IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Barış Özcan
13.09.2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Model and Predictions
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics results

Introduction

SpaceX, founded in 2002 by Elon Musk, is an American aerospace manufacturer and space transportation company with the primary goal of reducing space transportation cost. A key innovation that has set SpaceX apart from other space companies is its development and successful implementation of re-landable rockets. By designing rockets that can return to Earth and be reused multiple times, SpaceX has revolutionized the economics of space travel, significantly reducing costs and paving the way for more frequent and sustainable missions to space.

In this Project, as a Data Scientist, our primary aim is to determine landing outcome of the first stage. To do so, we will examine different datasets and create a machine learning pipeline, which predicts the landing outcome.

There are important questions that we want to answer in this project, such as:

- Identifying every element that impacts the result of the landing.
- The relationship between different variables and their effect on the final outcome.
- The optimal circumstances required to boost the likelihood of a successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The first historical rocket launch data was collected by SpaceX REST API
  - The second historical rocket launch data was scraped from Wikipedia
- Perform data wrangling
  - Categorical features were converted into numerical features using one-hot encoding.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Different classifiers were tested through Sci-Kit Learn framework

Data Collection

The data was acquired through a diverse array of methodologies. Initially, data collection entailed the utilization of a GET request directed at the SpaceX API. Subsequently, the response content was decoded into a JSON format by invoking the json() function, followed by its transformation into a pandas dataframe via the json_normalize function.

Simultaneously, data cleansing procedures were applied, encompassing the identification and handling of missing values where necessary. Additionally, web scraping was executed to extract Falcon 9 launch records from Wikipedia, employing the BeautifulSoup library. The principal objective was to extract the launch records represented as an HTML table, subsequently parsing the table content and converting it into a Pandas DataFrame for subsequent analytical purposes.

Data Collection – SpaceX API

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

```
response = requests.get(spacex_url)
```

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

```
# Calculate the mean value of PayloadMass column
mean=data_falcon9.PayloadMass.mean()
print(mean)

# Replace the np.nan values with its mean value
data_falcon9.PayloadMass=data_falcon9.PayloadMass.replace(np.nan,mean)
```

Retrieve rocket launch data through an API using a GET request.

Utilize the **json_normalize** method for the transformation of the JSON result into a dataframe.

Conducted data cleansing and filled missing values.

From: https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/1-spacex-data-collection-api.ipynb

Data Collection - Scraping

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

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

```
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()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            #print(flight_number)
            datatimelist=date_time(row[0])

            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            #print(date)

            # Time value
            # TODO: Append the time into launch_dict with key `Time`
            time = datatimelist[1]
            #print(time)

            # Booster version
```

Retrieve the Falcon 9 Launch Wikipedia page using its URL.

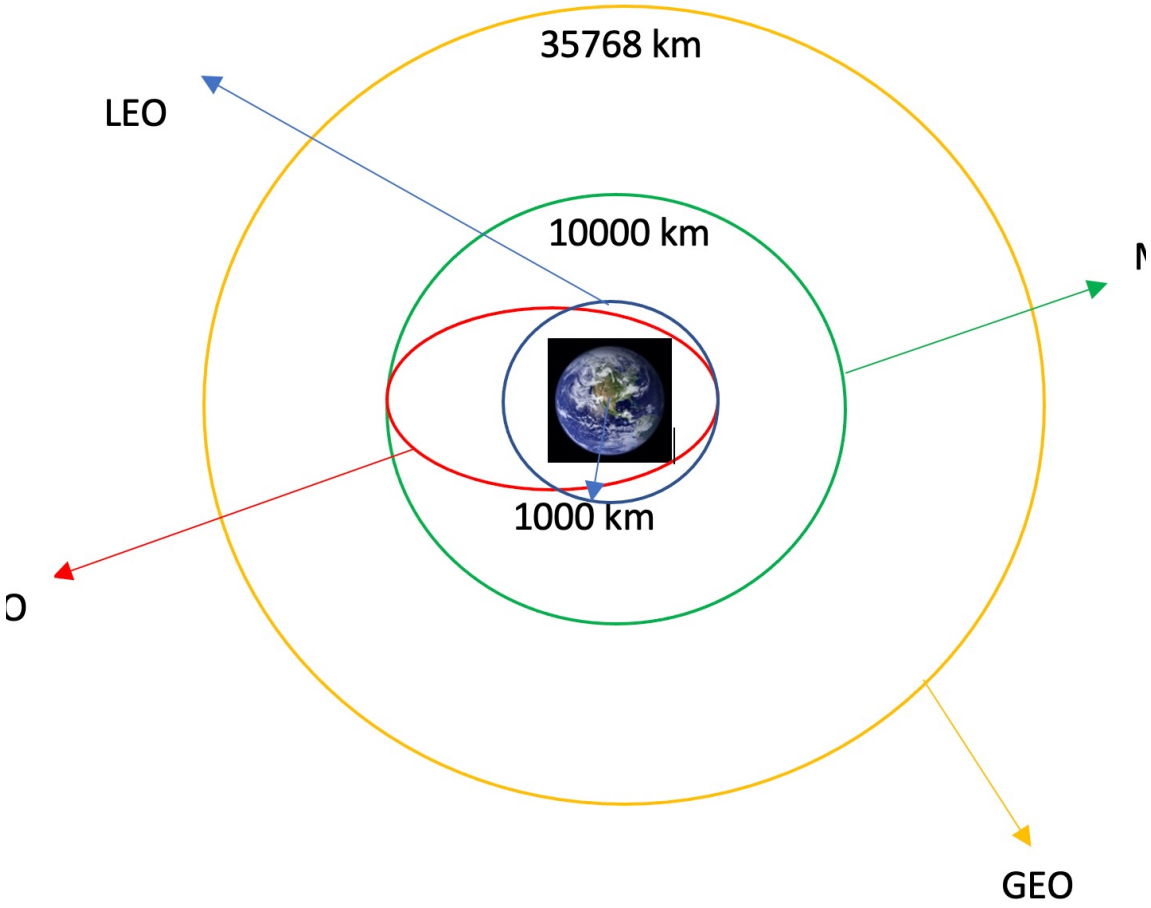
Generate a BeautifulSoup object from the HTML response.

Retrieve all column or variable names from the header of the HTML.

From: https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/2-jupyter-labs-webscraping.ipynb

Data Wrangling

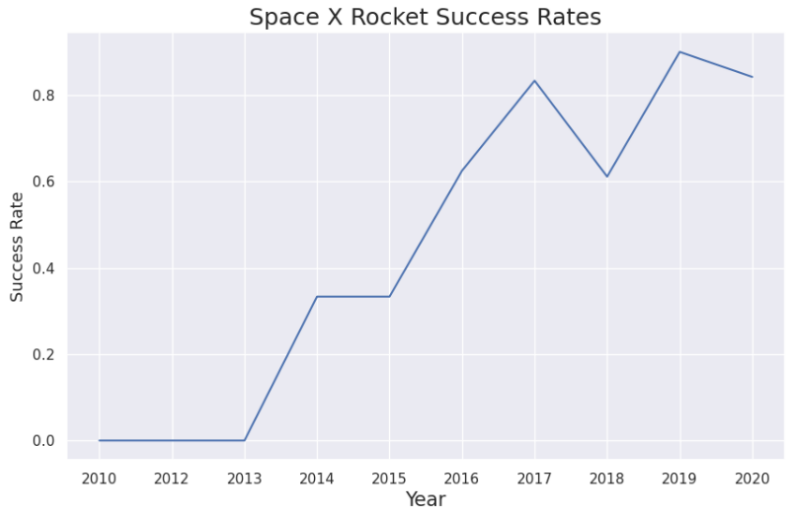
- Exploratory data analysis was conducted, and the training labels were determined.
- The number of launches at each site, along with the occurrence and quantity of each orbit, was calculated.
- A landing outcome label was created from the outcome column, and the results were exported to a CSV file.

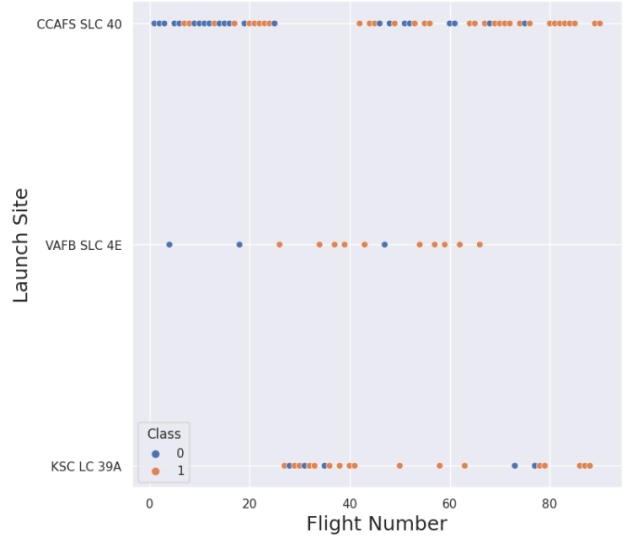


From: https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/3-labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

- Data exploration was performed 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, as well as the launch success yearly trend.





From: https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/5-jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

In the course of our investigation, multiple SQL queries were executed in order to enhance our comprehension of the dataset. These queries encompassed the following tasks:

- The presentation of launch site names.
- The retrieval of five records wherein launch sites commence with the alphanumeric sequence 'CCA.'
- The determination of the cumulative payload mass carried by boosters launched under the auspices of NASA, specifically those designated as 'CRS.'
- The calculation of the average payload mass transported by booster variants falling under the nomenclature 'F9 v1.1.'
- The compilation of dates marking the commencement of the initial instances of successful landing outcomes achieved on ground pads.
- The enumeration of booster appellations characterized by successful drone ship landings and payload masses exceeding 4000 yet remaining below 6000.
- The cataloging of the total count of mission outcomes categorized as either successful or unsuccessful.
- The listing of booster versions responsible for transporting the highest payload mass.
- The presentation of instances of failed landing outcomes on drone ships, including pertinent details such as booster versions and launch site names, all within the temporal confines of the year 2015.
- The arrangement and ranking of the count of landing outcomes, with a focus on success, observed during the period spanning from June 4, 2010, to March 20, 2017, arranged in a descending order.

Notebook URL: https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/4-jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

In our endeavor, a series of actions were taken to enhance the visual representation of launch site data on the Folium map. These actions involved:

- The tagging of all launch sites.
- The incorporation of map objects, including markers, circles, and lines, for the purpose of demarcating the outcomes of launch events, whether they be categorized as successes or failures, on the Folium map.

Furthermore, we executed the following steps:

- We assigned binary labels to the feature denoting launch outcomes, where '0' signified launch failures and '1' indicated successful launches.
- Utilizing color-coded marker clusters, we conducted an analysis to discern which launch sites exhibited comparatively high success rates in their launch operations.

Subsequently, the Haversine formula was applied to compute the distances between the launch sites and various landmarks. This computational approach was instrumental in addressing inquiries pertaining to the proximity of the launch sites in relation to the following:

- The closeness of the launch sites to railway networks, highways, and coastlines.
- The proximity of the launch sites to nearby urban centers or cities.

Notebook URL: https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/6-lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

Following these actions, an interactive dashboard was meticulously constructed using Plotly Dash, facilitating the dynamic exploration of pertinent data. This dashboard featured the following visualizations:

- Pie charts were generated to graphically depict the cumulative count of launches associated with specific launch sites.
- Scatter graphs were plotted to illustrate the correlation between the launch outcome (success or failure) and the payload mass (measured in kilograms) across various booster versions.

Python File URL: https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

Building the Model

The dataset was loaded into NumPy and Pandas, thereby facilitating data manipulation and analysis.

Data preprocessing techniques were applied to transform and prepare the dataset for modeling purposes. Subsequently, the dataset was partitioned into training and testing subsets to facilitate model evaluation.

A deliberation was undertaken to determine the appropriate type of machine learning algorithm to employ, taking into account the nature of the problem and the dataset characteristics.

Parameters and algorithms were configured for utilization within the GridSearchCV (Grid Search Cross-Validation) framework, which systematically explores various hyperparameter settings to optimize the model's performance. This optimized model was then fitted to the dataset, culminating in the model's training and fine-tuning process.

Evaluating the Model

The accuracy of each trained model was computed, providing insight into their predictive capabilities.

Tuned hyperparameters were obtained for each type of algorithm, reflecting the settings that yielded optimal performance during the GridSearchCV process.

Confusion matrices were generated and subsequently plotted, allowing for a visual representation of the model's classification performance, particularly in terms of true positives, true negatives, false positives, and false negatives. This aided in the assessment of the model's precision, recall, and overall classification accuracy.

Boosting the Model

Employing Feature Engineering and Algorithm Tuning involves the strategic manipulation and enhancement of dataset features, as well as the fine-tuning of machine learning algorithms to achieve improved predictive performance.

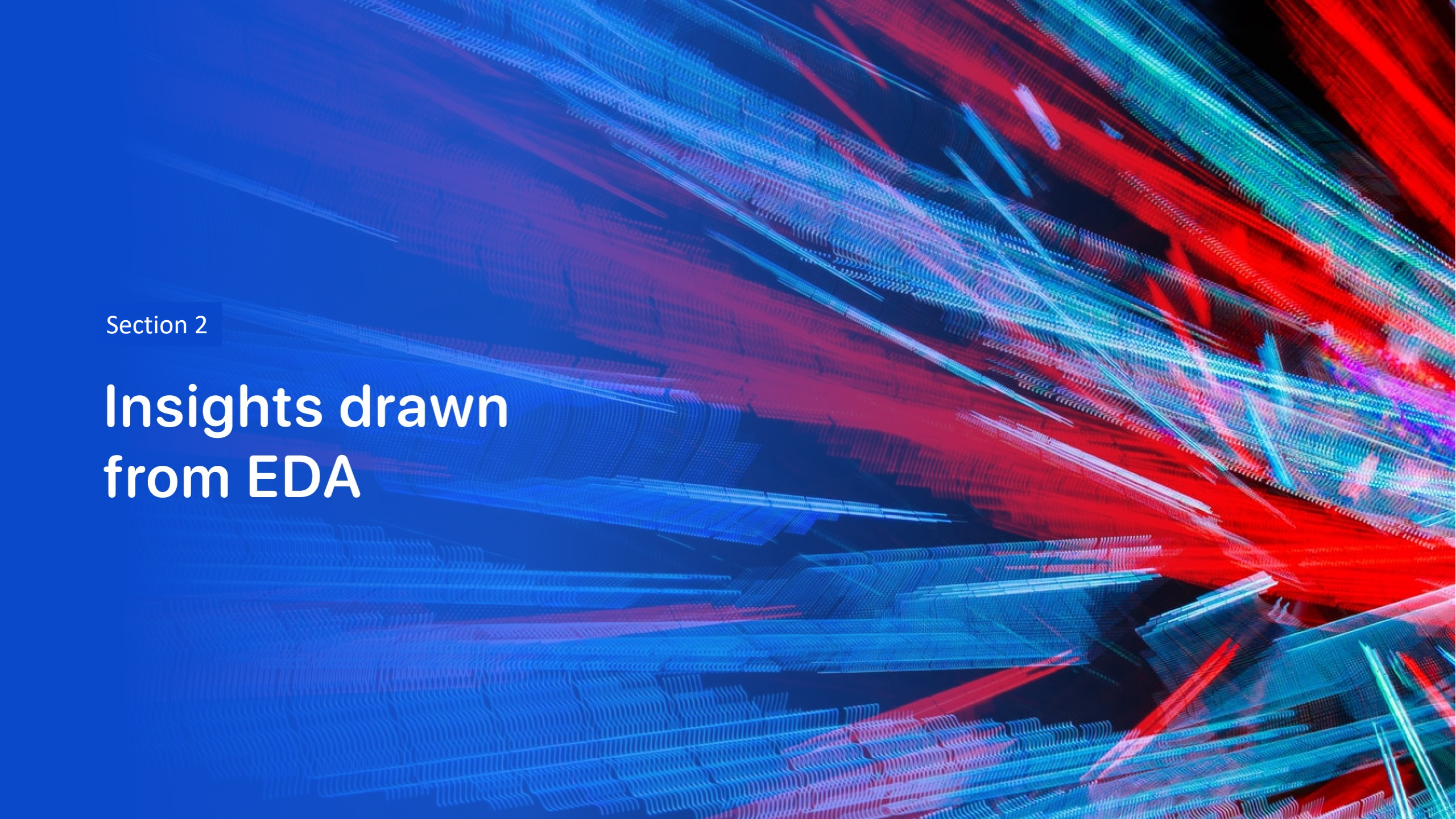
Finding the Best Model

The model exhibiting the highest accuracy score will be deemed the most proficient and effective among the models under consideration.

Notebook URL: https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/7-SpaceX_Machine_Learning_Prediction.jupyterlite.ipynb

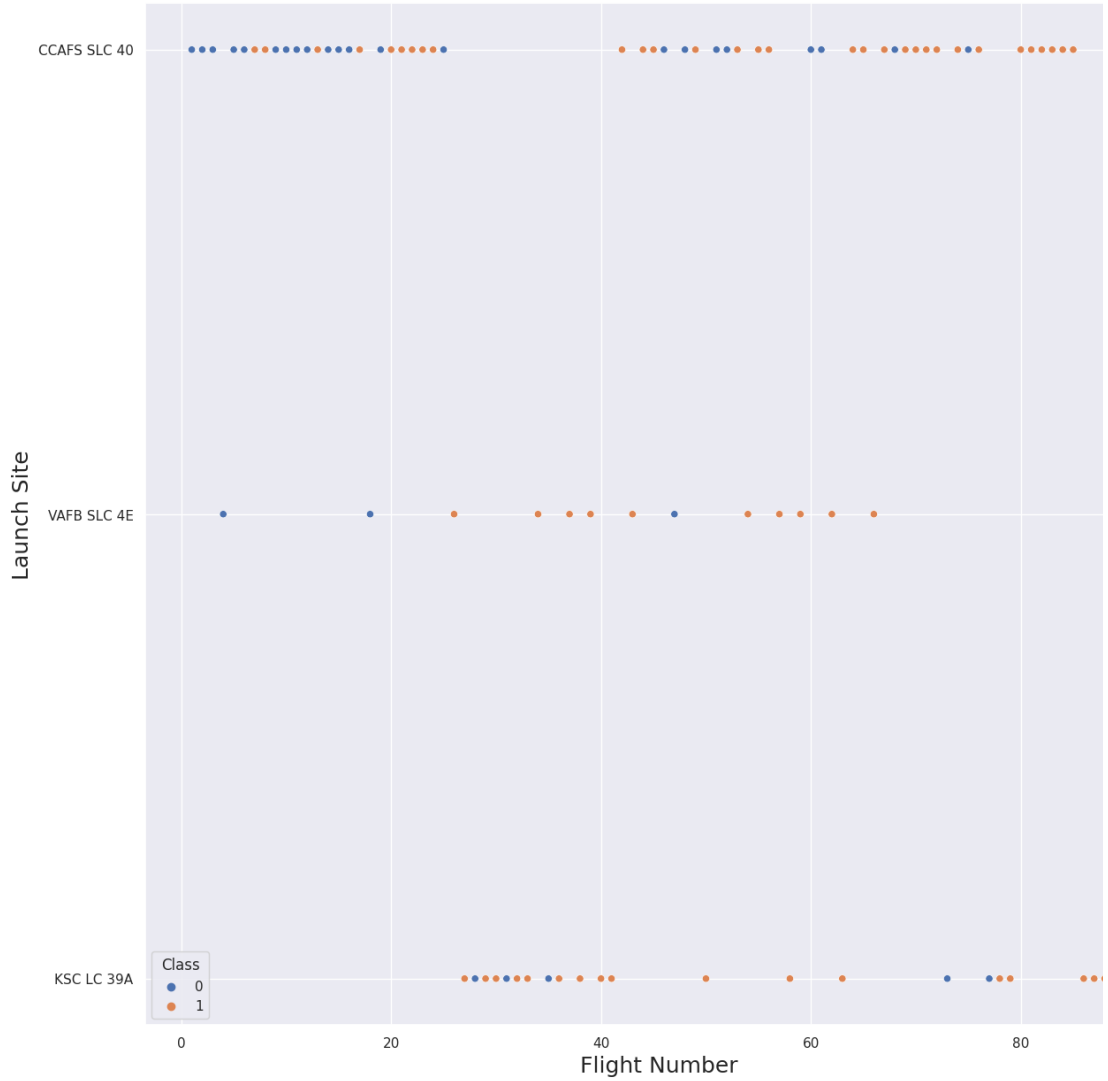
Results

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


Section 2
Insights drawn from EDA

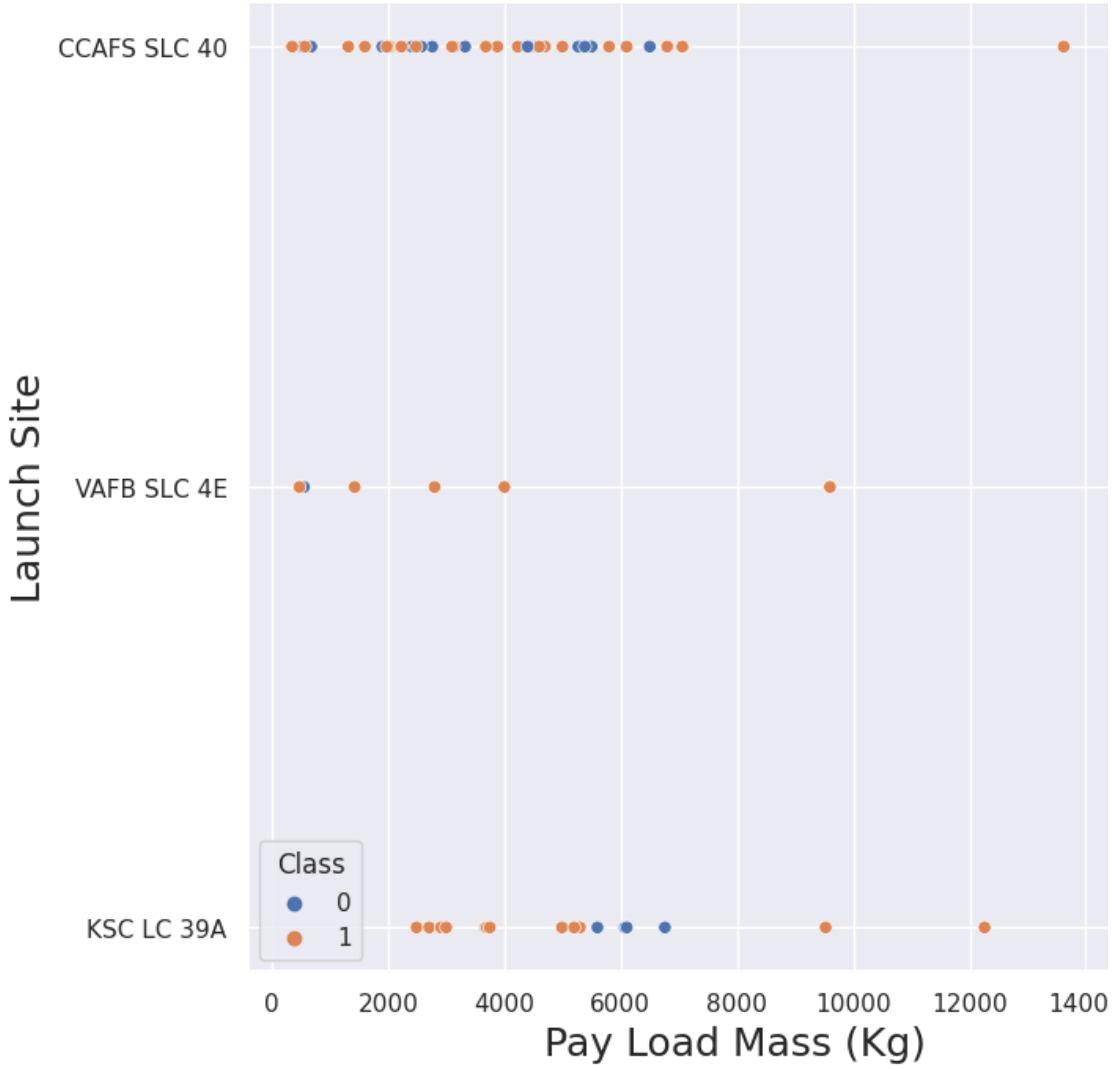
Flight Number vs. Launch Site

Based on the storyline, it was evident that a higher number of flights conducted at a launch site correlated with an increased likelihood of success at that site.

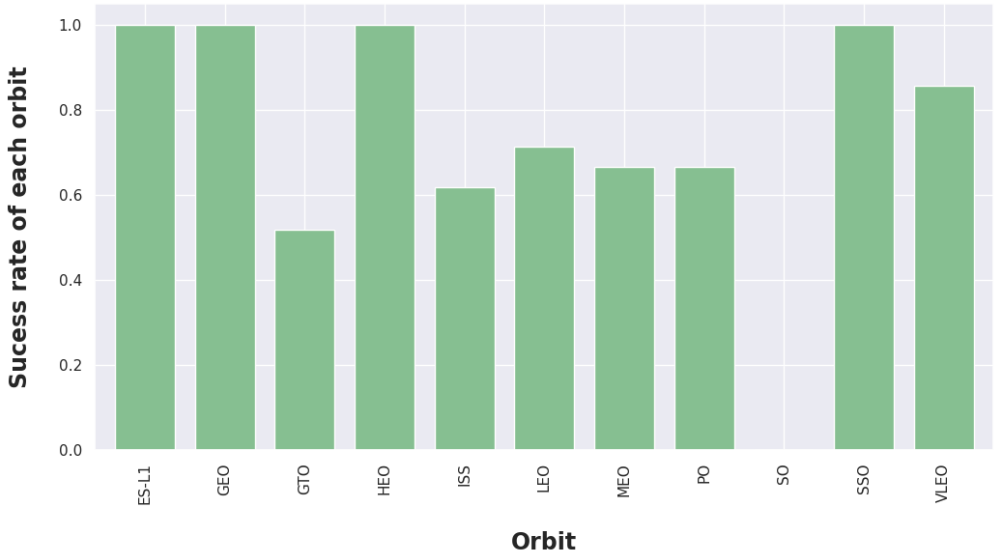


Payload vs. Launch Site

The plot illustrates that once the payload mass exceeds 7000 kilograms, the probability of a successful launch significantly rises. Nevertheless, there is no distinct pattern indicating that the success rate is reliant on the payload mass for a specific launch site.



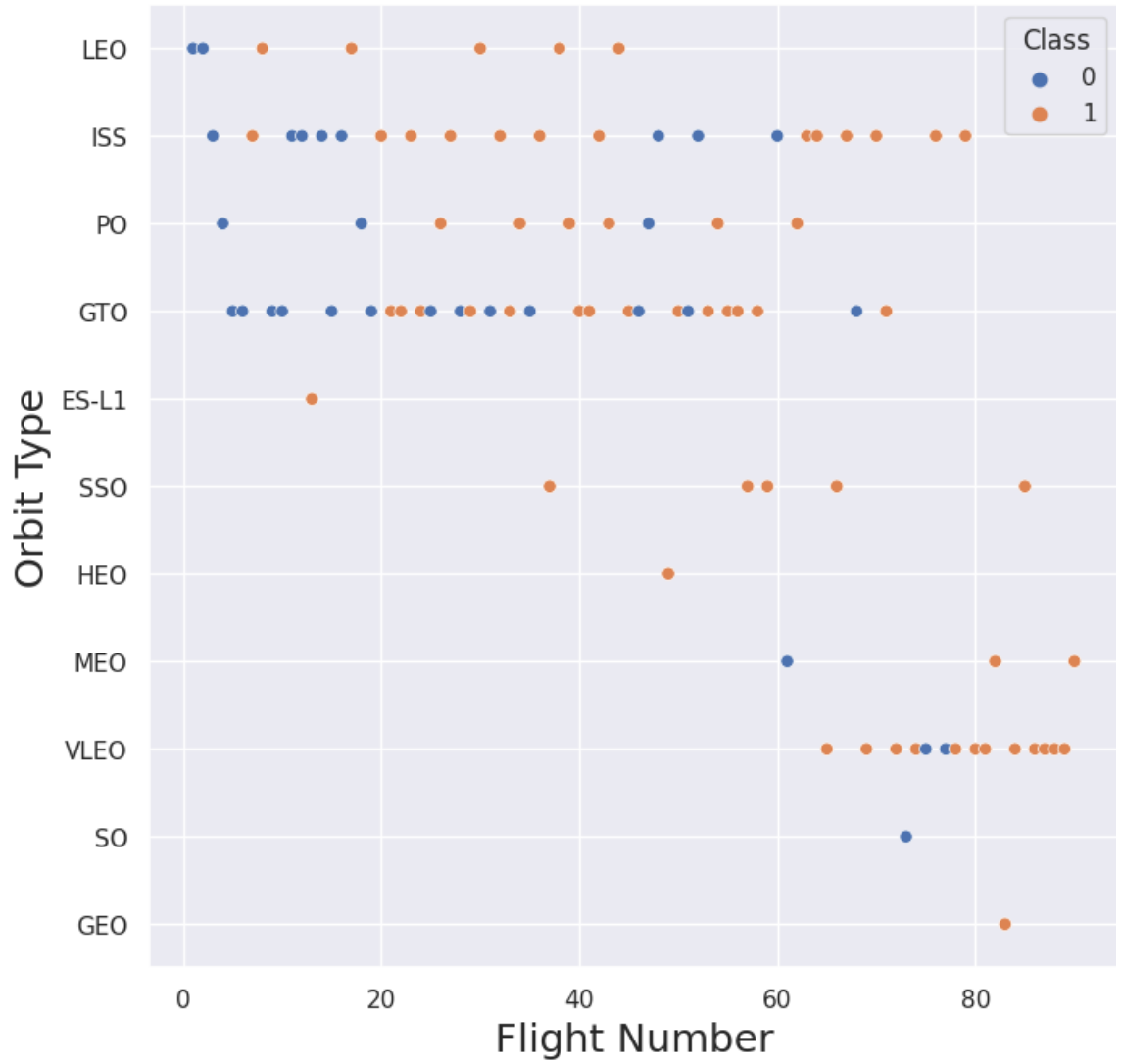
Success Rate vs. Orbit Type



This illustration portrayed how different orbits could potentially affect landing outcomes. Notably, certain orbits like SSO, HEO, GEO, and ES-L1 exhibited a 100% success rate, whereas the SO orbit yielded a 0% success rate. However, upon closer examination, it becomes evident that some of these orbits, such as GEO, SO, HEO, and ES-L1, had only one occurrence in the dataset. This indicates that more data is required to discern any patterns or trends before drawing any definitive conclusions.

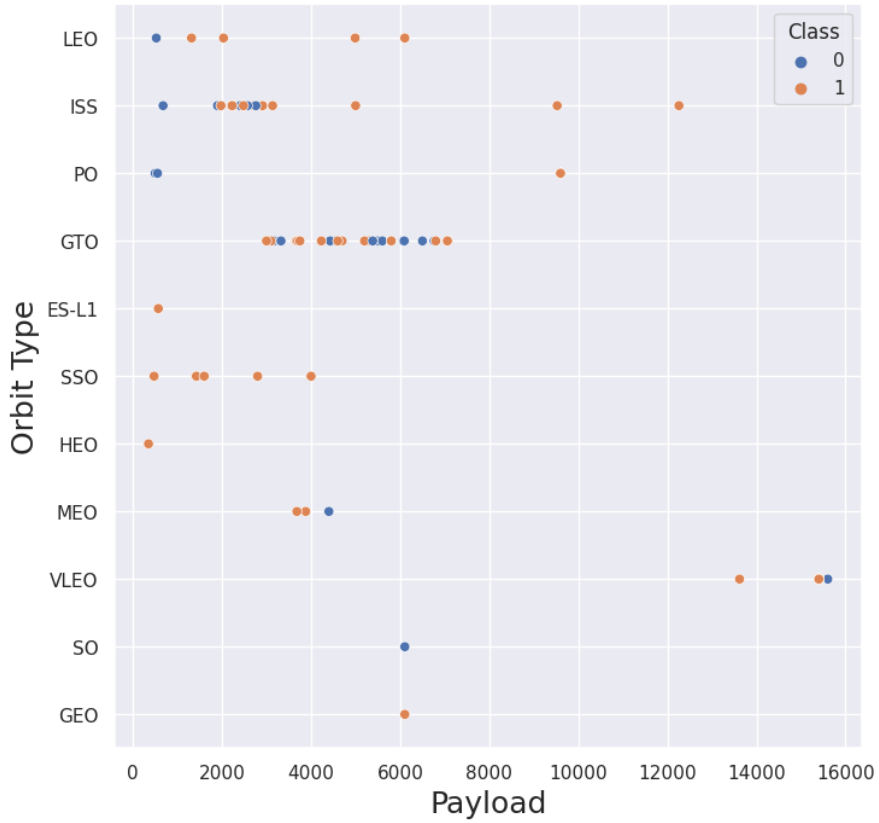
Flight Number vs. Orbit Type

The plot illustrates a general trend where an increase in the number of flights for each orbit is associated with a higher success rate, particularly in the case of the LEO orbit. However, the GTO orbit stands out as it does not exhibit any correlation between these two factors.

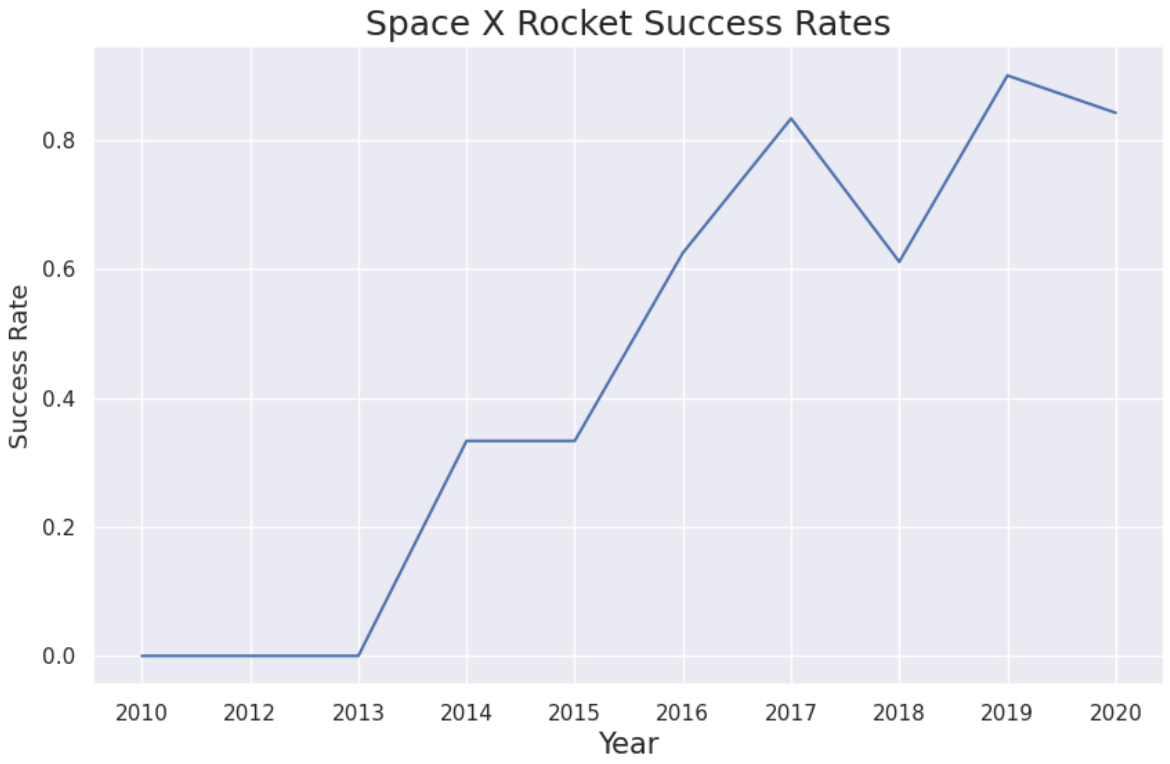


Payload vs. Orbit Type

A greater payload weight has a beneficial effect on LEO, ISS, and P0 orbits, but it has an adverse effect on MEO and VLEO orbits. In the case of the GTO orbit, there appears to be no discernible relationship between the two attributes. Meanwhile, once more, SO, GEO, and HEO orbits require additional data to identify any patterns or trends.



Launch Success Yearly Trend



By examining the plot, it's evident that the success rate has been steadily rising from 2013 to 2020.

All Launch Site Names

The key word "DISTINCT" was employed to display only the unique launch sites from the SpaceX data.

```
%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

Launch Site Names Begin with 'CCA'

The query depicted in the figure was utilized to showcase 5 records where the launch sites commence with "CCA."

```
%sql select LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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

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

Total Payload Mass

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

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

SUM (PAYLOAD_MASS__KG_)
45596

We determined the combined payload transported by NASA's boosters to be 45,596 units by utilizing the query above.

Average Payload Mass by F9 v1.1

We computed the mean payload weight transported by booster version F9 v1.1 to be 2,928.4.

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

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

AVG (PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

The observation was made that the initial achievement of a successful landing outcome on a ground pad occurred on the 22nd of December 2015.

```
%sql SELECT MIN (DATE) AS "First Successful Landing" FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

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

First Successful Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

The **WHERE** clause was employed to selectively isolate boosters that had achieved successful landings on a drone ship, and the **AND** condition was subsequently applied to ascertain successful landings with payload masses falling within the range of greater than 4000 but less than 6000.

```
%sql select BOOSTER_VERSION from SPACEXTBL 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

We employed the wildcard symbol '%' to conduct filtering based on WHERE MissionOutcome, encompassing both successful and unsuccessful outcomes.

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "succesful mission"  FROM SPACEXTBL  WHERE Mission_Outcome LIKE 'Success%'
```

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

succesful mission
100

```
%sql SELECT COUNT (MISSION_OUTCOME) AS "failure mission " FROM SPACEXTBL WHERE Mission_Outcome LIKE 'Fail%'
```

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

failure mission
1

Boosters Carried Maximum Payload

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Unique Boosters with the Heaviest Payload" FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

Unique Boosters with the Heaviest Payload
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

We identified the booster that transported the highest payload through the utilization of a subquery within the **WHERE** clause, in conjunction with the **MAX()** function.

2015 Launch Records

We applied a combination of filtering conditions, including the WHERE clause, LIKE, AND, and BETWEEN conditions, to extract information related to unsuccessful landing outcomes on drone ships, along with their respective booster versions and launch site names for the year 2015.

```
%sql SELECT DATE, Booster_Version, LAUNCH_SITE, Landing_Outcome  FROM SPACEXTBL  WHERE Landing_Outcome = 'Failure (drone ship)'  AND substr(Date,1,4)='2015';
```

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

Date	Booster_Version	Launch_Site	Landing_Outcome
2015-10-01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

We chose both the Landing outcomes and the **COUNT** of these landing outcomes from the dataset. We then employed the **WHERE** clause to narrow down the landing outcomes within the date range from June 4, 2010, to March 20, 2010. To organize the landing outcomes, we utilized the **GROUP BY** clause, and for arranging them in descending order, we implemented the **ORDER BY** clause.

```
%sql SELECT LANDING_OUTCOME as "Landing Result", COUNT(LANDING_OUTCOME) AS "Number of Times"  FROM SPACEXTBL \
WHERE DATE >= '2010-06-04' AND DATE <= '2017-03-20' GROUP BY LANDING_OUTCOME ORDER BY COUNT(LANDING_OUTCOME) DESC;
```

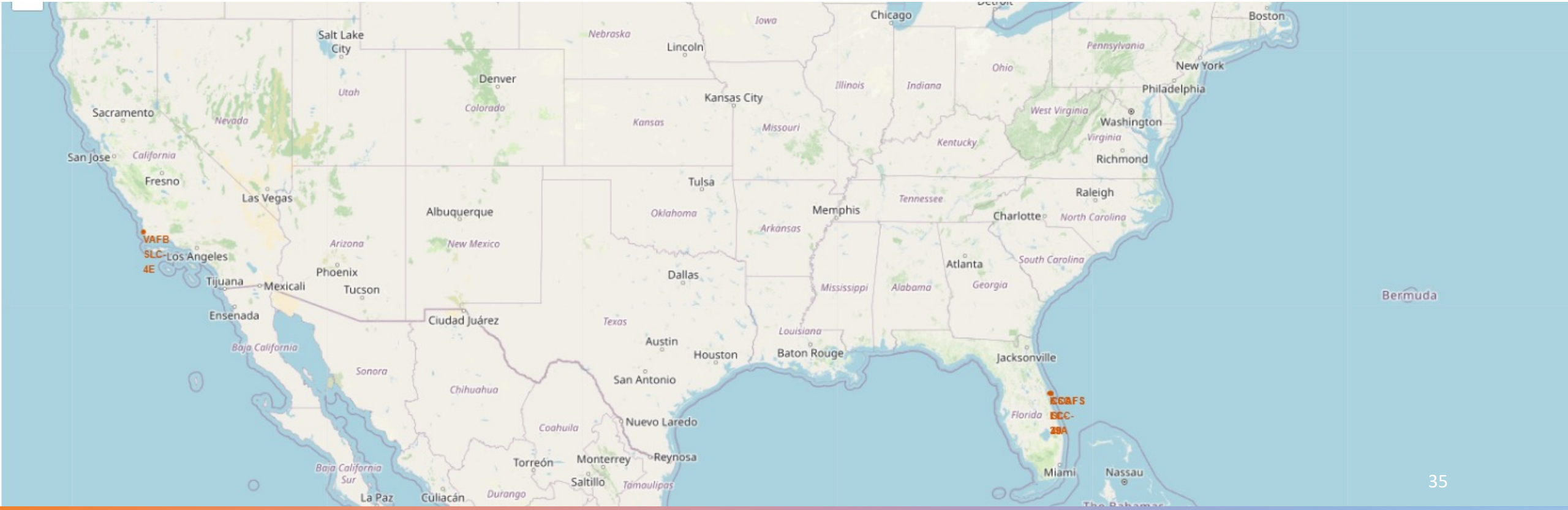
```
 * sqlite:///my_data1.db
Done.
```

Landing Result	Number of Times
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

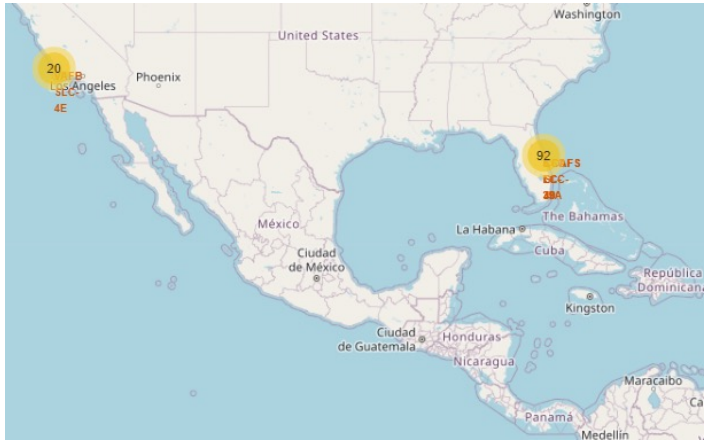
Location of the Launch Sites

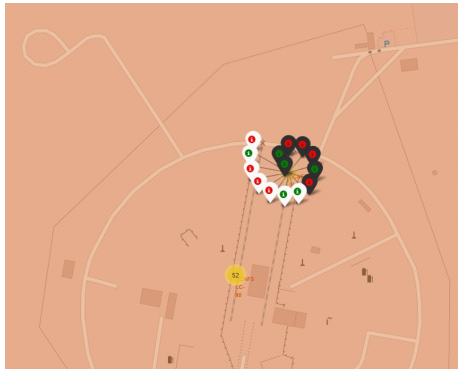
It can be observed that all SpaceX launch sites are situated within the boundaries of the United States.

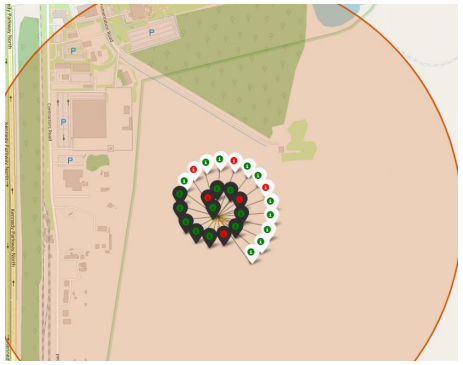


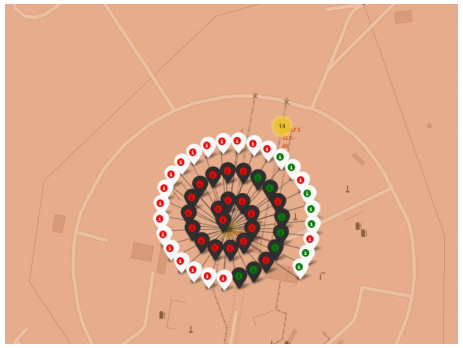
Markers Displaying Launch Sites with Color-Coded Labels for the Florida Site

Florida Launch Sites can be seen on right side with colored markes



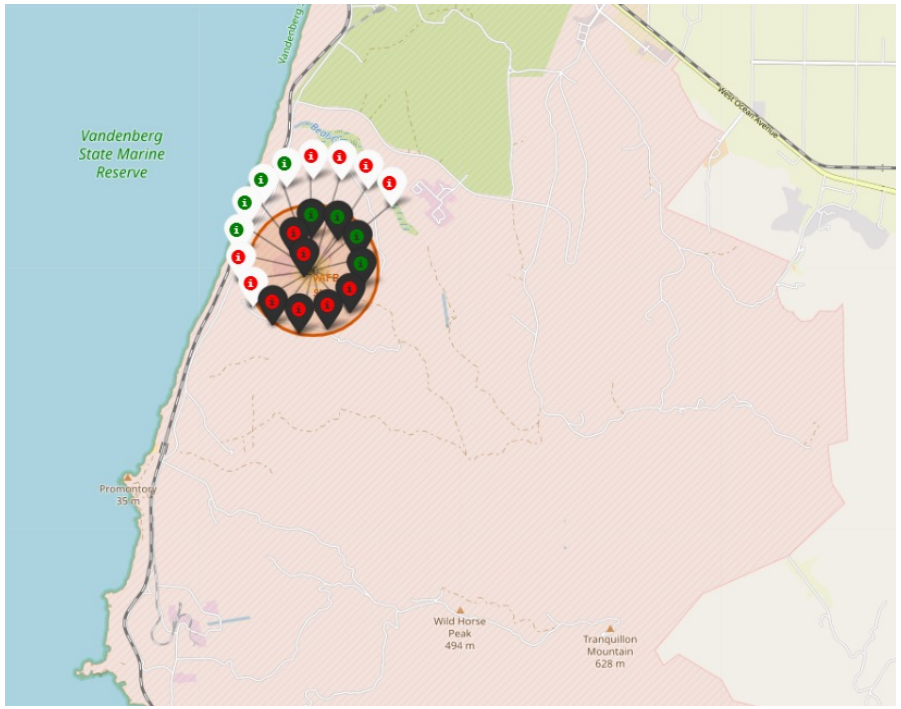


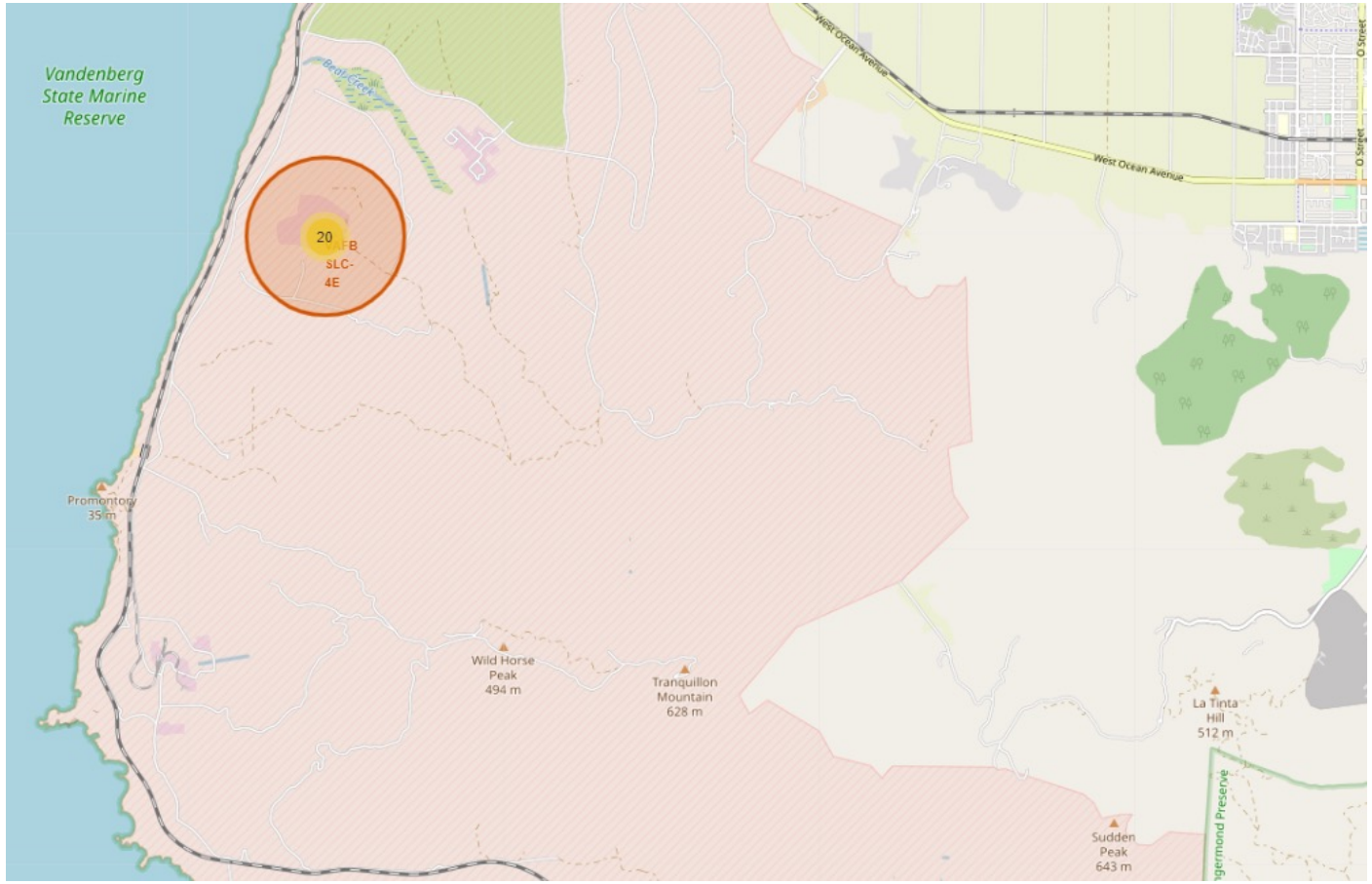




Markers Displaying Launch Sites with Color-Coded Labels for the California Site

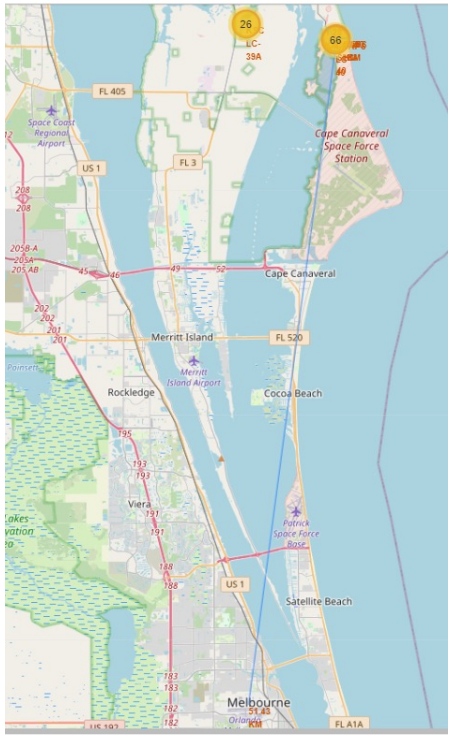
California Launch Sites can be seen on right side with colored markes

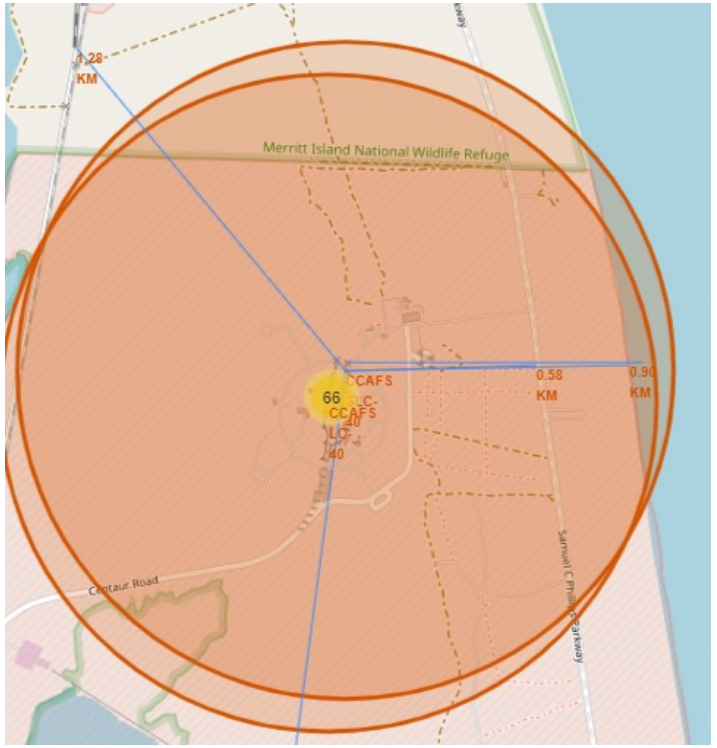




Launch Site Distance to Landmarks

- Launch sites are strategically located near the equator to leverage Earth's approximate 30km/sec eastward rotation, which aids in efficient orbit insertion by reducing fuel needs.
- They are ideally situated close to coastlines to ensure that launches primarily take place over the ocean. This serves dual safety purposes: firstly, it provides the crew with a water landing option if a launch abort is necessary, and secondly, it reduces risks to people and infrastructure from potential falling debris.
- Proximity to major highways is important for launch sites as it facilitates the easy movement of essential personnel and equipment.
- Being near railways is also advantageous, as it provides an avenue for transporting bulky cargo.
- To minimize risks to densely populated areas, these sites are intentionally distanced from major cities.





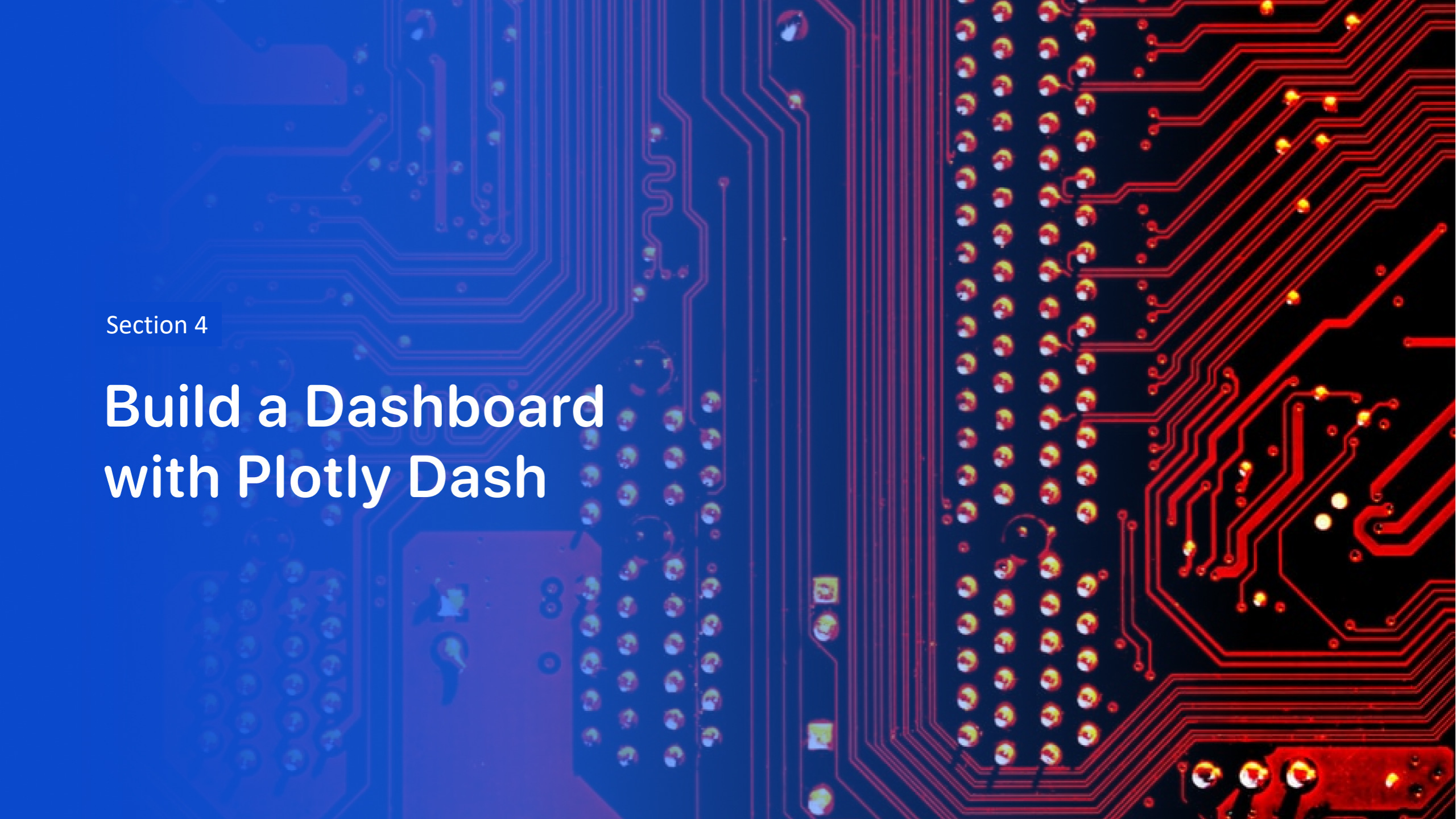

```
# Create a marker with distance to a closest city, railway, highway, etc.
# Draw a line between the marker to the launch site
closest_highway = 28.56335, -80.57085
closest_railroad = 28.57206, -80.58525
closest_city = 28.10473, -80.64531
```

```
distance_highway = calculate_distance(launch_site_lat, launch_site_lon, closest_highway[0], closest_highway[1])
print('distance_highway =',distance_highway, ' km')
distance_railroad = calculate_distance(launch_site_lat, launch_site_lon, closest_railroad[0], closest_railroad[1])
print('distance_railroad =',distance_railroad, ' km')
distance_city = calculate_distance(launch_site_lat, launch_site_lon, closest_city[0], closest_city[1])
print('distance_city =',distance_city, ' km')
```

```
distance_highway = 0.5834695366934144  km
distance_railroad = 1.2845344718142522  km
distance_city = 51.434169995172326  km
```

After you plot distance lines to the proximities, you can answer the following questions easily:

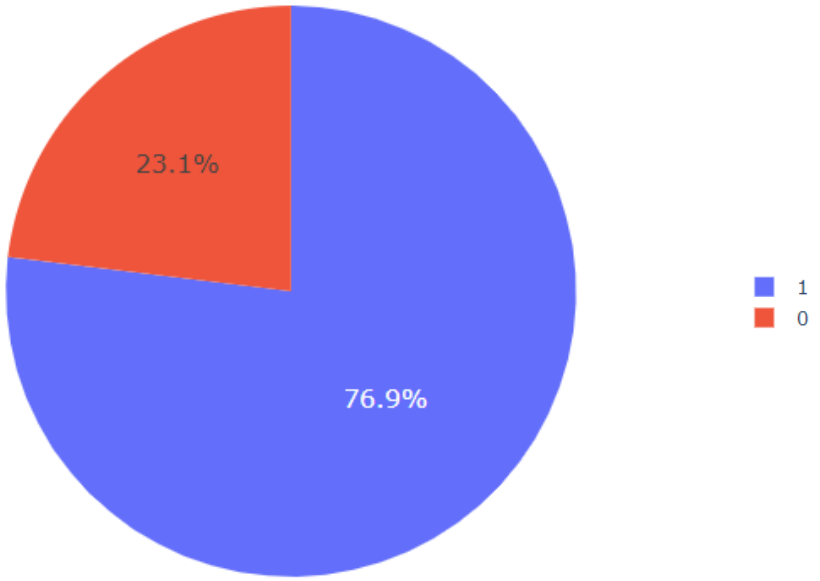
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes


Section 4
Build a Dashboard with Plotly Dash

The Success Percentage of Each Site

- KSC LC-39A is the site with the highest successful launch rate
- CCAFS SLC-40 is the site with the lowest successful launch rate

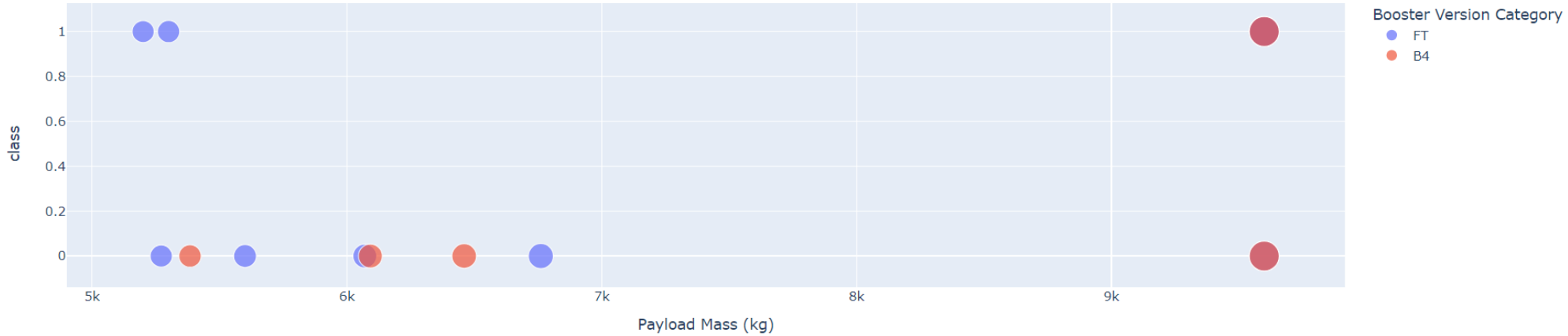
Launch Analysis of KSC LC-39A Launch Site



The graph shows that approximately 3 out of every 4 launches are successful.

Scatter Plot Comparing Payload to Launch Outcome

We can observe that the success rate is higher for payloads in the low weight category (0-5000 KG) compared to those in the heavy weight category (5000-10000 KG).



Section 5

Predictive Analysis (Classification)

Classification Accuracy

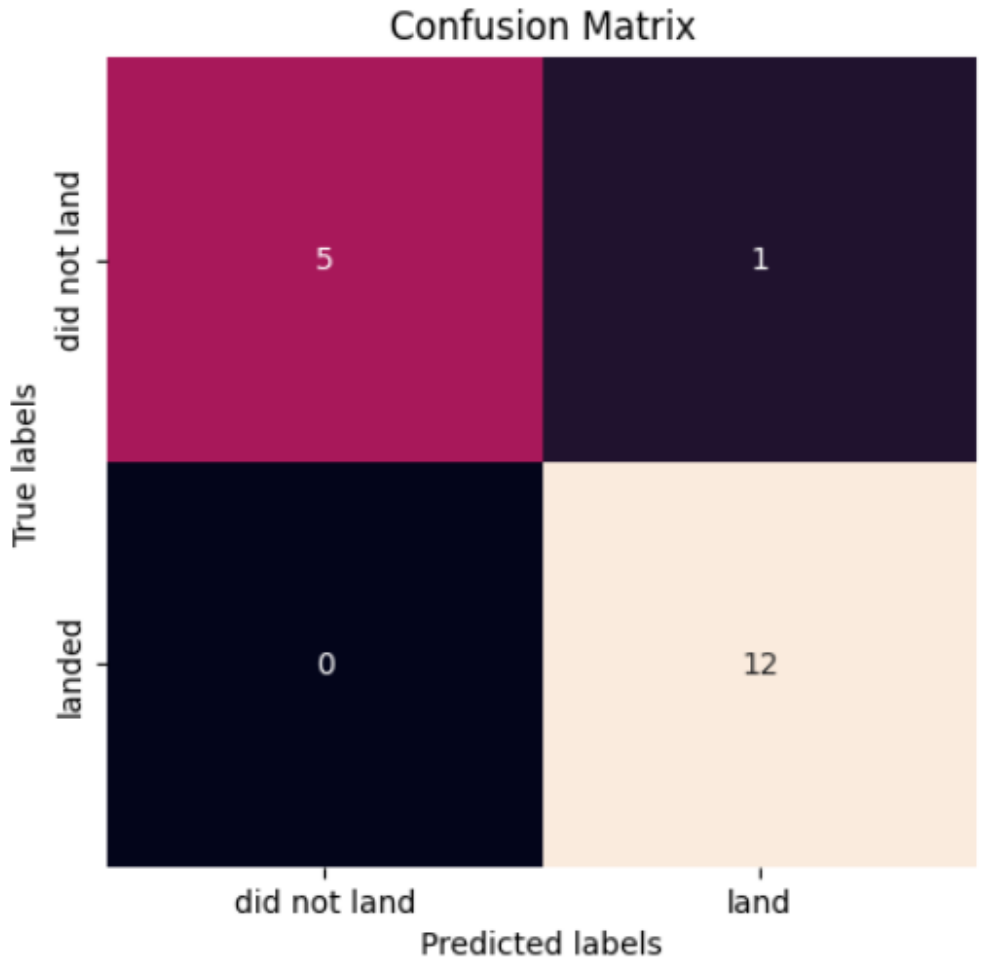
The model that exhibits the highest classification accuracy in the figure is the decision tree classifier with optimal parameters

```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

predictors = {
    'KNN': knn_cv,
    'SVM': svm_cv,
    'LogisticRegression': logreg_cv,
    'DecisionTree': tree_cv
}

best_predictor_name = None
best_params = None
best_score = 0
best_model=None
# Iterate over each GridSearchCV object
for name, predictor in predictors.items():
    # Iterate over each combination of parameters tested by the GridSearchCV
    for params in predictor.cv_results_['params']:
        # Set the parameters to the base model (estimator)
        predictor.estimator.set_params(**params)
        # Train the model with the current combination of parameters on the entire training data
        predictor.estimator.fit(X_train, Y_train)
        # Evaluate the model on the test data
        current_score = predictor.estimator.score(X_test, Y_test)
        if current_score > best_score:
            best_score = current_score
            best_predictor_name = name
            best_params = params
            best_model=predictor.estimator
            yhat=

print("Best Model: ", best_predictor_name)
print("Best Parameters on Test Data: ", best_params)
print("Score on Test Data: ", best_score)


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

```
Best Model:  DecisionTree
Best Parameters on Test Data:  {'criterion': 'gini', 'max_depth': 2, 'max_features': 'auto', 'min_samp
es_split': 5, 'splitter': 'best'}
Score on Test Data:  0.9444444444444444
```

Confusion Matrix

The decision tree classifier's confusion matrix indicates its capability to differentiate among the various classes. The model only shows weakness in terms of the false positive rate with one observation.



Conclusions

Our final assessment is that:

- Payloads weighing 5000kg or less outperformed their heavier counterparts.
- From 2013 onwards, there has been a consistent rise in SpaceX launch successes, pointing towards even better outcomes in upcoming years until 2020.
- Among all launch sites, KSC LC-39A stands out with a success rate of 76.9%.
- With a 100% success rate and multiple occurrences, the SSO orbit is the most reliable.
- For this dataset, the Tree Classifier Algorithm emerges as the top Machine Learning choice.

Appendix

Notebooks and Python Files in This Project

- https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/1-spacex-data-collection-api.ipynb
- https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/2-jupyter-labs-webscraping.ipynb
- https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/3-labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb
- https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/4-jupyter-labs-eda-sql-coursera_sqllite.ipynb
- https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/5-jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb
- https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/6-lab_jupyter_launch_site_location.jupyterlite.ipynb
- https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/7-SpaceX_Machine_Learning_Prediction.jupyterlite.ipynb
- https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/spacex_dash_app.py

Datasets in This Project

- https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/dataset_part_1.csv
- https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/dataset_part_2.csv
- https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/dataset_part_3.csv
- https://github.com/barisozcann/Applied-Data-Science-Capstone-SpaceX/blob/main/spacex_web_scraped.csv

Acknowledgments

- Thank you to Dr. Pooja, Romeo Kienzler, Joseph Santarcangelo, Polong Lin, Alex Aklson, Rav Ahuja, Saishruthi Swaminathan, SAEED AGHABOZORGI, Hima Vasudevan, Azim Hirjani, Aije Egwaikhide, Yan Luo, and Svetlana Levitan at IBM for creating the course and materials

Thank you!