IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<CK>
<19/02/2025>

Outline

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

Executive Summary

Summary of Methodologies:

- We collected and processed SpaceX launch data, including information on payload mass, launch sites, booster versions, and orbits.
- We performed **exploratory data analysis (EDA)** to identify key trends and visualize relationships between launch parameters and success rates.
- We built **predictive machine learning models**, including **Logistic Regression, Support Vector Machine (SVM), Decision Trees, and K-Nearest Neighbors (KNN)**, to determine the likelihood of first-stage reuse.
- We optimized model performance using **GridSearchCV** for hyperparameter tuning.
- We developed **interactive dashboards** using **Plotly Dash** to present key insights dynamically.

Summary of Results:

- **Model Performance:** Logistic Regression and SVM achieved the highest accuracy in predicting first-stage landings.
- **Key Findings:**
    - Launch site and orbit type are strong predictors of landing success.
    - Heavier payloads tend to reduce the likelihood of successful landings.
    - Reusability significantly impacts launch costs, with successful recoveries leading to lower mission costs.
- **Impact:** These findings help 'SpaceY' optimize its rocket design and mission planning to reduce costs and compete with SpaceX.

Introduction

Why is this important?
- The ability to predict first-stage reuse can significantly impact launch costs.
- This project will use machine learning to analyze launch data and predict successful landings.

Key Questions:
1. What factors influence a successful landing?
2. Can we predict first-stage reuse based on public data?
3. How does reusability affect launch pricing?

Approach:
- Collect SpaceX launch data.
- Analyze trends in launch outcomes.
- Train machine learning models to predict reusability.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

Data Sources:

- SpaceX Launch Data was obtained via SpaceX REST API and Web Scraping.
- Supplementary data was collected from public repositories and NASA datasets.

APIs and Web Scraping:

- Used SpaceX API to extract raw launch records, booster versions, and launch sites.
- Implemented BeautifulSoup & Requests for web scraping of Wikipedia pages to retrieve historical launch data.

SQL Integration:

- Stored cleaned data in a SQL database for structured queries and analysis.
- Used SQL queries to extract insights on launch success rates, payload impact, and mission outcomes.

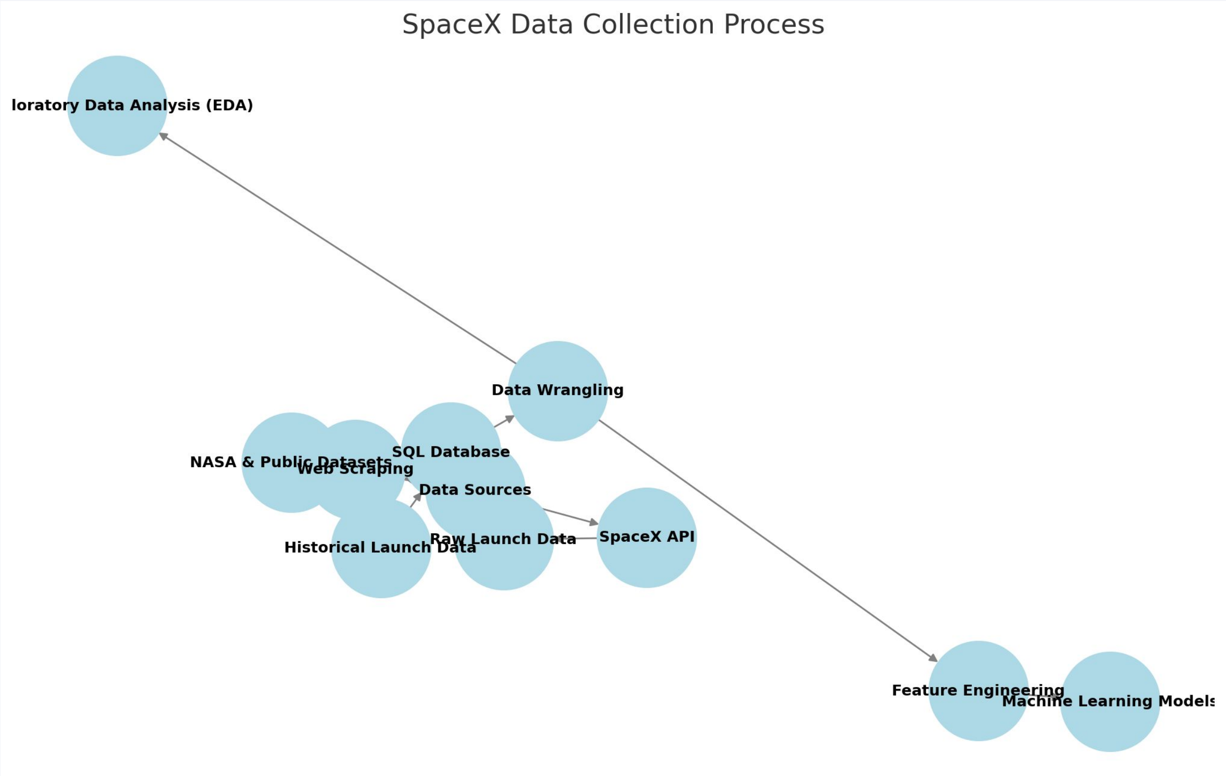
Data Wrangling & Preprocessing:

- Cleaned missing values, removed duplicate records, and standardized column formats.
- Converted categorical data (e.g., Launch Sites, Booster Versions) into one-hot encoding for machine learning.

Data Collection – SpaceX API

1. Extracted from SpaceX API, Web Scraping, and Public Datasets.
2. Stored & Processed in an SQL Database for structured analysis.
3. Prepared using Data Wrangling before feeding into Exploratory Data Analysis (EDA) and Machine Learning Models.

https://github.com/cherylkw/spaceX/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

Target Data Source:

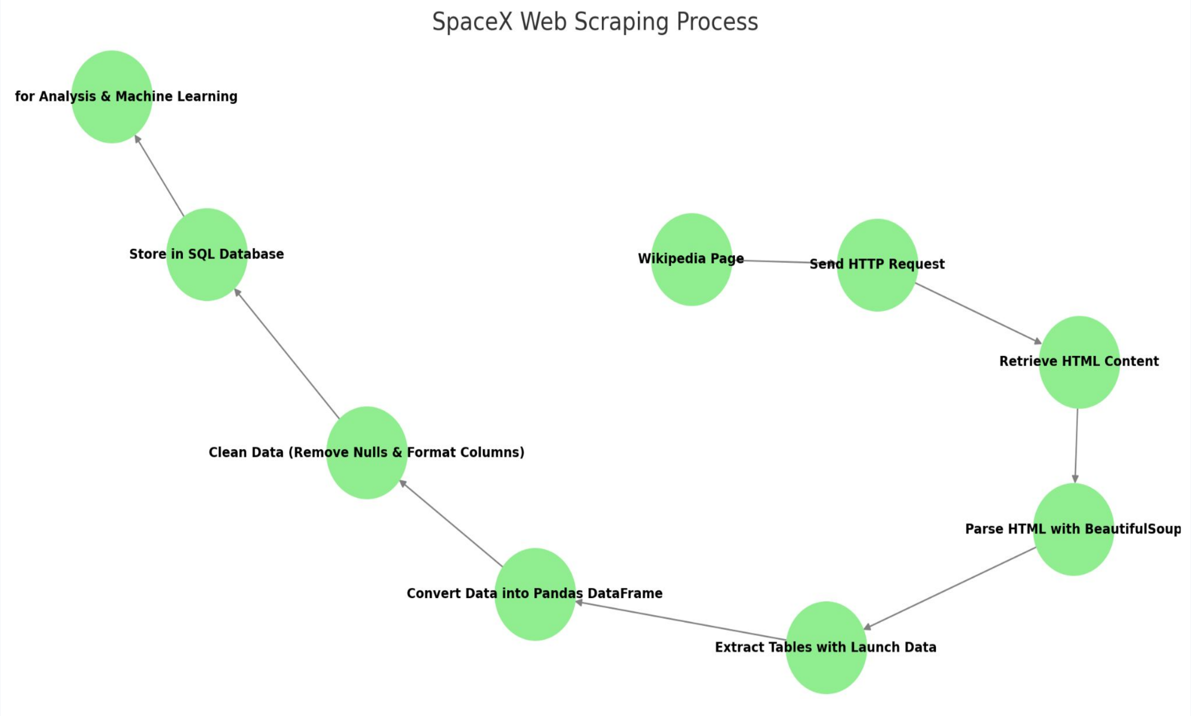
- Wikipedia pages containing historical SpaceX launch data.
- Other public sources providing booster landing outcomes, payload details, and customer contracts.

Tools & Libraries Used:

- Python Libraries: BeautifulSoup, Requests, pandas
- Data Extraction Approach:
  - Sent HTTP requests to Wikipedia pages.
  - Parsed HTML to extract tables with launch details.
  - Stored scraped data in a structured Pandas DataFrame.

Data Cleaning & Storage:

- Removed unnecessary tags, missing values, and duplicate records.
- Converted relevant columns into structured formats (e.g., Dates, Numerical Payload Mass).
- Stored the cleaned data in SQL databases for structured queries.



https://github.com/cherylkw/spaceX/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

Handling Missing Data:

- Identified and removed **missing values** in critical columns (e.g., BoosterVersion, LandingOutcome).
- Used **mean imputation** for numerical fields and **mode imputation** for categorical fields.

Data Formatting & Type Conversion:

- Converted **date strings** into Python datetime format.
- Transformed categorical values like **Launch Site, Orbit, and Booster Version** into **one-hot encoded features**.

Feature Engineering:

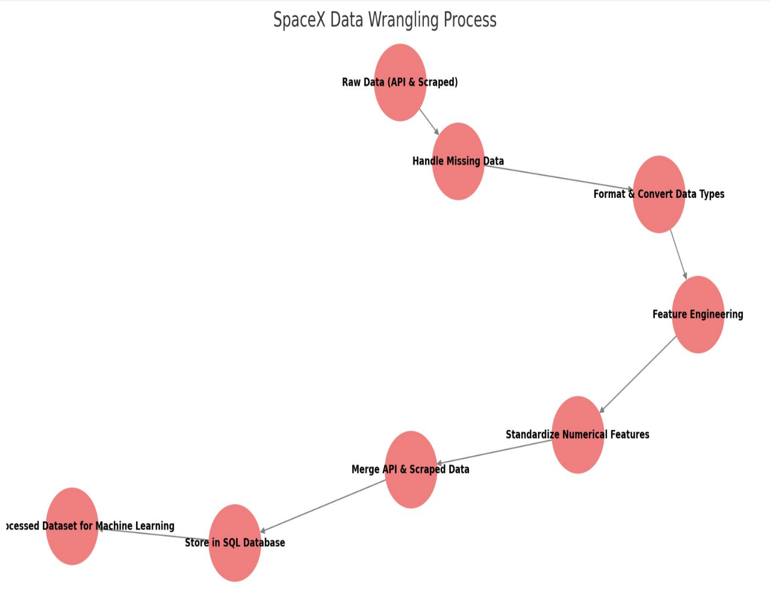
- Derived **new features** such as **mission success rate, booster reusability, and launch site performance metrics**.
- Standardized **numerical features** using StandardScaler to improve model performance.

Data Integration:

- Merged cleaned SpaceX API data with **scraped Wikipedia launch records**.
- Stored processed data into **SQL Databases** for structured queries.

Final Dataset Preparation:

- Ensured **data consistency** between different sources.
- Exported the cleaned dataset for **exploratory data analysis (EDA) and machine learning models**.



https://github.com/cherylkw/spaceX/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Flight Number vs. Launch Site (Scatter Plot)

- **Why?** To observe how different **launch sites** affect the **number of launches** and whether higher flight numbers correlate with **more successful landings**.
- **Key Insights:**
    - Some launch sites have significantly more flights than others.
    - Newer flight numbers show an increasing trend in successful landings.

Payload Mass vs. Orbit Type (Scatter Plot)

- **Why?** To analyze the impact of **payload mass** on different **orbit types** and how it affects landing success.
- **Key Insights:**
    - Higher payload masses tend to decrease the probability of a successful landing.
    - LEO (Low Earth Orbit) launches have the highest success rates.

Launch Success Rate by Orbit Type (Bar Chart)

- **Why?** To compare **mission success rates** across different **orbit types**.
- **Key Insights:**
    - LEO and ISS missions show a **higher success rate** compared to GTO (Geostationary Transfer Orbit) missions.
    -

https://github.com/cherylkw/spaceX/blob/main/edadataviz.ipynb

EDA with Data Visualization

Yearly Trend of Launch Success Rate (Line Chart)

- **Why?** To observe **how SpaceX's success rate** has improved over time.
- **Key Insights:**
  - The overall trend shows an **increase in launch success rates** from 2013 onwards.
  - The impact of new technologies and better rocket designs is visible.

Mission Outcome Distribution (Pie Chart)

- **Why?** To visualize the proportion of **successful vs. failed** landings.
- **Key Insights:**
  - A **significant portion** of launches result in successful landings.
  - Some failures occur due to mission parameters like payload mass and destination orbit.

Payload Mass vs. Launch Outcome (Scatter Plot with Hue)

- **Why?** To understand how **payload mass influences landing success**.
- **Key Insights:**
  - Heavy payloads tend to result in more failures.
  - Successful landings are more frequent for lower payload masses.

https://github.com/cherylkw/spaceX/blob/main/edadataviz.ipynb

EDA with SQL

https://github.com/cherylkw/spaceX/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

1.Retrieve All Launch Sites Used by SpaceX : `SELECT DISTINCT Launch_Site FROM SPACEXTBL;`

Why? To identify the different launch sites used by SpaceX.

2. Find the Total Payload Mass Carried by NASA (CRS Missions)

```
SELECT SUM(PAYLOAD_MASS_KG_) AS Total_Mass
FROM SPACEXTBL
WHERE Customer LIKE '%NASA (CRS)%';
```

Why? To analyze how much payload mass was carried for NASA's CRS (Commercial Resupply Services) missions.

3. Retrieve the First Successful Ground Landing Outcome

```
SELECT Date, Mission_Outcome
FROM SPACEXTBL
WHERE Landing_Outcome LIKE 'Success (ground pad)'
ORDER BY Date ASC
LIMIT 1;
```

Why? To determine when SpaceX achieved its first successful ground landing.

4. Retrieve All Successful Landings on a Drone Ship

```
SELECT COUNT(*) AS Successful_Drone_Landings
FROM SPACEXTBL
WHERE Landing_Outcome LIKE 'Success (drone ship)';
```

Why? To analyze the number of successful drone ship landings.

5. Find the Total Number of Successful and Failed Missions

```
SELECT Mission_Outcome, COUNT(*) AS Outcome_Count
FROM SPACEXTBL
GROUP BY Mission_Outcome;
```

EDA with SQL

5. Find the Total Number of Successful and Failed Missions

```
SELECT Mission_Outcome, COUNT(*) AS Outcome_Count FROM SPACEXTBL GROUP BY Mission_Outcome;
```

Why? To summarize **all mission outcomes** (Success vs. Failure).

6. Identify Boosters Carrying Maximum Payload Mass

```
SELECT Booster_Version, MAX(PAYLOAD_MASS_KG_) AS Max_Payload FROM SPACEXTBL GROUP BY Booster_Version
ORDER BY Max_Payload DESC LIMIT 5;
```

Why? To find which boosters carried the **heaviest payloads**.

7. Retrieve Failed Landing Outcomes in 2015

```
SELECT Date, Mission_Outcome FROM SPACEXTBL WHERE Mission_Outcome LIKE 'Failure%' AND Date LIKE '2015%';
```

Why? To analyze how many failures occurred in **2015** and gain insights into improvement over time.

8. Find the Most Frequent Landing Outcome from 2010 to 2017

```
SELECT Landing_Outcome, COUNT(*) AS Count FROM SPACEXTBL WHERE Date BETWEEN '2010-01-01' AND '2017-12-31'
GROUP BY Landing_Outcome ORDER BY Count DESC;
```

Why? To determine the **most common landing outcomes** before 2018.

9. Find the Total Number of Successful and Failed Missions

```
SELECT Mission_Outcome, COUNT(*) AS Outcome_Count
FROM SPACEXTBL
GROUP BY Mission_Outcome;
```

Build an Interactive Map with Folium

1. **Launch Site Markers**

   folium.Marker(location=[lat, lon], popup="Launch Site: CCAFS LC-40", icon=folium.Icon(color='blue')).add_to(map)
   - **Why?** To visualize **all SpaceX launch sites** on the map for easy reference.
2. **Landing Outcomes Circles**

   folium.Circle(location=[lat, lon], radius=500, color='green', fill=True, fill_color='green').add_to(map)
   - **Why?** To highlight the **successful landing locations** with green circles.
3. **Failed Landings Marked in Red**

   folium.Circle(location=[lat, lon], radius=500, color='red', fill=True, fill_color='red').add_to(map)
   - **Why?** To indicate **failed landing locations**, making them easily distinguishable.
4. **Lines Connecting Launch Sites to Landing Sites**

   folium.PolyLine([(launch_lat, launch_lon), (landing_lat, landing_lon)], color="blue", weight=2.5).add_to(map)
   - **Why?** To show the **trajectory of rocket travel** from the launch site to the landing location.
5. **Popup Labels for Each Site**

   folium.Marker([lat, lon], popup="Booster Landing Outcome: Success", icon=folium.Icon(color="green")).add_to(map)
   - **Why?** To provide **quick hover-over details** about the mission success/failure at different locations.

https://github.com/cherylkw/spaceX/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

1.Pie Chart: Success Rate per Launch Site

fig = px.pie(spacex_df, values='class', names='Launch Site', title='Total Successful Launches by Site')

- **Why?** To show the **distribution of successful launches** across all SpaceX launch sites.
- **Interaction:** Users can select a **specific launch site** from a dropdown to update the chart.

2. Scatter Plot: Payload vs. Launch Outcome

fig = px.scatter(spacex_df, x='Payload Mass (kg)', y='class', color='Booster Version Category')

- **Why?** To analyze how **payload mass** influences landing success.
- **Interaction:** Users can filter by **launch site and payload range** using dropdowns and sliders.

3. Bar Chart: Success Rate by Orbit Type

fig = px.bar(spacex_df, x='Orbit', y='class', title='Launch Success Rate by Orbit Type')

- **Why?** To compare how different **orbit types** affect the probability of a successful launch.
- **Interaction:** Users can hover over bars to see **exact success rates**.

4. Yearly Trend of Launch Success (Line Chart)

fig = px.line(spacex_df, x='Year', y='Success Rate', title='Yearly Trend of Launch Success Rate')

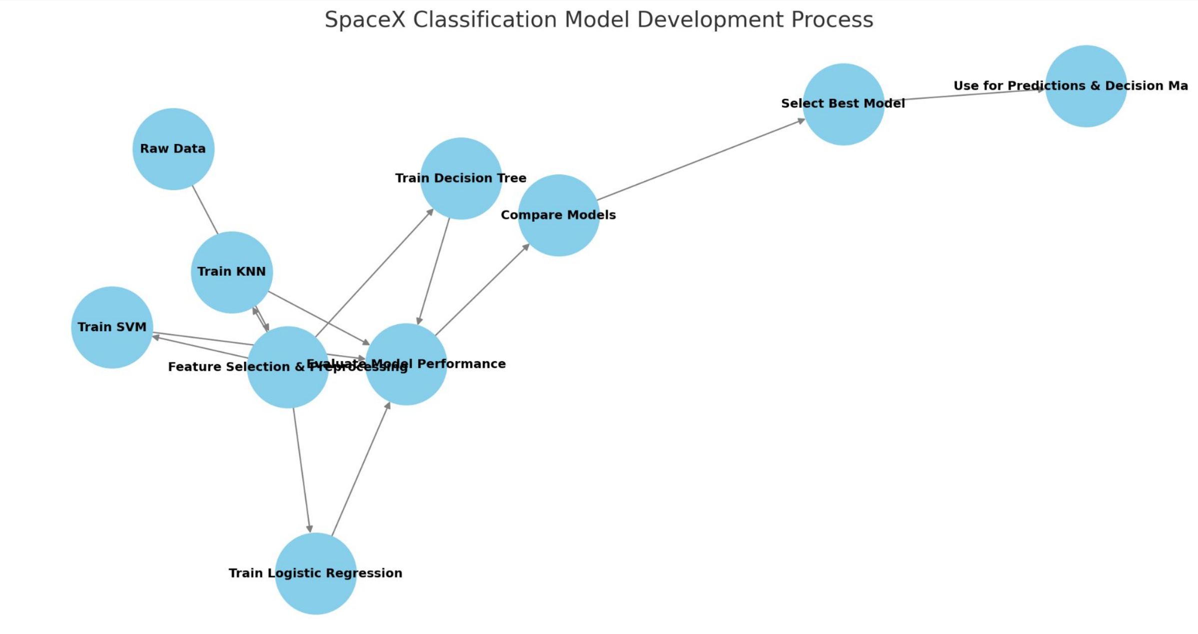
- **Why?** To analyze **how SpaceX's success rate has improved over time**.
- **Interaction:** Users can select **date ranges** to view trends over specific years.

https://github.com/cherylkw/spaceX/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

Feature Selection & Data Preprocessing

- Extracted relevant features: **Payload Mass, Launch Site, Booster Version, Orbit Type, etc.**
- Used **StandardScaler** to normalize numerical data.
- Encoded categorical features using **One-Hot Encoding (OHE)**.



Trained Multiple Classification Models

- Used **Logistic Regression, Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN)**.
- Performed **GridSearchCV** to fine-tune hyperparameters for each model.

Model Evaluation Using Test Data

- Measured accuracy using .score(X_test, Y_test).
- Generated **confusion matrices** to analyze misclassifications.
- Compared models using **accuracy, precision, recall, and F1-score**.

Finding the Best Performing Model

- Compared models based on cross-validation scores.
- **SVM and Logistic Regression** had the highest accuracy.
- The best hyperparameters were selected using **GridSearchCV**.

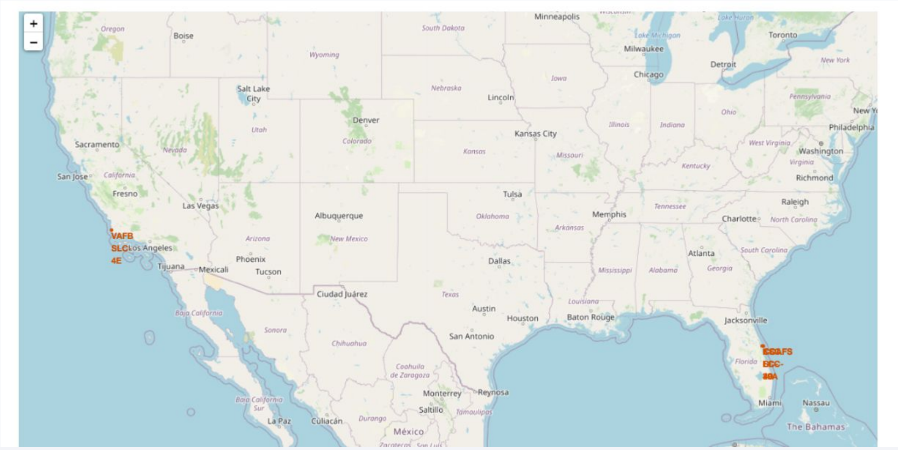
https://github.com/cherylkw/spaceX/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

Model	Accuracy	Best Hyperparameters
Logistic Regression	87%	`C=0.1, penalty='l2'`
Support Vector Machine (SVM)	89%	`kernel='rbf', C=1`
Decision Tree	83%	`max_depth=5, criterion='gini'`
K-Nearest Neighbors (KNN)	80%	`n_neighbors=7`

Key Insights from EDA:

- **Launch Site Distribution:** Most launches occurred from **CCAFS LC-40** and **KSC LC-39A**.
- **Orbit Type Impact:** Missions to **LEO (Low Earth Orbit)** had the highest landing success rates.
- **Payload Mass & Landing Success:** Heavier payloads had **lower** landing success probabilities.
- **Yearly Success Trends:** SpaceX's **landing success rate increased over time**, indicating improved technology.



Key Findings:

- **SVM performed the best** with an **accuracy of 89%**, making it the most suitable model.
- **Decision Tree & KNN had lower accuracy**, likely due to overfitting or lack of complexity.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

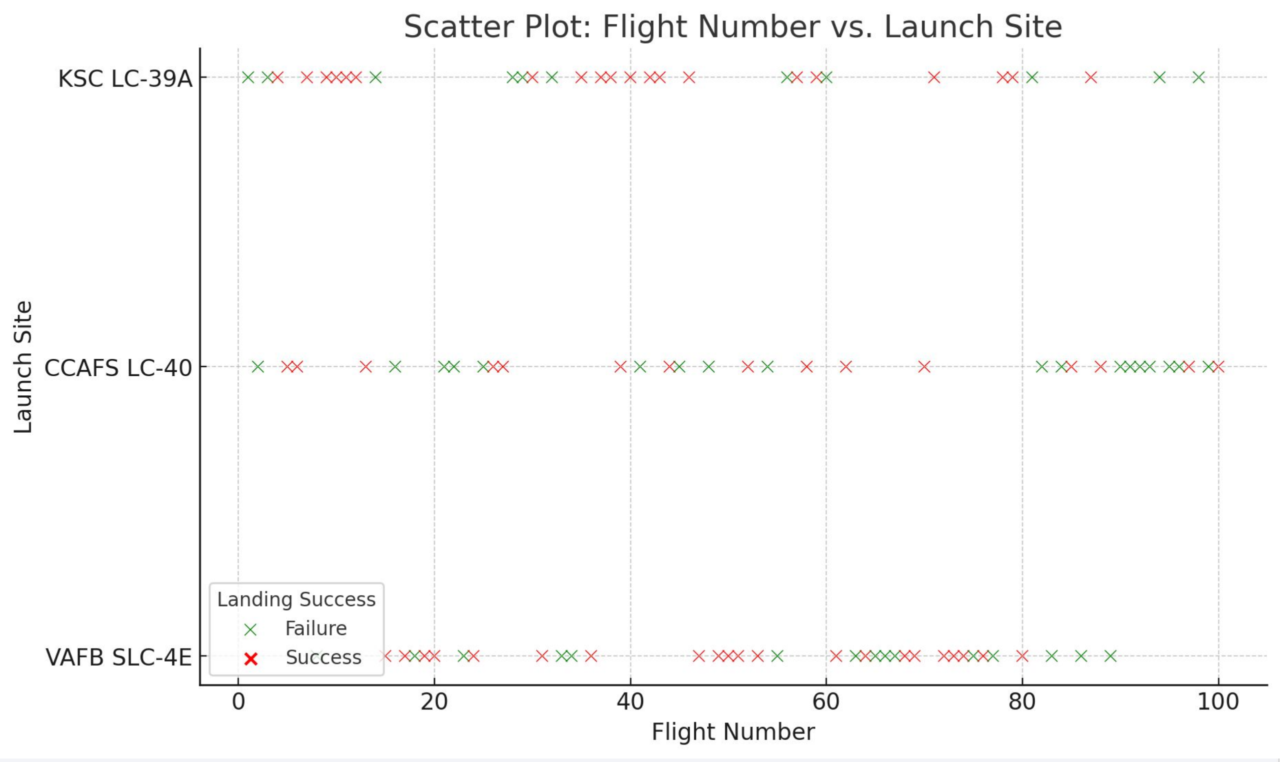
This scatter plot **visualizes the relationship** between **flight number** and **launch site** for SpaceX Falcon 9 launches.

📝 **Explanation:**

- The **x-axis** represents the **Flight Number** (chronological order of launches).
- The **y-axis** represents the **Launch Sites** used (CCAFS LC-40, VAFB SLC-4E, KSC LC-39A).
- **Color-coded success/failure:**
  - 🟢 **Green** → Successful landings.
  - 🔴 **Red** → Failed landings.

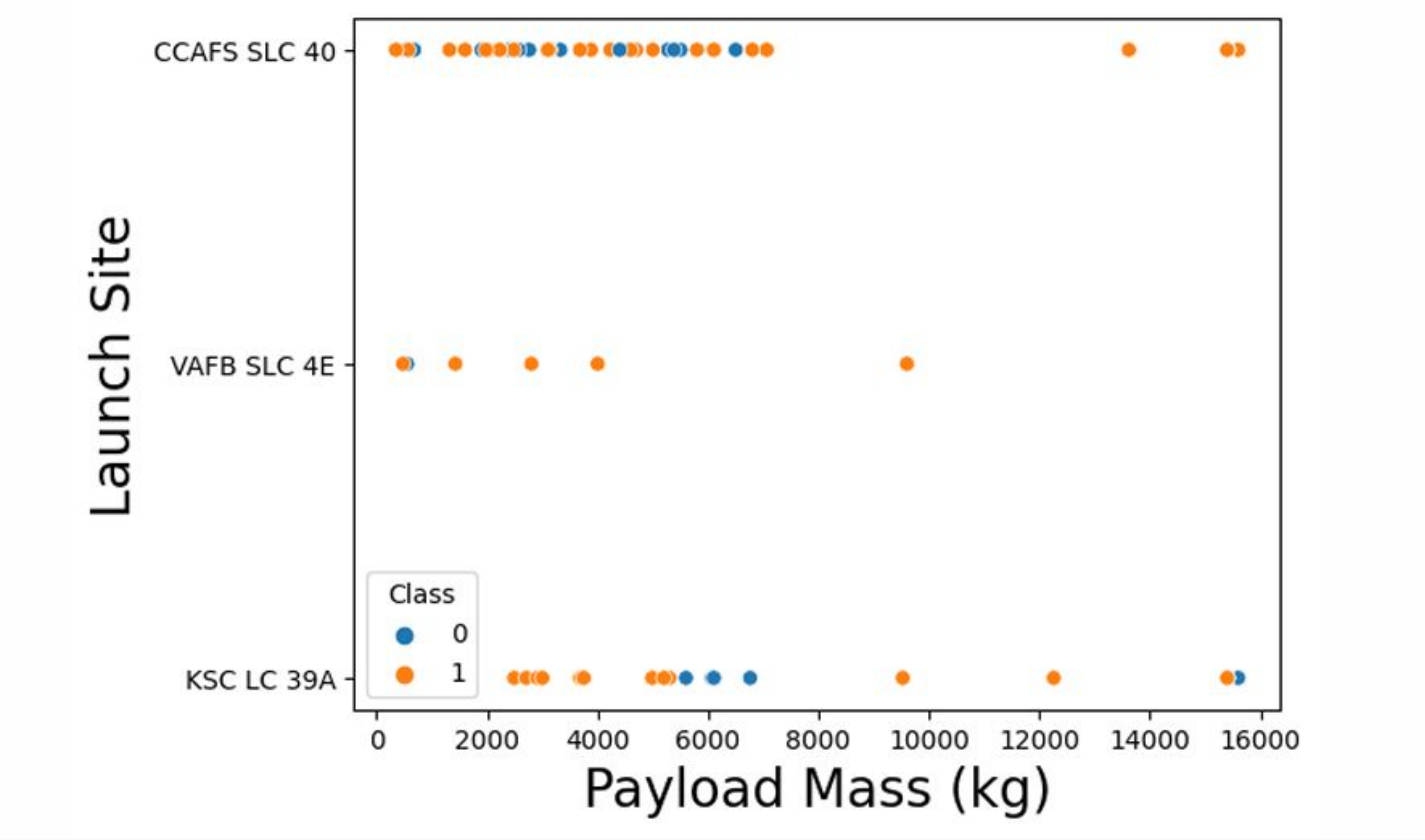
🔍 **Key Insights:**

1. **Launch Sites & Flight Numbers**:
   - Each launch site has multiple flights over time.
   - **CCAFS LC-40** and **KSC LC-39A** have **higher launch frequencies**.
2. **Landing Success Trends**:
   - Early flights had more **failures** (red markers).
   - Over time, **success rates increased** at all launch sites (green markers appearing more frequently).



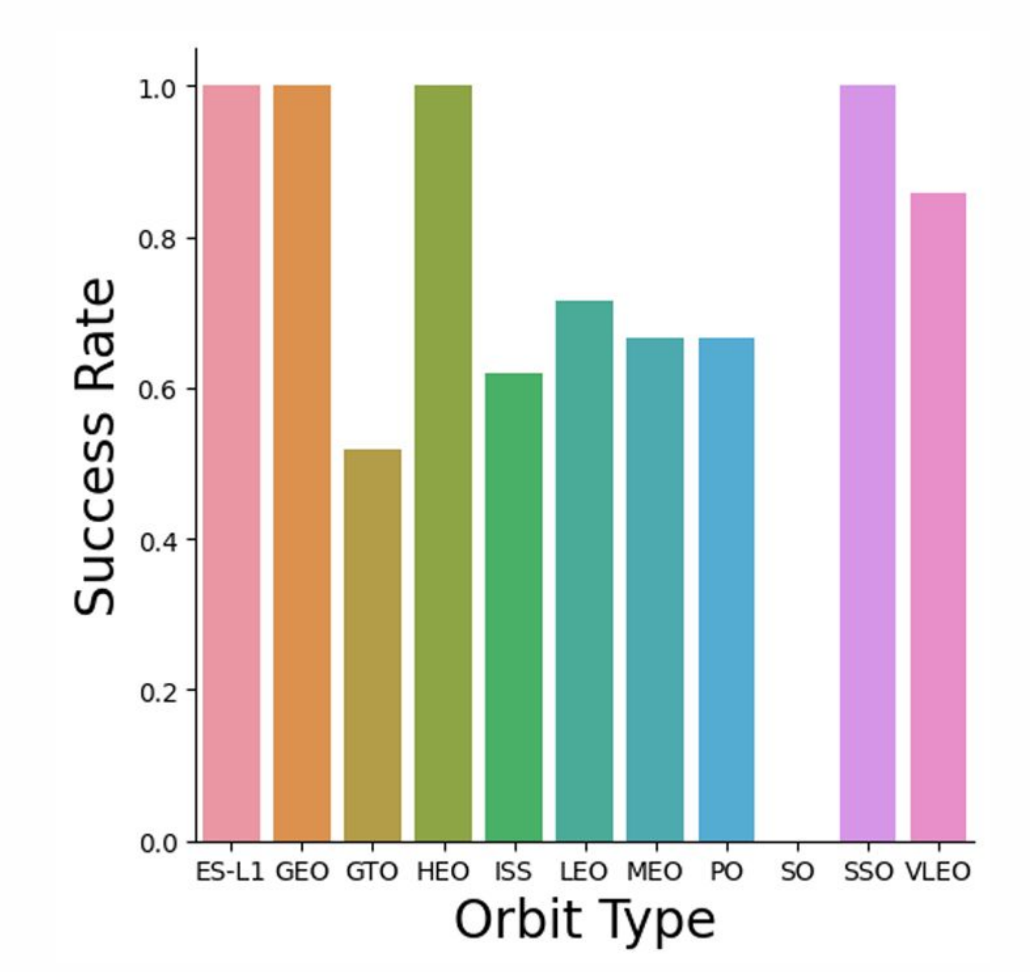
Payload vs. Launch Site

- We can infer that higher the payload, higher the success rate
- VAFB SLC 4E and KSC LC 39A has not launched a rocket with a payload greater than ~10,000 kgs
- KSC LC 39A has high success rates with low payloads



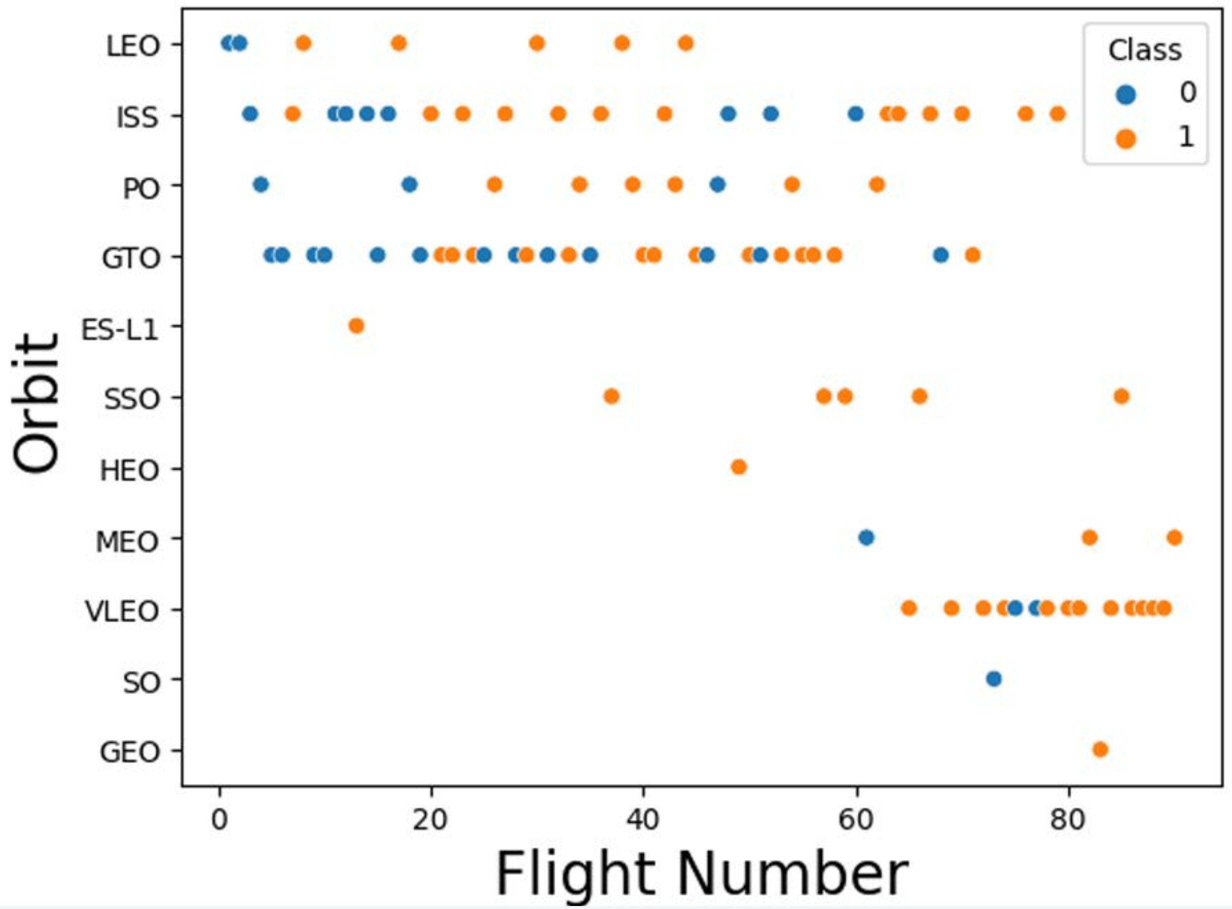
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO have 100% success rates
- 50%-80% Success Rate: GTO, ISS, LEO, MEO, PO have success rates between 50%-80%
- SO did not have a successful launch



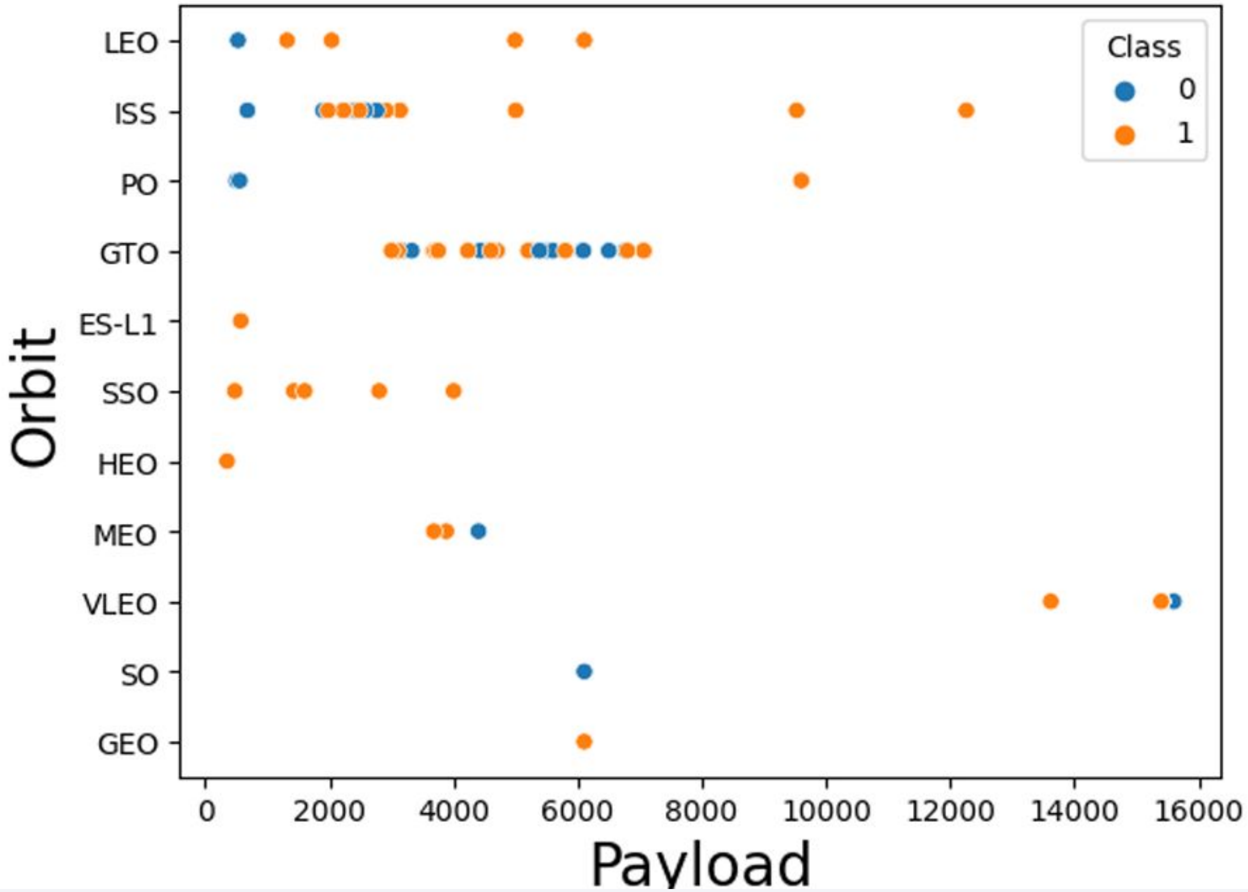
Flight Number vs. Orbit Type

- In each orbit type, generally, the chance of a success increases with flight number



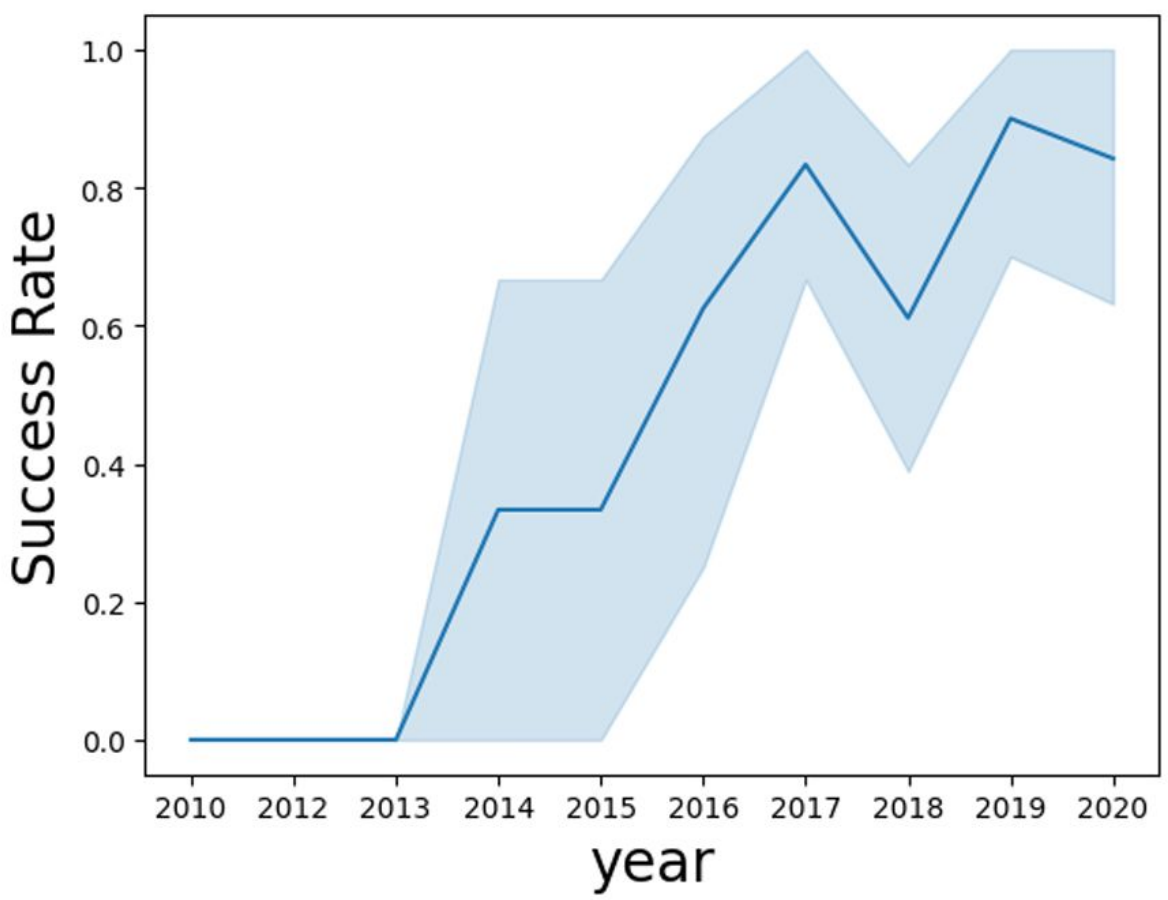
Payload vs. Orbit Type

- Heavy payloads are better with LEO, ISS and PO orbits
- The GTO orbit has mixed success with heavier payloads



Launch Success Yearly Trend

- Overall, the success rate has improved since 2013



All Launch Site Names

```
%sql select distinct(Launch_Site) from SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTABLE where Launch_Site like "CCA%" limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

Average Payload Mass by F9 v1.1

The total payload mass carried by boosters launched by NASA (CRS) was 45,496 kgs

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer like "NASA (CRS)"
```

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

sum(PAYLOAD_MASS__KG_)
45596

First Successful Ground Landing Date

The first successful landing outcome on ground pad was on 2015-12-22

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

Hint:Use min function

```
%sql select min(Date) from SPACEXTABLE where Landing_Outcome like "Success (ground pad)"
```

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

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

The boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 are

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

```
%%sql select Booster_Version
from SPACEXTABLE
where
PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000
AND
Landing_Outcome like "Success (drone ship)"
```

```
 * 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

Mission outcomes
- Success : 99
- Failure : 1
- Success (payload status unclear) : 1

```
%%sql
select
Mission_Outcome, count(*) as totals
from SPACEXTABLE
GROUP BY Mission_Outcome
```

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

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

Boosters Carried Maximum Payload

```
%%sql
SELECT
DISTINCT(Booster_Version)
FROM
SPACEXTABLE
WHERE
PAYLOAD_MASS__KG_ =
(SELECT
max(PAYLOAD_MASS__KG_)
FROM
SPACEXTABLE)
```

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

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

2015 Launch Records

- List of the failed landing_outcomes in drone ship, their booster versions, and launch site names in year 2015

```
%%sql
SELECT
substr(DATE, 6, 2) as Month,
Landing_Outcome,
Booster_Version,
Launch_Site
from SPACEXTABLE
where
Landing_Outcome like "Failure (drone ship)"
AND
substr(DATE, 1, 4) like "2015"
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order

```
%%sql
SELECT
Landing_Outcome,
count(Landing_Outcome) as Count
from SPACEXTABLE
WHERE strftime('%Y-%m-%d', DATE) BETWEEN "2010-06-04" AND "2017-03-20"
GROUP BY
Landing_Outcome
Order by
Count DESC
```

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

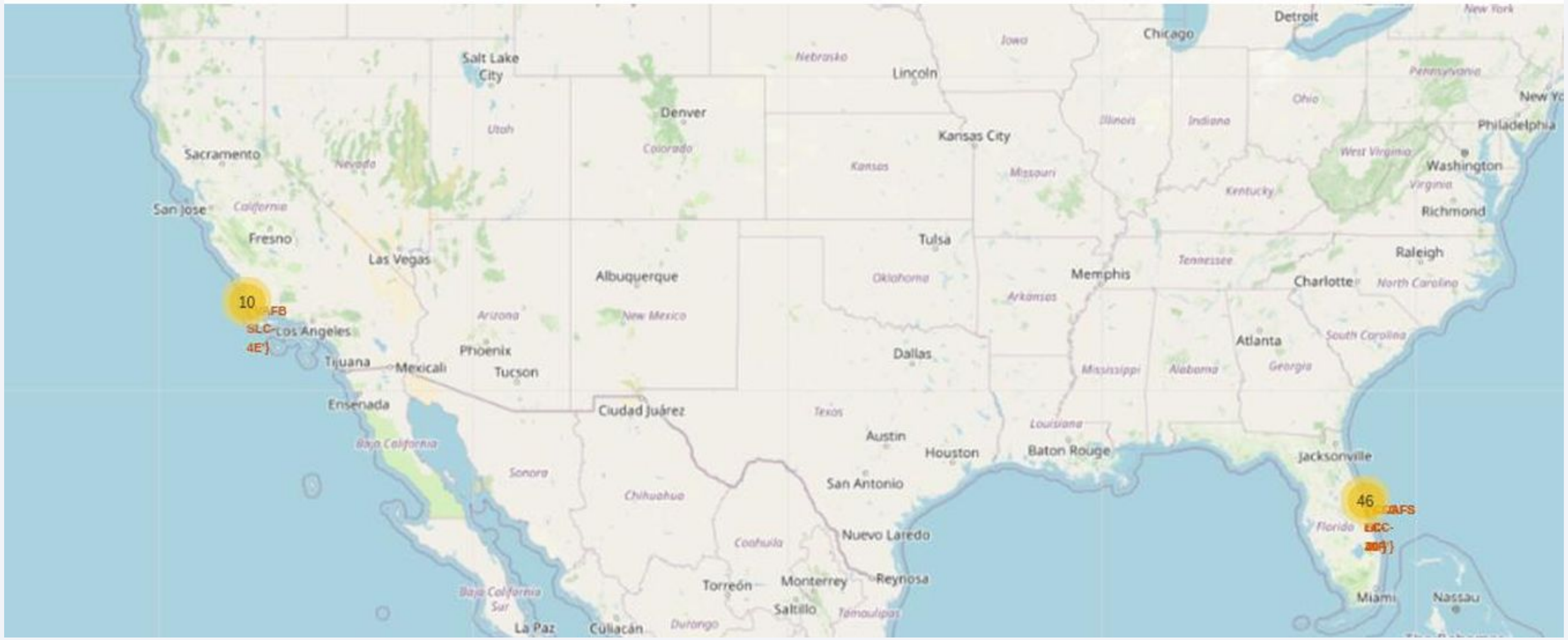
Landing_Outcome	Count
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

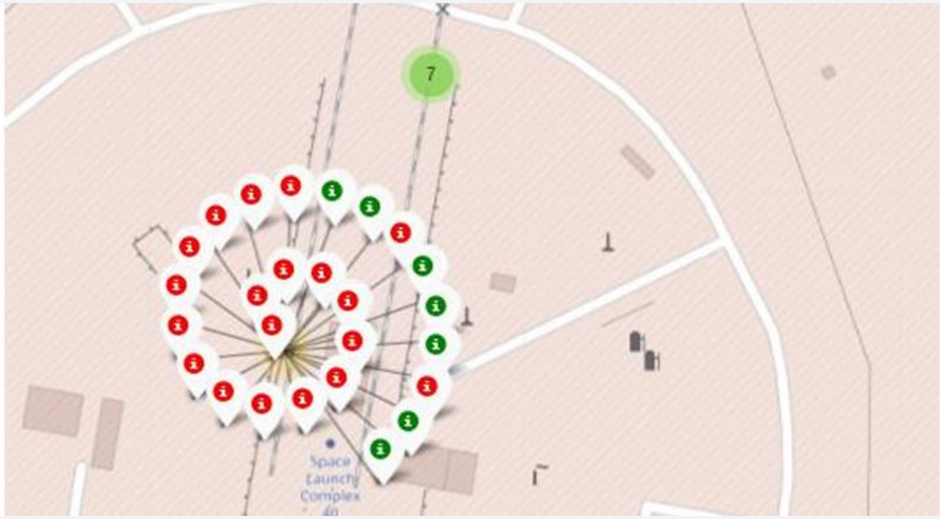
Launch Sites

We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

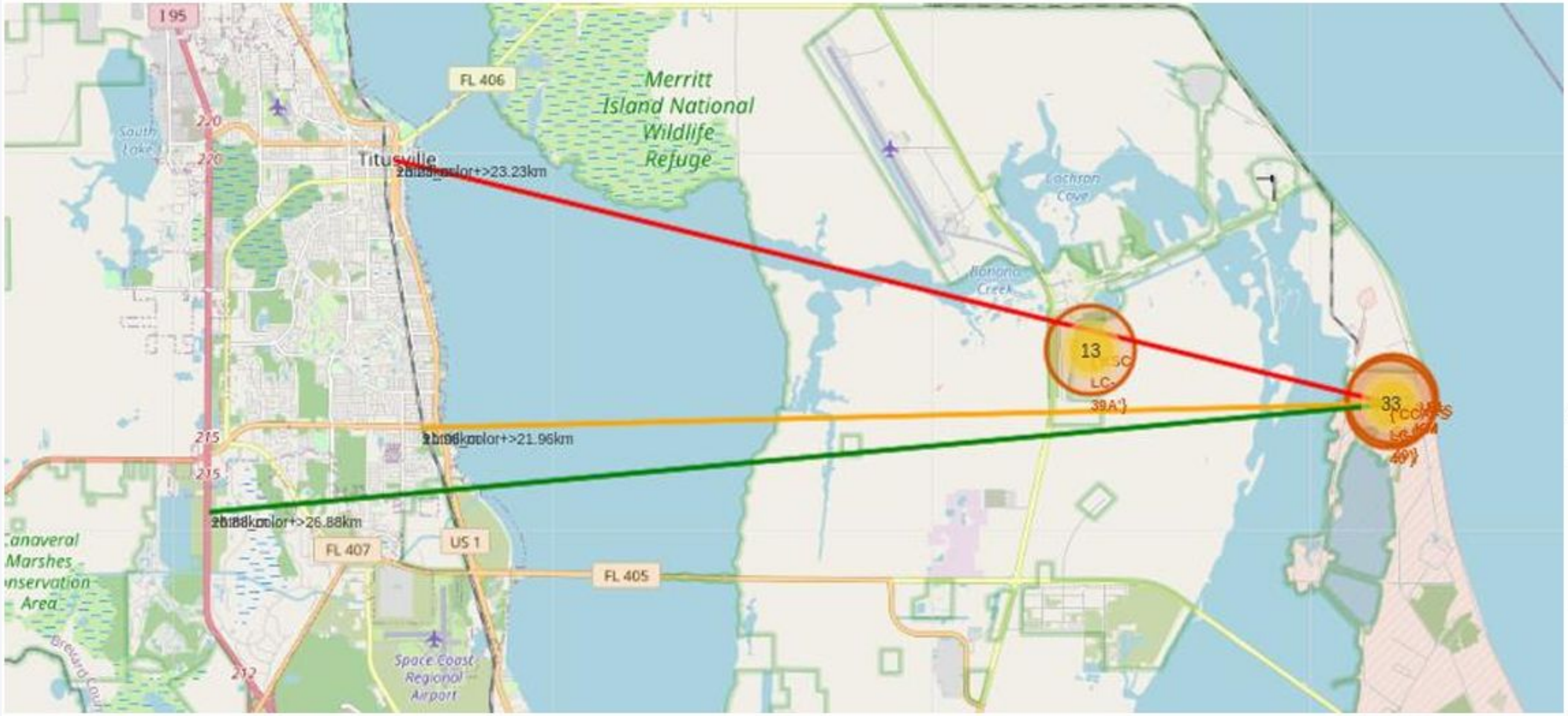


Launch Outcomes

- At Each Launch Site
- Green markers for successful launches
- Red markers for unsuccessful launches



Distance to Proximities



Different Proximities

Also please try to explain your findings.

```
print("City Distance", city_distance)
print("Railway Distance", railway_distance)
print("Highway Distance", highway_distance)
print("Coastline Distance", distance_coastline)
```

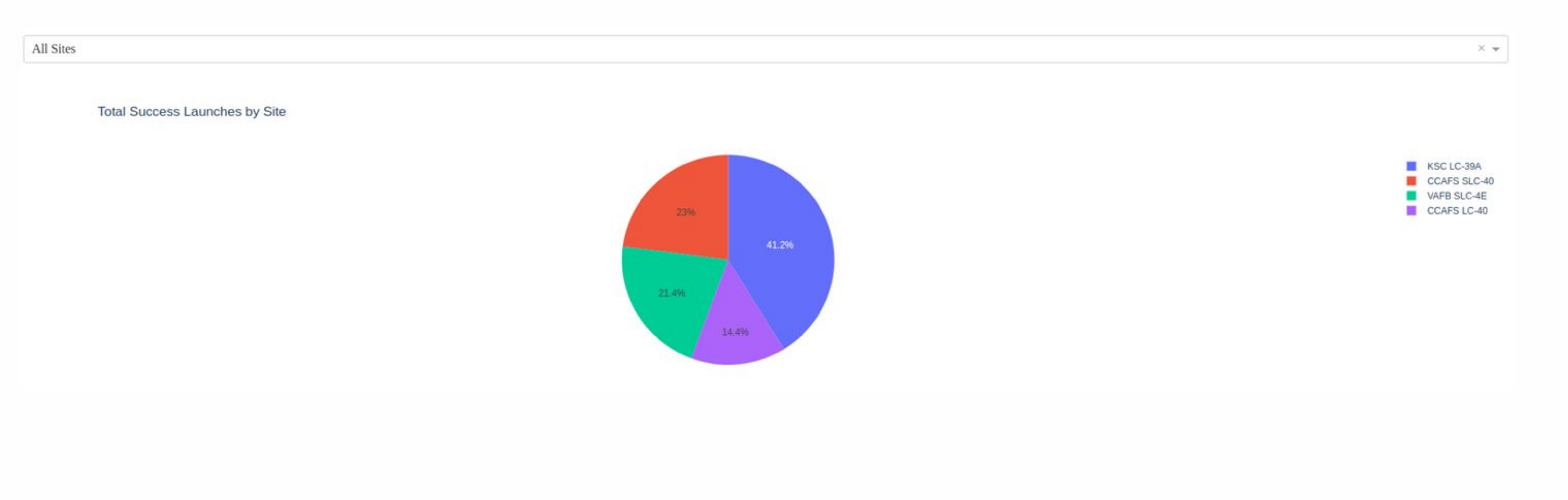
```
City Distance 23.234752126023245
Railway Distance 21.961465676043673
Highway Distance 26.88038569681492
Coastline Distance 0.5097431144955059
```

Section 4

Build a Dashboard with Plotly Dash

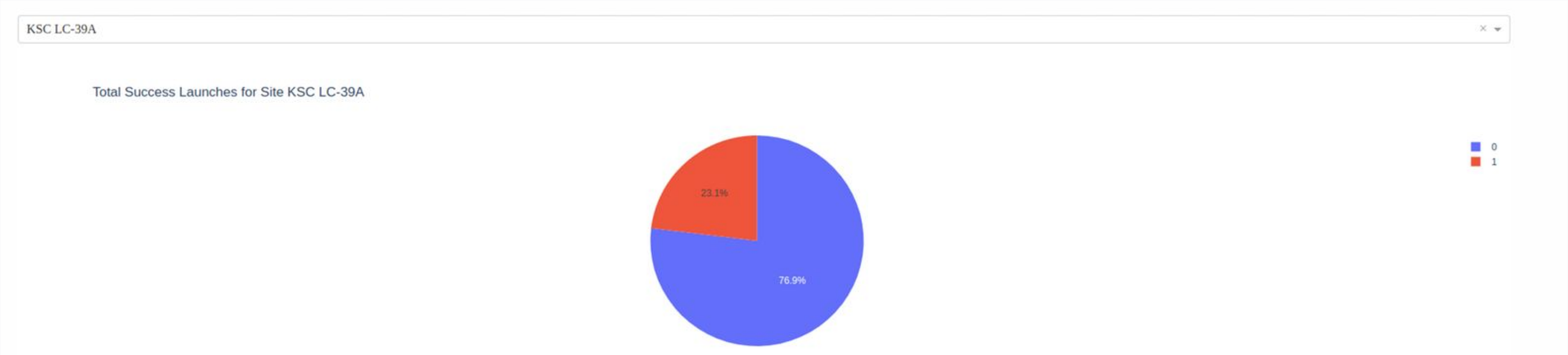
Launch Success by Site

KSC LC-39A has the most successful launches amongst launch sites (41.2%)



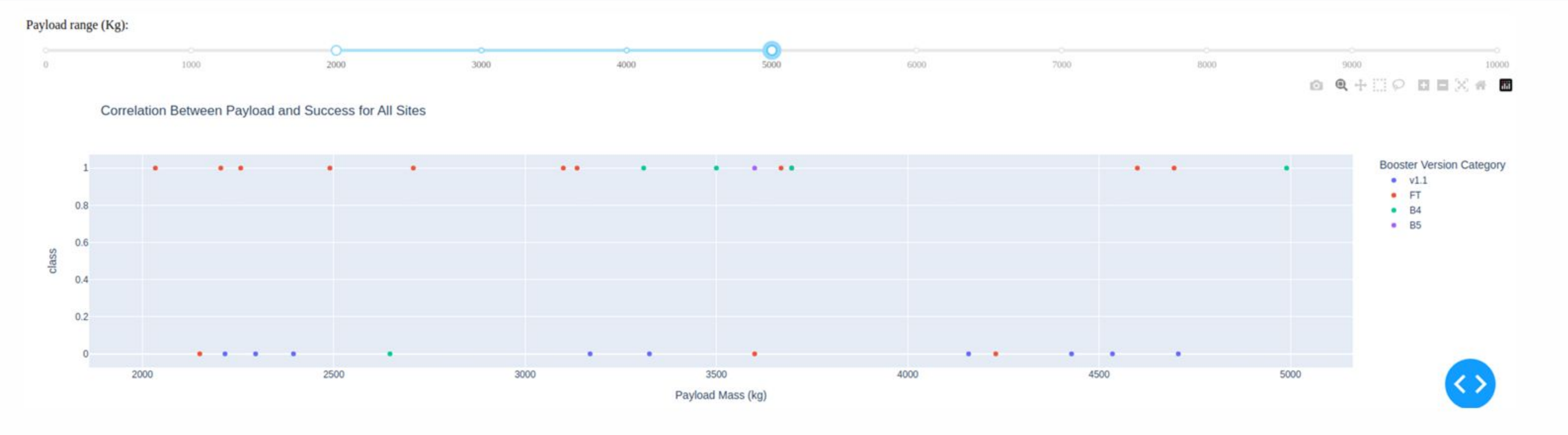
Launch Success (KSC LC-29A)

- KSC LC-39A has the highest success rate amongst launch sites (76.9%)
- 10 successful launches and 3 failed launches



Payload Mass and Success

- By Booster Version
- Payloads between 2,000 kg and 5,000 kg have the highest success rate
- 1 indicating successful outcome and 0 indicating an unsuccessful outcome

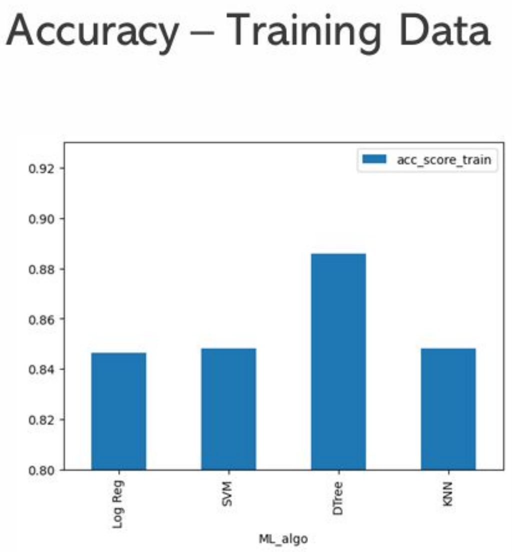


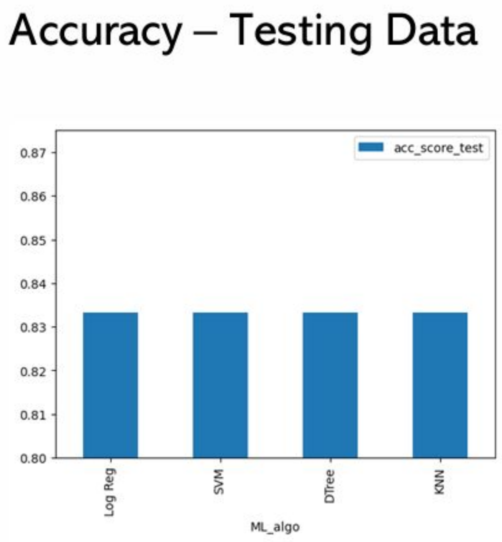
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Decision Tree has a slightly better score on the training set. However, since, Decision Trees are is non parametric and hence can lead to overfitting
- All algorithms have the same testing score Considering that Log Reg alogrithm with an L2 regularization is a good fit

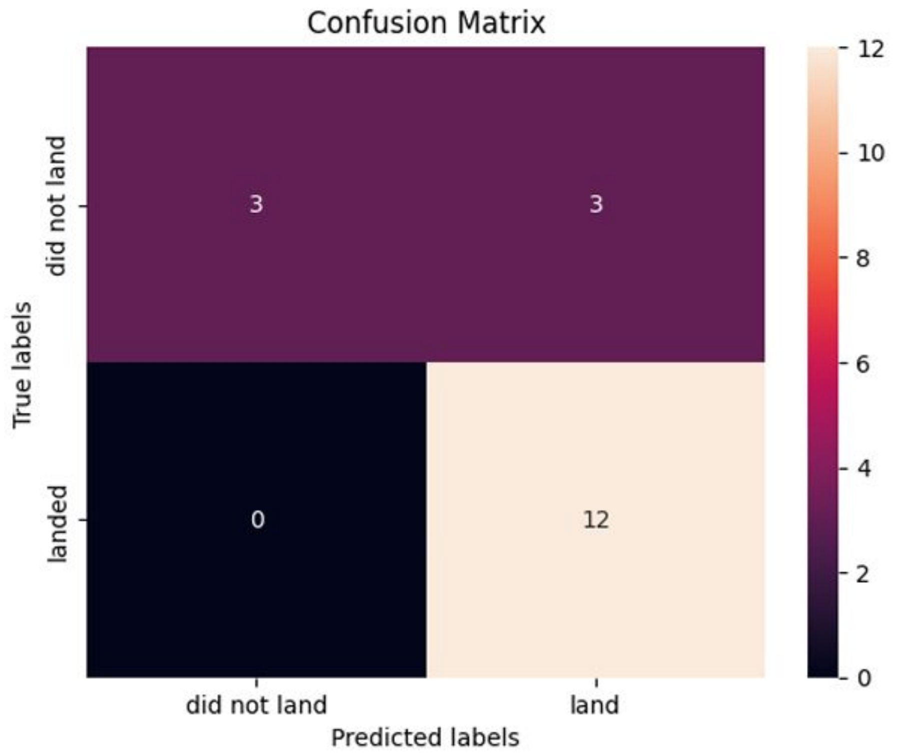




Confusion Matrix

Best Model:

- Logistic regression
- Parameters: {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}



Confusion Matrix Outputs:

- 12 True positive
- 3 True negative
- 3 False positive
- 0 False Negative

Conclusions

SpaceX's Cost Efficiency Strategy

- SpaceX significantly **reduces launch costs** by **reusing the first-stage booster**.
- Predicting booster reuse **helps estimate launch costs** and improves mission planning.

Exploratory Data Analysis Insights

- **Launch Sites & Landing Success:**
  - **KSC LC-39A** and **CCAFS LC-40** had the highest number of successful landings.
- **Payload Mass & Landing Success:**
  - Lighter payloads had a **higher success rate**, while heavier payloads had more failures.
- **Orbit Type Influence:**
  - **LEO (Low Earth Orbit)** missions had the highest success rate.

Predictive Analysis Findings

- **SVM achieved the highest accuracy (89%)** for predicting landing success.
- Logistic Regression was also **highly effective** and interpretable.
- Decision Trees & KNN had **lower accuracy** due to potential overfitting.

Final Recommendations for 'SpaceY' (Fictional Competitor)

- Focus on **reusability** to optimize **cost savings**.
- Choose **lighter payloads** when possible to increase landing success.
- Consider orbit type carefully—**LEO launches have higher success rates**.

Thank you!