Winning Space Race with Data Science

Itesiwaju Obasa
11/4/2024

Outline

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

Executive Summary

- The primary goal of this data science project was to predict the landing success of SpaceX's Falcon 9 launch modules. This predictive analysis aims to enhance understanding of the factors influencing successful launches and provide actionable insights for future missions. Initial exploratory data analysis was performed using SQL and Python to assess and visualize the data effectively. Advanced interactive visual analytics were conducted with tools like Folium and Plotly, emphasizing user interactivity and real-time data representation. The project progressed to predictive modeling, employing machine learning algorithms including Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Decision Trees.
- All models had the same accuracy of 83.4%. The general success rate across all launches was calculated at approximately 66.6%, with variations observed based on launch base and targeted orbit. This project highlights the potential of machine learning in enhancing launch strategies and operational decisions at SpaceX.

Introduction

In the race for commercial space travel, SpaceX has distinguished itself through the development of the Falcon 9 rocket, which significantly reduces launch costs by being partially reusable. The Falcon 9's first stage is designed to return to the launch site and land successfully, allowing it to be refurbished and reused in subsequent launches. This reusability feature is a cornerstone of SpaceX's cost-efficiency strategy, with Falcon 9 launches priced at approximately $62 million, compared to competitors' costs of over $165 million. The ability to predict whether the Falcon 9 first stage will successfully land is not only a technical challenge but also has substantial financial implications. Accurate predictions can directly influence the economic feasibility of launches, affecting overall cost-effectiveness and competitive pricing in the commercial space launch market. This project focuses on predicting the likelihood of a successful landing of the Falcon 9's first stage. The overarching goal is to develop a predictive model that can reliably forecast landing outcomes based on various launch parameters.

Section 1

Methodology

Methodology

Executive Summary

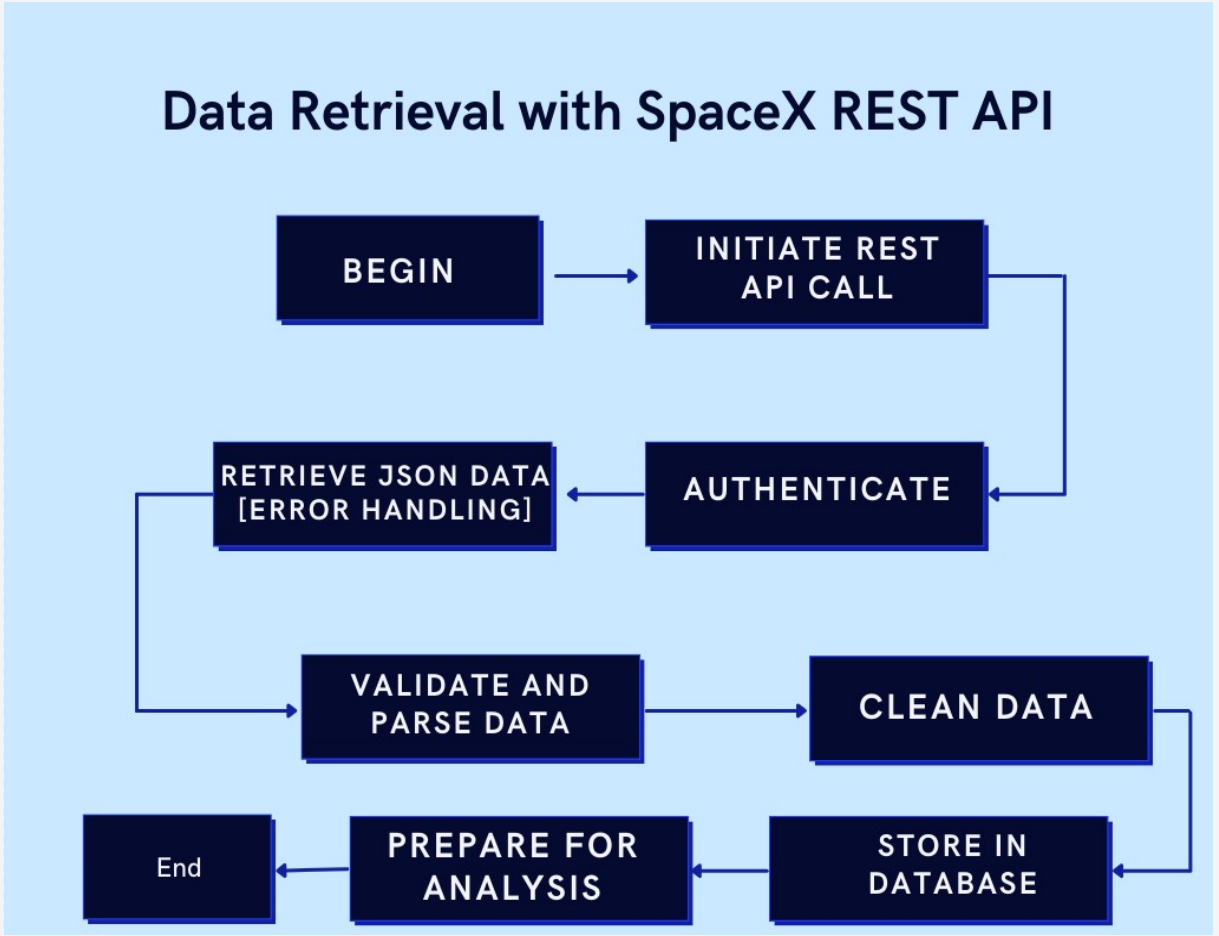
- Data collection methodology:
  - Data was extracted using a get request to the SpaceX API, and the data was filtered to tabulate only Falcon 9 launch data. Falcon 9 launch data was also obtained by web scraping from Wikipedia using the beautiful soup API.
- Perform data wrangling
  - The data was cleaned and the missing data in the payload mass column was replaced with the mean value
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We divided data into the training and test data. We create an ML algorithm object with a cross-validation value of 10. We tune to find the best parameters and calculate the accuracy.

Data Collection

- SpaceX API Data Retrieval
  - Accessed SpaceX's API to extract launch data.
  - Employed GET requests to fetch relevant launch parameters.
  - Collected data on launch dates, payloads, launch outcomes, and landing information.
- Web Scraping from Wikipedia
  - Implemented Python scripts with libraries like Beautiful Soup and requests.
  - Scraped structured data from SpaceX's launch history tables on Wikipedia.
  - Merged API data with the scraped Wikipedia data.
- Data Integration
  - Cleaned and transformed data to a uniform structure for analysis.
  - Conducted a preliminary review of the data to identify any anomalies or patterns.
  - Used exploratory data analysis techniques to summarize the main characteristics.

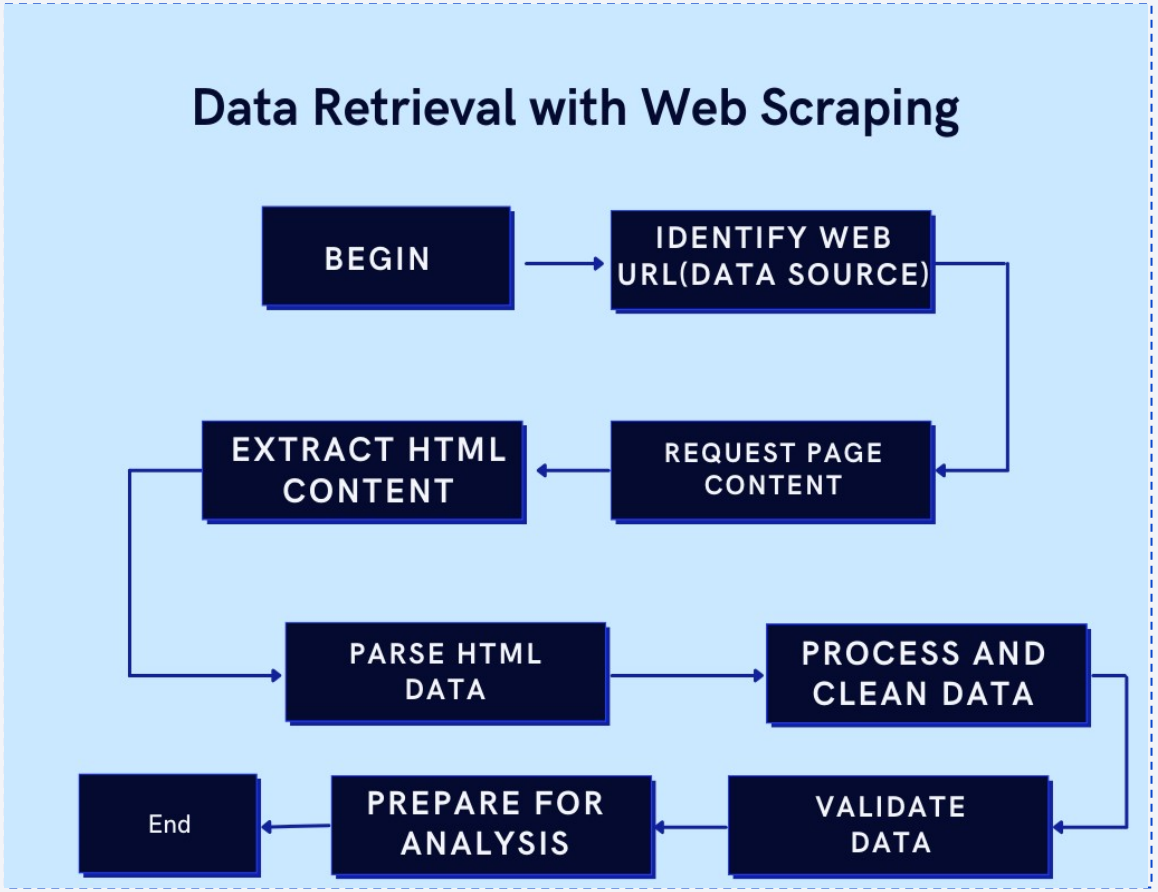
Data Collection – SpaceX API

- The data collection for our SpaceX project was streamlined through a series of methodical steps beginning with initiating HTTP GET requests to SpaceX's RESTful API to gather Falcon 9 launch data. We efficiently retrieved extensive datasets, including payloads, launch dates, and outcomes, formatted in JSON. Subsequent steps involved validation and error handling to maintain data integrity and manage any inconsistencies. The retrieved JSON data underwent parsing to distill necessary fields and thorough cleaning to eliminate nulls or unnecessary information. The resultant dataset was then systematically stored in a structured database, prepared for comprehensive exploratory data analysis (EDA) and further analytical processing.
- https://github.com/TesiX635/Data-Collection-Rest-API-/blob/24cecd99fb20c7be7711fdf52e95e6a9bae74b75/jupyter-labs-spacex-data-collection-api%20(2).ipynb



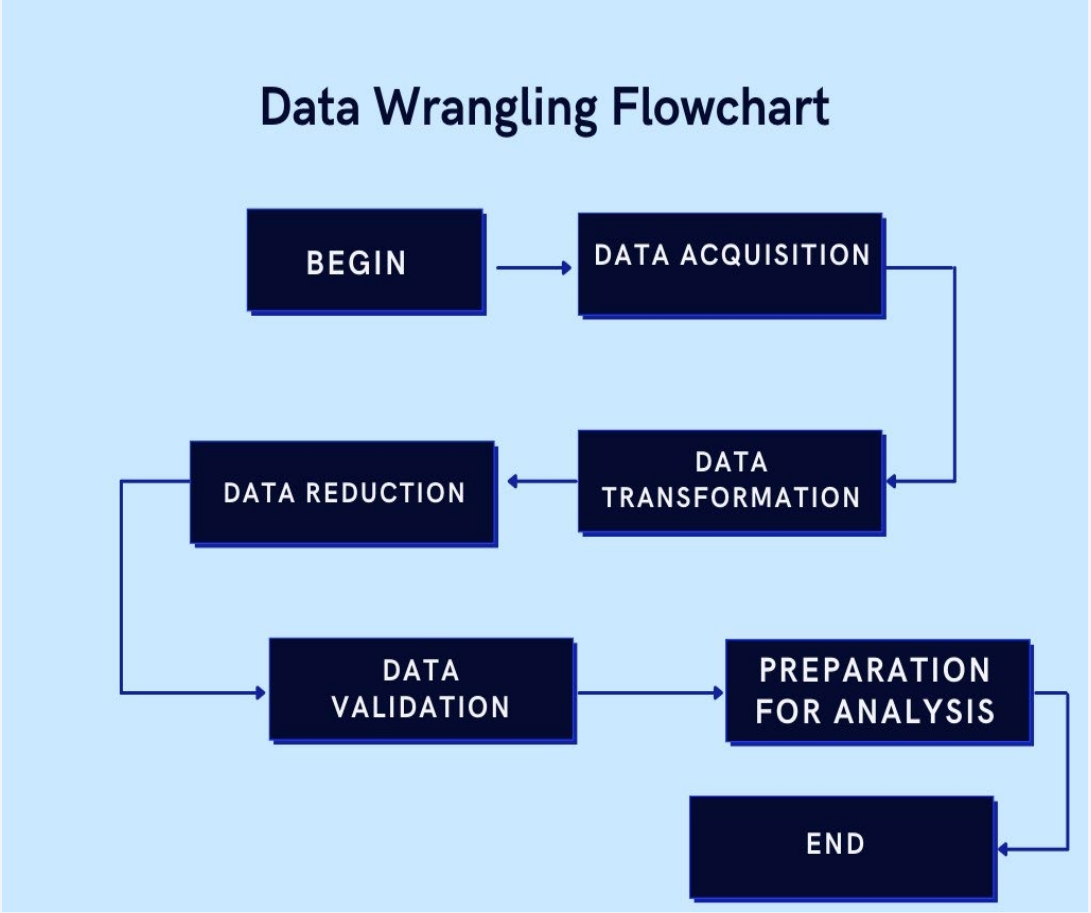
Data Collection - Scraping

- The web scraping process for SpaceX launch data from Wikipedia involved initiating an HTTP GET request for page content, which was then parsed using Beautiful Soup to extract launch data from HTML elements. We cleaned and formatted the data, converting it into structured JSON formats. The integrity of the data was ensured through thorough validation checks for completeness and consistency.
- https://github.com/TesiX635/Data-Collection-Rest-API-/blob/1f295c1532279a8b51cef23e9b2c1c39c5af962c/jupyter-labs-webscraping%20(2).ipynb



Data Wrangling

- The data wrangling process commenced with data acquisition, where we imported data from APIs and web scrapings into our processing environment. We then engaged in data cleaning to fill in missing values, eliminate duplicates, and rectify inconsistencies while standardizing formats for uniformity. The transformation phase involved converting data types and normalizing numerical data for analysis readiness. Data reduction was applied to discard irrelevant details and aggregate essential information. After which the data was validated and ready for further analysis.
- https://github.com/TesiX635/Data-Collection-Rest-API-/blob/a3f10d02b1ecd0c1be2be9e1470f4be1de440811/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb



EDA with Data Visualization

- In the EDA section, we plotted several graphs to get significant insight into the data via visualization. We used different plots such as the scatter plot which is ideal for observing the relationship between two quantitative variables. We used this to visualize the payload/flight number relationship, Launch Site/Flight number relationship, Launch Site/Payload relationship and the orbit/flight number relationship. We used the bar chart to see the success rate of each orbit, and to check if there's a relationship between the success rate and orbit type. Finally, we used the line graph to see the average yearly success rate trend, as line graphs are useful for showing trends over time.
- https://github.com/TesiX635/Data-Collection-Rest-API-/blob/5c80763453b745e8df2d0004986a2410245ee3f0/jupyter-labs-eda-dataviz%20(1).ipynb

EDA with SQL

- The queries were carried out as instructed in the labs, we selected the distinct launch sites from the data base using the select and distinct queries and we then found the record of launch sites that start with 'CCA' using the like query. We found the total payload mass for missions launched by NASA using the sum and where functions. Average payload mass carried by booster F9v.1.1 was found using the AVG function. We used the MIN(Date) function to select the date of the first successful landing from the table. The names of the boosters between 4000 and 6000kg payload, with successful landing on drone ships where selected. Used the count function to count the total number of success and failure mission outcomes and ranked the different landing outcomes between specific dates. These queries were designed to extract specific insights from the SpaceX launch dataset, ranging from identifying unique attributes, summarizing numerical data, to applying filters for custom requirements. A link to the GitHub is attached below
- https://github.com/TesiX635/Data-Collection-Rest-API-/blob/72aa0a7f4608850b2c3b642006740f42c7163ff2/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

- Map Objects created:
- Launch site markers: We plotted each launch site on the map using their latitude and longitude coordinates. Markers are added to indicate the exact locations of the launch sites, helping to visualize their geographical placement.
- MarkerCluster: A MarkerCluster is introduced to group close proximity markers, which helps to manage clutter on the map when zoomed out, as many launches occur from the same sites.
- Success/Failure Markers: Markers are color-coded—green for successful launches (class=1) and red for failures (class=0). This visual distinction quickly communicates the success rate of launches from each site.
- Distance Measurements: Using folium.PolyLine, we calculated and displayed the distance from a launch site to the nearest coastline. This is to understand the safety and logistical considerations for choosing launch sites, as proximity to the coast can influence launch window availability.
- https://github.com/TesiX635/Data-Collection-Rest-API-/blob/72f7c634ce02f2b6ff3093b1fe759ca8f42fd3b7/lab_jupyter_launch_site_location%20(1).ipynb

Build a Dashboard with Plotly Dash

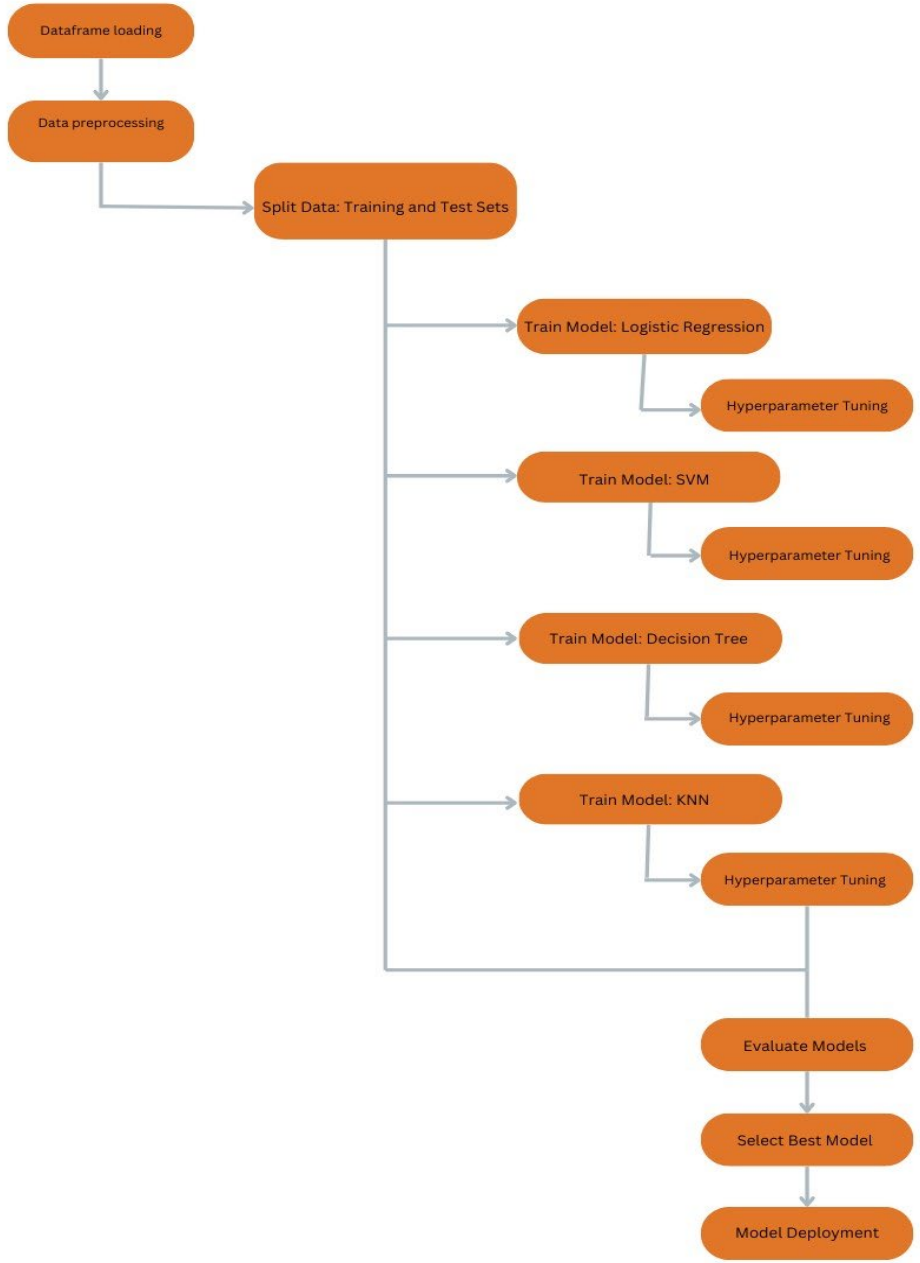
- Summary of plots/graphs and interactions added to the dashboard:
- Dropdown for Launch Site Selection: A dropdown menu allows us to filter the dashboard views for either all launch sites or a specific one. This interaction lets users focus their analysis on a single site or compare across all sites.
- Pie Chart for Launch Success Rates: When no specific launch site is selected, the dashboard displays a pie chart summarizing the total number of successful launches per site. If a site is selected, it shows the proportion of successful versus failed launches at that site. This visualization is crucial for understanding the reliability and success rates of different SpaceX launch sites or the company's overall success rate.
- Payload Range Slider: A slider is provided to select a range for the payload mass. It enables users to explore how payload mass affects launch success by limiting the data to specific mass ranges. This interaction can provide insights into the capabilities of SpaceX rockets regarding the mass they can carry to space successfully.
- Scatter Plot for Payload vs. Success: The scatter plot illustrates the correlation between payload mass and launch success. If 'ALL' sites are selected, it shows the data for all sites, and if a specific site is chosen, it filters the data accordingly. This plot is particularly useful to analyze whether heavier payloads influence the success rate of launches.
- https://github.com/TesiX635/Data-Collection-Rest-API-/blob/81302f7965ba1c3f239b014013f7c5c9a19e4c70/Ploty%20Assignment.py

Predictive Analysis (Classification)

- Summary of how we built, evaluated, improved, and found the best performing classification model:
- Data Preparation: The dataset is loaded and standardized using StandardScaler to ensure that the features contribute equally to the model training.
- Data Splitting: The dataset is split into training and testing sets using train_test_split, which is important for training the models and validating their performance on unseen data.
- Model Training and Hyperparameter Tuning: A logistic regression model is initialized and hyperparameters are tuned using GridSearchCV with 10-fold cross-validation to find the best performing parameters. Similarly, support vector machines (SVM), decision tree classifiers, and k-nearest neighbors (KNN) models are also trained with their respective hyperparameter tuning.
- Model Evaluation: The accuracy of each model is evaluated on the test set. Confusion matrices are plotted for each model to understand their performance in terms of false positives and false negatives.
- Model Selection: The best performing model is selected based on the test accuracy and the insights drawn from the confusion matrices. This model can then be used for predicting whether the first stage of the Falcon 9 rocket will land successfully.
- https://github.com/TesiX635/Data-Collection-Rest-API-/blob/6c691252f0a874b0f8b95d718795da1a9842206b/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(1).ipynb

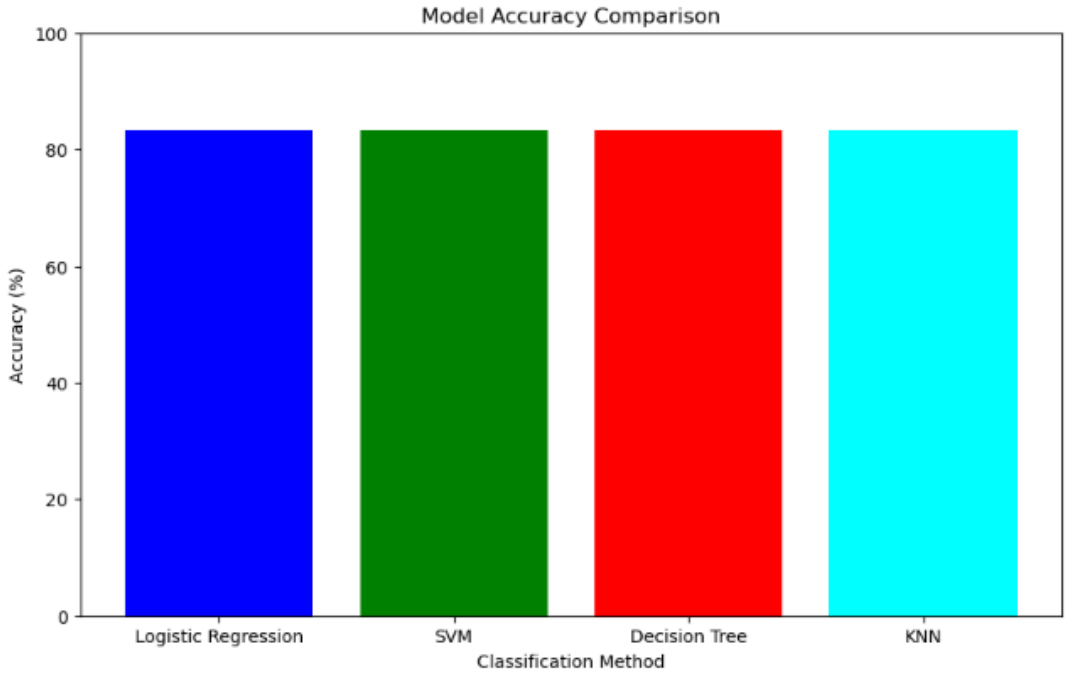
Predictive Analysis (Classification)

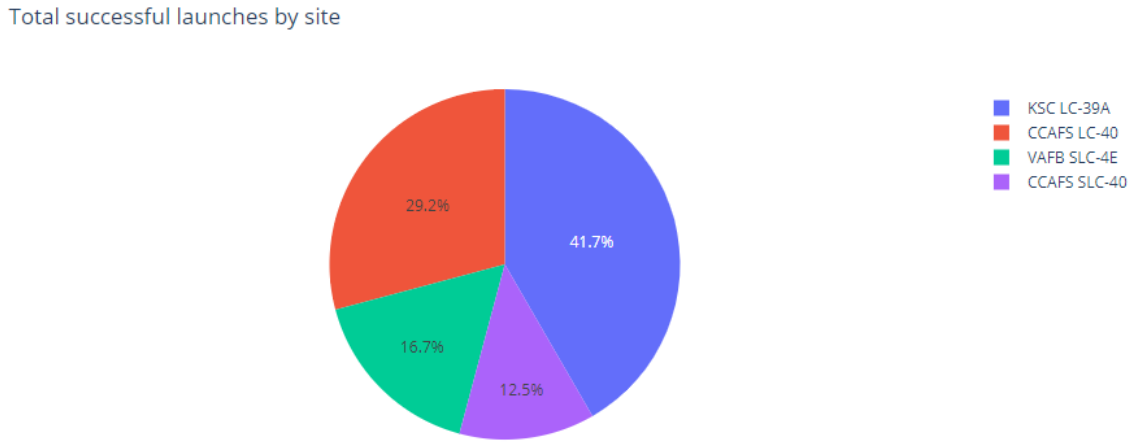
- This slide shows a flowchart for the machine learning predictive analysis. It is a short summary of the previous slide.



Results

- Exploratory data analysis gives interesting insight into the extracted data, we can see in the pie chart below the total successful launches from each of the sites. We can also determine the total successful and unsuccessful launches by further interacting with the dashboard
- Predictive analysis shows us that all the models have the same accuracy therefore any of these models will be optimal for model deployment, however more tuning and further scrutiny may find a particular method to be preferred.



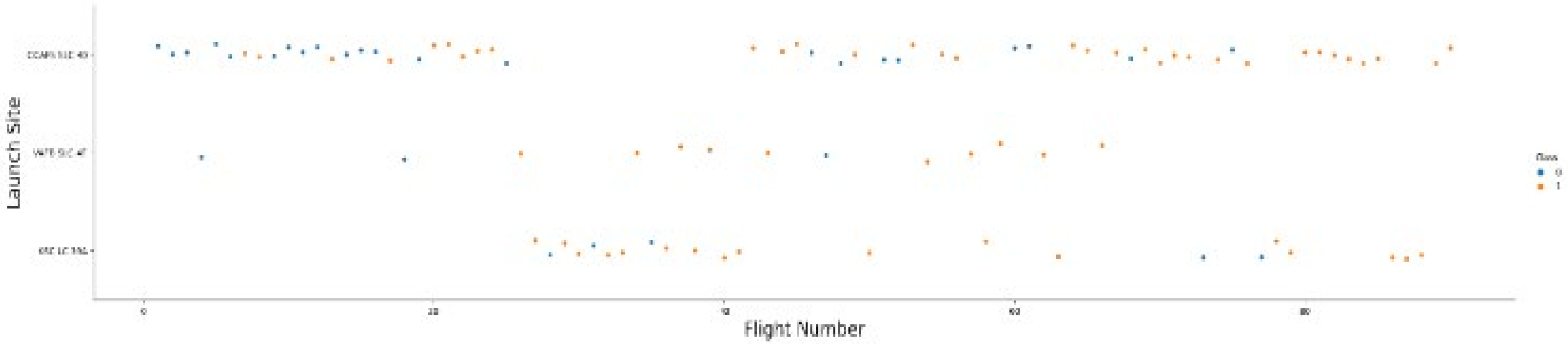


Section 2

Insights drawn from EDA

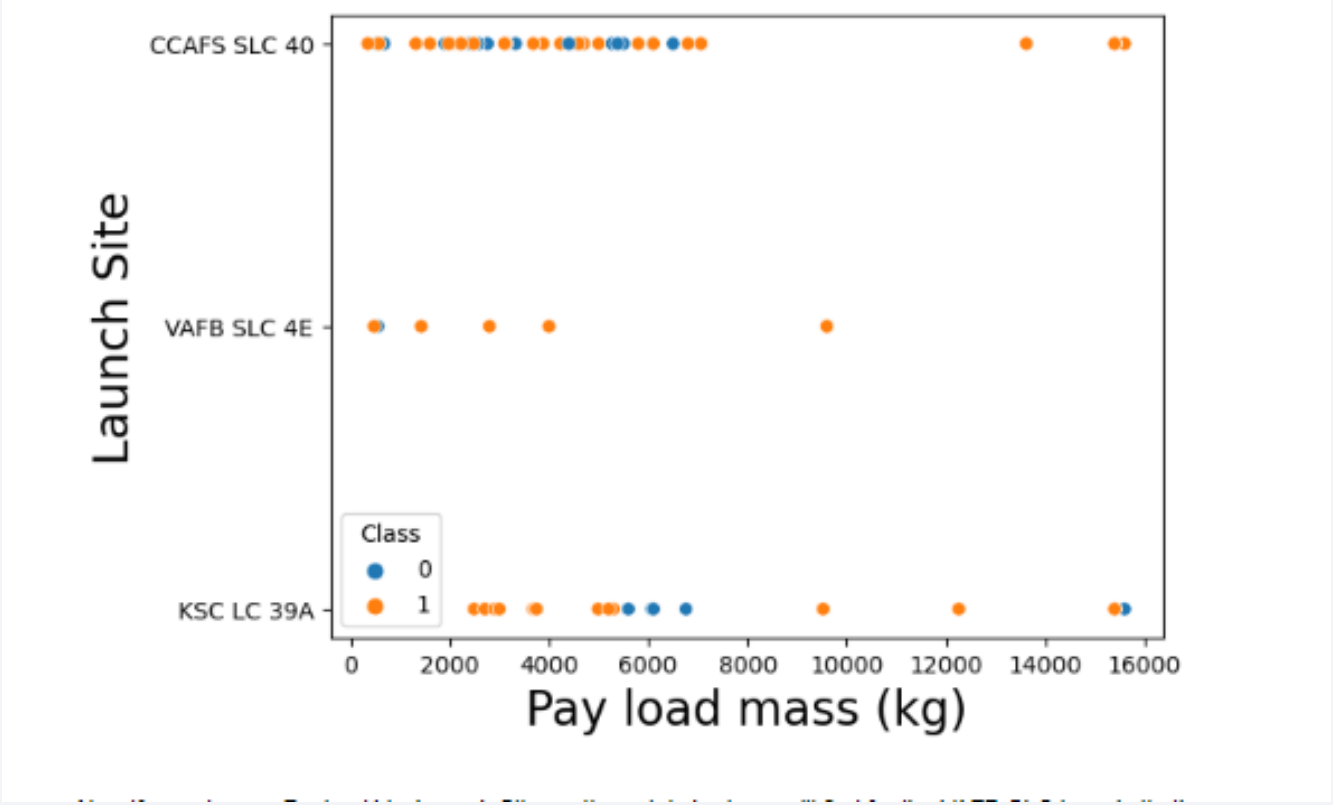
Flight Number vs. Launch Site

- We can see that in the first 20 launch attempts only 3 are successful and are mostly in the CCAFS SLC 40 with 2 failed attempts from the next site.
- In the next 20 flights there's only 3 unsuccessful and they mostly occur on the last launch site.
- In the last group of launches of about 30, we note that they mostly occur at the first launch site and there's only about 6 unsuccessful launches, 4 of which occurred on the first launch site, hoe ever the second launch site has only two launches in this period both of which are successful



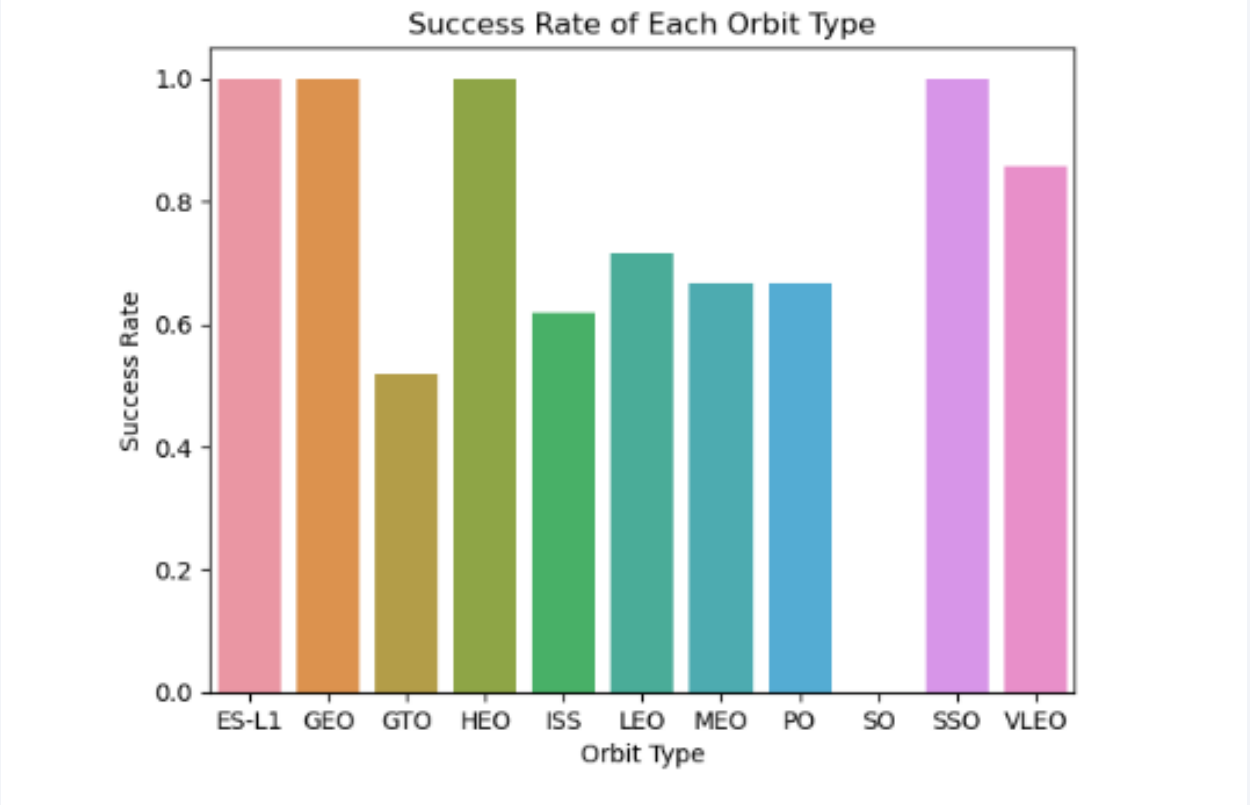
Payload vs. Launch Site

- We can see that most of the payloads are less than 8000kg
- Only one unsuccessful launch at a payload greater than 10000kg



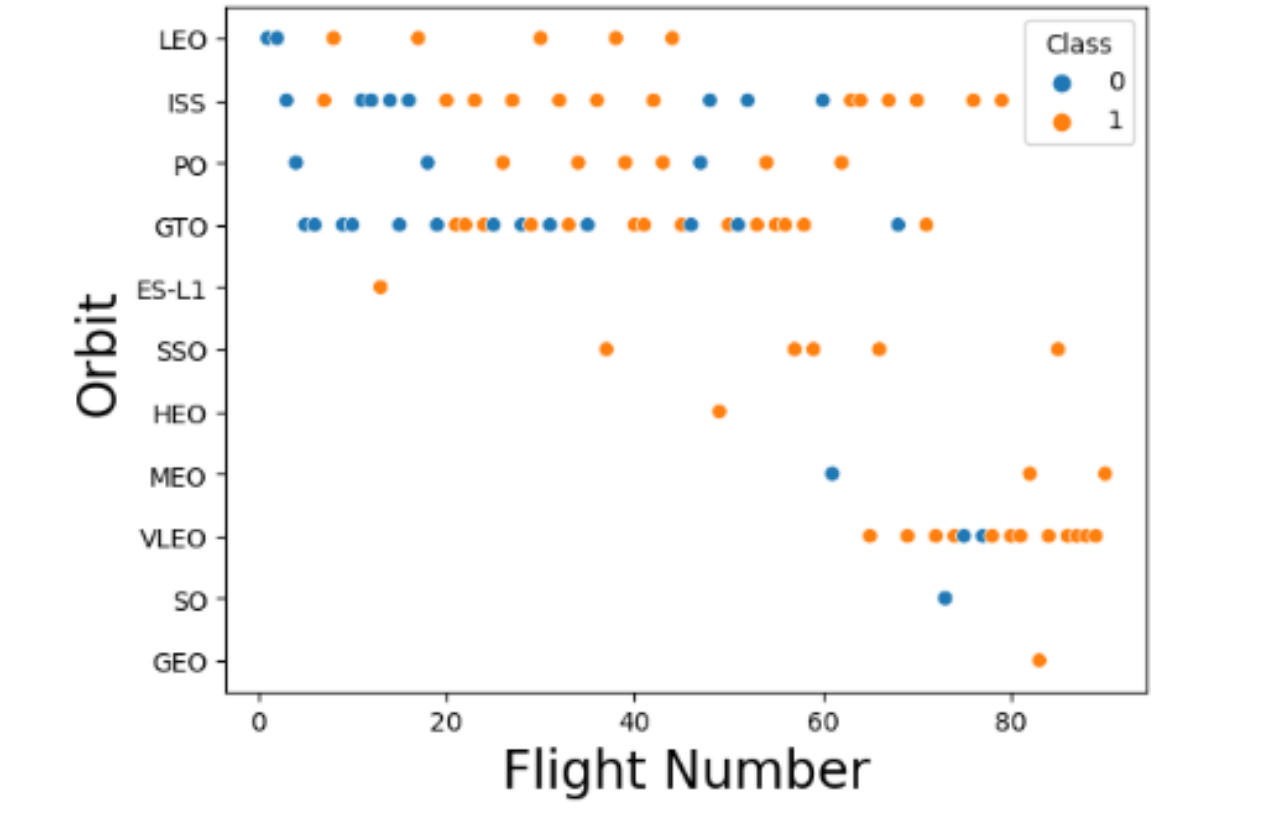
Success Rate vs. Orbit Type

- We can see from the graph that ES-L1, GEO, HEO, and SSO, all have a 100% success rate.
- The orbit SO has the lowest success rate of 0%, followed by GTO with a 50% success rate.
- The others range between 85% success rate and 60%



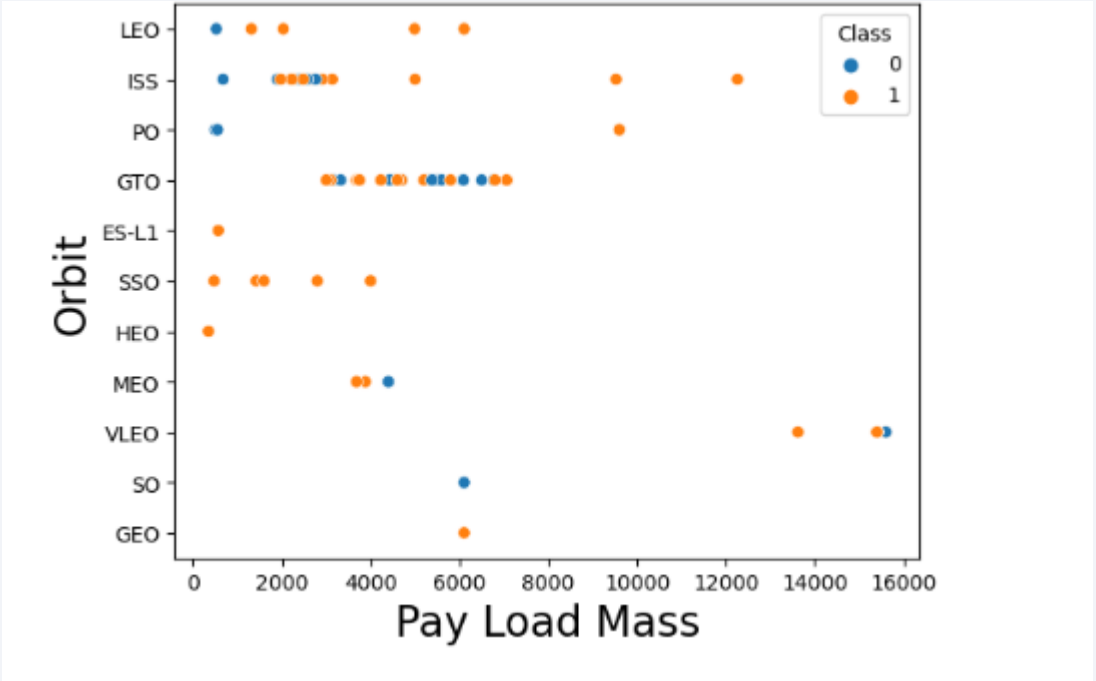
Flight Number vs. Orbit Type

- First successful launch was to the ISS orbit.
- GTO and ISS represent the greatest number of launch attempts.
- One launch to the SO and its unsuccessful, one launch the GEO and its successful.



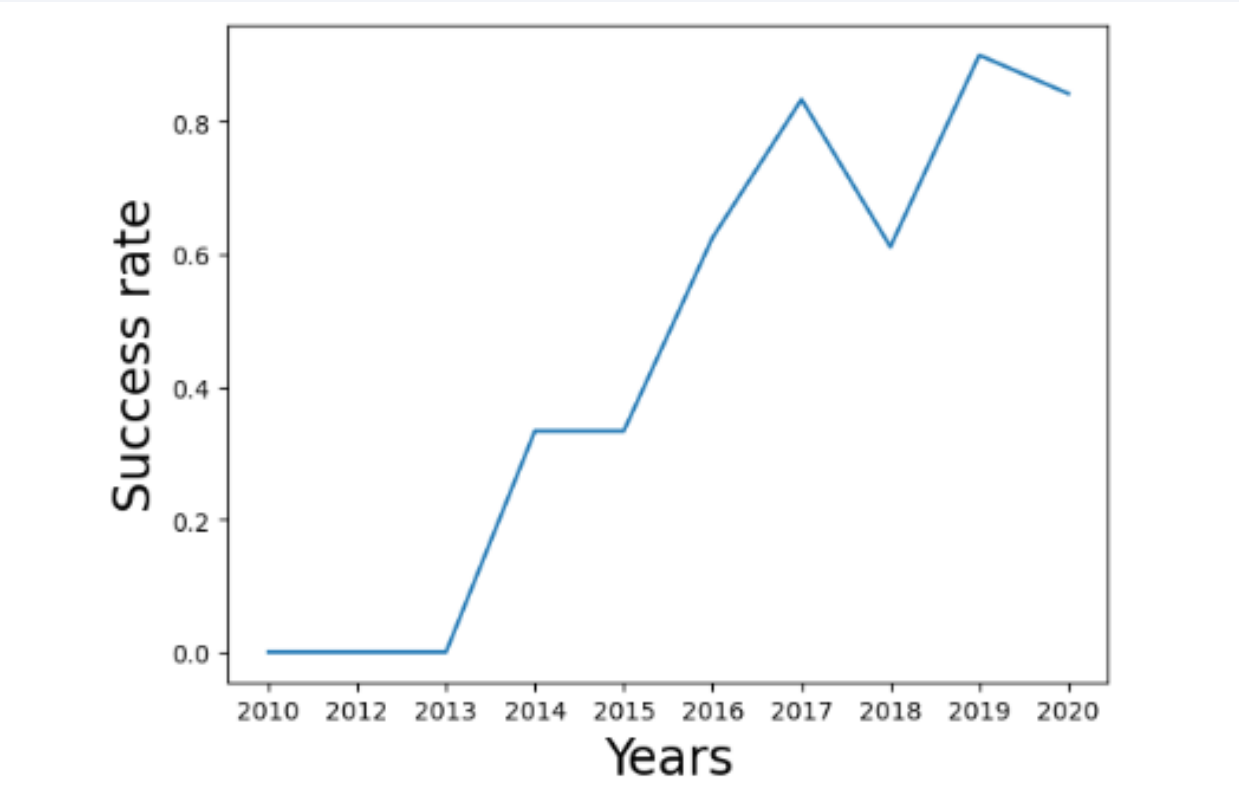
Payload vs. Orbit Type

- The payload mass for successful launches span a wide range. Unsuccessful launches are scattered across various payload masses without a clear pattern of payload size causing failure.
- Most of the unsuccessful launches are concentrated in a few orbit types (e.g., GTO, VLEO), while other orbits like LEO, SSO, and ISS (International Space Station) have predominantly successful launches.



Launch Success Yearly Trend

- There's a general upward trend in the success rate of launches over time, indicating improvements in technology, experience, and processes that have increased the likelihood of a successful launch as years progressed.
- Between 2013 and 2016, there's a steep increase in the success rate. This suggests a period of significant advancements or optimizations in launch technology or methodologies.
- The graph ends with a success rate that is notably higher than at the beginning, suggesting that since 2020, launch processes have matured and become more reliable.



All Launch Site Names

- We find the launch site name by using the select and distinct functions to select the column name from the table.

```
In [9]: %sql select DISTINCT Launch_site from SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

Out[9]:

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

Launch Site Names Begin with 'CCA'

- We use the select all function to pick the launch site column and use a where clause to filter and the like clause to filter to launch site names that begin with “CCA%”

In [11]:
```
%sql select * from SPACEXTABLE where Launch_site like 'CCA%' limit 5
```

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

Out[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

- We use the select function to retrieve the data, we sum the payload column with the sum function. WHERE Customer LIKE '%NASA (CRS)%' is a filter applied to select only the rows where the Customer column contains the string "NASA (CRS)".

```
In [17]: %sql select SUM(PAYLOAD_MASS__KG_) from  SPACEXTABLE where Customer like '%NASA (CRS)%'
 * sqlite:///my_data1.db
Done.
Out[17]: SUM(PAYLOAD_MASS__KG_)
                          48213
```

Average Payload Mass by F9 v1.1

- We use the select query to pick the payload mass column and we use the AVG query to calculate the average of this column. The where clause picks the row where the booster version is F9 v1.1

```
In [14]: %sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1'
 * sqlite:///my_data1.db
Done.
Out[14]: AVG(PAYLOAD_MASS__KG_)
         2928.4
```

First Successful Ground Landing Date

```
%sql select * from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)' limit 1
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	La
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	

- Here we find out the first landing outcome where ground pad success is equal to one using the select all query and the where clause.

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
- Here we select the booster version column from the table where the landing of the falcon 9 rocket was successful on a drone ship using the where clause. We use the and clause to extend our filter to account for the payload column greater than 400 but less than 6000

Total Number of Successful and Failure Mission Outcomes

- %sql select Mission_Outcome, Count(*) as Total_Outcomes from SPACEXTABLE where Mission_Outcome in ('Success', 'Failure (in flight)', 'Success (payload status unclear)') group by Mission_Outcome
- Here we select the mission outcome column. And we count the number of rows and append it to a column as total outcomes, and we use the where clause in conjunction with in to filter only those with a mission outcome if success, failure, and payload status unclear. And finally we group by mission outcome.

In [27]:
```
on_Outcome in ('Success', 'Failure (in flight)'
```

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

Out[27]:

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

Boosters Carried Maximum Payload

- %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
- We select the booster column from the table and we specify a where condition where the booster pay load mass mut be equal to the maximum payload in the entire table.

```
In [30]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE      PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)
 * sqlite:///my_data1.db
Done.
```

Out[30]:

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

```
In [32]: %sql select substr(Date, 6, 2) as Month, Booster_version, Launch_Site, Landing_Outcome from SPACEXTABLE WHE
 * sqlite:///my_data1.db
Done.
```

Out[32]:

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

- %sql select substr(Date, 6, 2) as Month, Booster_version, Launch_Site, Landing_Outcome from SPACEXTABLE WHERE substr(Date, 1,4) = '2015' and Landing_Outcome like "%Failure (drone ship)%"
- We select the date with the select query as month, as Month, Booster_version, Launch_Site, Landing_Outcome and we use a where clause to specify the date as 2015 and landing outcome like failure drone ship to filter on these case scenerios

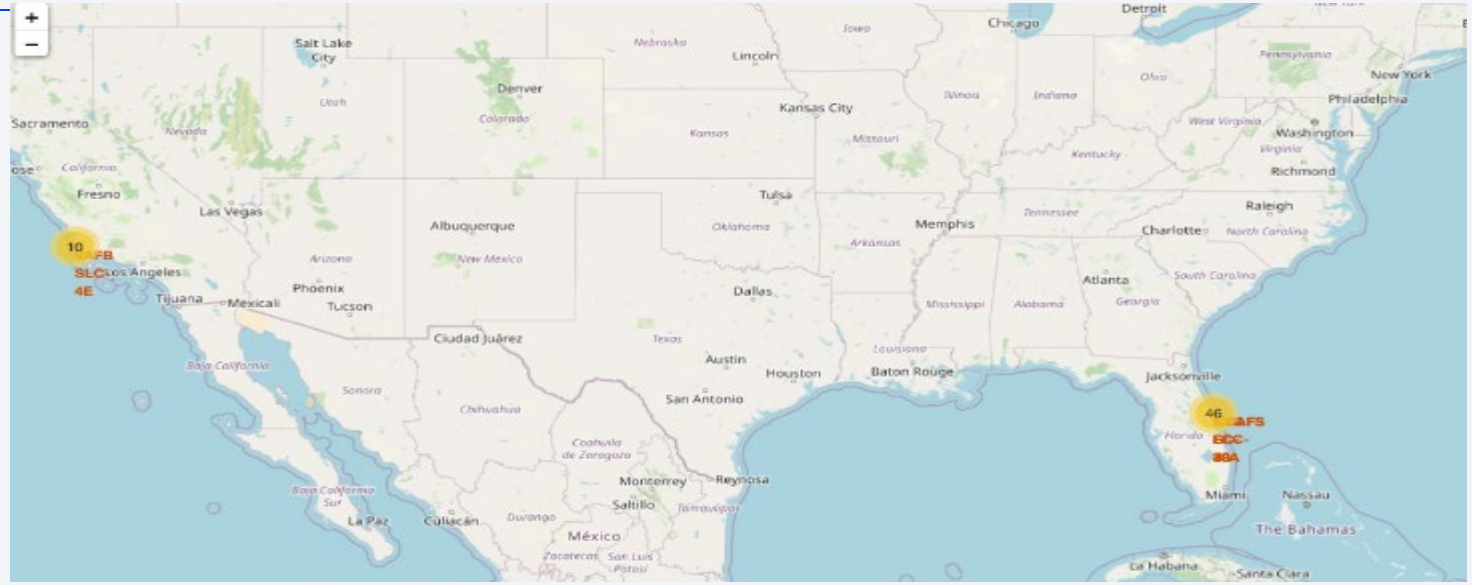
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- %sql select Landing_Outcome, count(Landing_Outcome) as Outcome_Count from SPACEXTABLE where Date Between '2010-06-04' AND '2017-03-20'


Section 3
Launch Sites
Proximities Analysis

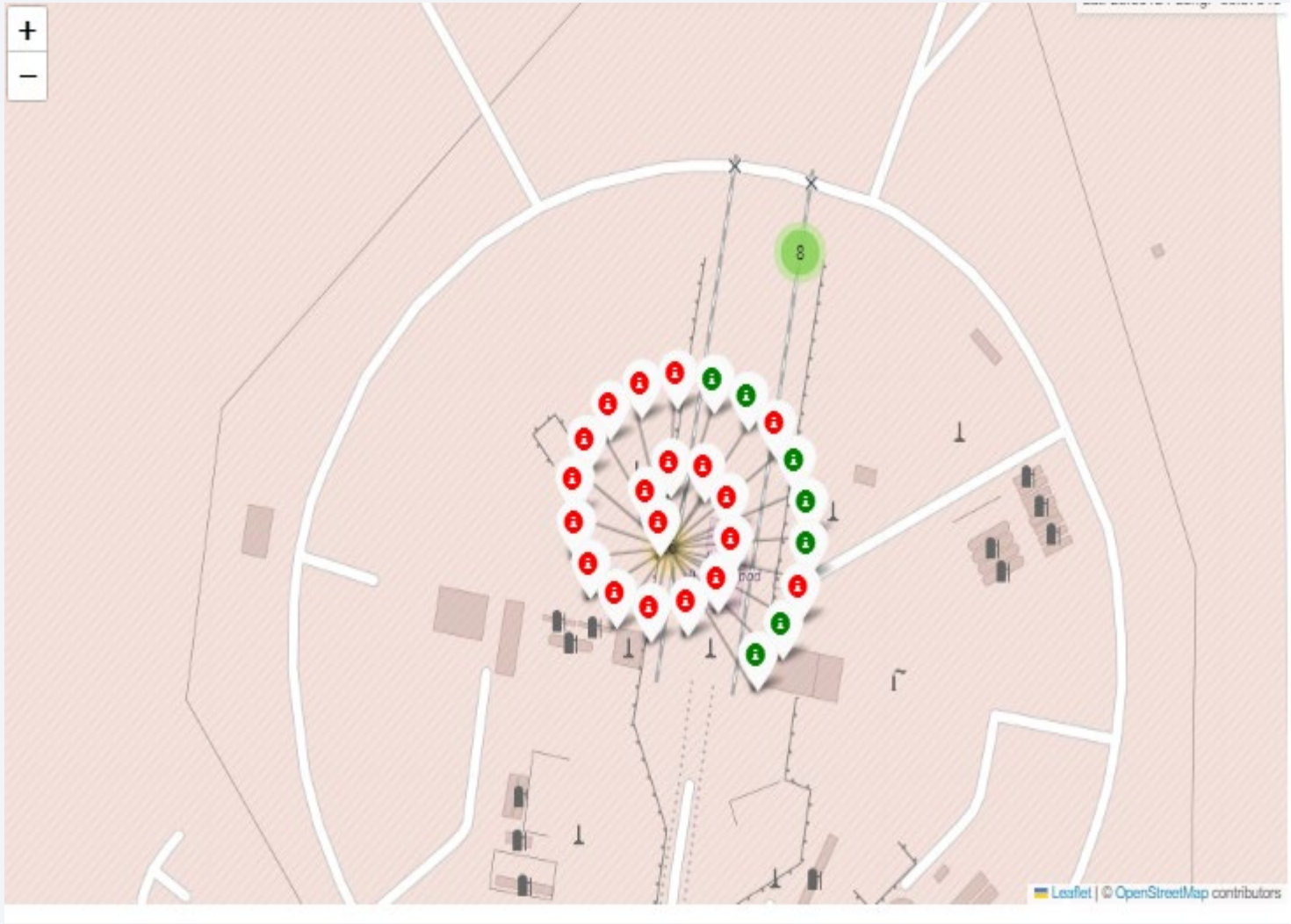
Launch Sites Locations Analysis with Folium

- We can see that the launch site have close proximities to the coast and other important infrastructures such as rail roads and high ways.



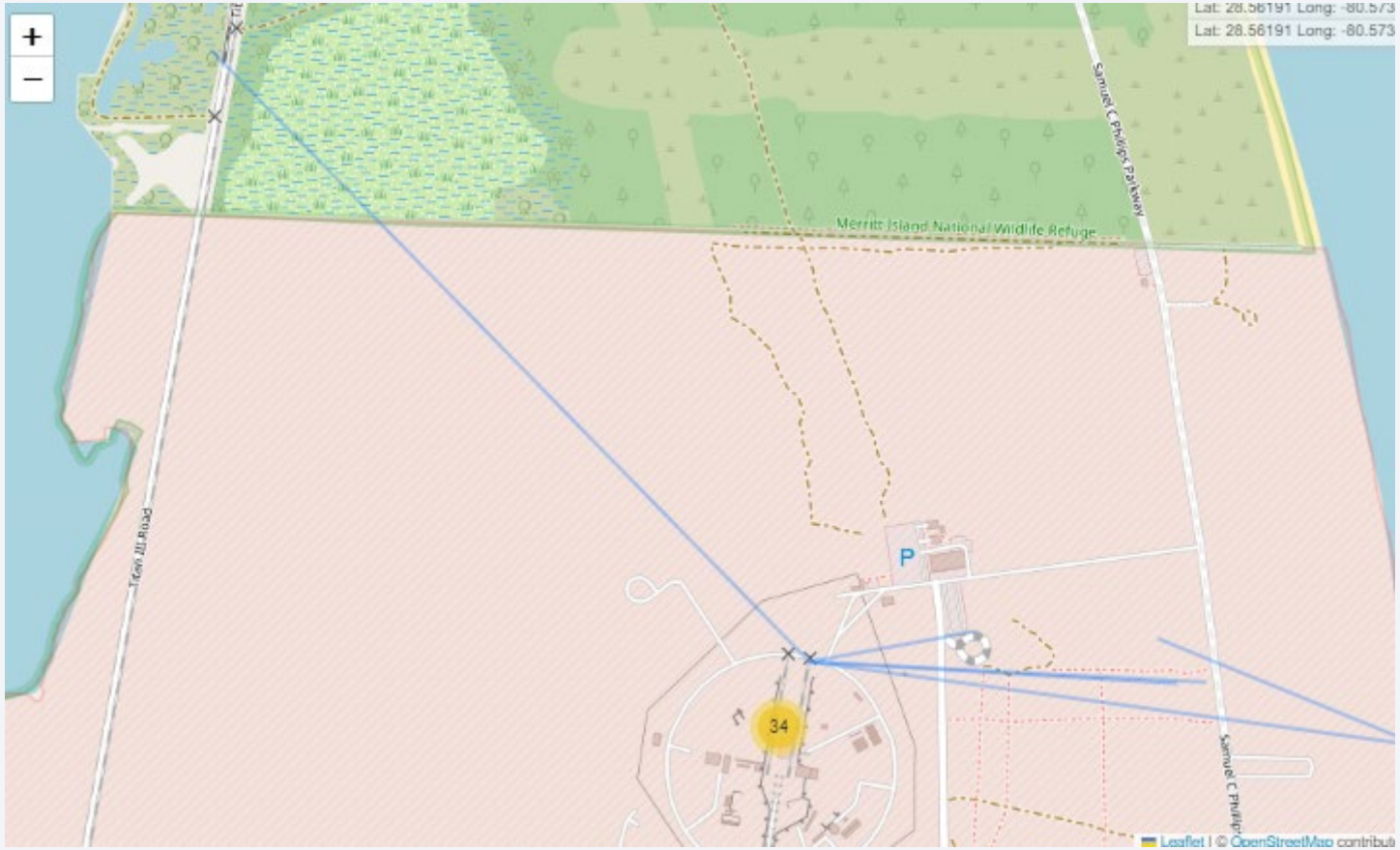
Launch Site Color Labels

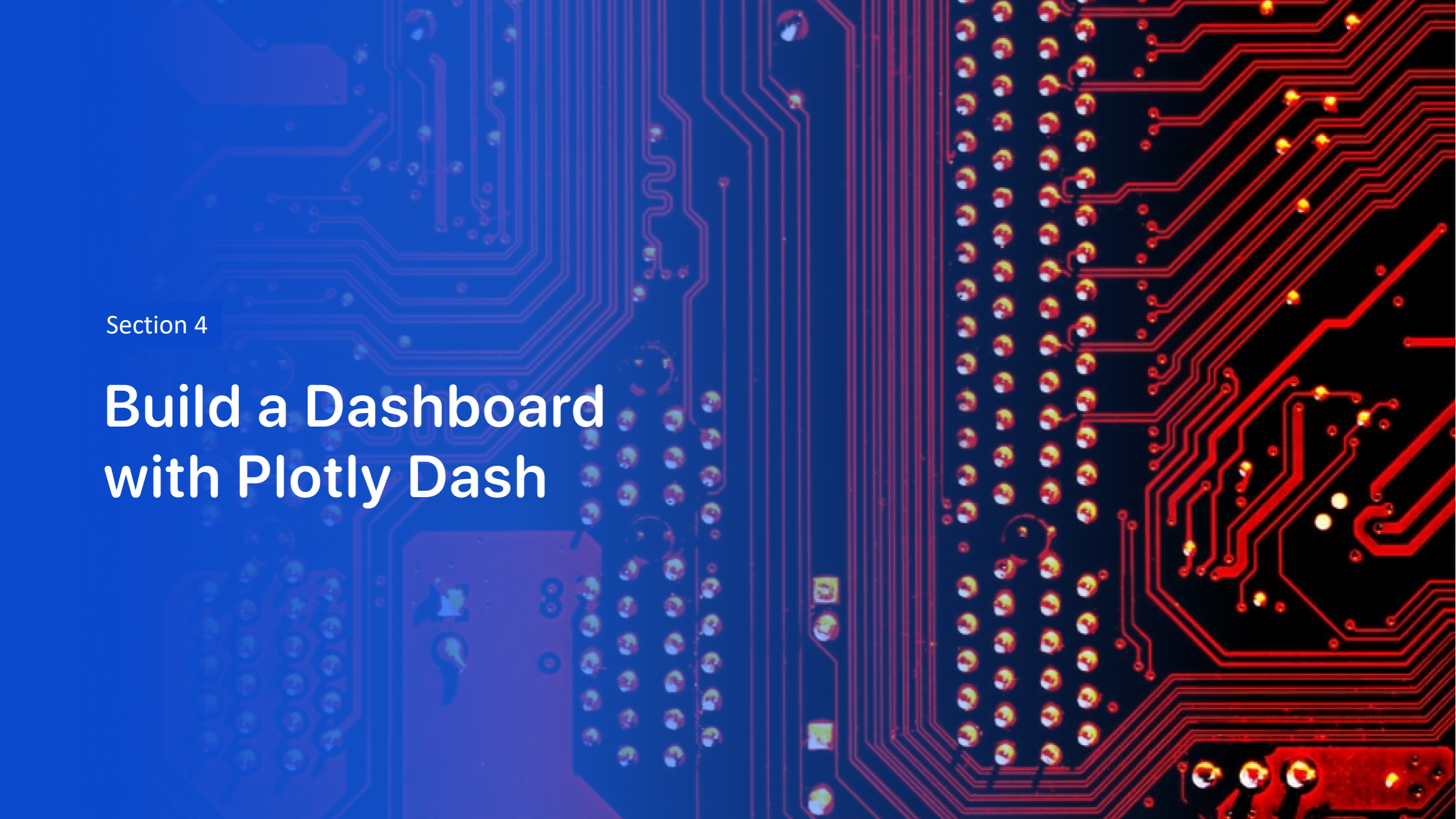
- We can see the color labels on each site that indicates the success rate from each launch site.



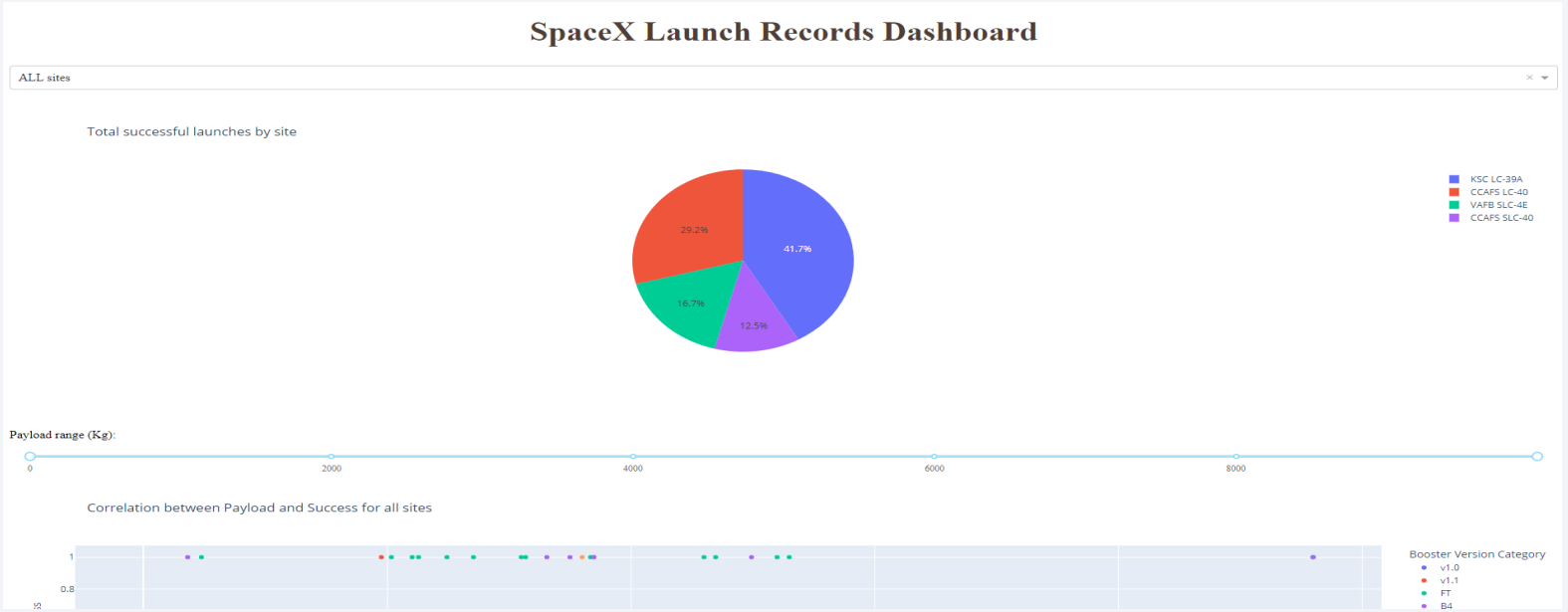
Launch Site Proximities

- WE see that the railway is 1.2km away from the launch site, the coast is 0.57km away and the highway is 0.3km away from the launch site



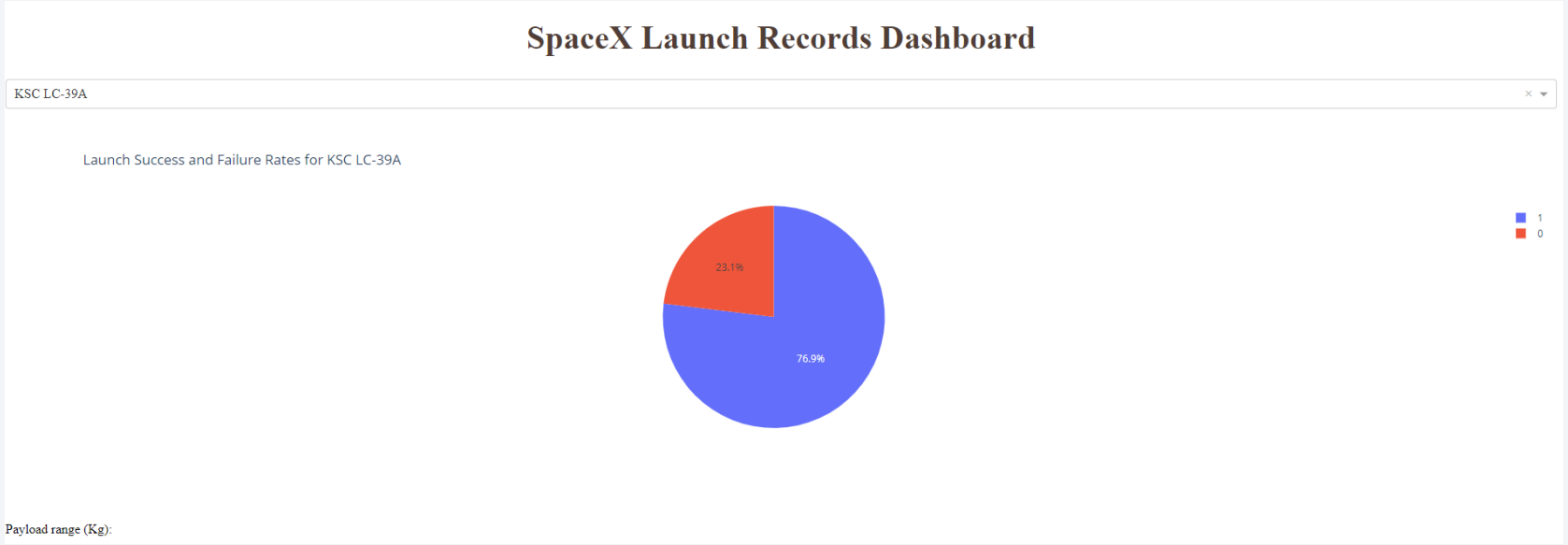

Section 4
Build a Dashboard with Plotly Dash

SpaceX Launch Record Dashboard



Here we can see the Launch site with the highest success margin is KLC LC 39A, followed by the CCAFS LC 40. However, the Other sites have a greater number of launches than the KLC.

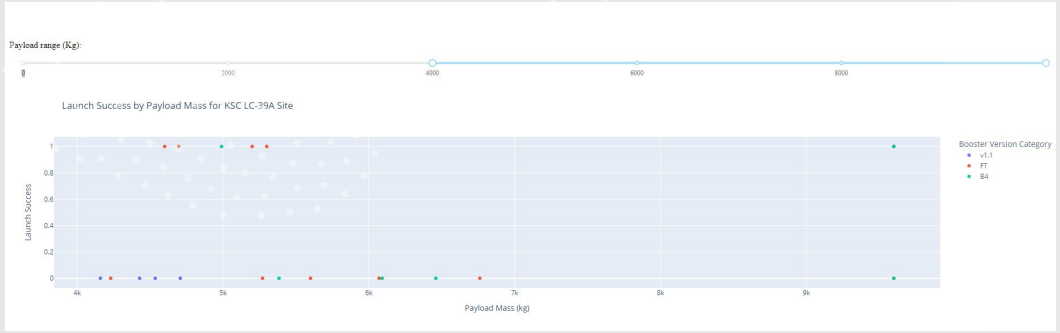
KSC LC-39A Success Statistics

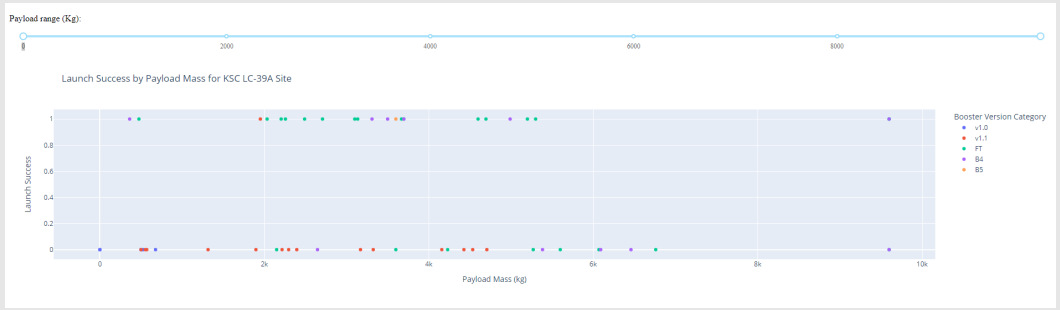


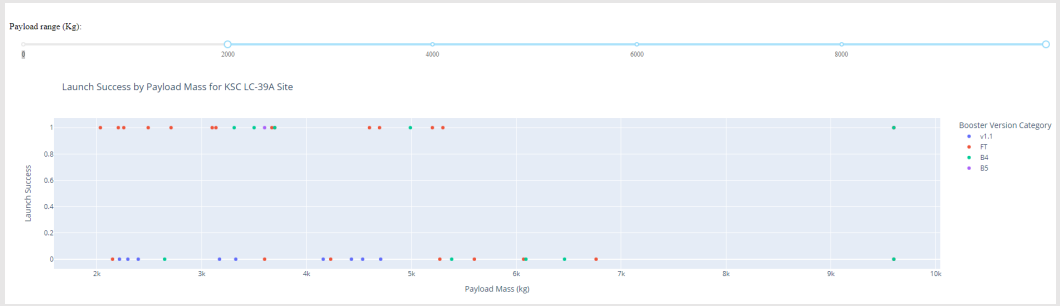
- Here we can see the distribution of the successful and unsuccessful launches. This is the statistic for the most successful launch site.

Payload Range vs Launch Success Chart

- We can find that the most successful booster across the payload mass from 0-6000kg is the FT booster and the B4 and B5 booster seems to be the preferred choice for lifting heavier payloads.
- V1.0 and v1.1 have less successful launches across all sites
- The most successful payload range seems to be between 0 and 6000kg





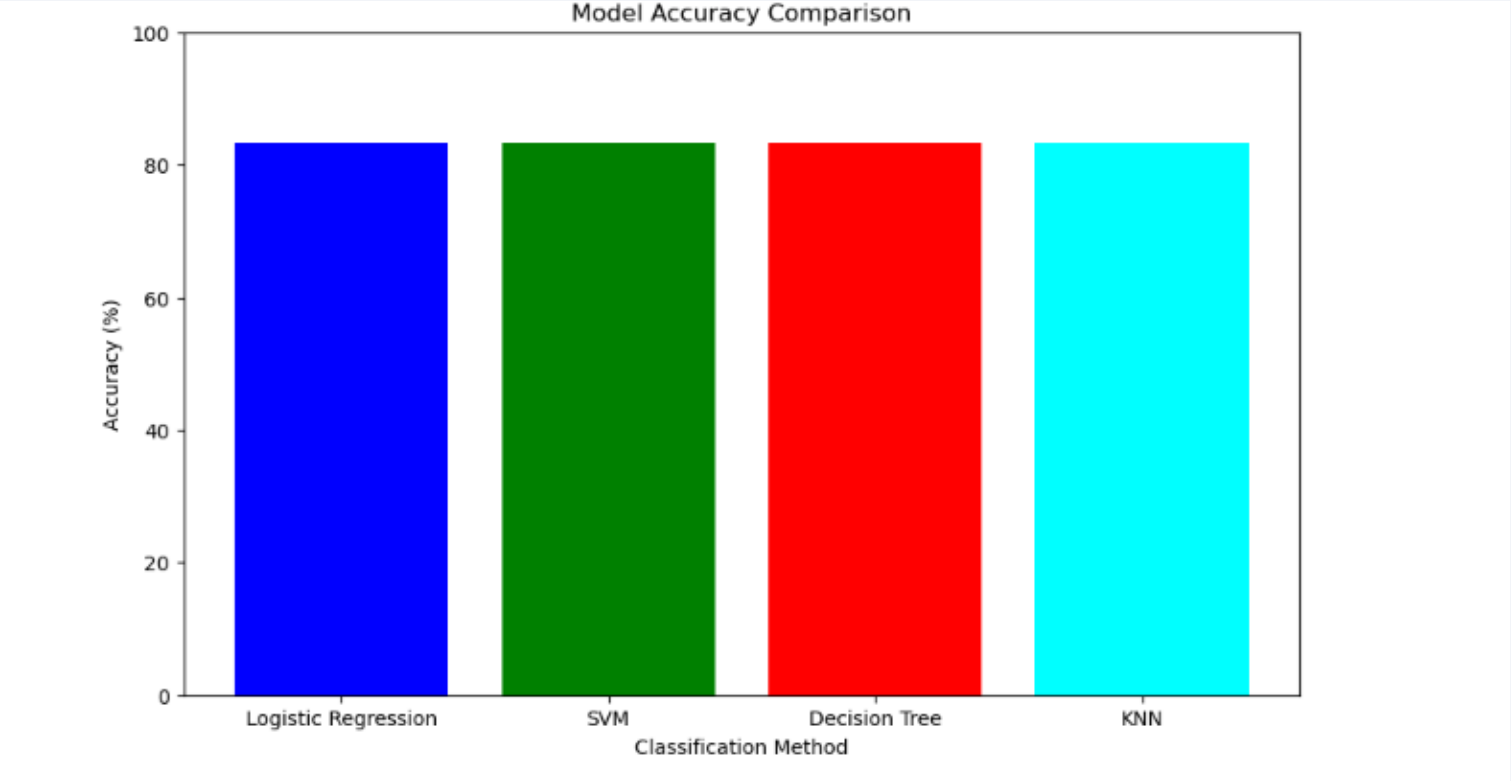


Section 5

Predictive Analysis (Classification)

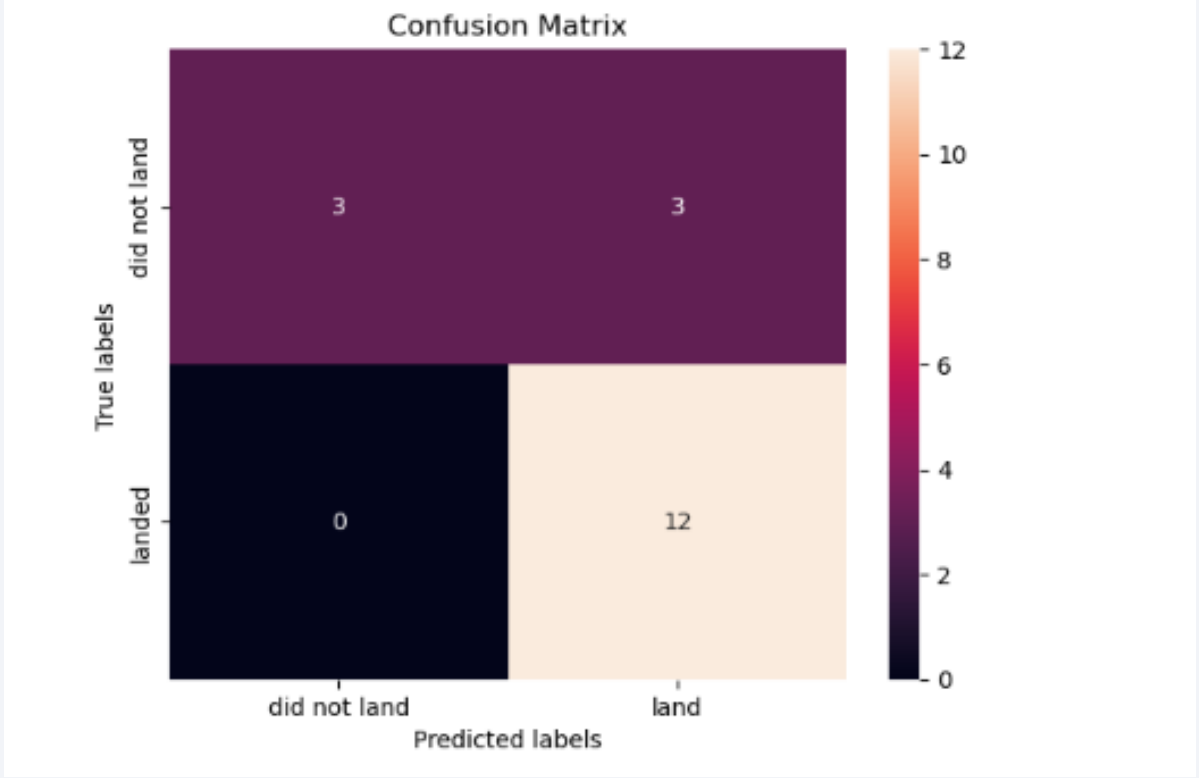
Classification Accuracy

- All classification methods had the same test accuracy of 83.4%



Confusion Matrix

- The models seem to be better at predicting successful landings better than unsuccessful ones.
- The number of misclassifications in the false positive section is 3. This is where the algorithm predicted that the booster landed when it did not land.
- There are 0 true negative misclassifications
- The diagonal from bottom-right to top-left shows the correctly predicted instances, while the off-diagonal shows the incorrectly predicted ones.



Conclusions

- The confusion matrix indicates that the ML algorithms have good performances in predicting successful landings (true positives) but may struggle with predicting unsuccessful landings given the false positives.
- The success rate graph shows improvement over time, which could be attributed to advancements in technology, accumulated experience, and refined processes.
- Unsuccessful launches do not display a clear pattern in relation to payload mass, indicating that factors other than payload mass could be influencing launch outcomes.
- The mapping analysis indicates that launch sites are generally chosen for their proximity to the coast, which is beneficial for launch safety and logistics.
- Understanding the success rate based on various factors like payload mass and orbit type can help in predicting the costs of launches and aid companies in making informed bids against competitors like SpaceX.
- The overall study underscores the value of using data-driven insights to improve launch success rates and operational efficiencies.

Appendix

Thank you!