IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Samorae Campbell
June 2, 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection and Wrangling
  - Exploratory Analysis using SQL, Pandas and Matplotlib
  - Interactive Visual Analysis and Dashboards with Folium and Plotly Dash
  - Predictive Analysis using Classification Models
- Summary of all results
  - Exploratory Analysis Results
  - Results of the Interactive Visual Analysis and Dashboards
  - Predictive Analysis Results

Introduction

- Project background and context
  - SpaceX currently advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars, while other providers cost upward of 165 million dollars each. Much of the savings is because SpaceX can reuse the first stage. Therefore if it can determined whether the first stage will land, the cost of a launch can also be determined. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. The project aims to develop a classification machine learning model that can predict whether the first stage of a Falcon 9 will land successfully.
- Problems we want to find answers
  - Are there any factors that influence a successful landing?
  - What are the relationships between particular rocket features, and do they have any impact on the success of a landing?
  - What are the variables needed to ensure the best successful landing rates?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The SpaceX data was collected by making requests to the SpaceX API, as well as by Web Scrapping the *List of Falcon 9 and Falcon Heavy launches* Wikipage updated on 9th June 2021.
- Perform data wrangling
  - The data was processed by one-hot encoding data fields and identifying and removing any irrelevant columns.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Here, we build, tune, evaluate different classification models on our dataset.

Data Collection

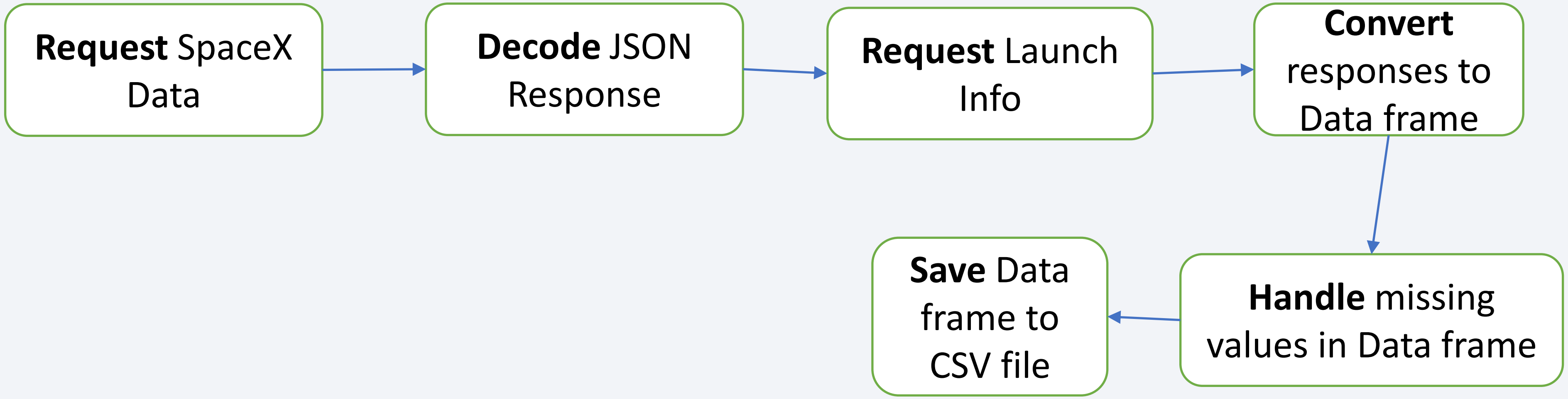
The datasets were collected using two methods:

- SpaceX REST API
  - We request and parse the SpaceX launch data using a GET request to a static Json URL, convert the data to a pandas data frame, process the data and then save the finished product as a CSV file.
- Web Scrapping the data from a known Wikipedia Webpage
  - We request the launch page from it s URL, then use the HTML elements to extract out data frame headers and content, process the data frame and then save the finished product as a CSV file.

Data Collection

SpaceX REST API

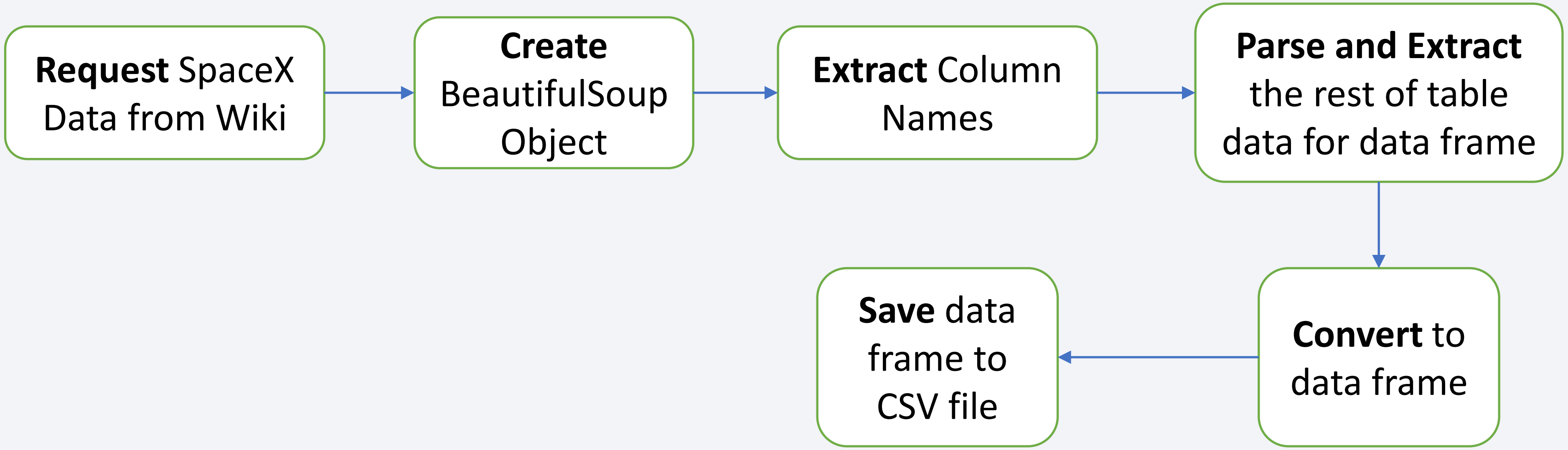
- We request and parse the SpaceX launch data using a GET request to a static Json URL, convert the data to a pandas data frame, process the data and then save the finished product as a CSV file.



Data Collection

Web Scrapping the data from a known Wikipedia Webpage

- We request the launch page from it s URL, then use the HTML elements to extract out data frame headers and content, process the data frame and then save the finished product as a CSV file.



Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts
- Steps
  - **Request** and **parse** the SpaceX launch data using the GET request
  - **Decode** the response using .json() and convert the response to a data frame using .json_normalize()
  - **Request** launch information from the SpaceX API using custom functions
  - **Create** a dictionary from the data
  - **Create** a data frame from the dictionary
  - **Filter** the data frame to only contain Falcon 9 launches
  - **Deal** with **missing values** of the Payload Mass by replacing them with the mean of the column
  - **Export** the resulting data frame to a CSV file

- [GitHub URL]()

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

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

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

```
# Calculate the mean value of PayloadMass column
df9_mean = data_falcon9['PayloadMass'].mean()
print(df9_mean)

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].fillna(df9_mean)
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
    - **Request** the Falcon9 Launch data from Wikipedia
    - **Create** BeautifulSoup object from HTML response
    - **Extract** the column names from the HTML table header
    - **Collect** the data frame data by parsing the HTML tables
    - **Create** dictionary from the data
    - **Create** data frame from the dictionary
    - **Export** the resulting data frame to a CSV file
- [GitHub URL]()

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```

```
column_names = []

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

print(column_names)
```

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

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

Data Wrangling

- Process Description
  - The data was processed by performing Exploratory Data Analysis (EDA) to find some patterns in the data and determine the best data features for training supervised models. In the data set, there are several different cases where the booster did not land successfully. We wrangle the data by identifying and labelling all cases that lead to a successful outcome, and those that lead to a failed outcome.

Load launch data → **Calculate** # of launches per site → **Calculate** # and occurrence of each orbit → **Calculate** # and occurrence of outcomes of the orbits → **Create** landing outcome label from Outcome column → **Populate** landing outcome from independent columns → **Save** data frame to CSV file.

- GitHub URL

Data Wrangling

- Steps
  - Perform an exploratory data analysis to determine the data labels
  - Examples of the EDA include calculations and visualizations such as:
    - # of launches for each site
    - # an occurrence of orbit
    - # and occurrence of mission outcome per orbit type
  - Create binary landing outcomes column as the target variable, and populate with values determined from the independent variables using one hot encoding.
  - Export the resulting data frame to a CSV file.

- [GitHub URL]

EDA with Data Visualization

A summary of the charts that were plotted are as follows:

- Scatter Charts – used to visualize the relationships between two numerical variables. The relationships identified could be used to aid the machine learning process.
- Bar Graphs – used to show comparisons of counts of discrete values of a categorical variable.
- Line Graphs – used to create visualization of how values of a variable change over time. This is useful for identifying additional trends and patterns in time series.

- GitHub URL

EDA with SQL

- A summary of SQL queries performed are as follows:
  - 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 succesful landing outcome in ground pad was acheived.
  - 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 landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

- GitHub URL

Build an Interactive Map with Folium

- Below, we summarize the map objects that were created and added to the folium map:
  - We added circles on the map to indicate and visualize the position of each launch site.
  - Markers were added to the map to visualize the launch outcomes for each site on the map. That way we can see which site had very high landing success rates and which do not.
  - After calculating the distance, lines were added to the map to visualize the closest distance of importance features (e.g., highways, rails, cities) from each launch site.

- GitHub URL

Build a Dashboard with Plotly Dash

- Below, we summarize the plots/graphs and interactions that were added to the dashboard:
  - **Pie Chart** showing the total launches by all launch sites, as well the total successful/unsuccessful launches for a specific site. These charts helps to organize and show the different values of a feature as percentages of a whole.
  - **Scatter Graph** showing the relationship between the success outcome and Payload Mass (/Kg) for the different rocket booster versions. Scatter graphs help us to observe and visualize the relationships between the two numeric features. We were able to observe the graph readings and determine in there were any correlations between the success outcome and the selected booster version.

- GitHub URL

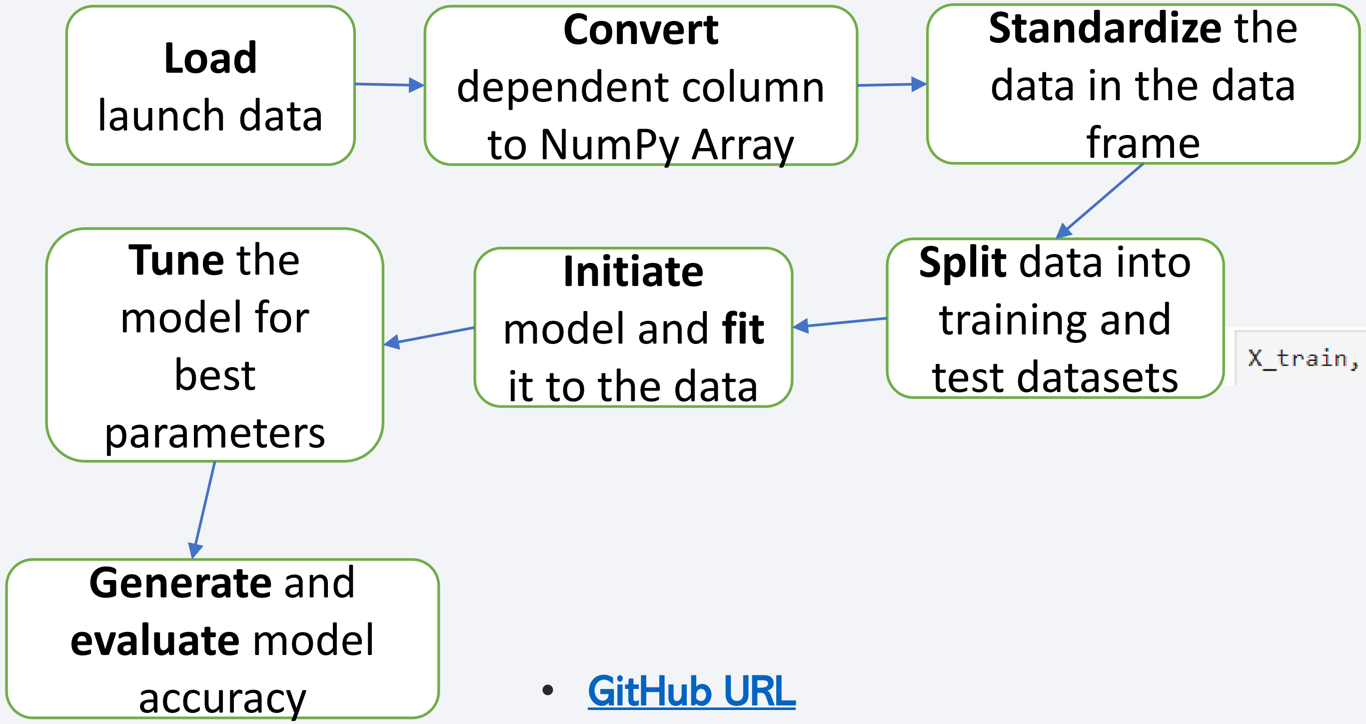
Predictive Analysis (Classification)

- Summary of how the best performing built, evaluated, improved, and found the best performing classification model
  - We built our classification models by doing the following:
    - Loading and transforming our data using Pandas and NumPy
    - Splitting the data into training and testing data sets
    - Set out parameters and algorithms to GridSearchCV, and then fitting out classification models to GridSearchCV using the training dataset.
  - We evaluated our classification models by measuring the accuracy score of each model.
  - We improved our classification models though Feature Engineering and tuning the hyperparameter for each model.
  - We found the best classification model by identifying the model with the highest accuracy score.

- GitHub URL

Predictive Analysis (Classification)

Predictive Analysis Flowchart




```
URL1 = "https://cf-courses-data.s3.us.cloud-object-storage
resp1 = await fetch(URL1)
text1 = io.BytesIO((await resp1.arrayBuffer()).to_py())
data = pd.read_csv(text1)
```

```
Y = data["Class"].to_numpy()
```

```
# students get this
transform = preprocessing.StandardScaler()
X = transform.fit_transform(X)
X
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

```
tree = DecisionTreeClassifier()
```

```
# Instantiate the GridSearchCV object: svm_cv
tree_cv = GridSearchCV(tree, parameters, cv=10)

# Fit it to the data
tree_cv.fit(X_train, Y_train)
```

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

- [GitHub URL]()

Results

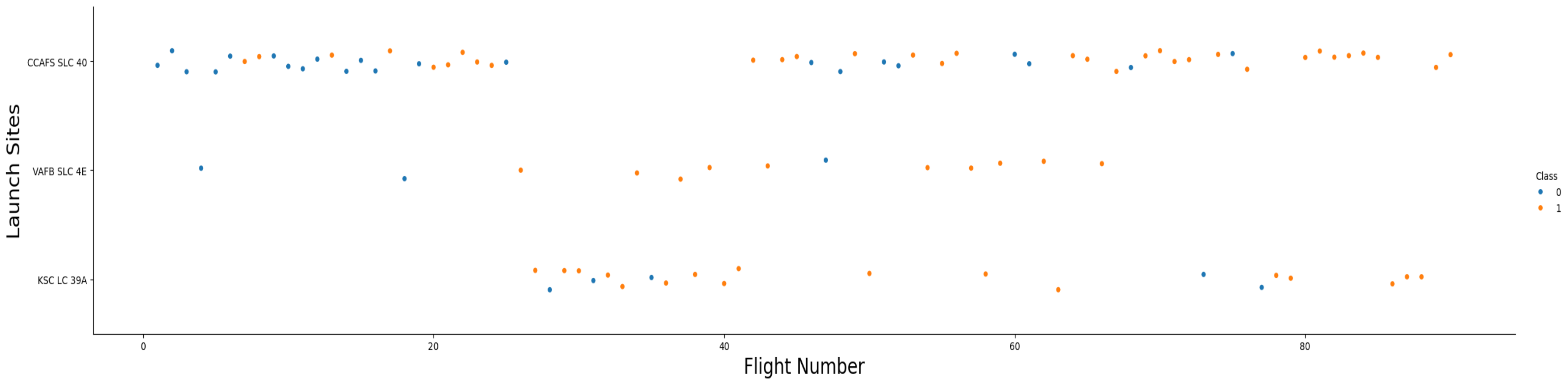
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

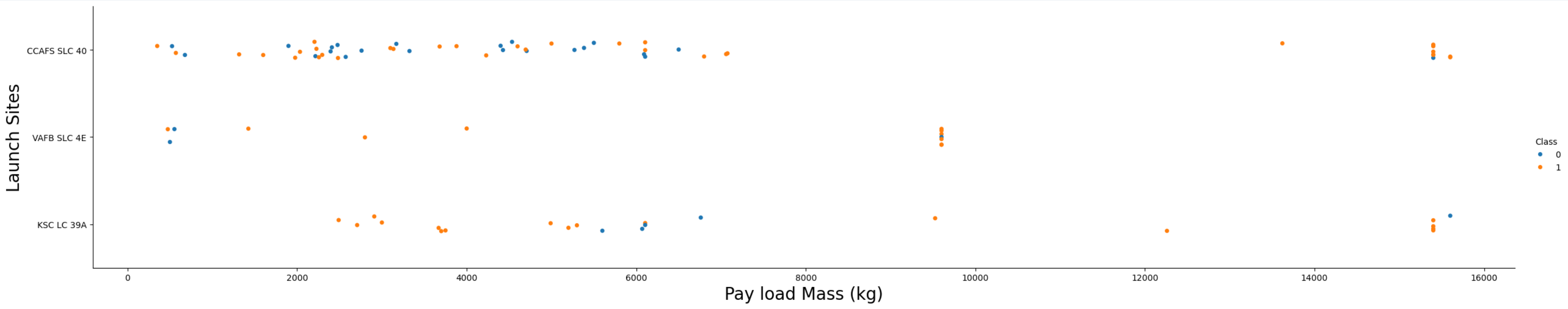
We see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.

Flight Number vs. Launch Site



- We see that different launch sites have different success rates. Generally, as the flight number increases, the chances of a successful landing of the first stage also increases.

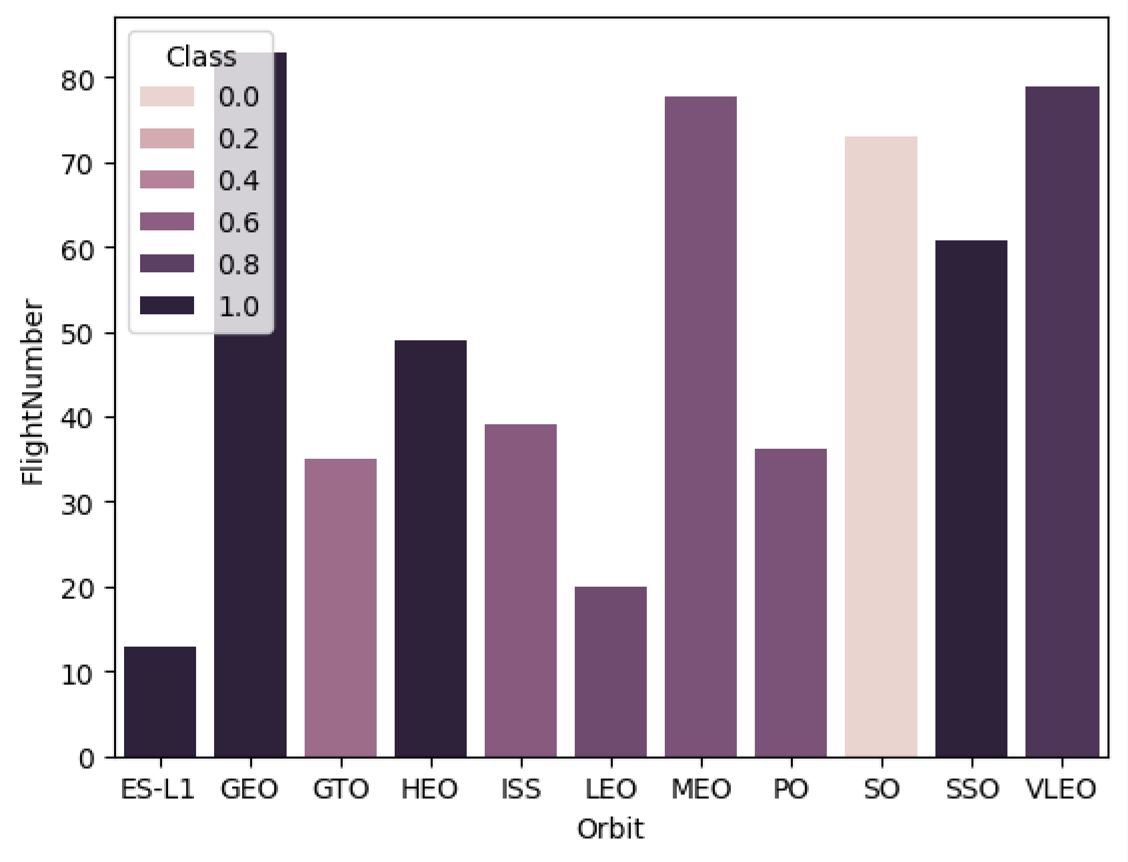
Payload vs. Launch Site



From the graph, it is hard to tell whether Pay load Mass has any impact on the success of a first stage landing.

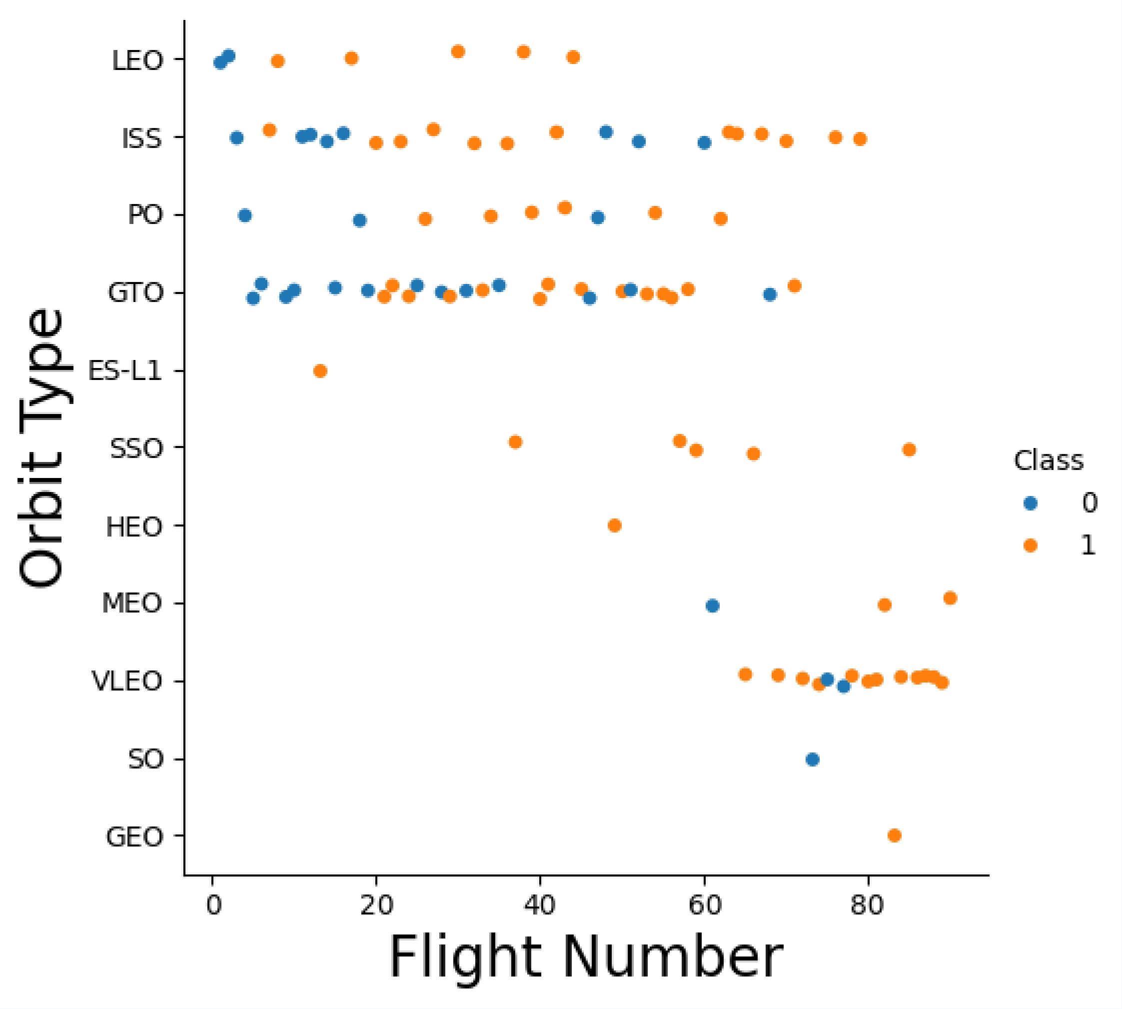
Success Rate vs. Orbit Type

From the graph of orbits, we see that ES-L1, GEO, HEO and SSO orbits have the highest success rates. However, ES-L1 has less than a third of the flight numbers needed to achieve the same success rate as the other three orbits.



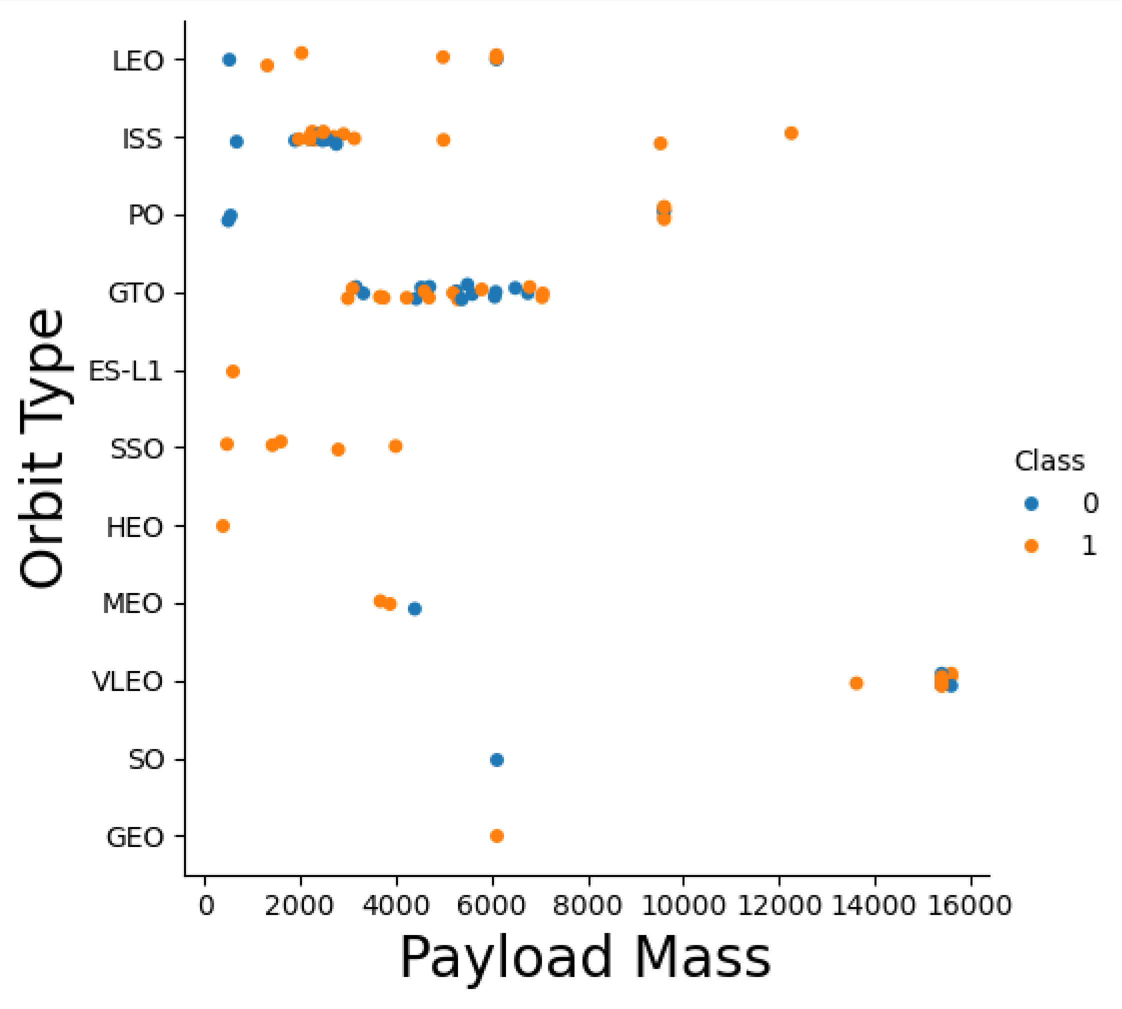
Flight Number vs. Orbit Type

From the graph of orbits, we see that 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.



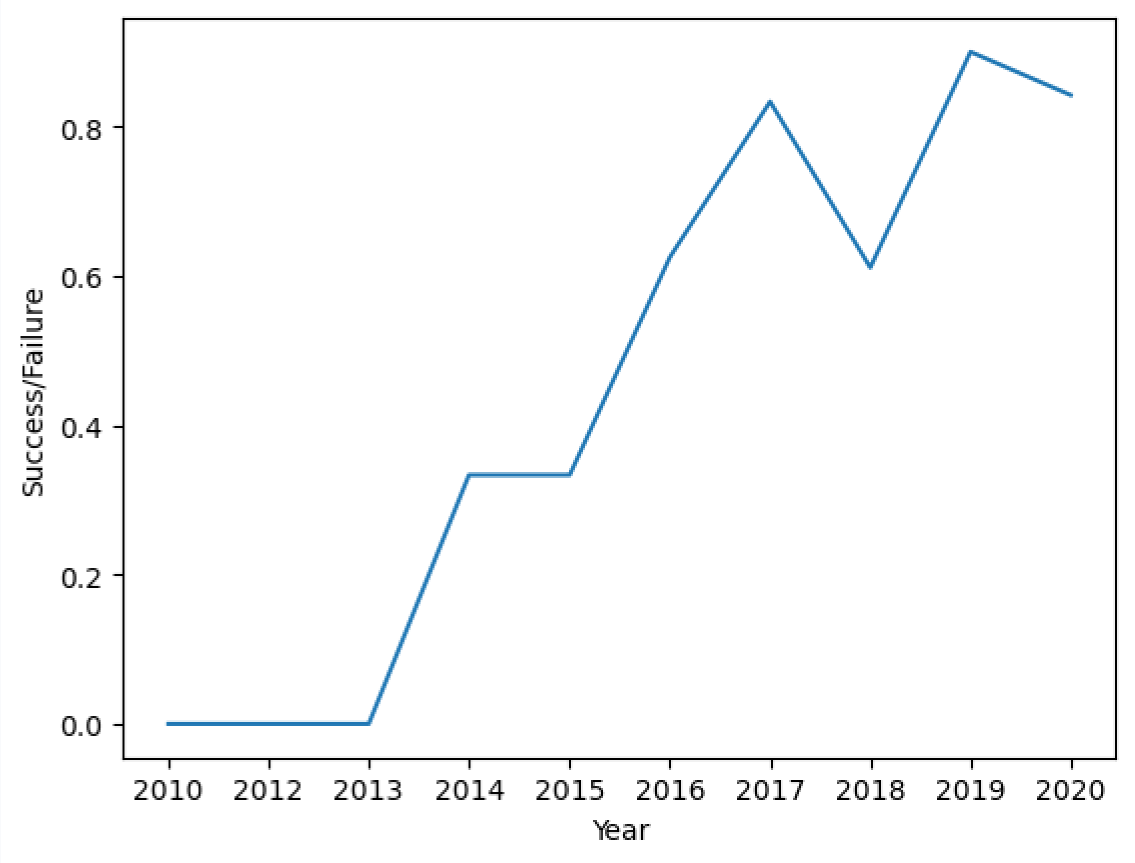
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



Launch Success Yearly Trend

Here we observe that the success rate since 2013 kept increasing until 2020.



All Launch Site Names

In this SQL query, we find the names of the unique launch sites with the help of the "DISTINCT" keyword.

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

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

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

Launch Site Names Begin with 'CCA'

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

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_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

In this SQL query, we find 5 records where launch sites begin with `CCA` with the help of the "LIKE" keyword, and then limiting the output to the first 5 rows.

Total Payload Mass

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

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

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

Total_Payload_Mass
45596

In this SQL query, we calculate the total payload carried by boosters from NASA by using the SUM operator on the rows of the output table which is filtered by the ‘Customer’ column.

Average Payload Mass by F9 v1.1

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

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

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

Total_Payload_Mass
2928.4

In this SQL query, we calculate the average payload mass carried by booster version F9 v1.1 by using the AVG operator on the rows of the output table which is filtered by the 'Booster_Version' column.

First Successful Ground Landing Date

```
[23]: %sql SELECT MIN(DATE) AS 'Minimum_Date' FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';

 * sqlite:///my_data1.db

Done.

[23]: ,,,,,,,,,,,
```

Minimum_Date
2015-12-22

In this SQL query, we find the dates of the first successful landing outcome on ground pad by using the MIN operator on the date columns for the rows of the output table which is filtered by the 'Landing_Outcome' column.

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

```
%sql SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_>4000 AND PAYLOAD_MASS__KG_<6000;
```

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

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

In this SQL query, we list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 by using the DISTINCT feature on the output table which is filtered by the 'Landing_Outcome' column as well as the Payload Mass column.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT(SELECT COUNT(Mission_Outcome) FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Success%') AS Successful_Mission_Outcomes,
(SELECT COUNT(Mission_Outcome) FROM SPACEXTBL WHERE Mission_Outcome LIKE '%Failure%') as Failed_Mission_Outcomes;
```

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

Successful_Mission_Outcomes	Failed_Mission_Outcomes
100	1

In this SQL query, we calculate the total number of successful and failure mission outcome by using the COUNT operator on the output table which is a subquery of another COUNT operation, filtered by the ‘Mission_Outcome” column.

Boosters Carried Maximum Payload

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

```
%sql SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL;
```

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

MAX(PAYLOAD_MASS__KG_)
15600

```
%sql SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

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

In this SQL query, we list the names of the booster which have carried the maximum payload mass by using the MAX operator on the output table which is a subquery of another MAX operation, filtered by the Payload Mass column.

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.

```
%sql SELECT substr(Date, 6,2),Mission_Outcome,Booster_Version,Launch_Site FROM SPACEXTBL WHERE substr(Date, 0,5)='2015' AND Landing_Outcome = 'Failure (drone ship)';
```

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

substr(Date, 6,2)	Mission_Outcome	Booster_Version	Launch_Site
01	Success	F9 v1.1 B1012	CCAFS LC-40
04	Success	F9 v1.1 B1015	CCAFS LC-40

In this SQL query, we list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 by using the substr operator on the output table which is filtered by the Date column as well as the Landing Outcome column.

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.

```
%sql SELECT Landing_Outcome, COUNT(Landing_Outcome) AS Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY Count DESC;
```

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

Landing_Outcome	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

In this SQL query, we 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 by using the COUNT operator on the output table which is filtered by the Date column and grouped by the Landing Outcome column.

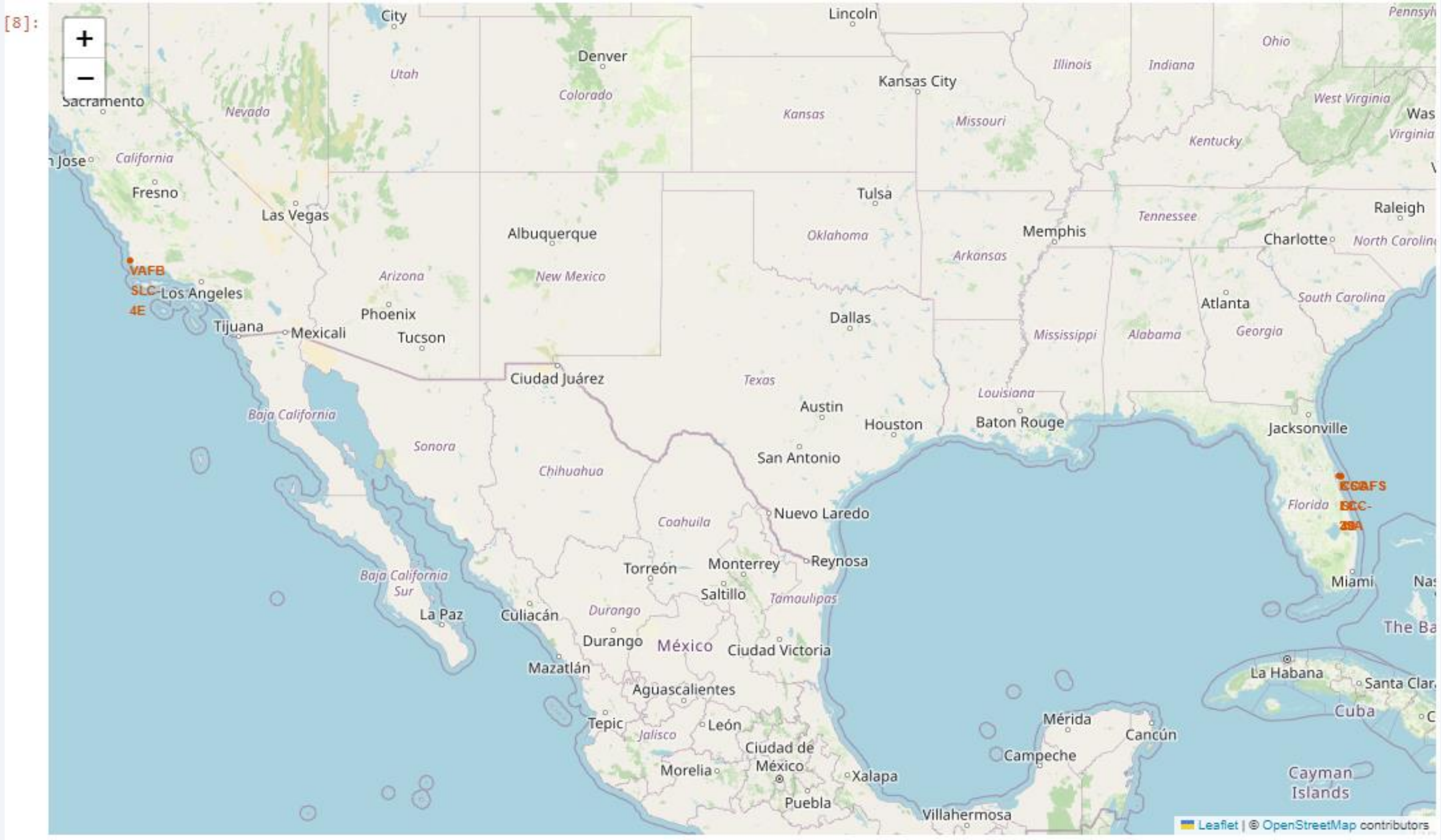
Section 3

Launch Sites Proximities Analysis

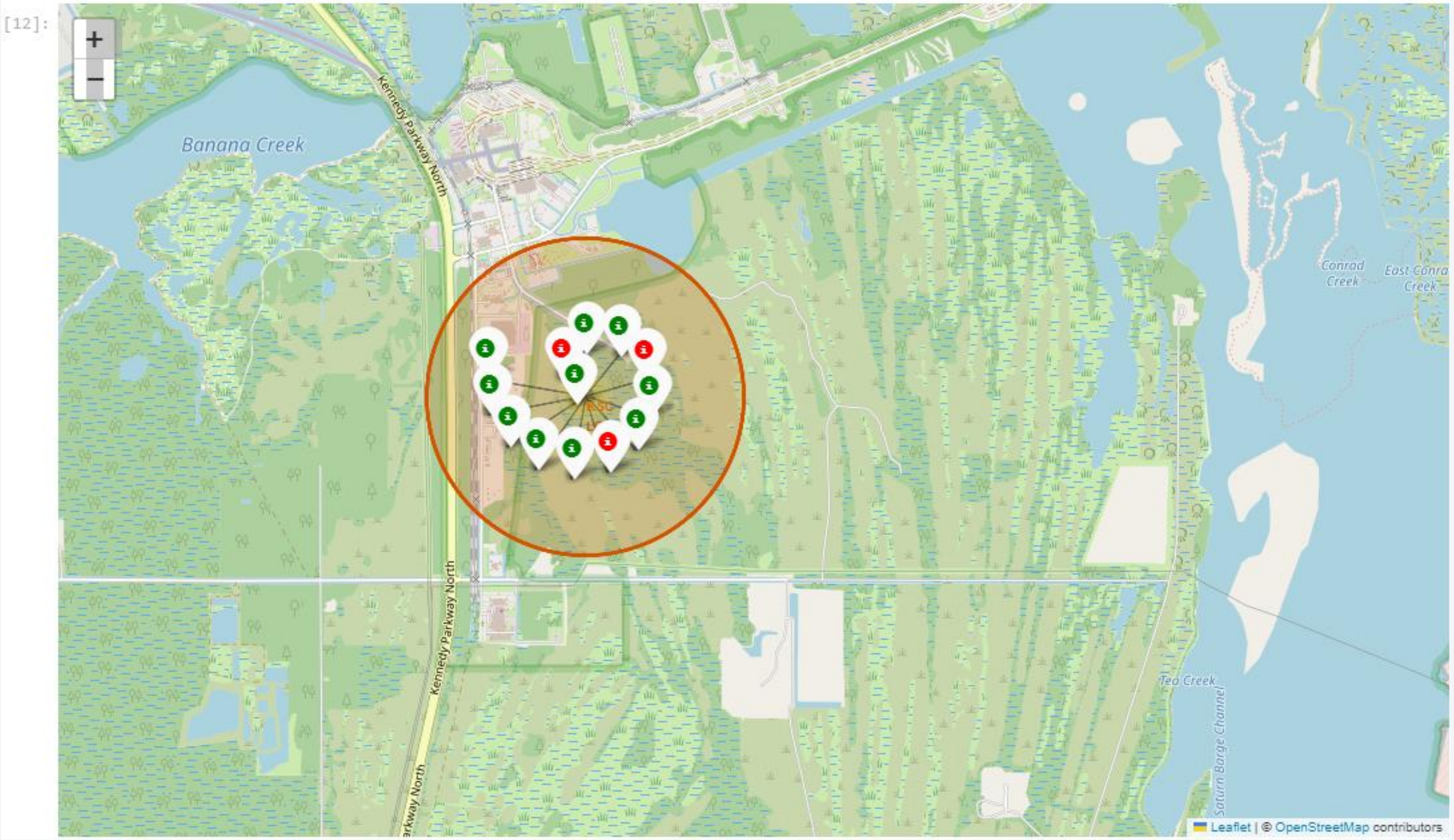
Global Map of All Launch Sites

The markers clearly identify the locations of the launch sites on the map. From the map, we can also see that:

- All launch sites in proximity to the Equator line.
- All launch sites in very close proximity to the coast.



Map Showing the Launch Outcomes for Site

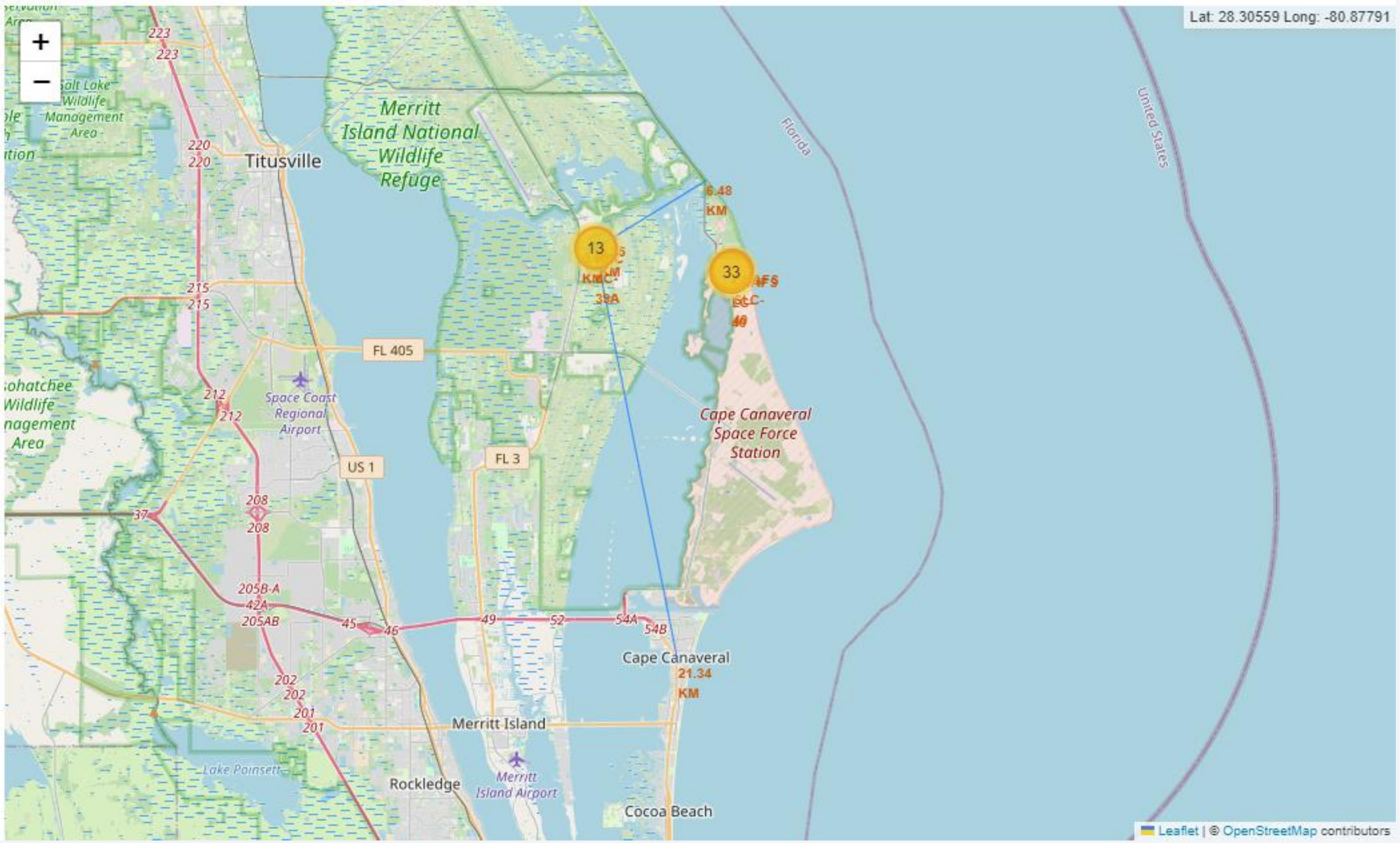


From the color-labeled markers in marker cluster for this site (**KSC-LC-39A**), we can easily identify that this launch site has a relatively high success rate, as there are only 3 red markers (unsuccessful landings) for 10 green markers (successful landings).

Distances of Important Elements from Launch Site

Based on the distance lines and proximities, we are able to answer the following questions:

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



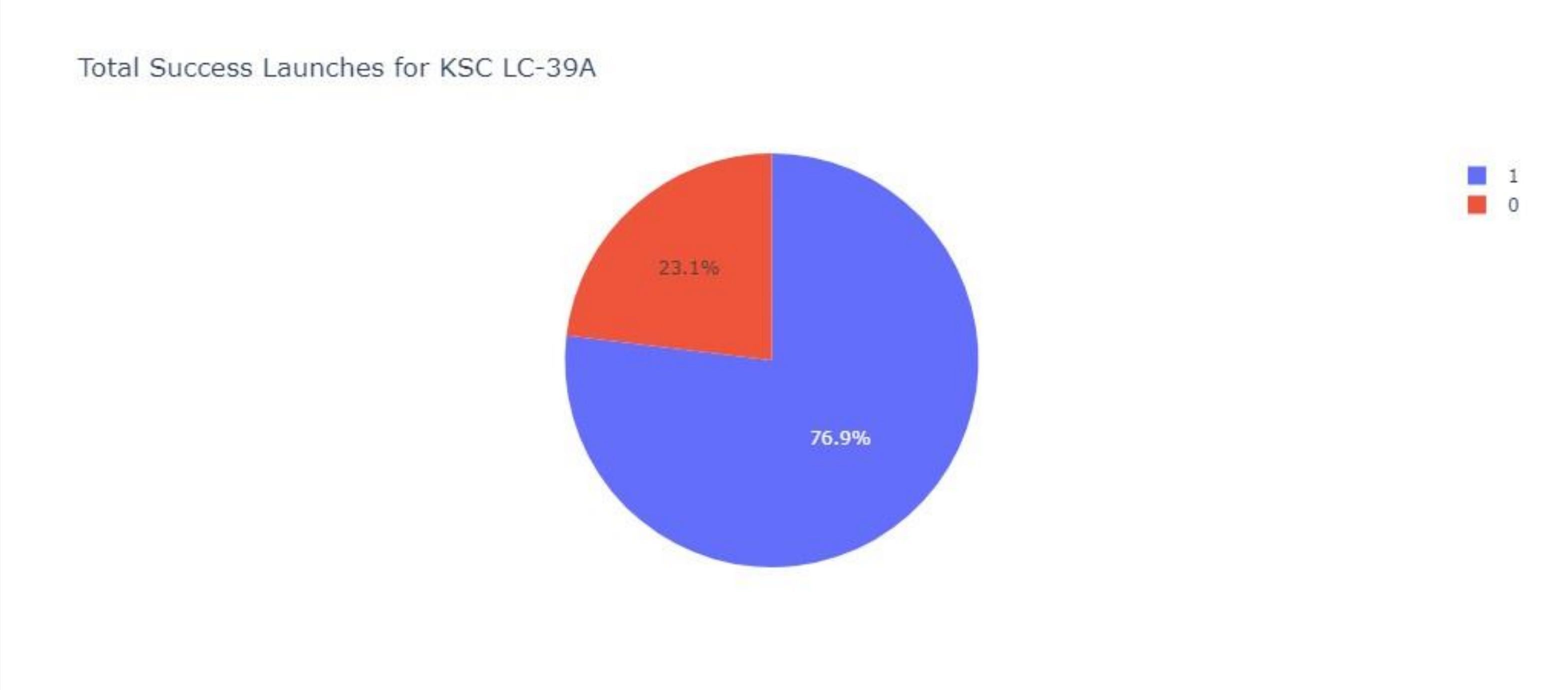
Section 4

Build a Dashboard with Plotly Dash

Launch Success Count for All Sites

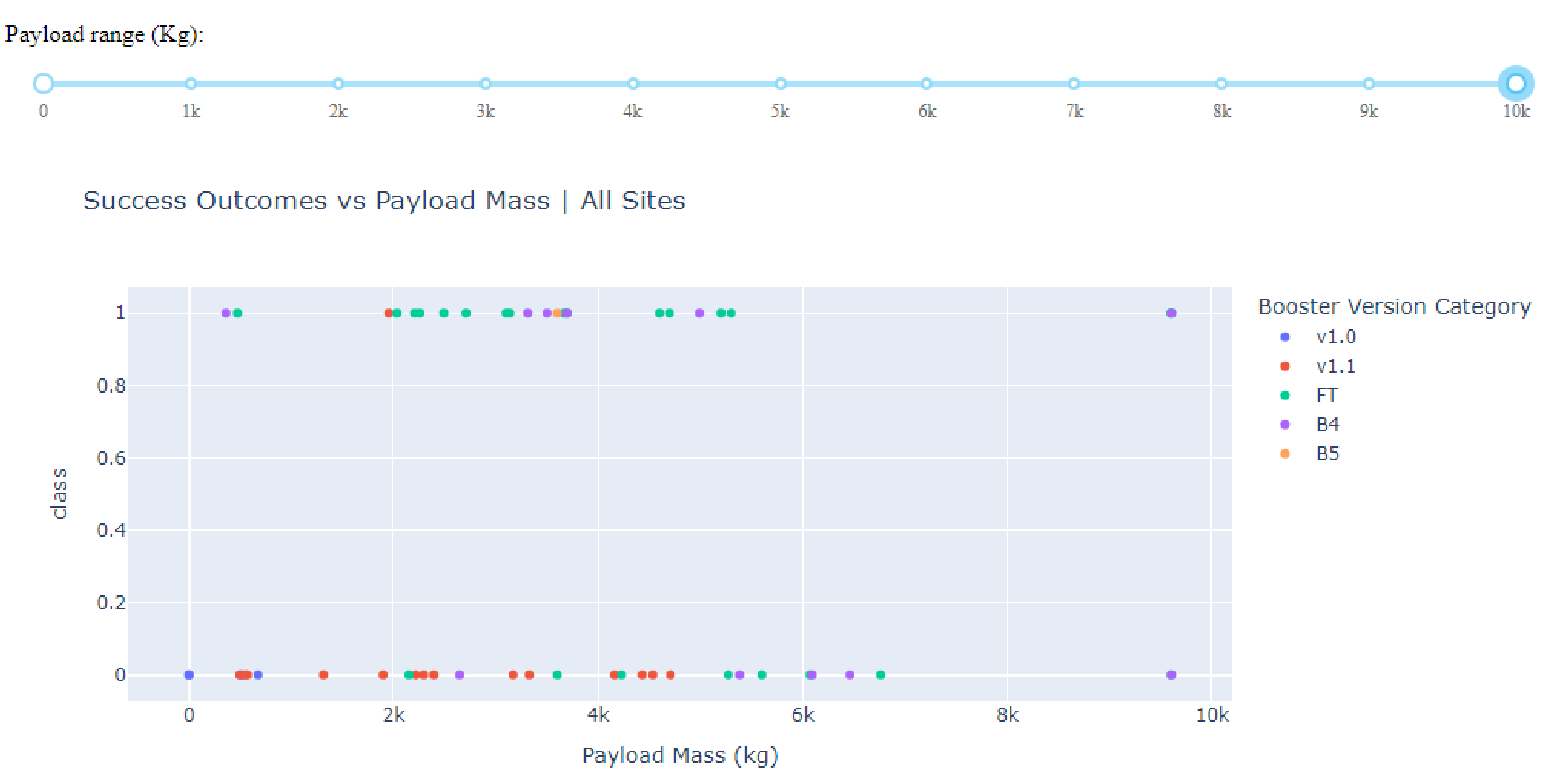
The graph shows a breakdown of the total successful landing outcomes among the launch sites. Based on the information shown, KSC-LC-39A has the most successful outcomes, while CCAFS-SLC-40 has the least.

Launch Rates for the Site with the Highest Success Ratio



The graph shows a breakdown of all landing outcomes for the most successful launch site. Based on the information shown, KSC-LC-39A has a total of 13 landings, 10 of which were successful, leaving the site with a 76.9% success rate. This can be translated to the amount of money saved by using this launch site.

Payload vs. Launch Success Outcome for All Sites

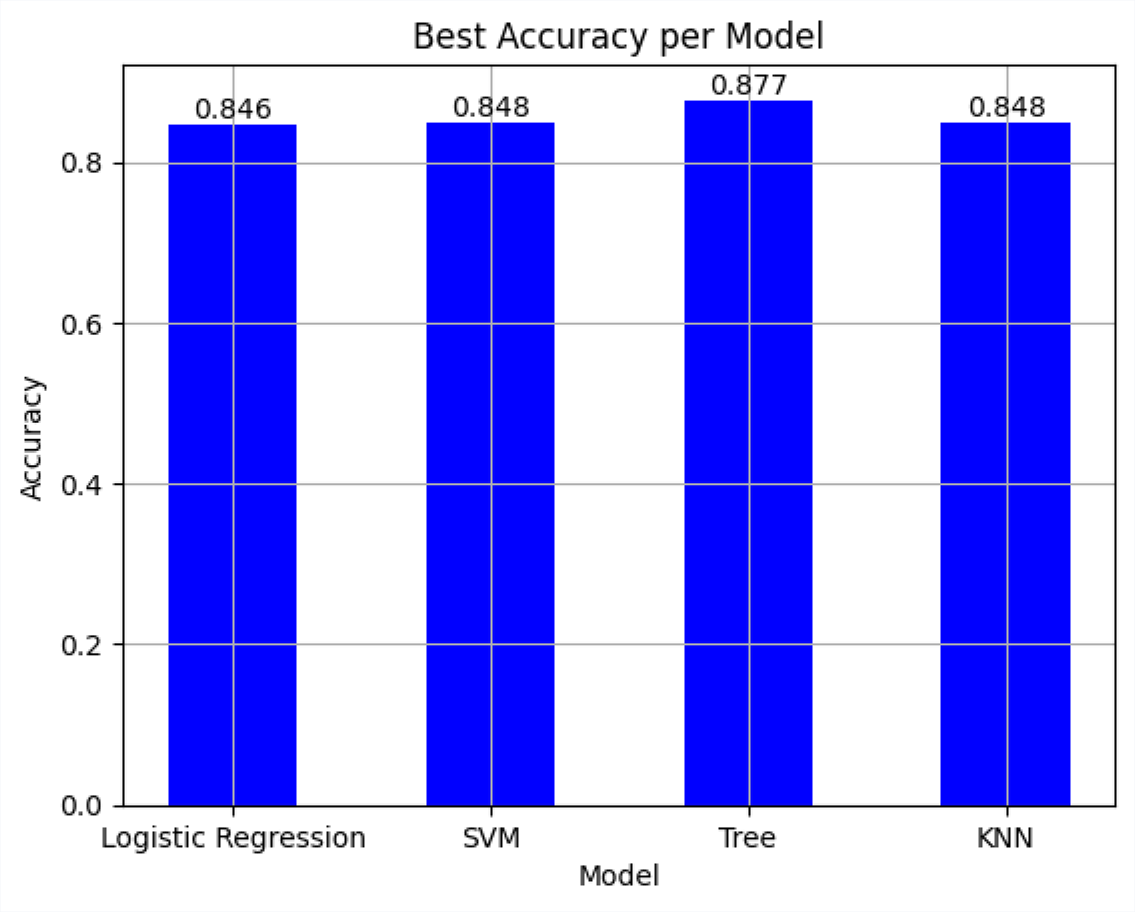


Using a demarcation of "0" for success and "1" for failure, the graph shows that companies should be focusing their efforts on using the v1.1 booster, as it shows clearly shows the largest success rates at payloads between 0 – 5,000 kg.

Section 5

Predictive Analysis (Classification)

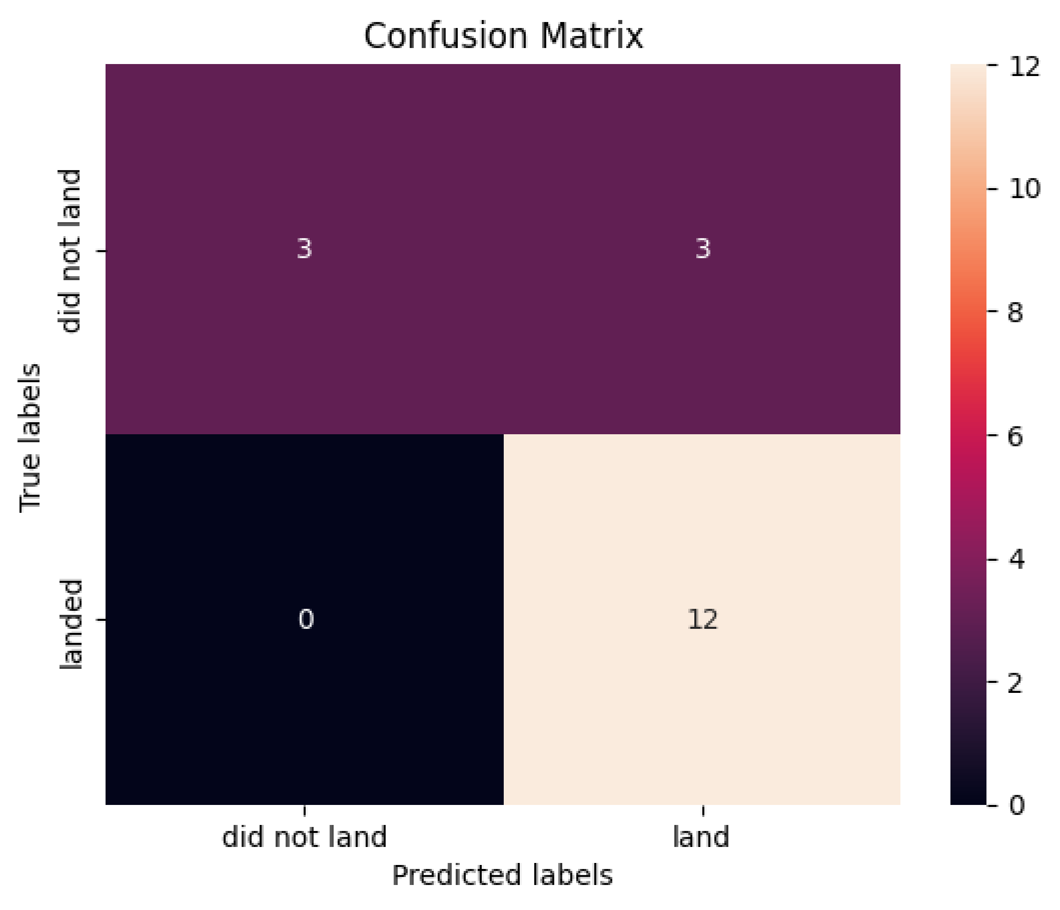
Classification Accuracy



The model with the highest accuracy is the Decision Tree.

Confusion Matrix

- The plot shows the confusion matrix of the Decision Tree. A total of 18 values were tested and of the 18, 15 were correctly predicted with an accuracy of 87%.



Conclusions

- EDA shows that the KSC-LC-39A launch site should be prioritized for a high probability for successful landing outcomes.
- Companies should prioritize v1.1 boosters at 0 – 5,000 kg payloads to increase successful landing outcomes.
- A Decision Tree is the best classifier for predicting successful landing outcomes.

Appendix

- Fin.

Thank you!