IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

John Pauline Pineda
October 14, 2023

Outline

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

Executive Summary

- Methodologies explored in the study included the following:
  - Data collection using REST API and web scraping
  - Pre-processing methods
  - Exploratory data analysis using visualization and SQL
  - Interactive visual analytics using Folium and Plotly Dash
  - Predictive analysis encompassing model development, evaluation and selection
- Study results provided the following:
  - Complete and pre-processed data appropriate for analysis and modeling
  - Determined effects of features to successful landing outcomes through EDA visualization and SQL
  - Determined effects of spatial features to successful launch outcomes through Folium
  - Determined effects of features to successful launch outcomes through Plotly Dash
  - Classification model for predicting landing outcomes with sufficiently high accuracy

Introduction

- Space exploration has traditionally been associated with enormous costs. However, SpaceX, under the leadership of Elon Musk, has emerged as a game changer in the industry.
- Notably, SpaceX has revolutionized the space exploration landscape by consistently achieving cost savings and cost-efficiency that sets it apart from its competitors.
- SpaceX pioneered the reusable rocket technology. The Falcon 9 rocket, equipped with a reusable first stage, allows multiple launches with the same hardware. By reusing significant portions of their launch vehicles, SpaceX dramatically reduces the cost per launch, driving overall mission expenses.
- This capstone project generally aims to investigate the features driving the successful landing of SpaceX's first stage through exploratory data analysis and predictive model development, thereby allowing reuse for multiple launches and achieving cost savings for the program.
    - In particular, a classification model will be formulated which could provide robust and reliable estimates of the Falcon 9 rocket's successful landing outcome from an optimal subset of observations and predictors, while delivering accurate predictions when applied to new unseen data.

Section 1

Methodology

Methodology

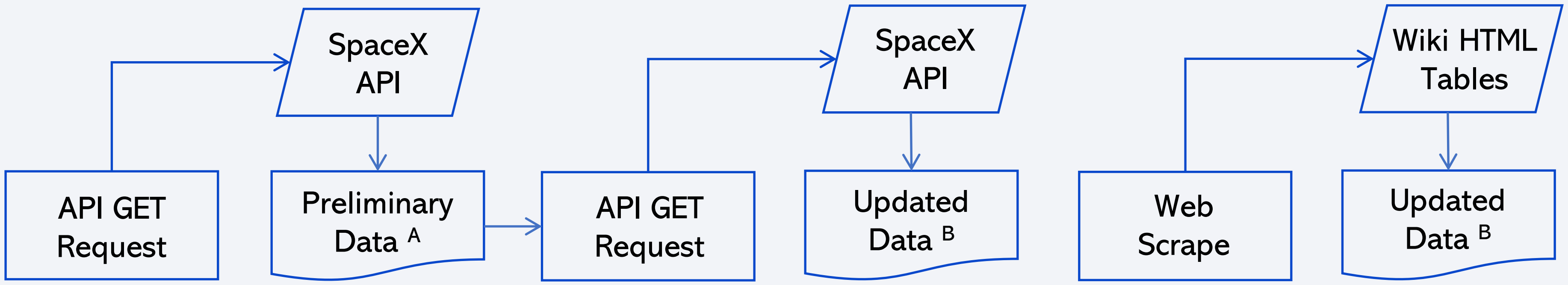
Executive Summary

- Data collection methodology
  - Sources: SpaceX REST API + web scraping
- Perform data wrangling
  - Pre-Processing: row + column filtering, missing data imputation, one-hot encoding, target creation
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Classification Models: decision tree, k-nearest neighbors, logistic regression, support vector machine
  - Train-Test Ratio: 80% train and hyperparameter tuning using10-fold cross validation + 20% test
  - Model Performance Evaluation: accuracy on train data with optimal hyperparameters + test data

Data Collection

- SpaceX launch data was gathered from 2 sources:
  - Source 1: SpaceX REST API containing data about SpaceX launches, including information about the rocket used, payload delivered, launch specifications, landing specifications and landing outcome, among others.
  - Source 2: web scraping of Falcon 9 launch data from relevant Wiki HTML tables
- Data collection process flowchart is presented as follows:

Option 1: SpaceX REST API **Option 2: World Wide Web**

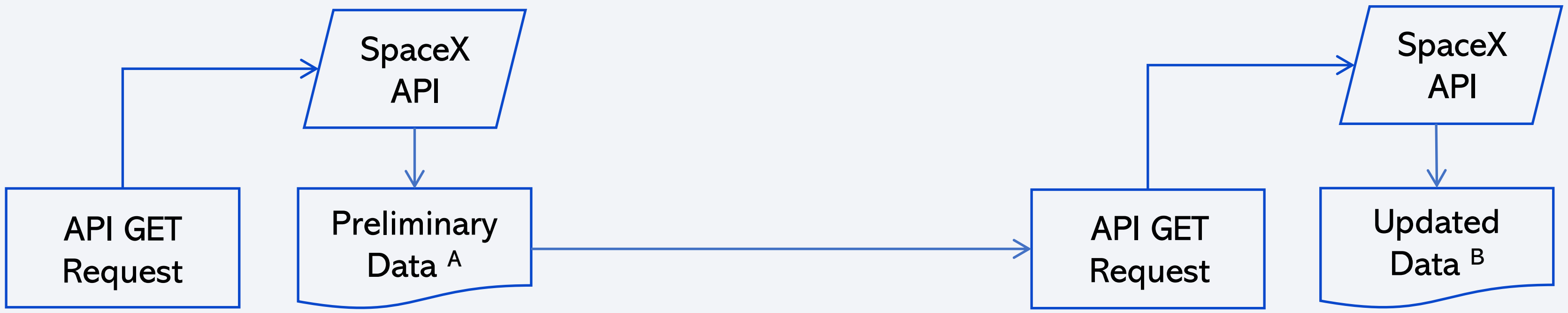


Data: [A] Rocket Launch [B] Rocket Launch + Booster + Launch Pad + Payload + Core

Data Collection – SpaceX API

- Data collection process flowchart with SpaceX REST calls is presented as follows:

Option 1: SpaceX REST API



API GET Request Steps

- Define SpaceX URL for Rocket Launch data
- Apply API GET request on SpaceX URL
- Decode the API response as JSON object
- Transform JSON object to a Pandas data frame
- Perform necessary feature sub-setting and row filtering
- Generate preliminary data from filtered data frame
- Create additional empty global variable list

API GET Request Steps

- Define different SpaceX URLs for Booster + Launch Pad + Payload + Core data
- Define auxiliary functions for applying API GET request on the SpaceX URLs
- Implement different auxiliary functions for API GET request
- Update global variable list from auxiliary function output
- Update preliminary data by combining with the list

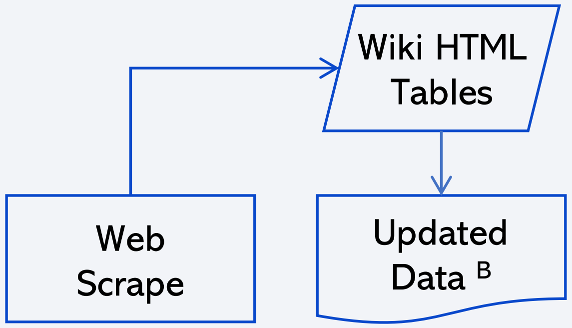
Data: [A] Rocket Launch [B] Rocket Launch + Booster + Launch Pad + Payload + Core

- GitHub URL of the completed SpaceX API calls Python notebook is provided here.

Data Collection – Scraping

- Data collection process flowchart with web scraping is presented as follows:

Option 2: World Wide Web



Web Scrape Steps

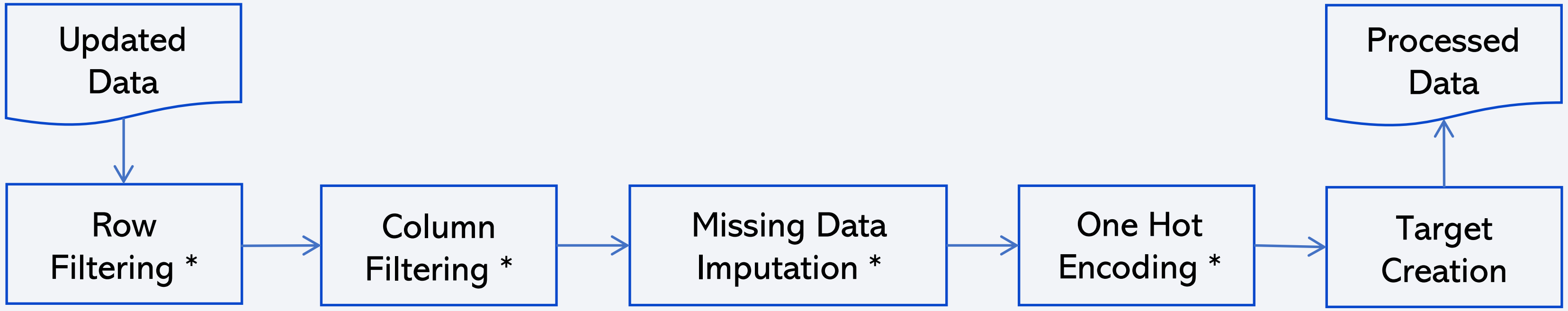
- Define WWW URL for Wiki HTML tables for Booster + Launch Pad + Payload + Core data
- Apply API GET request on WWW URL
- Decode the API response as BeautifulSoup object
- Extract all column names and row information from the BeautifulSoup object HTML tables
- Create a Pandas data frame from the parsed HTML tables

Data: B Rocket Launch + Booster + Launch Pad + Payload + Core

- GitHub URL of the completed web scraping Python notebook is provided here.

Data Wrangling

- Data wrangling was applied on the updated data set which involved cleaning, structuring, and transforming raw data into a format suitable for the subsequent analysis and modeling.
- Data wrangling process flowchart is presented as follows:



* Data wrangling steps which were incorporated under the data collection Python notebook

- GitHub URL of the data collection Python notebook which involved data wrangling operation is provided here.
- GitHub URL of the completed data wrangling Python notebook is provided here.

EDA with Data Visualization

- Exploratory data analysis involved formulating the appropriate charting tools to investigate and visualize the effects of the different features on landing success:
  - Categorical Plot: Effect of payload mass and flight number to landing outcome
  - Categorical Plot: Effect of launch site and flight number to landing outcome
  - Categorical Plot: Effect of payload mass and launch site to landing outcome
  - Categorical Plot: Effect of flight number and orbit to landing outcome
  - Categorical Plot: Effect of payload mass and orbit to landing outcome
  - Bar Plot: Effect of orbit to landing success rate
  - Line Plot: Effect of year to landing success rate
- GitHub URL of the completed EDA with data visualization Python notebook is provided here.

EDA with SQL

- Exploratory data analysis involved drill-down investigation using the following SQL queries:
  - Distinct Selection: SELECT statement + DISTINCT expression
  - String Patterns: SELECT statement + WHERE clause + LIKE predicate + LIMIT clause
  - Functions: SELECT statement + SUM|AVG |MIN functions + WHERE clause + LIKE predicate
  - Multiple Conditions: SELECT statement + WHERE clause + AND operator
  - Grouping Result Sets: SELECT statement + COUNT function + GROUP BY clause
  - Embedded Subquery: SELECT statement + WHERE clause + MAX function
  - Substring Extraction: SELECT statement + SUBSTR function + WHERE clause + AND operator
  - Range Conditions: SELECT statement + COUNT function + WHERE clause + BETWEEN | AND operators
  - Sorting Result Sets: SELECT statement + COUNT function + GROUP BY | ORDER BY clauses
- GitHub URL of the completed EDA with SQL Python notebook is provided here.

Build an Interactive Map with Folium

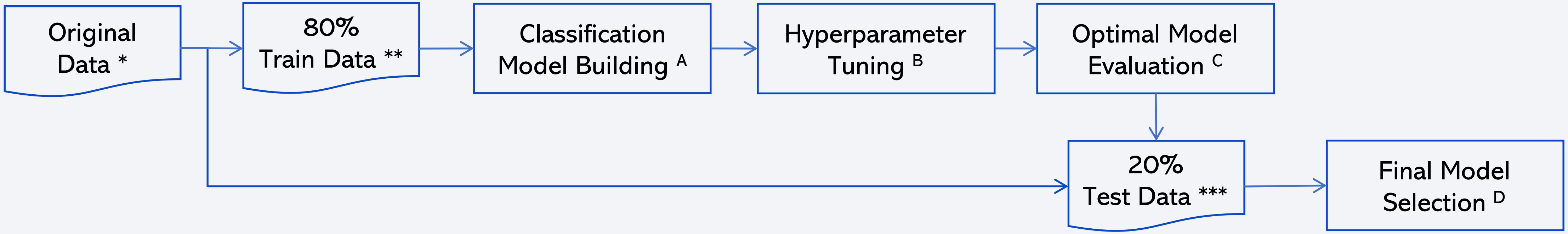
- Proximity analysis of launch data was conducted using an interactive map with given objects:
    - Circles: Indicates the launch sites
    - Markers: Indicates the names of the launch sites as a pop-up
    - Marker Cluster: Indicates individual launch record for each site color-coded based on launch outcome
    - Mouse Position: Indicates latitude and longitude information based on mouse position in the map
    - Lines: Connects the launch sites to identified landmarks (coastline, highway, railroad, major city)
    - Distance Circle: Indicates the distance between the launch sites and identified landmarks
- GitHub URL of the completed interactive Folium map Python notebook is provided here.
- NOTE: Despite enabling the 'Trust Notebook' setting, rendered maps still do not appear when the uploaded python notebook is viewed through Github. Screenshot of the complete code and map outputs were provided here as an alternative.

Build a Dashboard with Plotly Dash

- Interactive visual analytics involved formulating the appropriate dashboard charting tools to investigate the individual and combined effects of different features on launch success:
  - Pie Chart: Contribution of each launch site on the combined launch success count
  - Pie Chart: Distribution of the launch outcomes for each launch site
  - Categorical Plot: Effect of payload mass and booster version to launch outcome
  - Range Slider: Effect of changing the payload mass range to launch outcome
- GitHub URL of the completed Plotly Dash Python code is provided here.

Predictive Analysis (Classification)

- Predictive analysis involved the process steps described as follows :
  - Model Development: Included classification model development and tuning of model hyperparameters
  - Model Evaluation: Involved both internal and external evaluation using the accuracy metric
  - Model Selection: Involved the identification of the best performing model based on assessment



* 90 observations + 83 features | ** 72 observations + 83 features | *** 18 observations + 83 features
[A] 4 candidate models built using the Logistic Regression, Support Vector Machine, Decision Tree and K-Nearest Neighbors structures
[B] Hyperparameters of 4 candidate models fine-tuned and internally evaluated using a 10-fold cross-validation
[C] 4 candidate models with optimal hyperparameters were externally evaluated on the test set
[D] Best model among 4 optimal candidate models determined based on internal and external evaluation results

- GitHub URL of the completed Predictive analysis Python notebook is provided here.

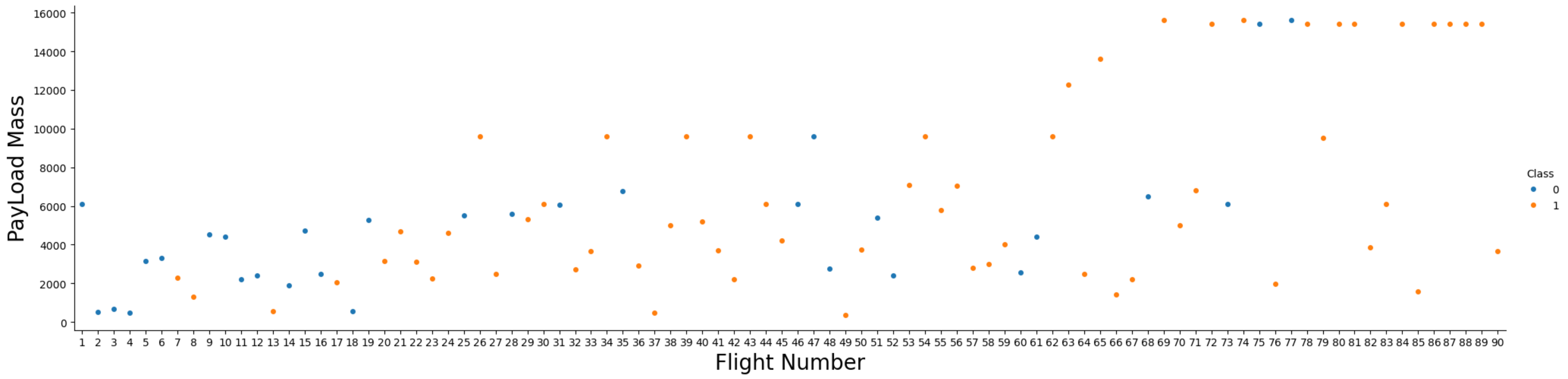
Results

- Exploratory data analysis results
  - Insights Drawn from EDA
- Interactive analytics demo in screenshots
  - Launch Sites Proximities Analysis
  - Build a Dashboard with Plotly Dash
- Predictive analysis results
  - Predictive Analysis (Classification)

Section 2

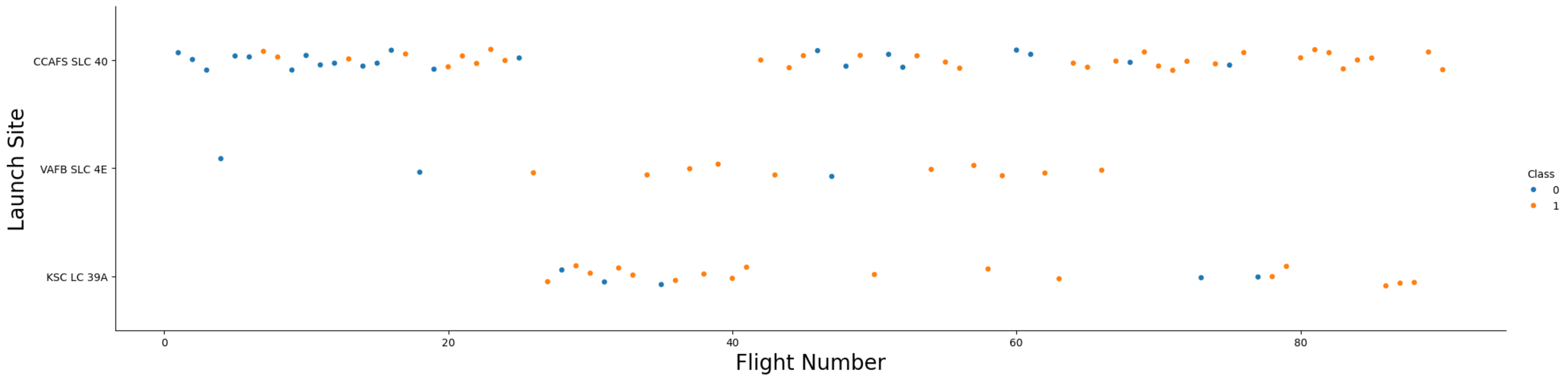
Insights drawn from EDA

Flight Number vs. Payload



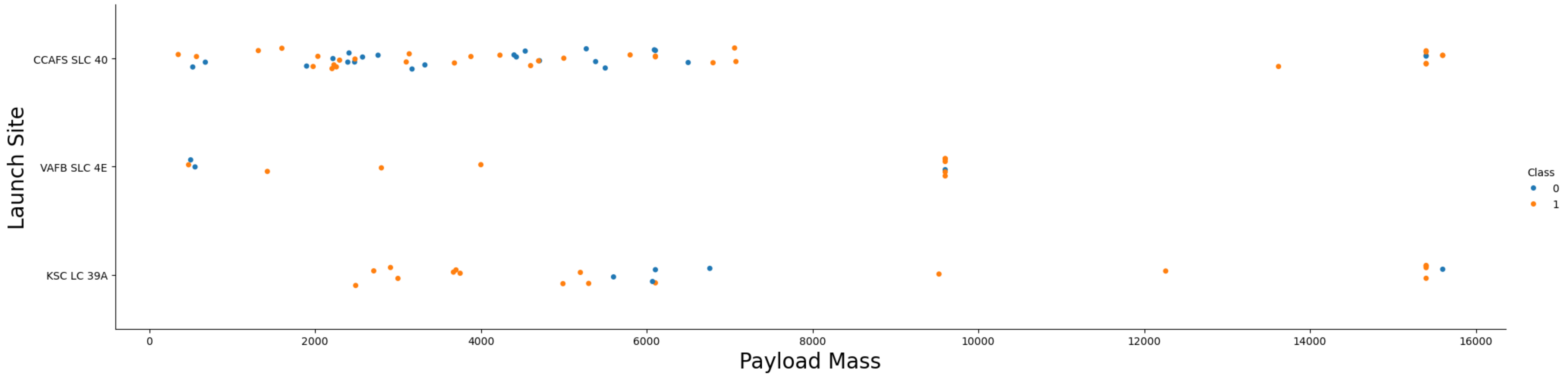
- Categorical Plot: Effect of Payload Mass and Flight Number to Landing Success Class
  - More successful landings generally observed as flight number increases.
  - Successful landings observed across varying ranges of payload mass including those >9K.

Flight Number vs. Launch Site



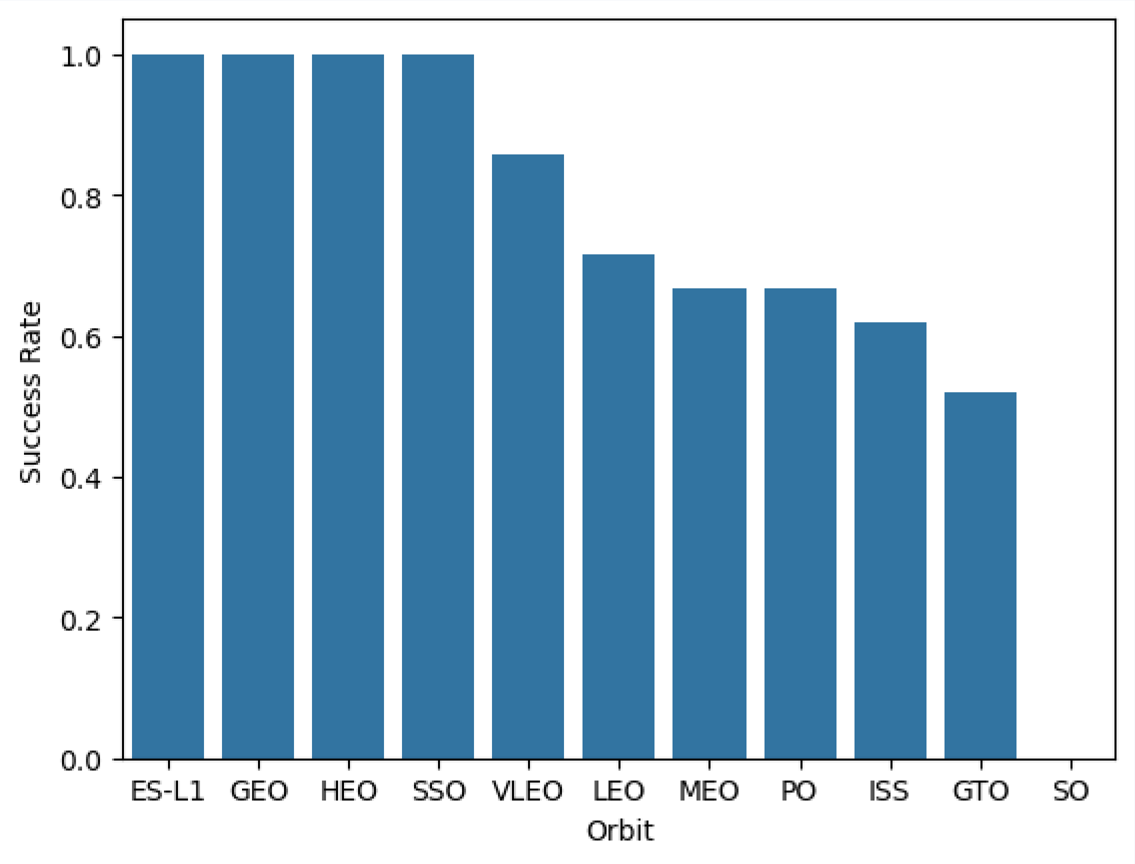
- Categorical Plot: Effect of Launch Site and Flight Number to Landing Success Class
  - More successful landings generally observed for VAFB SLC 4E and KSC LC 39A launch sites.
  - Increased flight numbers only observed for CCAFS SLC 40 and KSC LC 39A launch sites.

Payload vs. Launch Site



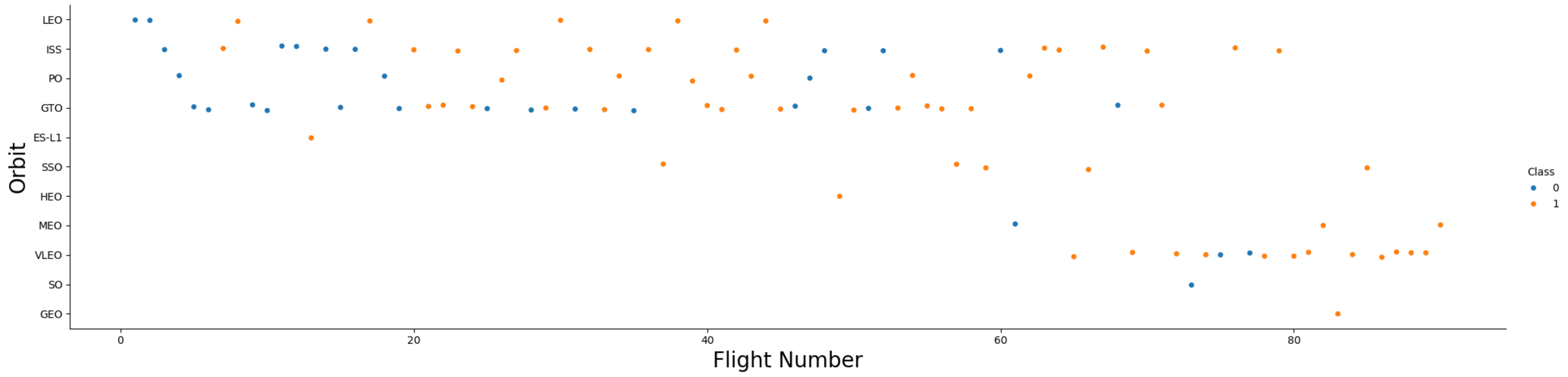
- Categorical Plot: Effect of Payload Mass and Launch Site to Landing Success Class
  - Successful landings observed across varying ranges of payload mass including those >9K.
  - Higher payload masses >10K only observed for CCAFS SLC 40 and KSC LC 39A launch sites.

Success Rate vs. Orbit Type



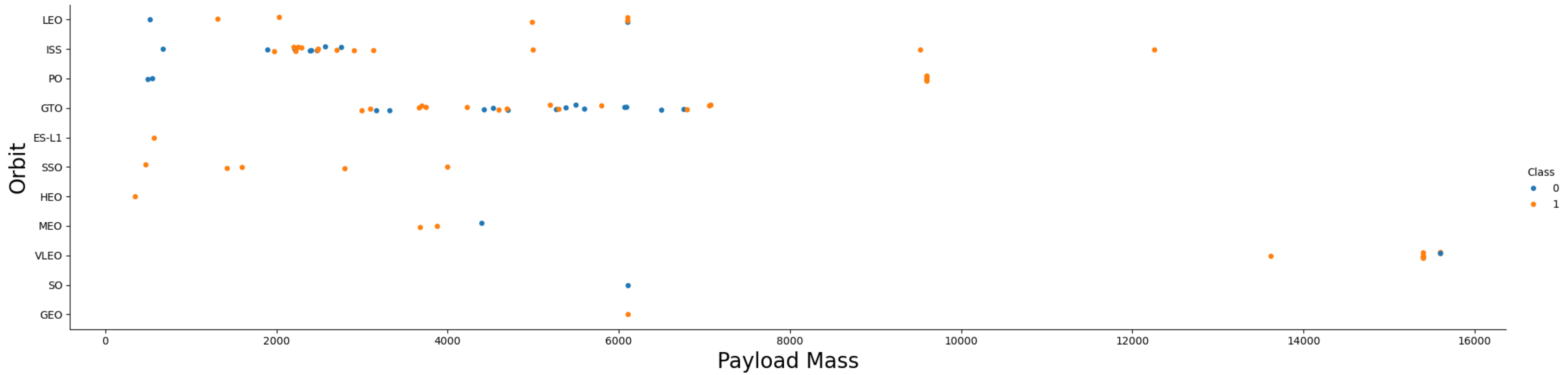
- Bar Plot: Effect of Orbit to Landing Success Rate
  - Higher landing success rate observed for the ES-L1, GEO, HEO and SSO orbits.

Flight Number vs. Orbit Type



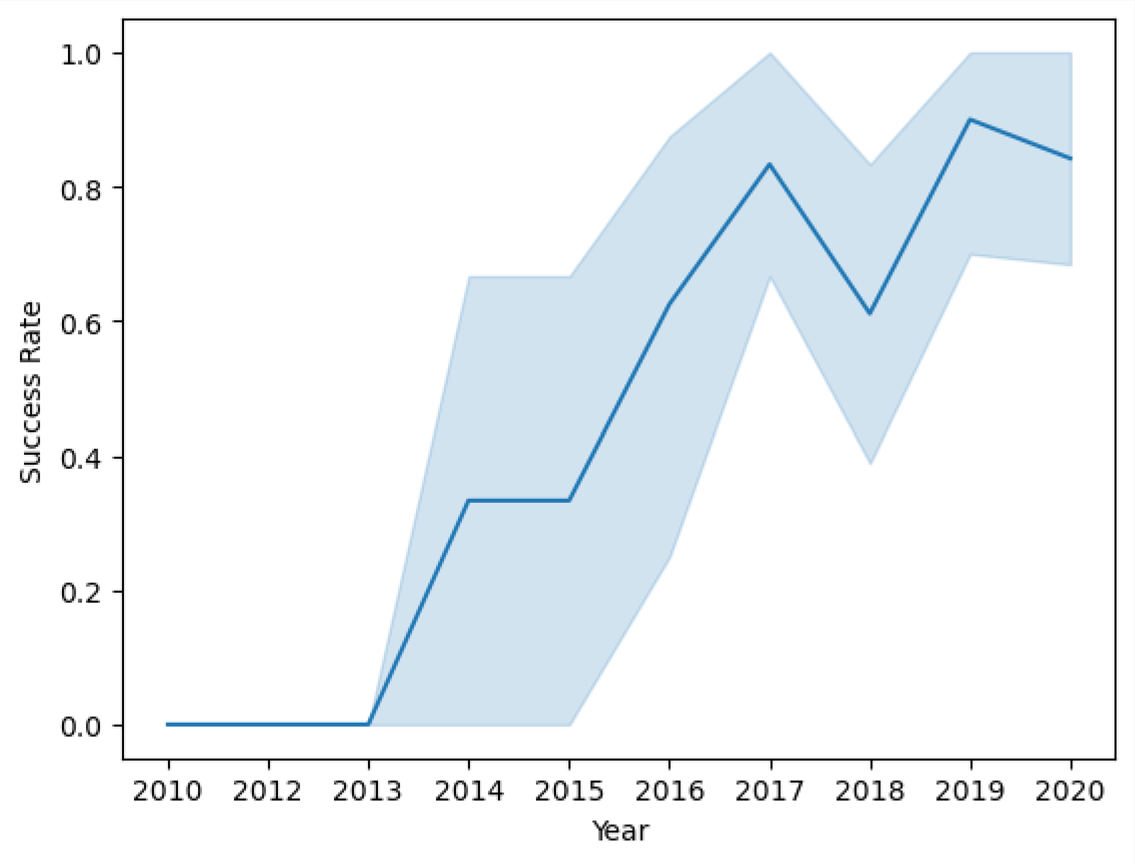
- Categorical Plot: Effect of Flight Number and Orbit to Landing Success Class
  - Previous observation on higher landing success rate for ES-L1, GEO and HEO orbits driven by low count.
  - Increased flight numbers only observed for the VLEO orbit.
  - Given a considerable number of flights, LEO, SSO and VLEO reported more successful landings.

Payload vs. Orbit Type



- Categorical Plot: Effect of Payload Mass and Orbit to Landing Success Class
  - Higher payload masses >9K only observed for ISS, PO and VLEO orbits.
  - Successful landings observed across varying ranges of payload mass including those >9K.

Launch Success Yearly Trend



- Line Plot: Effect of Year to Landing Success Rate
  - Higher landing success rate observed from year 2013 onwards.

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

- Distinct Selection: SELECT statement + DISTINCT expression
  - There were 4 unique launch sites included the space mission comprised of CCAFS LC-40, VAFB SLC-4E, KSC LC-39A and 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

- String Patterns: SELECT statement + WHERE clause + LIKE predicate + LIMIT clause
  - 5 records of launch sites beginning with the string ‘CCA’ were returned.

Total Payload Mass

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_)
AS "TOTAL PAYLOAD MASS", Customer
FROM SPACEXTABLE
WHERE Customer LIKE "%NASA (CRS)%";
```

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

TOTAL PAYLOAD MASS	Customer
48213	NASA (CRS)

- Functions: SELECT statement + SUM function + WHERE clause + LIKE predicate
  - The total payload mass carried by boosters launched by NASA (CRS) was 48213.

Average Payload Mass by F9 v1.1

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_)
AS "AVERAGE PAYLOAD MASS", Customer, Booster_Version
FROM SPACEXTABLE
WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

AVERAGE PAYLOAD MASS	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

- **Functions:** SELECT statement + AVG function + WHERE clause + LIKE predicate
  - The average payload mass carried by booster version F9 v1.1 was 2534.67.

First Successful Ground Landing Date

```
%%sql
SELECT MIN(DATE) FROM SPACEXTABLE
WHERE "Landing_Outcome"
LIKE "Success (ground pad)"
```

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

MIN(DATE)
2015-12-22

- Functions: SELECT statement + MIN function + WHERE clause + LIKE predicate
  - The first successful landing outcome in a ground pad was achieved on December 22, 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT * FROM SPACEXTABLE
WHERE Landing_Outcome = "Success (drone ship)" AND  PAYLOAD_MASS__KG_>4000  AND PAYLOAD_MASS__KG_<6000;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2016-06-05	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-11-10	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

- Multiple Conditions: SELECT statement + WHERE clause + AND operator
  - There were 4 boosters launched on the dates June 5, 2016; August 14, 2016; March 30, 2017; and November 10, 2017 which had success landing in drone ship and have payload masses greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT Mission_Outcome, COUNT(Mission_Outcome) AS "Total Number of Mission Outcomes"
FROM SPACEXTABLE
GROUP BY Mission_Outcome;
```

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

Mission_Outcome	Total Number of Mission Outcomes
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- Grouping Result Sets: SELECT statement + COUNT function + GROUP BY clause
  - There were 100 successful mission outcomes and 1 failed mission outcome.

Boosters Carried Maximum Payload

```
%%sql
SELECT Booster_Version,Payload, PAYLOAD_MASS__KG_
FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

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

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

- Embedded Subquery: SELECT statement + WHERE clause + MAX function
  - The booster versions which have carried the maximum payload mass included 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 and F9 B5 B1049.7.

2015 Launch Records

```
%%sql
SELECT SUBSTR(Date, 6, 2) AS MONTH, Booster_Version, Launch_Site, Landing_Outcome
FROM SPACEXTABLE
WHERE SUBSTR(Date, 0, 5) = "2015" AND Landing_Outcome = 'Failure (drone ship)';
```

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

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

- Substring Extraction: SELECT statement + SUBSTR function + WHERE clause + AND operator
  - There were two records for the months of April and October in 2015 with available details for the month names, failure landing outcomes in drone ship, booster versions and launch site.

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

```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome) AS "Landing Outcome Count"
FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY "Landing Outcome Count" DESC;
```

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

Landing_Outcome	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

- Range Conditions + Sorting Result Sets : SELECT statement + COUNT function + WHERE clause + BETWEEN | AND operators + GROUP BY | ORDER BY clauses
  - There were 8 types of landing outcomes between the dates June 4, 2010 and March 20, 2017 which were ranked by count in descending order.

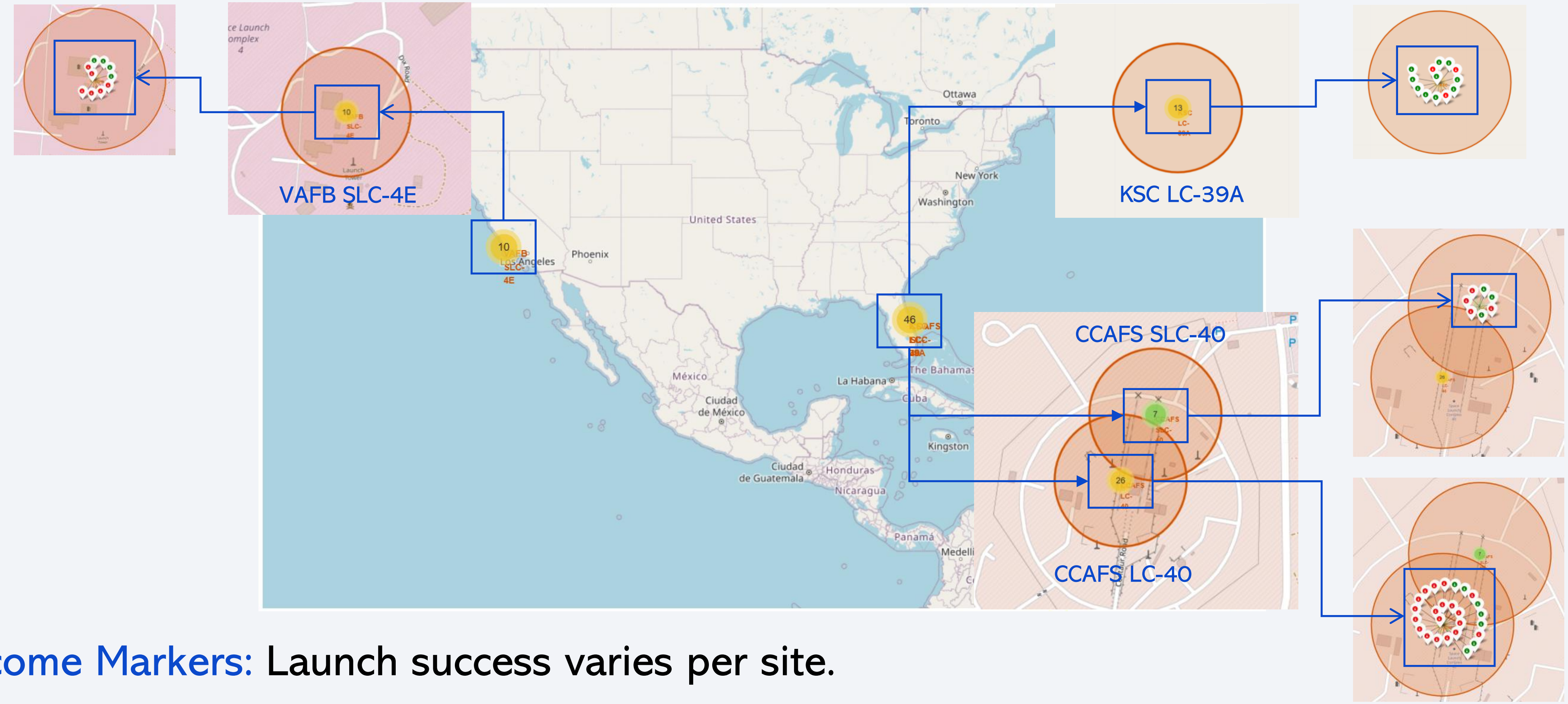
Section 3

Launch Sites Proximities Analysis

Location Markers of Launch Sites

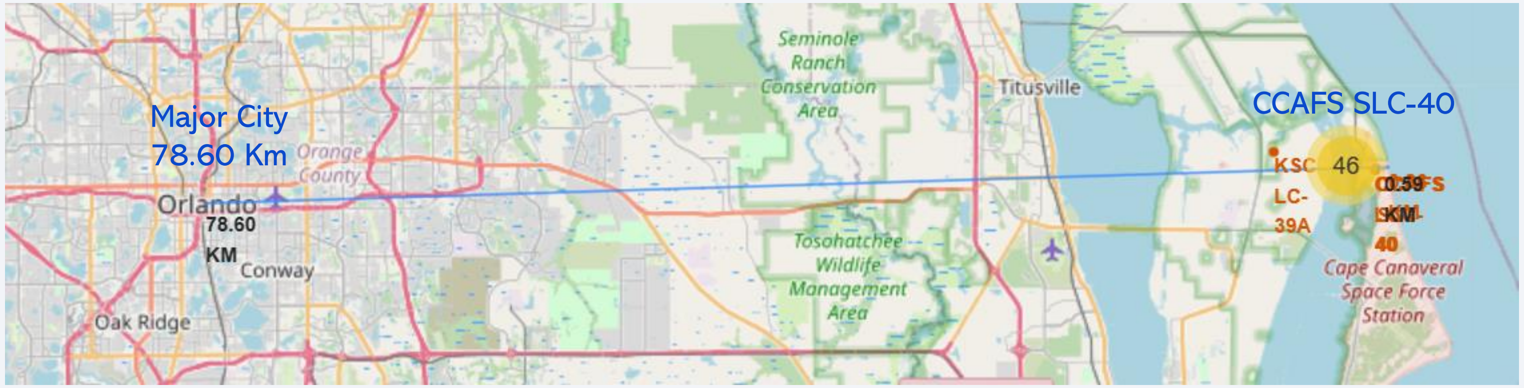
- **Location Markers:** US launch sites were normally situated at coastline locations.
  - West Coast: VAFB SLC-4E
  - East Coast: KSC LC-39A + CCAFS SLC-40 + CCAFS LC-40

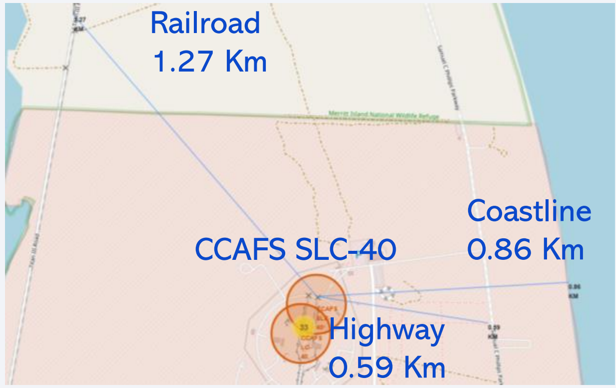
Color-Labeled Launch Outcomes by Launch Site



- Outcome Markers: Launch success varies per site.
  - The KSC LC-39A site demonstrated a relatively higher ratio of launch success as compared to the CCAFS SLC-40, CCAFS LC-40 and VAFB SLC-4E sites.

Proximity Analysis of Launch Site to Coastline, Highway, Railway and Major City



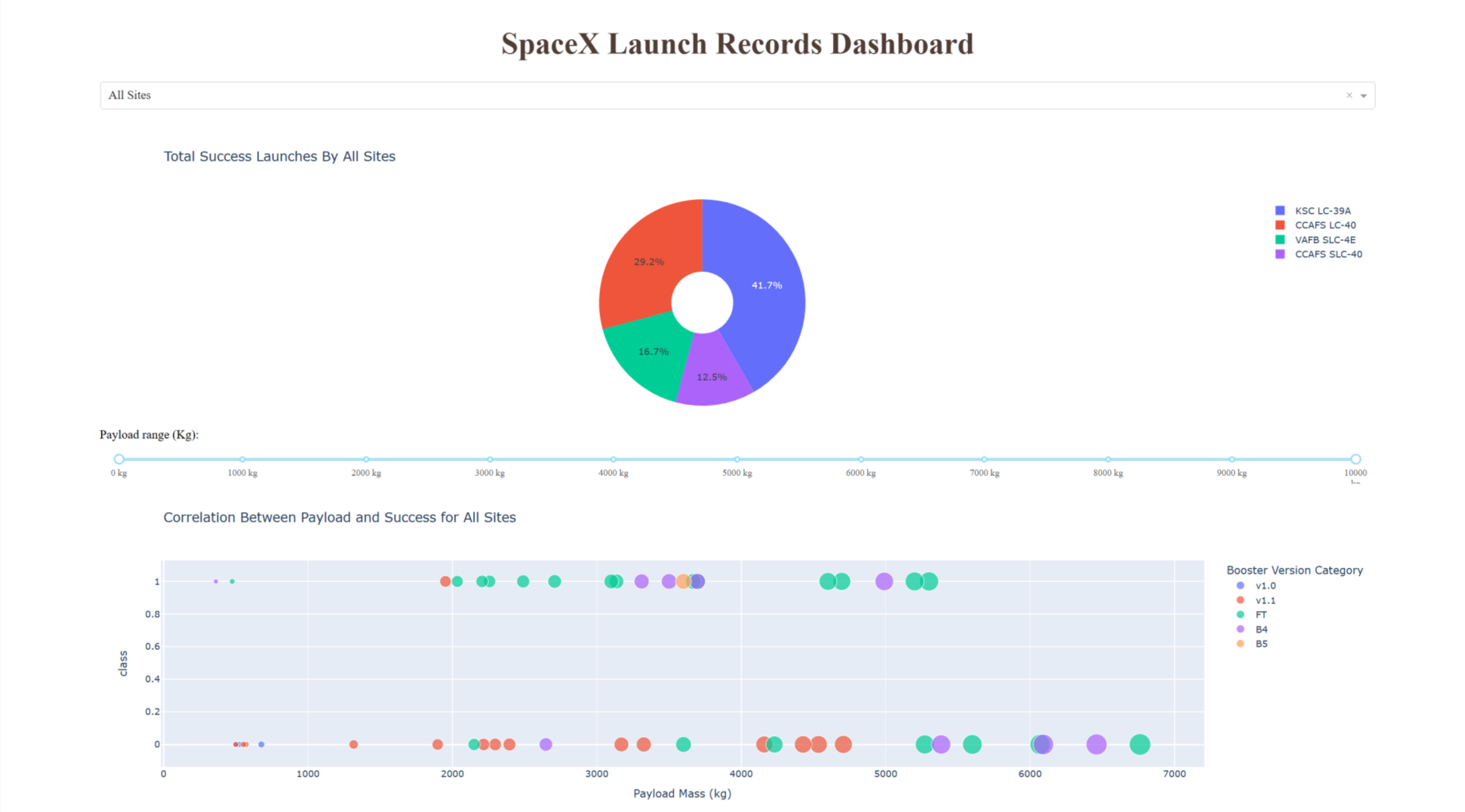


- Distance Markers: Launch sites were located far from major cities but relatively closer to coastlines, highways and railroads.
  - Taking CCAFS SLC-40 as an example, its approximate distance to a major city identified as Orlando was 78.60 Km.
  - The approximate distance of the CCAFS SLC-40 launch site to the coastline was 0.86 Km.
  - The approximate distance of the CCAFS SLC-40 launch site to a highway was 0.59 Km.
  - The approximate distance of the CCAFS SLC-40 launch site to a railroad was 1.27 Km.

Section 4

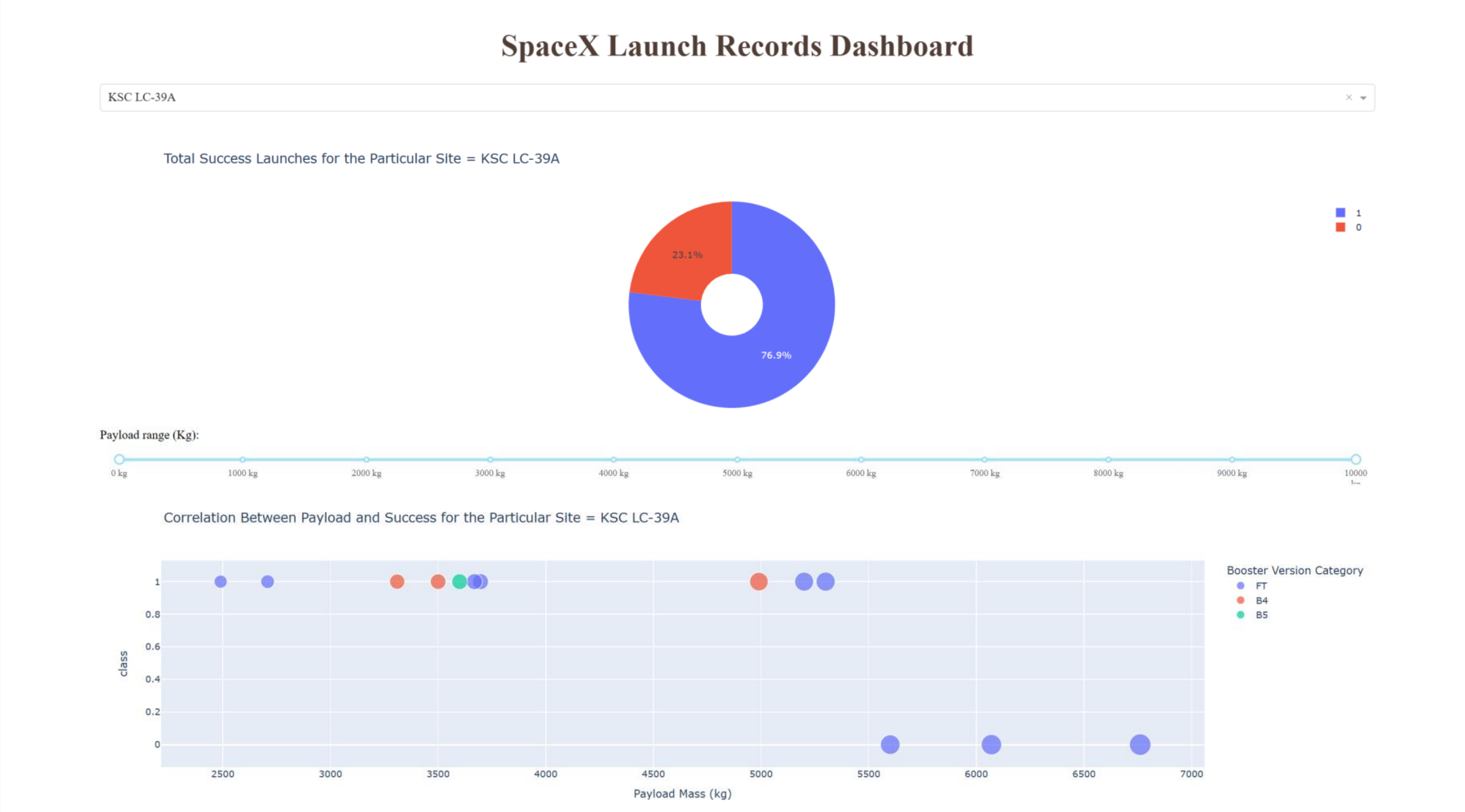
Build a Dashboard with Plotly Dash

Combined Launch Success Count For All Sites



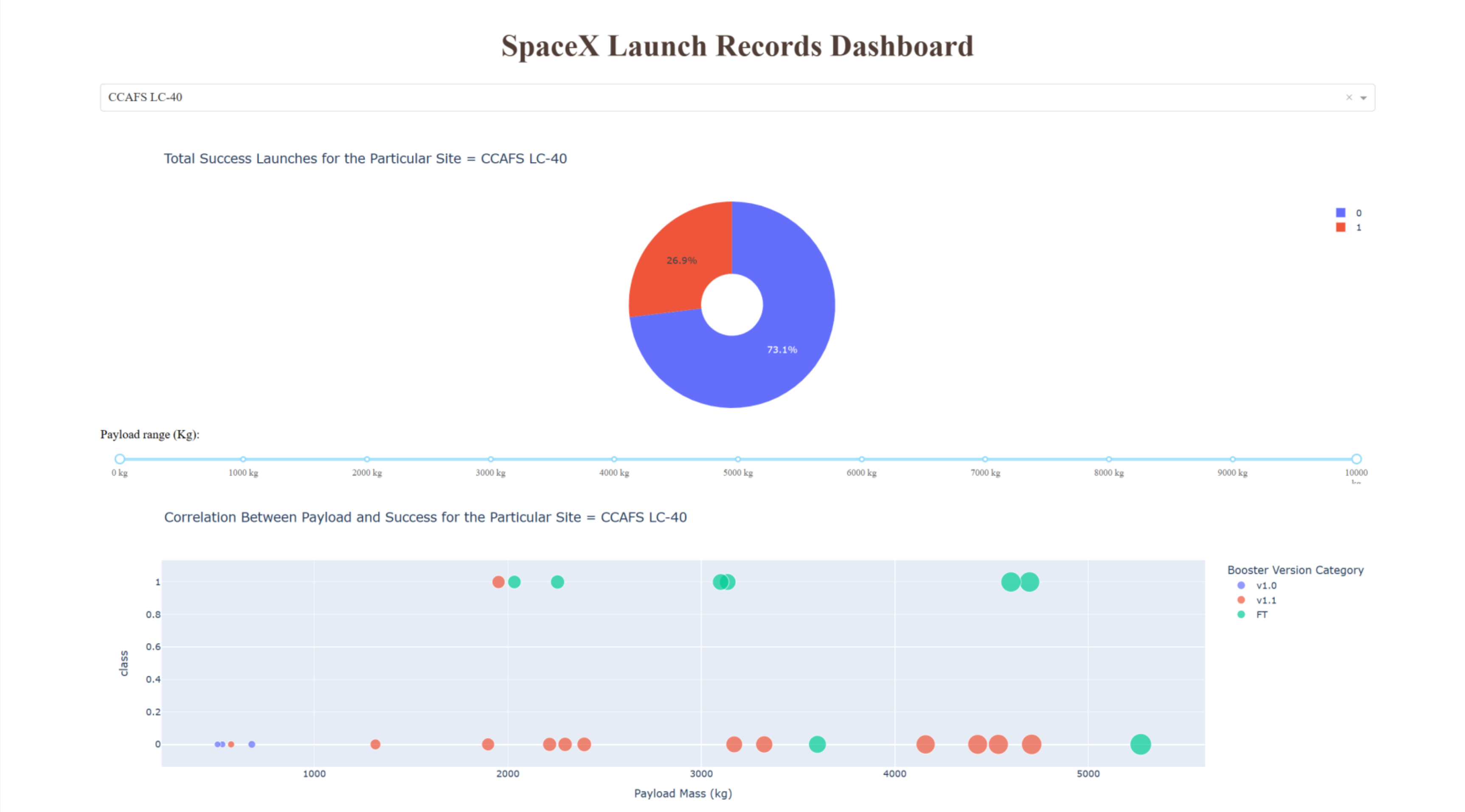
- Launch Success Count: The most number of successful launches was obtained from KSC LC-39A contributing 41.7% to the overall launch success count from all sites.

Launch Success Ratio Per Individual Site – Rank 1



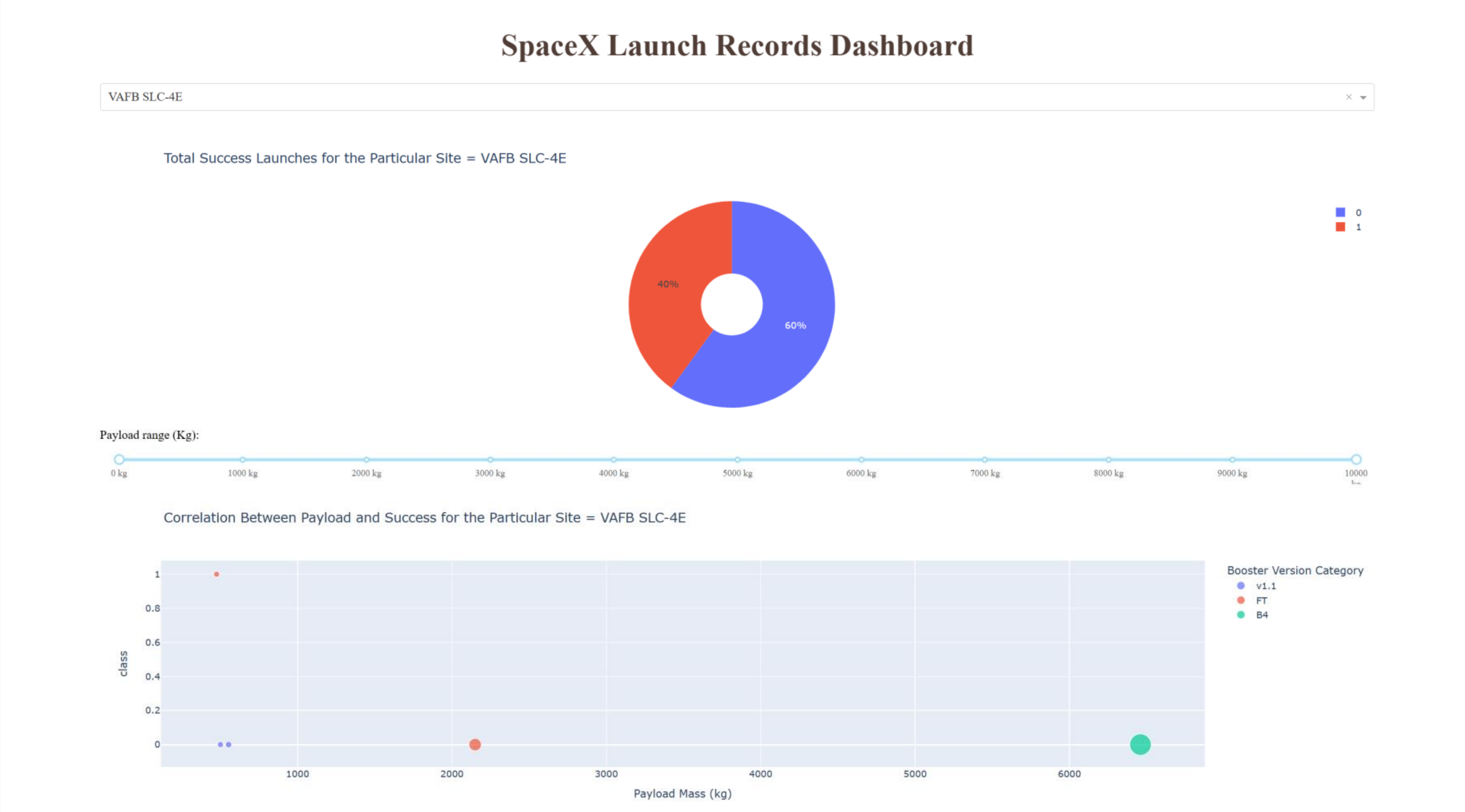
- Launch Success Ratio: Among all sites, the launch success ratio from KSC LC-39A was ranked first. 76.9% of all launches from KSC LC-39A were successful.

Launch Success Ratio Per Individual Site – Rank 2



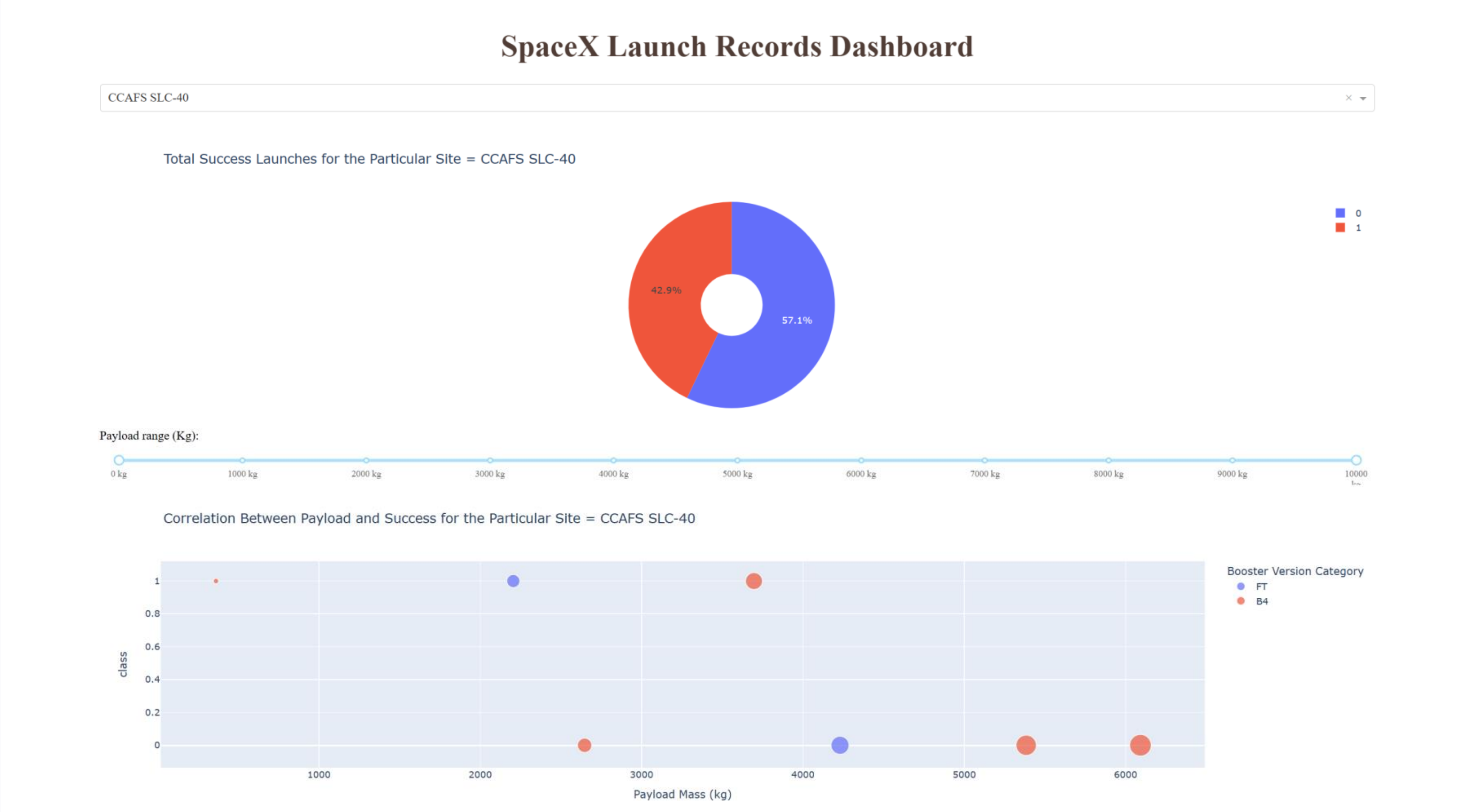
- Launch Success Ratio: Among all sites, the launch success ratio from CCAFS LC-40 was ranked second. 73.1% of all launches from CCAFS LC-40 were successful.

Launch Success Ratio Per Individual Site – Rank 3



- Launch Success Ratio: Among all sites, the launch success ratio from VAFB SLC-4E was ranked third. 60.0% of all launches from VAFB SLC-4E were successful.

Launch Success Ratio Per Individual Site – Rank 4

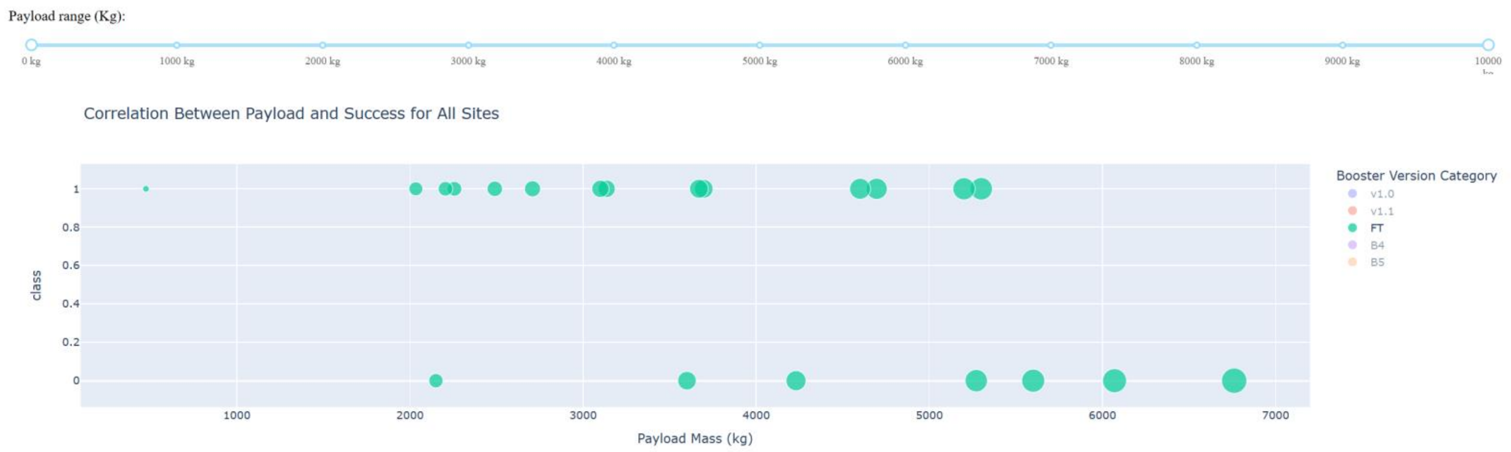


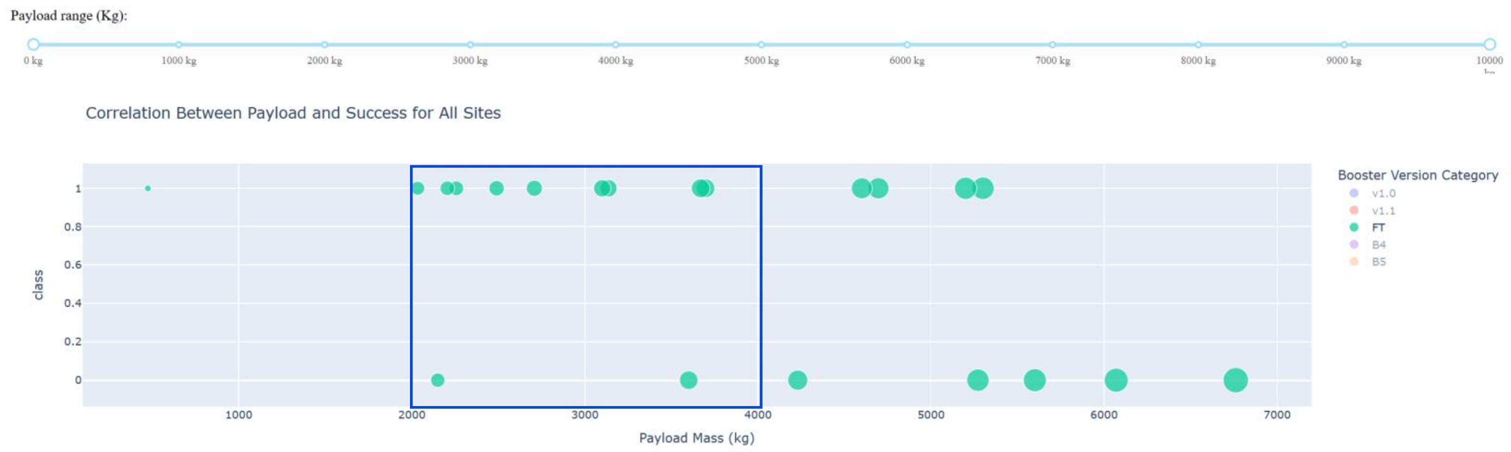
- Launch Success Ratio: Among all sites, the launch success ratio from CCAFS SLC-40 was ranked third. 57.1% of all launches from CCAFS SLC-40 were successful.

Launch Outcomes By Payload Ranges

- **Launch Outcomes By Payload:** At a payload range of 2000-4000 Kg, launch success ratio is at 60.0%. In contrast, launch success ratio is only 22.7% at 4000-7000 Kg.

Launch Outcomes By Booster Version



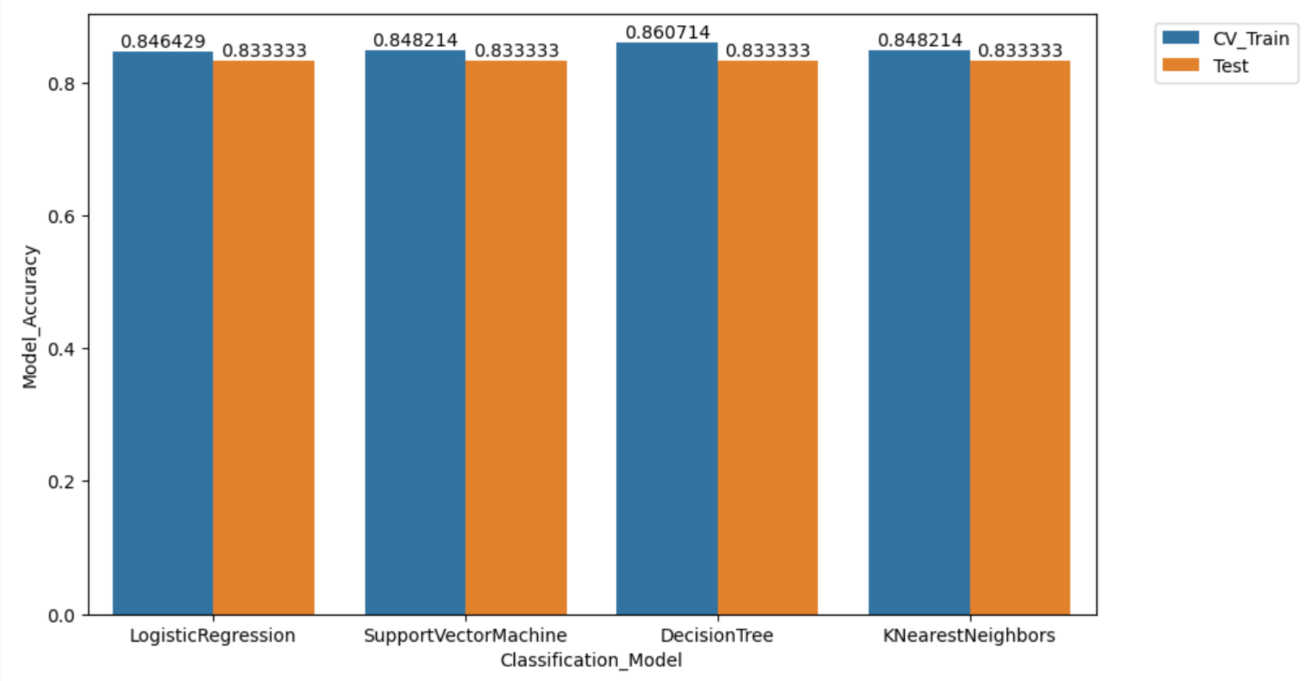


- Launch Outcomes By Booster Version: The highest launch success ratio was from booster FT version at 65.0%. This increases to 81.8% for 2000-4000 Kg payloads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



Optimal LR Model
C: 0.01
penalty: l2
solver: lbfgs

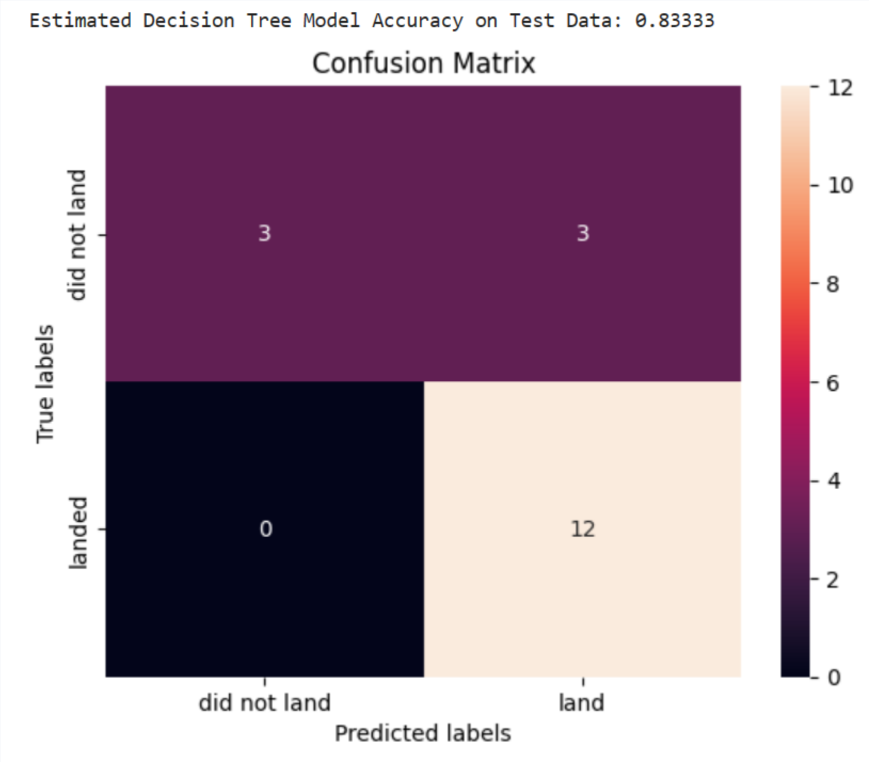
Optimal SVM Model
C:1.0
gamma: 0.03162
kernel: sigmoid

Optimal Decision Tree Model
criterion: gini
max_depth: 2
max_features: sqrt
min_samples_leaf: 1
min_samples_split: 2
splitter: best

Optimal KNN Model
algorithm: auto
n_neighbors: 10
p: 1

- Internal + External Model Evaluation: All 4 candidate optimal models equally achieved 83.33% accuracy on the test set. Higher accuracy was however observed for the decision tree model at 86.07% based on internal evaluation using a 10-fold cross-validation.
  - Cross-Validated Accuracy on Train: 86.07% (Decision Tree), 84.82% (KNN, SVM), 84.64% (LR)
  - Accuracy on Test: 83.33% (LR, SVM, Decision Tree, KNN)

Confusion Matrix



Optimal Decision Tree Model
criterion: gini
max_depth: 2
max_features: sqrt
min_samples_leaf: 1
min_samples_split: 2
splitter: best

- Classification Performance: With a classification accuracy determined as 86.07% for the test set, the final selected model based on a decision tree structure correctly predicted 15 (12 true positive and 3 true negative cases) out of 18 observations. 3 false positive cases were noted which the model predicted as successful landing, but actually failed.

Conclusions

- Data collection for the analysis involved SpaceX REST API and web scraping.
- Appropriate pre-processing methods including row and column filtering, missing data imputation, one-hot encoding, target creation were applied to prepare the data for subsequent analysis and modeling.
- EDA using visualization and SQL demonstrated the effects of features including payload mass, flight number, launch site and orbit to successful landing outcomes.
- Interactive visual analytics using Folium demonstrated the effects of spatial features including proximities to a coastline, highways, railways and major cities to launch sites.
- Interactive visual analytics using Plotly Dash demonstrated the effects of features including launch site, payload mass and booster version to successful launch outcomes.
- A decision tree classification model provided a robust and reliable prediction for successful landing outcomes with a cross-validated accuracy of 86.07% and independent testing accuracy of 83.33%.

Appendix

- Source Data
  - API URL: Launch Records | Rockets | Launch Pads | Payloads | Cores
  - Wikipedia URL: HTML Tables
- Python Notebooks | Codes
  - GitHub URL: SpaceX API Calls
  - GitHub URL: Web Scraping
  - GitHub URL: Data Wrangling
  - GitHub URL: EDA with Data Visualization
  - GitHub URL: EDA with SQL
  - GitHub URL: Interactive Visual Analytics with Folium
  - GitHub URL: Interactive Visual Analytics with Folium (alternative)
  - GitHub URL: Interactive Visual Analytics with Plotly Dash
  - GitHub URL: Predictive Analysis

Thank you!