IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Wilna Coronado

August 6, 2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

Summary of methodologies

- -Data Collection through API
- -Data Collection with Web Scraping
- -Data Wrangling
- -Exploratory Data Analysis with SQL
- -Exploratory Data Analysis with Data Visualization
- -Interactive Visual Analytics with Folium
- -Machine Learning Prediction

Summary of all results

- -Exploratory Data Analysis result
- -Interactive analytics in screenshots
- -Predictive Analytics result

Introduction

Project Background and Context

SpaceX's Falcon 9 rocket has revolutionized the aerospace industry with its reusable first stage, resulting in significant reductions in launch costs. While SpaceX offers launches at $62 million, other providers charge upwards of $165 million. The key to SpaceX's cost efficiency lies in the reusability of the Falcon 9's first stage. By determining whether the first stage will successfully land, we can ascertain the launch price. Public data and machine learning models can help predict whether SpaceX or a competing company can reuse the first stage. Understanding the factors influencing the successful landing of the Falcon 9 first stage can yield valuable insights for competitors and industry stakeholders.

Problems to Address

What factors determine if the rocket will land successfully?

Can we accurately predict the successful landing of the Falcon 9 first stage using historical data?

How does the prediction of landing success translate into cost savings for rocket launches?

Section 1

Methodology

Methodology

Executive Summary

- Collect data using SpaceX API and web scraping from Wikipedia.
- Perform Data Wrangling by applying One Hot Encoding to the data fields for machine learning, remove any irrelevant columns and handled missing values.
- Explore the data using SQL and data visualization techniques.
- Create visualizations using Folium and Plotly Dash.
- Construct classification models to predict landing outcomes and fine-tune and evaluate the models to identify the best parameters.

Data Collection

In our data collection process, we utilized two primary sources:

SpaceX REST API:

- We accessed SpaceX launch data directly from the SpaceX REST API.
- This API provides comprehensive information about launches, including details about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcomes.
- Our objective is to leverage this data for predicting whether SpaceX will attempt to land a rocket.

Web Scraping Wikipedia:

- Additionally, we obtained Falcon 9 launch data by web scraping Wikipedia using BeautifulSoup.
- The extracted data includes relevant details from Falcon 9 launches.
- This diverse dataset enhances our analysis and modeling efforts.

Data Collection – SpaceX API

Request data from the SpaceX API and decode the response into a Pandas dataframe.

Extract relevant information and transform it into a dataframe for data analysis.

Filter the dataframe to include only Falcon 9 launches and impute Missing Values.

Export the cleaned data to a CSV file.

[Link to GitHub Notebook]

Request data from the SpaceX API and decode the response into a Pandas dataframe.

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```
✓ 0.0s

```
response = requests.get(spacex_url)
```
✓ 2.1s

Extract relevant information and transform it into a dataframe for data analysis.

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```
✓ 0.0s

```
# Create a data from launch_dict
launch_dict_df = pd.DataFrame(launch_dict)
```
✓ 0.0s

Filter the dataframe to include only Falcon 9 launches and impute Missing

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_dict_df[launch_dict_df['BoosterVersion']=='Falcon 9']
```
✓ 0.0s

```
# Calculate the mean value of PayloadMass column
mean_payload_mass = round(data_falcon9['PayloadMass'].mean(),0)
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].fillna(mean_payload_mass, inplace=True)
```
✓ 0.0s

Export the cleaned data to a CSV file.

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```
✓ 0.0s

Data Collection – Web Scraping

Request Falcon 9 launch data from Wikipedia and create a BeautifulSoup object.

Extract column names and collect relevant data by parsing.

Create a dictionary and transform the dictionary into a Pandas dataframe.

Save the cleaned data to a CSV file.

Request Falcon 9 launch data from Wikipedia and create a BeautifulSoup object.

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldi
```

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

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

Extract column names and collect relevant data by parsing.

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all("table")
print(html_tables)
```

```
column_names = []
for row in first_launch_table.find_all('th'):
    columns = extract_column_from_header(row)
    if columns is not None and len(columns) > 0:
        column_names.append(columns)
```

Create a dictionary and transform the dictionary into a Pandas dataframe.

```
launch_dict= dict.fromkeys(column_names)
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
    # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight number.isdigit()
```

```
df=pd.DataFrame(launch_dict)
```

Save the cleaned data to a CSV file.

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

[Link to GitHub Notebook]

Data Wrangling

Exploratory Data Analysis and Training Labels:

Conducted exploratory data analysis to identify training labels.

Launch Site Statistics:

Calculated the number of launches at each site.

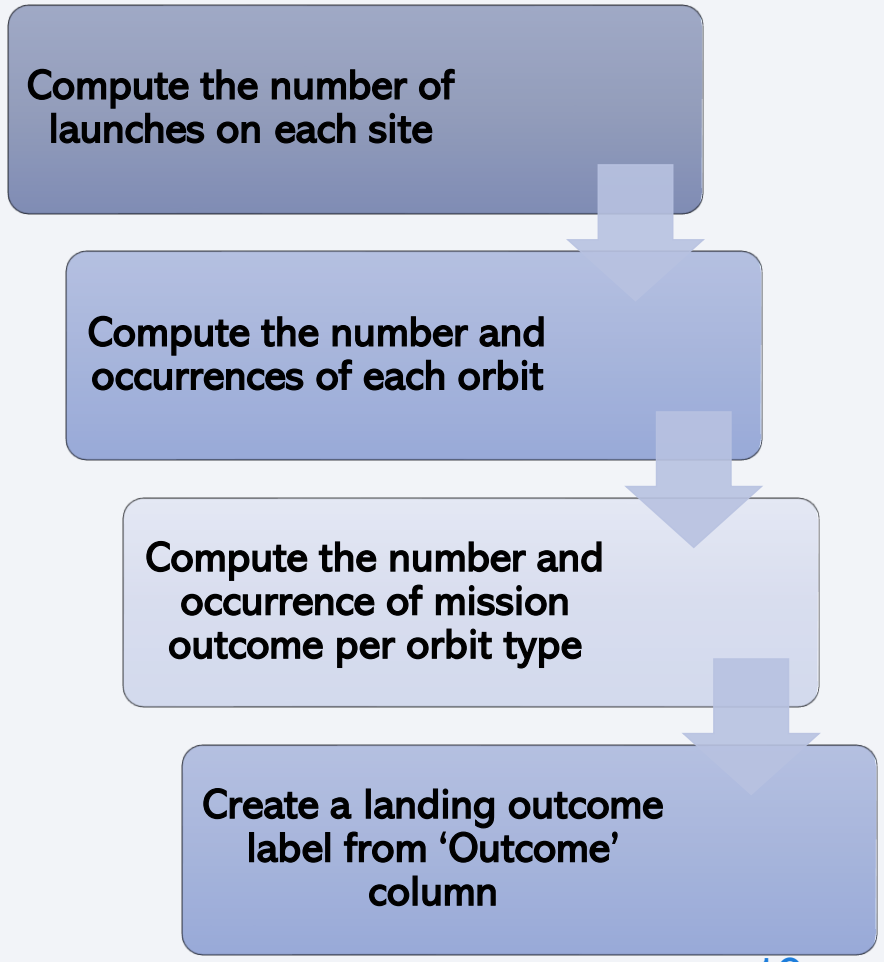
Orbit Information:

Determined the occurrence of each orbit type.

Landing Outcome Label:

Created a landing outcome label from the 'Outcome' column.

Exported the results to a CSV file.



Link to GitHub Notebook

EDA with Data Visualization

Scatter Plots are ideal for visualizing the relationship between two continuous variables.

Flight Number vs. Payload Mass: Is there's any correlation between the flight number and payload mass. Are newer flights carrying heavier payloads?
Flight Number vs. Launch Sites: Is there's any pattern or clustering in launch sites based on flight numbers. Are certain launch sites more common for specific flight numbers?
Payload vs. Launch Sites: How payload mass varies across different launch sites. Are there any trends or outliers?
Flight Number and Orbit Type: Are there consistent patterns on any certain flight numbers and orbit association?
Payload and Orbit Type: Does payload mass differs significantly based on the type of orbit.

Bar Chart (Success Rate of Each Orbit)

Bar charts are effective for comparing discrete categories. By plotting the success rate for each orbit type, we can easily identify which orbits have higher success rates. This information is valuable for decision-making and risk assessment.

Line Plot (Success Rate Over Time)

Line plots are excellent for showing trends over time. By plotting the success rate against dates (chronologically), we can observe how SpaceX's success rate has evolved. Are there any notable improvements or fluctuations? This helps stakeholders track progress and identify areas for improvement.

Link to GitHub Notebook

EDA with SQL

Summary of SQL queries that were used:

- Display unique launch site names in the space mission.
- Show 5 records where launch sites start with ‘CCA.’
- Retrieve the total payload mass carried by NASA (CRS) boosters.
- Calculate the average payload mass carried by booster version F9 v1.1.
- Identify the date of the first successful ground pad landing outcome.
- List booster names with drone ship success and payload mass between 4000 and 6000.
- Provide the total count of successful and failed mission outcomes.
- Determine booster versions that carried the maximum payload mass using a subquery.
- List failed drone ship landing outcomes, their booster versions, and launch site names for 2015.
- Rank landing outcomes (Failure or Success) between June 4, 2010, and March 20, 2017, in descending order.

Link to GitHub Notebook

Build an Interactive Map with Folium

Markers Indicating Launch Sites

Placed a blue circle at the coordinates of NASA Johnson Space Center, displaying its name in a popup using latitude and longitude.

Marked red circles at all launch site coordinates, showing their names in popups based on latitude and longitude, using a Folium map.

Colored Markers of Launch Outcomes

Added colored markers to indicate successful launches (green) and unsuccessful launches (red) at each launch site, highlighting sites with high success rates.

Distances Between a Launch Site to Proximities

Colored lines have been added to the map, illustrating the distance between the launch site CCAFS SLC40 and its proximity to the nearest coastline, railway, highway, and city.

Link to GitHub Notebook

Build a Dashboard with Plotly Dash

Dropdown List with Launch Sites:

Purpose: Allows users to select either all launch sites or a specific launch site.

Use case: Provides flexibility in filtering data based on launch location.

Pie Chart Showing Successful Launches:

Purpose: Visualizes the proportion of successful and unsuccessful launches.

Use case: Helps users quickly grasp the success rate and identify trends.

Slider of Payload Mass Range:

Purpose: Enables users to filter launches based on payload mass.

Use case: Useful for exploring correlations between payload mass and launch outcomes.

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

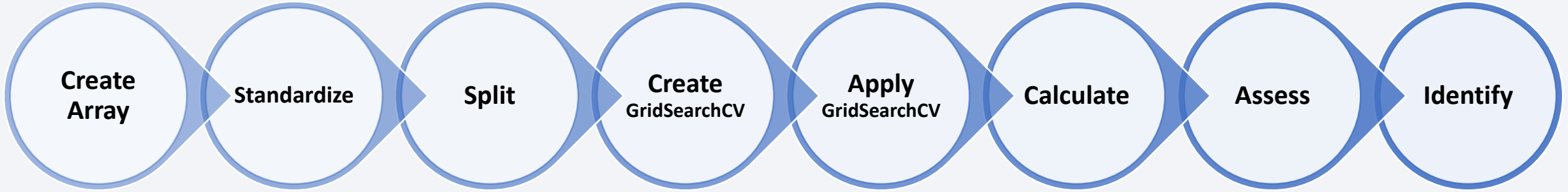
Purpose: Illustrates the relationship between payload mass and launch success, considering different booster versions.

Use case: Helps users analyze how payload mass affects success rates across booster types.

Link to GitHub Notebook

Predictive Analysis (Classification)

Scikit-learn is Machine Learning library that was used for predictive analysis.



- Create NumPy array from the Class column
- Standardize the data with StandardScaler. Fit and transform the data.
- Split the data using train_test_split
- Create a GridSearchCV object with cv=10 for parameter optimization
- Apply GridSearchCV on different algorithms: logistic regression, SVC, decision tree, K-Nearest Neighbor
- Calculate accuracy on the test data using .score() for all models
- Assess the confusion matrix for all models
- Identify the best model using Jaccard_Score, F1_Score and Accuracy

[Link to GitHub Notebook]

Results

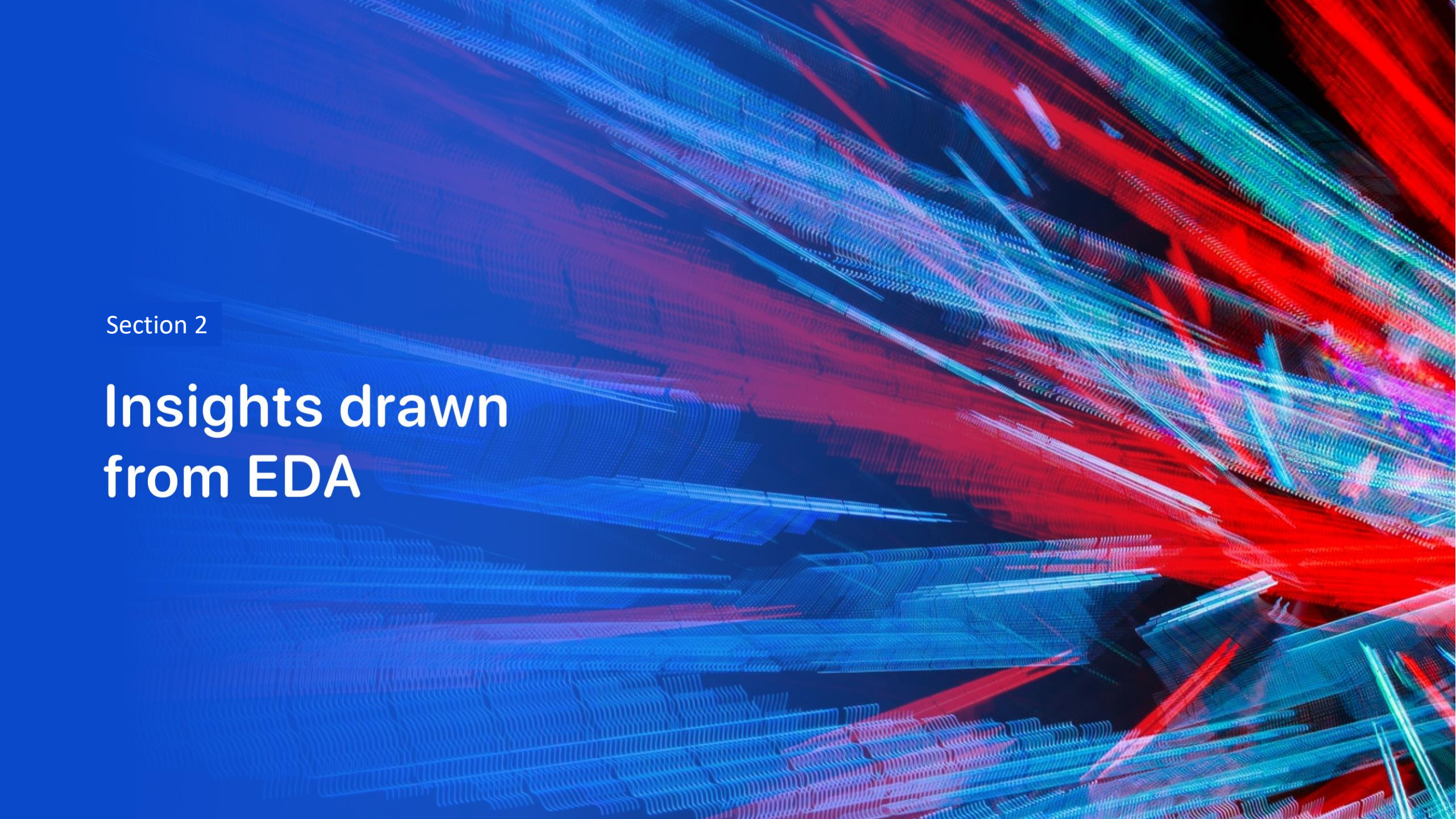
The **Exploratory Data Analysis** has shown us that successful landing outcomes are somewhat correlated with flight number. It was also apparent that successful landing outcomes have had a significant increase since the year 2015.

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

The **Visual Analytics** shows that all launch sites are located near the coast line. Perhaps, this makes it easier to test rocket landings in the water.

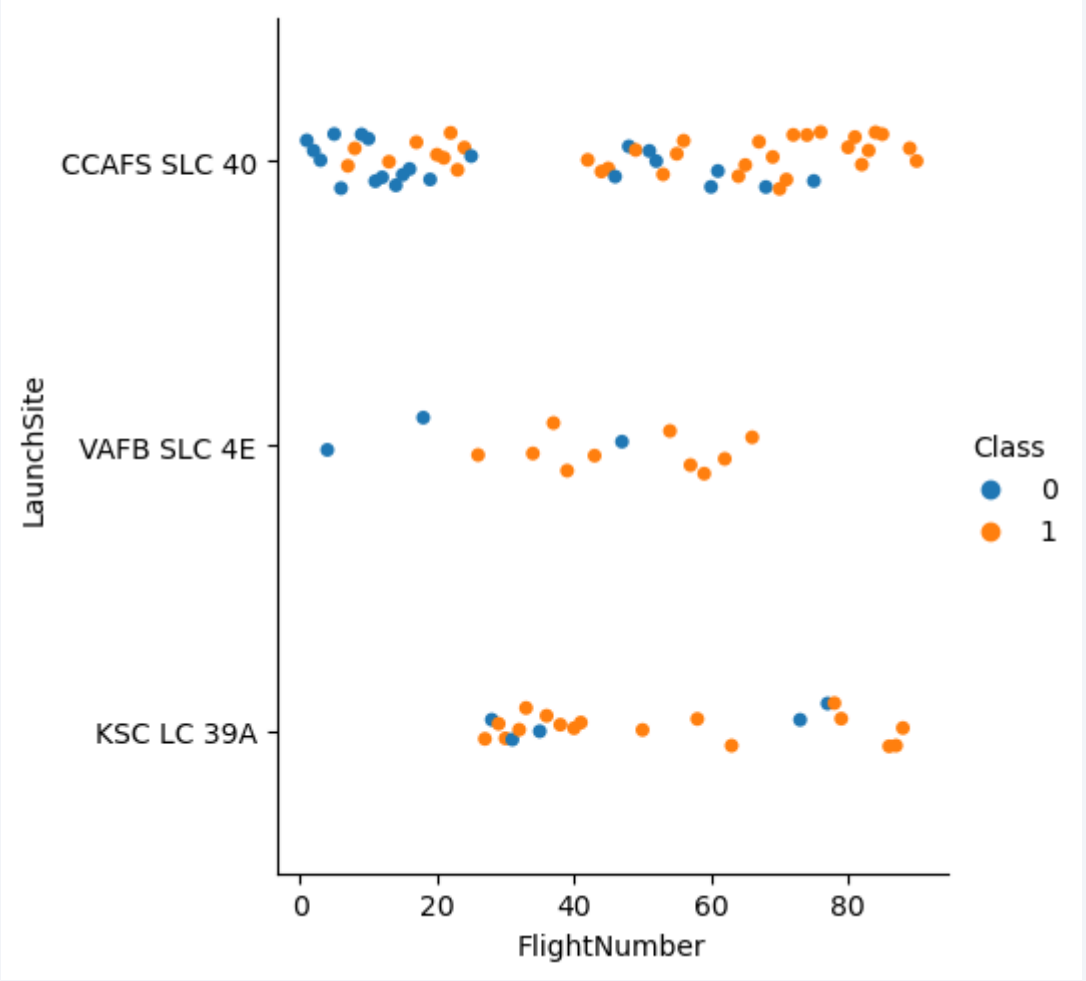
Launch sites are far enough away from anything a failed launch can damage (city, highway, railway), while still close enough to bring people and material to support launch activities.

Decision Tree model is the best **predictive model** for the dataset. The machine learning were able to predict the landing success of rockets with an accuracy score of 83.33%.


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

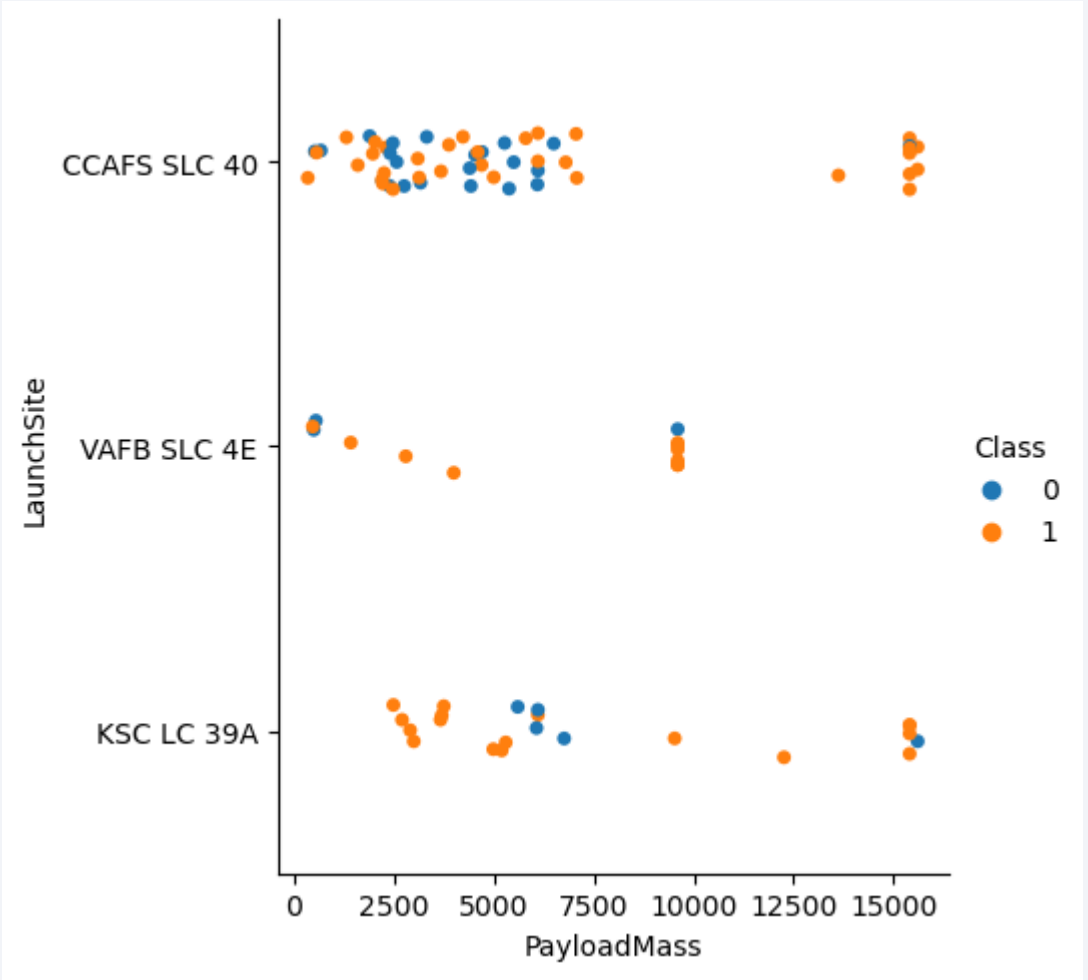
Earlier flights, represented by blue, exhibited a lower success rate, while later flights (orange) showed higher success rates. Approximately half of the launches originated from CCAFS SLC 40, whereas VAFB SLC 4E and KSC LC 39A had notably higher success rates.
Overall, we can infer that newer launches tend to achieve greater success.



Payload vs. Launch Site

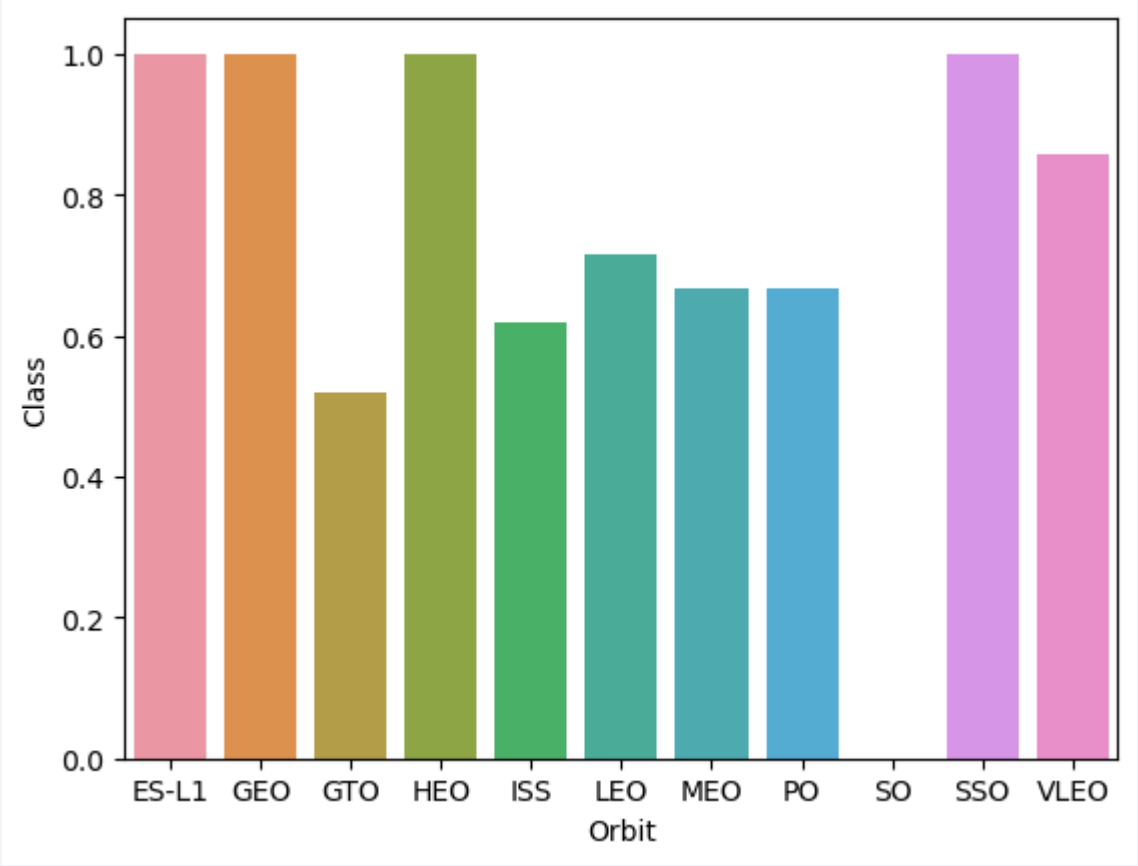
Generally, as payload mass (measured in kilograms) increases, the success rate of launches tends to rise.
Notably, most launches with a payload exceeding 7,000 kg achieved success.

Additionally, KSC LC 39A boasts a perfect success rate for launches with payloads below 5,500 kg, while VAFB SLC 4E has yet to launch payloads greater than approximately 10,000 kg.



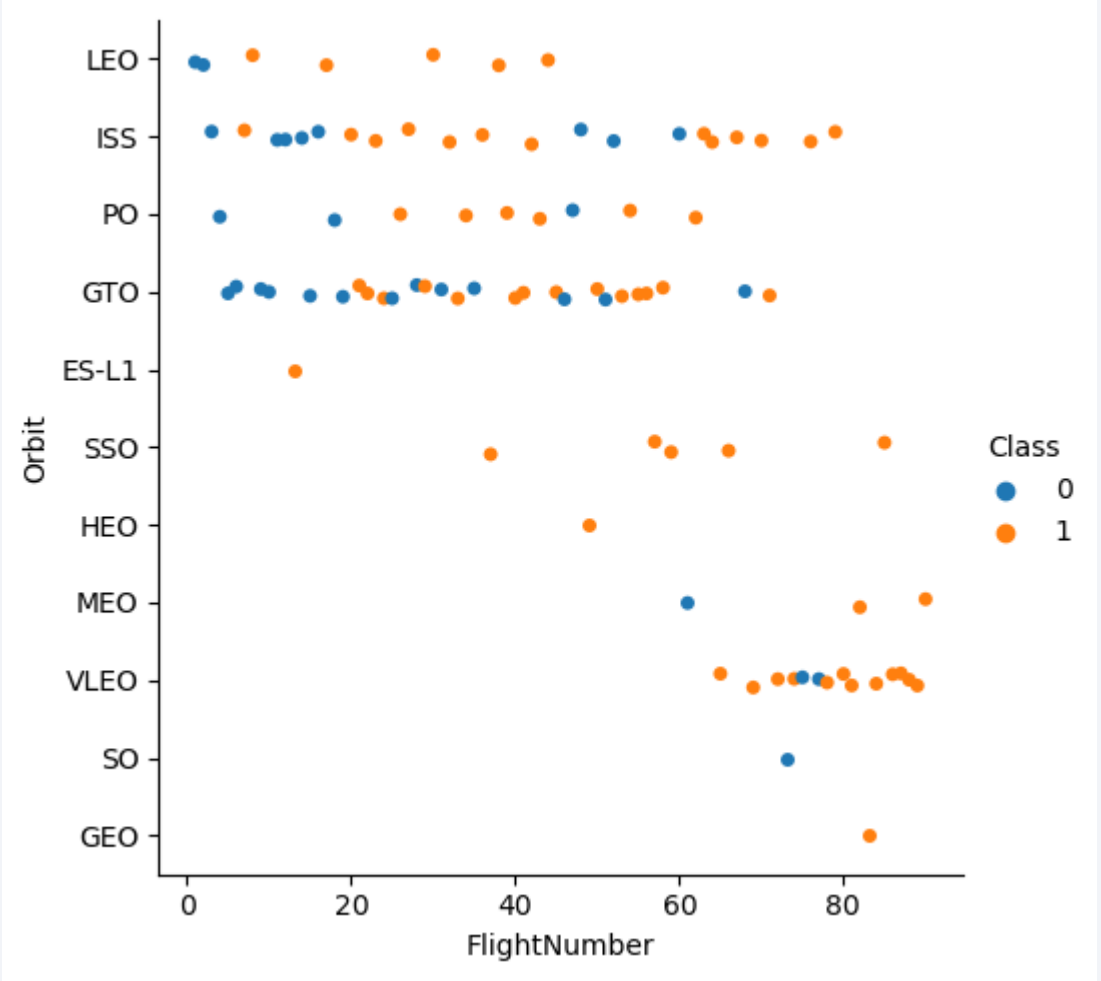
Success Rate vs. Orbit Type

- Missions to ES-L1, GEO, HEO, and SSO achieved a remarkable 100% success rate.
- Launches targeting GTO, ISS, LEO, MEO, and PO fell within the 50%-80% success range.
- Unfortunately, launches to SO (specific orbit) had a 0% success rate.



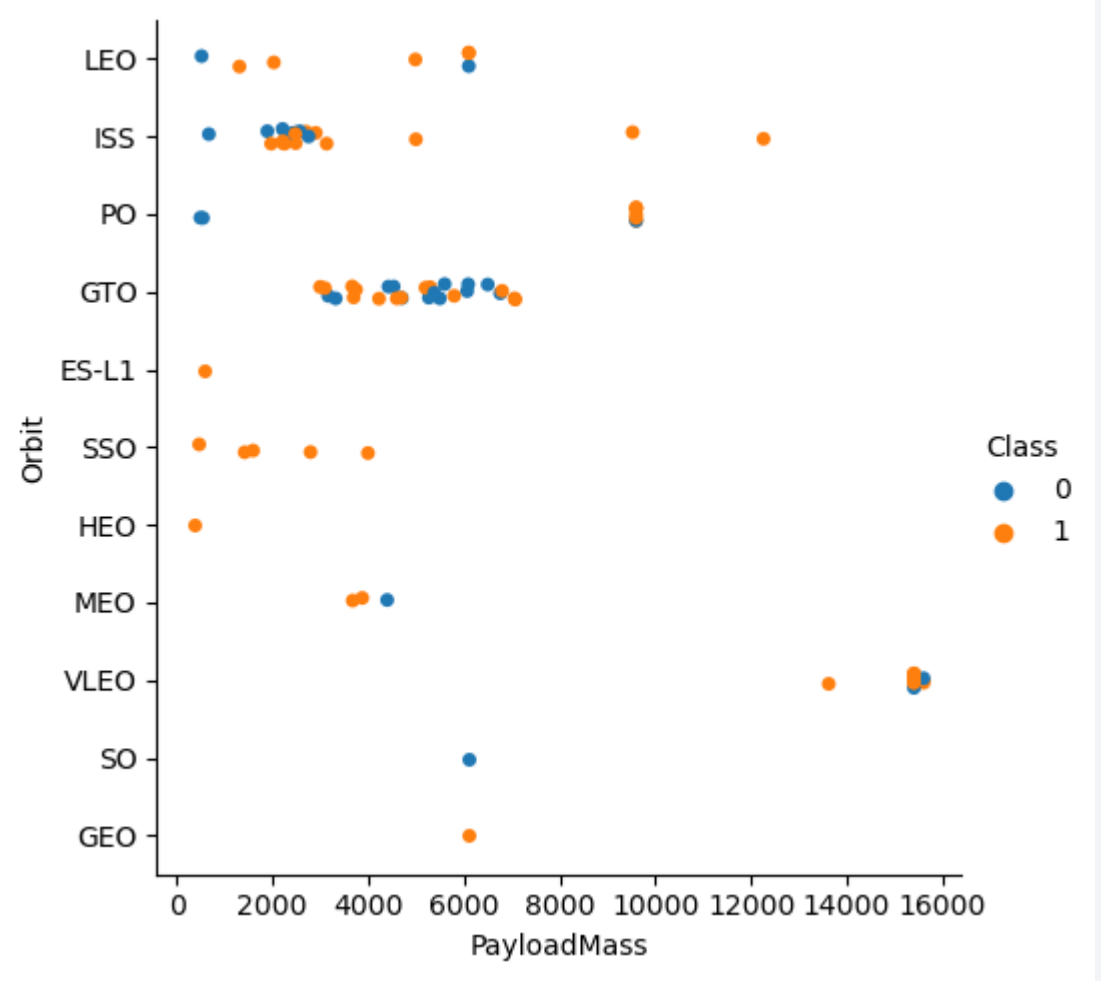
Flight Number vs. Orbit Type

The success rate tends to increase as the number of flights for each orbit grows. This trend is particularly evident in the Low Earth Orbit (LEO). However, the Geostationary Transfer Orbit (GTO) does not conform to this pattern.



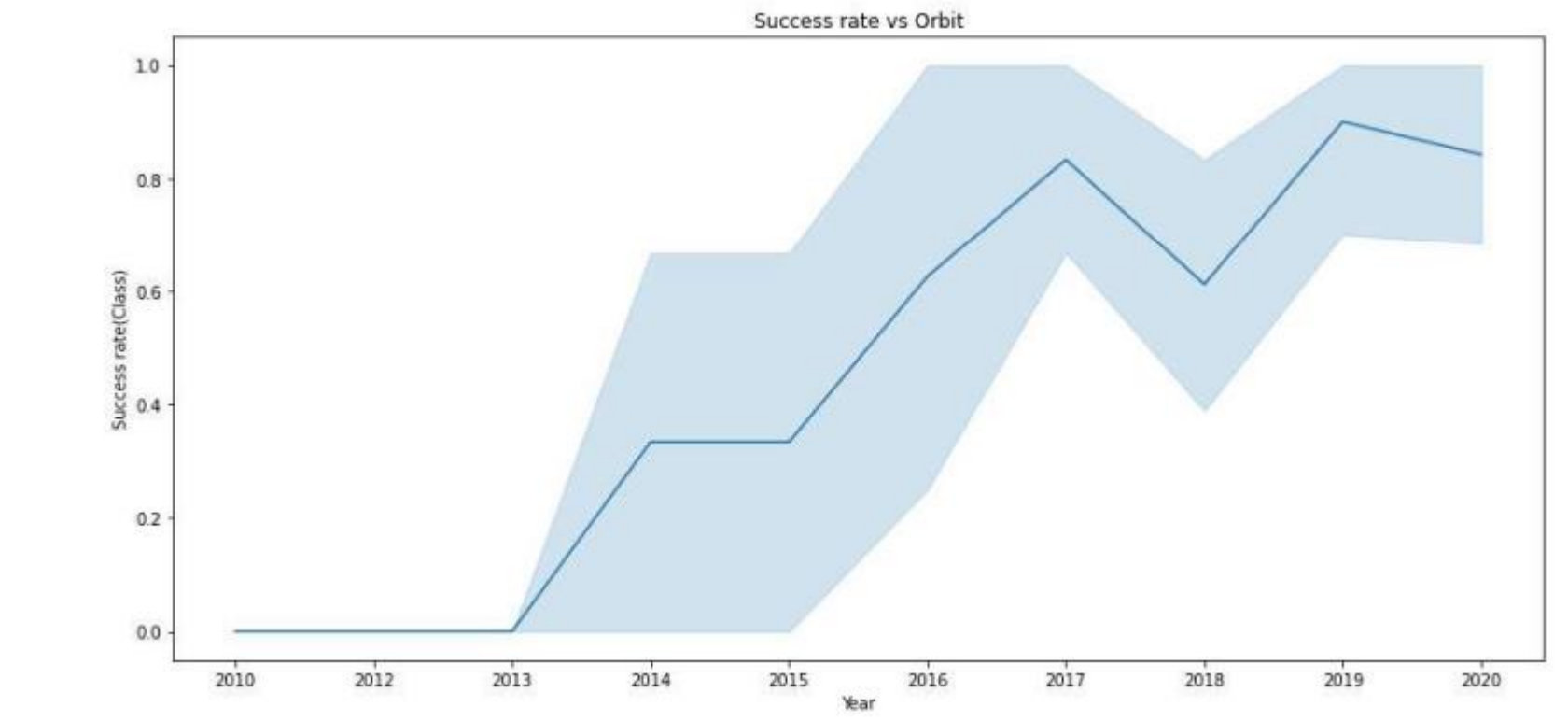
Payload vs. Orbit Type

Heavy payloads tend to perform well in Low Earth Orbit (LEO), International Space Station (ISS), and Polar Orbit (PO). However, the Geostationary Transfer Orbit (GTO) shows mixed success rates when handling heavier payloads.



Launch Success Yearly Trend

The success rate showed improvement between 2013 and 2017, as well as from 2018 to 2019. However, there was a decline in success rates from 2017 to 2018 and again from 2019 to 2020. Overall, since 2013, the success rate has seen positive progress.



All Launch Site Names

```
%sql SELECT distinct Launch_Site FROM SPACEXTABLE;
```

✓ 0.0s

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

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

There are only 4 Launch Sites where different rocket landings where attempted.

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5;
```

✓ 0.0s Python

```
 * 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

These are 5 records where launch sites begin with the letters 'CCA'.
As we can see, there are other organizations besides Space X that were testing their rockets

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) NASACRS_Payload_Mass from SPACEXTABLE where Customer= 'NASA (CRS)'
```

✓ 0.0s

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

NASACRS_Payload_Mass
45596

The total payload mass carried by boosters launched by NASA (CRS) is 45,596 kg

Average Payload Mass by F9 v1.1

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

✓ 0.0s

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

AVG_PAYLOAD_MASS
2928.4

The query calculates the average payload mass carried into space for the ‘F9 v1.1’ booster version, resulting in approximately 2928.4 kg

First Successful Ground Landing Date

```
%sql select min(Date) from SPACEXTABLE where Landing_Outcome='Success (ground pad)' order by Date;
```

✓ 0.0s

* sqlite:///my_data1.db
Done.

min(Date)
2015-12-22

The SQL query selects the minimum date (min(Date)) from a table named 'SPACEXTABLE' where the 'Landing_Outcome' is 'Success (ground pad)'.
The result indicates that the earliest successful ground pad landing occurred on December 12,2022.
This information could be relevant for analyzing the history of space missions, particularly those related to SpaceX and their landing outcomes over time.

Successful Drone Ship Landing with Payload between 4000 and 6000

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

✓ 0.0s

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

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

The SQL query results indicate that there are four instances where the 'BoosterVersion' had a landing outcome of 'Success (drone ship)' with the payload mass being greater than or equal to 4000 kg and less than 6000 kg. This information could be relevant for analyzing the performance of different booster versions under specified conditions.

Total Number of Successful and Failure Mission Outcomes

```
%sql select Mission_Outcome,count(*) from SPACEXTABLE group by Mission_Outcome;
```

✓ 0.0s

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

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

The SQL query results indicate that there was one occurrence of “Failure (in flight),” 98 occurrences of “Success,” and one occurrence of “Success (payload status unclear)” in the dataset.

Boosters Carried Maximum Payload

```
%sql select distinct Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE );
```

✓ 0.0s

```
 * 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

It shows that 12 unique boosters have carried the maximum payload mass of 15600 kg.

2015 Launch Records

```
select substr(Date, 6,2) Month,Landing_Outcome,Booster_Version,Launch_Site from SPACEXTABLE where substr(Date,0,5)='2015' and Landing_Outcome = 'Failure (drone ship)'
```

✓ 0.0s Python

```
 * 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

There were 2 failed landing_outcomes in drone ship for in year 2015.

We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

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

```
%sql select landing_outcome,count(*) count from SPACEXTABLE where Date BETWEEN '2010-06-04' AND '2017-03-20' group by landing_outcome order by 2 desc;
```

✓ 0.0s

* 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

- We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.
- We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

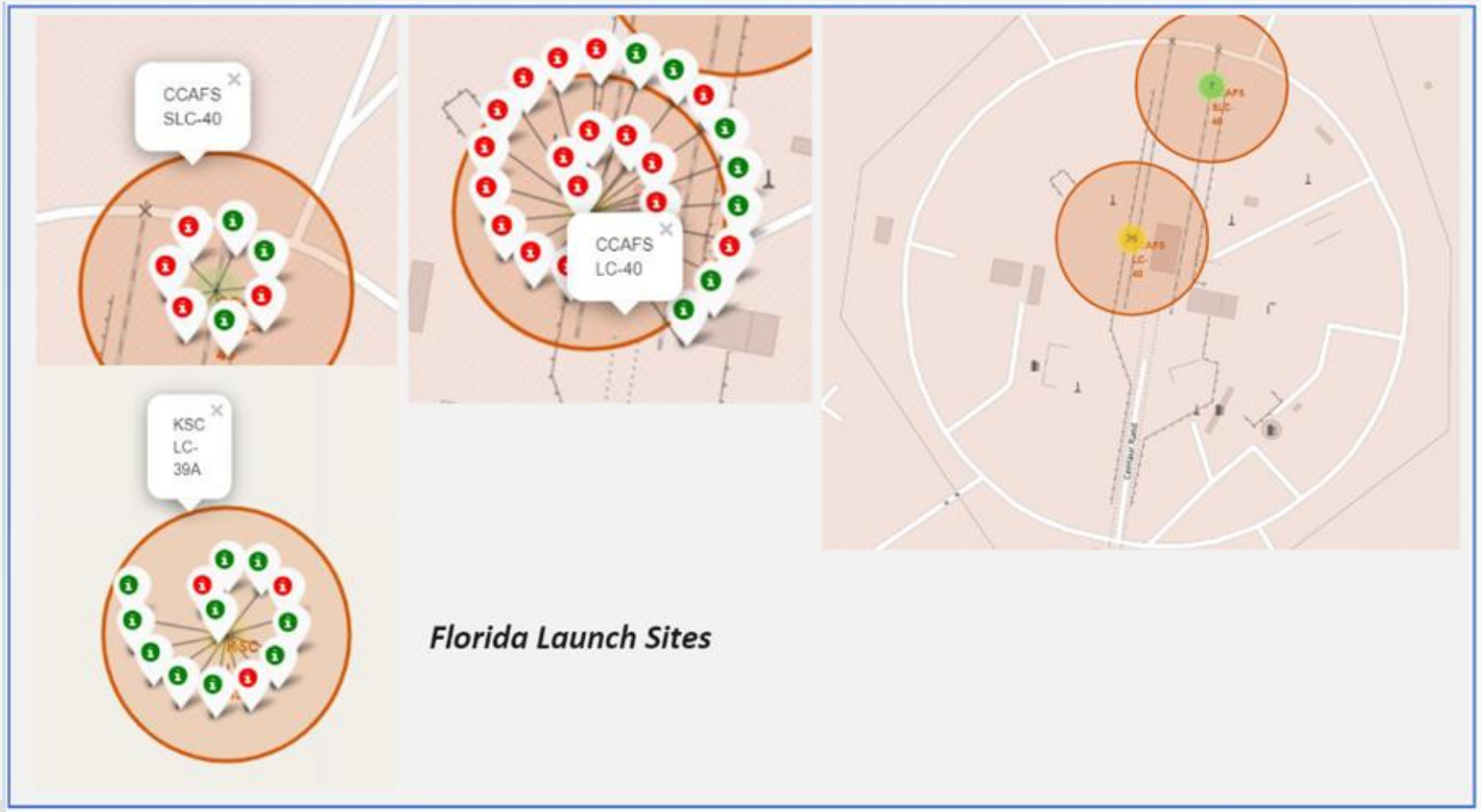
Section 3

Launch Sites Proximities Analysis

Launch Site Locations

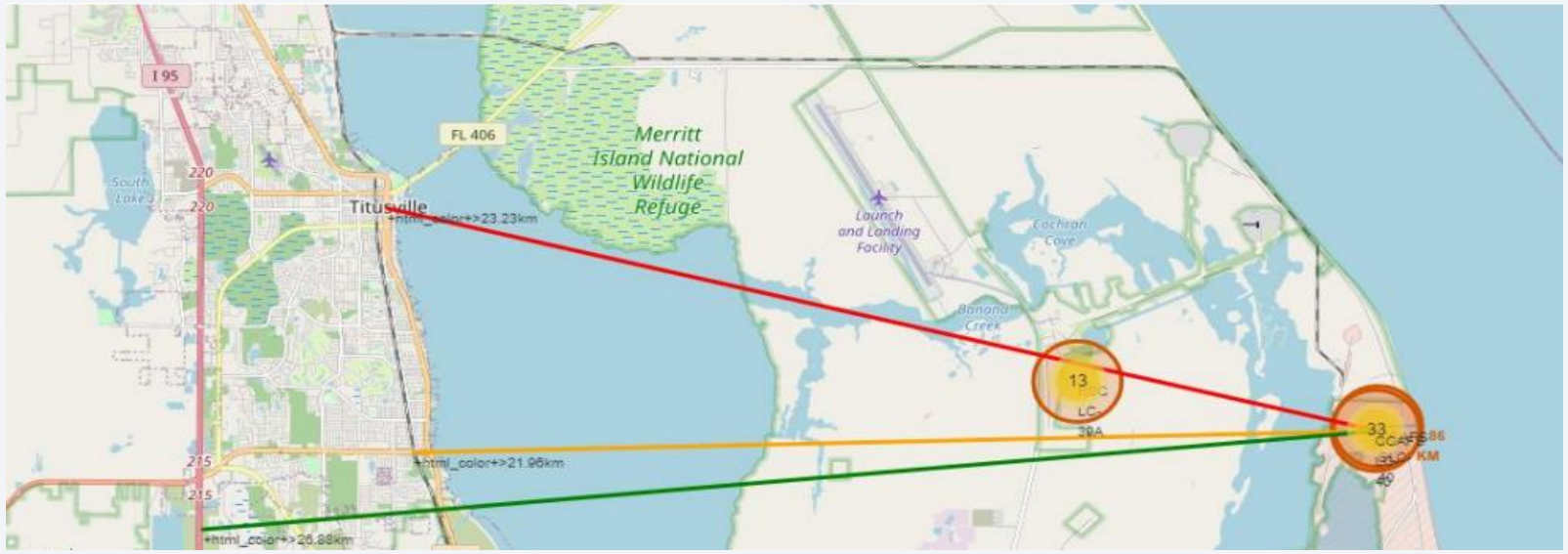
Launching rockets from sites near the equator, all of which are in very close proximity to the coast and a couple of thousand kilometers away from the equator line, benefits from Earth's rotation. This makes it easier to achieve an equatorial orbit, and the additional natural boost due to Earth's rotational speed helps reduce costs by minimizing the need for extra fuel and boosters.

Success Rate of Rocket Launches



The successful launches are represented by a green marker while the red marker represents failed rocket launches.
Launch site CCAFS SLC-40 has a 3/7 success rate (42.9%)

Distance to Proximities

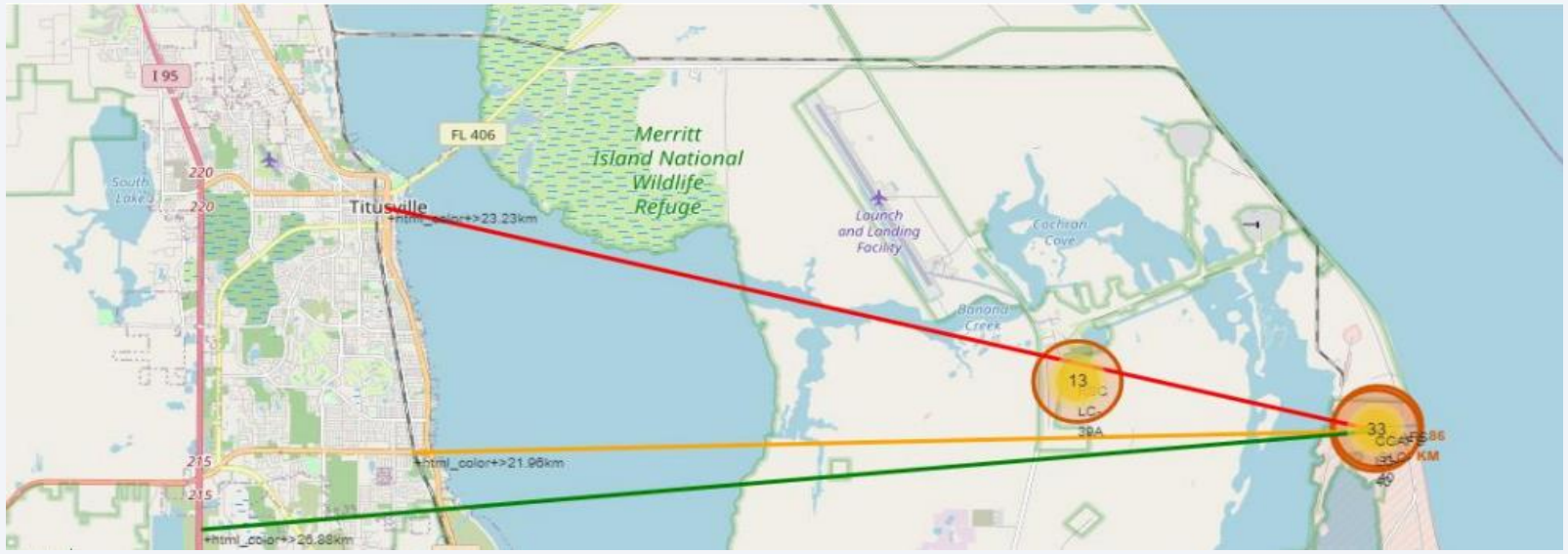


The image depicts a map with distances from a central point labeled “CCAFS SLC-40” to various proximities. Specifically, it shows the following distances:

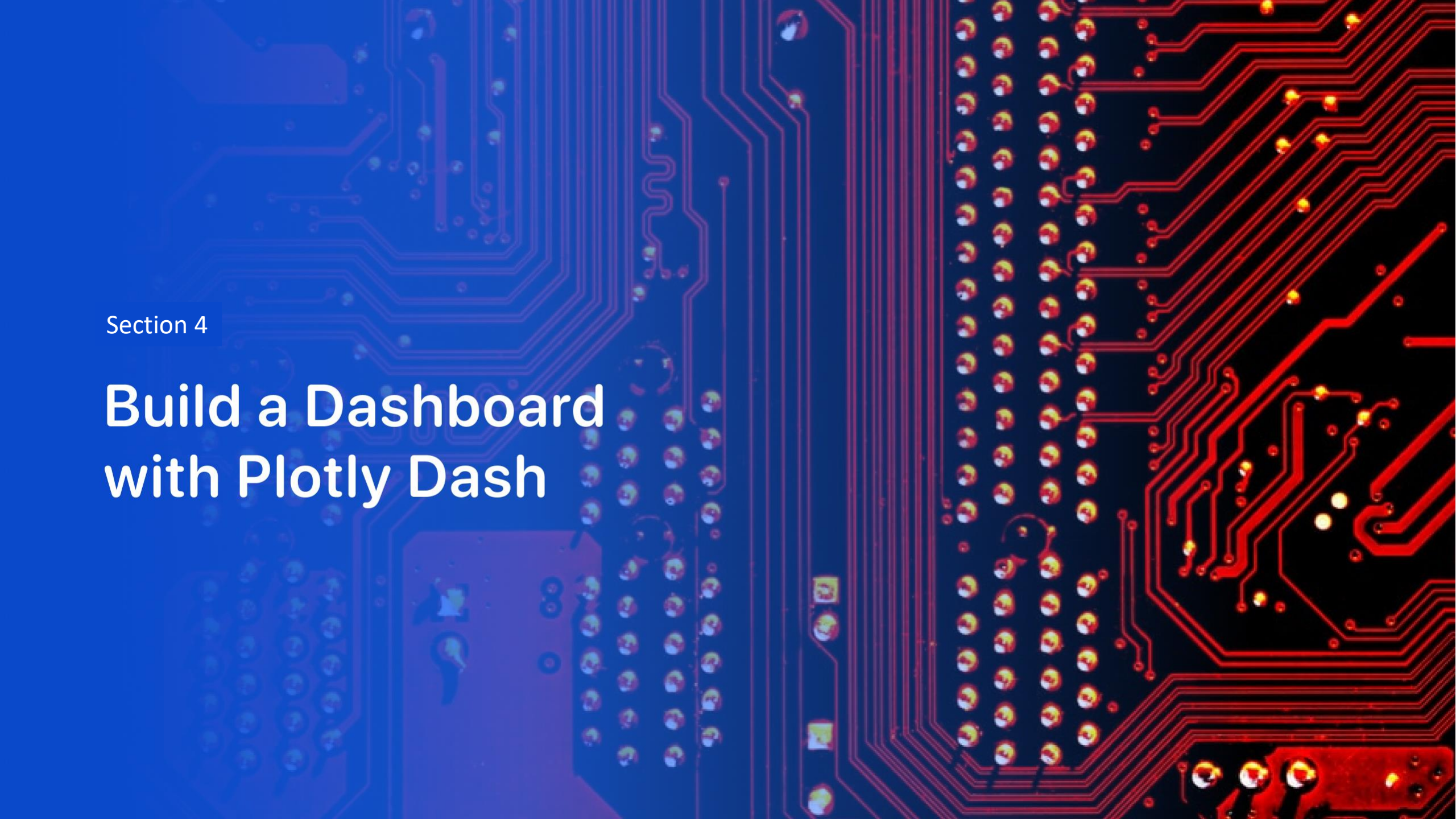
- **.86 km from the nearest coastline**
- **21.96 km from the nearest railway**
- **23.23 km from the nearest city**
- **26.88 km from the nearest highway**

These distances are visually represented on the map, connecting CCAFS SLC-40 to the respective points of interest. Such information could be valuable for logistical planning or assessing accessibility.

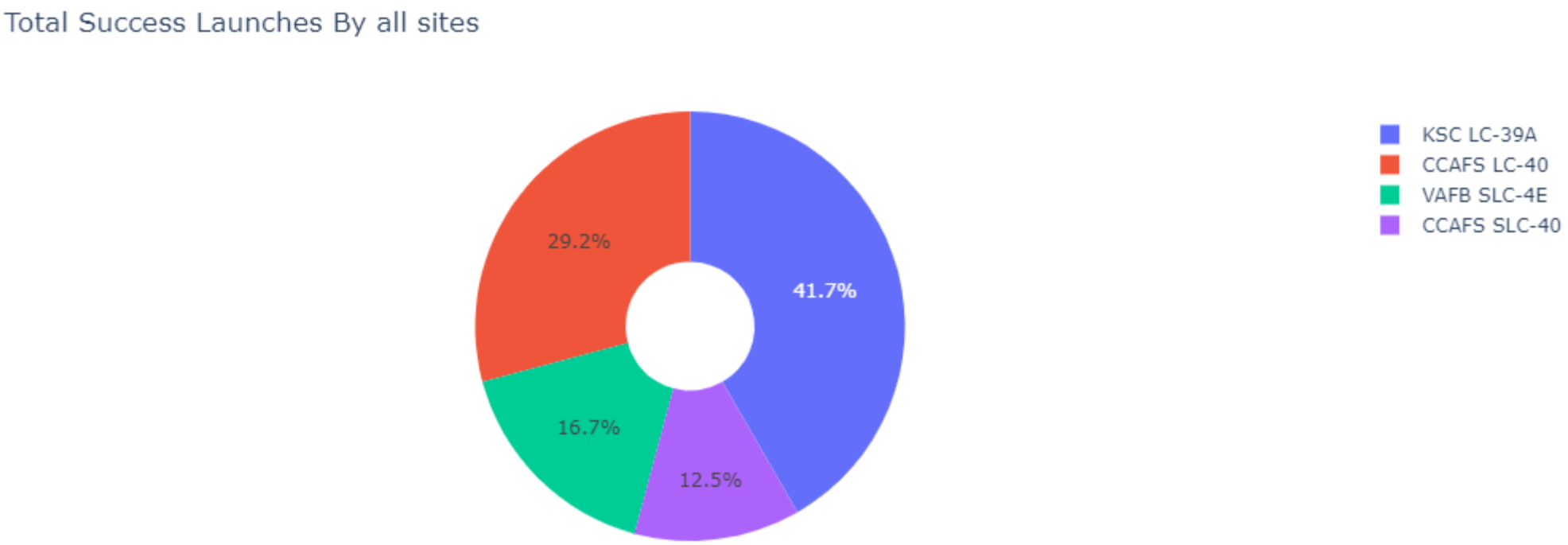
Distance to Proximities



• **Coasts**: help ensure that spent stages dropped along the launch path or failed launches don't fall on people or property.
• **Safety / Security**: needs to be an exclusion zone around the launch site to keep unauthorized people away and keep people safe.
• **Transportation/Infrastructure and Cities**: 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

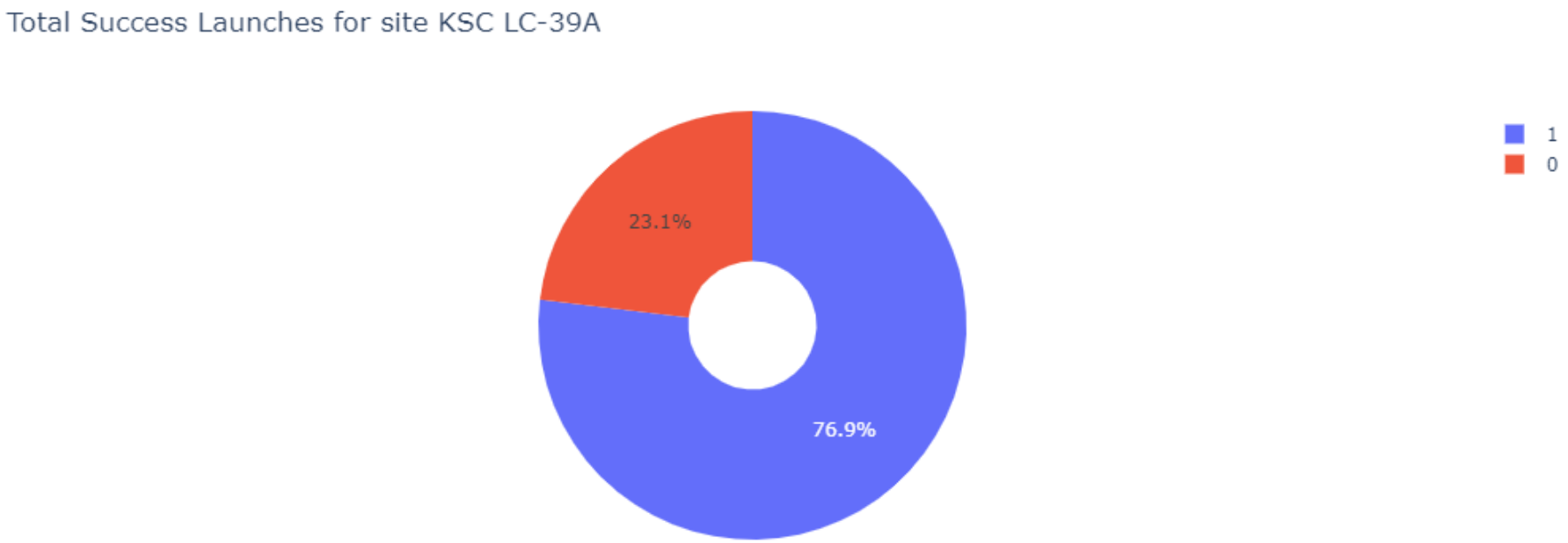

Section 4
Build a Dashboard with Plotly Dash

Successful Launches by Site



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

Total Successful Launches for Site KSC LC-39A

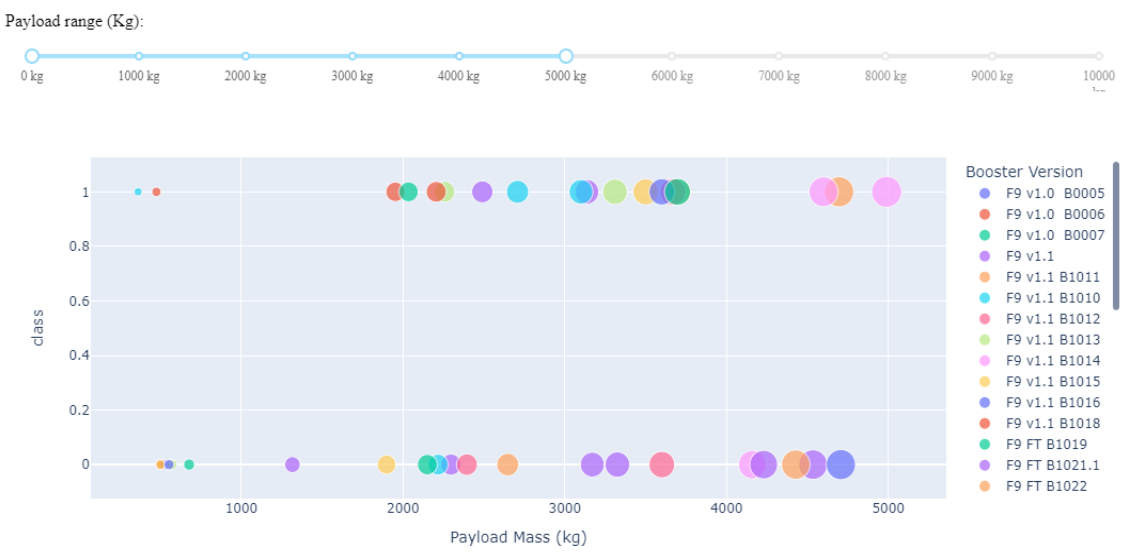


76.9% of the total launches at site KSC LC-39A were successful
(10 successful launches and 3 failed launches)

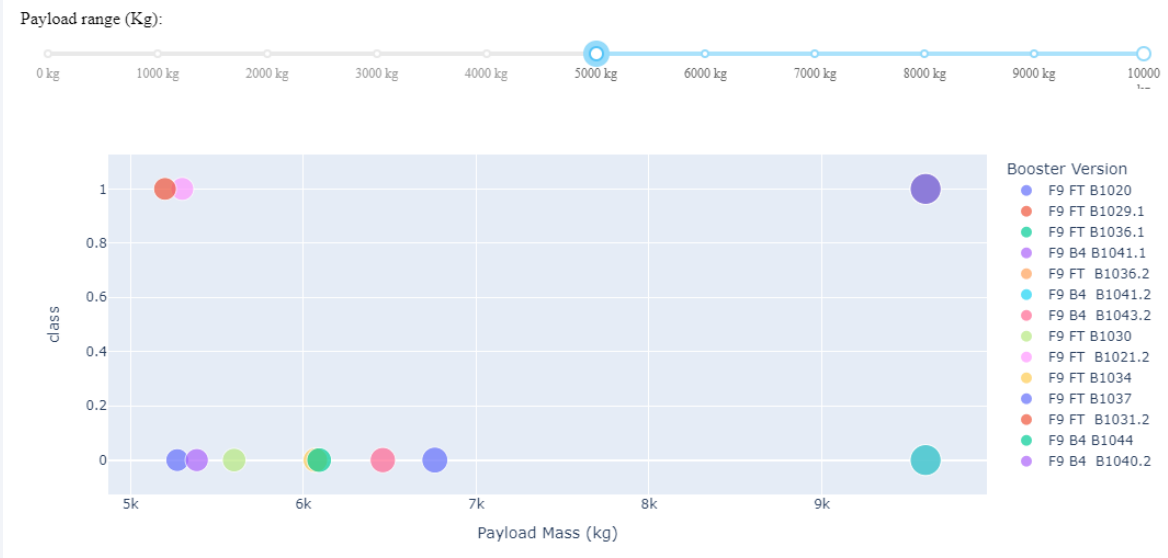
This is the highest success rate of all the different launch sites.

Payload Mass and Success

Low Weighted Payload 0kg – 5000kg



Heavy Weighted Payload 5000kg – 10000kg



The success rates for low weighted payloads is higher than the heavy weighted payloads

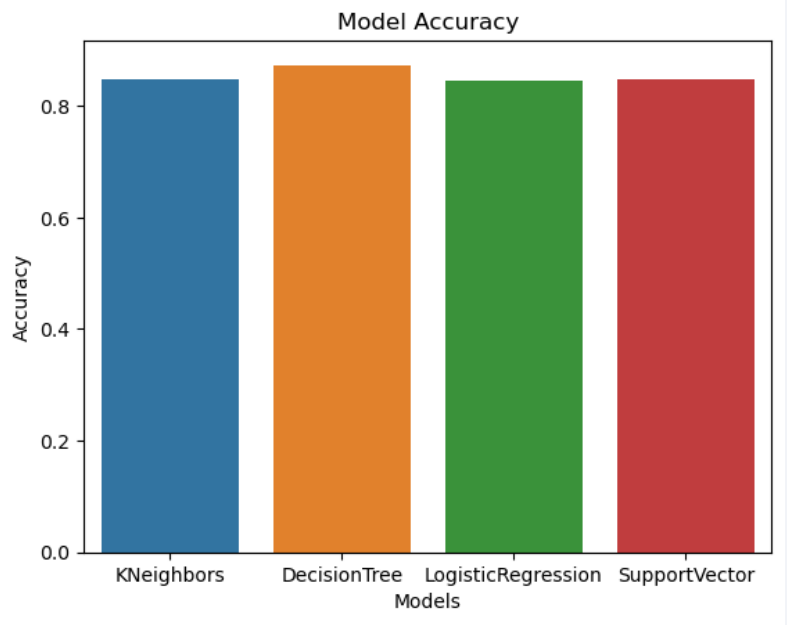
Section 5

Predictive Analysis (Classification)

Classification Accuracy

Accuracy

All the models achieved similar performance levels with identical scores and accuracy. This consistency is likely attributed to the small dataset. However, the Decision Tree model exhibited slightly better performance when considering the .best_score_, which represents the average across all cross-validation folds for a specific parameter combination.

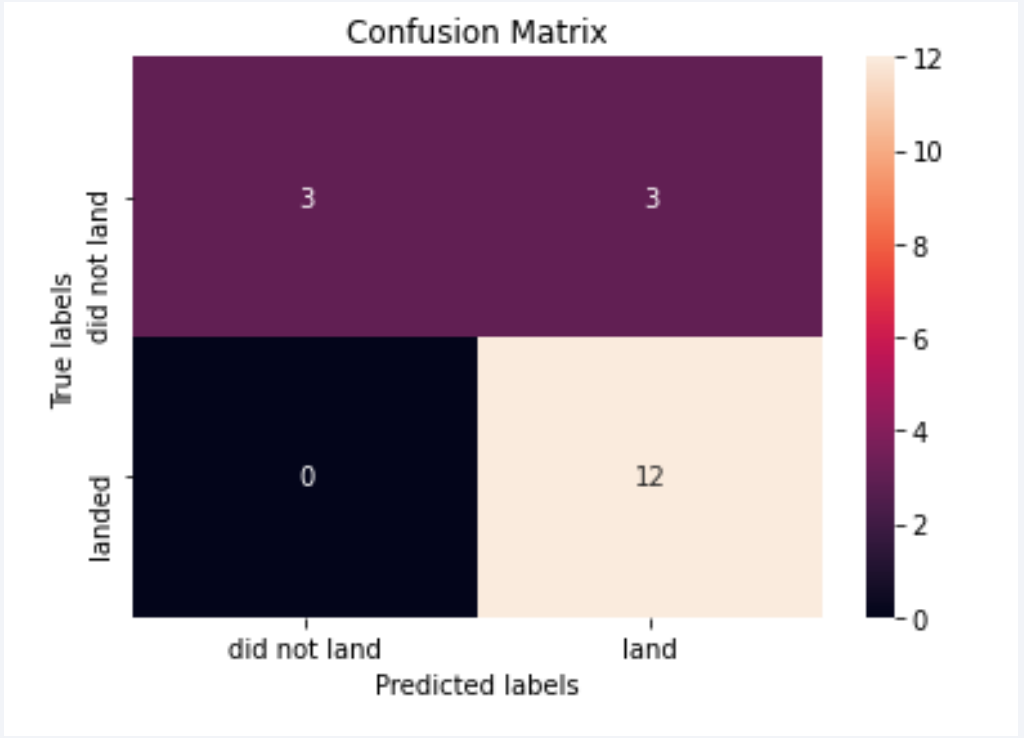



```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

After selecting the best hyperparameters for the decision tree classifier using the validation data, we achieved 83.33% accuracy on the test data.

Confusion Matrix

The decision tree classifier's confusion matrix indicates that it effectively distinguishes between various classes. However, a notable issue arises with false positives—instances where unsuccessful landings are incorrectly classified as successful by the model.



- **Precision** = TP / (TP + FP) = 12 / 15 = .80
- **Recall** = TP / (TP + FN) = 12 / 12 = 1
- **F1 Score** = 2 * (Precision * Recall) / (Precision + Recall) = 2 * (.8 * 1) / (.8 + 1) = .89
- **Accuracy** = (TP + TN) / (TP + TN + FP + FN) = .833

Conclusions

To compete with SpaceX, let's delve into an overview of their successful methods.

- Their launch sites are strategically located near coastlines, away from urban areas, allowing them to test rocket landings with minimal interference.
- Most of these launch sites are positioned near the equator, leveraging Earth's rotational speed to reduce the need for extra fuel and boosters.
- Across all launch sites, higher payload masses correlate with higher success rates.
- Site KSC LC-39A boasts the highest launch success rate among all SpaceX sites.
- Over time, SpaceX's launch success rates improve as they refine their processes.
- Orbits GEO, HEO, SSO, and ES-L1 exhibit the best success rates.

The data collected from these observations was used to train a machine learning model, achieving an 83.33% accuracy in predicting rocket landing outcomes.

Thank you!