IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mohamed Aziz Ajmi
May 10th, 2024

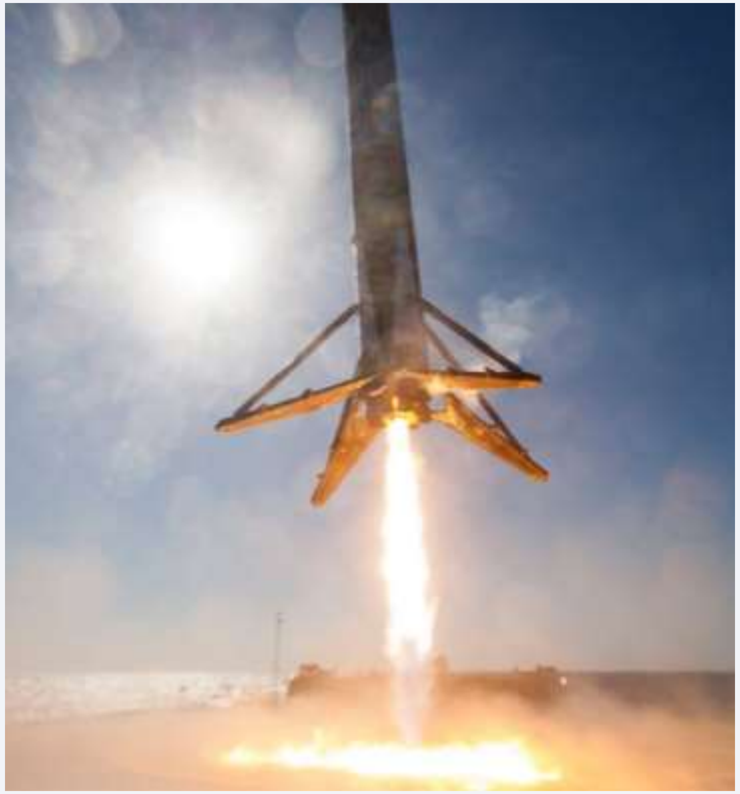
Executive Summary

In the dynamic realm of commercial space exploration, Space Y has embarked on a pioneering journey to transform rocket launch operations using predictive analytics. Drawing from an extensive dataset sourced from SpaceX API and web scraping, alongside meticulous preprocessing methods including data wrangling, exploratory data analysis (EDA), and data visualization, this initiative aimed to accurately forecast rocket launch outcomes.

Employing a range of machine learning algorithms such as Logistic Regression, Support Vector Machines (SVM), Classification Trees, and K-Nearest Neighbors (KNN), we strived to optimize decision-making processes and elevate the efficiency of launch operations. Following rigorous model evaluation and hyperparameter tuning, the standout performer emerged as the Classification Tree model, achieving an impressive accuracy of approximately 87.3%. These findings underscore the pivotal role of preprocessing methodologies in extracting actionable insights from intricate datasets, empowering Space Y to navigate the intricacies of the commercial space sector with confidence and precision.

Introduction

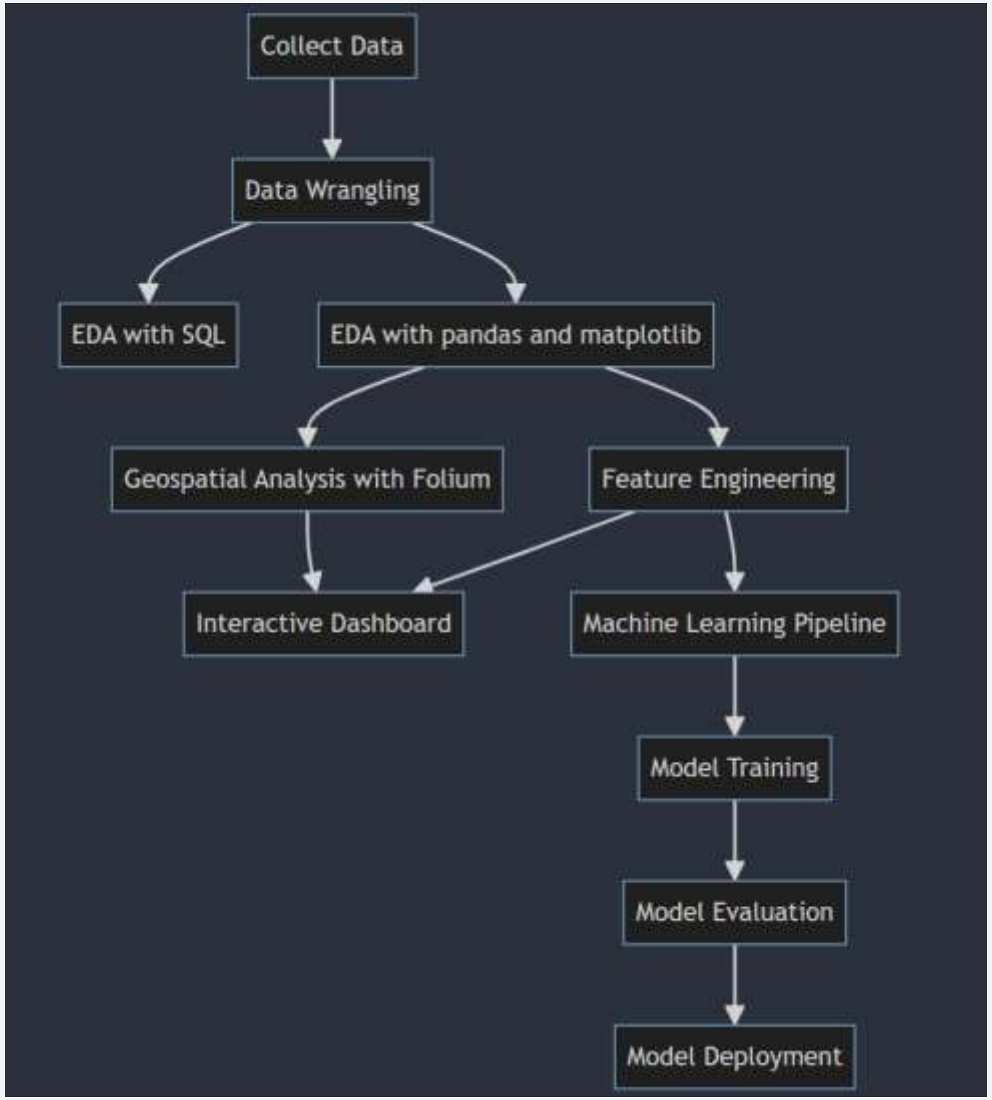
In today's era of expanding space exploration, companies like SpaceX have made space travel more accessible and innovative. Space Y, led by Allon Musk, is a newcomer aiming to shake up the industry. As data scientists working on Space Y's main project, our goal is to use data insights and predictive analytics to understand how space launches work, especially focusing on first-stage landings. By using machine learning and thorough data analysis, we want to give Space Y the tools they need to compete in the commercial space industry alongside established players like SpaceX.



Section 1

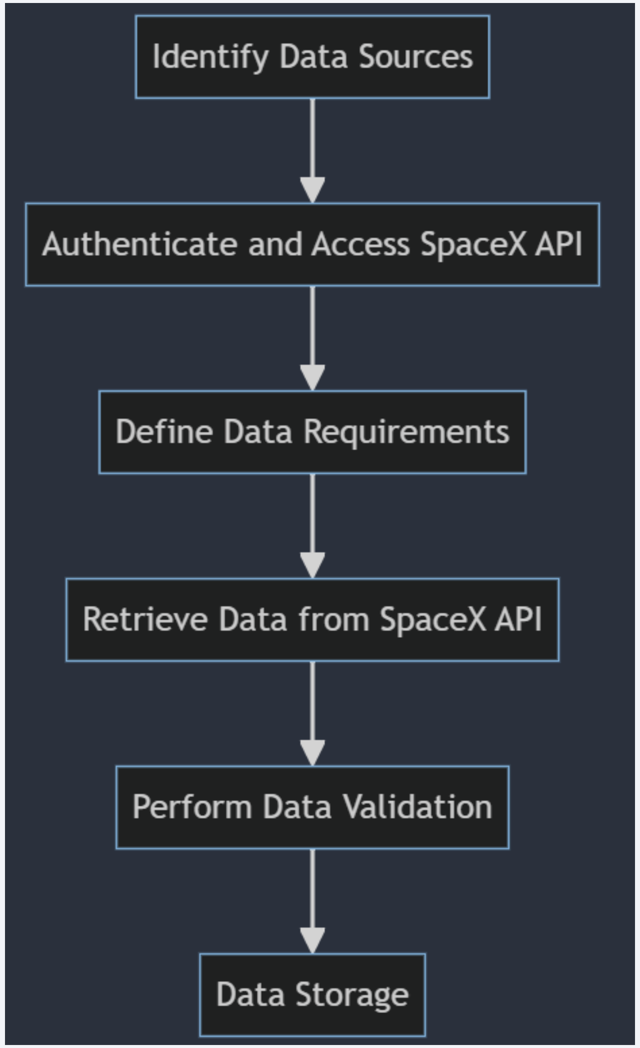
Methodology

Methodology



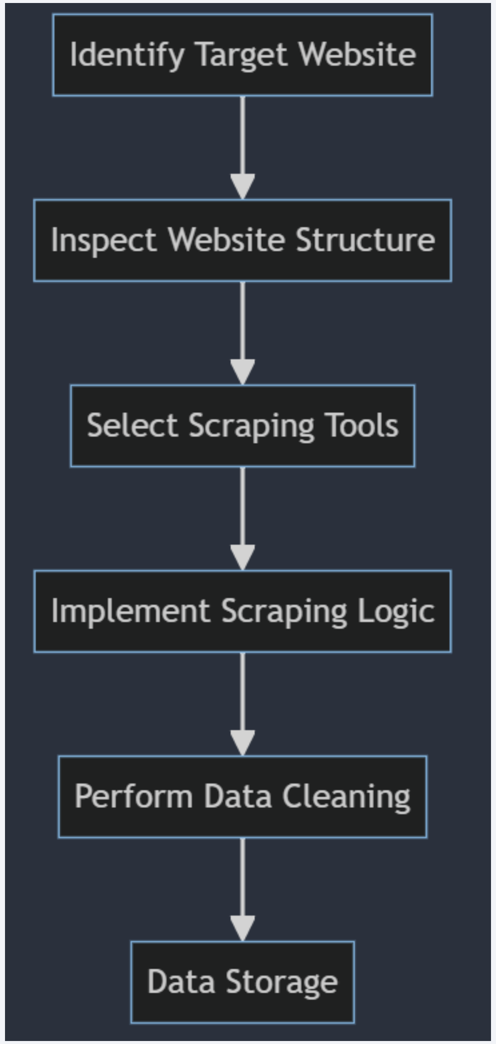
Data Collection – SpaceX API

1. Identify Data Sources: Begin by identifying relevant data sources, including the SpaceX API.
2. Authenticate and Access SpaceX API: Utilize authentication methods provided by the SpaceX API to gain access to the required data endpoints.
3. Define Data Requirements: Determine the specific data fields needed for the analysis, such as launch dates, mission details, rocket information, and landing outcomes.
4. Retrieve Data from SpaceX API: Implement requests to the SpaceX API to fetch the desired data, ensuring proper handling of pagination and rate limits.
5. Perform Data Validation: Validate the retrieved data to ensure accuracy, completeness, and consistency. This may involve checking for missing values, data formatting issues, and anomalies.



Data Collection - Scraping

1. Identify Target Website: Begin by identifying the target website containing the desired data. In this case, the Wikipedia page listing Falcon 9 and Falcon Heavy launches.
2. Inspect Website Structure: Analyze the structure of the target webpage to identify the location of the data elements to be scraped. This involves inspecting HTML tags, classes, and IDs.
3. Select Scraping Tools: Choose appropriate web scraping tools or libraries for the task. Popular options include BeautifulSoup in Python for HTML parsing.
4. Implement Scraping Logic: Develop scraping logic to extract the relevant data from the target webpage. This may involve navigating through HTML elements, selecting specific tables or sections, and extracting text or attributes.



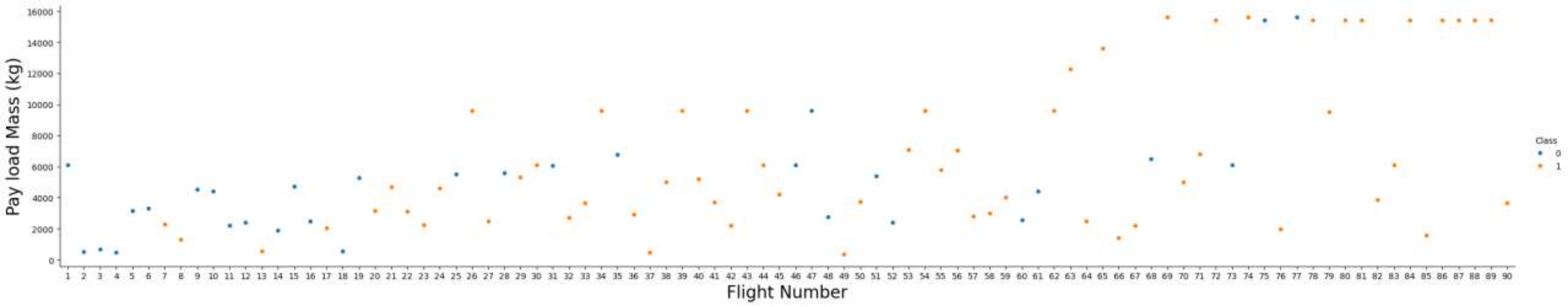
Data Wrangling

1. Identify and Handle Missing Values:
    - Calculate the percentage of missing values in each attribute to assess data completeness.
    - Implement strategies to handle missing values, such as imputation, deletion, or flagging.
2. Identify Numerical and Categorical Columns:
    - Distinguish between numerical and categorical columns to facilitate data analysis and transformation.
3. Calculate Number of Launches on Each Site:
    - Aggregate data to calculate the total number of launches on each launch site, providing insights into launch distribution.
4. Calculate Number and Occurrence of Each Orbit:
    - Analyze the data to determine the unique orbits and their respective frequencies, aiding in orbit-related analysis.
5. Calculate Number and Occurrence of Mission Outcomes for Orbits:
    - Determine the number and occurrence of mission outcomes associated with each orbit, providing insights into mission success rates.
6. Create Landing Outcome Label:
    - Utilize information from the 'Outcome' column to create a new categorical variable indicating the landing outcome, facilitating predictive modeling.

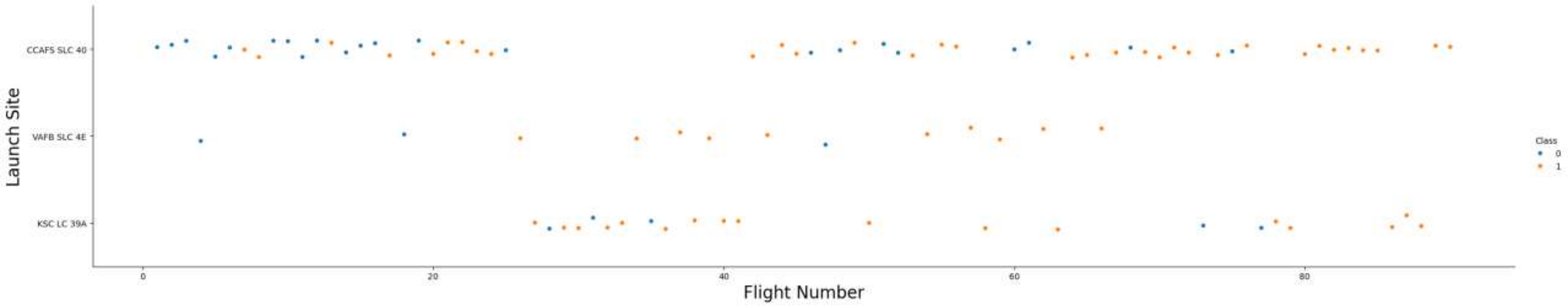
Section 2

Insights drawn from EDA

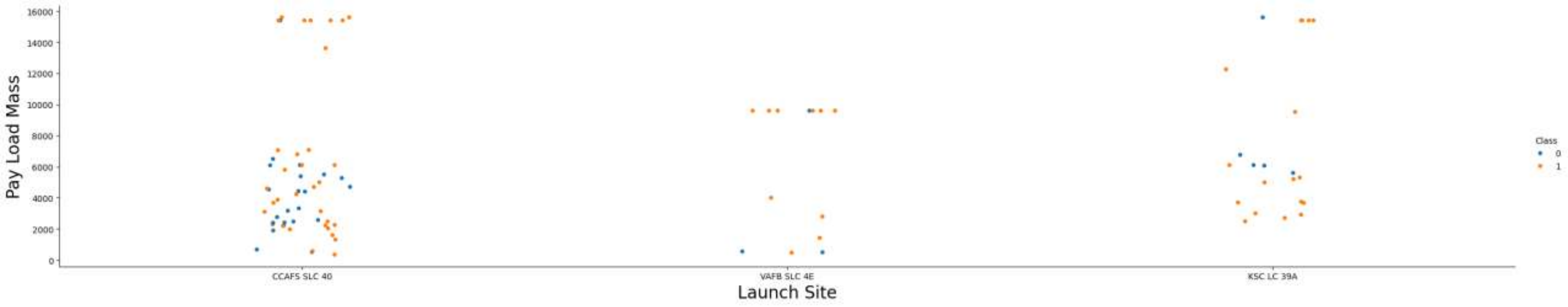
Flight Number vs. Payload Mass



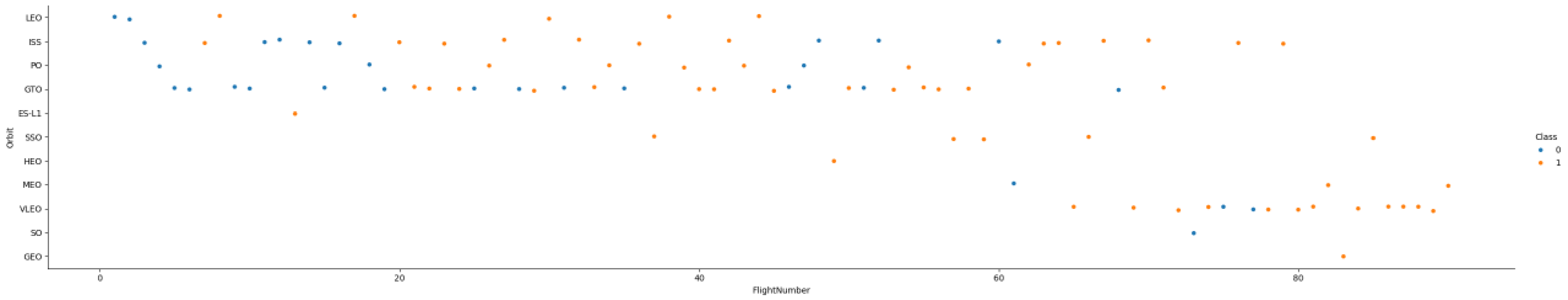
Flight Number vs. Launch Site



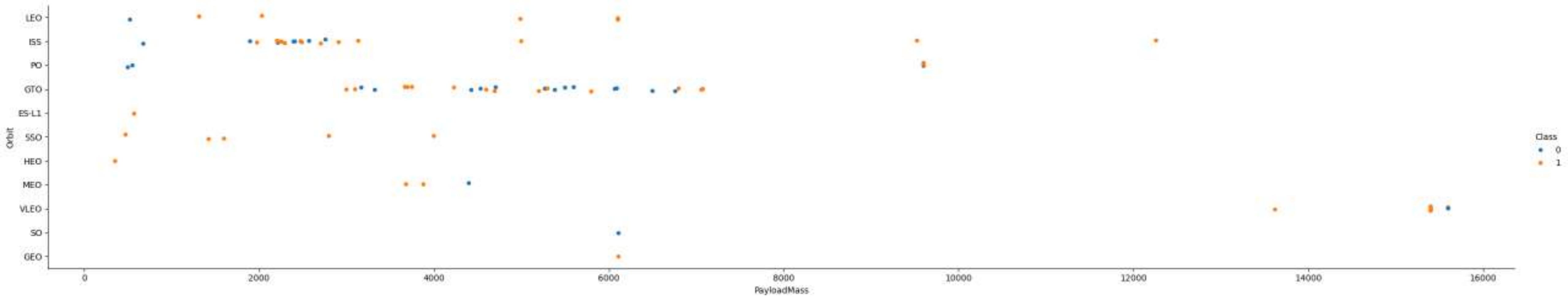
Launch Site vs. Payload Mass



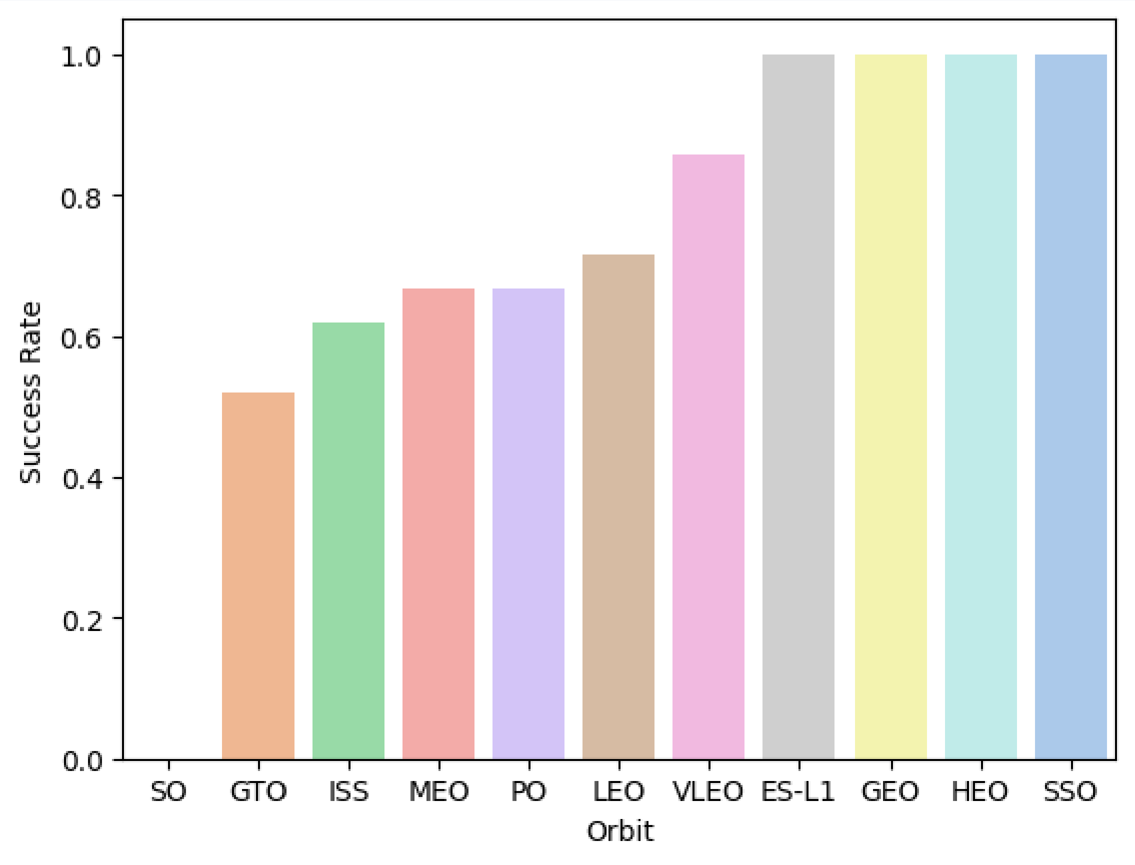
Flight Number vs. Orbit



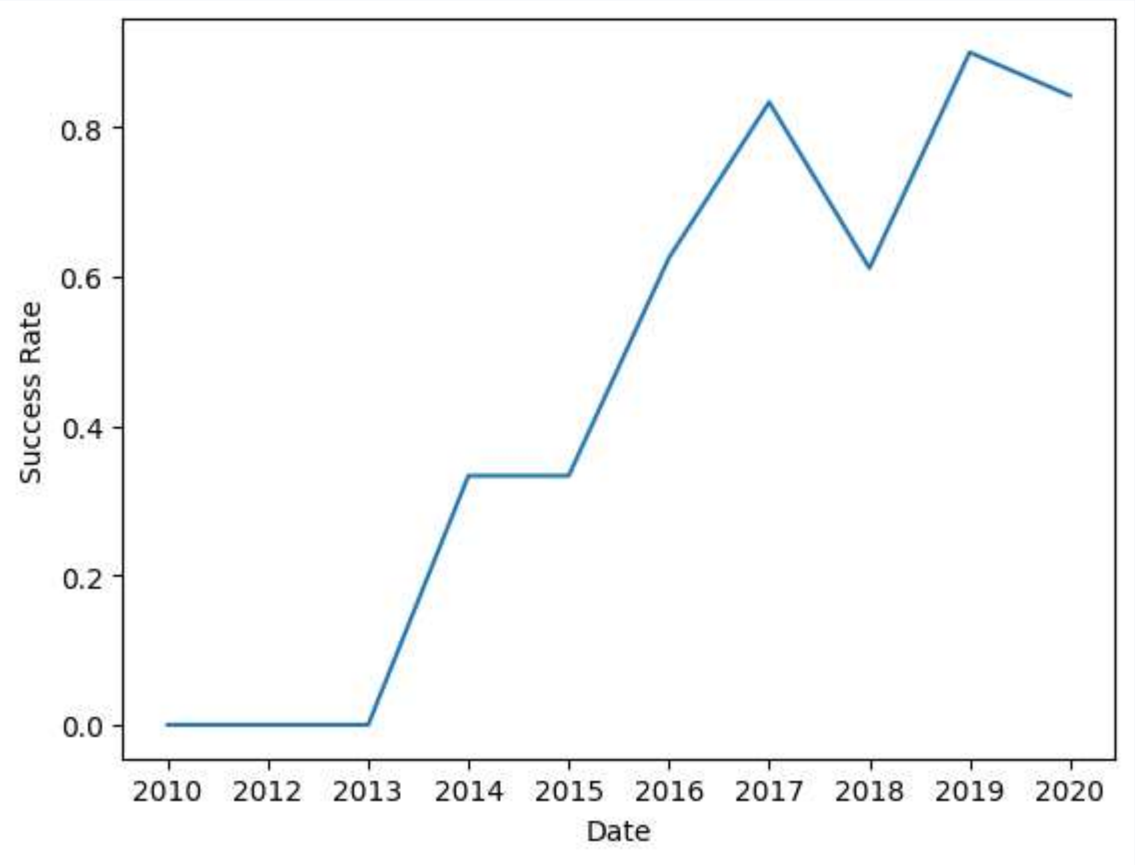
Payload Mass vs. Orbit



Success Rate for each Orbit



Success Rate Trend



EDA with SQL

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

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

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

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

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

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

G: List the total number of successful and failure mission outcomes

H. List the names of the booster_versions which have carried the maximum payload mass.

I. 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

J. 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.

Build an Interactive Map with Folium

- Task 1:
  - Markers: Created markers to represent each launch site on the map, providing a visual indication of their locations.
- Task 2:
  - Markers: Added markers to denote the success or failure of launches for each site on the map, distinguishing between successful and failed launches.
- Task 3:
  - Circles: Utilized circles to visualize the proximity around each launch site, helping to calculate distances between the launch site and its surroundings.

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

All Launch Site Names

```
%%sql

SELECT DISTINCT(Launch_Site)
FROM SPACEXTABLE
```

✓ 0.0s

* 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'

```
%%sql

SELECT *
FROM SPACEXTABLE
WHERE Launch_Site LIKE "CCA%"
LIMIT 5
```

✓ 0.0s Python

* sqlite:///my_data1.db
Done.

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

Total Payload Mass

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_)
FROM SPACEXTABLE
WHERE Customer LIKE "%NASA (CRS)%"
```

✓ 0.0s

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

SUM(PAYLOAD_MASS__KG_)
48213

Average Payload Mass by F9 v1.1

```
%%sql

SELECT AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTABLE
WHERE Booster_Version LIKE "F9 v1.1%"
```

✓ 0.0s

* sqlite:///my_data1.db

Done.

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

```
%%sql

SELECT MIN(Date)
FROM SPACEXTABLE
WHERE Landing_Outcome LIKE "%ground%"
```

✓ 0.0s

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

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql

SELECT Booster_Version,
Landing_Outcome,
PAYLOAD_MASS__KG_
FROM SPACEXTABLE
WHERE 400 < PAYLOAD_MASS__KG_
AND PAYLOAD_MASS__KG_ < 6000
AND Landing_Outcome LIKE "%drone%"
AND Landing_Outcome LIKE "%Success%"
```

✓ 0.0s

* sqlite:///my_data1.db
Done.

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1021.1	Success (drone ship)	3136
F9 FT B1022	Success (drone ship)	4696
F9 FT B1023.1	Success (drone ship)	3100
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1029.2	Success (drone ship)	3669
F9 FT B1038.1	Success (drone ship)	475
F9 FT B1031.2	Success (drone ship)	5200
F9 B4 B1042.1	Success (drone ship)	3500
F9 B5 B1046.1	Success (drone ship)	3600

Total Number of Successful and Failure Mission Outcomes

```
%%sql

SELECT Landing_Outcome, COUNT(Landing_Outcome)
FROM SPACEXTABLE
GROUP BY Landing_Outcome
```

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

Landing_Outcome	COUNT(Landing_Outcome)
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

```
%%sql

SELECT Booster_Version,
PAYLOAD_MASS__KG_
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ =
    (SELECT MAX(PAYLOAD_MASS__KG_)
     FROM SPACEXTABLE)
```

✓ 0.0s

* sqlite:///my_data1.db
Done.

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

```
%%sql

SELECT substr(Date, 6,2) AS month,
date,
Landing_Outcome,
Booster_Version
Launch_Site
FROM SPACEXTABLE
WHERE substr(Date,0,5)='2015'
AND Landing_Outcome LIKE "%drone%"
AND Landing_Outcome LIKE "%Failure%"
```

✓ 0.0s

* sqlite:///my_data1.db
Done.

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

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

```
%%sql

SELECT Landing_Outcome, count(*)
FROM SPACEXTABLE
GROUP BY Landing_Outcome
HAVING Date BETWEEN "2010-06-04" AND "2017-03-20"
ORDER BY count(*) DESC
```

✓ 0.0s

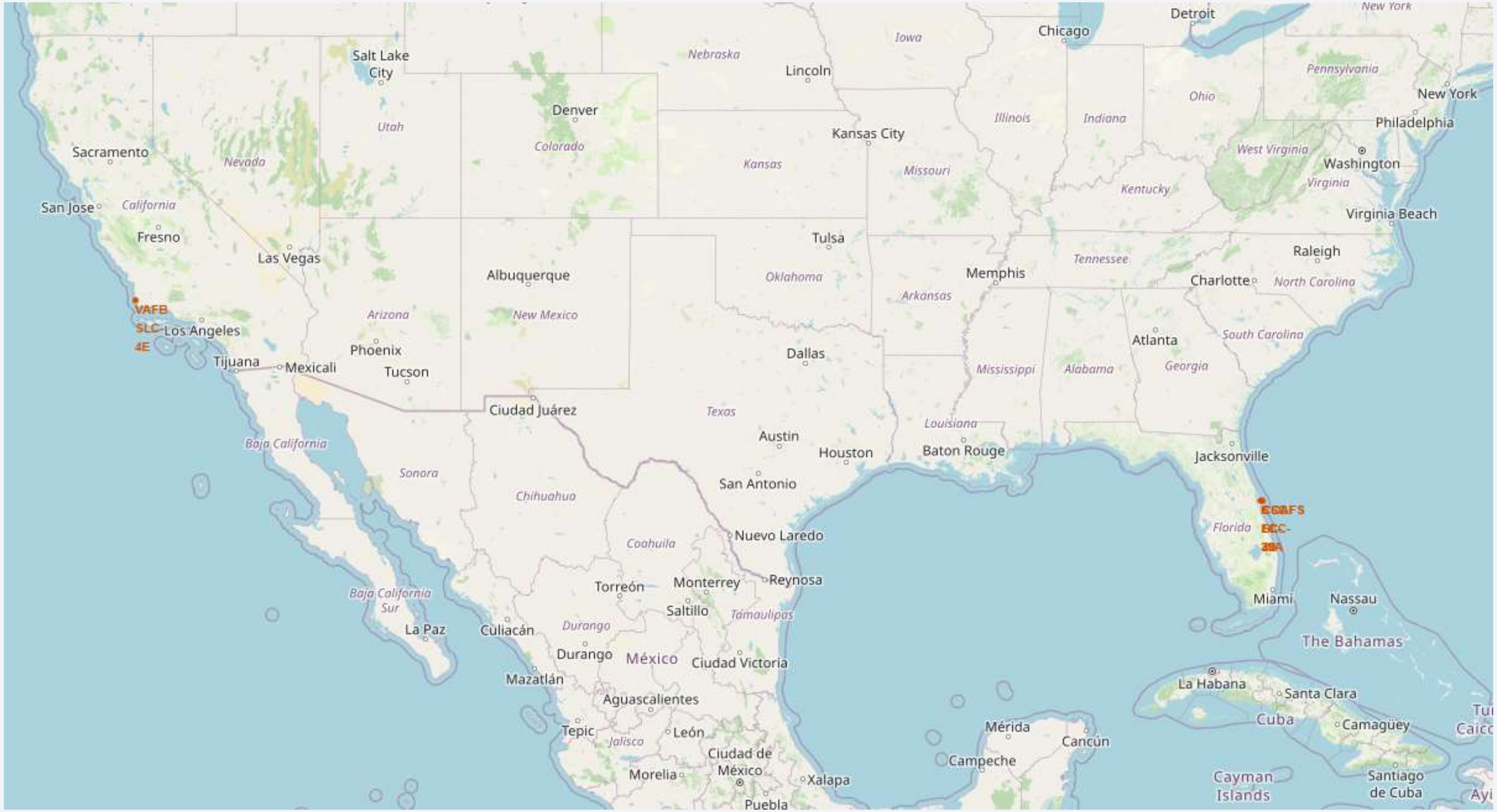
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	count(*)
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

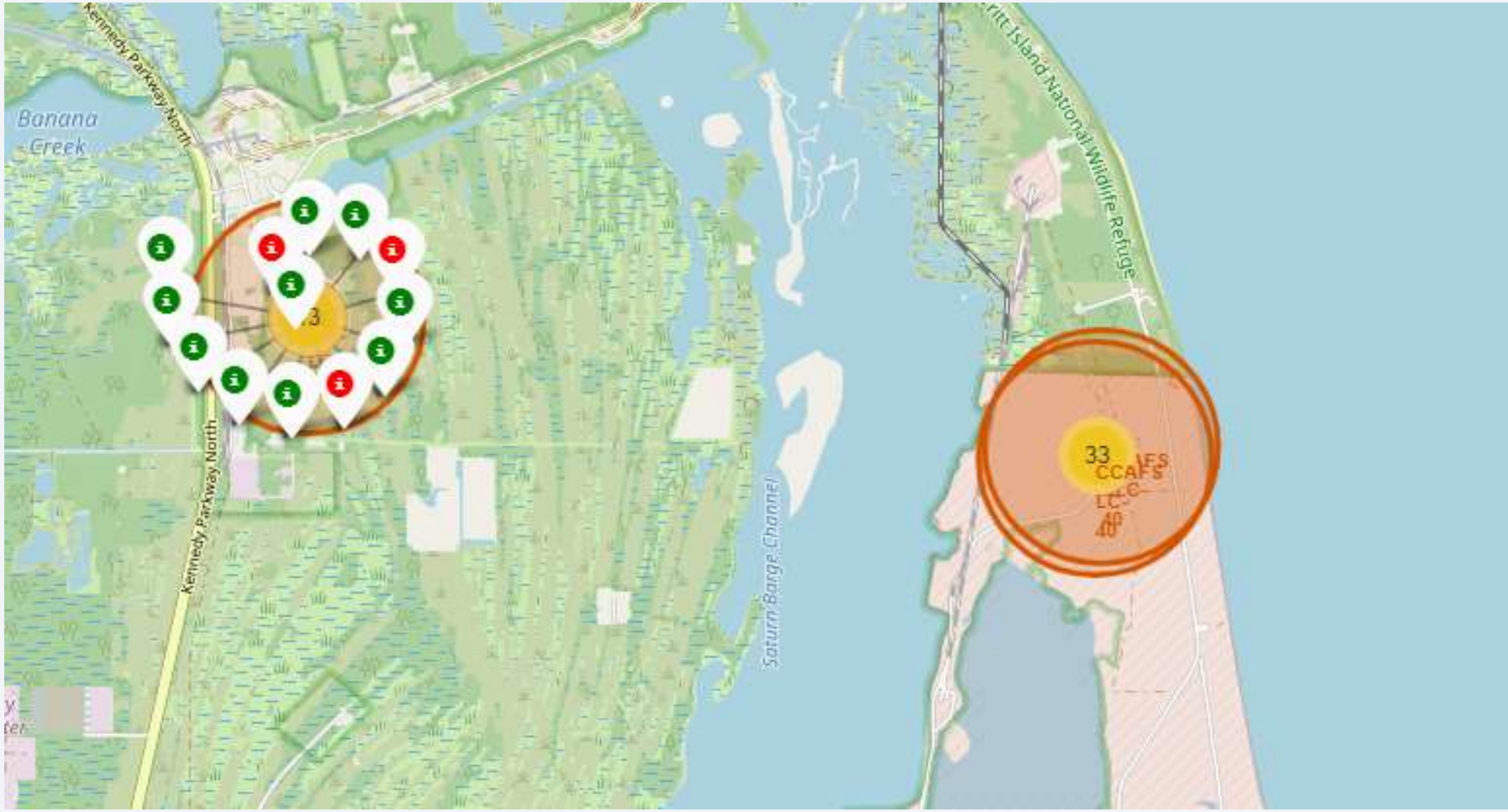
Section 3

Launch Sites Proximities Analysis

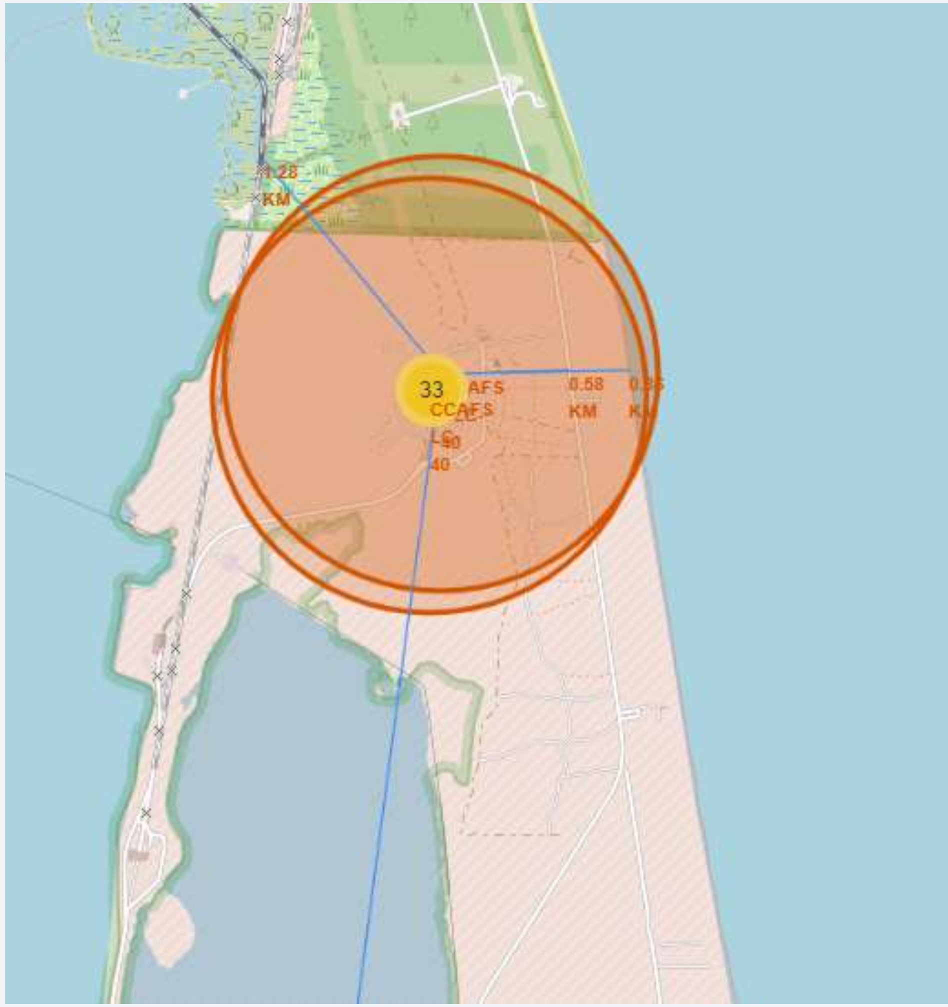
Launch Sites



Landing Outcomes



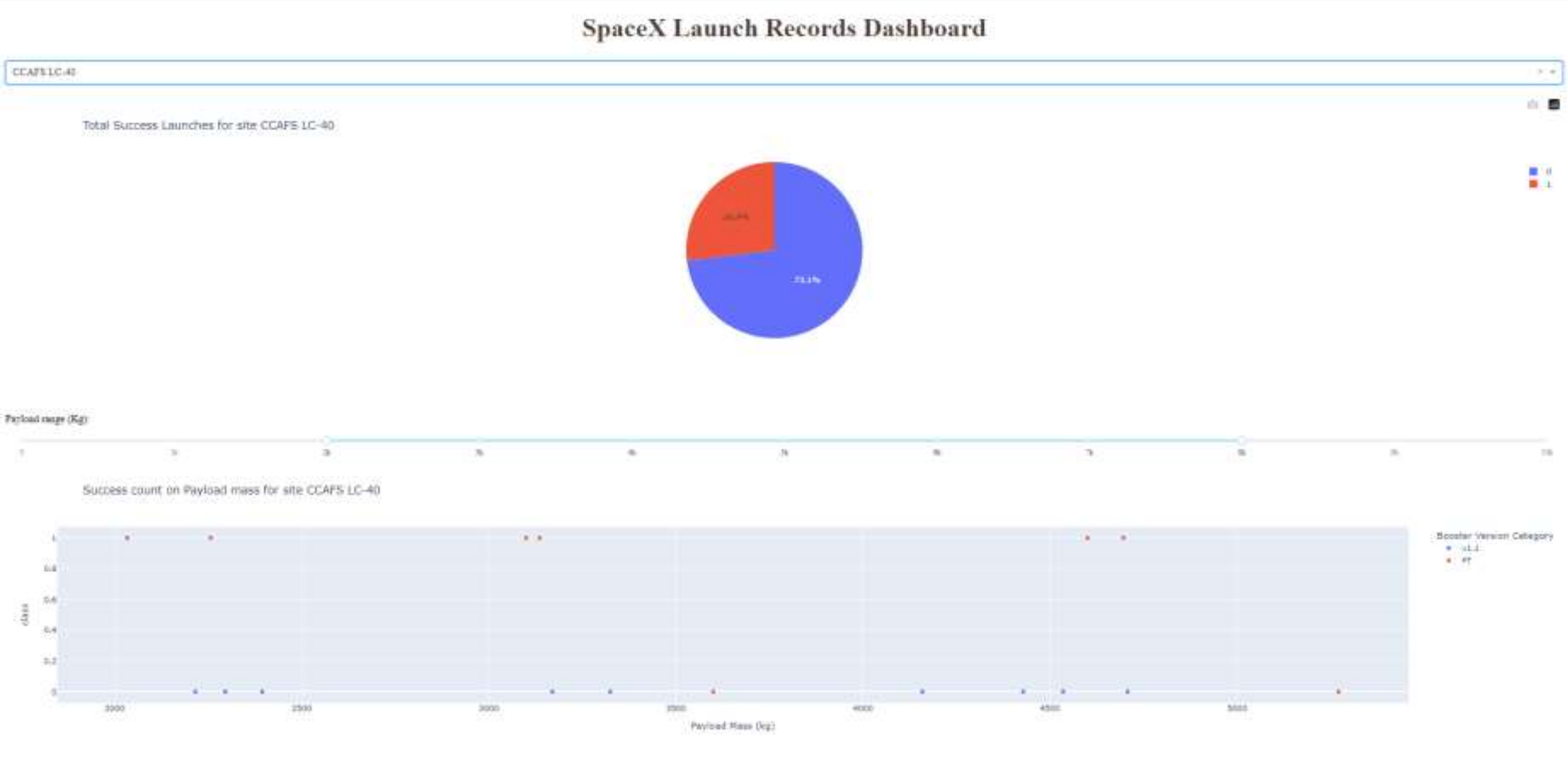
Proximities



Section 4

Build a Dashboard with Plotly Dash

Dashboard


SpaceX Launch Records Dashboard
Total Success Launches for site CCAFS LC-40
Payload range (Kg):
Success count on Payload mass for site CCAFS LC-40
Booster Version Category
class
2000
2500
3000
3500
4000
4500
5000
Payload Mass (kg)

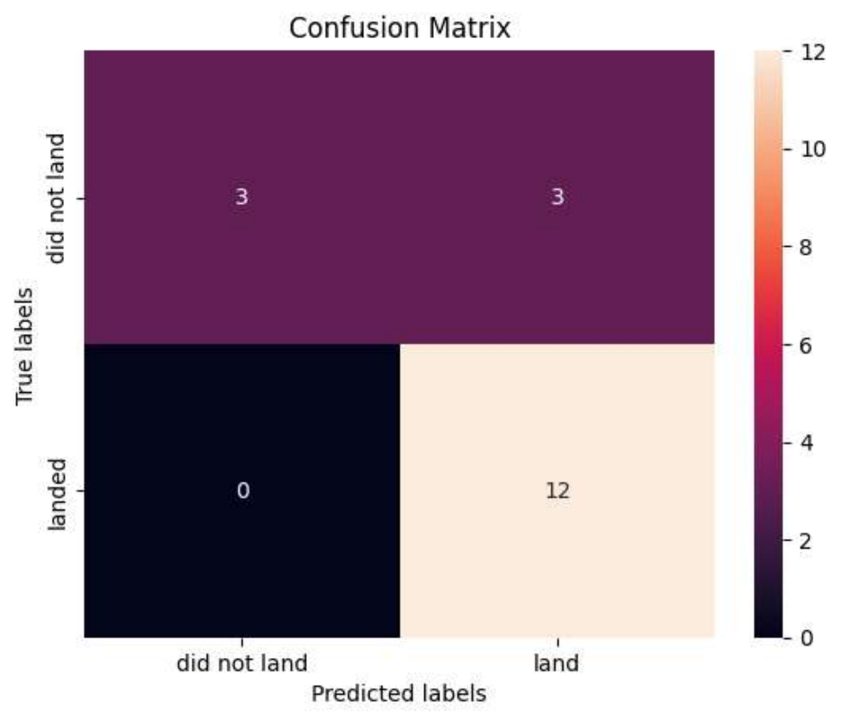
Section 5

Predictive Analysis (Classification)

Model Selection

During the predictive analysis phase, we kick off with exploratory data analysis (EDA) to delve into our dataset and define appropriate training labels. This entails organizing data into distinct classes, serving as our target variable for prediction. Following this, we standardize the data for consistency across different scales. We then divide the dataset into training and test subsets to facilitate model training and assessment. We proceed by exploring a range of machine learning algorithms like Support Vector Machines (SVM), Classification Trees, Logistic Regression, and K-Nearest Neighbors (KNN). For each algorithm, we optimize hyperparameters through techniques like grid search. Next, we evaluate each method's performance using test data, employing cross-validation, confusion matrix analysis, and classification scoring metrics. This thorough approach helps pinpoint the most effective predictive model, empowering us to make informed decisions and extract meaningful insights from the data.

Model Performance



Based on the results of our predictive modeling analysis, it is evident that all the models performed commendably well in terms of accuracy. The Logistic Regression and Support Vector Machine (SVM) models both achieved high accuracy scores of approximately 84.6% and 84.8%, respectively. Additionally, the Classification Tree model demonstrated even higher accuracy, reaching approximately 87.3%. Similarly, the K-Nearest Neighbors (KNN) model yielded a competitive accuracy score of approximately 84.8%. These findings indicate that each model was effective in capturing the underlying patterns within the data and making accurate predictions. However, while all models performed relatively well, the Classification Tree model stood out with the highest accuracy, suggesting its potential superiority in this particular predictive task. Overall, the results underscore the efficacy of machine learning techniques in predictive analysis and highlight the importance of selecting appropriate algorithms tailored to the specific characteristics of the dataset.

Conclusions

- The completion of this project emphasizes the crucial role of preprocessing techniques in predictive analytics for rocket launches.
- Thorough data collection efforts, including API data gathering and web scraping, yielded a comprehensive dataset crucial for model training.
- Preprocessing steps like data wrangling, exploratory data analysis (EDA), and data visualization upheld the dataset's quality and integrity.
- Insights gleaned from EDA, supported by SQL queries and visualization tools such as Folium, guided model selection and hyperparameter tuning.
- Evaluation of models revealed the Classification Tree as the standout performer, achieving an accuracy of around 87.3%.
- These results highlight the significance of robust preprocessing pipelines in extracting actionable insights and facilitating informed decision-making within the commercial space industry.

Thank you!