IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Bill Kamanzi
7 Aug 2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
    - EDA with Visulaization
    - EDA with SQL
    - Interactive Maps with Folium
    - Plotly Dash Dashboard
    - Predicitve Analytics
- Conclusion

Executive Summary

Summary of Methodologies:

The research aimed to identify factors for a successful rocket landing using the following methodologies:

- Collected and Wrangled data to create a success/fail outcome variable via SpaceX REST API and web scraping
- Visualized and Analyzed data with SQL, considering payload, launch site, flight number, and yearly trends
- Visualized launch sites with the highest success rates and successful payload ranges
- Built predictive models using logistic regression, SVM, decision tree, and KNN

Results:

Exploratory Data Analysis:

- Launch success improved over time.
- KSC LC-39A had the highest success rate among landing sites.
- Orbits ES-L1, GEO, HEO, and SSO have a 100% success rate.

Visualization/Analytics:

- Most launch sites are near the equator and close to the coast.

Predictive Analytics:

- All models performed similarly on the test set, with the decision tree model slightly outperforming the others.

Introduction

Background:

SpaceX’s Falcon 9 rocket has revolutionized space travel with its reusable first stage, significantly reducing the cost of launches. The cost of a Falcon 9 launch is approximately $62 million, compared to over $165 million for other providers. The ability to reuse the first stage is a major factor in this cost difference. SpaceX’s success in landing and reusing these stages is a key competitive advantage.

Context:

Predicting first stage landings helps SpaceX improve pricing and offers competitors insights for competitive bidding.

Github link:
https://github.com/kb546/IBM_Professional_Data_Science_Captsone_Project

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Via SpaceX API and Web Scraping
- Perform data wrangling
  - Describe how data was processed
- 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

Steps:

- Request Data: Use SpaceX API to get launch data.
- Data Wrangling: Clean and format data using Python libraries (requests, pandas, numpy).

Functions:

- getBoosterVersion: Fetches booster names.
- getLaunchSite: Retrieves launch site details.
- getPayloadData: Gets payload mass and orbit.
- getCoreData: Extracts core landing data and other attributes.

Process:

- Make API requests and convert responses into a Data frame.
- Filter to include only Falcon 9 launches.

Final Output: Data frame with launch details, filtered for Falcon 9 launches.

Data Collection – SpaceX API

- Used the GET request to the SpaceX API to collect data, decoded the response as a Json string, then proceeded to format it into a Pandas dataframe.

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[17]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storag
```

We should see that the request was successfull with the 200 status response code

```
[18]: response = requests.get(static_json_url)
      response.status_code
```

```
[18]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
[19]: # Use json_normalize meethod to convert the json result into a da
      df = response.json()
      data = pd.json_normalize(df)
```

Using the dataframe `data` print the first 5 rows

```
[20]: # Get the head of the dataframe
      data.head()
```

Data Collection - Scraping

- Performed an HTTP GET request to fetch the Falcon 9 Launch Wiki page and created a BeautifulSoup object from the response for parsing.

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

[6]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

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

[7]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text cont
soup = BeautifulSoup(response.text, 'html.parser')
```

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

[8]:
```
# Use soup.title attribute
soup.title
```

[8]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

Exploratory Data Analysis:

- Calculate Number of Launches at Each Site:
  - Determine launches per site
- Calculate Number of Occurrences for Each Orbit:
  - Determine occurrences per orbit
- Calculate Mission Outcome Occurrences:
  - Determine occurrences per outcome

Determine Training Labels:

- Identify Unsuccessful Landings:
  - Define bad outcomes
- Create Landing Outcome Labels:
  - Assign 0 for unsuccessful, 1 for successful
- Determine Success Rate:
  - Calculate success rate
- Export Processed Data:
  - Export to CSV for further analysis

Data Wrangling (Exploratory Data Analysis)

```
# Apply value_counts() on column LaunchSite
launch_site_counts = df['LaunchSite'].value_counts()
launch_site_counts
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

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

```
[11]: 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: Outcome, dtype: int64
```

```
# Apply value_counts on Orbit column
orbit_counts = df['Orbit'].value_counts()
orbit_counts
```

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

Data Wrangling (Determine Training Labels)

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

We create a set of outcomes where the second stage did not land successfully

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

```
[18]: # landing_class = 0 if bad_outcome
      # landing_class = 1 otherwise
      # Create the landing_class list based on whether Outcome is in bad_outcomes
      landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]

      # Print the resulting landing_class list
      print(landing_class)
```

This variable will represent the classification variable that represents the outcome c
not land successfully; one means the first stage landed Successfully

```
[19]: df['Class']=landing_class
      df[['Class']].head(8)
```

```
[16]: df.head(5)
```

We can use the following line of code to determine the success rate:

```
df["Class"].mean()
```

```
0.6666666666666666
```

We can now export it to a CSV for the next section,but to make the ansv
selected date range.

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

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

EDA with Data Visualization

Summary of Plots:

- **Number of Launches by Site:** Scatter plot to analyze the distribution of launches across different sites.
  - **Key Insight:** CCAFS SLC 40 is the most frequently used launch site with 55 launches, indicating its pivotal role in SpaceX's launch operations.
- **Number of Launches by Orbit Type:** Scatter plot to understand the preferred orbit types for Falcon 9 missions.
  - **Key Insight:** In LEO, a higher number of flights is associated with successful landings, while in GTO, the number of flights does not affect landing success.
- **Yearly Trend of Launch Success Rate:** Line chart illustrates the yearly trend in Falcon 9 launch success rates, highlighting improvements and identifying any fluctuations in performance over time.
  - **Key Insight:** The chart reveals a steady increase in Falcon 9's launch success rate from 2013 to 2020, indicating a consistent improvement in performance over the years.

EDA with SQL

A summary of the queries performed:

- Calculate the total payload mass carried by boosters for NASA (CRS) missions.
- Identify the date of the first successful landing on a ground pad.
- List the boosters that successfully landed on a drone ship and carried a payload mass between 4000 and 6000 kg.
- Count the total number of successful and failed mission outcomes.
- Determine the booster versions that carried the maximum payload mass.
- Show records with month names, failed drone ship landings, booster versions, and launch sites for the year 2015.
- Rank the count of different landing outcomes between 2010-06-04 and 2017-03-20 in descending order.

Build an Interactive Map with Folium

Markers Indicating Launch Sites:

- Added **blue** circle at NASA Johnson Space Center's coordinate with a popup label showing its name using its latitude and longitude coordinates
- Added **red** circles at all launch sites coordinates with a popup label showing its name using its name using its latitude and longitude coordinates

Colored Markers of Launch Outcomes:

- Added colored markers of successful (**green**) and unsuccessful (**red**) launches at each launch site to show which launch sites have high success rates

Distances Between a Launch Site to Proximities:

- Added colored lines to show distance between launch site CCAFS SLC40 and its proximity to the nearest coastline, railway, highway, and city.

Build a Dashboard with Plotly Dash

Summary of Dashboards:

Dropdown List with Launch Sites:

- Allow user to select all launch sites or a certain launch site

Slider of Payload Mass Range:

- Allow user to select payload mass range

Pie Chart Showing Successful Launches:

- Allow user to see successful and unsuccessful launches as a percent of the total

Scatter Chart Showing Payload Mass vs. Success Rate by Booster Version:

- Allow user to see the correlation between Payload and Launch Success

Predictive Analysis (Classification)

Data Preparation:

- Loaded SpaceX launch data.
- Standardized features and split data into training/testing sets.

Model Building and Tuning:

- **Logistic Regression:**
    - Best Params: C=0.01, penalty=l2, solver=lbfgs
    - Training Accuracy: 0.846 | Test Accuracy: 0.833
- **Decision Tree:**
    - Best Params: criterion=entropy, max_depth=8
    - Training Accuracy: 0.8625

Evaluation

- Plotted confusion matrices.
- Evaluated models with GridSearchCV.

Conclusion:

- Decision Tree performed best for predicting Falcon 9 landings.

Results Summary

Exploratory Data Analysis:

- Launch success has improved over time
- KSC LC-39A has the highest success rate among landing sites
- Orbits ES-L1, GEO, HEO and SSO have a 100% success rate

Visual Analytics:

- Most launch sites are near the equator, and all are close to the coast
- Launch sites are far enough away from anything a failed launch can damage (city, highway, railway), while still close enough to bring people and material to support launch activities

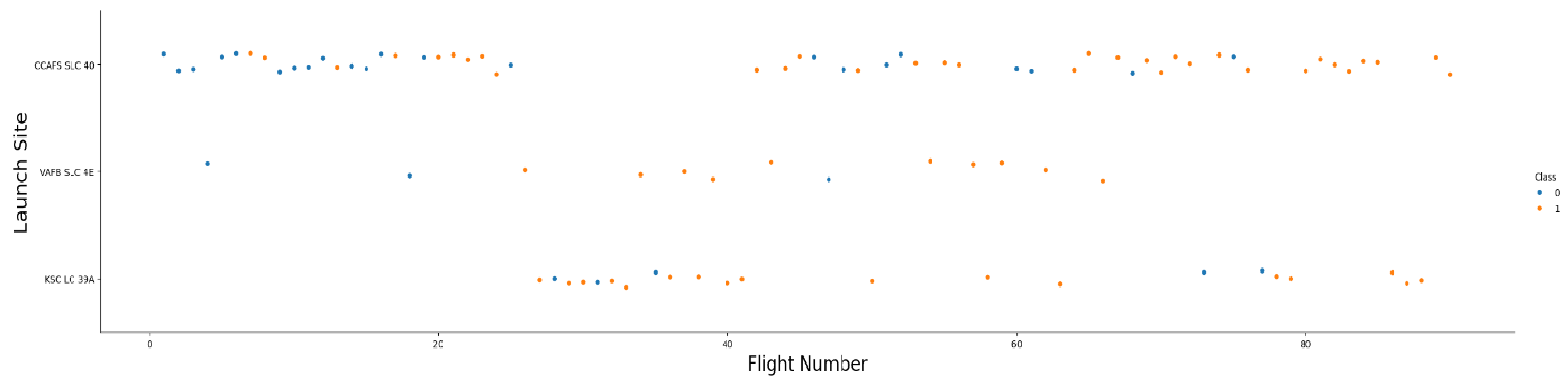
Predictive Analytics:

- All models have the same test accuracy of 83%, meaning that, according to the test data, they all perform equally well.

Section 2

Insights drawn from EDA

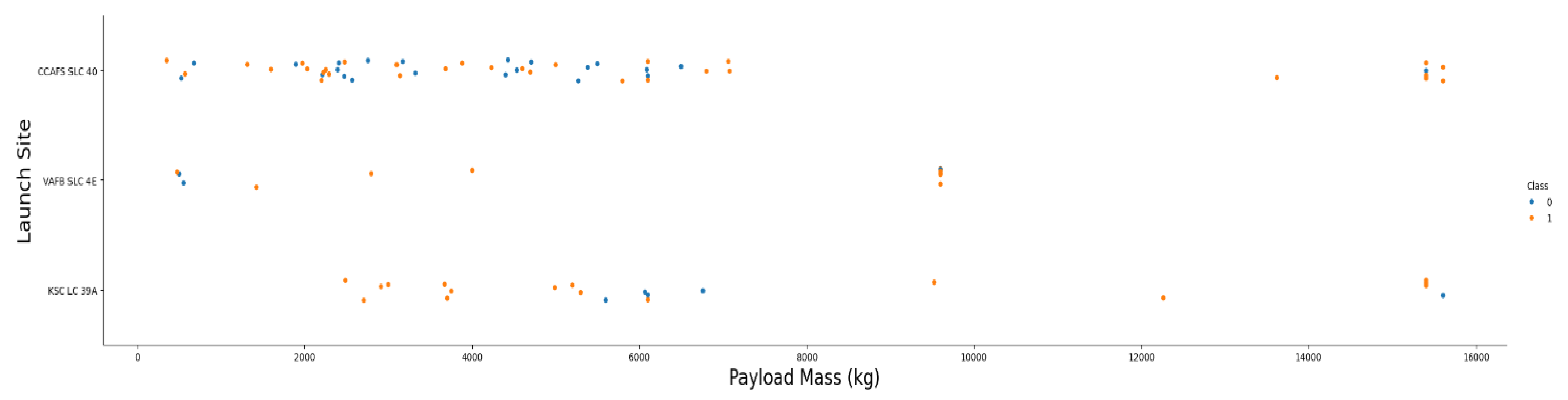
Flight Number vs. Launch Site



Key Insights:

- CCAFS SLC 40 and KSC LC 39A have higher flight numbers, indicating these sites are more frequently used.
- KSC LC 39A appears to have a higher proportion of successful landings compared to other sites, indicating it might offer conditions that favor successful landings.
- The pattern of successful landings increasing with flight numbers indicates a positive trend in SpaceX's landing technology and methods.

Payload vs. Launch Site



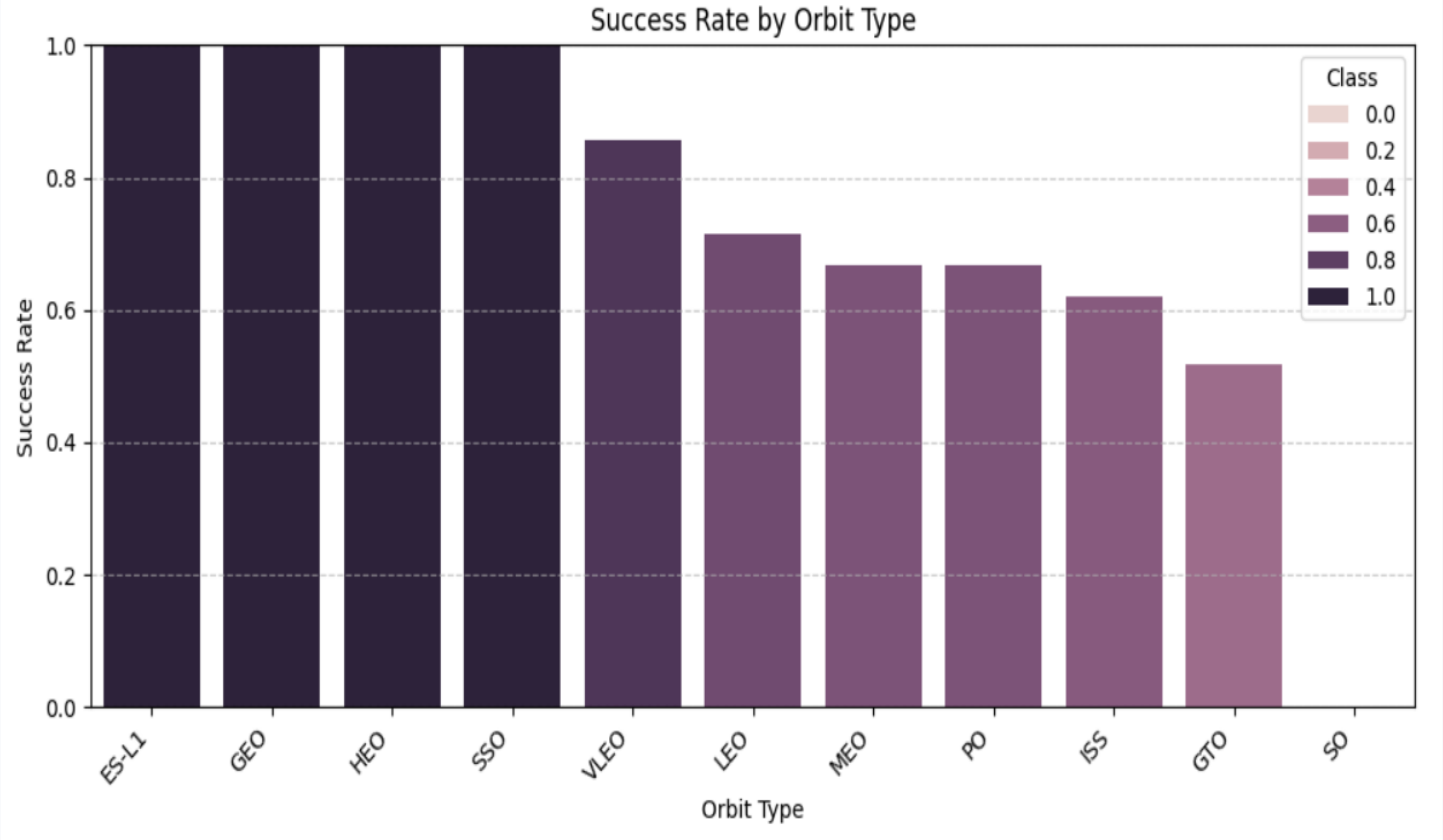
Key Insights:

- VAFB SLC 4E lacks heavy payload launches (>10,000 kg), suggesting it handles lighter payloads.
- KSC LC 39A and CCAFS SLC 40 show successful landings across various payload sizes, indicating versatility.

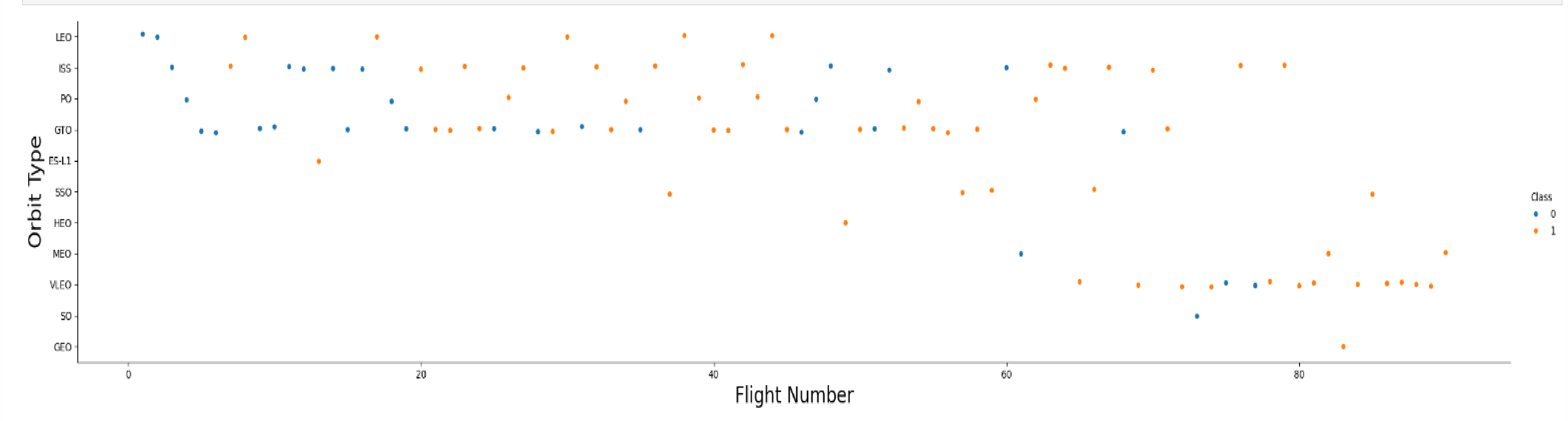
Success Rate vs. Orbit Type

Key Insights:

- ES-L1, GEO, HEO, SSO orbits have the highest success rate
- SO orbit has no success rate in landing.
- All the other orbits have moderate success rates in landings.



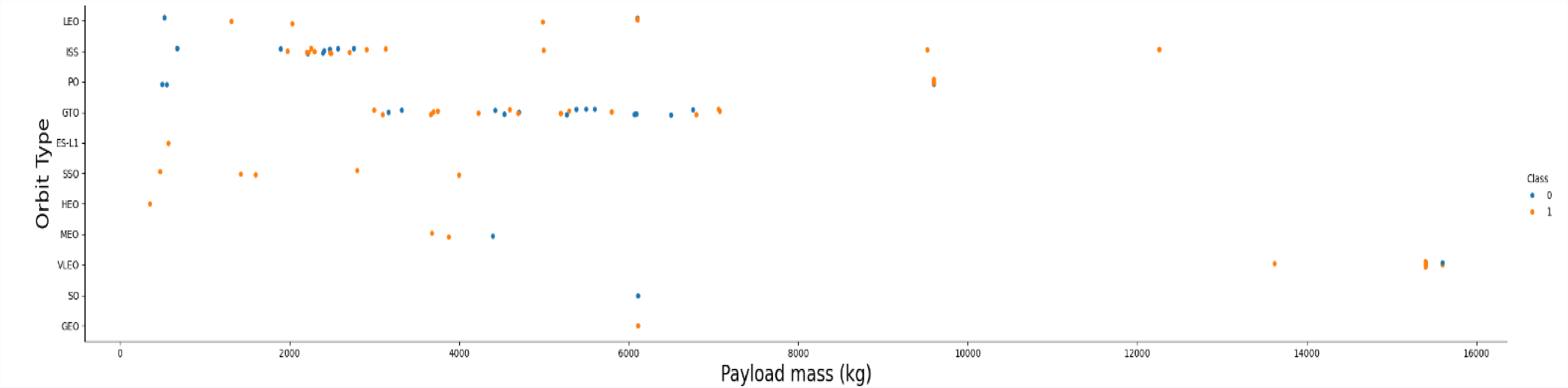
Flight Number vs. Orbit Type



Key Insights:

- LEO Orbit: Success rate increases with the number of flights.
- GTO Orbit: No clear relationship between flight number and success.

Payload vs. Orbit Type



Key Insights:

- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- For GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend

Key Insights:

- The success rate shows a steady increase from 2013 to 2020
- There is 0 success rate in landings from 2010 to 2013.

All Launch Site Names

- Keyword **distinct** queries unique launch site names from the SpaceX table.

[16]:
```
%sql select distinct(launch_site) from spacextable;
```

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

[16]:

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

Launch Site Names Begin with 'CCA'

- **SELECT *:** Retrieves all columns from the table.
- **FROM spacextable:** Specifies the table to query.
- **WHERE launch_site LIKE 'CCA%':** Filters records where the launch_site column starts with 'CCA'.
- **LIMIT 5:** Limits the result to 5 records.

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

Total Payload Mass

- **SELECT SUM(payload_mass__kg_):** Calculates the total payload mass.
- **FROM spacextable**: Specifies the table to query.
- **WHERE customer = 'NASA (CRS)':** Filters records where the customer is 'NASA (CRS)'.

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

```
%sql select SUM(payload_mass__kg_) from spacextable where customer = 'NASA (CRS)
```

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

SUM(payload_mass__kg_)
45596

Average Payload Mass by F9 v1.1

- **SELECT avg(payload_mass__kg_):** Calculates the average payload mass.
- **FROM spacextable**: Specifies the table to query.
- **WHERE booster_version LIKE '%F9 v1.1'**: Filters records where the booster version contains 'F9 v1.1'.

Display average payload mass carried by booster version 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_)
2928.4

First Successful Ground Landing Date

- **SELECT MIN(Date):** Retrieves the earliest date.
- **FROM SPACEXTABLE**: Specifies the table to query.
- **WHERE Landing_Outcome = 'Success (ground pad)'**: Filters records where the landing outcome was a successful ground pad landing.

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

Hint:Use min function

```
%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

- **WHERE "Landing_Outcome" = 'Success (drone ship)'**: Filters records where the landing outcome was a successful drone ship landing.
- **AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000**: Further filters records to include only those with payload mass between 4000 and 6000 kg.

Task 6

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 "Booster_Version" FROM spacextable WHERE "Landing_Outcome" = 'Success (drone ship)'
    AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

- **SELECT COUNT(mission_outcome):** Counts the total number of records in the mission_outcome column.
- **FROM spacextable**: Specifies the table to query.

List the total number of successful and failure mission outcomes

```
%sql select count(mission_outcome) from spacextable;
```

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

count(mission_outcome)
101

Boosters Carried Maximum Payload

- A subquery: **WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM spacextable):** Filters records to include only those with the maximum payload mass, identified by a subquery.

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

```
%%sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM spacextable
    WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM spacextable);
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- **SELECT SUBSTR(Date, 6, 2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site"**: Retrieves the month, landing outcome, booster version, and launch site columns, with the month extracted using SUBSTR.
- **WHERE "Landing_Outcome" LIKE 'Failure (drone ship)' AND SUBSTR(Date, 1, 4) = '2015':** Filters records for failures on drone ships in the 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.

```
%%sql SELECT SUBSTR(Date, 6, 2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site"
    FROM spacextable WHERE "Landing_Outcome" LIKE 'Failure (drone ship)' AND SUBSTR(Date, 1, 4) = '2015';
```

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

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

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

- **SELECT "Landing_Outcome", COUNT(*) AS count**: Retrieves the landing outcome and its count.
- **WHERE Date >= '2010-06-04' AND Date <= '2017-03-20':** Filters records between the specified dates.
- **GROUP BY "Landing_Outcome"**: Groups the results by landing outcome.
- **ORDER BY count DESC**: Orders the results by the count 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.

```
%%sql SELECT "Landing_Outcome", COUNT(*) as count FROM spacextable
    WHERE Date >= '2010-06-04' AND Date <= '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY count DESC;
```

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

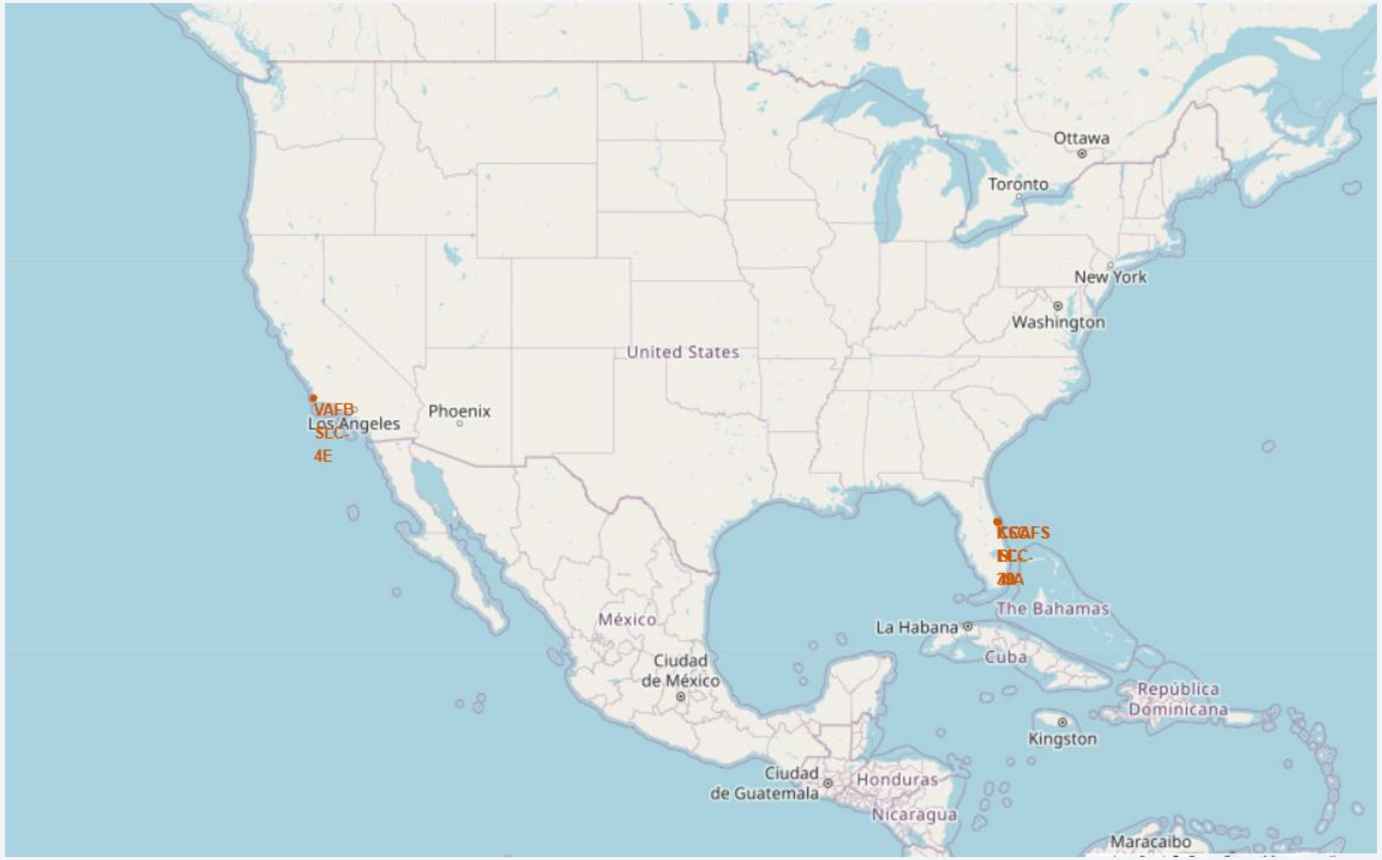
Landing_Outcome	count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

Launch Sites with Markers

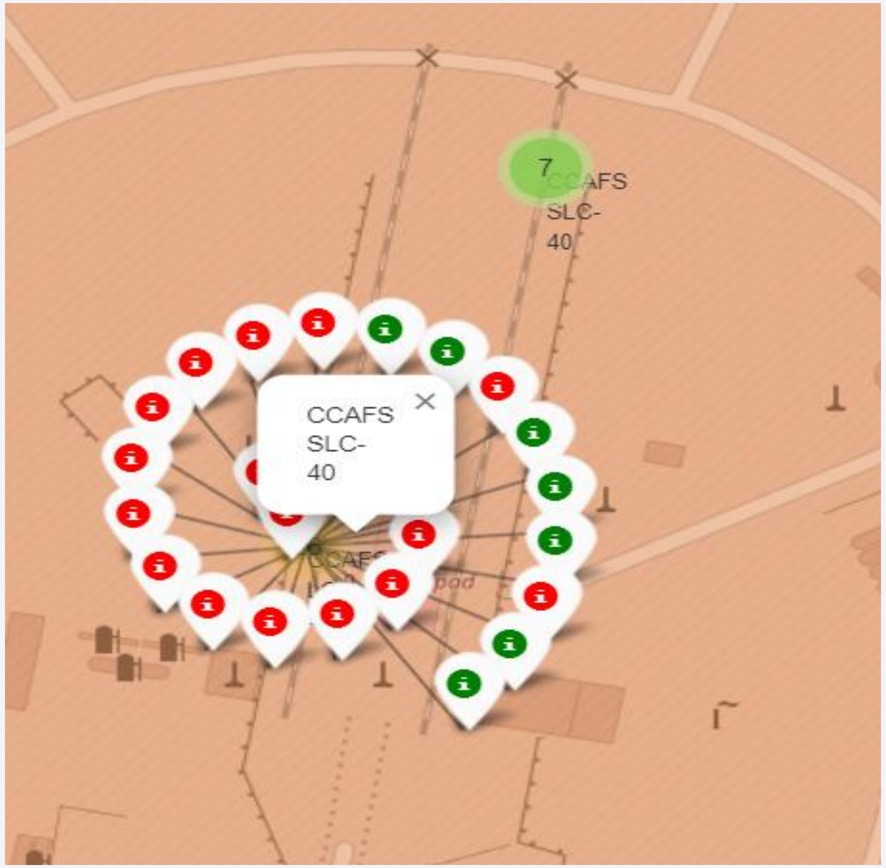
- **Near Equator:** The closer the launch site to the equator, the easier it is to launch to equatorial orbit, and the more help you get from Earth's rotation for a prograde orbit.
- Rockets launched from sites near the equator get an additional natural boost - due to the rotational speed of earth - that helps save the cost of putting in extra fuel and boosters.



Launch Outcomes at each Site

Outcomes:

- Green markers for successful launches
- Red markers for unsuccessful launches



Launch Site's Distance to Proximities

Distance to Proximities:

- 0.86 km from nearest coastline
- 21.96 km from nearest railway
- 23.23 km from nearest city
- 26.88 km from nearest highway

Safety:

- Coasts help ensure that spent stages dropped along the launch path or failed launches don't fall on people or property.

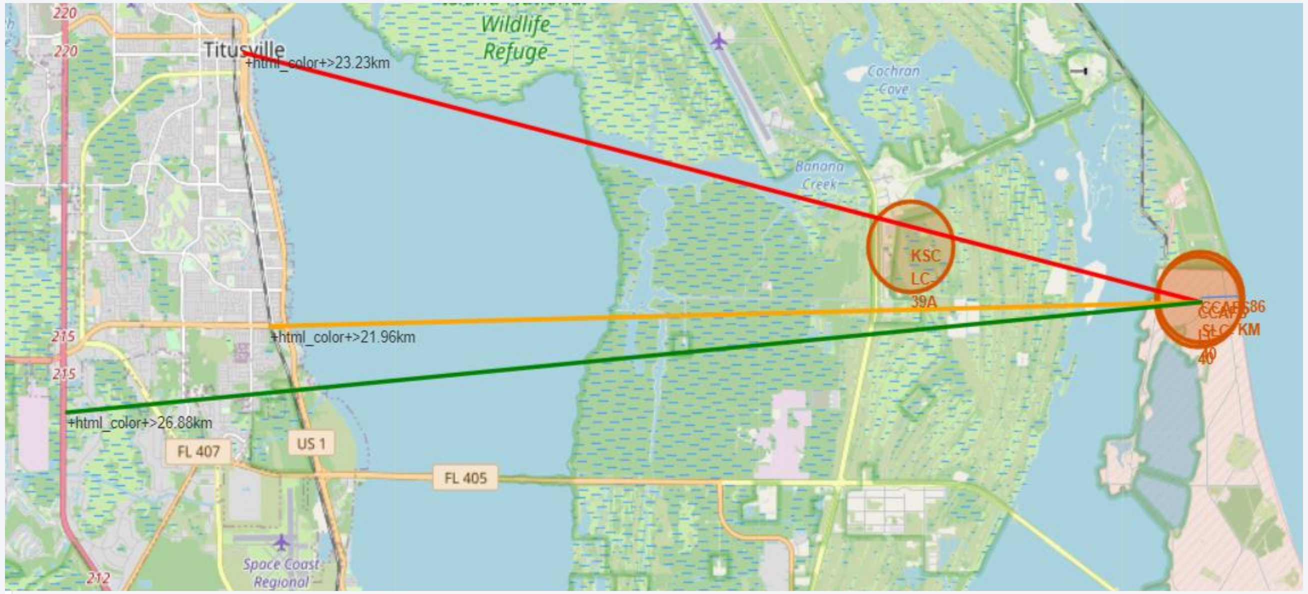
Proximity to Transportation/Infrastructure and Cities:

- Launch sites need to be away from anything a failed launch can damage, but still close enough to roads/rails/docks to be able to bring people and material to or from it in support of launch activities.

Proximity to the Equator:

- The closer a launch site is to the equator, the easier and more cost-effective it is to reach equatorial orbits. Earth's rotation provides a natural boost for prograde orbits, reducing the need for extra fuel and boosters.

Launch Site's Distance to Proximities (Code & Image)

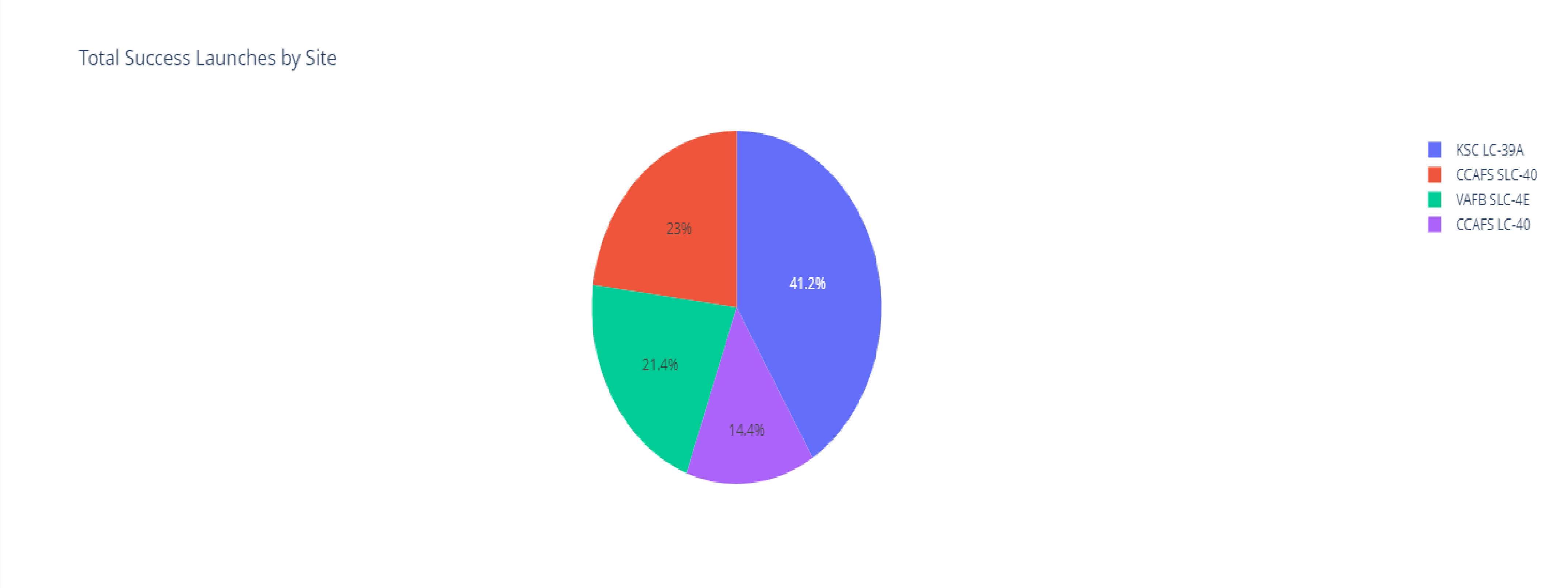



```python
# Create a marker with distance to a closest city, railway, highway, etc.
# Draw a line between the marker to the launch site
launch_coordinate = [28.562302, -80.57682]

city   = [28.61200, -80.80788]
railway = [28.55752, -80.80155]
highway   = [28.5402, -80.85079]

city_distance = calculate_distance(city[0], city[1], launch_coordinate[0], launch_coordinate[1])
railway_distance = calculate_distance(railway[0], railway[1], launch_coordinate[0], launch_coordinate[1])
highway_distance = calculate_distance(highway[0], highway[1], launch_coordinate[0], launch_coordinate[1])
```

Section 4

Build a Dashboard with Plotly Dash

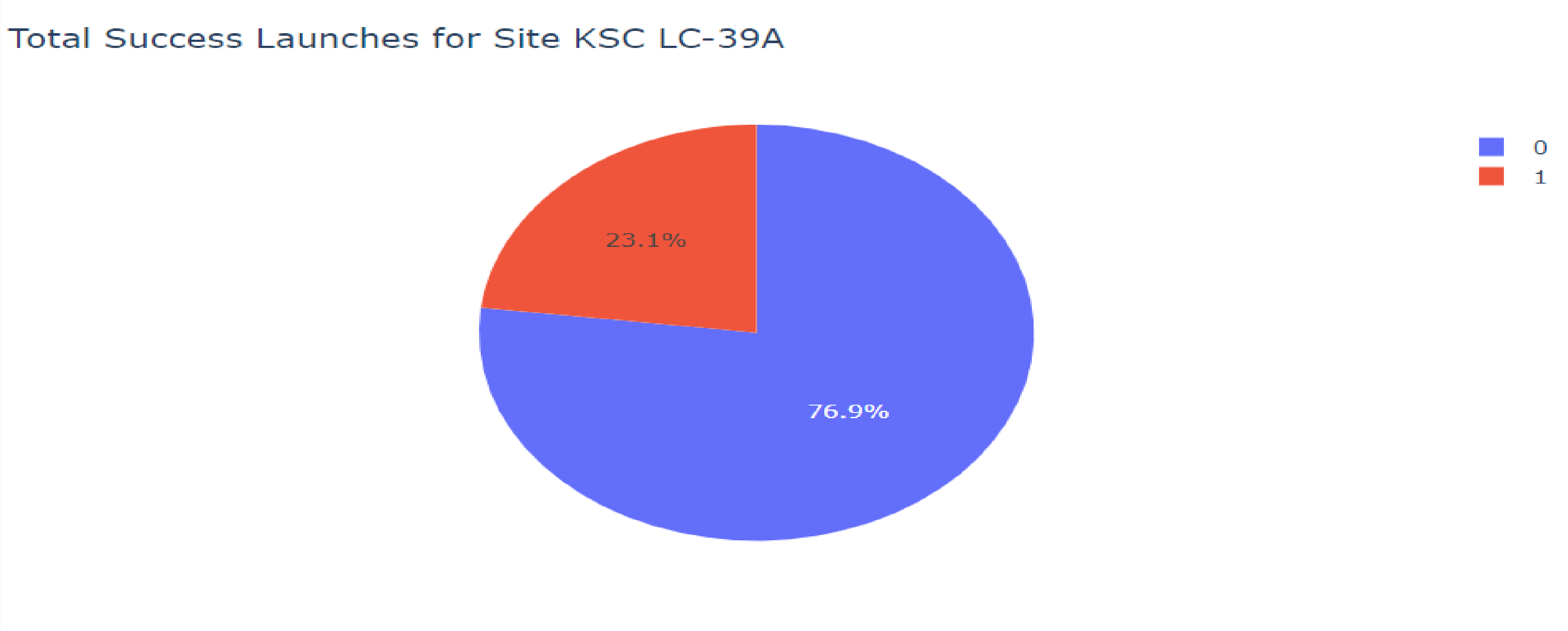
Launch Success by Site



Success as Percent of Total:

- KSC LC-39A has the most successful launches amongst launch sites (41.2%)

Launch Success (KSC LC-29A)



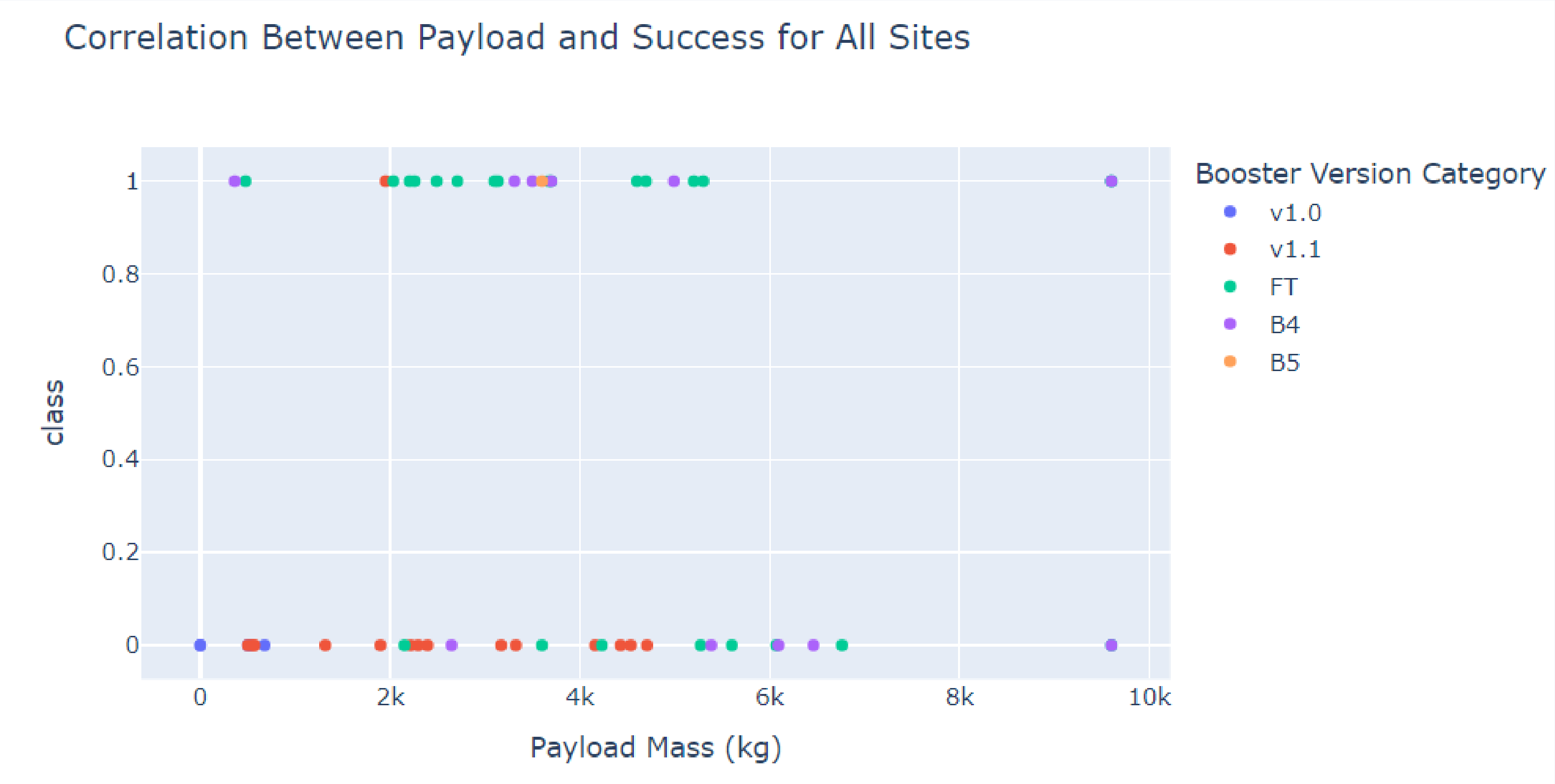
Success as Percent of Total:

- KSC LC-39A has the highest success rate amongst launch sites (76.9%)
- 10 successful launches and 3 failed launches

Payload Mass and Success

By Booster Version:

- Payloads between 2,000 kg and 5,000 kg have the highest success rate
- 1 indicating successful outcome and 0 indicating an unsuccessful outcome



Section 5

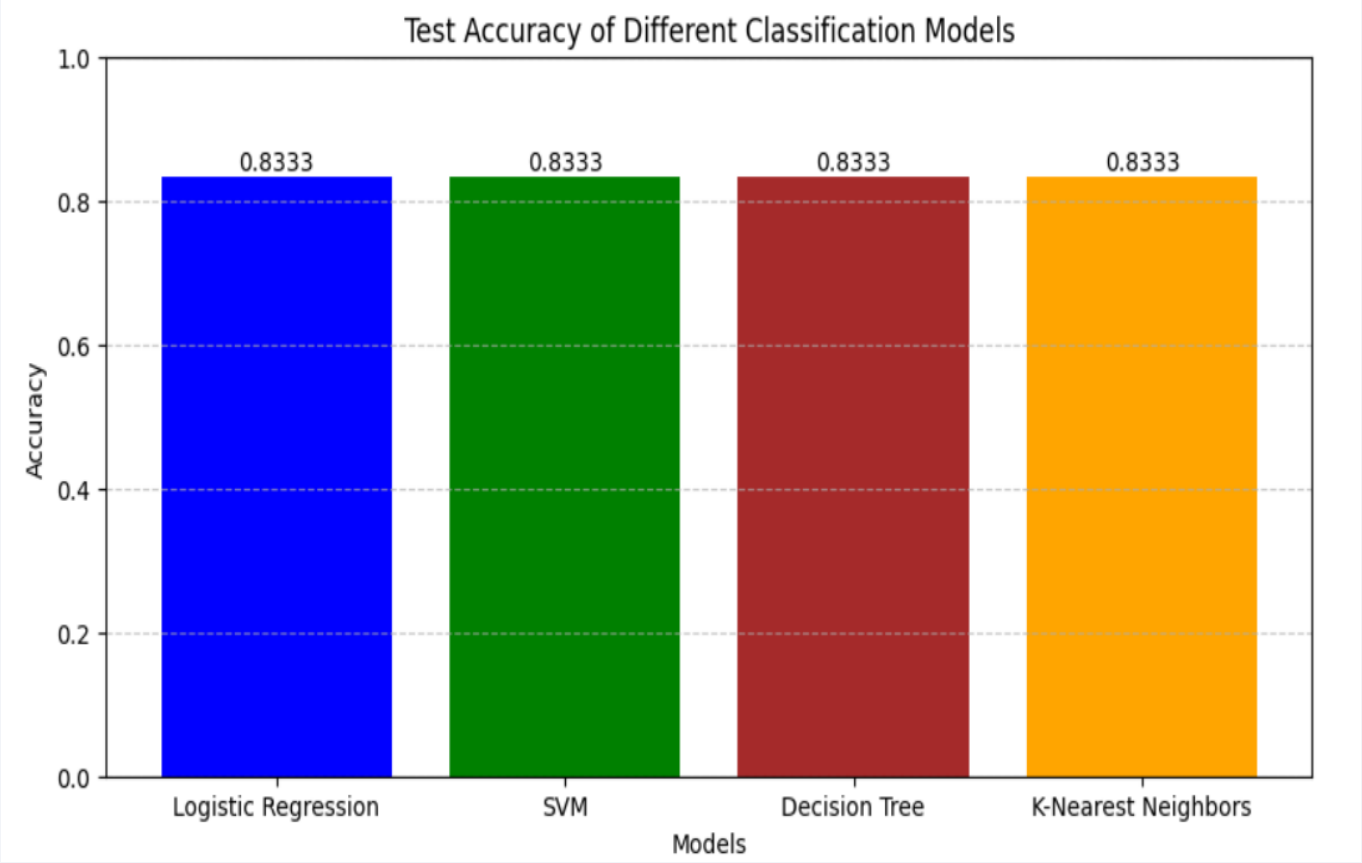
Predictive Analysis (Classification)

Classification Accuracy

All models have the same test accuracy of **0.8333**, meaning that, according to the test data, they all perform equally well.

The Decision Tree model had the highest **validation accuracy of 88.93%**, which means it was the most accurate on the validation set.

However, its test accuracy is the same as that of Logistic Regression and SVM. This discrepancy may suggest that the model overfit to the training data, capturing noise rather than general patterns.



Confusion Matrix

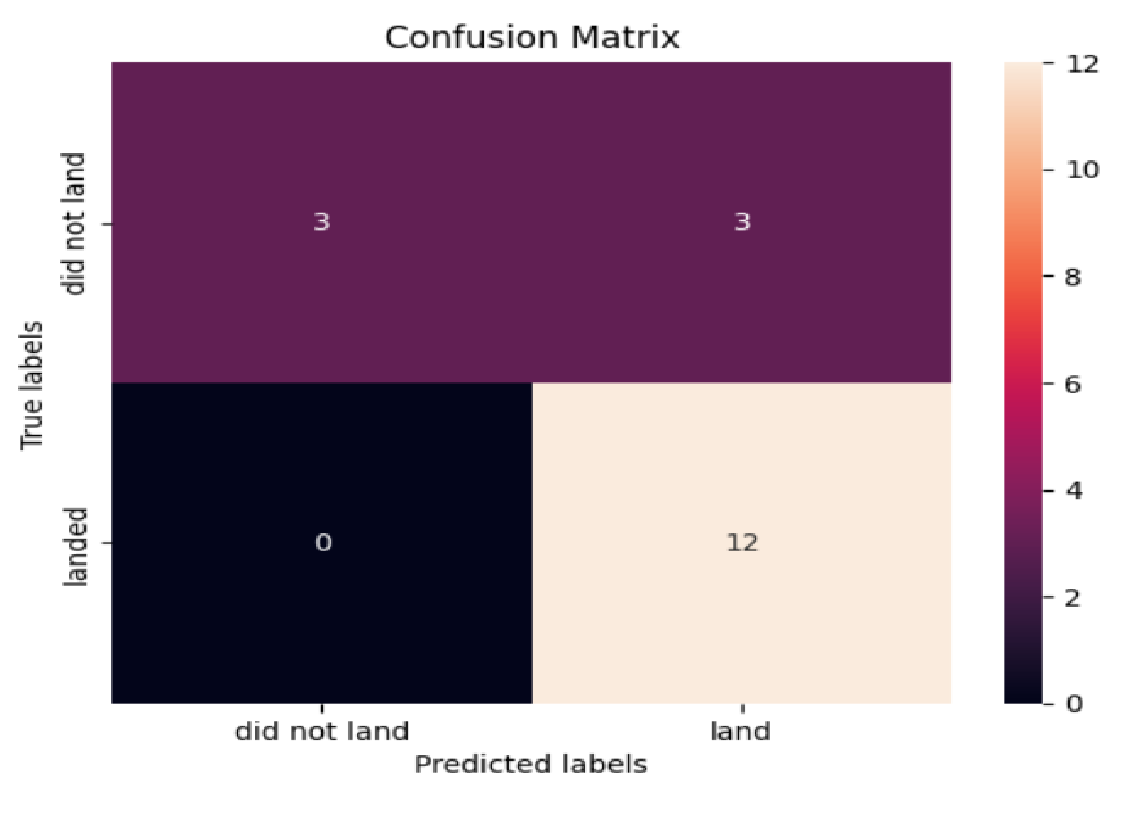
The confusion matrix for the Decision Tree model shows:

- True Positives (TP): 12
- False Positives (FP): 3
- False Negatives (FN): 3

Summary:

- **Precision:** 80% (proportion of positive predictions that are correct)
- **Recall:** 80% (proportion of actual positives identified correctly)
- **F1 Score:** 80% (a balanced measure that combines precision and recall)
- The model performs well with balanced precision and recall, but further tuning could enhance its accuracy.

```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

Research:

- Model Performance: The models performed similarly on the test set with the decision tree model slightly outperforming
- Equator: Most of the launch sites are near the equator for an additional natural boost - due to the rotational speed of earth - which helps save the cost of putting in extra fuel and boosters
- Coast: All the launch sites are close to the coast
- Launch Success: Increases over time
- KSC LC-39A: Has the highest success rate (100%) among launch sites.
- Orbits: ES-L1, GEO, HEO, and SSO have a 100% success rate
- Payload Mass: Across all launch sites, the higher the payload mass (kg), the higher the success rat

Thank you!