IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mazhar H

20.1.26

Outline

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

Executive Summary

Methodology Overview:

- Collected SpaceX launch data from APIs and Wikipedia, then cleaned, prepared, and stored it in a Db2 database for SQL-based analysis and EDA.
- Engineered features, standardized data, and created interactive visualizations using Folium and Plotly Dash.
- Built and tuned SVM, Decision Tree, and KNN models using GridSearchCV and evaluated them on test data.

Results Summary:

- Key factors affecting Falcon 9 first-stage landings were identified, with clear geographic and success-rate patterns.
- Model accuracies were 83.33% for SVM and KNN, and 94.44% for the Decision Tree.
- Overall, launch site and payload mass significantly influence landing success, with the Decision Tree emerging as the strongest predictor.

Introduction

Project Background:
This capstone project focuses on predicting whether the **Falcon 9 first stage** will land successfully. SpaceX's lower launch costs are largely driven by first-stage reusability, and accurate landing predictions can help estimate launch costs and support competitive decision-making in the aerospace market.

Key Questions:
- Which factors most affect Falcon 9 first-stage landing success?
- How effectively can machine learning models predict landing outcomes?
- Which model delivers the best prediction performance?

Section 1

Methodology

Methodology

- Data collection methodology:
  - Data is collected from SpaceX API and then the requested data is cleaned
  - Using Web Scraping Technique from Wikipedia
- Perform data wrangling
  - Data cleaning and filtering. Dealing with the missing and null values
  - Using one hot encoding for the binary classification of the data, as a part of data preparation step.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Applied varied ML predictive tools which helps in tuning, prediction and evaluation of models for best results

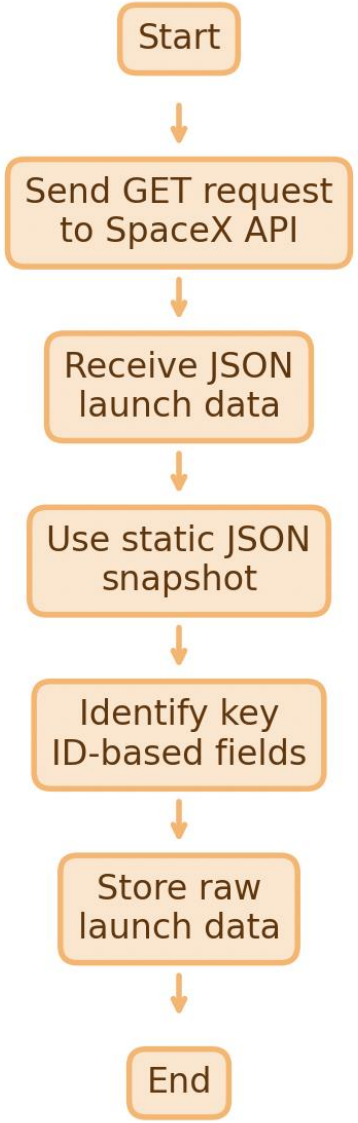
Data Collection

- Data for this project was collected using the public SpaceX REST API, specifically the v4/launches/past endpoint, which provides historical launch information in JSON format. To ensure consistency and reproducibility, a static snapshot of the API response was used instead of live requests.
- The initial dataset contained multiple identifier fields referencing rockets, payloads, launchpads, and booster cores. Additional SpaceX API endpoints were queried to extract meaningful information from these identifiers, including booster version, payload mass and orbit, launch site location, and core landing and reuse details.
- This process transformed raw API data into a structured dataset, forming the foundation for further data cleaning, analysis, and landing success prediction.
- Data Scraping is done to transform the raw data into a feature-rich engineered dataset.
- Data Wrangling is performed to remove unnecessary column, identify missing values and replace with the mean values instead, and retain only Falcon 9 reusable rockets for prediction modelling.

Data Collection – SpaceX API

- Data collection performed using public SpaceX REST API, which provided structured historical launch data in JSON format. Here, a **static** snapshot of API is used instead of live API calls.
- The initial API response consisted large number of nested fields and identifiers. Therefore, additional API endpoints such as (/v4/rockets, /v4/launchpads, /v4/cores etc.) were accessed.
- The collected data represents the raw factual record of Falcon launches with details on mission parameters, hardware, and landing attempts.

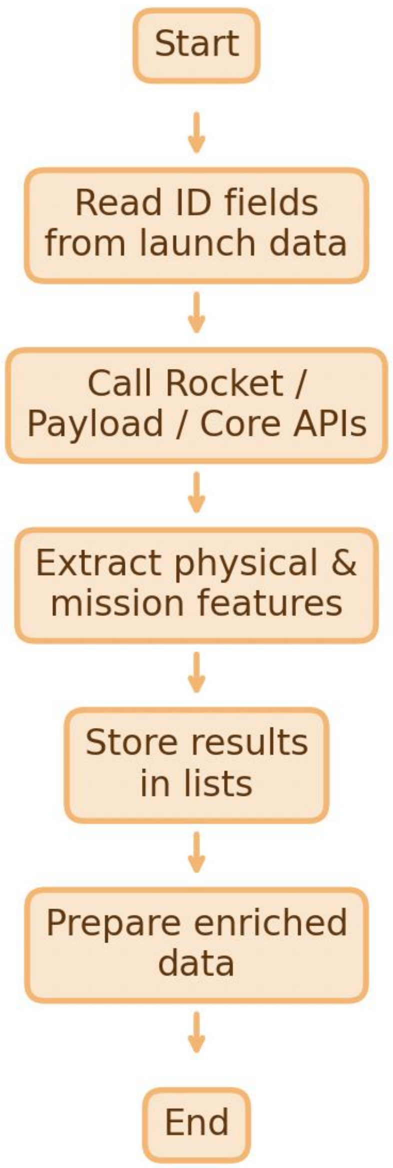
GitHub URL: https://github.com/marseaski/IBM-Applied-Data-Science-Capstone/blob/main/lab1-jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Post the API data collection, scraping was performed to obtain detailed information behind the ID references. Custom functions are written to retrieve info such as booster name, payload mass, orbit type, site etc.
- The **Wikipedia** webpage was requested using the requests library, and the HTML content was parsed using **BeautifulSoup**. The relevant launch table was identified, and column headers were extracted from table header (<th>) elements.
- Each table row was then parsed carefully to handle inconsistent formatting, references, and missing values.
- This scraping step transformed dataset from raw log of nested IDs to a feature rich engineered dataset.

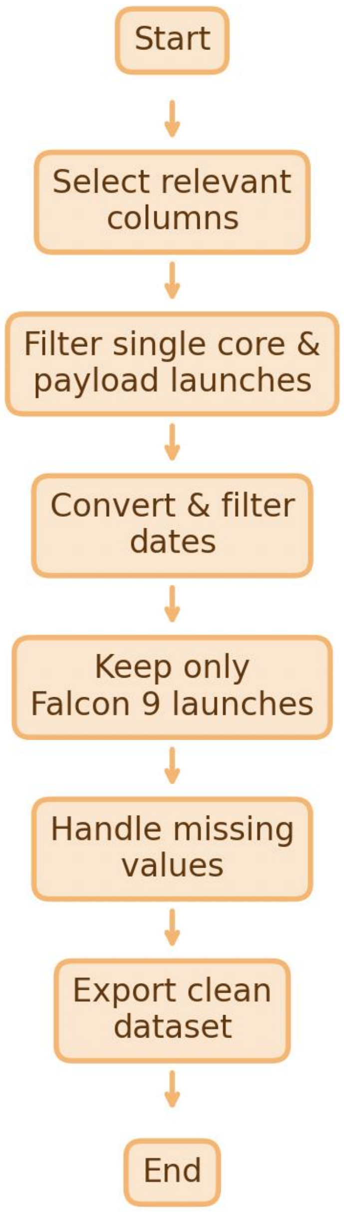
GitHub URL: https://github.com/marseaski/IBM-Applied-Data-Science-Capstone/blob/main/lab2-jupyter-labs-webscraping.ipynb



Data Wrangling

- Data Wrangling is done to convert the engineered dataset into clean, structured format suitable for feature engineering and ML models.
- Further, the dataset was filtered to done only to retain **Falcon 9** launches (as Falcon 1 are not reusable so not relevant to landing predictions).
- Understand launch patterns across sites, orbits, and landing outcomes. Landing result categories were mapped into a binary classification, successful landings were labelled as 1 and unsuccessful or no-attempt landings as 0.
- Missing data is handled using the mean values (payload column). Whereas, landing pad values are intentionally left. Finally, the dataset is exported as CSV file for EDA.

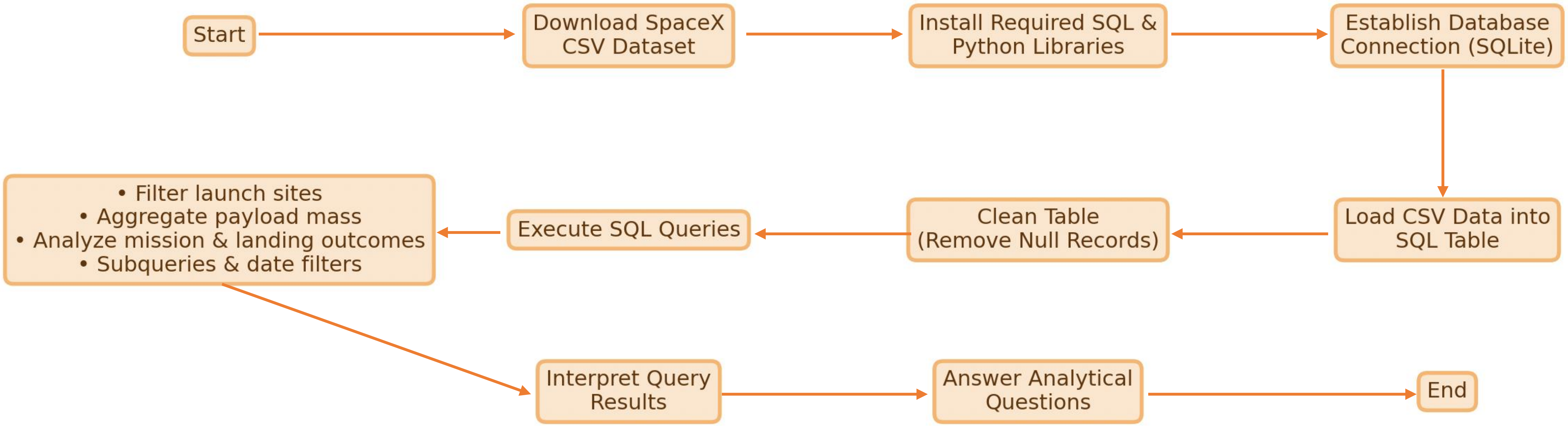
GitHub URL: https://github.com/marseaski/IBM-Applied-Data-Science-Capstone/blob/main/lab3-jupyter-spacex-Data%20wrangling.ipynb



EDA with SQL

- Dataset contains the records for payloads carried during SpaceX missions in detail. The primary objective was to understand the dataset structure, load it into a relational database, and perform structured queries to extract meaningful insights.
- CSV dataset is loaded into a **SQLite database.** A connection was established using **to_sql()** method.
- A series of SQL queries were then written to answer specific analytical questions. These queries involved:
    - Filtering records using **WHERE** and **LIKE** clauses
    - Aggregating data using functions such as **COUNT, SUM, AVG, MIN,** and **MAX**
    - Grouping results with **GROUP BY,** and applying subqueries to identify maximum payload missions.
- Temporal filtering was also performed to analyze landing outcomes. Through this process, SQL was used to explore launch patterns, booster performance, payload capacity trends, and mission success statistics.

EDA with SQL



GitHub URL: https://github.com/marseaski/IBM-Applied-Data-Science-Capstone/blob/main/lab4-jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with Data Visualization

- Exploratory Data Analysis is applied to uncover landing success patterns and performed feature engineering to transform mission data into a predictive modelling dataset.
- Relationships between variables such as flight number, payload mass, launch site, orbit type, and landing success were visually explored using Seaborn and Matplotlib:
    - Flight Number vs Payload Mass (Scatter / Categorical Plot)
    - Flight Number vs Launch Site (Scatter / Categorical Plot)
    - Payload Mass vs Launch Site (Scatter Plot)
    - Orbit Type vs Success Rate (Bar Type)
    - Flight Number vs Orbit Type (Scatter/Categorical Plot)
    - Payload Mass vs Orbit Type (Scatter Plot)
    - Year vs Average Success Rate (Line Chart)

GitHub URL: https://github.com/marseaski/IBM-Applied-Data-Science-Capstone/blob/main/lab5-edadataviz.ipynb

Build an Interactive Map with Folium

Folium Map objects were used, and their purpose are listed as follows:

- Folium **Map** object: Used as base interactive map to visualize geographic patterns of launch sites
- Circle markers **(folium.Circle):** Used to highlight physical location of each launch site
- Text markers **(folium.Marker):** Used to label each launch site directly on the map
- Marker Cluster **(MarkerCluster):** Group multiple launch records at the same coordinates
- Distance markers **(DivIcon):** Display numerical distance values (in km) between launch sites and nearby features such as coastlines, highways, railways, and cities.
- Polylines (**folium.PolyLine**): Visually connect launch sites to nearby geographical features and represent distance relationships spatially.
- Color coded markers: Visually differentiate successful launches (green) from failed (red).
- Mouse Position plugin: Interactively display latitude and longitude values when hovering

GitHub URL: https://github.com/marseaski/IBM-Applied-Data-Science-Capstone/blob/main/lab6-jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

The dashboard combines interactive filters and visualizations to analyze how launch site, payload mass, and booster type affect SpaceX launch success.

- **Launch Dropdown (interaction):** A dropdown menu allowing users to select *All Sites* or a specific launch site. *Enables interactive filtering so users can compare overall performance versus site-specific launch behaviour.*
- **Success Pie Chart (Plot):** A pie chart showing Total successful launches by site and Success vs failure distribution. *Provides an intuitive, high-level comparison of launch success rates across sites and within a single site.*
- **Payload Range Slider (Interaction):** A range slider to filter launches based on payload mass (kg). *Allows users to dynamically explore how payload mass influences launch success.*
- **Payload vs Launch Outcome (Scatter Plot):** Showing payload mass vs launch success (class), coloured by booster version category. *Identify correlations between payload weight, booster type, and launch success across sites.*
- **Linked Interactivity (Dropdown + Slider):** Callback-driven interaction. *Enables multi-dimensional analysis by combining site selection and payload filtering in real time.*

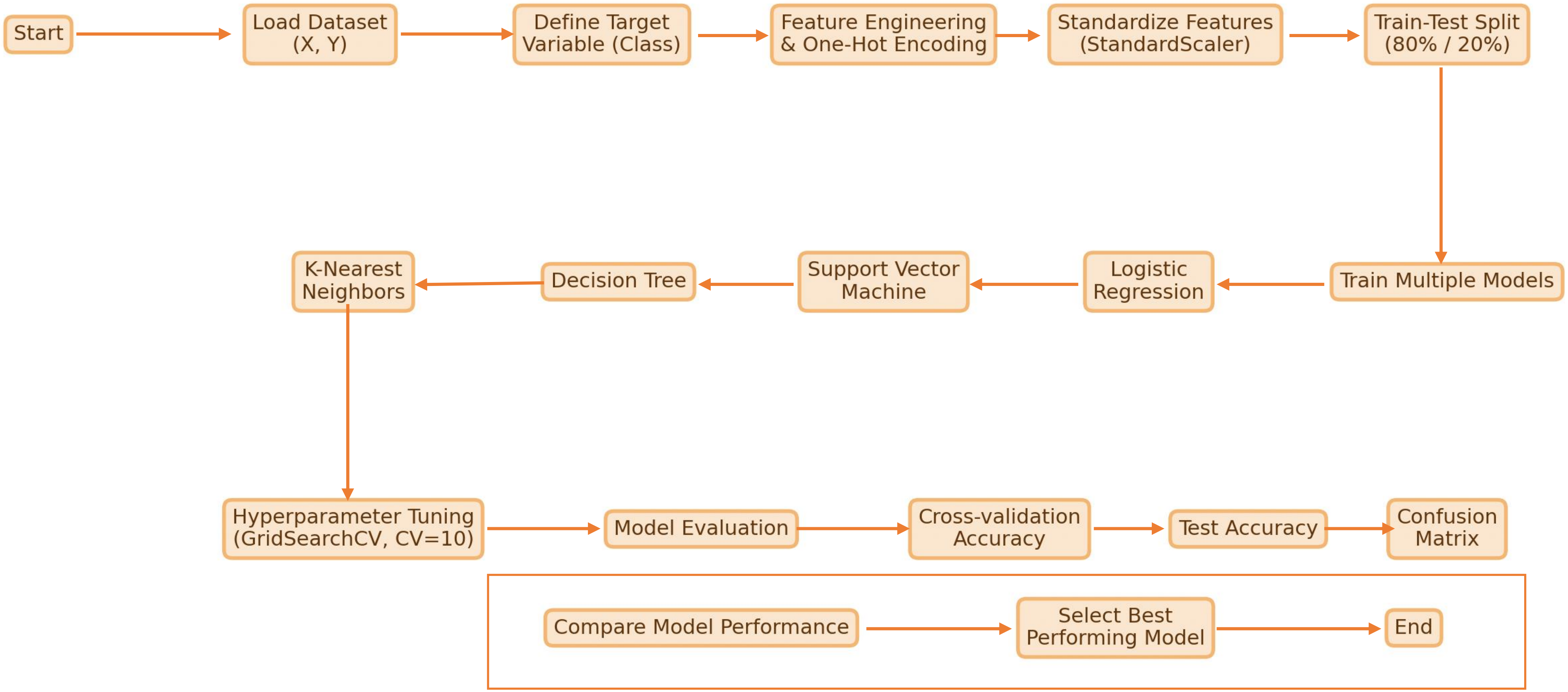
GitHub URL: https://github.com/marseaski/IBM-Applied-Data-Science-Capstone/blob/main/lab7-interactive_dashboard_plotly-dash.ipynb

Predictive Analysis (Classification)

A systematic ML pipeline with standardized features, cross-validated hyperparameter tuning, and multiple evaluation metrics was used to identify the best-performing Falcon 9 landing prediction model.

- **Problem Framework:** Defined the target variable *Class* to represent whether the Falcon 9 first stage landed successfully (1) or not (0).
- **Feature Preparation:** Used engineered numerical and one-hot encoded features and applied *StandardScaler* to normalize feature magnitudes for fair model learning.
- **Train-Test split of Dataset:** Split data into training (80%) and testing (20%) sets to evaluate generalization performance.
- **Model Building:** Trained four classification models: *Logistic Regression, Support Vector Machine, Decision Tree, and k-nearest neighbors.*
- **Hyperparameter Tuning:** Applied *GridSearchCV with 10-fold cross-validation* to find optimal hyperparameters for each model.
- **Model Evaluation:** Evaluated models using *cross-validation accuracy, test accuracy, and confusion matrices* to analyze misclassification patterns.
- **Model Comparison and Selection:** Compared models based on validation accuracy and test performance

Predictive Analysis (Classification)



GitHub URL: https://github.com/marseaski/IBM-Applied-Data-Science-Capstone/blob/main/lab8-SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

Section 2

Insights drawn from EDA

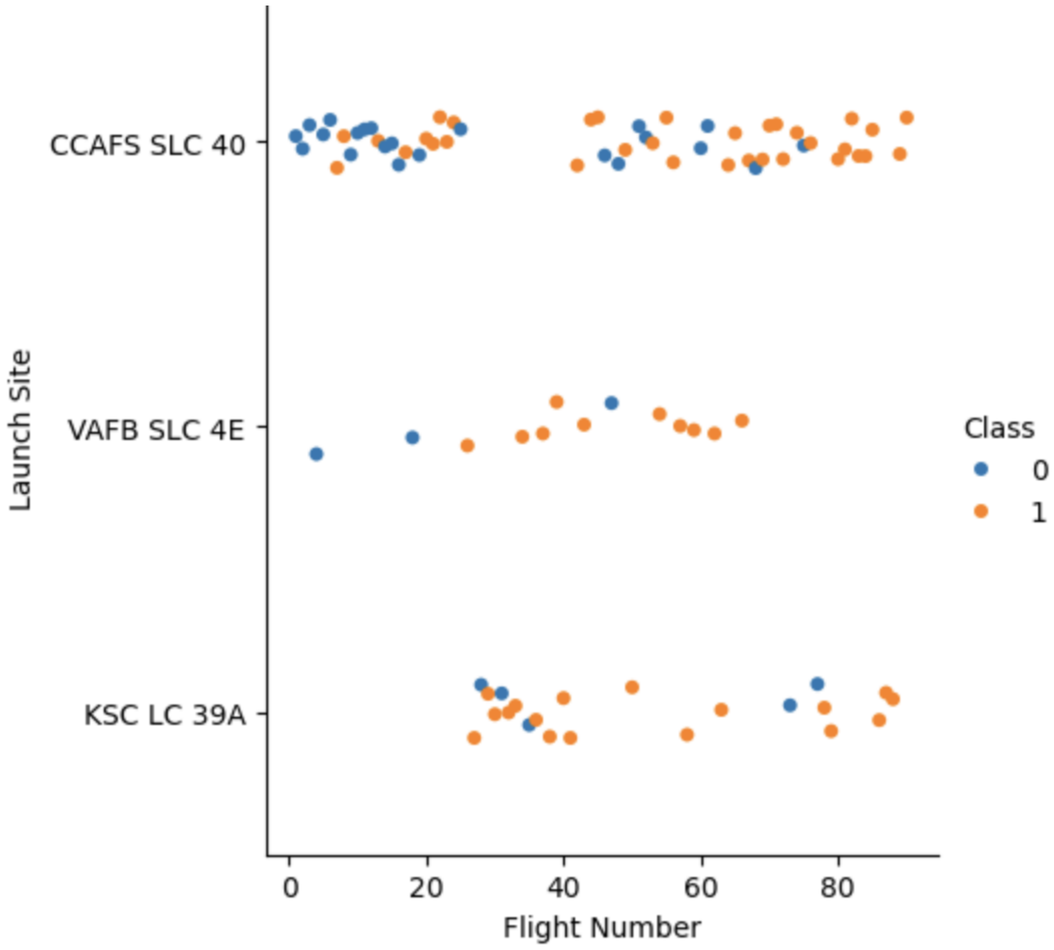
Flight Number vs. Launch Site

Plot signifies that as flight numbers increase, successful launches (orange) become more frequent.

Launch Site Comparison:

- **CCAFS SLC 40** has the most launches and shows improving success over time.
- **KSC LC 39A** mostly appears at higher flight numbers with high success rates.
- **VAFB SLC 4E** has fewer launches, with mixed outcomes.

Plot Insight: Launch success improves with experience and later flights.



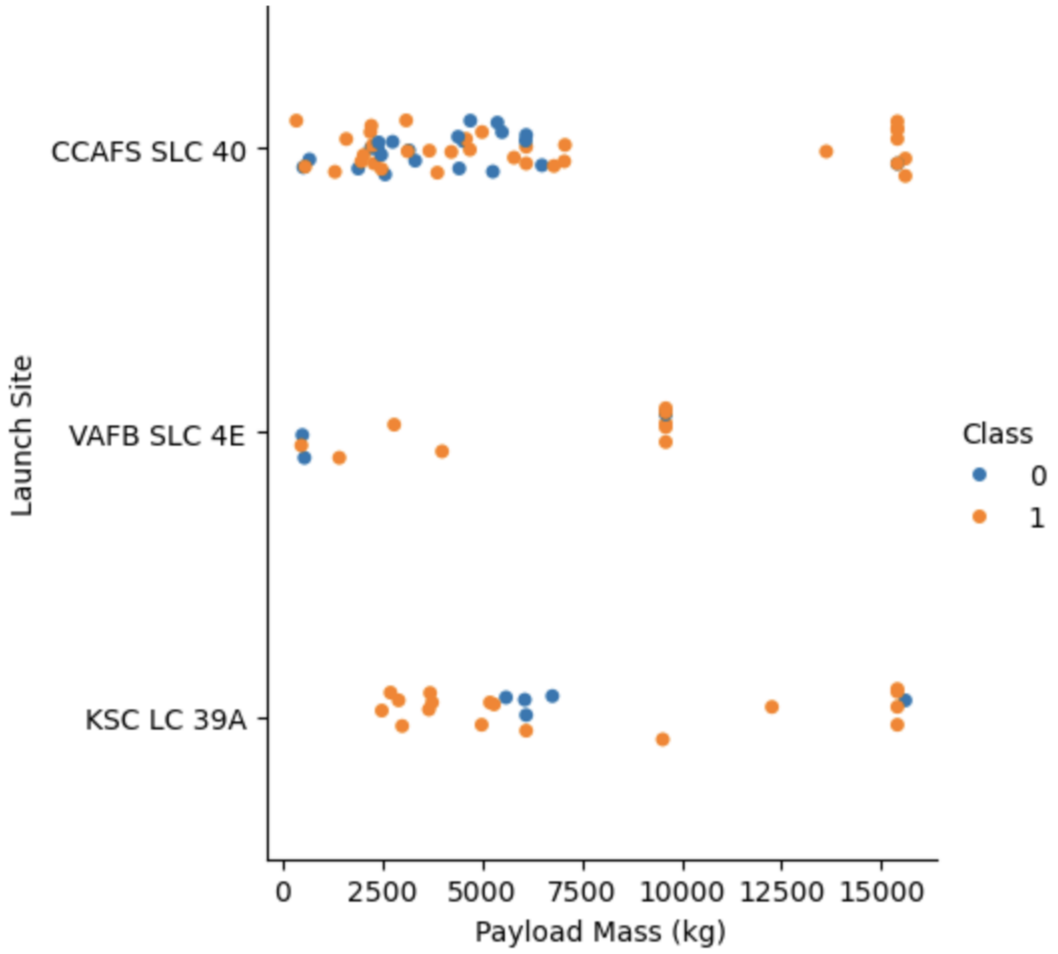
Payload vs. Launch Site

Plot signifies that successful launches (orange) dominate across most payload masses.

Launch Site Comparison:

- **CCAFS SLC 40** Wide range of payloads, mostly successful, including very heavy payloads.
- **KSC LC 39A** Medium to heavy payloads with high success rates.
- **VAFB SLC 4E** has fewer launches, generally lighter to mid-range payloads, mostly successful.

Plot Insight: Payload mass alone does not strongly limit success; high-mass payloads are often launched successfully



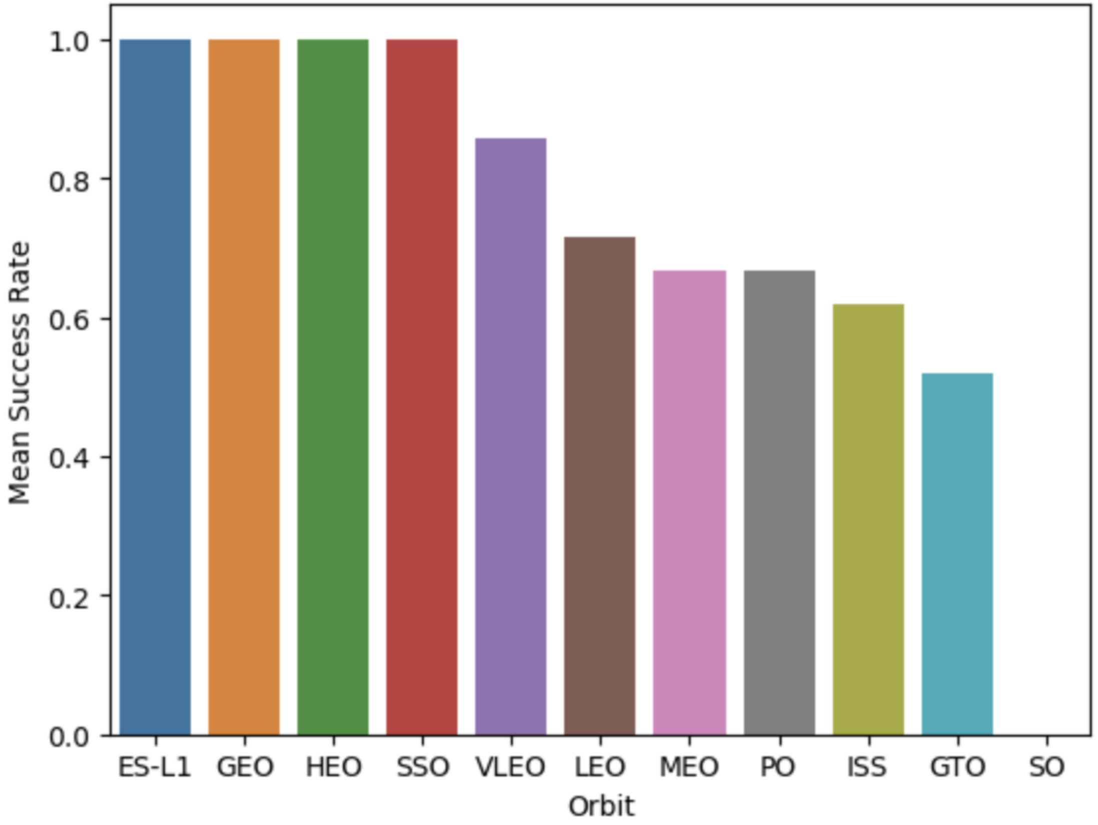
Success Rate vs. Orbit Type

Plot signifies mean success rate of the launches in the various orbits

Success with orbit type comparison:

- **Superior success:** ES-L1, GEO, HEO, SSO. *(100% success rate)*
- **High success:** VLEO (~0.86).
- **Moderate success:** LEO (~0.72), MEO (~0.67), PO (~0.67).
- **Lower success:** ISS (~0.62), GTO (~0.52).
- **Inferior to all:** SO (*0% success rate*)

Plot Insight: Mission success varies by orbit, with more complex or demanding orbits showing lower average success rates.



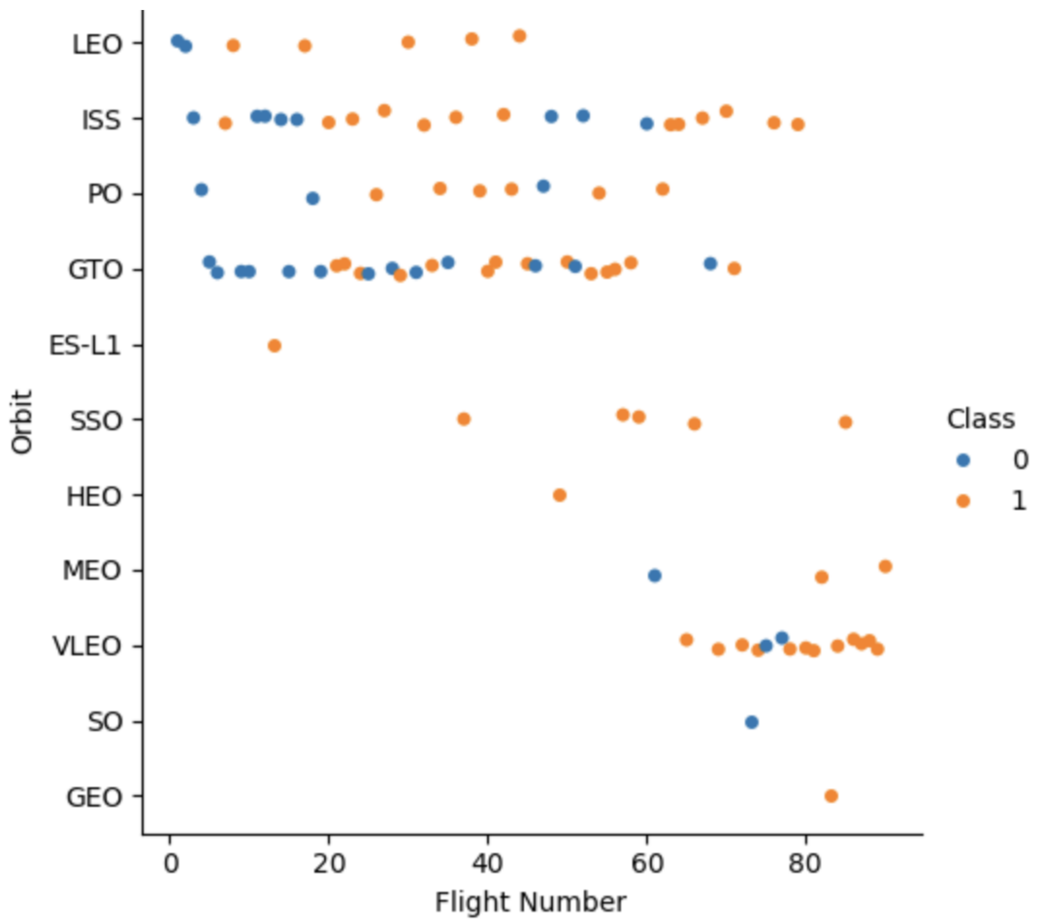
Flight Number vs. Orbit Type

Plot signifies that later flights show a higher proportion of successful launches across most orbits.

Flight number and Orbit Comparison:

- **LEO, ISS, GTO:** Most common orbits with mixed early results and improved success later.
- **SSO, ES-L1, HEO:** Fewer missions, mostly successful.
- **VLEO:** Appears only in later flights with high success.

Plot Insight: Launch success improves with experience, and success rates vary by orbit complexity and mission type.



Payload vs. Orbit Type

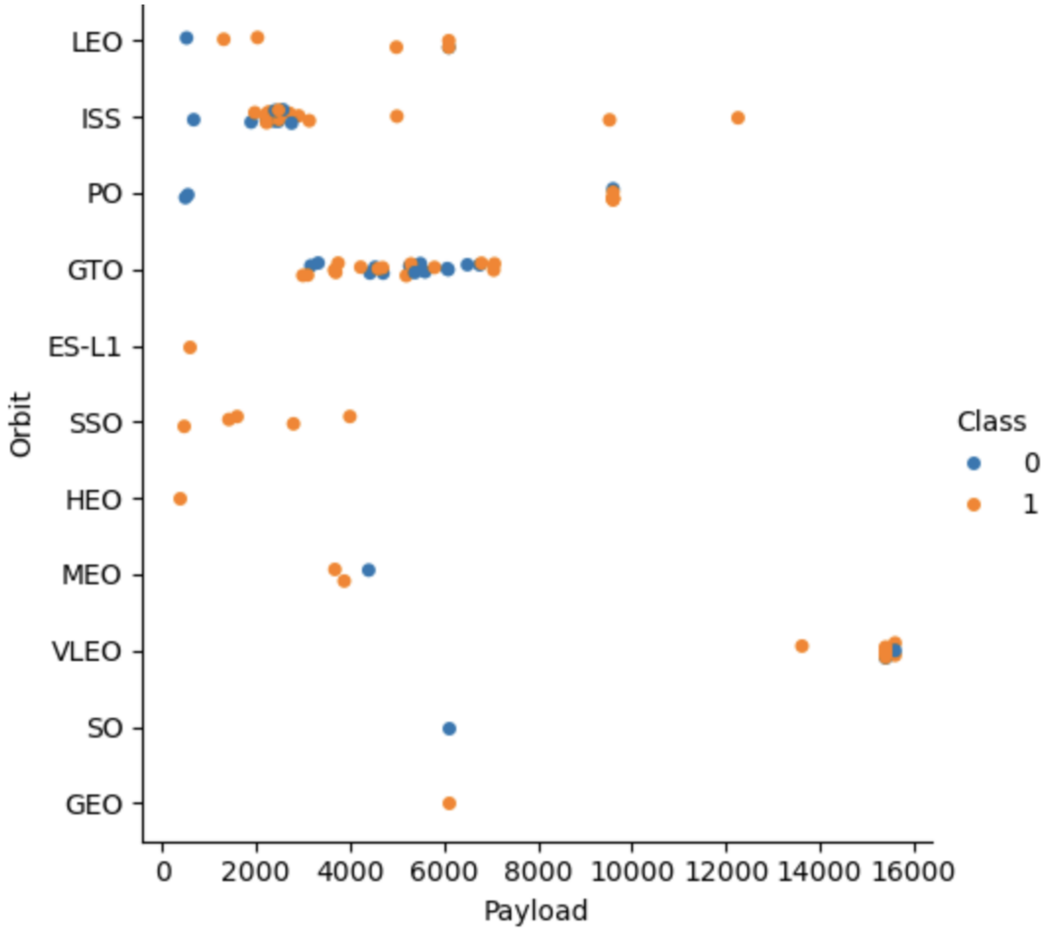
Plot signifies that different orbits cluster around typical payload ranges (e.g., GTO mid-range, VLEO very heavy).

Further, most payload–orbit combinations are successful (orange dominates).

Payload and orbit comparison:

- **GTO & ISS:** Many launches with mixed but mostly successful outcomes.
- **SSO, HEO, ES-L1:** Fewer missions, largely successful.
- **VLEO:** Very heavy payloads with high success.

Plot Insight: Success depends more on orbit type and mission maturity than payload mass alone



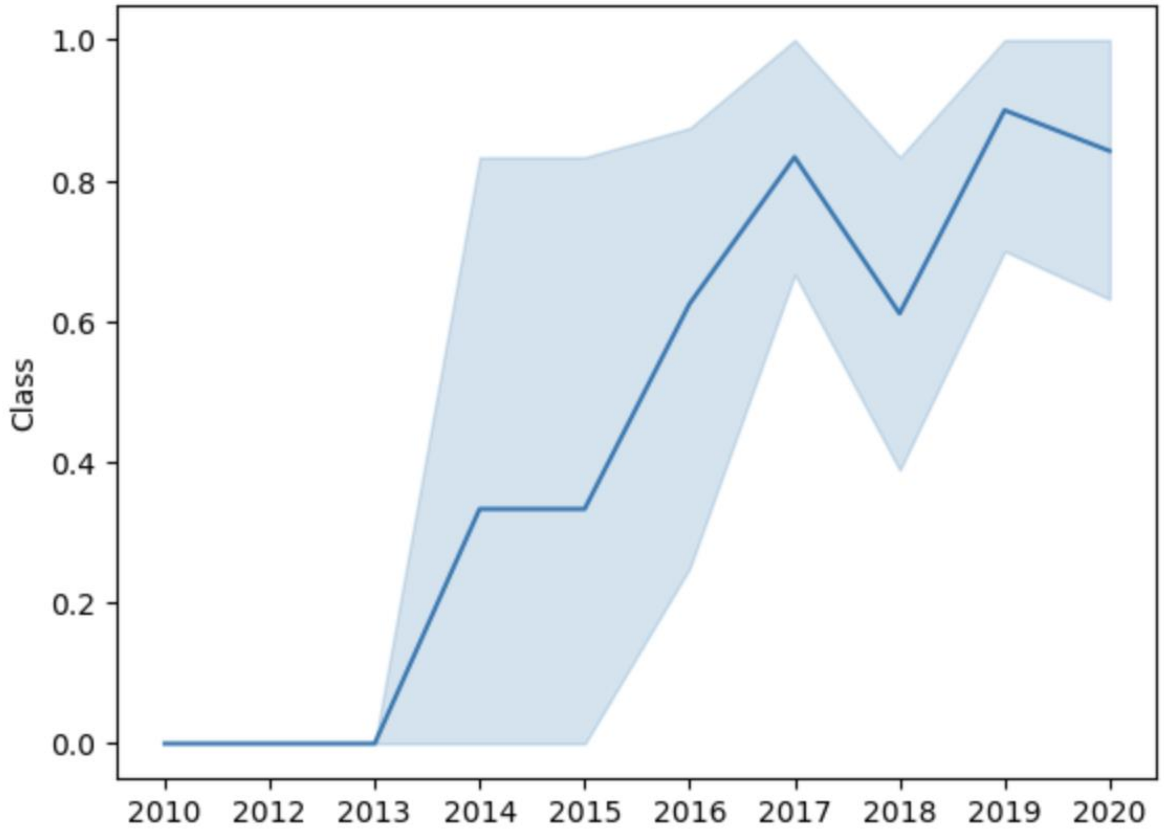
Launch Success Yearly Trend

Plot signifies that average launch success per year. Further, most payload–orbit combinations are successful (orange dominates).

Launch Success Trend:

- **Early years (2010–2013):** Mostly unsuccessful launches.
- **Mid period (2014–2016):** Rapid improvement in success rate.
- **Later years (2017–2020):** High and consistent success, with minor fluctuations.

Plot Insight: Launch success improves significantly over time, showing growing reliability and operational maturity.



All Launch Site Names

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

In [12]:

```
df.head()
%sql select distinct launch_site from SPACEXTABLE
```

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

Out[12]:

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

Select distinct: returns only unique values (removes duplicates).
Launch site: column containing launch site names.
SPACEXTABLE: table storing SpaceX mission data.

This query leads to a list of all unique launch sites used in the dataset

Launch Site Names Begin with 'CCA'

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

In [15]:
```
%sql select * from SPACEXTABLE where launch_site like 'CCA%' LIMIT 5
```

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

Out[15]:

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

Total Payload Mass

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

```
In [21]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer like 'NASA(CRS)%'

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

Out[21]:

sum(PAYLOAD_MASS__KG_)
45596

The SQL query calculates the total payload mass (in kilograms) for all SpaceX missions carried out for NASA Commercial Resupply Services (CRS).

- It uses SUM(PAYLOAD_MASS__KG_) to add together the payload masses of all matching launches in the SPACEXTABLE
- while the WHERE Customer LIKE 'NASA(CRS)%' condition filters the records to include only missions whose customer name starts with NASA(CRS).

Average Payload Mass by F9 v1.1

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

```
In [19]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where booster_version like 'F9 v1.1%'

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

Out[19]:

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

The SQL query calculates the total payload mass (in kilograms) for all SpaceX missions carried out by booster version F9 v1.1

- It uses avg(PAYLOAD_MASS__KG_) to add together the payload masses of all matching launches in the SPACEXTABLE
- while the WHERE Customer LIKE 'F9 v1.1%' condition filters the records to include only missions of booster versions.

First Successful Ground Landing Date

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
[20]: %sql SELECT MIN("Date") FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

[20]:

MIN("Date")
2015-12-22

The SQL query finds the earliest date of a successful mission in the dataset.
It uses MIN(date) to return the oldest launch date from SPACEXTABLE, while the WHERE MISSION_OUTCOME = 'Success' condition filters the data to include only missions that were completed successfully.

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
[19]: ISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

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

[19]:

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

The SQL query retrieves the unique booster versions that successfully landed on a drone ship while carrying a payload between 4,000 kg and 6,000 kg. The SELECT DISTINCT clause ensures only unique booster versions are listed, and the WHERE conditions filter the records to include only launches with a *successful drone ship landing* and payload masses within the specified range.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

[24]:
```
%sql select mission_outcome, count(*) from SPACEXTABLE group by mission_outcome
```

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

[24]:

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

The SQL query summarizes the launch results by mission outcome. It groups all records in SPACEXTABLE by mission_outcome and uses COUNT(*) to calculate how many missions fall into each outcome category, producing a simple count of successes, failures, and any other recorded outcomes.

Boosters Carried Maximum Payload

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

[25]:
```
%sql select booster_version, PAYLOAD_MASS__KG_ from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

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

[25]:

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

The SQL query returns the booster version(s) that carried the heaviest payload, by selecting rows whose payload mass equals the maximum payload mass in the table.

2015 Launch Records

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[28]:
```
%sql select Landing_Outcome, booster_version, launch_site from SPACEXTABLE where Date like '2015%'
```

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

[28]:

Landing_Outcome	Booster_Version	Launch_Site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
No attempt	F9 v1.1 B1014	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
No attempt	F9 v1.1 B1016	CCAFS LC-40
Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
Success (ground pad)	F9 FT B1019	CCAFS LC-40

The SQL query retrieves the landing outcome, booster version, and launch site for all SpaceX launches that occurred in 2015.

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

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, in descending order.

[29]:
```
%sql select Landing_Outcome, count(*) from (select * from SPACEXTABLE where date between '2010-06-04' and '2017-03-20') group by Landing_Outcome order by
```

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

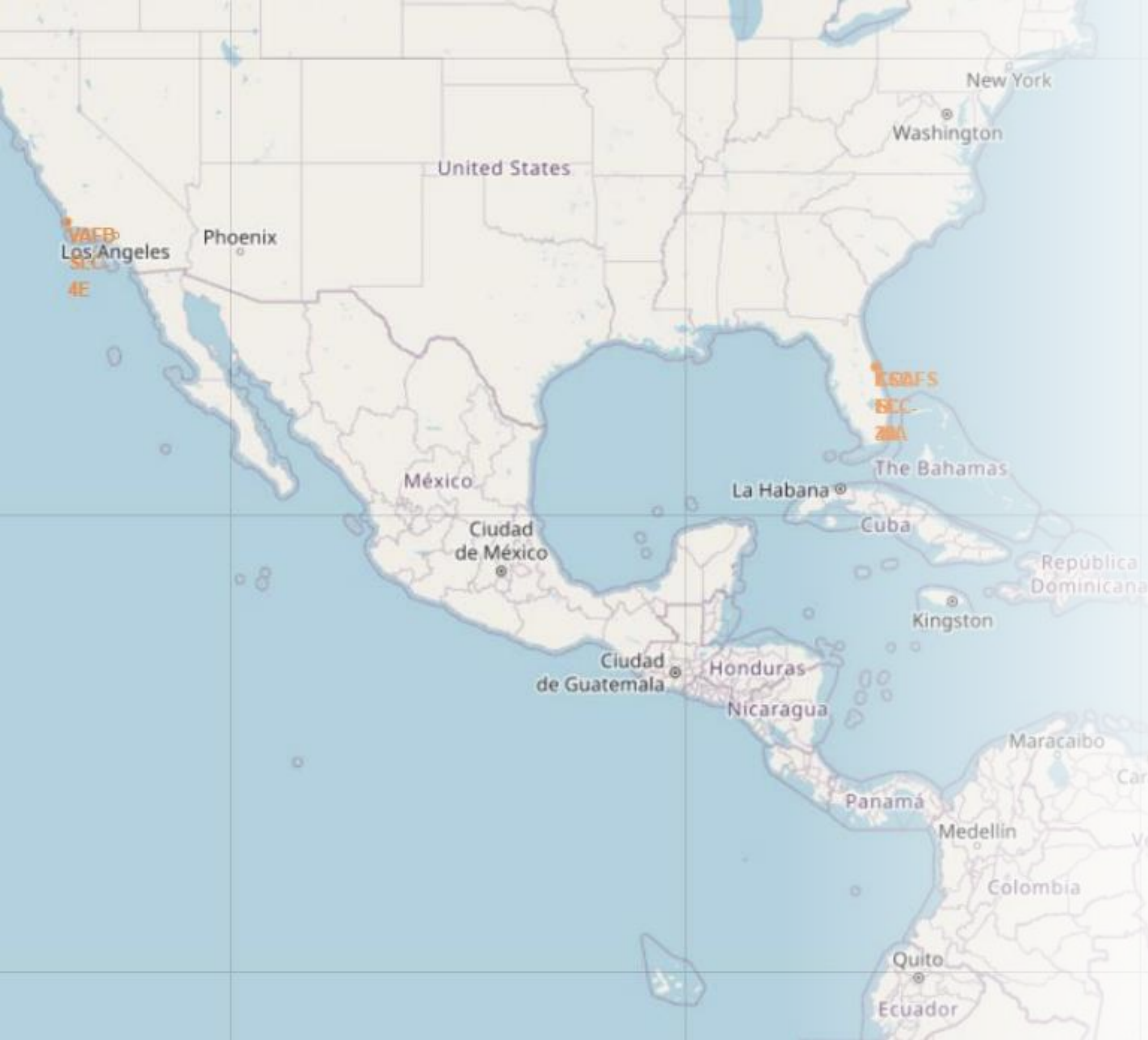
[29]:

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

The SQL query counts how many launches had each landing outcome between June 4, 2010 and March 20, 2017, and lists the results in descending order of landing outcome.

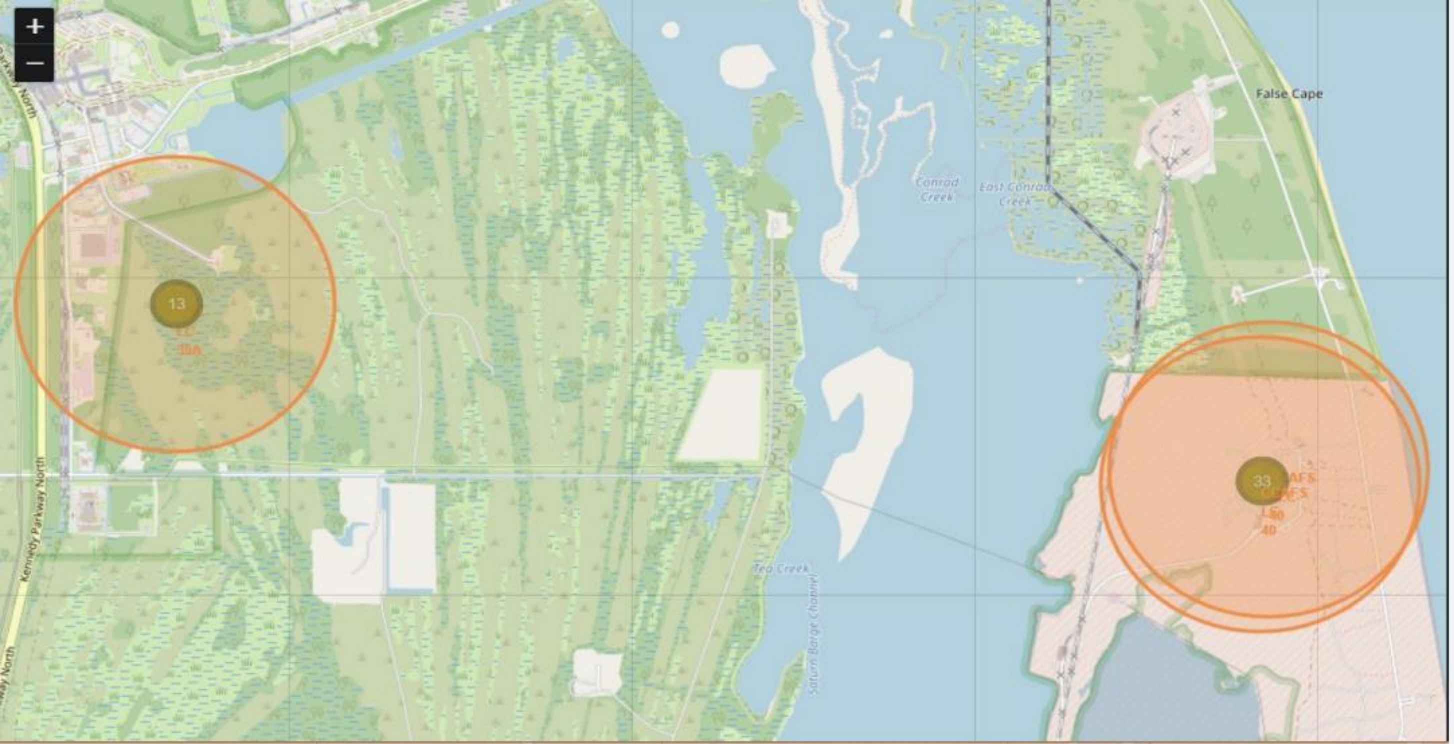
Section 3

Launch Sites Proximities Analysis



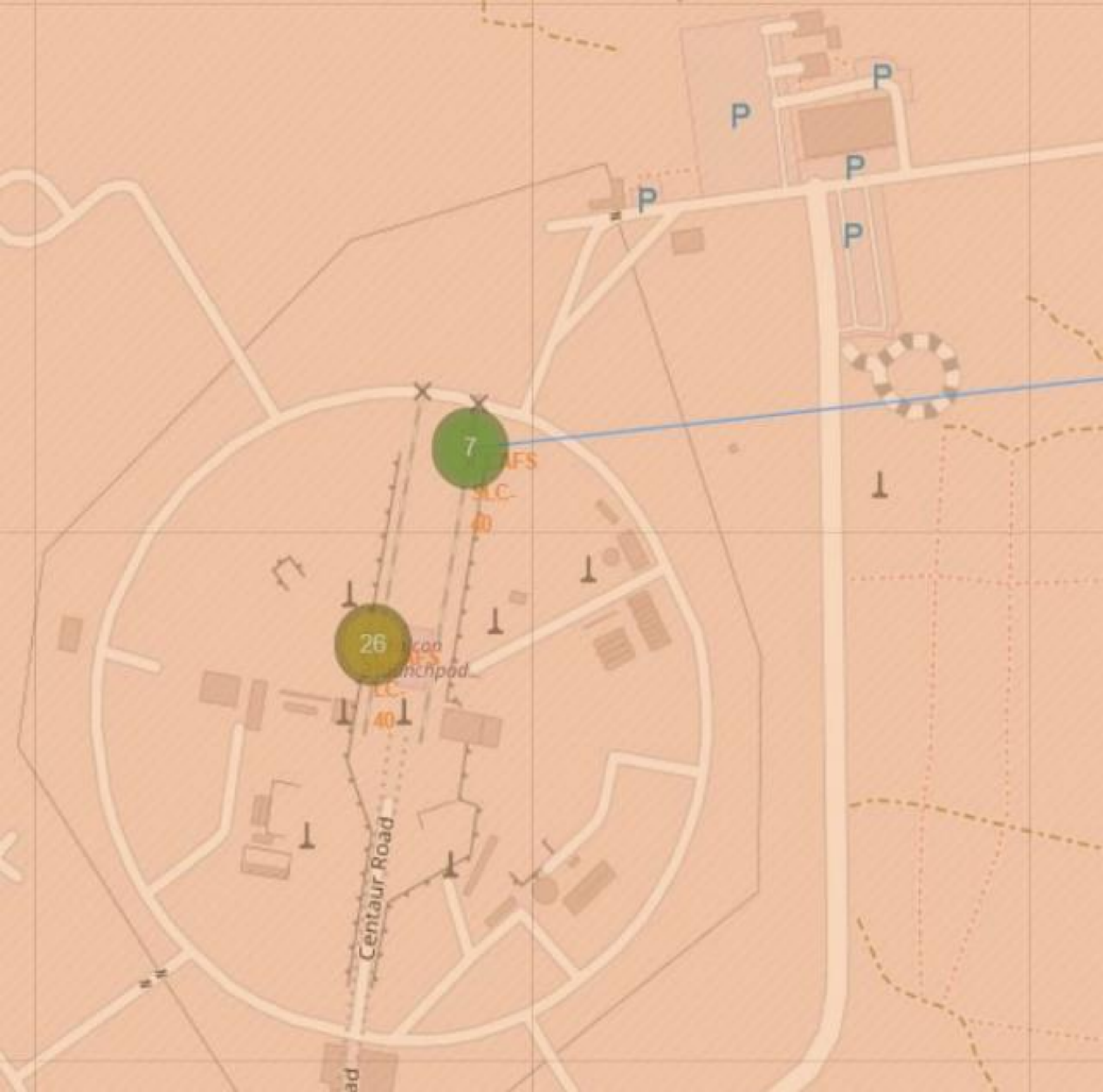
Folium: Marking Launch Sites

- Proximity to the Equator
  Not every launch site is located near the Equator. For example, Vandenberg Air Force Base (VAFB SLC-4E) sits at about 34.63° latitude, placing it much farther from the Equator than the Florida-based launch sites.
- Proximity to the Coast
  All launch sites are positioned close to the coastline. The Florida sites such as Cape Canaveral (CCAFS LC-40 / SLC-40) and Kennedy Space Center (KSC LC-39A) are located along the Atlantic coast, while Vandenberg Air Force Base (VAFB SLC-4E) is situated near the Pacific coast in California.



Folium: Success & Failures

- Clustered markers improve visualization by reducing clutter and making it easier to spot patterns in SpaceX launch data through color-coding and interactive details.
- For example, at CCAFS LC-40, 26 launches are shown with 19 red (unsuccessful) and 7 green (successful) markers, allowing quick evaluation of launch performance.



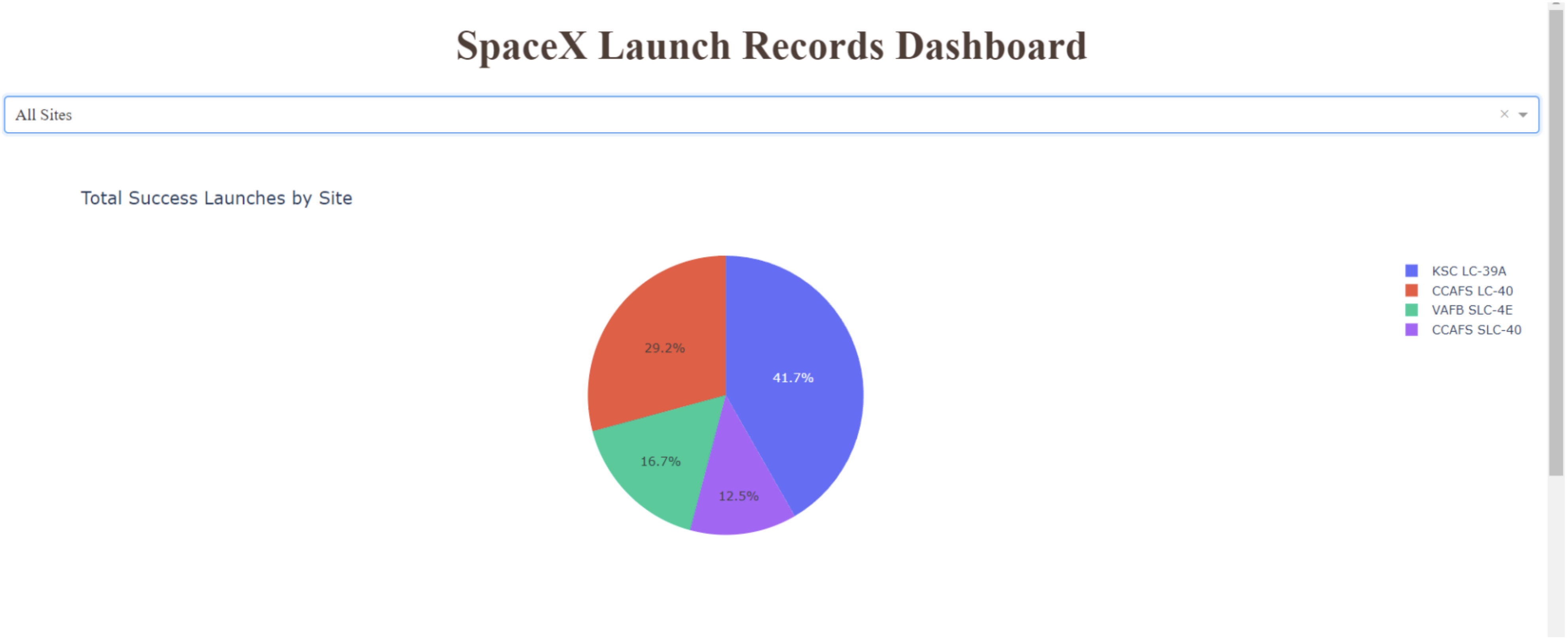
Folium: Launch site Proximity

- This visualization illustrates the distance from the CCAFS SLC-40 launch site to the nearest coastline, with the marker showing an approximate distance of 0.51 km.
- The straight-line PolyLine clearly emphasizes how close the launch complex is to the shore. Such coastal placement significantly reduces potential risks to nearby populated regions while enhancing overall launch safety.

Section 4

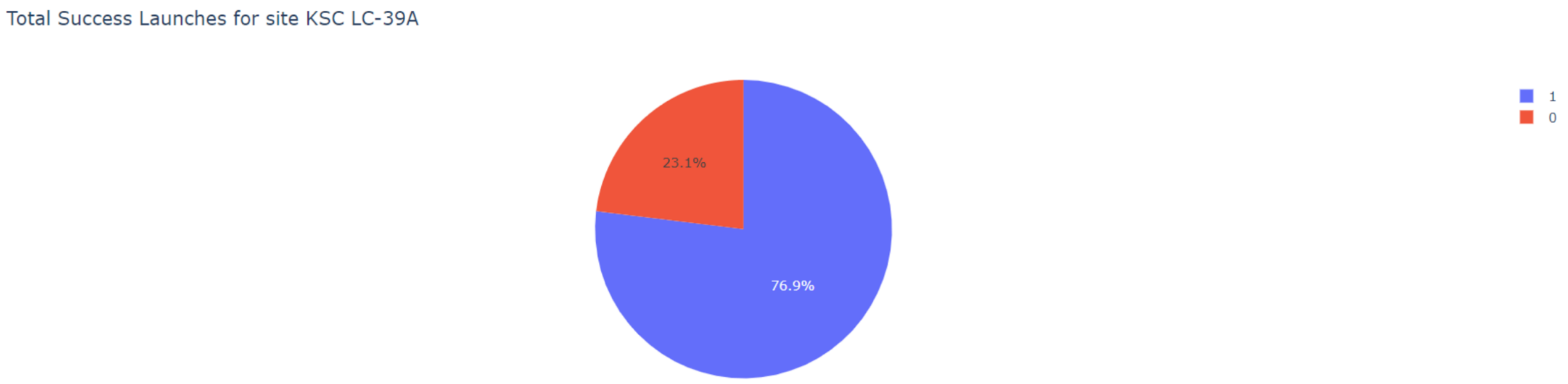
Build a Dashboard with Plotly Dash

Dashboard Launch Records



CCAFS LC-40: 29.2%, CCAFS SLC-40: 12.5%, VAFB SLC-4E: 16.7%, and KSC LC-39A: 41.7%
Among all launch sites, KSC LC-39A accounts for the largest share of successful launches at 41.7%, highlighting it as SpaceX's most dependable and consistently successful launch location.

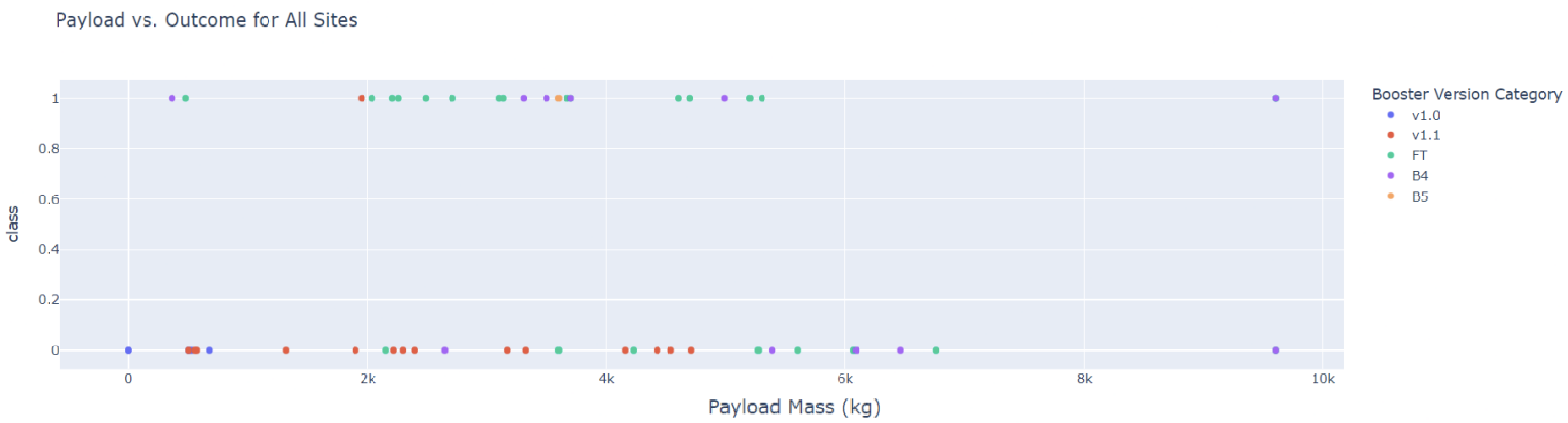
Dashboard Successful Launches



Key Insights:

- The large share of successful missions from KSC LC-39A demonstrates its strong performance and dependability as a launch facility.
- At KSC LC-39A, successful launches (Class 1) account for 76.9%, while unsuccessful launches (Class 0) make up 23.1%.
- This notably high success rate further confirms KSC LC-39A as one of SpaceX's most reliable launch sites.

Dashboard Payload Outcomes



Booster Version Performance:

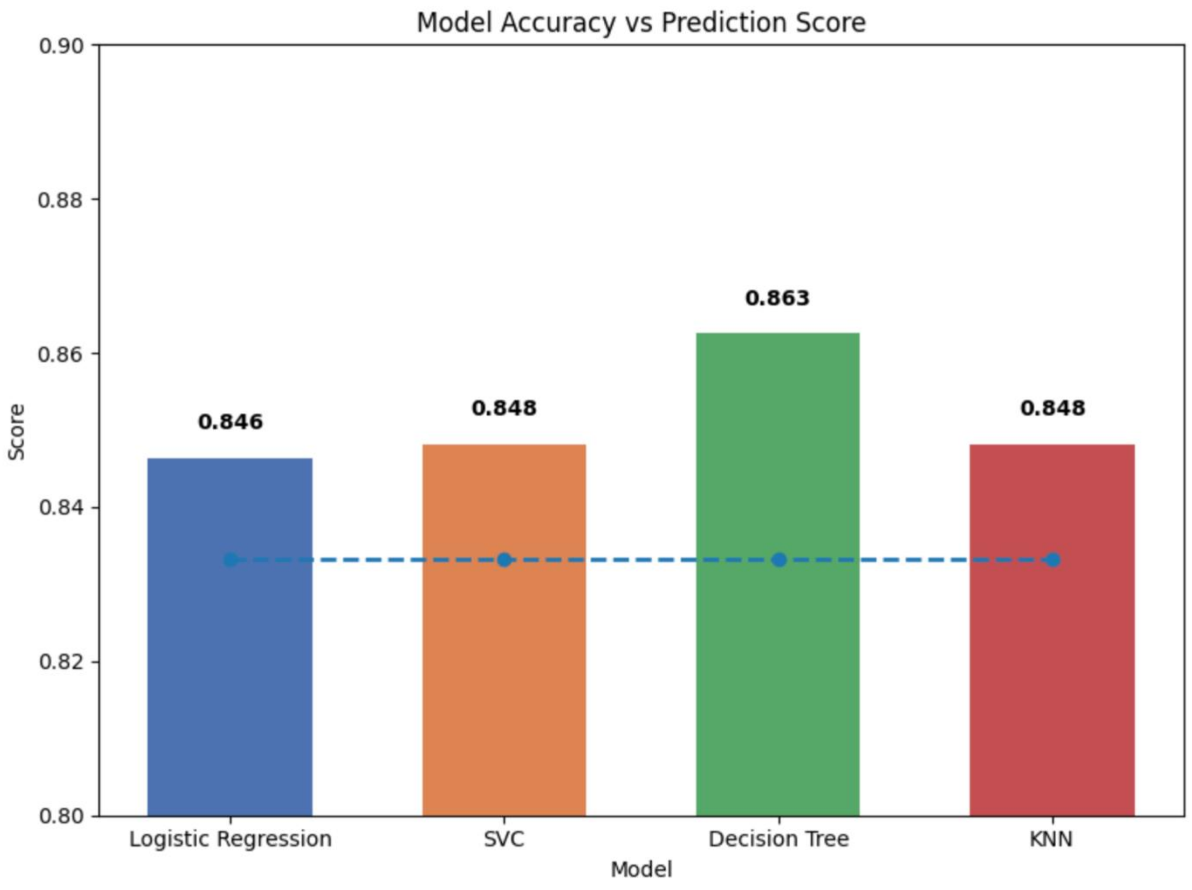
- The **FT** booster is the most commonly flown variant and demonstrates a strong success rate across a wide range of payload masses.
- **v1.0** booster has been used less often, making it harder to draw firm conclusions without additional data.
- Overall, there is no clear evidence that increasing payload mass leads to reduced launch success across booster versions.

Section 5

Predictive Analysis (Classification)

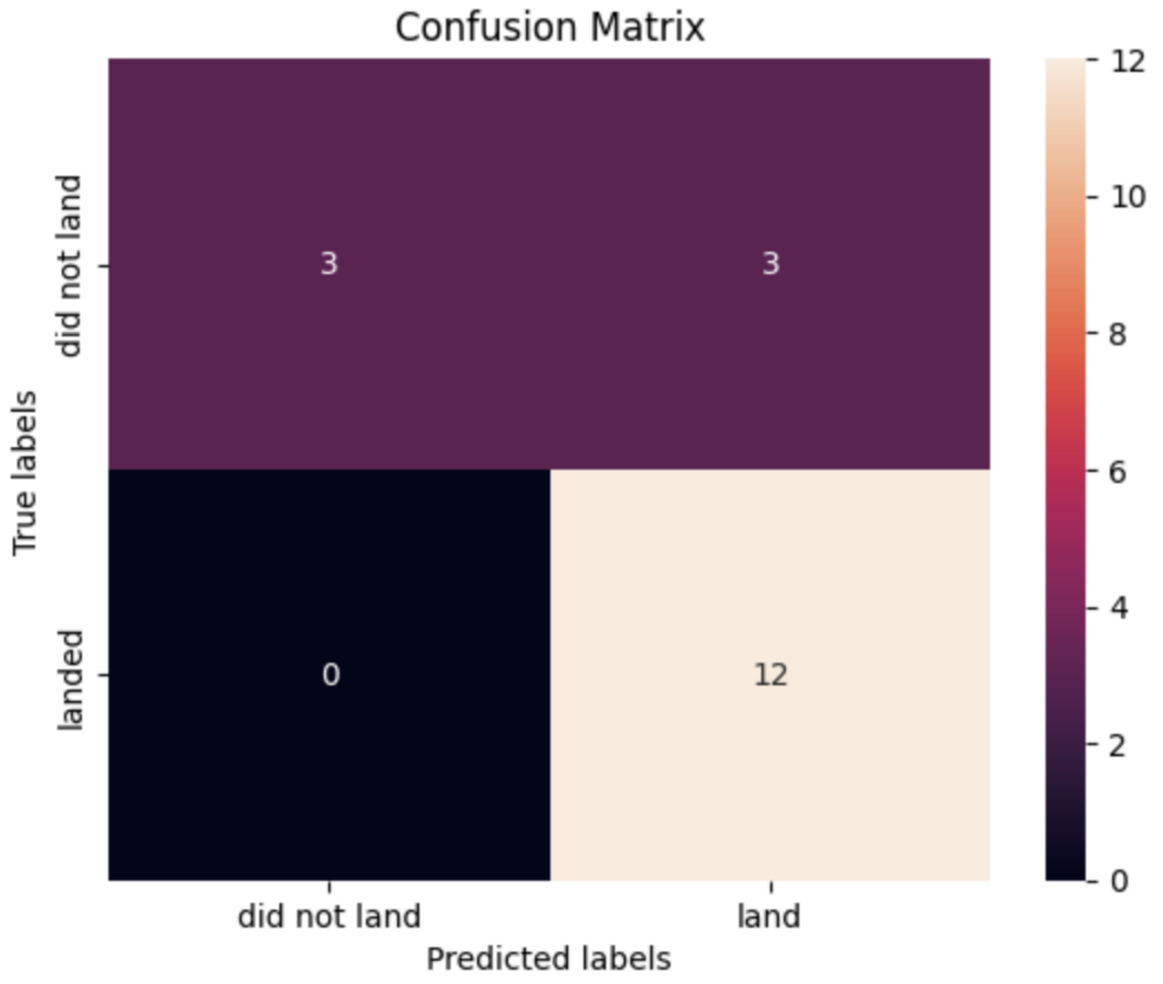
Classification Accuracy

- **Decision Tree stands out** with the highest accuracy (~0.863), showing better ability to capture complex patterns in the data.
- **Logistic Regression, SVC, and KNN perform similarly**, with accuracies clustered around 0.846–0.848, indicating comparable effectiveness.
- **Prediction scores are slightly lower than accuracies**, suggesting the models classify well but may need improved probability calibration.



Confusion Matrix

- Strong performance on successful landings: The model correctly predicts all successful landings (12 true positives), with no cases where a landed booster was misclassified as not landing.
- Errors in failed landing predictions: Out of 6 actual failed landings, only 3 are correctly identified, while 3 are incorrectly predicted as successful, indicating confusion in detecting failures.
- Bias toward predicting landings: The absence of false negatives and presence of false positives suggest the model tends to favour predicting a successful landing over a failure.



Conclusions

Summary & Conclusion:

- **CCAFS LC-40** stands out as the most successful launch site, contributing **43.7% of all successful launches**, suggesting favorable operational and environmental conditions.

- The **FT booster version** consistently performs well across different payload masses, highlighting its reliability and making it a strong choice for future missions.

- No strong relationship was found between payload mass and launch success, indicating that **booster type and launch site** have a greater impact on outcomes.

- Interactive visualizations built with **Folium and Plotly Dash** enhanced understanding of launch patterns and supported data-driven decision-making.

Overall, the analysis identifies key drivers of SpaceX launch success and provides a solid analytical foundation to support improved launch planning and the continued advancement of reusable rocket technology.

Thank you!