IBM Developer
SKILLS NETWORK

Rose-Marie PIPOKA
22/02/2025

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

Summary of methodologies

Data was collected via the SpaceX API and web scraping. It was cleaned, filtered for Falcon 9 launches, and missing PayloadMass values were replaced with the average. Additional details were enriched using the API.

Summary of all results

- After performing four classification models (Decision Tree, K-Nearest Neighbors, SVM, Logistic Regression), we observed that **the Decision Tree model** is the best-performing one. Therefore, we will focus on this model.
- Our analysis also shows that KSC LC-39A has the highest success rate among all launch sites

Introduction

Project background and context

With the rapid rise of commercial spaceflight, companies like Virgin Galactic, Rocket Lab, and Blue Origin are transforming access to space. However, **SpaceX**, founded by **Elon Musk**, stands out by significantly lowering launch costs through the **reuse of the Falcon 9's first stage**. As a result, a SpaceX launch costs around **$62 million**, compared to **over $165 million** for other providers.

In this project, as a **data scientist** at **Space Y**, a competitor of SpaceX, we will analyze **Falcon 9 launch data** to **predict whether the rocket's first stage will be successfully recovered**. This prediction is key to **estimating launch costs** and **guiding the company's strategic decisions**. To achieve this, we will train a **machine learning model** and develop an **interactive dashboard** to support decision-making.

Problems you want to find answers

- Witch factors influence the success of the first stage landing
- estimate the cost of each launch
- Pedict whether space X will reuse the first stage

Section 1

Methodology

Executive Summary

Data collection methodology:

The data used in this project is collected via the SpaceX REST API and web scraping. We use the API api.spacexdata.com/v4/launches/past to retrieve launch information, extracted in JSON format and normalized using the json_normalize function. Additionally, we use BeautifulSoup to scrape HTML tables containing Falcon 9 launch data.

Perform data wrangling

- The collected data has been cleaned and transformed.
- We filtered only Falcon 9 launches.
- We replaced the missing values in PayloadMass with their average and kept the missing values in LandingPad.
- We enriched the data by retrieving additional information via the API for columns such as Booster, Launchpad, Payload, and Core

Executive Summary

Perform exploratory data analysis (EDA) using visualization and SQL

After cleaning the data, we conducted an exploratory data analysis using various visualizations.

- We created correlation graphs between different attributes.
- We observed that the landing success rate has improved since 2013.
- We observed that launch sites have varying success rates.
- We also performed feature engineering by selecting relevant variables and creating dummy variables for categorical columns. Finally, we analyzed the data using SQL."

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification models

We used classification models such as Logistic Regression, Decision Trees, Support Vector Machines (SVM), K-Nearest Neighbors (KNN

Data Collection

To collect the data, we proceed as follow:

1. Import the necessary libraries (Requests, Pandas,numpy …)
2. Define four functions:
    - The first function takes the dataset and uses the "rocket" column to call the API and append the data to the list.
    - The second function takes the dataset and uses the "launchpad" column to call the API and append the data to the list.
    - The third function takes the dataset and uses the "payloads" column to call the API and append the data to the lists.
    - The last function takes the dataset and uses the "cores" column to call the API and append the data to the lists.
3. Requesting rocket launch data from SpaceX API with the following URL (spacex_url=https://api.spacexdata.com/v4/launches/past)
4. Requesting and parsing the SpaceX launch data using the GET request
5. The data from these requests will be stored in lists and will be used to create a new dataframe

You can see our approach through the organizational chart below.

Data Collection-API

Libraries importing
(Requests, Pandas, Numpy..)

Defining four functions
The four functions take a dataset and use different columns to call an API and append the retrieved data to lists. The first function uses the "rocket" column, the second uses "launchpad," the third relies on "payloads," and the last one uses "cores" to make the API calls and collect the results.

Requesting rocket launch data from SpaceX API using:

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

Make requestion JSON more consistent by using:
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
response=requests.get(static_json_url)

Use json_normalize meethod to convert the json result into a dataframe:

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

- Create an empty list for each column.
- Call each of the functions we defined earlier.
- Construct a dictionary to combine the data we have obtained.
- Use the dictionary to build the dataset by merging the columns.

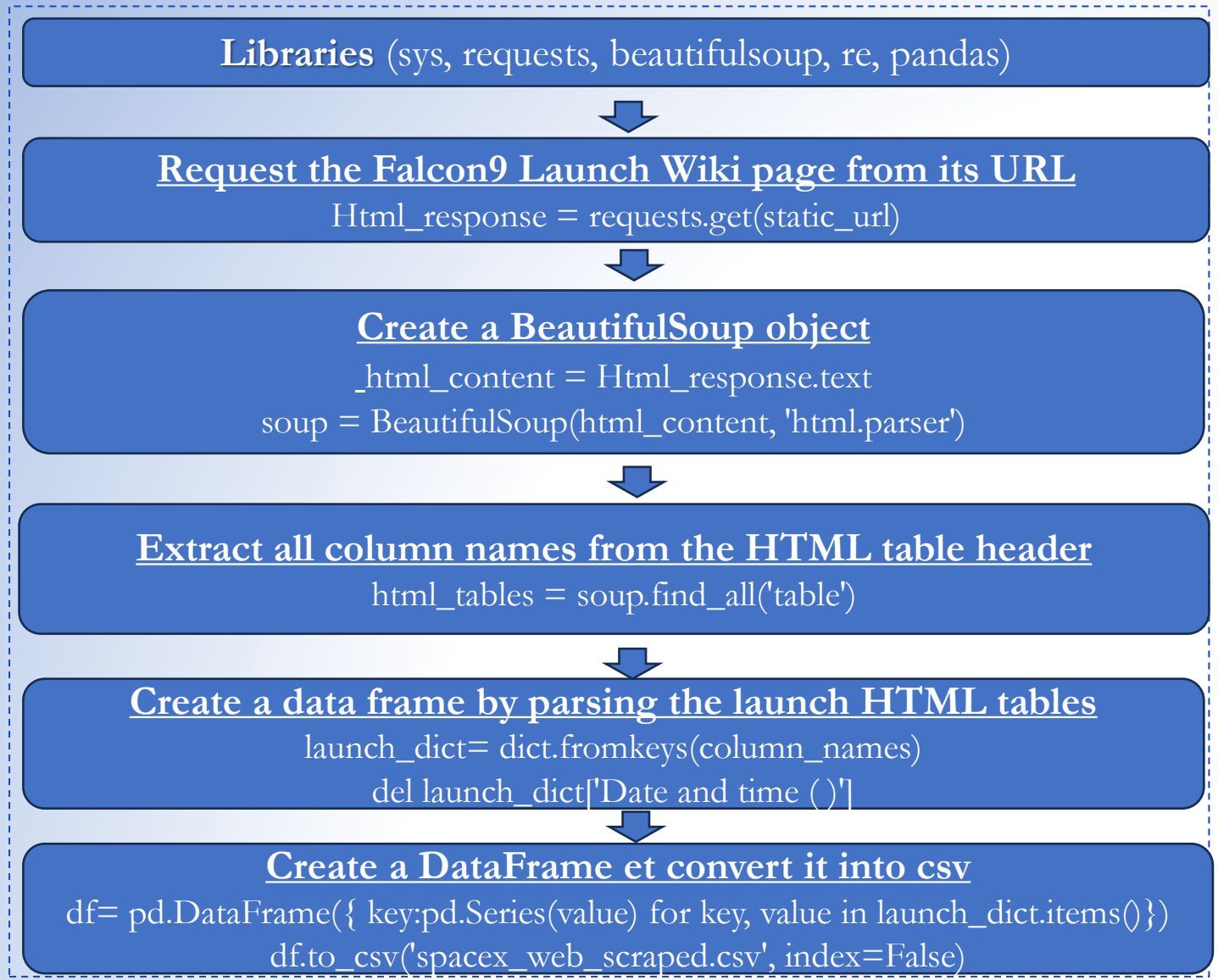
Github link – Data collecting API

Data Collection - Scraping

We perform Webscraping as follow:

- Importing libraries (sys, requests, beautifulsoup, re, pandas)
- Definition of 5 functions that will be used to process HTML tables retrieved via web scraping
- Performing an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.
- Create BeautifulSoup Object
- Collecting all relevant column names from the HTML table header
- Create a data frame by parsing the launch HTML table

[GitHub Link-Web scraping]



Data Wrangling

Loading the Dataset into dataframe

df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")

Identifing and calculate the percentage of the missing values

df.isnull().sum()/len(df)*100

Calculate the number of launches on each site

df['LaunchSite'].value_counts()

LaunchSite	Counts
• CCAFS SLC 40	55
• KSC LC 39A	22
• VAFB SLC 4E	13

Determine the number and occurrence of each orbit

df['Orbit'].value_counts()

Calculating the number and occurrence of mission outcomes for each orbit

landing_outcomes=df['Outcome'].value_counts()
landing_outcomes

Create a landing outcome label from Outcome column

landing_class = [0 if outcome in bad_outcomes else 1 for outcome in landing_outcomes.index]

Dave Data to csv:

df.to_csv("dataset_part_2.csv", index=False)

GitHub Link-Data wrangling

EDA with Data Visualization

1. We created a catplot to analyze the impact of FlightNumber and PayloadMass on landing success. A FlightNumber vs. PayloadMass chart was plotted, with the launch outcome overlaid

2. By a catplot, we visualize the relationship between Flight Number and Launch Site

3. We observed if there is any relationship between launch sites and their payload mass by making a scatterplot

4. We created a Bar Chart to find if there are any relationship between success rate and orbit type make a bar chart to

5. We created a scatterplot to see if there is any relationship between FlightNumber and Orbit type

6. We made a scatterplot to visualize the relationship between Payload Mass and Orbit type

7. We ploted a linechart to visualize the launch success
   yearly trend

GitHub Link - EDA

EDA with SQL

- Displaying the names of the unique launch sites in the space mission: %sql select distinct Launch_Site from SPACEXTABLE ;

- Displaying the names of the unique launch sites in the space mission: %sql select Launch_Site from SPACEXTABLE where Launch_Site LIKE 'CCA%' LIMIT 5;

- Displaying the total payload mass carried by boosters launched for NASA (CRS)

%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE Customer = 'NASA (CRS)';

- Displaying the average payload mass carried by the F9 v1.1 booster version:

%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1';

- List of the date when the first succesful landing outcome in ground pad was acheived:

%%sql select min(Date) as min_date from spacextbl where Landing__Outcome = 'Success (ground pad)';

- Listing the names of boosters that successfully landed on the drone ship and have a payload mass greater than 4000 but less than 6000:

%sql select Booster_Version,Landing_Outcome,PAYLOAD_MASS__KG_ from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000;

EDA with SQL

- Listing the total number of successful and failure mission outcomes:

```
%sql SELECT Landing_Outcome, COUNT(*) AS Total_Count FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success%' OR Landing_Outcome LIKE 'Failure%'GROUP BY Landing_Outcome;
```

- Listing the names of the booster_versions which have carried the maximum payload mass. Use a subquery:

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

- Listing the records displaying the month names, failure landing outcomes on the drone ship, booster versions, and launch sites for the months of the year 2015:

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

- 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:

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

GitHub Link – EDA with SQL

Build an Interactive Map with Folium

- We created a folium Map object, with an initial center location to be NASA Johnson Space Center at Houston, Texas.
- We used folium.Circle to add a highlighted circle area with a text label on a specific coordinate
- We created and add folium.Circle and folium.Marker for each launch site on the site map
- We created markers for all launch records. If a launch was successful (class=1), then we use a green marker and if a launch was failed, we use a red marker (class=0)
- For each launch result in spacex_df data frame, we added a folium.Marker to marker_cluster
- We Calculated the distances between a launch site to its proximities by adding a MousePosition on the map to get coordinate for a mouse
- We Created and added a folium.Marker on your selected closest coastline point on the map
- We Drawed a PolyLine between a launch site to the selected coastline point
- Created a marker with distance to a closest city, railway, highway

We added these objects to analyze and identify the various geographical factors that may influence the launch success rate. Identifying these factors will help determine the optimal location for building a launch site.

Build a Dashboard with Plotly Dash

On the Dashboard we created, we added:

- a launch Site Drop-down Input Component
- a callback function to render success-pie-chart based on selected site dropdown
- a range Slider to Select Payload
- a callback function to render the success-payload-scatter-chart scatter plot

We added those plots and interaction to show:

- Which site has the largest successful launches
- Which site has the highest launch success rate
- Which payload range(s) has the highest launch success rate
- Which payload range(s) has the lowest launch success rate

Github Link-Dashboard

Predictive Analysis (Classification)

After importing the dataset using pandas, we standardized some features using:

transform = preprocessing.StandardScaler()X = transform.fit_transform(X)

- We split the dataset using the **train_test_split function:**

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

- We used GridSearch and tested four models: **Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K-Nearest Neighbors Classifier**.

Logistic Regression

logreg_cv = GridSearchCV(lr, parameters1,cv=10)

logreg_cv.fit(X_train, Y_train)

Support Vector Maching

svm_cv = GridSearchCV(svm, parameters2, cv=10, scoring='accuracy', verbose=1, n_jobs=-1)achine

svm_cv.fit(X_train, Y_train)

Decision tree

tree_cv = GridSearchCV(tree, parameters3, cv=10, scoring='accuracy', verbose=1, n_jobs=-1)
tree_cv.fit(X_train, Y_train)

K-Nearest Neighbors

knn_cv = GridSearchCV(KNN, parameters4, cv=10, scoring='accuracy', verbose=1, n_jobs=-1)
knn_cv.fit(X_train, Y_train)

- For each model, we created a confusion matrix and calculated its accuracy.

accuracies = { "Decision Tree": tree_cv.best_score_,"K-Nearest Neighbors": knn_cv.best_score_, "SVM": svm_cv.best_score_, "Logistic Regression": logreg_cv.best_score_}

- Finally, we compared the accuracies of the four models to determine which one performed the best.

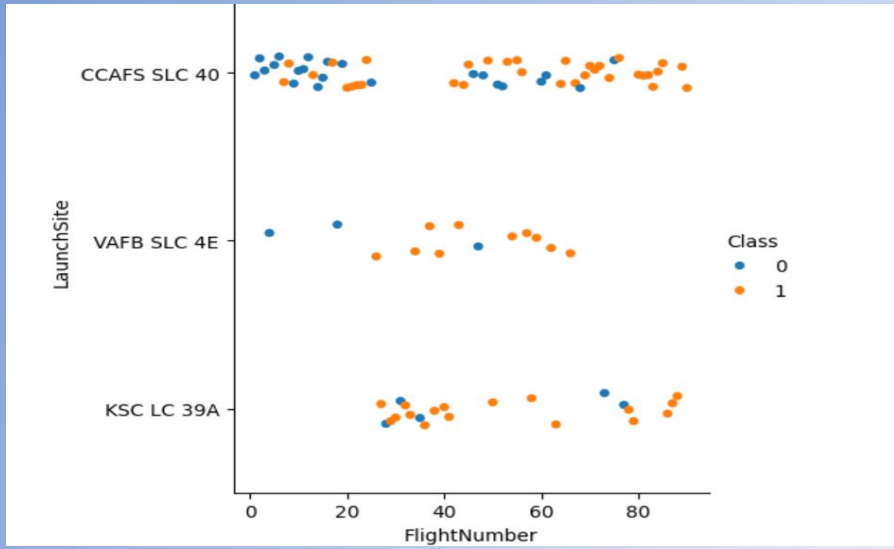
Results

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

Section 2

Flight Number vs. Launch Site

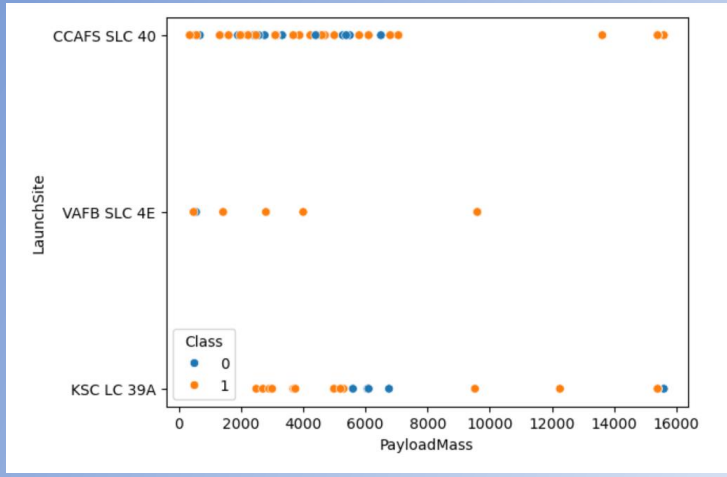
- scatter plot of Flight Number vs. Launch Site



- This scatter plot shows that for CCAFS SLC 40, failures and successes are mixed, but there seem to be more successful launches in recent flights.For VAFB SLC 4E, there is a mix of failures and successes without a clear trend.For KSC LC 39A, recent flights appear to have more successes than failures.

Payload vs. Launch Site

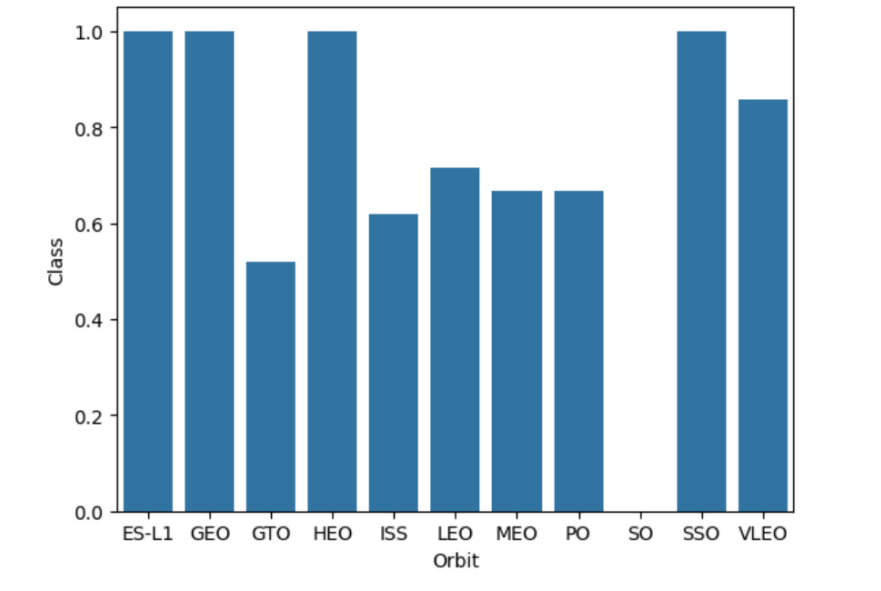
Scatter plot of Payload vs. Launch Site



This scatter plot shows that heavy payloads (>10,000 kg) mostly succeed, especially at KSC LC 39A and CCAFS SLC 40.Failures are more common for intermediate payloads (~6,000 kg), particularly at KSC LC 39A.The VAFB SLC 4E site has few failures, though the total number of launches there is low.

Success Rate vs. Orbit Type

Bar chart for the success rate of each orbit type



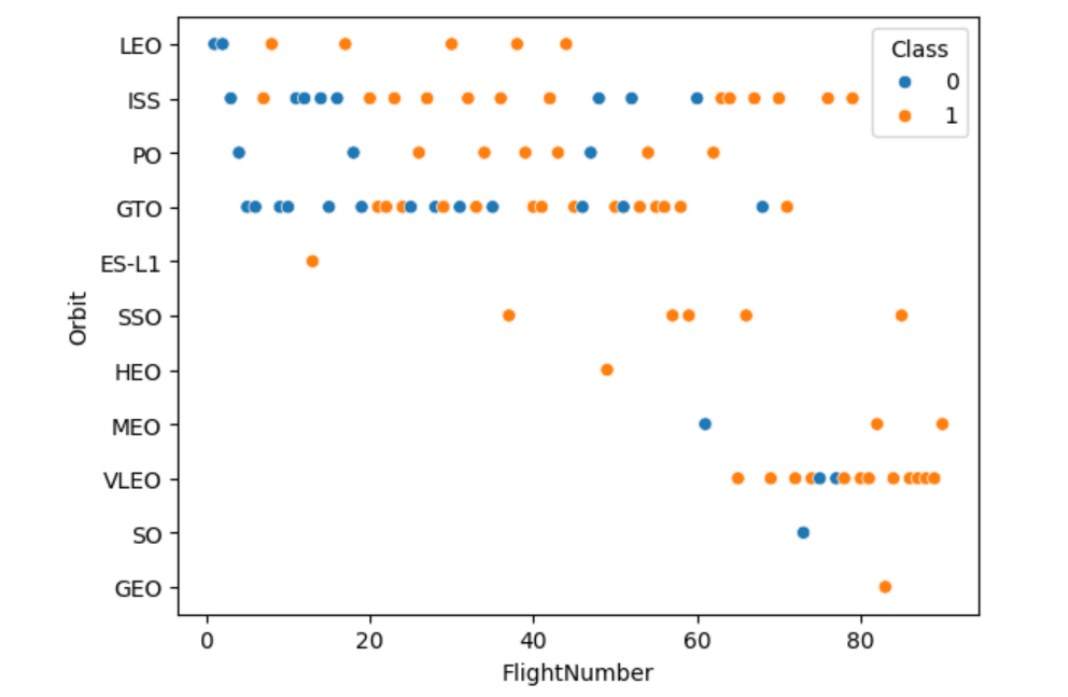
This bar chart represents the success rate (Class) of different orbit types. Here are the key observations:

- **ES-L1, GEO, HEO, and SSO** have a **100% success rate**, indicating that all launches to these orbits were successful.
- **GTO has the lowest success rate**, suggesting challenges in achieving successful launches for this orbit.
- **LEO, MEO, and PO** show **moderate success rates**, indicating variability in mission success.
- **VLEO also has a high success rate**, but slightly lower than the top-performing orbits.

This analysis suggests that certain orbits, particularly ES-L1, GEO, and SSO, have consistently high success rates, while others like GTO face more difficulties.

Flight Number vs. Orbit Type

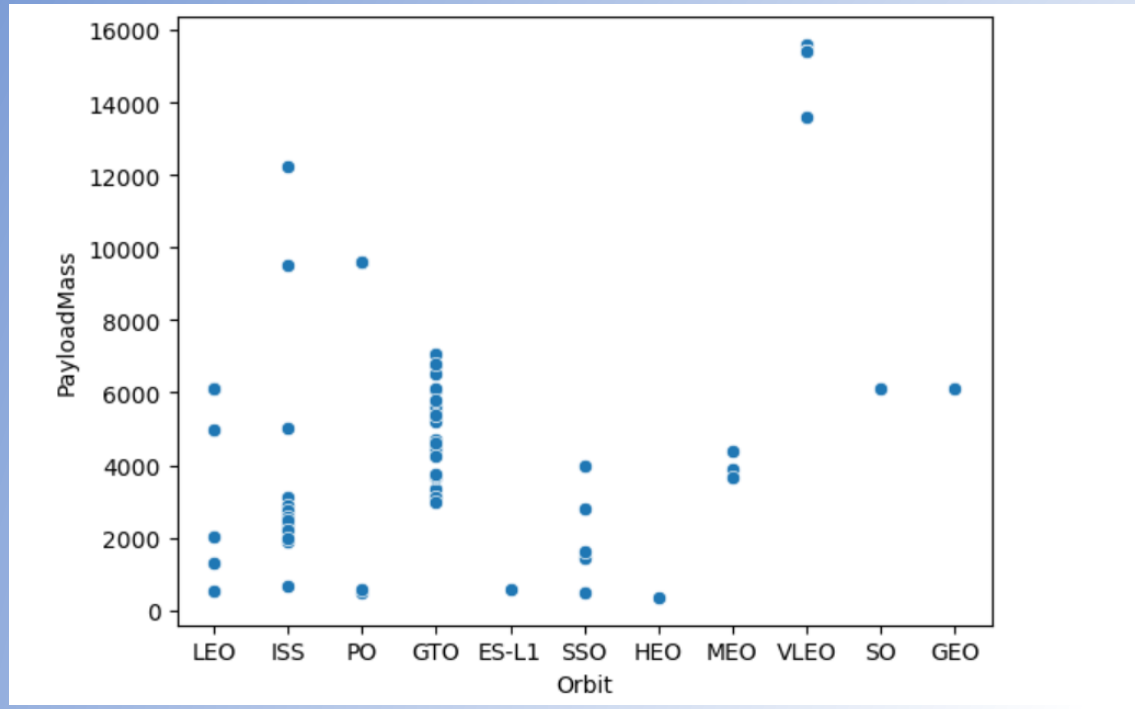
Scatter point of Flight number vs. Orbit type



We can see from this scatter plot that the SSO orbit has no successful launches. For LEO orbits, as the number of flights increases, the number of successful launches also increases. For VLEO, we can observe some successful launches toward the end.

Payload vs. Orbit Type

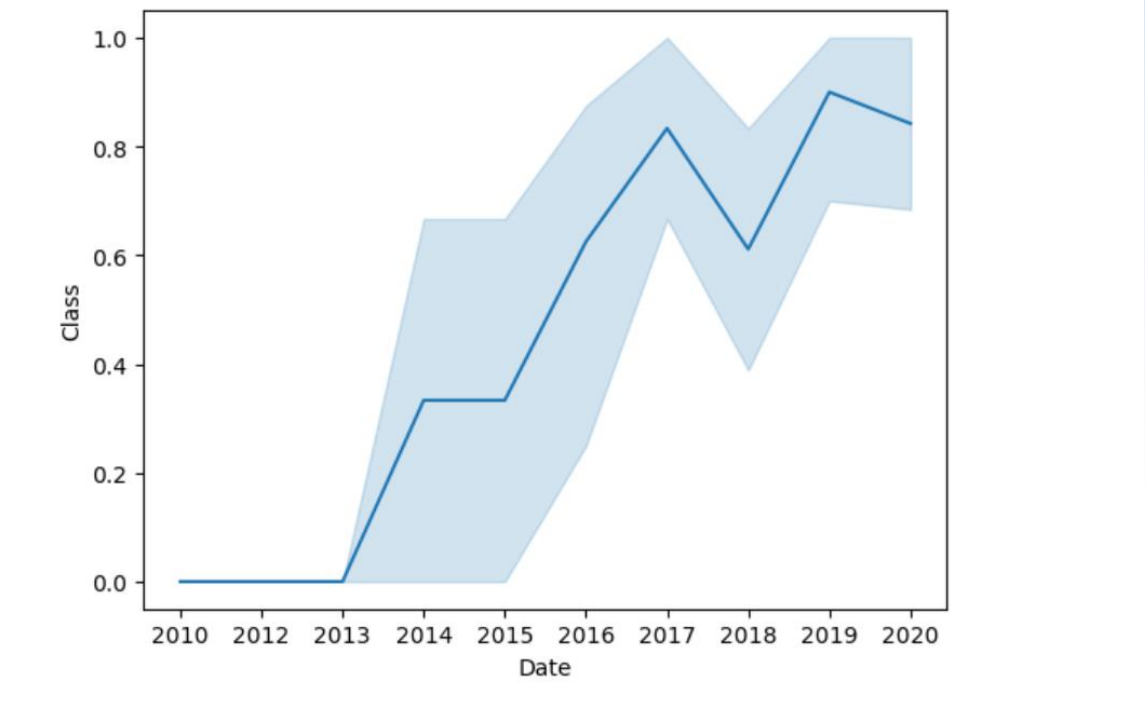
Scatter point of payload vs. orbit type



There is no correlation between PayloadMass and Orbit

Launch Success Yearly Trend

Line chart of yearly average success rate



This graph shows that the success rate of launches has increased since 2013.

All Launch Site Names

Unique launch sites names

%sql select distinct Launch_Site from SPACEXTABLE ;

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

There are four different launch site

Launch Site Names Begin with 'CCA'

5 records where launch sites begin with `CCA`

```
%sql select Launch_Site from SPACEXTABLE where Launch_Site
LIKE 'CCA%' LIMIT 5;
```

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

Total Payload Mass

Total payload carried by boosters from NASA

%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE Customer = 'NASA (CRS)';

SUM(PAYLOAD_MASS__KG_)
45596

The Total payload carried by boosters is 45596 KG

Average Payload Mass by F9 v1.1

Average payload mass carried by booster version F9 v1.1

%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1';

avg(PAYLOAD_MASS__KG_)
2928.4

The average of total payload carried by booster version F9 v1.1 is 2928.4

First Successful Ground Landing Date

- Dates of the first successful landing outcome on ground pad

%sql select Date, Landing_Outcome from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)' order by date asc;

Date	Landing_Outcome
2015-12-22	Success (ground pad)
2016-07-18	Success (ground pad)
2017-02-19	Success (ground pad)
2017-05-01	Success (ground pad)
2017-06-03	Success (ground pad)
2017-08-14	Success (ground pad)
2017-09-07	Success (ground pad)
2017-12-15	Success (ground pad)
2018-01-08	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql select Booster_Version,Landing_Outcome,PAYLOAD_MASS__KG_ from SPACEXTABLE where
Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

Total number of successful and failure mission outcomes

```
%sql SELECT Landing_Outcome, COUNT(*) AS Total_Count FROM SPACEXTABLE WHERE
Landing_Outcome LIKE 'Success%' OR Landing_Outcome LIKE 'Failure%'GROUP BY Landing_Outcome;
```

Landing_Outcome	Total_Count
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
Success	38
Success (drone ship)	14
Success (ground pad)	9

Boosters Carried Maximum Payload

Names of the booster which have carried the maximum payload mass

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

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

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

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

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

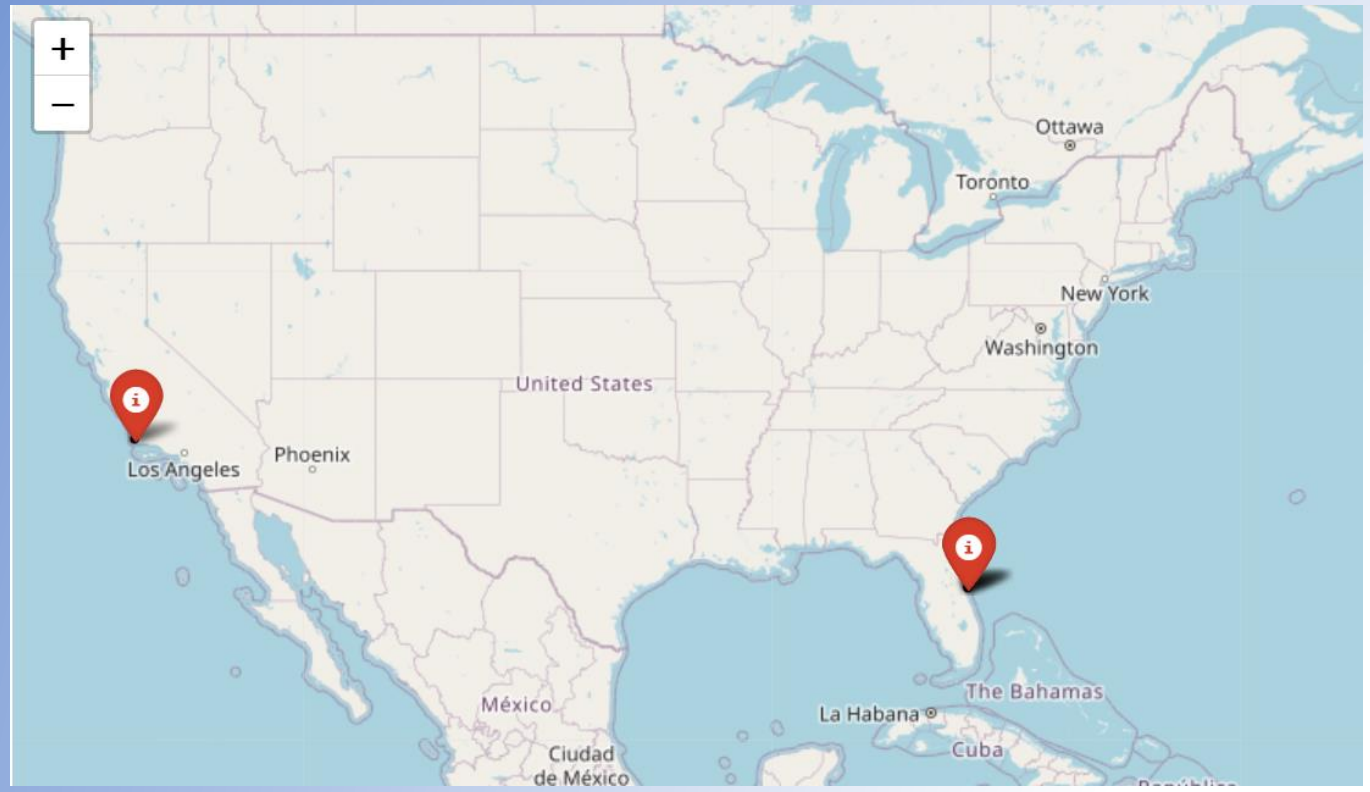
Ranking the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

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

Landing_Outcome	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

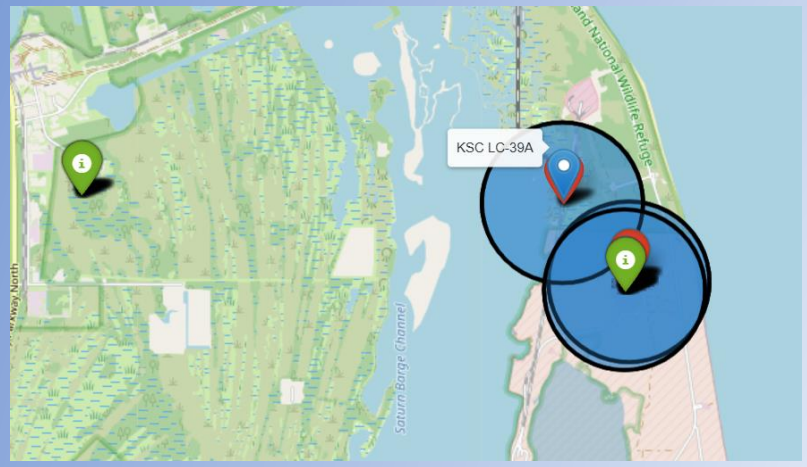
Section 3

Interactive Map Visualization with Folium



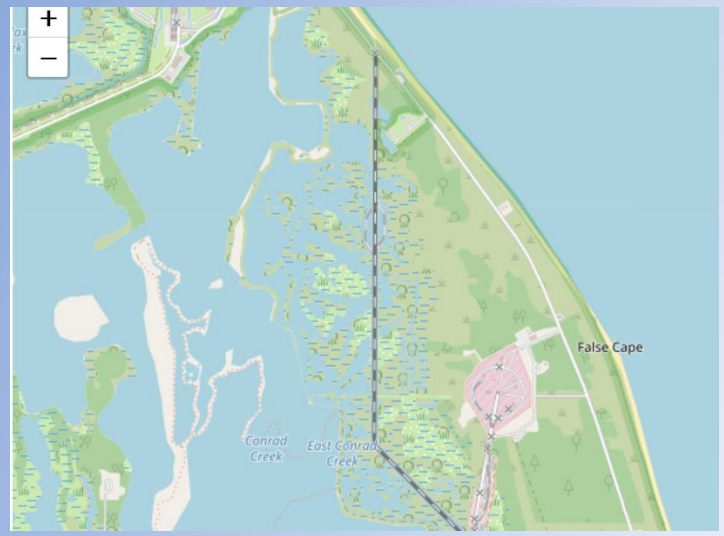
Folium Map showing the color-labeled launch outcomes

- Exploring the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map



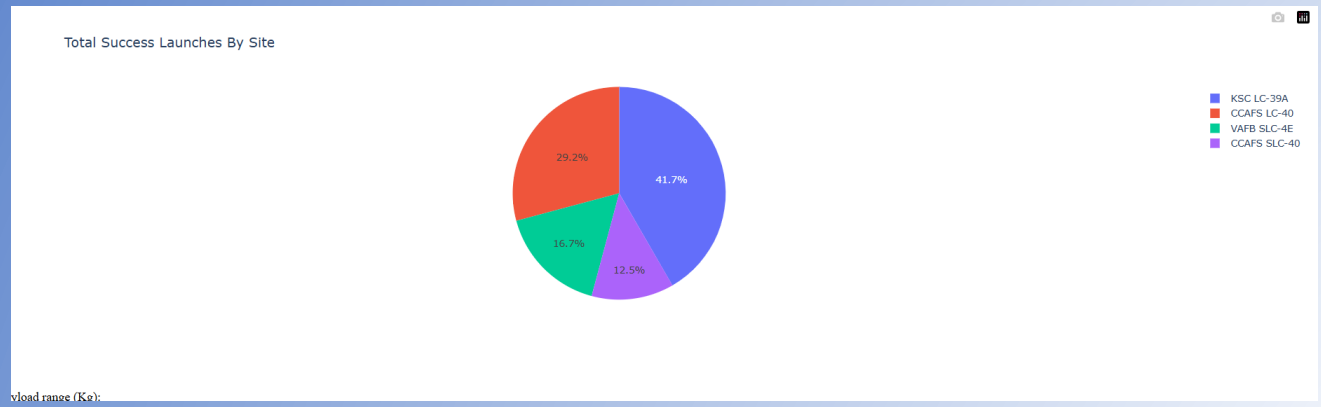
Folium Map showing a selected launch site to its proximities

- Explain the important elements and findings on the screenshot



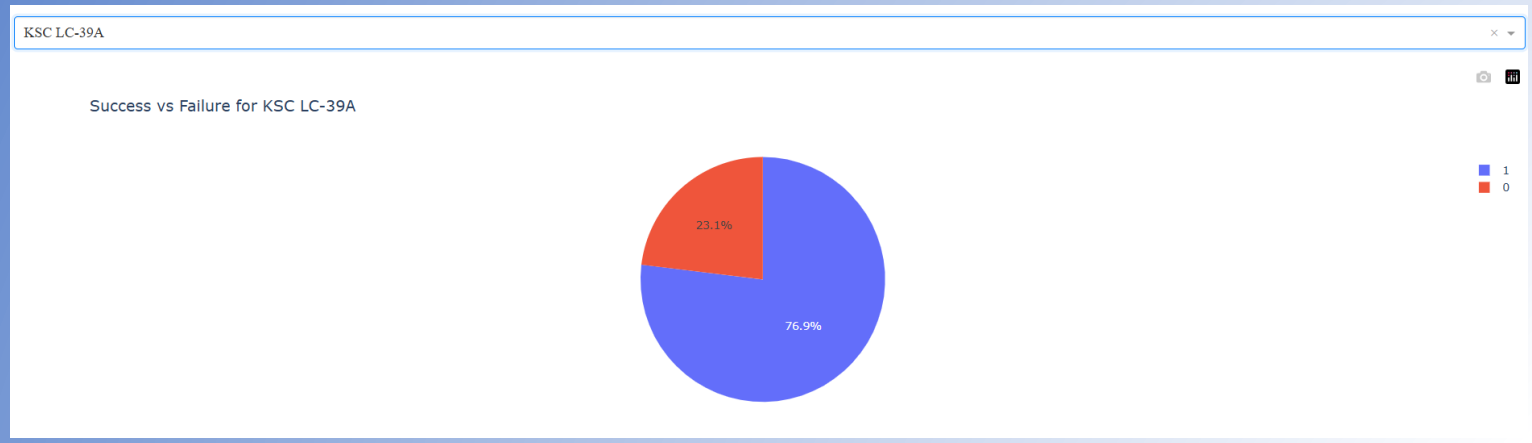
Section 4

Total success launches by sites



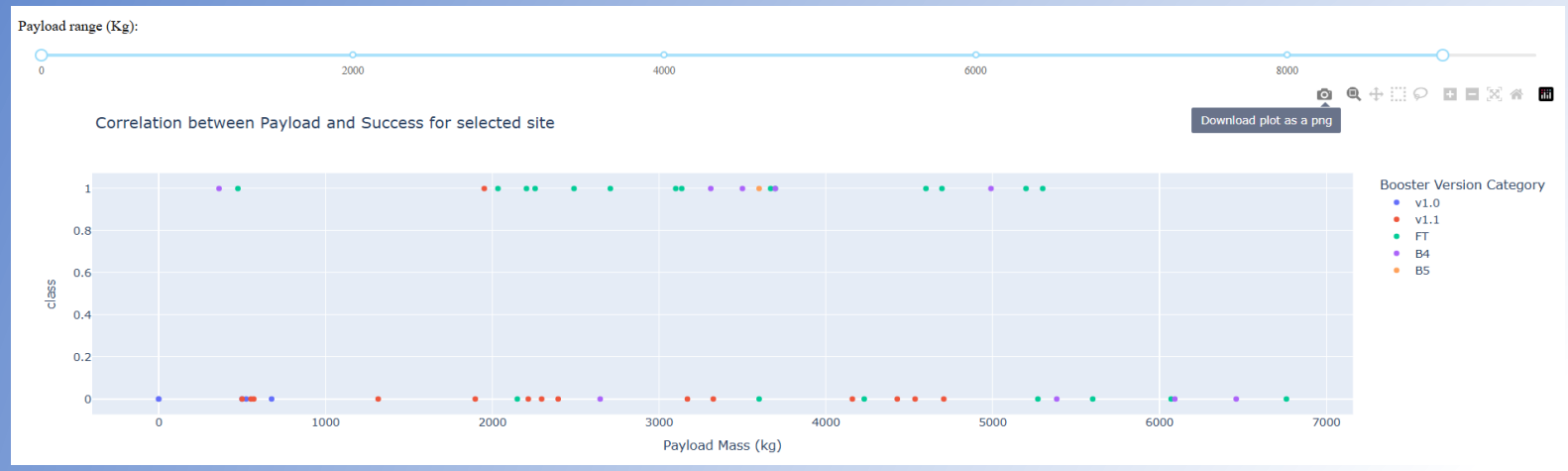
This screenshot indicates that the KSC LC-39A launch site has recorded the highest number of successful launches. Conversely, the site with the fewest successful launches is VAFB SLC-4E.

Pie chart of success vs failure for KSC LC-39A



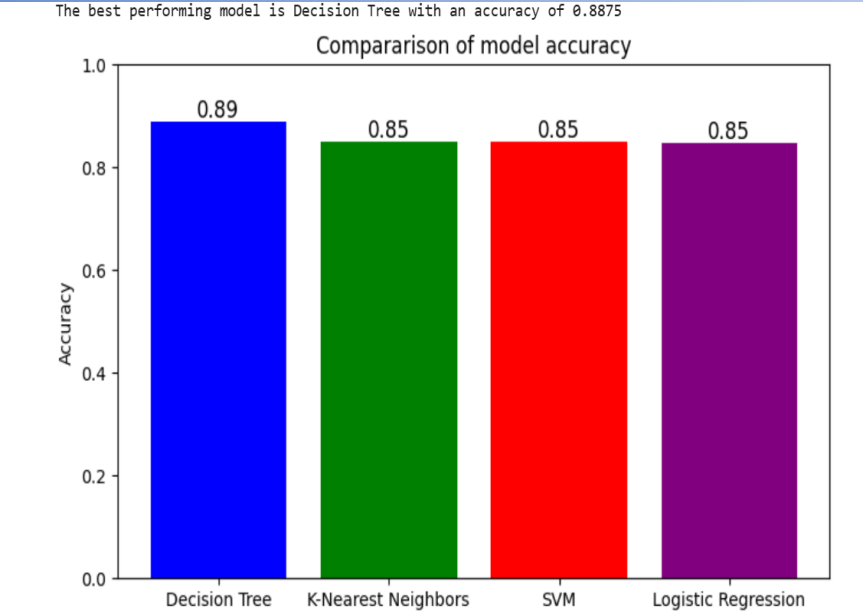
The data reveals that the KSC LC-39A launch site has a success rate of 76.9%, while 23.1% of the launches from this site have resulted in failure.

Correlation between payload and success for all site of launch



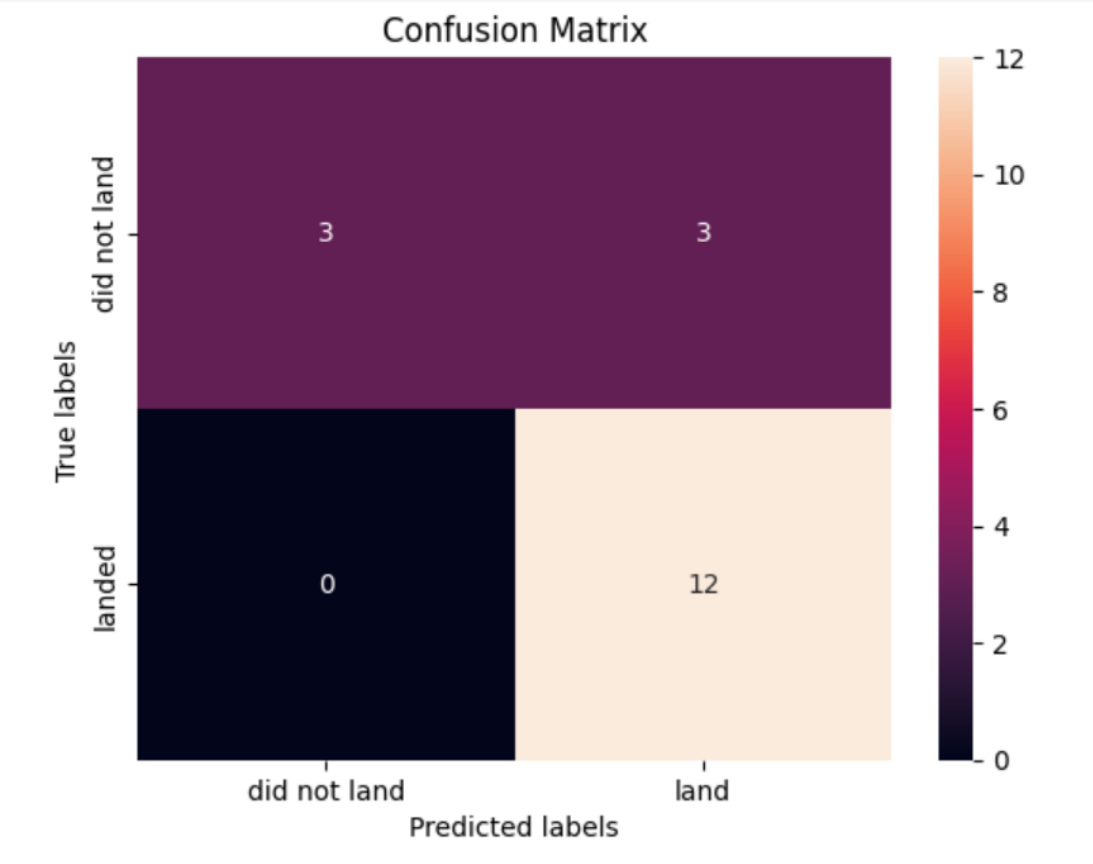
Section 5

Classification Accuracy



The best mode is decision Tree with an accuracy = 0.89

Confusion Matrix



Conclusions

- The best-performing model is the Decision Tree.
- KSC LC-39A is the launch site with the highest success rate.
- Payload mass and the number of launches are key factors to consider.
- Orbits like LEO, and GTO could be prioritized.