IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ali Taheri
Sep 12, 2024

Outline

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

Executive Summary

- The analysis of the SpaceX Falcon 9 launch data provides key insights into factors influencing the success of the first-stage landings. A logistic regression model was developed to predict the probability of a successful landing based on various features such as payload mass, orbit, and launch site. Here are the primary takeaways:

1. **Launch Site Success Rates**: Among the launch sites analyzed, CCAFS SLC 40 and KSC LC 39A demonstrated the highest success rates for Falcon 9 landings, indicating a potential operational advantage at these locations.
2. **Relationship Between Payload Mass and Landing Success**: There was a noticeable trend where lighter payloads were more likely to result in successful landings, though this relationship becomes more complex with other factors such as orbit type and whether the booster was reused.
3. **Trends Over Time**: Over the years, SpaceX has significantly improved its landing success rates, especially as the technology for reusing the first-stage booster has matured, resulting in cost savings.

Introduction

- In this presentation, the odds of Falcon 9 first stage landing successfully will be predicted. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if I can determine that the first stage will land, the cost of a launch can be determined. This information then can be used if an alternate company wants to compete for a rocket launch.

Section 1

Methodology

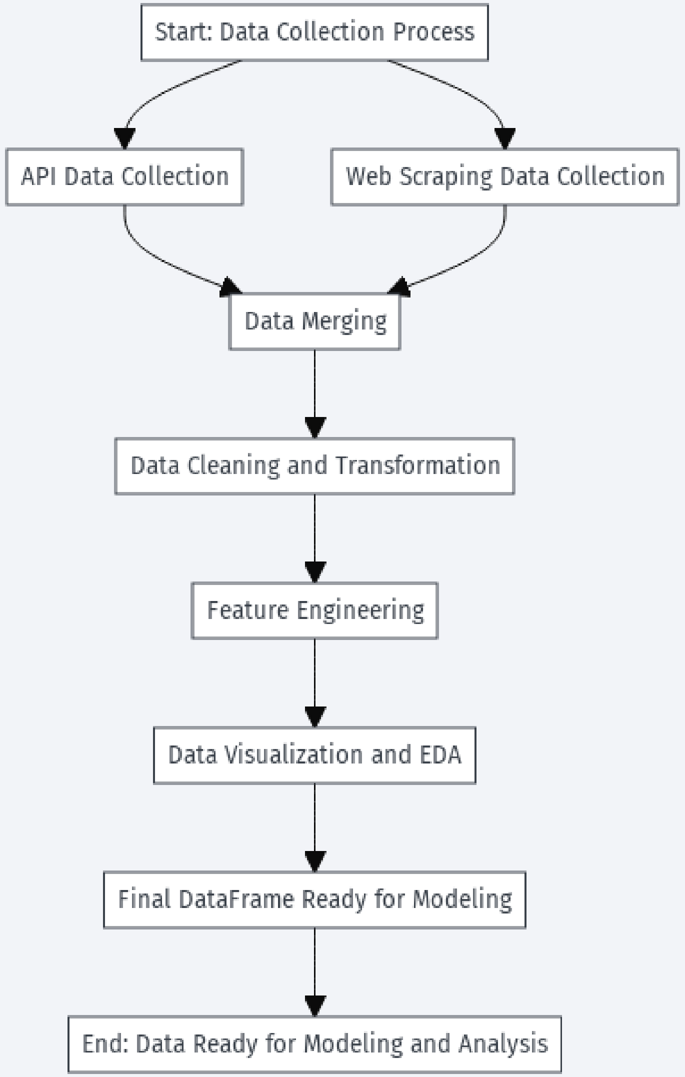
Methodology

Executive Summary

- Data collection methodology:
  - **Accessing the SPACEX API**
- Perform data wrangling
  - BeautifulSoup, Requests, Pandas, and API HTML were used.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - GridSearch and cross-validation techniques will be used on multiple classification models

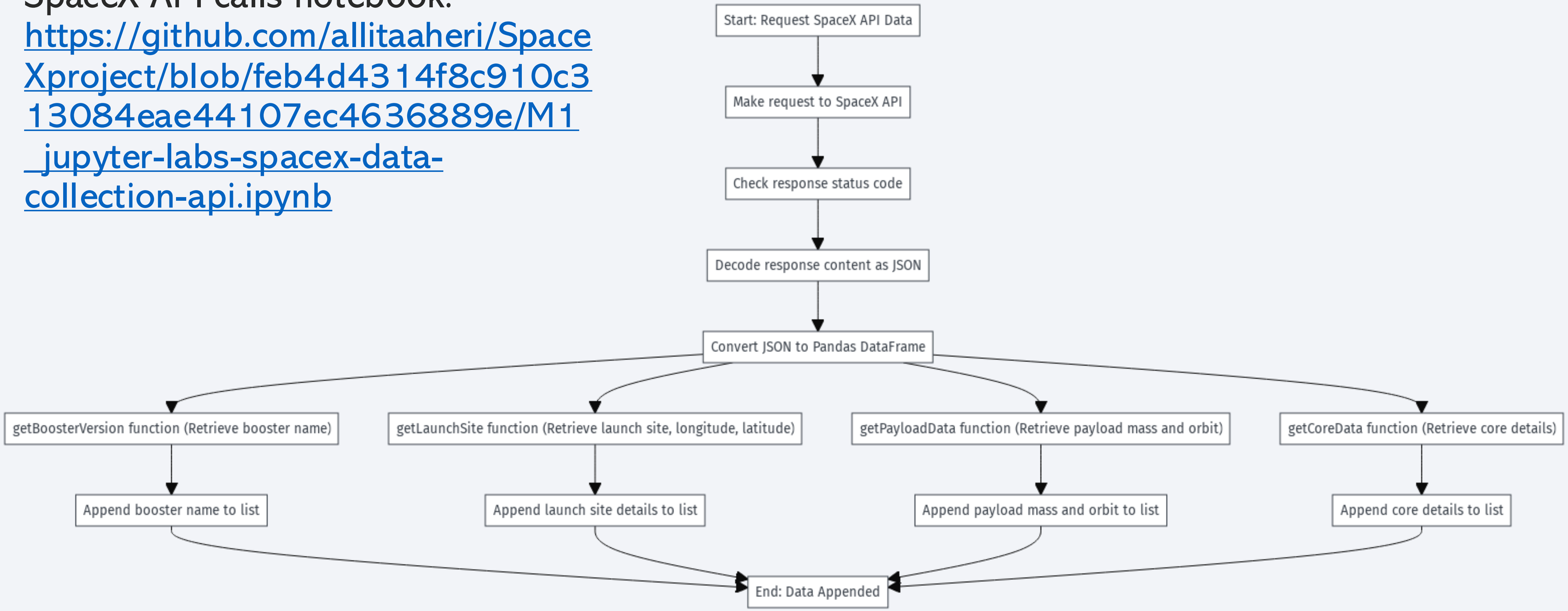
Data Collection

- **Endpoint:** I accessed SpaceX's RESTful API, which provides detailed information about past and upcoming launches, rockets, capsules, and more.
- **HTTP Requests:** Using the requests library in Python, I sent HTTP GET requests to the relevant API endpoints.
- Convert categorical data into numerical formats (e.g., one-hot encoding for categorical fields).
- Normalize/scale numerical data if required for modeling. Feature Engineering. Create new features such as LandingSuccess, LaunchYear, etc. Aggregate or transform data to create additional insights.
- Data Visualization and EDA: Perform exploratory data analysis (calculate statistics, identify trends).



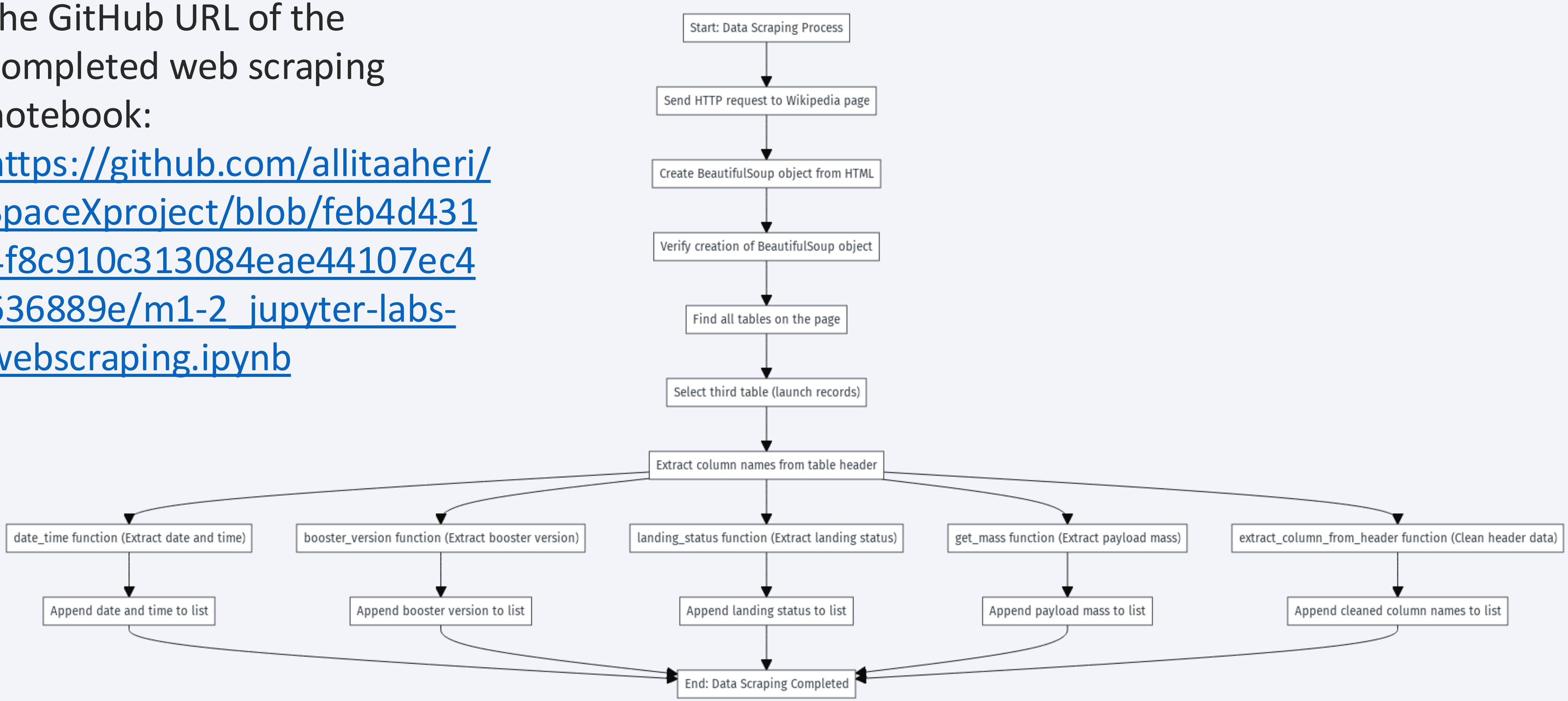
Data Collection – SpaceX API

- the GitHub URL of the completed SpaceX API calls notebook: https://github.com/allitaaheri/SpaceXproject/blob/feb4d4314f8c910c313084eae44107ec4636889e/M1_jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- the GitHub URL of the completed web scraping notebook: https://github.com/allitaaheri/SpaceXproject/blob/feb4d4314f8c910c313084eae44107ec4636889e/m1-2_jupyter-labs-webscraping.ipynb



Data Wrangling

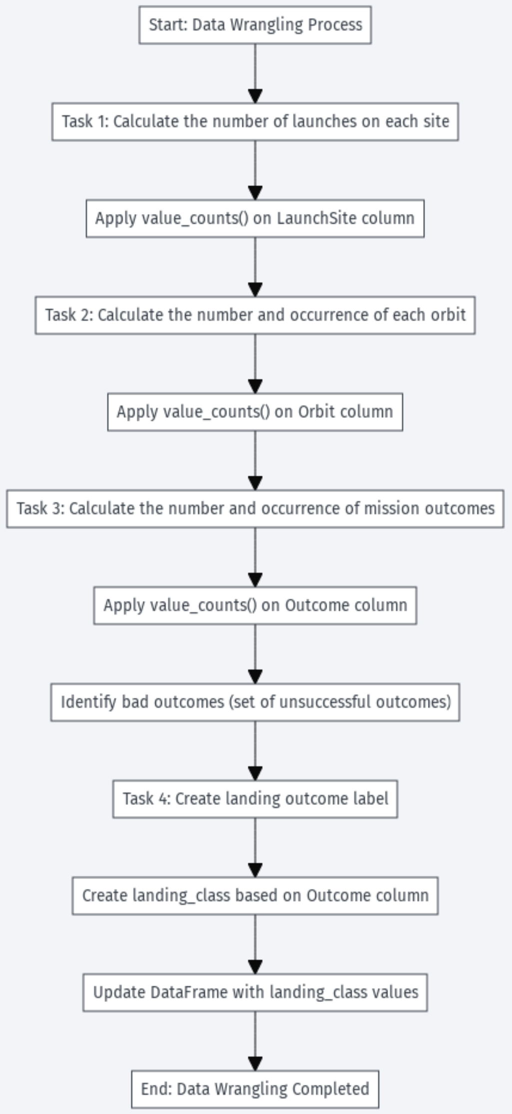
1. **Count Launches by Site** - The value_counts() method is applied to the LaunchSite column to determine the number of launches at each site.

2. **Count Occurrences of Orbits** - The value_counts() method is used on the Orbit column to identify the frequency of each orbit type.

3. **Analyze Mission Outcomes** - The Outcome column is analyzed using value_counts() to categorize successful and unsuccessful mission landings. Unsuccessful outcomes are identified and stored in the bad_outcomes set.

4. **Classify Landing Outcomes** - A new landing_class is created based on whether a mission outcome was successful (1) or not (0). This classification is added to the DataFrame as a new column.

- https://github.com/allitaaheri/SpaceXproject/blob/feb4d4314f8c910c313084eae44107ec4636889e/m1_3_labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- During the Exploratory Data Analysis (EDA), the following charts were plotted, and their purposes were as follows:
- 1. Flight Number vs. Payload Mass: A categorical plot (strip plot) was used to examine how both the flight number (indicating launch attempts over time) and payload mass affected the outcome of launches. This chart revealed that as flight numbers increased, indicating experience, the success rate improved. Additionally, higher payload masses tended to correlate with more launch failures.
- 2. Flight Number vs. Launch Site: Another categorical plot (strip plot) was created to show how flight numbers related to the success or failure of launches at different sites. This helped to uncover if specific launch sites were more prone to successful outcomes and how experience (represented by higher flight numbers) played a role.
- 3. Payload Mass vs. Launch Site: A scatter plot was used to check the relationship between payload mass and launch sites. This analysis revealed that heavy payloads (above 10,000 kg) were not launched from the VAFB-SLC launch site, providing insight into the site's limitations or preferences. https://github.com/allitaaheri/SpaceXproject/blob/feb4d4314f8c910c313084eae44107ec4636889e/jupyter-labs-eda-dataviz.ipynb

EDA with Data Visualization

- 4. Success Rate by Orbit: A bar chart was plotted to visualize the success rate for each orbit type by calculating the mean success rate (i.e., the mean of the "Class" column for each orbit). This allowed for identifying which orbits (e.g., LEO, ISS) had higher success rates and which were more challenging (e.g., GTO).
- 5. Flight Number vs. Orbit Type: A scatter plot was used to examine whether there was a relationship between flight numbers and orbit type. The chart showed that for LEO, higher flight numbers correlated with a higher success rate, while GTO orbit showed no clear pattern related to flight number.
- 6. Payload Mass vs. Orbit Type: This scatter plot was used to investigate how payload mass affected the success rate across different orbit types. The result indicated that heavy payloads led to more successful missions in Polar, LEO, and ISS orbits, while GTO missions showed a mixed success rate regardless of payload mass.
- 7. Yearly Trend of Success Rate: A line chart was plotted to show the average success rate over time, with years extracted from the launch dates. This chart revealed an increasing trend in success rates starting from 2013, stabilizing in 2014, and continuing to rise from 2015 onward, reflecting the learning curve and improvements made by SpaceX over time. These charts were chosen to provide insights into the key factors (flight experience, payload, launch site, orbit, and time) affecting SpaceX's launch outcomes and success rates.

EDA with SQL

- **Task 1**: Select distinct launch sites.
  - `SELECT DISTINCT "Launch_Site"`
- **Task 2**: Select records where launch sites start with 'CCA'.
  - `SELECT * WHERE "Launch_Site" LIKE 'CCA%'`
- **Task 3**: Sum of payload mass for NASA (CRS) launches.
  - `SELECT SUM("PAYLOAD_MASS__KG_") WHERE "Customer" LIKE '%NASA (CRS)%'`
- **Task 4**: Average payload mass for booster version F9 v1.1.
  - `SELECT AVG("PAYLOAD_MASS__KG_") WHERE "Booster_Version" = 'F9 v1.1'`
- **Task 5**: Find the date of the first successful ground pad landing.
  - `SELECT MIN("Date") WHERE "Landing_Outcome" = 'Success (ground pad)'`
- **Task 6**: Select boosters with successful drone ship landings and payloads between 4000 and 6000 kg.
  - `SELECT "Booster_Version" WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000`

- https://github.com/allitaaheri/SpaceXproject/blob/feb4d4314f8c910c313084eae44107ec4636889e/jupyter-labs-eda-sql-coursera_sqllite_module2.ipynb

EDA with SQL

- **Task 7**: Count of successful and failed mission outcomes.
  - `SELECT "Landing_Outcome", COUNT(*) GROUP BY "Landing_Outcome"`
- **Task 8**: Select boosters that carried the maximum payload.
  - `SELECT "Booster_Version" WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_"))`
- **Task 9**: Select failure outcomes in 2015, showing month, booster versions, and launch sites.
  - `SELECT substr("Date", 6, 2) AS Month, "Booster_Version", "Launch_Site" WHERE substr("Date", 0, 5) = '2015' AND "Landing_Outcome" = 'Failure (drone ship)'`
- **Task 10**: Rank landing outcomes by count between specific dates.
  - `SELECT "Landing_Outcome", COUNT(*) WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY COUNT(*) DESC`

- https://github.com/allitaaheri/SpaceXproject/blob/feb4d4314f8c910c313084eae44107ec4636889e/jupyter-labs-eda-sql-coursera_sqllite_module2.ipynb

Build an Interactive Map with Folium

- **Markers for Launch Sites**:
- **Map Object**: folium.Marker and folium.Circle.
- **Purpose**: Marked all SpaceX launch sites (e.g., CCAFS LC-40, KSC LC-39A, VAFB SLC-4E) on the map, each with a circle to highlight its location and a marker displaying the site's name. This helps visualize the geographical location of each launch site.
- **Colored Markers for Launch Outcomes**:
- **Map Object**: folium.Marker within a MarkerCluster.
- **Purpose**: Used green markers for successful launches and red markers for failed launches, indicating the outcome of each launch at specific locations. The MarkerCluster object was used to group markers with the same coordinates and simplify the visualization, allowing easy identification of success rates per site.
- **Mouse Position for Coordinates**:
- **Map Object**: MousePosition.
- **Purpose**: Enabled users to hover over the map and see the exact latitude and longitude of any point. This feature was used to determine the coordinates of key proximities such as railways, highways, and coastlines.

- https://github.com/allitaaheri/SpaceXproject/blob/337f17db23ca99851a49d3234a6362ce3e6de7ab/lab_jupyter_launch_site_location.ipynb

Build an Interactive Map with Folium

- **Distance Markers**:
- **Map Object**: folium.Marker with DivIcon.
- **Purpose**: Added markers showing the distance from launch sites to nearby points of interest, such as the coastline, highways, and cities. The markers displayed the calculated distance in kilometers, helping visualize proximity to essential infrastructure.
- **Polylines to Represent Distance**:
- **Map Object**: folium.PolyLine.
- **Purpose**: Drew lines between launch sites and key locations such as coastlines, cities, and highways. These lines represent the calculated distances, providing a clear visual connection between the sites and their surrounding infrastructure.

- https://github.com/allitaaheri/SpaceXproject/blob/337f17db23ca99851a49d3234a6362ce3e6de7ab/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Here's a summary of the plots/graphs and interactions added to the dashboard, along with the reasons for including them:
- **1. Dropdown Menu for Launch Site Selection:**
- A dropdown menu allows users to select a specific launch site or view data for all launch sites. interaction provides flexibility for users to filter the data by individual launch sites or view an aggregate of all sites, allowing for a detailed comparison of launch outcomes across different locations.
- **2. Pie Chart for Success Counts:**
- A pie chart visualizes either the total number of successful launches across all sites or the success vs. failure count for a selected site. This chart provides a clear and concise representation of the overall success rates, either globally or for individual launch sites. It helps users quickly understand how successful SpaceX launches have been at specific sites.
- https://github.com/allitaaheri/SpaceXproject/blob/feb4d4314f8c910c313084eae44107ec4636889e/spacex_dash_app.py

Build a Dashboard with Plotly Dash

- **3. Payload Range Slider:**
- A range slider allows users to select a specific payload mass range. This interaction helps narrow down the analysis by payload size, allowing users to examine how payload mass impacts the success rate. It's particularly useful for identifying trends related to heavier or lighter payloads.
- **4. Scatter Chart for Payload vs. Success:**
- A scatter plot showing the relationship between payload mass and the success of the launch, with different colors representing various booster versions. This plot helps analyze the correlation between payload mass and launch outcomes (success or failure) across different booster versions. It allows users to observe whether heavier payloads affect the success rate and compare performance across booster versions.
- These plots and interactions were added to allow users to explore the SpaceX launch data dynamically and uncover patterns related to launch site performance, payload mass, and booster version success rates.
- https://github.com/allitaaheri/SpaceXproject/blob/feb4d4314f8c910c313084eae44107ec4636889e/spacex_dash_app.py

Predictive Analysis (Classification)

Summary of the steps for Building, Evaluating, and Improving the Classification Model is presented.

https://github.com/allitaaheri/SpaceXproject/blob/337f17db23ca99851a49d3234a6362ce3e6de7ab/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

- **Logistic Regression**:
  - Created a `LogisticRegression` model.
  - Used `GridSearchCV` to tune hyperparameters (`C`, `penalty`, `solver`) with 10-fold cross-validation.
  - Best hyperparameters: `{'C': 1, 'penalty': 'l2', 'solver': 'lbfgs'}`.
  - Achieved validation accuracy of 82.14%.
- **Support Vector Machine (SVM)**:
  - Created an `SVC` model.
  - Tuned hyperparameters (`C`, `kernel`, `gamma`) using `GridSearchCV` and 10-fold cross-validation.
  - Best hyperparameters: `{'C': 1.0, 'gamma': 0.0316, 'kernel': 'sigmoid'}`.
  - Achieved validation accuracy of 84.82%.
- **Decision Tree**:
  - Created a `DecisionTreeClassifier`.
  - Used `GridSearchCV` to tune `criterion`, `max_depth`, `splitter`, and other parameters.
  - Best hyperparameters: `{'criterion': 'gini', 'max_depth': 18, 'splitter': 'best'}`.
  - Achieved validation accuracy of 86.25%.
- **K-Nearest Neighbors (KNN)**:
  - Created a `KNeighborsClassifier`.
  - Tuned `n_neighbors`, `algorithm`, and `p` using `GridSearchCV`.
  - Best hyperparameters: `{'n_neighbors': 6, 'algorithm': 'auto', 'p': 1}`.
  - Achieved validation accuracy of 83.39%.

Predictive Analysis (Classification)

- **Data Processing**:
- Created a target variable Class (landing success/failure). Standardized the features using StandardScaler for better model performance. Split the data into training (80%) and testing (20%) sets using train_test_split. For each model, accuracy on the test data was calculated.All models except the Decision Tree performed similarly on the test data with 83.3% accuracy.Confusion matrices were plotted to analyze the model performance, showing the breakdown of True Positives, False Positives, etc.

Final Results:

1. Decision Tree was the best-performing model on the training validation set but performed worse (77.8% accuracy) on the test set due to overfitting.
2. Logistic Regression, SVM, and KNN performed similarly with 83.3% accuracy on the test set.

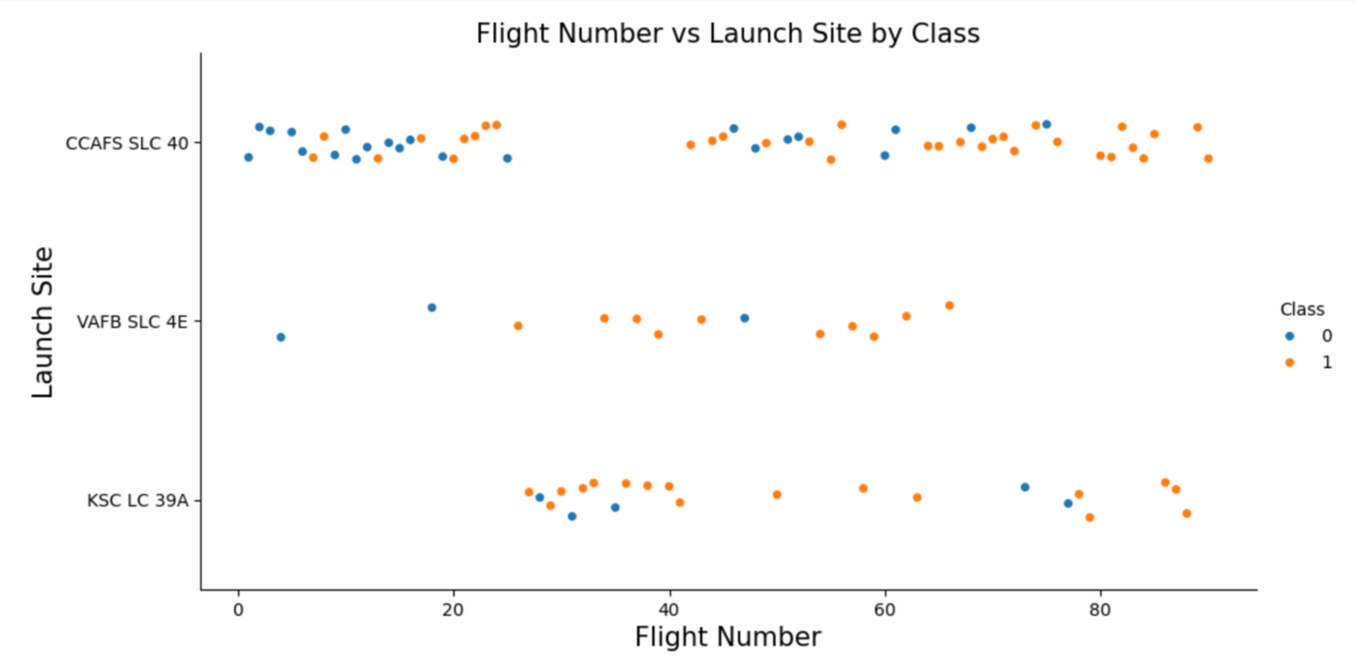
- https://github.com/allitaaheri/SpaceXproject/blob/337f17db23ca99851a49d3234a6362ce3e6de7ab/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Section 2

Insights drawn from EDA

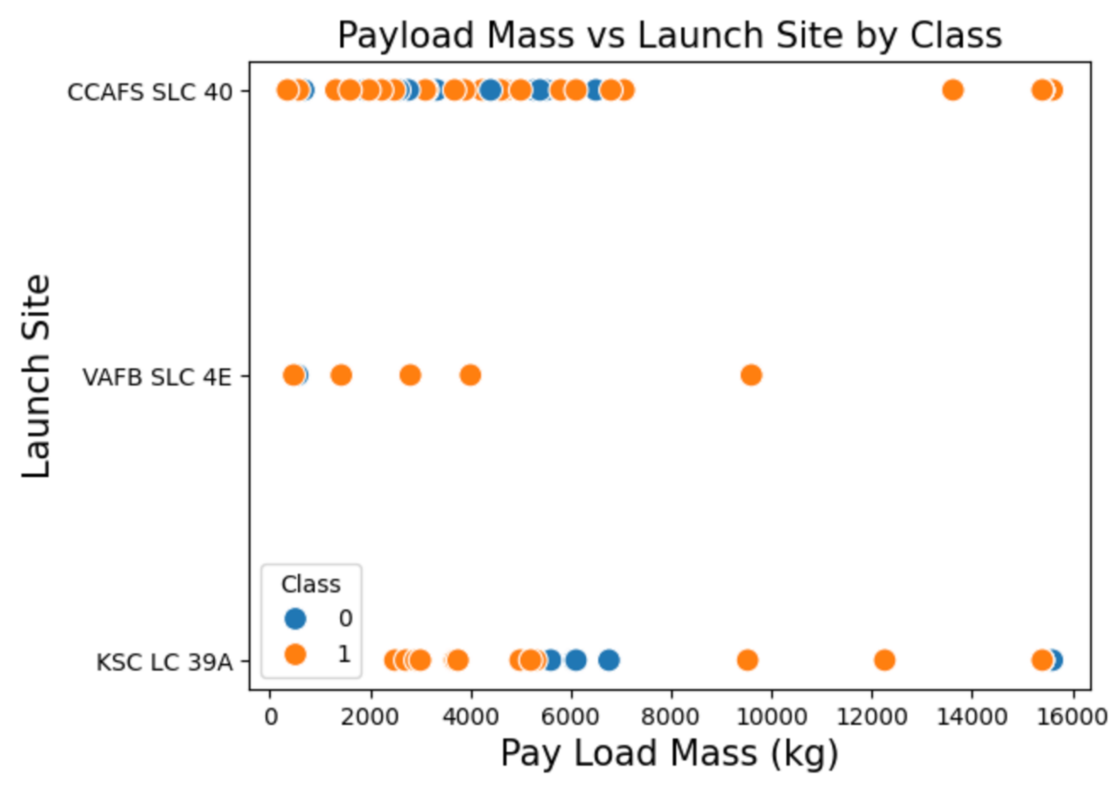
Flight Number vs. Launch Site

- **CCAFS SLC 40**: This site shows a mixture of successful and failed launches early on (low flight numbers), but as flight numbers increase, the site shows more consistent successes.
- **VAFB SLC 4E**: There are fewer launches from this site, but it shows a clear trend toward successful launches, particularly in later flight numbers.
- **KSC LC 39A**: This site also demonstrates mixed results in the early flight numbers but has more successful launches (Class 1) later.



Payload vs. Launch Site

- CCAFS SLC 40: This site shows a wide range of payload masses, from below 2,000 kg to over 15,000 kg. There are a mix of successful and failed launches for lower payload masses (below ~6,000 kg), but most of the heavier payloads tend to be successful.
- VAFB SLC 4E: This site handles a narrower range of payload masses, mostly between 5,000 and 10,000 kg. Almost all launches at this site are successful (orange dots) with a high success rate for various payloads.
- KSC LC 39A: Similar to CCAFS SLC 40, KSC LC 39A also handles a range of payloads, primarily under 6,000 kg, with a few heavy payloads around 15,000 kg. For lower payload masses, both successful and failed launches are observed, but for higher payloads, there are more successful outcomes.



Success Rate vs. Orbit Type

- **High Success Rate Orbits:**

  ES-L1, GEO, LEO, SSO: These orbits have a **100% success rate**. All launches to these orbits were successful, indicating strong reliability when targeting these specific orbits.

  VLEO: This orbit also has a high success rate, though not perfect, slightly below 90%.

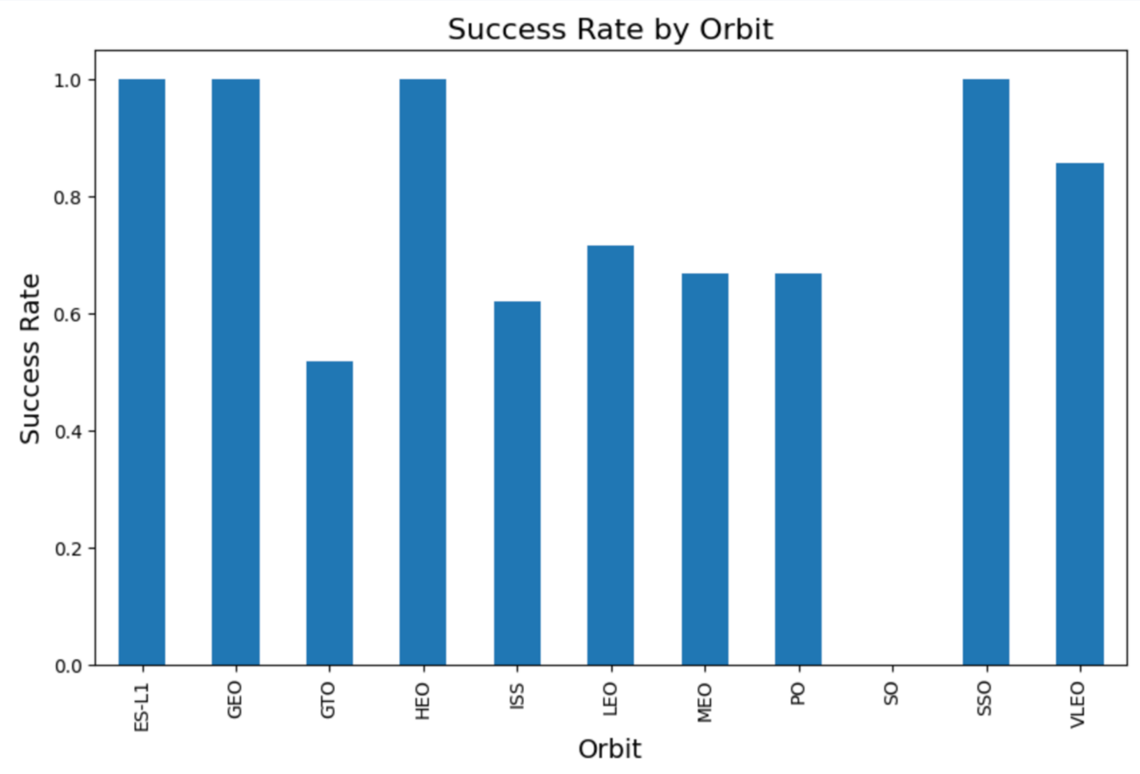
- **2. Moderate Success Rate Orbits:**

  ISS, MEO, PO: These orbits have success rates between **70% and 90%**. While there were some failures, the majority of launches to these orbits were successful.

- **3. Low Success Rate Orbits:**

  GTO: Approximately **50%** success rate, making it one of the least successful orbit types in this dataset.

  HEO: Success rate is around **60%**, slightly better than GTO, but still lower compared to other orbits.



Flight Number vs. Orbit Type

- **1. LEO Orbit:**

  Higher Flight Numbers: Success rate improves as the flight number increases, with more orange dots (successful launches) appearing in higher flight numbers.

  Early Failures: The earlier flight numbers have a mix of success (Class 1) and failures (Class 0), but failures decrease with experience.

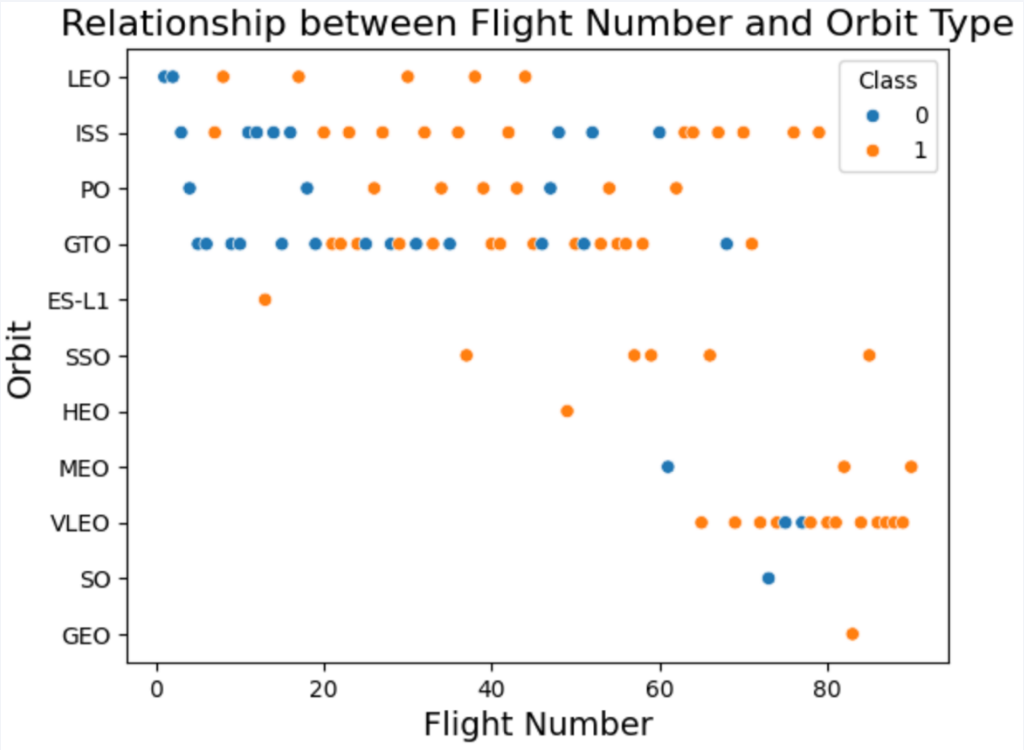
- **2. ISS Orbit:**

  Consistent Success: The **ISS orbit** shows relatively high success throughout, especially for higher flight numbers. However, there are a few failures scattered in earlier flights.

- **3. PO and GTO Orbits:**

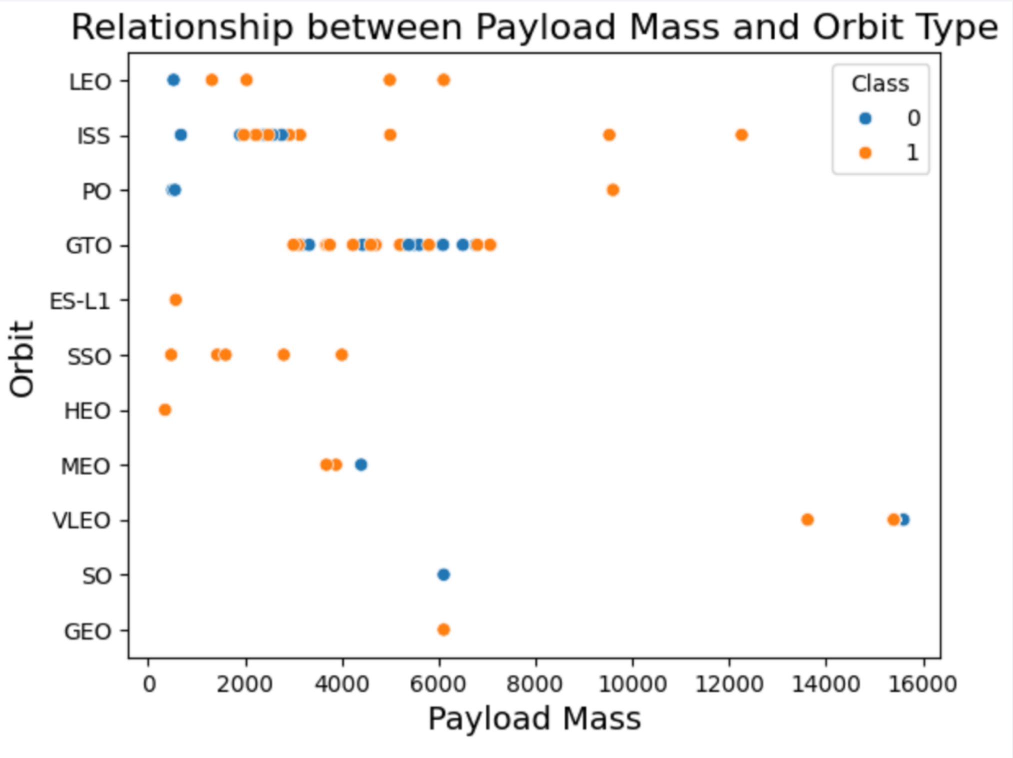
  Mixed Success: These orbits show a mix of success and failure throughout the flight numbers, with failures (Class 0) more common in the earlier stages of launches.

  GTO: Notably, GTO has a more even distribution of failures and successes, indicating challenges in achieving consistent success for this orbit.



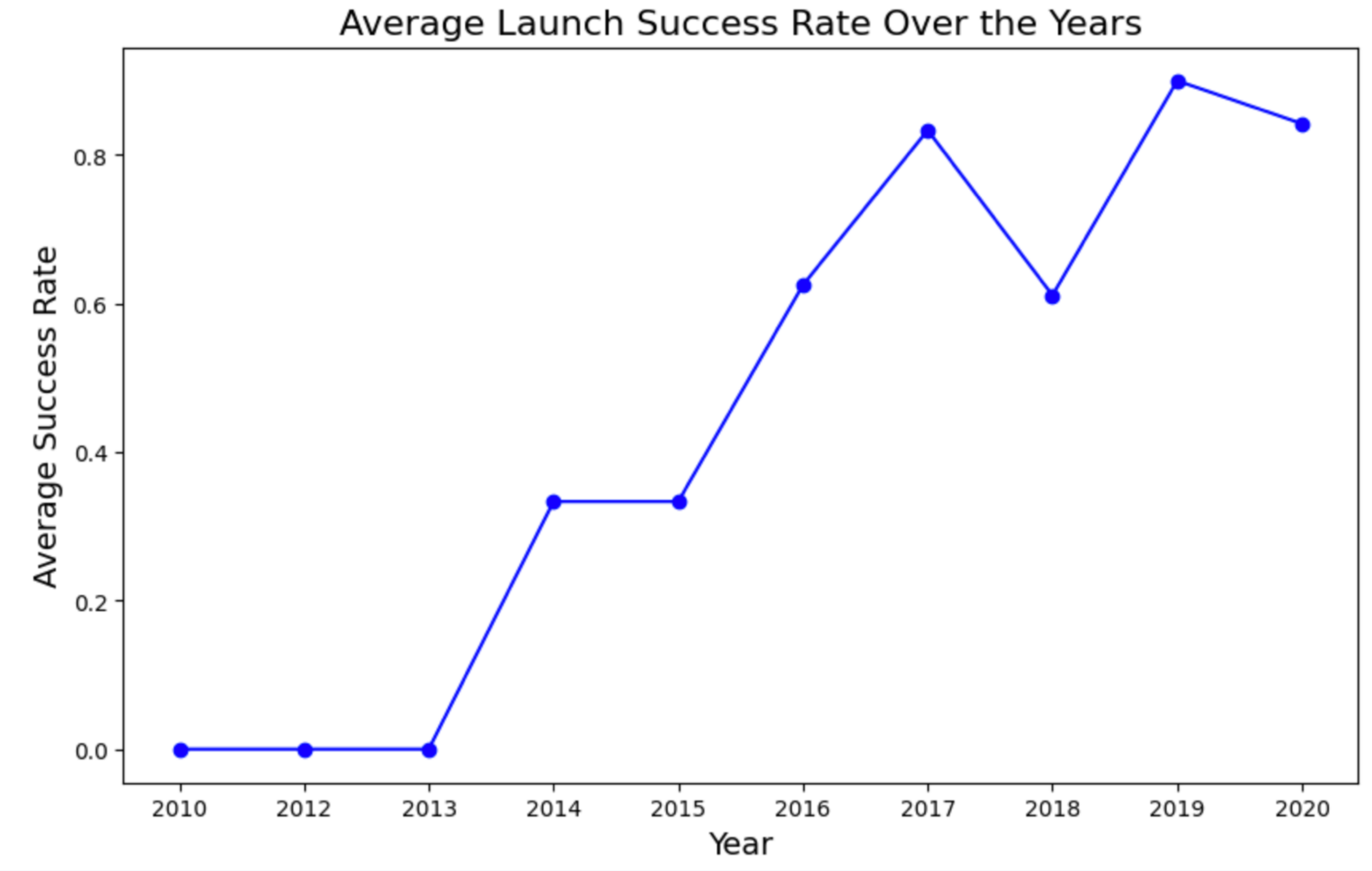
Payload vs. Orbit Type

- **LEO (Low Earth Orbit)**: A wide range of payload masses is observed, from very light (~1,000 kg) to very heavy (~15,000 kg). Despite the variety of payloads, the success rate (orange dots) remains high for most launches.
- **ISS (International Space Station)**: The payload mass here is concentrated between 4,000 kg and 6,000 kg. Although some failures (blue dots) exist, most launches are successful.
- **PO (Polar Orbit)**: Most of the launches in the PO orbit are failures (blue dots), with relatively low payload masses (~1,000 kg).
- **GTO (Geostationary Transfer Orbit)**: This orbit has a wide range of payloads, similar to LEO, ranging from 4,000 kg to 15,000 kg. GTO shows a more balanced mix of successes (orange) and failures (blue) compared to LEO, making it a more challenging orbit.
- **SSO (Sun-Synchronous Orbit)**: This orbit shows consistent success, even with relatively low payload masses (~2,000 kg). The dots are all orange, indicating high reliability in this orbit.



Launch Success Yearly Trend

- We 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

The 'DISTINCT' command retrieves a list of all unique launch site names from the SPACEX table in the database. The launch sites names are:

1. KSC LC-39A
2. CCAFS SLC-40
3. CCAFS LC-40
4. VAFB SLC-4E

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL;
```

[8] ✓ 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'

- The query returns 5 records where the launch site begins with "CCA", which corresponds to **CCAFS LC-40**. All launches listed are from this site and had varying payloads, mainly to **LEO (Low Earth Orbit)** or **LEO (ISS)**.
- The mission outcomes were successful in all cases, but landing outcomes varied, with some showing failures or no attempts at landing.

```
%sql SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40

Total Payload Mass

- Using the ‘SUM’ command, the total payload carried by boosters from NASA is calculated as 48,213 kilograms.

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") AS Total_Payload_Mass FROM SPACEXTBL WHERE "Customer" LIKE '%NASA (CRS)%';
```

Python

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

Total_Payload_Mass
48213

Average Payload Mass by F9 v1.1

- The average payload mass of 2928.4 kilograms, which is carried by booster version F9 v1.1, was calculated by using the combination of 'AVG' and 'WHERE' commands.

```python
%sql SELECT AVG("PAYLOAD_MASS__KG_") AS Average_Payload_Mass FROM SPACEXTBL WHERE "Booster_Version" = 'F9 v1.1';
```

Python

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

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

- The first successful landing outcome date was calculated using the ‘MIN’ command as December 22, 2015.

```
%sql SELECT MIN("Date") AS First_Successful_Landing FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (ground pad)';
```

Python

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

First_Successful_Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Using the conditional 'AND' statement, the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 are listed below:
- F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2

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

Python

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

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

Total Number of Successful and Failure Mission Outcomes

- **Successful Landings**: There are 38 successful landings, with 14 of them occurring on drone ships and 9 on ground pads.
  Failed Landings: Failures occurred 3 times, with additional failures on drone ships (5) and parachute failures (2).
  Controlled Landings in the Ocean: There were 5 landings where the outcome was a controlled descent into the ocean.”
  No attempt" outcomes: 22 total cases (21 + 1) where there was no landing attempt.

```
%sql SELECT "Landing_Outcome", COUNT(*) AS Outcome_Count FROM SPACEXTBL GROUP BY "Landing_Outcome";
```

Python

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

Landing_Outcome	Outcome_Count
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

- The booster which have carried the maximum payload mass are listed using a subquery strategy with two 'FROM' commands:

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

Python

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

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

2015 Launch Records

The two records which will display the month names, failure landing_outcomes in drone ship , booster versions, launch_site for the months in year 2015 are listed.

Since SQLLite does not support month names, the 'substr(Date, 6,2)' command was used to get the months and 'substr(Date,0,5)='2015" for year.

```
%sql SELECT substr("Date", 6, 2) AS Month, "Booster_Version", "Launch_Site", "Landing_Outcome" FROM SPACEXTBL WHERE substr("Date", 0, 5) = '2015' AND "Landing_Outcome" = 'Failure (drone ship)';
```

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)

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

- The count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order has been listed using ‘COUNT’, ‘AND’, ‘ORDER BY’ commands.

```
%sql SELECT "Landing_Outcome", COUNT(*) AS Outcome_Count FROM SPACEXTBL WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY Outcome_Count DESC;
```

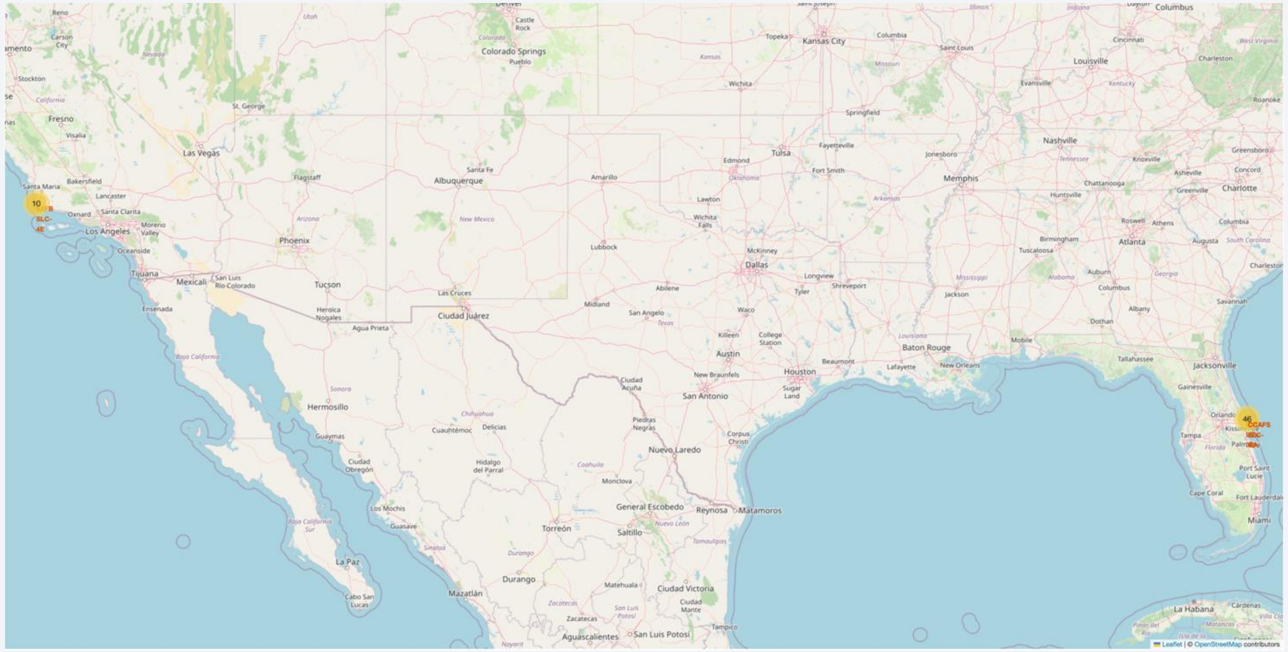
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

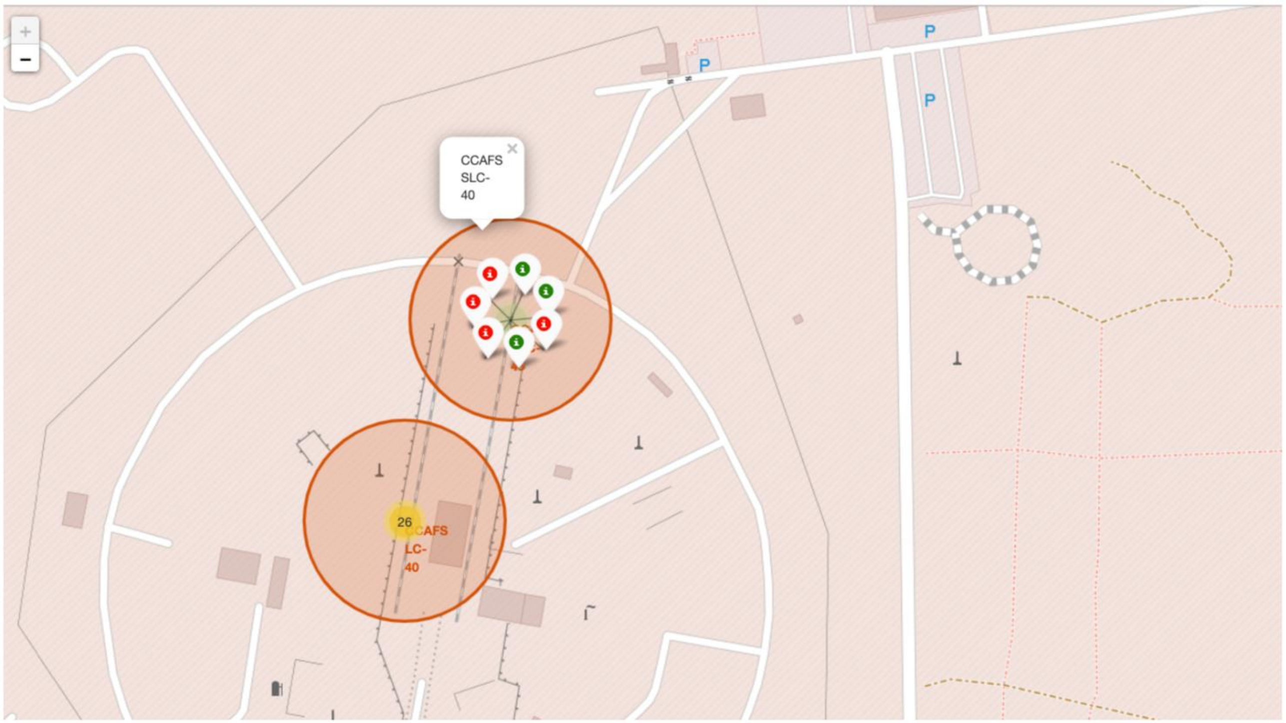
SPACEX Lunch Sites across USA

- The generated folium map includes all launch sites' location markers on a global map.



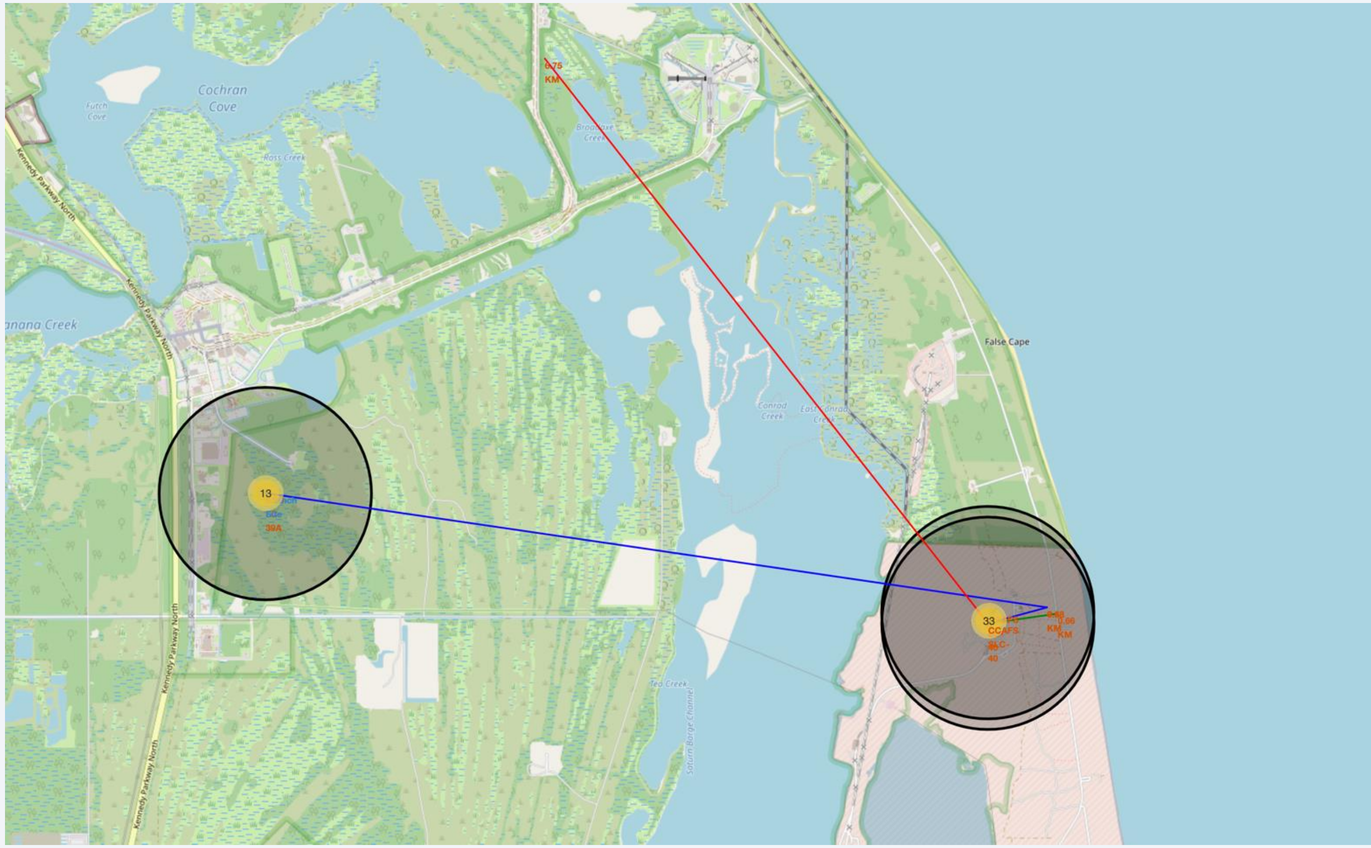
Color-coded map for lunch sites

- In the analysis, **Green markers** were used to represent **successful launches** (class=1) and **Red markers** were used to represent **failed launches** (class=0).
- Since launches occurred at only four distinct launch sites, and many launches shared the same coordinates, we utilized **marker clusters** to organize the data more efficiently. This allowed us to simplify the visualization by grouping markers with the same coordinates, making it easier to identify patterns.
- From the color-coded markers, it became straightforward to visually assess which launch sites had relatively high success rates. The clusters provided a clear picture of the proportion of successful and failed launches at each site, helping to highlight performance trends across different launch locations.



Lunch site distance to important locations

- The generated folium map shows the screenshot of a selected launch site to its proximities such as coastline, airport, highway, another lunch site.



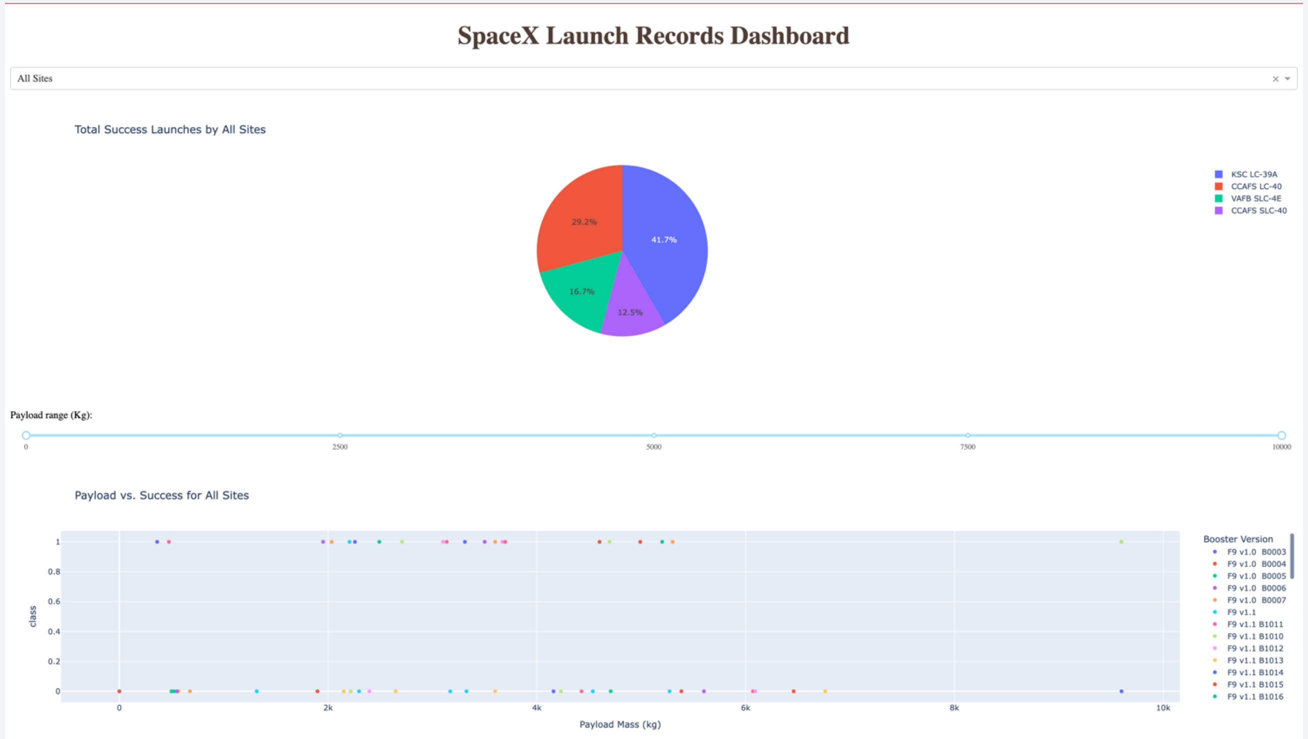
Section 4

Build a Dashboard with Plotly Dash

Launch success count for all sites

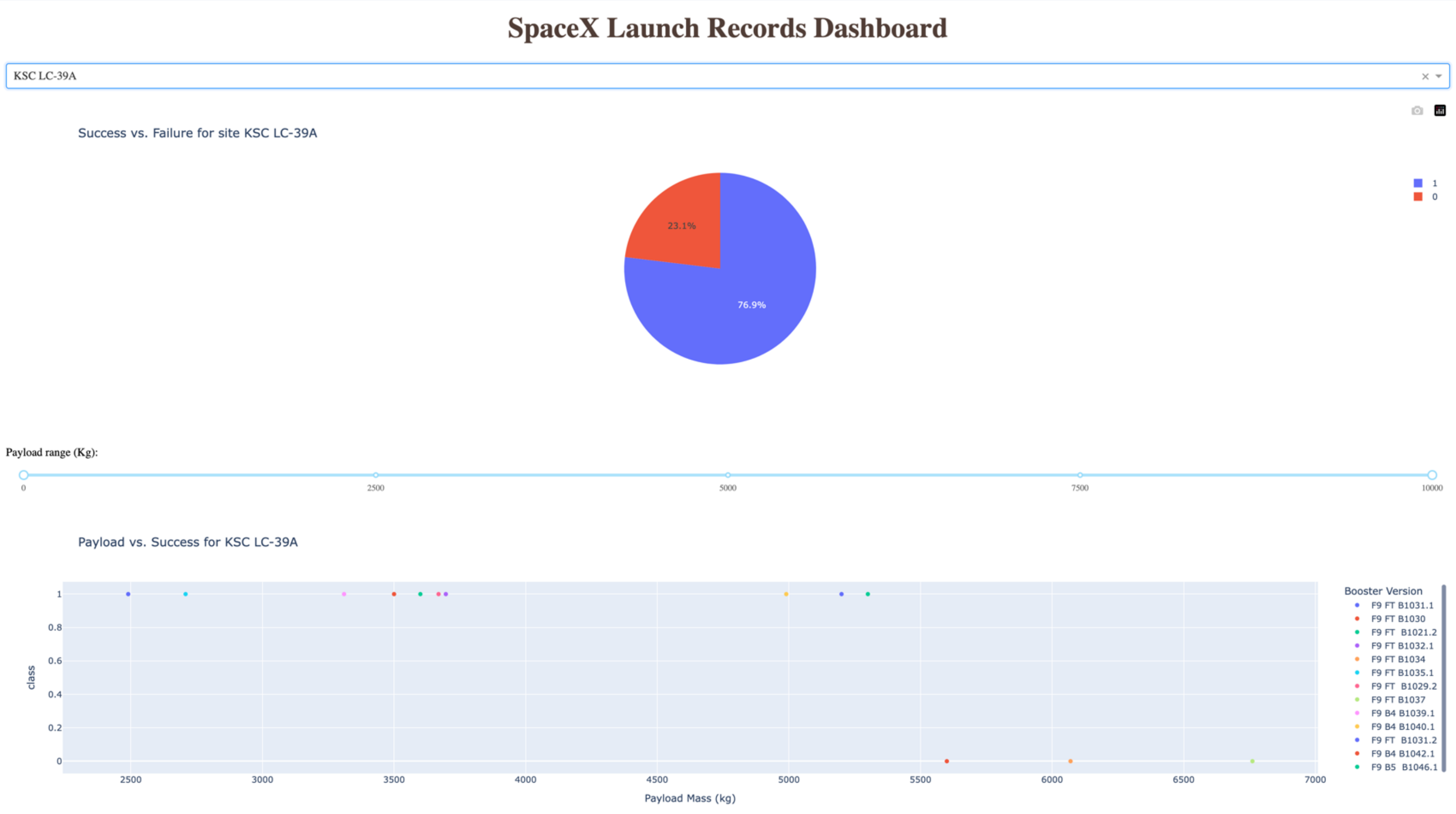
- **KSC LC-39A** is the most successful launch site, contributing to almost half of the successful launches. **Failures** are more common with lower payload masses, while **successes** are spread more evenly across the payload range. The variety of **booster versions** used across payload masses suggests ongoing iterations and experimentation in booster technology, but no clear booster version stands out in terms of launch success or failure.

- **Key Insight**: Multiple booster versions were used in SpaceX launches, but no clear pattern emerges showing that one version is more successful than others based on payload mass or launch outcome.



Pie chart of the most successful launch site

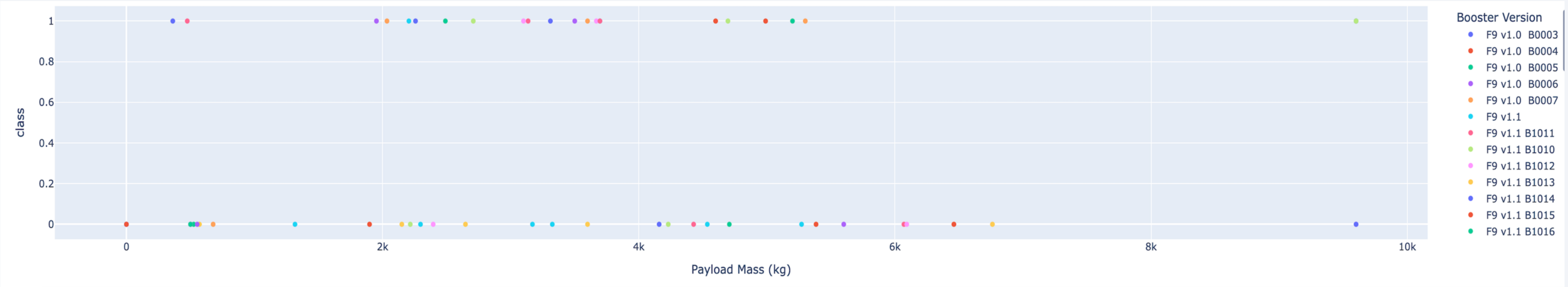
- The pie chart confirms that **KSC LC-39A** has a **strong performance track record**, with over three-quarters of launches resulting in successful outcomes. This reliability makes it a key site in SpaceX's launch operations. The failure rate, while present, is relatively low, which may be attributed to earlier trials or challenges with specific launches, but overall, this site has consistently delivered high success rates for a wide range of missions. The payload mass is mostly below 4000 kg.



Payload vs. Launch Outcome scatter plot for all sites

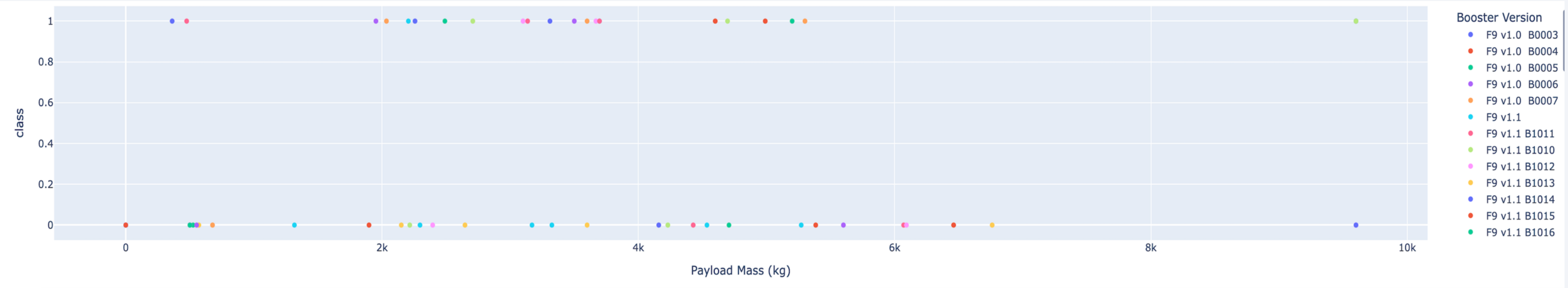
Booster Version and Success/Failure:

1. **F9 v1.0 (B0003 - B0007)**: Early booster versions like F9 v1.0 B0003, B0004, B0005, and others show a mix of success and failure, with several failures observed at lower payload masses (around 0-2000 kg).
2. **F9 v1.1 (B1011 - B1016)**: These later booster versions show more success at higher payload masses (above 2000 kg). However, some failures are still visible at payloads of around 0-2000 kg for the same versions. **F9 v1.1 B1013, B1014, and B1015** appear to have a more consistent success rate across a wide payload range, as most of their markers appear in the "successful" class (y=1).
3. **F9 FT B1031.1** and **F9 FT B1021.2**: These versions are seen at higher payloads (close to 6000 kg) with successful launches, indicating that newer booster versions were likely optimized for heavier payloads.



Payload vs. Launch Outcome scatter plot for all sites

1. **Payload and Booster Versions**:
   1. **Lower Payloads (0-2000 kg)**:
      1. Multiple failures are seen across different booster versions. This might indicate that launches with smaller payloads were more experimental or that issues occurred more frequently at this payload range.
   2. **Higher Payloads (4000-10,000 kg)**:
      1. More recent booster versions like **F9 FT B1031.1** and **F9 FT B1021.2** show consistent successes at higher payloads. These versions seem to be optimized for launches with larger payloads, reflecting their reliability.

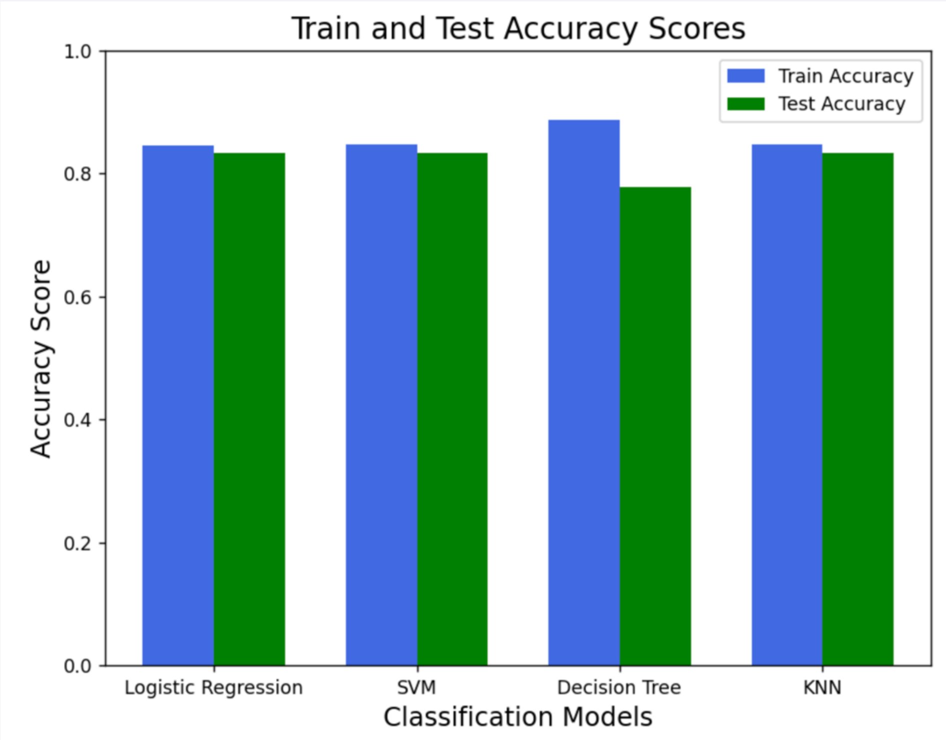


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- KNN and SVM Performed the best in both training and testing sets.



GridSearchCV
estimator: KNeighborsClassifier
KNeighborsClassifier

```python
print("tuned hpyerparameters :(best parameters) ",knn_cv.best_params_)
print("accuracy :",knn_cv.best_score_)
```

Python

```
tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 6, 'p': 1}
accuracy : 0.8339285714285714
```

GridSearchCV
estimator: SVC
SVC

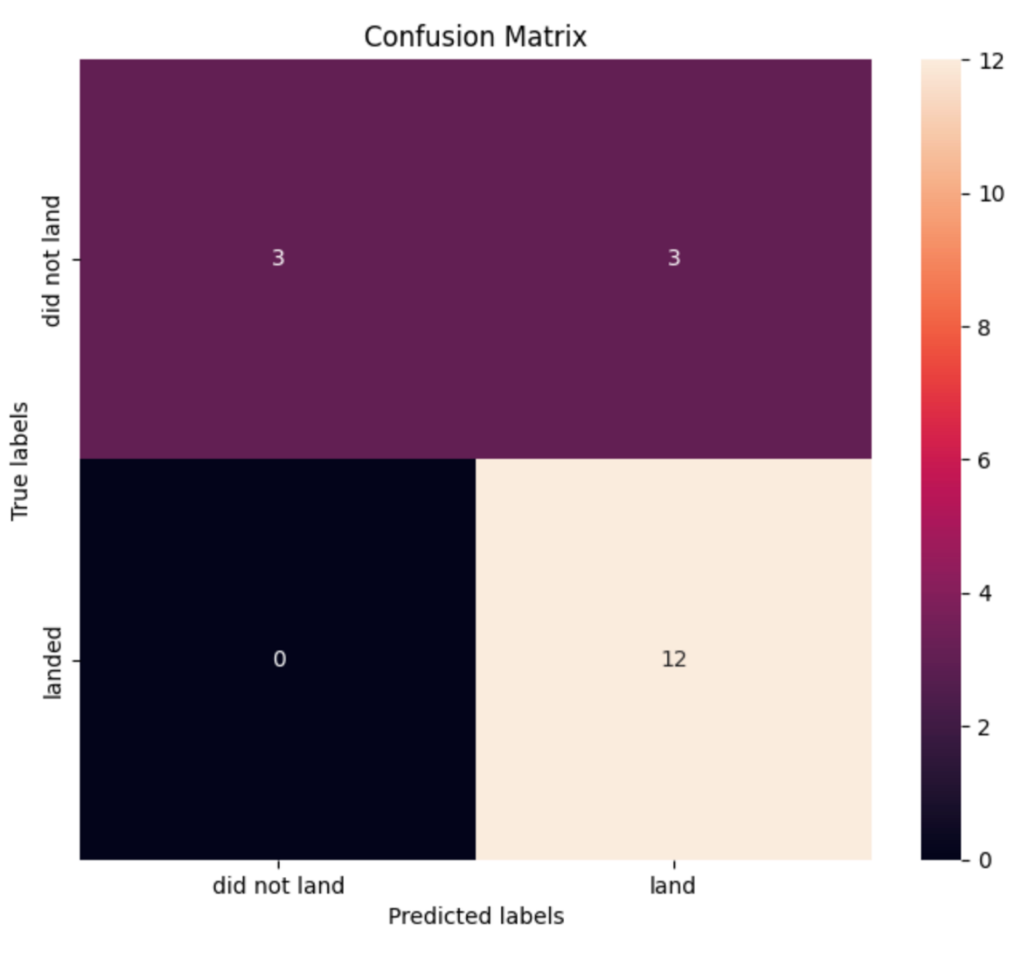
```python
print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
```

Python

```
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel':
accuracy : 0.8482142857142858
```

Confusion Matrix

- **Accuracy**: The model's overall performance is quite good, as it correctly predicted 15 out of 18 instances (3 + 12 = 15 correct predictions out of 18 total).
- **Error**: The error occurs in 3 instances where the model incorrectly predicted "did not land" for actual "landed" cases.
- **False Positive Rate**: The model has no false positives, which is good, indicating it did not incorrectly predict any non-landed cases as landed.
- **False Negative Rate**: There are 3 false negatives, where the model missed predicting a successful landing.



Conclusions

- Using API requests and webscrapping the SPACEX database was retrieved structured and prepared.
- Various SQL queries, modeling, visualization and data analysis techniques were employed to investigate various features including rocket booster version, lunch site, payload mass, etc.
- The success rate varies significantly depending on the orbit, with some orbits like **GTO** and **HEO** posing more challenges for successful missions, while others like **ES-L1, GEO, LEO, and SSO** have perfect success records. **Earlier booster versions** (like F9 v1.0) show more failures, particularly at lower payload masses, likely during the testing and development phase of SpaceX launches.
- **Later booster versions** (like F9 v1.1 and F9 FT) have shown significant improvement, with more successful launches at higher payload ranges, indicating the maturation of SpaceX's booster technology.
- Overall, **booster version improvements** seem correlated with **increasing payload capacity** and **launch success rates**, reflecting technological advancements over time.
- The KNN and SVM models perform reasonably well, but they are more prone to missing successful landings (false negatives) rather than incorrectly predicting landings that did not occur.

Thank you!