IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Aaron Rawago
7-Jun-2023

Outline

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

Executive Summary

Summary of Methodologies:

- Data Collection: First step was to collect data using a get request to the SpaceX API. The data was then subjected to data wrangling and formatting to ensure that it was in a usable format.
- Exploratory Data Analysis (EDA): EDA was performed to determine the training labels. This involves analyzing the data to find patterns, trends, and relationships that can be used to build models.
- SQL Queries: SQL queries were executed to gain better insights into the datasets stored in the database. This helps to identify trends and patterns in the data that may not be immediately apparent.
- Feature Engineering: EDA and feature engineering were performed using Pandas and Matplotlib to further refine the data for modeling. This involves transforming the data to create new features that may be more informative or useful for building the prediction models.
- Geospatial Analysis: Geographical patterns about launch sites were found using Folium, which is a Python library used for visualizing geospatial data.
- Dashboard Visualization: Interactive real-time dashboards were built using Plotly Dash to visualize the data and provide insights into trends and patterns.
- Prediction Modelling: Finally, classification models were built, trained, and tested to determine the best performance. This involves applying machine learning algorithms to the data to predict outcomes based on a set of input features.

Summary of all results:

- Data Collection: The data was collected, cleaned, formatted, and exported to a CSV file.
- Data Analysis: The data was analyzed, labeled with dependent and target variables, and split into training and testing sets. Maps, charts, and plots were created to gain insights into launch sites, landing success rates, payload mass, and booster versions.
- Predictive Models: Several models were built and evaluated to determine the best models, best accuracy, and confusion matrix. The goal of these models was to predict outcomes based on input variables and provide insights into the relationships between variables in the dataset.

Introduction

- **Project background and context**

SpaceX has emerged as a leader in the commercial space travel industry by making space travel more affordable through their innovative use of reusable rocket technology. One key factor that contributes to SpaceX's success in achieving this goal is their ability to reuse the first stage of their rockets, which significantly reduces the cost of launches. As a company in the same industry, SpaceY aims to understand and replicate this success. Therefore, this project will use SpaceX as a case study to analyze their approach to launching rockets, specifically focusing on the success of their first stage landings.

- **Problems we want to find answers**

The primary objective of this project is to help SpaceY replicate SpaceX success strategy through analysis and building predictive models that can determine successful launches of future Falcon 9 rockets. We will collect, clean, and analyze the data, identifying trends and insights to help us build several classification models. By doing so, we aim to develop an accurate and reliable model that can predict whether the first stage of a Falcon 9 rocket will land successfully. Through this project, we hope to gain insights into the factors that contribute to successful first stage landings, and to apply these insights to inform our own approach to rocket launches as a company in the commercial space travel industry.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  - Use get requests to fetch data from an API and load to csv and pandas DataFrames

- Perform data wrangling

  - Address Missing Values through replacement by the Mean value

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Use of One-Hot Encoding to build class features
  - Applying transformations for data standardization
  - Build Logistic Regression, Decision Trees, K-NN and SVM Models
  - Test each Model’s accuracy using the GridSearchCV

Data Collection

Description of how data sets were collected:

- Import Libraries and define functions
- Request rocket launch data from SpaceX API with URL
- Request & Parse SpaceX launch data using GET Request
- Decode the data and turn it into a Pandas dataframe
- Filter the dataframe to only include our target variable
- Deal with missing values and replace them
- Export the cleaned data into CSV

Data Collection – SpaceX API

- SpaceX Route: https://api.spacexdata.com/v4/
- SpaceX endpoint: launches/past
- We create user-defined functions that connect to the specific endpoints of the url, and fetch the data using get requests as below:

  - response = requests.get(url)

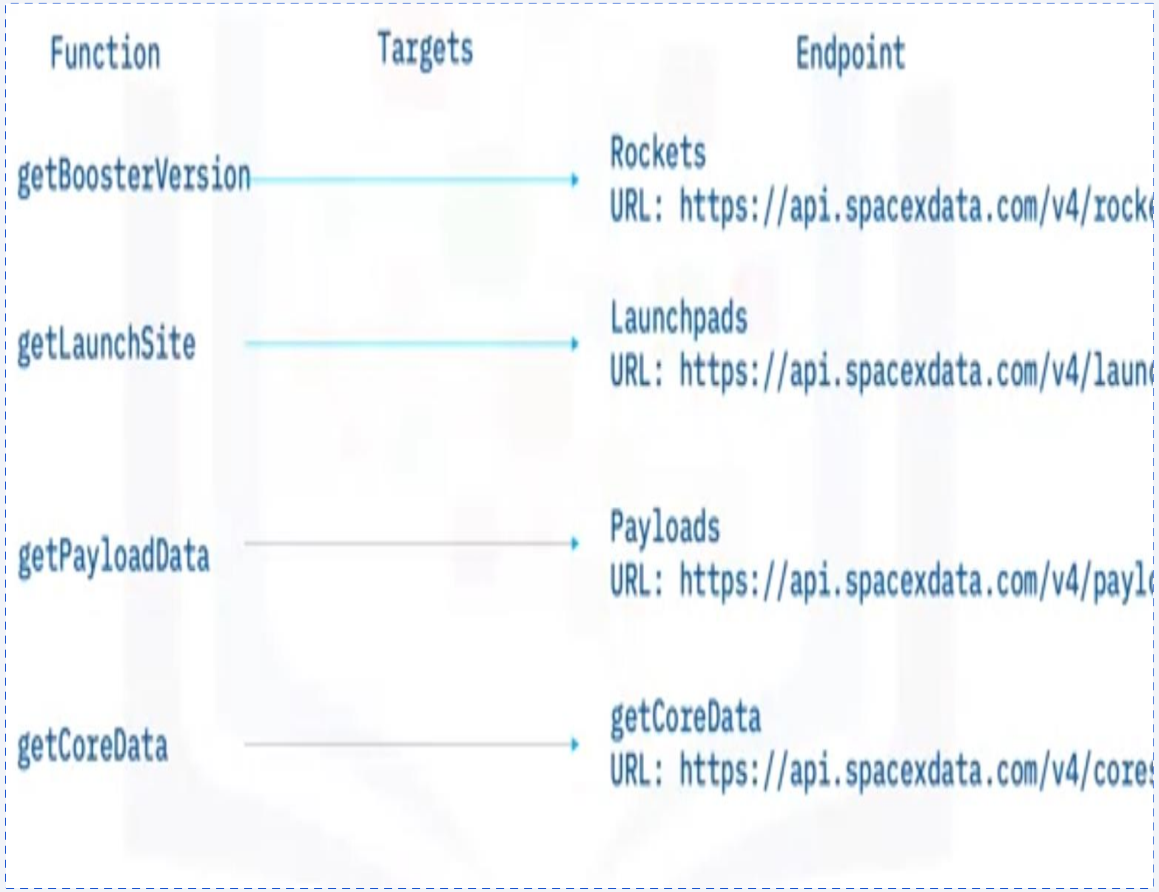
- Where 'url' is the 'route+endpoint'
- The response from the url is a json list of objects that needs to be converted to a pandas dataframe using the json_normalize() function.

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

- The pandas dataframe can be stored in local disk using the to_csv() function

GitHub URL of the completed SpaceX API calls notebook:

jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Data can be collected from Web Tables through webscrapping. We obtain Falcon 9 launch records with BeautifulSoup:

- We Create a BeautifulSoup object that takes in the response text from the get request as Input

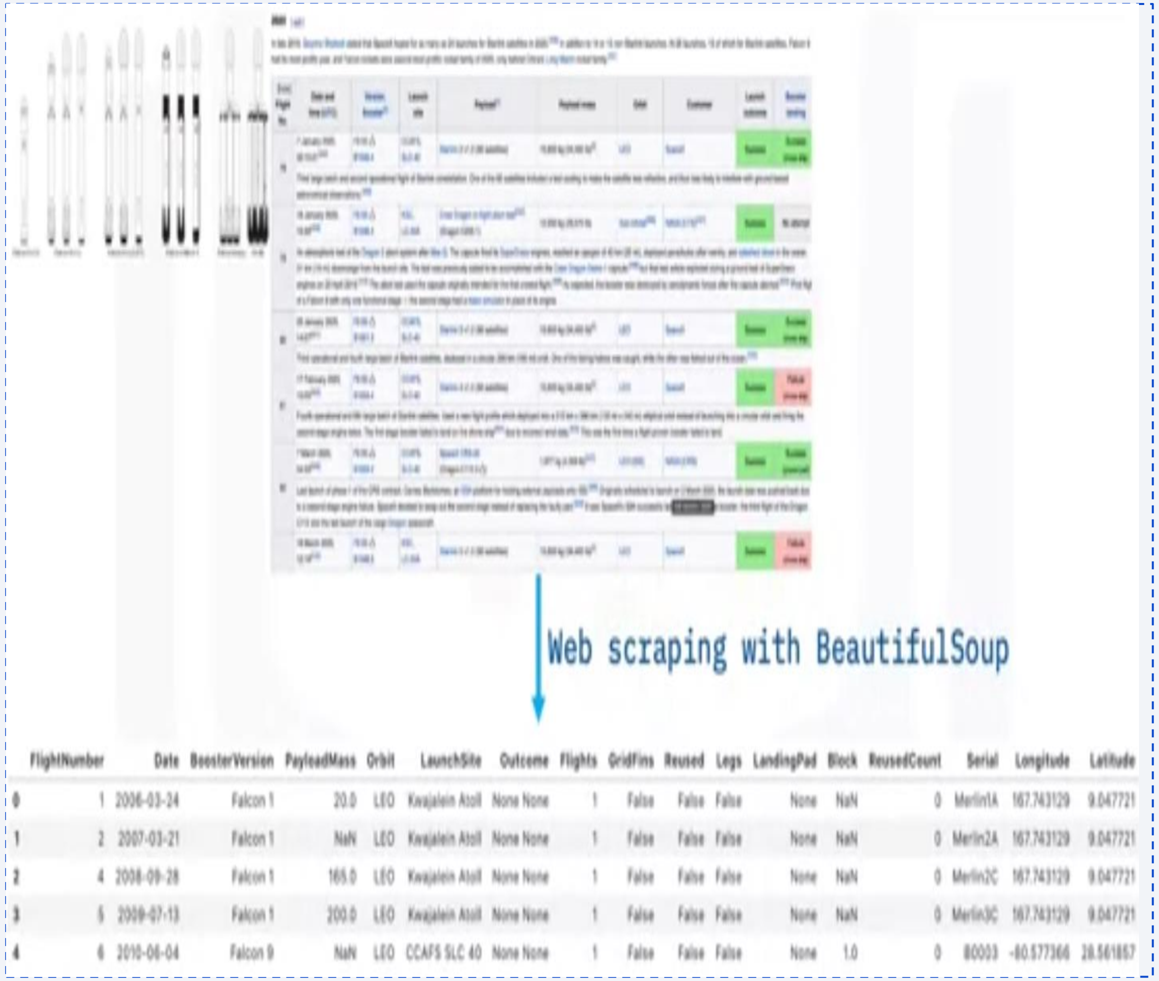
  soup = BeautifulSoup(response.text, html.parser)

- Next, we extract a Falcon 9 launch records HTML table from Wikipedia

  data = soup,find_all('tbody')

- Then, we parse the table and convert it into a Pandas data frame

  df = pd.DataFrame(launch_dict)



GitHub URL of your completed webscrapping notebooks: [jupyter-labs-webscraping.ipynb]

Data Wrangling

Description of how data was processed:

- We performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.
- We will mainly convert the landing outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.
- We calculated the number of launches that have taken place on each site.
- We calculated the number of occurrences of each orbit.
- We also calculated the number of occurrences of mission outcomes per orbit type.
- Finally, we created a landing outcome label from the outcome column to be used as the target categorical variable.

GitHub URL of your completed data wrangling related notebooks:

labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Summary of charts plotted and why we used those charts:

- We plotted an number of scatter plots, bar charts and line graphs to visualize the relationship between various components of data.

1. To visualize the relationship between flight number and launch site, we plotted a scatter plot that showed higher launch success rate at VAFB SLC 4E.
2. To visualize the relationship between payload and launch site, we plotted a scatter plot that showed no rockets launched for heavypayload mass(greater than 10000) at VAFB-SLC launch site.
3. We plotted a scatterplot to visualize the relationship between flight number and ortbit type. It showed 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.
4. We plotted a barchart to visualize the relationship between success rate of each orbit type. It that showed that High mission success rates were observed for orbits ES-L1, GEO, HEO, SSO, and VLEO.
5. We plotted a scatterplot to visualize the relationship between payload and orbit type. It showed 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.
6. We also plotted a line graph for the launch success yearly trend that showed that the success rate since 2013 kept increasing till 2020

GitHub URL of completed EDA with data visualization notebook: [jupyter-labs-eda-dataviz.ipynb]

EDA with SQL

Summary of the SQL queries performed:

- Understanding the SpaceX dataset through SQL Queries
- Display the names of unique launch sites to explore the data.
- Display the records of launch sites with CCA.
- Display the total payload mass carried by boosters launched by NASA (CRS).
- Display the average payload mass carried by booster version F9 V1.1.
- List the date when the first successful landing outcome in the past was achieved.
- Display the names of boosters with success in drone ships with payload between 4000-6000.
- List the total number of successful and failure mission outcomes.
- Display the names of booster versions that have carried the maximum payload mass.
- Query the drone failure outcome, booster version, and launch site for 2015.
- Rank the successful landing outcomes from June 2010 to March 2017.

GitHub URL of completed EDA with SQL notebook: jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Summary of map objects such as markers, circles, lines, etc. created and added to a folium map

- To understand the ideal location for building a launch site we analysed SpaceX existing launch site locations and whether the location and proximity of launch sites affected the success rate of a launch
- We build a Folium map object with NASA launch site coordinates as the center.
- We use the folium marker object to mark all launch sites on the map.
- We marked successful and failed mission sites on the map using a different marker.
- We then calculated the distance between launch site locations and their closest proximity to highways, railroads, coastlines, and cities.
- We then use the insights obtained from the map to draw conclusions about the suitability of launch site locations.

GitHub URL of completed interactive map with Folium map:

lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Summary of plots/graphs and interactions added to the dashboard:

- We create an input component for the dashboard with dropdown lists and range sliders to display the pie chart and scatter point chart
- We then include a dropdown input component to select the launch site
- Then implemented a callback function that generates a "Success pie chart" based on the selected launch site dropdown
- We also included a range slider to select the payload
- We added a callback function to display the "Success payload scatter plot" based on the selected payload range
- On Completion, we launched the interactive web dashboard using a private IP and port: 127.0.0.1 / 8050.

GitHub URL of completed Plotly Dash lab: lab_jupyter_launch_site_plotly_Dashboard.ipynb

Predictive Analysis (Classification)

Summarize on how we built, evaluated, improved, and found the best performing classification model

- Perform exploratory data analysis to determine the training labels, standardize the data, split it into training and test sets for classification, and test the models for accuracy to find the best performing model:
- Define a function to plot the confusion matrix.
- Create a NumPy array with the "Class" column and assign it to the variable Y.
- Standardize the data in X and assign it to the variable X.
- Split the data into training and testing sets.
- Create a logistic regression object and a GridSearchCV object to find the best parameters: {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}.
- Create an SVM object and a GridSearchCV object to find the best parameters: {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}.
- Create a decision tree classifier object and a GridSearchCV object to find the best parameters: {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitter': 'random'}.
- Create a KNN object and a GridSearchCV object to find the best parameters, which are {'algorithm': 'auto', 'n_neighbors': 10,'p': 1}.
- Compare all the models to determine the best performing one

GitHub URL of completed predictive analysis lab: [SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb]

Results

Exploratory data analysis results

- Based on the scatter plot of Flight number Vs Payload mass, it appears that different launch sites have varying success rates. For instance, the CCAFS LC-40 site has a success rate of 60%, while KSC LC-39A and VAFB SLC 4E have a success rate of 77%.
- The Payload Vs. Launch Site scatter point chart shows that there are no rockets launched for heavy payload mass (greater than10000) at the VAFB-SLC launch site.
- The scatter plot between orbit and Flight number suggests that in the LEO orbit, the success rate appears related to the number of flights. However, there seems to be no relationship between flight number when in GTO orbit.
- The scatter plot between Orbit and Payload mass reveals that for heavy payloads, the successful landing or positive landing rateis higher for Polar, LEO, and ISS. However, for GTO, it is difficult to distinguish as both positive landing rate and negative landing (unsuccessful mission) are present.

Predictive analysis results

- Based on the predictive analysis, we used the train_test_split function to split the data into 72 training samples and 18 test samples, with a test size of 0.2 and random state of 2.
- We then used four different classification models -Logistic regression, SVM, Decision tree, and KNN -and obtained their best parameters using GridSearchCV.
- We then plotted a confusion matrix for each model to visualize its performance.
- Findings: All models had the same accuracy on the test data, which was 0.8333333333333334. This suggests that all four models are similarly effective at predicting the outcome variable.

Section 2

Insights drawn from EDA

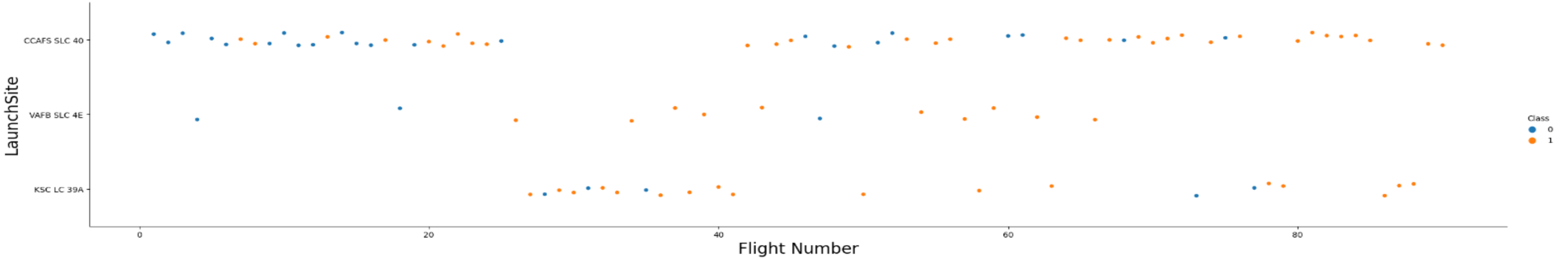
Flight Number vs. Launch Site

Scatter plot of Flight Number vs. Launch Site

TASK 1: Visualize the relationship between Flight Number and Launch Site

Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

In [4]:
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

Launches from flight number 18 and onwards had a higher success rate at the VAFB SLC launch site.

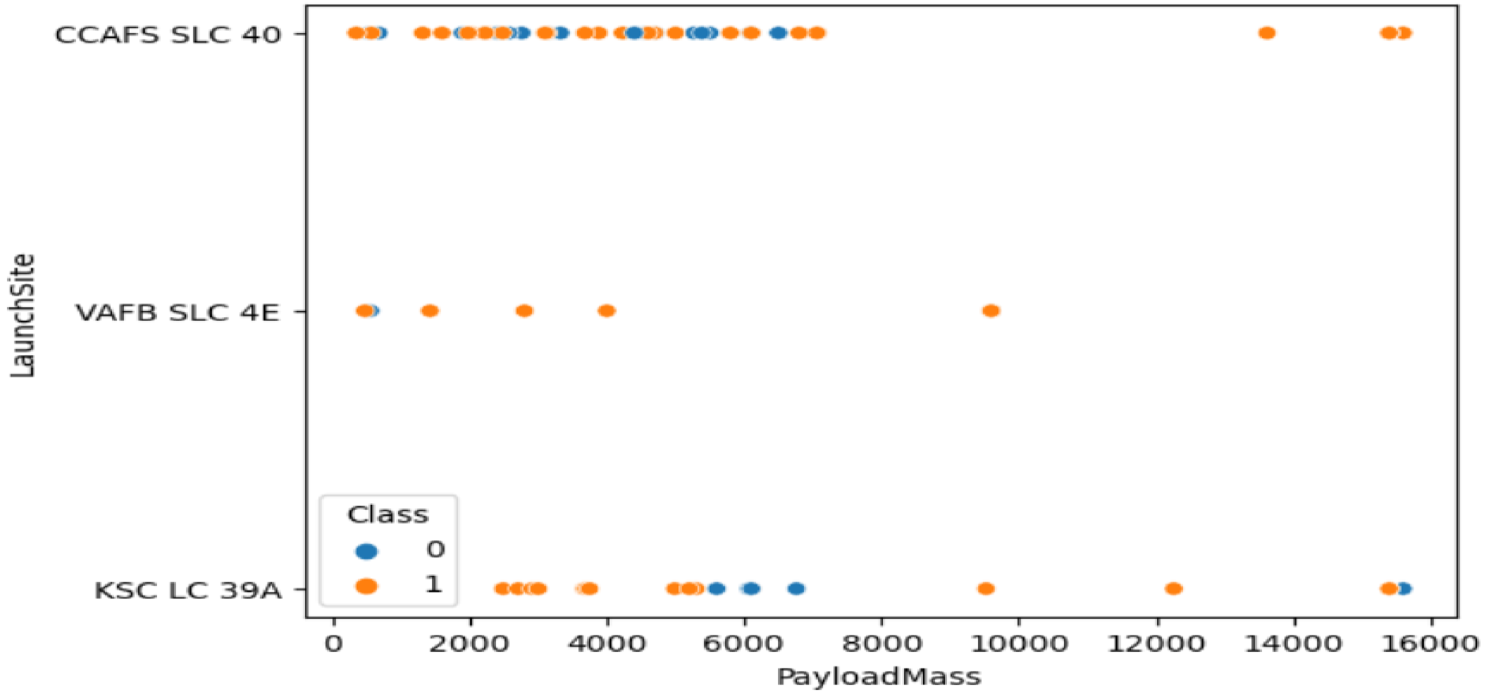
Payload vs. Launch Site

Scatter plot of Payload vs. Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

[9]:

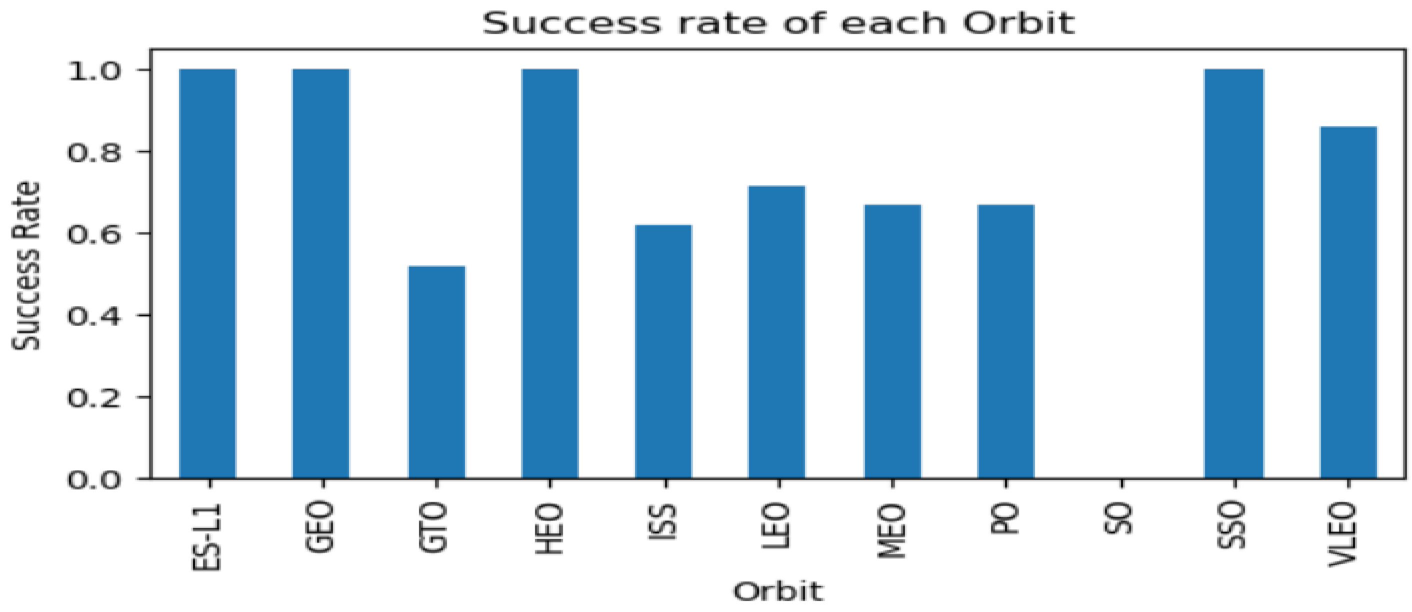
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and 
sns.scatterplot(data=df, x="PayloadMass", y="LaunchSite", hue="Class")
plt.show()
```



VAFB SLC had success with payload mass from 1000, while KSC had no success at +/-6000.

Success Rate vs. Orbit Type

Bar chart for the success rate of each orbit type



Analyze the ploted bar chart try to find which orbits have high sucess rate.

- High mission success rates were observed for orbits ES-L1, GEO, HEO, SSO, and VLEO

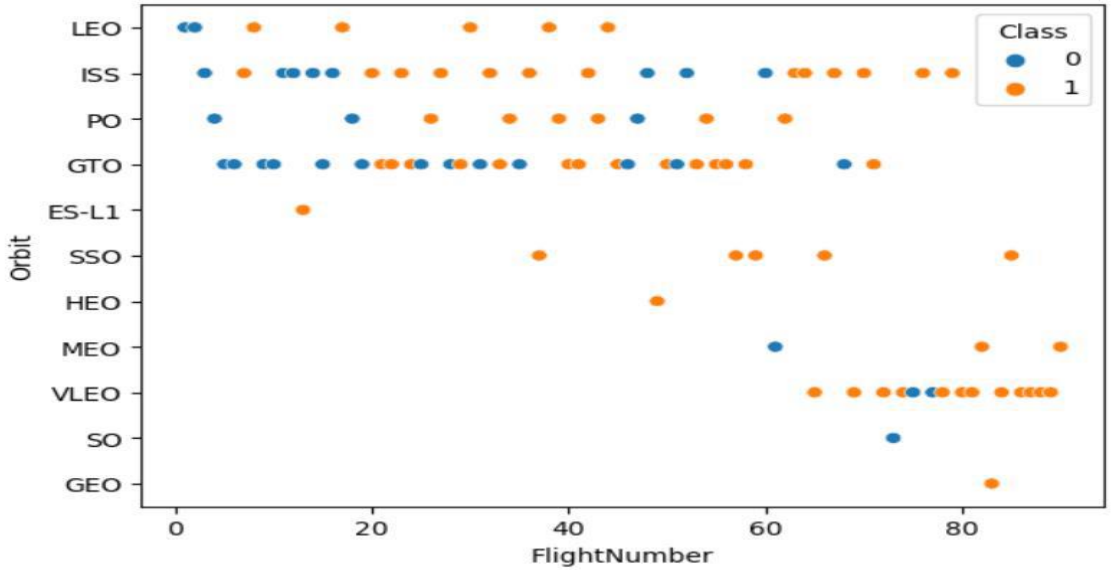
Flight Number vs. Orbit Type

Scatter point of Flight number vs. Orbit type

For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

In [34]:

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to
sns.scatterplot(data=df, x="FlightNumber", y="Orbit", hue="Class")
plt.show()
```



- There appears to be a relationship between success and flight number in the LEO orbit, while no such relationship exists in the GTO orbit.

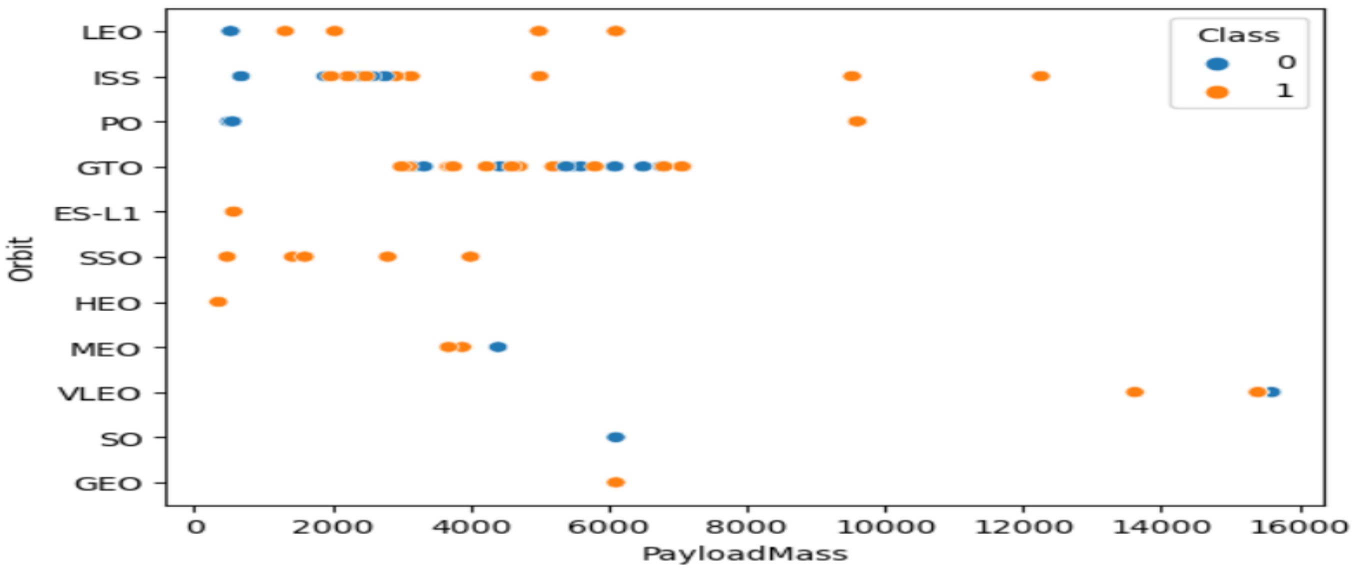
Payload vs. Orbit Type

Scatter point of payload vs. orbit type

Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type

In [60]:

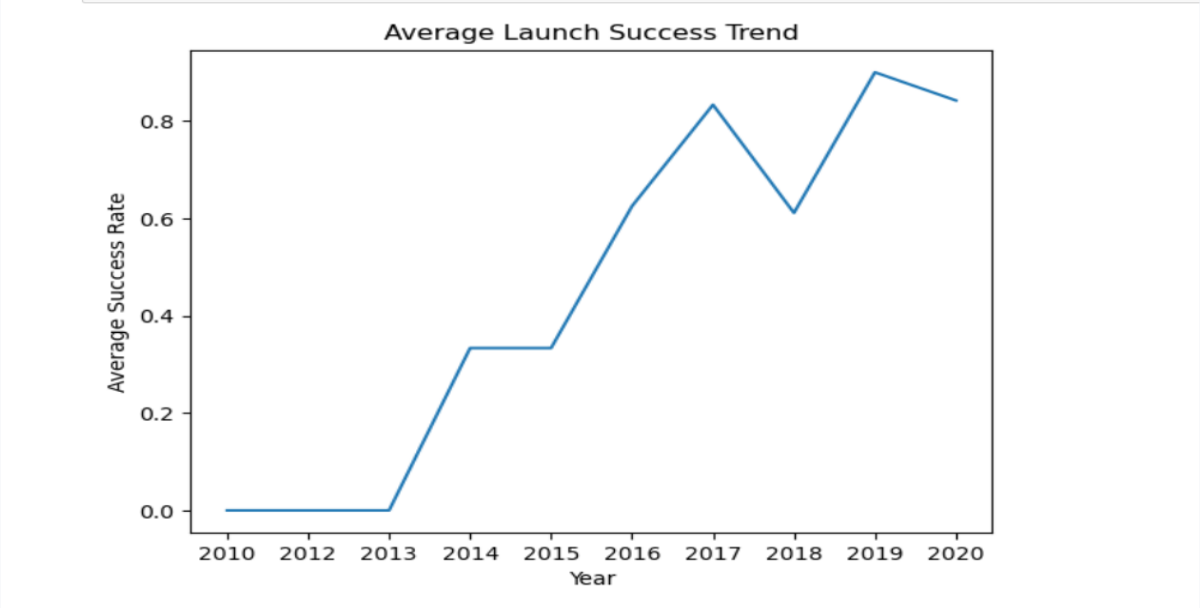
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the
sns.scatterplot(data=df, x="PayloadMass", y="Orbit", hue="Class")
plt.show()
```



- For heavy payloads, successful landings are more likely for Polar, LEO, and ISS orbits.
- However, for GTO orbit, it is difficult to distinguish between successful and unsuccessful missions.

Launch Success Yearly Trend

Line chart of yearly average success rate



- Success rates have been consistently increasing since 2013 until 2020.

All Launch Site Names

Unique launch sites:

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

```
%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
None

Space X has 4 Unique Launch Sites

Launch Site Names Begin with 'CCA'

5 records where launch sites begin with `CCA`

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

```
%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
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Total payload carried by boosters from NASA

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

```
%sql select sum(PAYLOAD_MASS__KG_) AS "Total Payload Mass (KG)" from spacextbl where customer = 'NASA (CRS)'
```

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

Total Payload Mass (KG)
45596.0

- 45,596 KG is to Total payload mass carried by booster from NASA (CRS)

Average Payload Mass by F9 v1.1

Average payload mass carried by booster version F9 v1.1

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

```
%sql select avg(payload_mass__kg_) AS 'Average Payload Mass (KG)' from spacextbl where Booster_Version like 'F9 v1.1%'
```

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

Average Payload Mass (KG)
2534.6666666666665

2,534.66 KG is the average payload mass carried by F9 v1.1 Boosters

First Successful Ground Landing Date

Dates of the first successful landing outcome on ground pad

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

Hint:Use min function

```
%sql select date as date from spacextbl where landing_outcome = 'Success (ground pad)' order by landing_outcome DESC Limit 1
```

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

date
22/12/2015

22nd Dec, 2015 was the first ever successful landing outcome on the ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000

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

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

```
%sql select 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

Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

```
%sql select mission_outcome,count(mission_outcome) as 'Total Number' from spacextbl group by mission_outcome order by count(mission_outcome) Desc
```

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

Mission_Outcome	Total Number
Success	98
Success (payload status unclear)	1
Success	1
Failure (in flight)	1
None	0

There was a high percentage of Successful Mission Outcomes than Failed Missions

Boosters Carried Maximum Payload

List of the names of boosters which have carried the maximum payload mass

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

```
%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

2015 Launch Records

List of failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql

select substr(date,4,2) as month, booster_version, launch_site, landing_outcome
from (select * from spacextbl where substr(date,7,4) = '2015')
where landing_outcome = 'Failure (drone ship)'
```

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

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

- In the months of Apr and Oct'2015, there were 2 Failed Drone ship landings, both at CCAFS LC-40 Landing Site.
- Booster Version F9 v1.1 B1012 and B1015 each had a failed landing in 2015

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

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

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%%sql

select landing_outcome, count(landing_outcome) As 'Total' from spacextbl
where julianday(substr(date,7,4)||"-"||substr(date,4,2)||"-"||substr(date,1,2)) between julianday('2010-06-04') and julianday('2017-03-20')
group by landing_outcome
order by count(landing_outcome) DESC
```

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

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

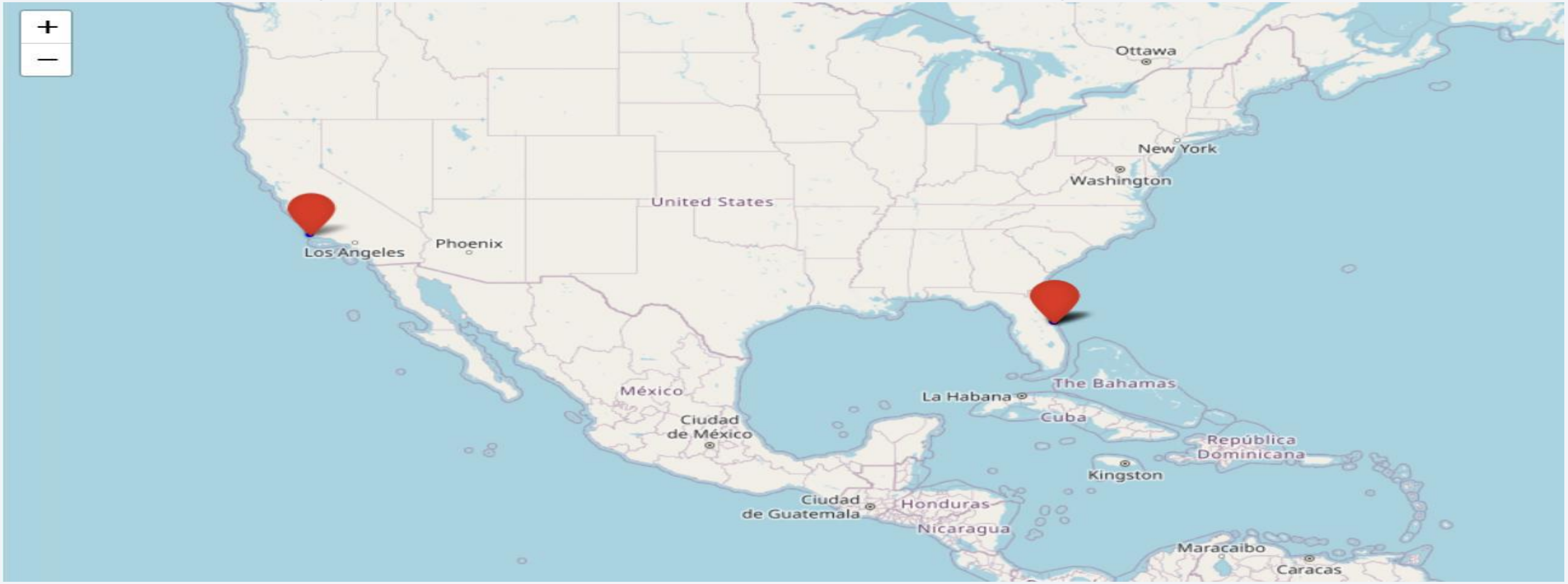
- There was an equal number of successful ground pad and drone ship landings
- There was a high number of “No Attempted’ landings in this period

Section 3

Launch Sites Proximities Analysis

Launch Site Locations on Map

Folium map showing all launch sites' location markers on a global map

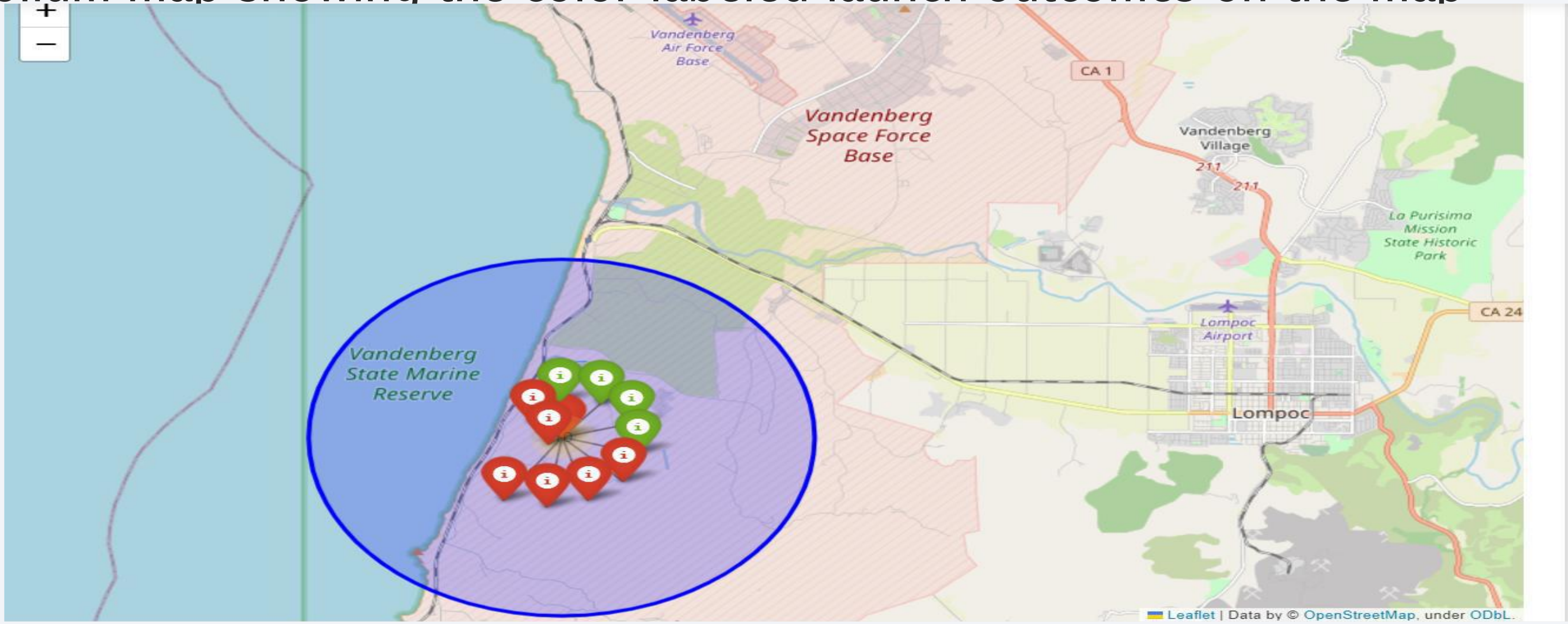


Important elements and findings on the screenshot:

- Launch Sites are located in the coastlines and never inland

Color-labeled Launch Outcomes on Map

Folium map showing the color-labeled launch outcomes on the map

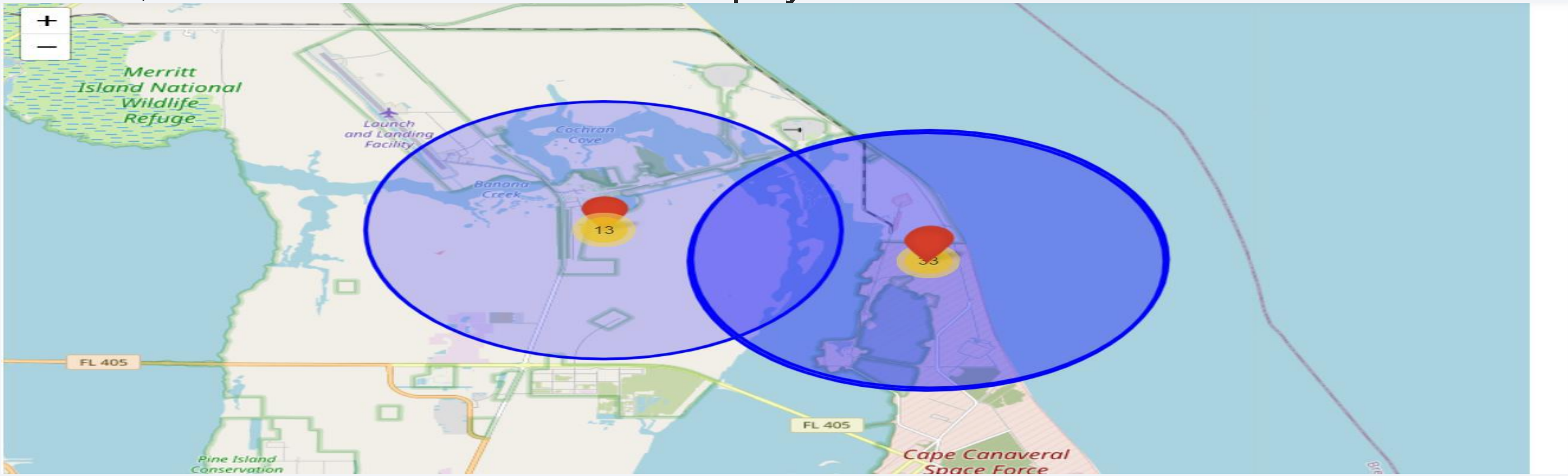


Important elements and findings on the screenshot:

- Folium map with clusters and markers indicating successful and failed mission launch sites

Launch Site Proximity

Folium map showing a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed



Important elements and findings on the screenshot:

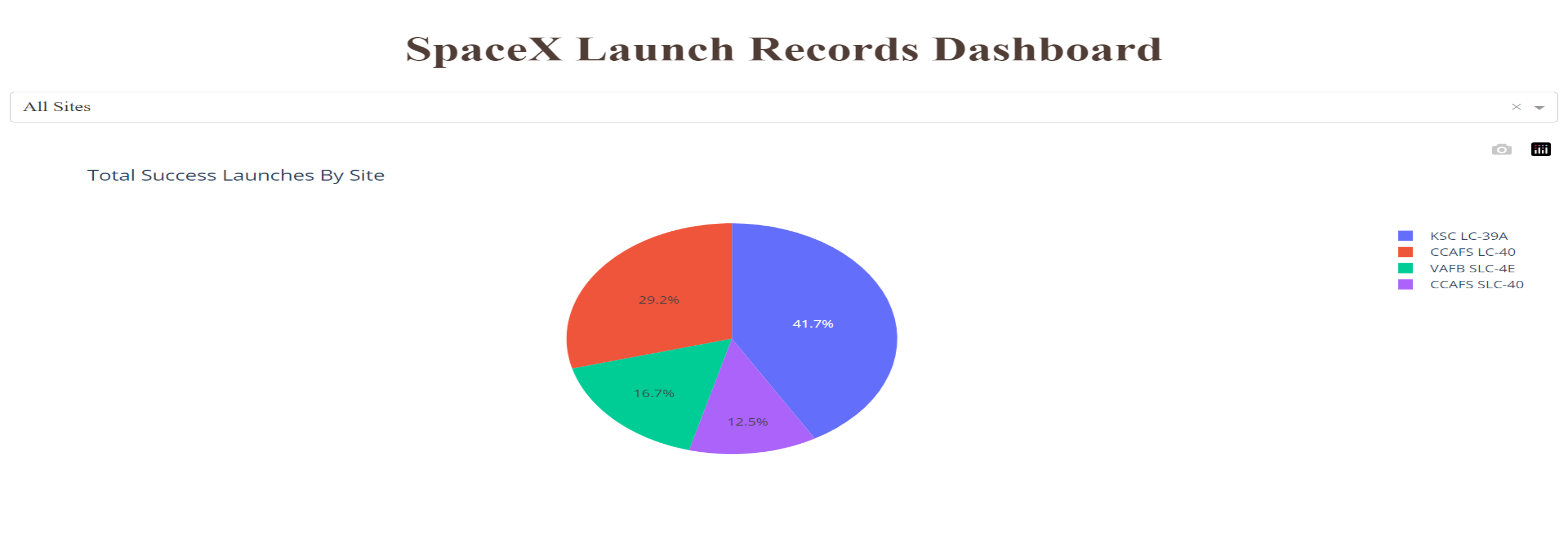
- Launch sites are always in close proximity to the coastlines

Section 4

Build a Dashboard with Plotly Dash

Launch Success Count for All Sites

Screenshot of launch success count for all sites, in a piechart

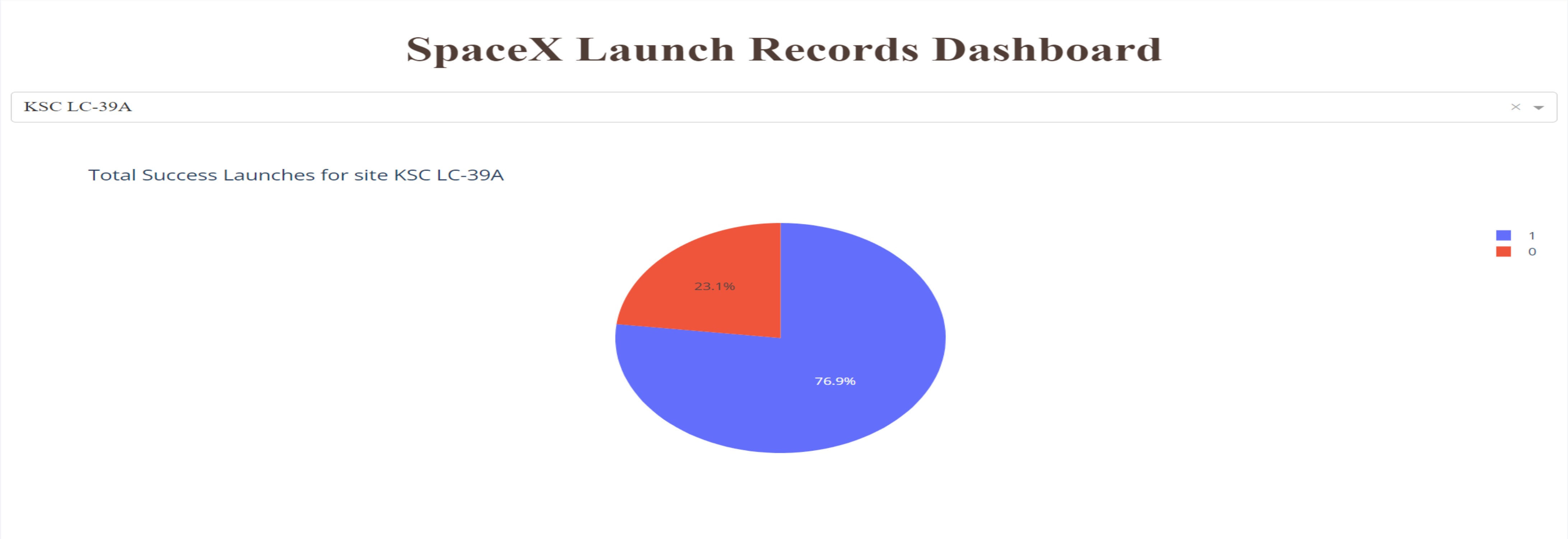


Important elements and findings on the screenshot:

- Successful launches by launch sites, with KSC LC 39A and CCAFS LC 40 having the highest success rates.

Launch Sites with Highest Launch Success

Screenshot of the pie chart for the launch site with highest launch success ratio



The important elements and findings on the screenshot

- KSC LC-39A had highest success rate of 76.9% success, 23.1% failed missions.

Payload vs Launch Outcomes

Screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

The important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

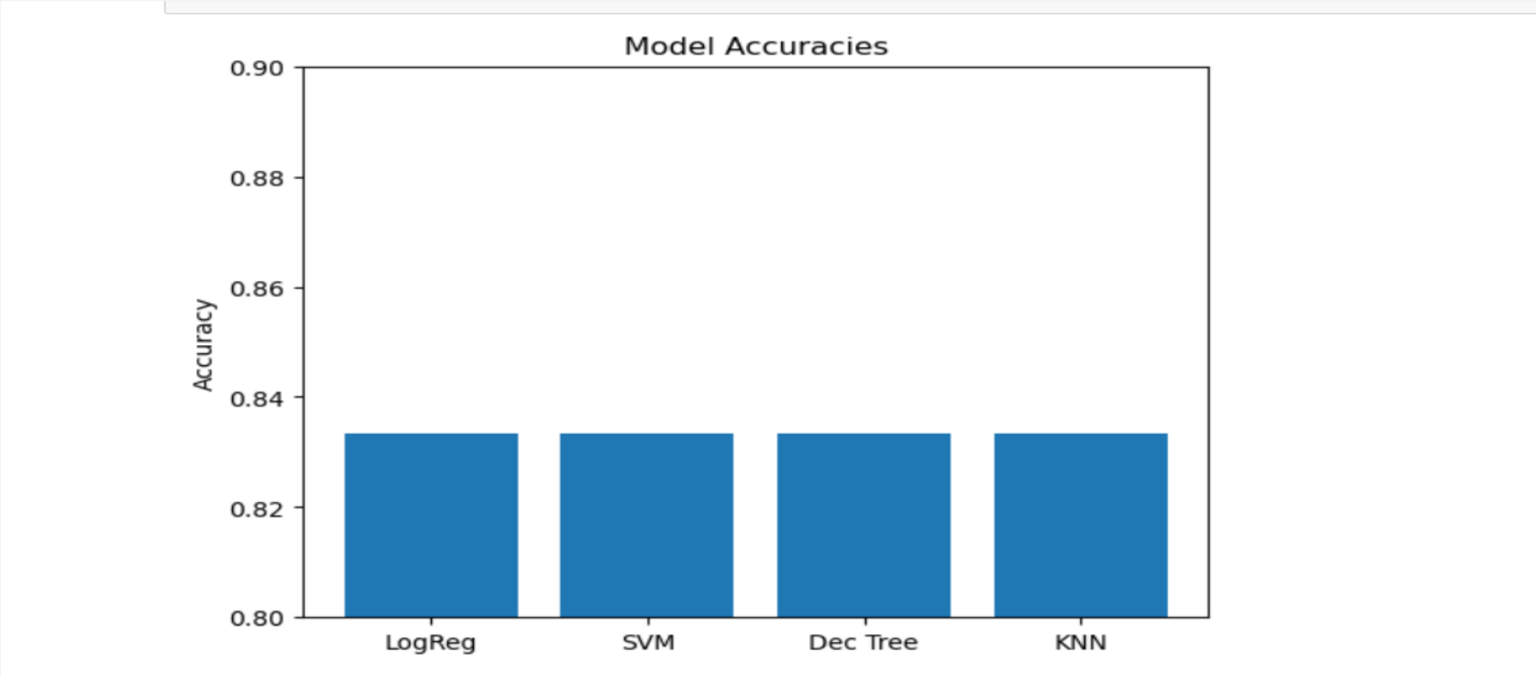
- Booster v1.0 and FT had the highest success rate for payloads up to 10k and 6k maximum, respectively, while others had success rates below 5k.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Visual of the built models accuracy for all built classification models, in a bar chart

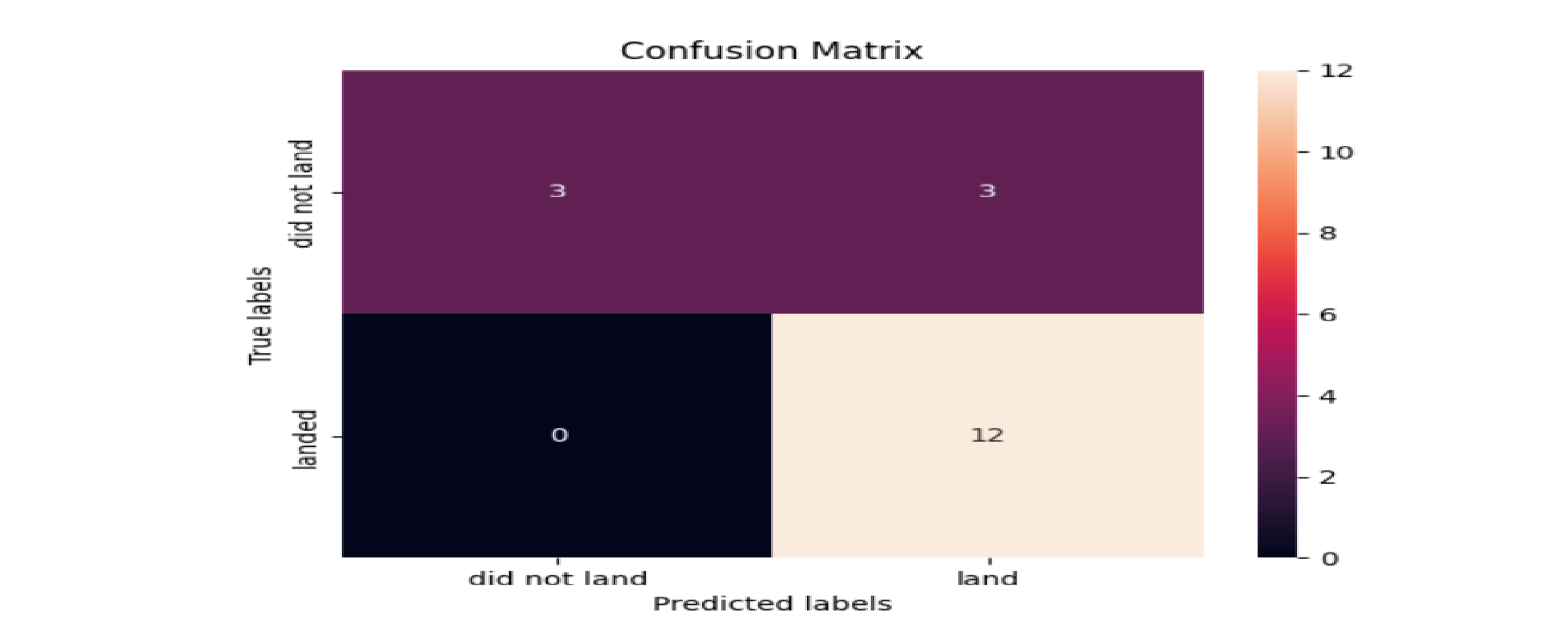


Find which model has the highest classification accuracy

- Test set accuracy for all models was 0.8333333333333334, but the Decision tree classifier had a higher accuracy of 0.9017857142857144 in the training set.

Confusion Matrix

- Confusion matrix of the best performing model with explanation



The confusion matrix reveals that the Decision tree classifier effectively identifies the different classes, but there are several false positives.

Conclusions

- Based on the bar chart visualization, we found that the orbits with the highest success rates for missions are ES LI, GEO, HEO, and SSO.
- •The scatter plot we generated from the Plotly interactive dashboard revealed that most booster versions were successful in launching payloads with masses between 2,000 to 6,000, with FT being the most successful booster version followed by B4, which has a payload capacity of 10,000.
- •We analyzed the data using a Plotly pie chart and found that KSC LC-39A was the launch site with the highest success rate of 41.7%, followed by CCAFS LC-40 with 29.2%, VAFB SLC-4E with 16.7%, and CCAFS SLC-40 with the lowest success rate of 12.5%.
- •Using the Folium map, we discovered that launch sites were mostly located in the coastlines, but not within the proximity of any city.
- •Our classification model showed that the Decision tree classifier was the best prediction model, with a training accuracy of 0.9017857142857144 and a test accuracy of 0.8333333333333334, which was the same as other models tested.

Appendix

Relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

- All the notebooks, codes, and assets related to this project are available in the GitHub URL link provided in each applicable slide.

Thank you!