Winning Space Race with Data Science

<Brandon Wong>
<09/21/2023>

Outline

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

Executive Summary

In this culminating project, we will endeavor to forecast the successful landing of the SpaceX Falcon 9 first stage by leveraging multiple classification algorithms within the realm of machine learning.

The primary stages of this endeavor encompass:
1. Data acquisition, refinement, and structuring.
2. In-depth exploration and analysis of the data.
3. Creation of interactive data visualizations.
4. Employing machine learning for predictive modeling.

Our visual representations demonstrate a correlation between certain characteristics of rocket launches and their outcomes, specifically, whether they are successful or not.
Furthermore, it is deduced that the decision tree algorithm appears to be the most promising choice among machine learning models for forecasting the successful landing of the Falcon 9 first stage.

Introduction

- Within this capstone project, our primary aim is to forecast the successful landing of the Falcon 9 first stage. SpaceX promotes Falcon 9 rocket launches on their website at a cost of 62 million dollars, a substantially lower figure compared to other providers, whose prices soar to as much as 165 million dollars per launch. This significant cost advantage largely stems from SpaceX's ability to recycle the first stage of the rocket. Consequently, by ascertaining whether the first stage will achieve a successful landing, we can effectively determine the overall cost of a launch. This valuable information can prove instrumental for alternative companies seeking to compete with SpaceX in rocket launch bids.
- It's worth noting that most of the unsuccessful landings are intentional and planned, with SpaceX occasionally opting for controlled ocean landings.
- In essence, the central question we endeavor to address is as follows: Given a specific set of characteristics related to a Falcon 9 rocket launch, encompassing parameters such as payload mass, orbit type, launch site, and more, can we predict whether the first stage of the rocket will successfully achieve its landing?

Section 1

Methodology

Methodology

- The comprehensive approach encompasses the following key steps:

1. Gathering, cleansing, and organizing data, achieved through:
    1. Utilizing the SpaceX API
    2. Employing web scraping techniques
2. Conducting exploratory data analysis (EDA), facilitated by:
    1. Leveraging Pandas and NumPy
    2. Utilizing SQL for data manipulation
3. Creating data visualizations using a range of tools and libraries, including:
    1. Matplotlib and Seaborn for traditional data visualization
    2. Folium for geographical data representation
    3. Dash for interactive data displays
4. Employing various machine learning models for predictive analysis, including:
    1. Logistic regression
    2. Support vector machine (SVM)
    3. Decision tree
    4. K-nearest neighbors (KNN)

- These methodologies are instrumental in our quest to predict the successful landing of the Falcon 9 first stage and provide valuable insights into the cost estimation of rocket launches.

Data Collection

- Web Scraping Details:
- The data is obtained through web scraping from the following URL: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922.
- The web page exclusively contains information related to Falcon 9 launches.
- Upon completion of the web scraping process, we acquire a dataset comprising 121 rows or instances and 11 columns or features. The initial rows of the dataset are illustrated in the image below

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- https://github.com/brandon-s-wong/SpaceX-Falcon-9-Landing-Prediction-Model/blob/main/SpaceX%20Falcon%209%20First%20Stage%20Landing%20Prediction.ipynb

1. Get request for rocket launch data using API

In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [7]:
```
response = requests.get(spacex_url)
```

2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```

Data Collection - Scraping

- Applied web scraping to webscrape Falcon 9 launch records
- Parsed table and converted into pandas
- https://github.com/brandon-s-wong/SpaceX-Falcon-9-Landing-Prediction-Model/blob/main/SpaceX%20Falcon%209%20First%20Stage%20Landing%20Prediction.ipynb

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]:
```
200
```

2. Create a BeautifulSoup object from the HTML response

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

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

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Extract all column names from the HTML table header

In [10]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

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

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

Data Wrangling

- Subsequent to data retrieval, thorough preprocessing is conducted to ensure data quality. This involves eliminating missing entries and applying one-hot encoding to categorical features.
- An additional column, denoted as 'Class,' is introduced into the data frame. The 'Class' column assigns a value of 0 to launches classified as failures and 1 to those deemed successful.
- Ultimately, the dataset is refined to consist of 90 rows or instances and 83 columns or features, after data cleansing and transformation processes have been carried out.
- https://github.com/brandon-s-wong/SpaceX-Falcon-9-Landing-Prediction-Model/blob/main/SpaceX%20Falcon%209%20First%20Stage%20Landing%20Prediction.ipynb

EDA with Data Visualization

- Explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- Each needed a specific type of chart to compare with each other, such as launch success over time
- https://github.com/brandon-s-wong/SpaceX-Falcon-9-Landing-Prediction-Model/blob/main/SpaceX%20Falcon%209%20First%20Stage%20Landing%20Prediction.ipynb

EDA with SQL

- We successfully imported the SpaceX dataset directly into a PostgreSQL database from within our Jupyter notebook environment. This allowed us to seamlessly integrate the data for analysis. Our exploratory data analysis (EDA) was conducted using SQL queries, providing us with valuable insights from the dataset.
- Some of the key insights we obtained include:

1. We identified the unique launch site names involved in space missions.
2. Calculated the total payload mass carried by boosters launched under NASA's Commercial Resupply Services (CRS) program.
3. Determined the average payload mass carried by booster version F9 v1.1.
4. Counted the total number of missions categorized as successful and those resulting in failure.
5. Identified instances of failed landings on drone ships, along with the associated booster version and launch site names.

https://github.com/brandon-s-wong/SpaceX-Falcon-9-Landing-Prediction-Model/blob/main/SpaceX%20Falcon%209%20First%20Stage%20Landing%20Prediction.ipynb

Build an Interactive Map with Folium

- In our data visualization process, we implemented various map objects and markers on a Folium map to represent the success or failure of launches at different launch sites. Here's a summary of the key steps and findings:

1. **Mapping Launch Outcomes:**
    1. We utilized markers, circles, and lines to visually depict the outcomes (success or failure) of launches for each launch site on the Folium map.
    2. The launch outcomes were assigned binary labels: 0 for failure and 1 for success.
2. **Identifying High Success Rate Launch Sites:**
    1. By examining the color-labeled marker clusters on the map, we identified launch sites with relatively high success rates. These sites were characterized by a higher concentration of success markers.
3. **Calculating Distances:**
    1. We computed the distances between each launch site and its surrounding features, such as railways, highways, coastlines, and cities.
4. **Proximity to Railways, Highways, and Coastlines:**
    1. Our analysis revealed whether launch sites are situated near railways, highways, or coastlines. This information is valuable for assessing safety and accessibility factors.
5. **Distance from Cities:**
    1. We examined whether launch sites maintain a certain distance from populated areas, providing insights into safety regulations and urban planning considerations.

- By visualizing these aspects on the Folium map and conducting distance calculations, we gained a comprehensive understanding of the geographic context and safety measures associated with SpaceX launch sites. This information can be pivotal for ensuring the safe and efficient operation of rocket launches.

https://github.com/brandon-s-wong/SpaceX-Falcon-9-Landing-Prediction-Model/blob/main/SpaceX%20Falcon%209%20First%20Stage%20Landing%20Prediction.ipynb

Build a Dashboard with Plotly Dash

- In our project, we harnessed the capabilities of Plotly Dash to construct an interactive dashboard. This dashboard provided an engaging and dynamic platform for data exploration and visualization. Here are some of the key components and visualizations integrated into the dashboard:

1. **Total Launches by Sites - Pie Charts:**
    1. We incorporated pie charts to visualize the distribution of total launches across various launch sites. This allowed for a quick and intuitive understanding of the launch frequency at each site.
2. **Relationship between Outcome and Payload Mass (Kg) - Scatter Graphs:**
    1. Scatter graphs were employed to illustrate the relationship between launch outcomes (success or failure) and the payload mass (measured in kilograms) for different booster versions. This visualization helped identify patterns and trends in how payload mass influences the success or failure of launches.

- The interactive nature of the dashboard allowed users to interact with these visualizations, providing a dynamic and user-friendly experience for exploring the SpaceX dataset.
- By incorporating Plotly Dash, we not only enhanced the accessibility of our data but also enabled users to gain valuable insights and make data-driven decisions through interactive exploration.

https://github.com/brandon-s-wong/SpaceX-Falcon-9-Landing-Prediction-Model/blob/main/SpaceX%20Falcon%209%20First%20Stage%20Landing%20Prediction.ipynb

Predictive Analysis (Classification)

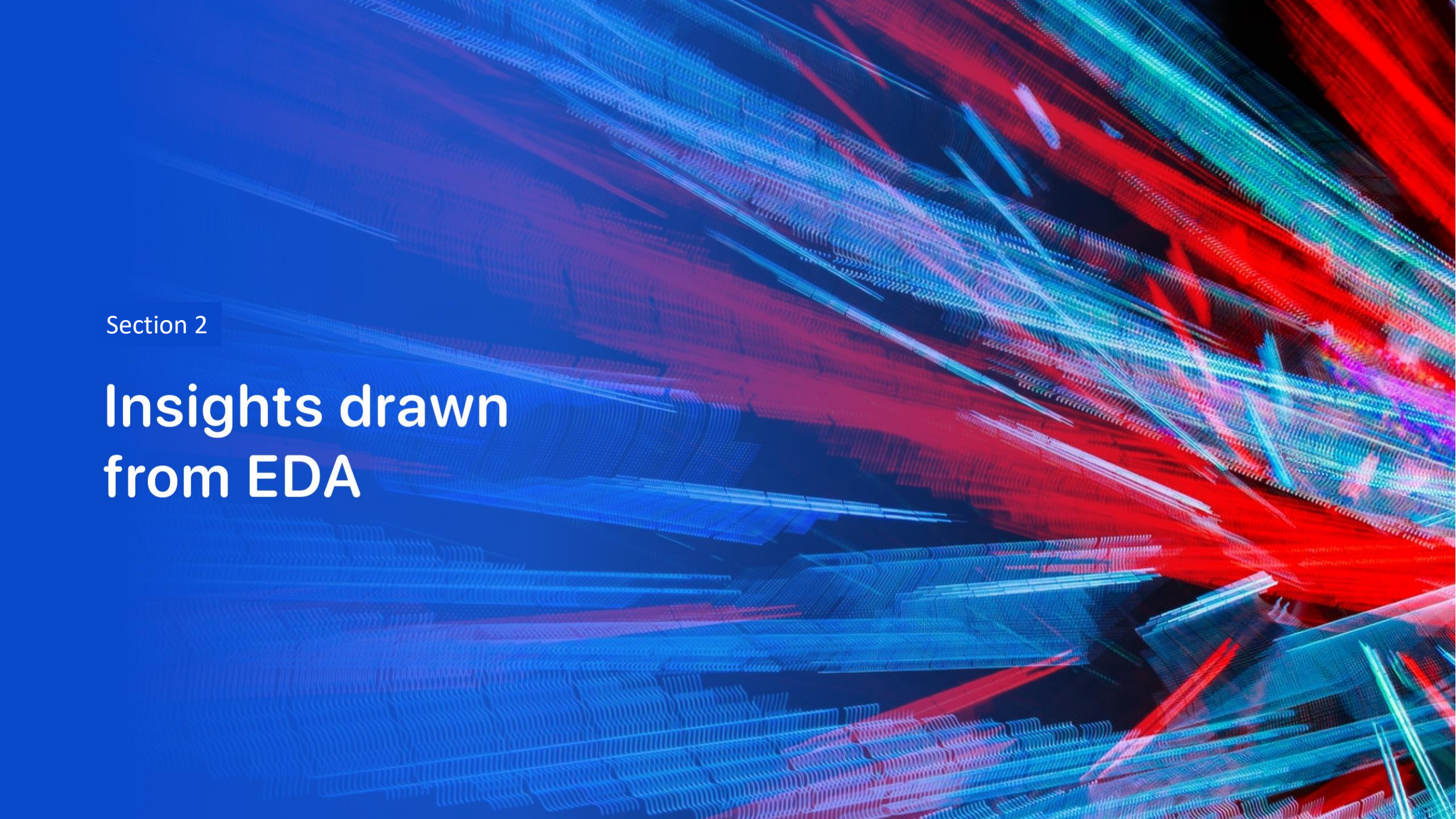
- In our data analysis and machine learning process, we followed a systematic approach to develop and fine-tune classification models. Here's an overview of the key steps involved:

1. **Data Loading and Transformation:**
    1. We loaded the dataset using NumPy and Pandas, facilitating data manipulation and preprocessing.
    2. Data transformation techniques were applied to prepare the dataset for model training.
2. **Data Splitting:**
    1. We divided the dataset into training and testing subsets. This segregation allowed us to train our models on one portion of the data and evaluate their performance on another, ensuring a robust assessment.
3. **Model Building:**
    1. Multiple machine learning models were constructed to predict launch outcomes.
    2. We experimented with various algorithms to explore their predictive capabilities.
4. **Hyperparameter Tuning with GridSearchCV:**
    1. To enhance model performance, we utilized GridSearchCV to systematically search for optimal hyperparameters for each model.
    2. GridSearchCV helps identify the best combination of hyperparameters by cross-validating different parameter settings.
5. **Performance Metric - Accuracy:**
    1. We evaluated the models using accuracy as the primary performance metric. Accuracy measures the ratio of correctly predicted outcomes to the total predictions made by the model.
6. **Feature Engineering:**
    1. To further improve model performance, we engaged in feature engineering, which involves selecting, modifying, or creating features that enhance the model's ability to make accurate predictions.
7. **Algorithm Tuning:**
    1. Algorithm tuning was performed to fine-tune the models and optimize their predictive power.
    2. This process involved adjusting various model-specific parameters to achieve the best possible results.
8. **Selecting the Best Model:**
    1. After evaluating the models, we identified the best-performing classification model based on accuracy and potentially other relevant criteria.

https://github.com/brandon-s-wong/SpaceX-Falcon-9-Landing-Prediction-Model/blob/main/SpaceX%20Falcon%209%20First%20Stage%20Landing%20Prediction.ipynb

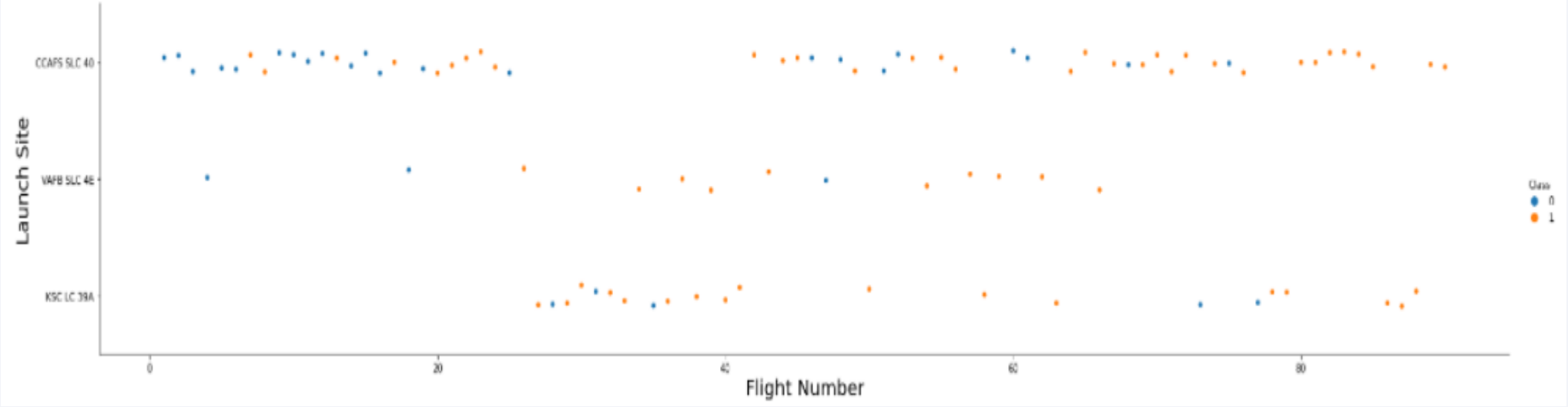
Results

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


Section 2
Insights drawn from EDA

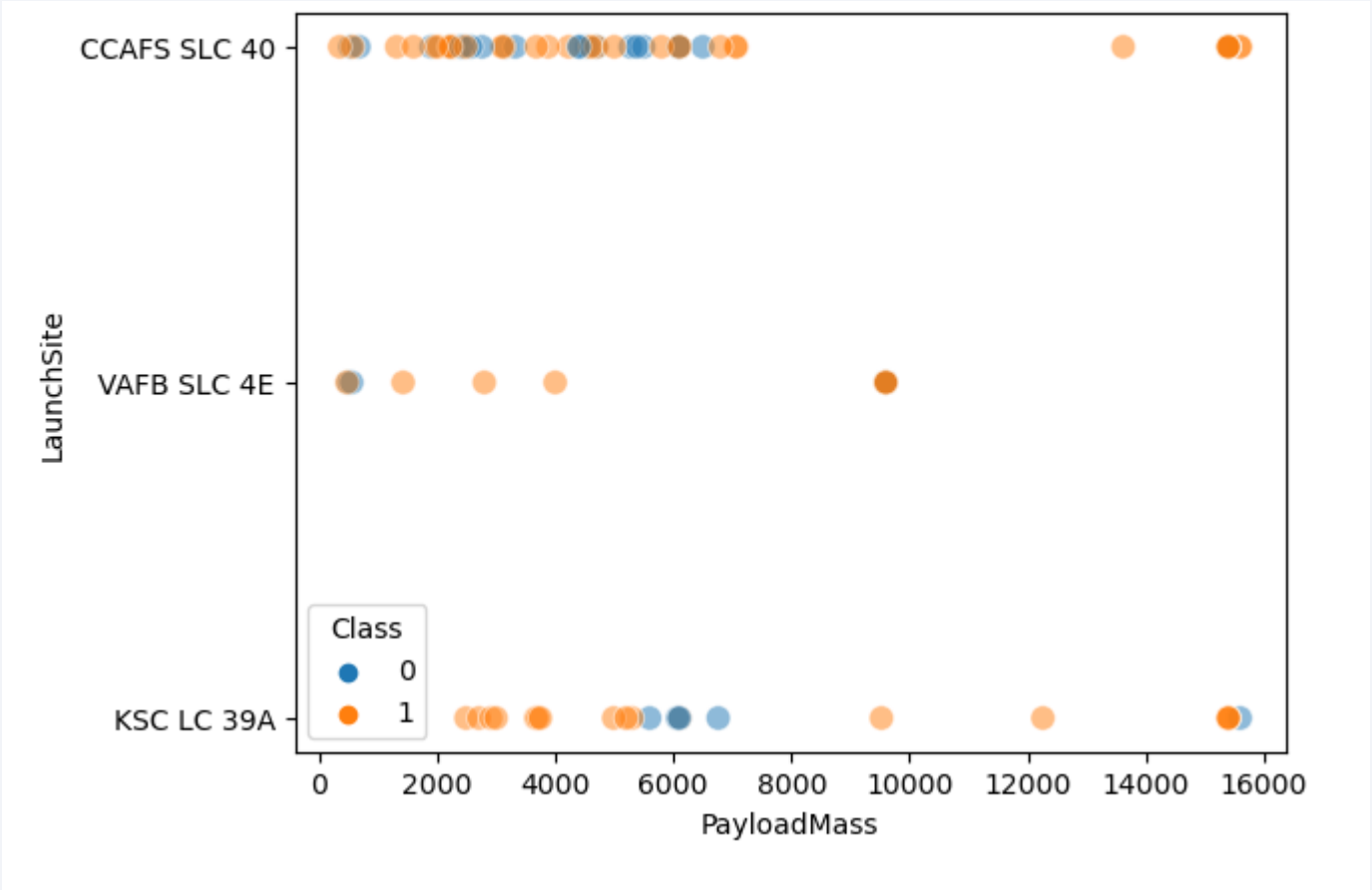
Flight Number vs. Launch Site

- We found that the larger the flight at a launch flight, the greater the success rate



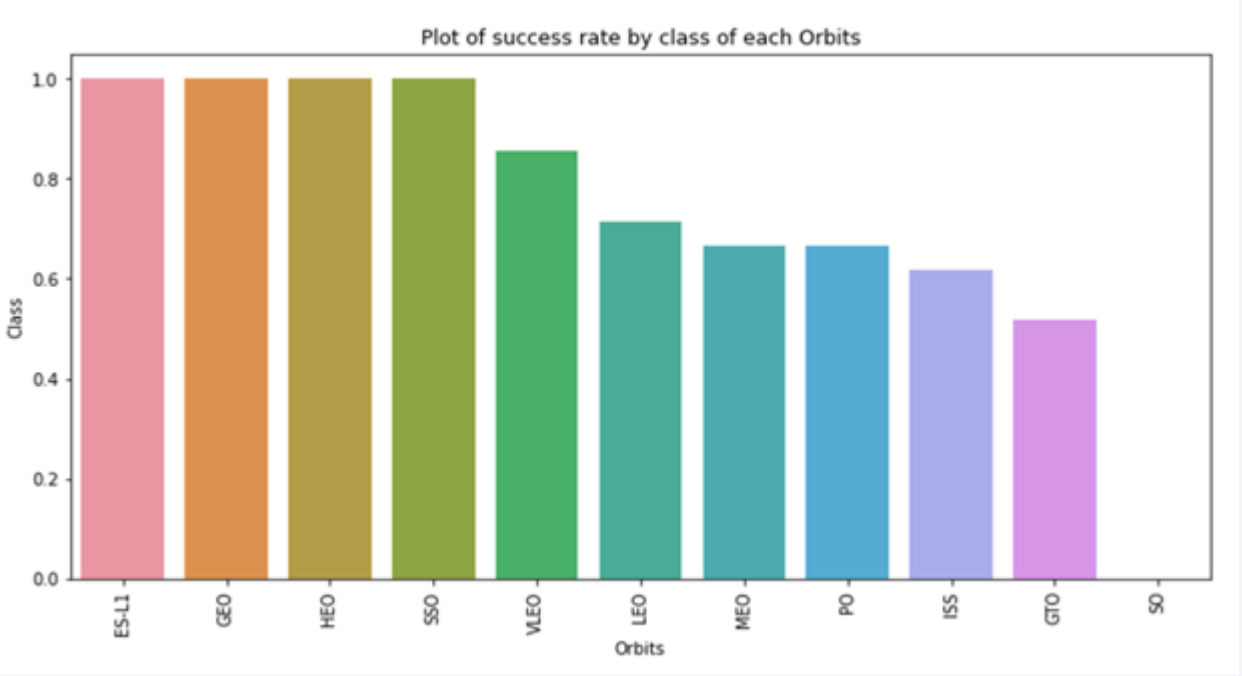
Payload vs. Launch Site

- CAAFS SLC 40 Launchsite had greater payload at 160000



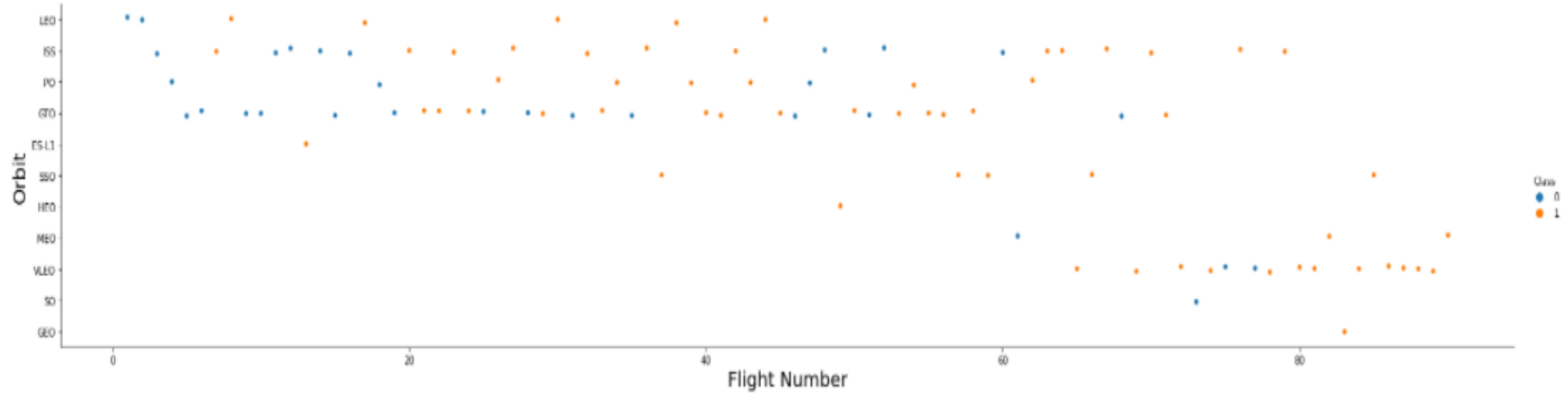
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO. SSO, VLEO had highest success rate



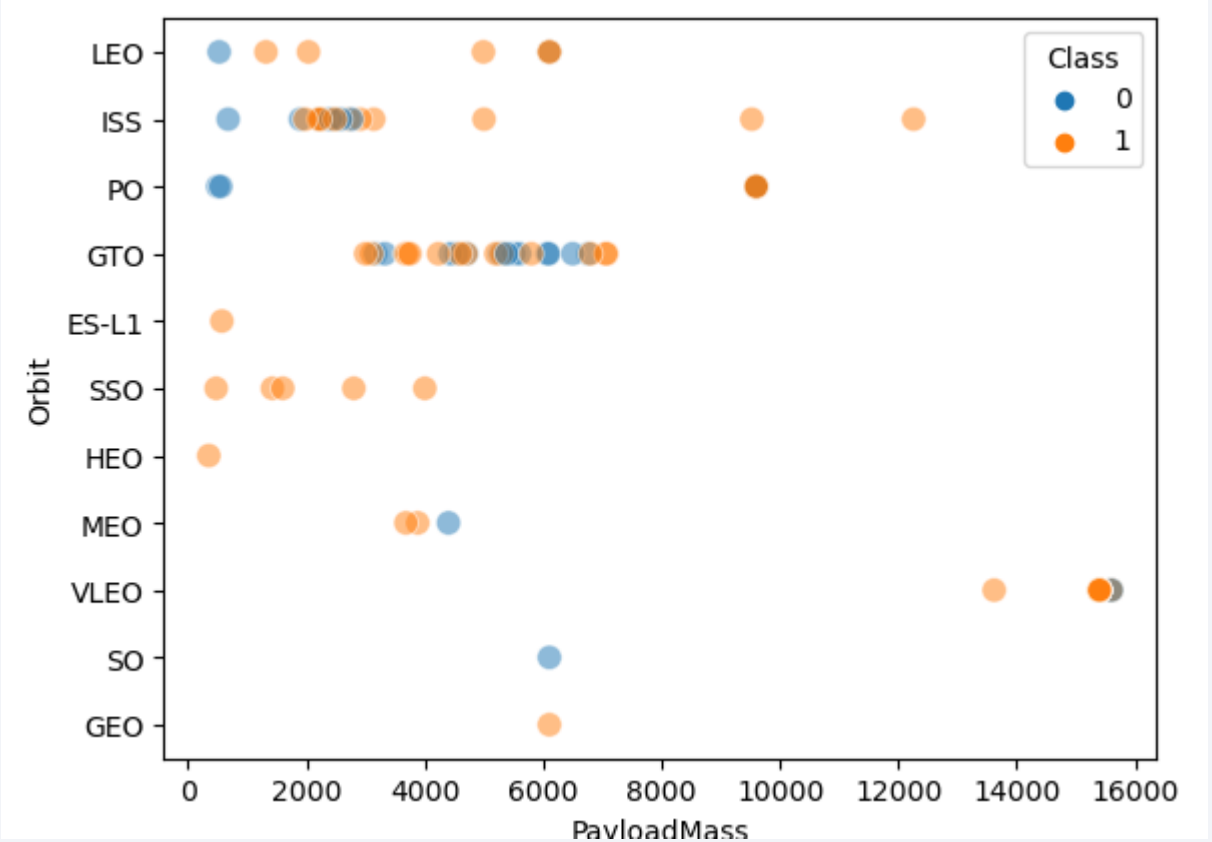
Flight Number vs. Orbit Type

- The plot below illustrates Flight Number vs. Orbit type. We notice that in the Low Earth Orbit (LEO), the success rate is correlated with the number of flights, while in the Geostationary Transfer Orbit (GTO), there is no discernible relationship between flight number and the orbit's success.



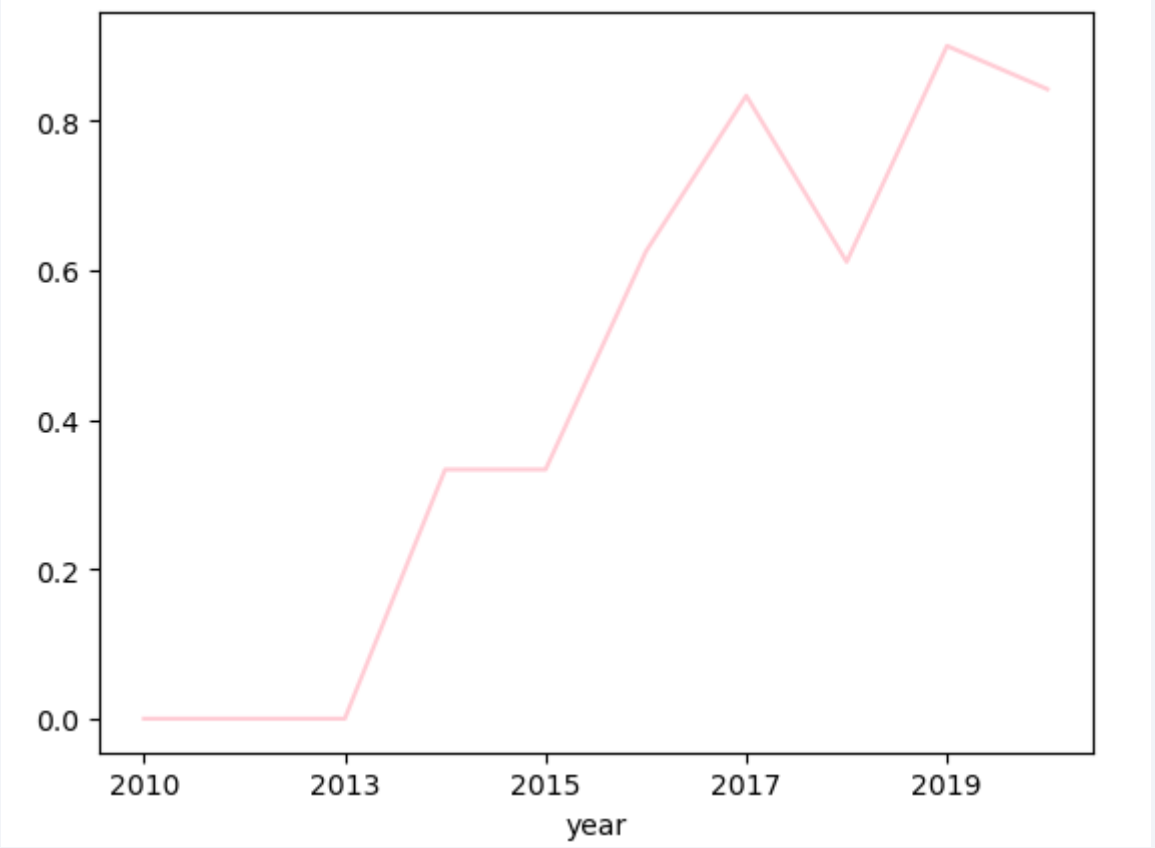
Payload vs. Orbit Type

- Heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- Success rate increases until 2018 and 2020



All Launch Site Names

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

In [63]:
```
%%sql

SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL ORDER BY 1;
```

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

Out[63]:

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

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with 'CCA'

In [64]:

```
%%sql

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

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

Total Payload Mass

Display total payload mass carried by boosters launched by NASA

In [66]:
```
%%sql

SELECT SUM (PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE CUSTOMER='NASA (CRS)';
```

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

Out[66]:

SUM (PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

In [67]:
```
%%sql

SELECT AVG (PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Booster_Version like 'F9 v1.1%';
```

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

Out[67]:

AVG (PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

List date when first successful landing outcome in ground pad was achieved

In [68]:

```
%%sql

SELECT MIN(Date)
FROM SPACEXTBL
WHERE Mission_outcome LIKE 'Success%';
```

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

Total Number of Successful and Failure Mission Outcomes

List total number of successful and failure mission outcomes

In [76]:
```
%%sql

SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME)
FROM SPACEXTBL
WHERE MISSION_OUTCOME like 'Success%'
```

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

Out[76]:

Mission_Outcome	COUNT(MISSION_OUTCOME)
Success	100

Boosters Carried Maximum Payload

List names of boosterversions which have carried max payload mass

[77]:
```
%%sql

SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

t[77]:

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 records for launch sites in the months in year 2015

In [78]:

```
%%sql

SELECT BOOSTER_VERSION,LAUNCH_SITE,LANDING_OUTCOME
FROM SPACEXTBL
WHERE LANDING_OUTCOME LIKE "Failure%" and DATE('2015')
```

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

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

Rank the count of landing outcomes between dates 2010 - 2017 in descending order

In [61]:
```
%%sql

SELECT landing_outcome, COUNT(*) AS "Count"
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' and '2017-03-20'
GROUP BY landing_outcome
ORDER BY Count DESC;
```

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

Out[61]:

Landing_Outcome	Count
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

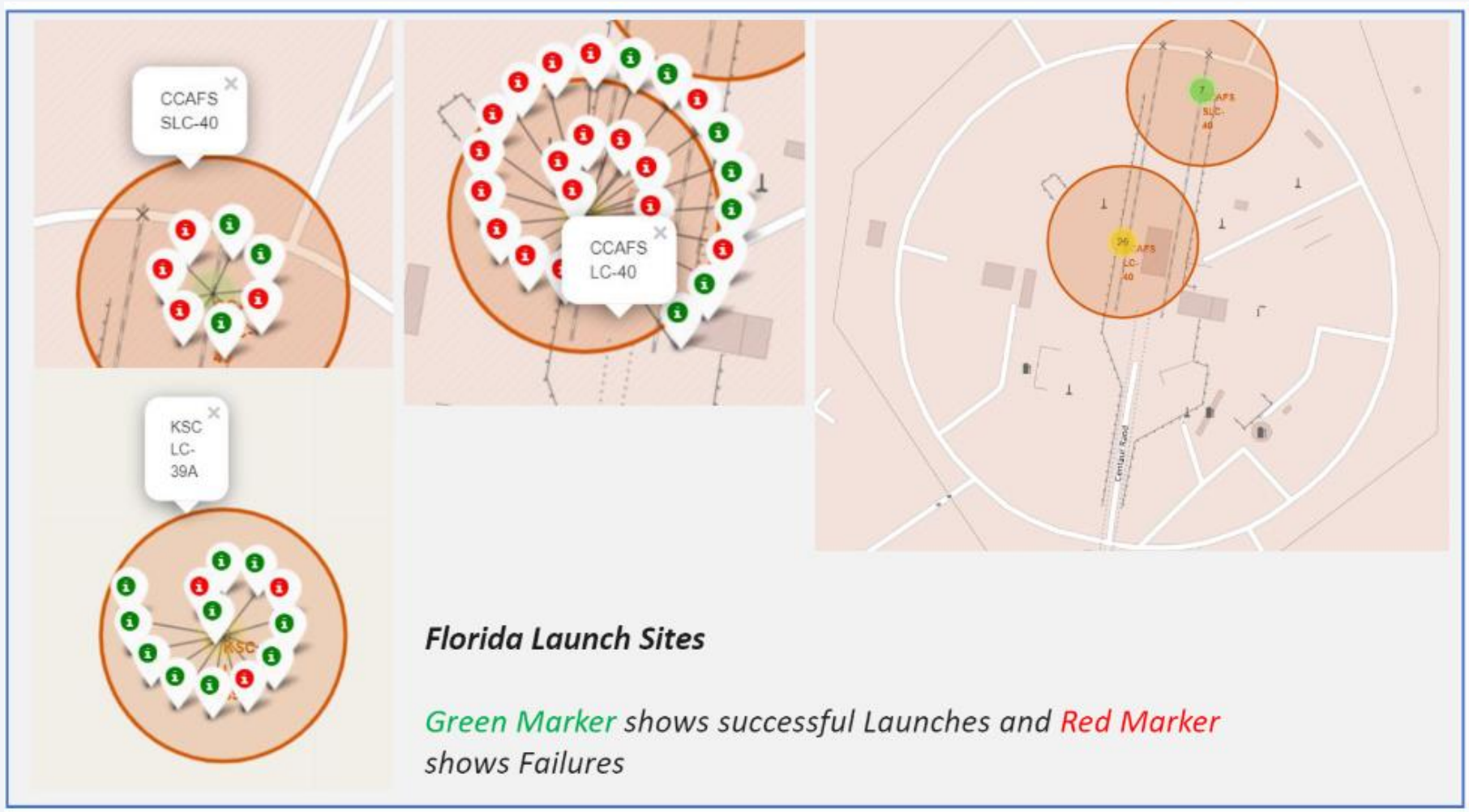
Section 3

Launch Sites Proximities Analysis

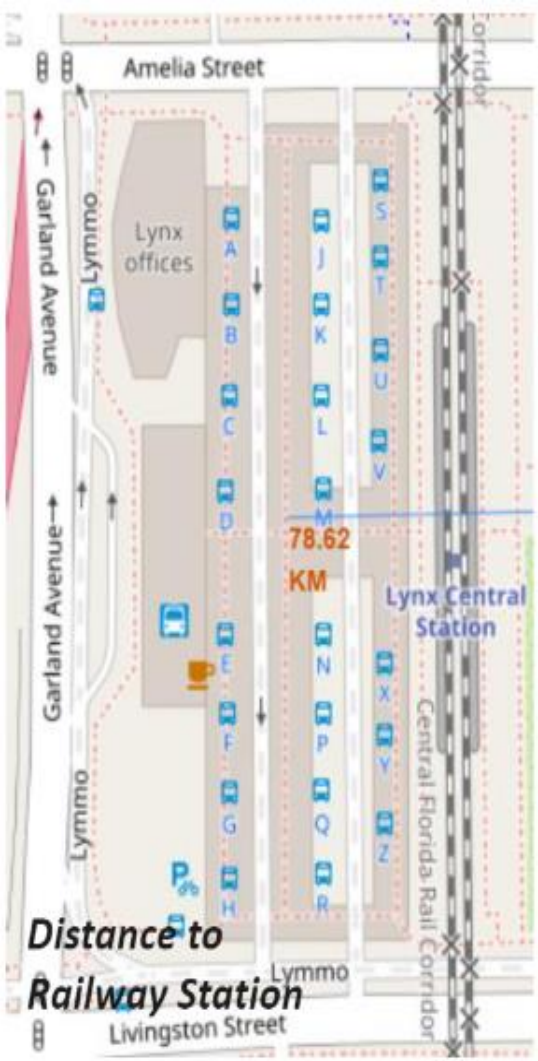
Global map markers

We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

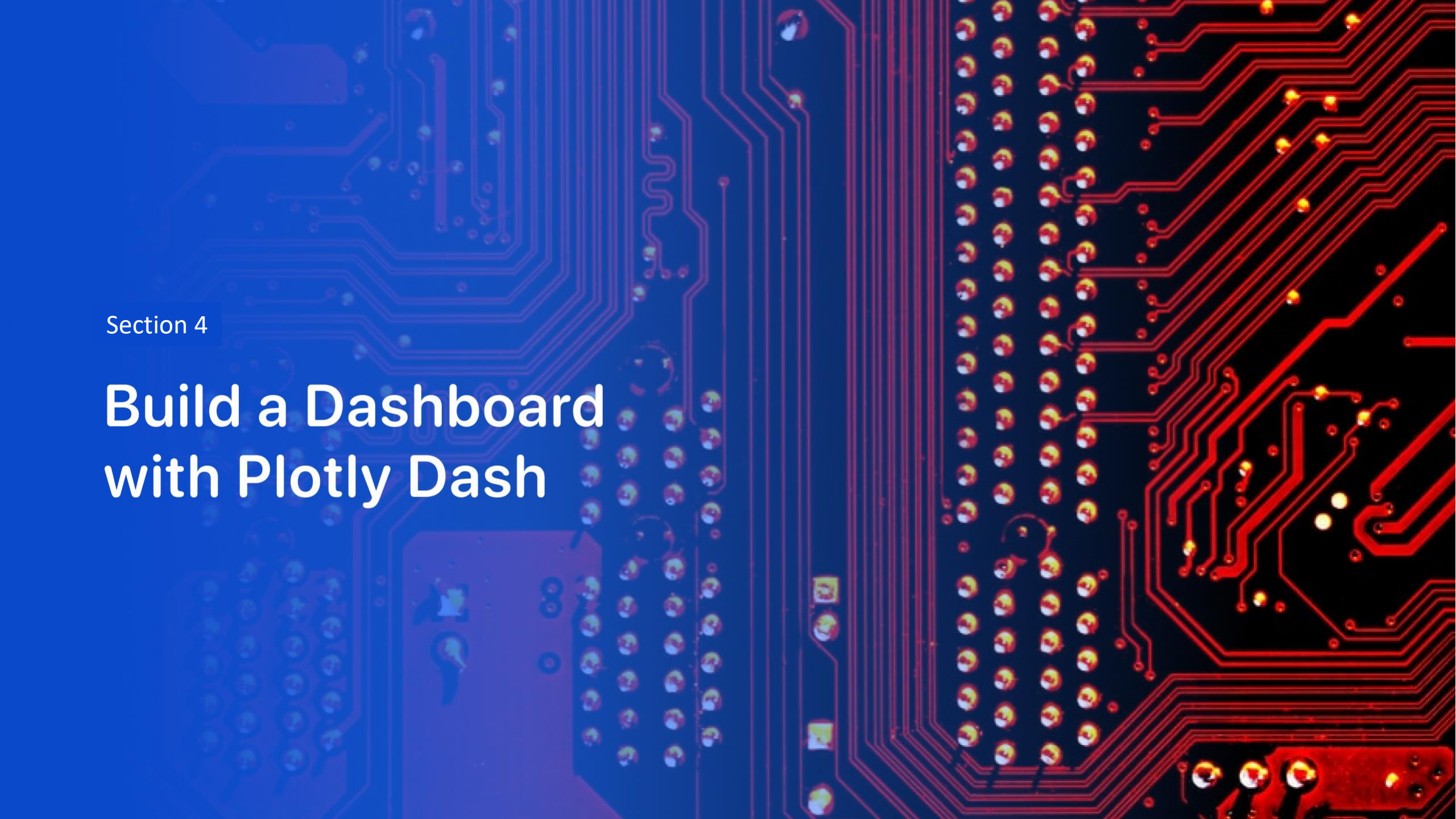
Launch sites with color



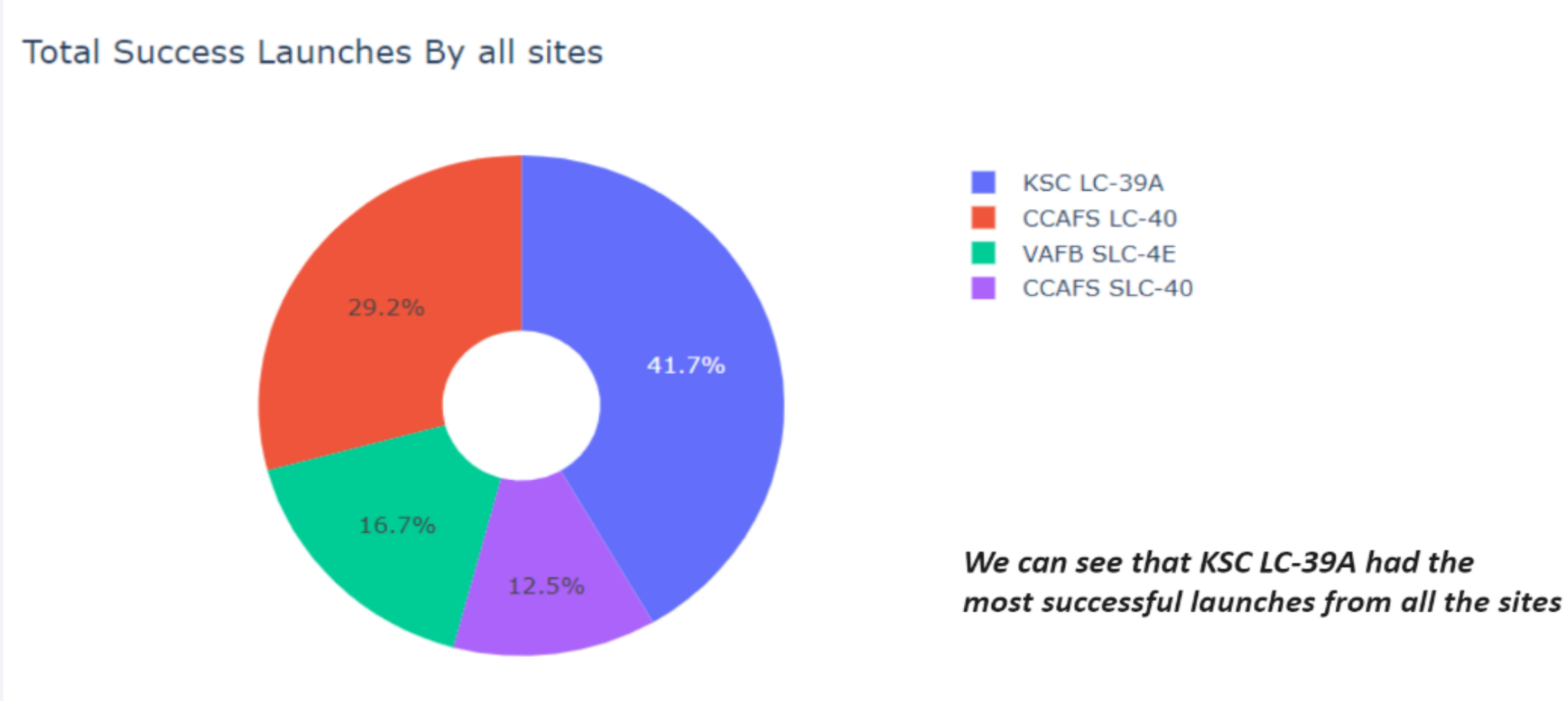
Distance to landmarks



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

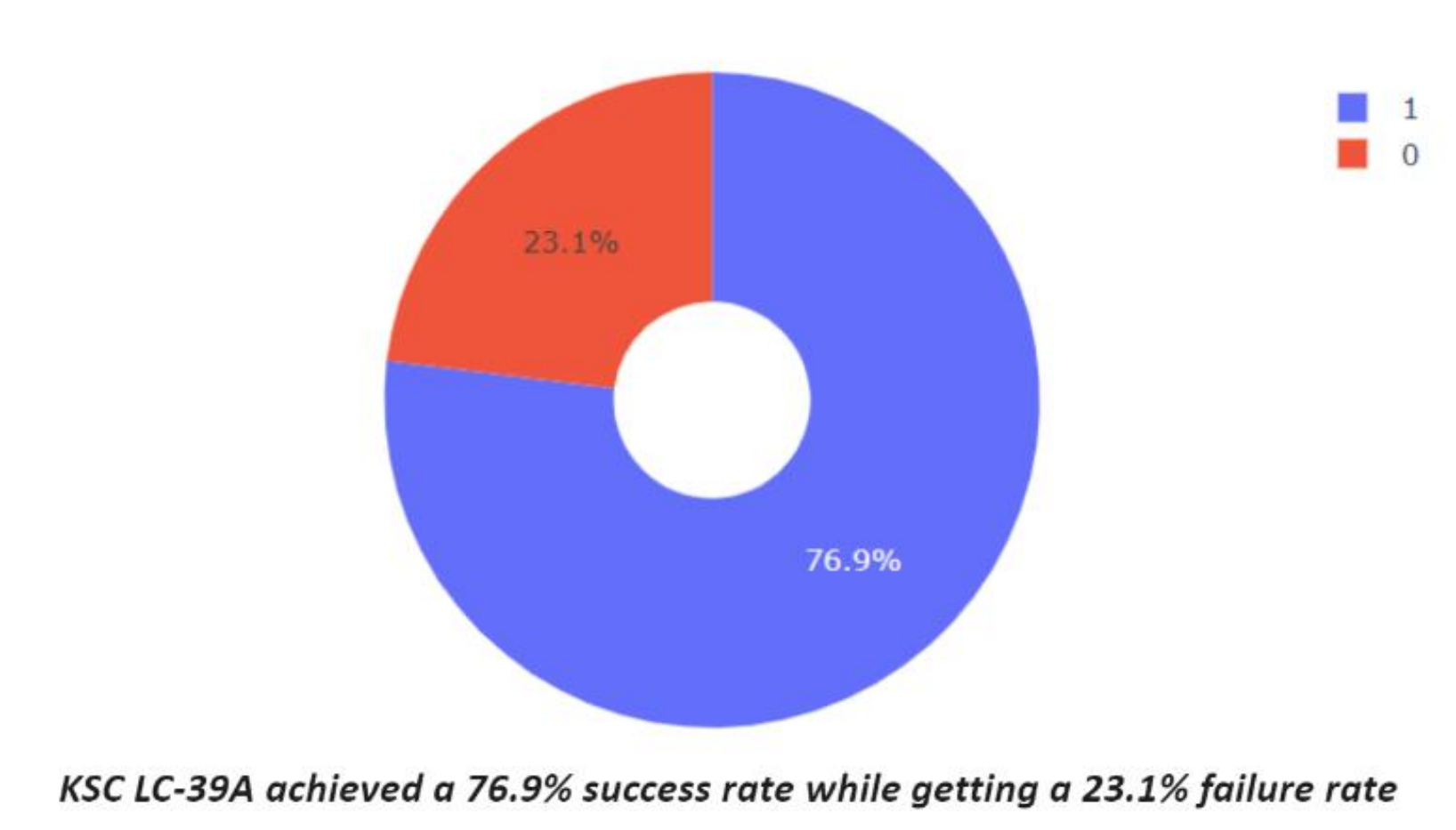

Section 4
Build a Dashboard with Plotly Dash

Pie chart with success percentage



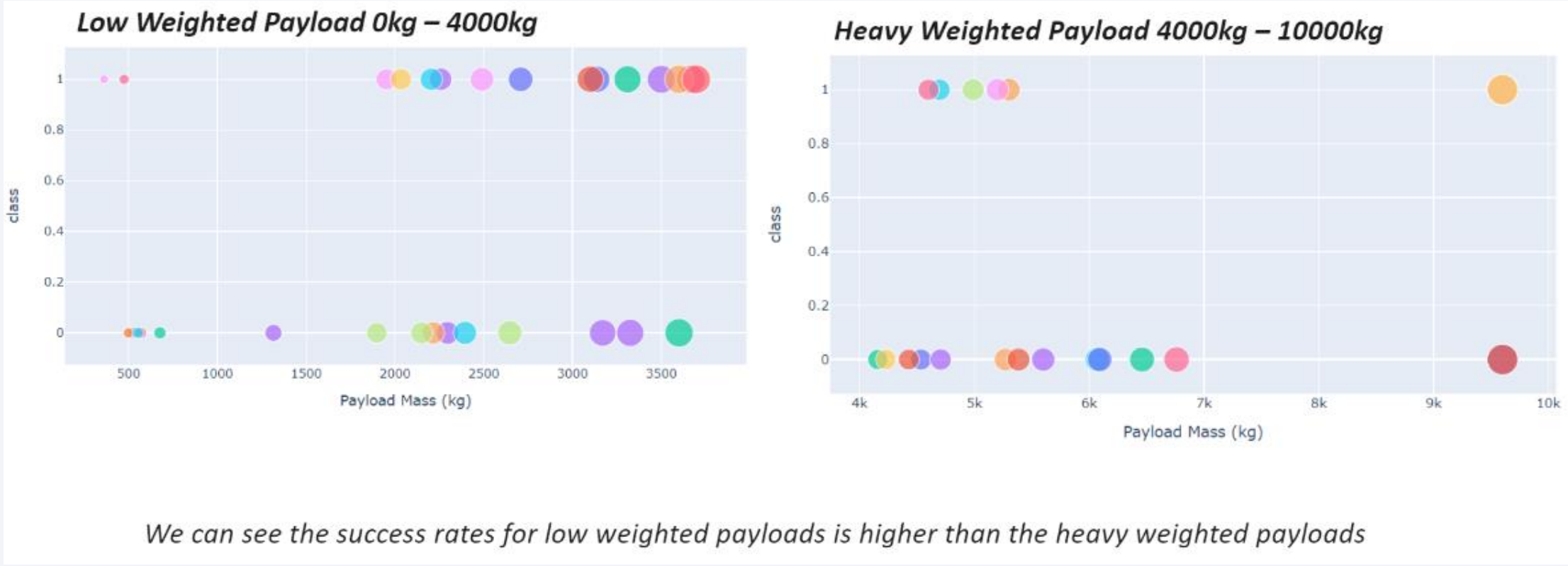
We can see that KSC LC-39A had the most successful launches from all the sites

Pie chart with Launch site



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Scatter of Payload vs Launch



We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Decision tree classifier is model with highest classification accuracy

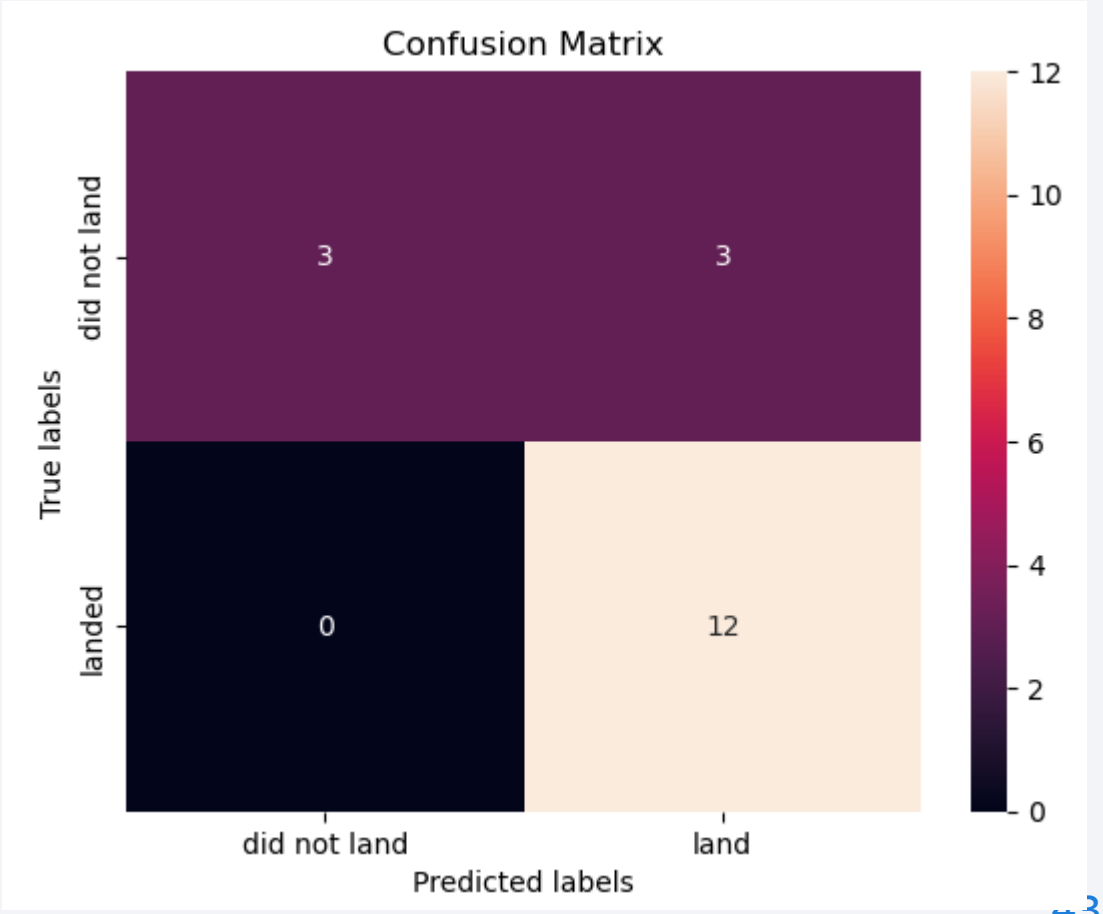
Create a decision tree classifier, create a GridSearchCV object, and fit the object to find the best parameters

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()

tree_cv = GridSearchCV(tree, param_grid=parameters,scoring='accuracy', cv=10)
tree_cv.fit(X_train, Y_train)
tree_cv.best_params_

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

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 14, 'max_features': 'sqrt', 'min_sam
ples_leaf': 4, 'min_samples_split': 5, 'splitter': 'random'}
accuracy : 0.875
```

Confusion Matrix

- The confusion matrix for the decision tree classifier indicates its ability to differentiate between distinct classes. However, a notable challenge lies in the occurrence of false positives, where the classifier incorrectly labels unsuccessful landings as successful ones.



Conclusions

- Based on our analysis, we can draw the following conclusions:

1. A positive correlation exists between the number of flights at a launch site and its success rate. As the flight count increases, so does the likelihood of a successful launch.
2. The launch success rate demonstrated an upward trend starting from 2013 and continued until 2020.
3. The orbits ES-L1, GEO, HEO, SSO, and VLEO exhibited the highest success rates, suggesting these orbits have a history of successful launches.
4. KSC LC-39A emerged as the launch site with the highest number of successful launches among all sites.
5. Based on our analysis, the Decision Tree classifier appears to be the most suitable machine learning algorithm for the task at hand, demonstrating strong predictive capabilities for SpaceX launch outcomes.

Thank you!