IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Reshma Karkal
21.01.2026

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- Summary of all results
  - EDA results
  - Interactive analytics
  - Predictive analysis

Introduction

- **Project Background and Context:**

  In this capstone project, we aim to predict the successful landing of the Falcon 9 first stage. SpaceX advertises rocket launches at a significantly lower cost compared to other providers, largely due to their ability to reuse the first stage of the rocket. By accurately predicting landing success, we can estimate launch costs and provide valuable insights for companies bidding against SpaceX.

- **Problems We Want to Find Answers To:**
  - What factors influence the successful landing of the Falcon 9 first stage?
  - How can we accurately predict the landing outcome using machine learning models?
  - Which machine learning model performs best in predicting the landing success?

Section 1

Methodology

Methodology

- **Executive Summary:** This project employs a comprehensive approach to predict the successful landing of the Falcon 9 first stage, incorporating data collection, processing, exploratory analysis, interactive visualizations, and predictive modeling.
- Data collection methodology:
  - SpaceX Open-Source Rest API
  - Web Scraping from Wikipedia page 'List of Falcon 9 and Falcon Heavy Launches'
- Perform data wrangling
  - Transforming categorical data using One Hot Encoding for machine learning algorithms and removing any empty or unnecessary information from the dataset.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistic Regression, K-Nearest Neighbors, Support Vector Machines, and Decision Tree models have been developed to determine the most effective classification method.

Data Collection

- The data sets are collected using 2 methods:

1) Request to the SpaceX API

  - Gathered SpaceX's past launch data via their open-source API.
  - Retrieved and processed this data with GET request.
  - Ensured the data included only Falcon 9 launches.
  - Filled in missing payload weights from secret missions with average values.

2) Web Scraping

  - Requested past Falcon 9 and Falcon Heavy launch data from Wikipedia's relevant page.
  - Accessed the Falcon 9 Launch page via its direct Wikipedia link.
  - Extracted all the column names from the HTML table.
  - Parsed and transformed the table into a Pandas data frame suitable for analysis.

Data Collection

- Step 1: SpaceX API Request

- Step 2: Web Scraping Wikipedia

- Step 3: Data Integration

Data Collection – SpaceX API

Step 1: Initiate API Request

- Use Python's `requests` library to connect to the SpaceX API.
- Endpoint: `https://api.spacexdata.com/v4/launches`

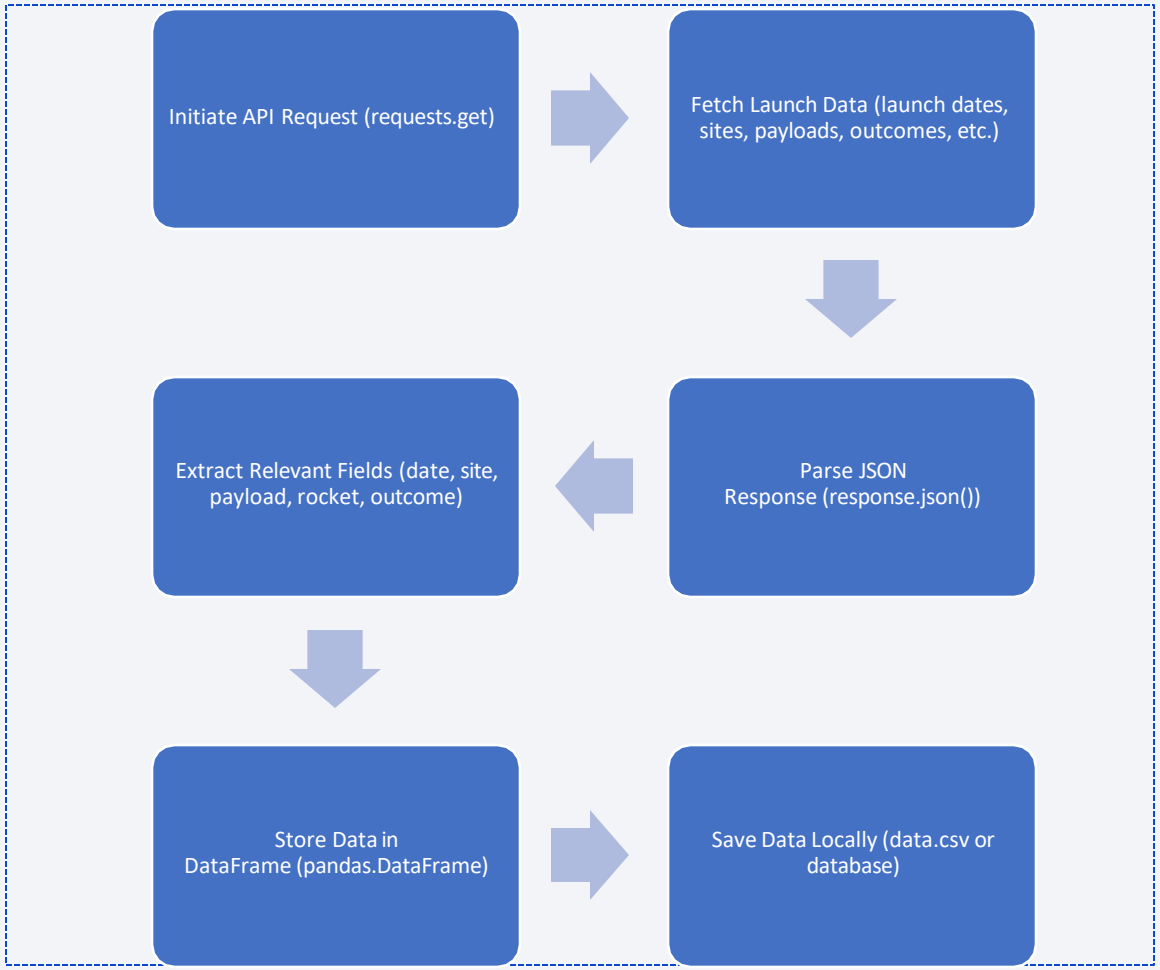
Step 2: Parse API Response

- Convert API response from JSON to a Python dictionary.
- Extract relevant fields: launch date, launch site, payload mass, rocket type, outcome.

Step 3: Store Data Locally

- Save extracted data into a pandas DataFrame.
- Store the DataFrame locally for further processing.

GitHub URL: Module 01-1 Hands-on Lab Complete the Data Collection API Lab.ipynb



Data Collection - Scraping

Step 1: Initiate Web Scraping

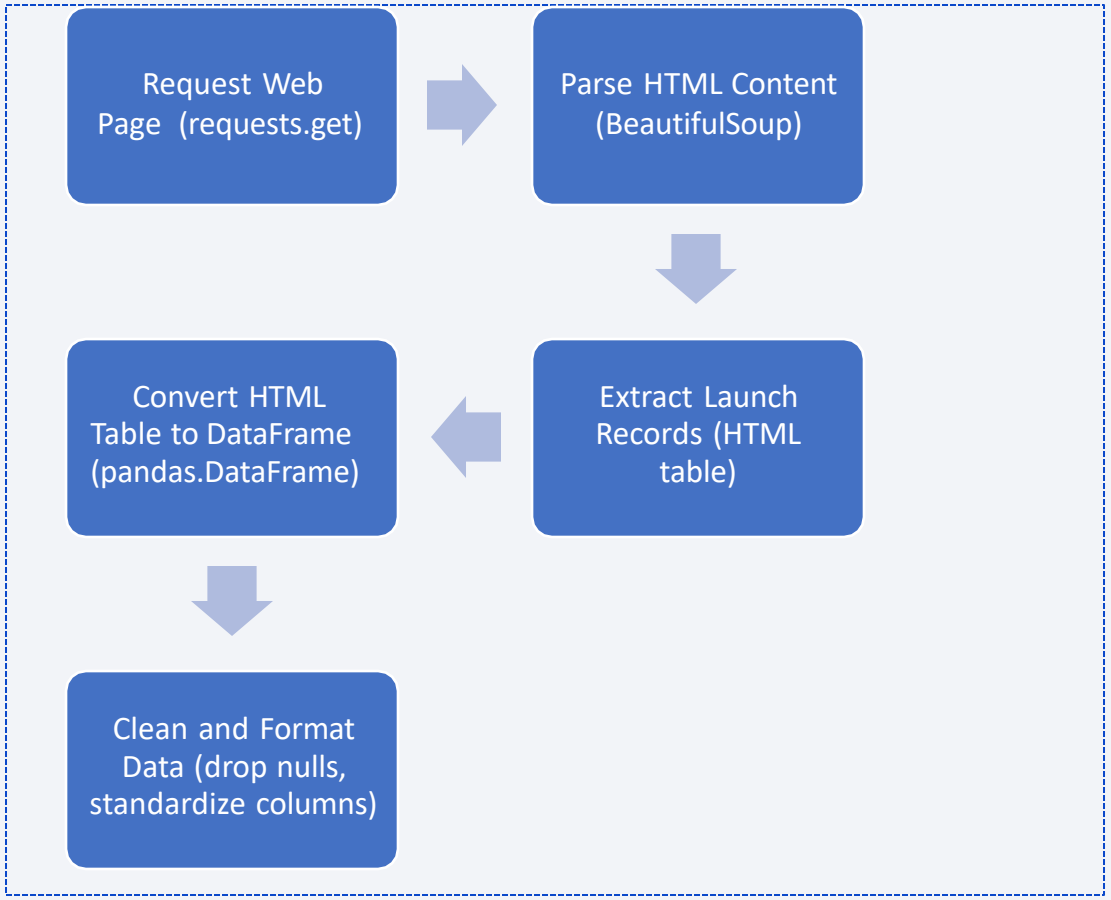
- Use Python's `requests` library to fetch the HTML content of the Wikipedia page.
- Target URL: `https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches`

Step 2: Parse HTML Content

- Use `BeautifulSoup` to parse the HTML content.
- Extract the HTML table containing Falcon 9 launch records.

Step 3: Convert to DataFrame

- Convert the extracted HTML table into a pandas DataFrame.
- Clean and format the DataFrame, ensuring data consistency.



GitHub URL: Module 01-2 Hands-on Lab Complete the Data Collection with Web Scraping lab.ipynb

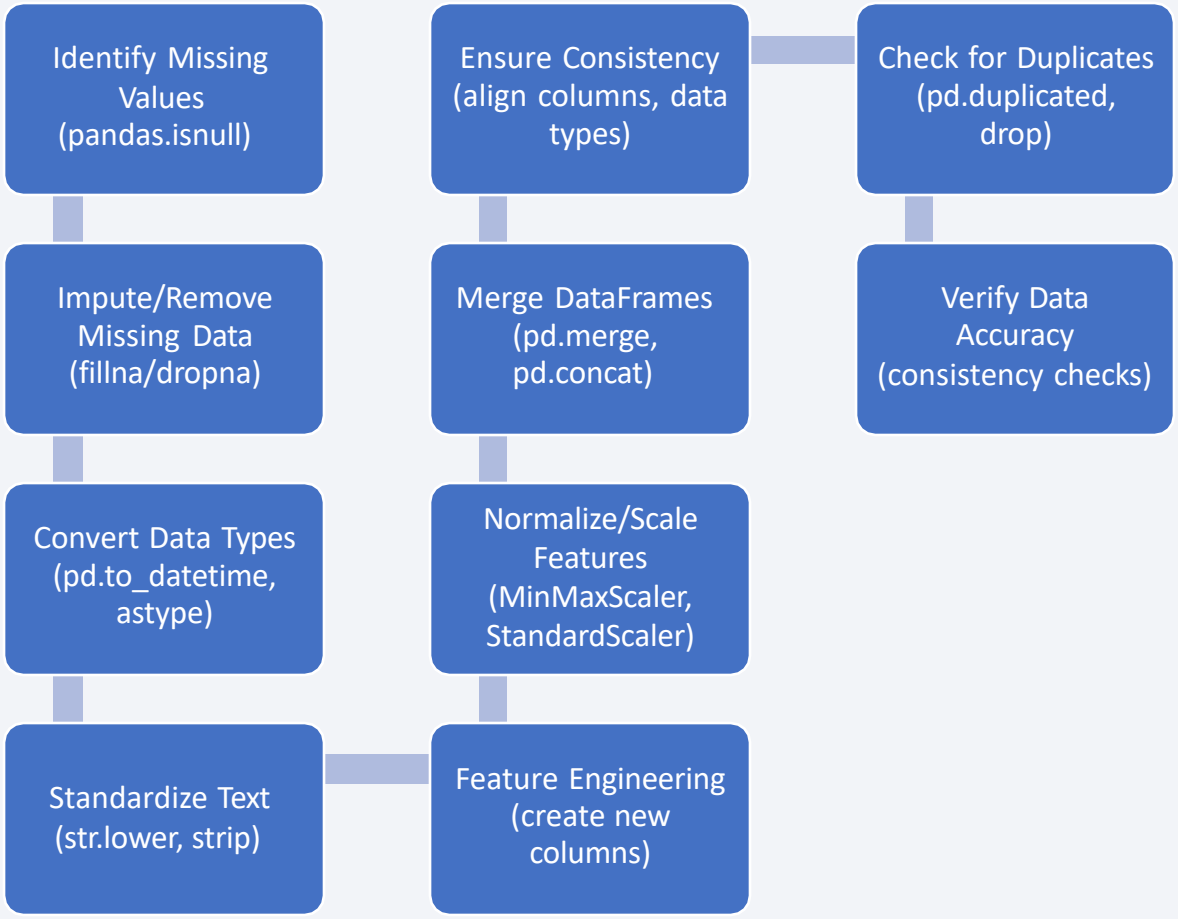
Data Wrangling

Overview: Data wrangling involves cleaning, transforming, and organizing raw data into a structured format suitable for analysis.

- Step 1: Data Cleaning
  - Identify and fill or remove missing values in the dataset.
  - Use appropriate imputation techniques or drop rows/columns with excessive missing data.
- Step 2: Data Transformation
  - Convert data types to appropriate formats (e.g., date-time, numerical).
  - Standardize text (e.g., lowercase, remove whitespace).
  - Create new features from existing data (e.g., extract year from date).
  - Normalize/scale numerical features to ensure consistency.
- Step 3: Data Integration
  - Merge datasets collected from different sources (API, web scraping) into a single cohesive dataset.
  - Ensure consistent column names and data formats across datasets.
- Step 4: Data Validation
  - Check for duplicate records and remove them.
  - Verify the accuracy and consistency of data entries.

GitHub URL: Module 01-3 Hands-on Lab Data wrangling.ipynb

Data Wrangling



EDA with Data Visualization

- **Charts Plotted:**

1. **Histograms:**

Used to visualize the distribution of numerical variables such as launch success rates, payload mass, and flight number. Helps in understanding the spread and central tendency of the data, identifying outliers, and assessing data skewness.

2. **Bar Charts:**

Used to compare categorical variables such as launch outcomes (success/failure) across different categories like launch sites or rocket types. Provides a clear comparison of frequencies or proportions within categorical data, highlighting patterns or trends.

3. **Line Charts:**

Used to track trends over time, such as the success rate of Falcon 9 launches across different years. Reveals temporal patterns and helps in understanding performance trends or changes over specific periods.

4. **Scatter Plots:**

Used to explore relationships between two numerical variables, such as payload mass vs. launch success. Identifies correlations or dependencies between variables, visualizing how one variable changes concerning another.

5. **Heatmaps:**

Used to visualize correlation matrices between multiple numerical variables. Helps in identifying strong correlations (positive or negative) between variables, aiding feature selection or understanding multicollinearity.

6. **Box Plots:**

Used to display the distribution of numerical data through their quartiles. Visualizes the spread and skewness of data, highlighting outliers and comparing distributions across different categories.

- **Github URL:** Module 02-1 Hands-on Lab Complete the EDA with SQL.ipynb

EDA with SQL

- Performed SQL queries:
    - Displaying the names of the unique launch sites in the space mission
    - Displaying 5 records where launch sites begin with the string ‘CCA'
    - Displaying the total payload mass carried by boosters launched by NASA (CRS)
    - Displaying average payload mass carried by booster version F9 v1.1
    - Listing the date when the first successful landing outcome in ground pad was achieved
    - Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
    - Listing the total number of successful and failure mission outcomes
    - Listing the names of the booster versions which have carried the maximum payload mass
    - Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015
    - Ranking 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

Github URL: Module 02-2 EDA with Visualization Lab .ipynb

Build an Interactive Map with Folium

- Markers of all Launch Sites:
  - Added Marker with Circle, Popup Label and Text Label of NASA Johnson Space Center using its latitude and longitude coordinates as a start location.
  - Added Markers with Circle, Popup Label and Text Label of all Launch Sites using their latitude and longitude coordinates to show their geographical locations and proximity to Equator and coasts.

- Colored Markers of the launch outcomes for each Launch Site
  - Added colored Markers of success (Green) and failed (Red) launches using Marker Cluster to identify which launch sites have relatively high success rates.

- Distances between a Launch Site to its proximities:
  - Added colored Lines to show distances between the Launch Site KSC LC-39A (as an example) and its proximities like Railway, Highway, Coastline and Closest City.

Github URL: Module 03-1 Hands-on Lab Interactive Visual Analytics with Folium lab.ipynb

Build a Dashboard with Plotly Dash

- Launch Sites Dropdown List:
  - Added a dropdown list to enable Launch Site selection.
- Pie Chart showing Success Launches (All Sites/Certain Site):
  - Added a pie chart to show the total successful launches count for all sites and the Success vs. Failed counts for the site, if a specific Launch Site was selected.
- Slider of Payload Mass Range:
  - Added a slider to select Payload range.
- Scatter Chart of Payload Mass vs. Success Rate for the different Booster Versions:
  - Added a scatter chart to show the correlation between Payload and Launch Success.

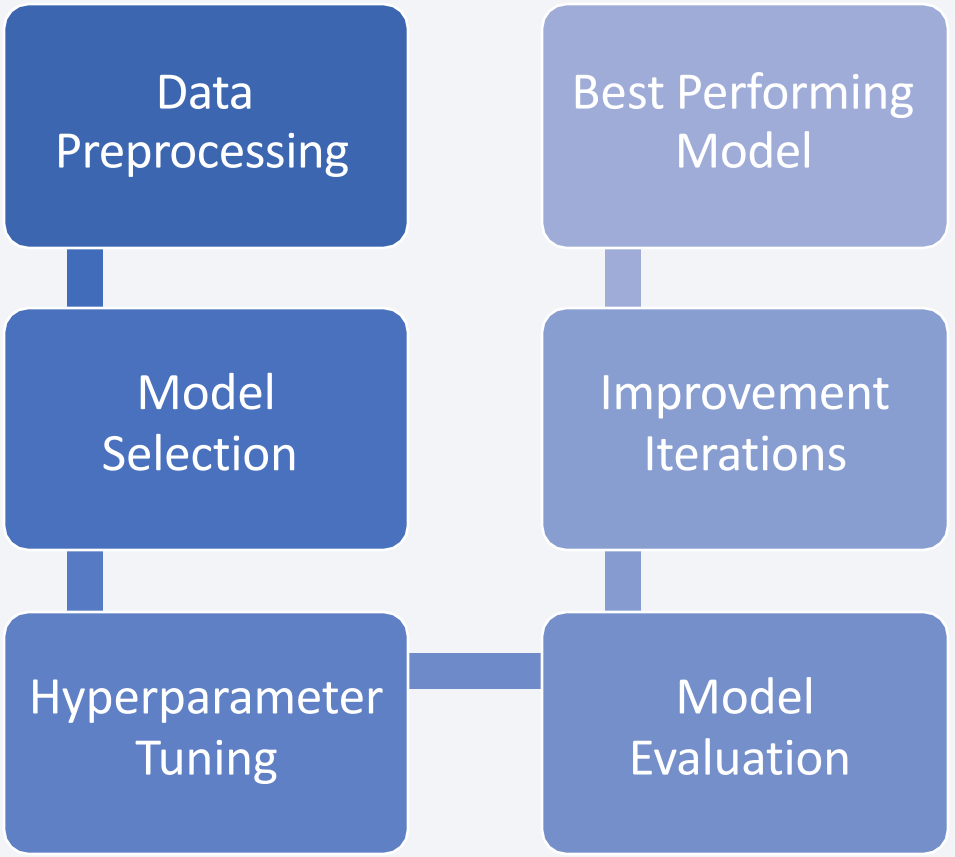
Github URL: Module 03-2 Hands-on Lab- Build an Interactive Dashboard with Ploty Dash.py

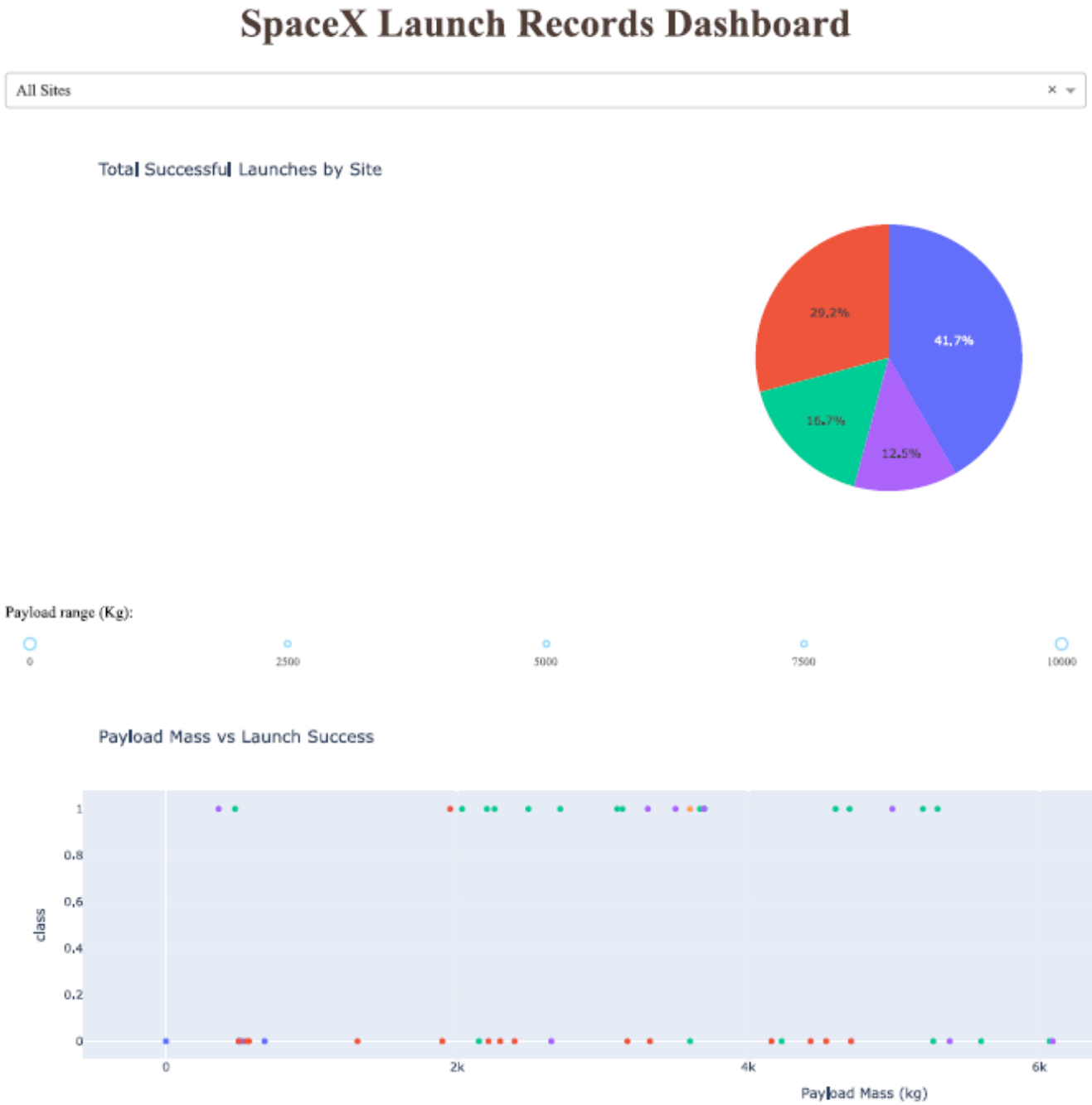
Predictive Analysis (Classification)

1. Data Preprocessing:
   - Standardized features to ensure all variables contribute equally.
   - Split data into training and test sets for model validation.
2. Model Selection:
   - Explored multiple classification algorithms: SVM, Decision Trees, and K-Nearest Neighbors (KNN).
   - Chose algorithms suitable for binary classification tasks based on project requirements.
3. Hyperparameter Tuning:
   - Used GridSearchCV to systematically search for optimal hyperparameters.
   - Tuned parameters such as C (SVM), max_depth (Decision Trees), and n_neighbors (KNN).
4. Model Evaluation:
   - Evaluated models using cross-validation techniques to ensure robustness and generalizability.
   - Utilized metrics like accuracy, precision, recall, and F1-score to assess model performance.
5. Improvement Iterations:
   - Iteratively adjusted models based on insights from validation results.
   - Fine-tuned hyperparameters to maximize predictive accuracy and reliability.
6. Selection of Best Performing Model:
   - Identified the model with the highest accuracy on the test set as the best performer.
   - Considered both training and test set performance to avoid overfitting and ensure real-world applicability.

Github URL: Module 04 SpaceX_Machine Learning Prediction_Part_5.ipynb

Predictive Analysis (Flowchart)

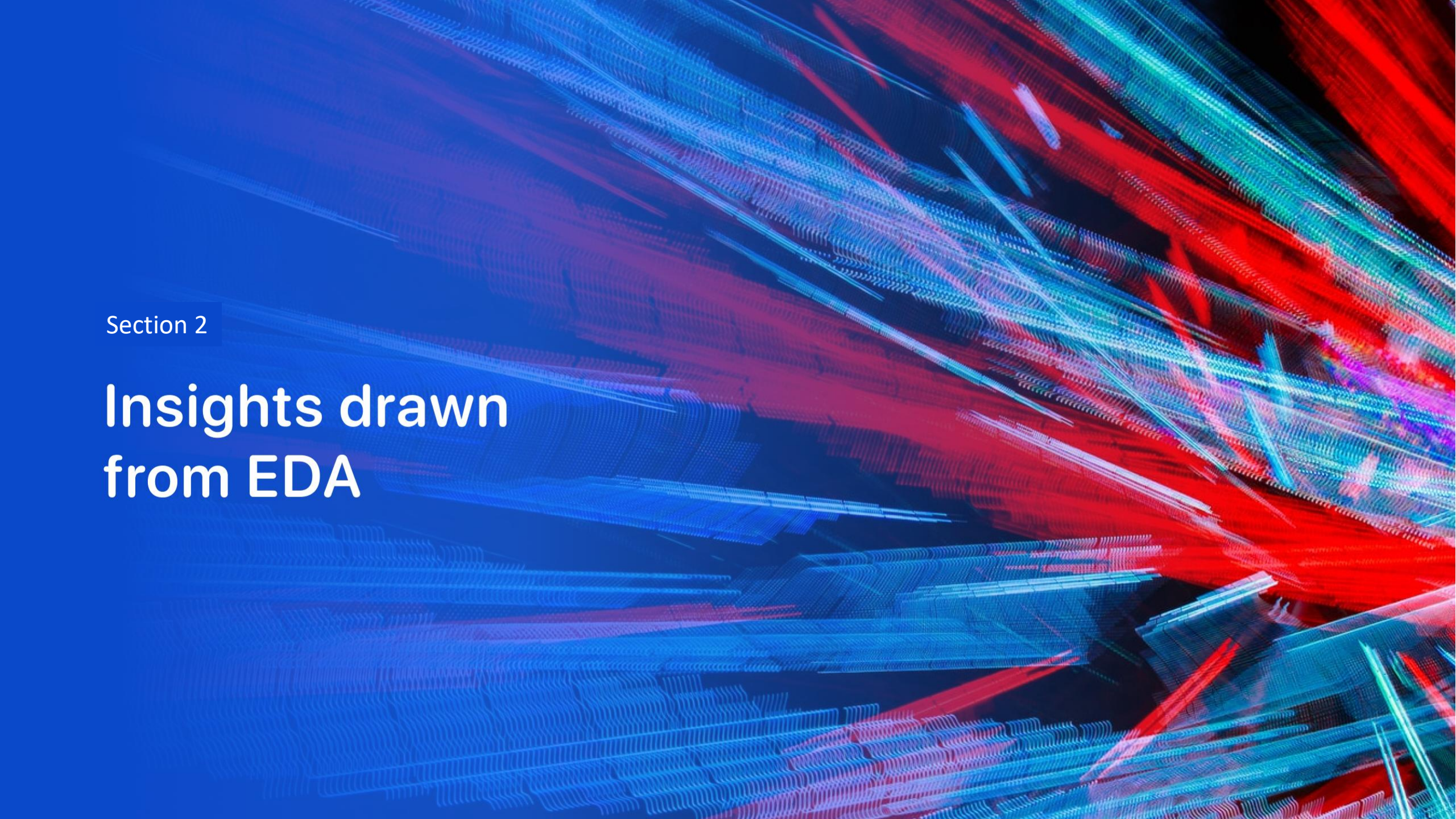




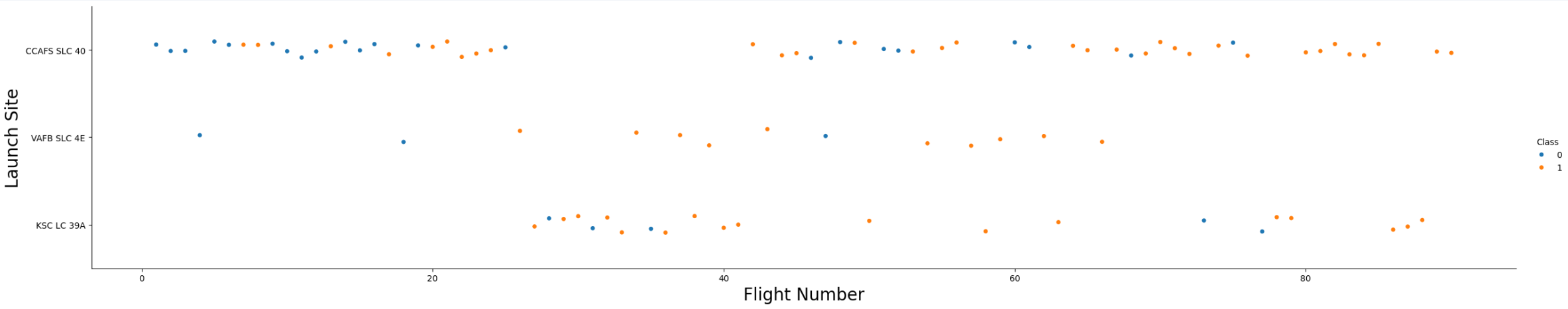
Results

- LR, SVM, KNN are top-performing models for forecasting outcomes in this data.
- Lighter payloads have a higher performance compared to heavier ones.
- The likelihood of a SpaceX launch succeeding increases with the number of years of experience, suggesting a trend towards flawless launches over time.
- Launch Complex 39A at Kennedy Space Center has the highest number of successful launches compared to other launch sites.
- GEO, HEO, SSO, ES L1 orbit types exhibit the highest rates of successful launches.

Github URL:
spacex_launch_records_dashboard.pdf

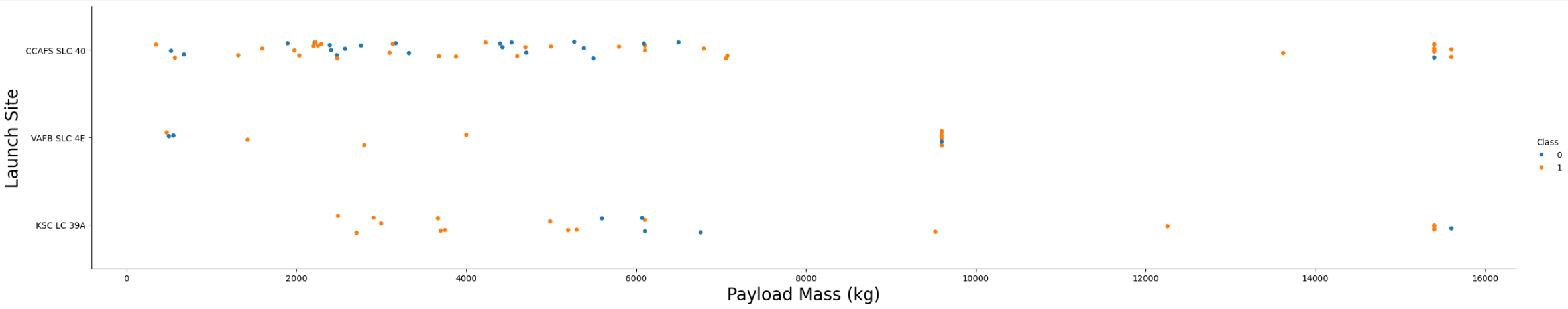

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



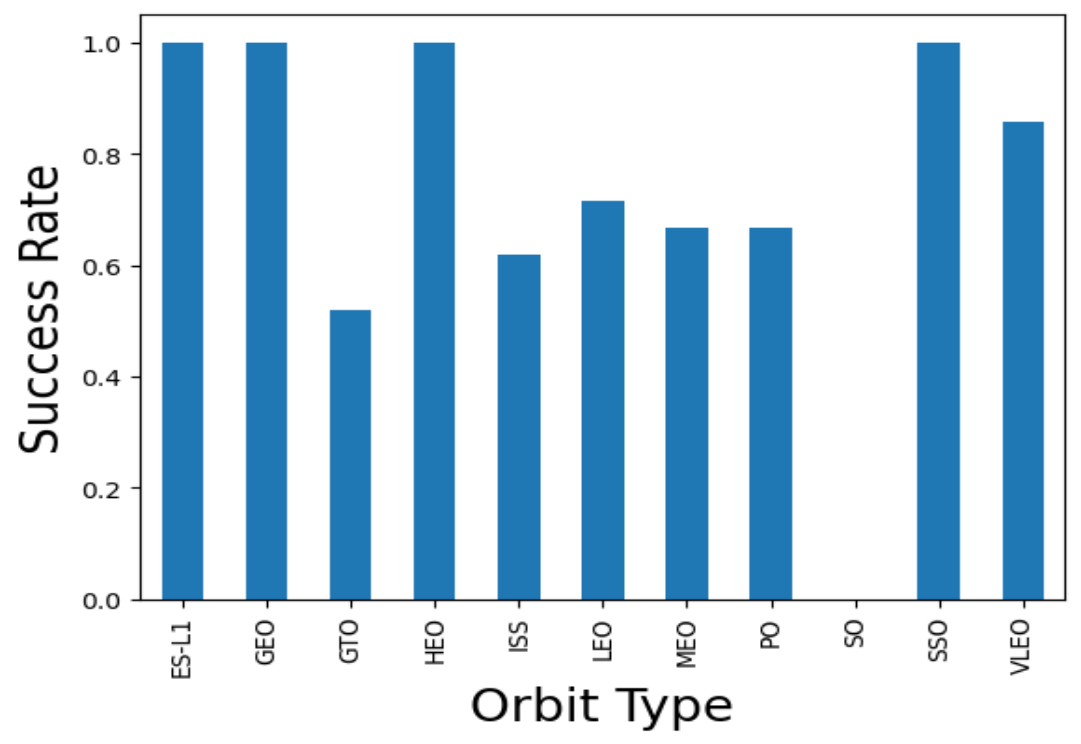
- Total number of launches from launch site CCAFS SLC 40 are significantly higher than the other launch sites.

Payload vs. Launch Site



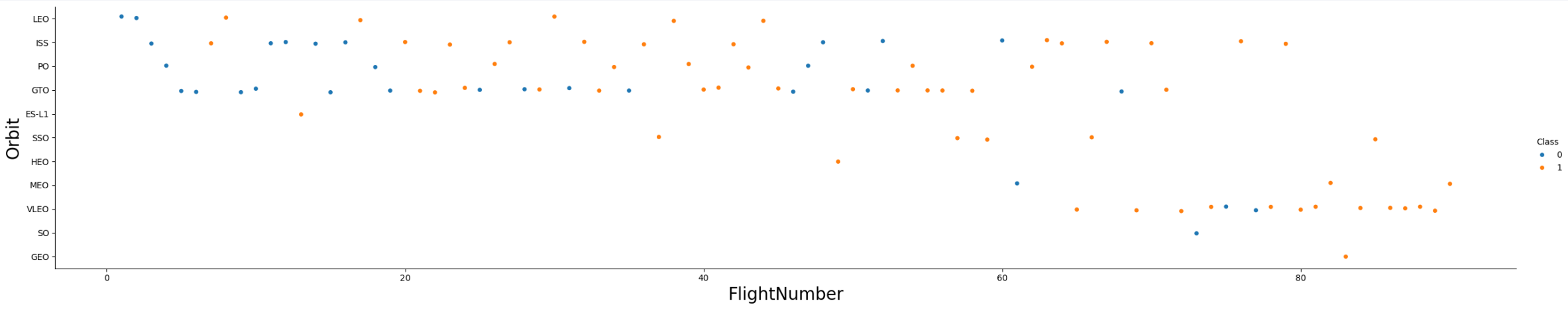
- Payloads with lower mass are have more launches compared to those with higher mass across all three launch sites.

Success Rate vs. Orbit Type



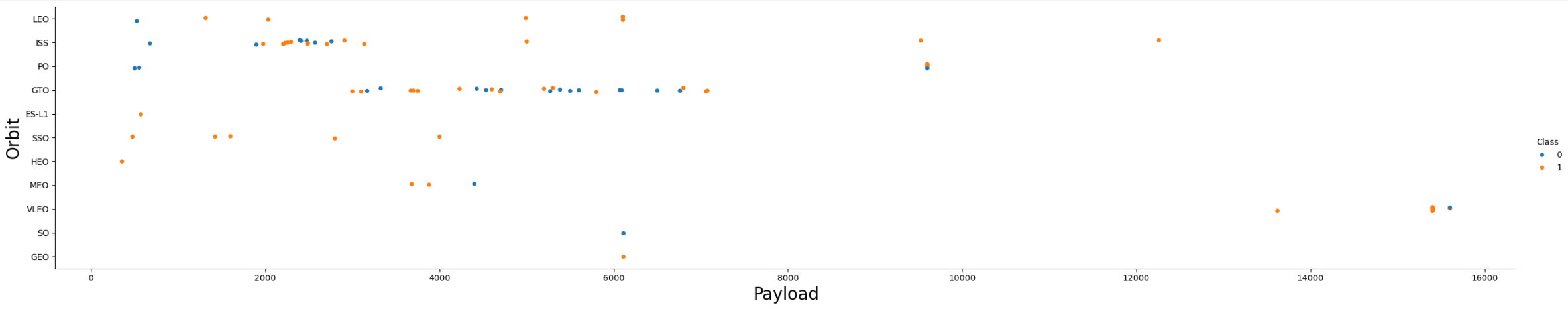
- Orbit types ES-L1, GEO, HEO, SSO have the highest success rate among all.

Flight Number vs. Orbit Type



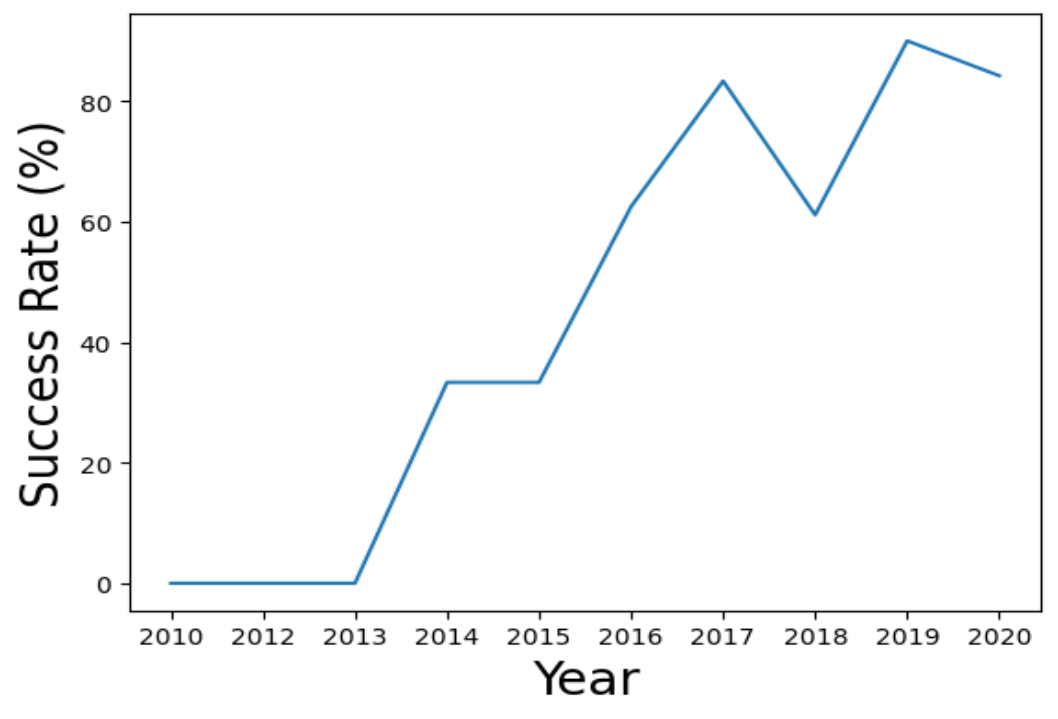
- LEO, ISS, PO, GTO orbits have the most launches in the earlier years, but it slowly shifted to VLEO orbit in the later years.

Payload vs. Orbit Type



- Heavy payloads tend to have higher successful landing rates for PO, LEO, and ISS orbits, but for GTO orbit, success is less predictable with an almost equal mix of successes and failures.

Launch Success Yearly Trend



- The success rate of launches have been increasing since 2013 till 2020, possibly due to technology advancement and experience.

All Launch Site Names

Task 1

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

```
%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

- Performed an SQL query to obtain all launch site names

Launch Site Names Begin with 'CCA'

Task 2

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

- Performed an SQL query to obtain 5 launch site names that begin with ‘CCA’

Total Payload Mass

Task 3

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where customer = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

- Performed an SQL query to obtain the total payload mass carried by boosters launched by NASA (CRS)

Average Payload Mass by F9 v1.1

Task 4

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

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

- Performed an SQL query to calculate the average payload mass carried by booster version F9 v1.1

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

```
%%sql
SELECT min(Date)
FROM SPACEXTBL
WHERE Landing_Outcome = 'Success (ground pad)';
```

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

min(Date)
2015-12-22

- Performed an SQL query to find the dates of the first successful landing outcome on ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000

- Performed an SQL query to list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
In [9]: %sql select booster_version from SPACEXDATASET where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ between 4
        000 and 6000;
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[9]:

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

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [23]:
```
%sql select distinct Mission_Outcome, count(*) from SPACEXTABLE group by Mission_Outcome
```

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

Out[23]:

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

- Performed an SQL query to calculate the total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

- Performed an SQL query to list the names of the booster which have carried the maximum payload mass

```
In [11]: %sql select booster_version from SPACEXDATASET where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXDATASET);
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[11]:

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

- Performed an SQL query to list the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
In [12]: %%sql select monthname(date) as month, date, booster_version, launch_site, landing__outcome from SPACEXDATASET
             where landing__outcome = 'Failure (drone ship)' and year(date)=2015;
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[12]:

MONTH	DATE	booster_version	launch_site	landing__outcome
January	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- Performed an SQL query to 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

```
In [13]: %%sql select landing__outcome, count(*) as count_outcomes from SPACEXDATASET
               where date between '2010-06-04' and '2017-03-20'
               group by landing__outcome
               order by count_outcomes desc;
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

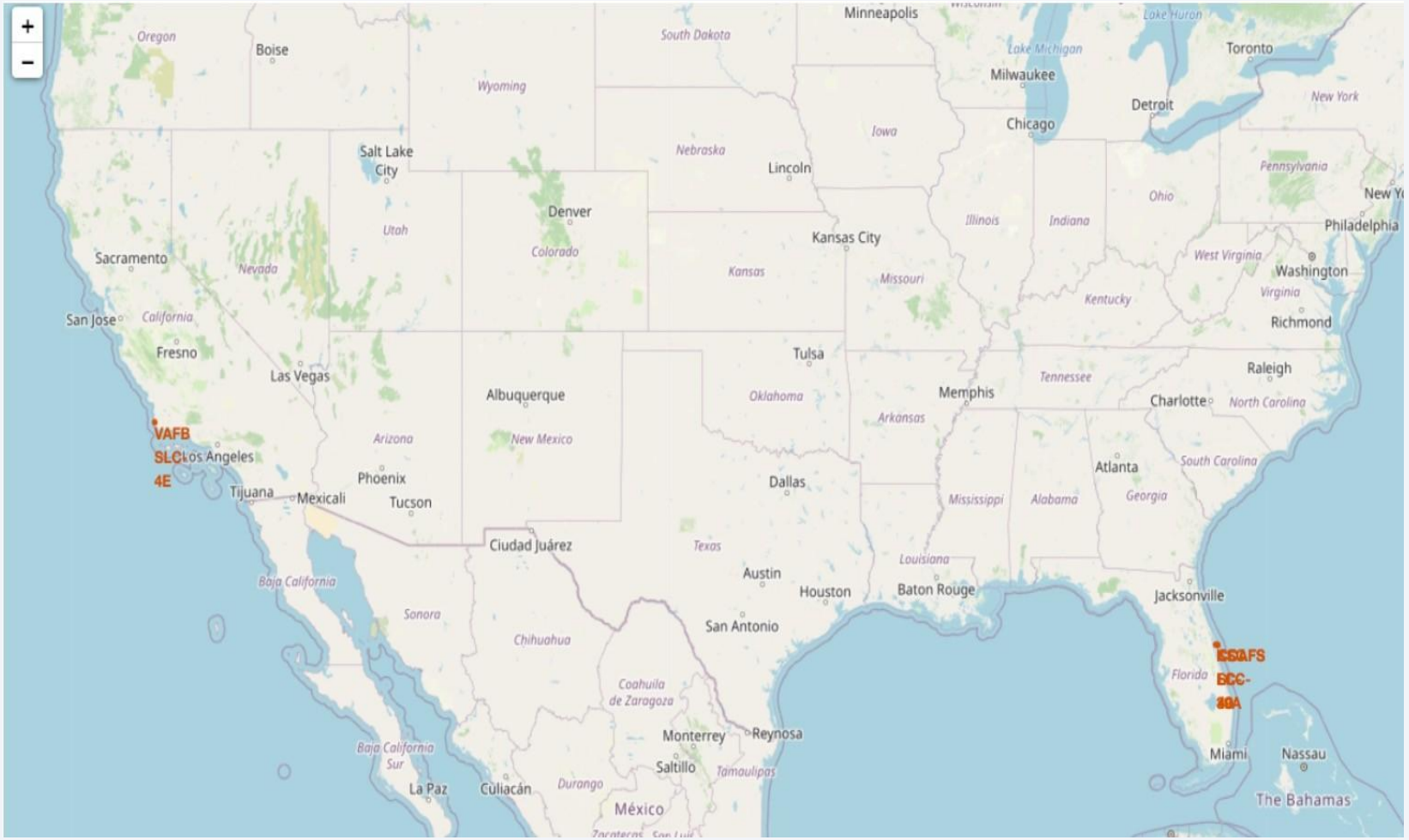
Out[13]:

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

Section 3

Launch Sites Proximities Analysis

All launch sites on a map

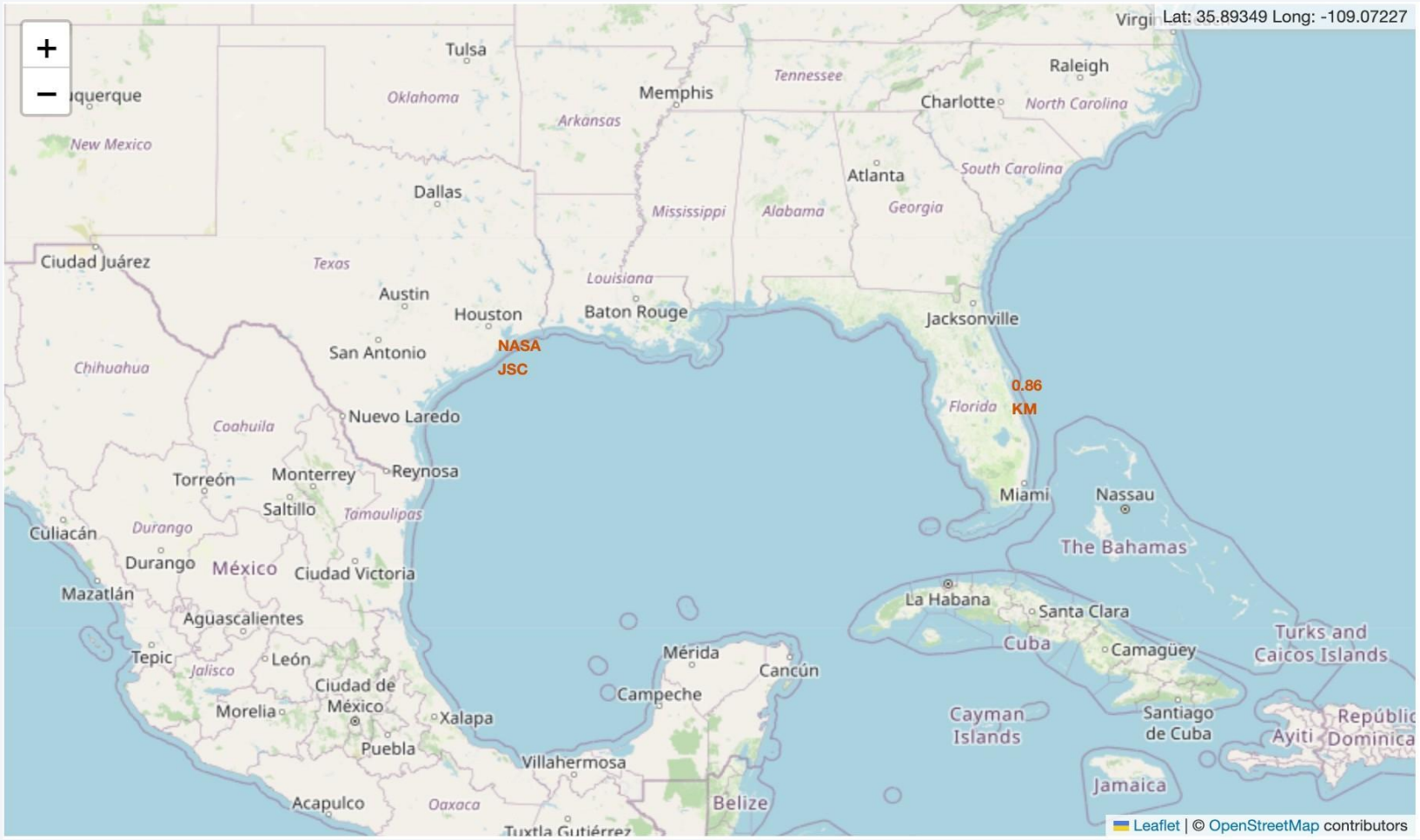


- The launch sites are labelled by a marker with their names on the map.

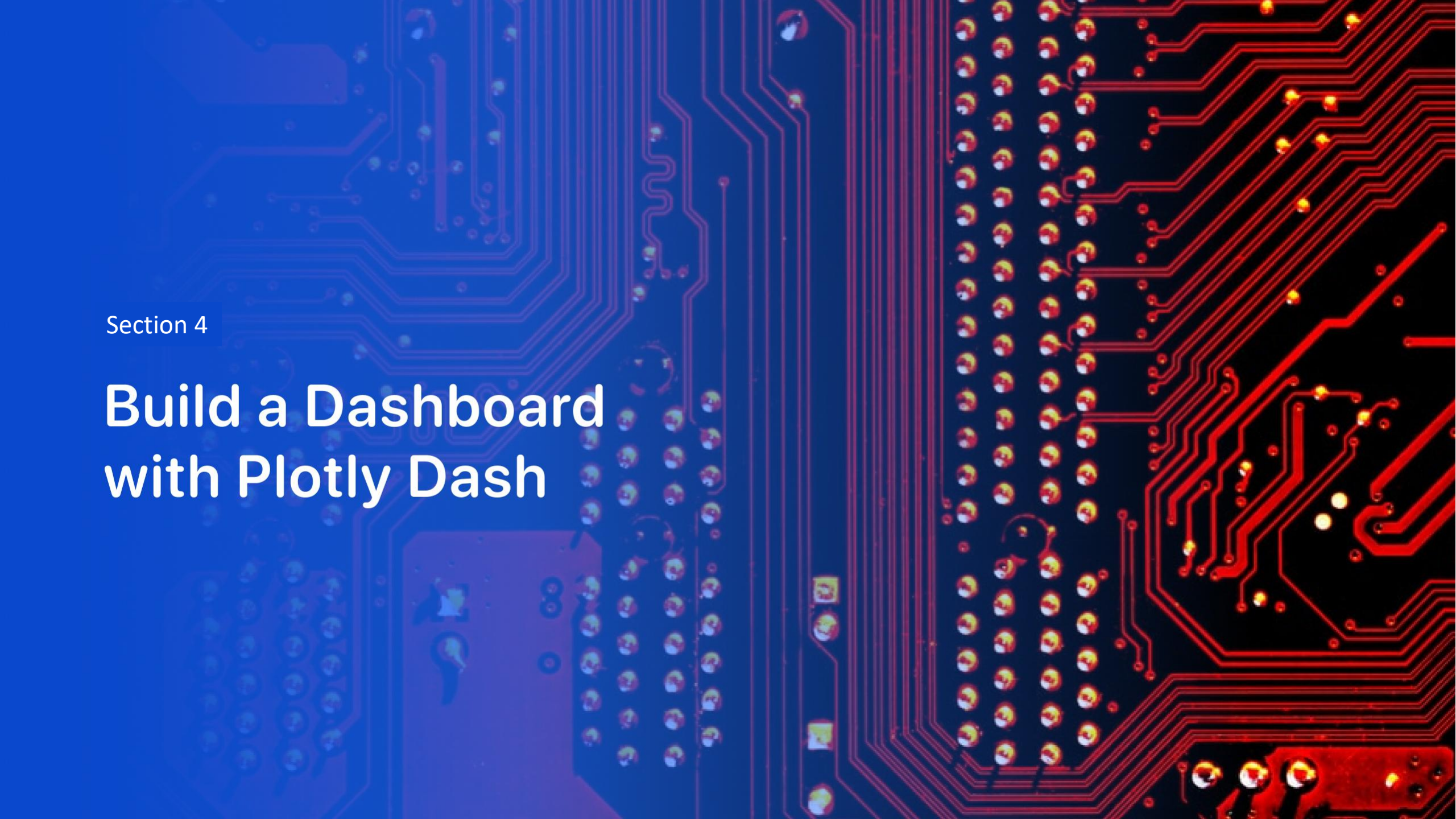
All success/failed launches for each site on the map

- The launch records are grouped in clusters on the map, then labelled by green markers for successful launches, and red markers for unsuccessful ones.

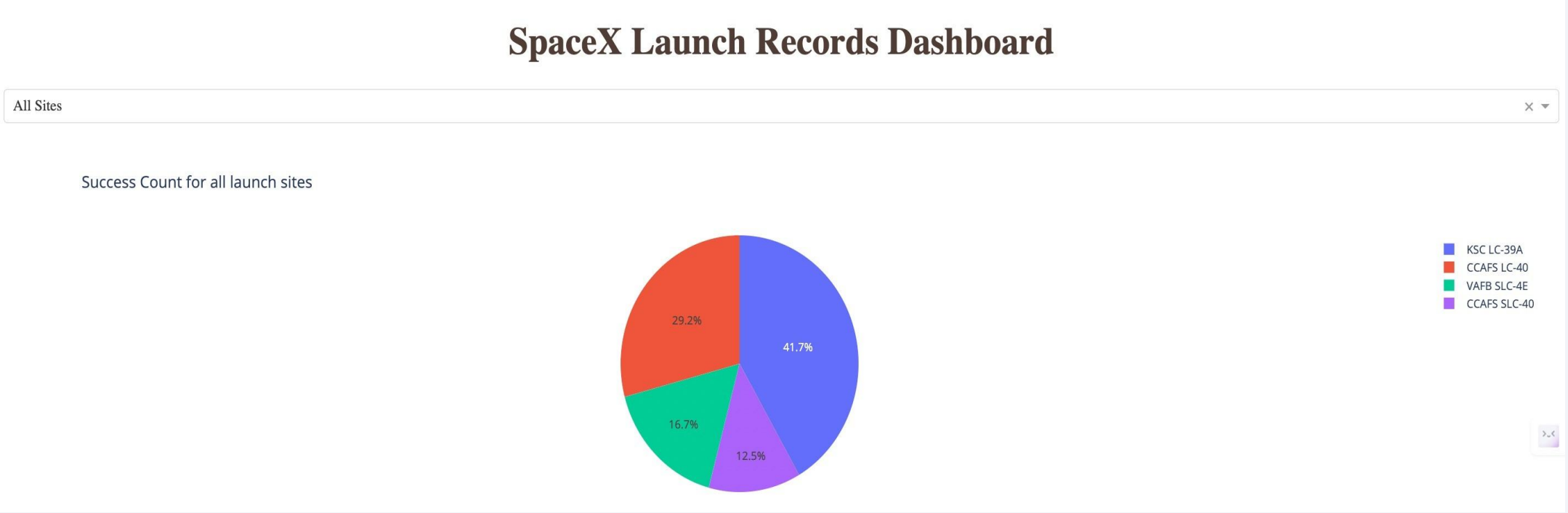
Distances between a launch site to its proximities



- The closest coastline from NASA JSC is marked as a point using MousePosition and the distance between the coastline point and the launch site, which is approximately 0.86 km.

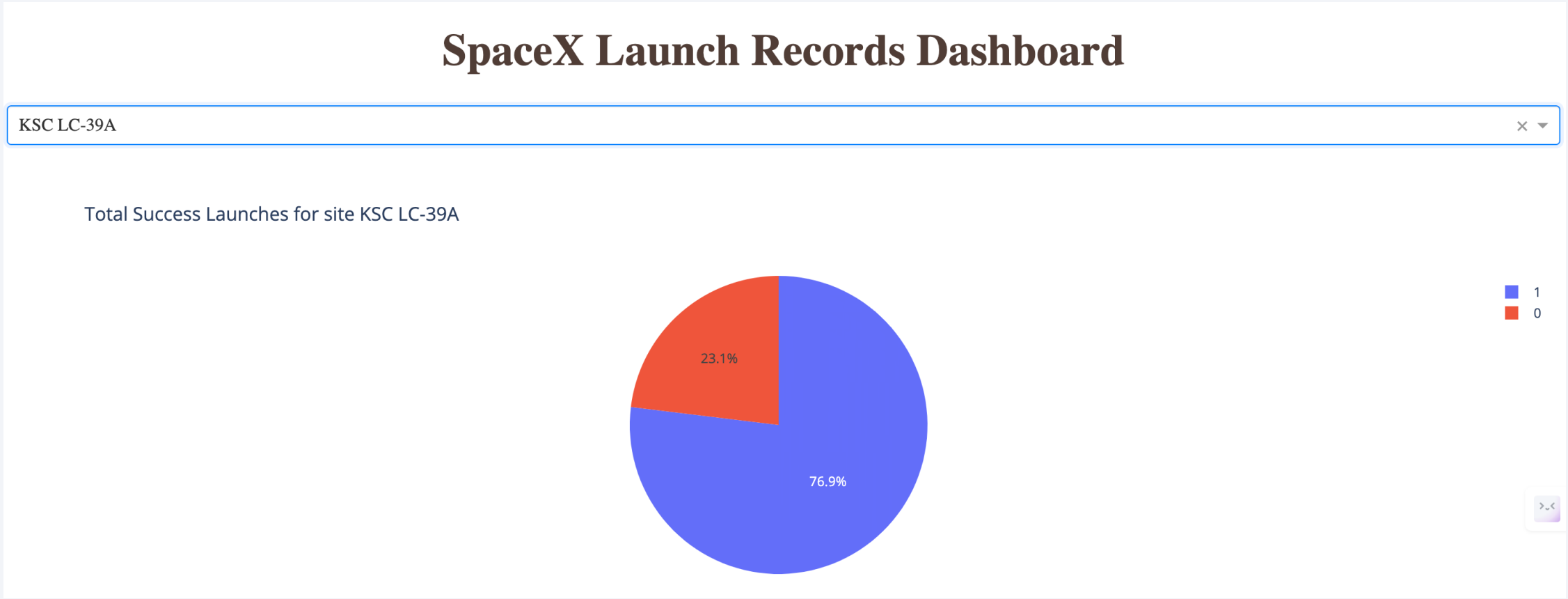

Section 4
Build a Dashboard with Plotly Dash

Total success launches for all sites



- KSC LC-39A has the highest amount of success launches with 41.7% from the entire record, whereas CCAFS SLC-40 has the lowest amount of success launches with only 12.5%.

Success ratio of the launch site with the highest success launches



- KSC LC-39A which is the launch site with highest amount of success, has a 76.9% success rate for the launches from its site, and 23.1% failure rate.

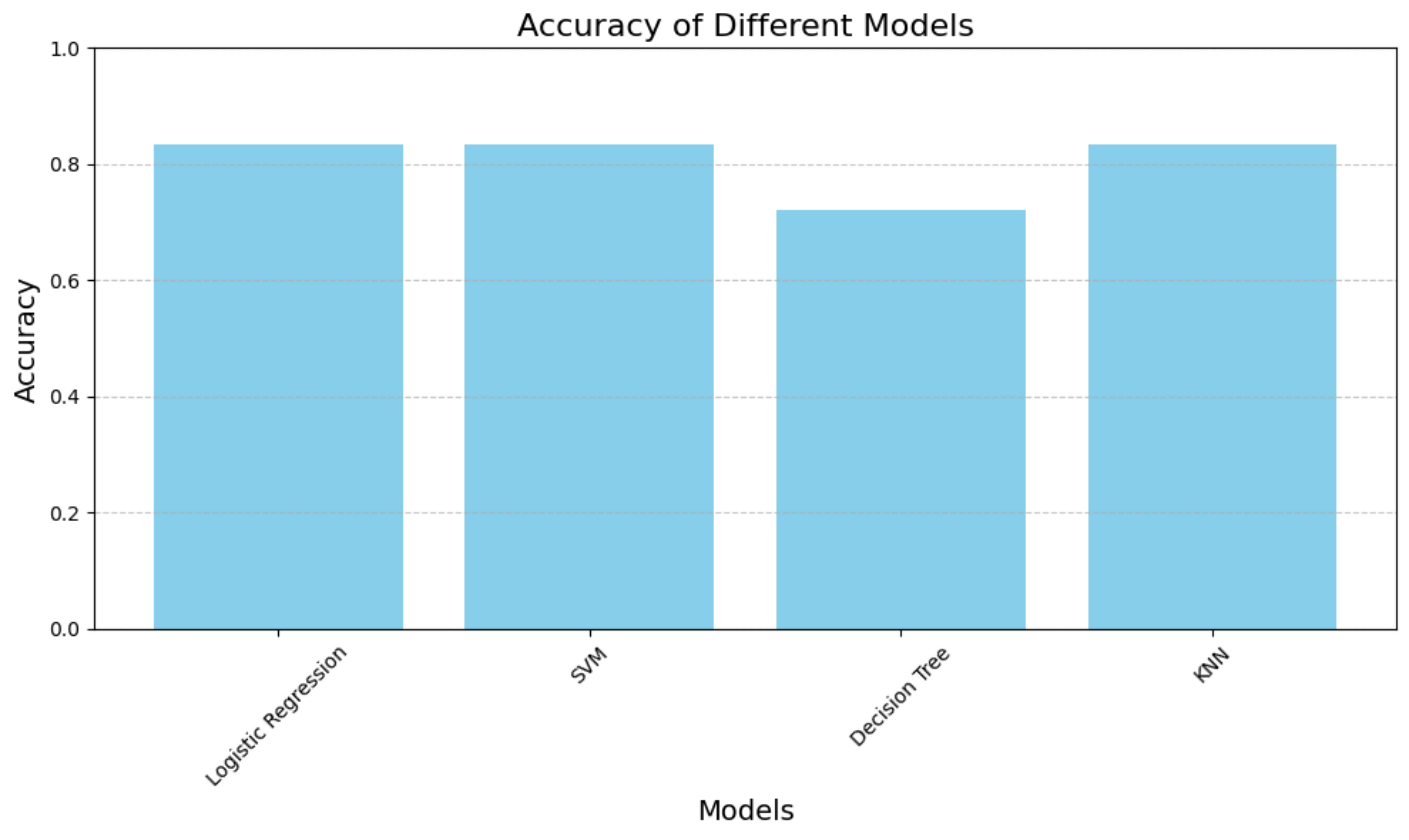
Payload vs. launch outcome

- The payload range that has the highest success launches is between 2,000 to 4,000 kg, which can be seen by the most number of plots in that range, followed by the payload range of 4,000 to 6,000 kg, with the second most number of plots.
- Booster version FT (green spots) has the highest success launches, followed by B4 (purple spots) with the second highest success launches, among all booster versions.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



TASK 12

Find the method performs best:

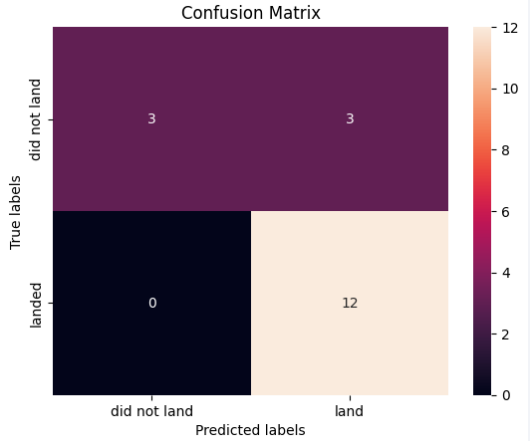
```
print('LR Accuracy:', '{:.2%}'.format(logreg_accuracy))
print( 'SVM Accuracy:', '{:.2%}'.format(svm_accuracy))
print('Decision Tree Accuracy:', '{:.2%}'.format(tree_accuracy))
print('KNN Accuracy:', '{:.2%}'.format(knn_accuracy))
```

```
LR Accuracy: 83.33%
SVM Accuracy: 83.33%
Decision Tree Accuracy: 72.22%
KNN Accuracy: 83.33%
```

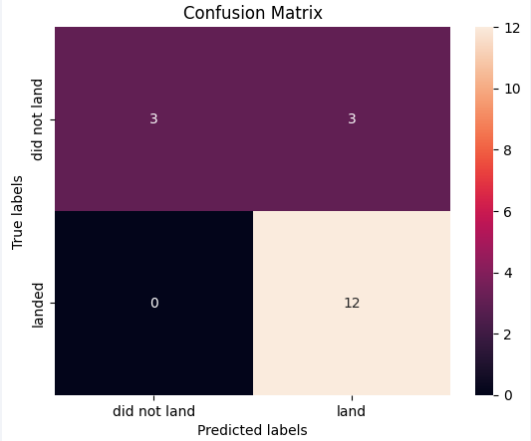
- The model that performed best are LR, SVM, KNN where all 3 achieved the highest accuracy of 83.33%.

Confusion Matrix

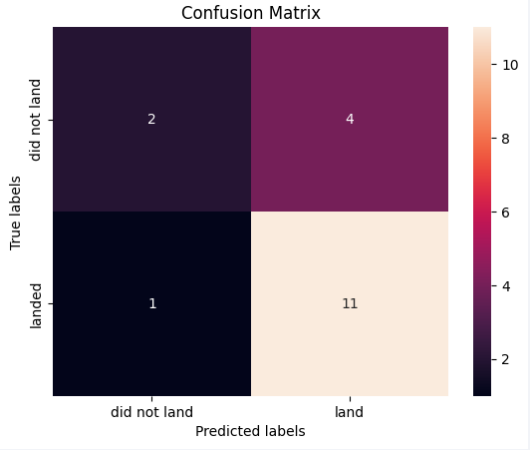
Confusion Matrix of LR



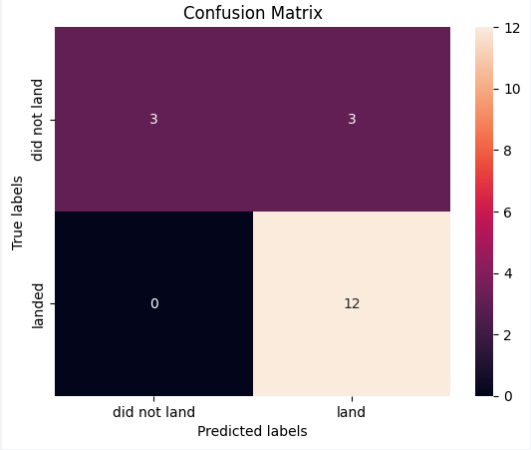
Confusion Matrix of SVM



Confusion Matrix of Decision Tree



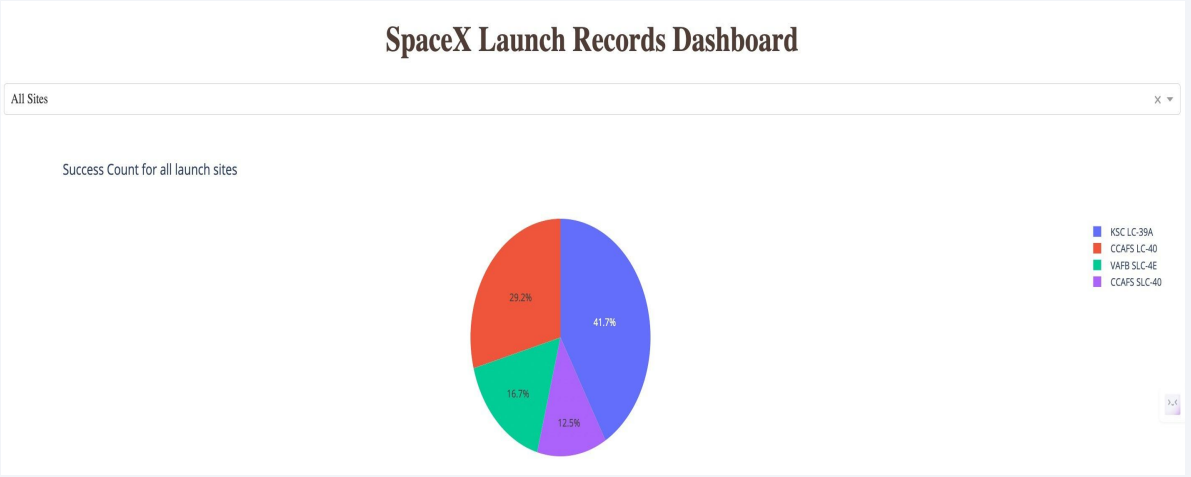
Confusion Matrix of KNN



- LR, SVM, KNN models are good as their confusion matrix show that they predicted all 12 successful landing correctly, with 0 error.
- However, the Decision Tree model only predicted 11 successful landing correctly, with one of them wrongly predicted as a failed / did not land.
- LR, SVM, KNN models have the same accuracy of 83.33% as displayed earlier, hence the same confusion matrix.

Conclusions

- Decision Tree Model is the best algorithm for this dataset.
- Launches with a low payload mass show better results than launches with a larger payload mass.
- Most of launch sites are in proximity to the Equator line and all the sites are in very close proximity to the coast.
- The success rate of launches increases over the years.
- KSC LC-39A has the highest success rate of the launches from all the sites.
- Orbits ESL1, GEO, HEO and SSO have 100% success rate.
- KSC LC-39A has the most successful launches overall



Appendix

Git repository containing all the relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets created during this project:

https://github.com/Reshma11K/Applied-Data-Science-Capstone

Thank you!