend of launch success.

- ✓ GitHub URL : https://github.com/Sivaraman-VishalRamanathan/FINAL_CAPSTON_PROJECT_IBM_DATA_SCIENCE/blob/main/Vishal_Ramanathan_20231219jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb





EDA with SQL

- ✓ The SpaceX dataset was loaded into a PostgreSQL database without requiring us to exit the Jupyter notebook.
- ➢ To extract meaning from the data, we used SQL and EDA. We created queries to, for example, find out :
- ✓ The space mission's distinct launch sites' names.
- ✓ The total mass of payload that NASA's CRS boosters have carried.
- ✓ The rocket version F9 v1.1's average payload mass.
- ✓ The total number of outcomes from missions that were successful or unsuccessful.
- ✓ Drone ship's unsuccessful landing results, along with the names of the launch sites and booster versions.
- ✓ GitHub URL : https://github.com/Sivaraman-VishalRamanathan/FINAL_CAPSTON_PROJECT_IBM_DATA_SCIENCE/blob/main/Vishal_Ramanathan_20231219_jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- ✓ After initialising the map and adding a folium circle and marker for each launch site that has been provided, mark all of the launch sites on it using the NASA coordinates.
- ✓ Determine and label triumphs and setbacks, create clusters, create an icon as a text title, give an icon colour, and launch sites by providing dummy variables, such as 1 for success and 0 for failure.
- ✓ The process of measuring the distance between a launch site and its surroundings involves locating launch sites, figuring out the distance between latitude and longitude, making a folium marker to indicate the distance, and adding these characteristics to the map.
- ✓ The Space X launch sites were marked with folium markers, along with the locations of nearby significant landmarks such as cities, highways, trains, and beaches.
- ✓ The launch sites were connected to the closest land points using polylines.
- ✓ Failures to launch rockets are shown by red.
- ✓ Green stands for success.
- ✓ GitHub URL : https://github.com/Sivaraman-VishalRamanathan/FINAL_CAPSTON_PROJECT_IBM_DATA_SCIENCE/blob/main/Vishal_Ramanathan_20231220_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- ✓ A Plotly Dash dashboard was enhanced with the following plots to provide an interactive data visualisation.
- ✓ Pie chart (px.pie()) displaying each site's total number of successful launches.
- ✓ This makes it easy to identify the most successful websites.
- ✓ A dcc.Dropdown() object might be used to filter the chart and view the success/failure ratio for a certain site.
- ✓ The scatter graph (px.scatter()) illustrates the relationship between the payload mass (kg) and the outcome (success or failure).
- ✓ This could have ranges of payload masses blocked off (using a RangeSlider() object).
- ✓ Moreover, booster version could be used to filter it.
- ✓ GitHub URL : https://github.com/Sivaraman-VishalRamanathan/FINAL_CAPSTON_PROJECT_IBM_DATA_SCIENCE/blob/main/Vishal_Ramanathan_Dash_app.py

Predictive Analysis (Classification)

- ✓ To create, assess, and determine which categorization model performed the best, the following procedures were followed:
- ➢ **Model Creation :**
- ✓ Download the dataset.
- ✓ Carry out the appropriate preprocessing and standardisation of data.
- ✓ Use train_test_split to divide data into training and test data sets.
- ✓ Choose the most suitable kind of machine learning algorithms.
- ✓ Generate a dictionary of parameters and a GridSearchCV object.
- ✓ Align the item with the specifications.
- ✓ Train the model using the training data set.

Predictive Analysis (Classification) contd.

➢ Assessment of the Model :

✓ Verify the adjusted hyperparameters (best_params_) and accuracy (score and best_score_) using the GridSearchCV object that is produced.

✓ Create a plot and study the confusion matrix.

➢ Choosing the optimal model for FIT classification :

✓ Examine the accuracy ratings of each selected algorithm.

✓ The model that performs the best is the one with the highest accuracy score.

✓ GitHub URL : https://github.com/Sivaraman-VishalRamanathan/FINAL_CAPSTON_PROJECT_IBM_DATA_SCIENCE/blob/main/Vishal_Ramanathan_20231220_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

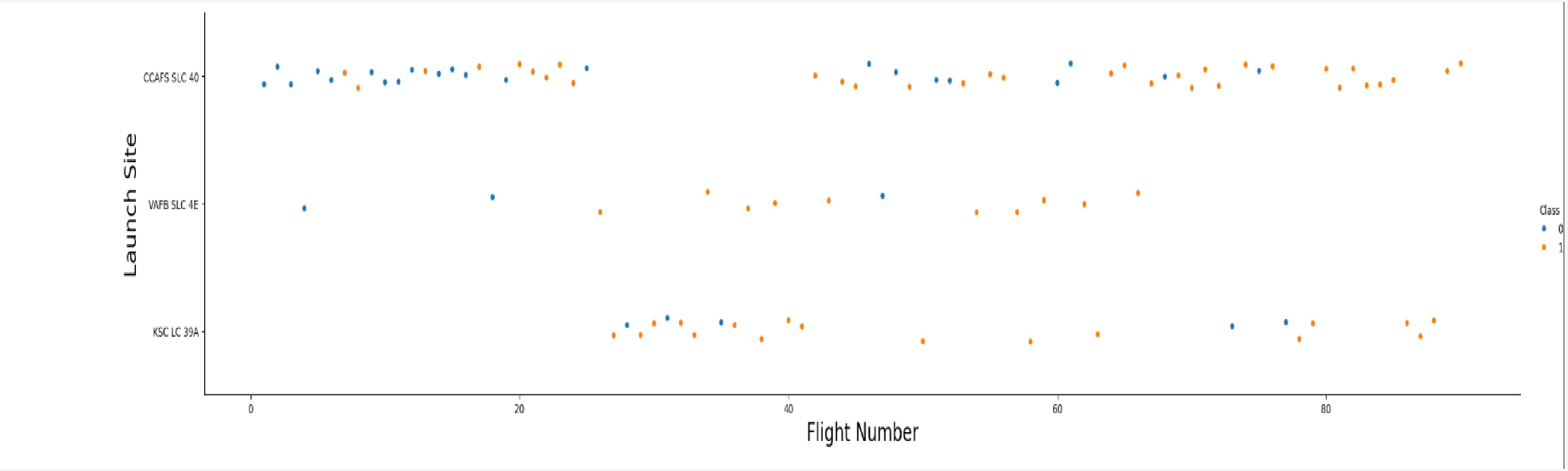
Results

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

Section 2

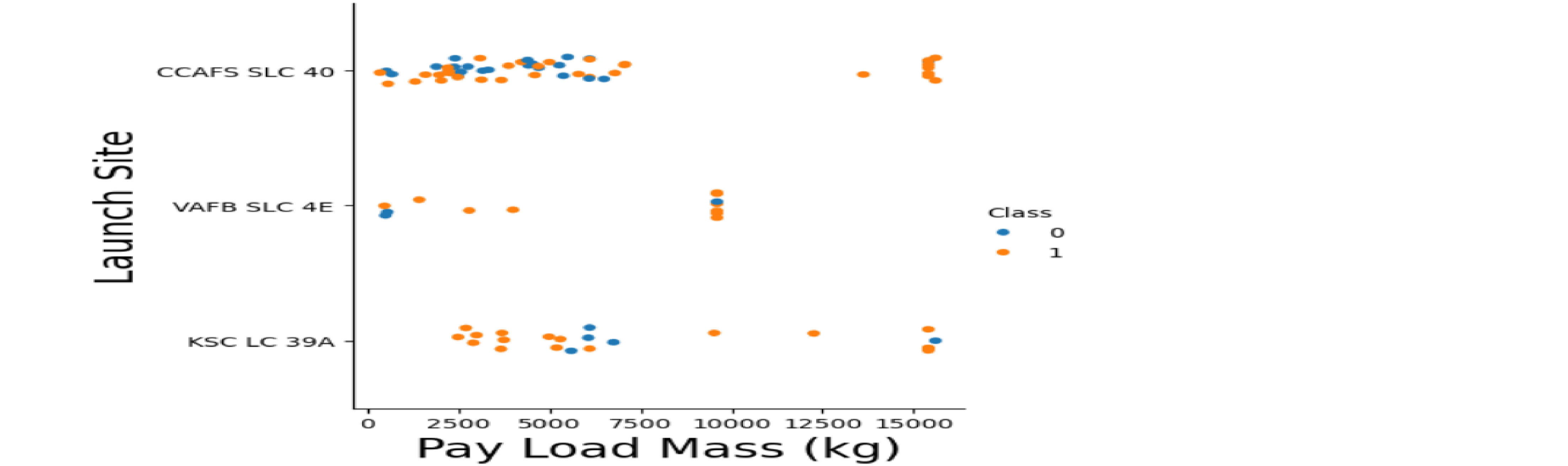
Insights drawn from EDA

Flight Number vs. Launch Site



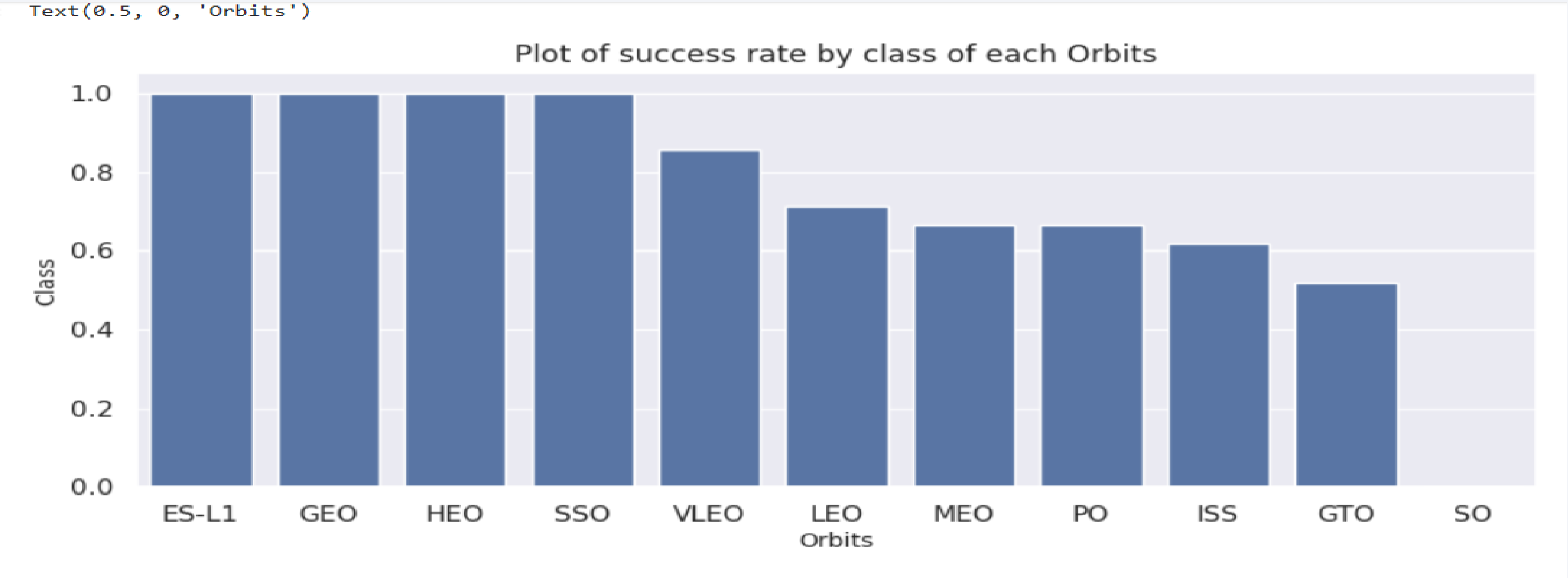
- ✓ The number of flights is rising, and launch sites are seeing an increase in success rates as well.
- ✓ As the number of flights rose, it seems that more landings were successful.
- ✓ The most landings were at launch location CCAFS SLC 40.

Payload vs. Launch Site



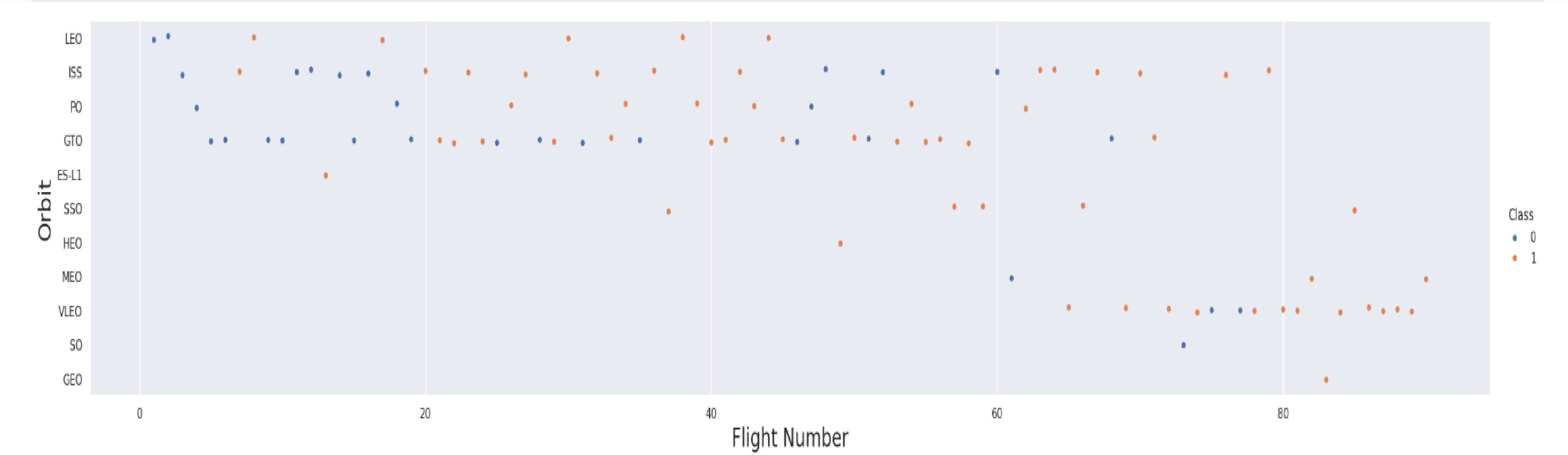
- ✓ As the payload mass increases, launch sites are seeing a rise in success rates as well.
- ✓ As you can see from the scatter point chart, no rockets with a significant payload mass (more than 10,000) have been launched from the VAFB-SLC launch site.

Success Rate vs. Orbit Type



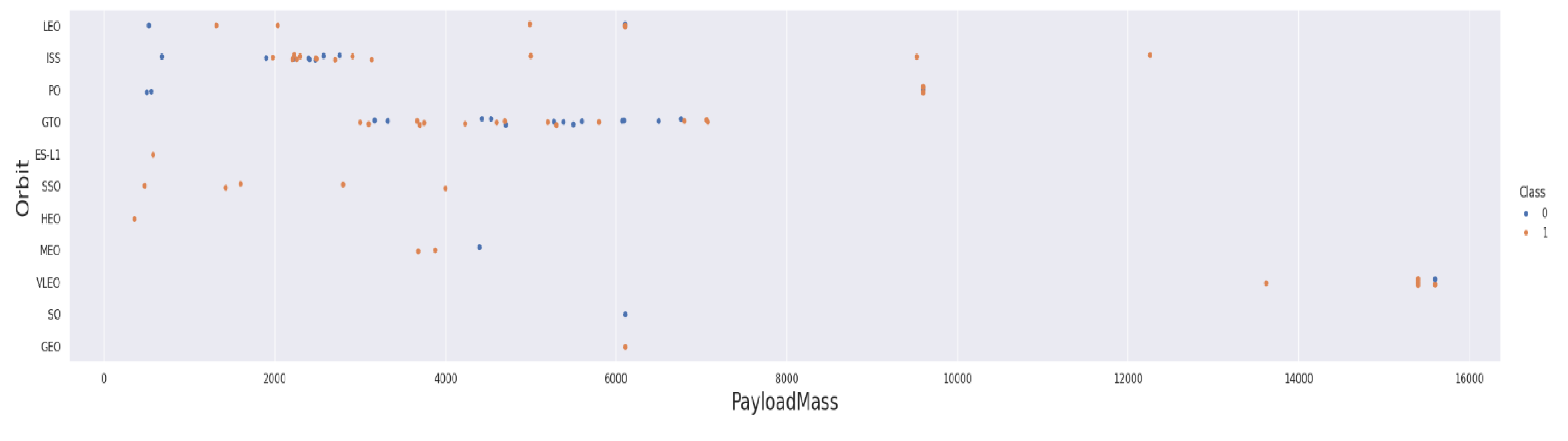
- ✓ The plot indicates that the highest success rates were attained by ES-L1, GEO, HEO, SSO, and VLEO.
- ✓ SO has a success rate of 0.

Flight Number vs. Orbit Type



✓ We see that while there is no correlation between flight number and orbit in the GTO orbit, success in the LEO orbit is correlated with the number of flights.

Payload vs. Orbit Type



- ✓ We can see that more successful landings for PO, LEO, and ISS orbits occur when heavy payloads are carried.
- ✓ The first thing to observe is the impact on ISS of the Pay load Mass between 2000 and 3000.
- ✓ For GTO, on the other hand, it is difficult to discern between positive landing rate and negative landing rate (unsuccessful mission).
- ✓ GTO is also impacted by pay load mass between 3000 and 7000.

Launch Success Yearly Trend

- ✓ It is evident that between 2013 and 2020, the success rate grew dramatically.
- ✓ The success rate has significantly increased since 2013. It did, however, somewhat decline in 2018 before gaining strength again.

All Launch Site Names

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

```
[17]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

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

[17]:

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

✓ "DISTINCT" key word allows us to obtain the unique values.

Launch Site Names Begin with ‘CCA’

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

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

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

[19]:

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

✓ Using "LIMIT," we are only able to obtain 5 rows.

Total Payload Mass

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

```
[20]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS FROM SPACEXTBL WHERE CUSTOMER LIKE 'NASA (CRS)'
```

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

[20]:

TOTAL_PAYLOAD_MASS
45596

✓ Using "SUM," we can obtain the total of all values.

Average Payload Mass by F9 v1.1

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

```
[22]: %sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVERAGE_PAYLOAD_MASS FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE 'F9 v1.1'

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

[22]:

AVERAGE_PAYLOAD_MASS
2928.4

✓ The "AVG" function allows us to obtain the average of all values.

First Successful Ground Landing Date

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

Hint:Use min function

```
[26]: %sql SELECT MIN(DATE) AS FIRST_SUCCESSFUL_LANDING_OUTCOME FROM SPACEXTBL WHERE LANDING_OUTCOME LIKE 'Success (ground pad)';
```

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

[26]:

FIRST_SUCCESSFUL_LANDING_OUTCOME
2015-12-22

✓ As we can see, on December 22, 2015, the first ground pad was successfully launched.

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
[27]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_M
```

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

[27]:

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

✓ The landing result was identified as a "success drone" after the payload mass data was only collected between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

[32]:
```
%sql SELECT MISSION_OUTCOME, COUNT(*) AS SUCCESS_OUTCOME FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%'
```

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

[32]:

Mission_Outcome	SUCCESS_OUTCOME
Success	100

[33]:
```
%sql SELECT MISSION_OUTCOME, COUNT(*) AS FAILURE_OUTCOME FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Failure%'
```

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

[33]:

Mission_Outcome	FAILURE_OUTCOME
Failure (in flight)	1

✓ We may use "COUNT" and LIKE "Success%" and "Failure%" to obtain the total number of successful and Failure missions.

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
[34]: %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.
```

[34]:

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

✓ We can get the maximum payload masses by using “MAX” function.

2015 Launch Records

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[55]:
```
%sql SELECT SUBSTR(DATE, 6,2) as MONTH, LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE SUBSTR(DATE, 0,5) LIKE '2015
```

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

[55]:

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

✓ Month(DATE) can be used to obtain the months, and "2015" is the year value that we supplied to the WHERE function.

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

Rank 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.

```
[76]: %sql SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS TOTAL_COUNT FROM SPACEXTBL where date BETWEEN '2010-06-04' and '2017-03-20' GROU
```

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

[76]:

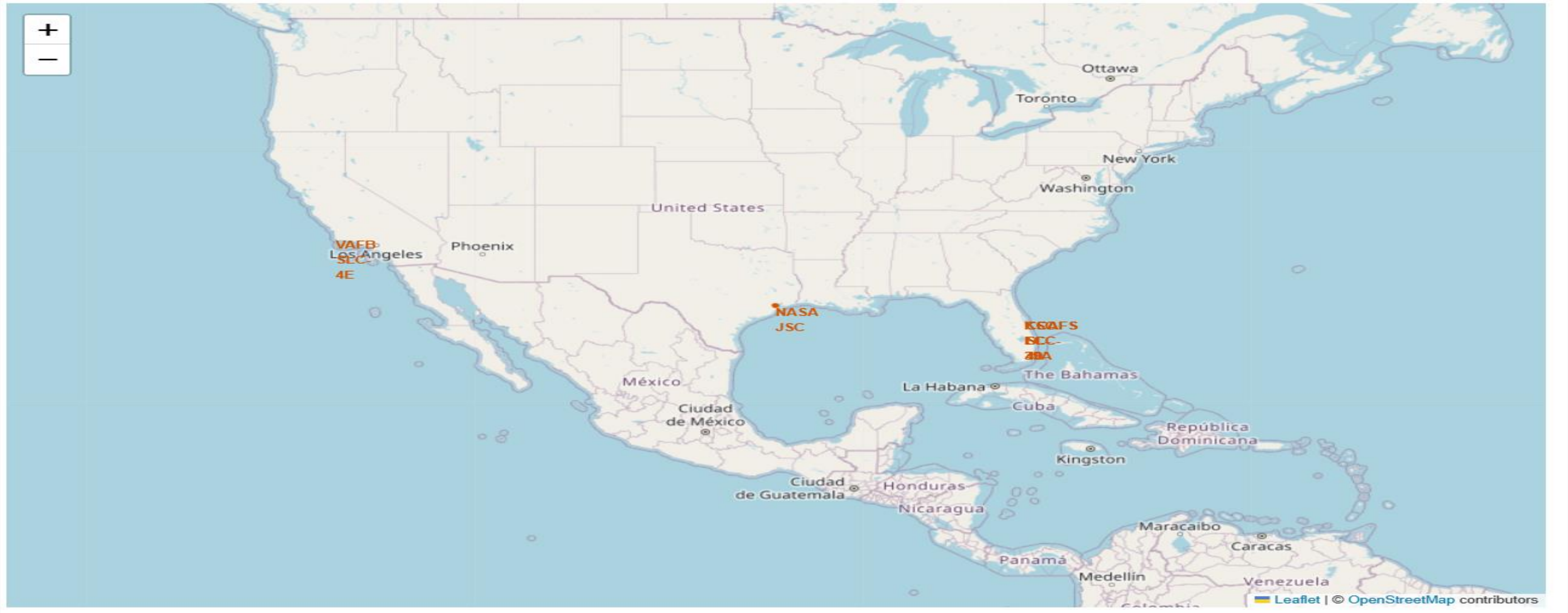
Landing_Outcome	TOTAL_COUNT
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

✓ We may arrange the values in decreasing order by using "ORDER," and we can count all of the numbers as we did before by using "COUNT."

Section 3

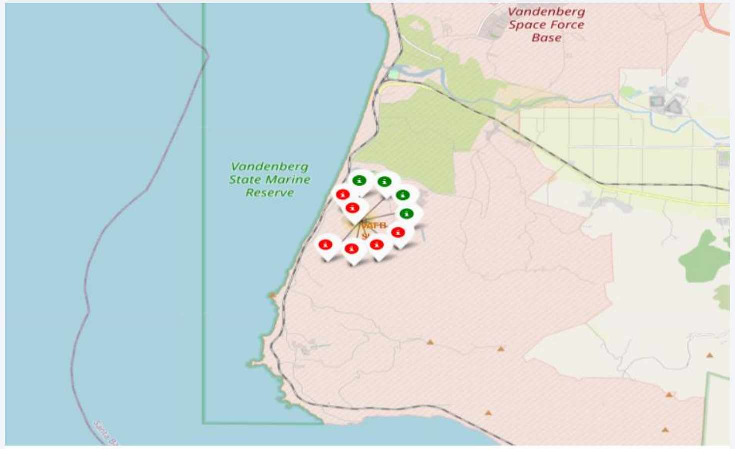
Launch Sites Proximities Analysis

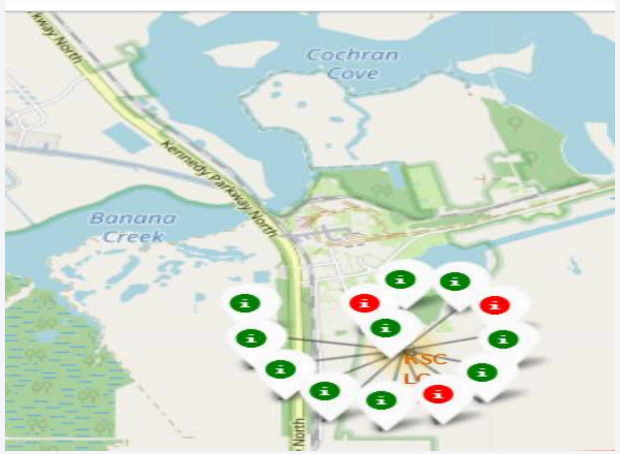
All Launch Sites Location Markers

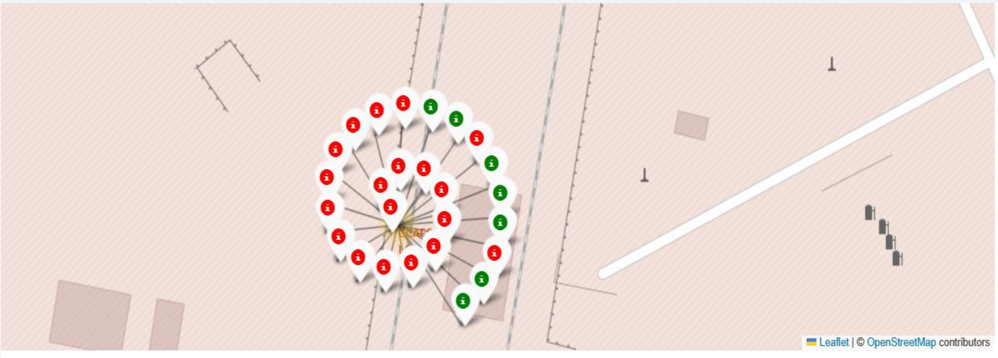


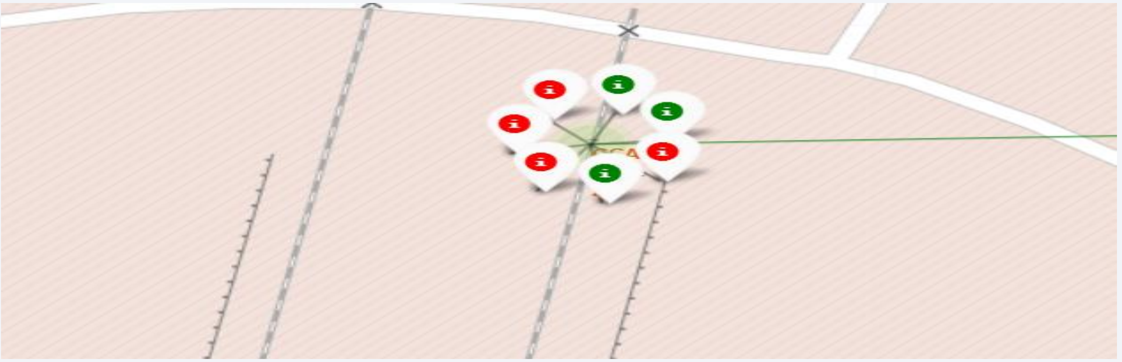
✓ Every launch occurs close to Florida, California, and the United States.

Color Labeled Launch Outcomes



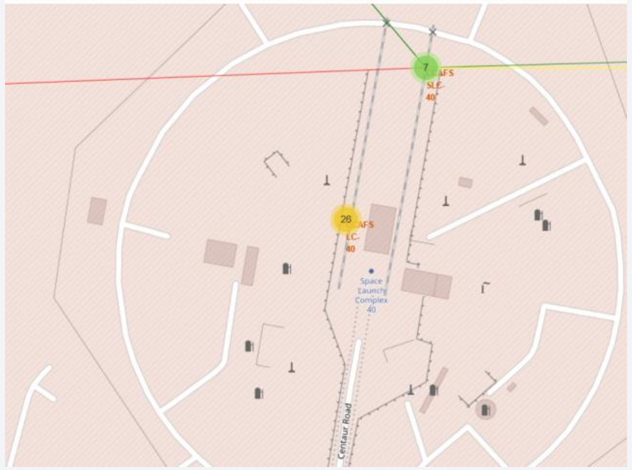






✓ Green means successful & Red means Failure

Launch Sites in Close Distance

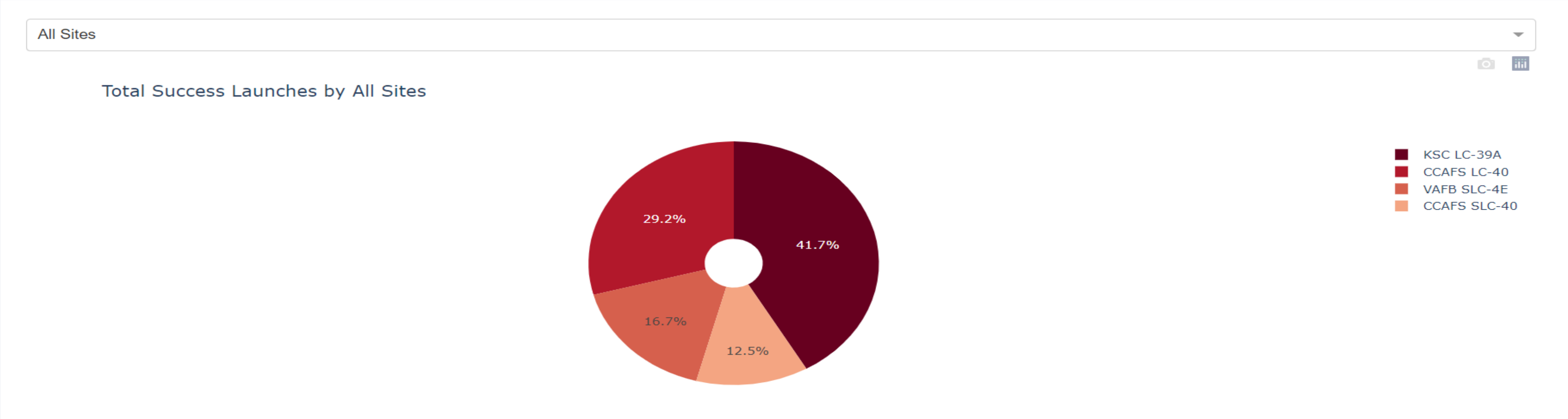


✓ The distances between the launch sites and their proximity to railway tracks were quite short.

Section 4

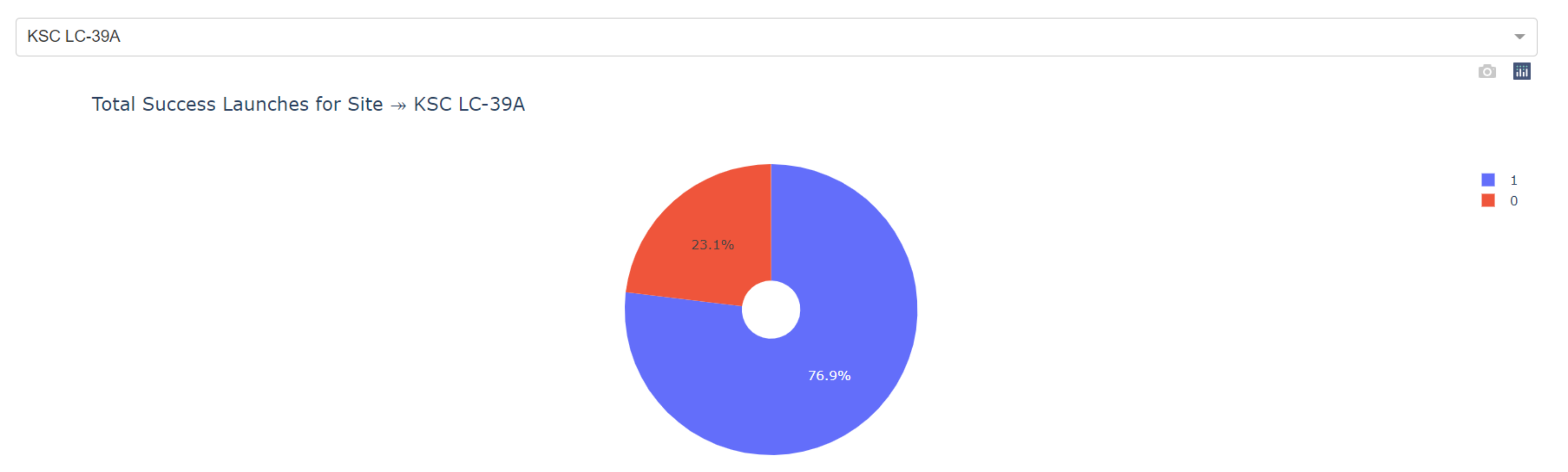
Build a Dashboard with Plotly Dash

Launch Performance Index



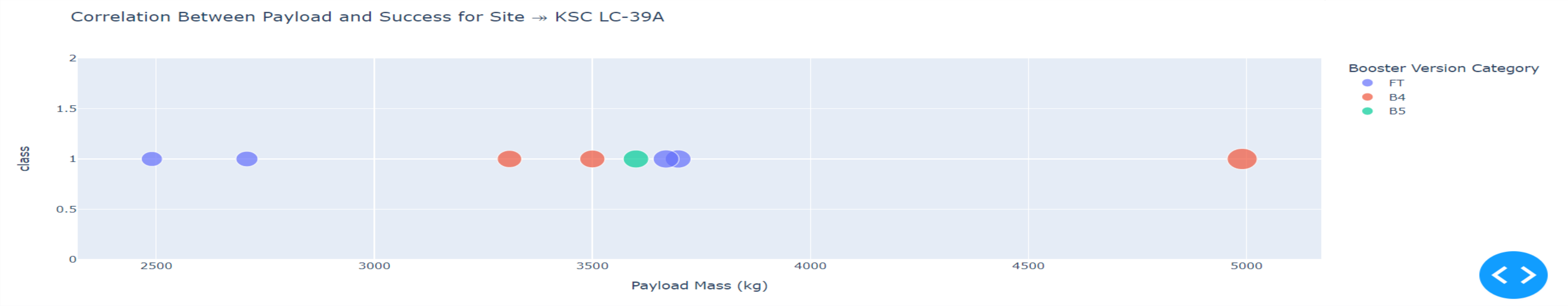
- ✓ With 41.7%, KSC LC 39A has the highest success rate.
- ✓ CCAFS LC 40 follows with a 29.2% share.
- ✓ Finally, with 16.7% and 12.5%, respectively, VAFB SLC 4E and CCAFS SLC 40.

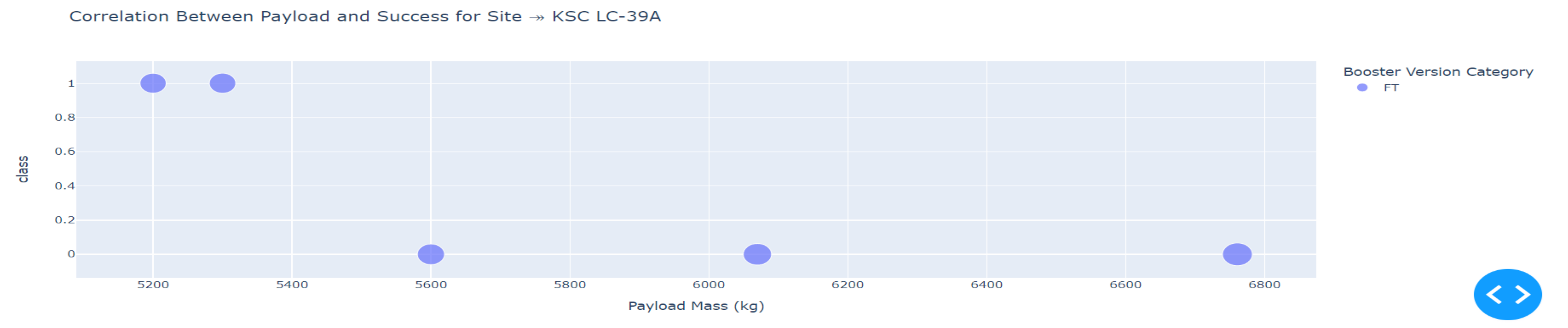
Launch Location with Best Rating



✓ With a payload range of 2000 kg to 10,000 kg, the KSC LC 39A gets the greatest score of 76.9%, while the FT booster version has the highest score.

Payload vs. Launch Outcome



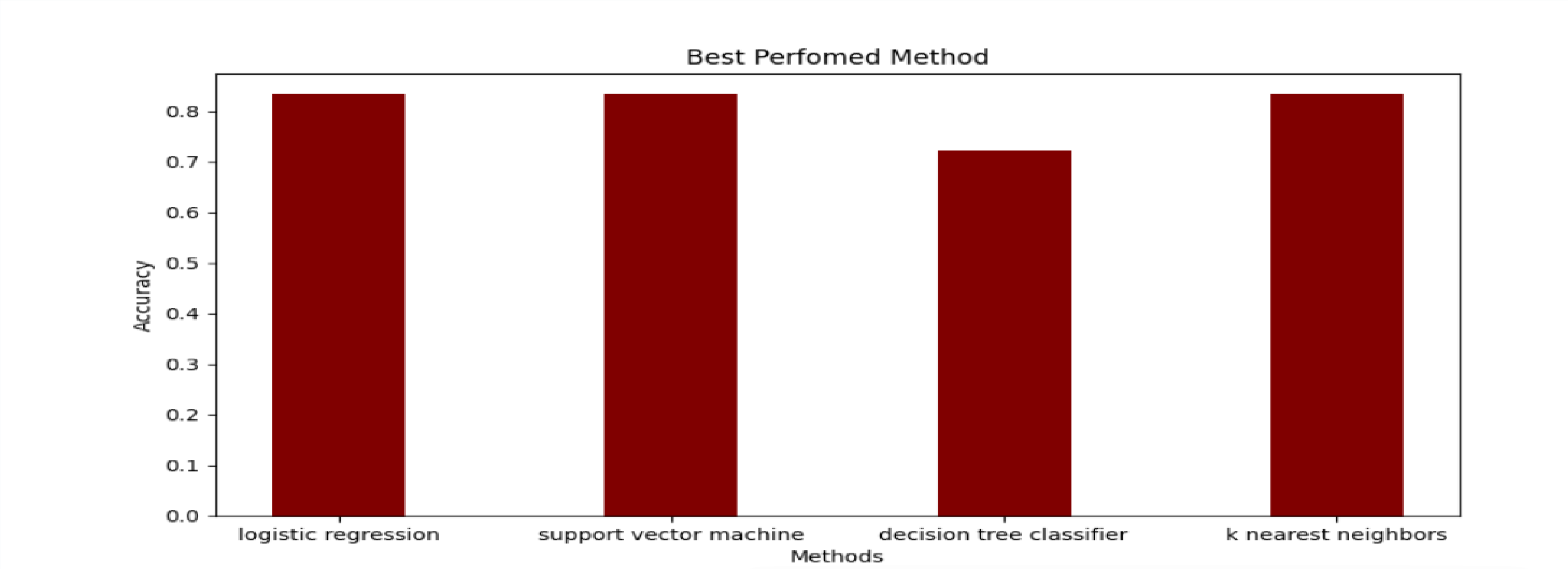


- ✓ First Screenshot shows the payload 0kg - 5000kg.
- ✓ Second Screenshot shows payload 6000kg -10000kg.

Section 5

Predictive Analysis (Classification)

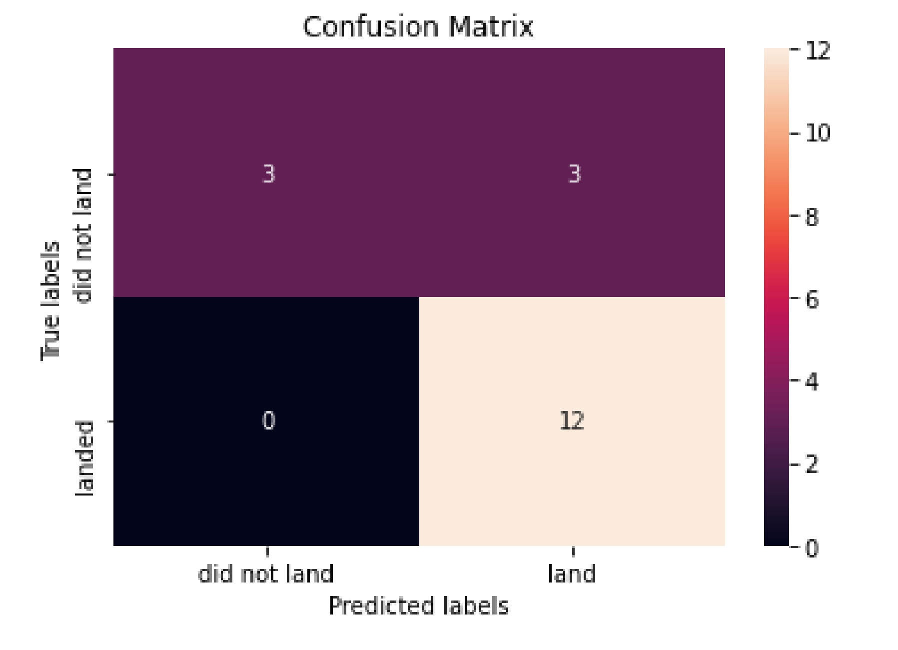
Classification Accuracy



✓ Here K nearest Neighbors have the highest accuracy and other three methods have the same accuracy hence we are choosing K nearest for classification.

Confusion Matrix

- ✓ The decision tree classifier's confusion matrix demonstrates the classifier's ability to discriminate between the various classes.
- ✓ False positives are the main issue.i.e., the classifier interprets an unsuccessful landing as a successful landing.
- ✓ The selected Logistic Regression model's confusion matrix is displayed in the graphic.
- ✓ Only three labels were incorrectly predicted by the model.



Conclusions

➢ We can then draw the following conclusion :

- ✓ The success rate at a launch site increases with the number of flights conducted there.
- ✓ The launch success rate increased from 2013 to 2020.
- ✓ The success percentage of rocket landings rose dramatically starting in 2015. It was also evident that the number of flights had an impact on landing success.
- ✓ Orbits with the highest success rate were ES-L1, GEO, HEO, SSO, and VLEO.
- ✓ Out of all the sites, KSC LC-39A had the most successful launches.
- ✓ For this problem, the optimal machine learning algorithm is the decision tree classifier.
- ✓ Their launch sites are all situated away from neighbouring cities and close to the coast. They were able to test their rocket landings with minimal interruption thanks to this.
- ✓ A machine learning model that can forecast the outcome of rocket launches with an accuracy of 83.33% was trained using all of this data.

Appendix

- ✓ Relevant codes have been submitted to the Github supplied below :

- ✓ Github link : https://github.com/Sivaraman-VishalRamanathan/FINAL_CAPSTON_PROJECT_IBM_DATA_SCIENCE/tree/main

Thank you!