IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Veronika Grundmane
02/08/23

Outline

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

Executive Summary

- The procedures utilized to examine the data are as outlined below: First, data was gathered from the SpaceX API, and information was extracted from the SpaceX Wikipedia webpage using web scraping techniques. Subsequently, comprehensive data preparation was carried out, encompassing tasks such as Exploratory Data Analysis (EDA), SQL manipulation, creation of data visualizations through dashboards, and the application of additional analytical methods. Once the data was appropriately prepared, a range of Machine Learning models were utilized, with a focus on optimizing hyperparameters. These models were employed to categorize successful landings, and the accuracy scores of each model were assessed.
- Summary: Four distinct machine learning classification models were applied to the processed data, specifically the K Nearest Neighbors, Decision Tree Classifier, Support Vector Classifier, and Logistic Regression models. These models collectively achieved an accuracy rate of x%. However, in order to enhance the precision of target classification and instill greater confidence in the results, further analysis and additional data are imperative.

Introduction

- SpaceX stands out as a prominent player in the realm of rocket launches, offering its services at a significantly lower cost of $62 million, in stark contrast to other providers who demand $162 million for the same service. One of SpaceX's key cost-saving strategies revolves around the reuse of the initial stage of their rockets. This approach not only leads to substantial cost savings but also contributes to the company's ability to offer their services at a more affordable price point.

- Space Y aims to rival Space X in the space launch industry and is actively seeking to estimate various factors associated with their launches. These factors include the total cost of launch operations as well as the metrics related to successful landings. By gaining a comprehensive understanding of these variables, Space Y can better position itself to compete effectively in the market and make informed decisions regarding its launch strategies and pricing models.

- The objective of this project is to analyze historical data regarding previous landing events and utilize this information to predict the likelihood of a successful landing for the first stage of a rocket. Through this analysis, Space Y aims to assess its potential to compete with SpaceX. By examining the viability of achieving consistent successful landings, Space Y can make informed decisions about its capability to establish itself as a competitor in the space launch industry alongside SpaceX.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data is collected from the combination of SpaceX public API and web scraping of the SpaceX Wikipedia webpage.
- Perform data wrangling
  - Conducted a value count analysis to quantify occurrences.
  - Eliminated NaN (null) values from the dataset.
  - Employed one-hot encoding to transform categorical variables.
  - Categorized instances of successful and unsuccessful landings.
  - Examined the characteristics of various features.
  - Organized and refined the data in preparation for subsequent modeling tasks.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The insights gained from the aforementioned steps were incorporated into the process. The data was normalized to ensure consistency and comparability. Subsequently, the dataset was divided into training and testing subsets. With the assistance of hyperparameter tuning, classification models were developed. The model with the highest accuracy scores was identified as the best-performing one. This comprehensive approach, integrating data analysis, normalization, splitting, and advanced model tuning, allows for the creation of effective classification models tailored to the specific task of predicting successful landings.

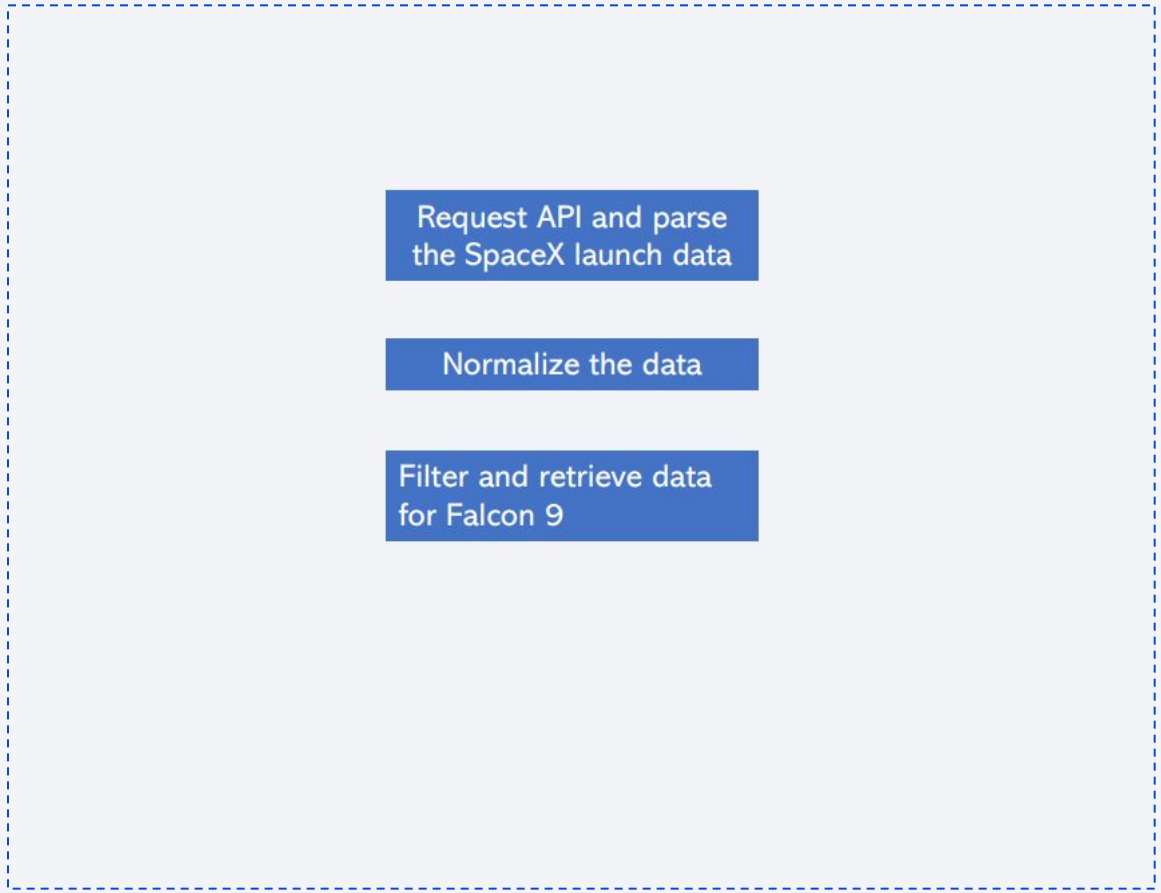
Data Collection

- The data collection procedure primarily encompassed two key steps:
- Initial data acquisition involved the retrieval of datasets from the Space X API. The specific Space X API endpoint utilized for this purpose was: https://api.spacexdata.com/v4/rockets/
- Subsequently, additional data was procured utilizing web scraping methodologies. This involved extracting data from a Wikipedia webpage that is accessible at: https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches
- These two steps were integral to gathering the requisite data for the analysis. The API usage provided structured information directly, while web scraping facilitated the extraction of specific data from the Wikipedia page.

Data Collection – SpaceX API

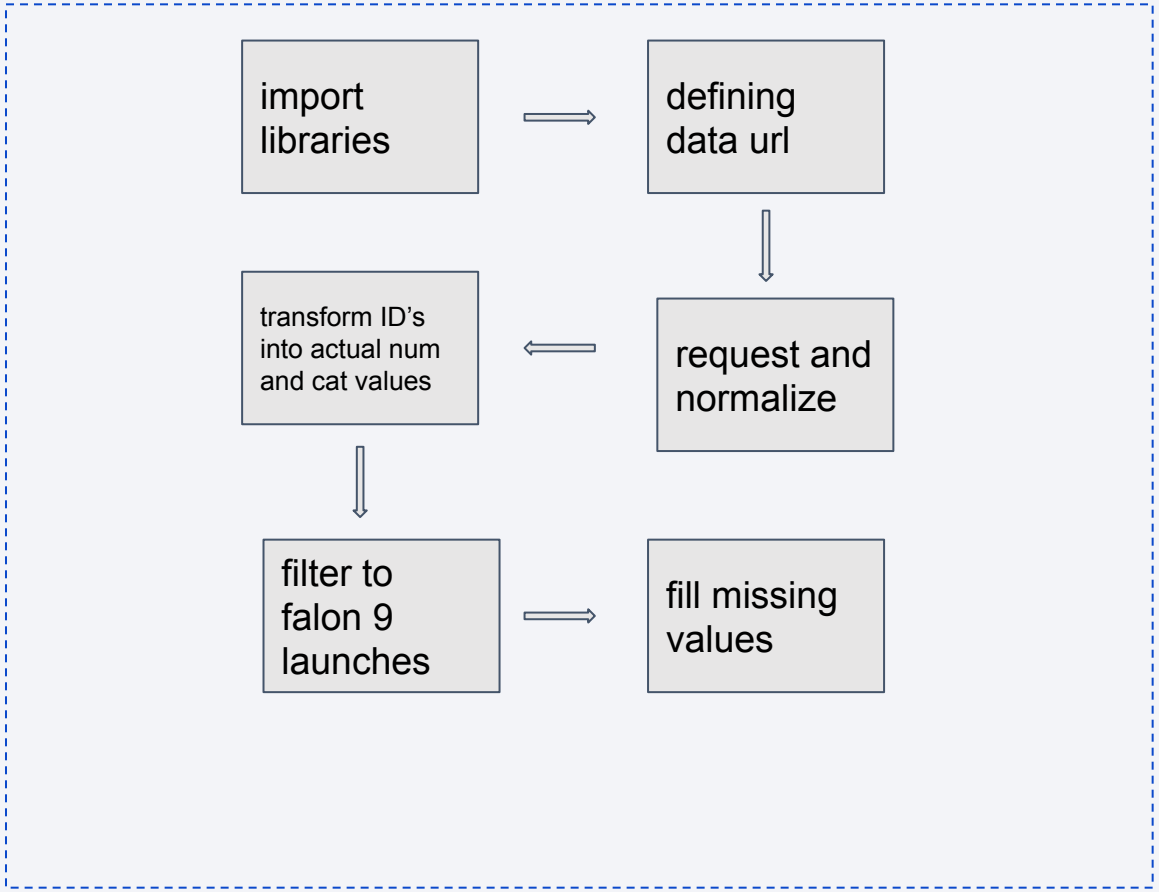
- SpaceX provides a publicly accessible Application Programming Interface (API) that offers access to various data related to Falcon 9 launches. This API has been utilized in this context to retrieve detailed information about Falcon 9 rocket launches, including launch data, rocket specifications, and landing outcomes. This API serves as a valuable resource for obtaining accurate and up-to-date information about SpaceX's Falcon 9 missions and their associated details.

- https://github.com/Grundmane/FinalProjectMachineLearning/blob/main/jupyter_labs_spacex_data_collection_api.ipynb



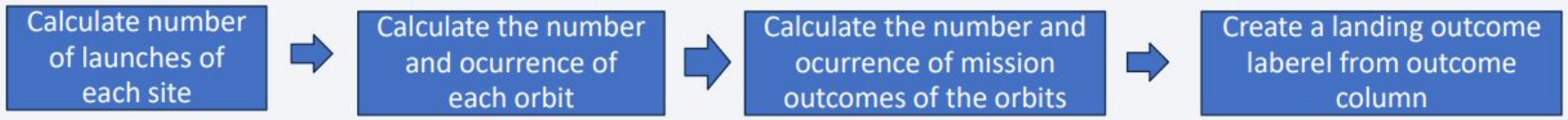
Data Collection - Scraping

- https://github.com/Grundmane/FinalProjectMachineLearning/blob/main/jupyter_labs_webscraping_(1).ipynb



Data Wrangling

- https://github.com/Grundmane/FinalProjectMachineLearning/blob/main/labs_jupyter_spacex_Data_wrangling.ipynb



EDA with Data Visualization

The visualizations that were created in the project are as follows:

- A chart depicting the relationship between Payload Mass and Flight Number.
- A chart illustrating the distribution of Flight Numbers across different Launch Sites.
- A visualization showcasing the correlation between Payload Mass and Launch Site.
- A graph showcasing the Success Rate for each orbit type.
- A chart detailing the relationship between Flight Number and Orbit Type, sorted by class.
- A visualization displaying the connection between Payload and Orbit Type, sorted by class.
- Lastly, a graphical representation of the Launch Success trend over the years.

These charts provide valuable insights into various aspects of the data and aid in understanding patterns, trends, and relationships within the dataset.

https://github.com/Grundmane/FinalProjectMachineLearning/blob/main/jupyter_labs_eda_dataviz_(1).ipynb

EDA with SQL

- • With SQL Queries, we performed:
- 1. Display unique launch sites
- 2. Display 5 records where launch sites begin with the string 'CCA’ 3. Display the total payload mass carried by boosters launched by NASA (CRS)
- 4. Display average payload mass carried by booster version F9 v1.1
- 5. List the date when the first succesful landing outcome in ground pad was achieved 106. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- 7. List the total number of successful and failure mission outcomes
- 8. List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- 9. List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
- 10. Rank the count of landing outcomes (such asFailure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

https://github.com/Grundmane/FinalProjectMachineLearning/blob/main/jupyter_labs_eda_sql_coursera_sqllite.ipynb

Build an Interactive Map with Folium

With Folium, a Python library for creating interactive maps, you performed the following tasks:

- Marked All Launch Sites on the Map: You plotted markers on the map to indicate the locations of all the launch sites. This provides a geographical overview of where the launch activities are centered.
- Marked Success/Failed Launches for Each Site: You used markers with different colors or icons to distinguish between successful and failed launches at each launch site. This visual representation helps in quickly identifying the outcomes associated with each launch site.
- Clustered the Launch Sites: You employed clustering techniques to group nearby launch sites together on the map. This approach enhances map readability and prevents overcrowding, especially in cases where there are many launch sites in close proximity.
- Calculated Distances Between Launch Sites: You calculated distances between launch sites and their neighboring points. This information could be used for analyzing launch site proximity and distribution.
- Drew a PolyLine: You drew a polyline (a connected line segment) on the map, indicating the path between a launch site and a selected coastline point. This could be useful for visualizing the trajectory or flight path of a rocket launch.

The combination of these actions in Folium allowed you to create an interactive map that provides insights into the success rates of launches at different sites, their spatial distribution, and the geographical context of their launch trajectories. This type of visualization aids in understanding patterns and relationships within the launch data in a more intuitive manner.

https://github.com/Grundmane/FinalProjectMachineLearning/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

The described features create an interactive and insightful visualization platform for analyzing space launch data. Here's a summary of what each feature accomplishes:

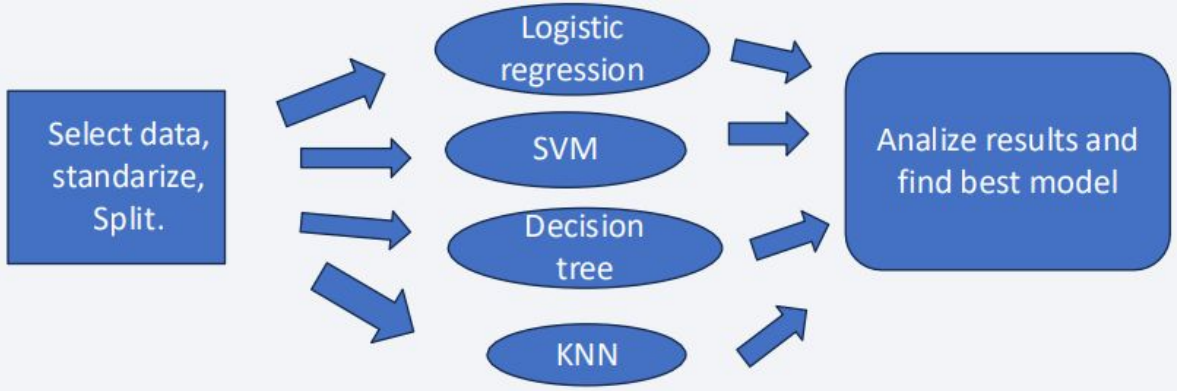
- Dropdown List with Launch Sites: This feature offers users the option to select either all launch sites or a specific launch site. It allows users to narrow down their focus and analyze data based on their chosen criteria.
- Pie Chart Showing Successful Launches: A pie chart visualizes the distribution of successful launches versus unsuccessful ones. It provides a clear representation of the proportion of successful and unsuccessful outcomes.
- Slider of Payload Mass Range: This slider enables users to dynamically adjust the range of payload masses they want to analyze. It allows for filtering data based on payload mass, offering a customizable view.
- Scatter Chart Showing Payload Mass vs. Success Rate by BoosterVersion: The scatter chart plots payload mass against success rate for different booster versions. It enables users to identify any trends or correlations between payload mass and success rate.
- Allow User to See Successful and Unsuccessful Launches as a Percent of the Total: This feature provides a toggle or checkbox option that allows users to switch between viewing successful and unsuccessful launches as a percentage of the total launches. This facilitates a comparative analysis of success rates.
- Allow User to See the Correlation Between Payload and Launch Success: This feature may involve displaying a correlation coefficient value or a visual representation (such as a heatmap) of the correlation between payload mass and launch success. It helps users understand the relationship between these two variables.

Combining these features creates an interactive dashboard that empowers users to explore and analyze space launch data from various angles. Users can customize their views, investigate patterns, and draw insights from the visualizations, contributing to a deeper understanding of the data and its underlying trends.

https://github.com/Grundmane/FinalProjectMachineLearning/blob/main/7.%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash%20-%20spacex_dash_app.py

Predictive Analysis (Classification)

- Data Preparation:
  - Select the relevant data for the analysis.
  - Standardize the features if needed to bring them to a common scale.
  - Split the data into training and testing sets.
- Logistic Regression:
  - Create a Logistic Regression model.
  - Implement GridSearchCV to find the best hyperparameters for the model.
  - Calculate the accuracy of the Logistic Regression model.
  - Analyze the confusion matrix to understand the model's performance.
- Support Vector Machine (SVM):
  - Create a Support Vector Machine (SVM) classifier object.
  - Implement GridSearchCV to determine the best hyperparameters.
  - Fit the model with the training data and find the best parameters.
  - Calculate the accuracy of the SVM model.
- Decision Tree Classifier:
  - Develop a Decision Tree classifier.
  - Apply GridSearchCV to identify optimal hyperparameters.
  - Fit the classifier with the training data and find the best parameters.
  - Calculate the accuracy of the Decision Tree model.
- K Nearest Neighbors (KNN):
  - Instantiate a K Nearest Neighbors classifier.
  - Utilize GridSearchCV to discover the most suitable hyperparameters.
  - Fit the classifier using the training data and determine the best parameters.
  - Calculate the accuracy of the KNN model.
- Comparative Analysis:
  - Analyze and compare the results obtained from the Logistic Regression, SVM, Decision Tree, and KNN models.
  - Evaluate the accuracy and other relevant metrics for each model.
  - Select the model that performs the best based on the evaluation metrics.



- https://github.com/Grundmane/FinalProjectMachineLearning/blob/main/SpaceX_Machine_Learning_Prediction_Part_5_jupyterlite.ipynb

Results

- Launch Success Improvement Over Time: The success rate of launches has demonstrated an improvement over the years. This indicates advancements in technology and operational practices.
- KSCLC-39A Success Rate: Among the various landing sites, KSCLC-39A stands out with the highest success rate, possibly due to specific conditions that favor successful landings.
- Orbit Success Rates: Orbits like ES-LI, GEO, HEO, and SSO exhibit a remarkable 100% success rate. This could be due to the suitability of these orbits for specific mission types.

Visual Analytics Insights:

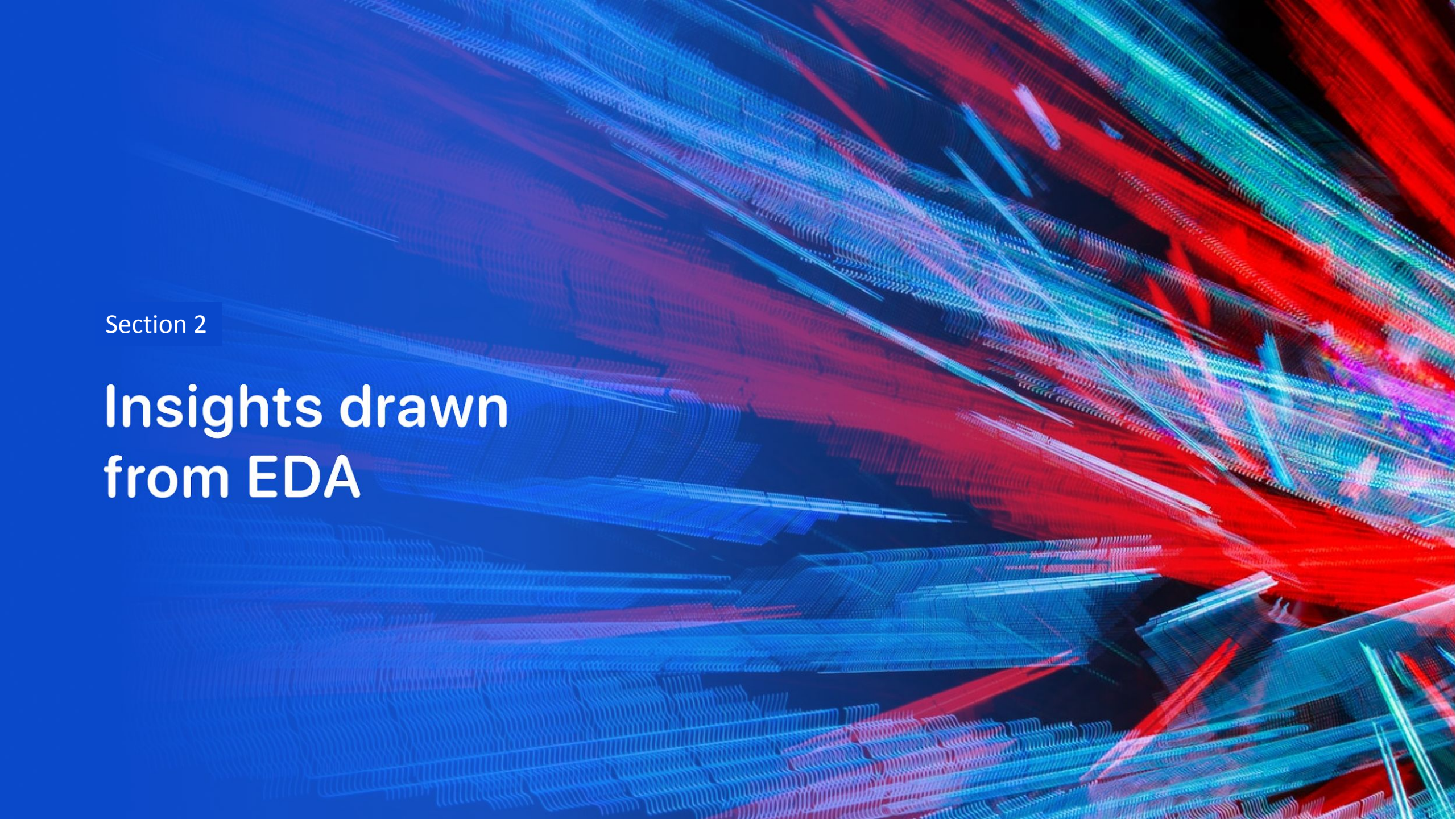
- Launch Site Locations: Most launch sites are strategically positioned near the equator, which benefits rocket launches due to the Earth's rotation speed being highest at the equator.
- Safety Considerations: Launch sites are located a sufficient distance from populated areas, highways, railways, and other critical infrastructure. This minimizes the potential damage in case of a launch failure.
- Accessibility: Despite safety precautions, launch sites remain conveniently located to allow transportation of personnel and resources to support launch activities.

Predictive Analytics Insights:

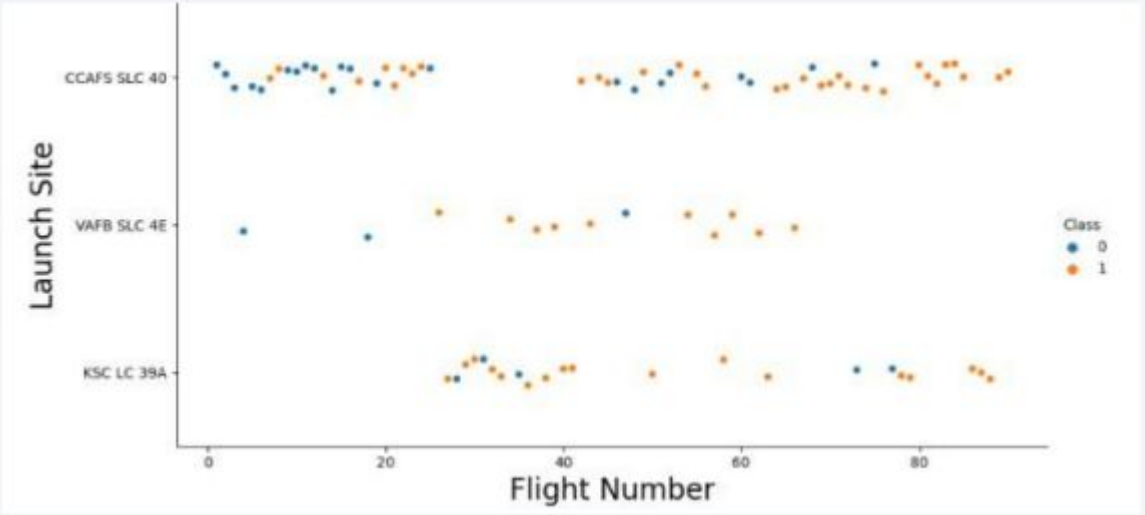
- Decision Tree Model: Among the predictive models evaluated, the Decision Tree model stands out as the best performer. This model likely demonstrated the highest accuracy in predicting launch success.

Section 2

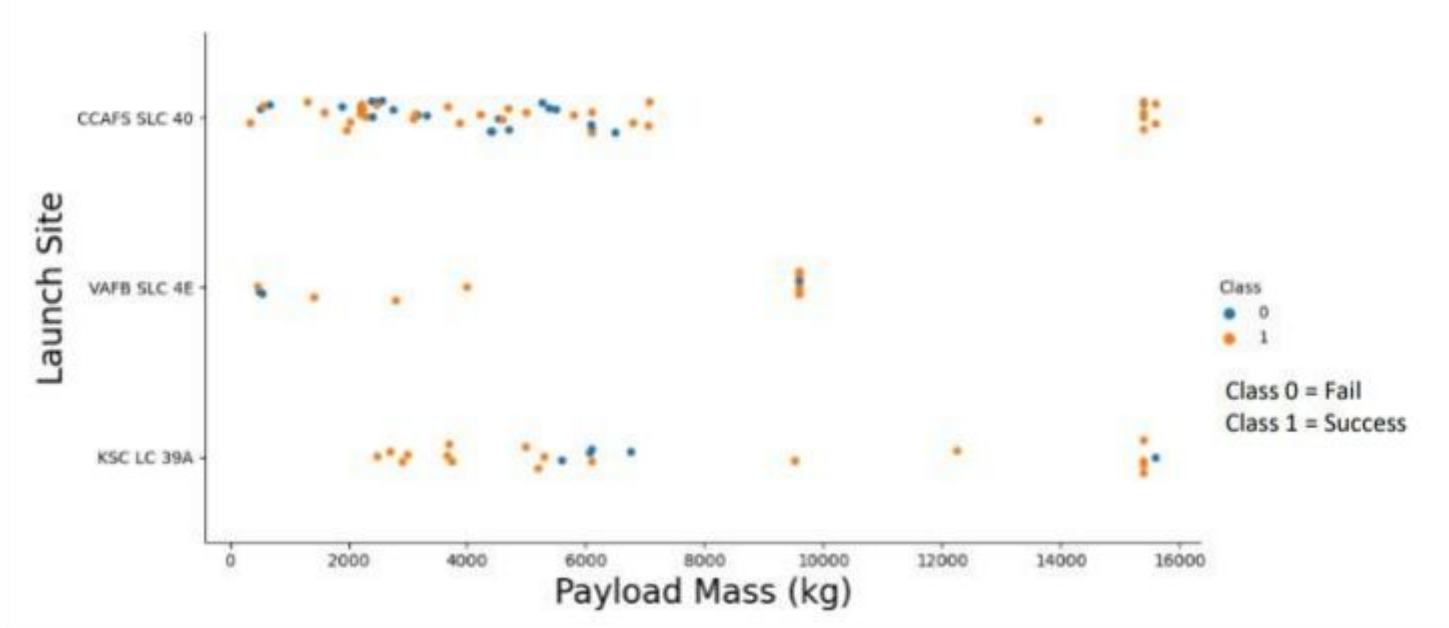
Insights drawn from EDA



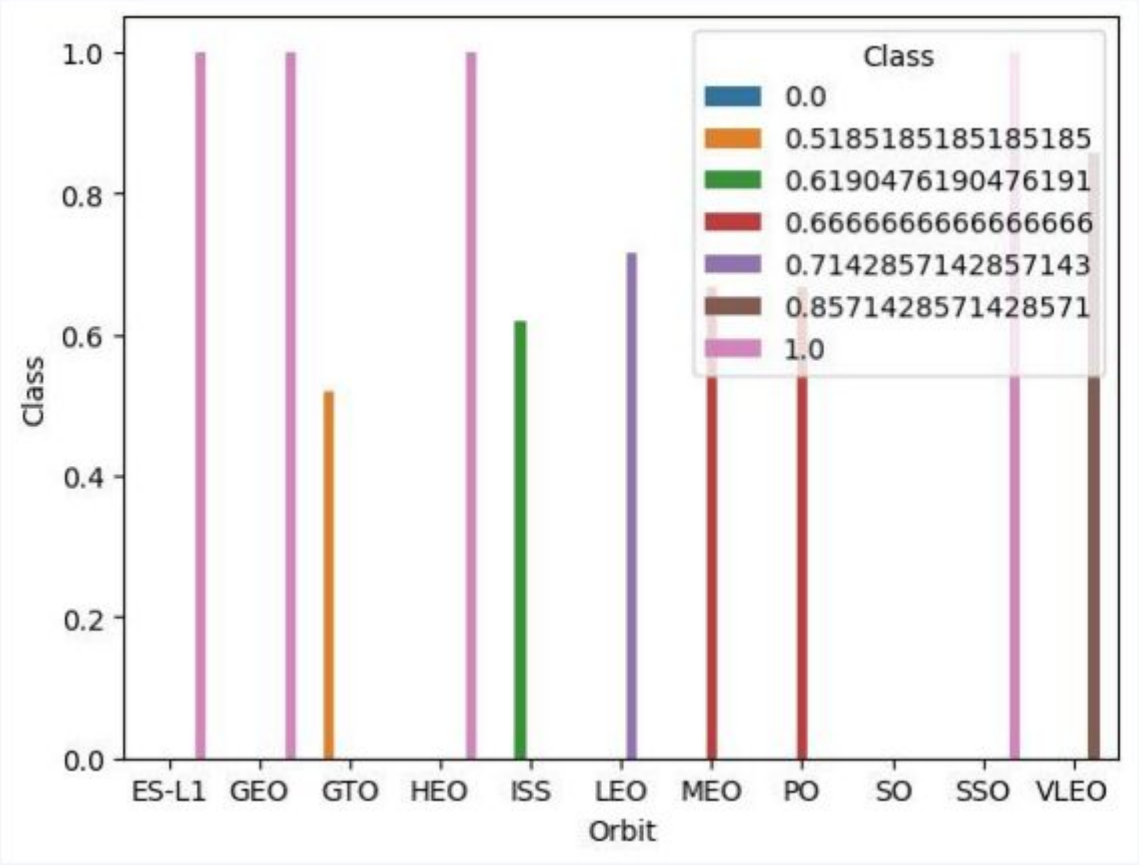
Flight Number vs. Launch Site



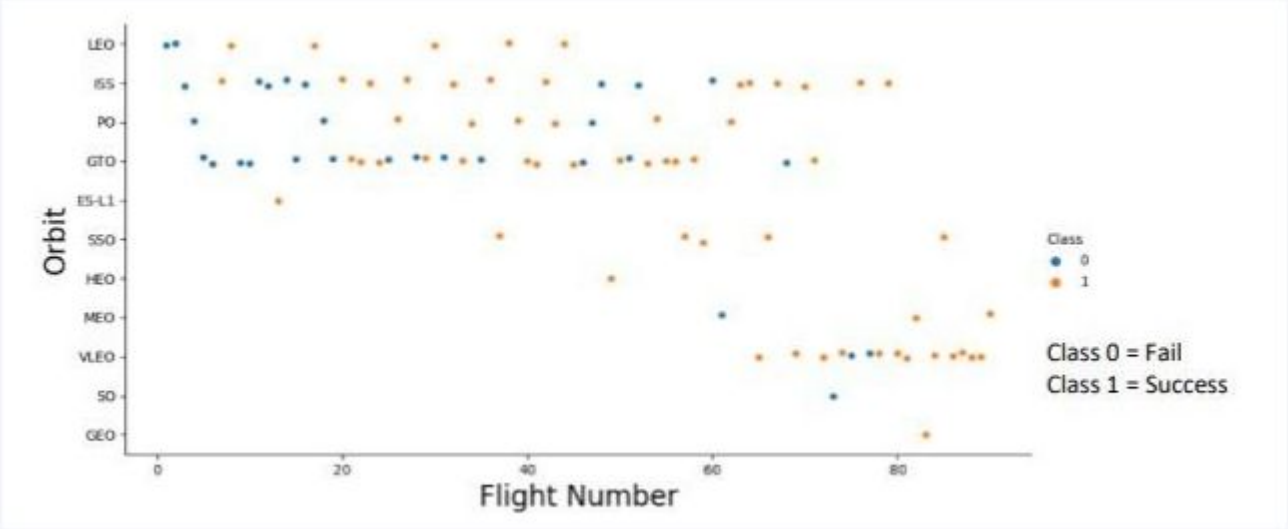
Payload vs. Launch Site



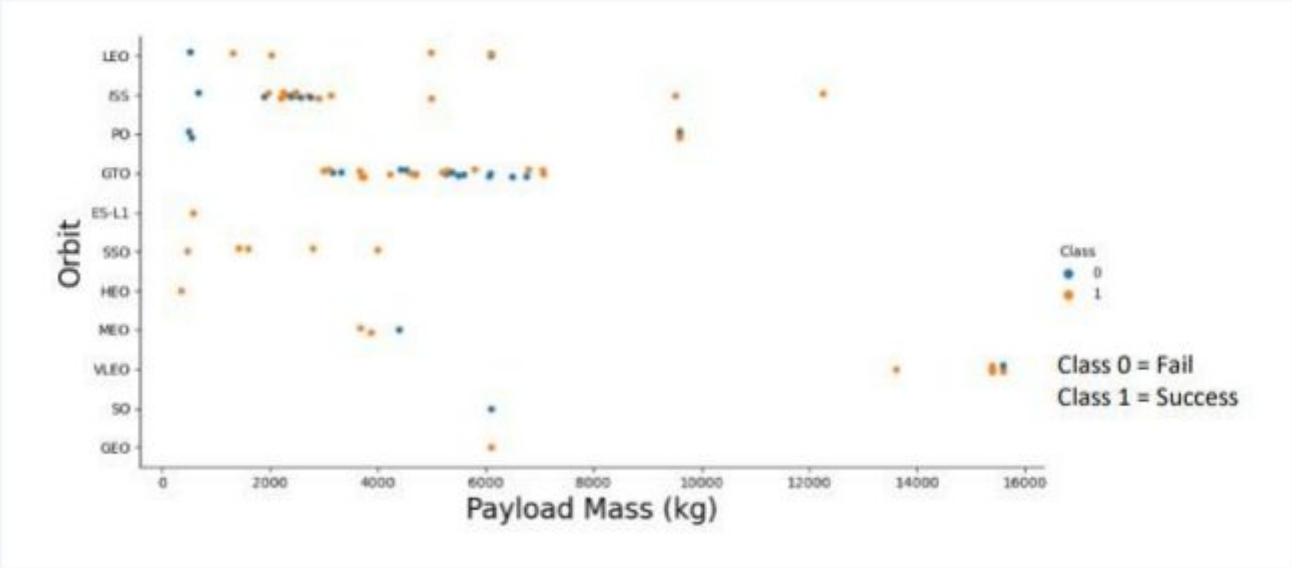
Success Rate vs. Orbit Type



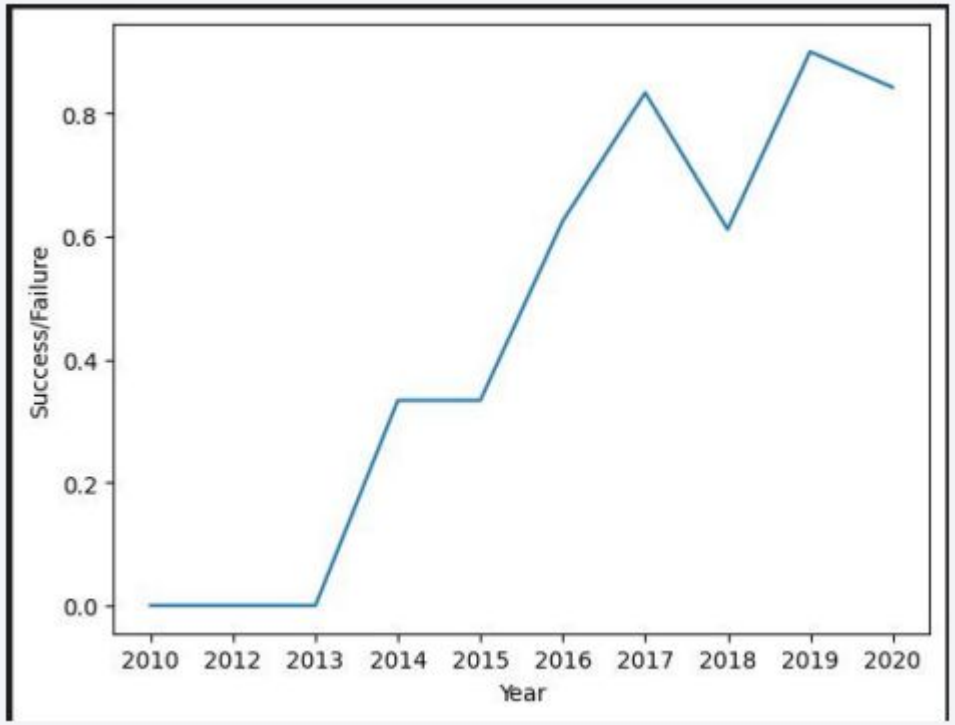
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

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

```
1  %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'

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

+ Code + Markdown

```
1  %sql select * from SPACEXTBL where launch_site like 'CCA%' limit 5;
```

Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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

```
1  %sql select sum(payload_mass__kg_) from SPACEXTBL where customer = 'NASA (CRS)';
```

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

sum(payload_mass__kg_)
45596

Average Payload Mass by F9 v1.1

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

```
1  %sql select avg(payload_mass__kg_) from SPACEXTBL where booster_version like 'F9 v1.1%';
```

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

avg(payload_mass__kg_)
2534.6666666666665

First Successful Ground Landing Date

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

Hint:Use min function

```
1  %sql select min(DATE) from SPACEXTBL where "Landing_Outcome" = 'Success (ground pad)';
```

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

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
1  %sql select distinct(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

List the total number of successful and failure mission outcomes

```
1  %sql select  mission_outcome,count(*) as Counter from SPACEXTBL group by mission_outcome;
```

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

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

Boosters Carried Maximum Payload

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

```
1  %sql select distinct(booster_version) from SPACEXTBL where payload_mass__kg_  = (select max(payload_mass__kg_) from SPACEXTBL);
2
```

```
 * 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 the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
1  %sql select  landing_outcome,booster_version,launch_site,date from SPACEXTBL where Date like '2015%' and landing_outcome = 'Failure (drone ship)';
2
```

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

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

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.

```
1  %sql select landing_outcome,count(landing_outcome) as count from SPACEXTBL where DATE BETWEEN '2010-06-04' and '2017-03-20' GROUP BY landing_outcome ORDER BY count(landing_outcome
2
```

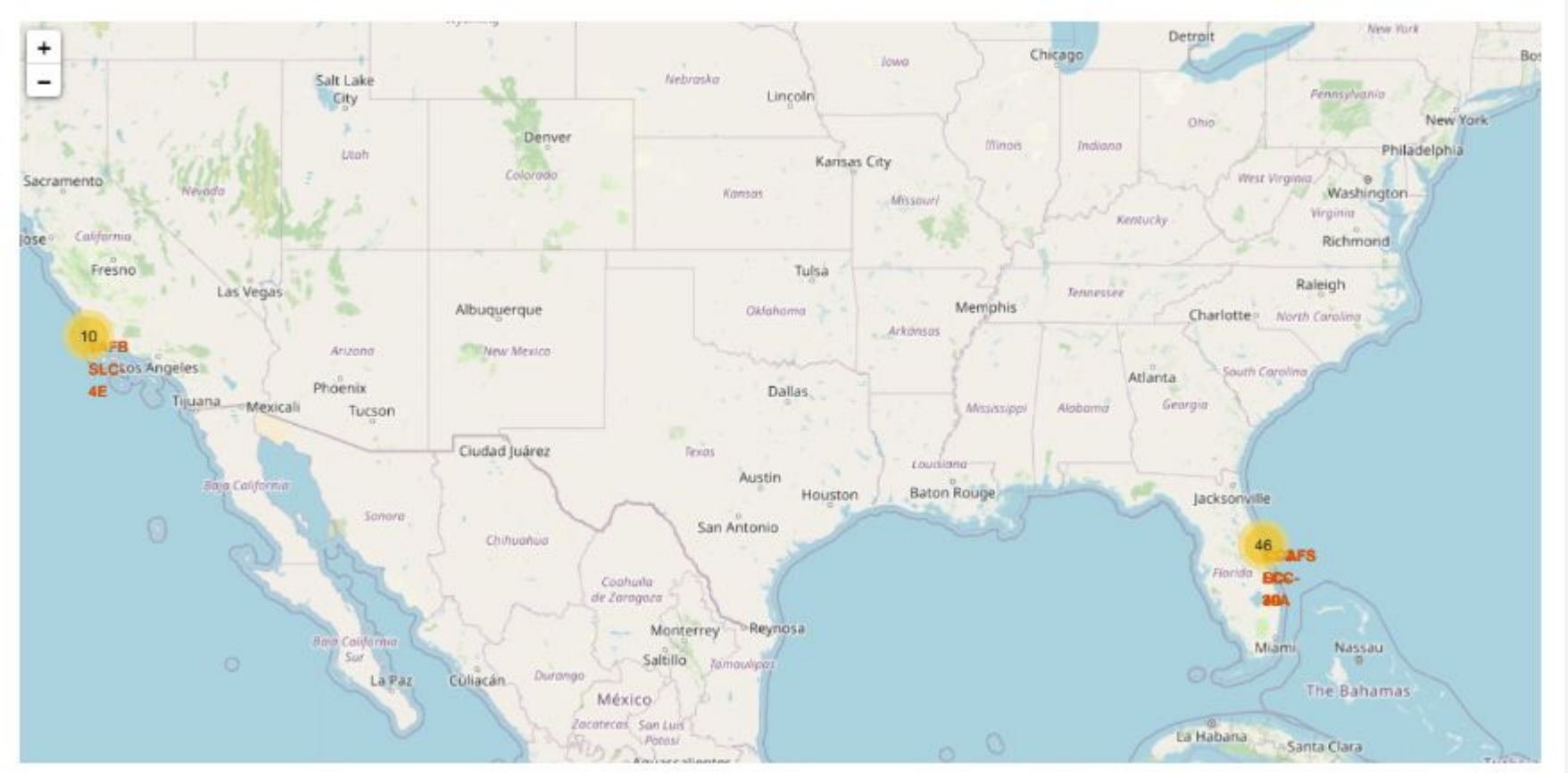
Python

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

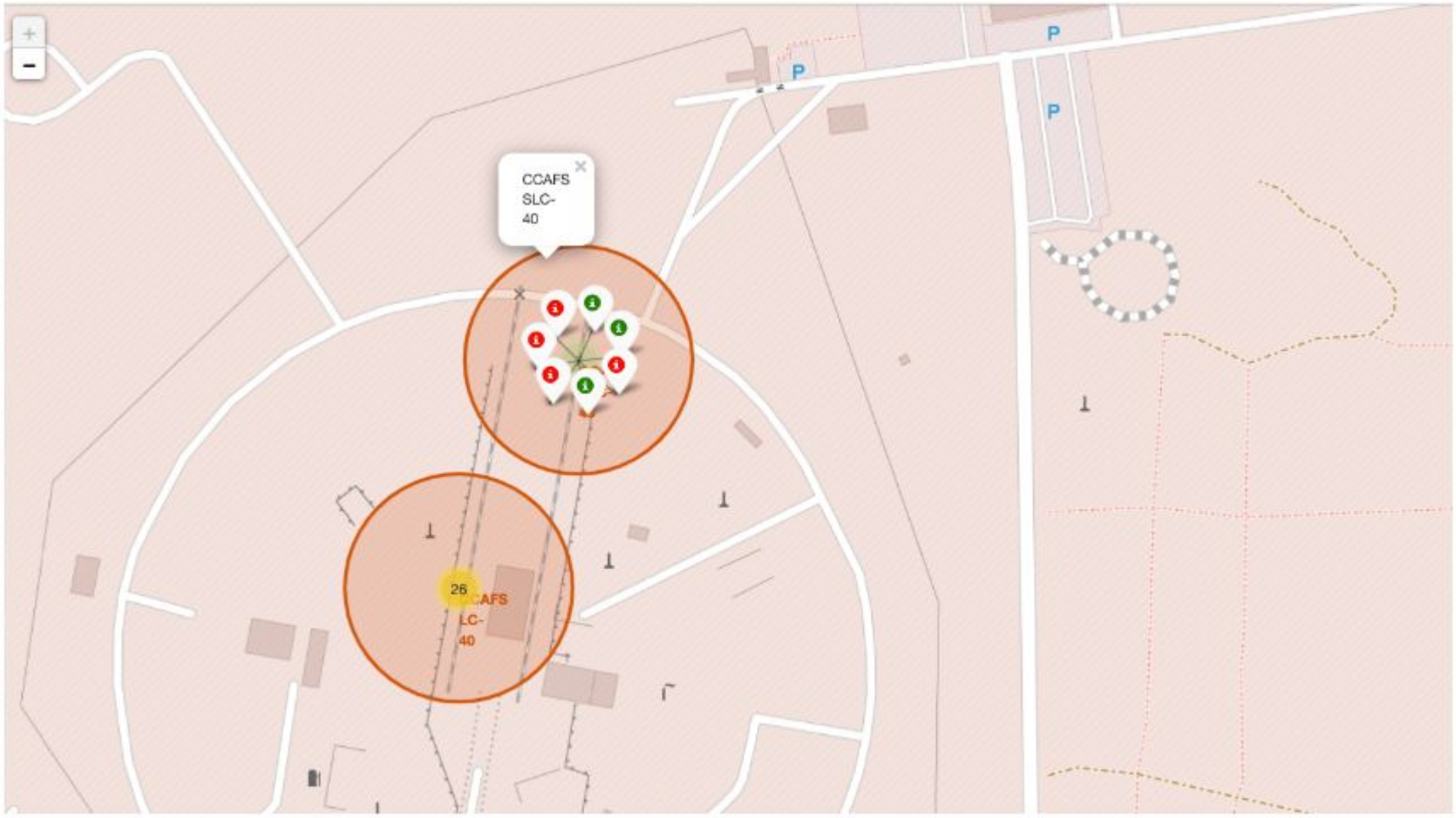
Landing_Outcome	count
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1


Section 3
Launch Sites
Proximities Analysis

<Folium Map Screenshot 1>

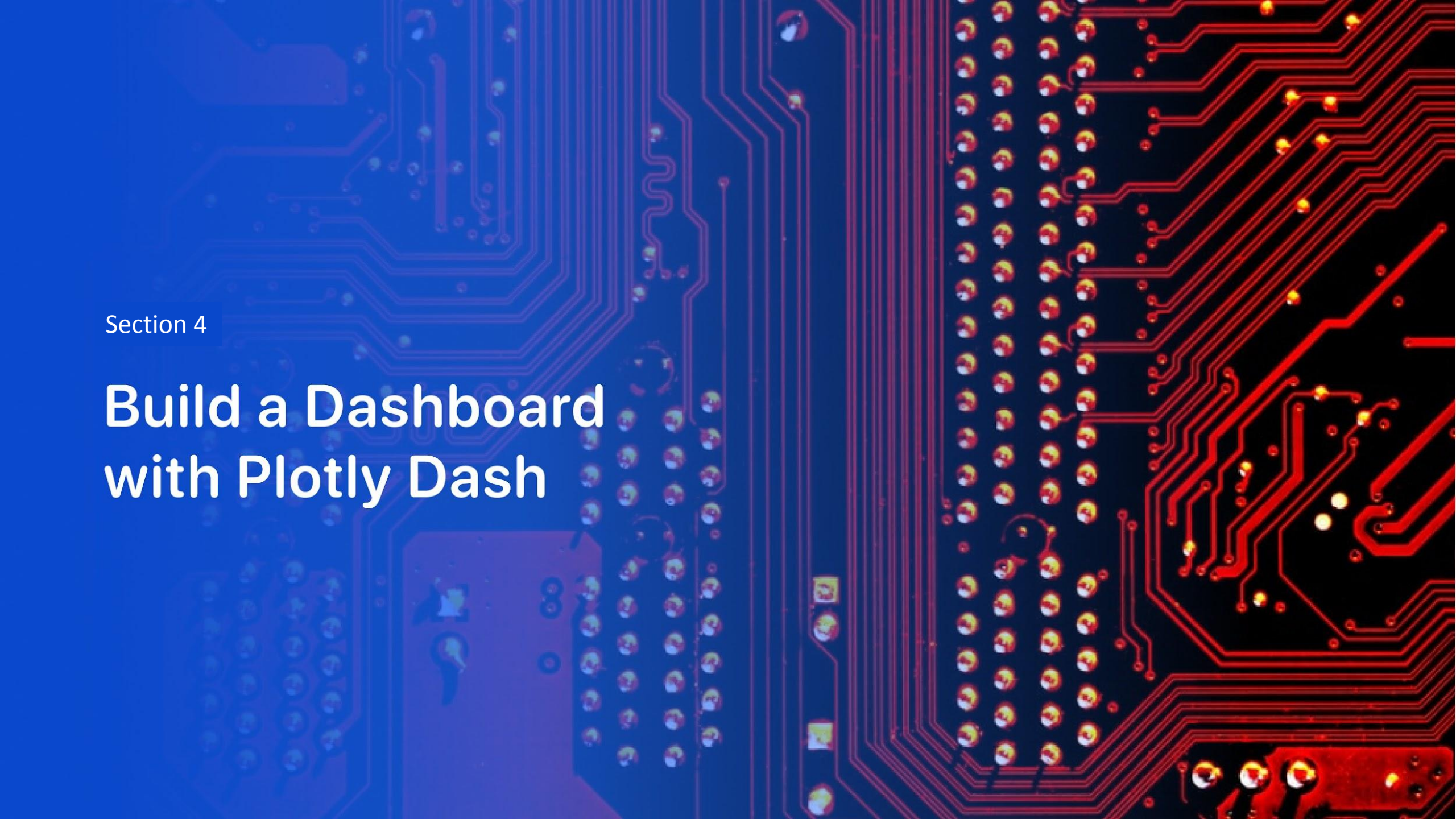


<Folium Map Screenshot 2>

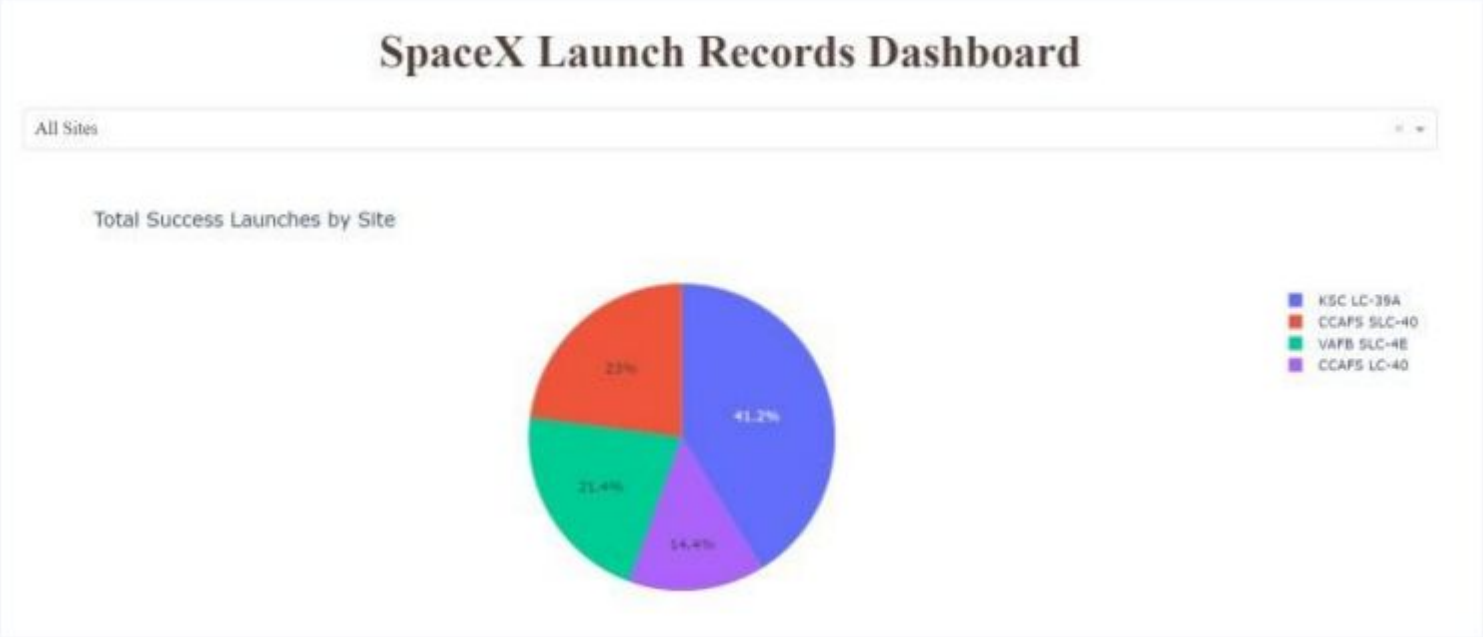


From the color-labeled markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rates.

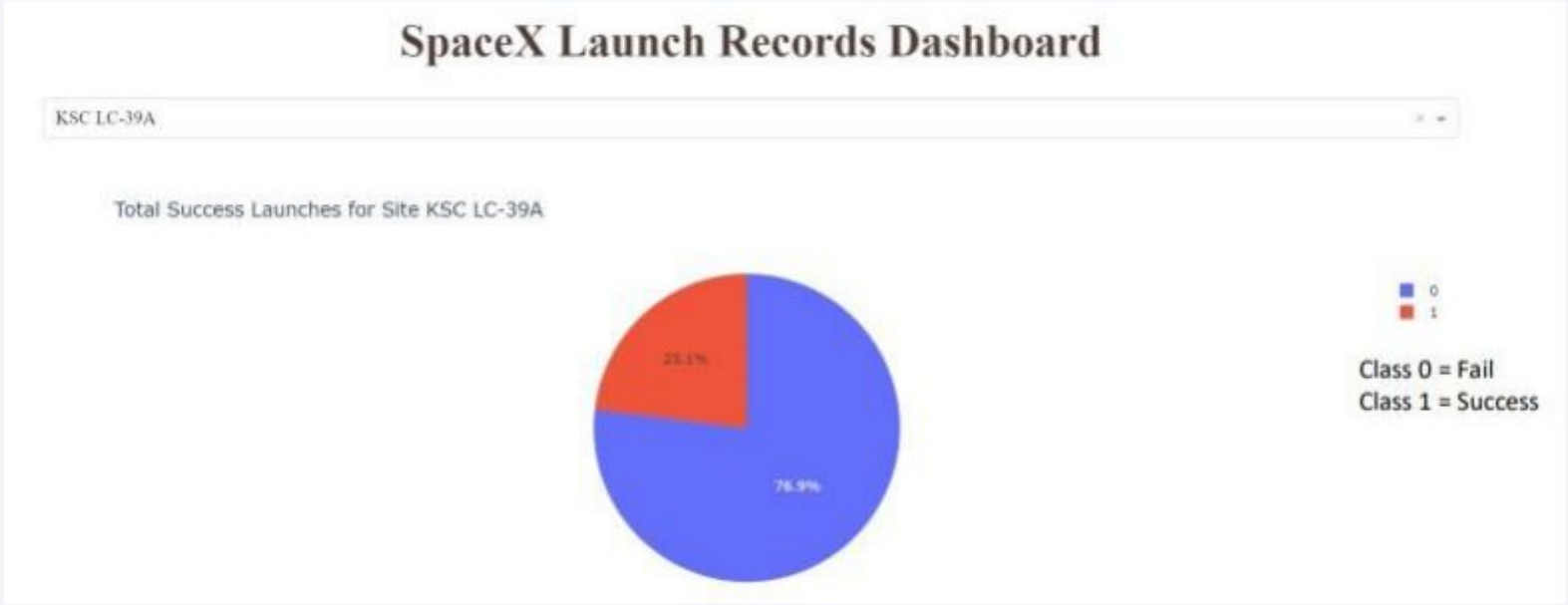
<Folium Map Screenshot 3>


Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



<Dashboard Screenshot 2>



<Dashboard Screenshot 3>

Section 5

Predictive Analysis (Classification)

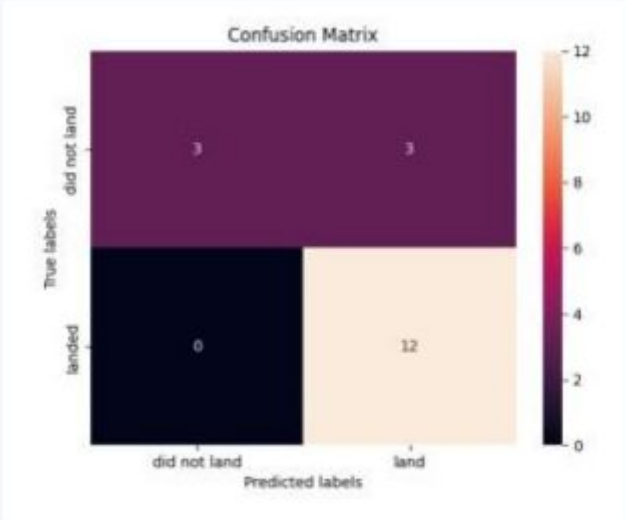
Classification Accuracy

The models demonstrated similar performance levels with consistent scores and accuracy. This uniformity is probably a result of the limited size of the dataset. Among these models, the Decision Tree model exhibited a marginal superiority, particularly evident when examining the `.best_score_` parameter.

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

Confusion Matrix

- A confusion matrix provides an overview of how well a classification algorithm performs.
- Interestingly, all the confusion matrices were exactly the same.



Conclusions

- Coastal Proximity: All the launch sites are situated in close proximity to the coast.
- Orbit Outcomes: Orbits like ES-LI, GEO, HEO, and SSO exhibit a remarkable 100% success rate.
- Payload Mass Effect: Irrespective of launch site, there's a positive correlation between higher payload mass (kg) and a higher likelihood of success.
- Model Comparison: The models showcased similar performance levels on the test set, with the decision tree model slightly outperforming others.
- Equatorial Advantage: Many launch sites are strategically located near the equator, leveraging Earth's rotational speed for added efficiency. This minimizes the need for extra fuel and boosters, leading to cost savings.
- Success Trend: The success rate of launches has shown an upward trend over time.
- KSC LC-39A Superiority: KSC LC-39A stands out with the highest success rate among launch sites, achieving a 100% success rate for launches with payloads under 5,500 kg.

Thank you!