IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Xing Liu
2025-04-02

Outline

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

Executive Summary

- **Summary of methodologies** Page 6
- **Summary of all results:**
- **Data Collection and Wrangling**:
  - The dataset was successfully cleaned and structured, with missing payload mass values replaced by the mean.
  - The features such as payload mass, orbit type, launch site, booster version, landing pad were extracted and prepared for analysis.
  - **c**reate a landing outcome label (**Class**)
- **Exploratory Data Analysis**:
  - **Launch Sites**, **Payload Mass and Orbit Types** were identified as the primary features determining successful Falcon 9 booster recovery.
- **Predictive Modeling**:
  - **Model Performance**: The Decision Tree model outperformed all other models.
  - **Test Set Performance**: All models achieved a good test score.
- **Interactive Visual Analytics**:
  - **Launch Site Analysis**: Interactive maps revealed patterns in launch site locations and their success rates.
- **Dashboard Insights**: The Plotly Dash dashboard allowed stakeholders to explore data trends, such as the relationship between payload mass, booster version, launch sites and success rate, dynamically.

Introduction

SpaceX has revolutionized the space industry by offering Falcon 9 rocket launches at $62 million—significantly lower than competitors' prices, which often exceed $165 million per launch. This cost advantage stems primarily from SpaceX's ability to **reuse the first stage** of its rockets.

Through comprehensive analysis of SpaceX's historical launch data—including **data collection, wrangling, exploratory data analysis (EDA), interactive visual analytics, feature engineering and machine learning modeling**—we have developed:

- **A predictive model** to determine the likelihood of a successful Falcon 9 first-stage landing.
- **Cost insights** to benchmark competitive launch pricing.

- This information can be used if an alternate company (SpaceY) wants to bid against SpaceX for a rocket launch.

Section 1

Methodology

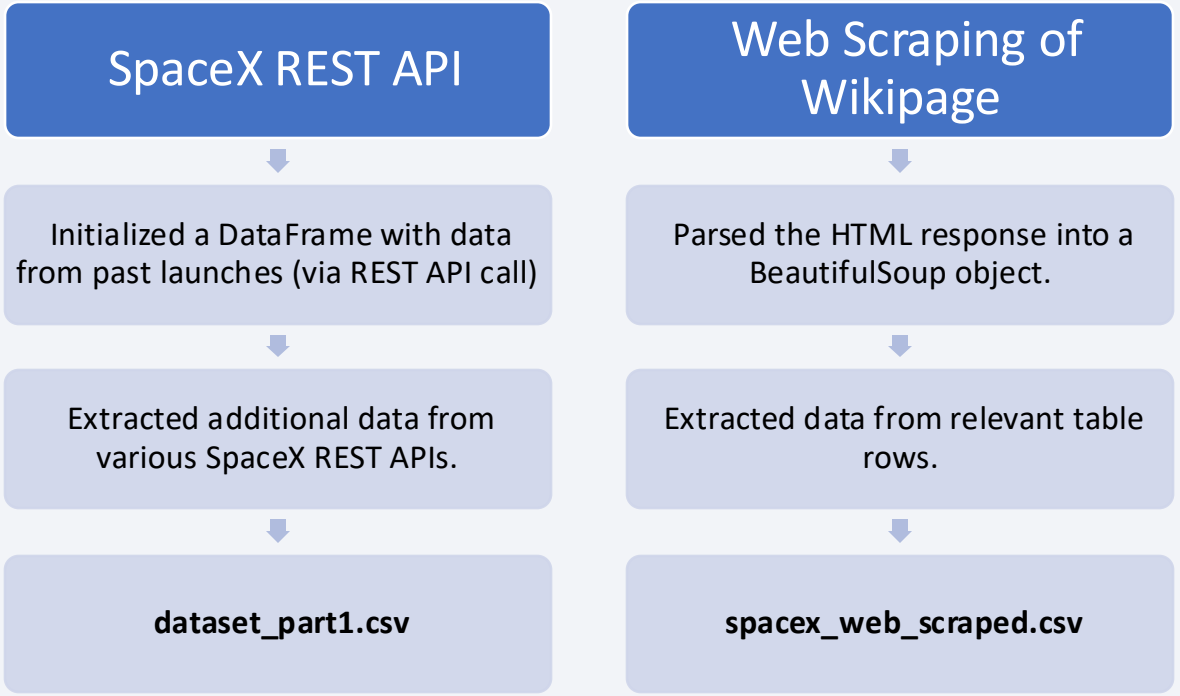
Methodology

Executive Summary

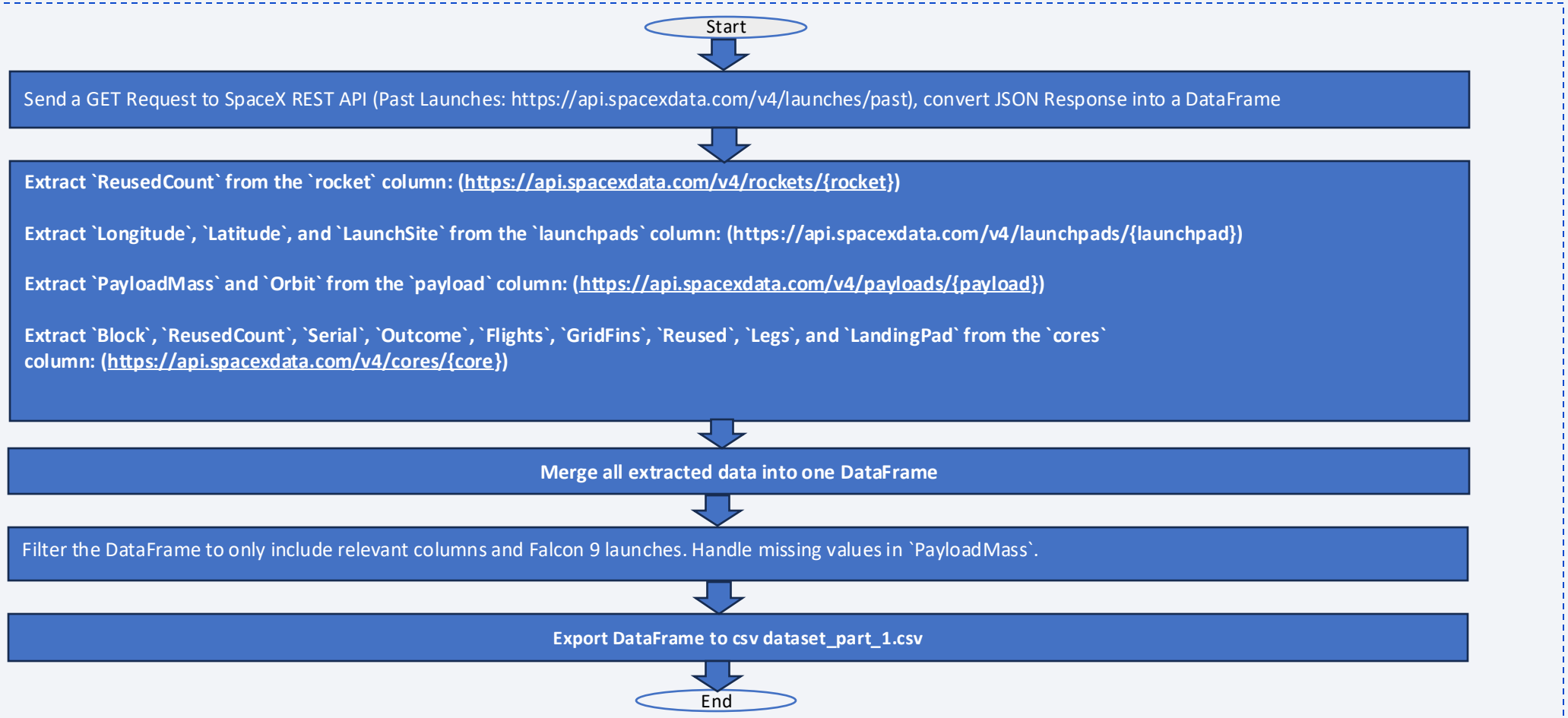
- Data collection methodology:
  - Data collection from SpaceX REST APIs, Web scraping, see detail on Page 7
- Perform data wrangling
  - Dealing with missing values, add a new label, see detail on Page 10
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- The **dataset_part1.csv** was collected from the **SpaceX REST APIs**. A DataFrame was initialized with data from past launches, and additional data (such as BoosterVersion, LaunchSite, PayloadMass, Outcome, and LandingPad) was subsequently extracted from other SpaceX REST APIs.
- The **spacex_web_scraped.csv** was collected through web scraping of a Wikipedia page. The **HTML response** was parsed into a **BeautifulSoup** object, and relevant tables were iterated over to create a DataFrame.

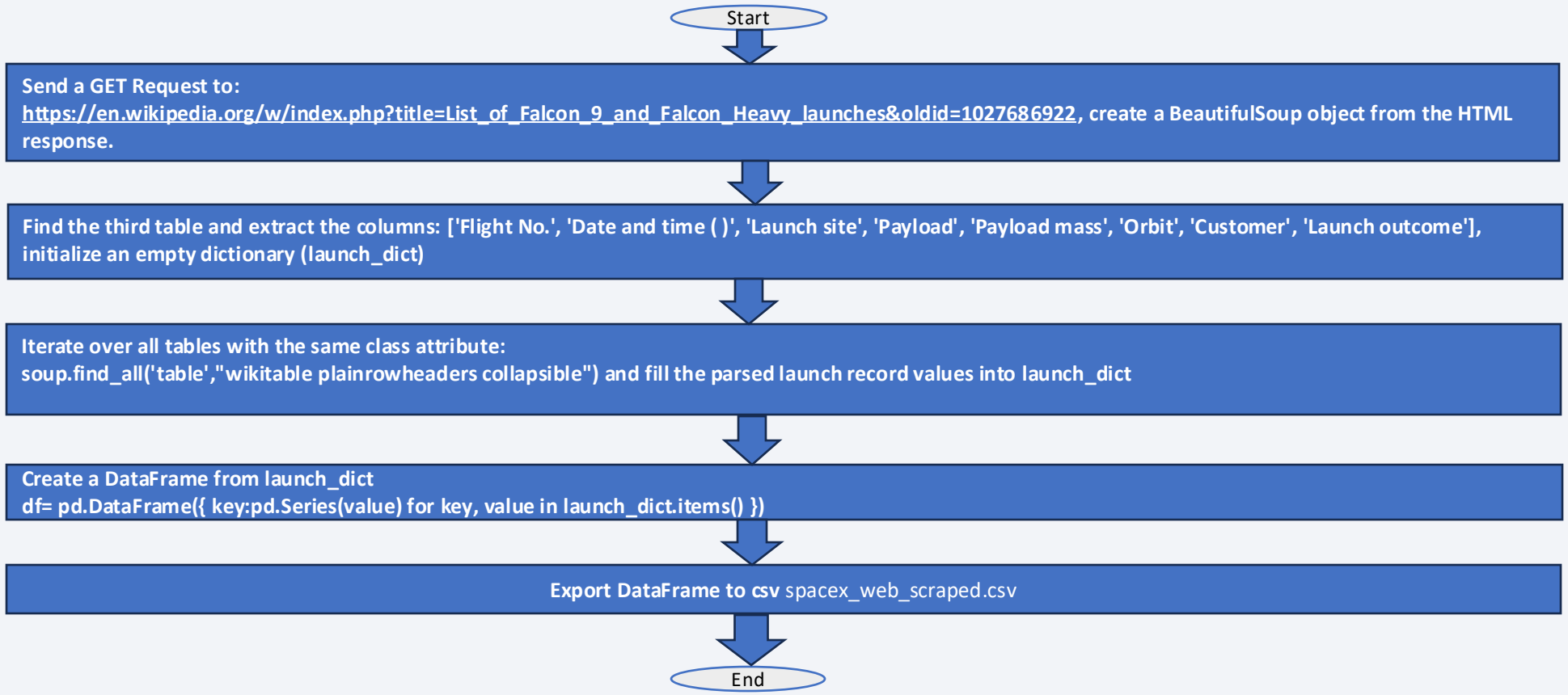


Data Collection – SpaceX API



https://github.com/bigtomate/python_data_science_ML_capstone_project/blob/master/module1_intro/jupyter-labs-spacex-data-collection-api-v2.ipynb

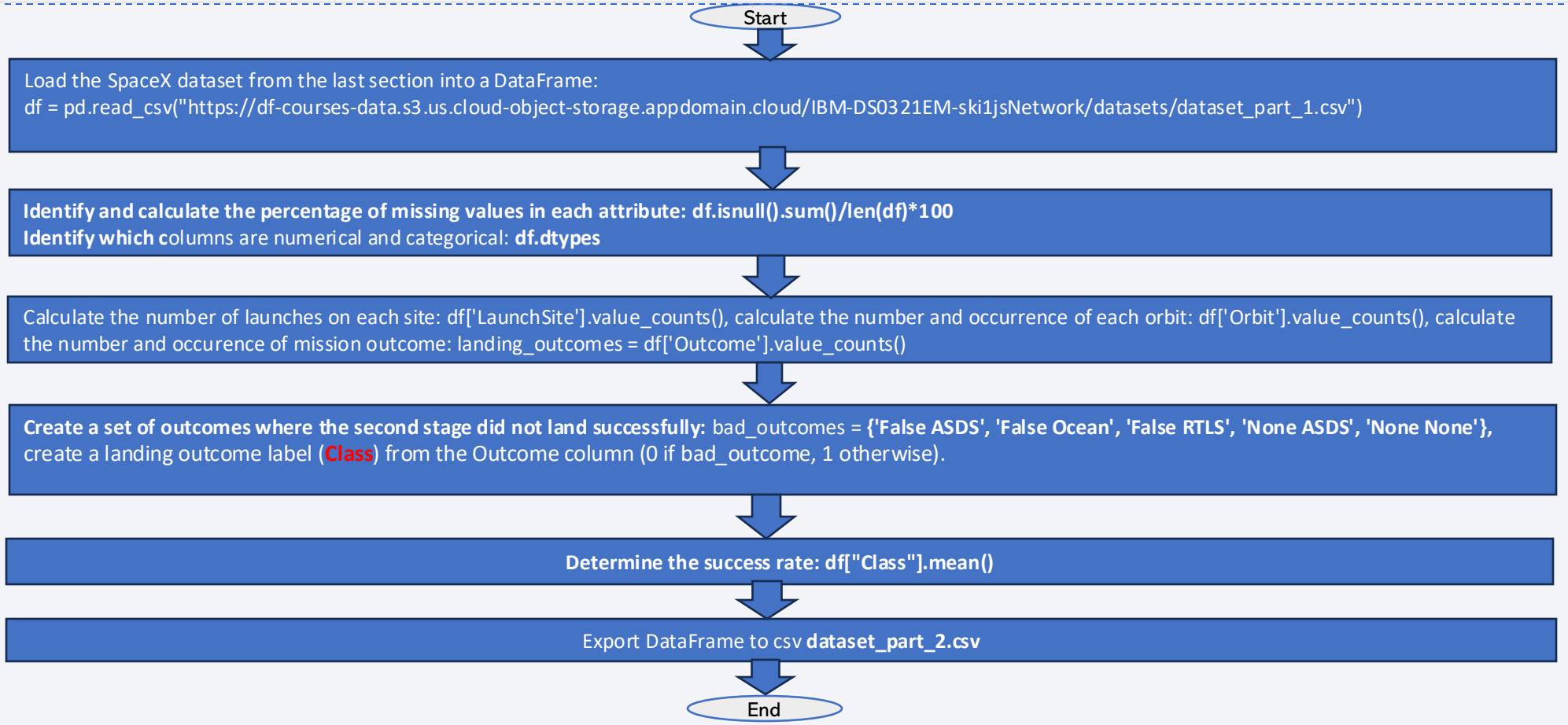
Data Collection - Scraping



https://github.com/bigtomate/python_data_science_ML_capstone_project/blob/master/module1_intro/jupyter-labs-webscraping-v3.ipynb

Data Wrangling

In the previous section, we replaced the missing values in `PayloadMass` with the mean of `PayloadMass`. Now, we will add a label **Class** corresponding to the `Outcome` column.



https://github.com/bigtomate/python_data_science_ML_capstone_project/blob/master/module1_intro/labs-jupyter-spacex-Data%20wrangling-v4.ipynb

EDA with Data Visualization

Exploratory Data Analysis (EDA) was performed using scatter plots (numerical variable), bar plot (categorical variable) to reveal relationships between features and gain preliminary insights into their importance for launch success. These visualizations were used for feature engineering. Using line plot to identify launch success over time.

- Scatter plot charts:

  FlightNumber vs. PayloadMass and overlay the outcome of the launch (success or failure).

  FlightNumber vs. Launch Site and overlay the outcome of the launch.

  FlightNumber vs. Oribit and overlay the outcome of the launch.

  Payload vs. Orbit and overlay the outcome of the launch.

- Bar plot chart: Visualize the relationship between the success rate and each orbit type.
- Line plot: Visualize the average launch success trend over the years.

https://github.com/bigtomate/python_data_science_ML_capstone_project/blob/master/module2_EDA/jupyter-labs-eda-dataviz-v2.ipynb

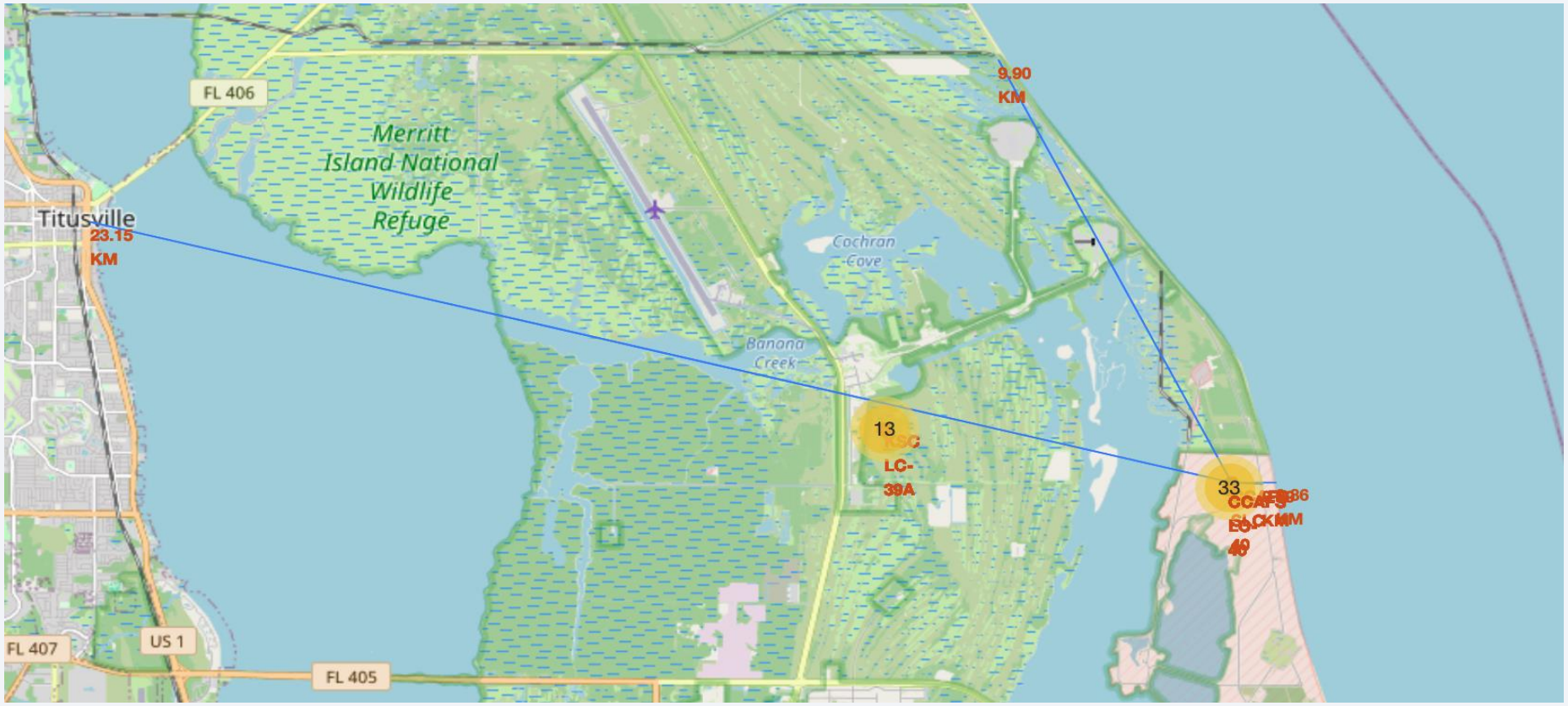
EDA with SQL

- The names of the unique launch sites: %sql SELECT distinct "Launch_Site" from SPACEXTABLE;
- 5 records where launch sites begin with the string 'KSC': %sql SELECT * from SPACEXTABLE where "Launch_Site" like "KSC%" limit 5;
- The total payload mass carried by boosters launched by NASA (CRS): %sql SELECT sum("PAYLOAD_MASS__KG_") as "PAYLOAD_MASS__KG_" from SPACEXTABLE where "Customer" = "NASA (CRS)";
- The average payload mass carried by booster version F9 v1.1: %sql SELECT avg("PAYLOAD_MASS__KG_") as "PAYLOAD_MASS__KG_" from SPACEXTABLE where "Booster_Version" = "F9 v1.1";
- List the date where the succesful landing outcome in drone ship was acheived: %sql SELECT "Date" as "Successful_in_drone_ship_date" from SPACEXTABLE where "Landing_Outcome" = "Success (drone ship)";
- List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000: %sql SELECT "Booster_Version" from SPACEXTABLE where "PAYLOAD_MASS__KG_" > 4000 and "PAYLOAD_MASS__KG_" < 6000 and "Landing_Outcome" = "Success (ground pad)";
- List the total number of successful and failure mission outcomes: %sql SELECT COUNT(CASE WHEN "Mission_Outcome" LIKE 'Failure%' THEN 1 END) AS "Failure", COUNT(CASE WHEN "Mission_Outcome" like "Success%" THEN 1 END) AS "SUC" FROM SPACEXTABLE;
- List the names of the booster_versions which have carried the maximum payload mass: %sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
- List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017:%sql select substr(Date,6,2) as month, "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTABLE where substr(Date,0,5)="2017" and "Landing_Outcome" = "Success (ground pad)";
- Rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order: %sql SELECT "Landing_Outcome", COUNT(*) AS outcome_count, RANK() OVER (ORDER BY COUNT(*) DESC) AS rank FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY rank desc;

https://github.com/bigtomate/python_data_science_ML_capstone_project/blob/master/module2_EDA/jupyter-labs-eda-sql-edx-sqllite-v1.ipynb

Build an Interactive Map with Folium

Use Marker, Circle to label the locations and proximities of launch sites (e.g., coastlines, railroads ect.), and visualize these distances using Polylines. Additionally, MarkerCluster to group markers for better visualization.



https://github.com/bigtomate/python_data_science_ML_capstone_project/blob/master/Module3_Interactive%20Visual%20Analytics%20and%20Dashboard/lab-jupyter-launch-site-location-v1.ipynb

Build a Dashboard with Plotly Dash

Pie charts and Scatter point charts have been added to the dashboard.

- **Pie Chart**:
  - By default, the pie chart displays the **success rate** across all launch sites.
  - When a specific launch site is selected from the dropdown list, the pie chart updates to show the success rate for the selected site.
- **Scatter Plot**:
  - The scatter plot visualizes the relationship between **Payload Mass (kg)** and the target variable **Class** (success or failure).
  - It includes an overlay of the **Booster Version** to provide additional context.
  - By default, the scatter plot includes data from **all launch sites** and focuses on payloads with a mass range of **0 to 10,000 kg**.
  - When a specific launch site is selected, the scatter plot updates to display data only for the selected site.
  - A range slider allows users to adjust the payload mass range, enabling observation of how **Payload Mass** impacts the target variable **Class**.
- **Interactive Components**:
  - **Dropdown List**:
    - Allows users to select a specific launch site.
    - Updates both the pie chart and scatter plot to reflect data for the selected site.
  - **Range Slider**:
    - Enables users to adjust the payload mass range (0 - 10,000 kg by default).
    - Updates the scatter plot to show how changes in payload mass affect the target variable **Class**.

https://github.com/bigtomate/python_data_science_ML_capstone_project/blob/master/Module3_Interactive%20Visual%20Analytics%20and%20Dashboard/spacex_dash_app.py

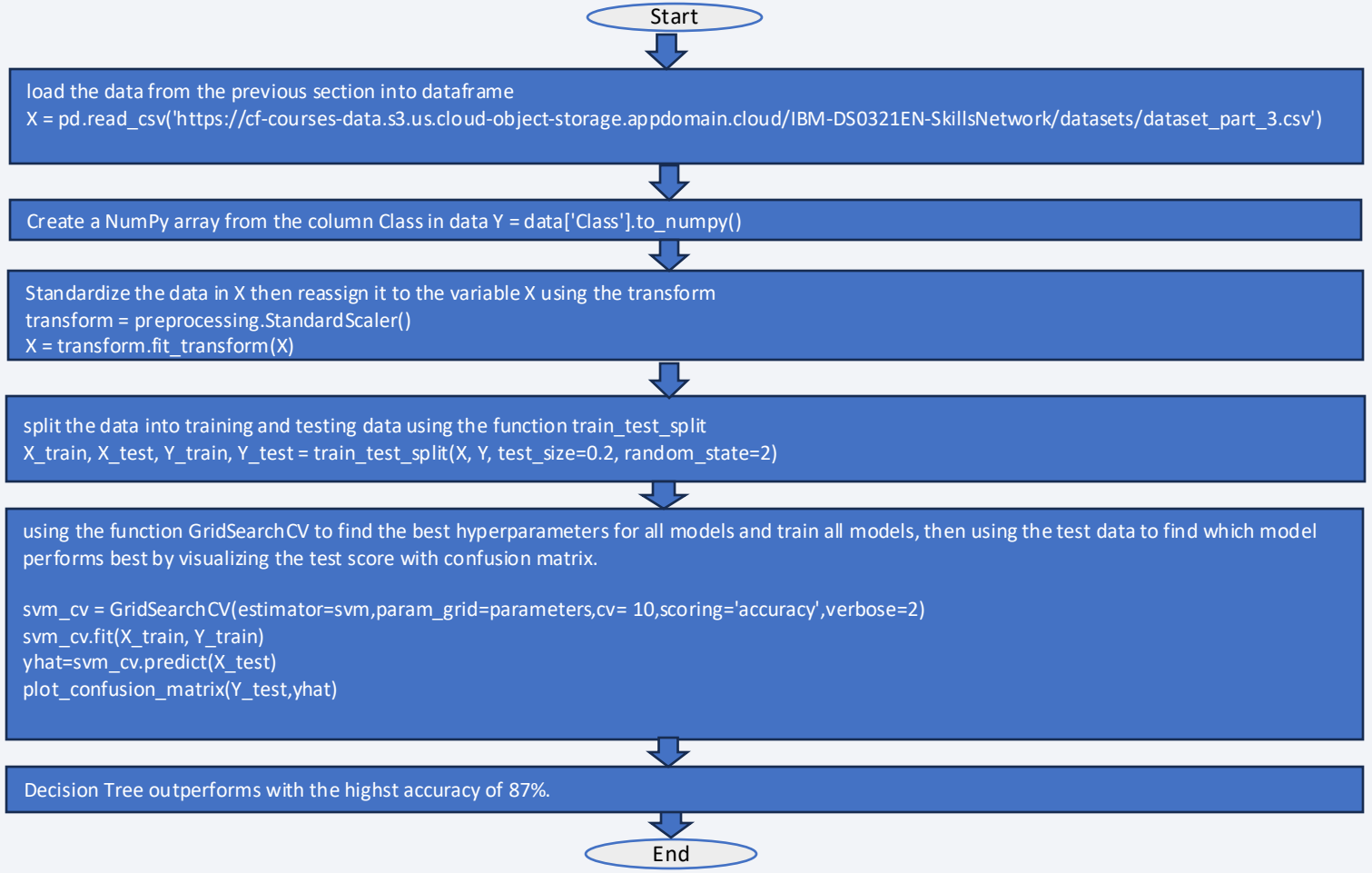
Predictive Analysis (Classification)

Perform exploratory Data Analysis and determine Training Labels

- create a column for the class
- Standardize the data
- Split into training data and test data

Find best Hyperparameter for Logistic Regression,SVM, Decision Tree and KNN

- Find the method performs best using test data

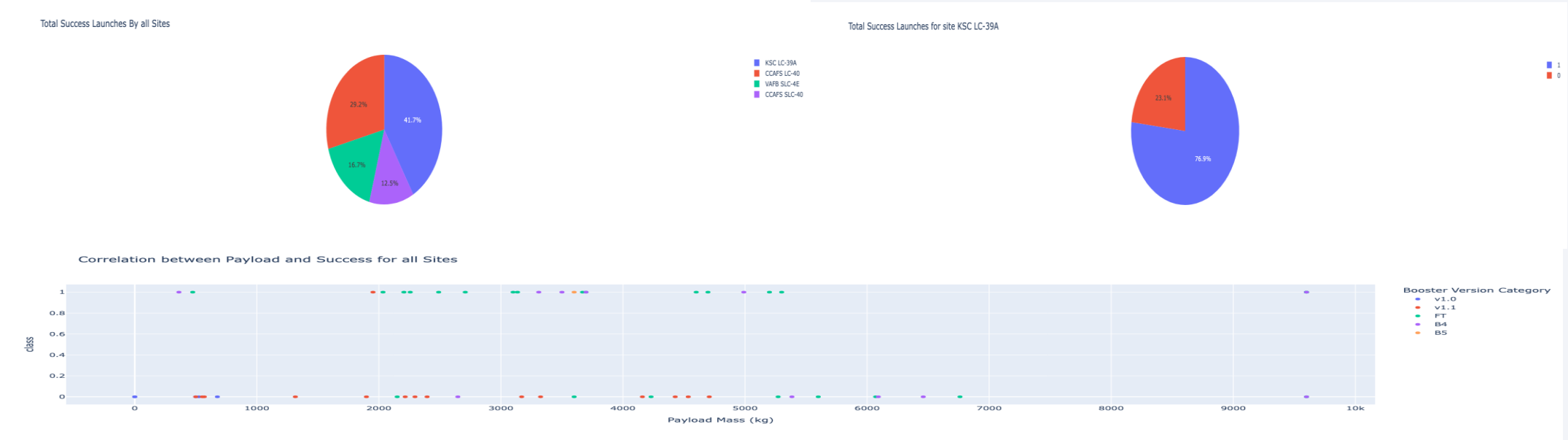


https://github.com/bigtomate/python_data_science_ML_capstone_project/blob/master/module4_Predictive_Analysis_Classification/SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb

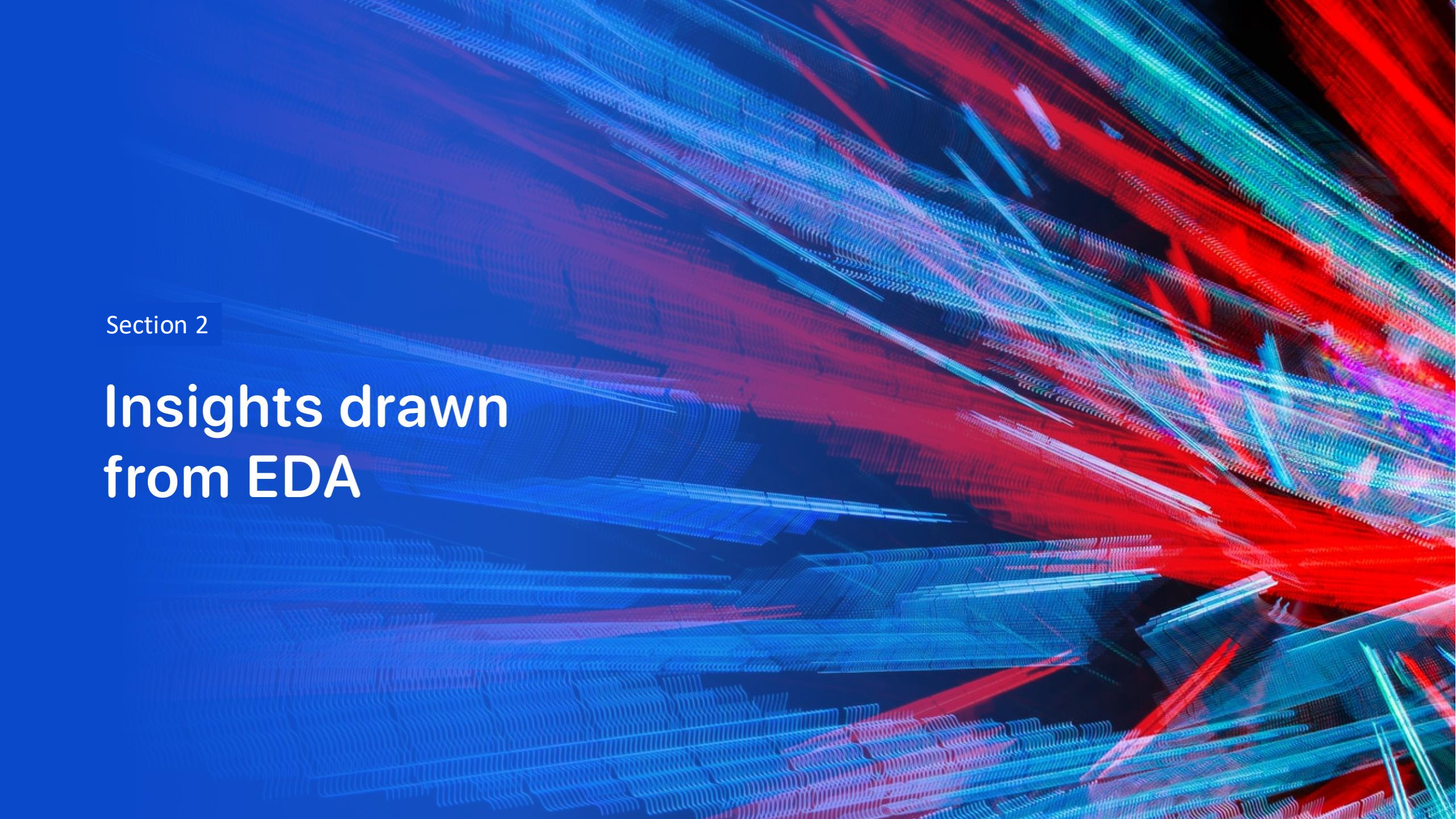
Results

- **Exploratory Data Analysis Results**:
  - **Launch Sites**: Different launch sites had varying success rates, CCAFS SLC-40 with at 60% and KSC LC-39A at 77%.
  - **Payload Mass**: Heavier payloads (above 10,000 kg) are more likely to succeed (Class = 1).
  - **Orbit Types**: Certain orbits (e.g., SSO, VLEO) had higher success rates.

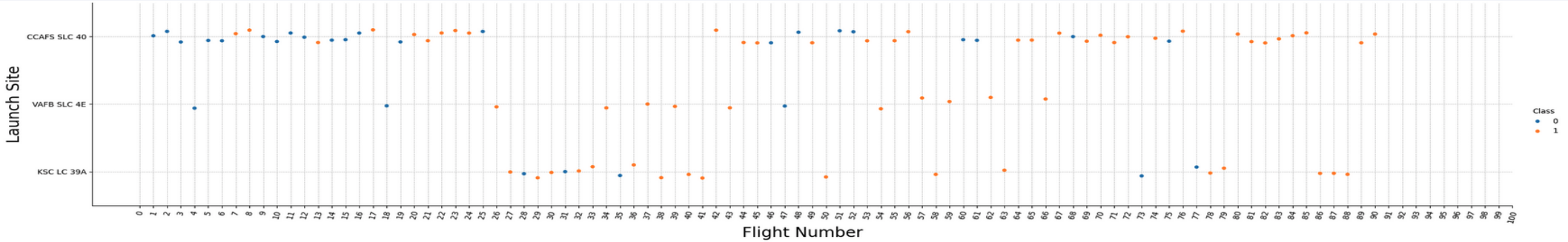
- **Interactive analytics demo in screenshots:**



- **Predictive analysis results:**
  - **Model Performance**: The Decision Tree model had the highest cross-validation accuracy of the 87%.
  - **Test Set Performance**: All models achieved a test set score of 0.83, the Decision Tree model has a test score of 0.89.


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



Key Observations

KSC LS-39A:

- Has the hightst success rate and it is the second most frequently used launch site.

CCAFS SLC-40:

- Has the highest frequency of flights, indicating it is the most frequently used launch site.
- Higher Flight Numbers are more likely to succeed (Class 1), suggesting improved reliability over time.

Anomalies in CCAFS SLC-40 and KSC LC-39A:

- There are few anomalies (e.g., failures) around Flight Numbers 73 to 78 at these sites.
- These anomalies could represent technical issues, experimental missions, or other unique circumstances.

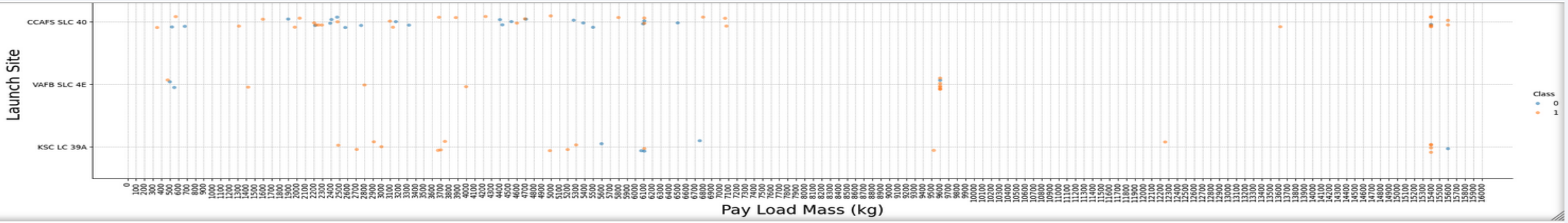
VAFB SLC-4E:

- No flights are observed after Flight Number 66, indicating this site may have been less active or used only for specific missions (e.g., polar orbits) during earlier phases.

Summary

- KSC LS-39A has the hightst success rate and it is the second most frequently used launch site.
- CCAFS SLC-40 is the most active site, with higher Flight Numbers correlating with success.
- Anomalies in CCAFS SLC-40 and KSC LC-39A around Flight Numbers 73 to 78 suggest irregularities in those missions.
- VAFB SLC-4E shows no activity after Flight Number 66, indicating limited usage or a shift in mission focus.

Payload vs. Launch Site



Key Observations

KSC LS-39A:

- Has the hightst success rate and the second most various payload mass range.

CCAFS SLC-40:

- Shows the most variation in payload mass, ranging from low to high values.
- Heavier payloads (e.g., above 10,000 kg) are more likely to succeed (Class 1).

Anomalies in CCAFS SLC-40 and KSC LC-39A: Two anomalies are observed:

- CCAFS SLC-40: A failure (Class 0) at a payload mass of approximately 15,400 kg.
- KSC LC-39A: A failure (Class 0) at a payload mass of approximately 15,600 kg. These anomalies could indicate technical challenges or unique mission conditions at these payload ranges.

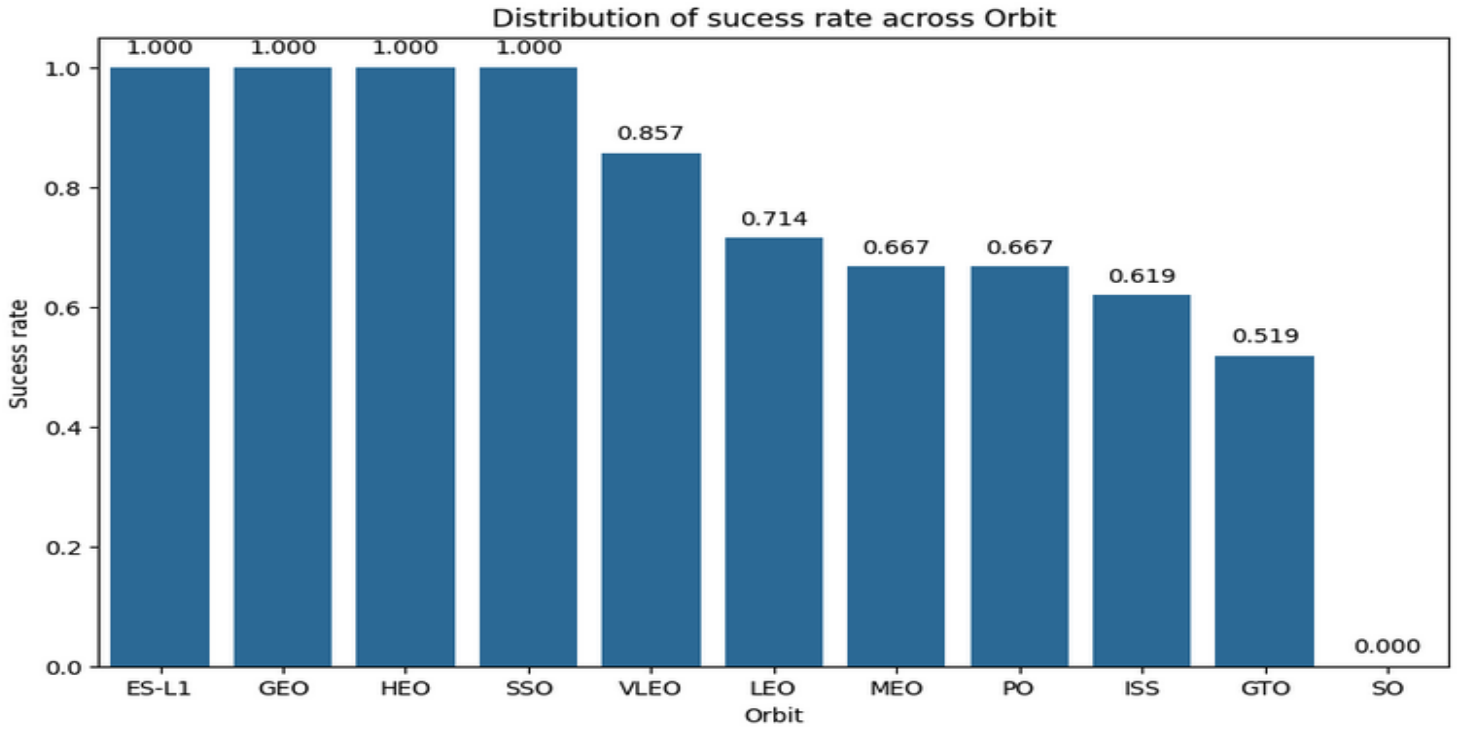
VAFB SLC-4E:

- No rockets were launched with a payload mass greater than 10,000 kg.

Summary

- Heavier payloads (above 10,000 kg) are more likely to succeed (Class 1) However, there are exceptions (anomalies) at very high payload masses (~15,400–15,600 kg), where failures occurred.

Success Rate vs. Orbit Type



Analyze the ploted bar chart try to find which orbits have high sucess rate.

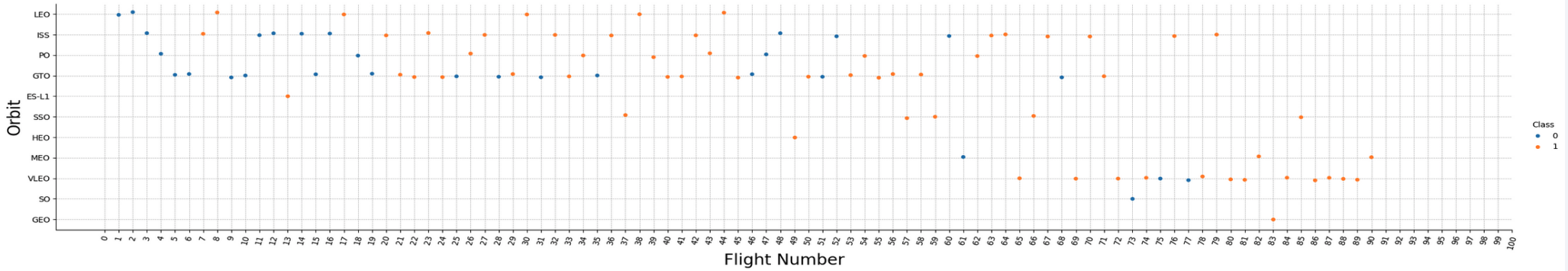
Summary

- High Success Orbits (ES-L1, GEO, HEO, SSO, VLEO): Well-established, reliablel.
- Moderate Success Orbits (LEO, MEO, PO, ISS): Specialized missions with moderate complexity and risk.
- Low Success Orbits (GTO): Advanced or experimental missions with higher risk due to complexity or distance.
- Very Low/Zero Success Orbits (SO): Rarely used or associated with high-risk, experimental missions.

Insights

- The success rate varies significantly across orbits, reflecting differences in mission complexity, technology maturity, and operational challenges.
- Missions to well-established orbits are more reliable, while experimental or advanced orbits pose higher risks.

Flight Number vs. Orbit Type



You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

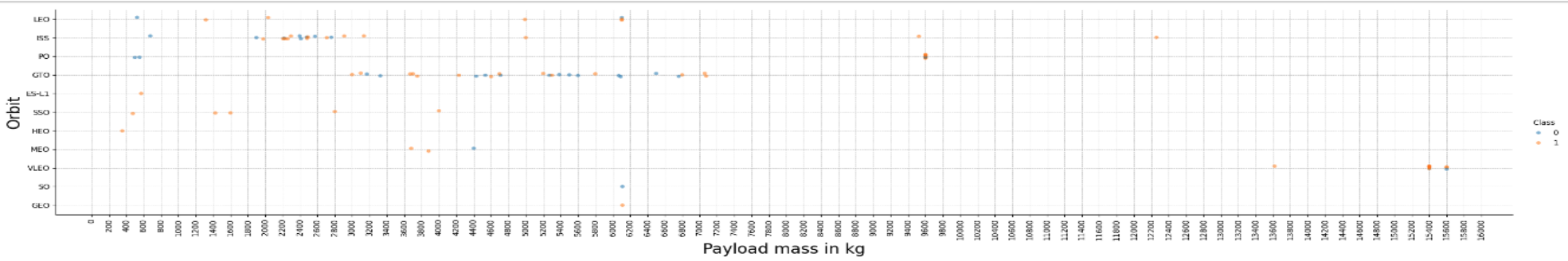
Key Observations

- High Success Orbits :
  - SSO (100% success rate) has no failures.This indicates that SSO missions are highly reliable and well-optimized.
  - VLEO (85% success rate) shows the most frequency of flights with two failures
  - ES-L1, GEO, and HEO (100% success rate): Each has only one flight, making it difficult to generalize their success rates.
- Anomalies in ISS, PO, MEO, and VLEO:
  - These orbits show failures (Class 0) at specific Flight Numbers (e.g., between 60 and 78). PO at 47. These anomalies could indicate technical challenges, unique mission conditions, or experimental missions.
- SO (0% success rate): Only one flight was recorded, and it was a failure (Class 0). This suggests that SO missions may be high-risk or experimental.
- GTO (51% success rate): No clear relationship between Flight Number and success rate. Success and failure are distributed across different Flight Numbers.

Summary

- General Trend: Higher Flight Numbers are more likely to succeed (Class 1), indicating improved reliability and experience over time.
- Exceptions: Failures (Class 0) occur in ISS, PO, MEO, and VLEO orbits, particularly between Flight Numbers 60 and 78. These anomalies highlight specific challenges or risks associated with these orbits.
- SSO stands out as the most reliable orbit, with no failures and the highest frequency of flights.

Payload vs. Orbit Type



Key Observations
- High Success Orbits:
  - SSO (100% success rate): All payloads are less than 4,000 kg, indicating that lighter payloads in SSO are highly reliable.
  - VLEO (85% success rate): with two failures. All payloads are greater than 10,000 kg, suggesting that heavier payloads in VLEO are generally successful but come with some risk.
  - ES-L1, GEO, and HEO (100% success rate): each has only one recorded flight. (ES-L1 and HEO) Payload masses are less than 600 kg, GEO has a payload mass about 6100 kg. Indicating limited data but high reliability for these orbits.

- Anomalies in ISS, PO, MEO, and VLEO:
  - ISS: One failure at a payload mass of ~2,780 kg.
  - PO: One failure at a payload mass of ~9,600 kg.
  - MEO: One failure at a payload mass of ~4,400 kg.
  - VLEO: Two failures at payload masses of ~15,400 kg and ~15,600 kg, highlighting challenges with very heavy payloads in this orbit.

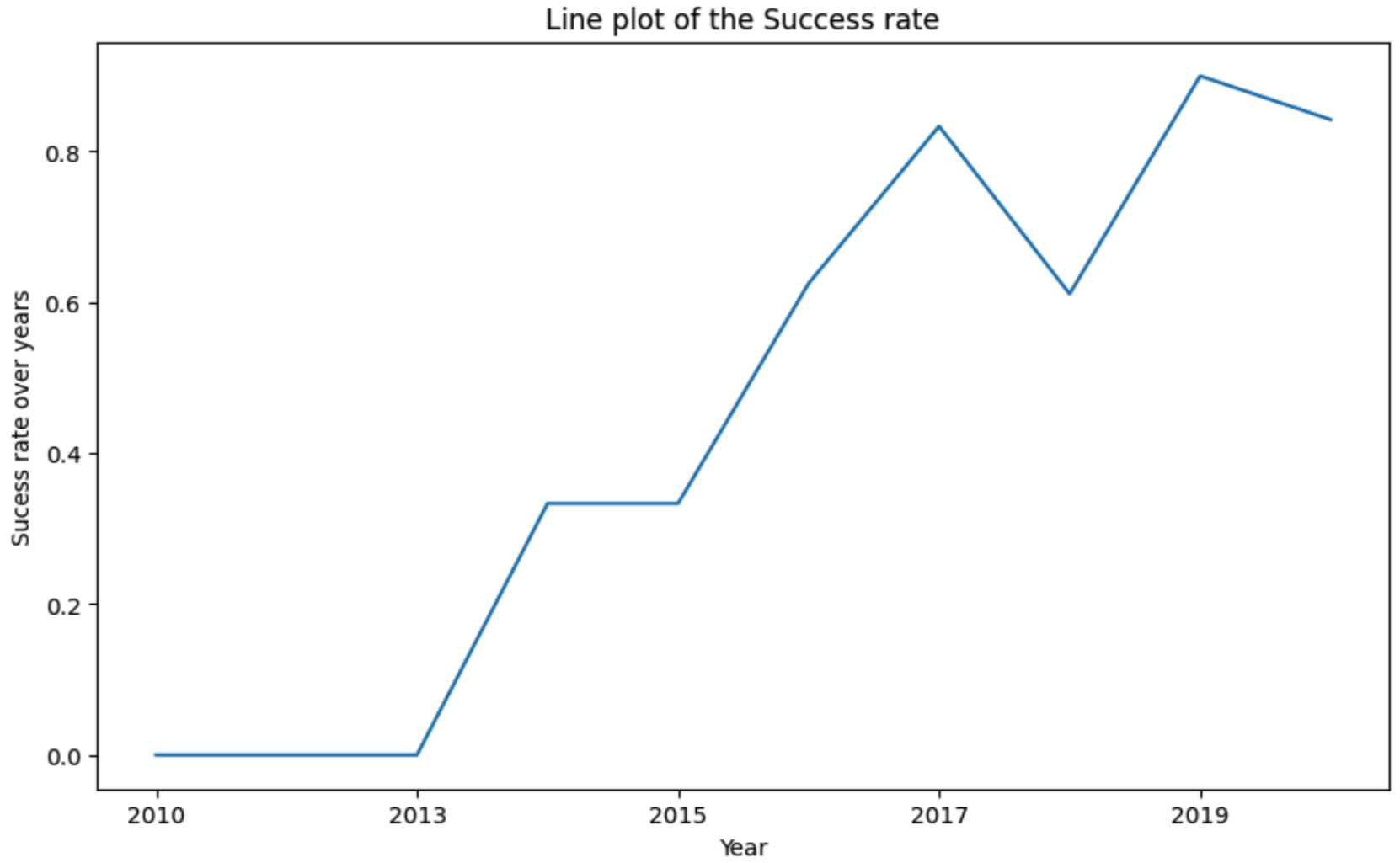
- SO: Only one flight was recorded, with a payload mass of ~6,100 kg, and it was a failure (Class 0). This suggests that SO missions may be high-risk or experimental.

- GTO: No clear relationship between Flight Number and success rate. Successes and failures are distributed across various payload mass value, indicating variability in mission outcomes.

Summary
- General Trend:
  - Heavier payloads (above 10,000 kg) are more likely to succeed (Class 1), especially in orbits like VLEO.However, there are exceptions (anomalies) at very high payload masses (~15,400–15,600 kg), where failures occurred.
- Exceptions: Failures (Class 0) occur in ISS, PO, MEO, and VLEO orbits, particularly in VLEO at very high payload masses (~15,400–15,600 kg). These anomalies highlight specific challenges or risks associated with these orbits and payload ranges.

- Reliable Orbits:
  - SSO stands out as the most reliable orbit for lighter payloads (under 4,000 kg), with no failures.
  - VLEO is also reliable for heavier payloads (above 10,000 kg), despite two failures at very high masses.

Launch Success Yearly Trend



You can observe that the success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.

All Launch Site Names

```
%sql SELECT distinct "Launch_Site" from SPACEXTABLE;
```

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

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

The keyword **DISTINCT** retrieves unique values from a database table column.

Launch Site Names Begin with 'KSC'

```
%sql SELECT * from SPACEXTABLE where "Launch_Site" like "KSC%" limit 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

- **WHERE "Launch_Site" LIKE "KSC%"**:
  - Filters the rows where the Launch_Site column **starts with the string "KSC"**.
  - The LIKE operator is used for pattern matching.
  - The % symbol is a wildcard that matches **any sequence of characters**. So, "KSC%" means "any value that starts with KSC".
- **LIMIT 5**: Restricts the result to **only 5 rows**.

Total Payload Mass

```
%sql select distinct "Customer" from SPACEXTABLE where "Customer" like "NASA%";
```

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

Customer
NASA (COTS) NRO
NASA (COTS)
NASA (CRS)
NASA (LSP) NOAA CNES
NASA (LSP)
NASA (CCD)
NASA (CRS), Kacific 1
NASA (CTS)
NASA (CCDev)
NASA (CCP)
NASA / NOAA / ESA / EUMETSAT

```
%sql SELECT sum("PAYLOAD_MASS__KG_") as "PAYLOAD_MASS__KG_" from SPACEXTABLE where "Customer" like "NASA%";
```

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

PAYLOAD_MASS__KG_
99980

- **SELECT SUM("PAYLOAD_MASS__KG_")**: This calculates the **total sum** of the PAYLOAD_MASS__KG_ column, which represents the payload mass in kilograms.
- **AS "PAYLOAD_MASS__KG_"**: This renames the result column to PAYLOAD_MASS__KG_ for clarity in the output.
- **WHERE "Customer" LIKE "NASA%"**:
  - Filters the rows where the Customer column **starts with the string "NASA"**.
  - The LIKE operator is used for pattern matching.
  - The % symbol is a wildcard that matches **any sequence of characters**. So, "NASA%" means "any value that starts with NASA".

Average Payload Mass by F9 v1.1

```
%sql SELECT avg("PAYLOAD_MASS__KG_") as "PAYLOAD_MASS__KG_" from SPACEXTABLE where "Booster_Version" = "F9 v1.1";
```

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

PAYLOAD_MASS__KG_
2928.4

- **SELECT AVG("PAYLOAD_MASS__KG_")**:
  - This calculates the **average value** of the PAYLOAD_MASS__KG_ column, which represents the payload mass in kilograms.
- **AS "PAYLOAD_MASS__KG_"**:
  - This renames the result column to PAYLOAD_MASS__KG_ for clarity in the output.
- **FROM SPACEXTABLE**:
  - Specifies the table from which to retrieve the data. In this case, the table is named **SPACEXTABLE**.
- **WHERE "Booster_Version" = 'F9 v1.1'**:
  - Filters the rows where the Booster_Version column is exactly equal to "F9 v1.1". This ensures that only missions using the **F9 v1.1 booster version** are considered.

First Successful Ground Landing Date

```
%sql SELECT min("Date") as "First_Successful_Drone_Ship_Date" from SPACEXTABLE where "Landing_Outcome" = "Success (drone ship)";
```

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

First_Successful_Drone_Ship_Date
2016-04-08

- **SELECT MIN("Date")**:
  - Retrieves the **minimum value** of the Date column, which represents the **earliest date** in the dataset.
  - In this context, it finds the **first successful drone ship landing date**.
- **AS "First_Successful_Drone_Ship_Date"**:
  - Renames the result column to First_Successful_Drone_Ship_Date for clarity in the output.
- **WHERE "Landing_Outcome" = 'Success (drone ship)'**:
  - Filters the rows where the Landing_Outcome column is exactly equal to "Success (drone ship)".
- This ensures that only **successful drone ship landings** are considered.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
query = """
SELECT
    "Booster_Version"
FROM
    SPACEXTABLE
WHERE
    "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000 AND "Landing_Outcome" = 'Success (drone ship)';
"""
%sql {query}
```

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

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

- **SELECT "Booster_Version"**:
  - This specifies that we want to retrieve the **Booster_Version** column from the table.
- **WHERE Clause**:
  - This filters the rows based on three conditions:
    - **"PAYLOAD_MASS__KG_" > 4000**: The payload mass must be **greater than 4000 kg**.
    - **"PAYLOAD_MASS__KG_" < 6000**: The payload mass must be **less than 6000 kg**.
    - **"Landing_Outcome" = 'Success (drone ship)'**: The landing outcome must be a **successful drone ship landing**.

Total Number of Successful and Failure Mission Outcomes

```
query = """
SELECT
    COUNT(CASE WHEN "Mission_Outcome" LIKE 'Failure%' THEN 1 END) AS "Failure",
    COUNT(CASE WHEN "Mission_Outcome" LIKE 'Success%' THEN 1 END) AS "SUC"
FROM
    SPACEXTABLE;
"""
%sql {query}
```

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

Failure	SUC
1	100

- **SELECT Clause**:
  - **COUNT(CASE WHEN "Mission_Outcome" = 'Failure (in flight)' THEN 1 END) AS "Failure"**:
    - Counts the number of rows where the Mission_Outcome is exactly "Failure (in flight)".
    - The result is labeled as "Failure".
  - **COUNT(CASE WHEN "Mission_Outcome" LIKE 'Success%' THEN 1 END) AS "SUC"**:
    - Counts the number of rows where the Mission_Outcome starts with "Success" (e.g., "Success", "Success (drone ship)", etc.).
    - The result is labeled as "SUC".

Boosters Carried Maximum Payload

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

```
 * 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

- **SELECT "Booster_Version"**:
  - This specifies that we want to retrieve the **Booster_Version** column from the table.
- **WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)**:
  - This is a **subquery** that filters the rows where the PAYLOAD_MASS__KG_ column is equal to the **maximum payload mass** in the entire table.
  - The subquery (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE) calculates the **maximum value** of the PAYLOAD_MASS__KG_ column.
  - The outer query then retrieves the Booster_Version for the row(s) where the payload mass equals this maximum value.

2015 Launch Records

```
query = """
SELECT
    substr(Date, 6, 2) AS month,
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM
    SPACEXTABLE
WHERE
    substr(Date, 0, 5) = "2017"
    AND "Landing_Outcome" = "Success (ground pad)"
"""
%sql {query}
```

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

month	Landing_Outcome	Booster_Version	Launch_Site
02	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
05	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
06	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
08	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
09	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
12	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

- **SELECT Clause**:
  - substr(Date, 6, 2) AS month: Extracts the month from the Date column. The substr function is used to get a substring from the Date column. Here, it starts at the 6th character and extracts 2 characters, which corresponds to the month part of the date (e.g., "01" for January, "02" for February, etc.).
  - "Landing_Outcome": Selects the column that indicates the outcome of the landing.
  - "Booster_Version": Selects the column that indicates the version of the booster used.
  - "Launch_Site": Selects the column that indicates the site from which the launch occurred.
- **WHERE Clause:**
  - substr(Date, 0, 5) = "2017": Filters the rows to include only those where the year part of the Date column is "2017". The substr function is used to extract the first 5 characters of the Date column, which corresponds to the year.
  - AND "Landing_Outcome" = "Success (ground pad)": Further filters the rows to include only those where the Landing_Outcome is "Success (ground pad)", indicating a successful landing on a ground pad.

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

```
query = """
SELECT
    "Landing_Outcome",
    COUNT(*) AS outcome_count,
    RANK() OVER (ORDER BY COUNT(*) DESC) AS rank
FROM
    SPACEXTABLE
WHERE
    "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY
    "Landing_Outcome"
ORDER BY
    rank DESC;
"""
%sql {query}
```

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

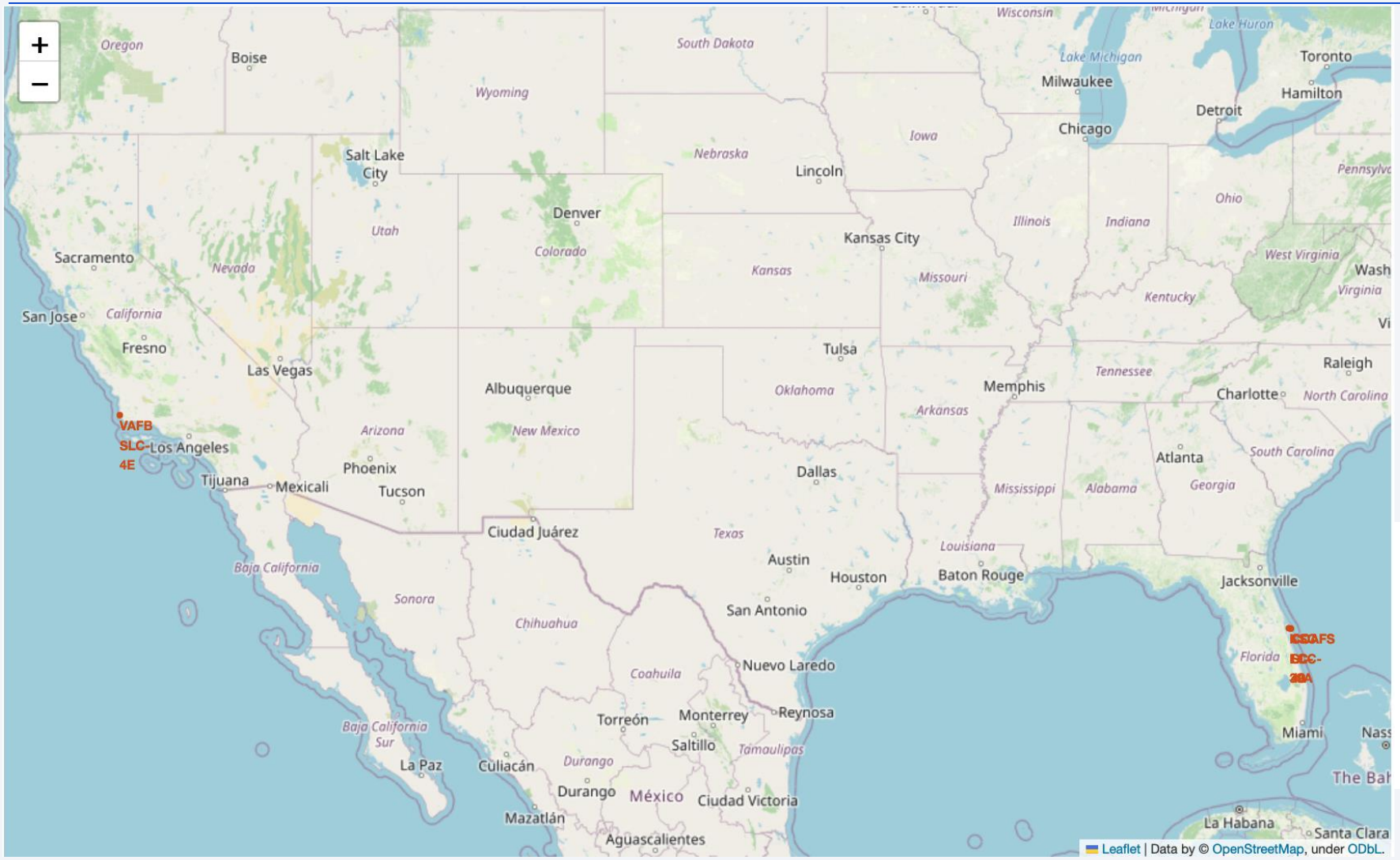
Landing_Outcome	outcome_count	rank
Precluded (drone ship)	1	8
Uncontrolled (ocean)	2	6
Failure (parachute)	2	6
Success (ground pad)	3	4
Controlled (ocean)	3	4
Success (drone ship)	5	2
Failure (drone ship)	5	2
No attempt	10	1

- **Filters** rows where the Date is between **2010-06-04** and **2017-03-20**.
- **Groups** the data by Landing_Outcome and **counts** the occurrences of each outcome.
- **Ranks** the outcomes based on their count in **descending order** (most frequent gets rank 1).
- **Returns** the outcomes, their counts, and their ranks, sorted by rank (most frequent first).

Section 3

Launch Sites Proximities Analysis

SpaceX Launch Sites



The screenshot shows four launch sites. **Vandenberg Space Force Base (VAFB) SLC-4E** is located on the west coast of the United States, near Lompoc, **California**. On the east coast, three launch sites are clustered in **Florida**: **Cape Canaveral Space Force Station (CCAFS) LC-40**, **Kennedy Space Center (KSC) LC-39A** and **(CCAFS) SLC-40**.

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

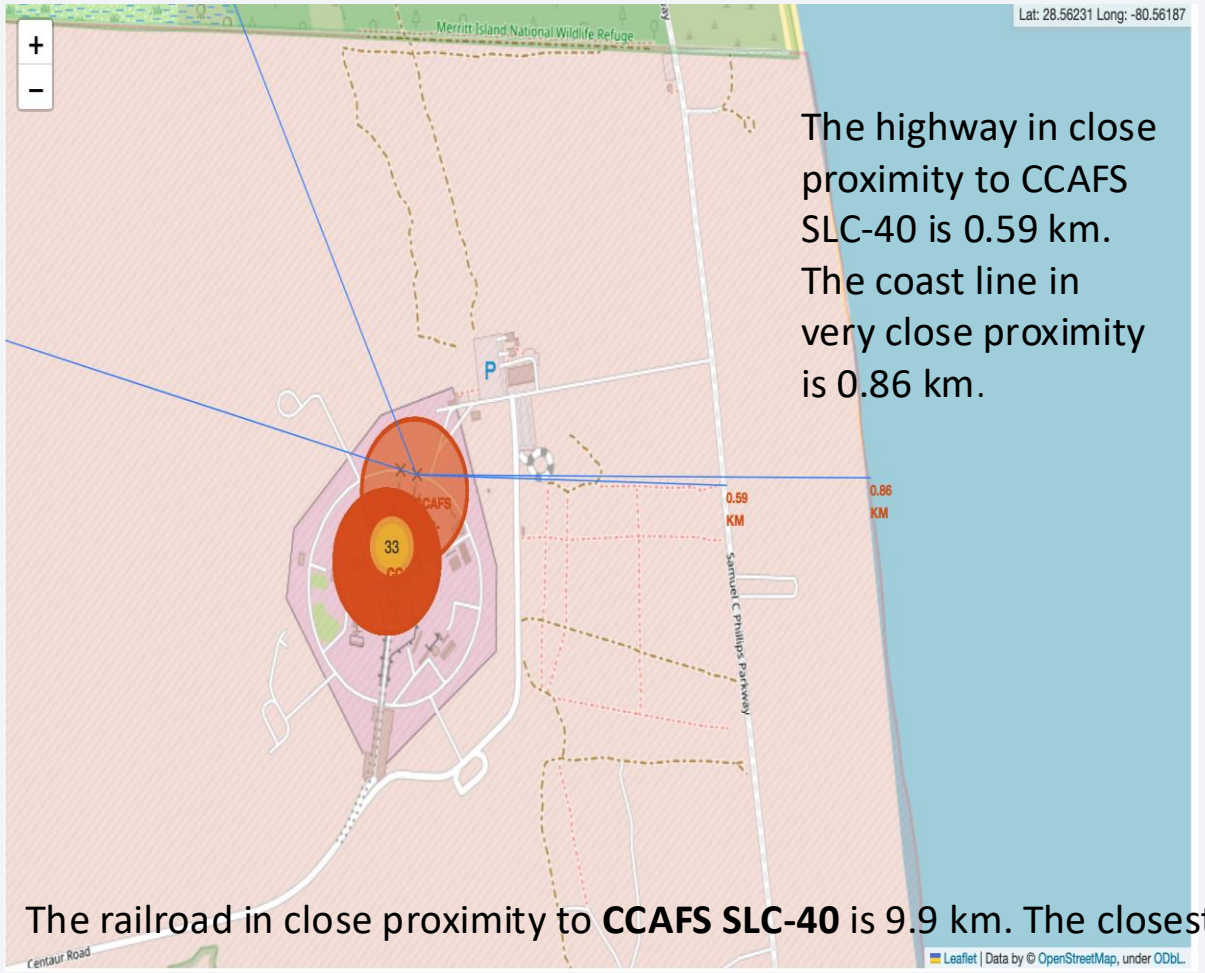
Launch outcomes of CCAFS SLC-40 and CCAFS LC-40

The screenshot displays the launch outcomes for the **CCAFS SLC-40** launch site. A total of **7** launches are marked, with **3** successful launches (**Class 1**, marked in green) and 4 failed launches (**Class 0**, marked in red). Additionally, a yellow marker cluster indicates **26** launches at **CCAFS LC-40**, providing a visual overview of the total number of launches and their success rates.

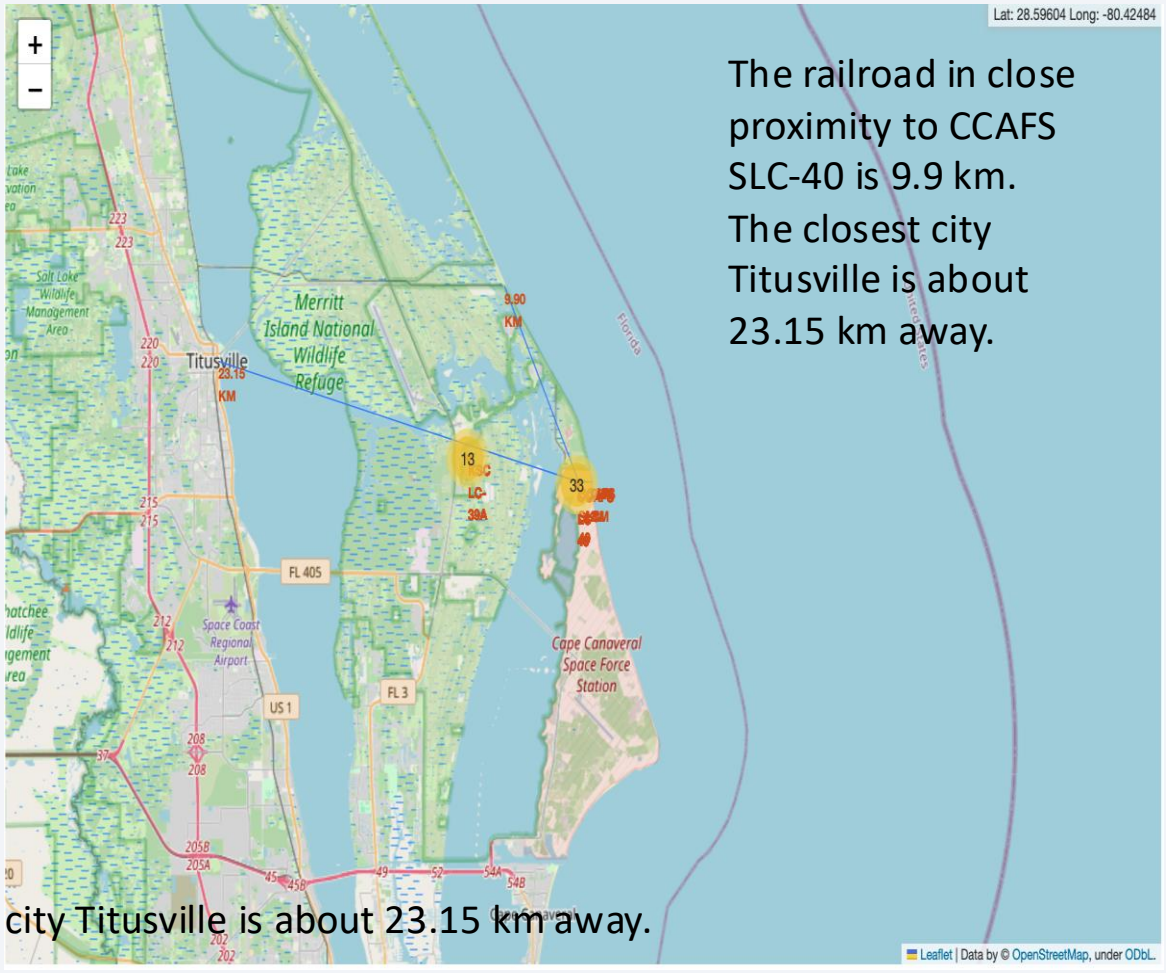
```
spacex_df[spacex_df['Launch Site'] == 'CCAFS SLC-40']
```

	Launch Site	Lat	Long	class	marker_color
49	CCAFS SLC-40	28.563197	-80.57682	1	green
50	CCAFS SLC-40	28.563197	-80.57682	1	green
51	CCAFS SLC-40	28.563197	-80.57682	0	red
52	CCAFS SLC-40	28.563197	-80.57682	0	red
53	CCAFS SLC-40	28.563197	-80.57682	0	red
54	CCAFS SLC-40	28.563197	-80.57682	1	green
55	CCAFS SLC-40	28.563197	-80.57682	0	red

The proximities to launch site CCAFS SLC-40

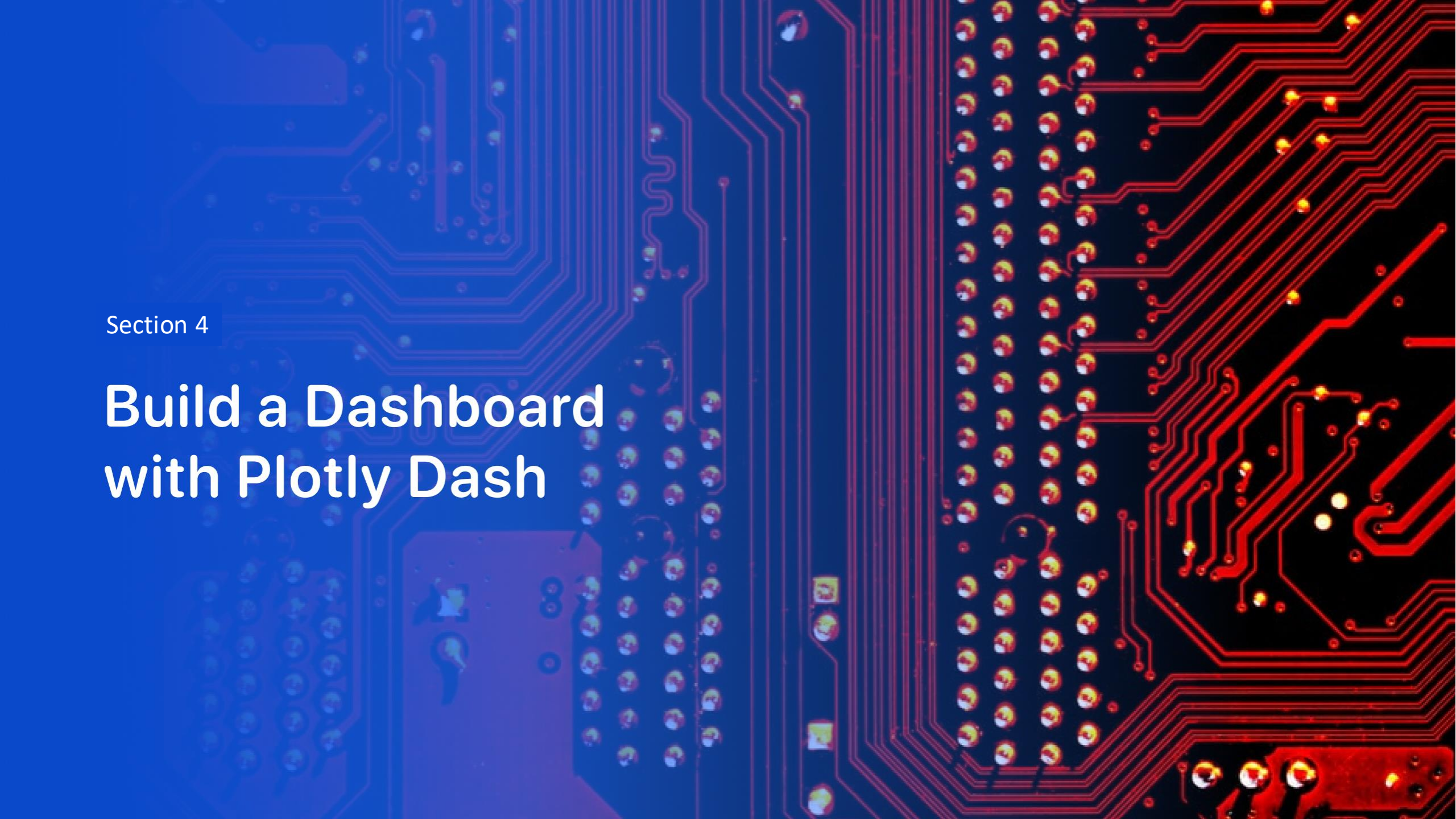


The highway in close proximity to CCAFS SLC-40 is 0.59 km. The coast line in very close proximity is 0.86 km.

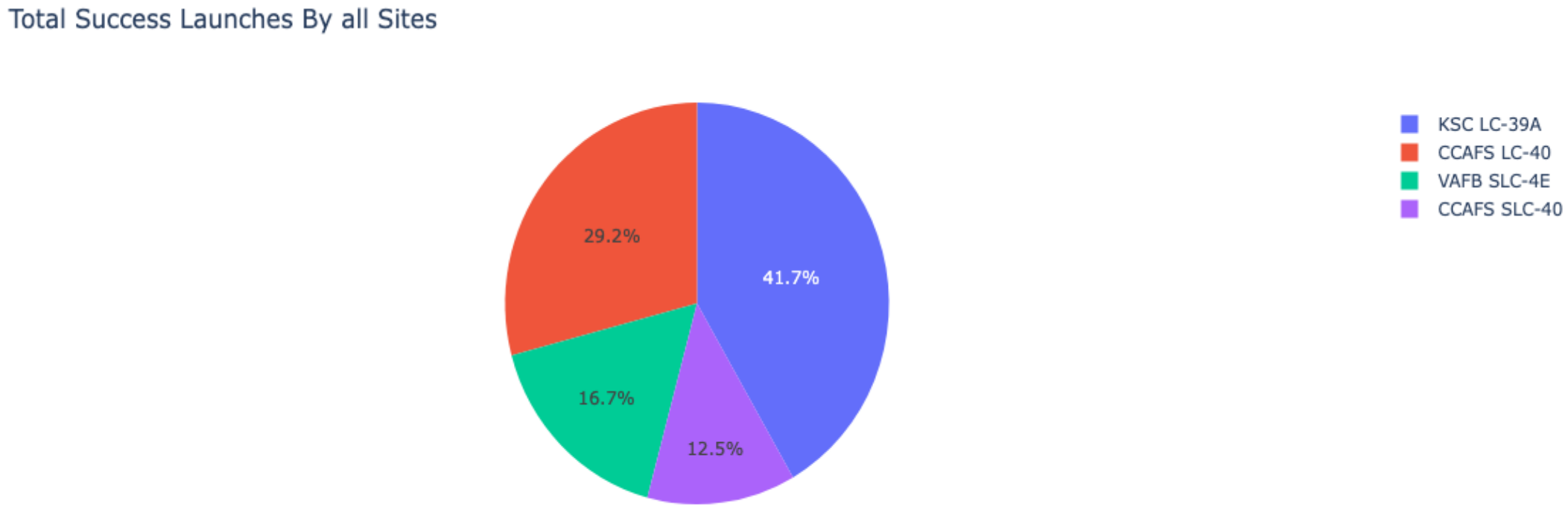


The railroad in close proximity to CCAFS SLC-40 is 9.9 km. The closest city Titusville is about 23.15 km away.

The railroad in close proximity to **CCAFS SLC-40** is 9.9 km. The closest city Titusville is about 23.15 km away.

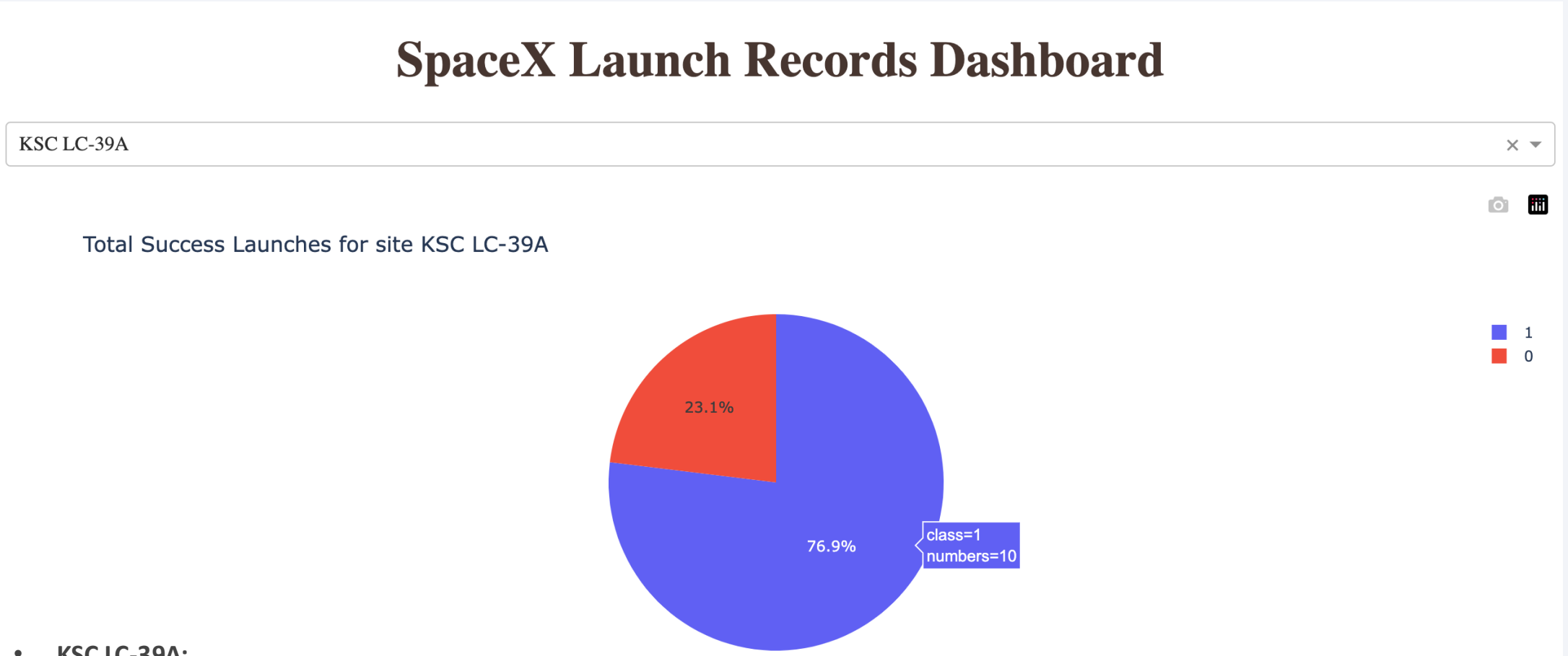

Section 4
Build a Dashboard with Plotly Dash

Dashboard Pie chart success rate of all sites



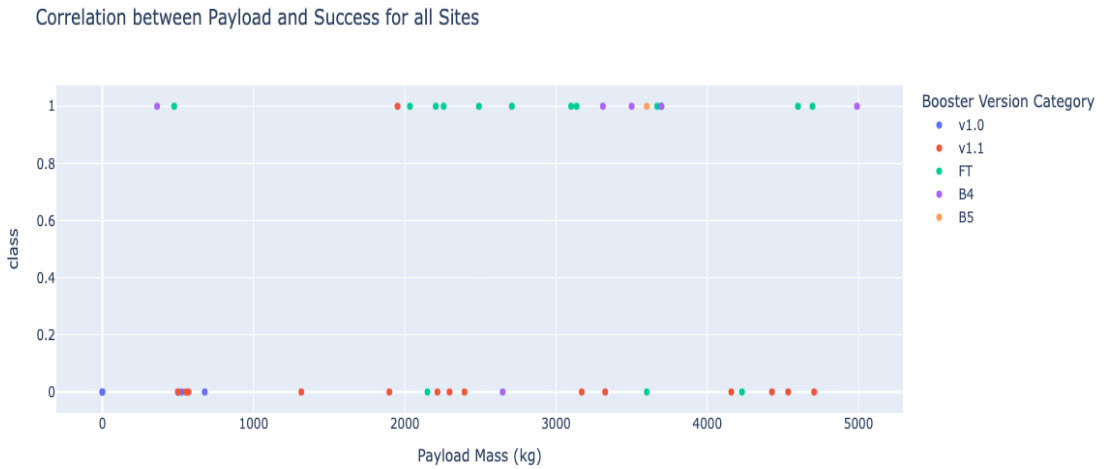
- **KSC LC-39A**:
  - Success rate: **41.7% (**10 successful launches out of 24 total successful launches **)**
  - Total launches: Fewer than CCAFS LC-40
- **CCAFS LC-40**:
  - Success rate: **29.2%**
- **Key Observation**:
  - **KSC LC-39A** has a higher success rate (41.7%) compared to **CCAFS LC-40** (29.2%).
  - Despite having fewer total launches, **KSC LV-39A** outperforms **CCAFS LC-40** in terms of success rate.
- **Conclusion: KSC LC-39A** has a higher success rate with fewer launches compared to **CCAFS LC-40**.

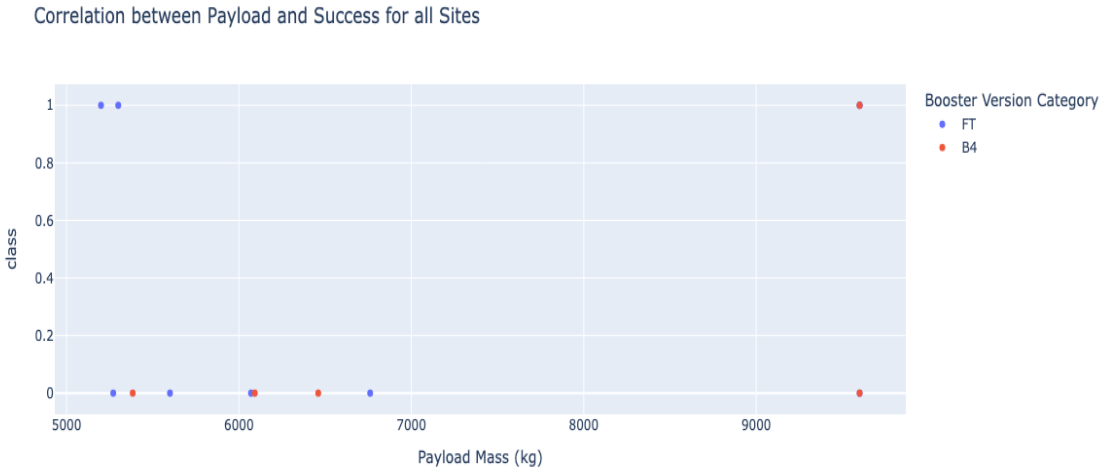
Dashboard Pie Chart highest launch success ratio



- **KSC LC-39A:**
  - Success rate all sites : **41.7%** overall successful launches of all sites, **KSC LC-39A** has the highest launch success ratio.
  - Success rate **76.9%** (10 successful launches out of 13 total launches)

Payload vs. Launch Outcome overlay the Booster Version





Key Observations:

- **Overall Correlation Between Payload and Success Rate**
  - The success rate generally decreases as payload mass increases, particularly beyond 5000 kg.
  - Lighter payloads (below 5000 kg) tend to have higher launch success rates.
- **Booster-Specific Performance:**
  - **FT Booster:**
    - Demonstrates **the largest success rate** in the **350–5000 kg payload range**.
  - **B4 and B5 Boosters:**
    - **B5** has a **100% success rate**, but this is based on only **a single launch**, making it statistically unreliable.
    - **B4** performs well in the **350–5000 kg range** but shows **no successful launches above 5000 kg**.

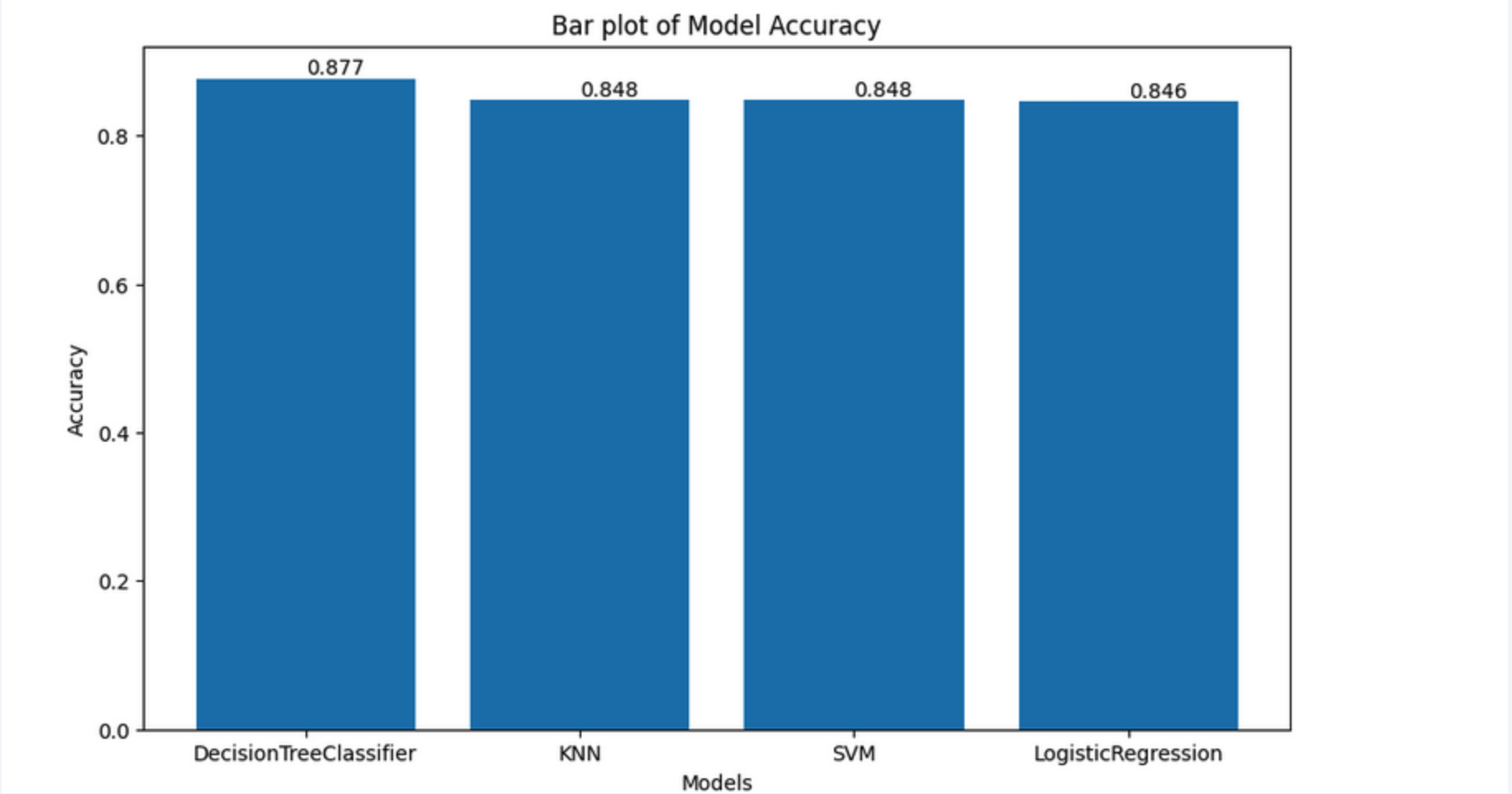
Conclusion:

- **Optimal Payload Range:** For the highest probability of success, payloads should ideally stay within **350–5000 kg**, especially when using the **FT booster**.
- **Heavier Payloads (>5000 kg):** Current data suggests a higher risk of failure, particularly with the B4 booster.
- **B5's Perfect Record:** While promising, more launches are needed to confirm its reliability.

Section 5

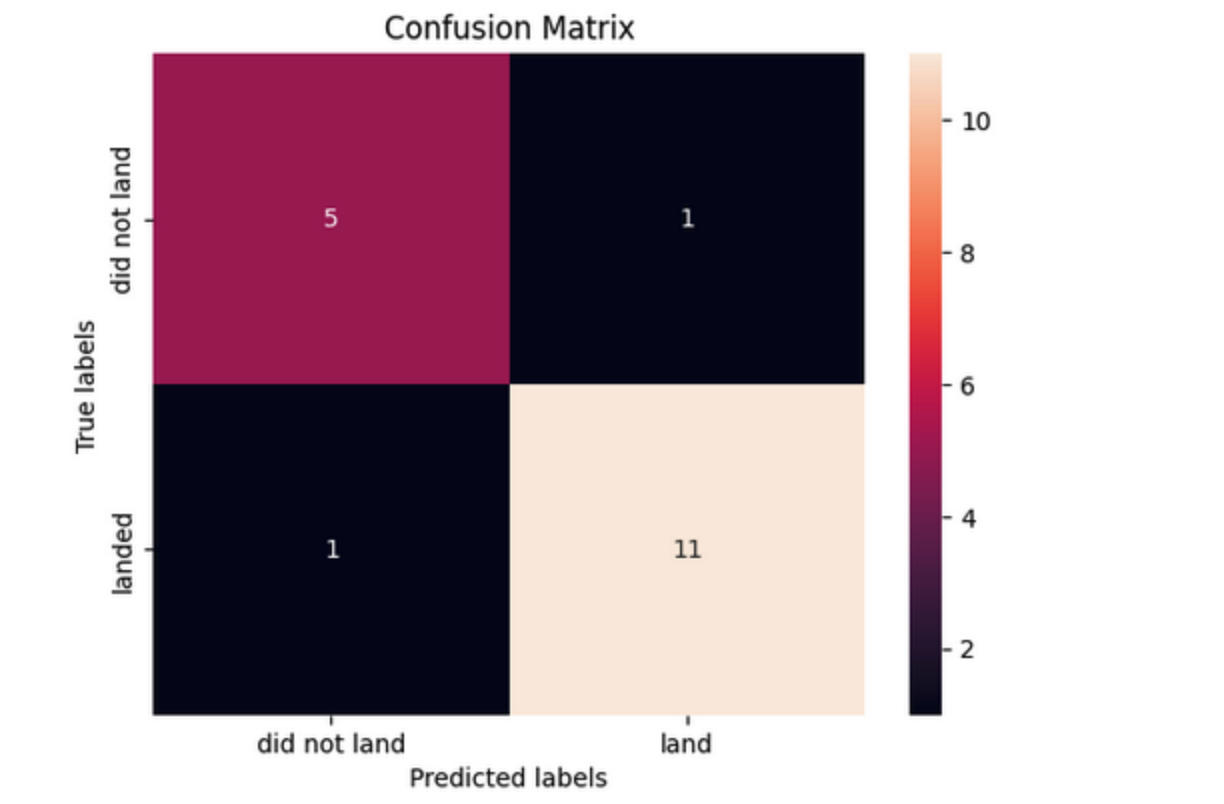
Predictive Analysis (Classification)

Classification Accuracy



- Decision Tree classification model has the highest classification accuracy.

Confusion Matrix



Y_test
1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1
Y_hat
1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1
The Confusion matrix represents the Y_test and Y_hat values from **Decision Tree Classification model**. Which has the best performance.

Conclusions

Launch Site Performance

- **KSC LC-39A** demonstrated the highest success rate across all payload mass ranges (lightweight to heavy), particularly excelling with heavy payloads (>10,000 kg).
- **CCAFS SLC-40**, the most active launch site, maintained moderate success rates but showed consistent performance with heavy payloads.
- **VAFB SLC-4E** achieved high success rates but had no missions after Flight 66 and no heavy payload launches.
- **Trend**: Higher payload masses (>10,000 kg) correlated with increased success rates, suggesting optimized engineering for heavy missions.

Orbit Type Reliability

- **SSO** : 100% success rate, indicating exceptional mission reliability for lightweight payload mass (< 4,000 kg).
- **VLEO** : 85% success rate, reflecting strong operational optimization for heavy payload mass (> 10,000 kg).

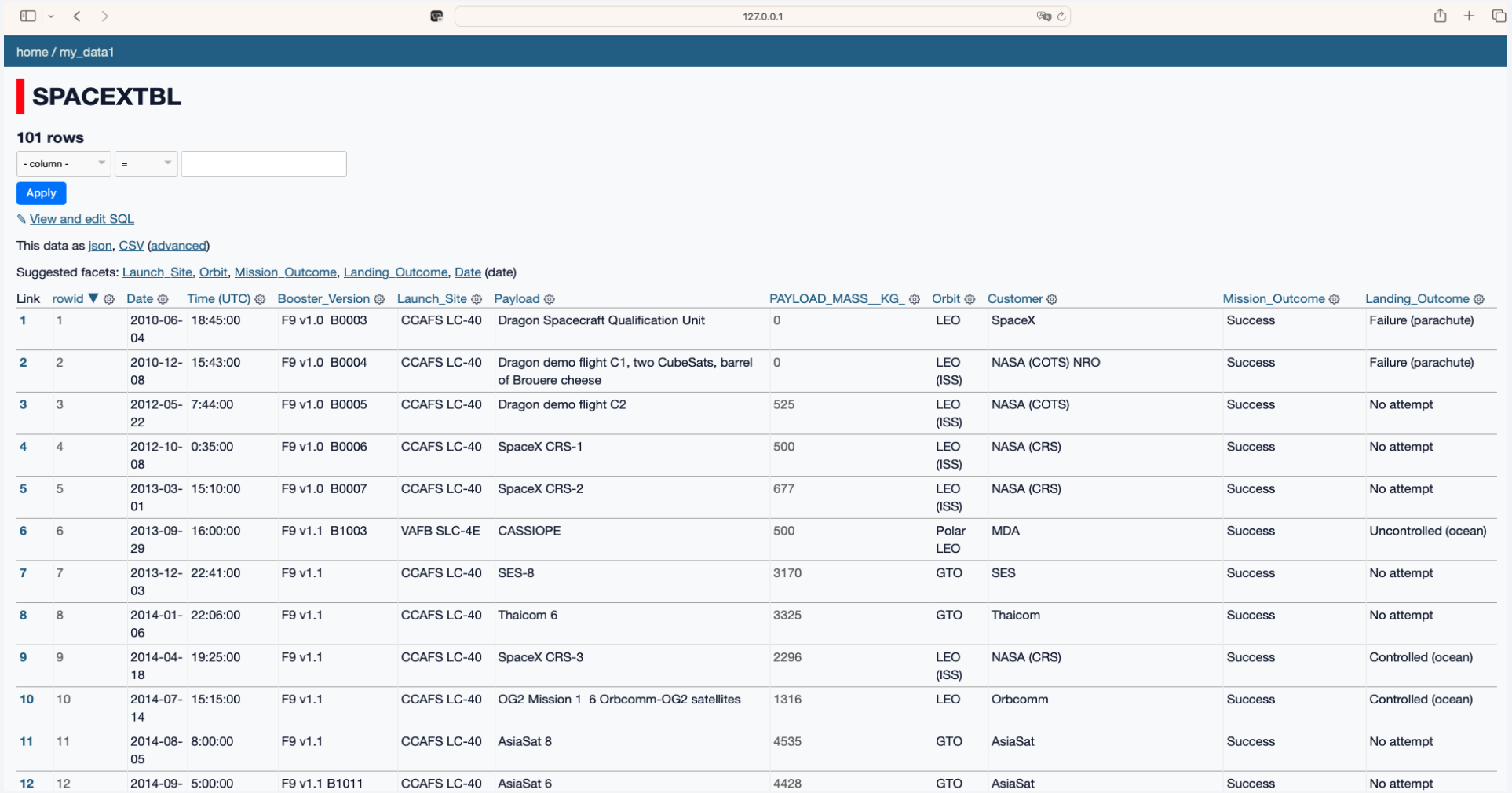
Predictive Modeling

- The **Decision Tree model** outperformed all other classifiers (accuracy: 87%), validating its robustness for launch success prediction.

Historical Improvement

- Success rates have **steadily increased since 2013**, reflecting SpaceX's advancements in booster reusability and launch optimization.

Appendix



127.0.0.1

home / my_data1

SPACEXTBL

101 rows

- column - | = | Apply

View and edit SQL

This data as json, CSV (advanced)

Suggested facets: Launch_Site, Orbit, Mission_Outcome, Landing_Outcome, Date (date)

Link	rowid	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
1	1	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2	2	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)
3	3	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
4	4	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
5	5	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
6	6	2013-09-29	16:00:00	F9 v1.1 B1003	VAFB SLC-4E	CASSIOPE	500	Polar LEO	MDA	Success	Uncontrolled (ocean)
7	7	2013-12-03	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt
8	8	2014-01-06	22:06:00	F9 v1.1	CCAFS LC-40	Thaicom 6	3325	GTO	Thaicom	Success	No attempt
9	9	2014-04-18	19:25:00	F9 v1.1	CCAFS LC-40	SpaceX CRS-3	2296	LEO (ISS)	NASA (CRS)	Success	Controlled (ocean)
10	10	2014-07-14	15:15:00	F9 v1.1	CCAFS LC-40	OG2 Mission 1 6 Orbcomm-OG2 satellites	1316	LEO	Orbcomm	Success	Controlled (ocean)
11	11	2014-08-05	8:00:00	F9 v1.1	CCAFS LC-40	AsiaSat 8	4535	GTO	AsiaSat	Success	No attempt
12	12	2014-09-	5:00:00	F9 v1.1 B1011	CCAFS LC-40	AsiaSat 6	4428	GTO	AsiaSat	Success	No attempt

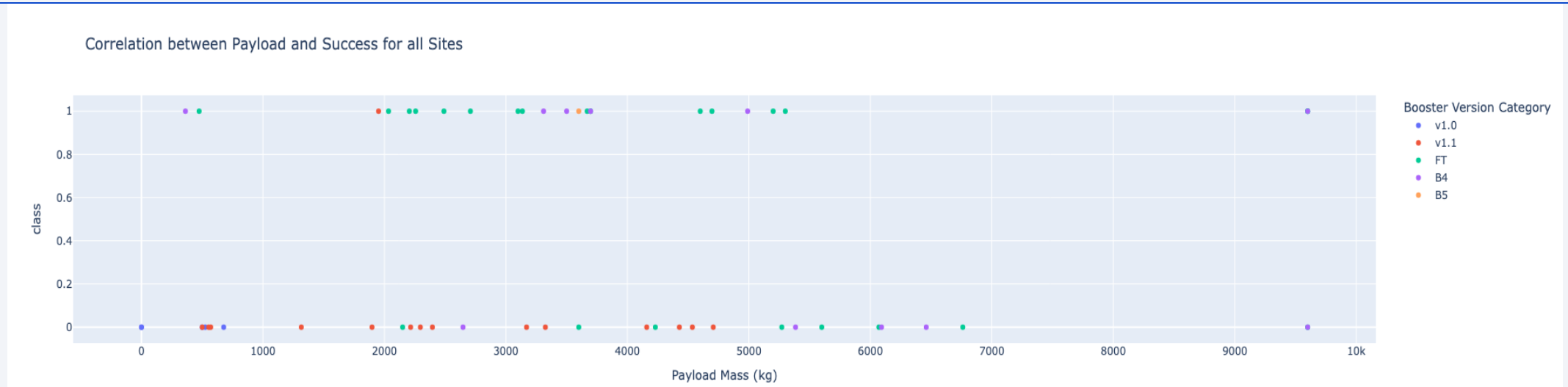
Alternative to sql magic command using sql queries on a database UI : https://github.com/simonw/datasette

Appendix

```
[66]: # Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
plt.figure(figsize=(20,10))
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5, alpha = 0.5)
plt.grid(True, which='both', axis='both', linestyle='--', linewidth=0.5)
plt.xticks(ticks=np.arange(0, 16001, 100), rotation = 90)
plt.xlabel("Pay Load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

Using **xticks** ticks attribute and **grid** for better readable payload mass on x-axis in this case. Using **alpha = 0.5** to avoid the overlapping.

Appendix




```python
def get_fig(dataframe, entered_site):
    payload_xaxis = [0, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000]
    fig = px.scatter(
        dataframe,
        x="Payload Mass (kg)",
        y="class",
        color="Booster Version Category",
        title=f"Correlation between Payload and Success for {entered_site}",
    )
    fig.update_xaxes(
        tickvals=payload_xaxis,
        title_text='Payload Mean (kg)'
    )

    fig.update_layout(
        xaxis_title="Payload Mass (kg)",
        yaxis_title="class",
        showlegend=True
    )
    return fig
```

Using **ticksvals** in **update_axes** for better readable payload mass on x-axis in this case.

Thank you!