IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dushyant Patel
November 5th, 2024

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 in screenshots
  - Predictive Analytics result

Introduction

- **Project Background and Context**
  - SpaceX promotes its Falcon 9 rocket launches on its website at a cost of $62 million, significantly lower than competitors, who charge upwards of $165 million. This cost advantage primarily stems from SpaceX's ability to reuse the first stage of the rocket. Therefore, accurately predicting whether the first stage will successfully land can help assess the overall cost of a launch. This information could also be valuable for alternate companies looking to compete with SpaceX for rocket launch contracts. The objective of this project is to develop a machine learning pipeline to forecast the success of the first stage landing.
- **Key Questions to Address**
  1. What factors influence the successful landing of the rocket?
  2. How do various features interact to determine the landing success rate?
  3. What operational conditions are necessary to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to 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
  - How to build, tune, evaluate classification models

Data Collection

- Data collection was conducted through various methods
    - Initially, we made GET requests to the SpaceX API to retrieve data.
    - We then decoded the response content as JSON using the .json() function and transformed it into a pandas DataFrame with the .json_normalize() method.
    - After that, we cleaned the data by checking for missing values and filling in those values where necessary.
    - Additionally, we performed web scraping on Wikipedia to obtain Falcon 9 launch records using BeautifulSoup.
    - The goal was to extract the launch records from an HTML table, parse the table, and convert it into a pandas DataFrame for further analysis.

Data Collection – SpaceX API

- We utilized GET requests to the SpaceX API to collect data, followed by cleaning the retrieved data and performing basic data wrangling and formatting.
- GitHub URL: https://github.com/dkpatel369/Applied-Data-Science-Capstone/blob/de298a4b66f47b403d53e37031196b6dbb218944/Hands-On%20Lab%3A%20Data%20Collection.ipynb

In [4]:

```python
# Takes the dataset and uses the payloads column to call the API and append the data to the lists
def getPayloadData(data):
    for load in data['payloads']:
       if load:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

From `cores` we would like to learn the outcome of the landing, the type of the landing, number of flights with that core, whether gridfins were used, wheter the core is reused, wheter legs were used, the landing pad used, the block of the core which is a number used to seperate version of cores, the number of times this specific core has been reused, and the serial of the core.

In [5]:

```python
# Takes the dataset and uses the cores column to call the API and append the data to the lists
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

Now let's start requesting rocket launch data from SpaceX API with the following URL:

In [6]:

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

In [7]:

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

Check the content of the response

In [8]:

```python
print(response.content)
```

Data Collection - Scraping

- We applied web scrapping to web scrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas DataFrame.
- GitHub URL: https://github.com/dkpatel369/Applied-Data-Science-Capstone/blob/de298a4b66f47b403d53e37031196b6dbb218944/Hands-On%20Lab%3A%20Data%20Collection%20with%20Web%20Scraping.ipynb

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

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

Create a `BeautifulSoup` object from the HTML `response`

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(page.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>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

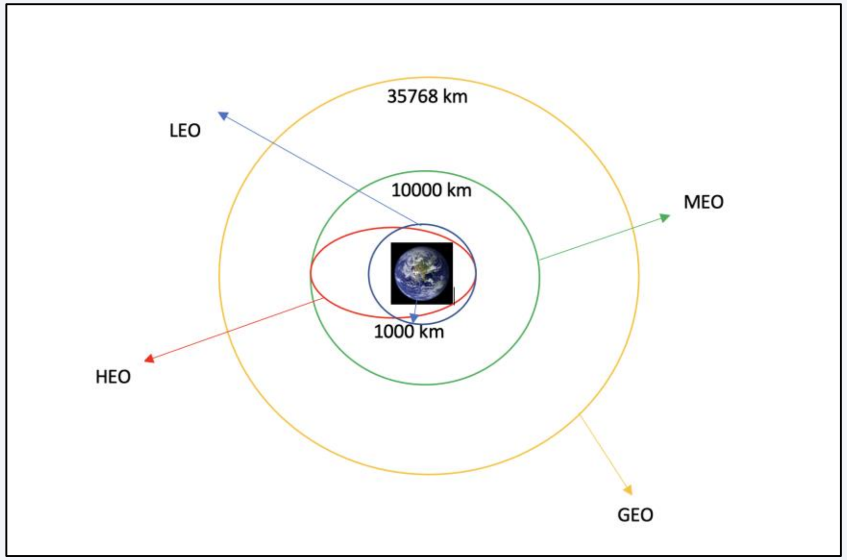
In [8]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

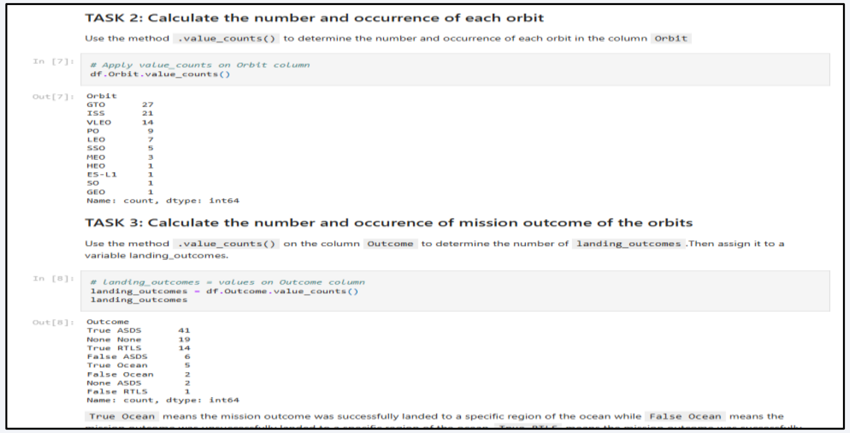
Starting from the third table is our target table contains the actual launch records.

In [9]:
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

Data Wrangling

- We conducted exploratory data analysis to define the training labels. We calculated the number of launches at each site and analyzed the frequency of each orbit type. Additionally, we generated landing outcome labels from the outcome column and exported the results to a CSV file.
- The handful of mission outcome types were converted to a binary classification where 1 means that the Falcon 9 first stage landing was a success and 0 means that it was a failure.
- GitHub URL: https://github.com/dkpatel369/Applied-Data-Science-Capstone/blob/de298a4b66f47b403d53e37031196b6dbb218944/Hands-On%20Lab%3A%20Data%20Wrangling.ipynb





TASK 2: Calculate the number and occurrence of each orbit

Use the method `.value_counts()` to determine the number and occurrence of each orbit in the column `Orbit`

In [7]:
```
# Apply value_counts on Orbit column
df.Orbit.value_counts()
```

Out[7]:
```
Orbit
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
HEO       1
ES-L1     1
SO        1
GEO       1
Name: count, dtype: int64
```

TASK 3: Calculate the number and occurence of mission outcome of the orbits

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes`.Then assign it to a variable landing_outcomes.

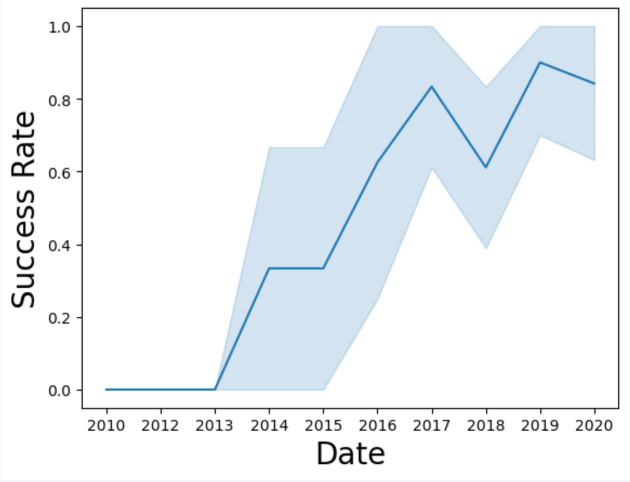
In [8]:
```
# landing_outcomes = values on Outcome column
landing_outcomes = df.Outcome.value_counts()
landing_outcomes
```

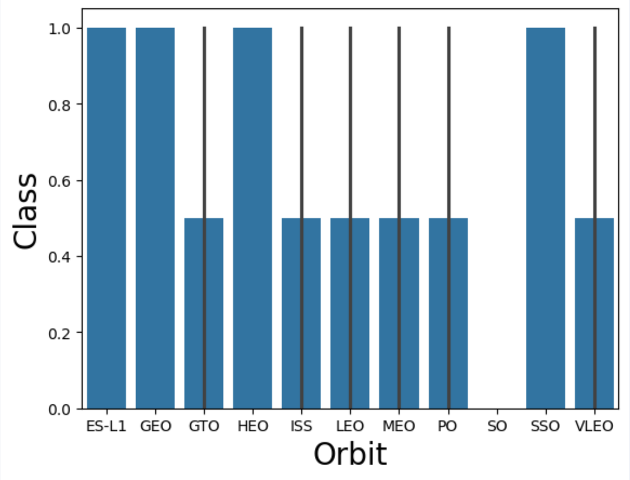
Out[8]:
```
Outcome
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: count, dtype: int64
```

`True Ocean` means the mission outcome was successfully landed to a specific region of the ocean while `False Ocean` means the

EDA with Data Visualization

- We examined the data by visualizing the relationships between flight number and launch site, payload and launch site, success rates for each orbit type, flight number and orbit type, as well as the yearly trend of launch success.
- GitHub URL: https://github.com/dkpatel369/Applied-Data-Science-Capstone/blob/de298a4b66f47b403d53e37031196b6dbb218944/Hands-On%20Lab%3A%20EDA%20with%20Visualization%20Lab.ipynb





EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database directly from the Jupyter notebook. We then performed exploratory data analysis using SQL to gain insights from the data. We wrote queries to determine:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS).
  - The average payload mass for the booster version F9 v1.1.
  - The total number of successful and failed mission outcomes.
  - The failed landing outcomes on drone ships, including the corresponding booster versions and launch site names.
- GitHub URL: https://github.com/dkpatel369/Applied-Data-Science-Capstone/blob/de298a4b66f47b403d53e37031196b6dbb218944/Hands-on%20Lab%3A%20Complete%20the%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
    - Are launch sites near railways, highways and coastlines.
    - Do launch sites keep certain distance away from cities.
- GitHub URL: https://github.com/dkpatel369/Applied-Data-Science-Capstone/blob/de298a4b66f47b403d53e37031196b6dbb218944/Hands-on%20Lab%3A%20Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

- The input dropdown is used to select one or all launch sites for the pie chart and scatterplot.
- The pie chart displays one of two things:
    - For All Sites – the distribution of successful Falcon 9 first stage landings between the sites
    - For One Site – the distribution of successful and failed Falcon 9 first stage landings for that site
- The input slider is used to filter the payload masses for the scatterplot.
- The scatterplot displays the distribution of Falcon 9 first stage landings split by payload mass, mission outcome and by booster version category
- GitHub URL: https://github.com/dkpatel369/Applied-Data-Science-Capstone/blob/de298a4b66f47b403d53e37031196b6dbb218944/Hands-on%20Lab%3A%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.py

Predictive Analysis (Classification)

- The dataset was split into training and testing sets.
- Logistic Regression, SVM (Support Vector Machine), Decision Tree, and KNN (k-Nearest Neighbors) machine learning models were trained on the training data set.
- Hyper-parameters were evaluated using GridSearchCV() and the best was selected using ‘.best_params_’.
- Using the best hyper-parameters, each of the four models were scored on accuracy by using the testing data set.
- GitHub URL: https://github.com/dkpatel369/Applied-Data-Science-Capstone/blob/de298a4b66f47b403d53e37031196b6dbb218944/Hands-on%20Lab%3A%20Complete%20the%20Machine%20Learning%20Prediction%20lab.ipynb

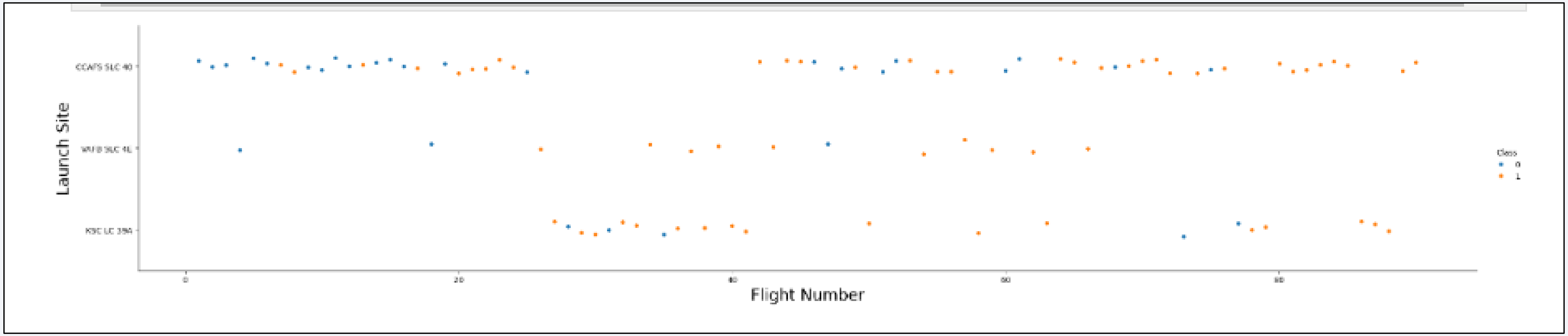
Results

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

Section 2

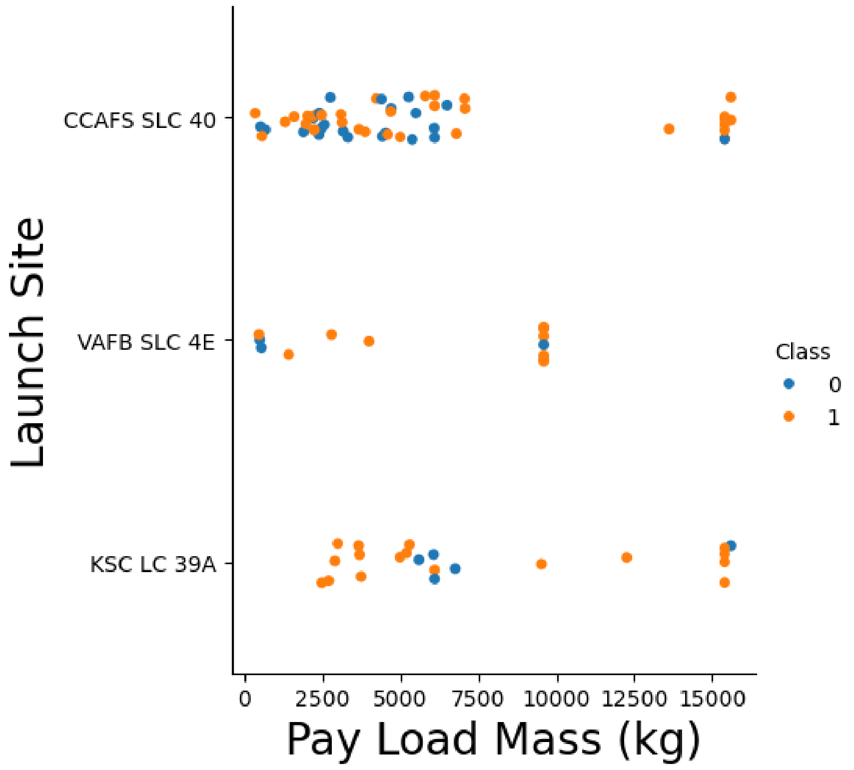
Insights drawn from EDA

Flight Number vs. Launch Site



From the plot, we observed that a higher number of flights at a launch site is associated with a greater success rate.

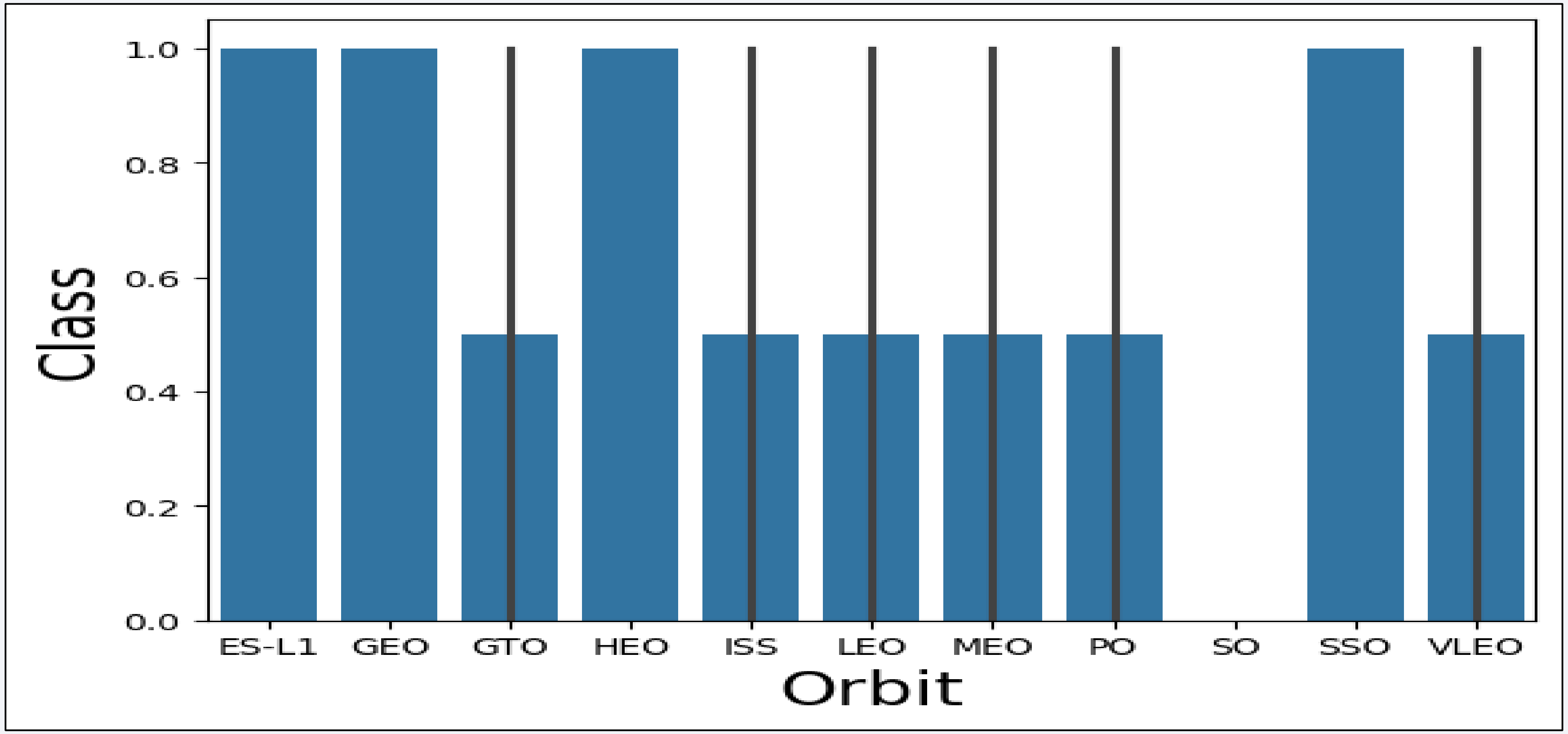
Payload vs. Launch Site



Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

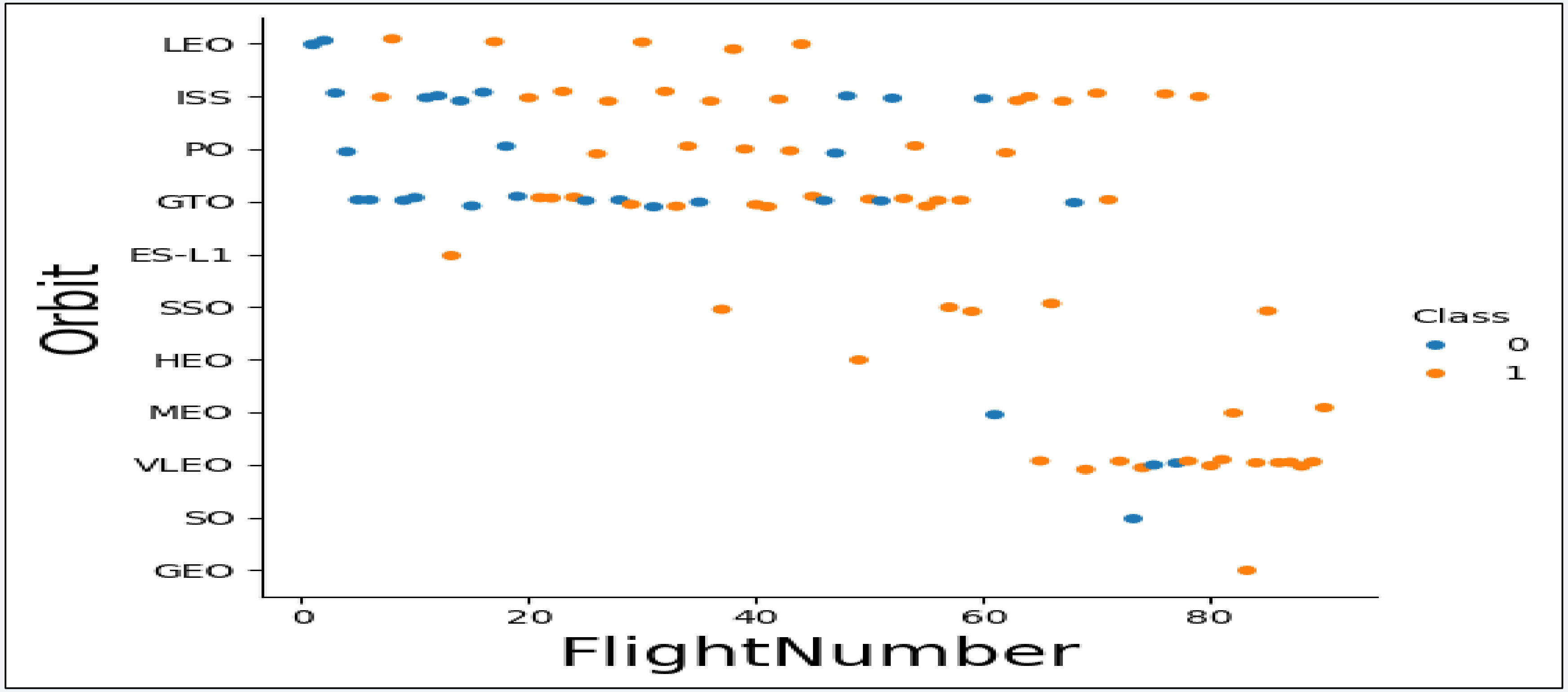
- For the CCAFS SLC 40 launch, the payload mass and the landing outcome appear not to be strongly correlated.
- The failed landings at the KSC LC 39A launch site are all grouped around a narrow band of payload masses.

Success Rate vs. Orbit Type



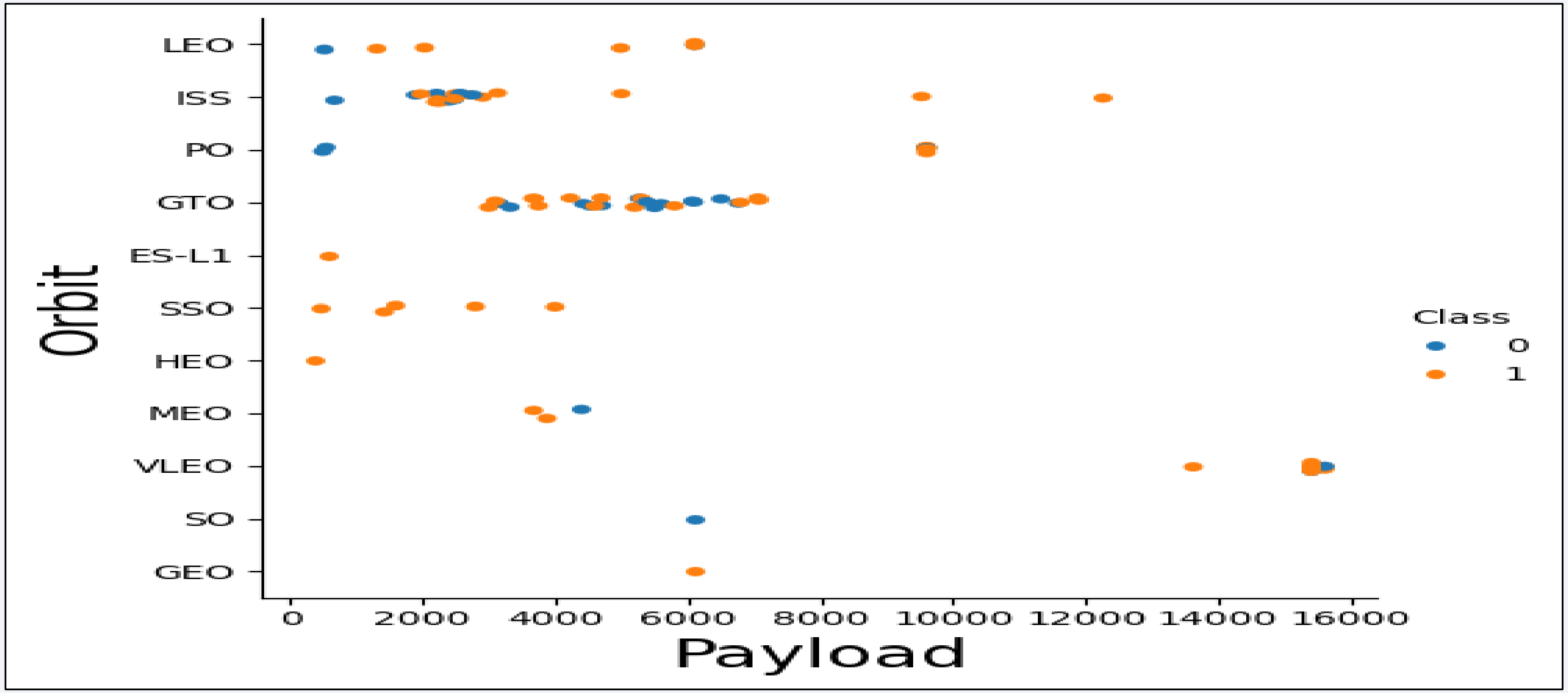
- ES-L1, SSO, HEO and GEO orbits have no failed first stage landings.
- SO orbits have no successful first stage landings.

Flight Number vs. Orbit Type



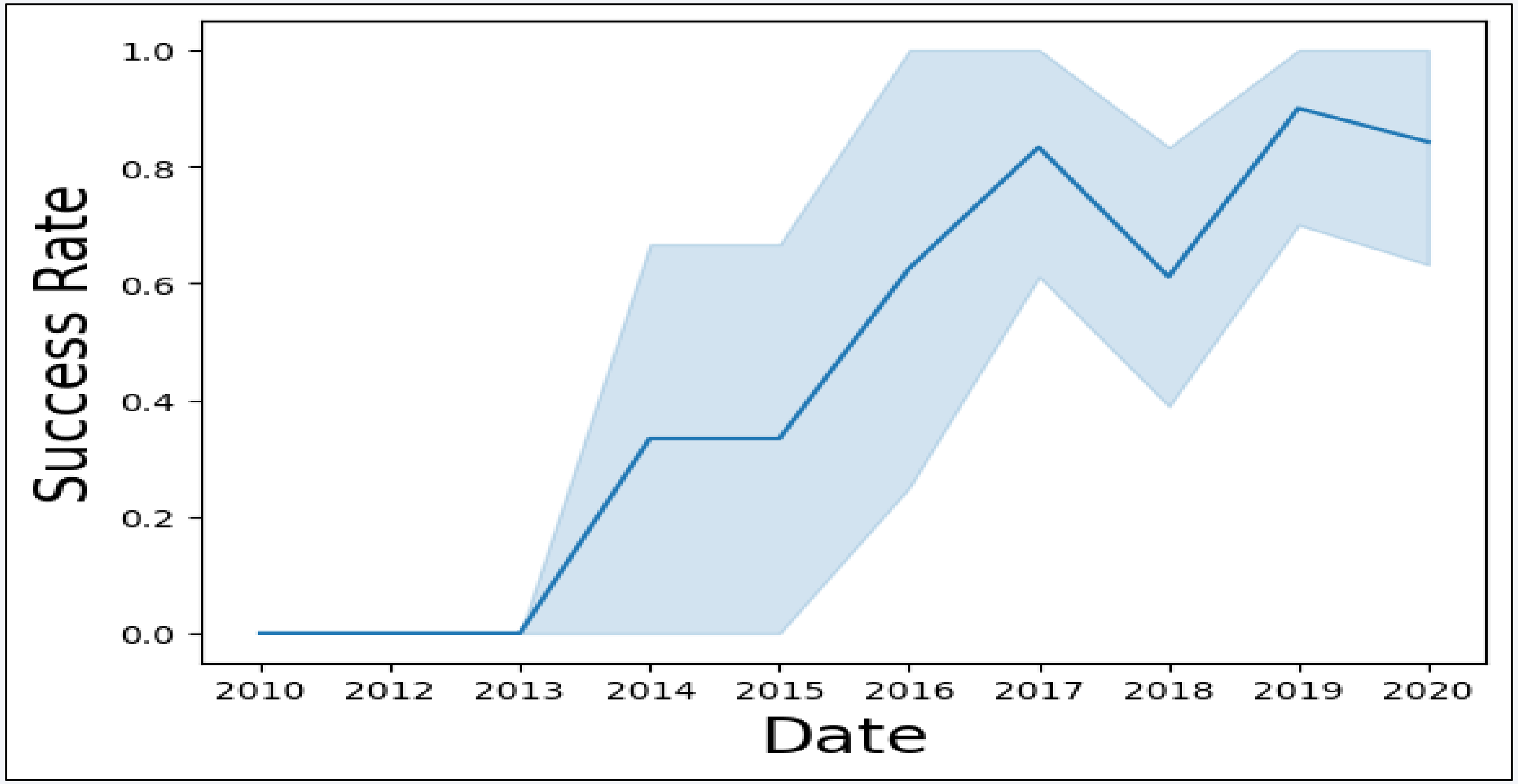
- There is a correlation between flight number and success rate with larger flight numbers being associated with higher success rates.

Payload vs. Orbit Type



- Some orbit types have better success rates than others.
- Success rate appears to have no obvious correlation with payload mass.

Launch Success Yearly Trend



The success rate has increased significantly over the years.

All Launch Site Names

- Unique Launch Sites:

Task 1

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

In [10]:

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

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

Out[10]:

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

- There are 4 unique launch sites.

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

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

In [11]:

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

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

Out[11]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- The total payload carried by boosters from NASA is 45596 kg.

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

In [12]:
```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Customer = 'NASA (CRS)';
```

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

Out[12]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1 is 340.4 kg.

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

In [13]:

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Booster_Version LIKE 'F9 v1.0%';
```

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

Out[13]:

AVG(PAYLOAD_MASS__KG_)
340.4

First Successful Ground Landing Date

- First successful landing outcome on ground pad date is Dec. 22^{nd}, 2015.

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

Hint:Use min function

In [15]:

```
%%sql
SELECT MIN(Date)
FROM SPACEXTBL
WHERE Landing_Outcome = 'Success (ground pad)';
```

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

Out[15]:

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List of names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

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

In [17]:

```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE LANDING_OUTCOME = 'Success (drone ship)'
    AND 4000 < PAYLOAD_MASS__KG_ < 6000;
```

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

Out[17]:

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

In [18]:

```
%%sql
SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME;
```

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

Out[18]:

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

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass.

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

In [19]:
```
%%sql
SELECT DISTINCT BOOSTER_VERSION
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL);
```

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

Out[19]:

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

- Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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.

In [22]:

```
%%sql
SELECT LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE Landing_Outcome = 'Failure (drone ship)'
    AND strftime('%Y', date) = '2015'
```

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

Out[22]:

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

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

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.

In [23]:

```
%%sql
SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING_OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

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

Out[23]:

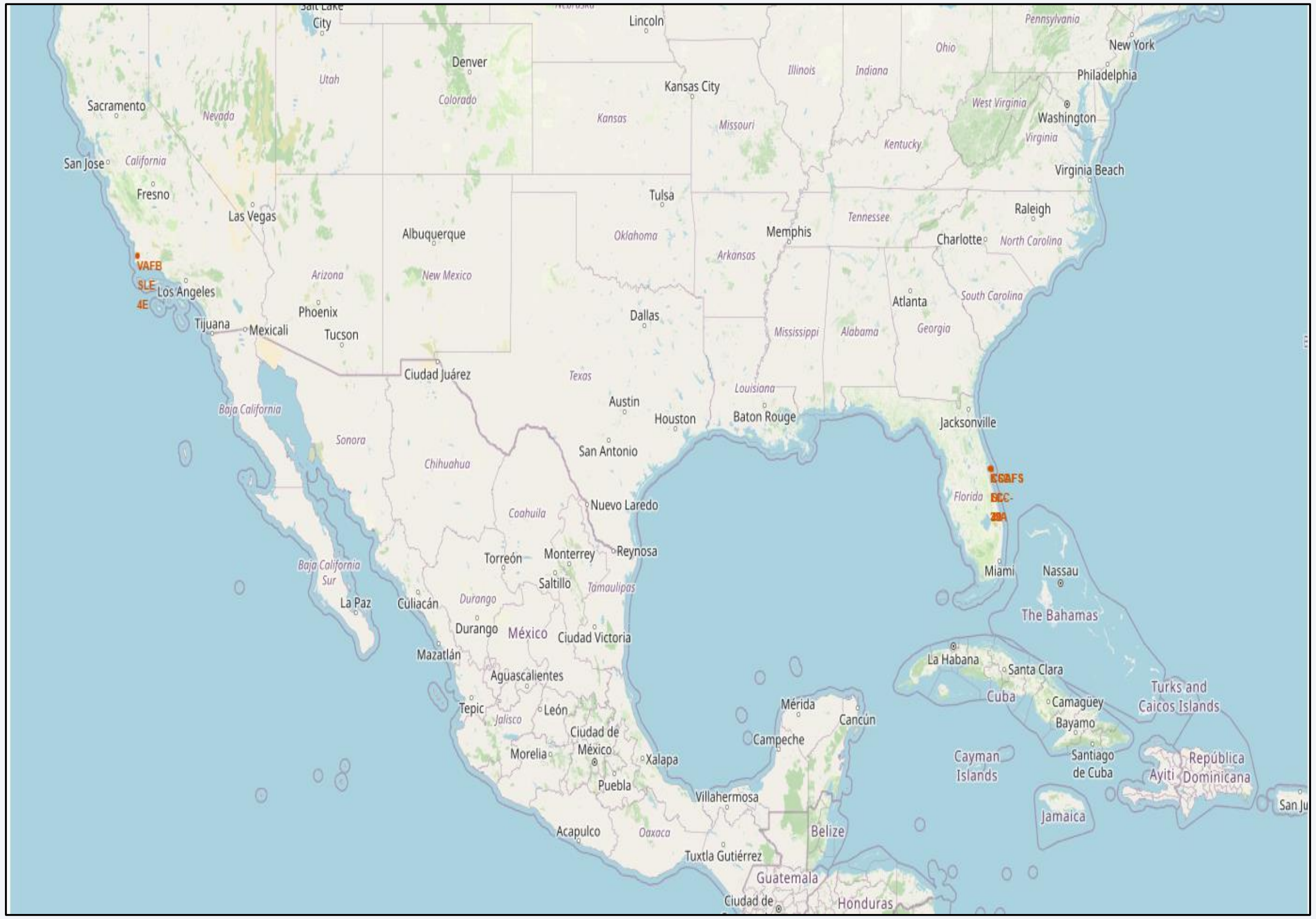
Landing_Outcome	TOTAL_NUMBER
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

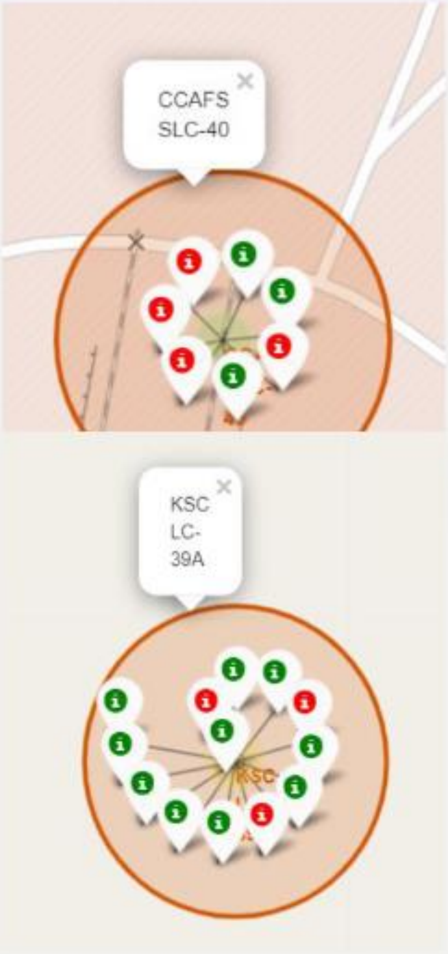
Launch Sites Proximities Analysis

All Launch Sites (GLOBAL)

- VAFB SLC-4E (California, USA)
  - Vandenberg Air Force Base Space Launch Complex 4E
- KSC LC-39A (Florida, USA)
  - Kennedy Space Center Launch Complex 39A
- CCAFS LC-40 (Florida, USA)
  - Cape Canaveral Air Force Station Launch Complex 40
- CCAFS SLC-40 (Florida, USA)
  - Cape Canaveral Air Force Station Space Launch Complex 40



Markers showing launch sites with color labels



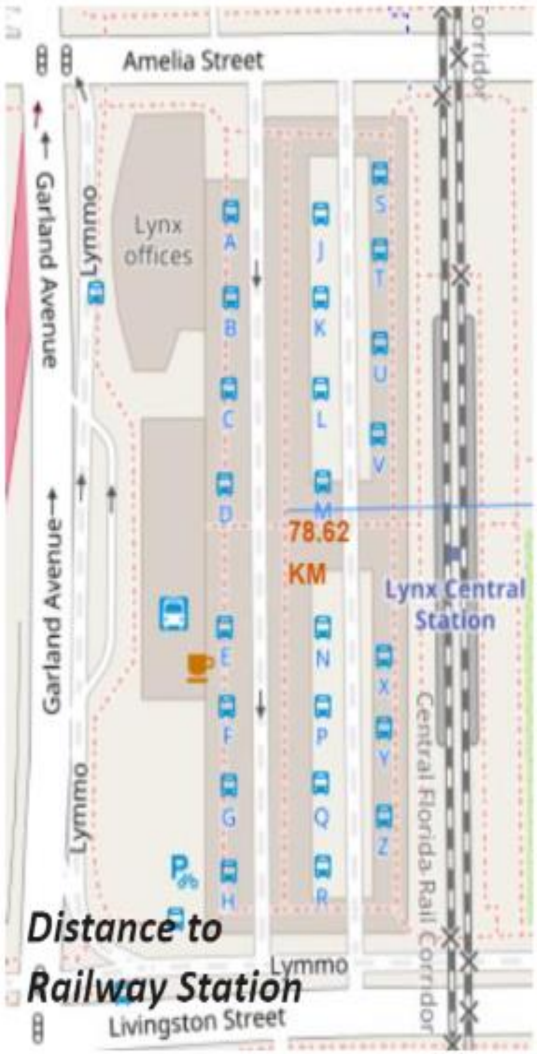
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

Launch Site distance to landmarks

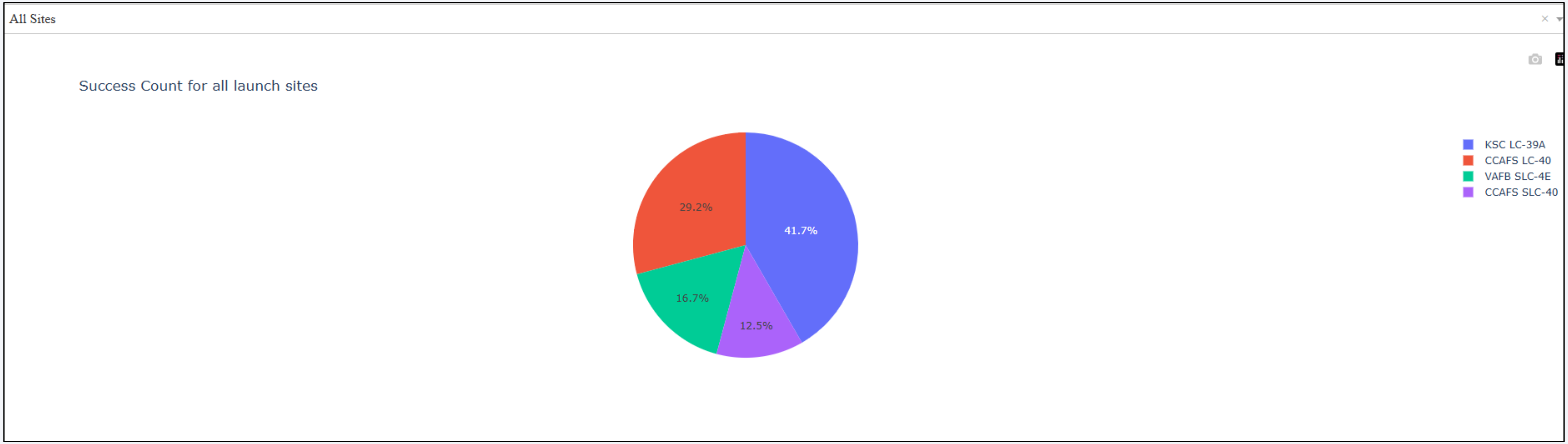


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

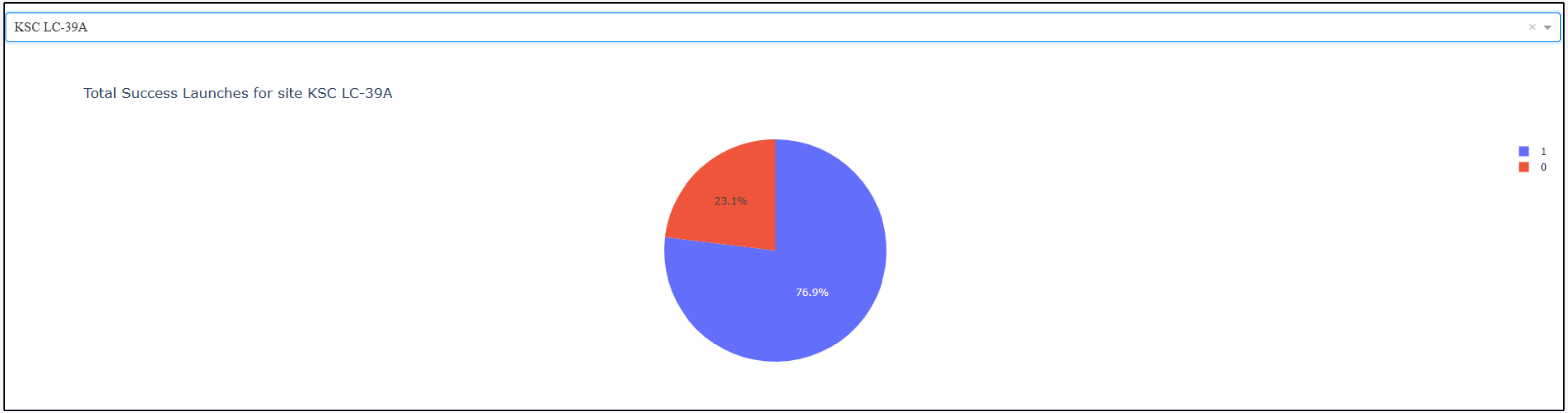
Build a Dashboard with Plotly Dash

Success Counts for all Launch Sites



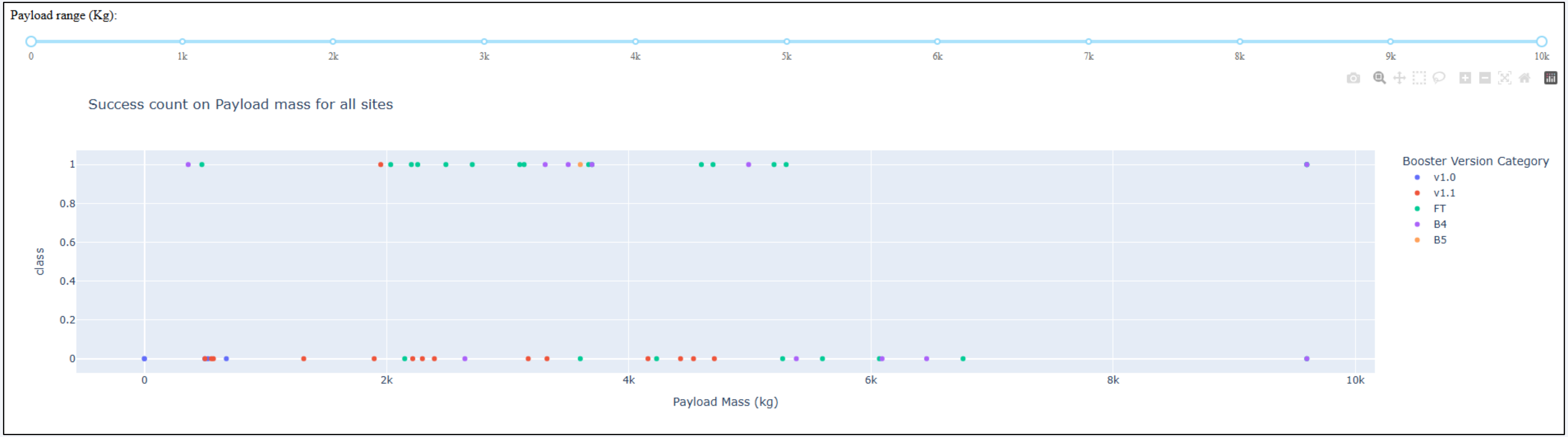
- The dropdown menu allows the selection of one or all launch sites.
- With all launch sites selected, the pie chart displays the distribution of successful Falcon 9 first stage landing outcomes between the different launch sites.
- The greatest share of successful Falcon 9 first stage landing outcomes (at 41.7% of the total) occurred at KSC LC-39A.

Launch site with Highest Launch Success Ratio



KSC LC-39A has the highest launch success ratio of 76.9% and failure of 23.1%

Payload vs Launch Outcome Scatter Plot

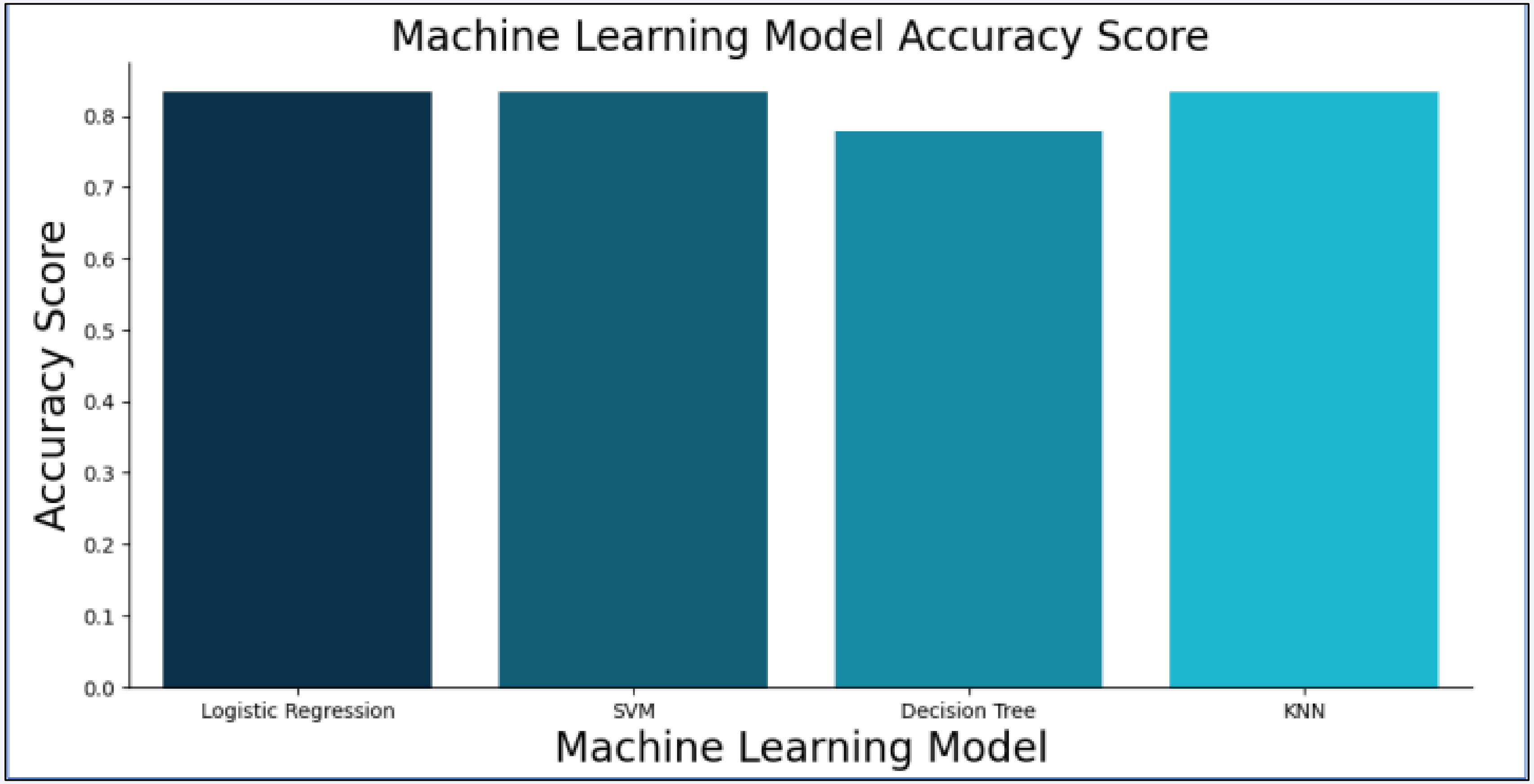


- The payload range from about 2,000 kg to 5,000 kg has the highest success rate.
- The ‘FT’ booster version category has the largest success rate.

Section 5

Predictive Analysis (Classification)

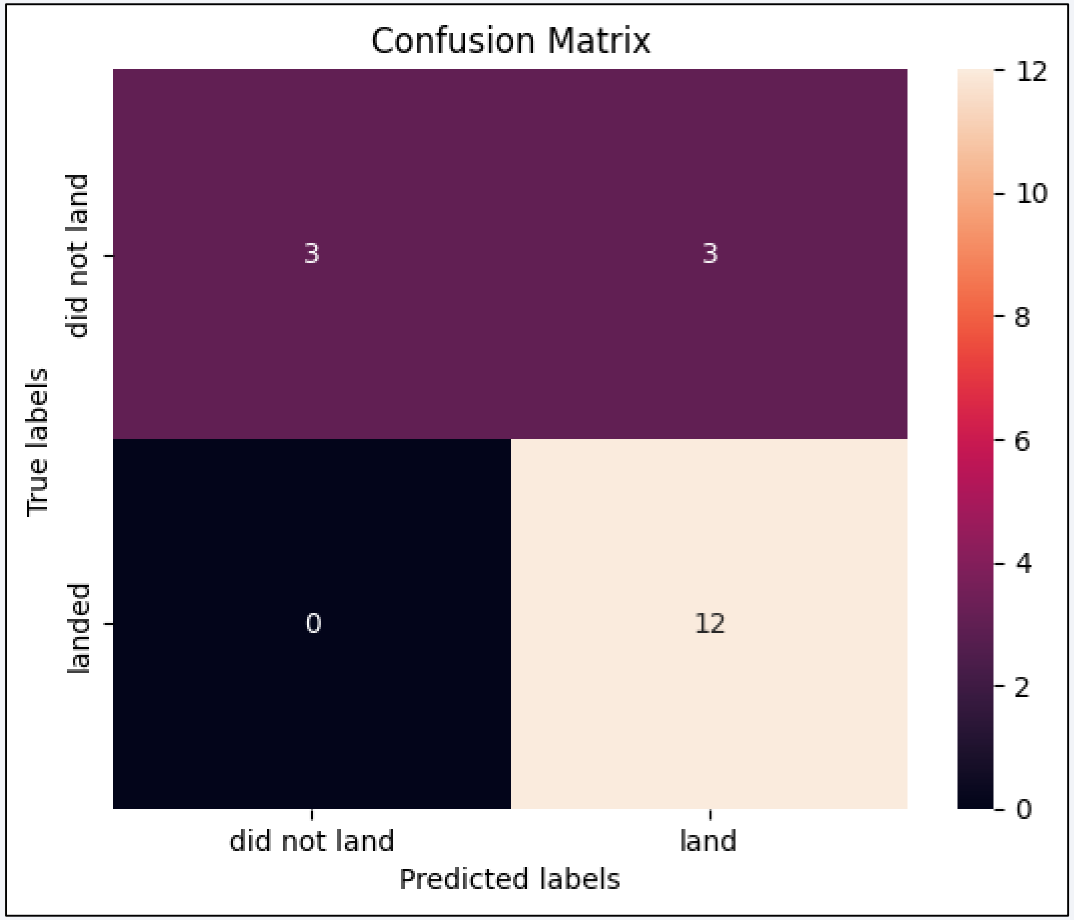
Classification Accuracy



- All models performed equally well except for the Decision Tree model which performed poorly relative to the other models.

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.
- Prediction Breakdown:
    - 12 True Positives and 3 True Negatives
    - 3 False Positives and 0 False Negatives



Conclusions

- A higher volume of flights at a launch site correlates with a greater success rate.
- The launch success rate showed an upward trend from 2013 to 2020.Orbits ES-L1, GEO, HEO, SSO, and VLEO achieved the highest success rates.
- The KSC LC-39A site had the highest number of successful launches among all sites.
- The Decision Tree classifier proved to be the most effective machine learning algorithm for this analysis.

Appendix

- Initial Data Sets
  - SpaceX API (JSON): https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json
  - Wikipedia (Webpage): https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
  - SpaceX (CSV): https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN
  - Launch Geo (CSV): https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_geo.csv
  - Launch Dash (CSV): https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_dash.csv

Thank you!