IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Erandi T. Sandarenu

08/29/2024

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 Pandas & Matplotlib

  - Interactive Visual Analytics with Folium & Dashboard

  - Machine Learning Predictive Analysis

- Summary of all results

  - Results from the EDA

  - Interactive Visuals

  - Predictive Analysis Results

Introduction

- Project background and context

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. Spaces X's Falcon 9 launch like regular rockets. But unlike other rocket providers, SpaceX's Falcon 9 can recover the first stage. The aim of the project is to determine the cost of each launch & also determine if SpaceX will reuse the first stage.

- Problems you want to find answers

-What are the factors which determine if the rocket will land successfully?

- What are the relationships among various features that determine the success rate of a successful landing?

- What are the conditions required to ensure a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from SpaceX REST API and web scraping related Wiki pages.
- 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
  - Machine Learning Pipeline was built to predict if the first stage of Falcon 9 lands successfully.

Data Collection

- From a SpaceX API

  - Initially data was collected using get request to the SpaceX REST API.

  - Then the response, the structured JSON data, was converted to a Pandas dataframe using json_normalize function.

  - Next, the data was cleaned by checking the missing values and replacing the missing values where necessary.

- Using Web Scraping

  - Also, using the BeautifulSoup package, we did web scrape some HTML tables that contain Falcon 9 launch records.

  - Next, the data was parsed from those tables and converted into a Pandas dataframe for further visualization and analysis.

Data Collection – SpaceX API

- We requested rocket launch data from SpaceX API using request.get() function , cleaned the dataset and replaced missing values.

- Link to the Notebook : https://github.com/Erandi-Sandarenu/IBM_Data_Science_Capstone_SpaceX/blob/main/01-spacex-data-collection-api.ipynb

1. Get request to rocket launch data using API

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

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

2. Use json_normalize method to convert json file into Pandas dataframe

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

3. Filter the dataframe to only include Falcon 9 launches

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df[df.BoosterVersion == 'Falcon 9']
data_falcon9
```

4. Perform data cleaning & filling the missing values

```
# Calculate the mean value of PayloadMass column
Mean_PayloadMass = data_falcon9.PayloadMass.mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, Mean_PayloadMass)
```

Data Collection - Scraping

- First we requested the Falcon 9 Launch Wiki page from its URL
- Then, we extracted all column/variable names from the HTML table header
- Next, we created a Pandas dataframe by parsing the launch HTML tables
- Link to the Notebook : https://github.com/Erandi-Sandarenu/IBM_Data_Science_Capstone_SpaceX/blob/main/02-spacex-webscraping.ipynb

1. Use HTTP GET method to request the Falcon 9 Launch HTML page.

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

2. Create a BeautifulSoup object from the HTML response

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response)
```

```
# Use soup.title attribute
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Use extract_column_from_header() to extract column name one by one

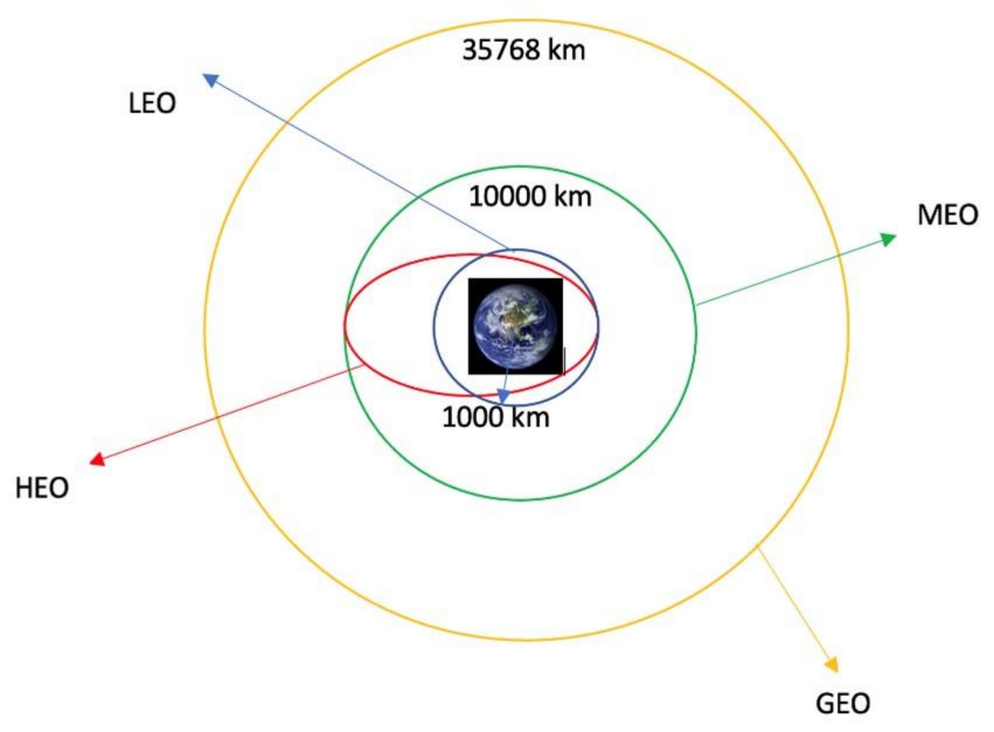
```
column_names = []
labels = first_launch_table.find_all('th')
for label in labels:
    name = extract_column_from_header(label)
   # header = str(label.text).strip()
    #header = str(header).split("($)Footnote", 1)[0]
    if name != None:
        if len(name) > 0:
            column_names.append(name)

# 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
```

4. Create a dataframe by parsing the launch HTML tables

5. Export data to CSV

Data Wrangling

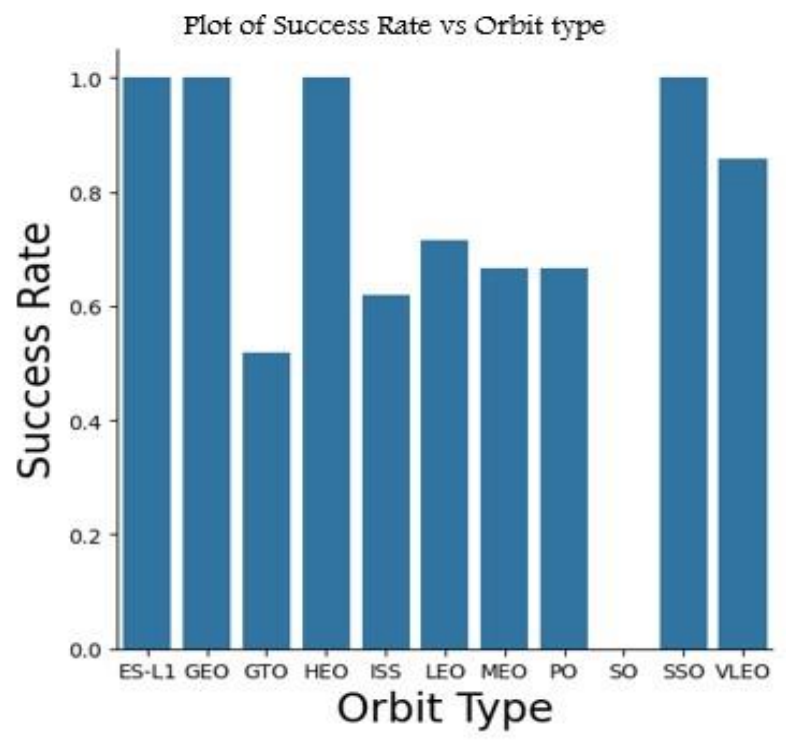
- First, we did the EDA by identifying & calculating the missing values in each attribute and identifying which columns are numerical & categorial.
- Then we calculated the number of launches in each site, and the number and occurrence of each orbit.
- Also, we calculated the number and occurrence of mission outcome of each orbit.
- Lastly, we created a landing outcome label from outcome column.

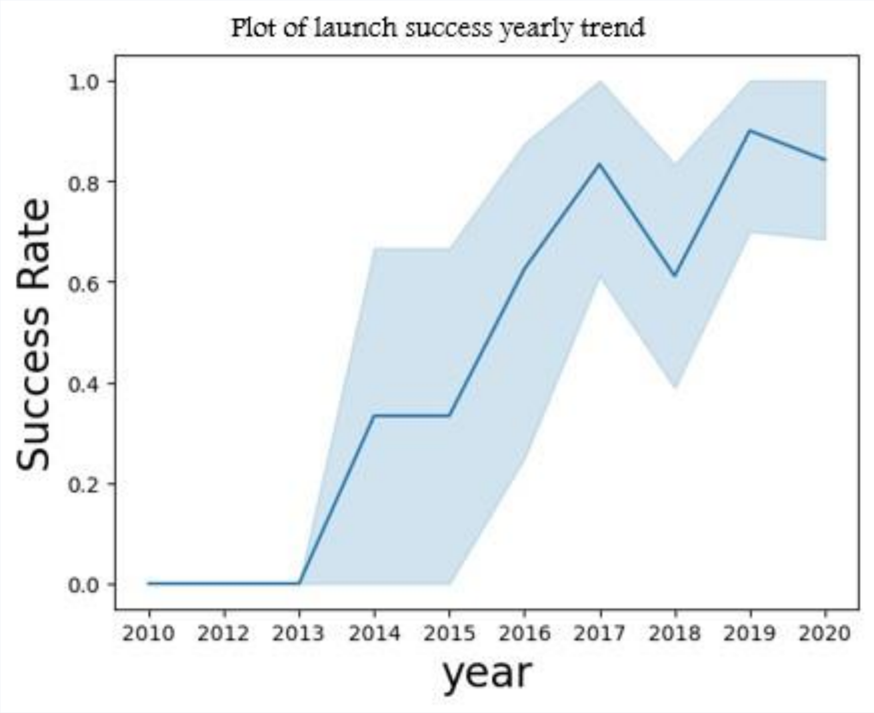


- Link to the Notebook : https://github.com/Erandi-Sandarenu/IBM_Data_Science_Capstone_SpaceX/blob/main/03-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- We used visualization to figure out the relationships between Flight Number & Launch Site, Payload Mass & Launch Site, success rate at each orbit type, Flight Number & Orbit type, Payload Mass & Orbit type and the launch success yearly trend.





Link to the Notebook : https://github.com/Erandi-Sandarenu/IBM_Data_Science_Capstone_SpaceX/blob/main/05-spacex-eda-data-visualization.ipynb

EDA with SQL

- We applied EDA with SQL to get insights from the data. We wrote queries to find out the followings :

  - Names of the unique launch sites in the space mission
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - Total number of successful and failure mission outcomes
  - Records for the failed landing outcomes in drone ship, their booster versions and launch site names

- Link to the Notebook : https://github.com/Erandi-Sandarenu/IBM_Data_Science_Capstone_SpaceX/blob/main/04-spacex-eda-sql.ipynb

Build an Interactive Map with Folium

- We marked all launch sites on a map using each site's latitude and longitude coordinates and we added map objects such as circles, markers, lines to indicate successful and failed launches.
- We assigned the feature launch outcomes to class 0 for failure and to class 1 for success.
- Adding marker cluster is a good way to simplify a map containing many markers having the same coordinates.
- Using the color-labeled markers in marker clusters, we were able to easily identify which launch sites have relatively high success rates.

Build an Interactive Map with Folium – Cont'd

- Then, we calculated the proximities of launch sites and got answers for the followings :

  - Are launch sites in close proximity to railways, highways, coastline?

  - Do launch sites keep certain distance away from cities?

- Link to the Notebook : https://github.com/Erandi-Sandarenu/IBM_Data_Science_Capstone_SpaceX/blob/main/06-spacex-interactive-visual-analytics-folium.ipynb

Folium

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with the following steps :

  - Adding a launch site drop-down input component

  - Adding a callback function to render success-pie-chart based on selected site dropdown

  - Adding a ranger slider to a selected payload

  - Adding a callback function to render the success-payload-scatter-chart scatter plot

- We were able to find the site which has the largest successful launches, which has the highest success rate, the payload range(s) which has the highest & lowest launch success rate, and F9 booster version which has the highest launch success rate by using the dashboard we built.

- Link to the Notebook : https://github.com/Erandi-Sandarenu/IBM_Data_Science_Capstone_SpaceX/blob/main/07-spacex_dash_app.py

Predictive Analysis (Classification)

- We loaded the data using Numpy and Pandas, standardized them, and split them into training and test data using 'train_test_split' function.
- Then, we created logistic regression, SVM, decision tree classifier, KNN obejcts and performed GridSearchCV object.
- Also we calculated the accuracy for each method mentioned above and plotted the confusion matrix.
- We found the best performing classification model.
- Link to the Notebook : https://github.com/Erandi-Sandarenu/IBM_Data_Science_Capstone_SpaceX/blob/main/08-spacex-machine-learning-prediction.ipynb

Results

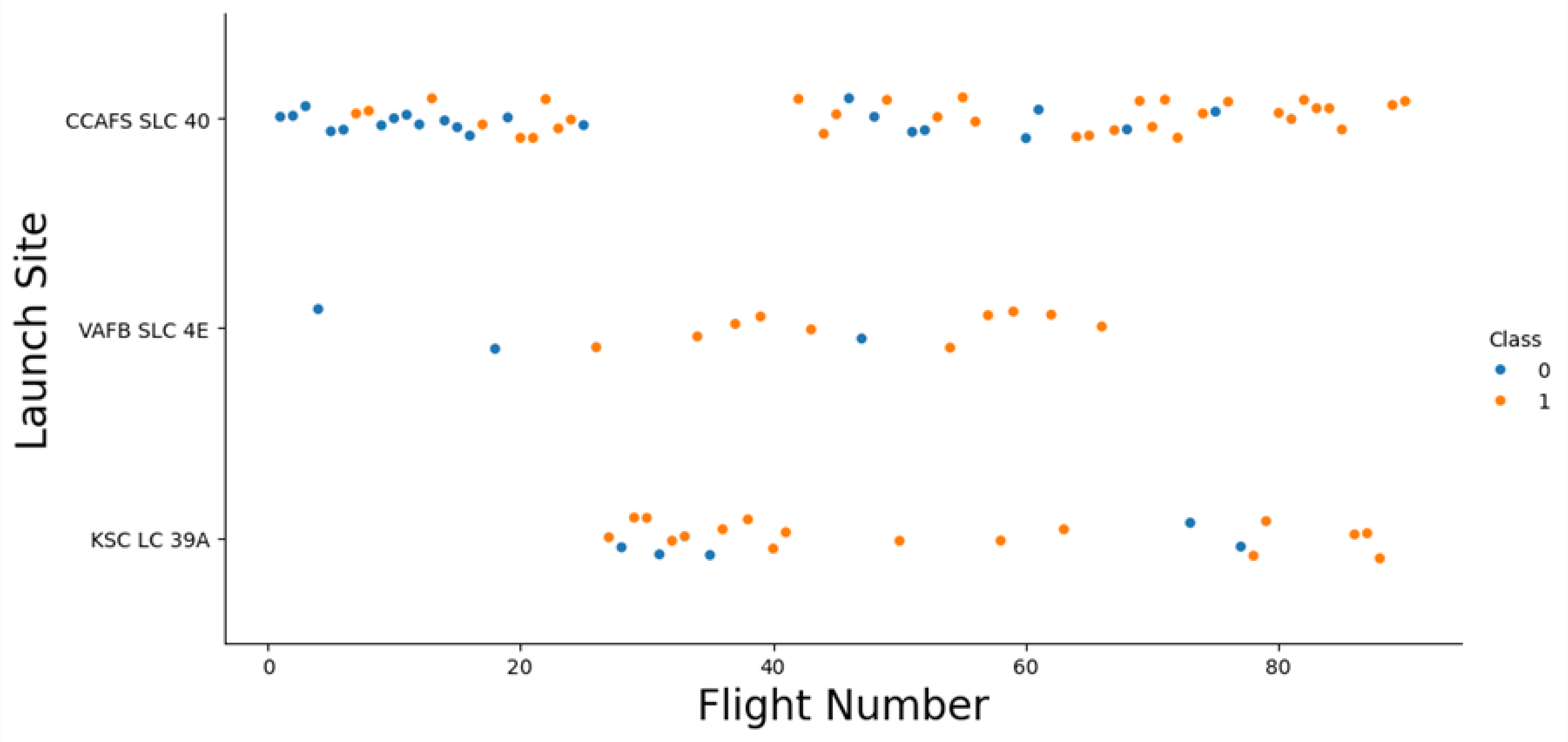
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

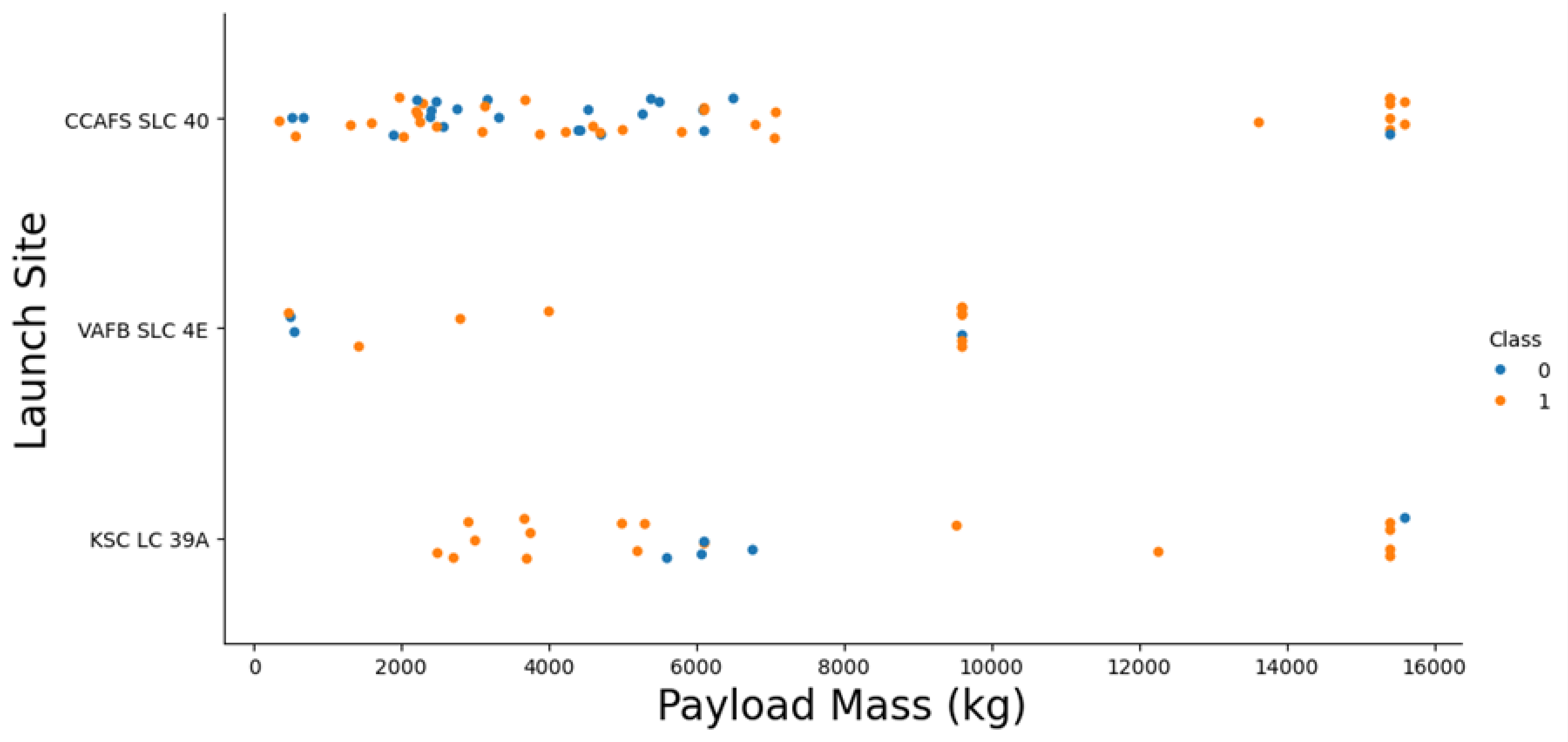
Flight Number vs. Launch Site

- As shown in the figure, when the Flight Number increases at a launch site, the success rate increases.



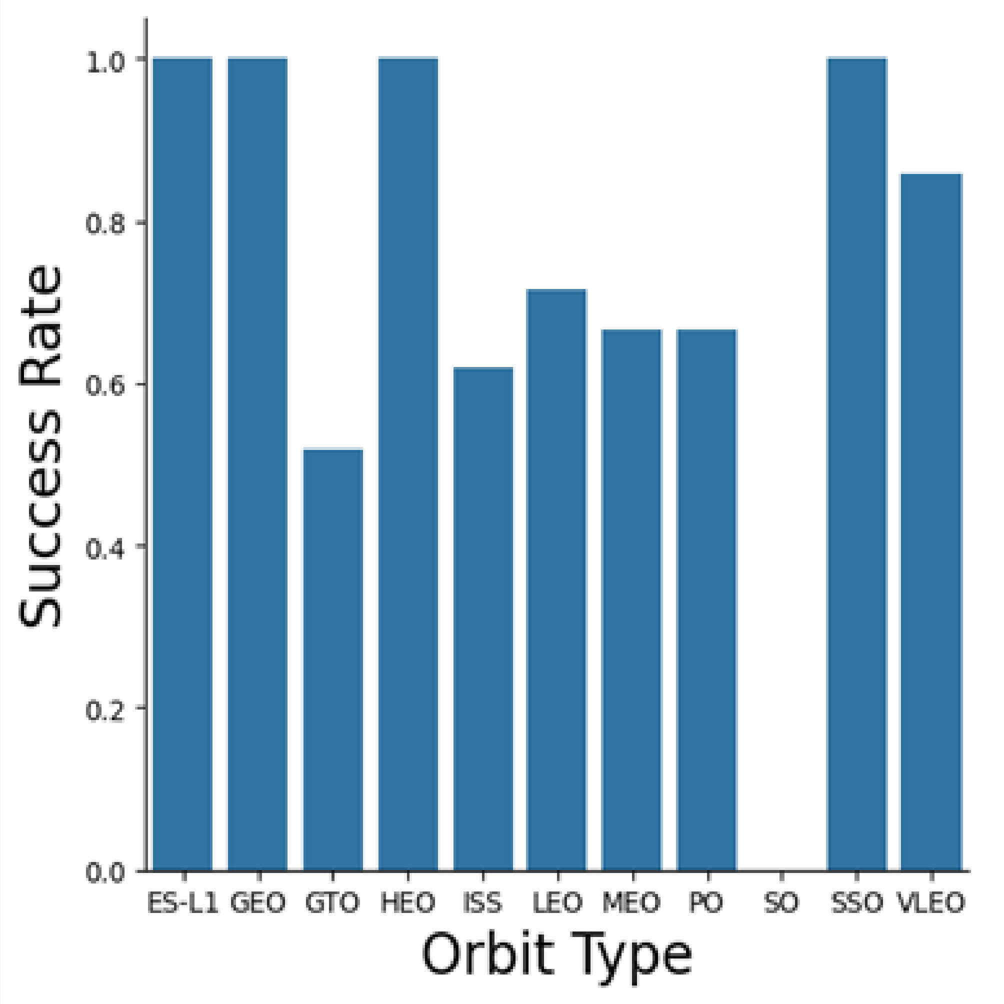
Payload vs. Launch Site

- The greater the payload mass for launch site CCAFS SLC 40, the higher the success rate for the rocket.



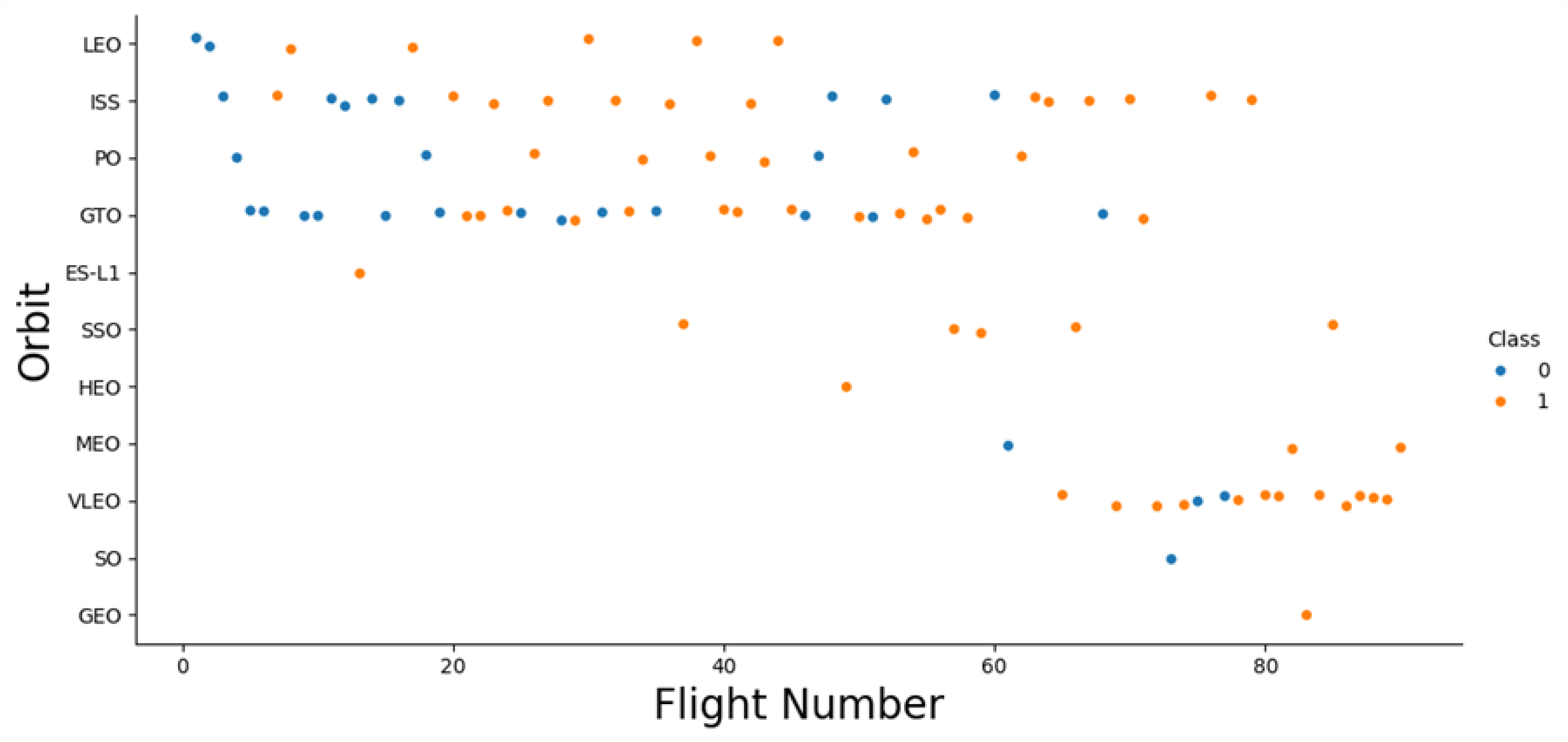
Success Rate vs. Orbit Type

- From the bar chart we can observe that the orbit types ES-L1, GEO, HEO and SSO have the highest success rate which is 1.0.
- Also, for the orbit type SO has 0.0 success rate.



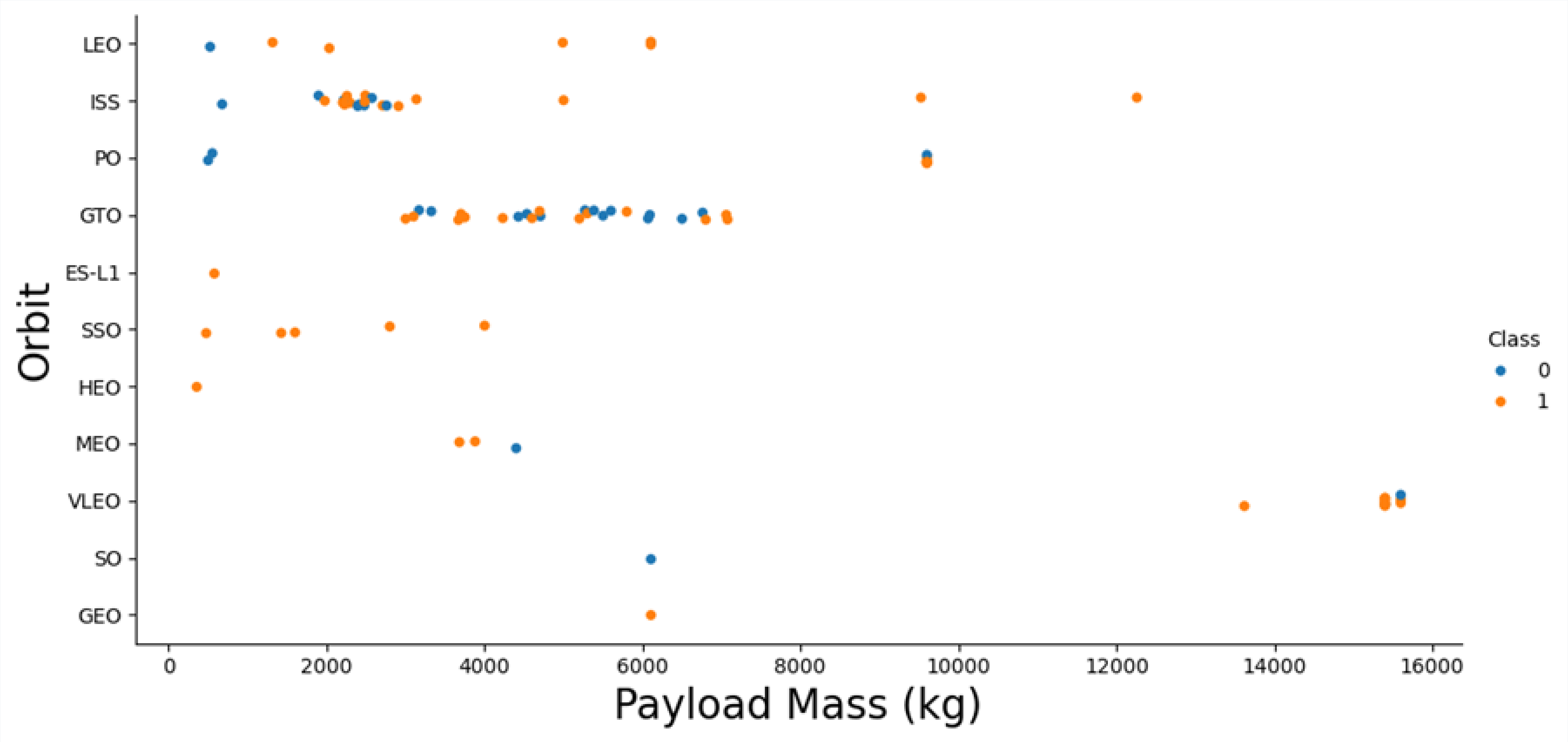
Flight Number vs. Orbit Type

- According to the figure below, the VLEO orbit has a higher success rate for higher flight numbers. And the orbit ISS shows high success rate for all flight numbers.

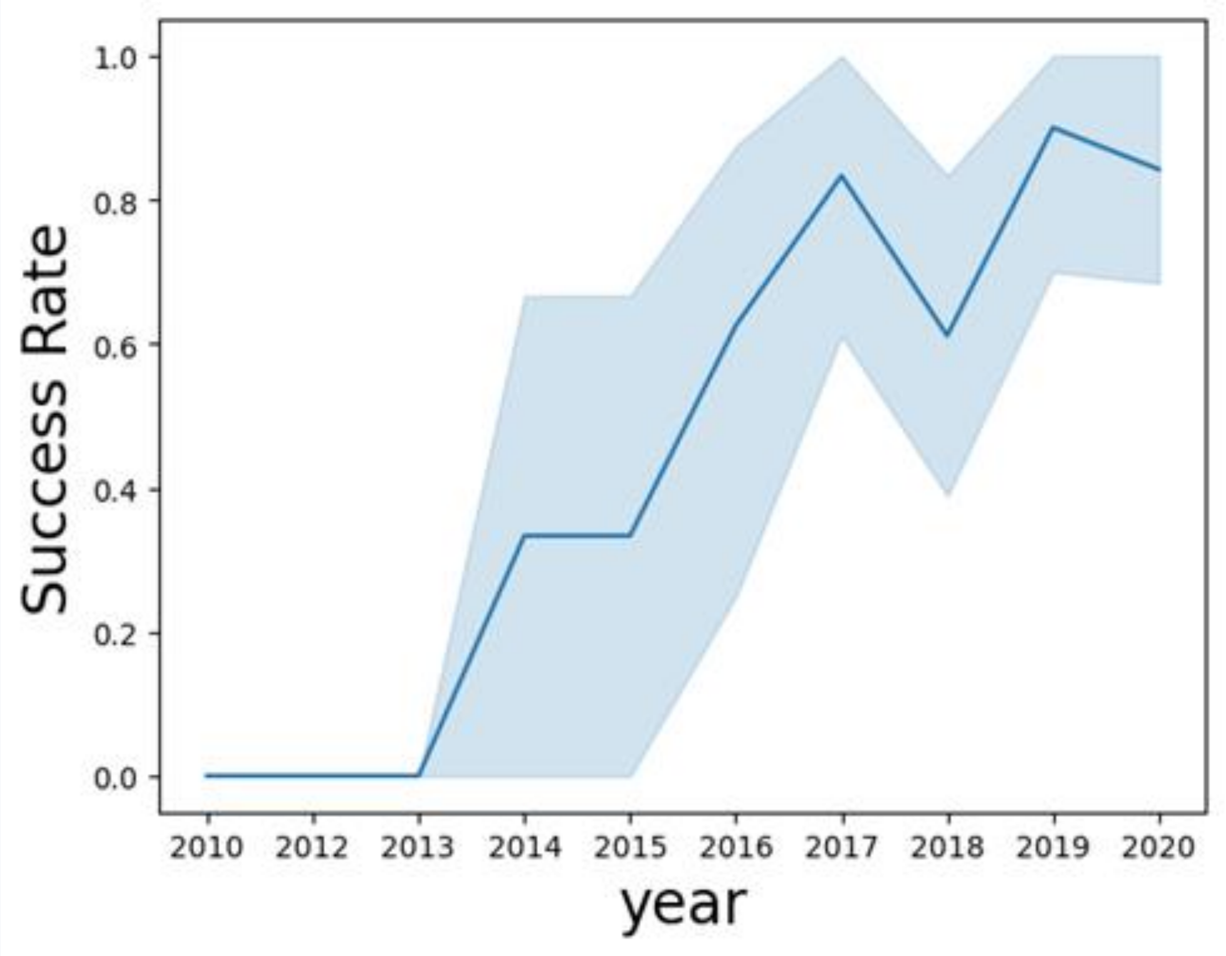


Payload vs. Orbit Type

- With heavy payloads, the launches are successful for the orbits VLEO and ISS.
- The successful launches can be observed in SSO orbit only for lower payloads.



Launch Success Yearly Trend



- As shown in the plot, the success rate was increasing from 2013 to 2020, with a slight decrement in 2018.

All Launch Site Names

- Using the key word **DISTINCT**, we can show the unique launch sites from the SpaceX data.

```
%sql select DISTINCT "Launch_Site" from SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTABLE 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
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parach
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 (parach
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No atte
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No atte
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No atte

- We used the **LIKE** key word to filter the launch site names begin with 'CCA', and the **LIMIT** key word to get first 5 results.

Total Payload Mass

- We calculated the total payload mass carried by boosters launched by NASA (CRS) as 48213 kg.

```
%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Customer" like 'NASA (CRS)%'
```

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

SUM(PAYLOAD_MASS__KG_)
48213

Average Payload Mass by F9 v1.1

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Booster_Version" LIKE 'F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

- The average payload mass carried by booster version F9 v1.1 is approximately 2534 kg.

First Successful Ground Landing Date

- We found that the date of the first successful landing outcome on ground pad was 22nd December, 2015.

```
%sql select MIN("Date") from SPACEXTABLE where "Landing_Outcome"='Success (ground pad)'
```

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

MIN("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Using **WHERE** clause, we filterd results for boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.

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

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

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- We used the **COUNT** and **GROUPBY** key words to calculate the number of successful and failure mission outcomes. According to the results there were 100 successful missions and only 1 failure.

```
%sql select "Mission_Outcome", count("Mission_Outcome") from SPACEXTABLE GROUP BY "Mission_Outcome"
```

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

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

Boosters Carried Maximum Payload

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

```
%sql SELECT BOOSTER_VERSION \
FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

- To get a list of the names of the boosters which have carried the maximum payload mass, we used the **MAX** key word.

2015 Launch Records

```
%sql SELECT CASE SUBSTR("Date", 6, 2) \
WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'March' \
WHEN '04' THEN 'April'WHEN '05' THEN 'May'WHEN '06' THEN 'June' \
WHEN '07' THEN 'July'WHEN '08' THEN 'August'WHEN '09' THEN 'September' \
WHEN '10' THEN 'October'WHEN '11' THEN 'November'WHEN '12' THEN 'December' \
ELSE 'Unknown'  END AS month, \
"Landing_Outcome"='Failure (drone ship)',"Booster_Version","Launch_Site"FROM SPACEXTABLE \
WHERE SUBSTR("Date", 0, 5) = '2015';
```

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

month	"Landing_Outcome" ='Failure (drone ship)'	Booster_Version	Launch_Site
January	1	F9 v1.1 B1012	CCAFS LC-40
February	0	F9 v1.1 B1013	CCAFS LC-40
March	0	F9 v1.1 B1014	CCAFS LC-40
April	1	F9 v1.1 B1015	CCAFS LC-40
April	0	F9 v1.1 B1016	CCAFS LC-40
June	0	F9 v1.1 B1018	CCAFS LC-40
December	0	F9 FT B1019	CCAFS LC-40

- According to the result obtained, we can see that the landing outcome was a failure in the months January and April.

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

```
%sql select "Landing_Outcome",count("Landing_Outcome") from SPACEXTABLE \
where "Date" between 20100604 and 20170320 \
group by "Landing_Outcome" order by 2 desc
```

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

Landing_Outcome	count("Landing_Outcome")
Success (drone ship)	12
No attempt	12
Success (ground pad)	8
Failure (drone ship)	5
Controlled (ocean)	4
Uncontrolled (ocean)	2
Precluded (drone ship)	1

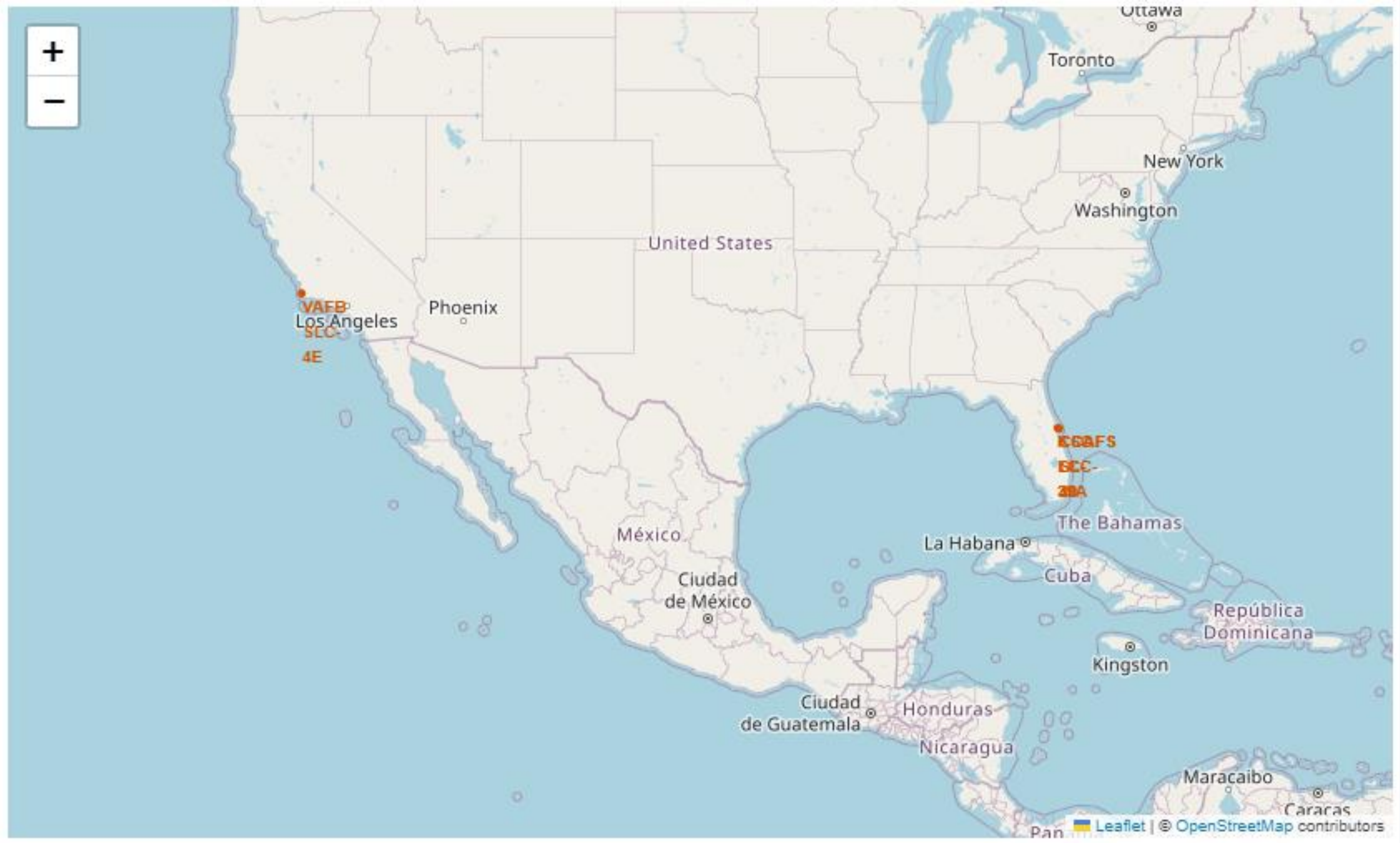
- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.

- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

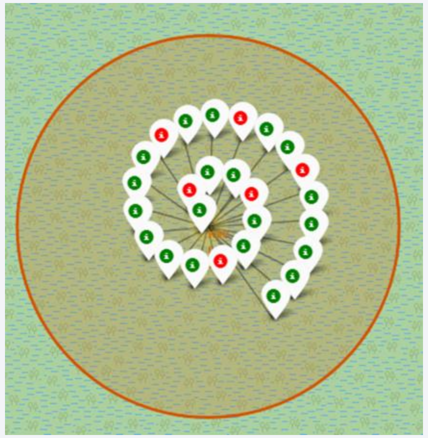
Section 3

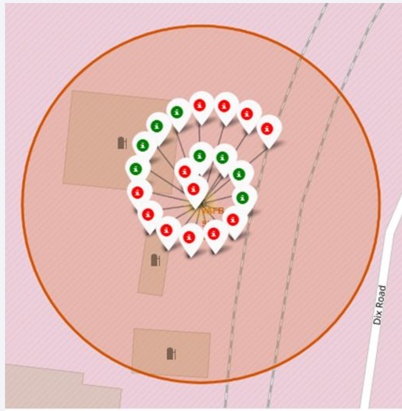
Launch Sites Proximities Analysis

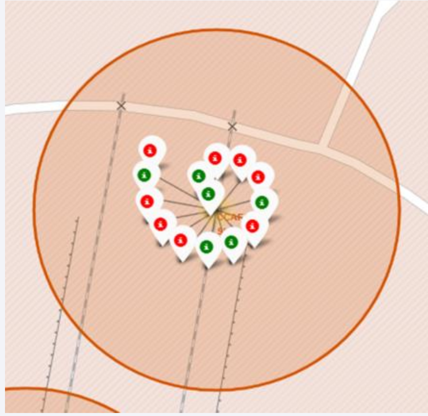
Locations for all launch sites



- As shown in the map, SpaceX launch sites are located in coastal areas like Florida and California.



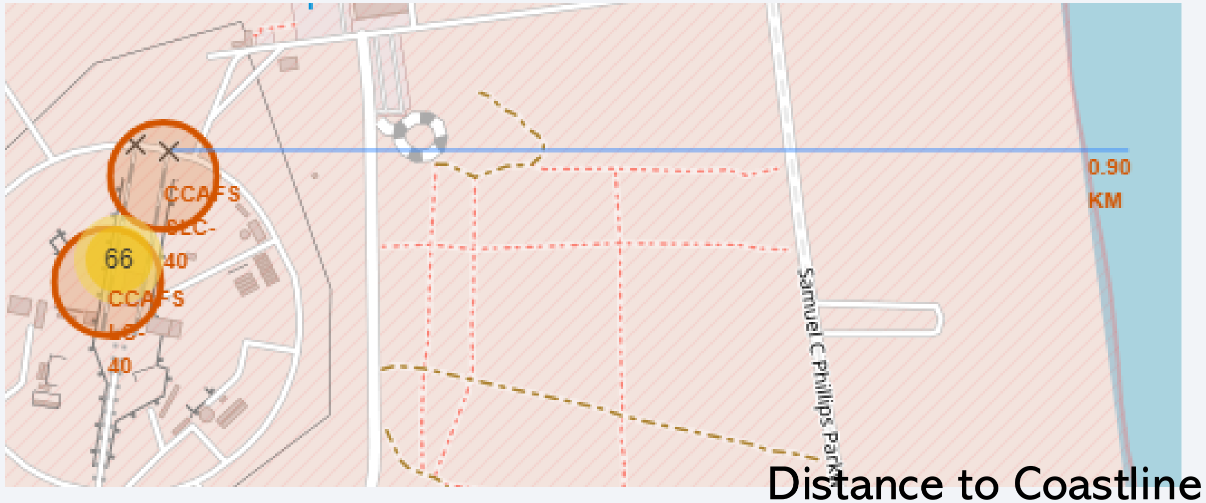




Color-labeled launch outcomes

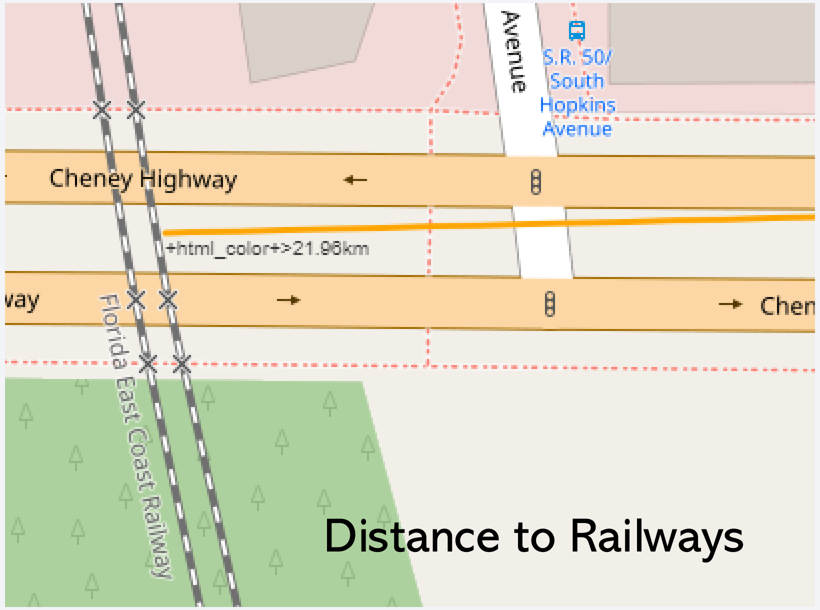
- Green markers show the successful launches while red shows failures.

Launch site distances to landmarks



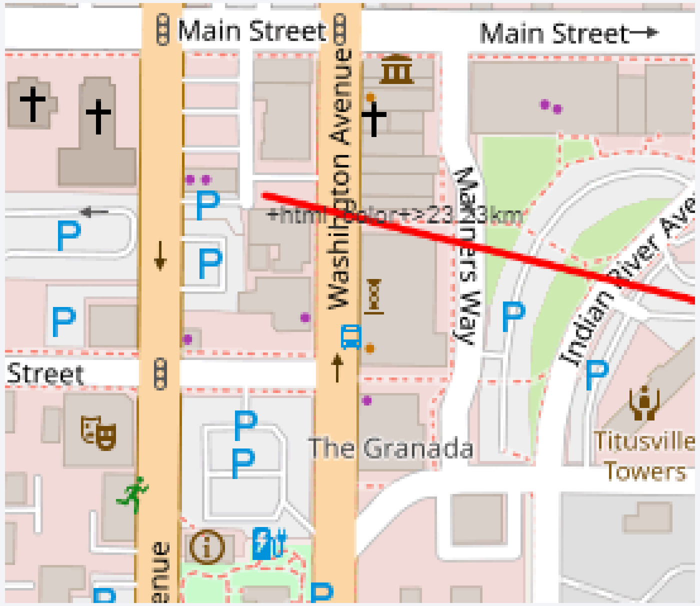

```
City Distance 23.234752126023245
Railway Distance 21.961465676043673
Highway Distance 26.88038569681492
Coastline Distance 0.8627671182499878
```

Distance to Coastline



Distance to Railways

Distance to Highways



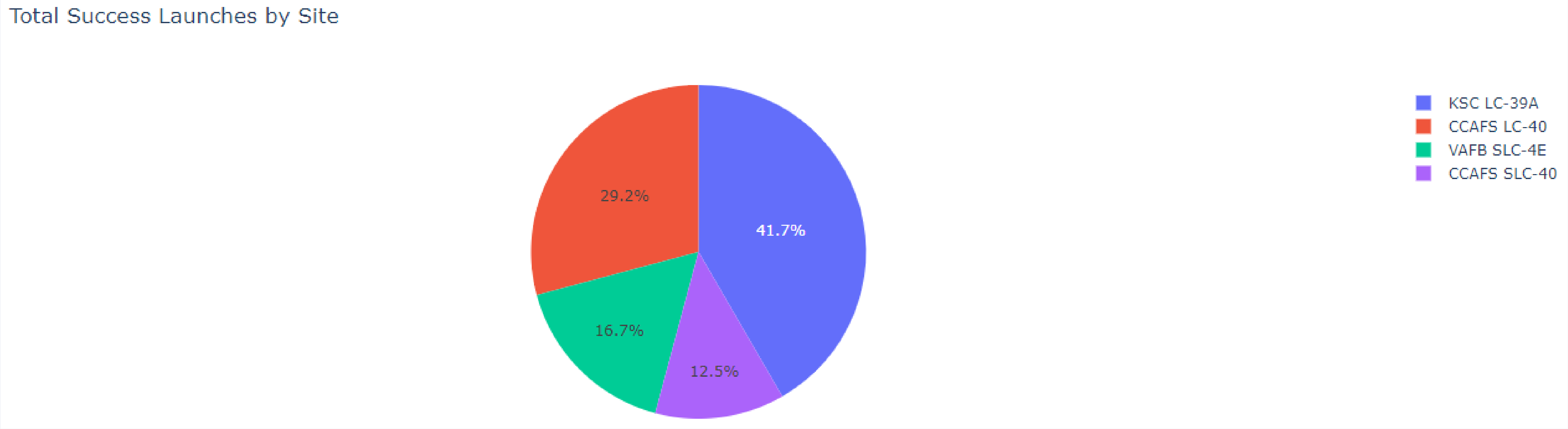
Distance to City

Section 4

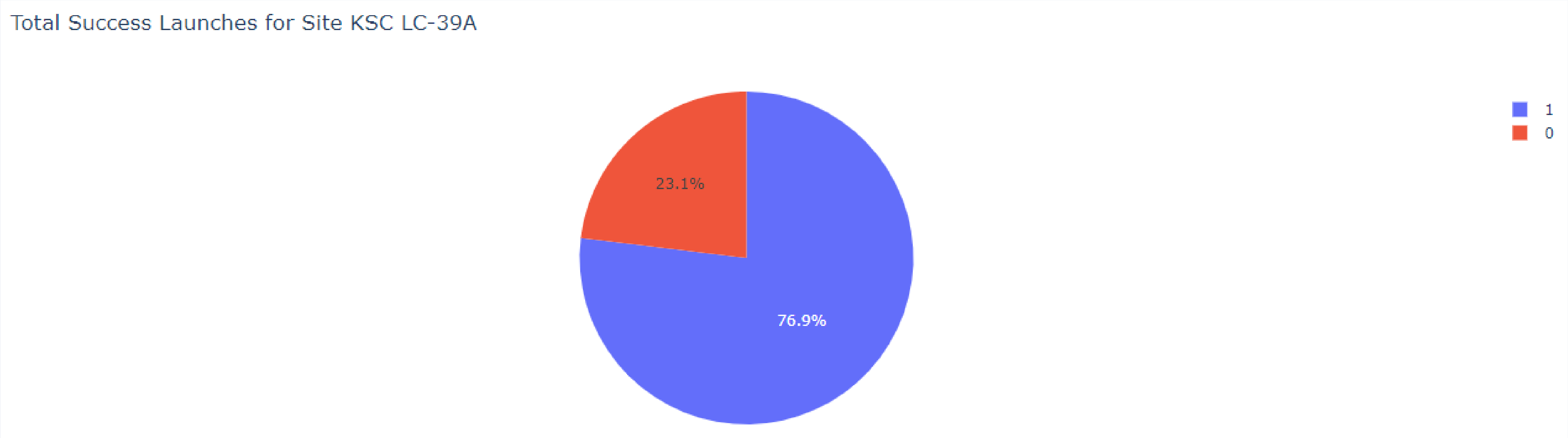
Build a Dashboard with Plotly Dash

Pie Chart for Success Percentages by all sites

- According to the pie chart, KSC LC-39A had the largest number of successful launches among all sites and it was 41.7%.



Pie Chart for the launch site with highest success ratio



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

Scatter plots for Payload vs. Launch Outcome

- The first figure shows for the site CCAFS LC-40 for the lower payload and FT booster version had the highest success count.
- The second plot indicates for the site VAFB SLC-4E for the higher payload. But one launch from the version B4 had succeeded.

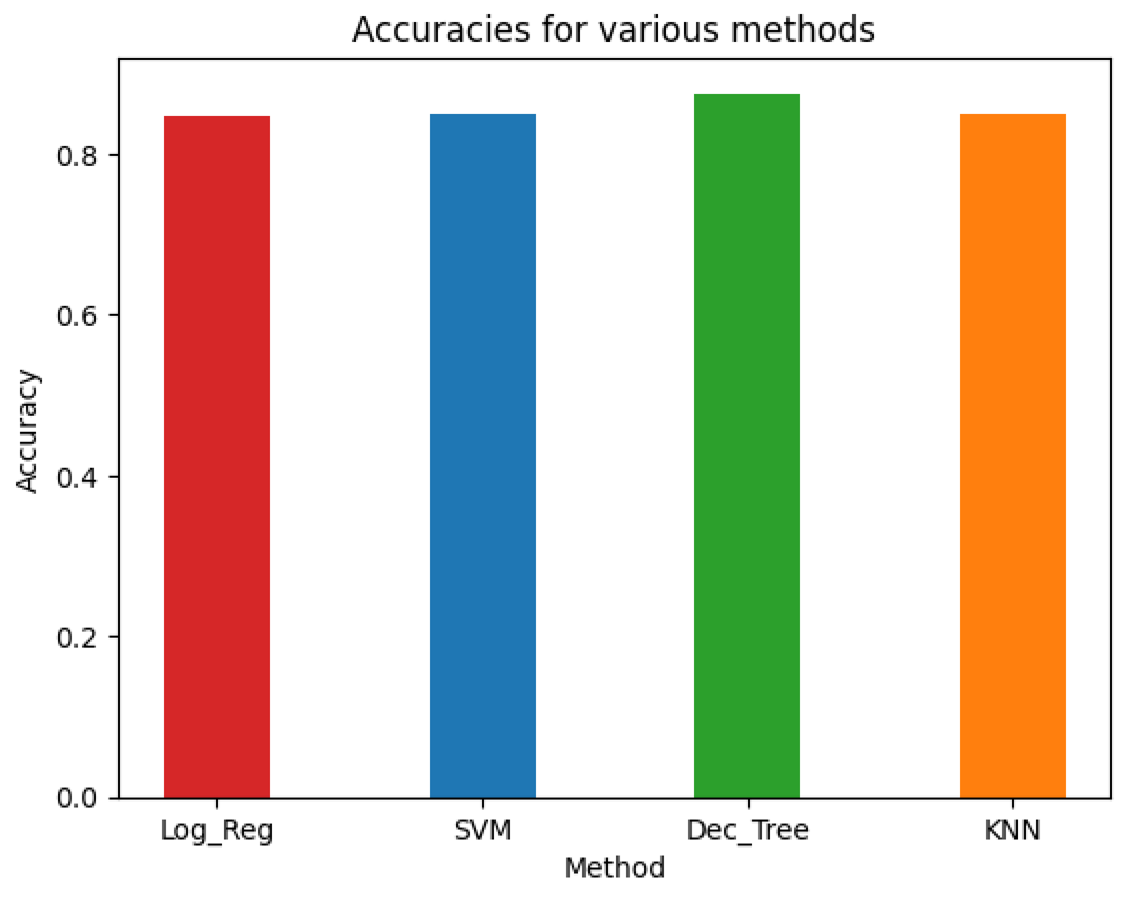
Section 5

Predictive Analysis (Classification)

Classification Accuracy

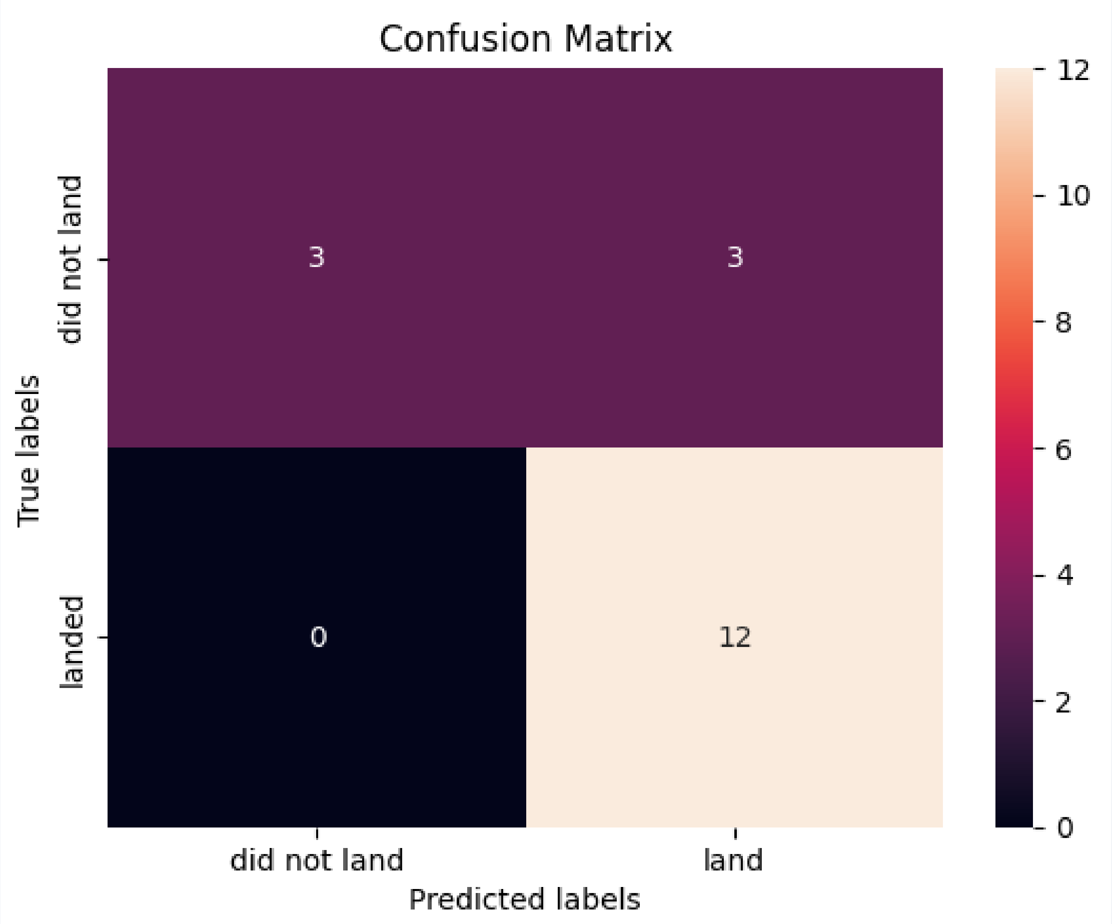
- The decision tree classifier is the model with the highest classification accuracy.

	ML Method	Accuracy
0	Logistic Regression	0.846429
1	Support Vector Machine	0.848214
2	Decision Tree	0.875000
3	K Nearest Neighbour	0.848214



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.



Conclusions

We can conclude that :

- The larger the flight number at a launch site, the greater the success rate at a launch site.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- Launch success rate started to increase from 2013 to 2020.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!