IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nathan Staab
September 16, 2025

Outline

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

Executive Summary

This study analyzed Falcon 9 first-stage landing success rates to estimate true launch costs and develop a predictive model using launch-specific features. These insights enable more competitive bidding for NASA and other agency contracts.

SpaceX advertises a $62M launch cost—significantly lower than the $165M charged by other providers—primarily due to its ability to reuse the first stage. Predicting landing outcomes allows for more accurate cost forecasting.

From June 2010 to November 2020 , Falcon 9 achieved a X% landing success rate across 90 launches, resulting in an average cost of $X per launch. To compete with SpaceX, providers must offer launches below $XM. If current trends continue, the average cost is projected to reach $X in one year and $X in two years, further narrowing the competitive price point toward $62M.

Introduction

- Assess Falcon 9 first-stage landing success rates to estimate true launch costs and build a predictive model using launch-specific features.
- SpaceX's reusable rocket technology offers significant cost advantages, advertising launches at $62M versus $165M from traditional providers.
- Understanding and forecasting landing outcomes enables more accurate cost modeling—critical for placing competitive bids on NASA and other agency contracts.
- Analysis spans 90 launches from 2010 to 2020, focusing on landing outcomes, cost implications, and future projections.

Section 1

Methodology

Methodology

Executive Summary

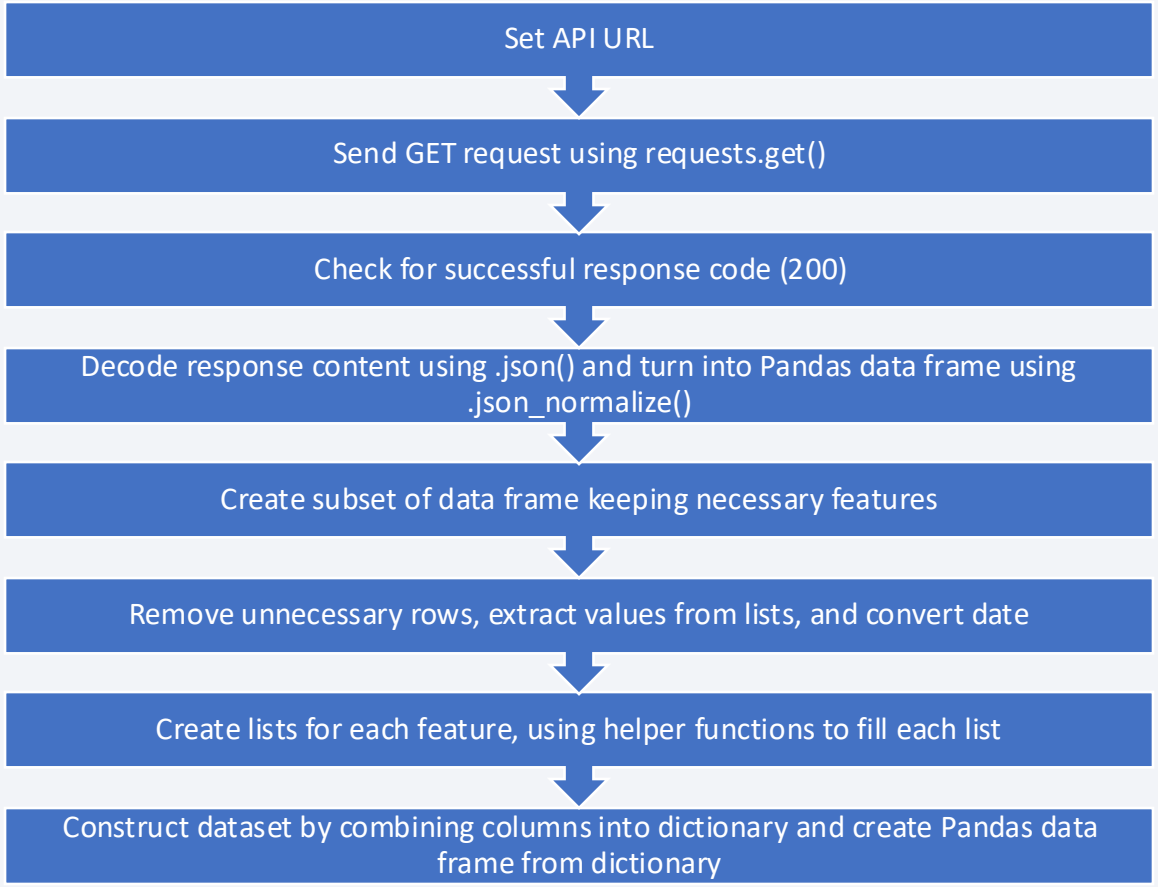
- Data collection:
  - Falcon 9 launch records were extracted from HTML tables on Wikipedia using web scraping and from the SpaceX API using GET requests.
  - HTML tables and JSON files were parsed to a Pandas data frame.
- Data wrangling
  - Payload Mass was imputed using the mean for 5 of 90 launches missing this feature.
  - Created landing outcome label to determine success rate and for predictive modeling.
- Exploratory data analysis (EDA) using visualization and SQL
- Interactive visual analytics using Folium and Plotly Dash
- Predictive analysis using classification models
  - Data was transformed using Standard Scaler to prevent bias and improve model performance.
  - To identify the most accurate model for predicting the successful landing of the first stage, Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K-Nearest Neighbor algorithms were evaluated through hyperparameter tuning and cross-validation.

Data Collection

- Data was collected from the SpaceX API using HTTP GET requests, facilitated by the *requests* library in a Python notebook. This library enabled seamless interaction with the API endpoints.
- To streamline the process, helper functions were implemented to extract specific information from various API endpoints using unique identification numbers. The initial JSON responses were parsed and normalized into Pandas data frames for easier manipulation and analysis.
- Subsequent API calls were made to retrieve detailed launch data using launch IDs. The responses from these requests were stored in lists, and additional helper functions were used to extract relevant variables from each API response.
- Finally, the collected data was organized by combining the list variables into a dictionary structure. This dictionary was then converted into a comprehensive Pandas data frame, forming the final dataset ready for analysis.

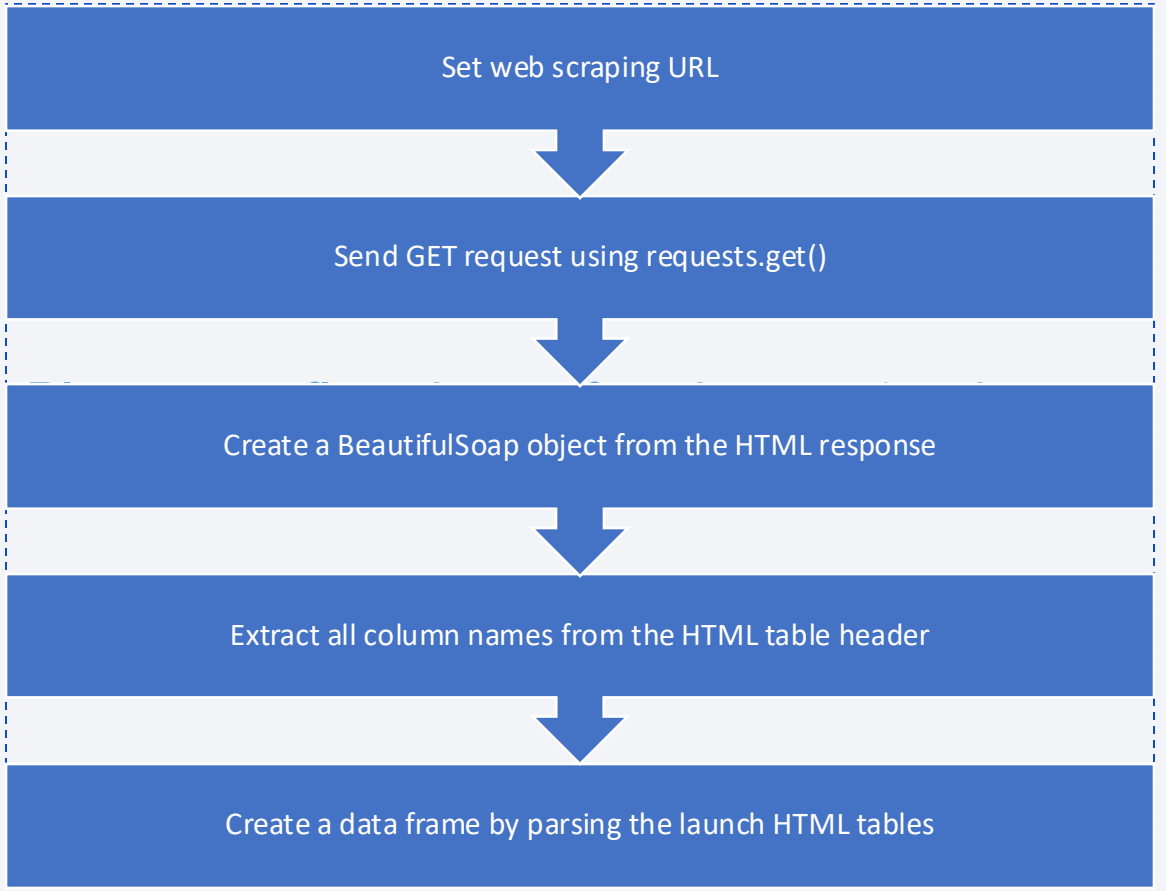
Data Collection – SpaceX API

- Data was collected from the SpaceX API using HTTP GET requests.
- GitHub URL to SpaceX API notebook: https://github.com/njstaab/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api-v2.ipynb



Data Collection – Scraping

- Web scraping was used to collected data on Falcon 9 launch records from a Wikipedia paged titled “List of Falcon 9 and Falcon Heavy launches” located at the following URL: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- GitHub URL to SpaceX web scraping notebook: https://github.com/njstaab/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Dataset was reviewed for missing values and to understand the features and data types available for analysis.
- Feature exploration within the dataset revealed different landing locations such as ocean, ground pads, and drone ships.
- For further analysis and to enable predictive modeling, landing outcomes were consolidated to a landing outcome label column with “1” representing successful landings and “0” representing unsuccessful landings.
- GitHub URL for data wrangling notebook: https://github.com/njstaab/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling-v2.ipynb

EDA with Data Visualization

- A number of charts (included in Section 2) were plotted to provide additional insights on landing outcomes over time and how various features may have influenced landing outcomes.
- These charts include Pay Load Mass vs Flight Number, Flight Number vs Launch Site, Payload vs Launch Site, Success Rate by Orbit, Flight Number vs Orbit, Payload vs Orbit, and Launch Success Rate by Year.
- GitHub URL for EDA with data visualization notebook: https://github.com/njstaab/Applied-Data-Science-Capstone/blob/main/jupyter_labs_eda_dataviz_v2.ipynb

EDA with SQL

- SQL queries were executed against a database containing the dataset to answer additional questions, including: unique launch site names, total payload mass carried by the boosters, average payload mass by booster version, date of the first successful landing on a ground pad, the names of boosters that landed successfully and had a payload >= 4,000 but less than 6,000, total number of successful and failed landings, which boosters have carried the max payload, month names of records, and rank landing outcomes.
- GitHub URL for EDA with SQL notebook: https://github.com/njstaab/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- An interactive map was created to better understand launch success rate based on launch site location. The interactive map includes objects such as markers to illustrate the number of successful and failed launches at each site in an effort to find optimal launch site locations.
- GitHub URL for interactive map: https://github.com/njstaab/Applied-Data-Science-Capstone/blob/main/lab-jupyter-launch-site-location-v2.ipynb

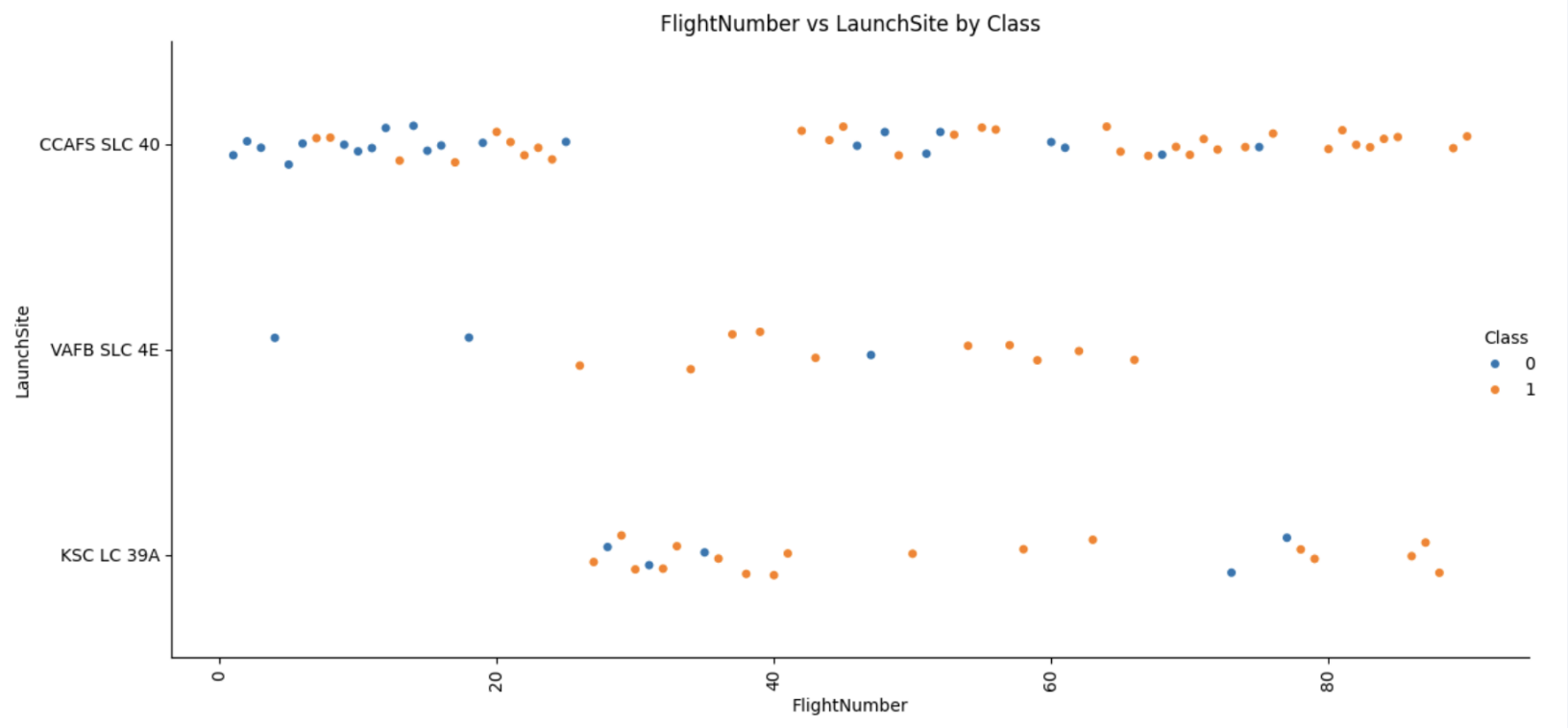
Predictive Analysis (Classification)

- Preprocess techniques such as One-Hot Encoding and StandardScaler were used to transform categorical variables to numerical columns and normalize values to ensure each feature contributes equally to the model.
- The dataset was then split 80/20 into training and test subsets using the train_test_split function.
- Several predictive classification models were created and compared for accuracy including: logistic regression, supper vector machine, decision tree classifier, and k nearest neighbors.
- GridSearchCV was was used to test different hyperparameter combinations using cross-validation (cv=10) to find the best model settings with the most reliable performance estimate.
- GitHub URL for predictive analysis notebook: https://github.com/njstaab/Applied-Data-Science-Capstone/blob/main/SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb

Section 2

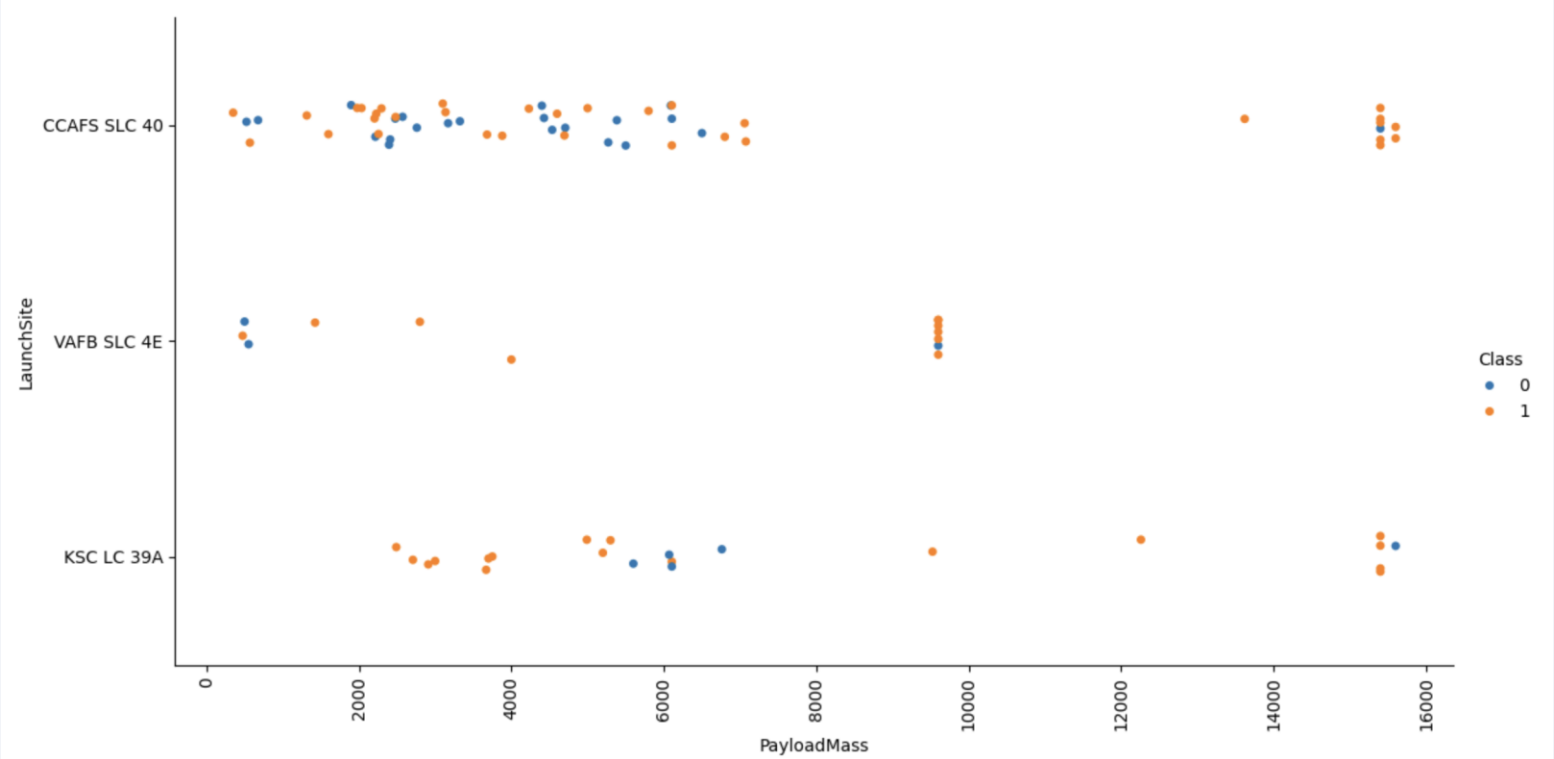
Insights drawn from EDA

Flight Number vs. Launch Site



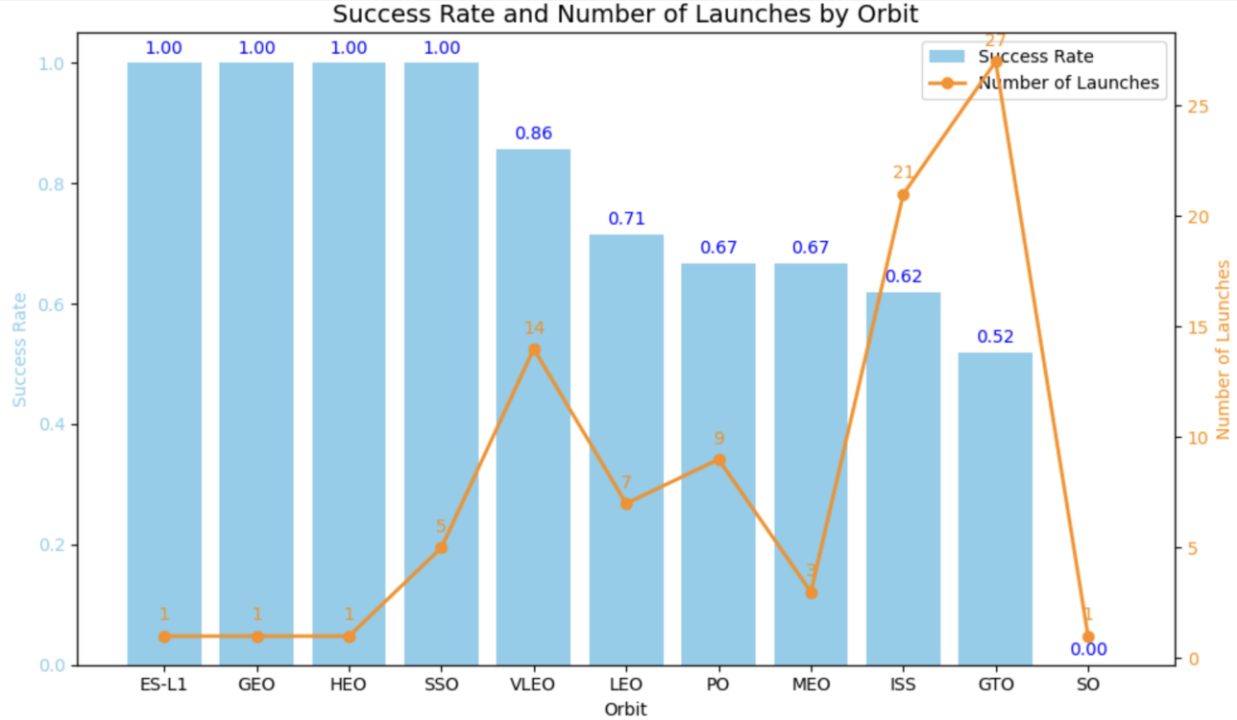
- CCAFS had the most launches (55) and lowest rate of success (60%).
- However, success rate for the last 20 flights at CCAFS was 85%, compared to 35% for the first 20 flights.
- The KSC location had the highest rate of success at 77.3%, but only marginally higher than the VAFB site at 76.9%.

Payload vs. Launch Site



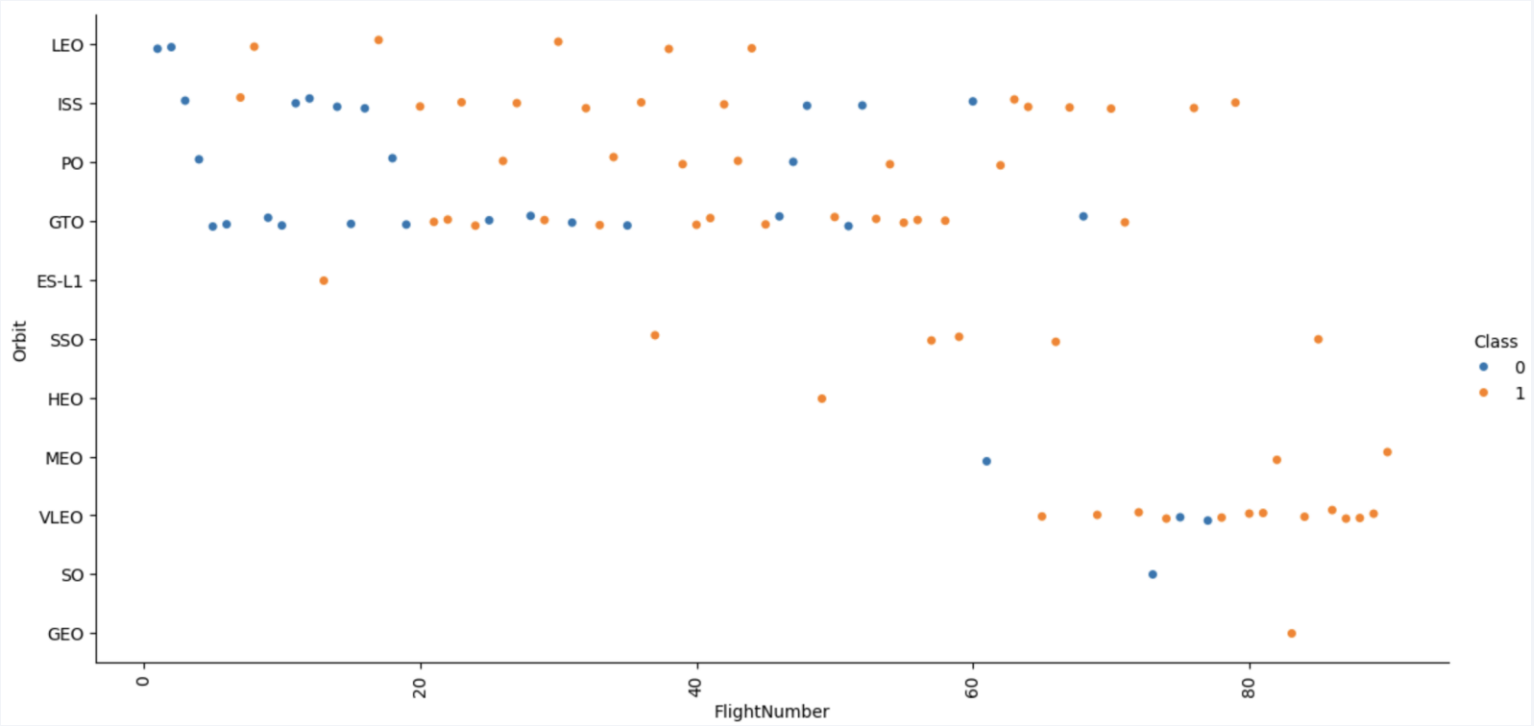
- The VAFB launch site had no payloads over 10,000.
- Most payloads (67) were under 8,000 with 23 payloads above 8,000.
- Larger payloads (above 8,000) had a higher success rate (87%) compared to payloads below 8,000 (60%).

Success Rate vs. Orbit Type



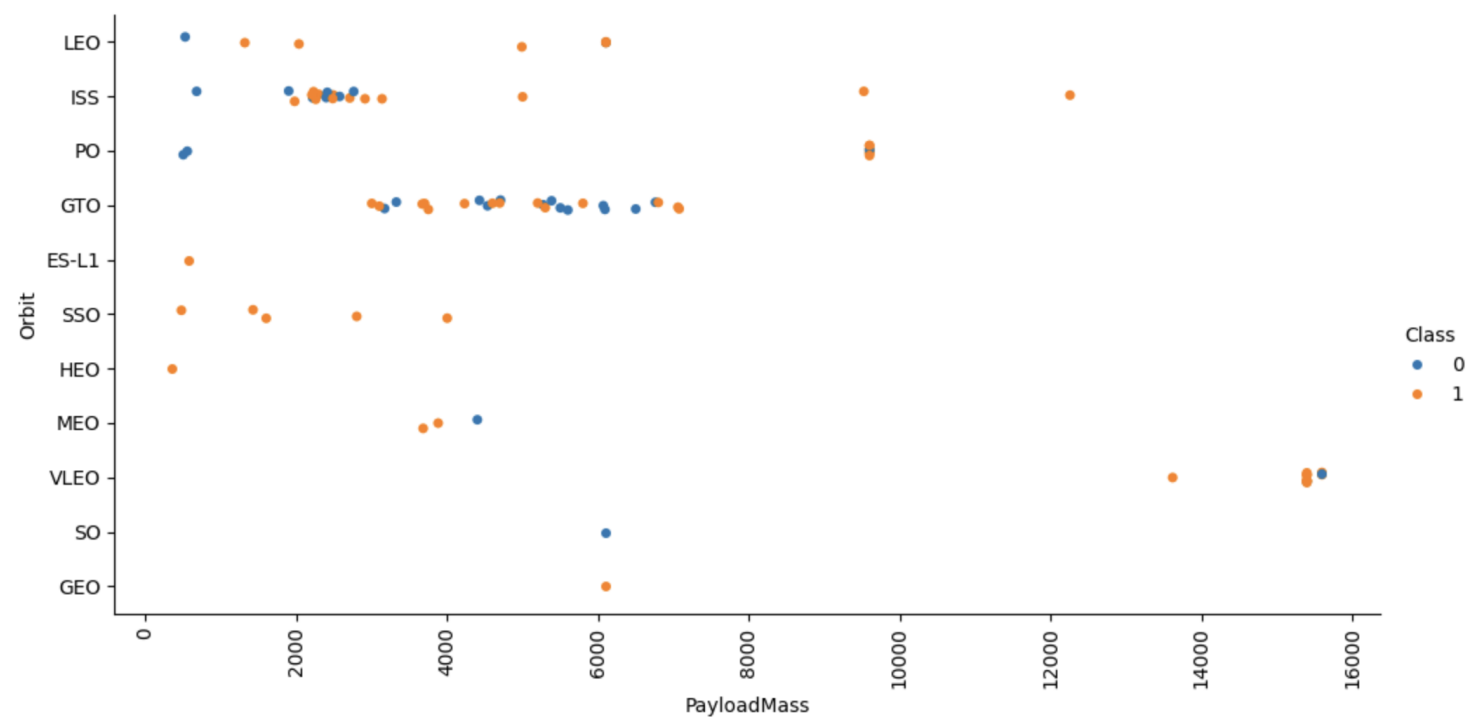
- The ES-L1, GEO, HEO, and SSO orbits had a landing success rate of 100%, however these orbits accounted for a small number of launches.
- For orbits with at least 5 launches, SSO had the highest landing success rate (100%) and GTO orbits had the lowest success rate (52%).

Flight Number vs. Orbit Type



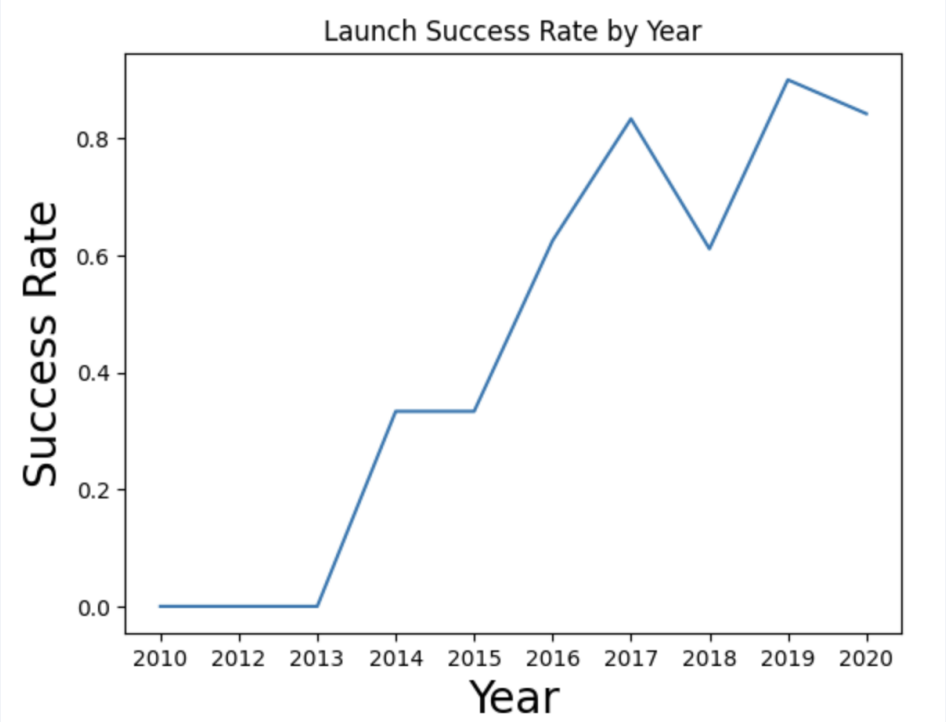
- Many of initial launches (first 20) were unsuccessful, resulting in low success rates for these orbit types.
- Some orbits had 100% success rate but are very small samples (i.e., HEO, SSO, GEO).
- To account for small sample sizes and recent improvements in landing success, VLEO orbits had the highest success rate (86%) over the last 45 flights (second half of the dataset) among orbits with at least 5 records.

Payload vs. Orbit Type



- Most launches had payloads less than 8,000.
- For LEO, ISS, and PO, heavier payloads had more successful landings.
- For GTO, payload did not significantly influence landing success.
- ES-L1, SSO, HEO, and GEO orbits had a 100% success rate, but these figures carry less weight due to small sample sizes.

Launch Success Yearly Trend



- Success rate experienced a slight dip in 2020 and a more notable dip in 2018, but is overall upward trending.

All Launch Site Names

```
1 # Display the names of the unique lanuch sites in the space mission using SQL queries
2 %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

- Four unique launch sites were identified in the dataset containing historical launch records using the above SQL query.
- The `%sql` *magic command* was used since the query was executed from a Python notebook. `SELECT DISTINCT` instructs the query to only return unique values from the `Launch_Site` column in the `SPACAEXTABLE` table.

Launch Site Names Begin with 'CCA'

```
1 # Display 5 records where lanuch sites begin with the string 'CCA' using SQL queries
2 %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
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (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 (ISS)	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 attempt
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

- The five records for launch sites containing “CAA” were identified using the above SQL query. The query includes a wildcard (%) to capture all launch sites starting with “CAA”. `SELECT *` instructs the query to return all columns and `LIMIT 5` instructs the query to only return the first 5 rows that meet the `WHERE` and `LIKE` conditions.

Total Payload Mass

```
# Display the total payload mass carried by boosters launched by NASA (CRS) using SQL queries
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
SUM(PAYLOAD_MASS__KG_)
45596
```

- Total payload mass was 45,596 kg for launches where `NASA (CRS)` was the customer.
- The query calculates the sum of all values in the `PAYLOAD_MASS_KG` column of the dataset in the `SPACEXTABLE` table. Furthermore, the query uses the `WHERE` condition to only return the payload sum for records where customer name is equal to `NASA (CRS)`.

Average Payload Mass by F9 v1.1

```
1 # Display average payload maass carried by booster version F9 v1.1 using SQL queries
2 %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

- Average payload mass for launches with the F9 v1.1 booster was 2,928.4 kg.
- The query calculates the average value of the `PAYLOAD_MASS_KG` column in the `SPACEXTABLE` table in the database where the `Booster_Version` column is equal to “F9 v1.1”.

First Successful Ground Landing Date

```
1 # List the date when the first successful landing outcome in ground pad was acheieved using SQL queries
2 %sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)';
```

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

MIN(Date)
2015-12-22

- The date of the first successful landing on a ground pad was December 22, 2015.
- The query utilizes the `MIN` condition to find the lowest/oldest date value in the `Date` column of the `SPACEXTABLE` table and the `WHERE` condition to only include rows with the desired landing outcome.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
1 # List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000 using SQL queries
2 %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 60
```

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

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

- The above query lists the 4 boosters with successful drone ship landings and payloads between 4,000 and 6,000 kg.
- `SELECT` specifies the column values to return, `FROM` specifies the table name, `WHERE` specifies the desired landing type, `AND` filters the results to the desired weight conditions (between 4,000 and 6,000).

Total Number of Successful and Failure Mission Outcomes

```
# List the total number of successful and failure mission outcomes using SQL queries
%sql SELECT Mission_Outcome, COUNT(*) FROM SPACEXTABLE GROUP BY Mission_Outcome;
```

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

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

- The above query lists the mission outcomes with a very high success rate. 100 of the 101 missions were considered successful.
- `SELECT` specifies the column, `COUNT(*)` instructs the query to count all records, `GROUP` instructs the query to group the counts by mission outcome category.

Boosters Carried Maximum Payload

```
1 # Display Booster_Version and Payload_Mass_kg where Payload_Mass_kg is equal to 15,600
2 %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = 15600 ORDER BY Booster_Version ASC;
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

- The above SQL query lists the boosters that carried the max payload (15,600 kg).
- The `WHERE` condition instructs the query to return a list of boosters where `PAYLOAD_MASS_KG` is equal to 15,600 (max payload was identified using the `MAX` function in a separate query) and orders the list alphabetically by booster.

2015 Launch Records

```
1 # List records with month name, failure landing_outcomes in drone ship, booster versions, launch_site for the months in 2015 using SQL queries
2 %%sql SELECT
3   CASE substr(Date, 6,2)
4     WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'March' WHEN '04' THEN 'April'
5     WHEN '05' THEN 'May' WHEN '06' THEN 'June' WHEN '07' THEN 'July' WHEN '08' THEN 'August' WHEN '09' THEN 'September' WHEN '10'
6     THEN 'October' WHEN '11' THEN 'November' WHEN '12' THEN 'December'
7   END AS Month, Landing_Outcome, Booster_Version, Launch_Site
8 FROM SPACEXTABLE
9 WHERE Landing_Outcome = 'Failure (drone ship)'
10 AND substr(Date,0,5) = '2015';
```

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

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

- The above SQL query lists the failed drone ship landing records, booster version, and launch site for year 2025.
- The `CASE` function takes s substring of the value in the `DATE` column. It starts at character position 6 and takes 2 characters. The result of the substring is used to determine the appropriate month name (ex: 01 = January, 02 = February, etc.). The `END` function renames the column as `MONTH`. `WHERE` instructs the query to only return records for failed drone ship landings. The `AND` function instructs the query to only include records that contain the year 2015.

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

```
# Rank the count of landing outcomes between date 2010-06-04 and 2017-03-02 in descending order using SQL queries
%%sql SELECT
  Landing_Outcome,
  COUNT(*) AS Count
FROM SPACEXTABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY Count DESC;
```

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

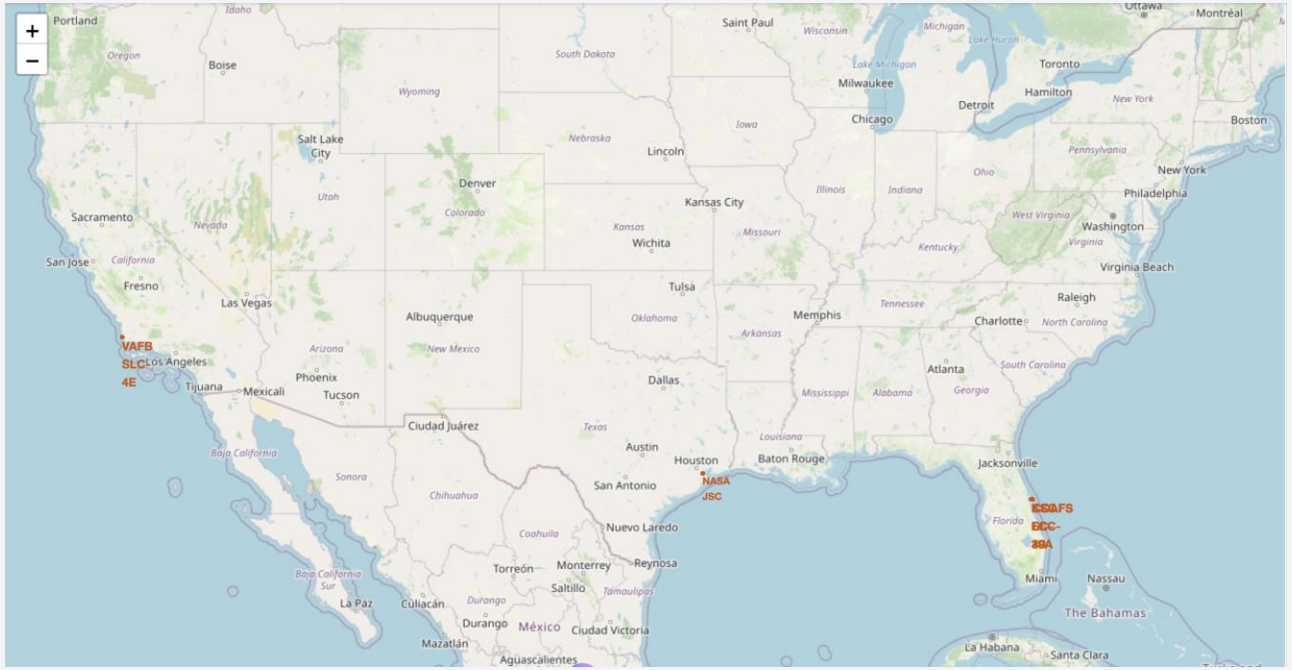
Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

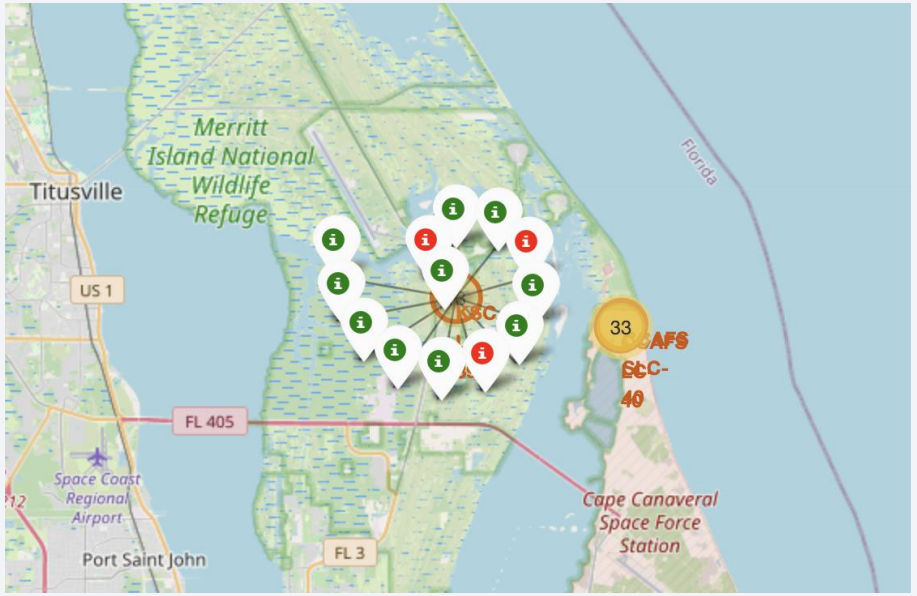
- The above query ranks landing outcomes by number of records for each category between the specified dates. The query indicates "No attempt" was the most frequent landing outcome, followed by successful drone ship landings, followed by failed drone ship landings, etc.
- `%%sql` is used to separate to break the query into multiple lines (so it's not cutoff in the code cell). The `COUNT` function instructs the query to count the number of records for each landing outcome. `WHERE` instructs the query to only count records for the specified dates. `GROUP BY` and `ORDER BY` group the results by landing outcome and sort by count in descending order.

Section 3

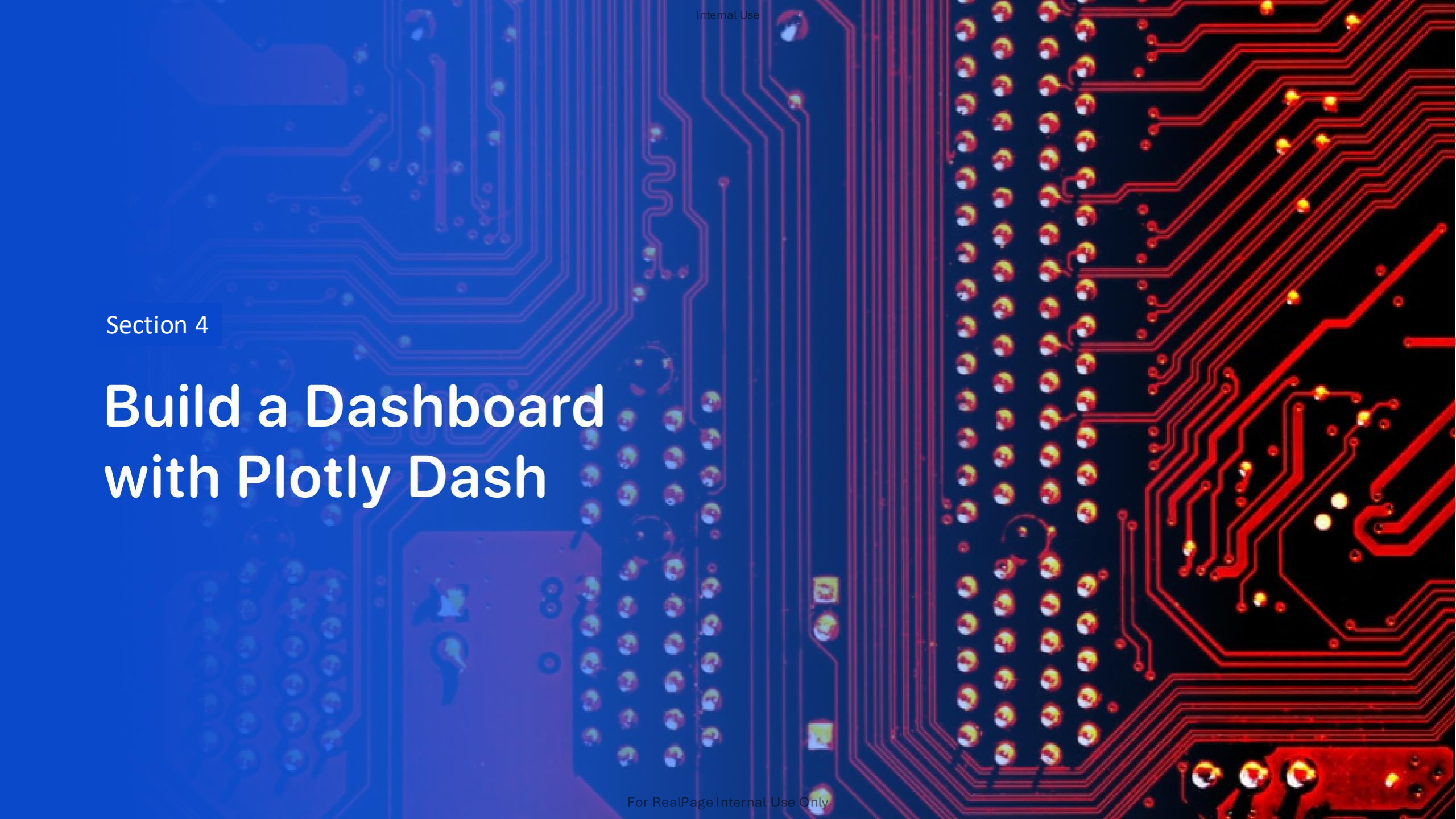
Launch Sites Proximities Analysis

Folium Launch Site Map





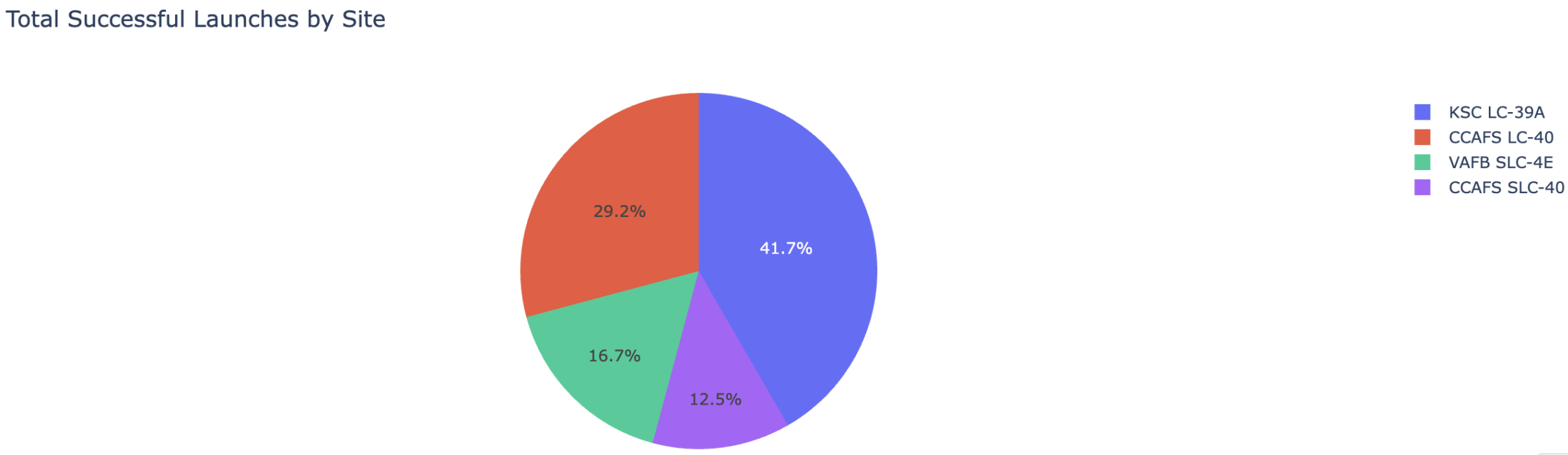
- The above interactive map was created using the Folium package for Python and illustrates the four launch sites with one site in California, one site in Texas, and two sites in Florida.
- The map includes markers to represent the number of successful and failed launches for each site, indicated by green and red, respectively.
- The KSC LC-39A launch site had the highest landing success rate at 10 out of 13, or 76.9%.



Section 4

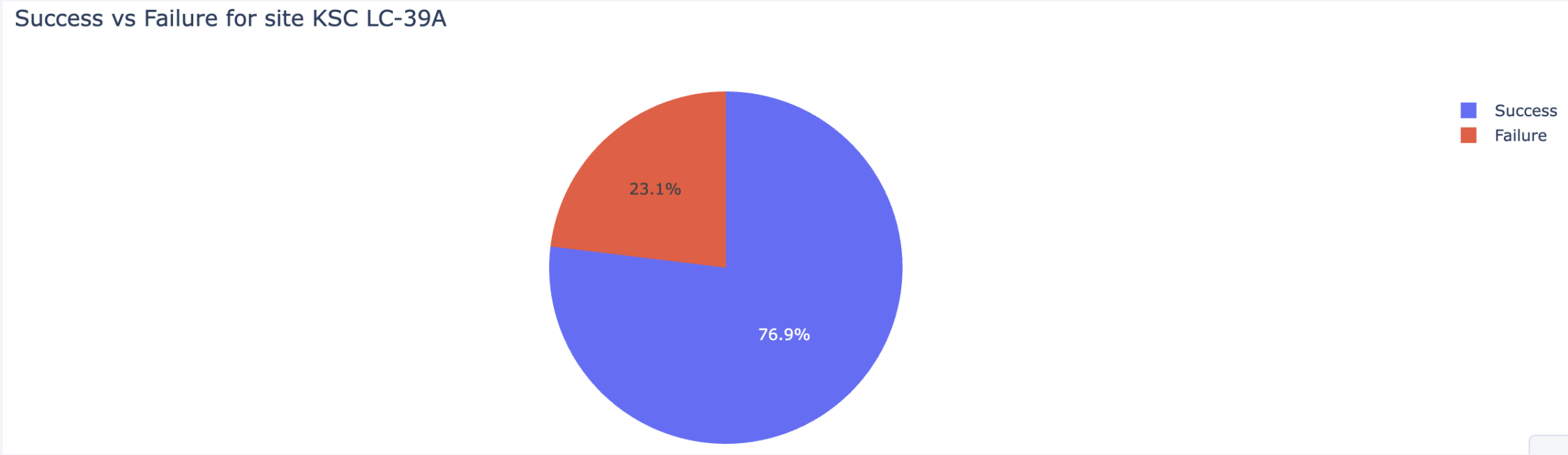
Build a Dashboard with Plotly Dash

Total Successful Launches by Site – Plotly Dash Dashboard



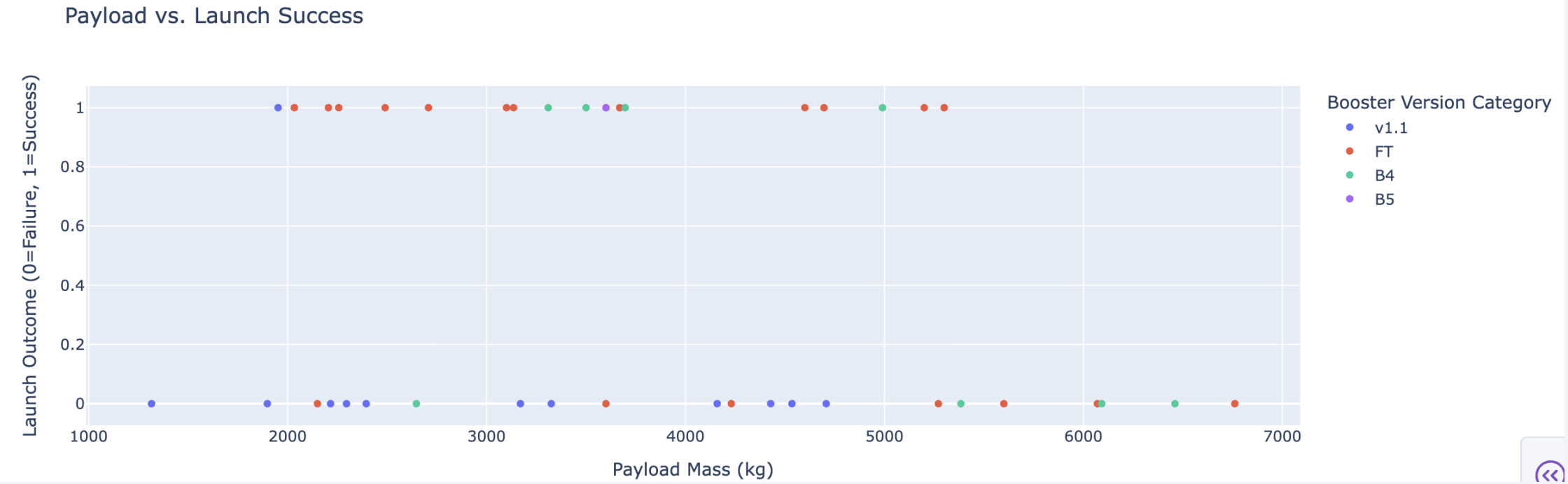
- KSC LC-39A accounted for the most successful launches among the group. Note, these percentages are based on the distribution of successful launches only. The individual success rate for each site may vary from the figures noted above when you account for total launches (i.e., include both successful and failed launches in the same underlying dataset).

Landing Success by Launch Site – Ploty Dash Dashboard



- KSC LC-39A had the highest rate of success with 76.9% of launches landing successful.

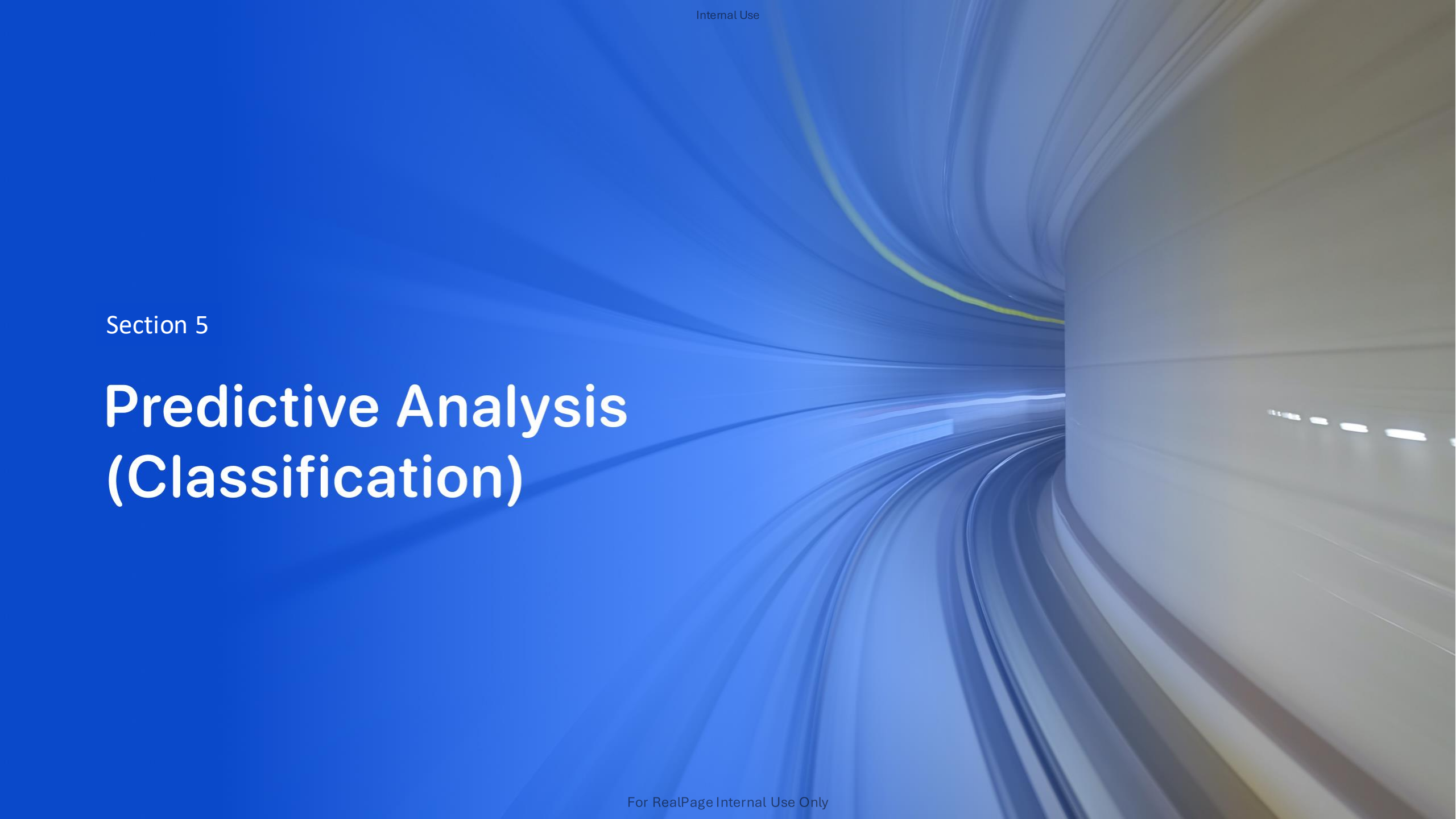
Booster Version and Payload vs Launch Success



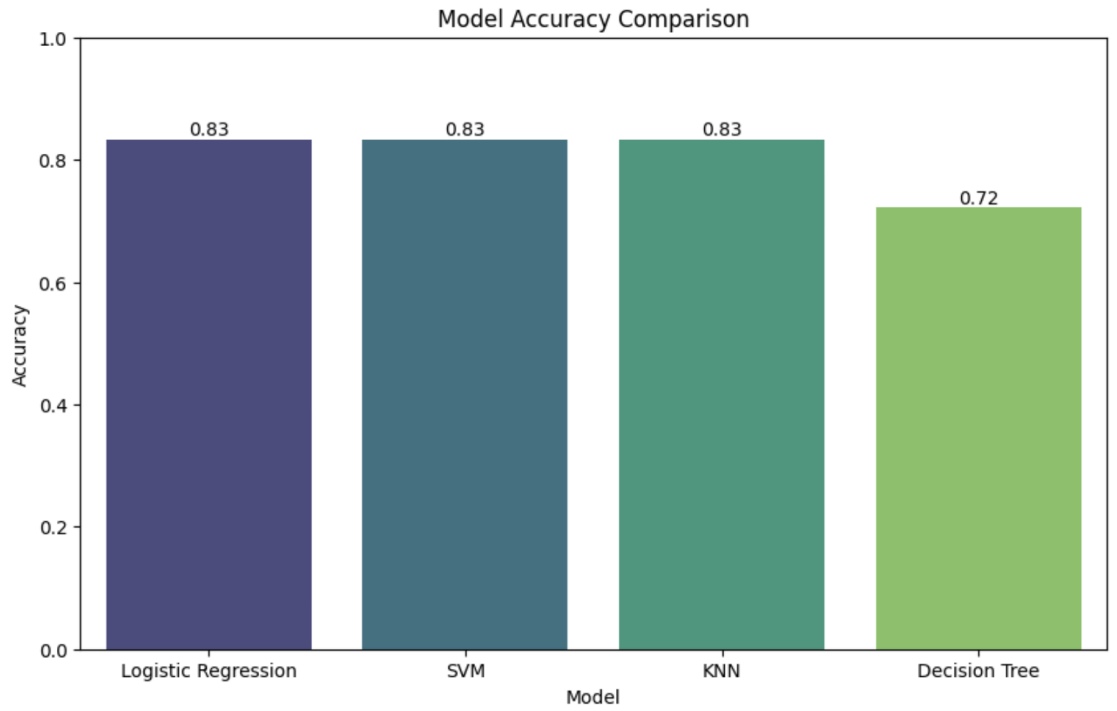
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.
- There were unsuccessful landings across the full range of payloads. However, most successful landings occurred with payloads between 2,000 and 4,000 kg or between 4,500 and 5,500 kg.
- The data suggests once payloads exceed 5,500 kg, there is nearly a 0% rate of success, with only one successful landing above this mark (not shown on chart but occurred at 9,600 kg).

Section 5

Predictive Analysis (Classification)

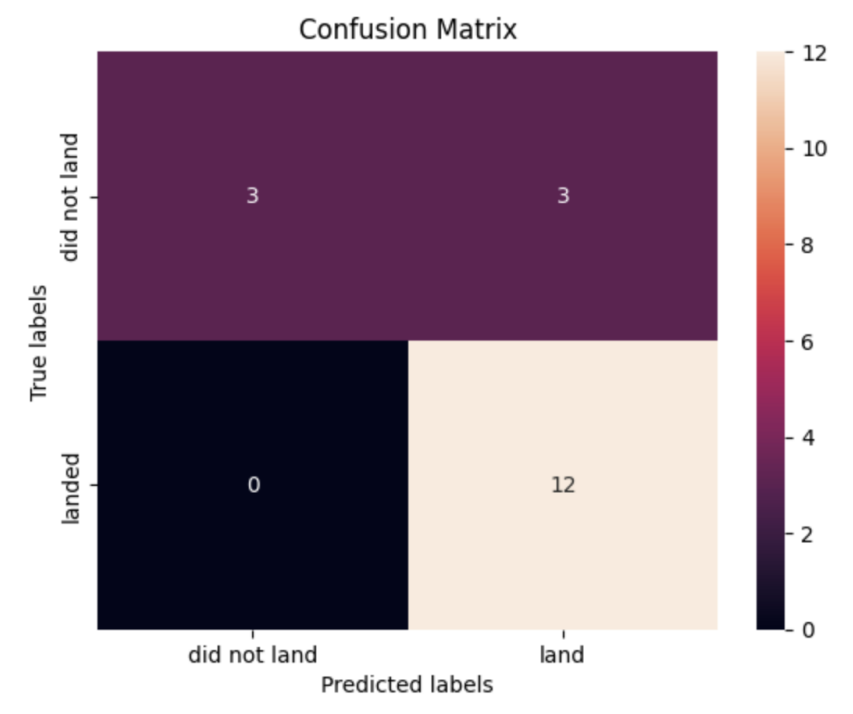


Classification Accuracy



- The Logistic Regression, SVM, and KNN models each achieved about 83% accuracy, likely due to the dataset's limited size and feature variability.
- The dataset is imbalanced, with 67% of records representing successful landings; thus, a naive model predicting all successes would achieve similar baseline accuracy.
- Although all models outperform the baseline, larger-scale testing is needed to identify the most accurate predictor. Given the comparable results, the simplest and most interpretable model is preferred.
- Because failed landings are costly, recall should be prioritized to reduce false negatives in future model evaluations.

Confusion Matrix



- The Confusion Matrix is identical for the Logistic Regression, SVM, and KNN models therefore accuracy and other model evaluation metrics (recall, precision, and f1 scores) were also the same.

Conclusions

- Since Logistic Regression, SVM, and KNN all achieved 83% accuracy, the Logistic Regression model is recommended for its interpretability and lower complexity.
- Logistic Regression coefficients suggest that the presence of landing legs and grid fins are strong predictors of successful landings, though further testing is needed to confirm the statistical significance of these relationships.
- As more launch data becomes available and feature diversity increases, model evaluation can be refined to identify the most accurate predictor.
- If stakeholders wish to emphasize the model's ability to predict failed landings, particularly given their high cost, it may be beneficial to consider recall as a key evaluation metric in future analyses.

Thank you!