IBM Developer
SKILLS NETWORK

IBM Data Science Capstone Project

Samuel Akuffo

8th July, 2024

Outline

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

Executive Summary

- Summary of methodologies
  - ➢ Data Collection Using an API.
  - ➢ Data Collection with Web Scraping.
  - ➢ Exploratory Data Analysis Using SQL
  - ➢ Further Exploratory Data Analysis With Visualization.
  - ➢ Interactive Visual Analysis With Folium Library.
  - ➢ Interactive Dashboard with Plotly Dash.
  - ➢ Predictive Analysis With Machine Learning.

Executive Summary

- Summary of all results
    - There were several key findings from all the analysis conducted such as:
        - ➢ Exploratory Data Analysis.
            - ➢ Insights: The EDA revealed critical patterns and trends within the data, such as how the number of flights affected whether the first stage can be re-used and also how the payload mass impacts whether the first stage will return or not.
            - ➢ Statistical Findings: Key statistical metrics were identified, such as averages, maximum, minimum, total and ranges.
            - ➢ Data Distributions: The data distributions for various features were analyzed, showing that features such as the flight number, the orbits the launches were made into and the payload mass had significant impact on other features.

Executive Summary

- Summary of all results

  - ➢Screenshots of Interactive Analytics.

    - ➢Visualizations: Interactive visualizations were created to better understand the data. Screenshots illustrate launch sites and their rates of success, how the number of flights affect the success rates, time series trends showing the average success rates.

    - ➢Tool Utilized: Tools like Seaborn, Matplotlib, Folium and Plotly Dash were used to generate interactive maps and dashboards, providing dynamic insights into the data.

Executive Summary

- Summary of all results
  - Predictive Analytics Results.
    - Models Used:
      - K-Nearest Neighbors (KNN).
      - Support Vector Machines (SVM).
      - Logistic Regression.
      - Decision Tree Classifier.
    - Key Insights: The KNN, SVM, and Logistic Regression models performed similarly well, indicating strong predictive capabilities. The Decision Tree Classifier, while slightly less accurate, still provides valuable classification performance.
    - Recommendations: Future work could focus on tuning these models or exploring ensemble methods to improve overall accuracy.

Introduction

- Project background and context
  - The goal of this capstone project for SpaceY is to predict the successful landing of the Falcon 9 first stage. SpaceX, known for its cost-effective rocket launches at $62 million compared to other providers' $165 million, achieves significant savings by reusing the first stage. Accurate predictions of the first stage landing can help determine the launch cost and provide valuable insights for SpaceY to compete with SpaceX in rocket launches. This project provides an overview of the problem and the necessary tools to complete the analysis.

Introduction

- Problems to be understood:
  - How do features such as payload mass, launch site, number of flights, and orbits affect the successful landing of the first stage?
  - Does the rate of successful landings increase over the years?
  - What is the price of each launch?
  - Will SpaceX reuse the first stage?
  - What is the best algorithm that can be used for binary classification in this case?
  - Is there a relationship between launch sites and success rates?

Section 1

Methodology

Executive Summary

- Data collection methodology:
    - Data was collected using the SpaceX API through Python.
    - Further data was collected by web scraping a Wikipedia page titled: List of Falcon 9 and Falcon Heavy launches
- Perform data wrangling
    - Data Analysis
    - Dealing with missing values.
    - Assigning new data types
    - Binary Encoding

Methodology

Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash

Methodology

Executive Summary

- Perform predictive analysis using classification models
  - Building Models:
    - Models were built using the Scikit-Learn Library in Python and various classification models including Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Decision Trees were implemented.
  - Tuning Models:
    - The models were tuned using Scikit-Learn’s preprocessing tools, train/test split function and Grid Search tool.
  - Evaluating Models:
    - The models were evaluated using using cross-validation and best score and parameters attributes from Grid Search, accuracy scores and confusion matrix.

Data Collection

- **How Data Sets Were Collected:**
  - The data sets for this project were collected using two main methods: API data retrieval and web scraping.

    1. API Data Retrieval:
       - Source: SpaceX REST API
       - Endpoints: Data was gathered from endpoints such as api.spacexdata.com/v4/launches/past, which provide detailed information about SpaceX launches, including rocket details, payloads, launch specifics, landing outcomes, etc.
       - Method: A GET request was performed using the requests library in Python to obtain the launch data. The response, in JSON format, was converted to a pandas DataFrame using the json_normalize function, which flattened the structured JSON data into a table format.

Data Collection

- **How Data Sets Were Collected:**

  2.Web Scraping:

  - Source: Wikipedia page titled - List of Falcon 9 and Falcon Heavy launches.
  - Method: The BeautifulSoup package in Python was used to scrape HTML tables containing Falcon 9 launch records. The scraped data was parsed and converted into a pandas DataFrame for further visualization and analysis.

Data Collection

- **How Data Sets Were Collected:**

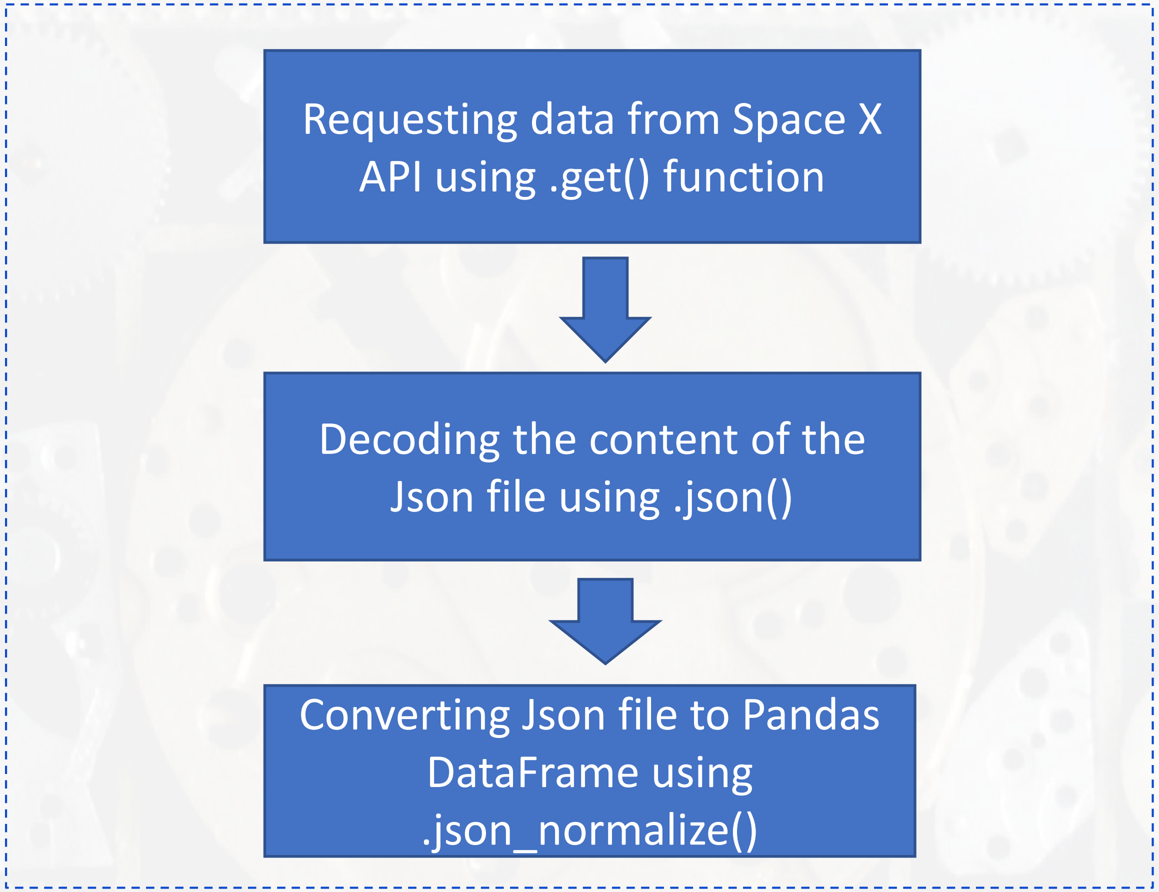
The data sets for this project were collected using two main methods: API data retrieval and web scraping.

- You need to present your data collection process use key phrases and flowcharts

Data Collection - SpaceX API

- Data was collected using the SpaceX API which provides valuable data on all aspects of rockets launched by them.
- An HTTP request was made using the GET() function.
- The response content was decoded to a Json file using the .json() function.
- The Json file was then converted to a pandas DataFrame using the json_normalize() function

- **GitHub URL:** https://github.com/KwameSA/DataScienceCapstoneProject/blob/main/1-Jupyter-Labs-SpaceX-Data-Collection-Api.ipynb



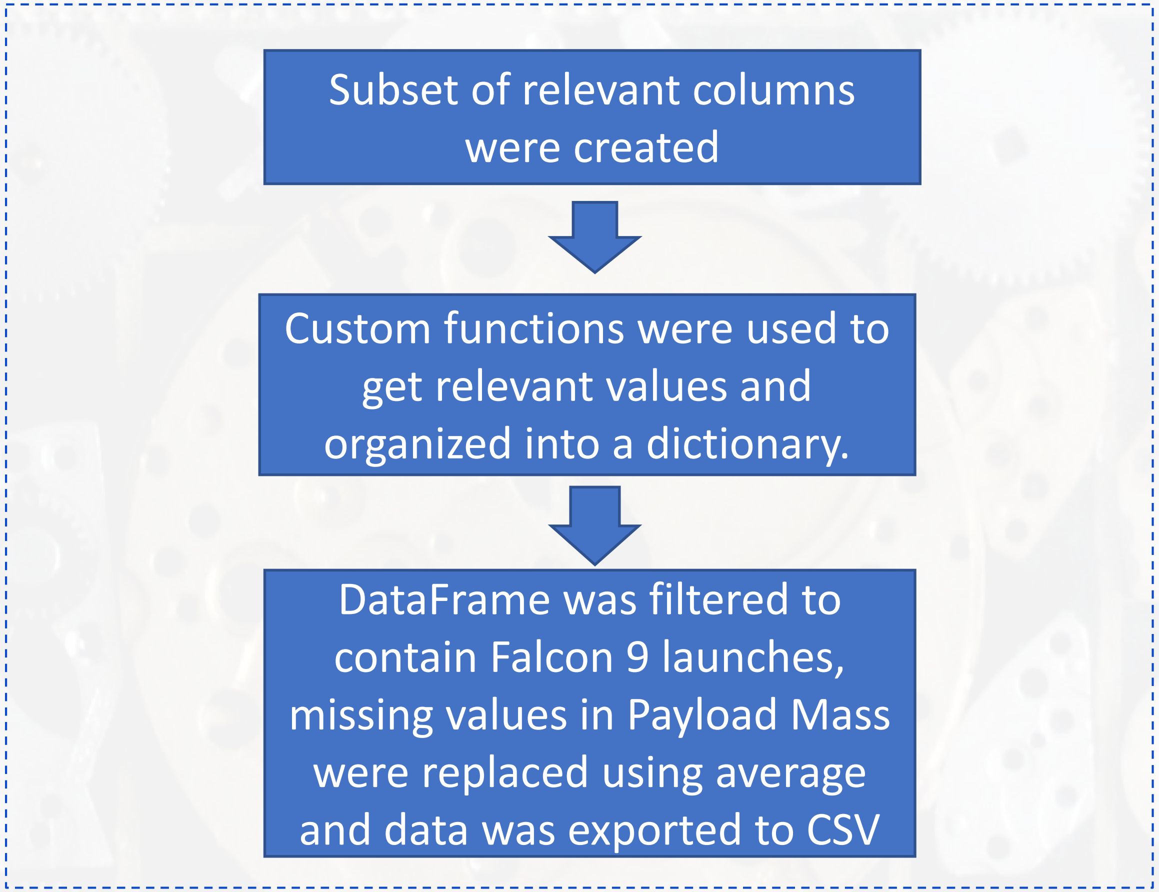
Requesting data from Space X API using .get() function

Decoding the content of the Json file using .json()

Converting Json file to Pandas DataFrame using .json_normalize()

Data Collection - SpaceX API

- A subset of relevant features containing needed information such as rocket, payloads, launchpad, cores, flight_number, and date_utc were separated from the earlier DataFrame.
- Custom functions were used to get the needed values and was organized into a dictionary.
- A DataFrame was created from the dictionary
- The DataFrame was filtered to contain only Falcon 9 launches.
- Missing values in the Payload Mass (kg) column were replaced using the calculated mean.
- Data was exported into a CSV file.
- **GitHub URL:** https://github.com/KwameSA/DataScienceCapstoneProject/blob/main/1-Jupyter-Labs-SpaceX-Data-Collection-Api.ipynb



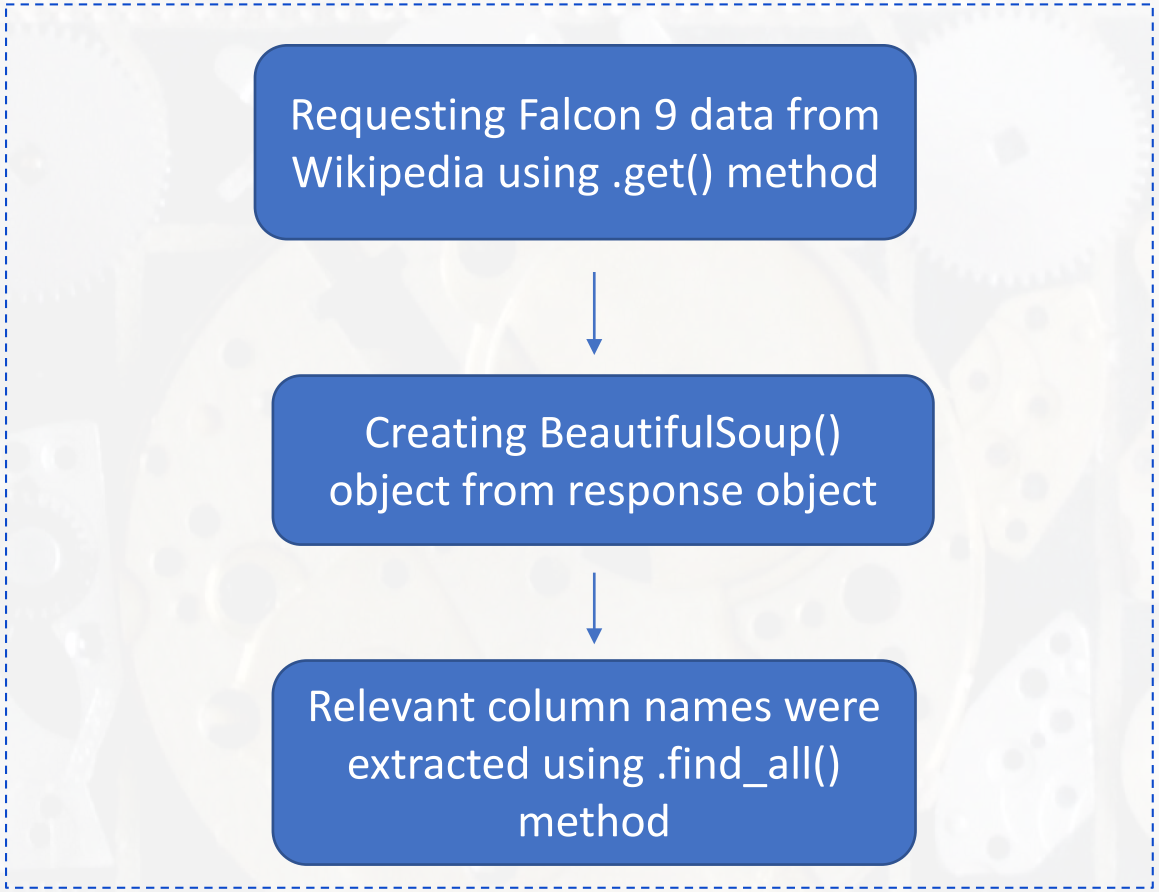
Subset of relevant columns were created

Custom functions were used to get relevant values and organized into a dictionary.

DataFrame was filtered to contain Falcon 9 launches, missing values in Payload Mass were replaced using average and data was exported to CSV

Data Collection - Web Scraping

- Data was scraped from a Wikipedia page titled - List of Falcon 9 and Falcon Heavy launches which contains historical data about Falcon 9 launches
- Helper functions were created to help process scraped data.
- An HTTP request was made using the .get() function and then a BeautifulSoup object was created from the HTML response.
- All relevant feature names were extracted from HTML header using different methods and predefined functions.
- **GitHub URL:** https://github.com/KwameSA/DataScienceCapstoneProject/blob/main/2-Jupyter-Labs-Webscraping.ipynb



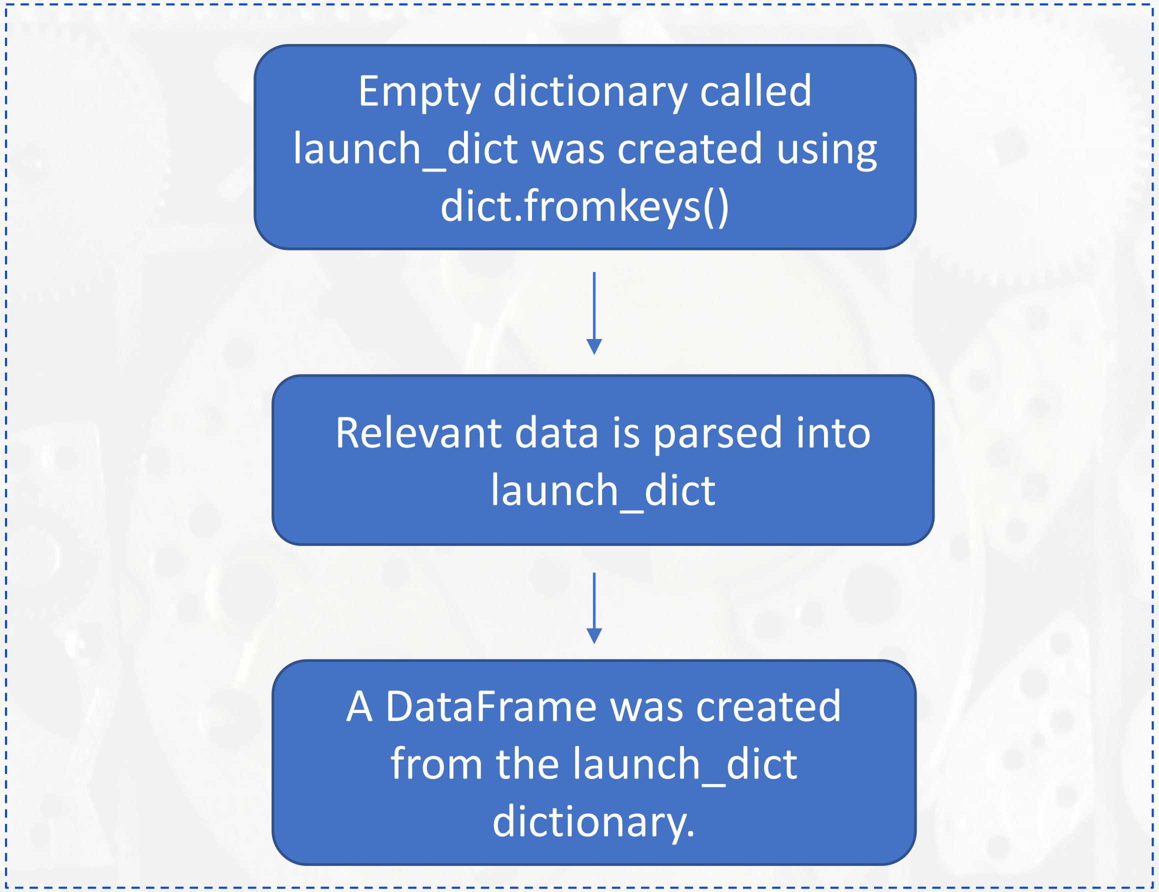
Requesting Falcon 9 data from Wikipedia using .get() method

Creating BeautifulSoup() object from response object

Relevant column names were extracted using .find_all() method

Data Collection - Web Scraping

- An empty dictionary was created with keys from the extracted column names called launch_dict..
- Data elements were parsed and filled into the launch_dict dictionary
- A DataFrame wss then crweated from the dictionary.
- **GitHub URL:** https://github.com/KwameSA/DataScienceCapstoneProject/blob/main/2-Jupyter-Labs-Webscraping.ipynb



Empty dictionary called launch_dict was created using dict.fromkeys()

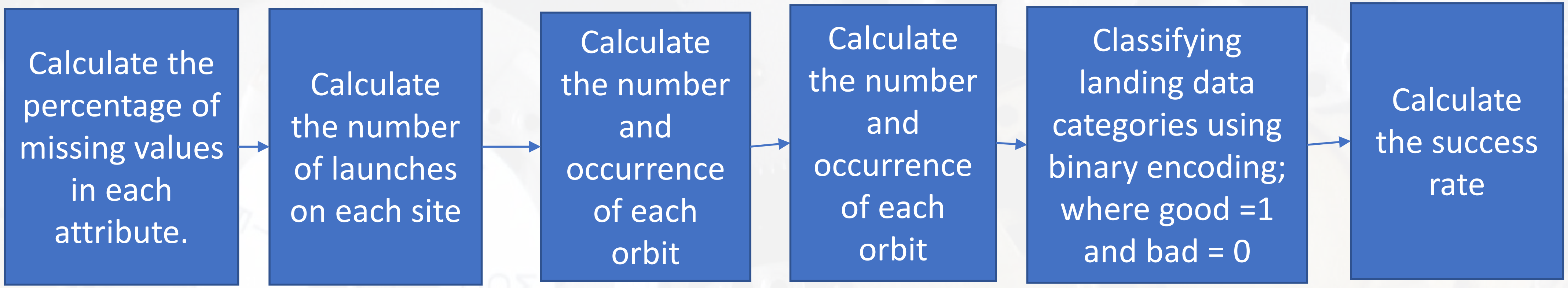
Relevant data is parsed into launch_dict

A DataFrame was created from the launch_dict dictionary.

Data Wrangling

- Exploratory Data Analysis (EDA) was conducted to find patterns in data and also help define labels to be used for supervised learning in the form of Classification.
  - What the EDA helped calculate:
    - Percentage of missing values in each attribute.
    - Number of launches on each site.
    - Number and occurrence of each orbit
    - Number and occurence of missing outcomes of the orbits
    - Average success rate.
  - Defined labels for Classification:
    - Classifying landing data categories as good or bad into a list where a bad outcome is represented by 0 and a good outcome is represented by 1.
    - List is cast into the landing outcome column of our DataFrame.

Data Wrangling



Calculate the percentage of missing values in each attribute.

Calculate the number of launches on each site

Calculate the number and occurrence of each orbit

Calculate the number and occurrence of each orbit

Classifying landing data categories using binary encoding; where good =1 and bad = 0

Calculate the success rate

- GitHub URL: https://github.com/KwameSA/DataScienceCapstoneProject/blob/main/3-Labs-Jupyter-Spacex-Data%20Wrangling.ipynb

EDA with Data Visualization

- Features which where visualized:
  - We explored the data by visualizing the relationship between flight, number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- Charts used and why:
  - Scatter Plots: This type of plot shows the relationship between independent and dependent features. Helps determine best features for our models.
  - Bar Charts: This shows comparisons among discrete categories. It defines the relationship between the specific categories being compared and a measured value.
  - Line Charts: This show trends in data over time (time series).
- **GitHub URL:**
  - https://github.com/KwameSA/DataScienceCapstoneProject/blob/main/5-EDA-Data-Vizualization.ipynb

EDA with SQL

- Summary of SQL queries performed:
  - Displayed the names of the unique launch sites in the space mission.
  - Displayed 5 records where launch sites began with the string ‘CCA‘.
  - Displayed the total payload mass carried by boosters launched by NASA (CRS).
  - Displayed the average payload mass carried by booster version F9 v1.1.
  - Listed the date when the first successful landing outcome was achieved in ground pad.
  - Listed the names of the boosters which had success in drone ship and payload mass greater than 4000 but less than 6000.
  - Listed the total number of successful and failed mission outcomes

- Listed the names of the booster versions which have carried the maximum payload mass
- Listed the failed landing outcomes in drone ship, their booster versions and launch sites for the months in the year 2015.
- Ranked the count of landing outcomes between the dates 2010-06-04 and 2017-03-20 in descending order
- **GitHub URL:** https://github.com/KwameSA/DataScienceCapstoneProject/blob/main/4-Jupyter-Labs-EDA-SQL-Coursera_Sqllite.ipynb

Build an Interactive Map with Folium

- **Map objects created and added to the folium map:**
  - Markers, circles, lines and marker clusters were used with Folium Maps
- **Why those objects were added:**
  - Markers were used to indicate the launch sites.
  - Circles were used to indicate highlighted areas around specific coordinates, like NASA Johnson Space Center.
  - Marker clusters were used to indicate groups of events such as successful and failed launches in each coordinate, and this helped identify the success rates of the launch sites.
  - Lines were used to indicate distances between the launch sites and their proximities such as railways, cities, coastlines and highways .
- GitHub URL:
  - https://github.com/KwameSA/DataScienceCapstoneProject/blob/main/6-Lab_Jupyter_Launch_Site_Location.ipynb

Build a Dashboard with Plotly Dash

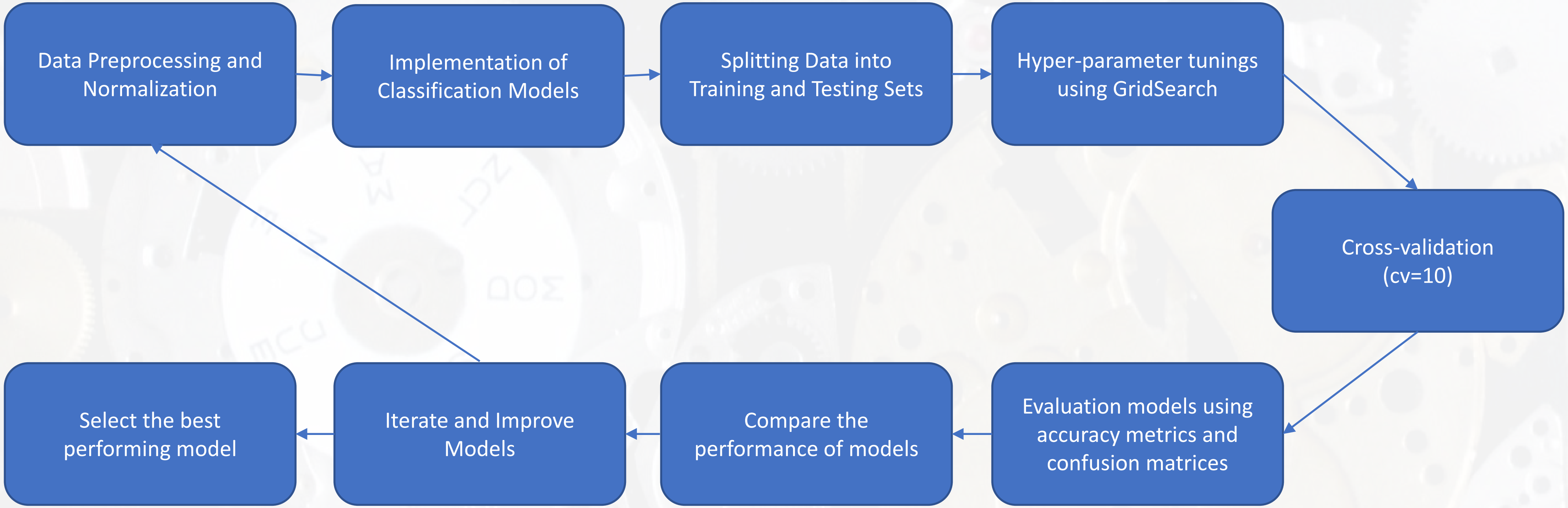
- **Dashboard:**
  - An interactive dashboard was built using Plotly dash.
  - Pie Charts and Scatter Plots were used to make relevant deductions from our Falcon 9 rocket launch data.
- **Purpose of plots and interactions:**
  - The pie charts were plotted to show the total launches of all the launch sites and those of each specific site.
  - The scatter plots were plotted to show the relationship with Outcome and Payload Mass (Kg) for the different booster versions.
- **GitHub URL:**
  - https://github.com/KwameSA/DataScienceCapstoneProject/blob/main/7-spacex_dash_app.py

Predictive Analysis (Classification)

- **Summary of Model Development:**
  - **Model Building:**
    - Implemented classification models including Logistic Regression, KNN, SVM and Decision Tree.
    - Split data into training and testing sets
  - **Hyper-parameter Tuning:**
    - Used GridSearch for optimization
    - Configured cross-validation (cv=10)
  - **Model Evaluation:**
    - Evaluated model by using accuracy scores and confusion matrices.
    - Compared performance metrics of the different models.
  - **Model Improvement:**
    - Iteratively improved models based on metrics
    - Re-evaluated after tuning
  - **Best Model Selection:**
    - Selected best model based on highest accuracy and cross-validation results
- **GitHub URL:**
  - https://github.com/KwameSA/DataScienceCapstoneProject/blob/main/8-SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Predictive Analysis (Classification)

- **Model Development Process:**



Data Preprocessing and Normalization

Implementation of Classification Models

Splitting Data into Training and Testing Sets

Hyper-parameter tunings using GridSearch

Cross-validation (cv=10)

Evaluation models using accuracy metrics and confusion matrices

Compare the performance of models

Iterate and Improve Models

Select the best performing model

- GitHub URL:https://github.com/KwameSA/DataScienceCapstoneProject/blob/main/8-SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Exploratory data analysis results

  - Space X uses 4 different launch sites.
  - The total payload mass carried by boosters launched by NASA (CRS) is 45596 kg.
  - The average payload of F9 v1.1 booster is 2,928.4 kg;
  - The first success landing outcome happened on 22nd December 2015.
  - Falcon 9 booster versions; F9 FT B1022, F9 FT B1026, F9 FT B1021.2 and F9 FT B1031.2 were successful at landing in drone ships and had payload masses greater than 4000 but less than 6000 kg.
  - Almost 100% of mission outcomes were successful with only one failure out of 101 launches.
  - Only 12 booster versions had carried the maximum payload mass.
  - Two booster versions failed at landing in drone ships in 2015: F9 v1.1 B1012 and F9 v1.1 B1015;
  - The launch success rate since 2013 kept increasing till 2020which is the range covered in our project.
  - The orbits ES-L1, GEO, HEO and SSO had the highest success rates.

Results

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

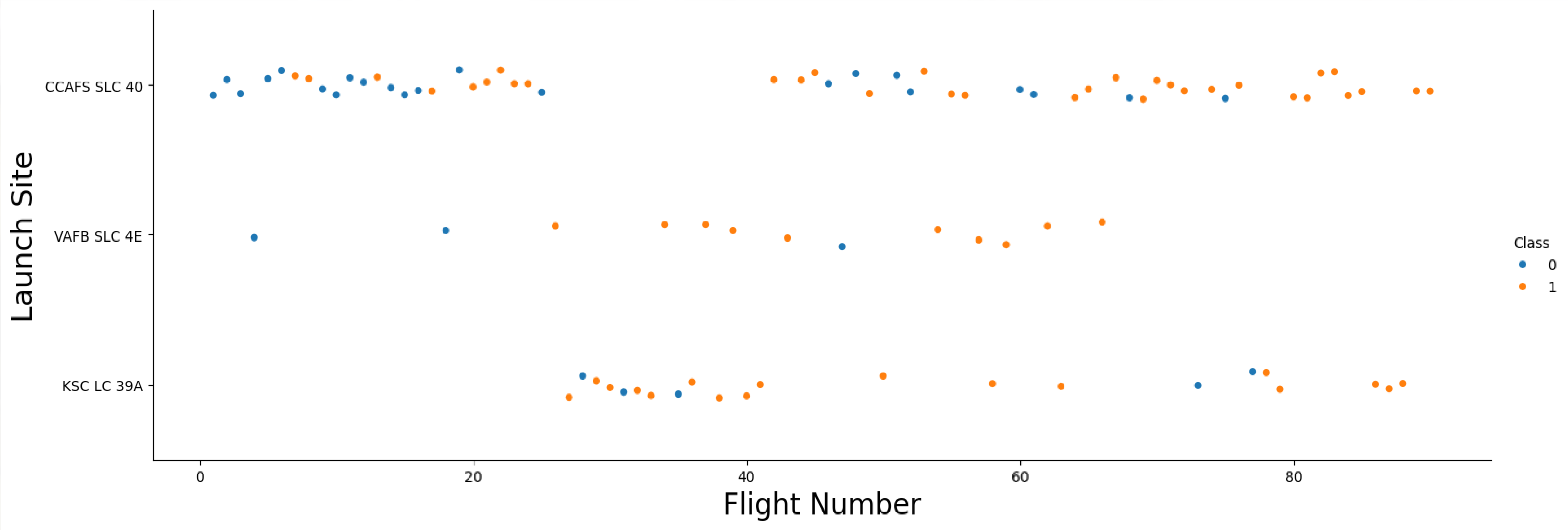
Results

- Predictive analysis results
  - The Decision Tree Classifier was the best model.
  - All the models used, K-Nearest Neighbor, Support Vector Machines, Logistic Regression and Decision Tree Classifier all performed really well on the test data after training.
  - The least accuracy on the test data was 83.33% (KNN, SVM and Logistic Regression) and the highest was 89%(Decision Tree Classifier).
  - Confusion Matrices were used to represent how well the models were able to distinguish between the different classes

Section 2

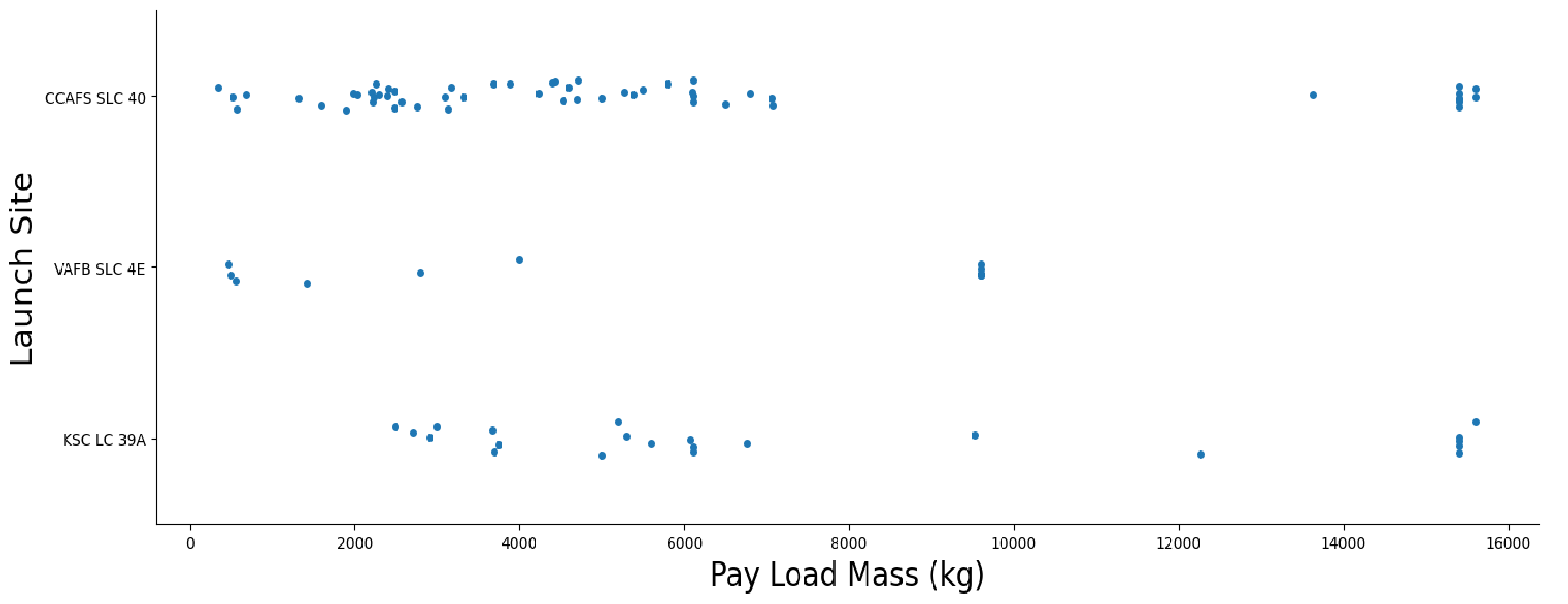
Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site:
  - The success of the flights increased across the launch sites as the number of flights increased.
  - Most of the launches were made at the CCAFA SLC 40 launch site.



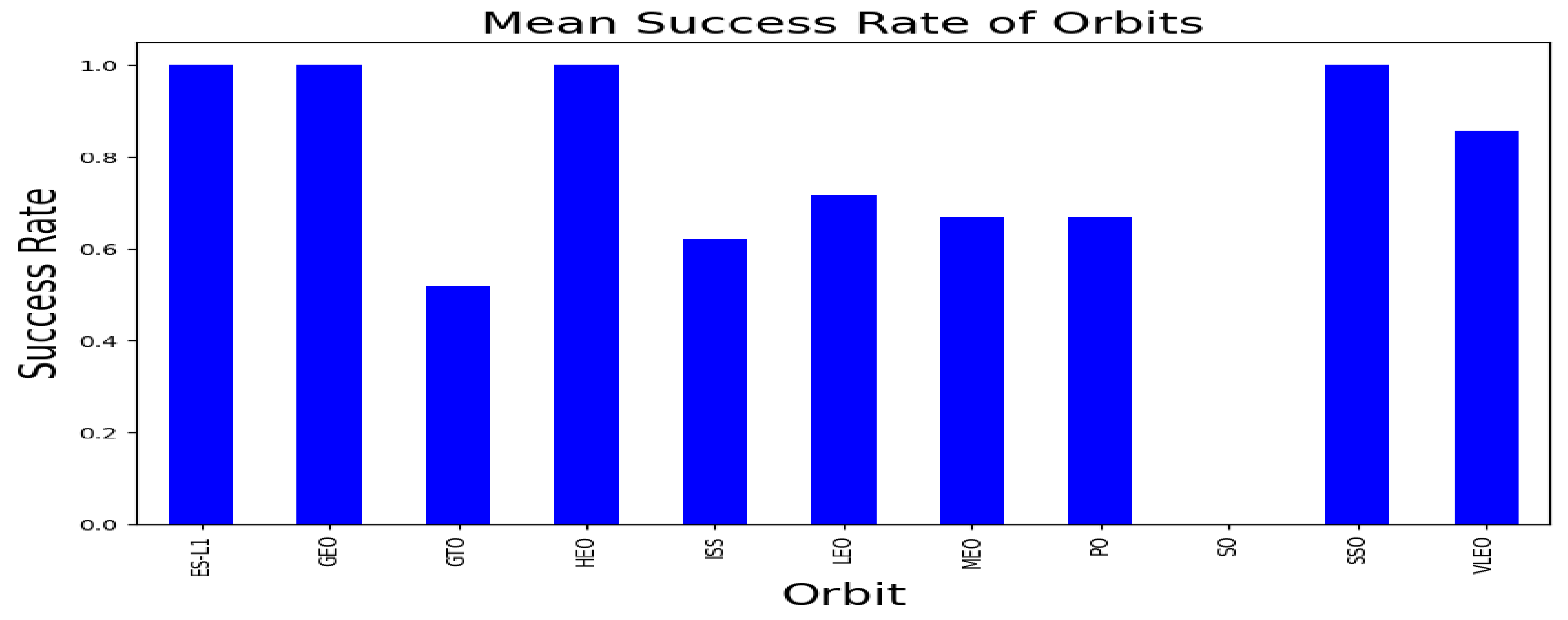
Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site:
  - Both CCAFA SLC 40 and KSC LC 33A have rockets launched for heavy payload masses (greater than 10000).
  - VAFB-SLC 4E launch site has no rockets launched for heavy payload masses.



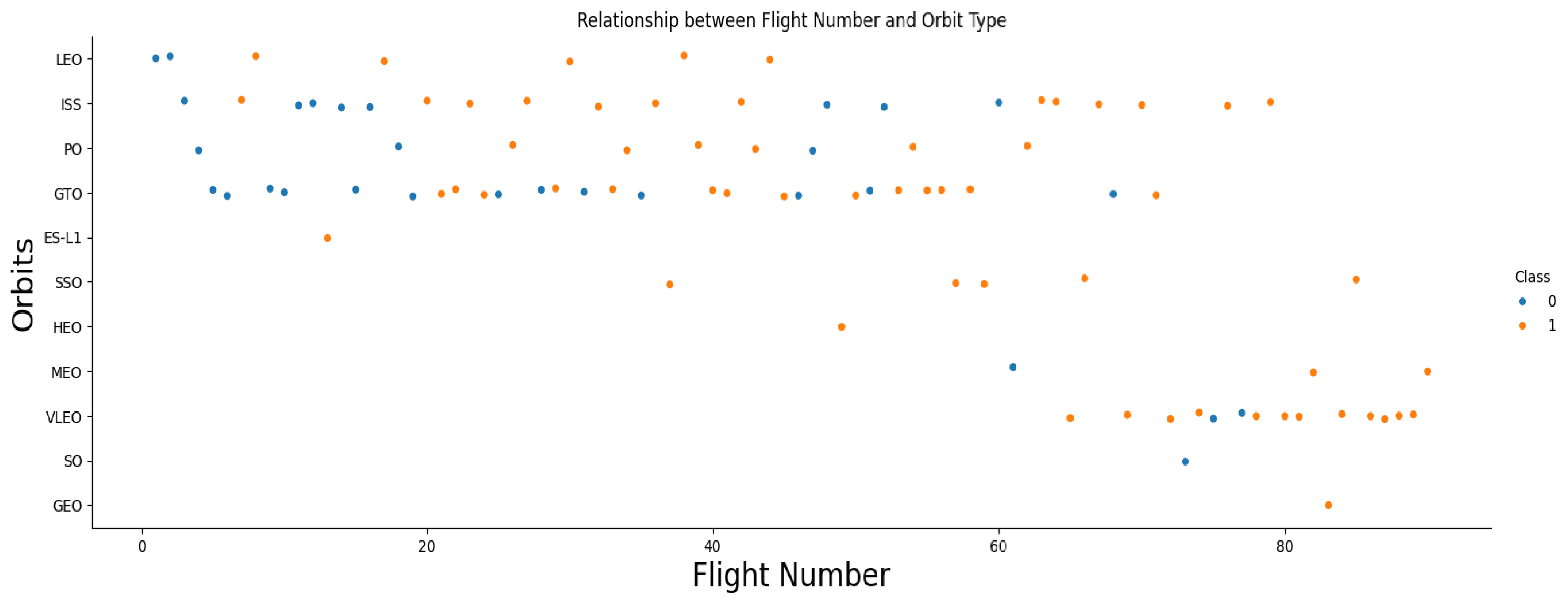
Success Rate vs. Orbit Type

- Bar chart showing the success rate of each orbit type:
  - The orbits ES-L1, GEO, HEO and SSO had the highest success rates.
  - The GTO and SO orbits had the lowest success rate.



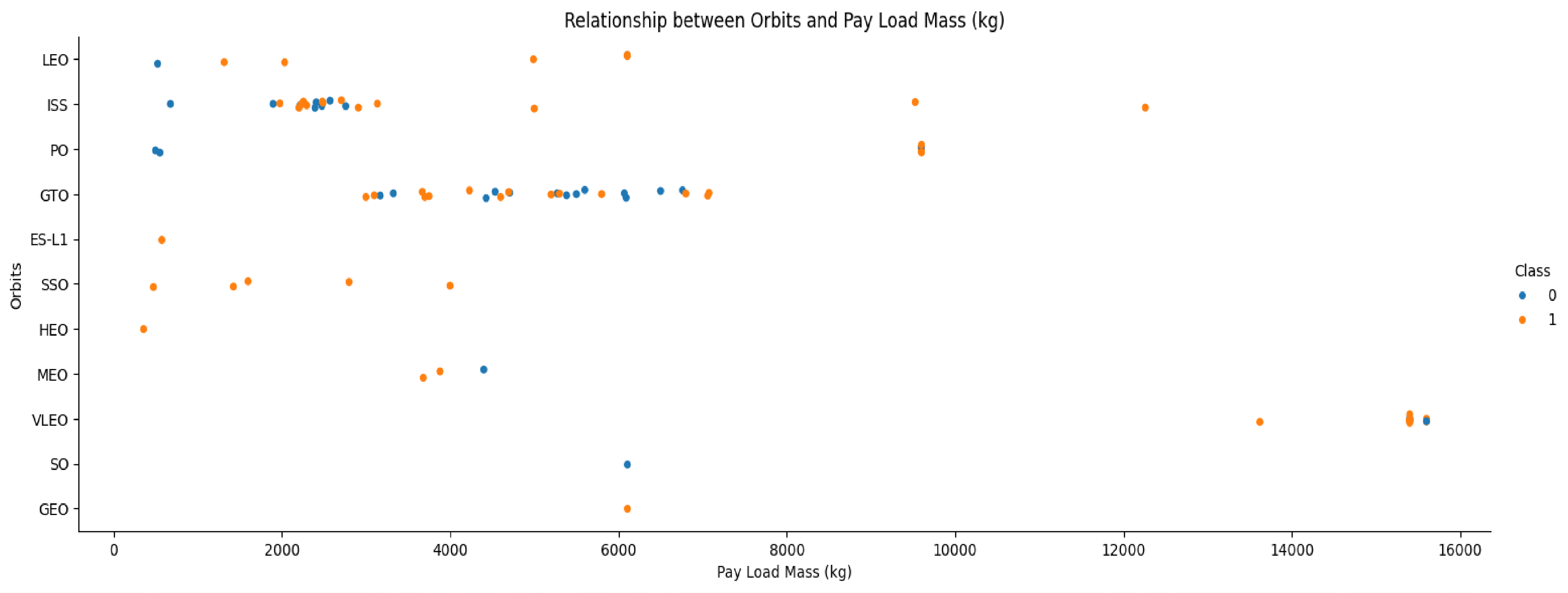
Flight Number vs. Orbit Type

- Scatter plot of Flight number vs. Orbit type:
  - In the LEO orbit the success appears to be related to the number of flights.
  - There seems to be no relationship to the number of flights when in GTO orbit.



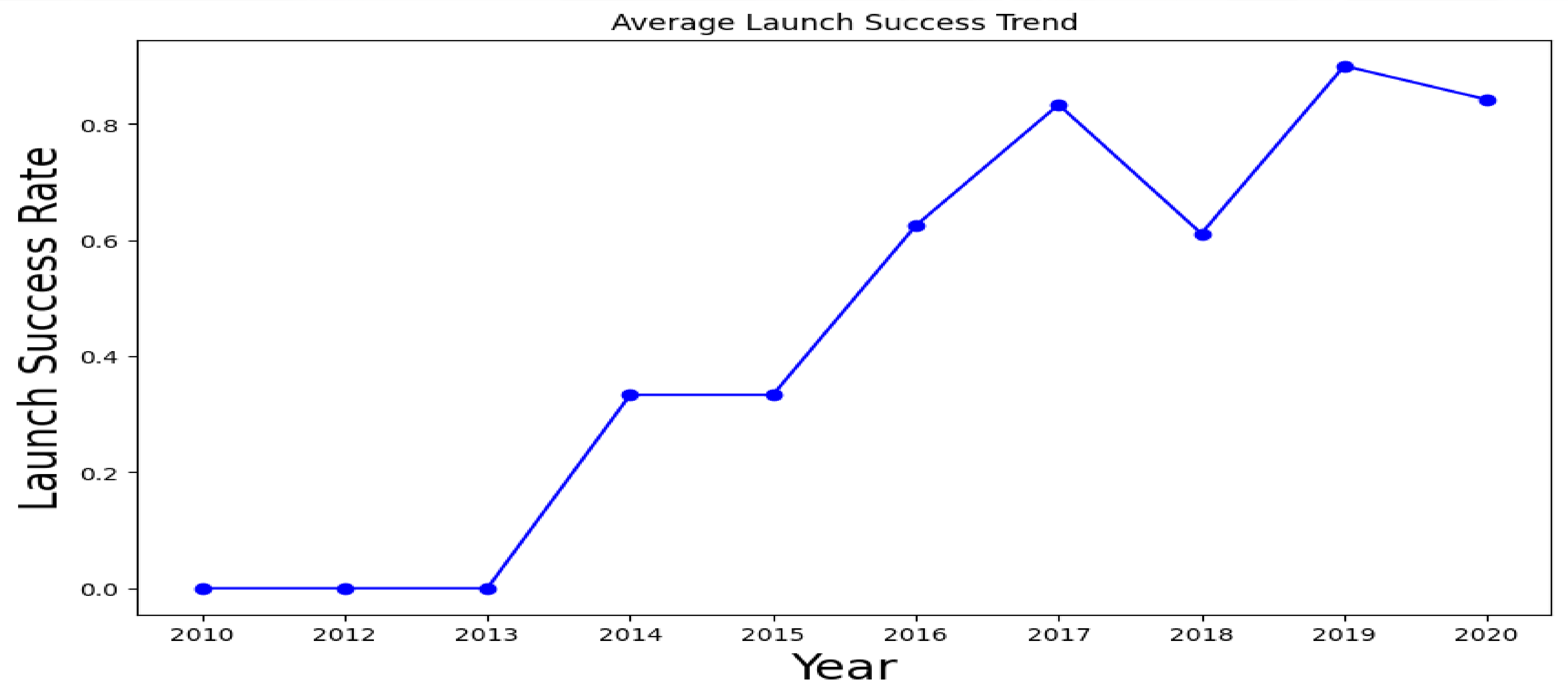
Payload vs. Orbit Type

- Scatter plot of payload vs. orbit type:
  - The successful landing rate are more for Polar, LEO and ISS when the payloads are heavy.
  - Both positive and negative landing rate cannot be clearly distinguished for the GTO orbit.



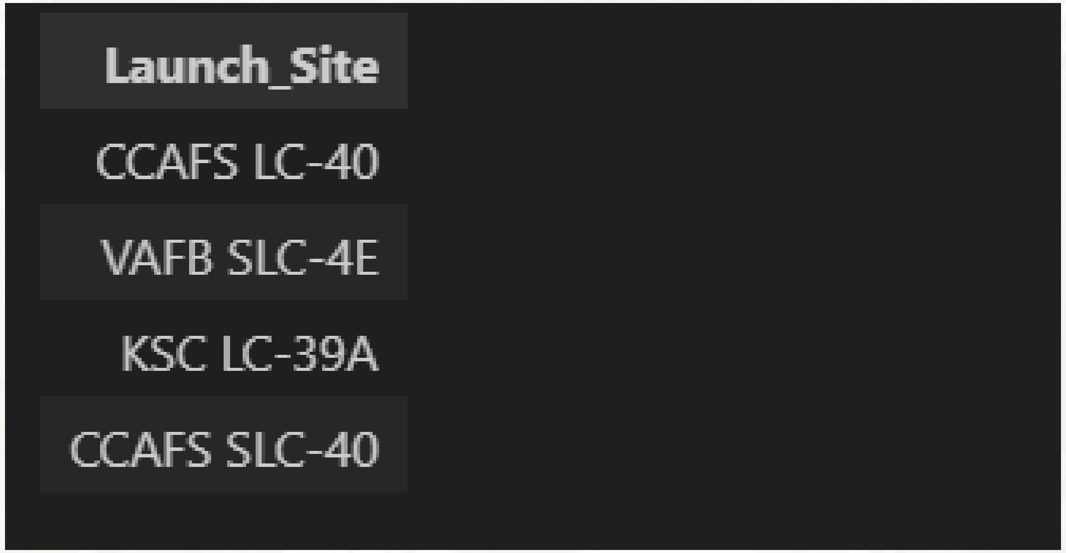
Launch Success Yearly Trend

- Line chart of yearly average success rate:
  - The launch success rate since 2013 kept increasing till 2020which is the range covered in our project.



All Launch Site Names

- The names of the unique launch sites:

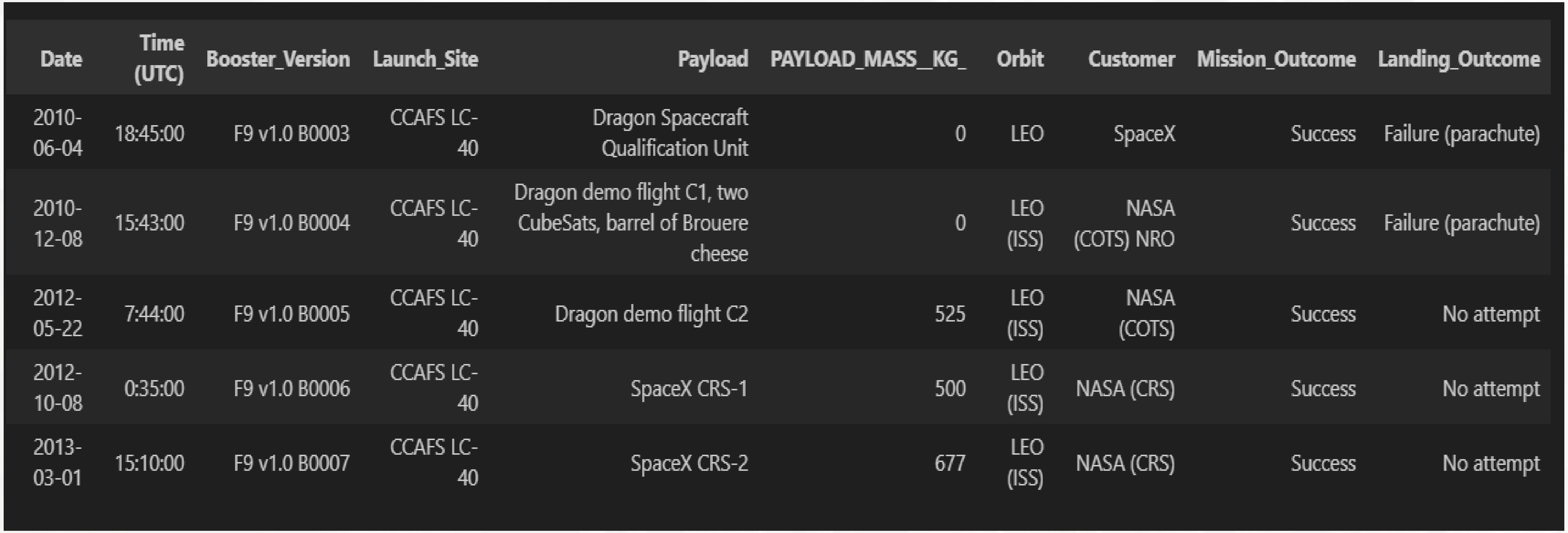


Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

- The DISTINCT() query was used to get the unique names of all the launch sites available in the data.

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`:

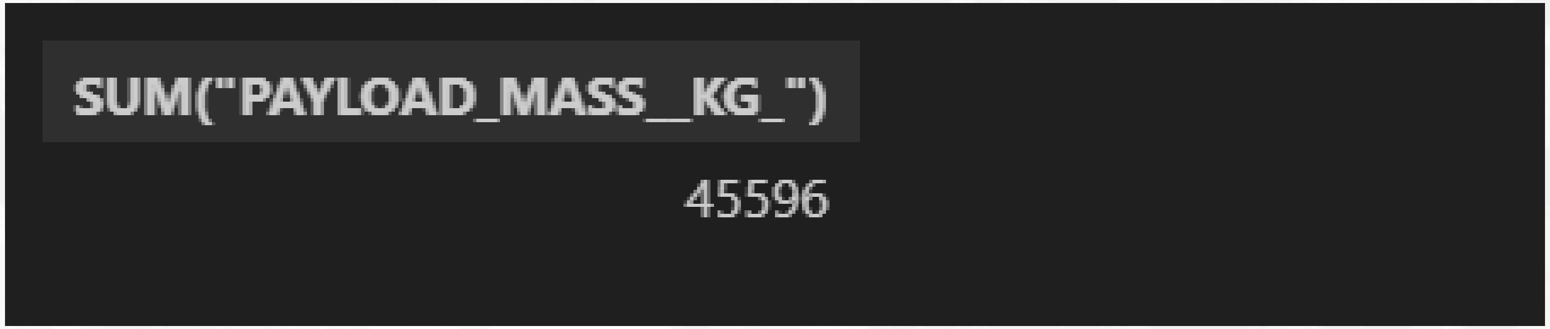


Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- The LIMIT query was used to limit the results to 5 records

Total Payload Mass

- Total payload carried by boosters from NASA:

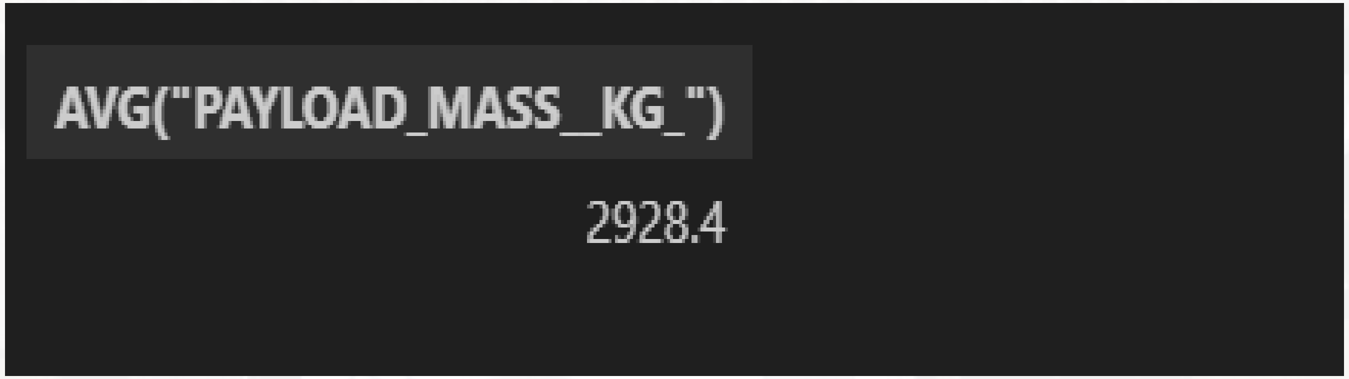


SUM("PAYLOAD_MASS__KG_")
45596

- The aggregate function SUM() was used on the payload mass (kg) column to get the total payload.

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1:

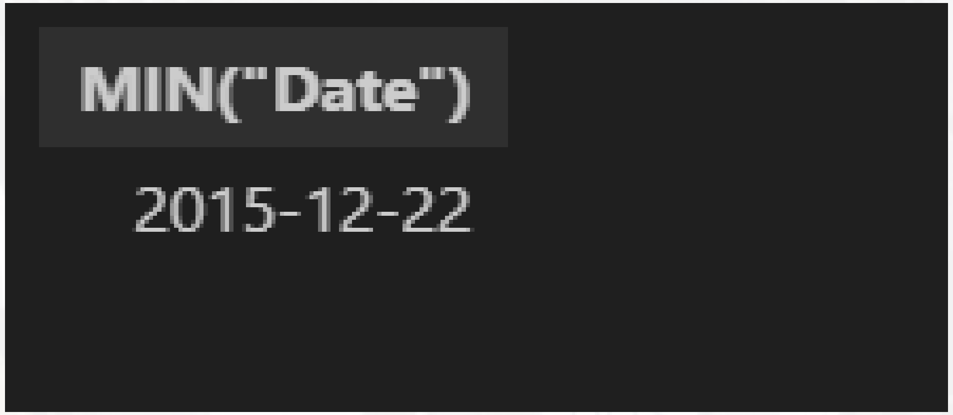


AVG("PAYLOAD_MASS__KG_")
2928.4

- The aggregate function AVG() was used to get the average payload mass carried by booster version F9 v1.1.

First Successful Ground Landing Date

- Dates of the first successful landing outcome on ground pad:

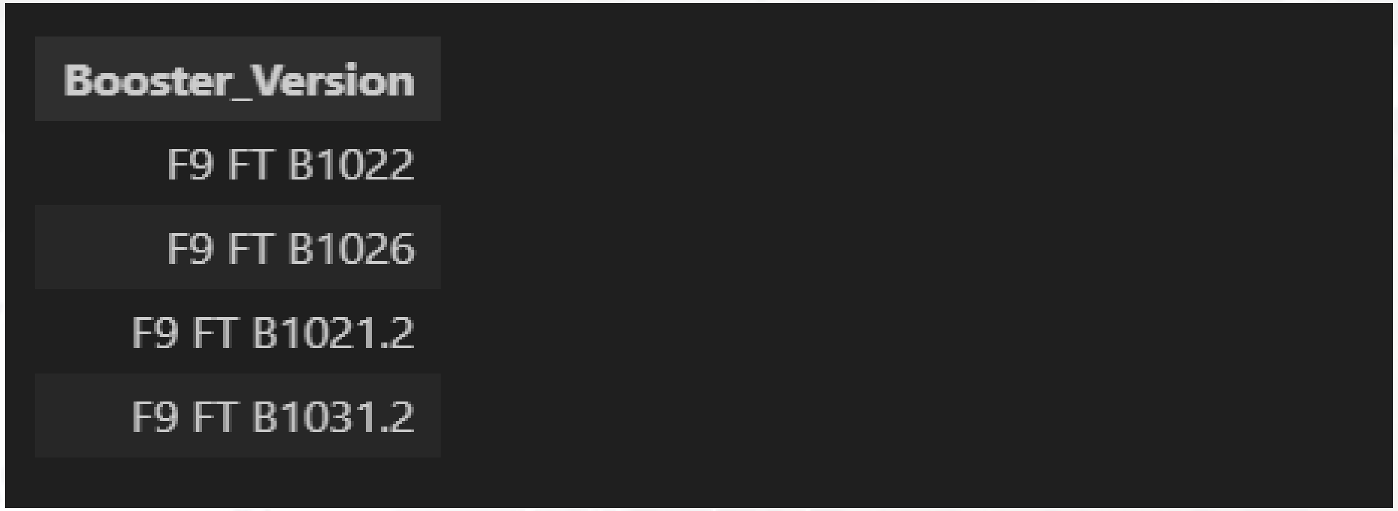


MIN("Date")
2015-12-22

- The aggregate function MIN() was used to find the date of the first successful landing outcome on ground pad.

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000:

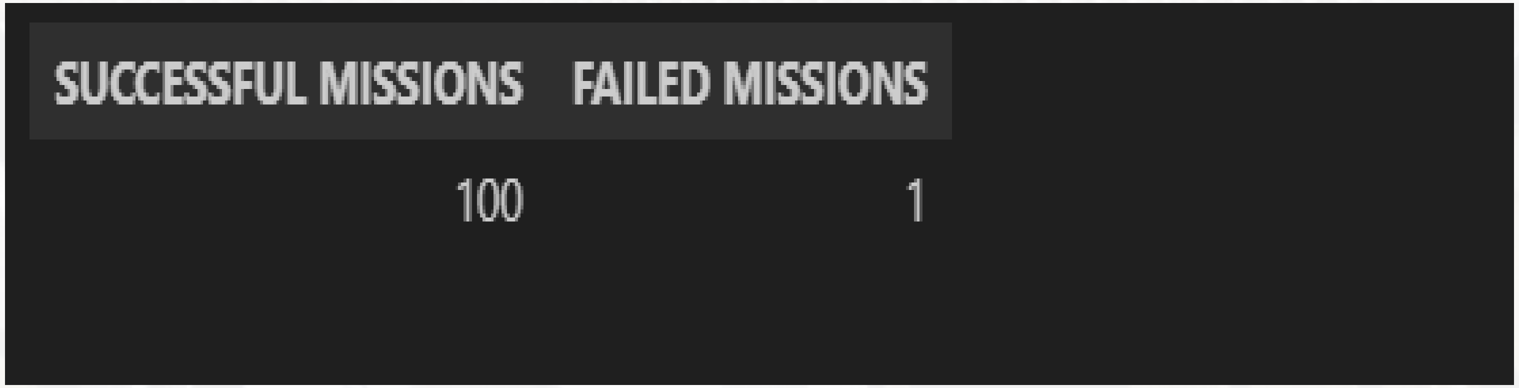


Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- First, a query was made to find where "Landing_Outcome" = 'Success (drone ship)' and then the BETWEEN operator was used to find the range between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failed mission outcomes:

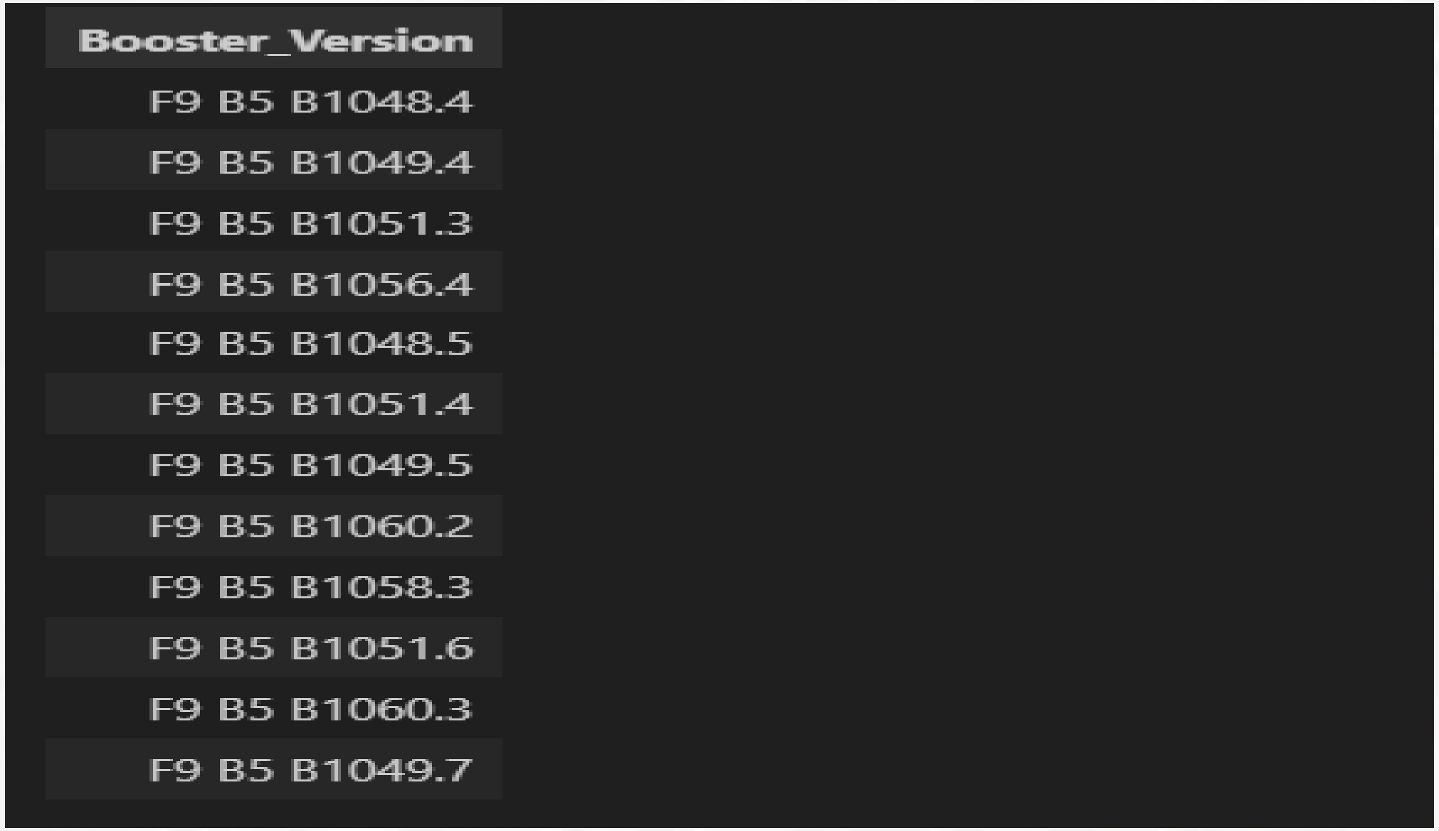


SUCCESSFUL MISSIONS	FAILED MISSIONS
100	1

- A query was made using the aggregate function SUM() to find the total f the successful missions and failed missions and aliases were used to give them column headers.

Boosters Carried Maximum Payload

- Names of the boosters which have carried the maximum payload mass:

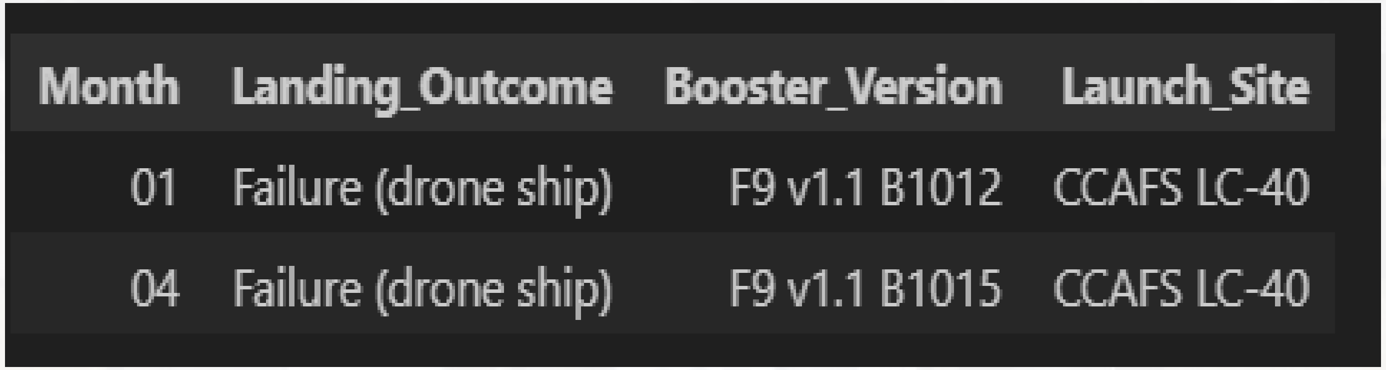


Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- The booster version column was first selected from the SpaceX table and then a subquery was used in conjunction with the aggregate function MAX() to find the boosters which have carried the maximum payload mass.

2015 Launch Records

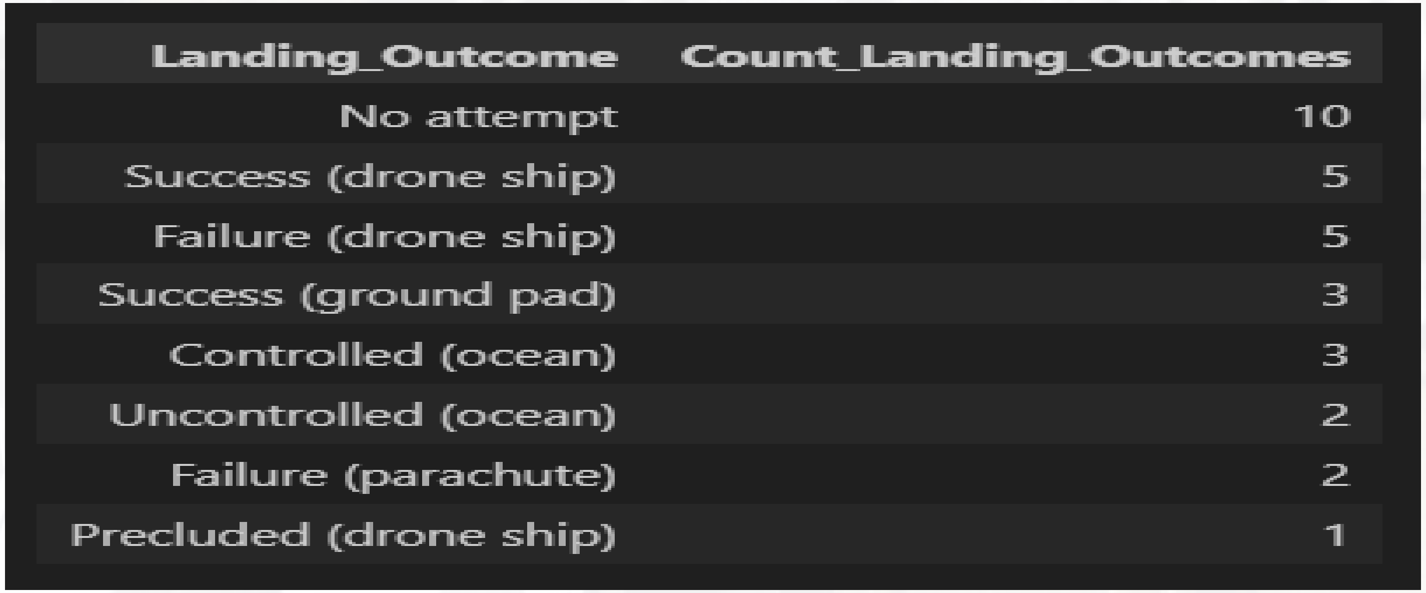
- Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015:



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

- Because sqllite does not support month names, the substring function, subst() was used to get the months where drone ship landings failed.

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

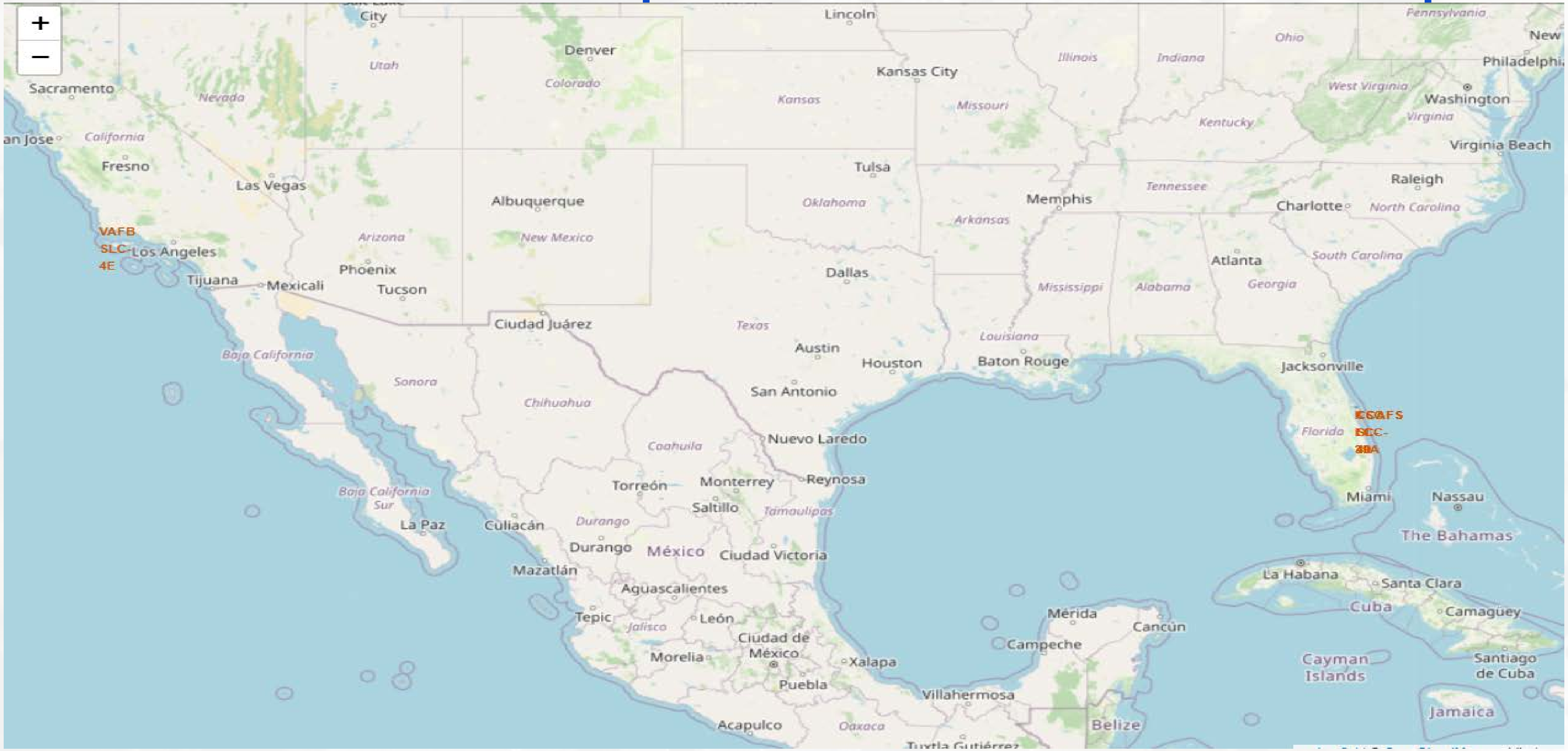


Landing_Outcome	Count_Landing_Outcomes
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- The count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20 were ranked in descending order using the DESC() function.

Section 3

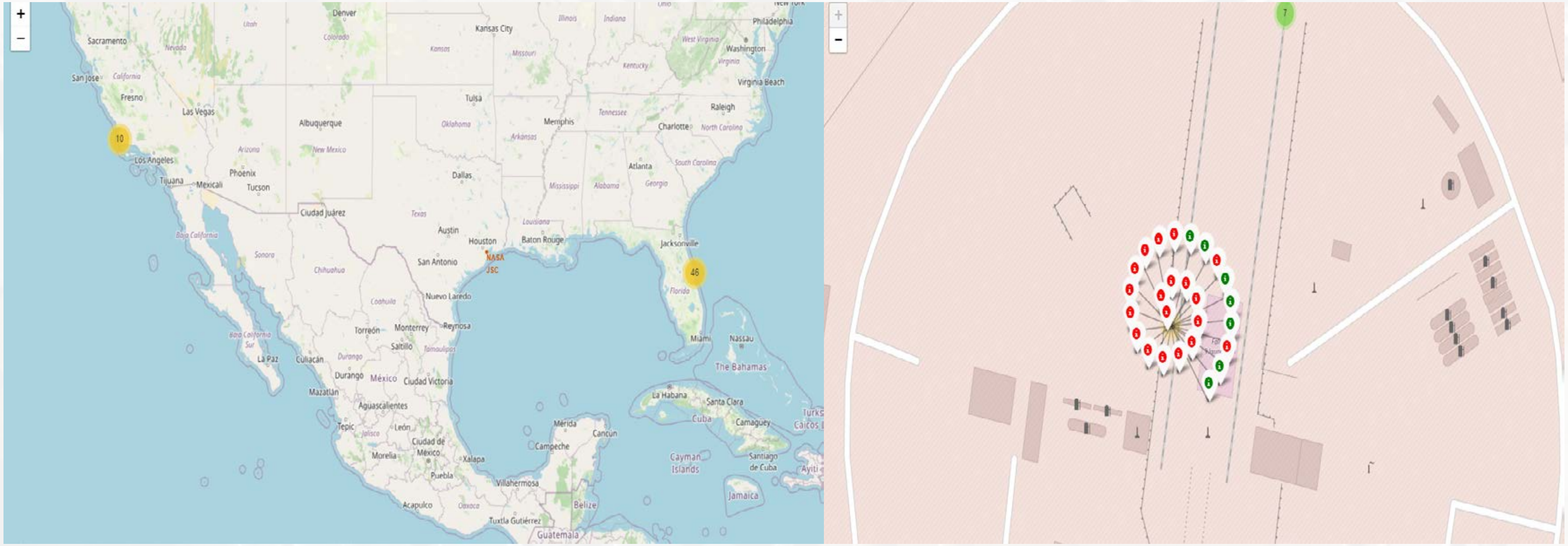
Total Launch Sites presented on Folium Map



- All launch sites are in close proximity to the coast. This is probably a safety measure due to emissions and also debris from launches.

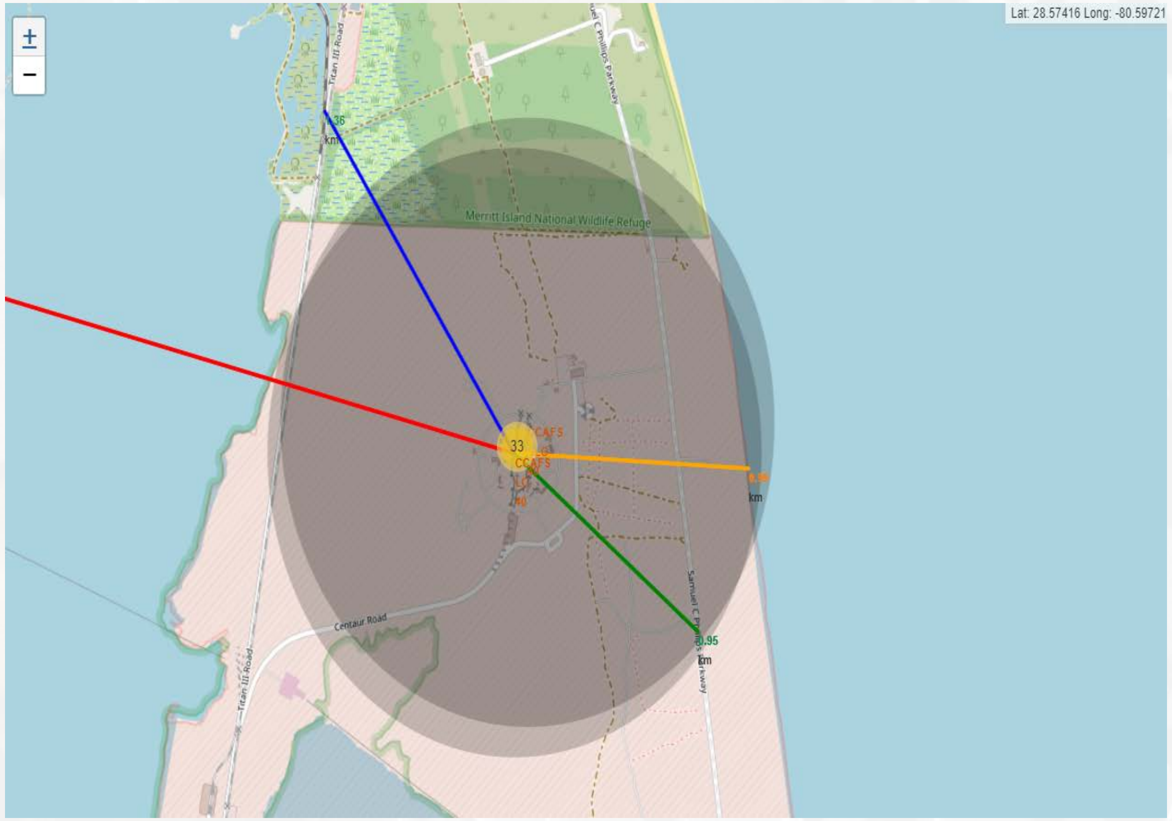
- The launch sites on the west coast are not as close in proximity to the equator line like those in the east coast.

Launch Outcomes - Color Labeled



- Green Marker = Successful Launch.
- Red Marker = Failed Launch.
- The Kennedy Space Center seems to have a high success rate.

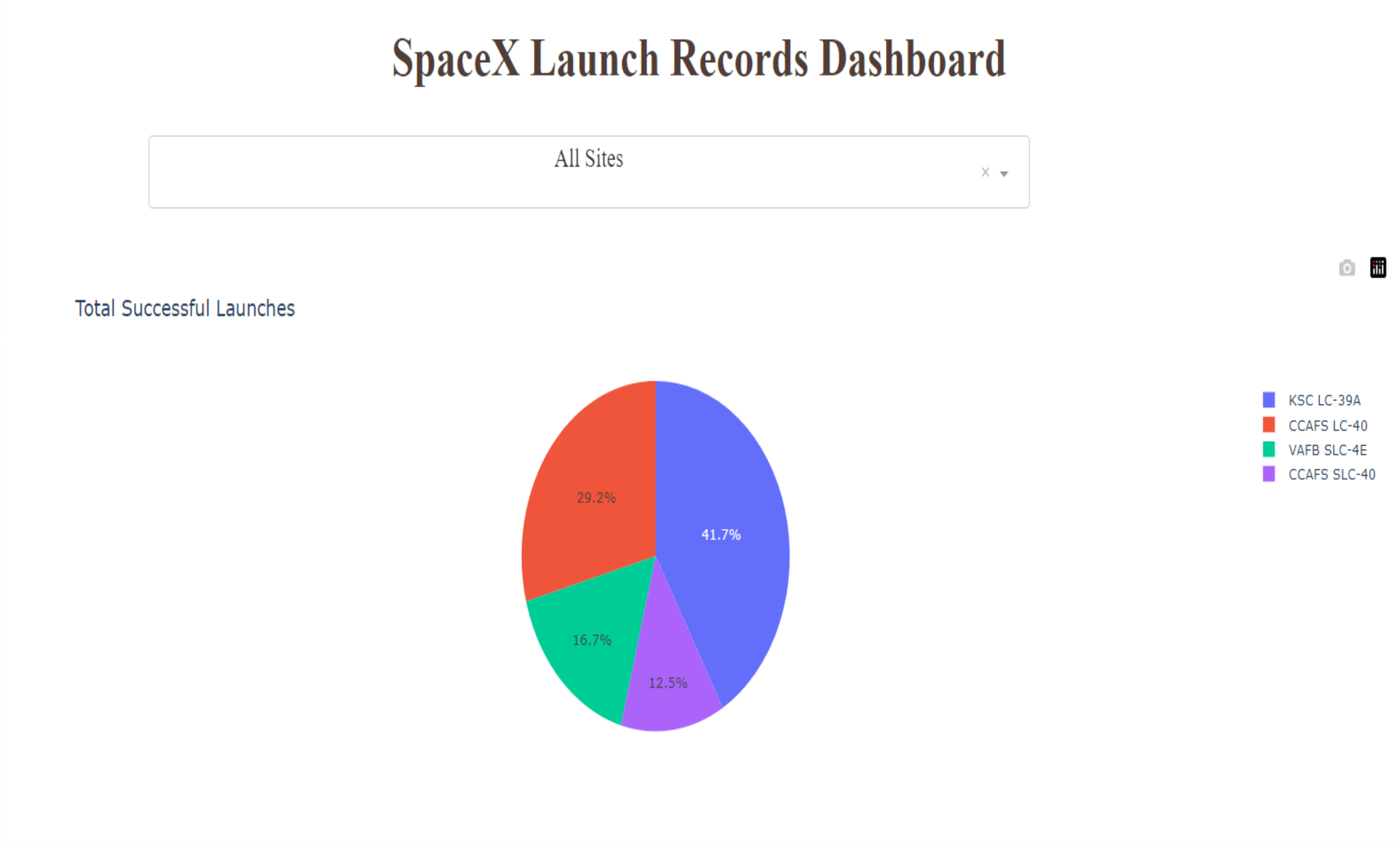
Distance from Launch Site to it’s Proximities:


Lat: 28.57416 Long: -80.59721
+
−
Titan III Road
Samuel C Phillips Parkway
36
km
Merritt Island National Wildlife Refuge
33
CCAFS
CCAFS
LC
40
km
Centaur Road
0.95
km
Titan III Road

- For the CCAFS SLC-40 launch site:
  - Relatively close railway is the NASA Railroad (1.36 km)
  - Relatively close highway is the Samuel C. Phillips Parkway (0.95 km)
  - Relatively close city is Titusville (23.21 km)
  - Relatively close coastline is the Florida Coastline (0.96 km)

Section 4

The Launch Success Count for All Sites:



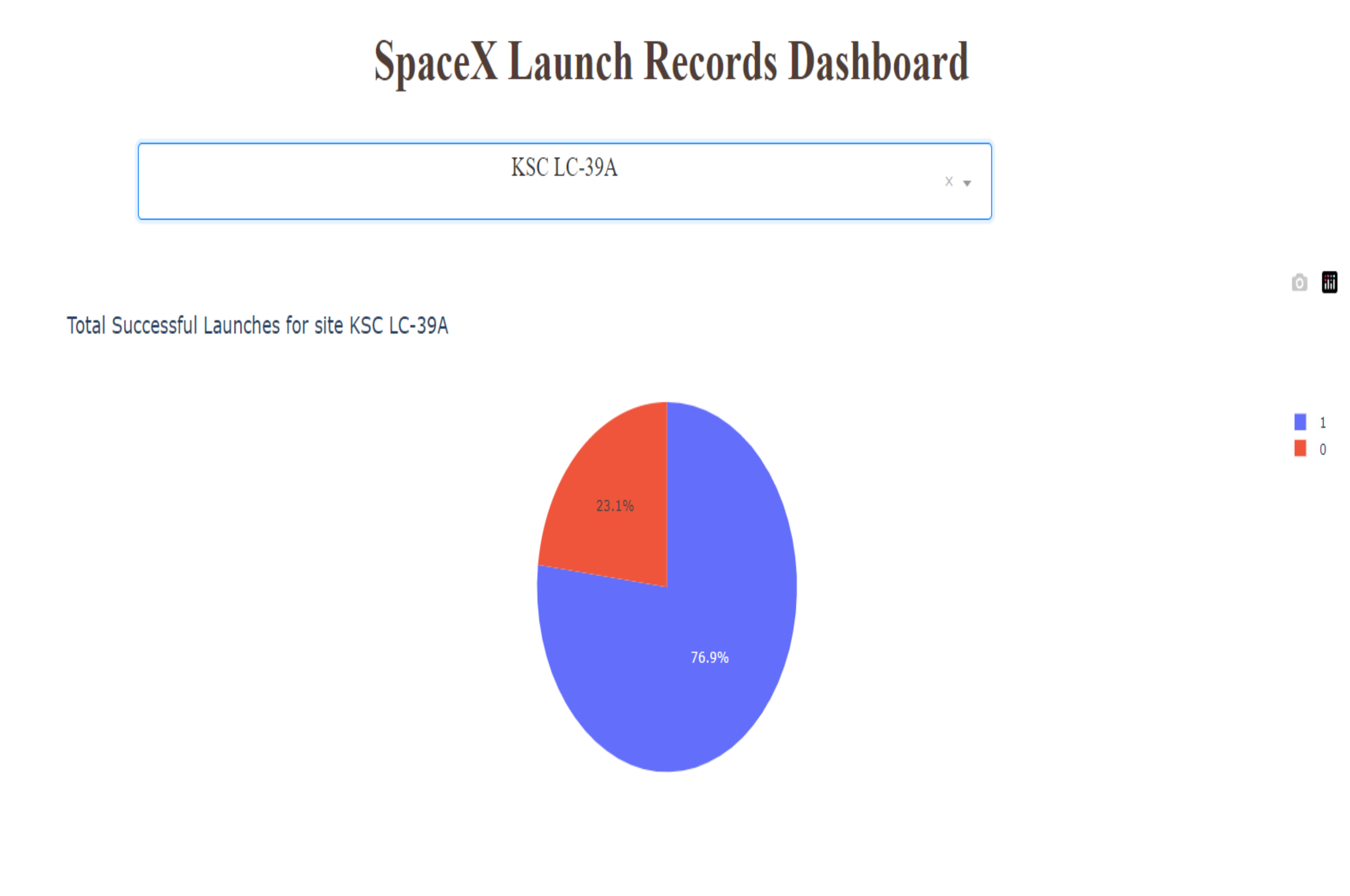
SpaceX Launch Records Dashboard

All Sites

Total Successful Launches

- The chart clearly shows that from all the sites and KSC LC-39A has the most successful launches.

Launch Site with the Highest Launch Success Ratio:



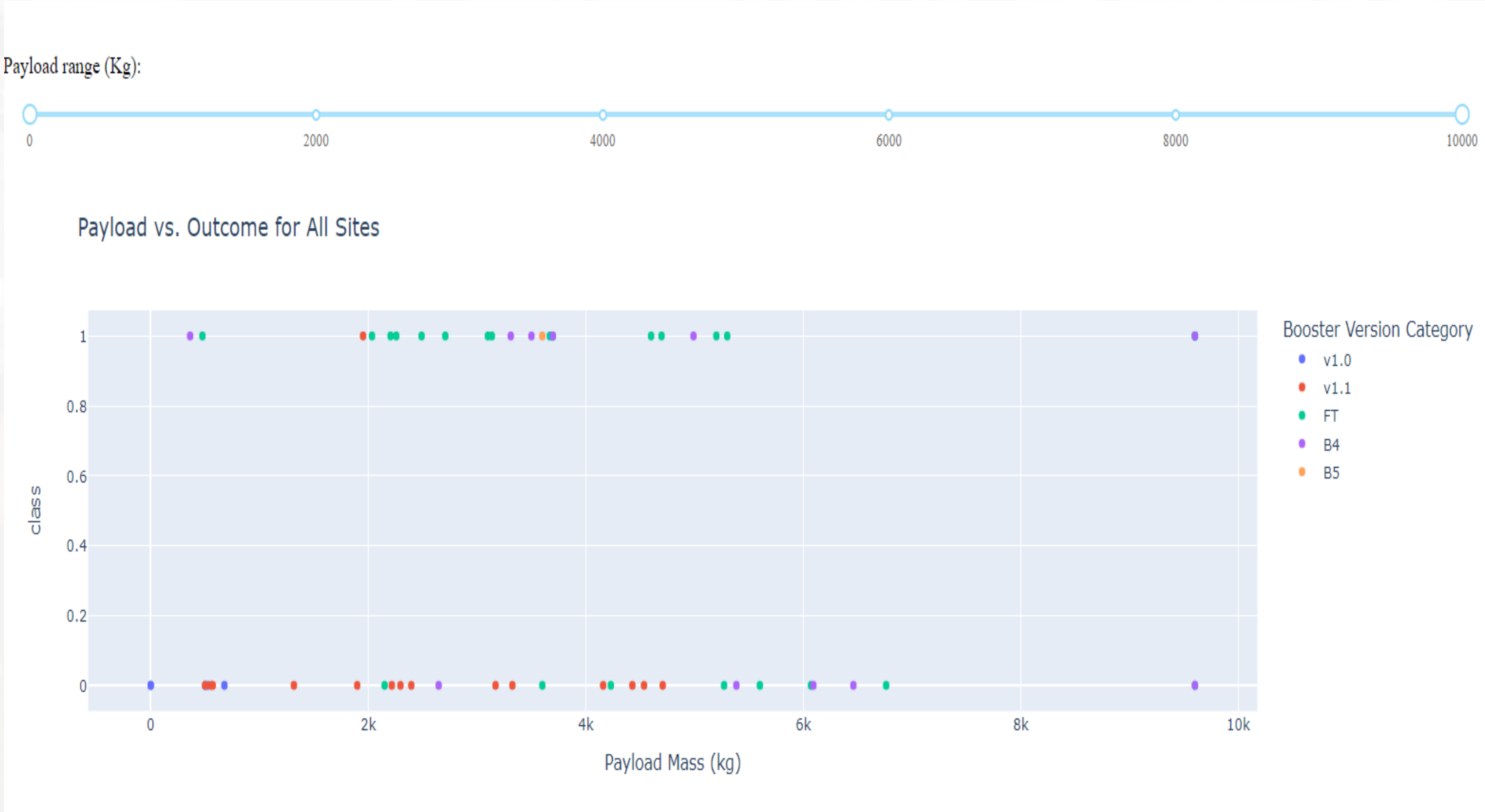
SpaceX Launch Records Dashboard

KSC LC-39A

Total Successful Launches for site KSC LC-39A

- The KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and only 3 failed landings.

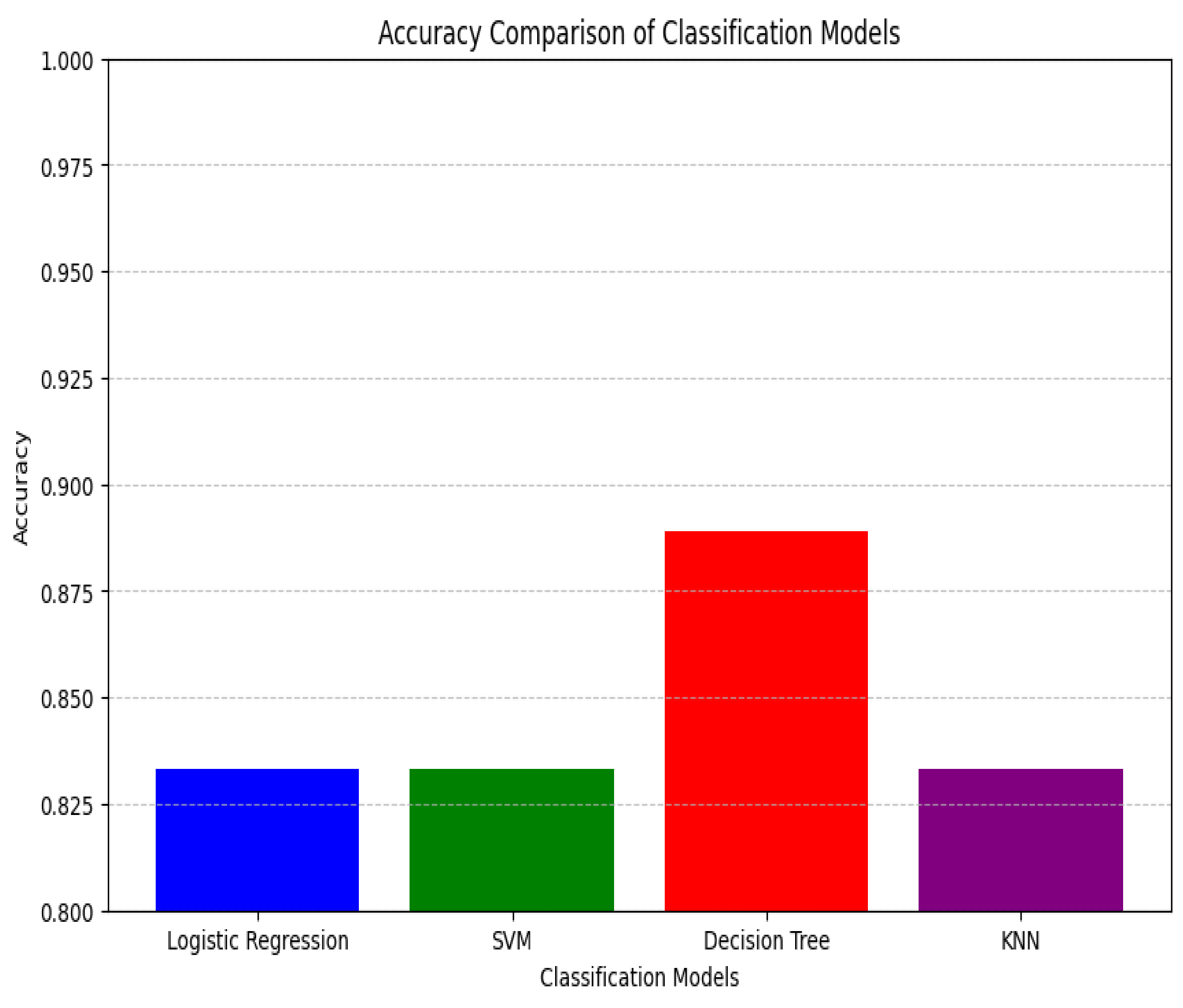
Payload vs. Launch Outcome for all sites:



- The charts shows that payloads between 2000 and 5500 kg have the highest success rate.

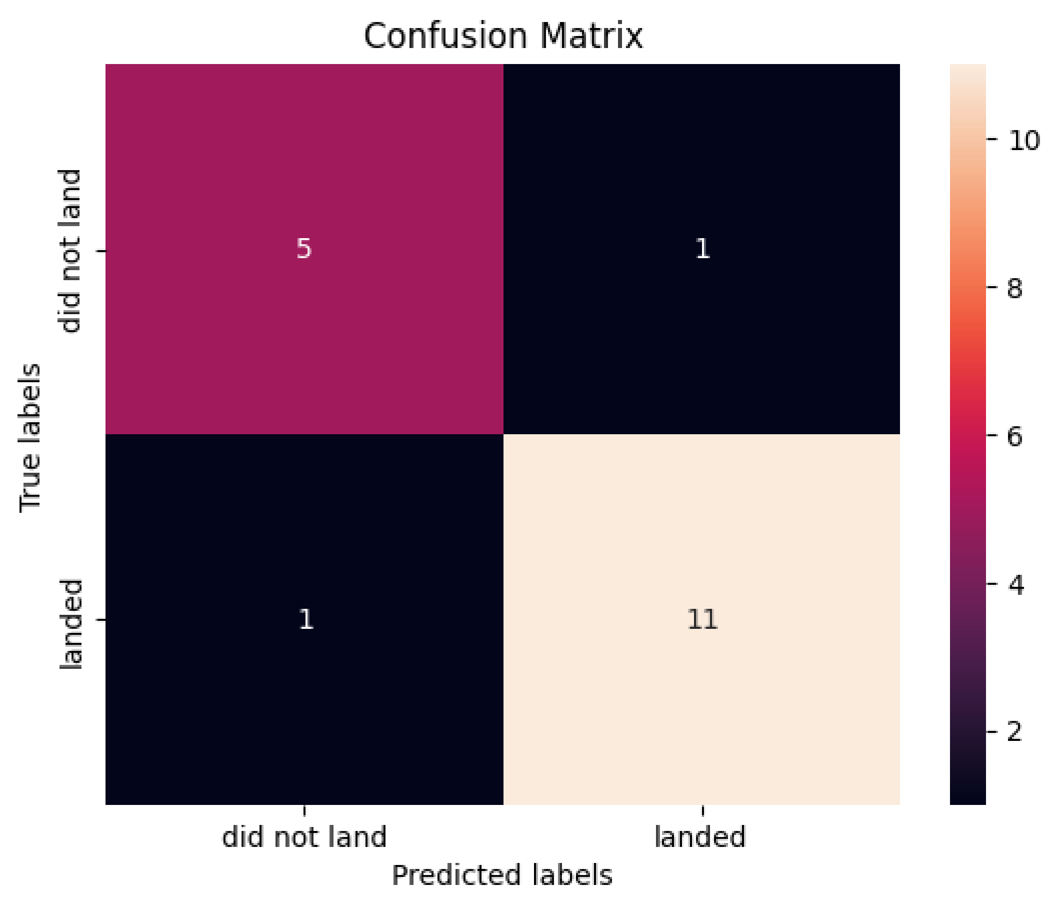
Section 5

Classification Accuracy



- The decision tree classifier has the bes classification accuracy with an accuracy of 88%(0.88) on the test set.

Confusion Matrix



- The decision tree classifier correctly predicted 5 out of 6 launches that did not land and 11 out of 12 launches that landed.

Conclusions

- The decision tree classifier is the best model and can be used to predict successful landings and help SpaceY to become a competitor.
- The best launch site is the Kennedy Space Center (KSC LC-39A).
- Although most of mission outcomes are successful, successful landing outcomes seem to improve over time, as processes and rocket technology become better.
- Launches with heavy payload mass, that is payload mass above 10,000 are risky.

Appendix

- Adjusting the “max_features” parameter in the Decision Tree Classifier enabled us to get the right results.
- I updated the original instructions in order to get the bar chart of the machine learning models showing their accuracy.