IBM Developer
SKILLS NETWORK

Digital LEARNING
& SKILLS ENRICHMENT INITIATIVE

Winning Space Race with Data Science

Mehwish Younus

Outline

Executive Summary

Summary of Methodologies

- **Data Preprocessing:** Cleaned and transformed the raw dataset to ensure it was suitable for analysis and model training.
- **Feature Engineering**: Created and selected relevant features to improve model performance and predictive power.
- **Model Selection:** Tested and compared multiple machine learning algorithms to identify the best-performing model.
- **Model Evaluation:** Used metrics such as accuracy score to evaluate model performance.
- **Deployment Consideration:** Explored how the final model could be applied in real-world scenarios to assist in SpaceX launch cost estimation.

Executive Summary

Summary of all results

- SpaceX has **significantly improved** landing success over time.
- **Orbit type** impacts success more than payload mass.
- More data could enhance **model performance** and algorithm differentiation.
- Current features show consistent predictive power.

Introduction

Project Background and Context

In this capstone project, we aim to predict whether the first stage of the SpaceX Falcon 9 rocket will successfully land. SpaceX offers launches at a significantly reduced cost—around $62 million—compared to other providers that charge upwards of $165 million. A key factor in this cost advantage is the reusability of the Falcon 9's first stage.

By predicting landing success, we can estimate the true cost of a launch. This insight can be especially valuable to competitors or stakeholders evaluating bids or comparing launch providers. Understanding the likelihood of first-stage recovery is essential in assessing the economics of space launches.

Introduction

Problems We Aim to Solve

- What factors influence the success or failure of a Falcon 9 first stage landing?
- Are there any observable patterns or trends in the launch data?
- What percentage of first-stage landings are successful versus unsuccessful?
- Which machine learning model provides the most accurate prediction of landing outcomes?

Section 1

Methodology

Methodology

Data Sources:

- Retrieved launch data using API requests to the official SpaceX API.
- Supplemented data with web scraping from Wikipedia's "List of Falcon 9 and Falcon Heavy launches" page.

Data Wrangling & Preparation:

- Cleaned and structured the data for analysis.
- Aggregated statistics such as: Launch counts per site, Orbit types and their frequencies, Mission outcome distributions.

Feature Engineering:

- Created a binary landing outcome label:1 for successful/soft landings 0 for failures/hard landings.

Exploratory Data Analysis (EDA):

- Conducted visual analysis using Matplotlib, Seaborn, and SQL queries.
- Developed interactive visualizations with Folium (for mapping launch sites) and Plotly Dash (for dynamic dashboards).

Predictive Modeling:

- Applied classification algorithms: Decision Tree, SVM, Logistic Regression, and K-Nearest Neighbors.
- Tuned model hyper parameters using cross-validation to enhance accuracy.

Data Collection

We collected data from two primary sources:

1. SpaceX API

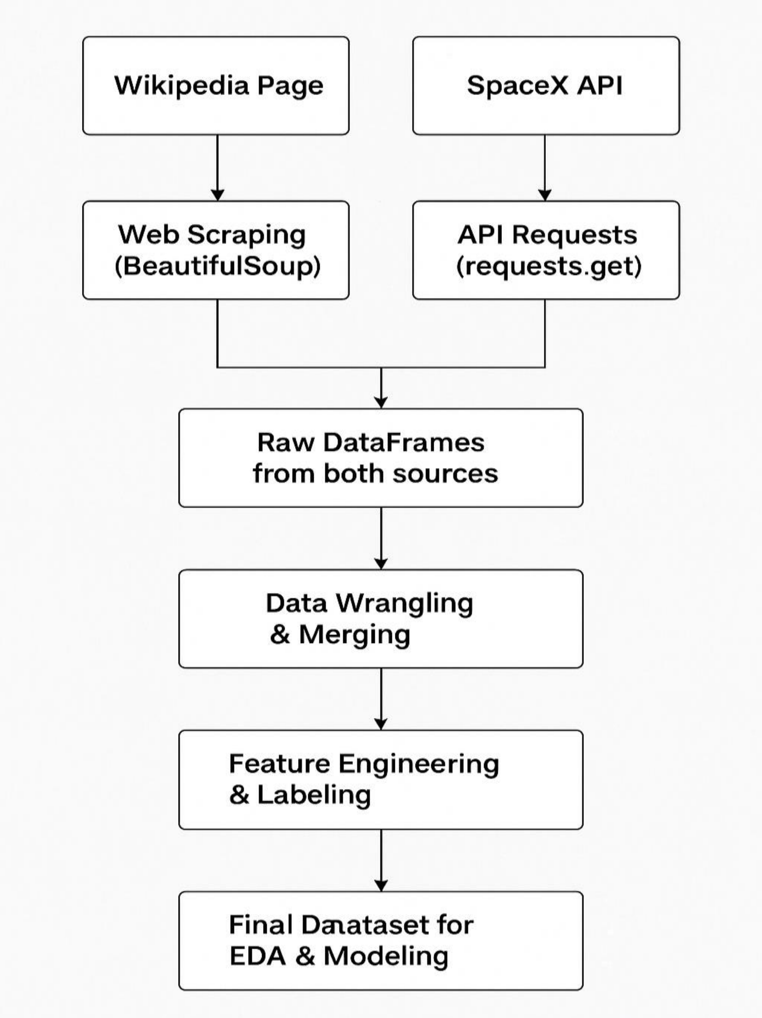
- Used requests.get() to retrieve structured launch data programmatically.
- Included rocket type, payload, mission outcome, and landing details.

2. Wikipedia Scraping

- Scraped the page “List of Falcon 9 and Falcon Heavy launches” using BeautifulSoup.
- Extracted supplementary information such as mission names, detailed outcomes, and specific date.

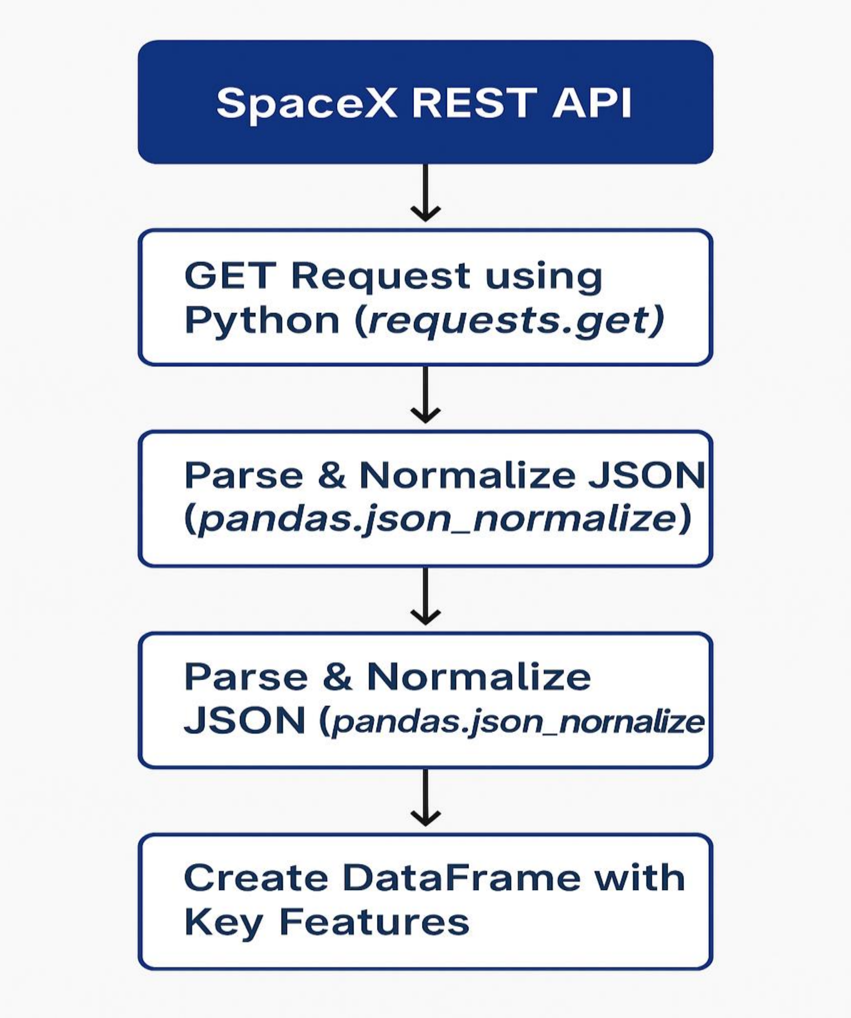
After data retrieval:

- Performed **data wrangling** to clean, merge, and format the data.
- Created **custom features** such as landing outcome labels and mission success indicators



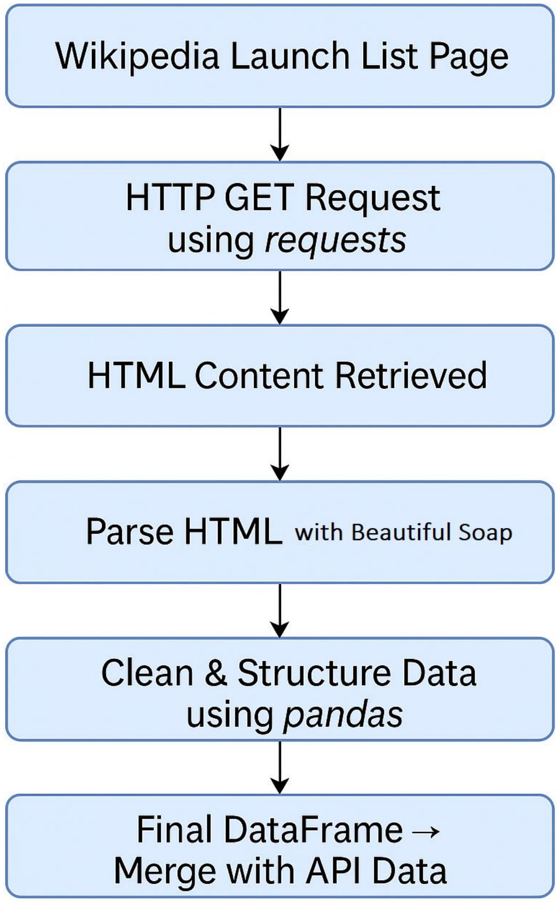
Data Collection – SpaceX API

- Accessed structured launch data via SpaceX's public REST API
- Utilized Python's requests library to retrieve JSON data
- Targeted key endpoints
- Parsed and normalized the data
- Extracted essential features: launch date, payload mass, orbit type, booster version, etc.
- Merged this data into a structured data frame for further analysis and model training
- GitHub url: [DataCollectionAPI]()



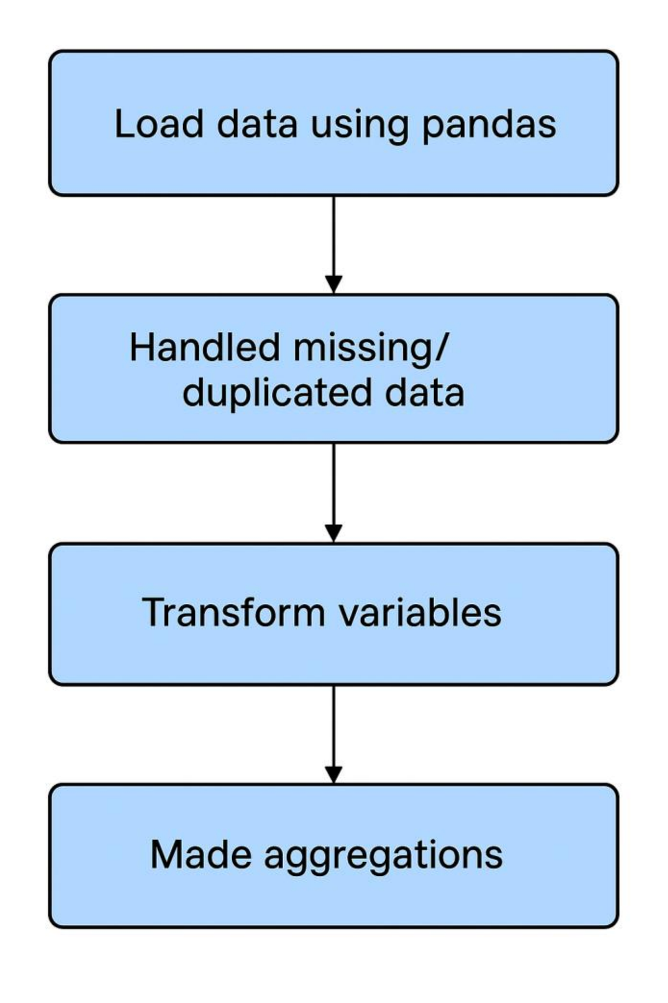
Data Collection - Scraping

- Used requests to fetch the HTML content of the page
- Parsed HTML using BeautifulSoup to navigate and extract relevant tables
- Cleaned and converted table data into structured format using pandas
- Extracted key details: launch date, launch site, outcome
- Combined with API data for a more comprehensive dataset
- GitHub Url: [WebScraping]



Data Wrangling

- Handled missing and duplicated data
- Transformed variables to suitable data types
- Calculated the number of launches on each site
- Calculated the number and occurrence of each orbit
- Created a landing outcome label from Outcome column
- Determined the number and occurrence of mission outcome of the orbits
- GitHub Url: [DataWrangling]



EDA with Data Visualization

- **Scatter Plots:** Plotted scatter charts to investigate relationships between variables such as Flight Number, Launch Site, Orbit Type, and Payload Mass.
- **Bar Charts:** Created bar charts to identify which orbit type had the highest success rate.
- **Line Graph:** Visualized the yearly trend of successful launches.
- GitHub URL: EDAwithVisualization

EDA with SQL

- **Aggregate Queries:** Used SQL aggregate functions like COUNT, AVG, MAX, and MIN to summarize data.
- **Grouping and Filtering:** Applied GROUP BY and WHERE clauses to segment data and focus on specific subsets.
- **Joins:** Combined data from multiple tables to enrich the analysis.
- **Ordering:** Sorted data to identify trends and outliers.
- GitHub URL: [EDAwithSQL]

Build an Interactive Map with Folium

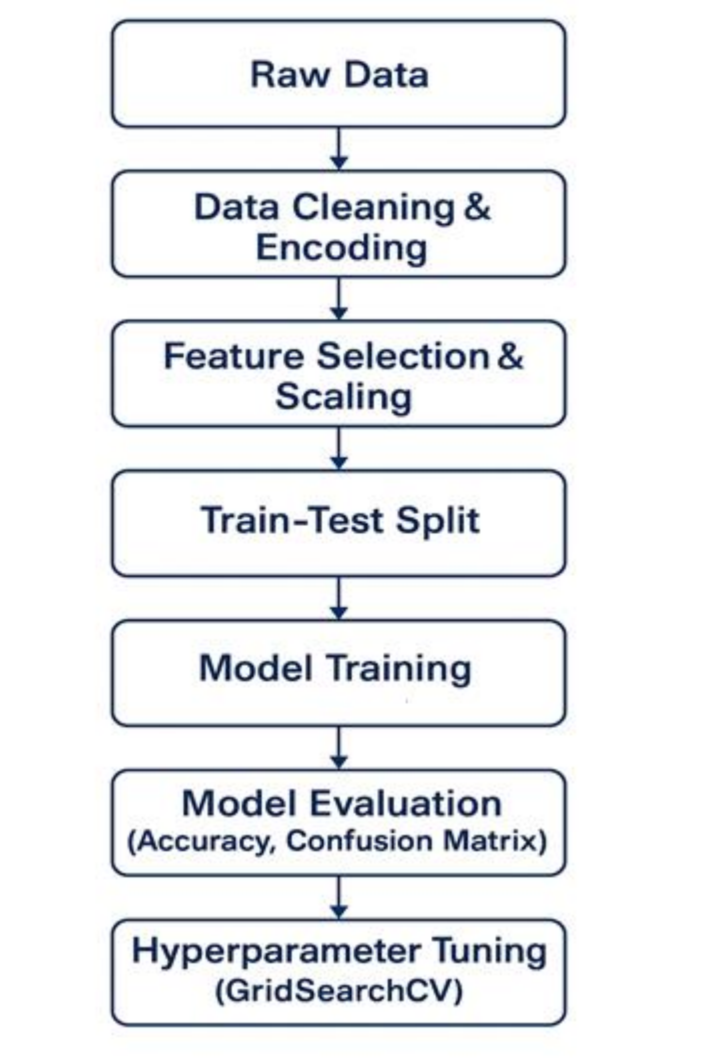
- **Markers Purpose:** Marked each launch site location.
  - **Why:** Helped visually identify the geographic location of SpaceX launch pads across the U.S.
- **Circle Markers Purpose:** Indicated launch activity intensity (or success/failure rate) at each site.
  - **Why:** The size and color variation helped convey quick insights into launch frequencies or success ratios per site.
- GitHub URL: FoliumMap

Build a Dashboard with Plotly Dash

- **Launch Site Success Rate (Pie Chart):** Visualizes the proportion of successful landings across different launch sites.
  - **Why:** Offers a quick, intuitive understanding of which sites have been most reliable.
- **Payload vs. Success Correlation (Scatter Plot):** Shows the relationship between payload mass and landing outcome.
  - **Why:** Helps assess whether heavier payloads influence the likelihood of successful recovery.
- GitHub URL: PlotlyCode
- GitHub URL: Dash

Predictive Analysis (Classification)

- **Data Preprocessing:** Encoded categorical variables (e.g., Orbit, Launch Site).
- **Normalization:** Normalized numerical features (e.g., Payload Mass).
- **Train-Test Split:** Divided dataset into training and testing set. Ensured class balance in split.
- **Model Selection:** Tried various classification algorithms.
- **Model Evaluation:** Evaluated models using accuracy score and confusion matrix.
- **Model Tuning:** Applied GridSearchCV for hyperparameter optimization.
- **Final Comparison:** All models showed similar accuracy (~83.3%) due to limited data.
- GitHub URL: [MachineLearning]()



Results

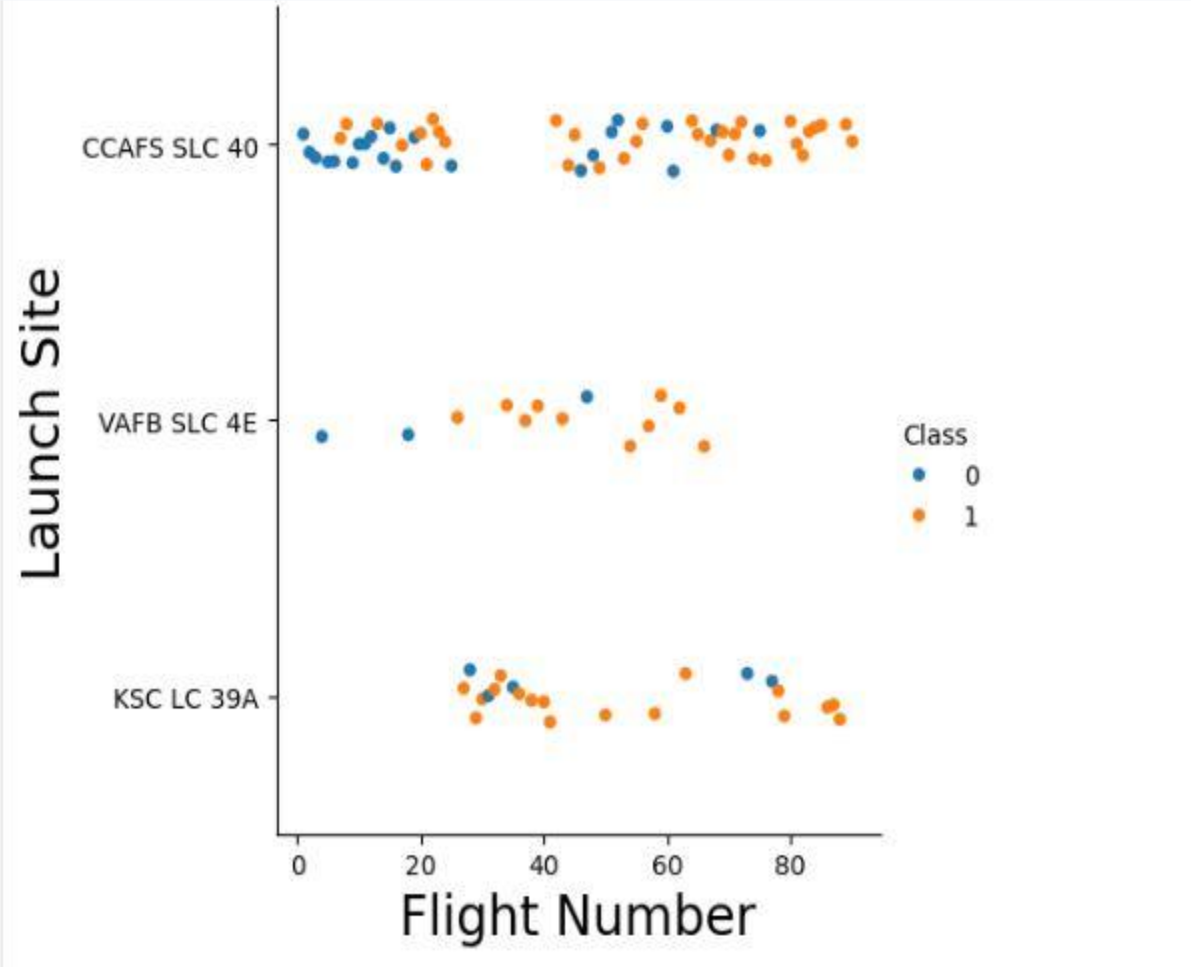
- In this project, we explored SpaceX launch data to understand and predict the success of first-stage rocket landings.
- Through exploratory data analysis, we found that missions to LEO and ISS had higher success rates, landing outcomes improved significantly over the years, and payload mass had no strong correlation with landing success.
- An interactive dashboard was developed with scatter plots and pie charts allowing users to explore trends by orbit, payload, and launch site using dropdown filters and tooltips.
- For predictive analysis, we applied various classification models—Logistic Regression, Decision Tree, SVM, and KNN—all of which achieved a similar accuracy of ~83.3%, likely due to the small-to-moderate dataset size. Despite hyper parameter tuning, model performance remained consistent.
- Overall, the project demonstrated how data analytics can uncover insights and support future launch decision-making.

Section 2

Insights drawn from EDA

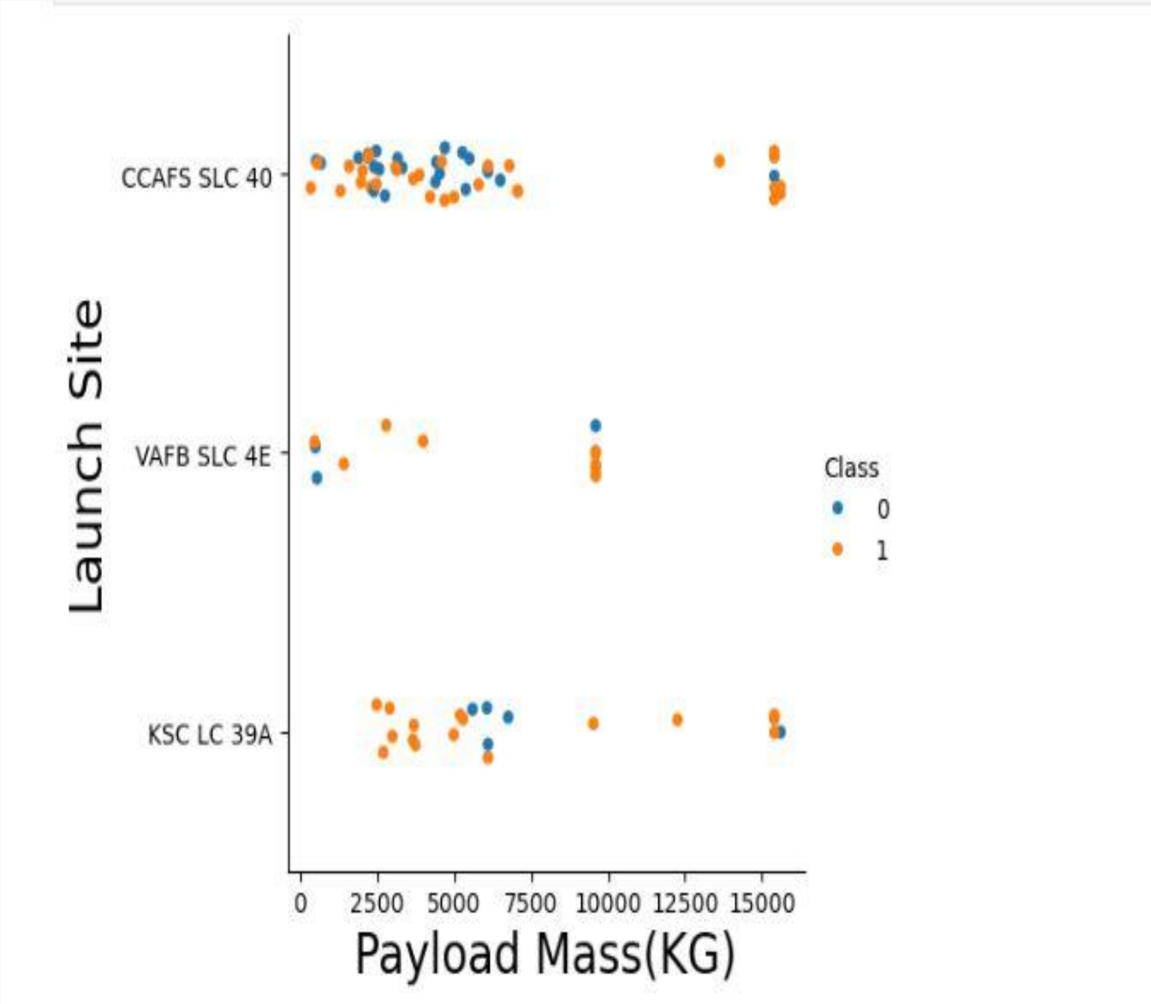
Flight Number vs. Launch Site

- **Trend Over Time:** The success rate improves as flight numbers increase, showing SpaceX learned and optimized from early failures.
- **Launch Site Impact:** Some launch sites (like KSC) seem to have better performance, but that may be because they hosted more modern missions after the technology matured.
- **Class 1 (successful landings)** dominate later launches compared to early ones.



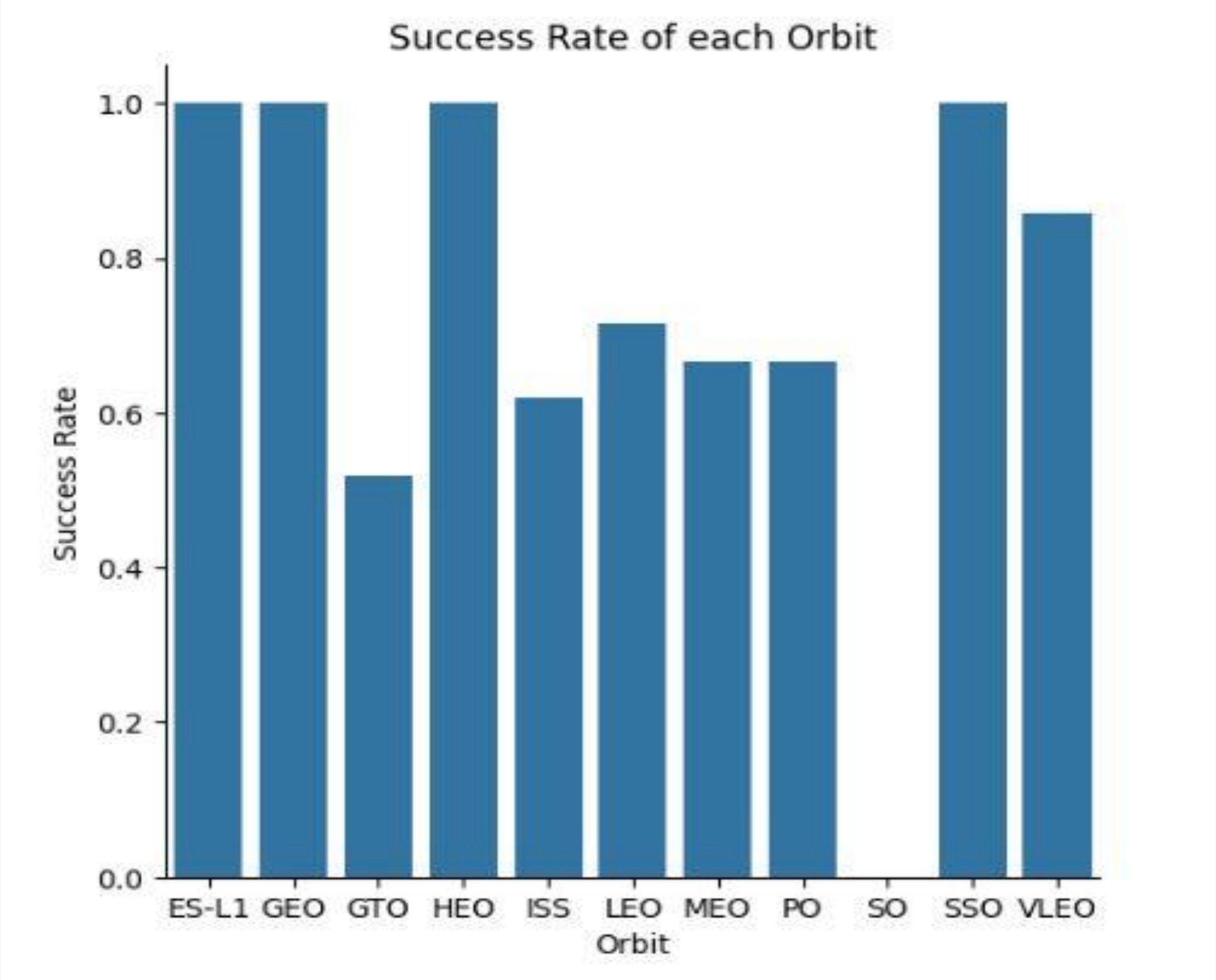
Payload vs. Launch Site

- **High payload mass does NOT drastically reduce success rates.**
  - Even with heavier payloads (up to 15,000 kg), SpaceX achieved several successful landings.
- **Launch site affects success probability slightly, but payload mass alone isn't the biggest factor**.
- **KSC LC 39A** shows particularly strong success rates for moderate payloads (~4,000–6,000 kg range).



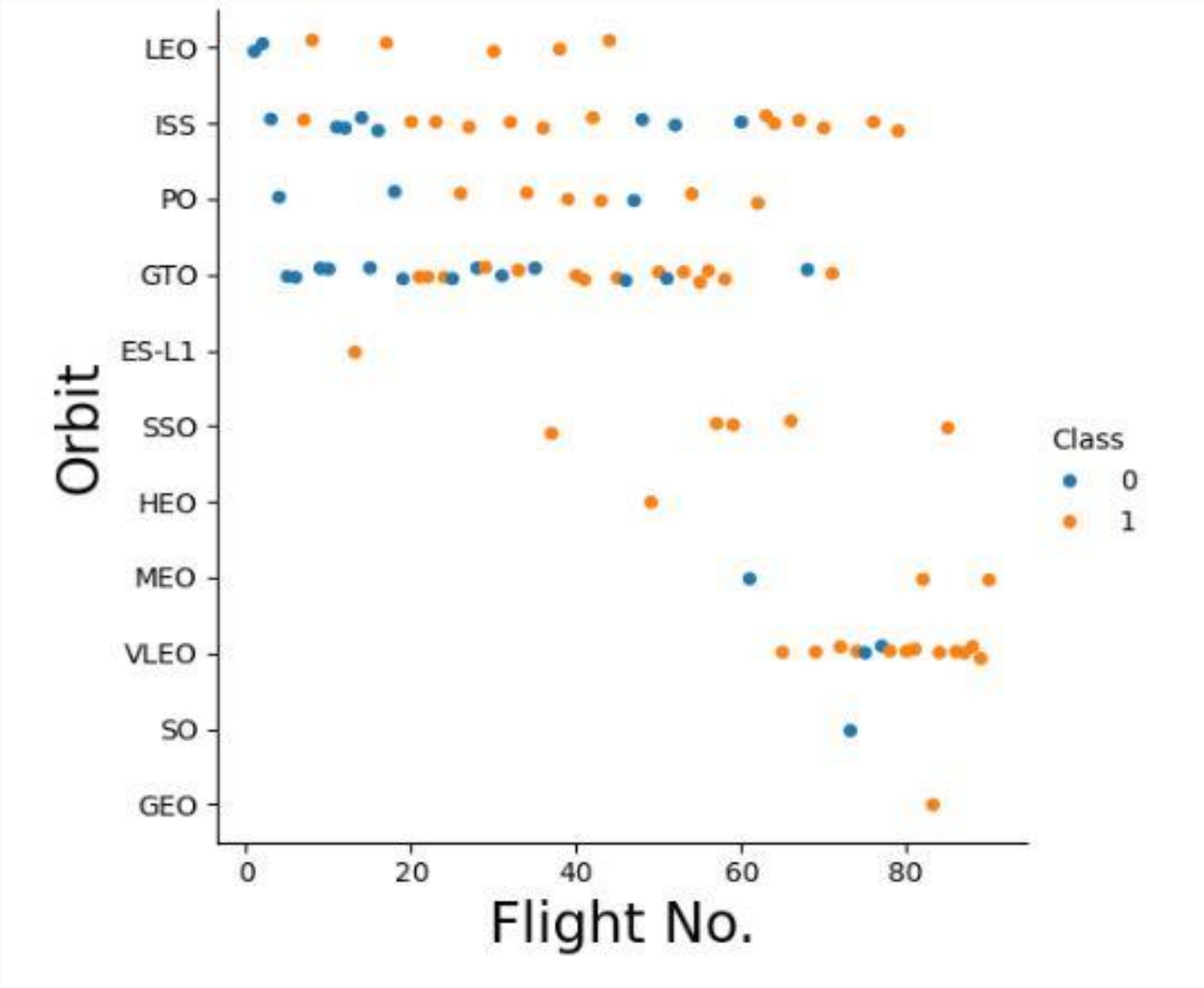
Success Rate vs. Orbit Type

- **Missions to higher orbits** (like GEO and ES-L1) still maintain a very high landing success, showing SpaceX's capability in handling complex missions while achieving first-stage recovery.
- **GTO missions are riskier for landing success** — likely because reaching geostationary orbit requires very high velocities, leaving less fuel for controlled landings.
- Missions to the **ISS and LEO** have moderate success, possibly due to their operational focus on cargo resupply and lower altitude but frequent launches.



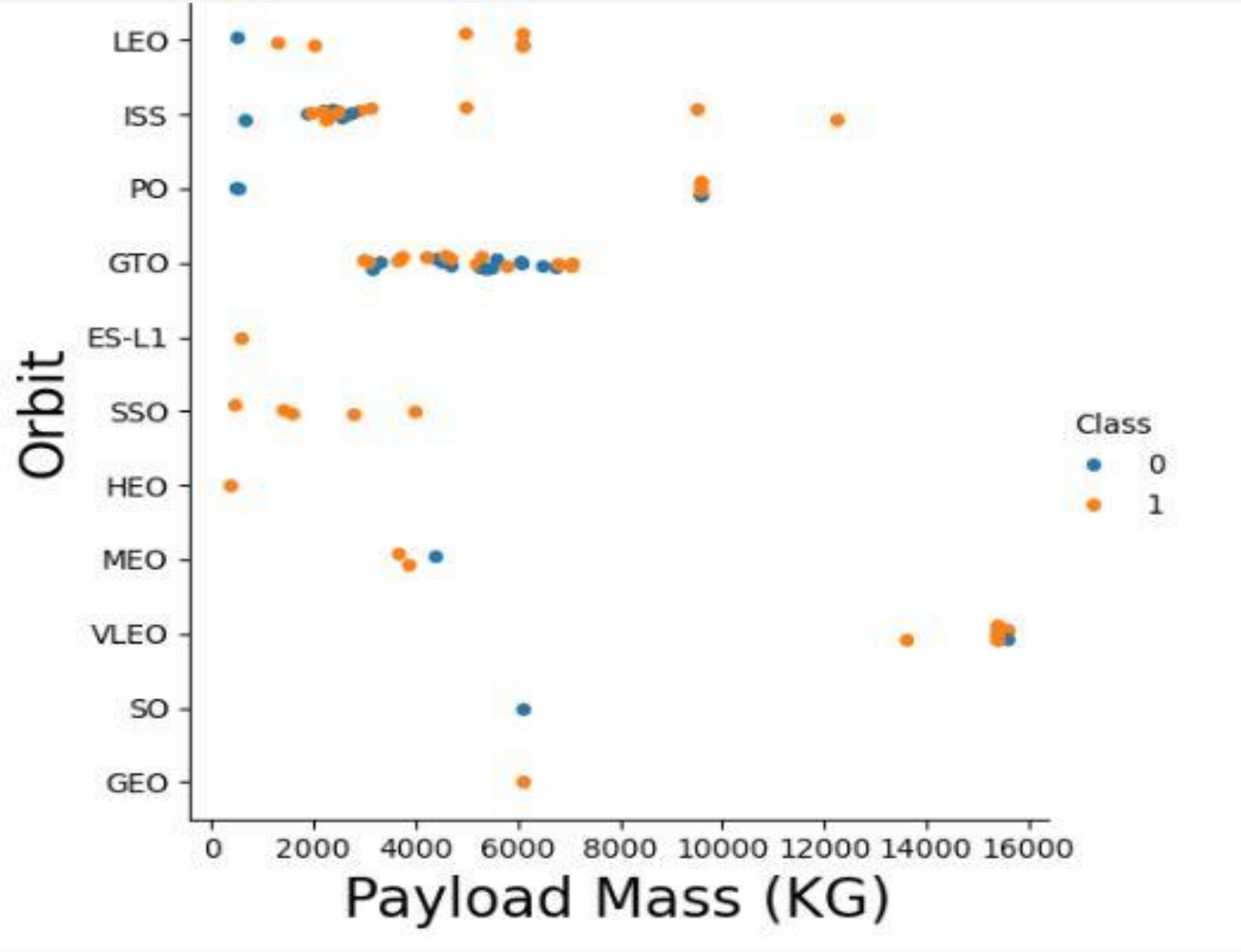
Flight Number vs. Orbit Type

- **Orbit-wise success rates vary:** LEO and ISS orbits have higher landing success rates compared to GTO and PO.
- **Higher orbit complexity often leads to failures:** GTO and HEO show more Class 0 outcomes, indicating tougher recovery conditions.
- **Success improved with experience:** Later flights show more successful landings, reflecting SpaceX's iterative improvements.
- **Rare orbits = sparse data:** For orbits like ES-L1, SO, and GEO, data is limited, so conclusions are less certain.



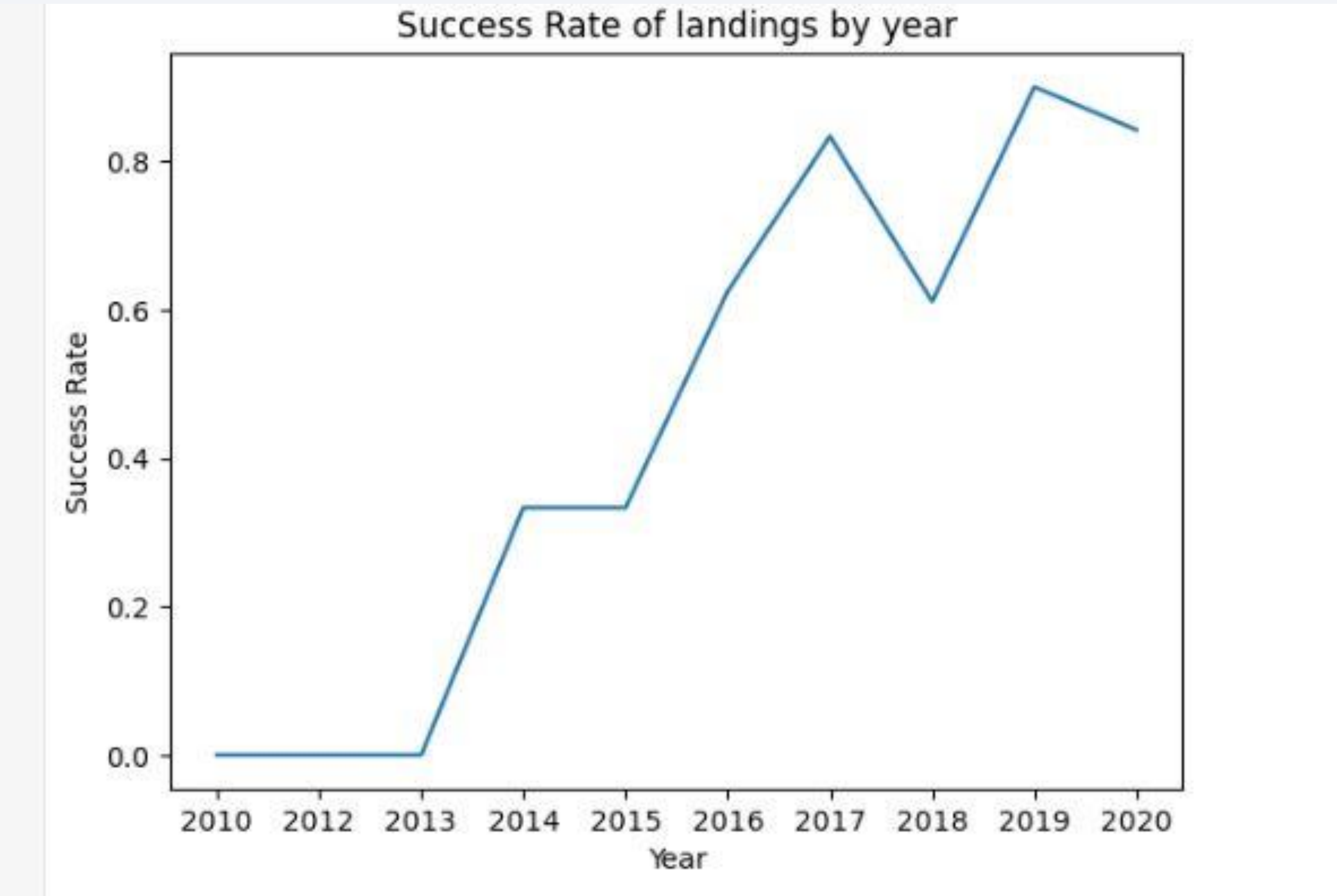
Payload vs. Orbit Type

- **No strong correlation** between payload mass and landing success.
- **GTO and VLEO orbits** support the **heaviest payloads** (up to ~16,000 KG), with both success and failure outcomes.
- **Lighter payloads (0–6,000 KG)** are distributed across most orbits and exhibit mixed outcomes.
- **SSO, PO, and ISS** tend to carry **smaller payloads** with relatively high landing success.



Launch Success Yearly Trend

- **No successful landings** occurred before 2014.
- **First major improvements** started in 2014, with a consistent upward trend through 2017.
- **Peak success rate (~90%)** occurred in **2019**, showing major advancements in technology and operations.
- **2018 saw a dip**, likely due to experimental missions or anomalies.
- Overall, a **clear improvement trend** is visible from 2014 to 2020.



All Launch Site Names

Task 1

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

[10]:
```
con = sqlite3.connect("my_data1.db")
cursor = con.cursor()
results=cursor.execute('SELECT DISTINCT Launch_Site FROM SPACEXTABLE;')
rows=results.fetchall()
print("The unique launch sites are:")
for r in rows:
    print(r[0])
cursor.close()
con.close()
```

```
The unique launch sites are:
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'

Task 2

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

[11]:
```
con = sqlite3.connect("my_data1.db")
cursor = con.cursor()
results = cursor.execute("SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE '%CCA%' LIMIT 5;")
rows = results.fetchall()
print("The launch sites that match 'CCA' are:")
for r in rows:
    print(r)
cursor.close()
con.close()
```

```
The launch sites that match 'CCA' are:
('2010-06-04', '18:45:00', 'F9 v1.0  B0003', 'CCAFS LC-40', 'Dragon Spacecraft Qualification Unit', 0, 'LEO', 'SpaceX', 'Success', 'Failur
e (parachute)')
('2010-12-08', '15:43:00', 'F9 v1.0  B0004', 'CCAFS LC-40', 'Dragon demo flight C1, two CubeSats, barrel of Brouere cheese', 0, 'LEO (IS
S)', 'NASA (COTS) NRO', 'Success', 'Failure (parachute)')
('2012-05-22', '7:44:00', 'F9 v1.0  B0005', 'CCAFS LC-40', 'Dragon demo flight C2', 525, 'LEO (ISS)', 'NASA (COTS)', 'Success', 'No attemp
t')
('2012-10-08', '0:35:00', 'F9 v1.0  B0006', 'CCAFS LC-40', 'SpaceX CRS-1', 500, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'No attempt')
('2013-03-01', '15:10:00', 'F9 v1.0  B0007', 'CCAFS LC-40', 'SpaceX CRS-2', 677, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'No attempt')
```

Total Payload Mass

Task 3

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

[12]:
```
con=sqlite3.connect("my_data1.db")
cursor=con.cursor()
results=cursor.execute("SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE CUSTOMER='NASA (CRS)';")
rows=results.fetchone()
print(f"Payload mass in KG of NASA CRS is:{rows[0]}")
cursor.close()
con.close()
```

```
Payload mass in KG of NASA CRS is:45596
```

Average Payload Mass by F9 v1.1

Task 4

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

```
[13]: con=sqlite3.connect("my_data1.db")
cursor=con.cursor()
results=cursor.execute("SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version='F9 v1.1';")
rows=results.fetchone()
print(f"Average Payload Mass KG of Booster F9 v1.1 is:{rows[0]}")
cursor.close()
con.close()
```

```
Average Payload Mass KG of Booster F9 v1.1 is:2928.4
```

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

[14]:
```
con = sqlite3.connect("my_data1.db")
cursor = con.cursor()
results = cursor.execute("""
    SELECT MIN(DATE), Landing_Outcome
    FROM SPACEXTABLE
    WHERE Landing_Outcome='Success (ground pad)';
""")
rows = results.fetchone()
print(f"The date when the first successful landing outcome is achieved: {rows[0]}")
cursor.close()
con.close()
```

```
The date when the first successful landing outcome is achieved: 2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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

[15]:

```
con=sqlite3.connect("my_data1.db")
cursor=con.cursor()
results=cursor.execute("""SELECT Booster_Version, Landing_Outcome, PAYLOAD_MASS__KG_ FROM SPACEXTABLE
WHERE Landing_Outcome='Success (drone ship)' AND PAYLOAD_MASS__KG_ >4000 AND PAYLOAD_MASS__KG_<6000;""")
rows=results.fetchall()
for r in rows:
    print(r)
cursor.close()
con.close()
```

```
('F9 FT B1022', 'Success (drone ship)', 4696)
('F9 FT B1026', 'Success (drone ship)', 4600)
('F9 FT  B1021.2', 'Success (drone ship)', 5300)
('F9 FT  B1031.2', 'Success (drone ship)', 5200)
```

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

[24]:
```
con=sqlite3.connect("my_data1.db")
cursor=con.cursor()
results=cursor.execute("""SELECT Mission_Outcome, Count(*) FROM SPACEXTABLE GROUP BY Mission_Outcome;""")
rows=results.fetchall()
print("Mission Outcome Counts:")
for row in rows:
    print(f"Outcome: {row[0]}, Count: {row[1]}")
cursor.close()
con.close()
```

```
Mission Outcome Counts:
Outcome: Failure (in flight), Count: 1
Outcome: Success, Count: 98
Outcome: Success , Count: 1
Outcome: Success (payload status unclear), Count: 1
```

Boosters Carried Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

[33]:
```
con=sqlite3.connect("my_data1.db")
cursor=con.cursor()
results=cursor.execute("""SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
rows=results.fetchall()
for row in rows:
    print(f"Booster Versions:{row[0]}")
cursor.close()
con.close()
```

```
Booster Versions:F9 B5 B1048.4
Booster Versions:F9 B5 B1049.4
Booster Versions:F9 B5 B1051.3
Booster Versions:F9 B5 B1056.4
Booster Versions:F9 B5 B1048.5
Booster Versions:F9 B5 B1051.4
Booster Versions:F9 B5 B1049.5
Booster Versions:F9 B5 B1060.2
Booster Versions:F9 B5 B1058.3
Booster Versions:F9 B5 B1051.6
Booster Versions:F9 B5 B1060.3
Booster Versions:F9 B5 B1049.7
```

2015 Launch Records

Task 9

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.

```
[ ]: import sqlite3

# Connect to the SQLite database
con = sqlite3.connect("my_data1.db")
cursor = con.cursor()

# Query to get the required records for 2015 with failure landing on drone ship
results = cursor.execute("""
    SELECT
        CASE
            WHEN substr(Date, 6, 2) = '01' THEN 'January'
            WHEN substr(Date, 6, 2) = '02' THEN 'February'
            WHEN substr(Date, 6, 2) = '03' THEN 'March'
            WHEN substr(Date, 6, 2) = '04' THEN 'April'
            WHEN substr(Date, 6, 2) = '05' THEN 'May'
            WHEN substr(Date, 6, 2) = '06' THEN 'June'
            WHEN substr(Date, 6, 2) = '07' THEN 'July'
            WHEN substr(Date, 6, 2) = '08' THEN 'August'
            WHEN substr(Date, 6, 2) = '09' THEN 'September'
            WHEN substr(Date, 6, 2) = '10' THEN 'October'
            WHEN substr(Date, 6, 2) = '11' THEN 'November'
            WHEN substr(Date, 6, 2) = '12' THEN 'December'
        END AS Month_Name,
        Booster_Version,
        Landing_Outcome,
        Launch_Site
    FROM SPACEXTABLE
    WHERE Landing_Outcome = 'Failure (drone ship)'
      AND substr(Date, 0, 5) = '2015';
""")

# Fetch all results
rows = results.fetchall()

# Print the results
print("Failure Landing Outcomes in Drone Ship for 2015 (with Month Names):")
for row in rows:
    print(f"Month: {row[0]}, Booster Version: {row[1]}, Landing Outcome: {row[2]}, Launch Site: {row[3]}")

# Close the cursor and connection
cursor.close()
con.close()
```

```
Failure Landing Outcomes in Drone Ship for 2015 (with Month Names):
Month: January, Booster Version: F9 v1.1 B1012, Landing Outcome: Failure (drone ship), Launch Site: CCAFS LC-40
Month: April, Booster Version: F9 v1.1 B1015, Landing Outcome: Failure (drone ship), Launch Site: CCAFS LC-40
```

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

Task 10

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.

[37]:

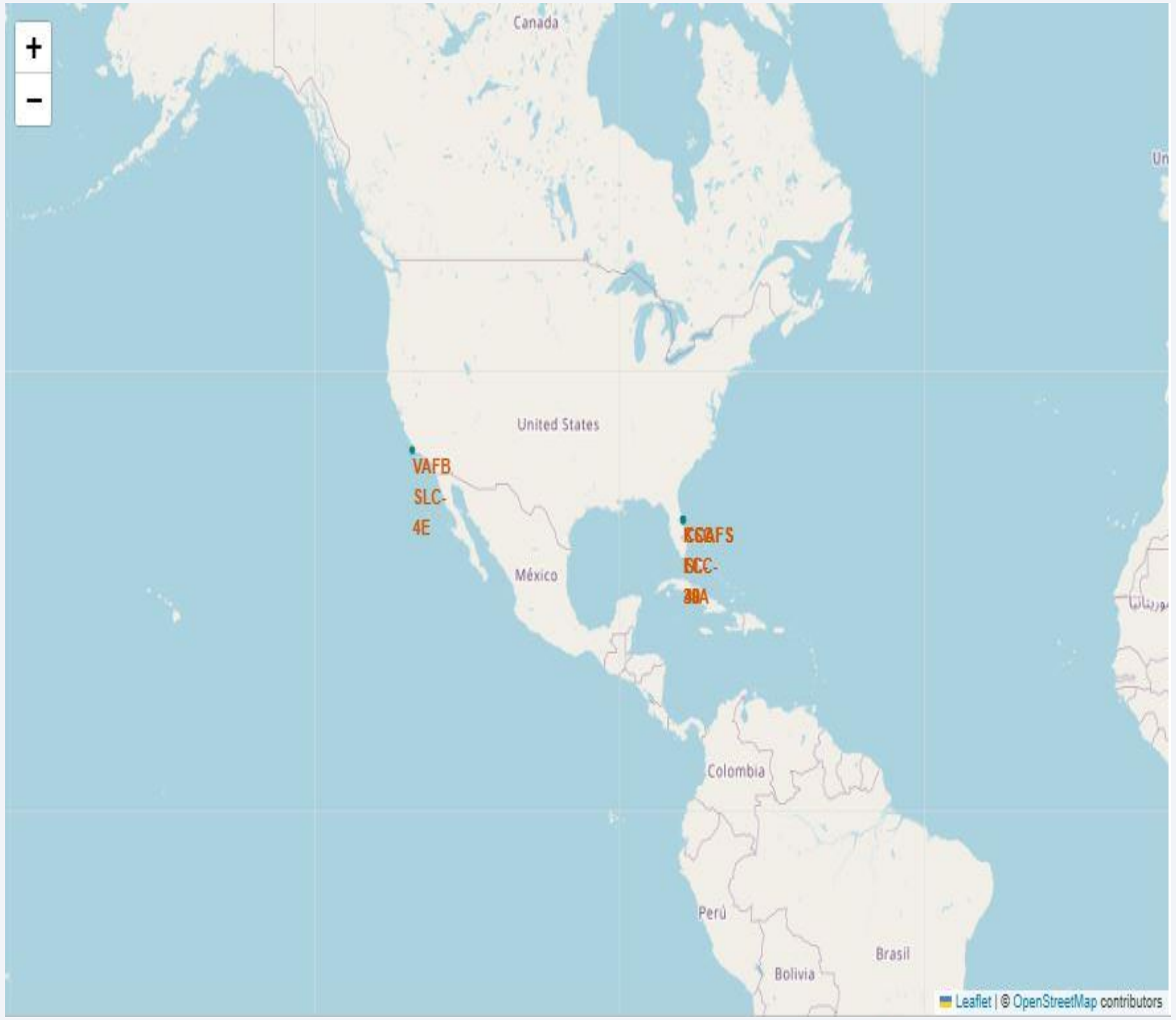
```
con = sqlite3.connect("my_data1.db")
cursor = con.cursor()
results = cursor.execute("""
    SELECT Landing_Outcome, COUNT(*)
    FROM SPACEXTABLE
    WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY Landing_Outcome
    ORDER BY COUNT(*) DESC;
""")
rows = results.fetchall()
print("Landing Outcome Counts (from 2010-06-04 to 2017-03-20):")
for row in rows:
    print(f"Outcome: {row[0]}, Count: {row[1]}")
cursor.close()
con.close()
```

```
Landing Outcome Counts (from 2010-06-04 to 2017-03-20):
Outcome: No attempt, Count: 10
Outcome: Success (drone ship), Count: 5
Outcome: Failure (drone ship), Count: 5
Outcome: Success (ground pad), Count: 3
Outcome: Controlled (ocean), Count: 3
Outcome: Uncontrolled (ocean), Count: 2
Outcome: Failure (parachute), Count: 2
Outcome: Precluded (drone ship), Count: 1
```

Section 3

Launch Sites Proximities Analysis

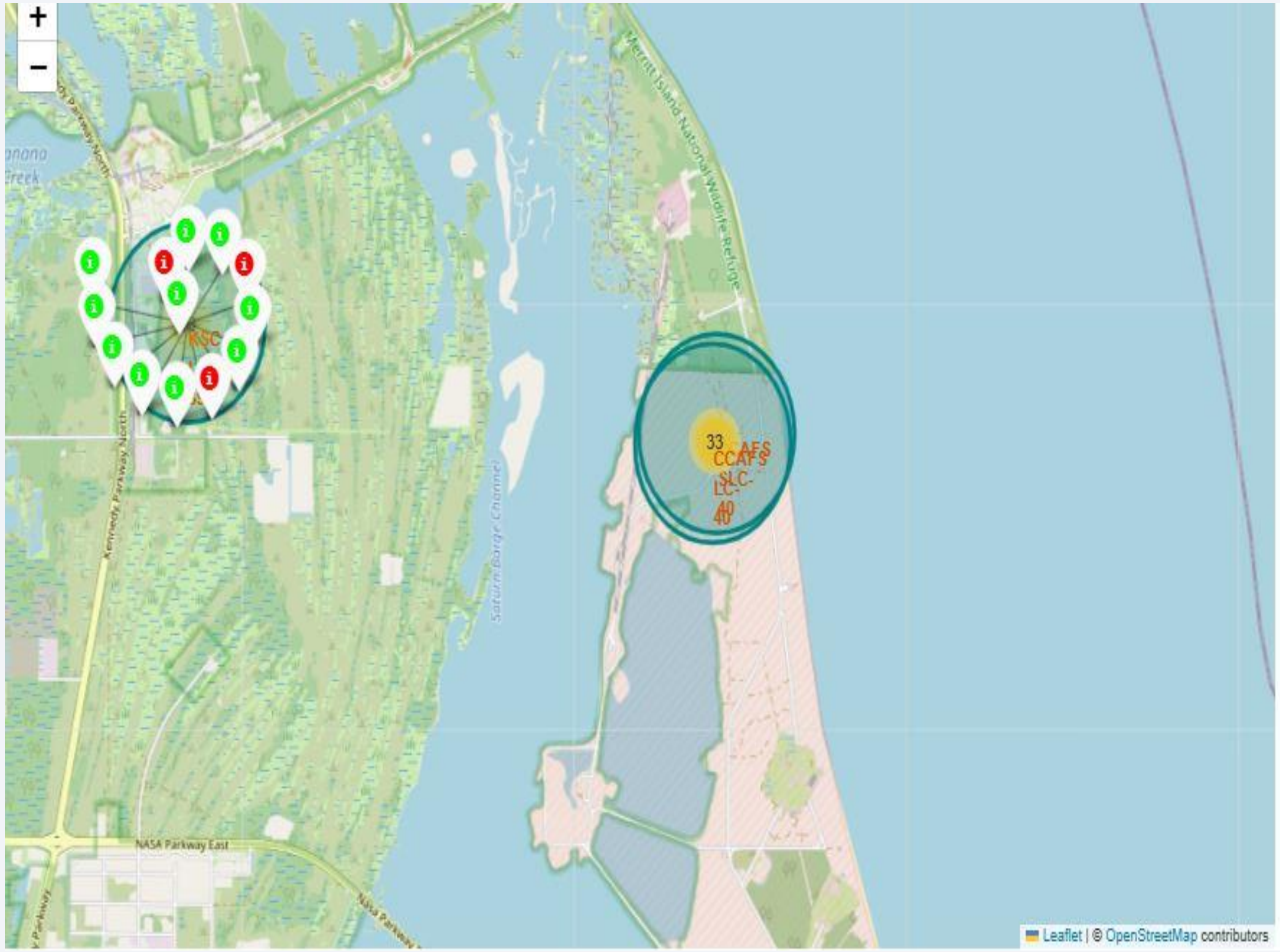
Launch Sites



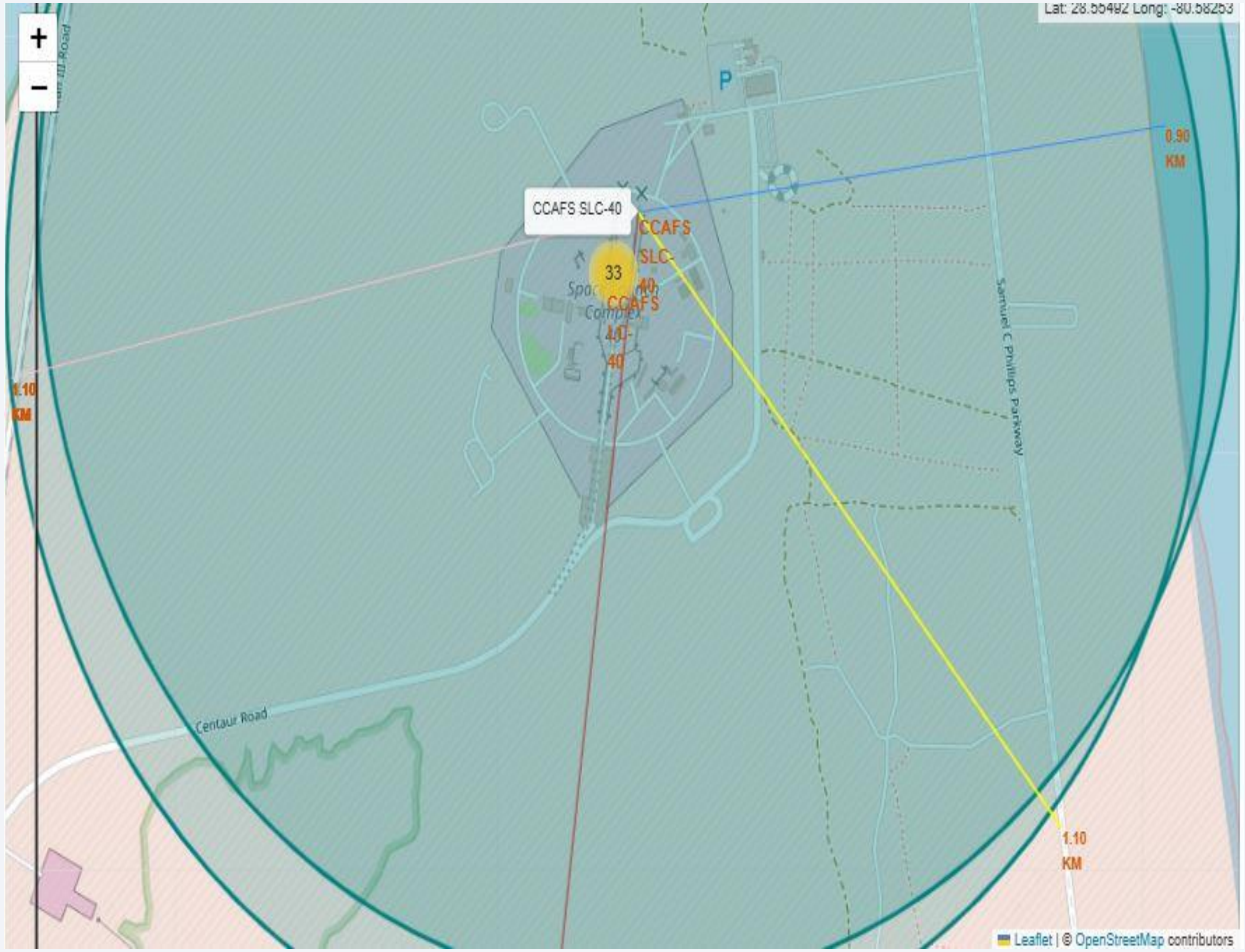
- All four launch sites — CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, and VAFB SLC-4E — were plotted on the Folium map using circles and markers.
- CCAFS LC-40, CCAFS SLC-40, and KSC LC-39A are all located close to Cape Canaveral, Florida.
- VAFB SLC-4E is the only launch site located on the West Coast, at Vandenberg Air Force Base in California.
- This geographical separation helps SpaceX serve different mission profiles: East Coast (Cape Canaveral): Ideal for missions to equatorial orbits (like GTO). West Coast (Vandenberg AFB): Preferred for polar or sun-synchronous orbits (SSO).

Launch Outcomes

- On the Folium map, each launch was represented by an icon inside a MarkerCluster object to group launches visually by proximity.
- Color Coding:Green icons → Represent successful landings (Class 1).
- Red icons → Represent unsuccessful landings (Class 0).
- Clustering allowed the map to handle overlapping launches neatly, making it easier to zoom in and explore individual launches.



Launch Site Proximities

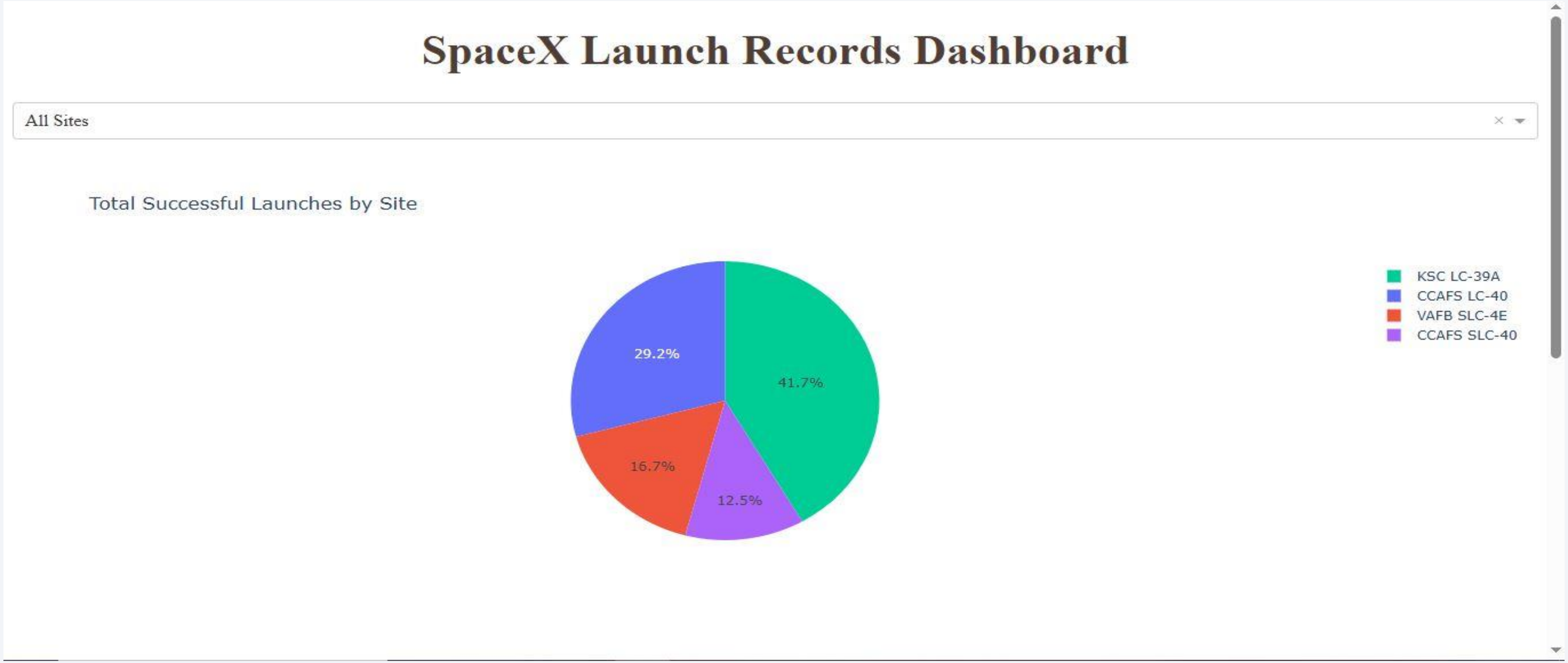


- The distance from CCAFS SLC-40 to various nearby features (coastline, railway, highway, city) was calculated to assess proximity risks.
- Findings:
  - The nearest feature to the launch site was the ocean, offering a natural buffer zone.
  - Railways and highways were located at moderate distances.
  - The nearest city was the farthest among the features, ensuring public safety in the event of a launch failure.
- Reasoning:
  Keeping populated areas at a greater distance minimizes risk to civilians and reduces potential launch-related hazards.

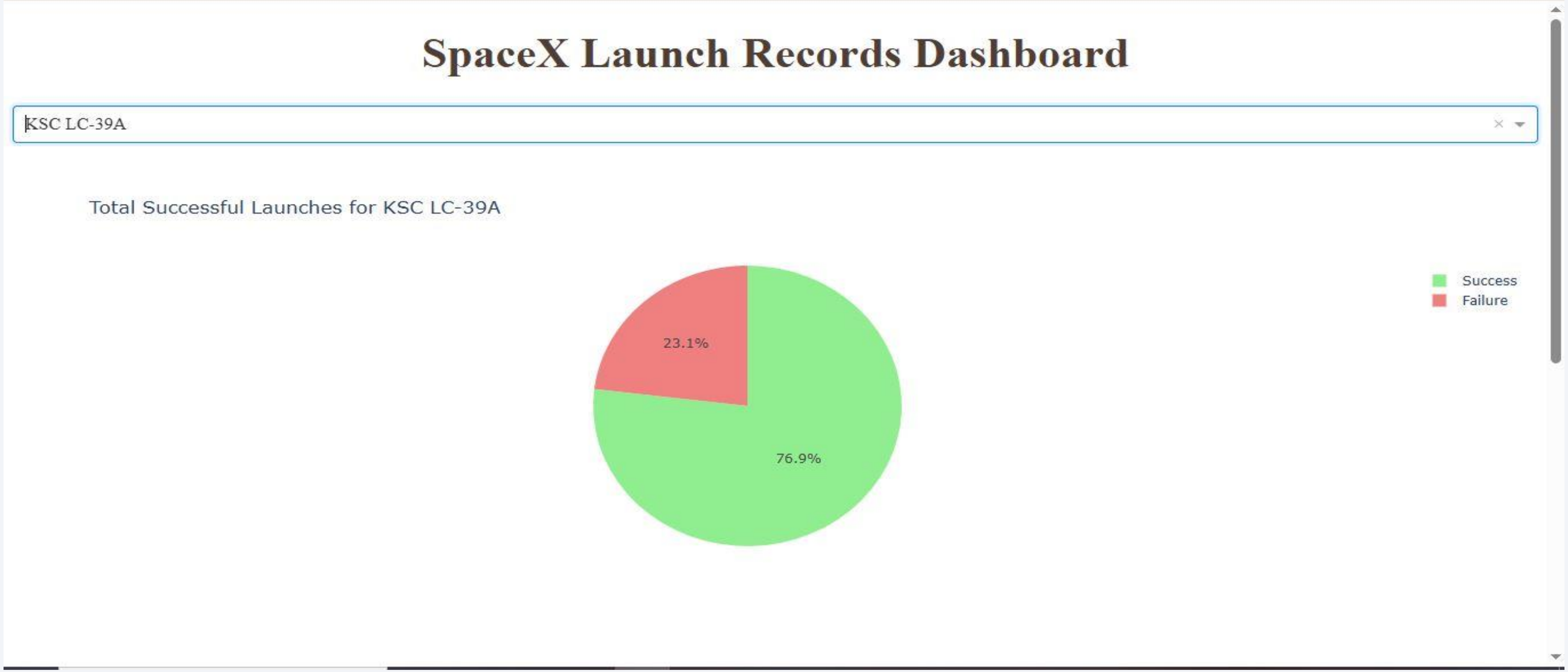
Section 4

Build a Dashboard with Plotly Dash

Total Successful Launches by Site



Site with most successful launch rate



Launch Site Rate Analysis

- KSC LC 39A showed the highest success rate among all sites: Overall success rate: 41.7% of total launches. Further breakdown (Pie Chart Analysis):76.9% of launches were successful.23.1% of launches were unsuccessful.
- CCAFS SLC 40 had the lowest success rate: Only 12.5% of launches from this site were successful.
- **Key Insights:**
  - KSC LC 39A is SpaceX’s most reliable site based on success proportions, likely due to newer booster versions, improved technology, and operational experience.
  - CCAFS SLC 40, despite being a historic site, had more early-stage launches when SpaceX was still perfecting the landing process.

Correlation b/w Payload & Launch Success

Correlation b/w Payload & Launch Success

Overall Analysis Across All Sites:

- Payload mass (kg) did not show a strong influence on landing success.
- Launch success rates appeared relatively consistent across different payload weights.

Booster Version Insight:

- The F9 v1.1 booster category had the most unsuccessful landings, regardless of payload size.
- This suggests that technology limitations in early booster versions contributed more to landing failures than payload mass itself.

Site-Specific Analysis – KSC LC 39A:

- Lighter payloads were more likely to have successful landings.
- Heavier payloads — especially on FT (Full Thrust) booster versions — tended to fail more often.

Key Insights:

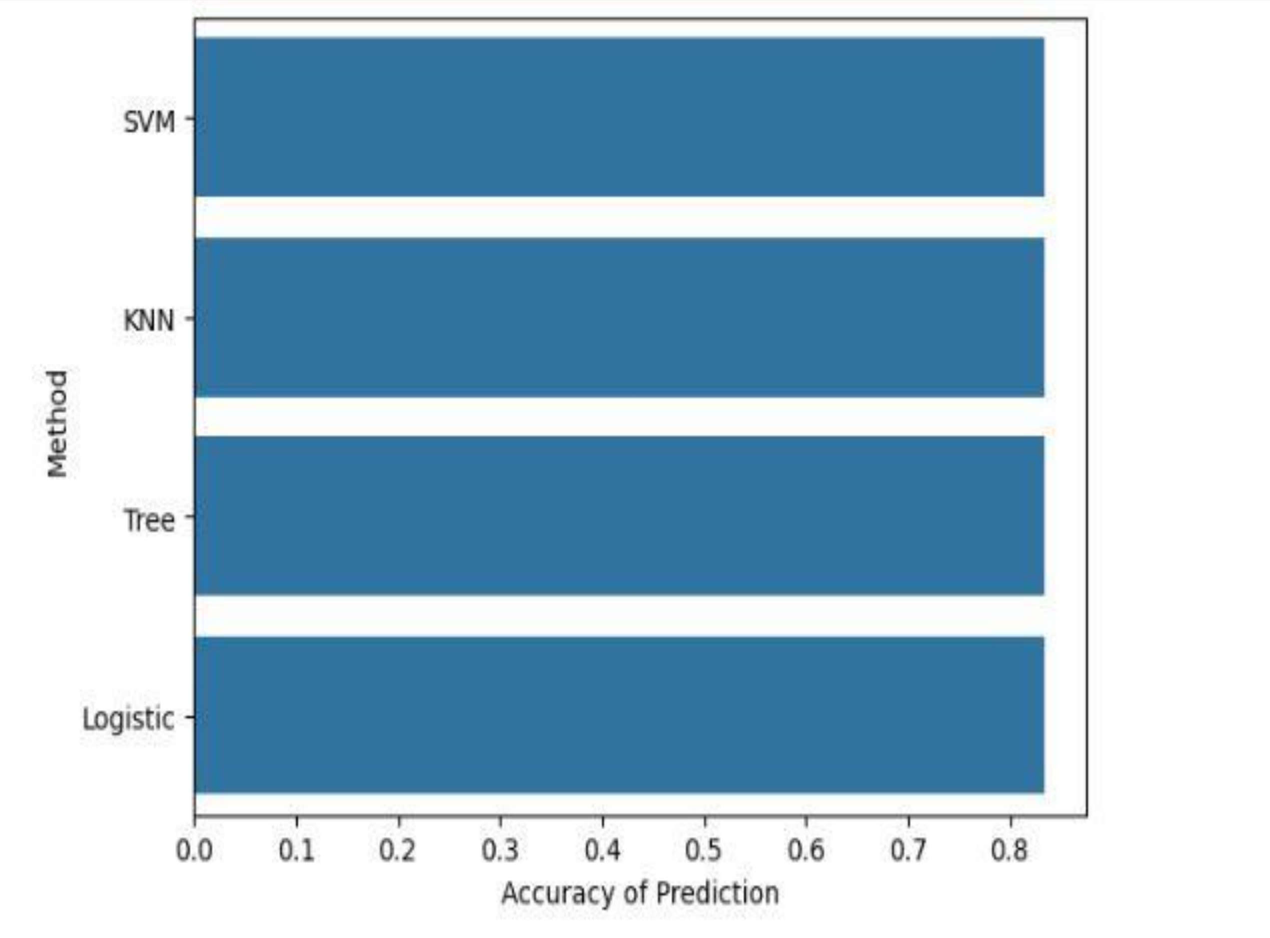
- Booster technology upgrades (moving from v1.1 to later versions like FT and Block 5) were crucial in improving landing success.
- At KSC LC 39A, a pattern emerged where lighter missions fared better, but heavier missions posed challenges even with newer boosters.

Section 5

Predictive Analysis (Classification)

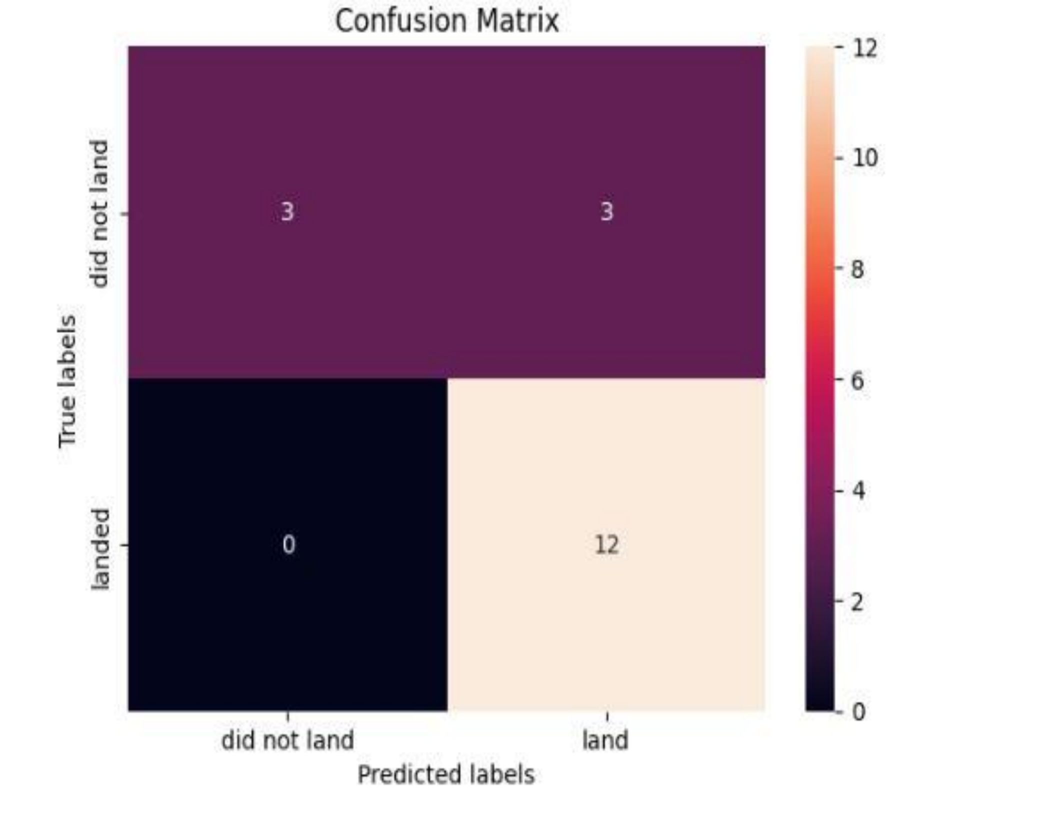
Classification Accuracy

- **All models achieved similar accuracy (~83.3%).**
- **The small to moderate dataset size** limited the ability of algorithms to distinguish themselves in performance.



Confusion Matrix

- The confusion matrix results for all algorithms combined show the following:
  - **True Positives (TP): 12**
    → 12 successful landings correctly predicted as successful.
  - **True Negatives (TN): 3**
    → 3 failed landings correctly predicted as failures.
  - **False Positives (FP): 3**
    → 3 cases where the model incorrectly predicted a success when it was actually a failure.
  - **False Negatives (FN): 0**
    → No cases where the model missed a successful landing (no successes incorrectly classified as failures).



Confusion Matrix

- **Key Insights:**
  - **High True Positive Rate:** The model is **very good at predicting successful landings**.
  - **No False Negatives:** The model **never missed** an actual success, which is important for business decisions (e.g., launch costing, insurance).
  - **Some False Positives:** There is a small risk where a failure might be misclassified as a success, but it is relatively limited.

Conclusions

- We performed extensive data wrangling to explore and clean launch data.
- Visualizations revealed key insights into orbit types, payloads, and landing outcomes.
- A clear trend of increasing landing success over the years was observed.
- We developed several machine learning models—Logistic Regression, Decision Tree, SVM, and KNN—all achieving ~83% accuracy.
- Hyperparameter tuning did not significantly improve performance due to the moderate dataset size.
- A Folium map was used to visualize launch sites and outcomes, enhancing spatial understanding.

Appendix

Tools & Technologies Used:

- **Languages/Libraries:** Python (pandas, matplotlib, seaborn, scikit-learn)
- **APIs & Web:** requests, BeautifulSoup (for scraping Wikipedia and API data)
- **Dashboards:** Dash, Plotly (for interactive visualizations)
- **SQL:** Used Jupyter Notebook with SQL magic for querying structured datasets
- **ChatGPT:** Used AI tool to assist in coding related queries and presentation refinement

Appendix

Dataset Overview:

1		Flight Number	Launch Site	class	Payload Mass (kg)	Booster Version	Booster Version Category
2	0	1	CCAFS LC-40	0	0	F9 v1.0 B0003	v1.0
3	1	2	CCAFS LC-40	0	0	F9 v1.0 B0004	v1.0
4	2	3	CCAFS LC-40	0	525	F9 v1.0 B0005	v1.0
5	3	4	CCAFS LC-40	0	500	F9 v1.0 B0006	v1.0
6	4	5	CCAFS LC-40	0	677	F9 v1.0 B0007	v1.0
7	5	7	CCAFS LC-40	0	3170	F9 v1.1	v1.1
8	6	8	CCAFS LC-40	0	3325	F9 v1.1	v1.1
9	7	9	CCAFS LC-40	0	2296	F9 v1.1	v1.1
10	8	10	CCAFS LC-40	0	1316	F9 v1.1	v1.1
11	9	11	CCAFS LC-40	0	4535	F9 v1.1	v1.1
12	10	12	CCAFS LC-40	0	4428	F9 v1.1 B1011	v1.1
13	11	13	CCAFS LC-40	0	2216	F9 v1.1 B1010	v1.1
14	12	14	CCAFS LC-40	0	2395	F9 v1.1 B1012	v1.1
15	13	15	CCAFS LC-40	0	570	F9 v1.1 B1013	v1.1
16	14	16	CCAFS LC-40	0	4159	F9 v1.1 B1014	v1.1

Thank you!