IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jim Whiting
Capstone project
May 8, 2025

Outline – the flow of this presentation

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

Note: this presentation is intended for an audience of data scientists; there are also some executive-focused slides as well – namely the Executive Summary and Conclusion.

The format of the presentation was via a template.

More detailed info on models and conclusion on slides 48-53.

Executive Summary – Overview of methodologies

Data Collection

- **SpaceX REST API**: Pulled launch metadata (flight #, booster version, payload mass, landing outcome, reuse count, etc.).
- **Web Scraping:** Beautiful Soup extraction of Wikipedia Falcon 9/Heavy launch tables; merged into unified dataset.

Data Wrangling & EDA

- Imputed missing payload masses (mean substitution) and encoded categorical flags (e.g., GridFins, Legs).
- Calculated baseline landing success rate (66.7 %) and class labels.
- **SQL-driven summaries:** site counts, orbit success rates, yearly trends.
- **Visualization:** static charts (payload vs. success), Folium maps of landing sites, interactive Plotly Dash dashboard.

Predictive Modeling

- **Preprocessing**: StandardScaler on numeric inputs; one-hot encoding where needed; 80/20 train/test split.
- **Algorithms**: SVM, CART decision tree, K-nearest neighbors, logistic regression.
- **Hyperparameter tuning**: 10-fold GridSearchCV for each model (e.g. decision tree depth, SVM boundary conditions and kernel use, KNN neighbor number, penalizing complexity in logistic regression).
- **Evaluation**: 5-fold CV metrics, confusion matrices, precision-recall vs. “always-land” baseline.

Executive Summary – Key results and take-aways

Baseline vs. Models:
- "Always-land" was 66.7% accuracy
- SVM: 86.4% (~20% lift)
- Decision Tree: 84.8 % (+18% lift; full interpretability)

Other Models:
- KNN*: ~82% (best at k = X, Manhattan distance)
- Logistic Regression.: ~81% accuracy; L2 regularization, lbgs* used.

Economic Impact:
- Each recovered booster ~$25 to $40 M saved per flight (if pre-refurbished).
- Enables up to 25% price discounts while preserving margin, allowing for investment in R&D.

Data & Modeling Gaps:
- No live weather or telemetry inputs yet, presently resulting in a "signal ceiling".
- Public/scraped data noise; needs cleanup when real data becomes available.
- No operational feedback loop yet, would need to account for drift risk.

Next Steps:
- Deploy pruned decision-tree API as v1.
- Integrate real-time launch telemetry.
- Retrain monthly; monitor performance vs. baseline.
- **More detailed info on models and conclusion on slides 48-53.**

***lbfgs**: Stands for **Limited-memory Broyden–Fletcher–Goldfarb–Shanno**, a popular optimization algorithm used to efficiently find the best model parameters.
The KNN model did best when it looked at the **seven most similar past launches to decide landing outcome, measuring similarity by simply adding up how much each characteristic differed.

Introduction

- Project background and context:
  - Fictitious company wants to compete with SpaceX
  - Need to be able to compete based on launch price
  - Being able to predict launch price is highly important…
  - … not a simple problem – it has to do with a few different independent research problems

- Research problems we are trying to solve to help predict price (using machine learning)
  - Use public data to predicting whether first stage will land (to be re-used again); a gating factor for lower launch price
  - Quantify the impact of rocket stage recovery on rocket launch costs
  - Forecast whether first stage can be re-used
  - Assess if machine learning predictions could be useful to support decisions about launch pricing

Section 1

Methodology

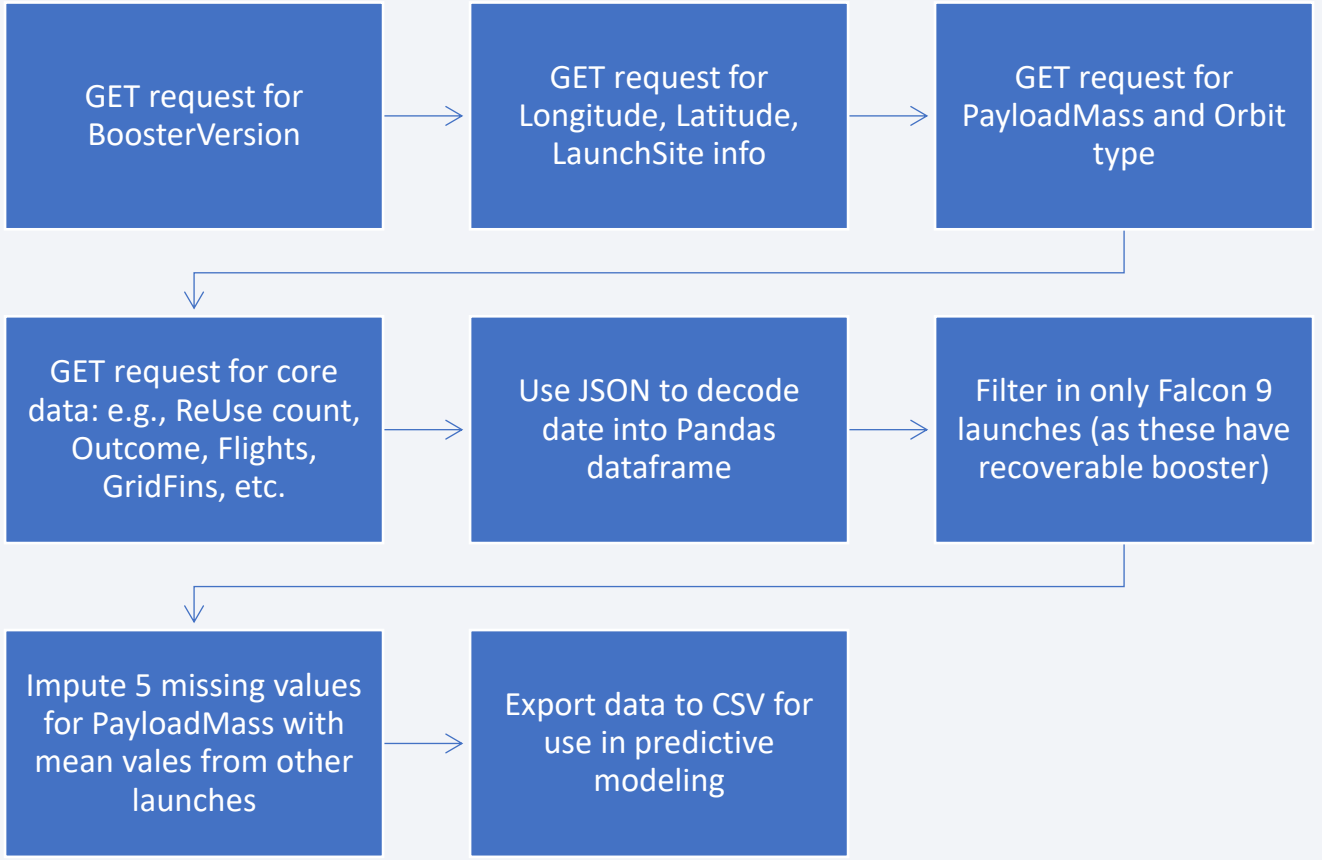
Methodology

- Data collection methodology:
  - SpaceX REST API for info regarding launches, rocket type, payload delivered, launch and landing specs and outcome
  - Scraped public Falcon 9 launch records from Wikipedia pages via Python Beautiful Soup
- Perform data wrangling
  - Parsed data from html tables
  - Transformed and filtered raw data collected into data that we could use (i.e., Falcon 9)
  - Imputed null values with means for payload mass
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Built, tuned, evaluated classification models
  - Classification models used: Support Vector Model (SVM), K Nearest Neighbors (KNN), Logistic Regression, Decision Tree

Data Collection – SpaceX API

GitHub link to code: https://github.com/whitingjim/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api-v2.ipynb

- Data collection from SpaceX API via REST calls to get needed info
- Examples of retrieved variables:
  - FlightNumber, Date, BoosterVersion
  - PayloadMass, Orbit, Launchsite, Legs
  - GridFins, LandingPad, Latitude, Longitude, ReusedCount, etc…
- Five (5) missing values for PayloadMass replaced with mean values (for sake of modeling modeling)

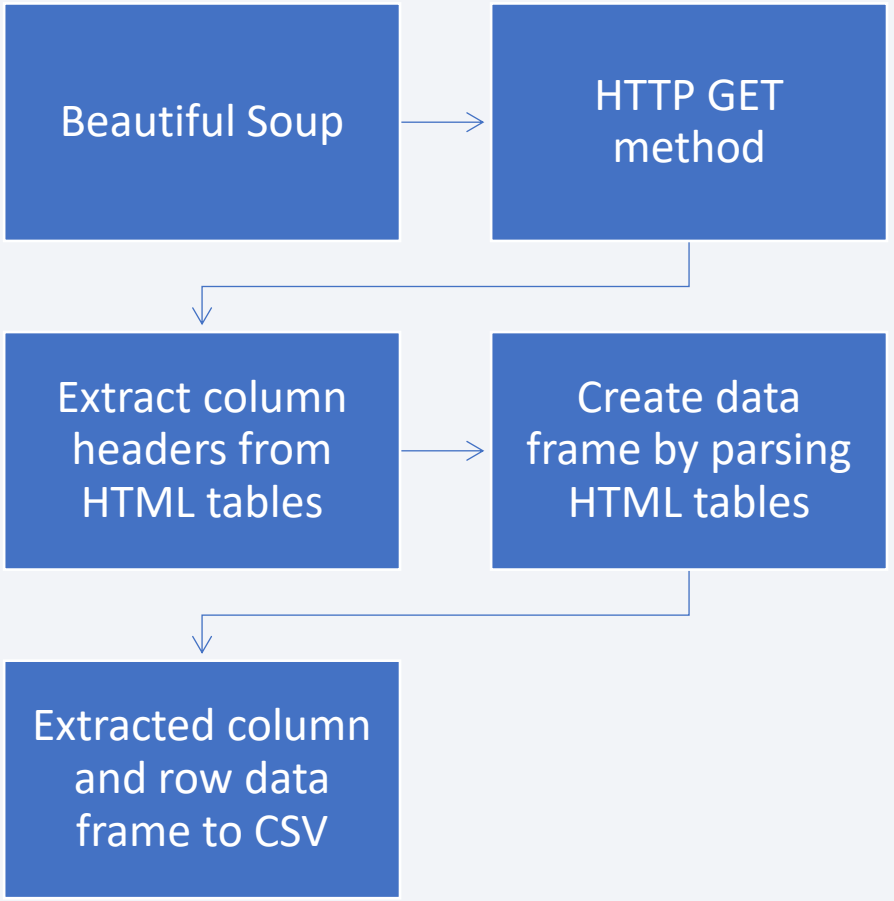


SpaceX API URL: spacex_url=https://api.spacexdata.com/v4/launches/past
https://api.spacexdata.com/v4/launches/past

Data Collection – Web Scraping

GitHub link to code: https://github.com/whitingjim/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb

- Beautiful Soup (BS) used to retrieve data via HTTP GET data to BS tables
- Extract column and row data from HTML tables into BS Object, and then data frame
- Data extracted to CSV for further use
- Variables:
  - Flight No., Launch site, Payload, Payload mass
  - Orbit, Customer, Launch Outcome, Version Booster, Booster landing, Date, Time

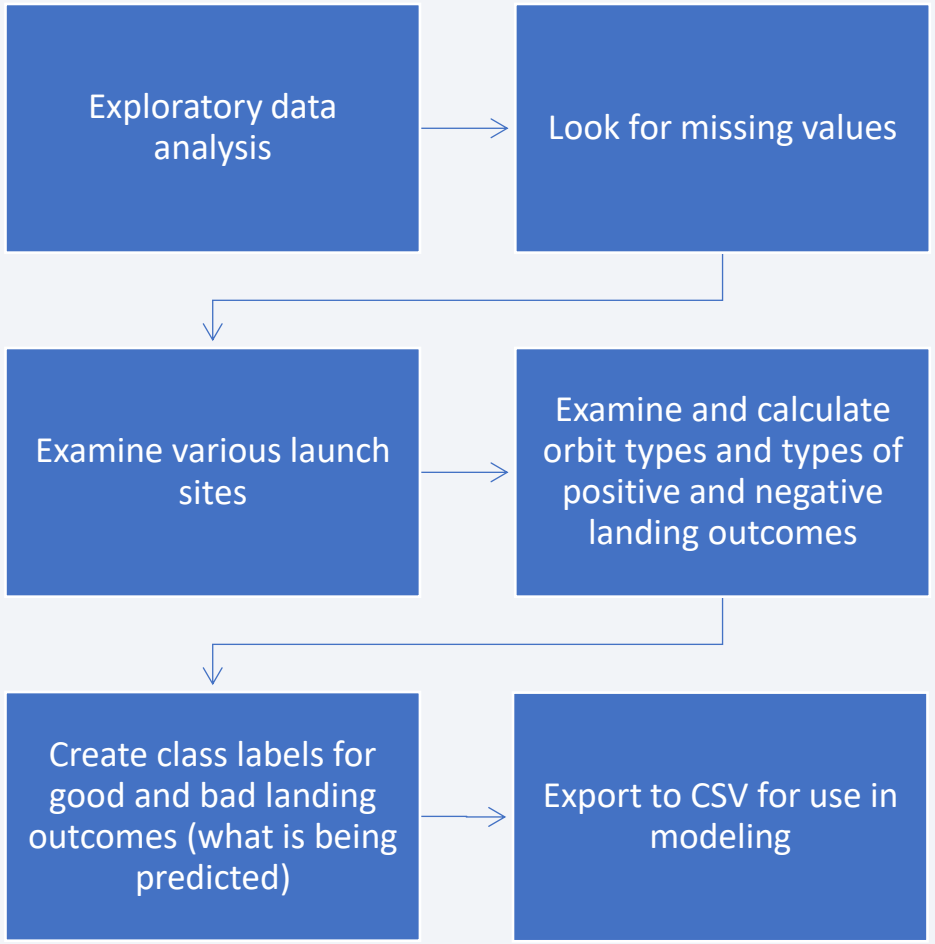


Wikipedia Data source: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Data Wrangling

GitHub link to code: https://github.com/whitingjim/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling-v2.ipynb

- Performed exploratory data analysis
- Missing values analysis, reach deeper understanding of data characteristics like:
  - Launch sites
  - Desired orbit types for satellite delivery
  - Landing outcomes (successful or unsuccessful)
- Created class labels columns for sake of prediction (successful or unsuccessful landing)
- Calculated landing success rate of 66.67%, used as baseline for predictive model lift comparisons.



EDA with Data Visualization

GitHub link to code: https://github.com/whitingjim/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-dataviz-v2.ipynb

Summary of charts plotted	Why visualization chosen
• Category (scatter) plot of payload mass vs flight number attempts vs class variable (successful or unsuccessful landing)	• Understand relationship, if any, between payload and volume of flights vs successful and unsuccessful landings
• Category (scatter) launch site and flight number attempts vs class variable (successful or unsuccessful landing)	• Understand the relationship, if any, between launch site and class variable (successful vs unsuccessful landing)
• Category (scatter) plot of payload mass vs launch site vs class variable (successful or unsuccessful landing)	• Understand the relations, if any, between payload mass, launch site, and class variable (successful vs unsuccessful landing)
• Bar chart to examine relationship between landing success rate (class variable) and orbit type	• Visual check for success rate for landing (class variable) for each orbit type
• Category (scatter) plot between flight number (volume), orbit type and landing success rate (class variable)	• Examine if there is a relationship between Flight number and orbit type
• Category (scatter) plot between payload mass, orbit type and landing success rate (class variable)	• Examine relationship, if any, between payload mass and orbit type on the class variable (landing success rate)
• Line and bar charts between year of launch and average annual success rate for successful landing (class variable)	• Examine the average successful launch rate trend over years of time

EDA with SQL -- Summary

GitHub link to code: https://github.com/whitingjim/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

- Displayed names of unique launch sites for space missions
- Displayed only launch sites that began with prefix of 'CCA'
- Displayed aggregated mass of payloads across all NASA launches
- Displayed aggregated mass of payloads, by [for each] customer
- Displayed average payload mass across Falcon 9 version 1.1 variant
- Displayed first successful landing date when booster returned to a ground pad (as compared to ocean, or ocean drone pad)
- Displayed names of boosters which have successfully landed on drone ships that have a pay load mass greater than 4000 KG and less than 6000 KG
- Listed total number of successful and failure mission outcomes
- Listed all booster versions that have carried the maximum payload mass
- Listed month name (sequential number), failed landing outcome, booster version, and launch site during the year of 2015
- Listed count, in descending order, of landing type outcomes between June 4, 2010 and March 20, 2017.

Folium – interactive map features

GitHub link to code: https://github.com/whitingjim/Applied-Data-Science-Capstone/blob/main/lab-jupyter-launch-site-location-v2.ipynb

- Folium used to create interactive map using the following objects:
  - Markers – used to indicate where there were successful (green) and unsuccessful (red) launches
  - Circles and text labels used for identifying NASA Space Center, launch site names and space missions (along with their names) throughout the USA
  - **Clusters** of circles and markers which allowed “zoom-in and zoom-out” were used allow easier identification of several launch sites that shared close proximity, and indicated launch outcomes
  - Lines used in Florida to indicate measurement of launch site proximities to nearby city, highways, and coast.

Plotly Dash: an overview of the graphs

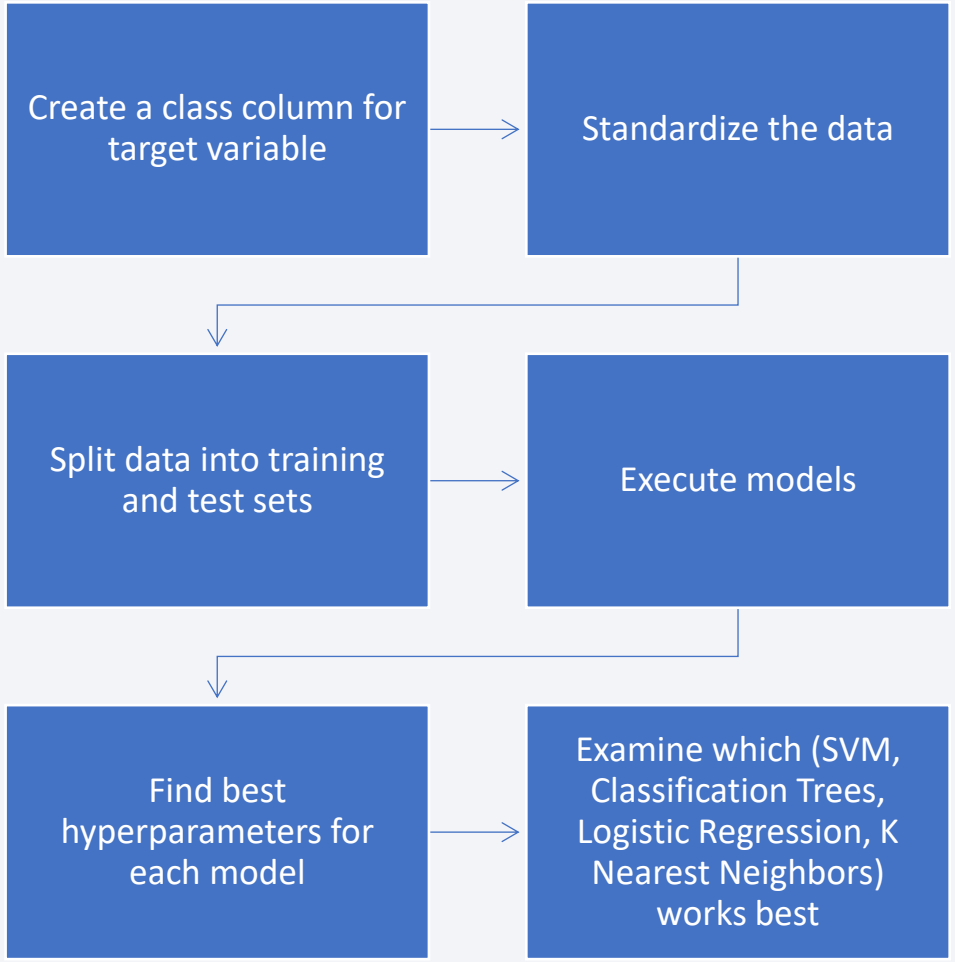
- Dashboard created using Plotly Dash using the following visualizations:
  - Pie graph indicating percentage of total launch successes by launch site
  - Scatter plot which shows payload mass by launch success or failure, with color coded dots for booster version category
- These plots, together, allow one to examine, isolate and view interactions by launch success by site, weight of payload, and booster version

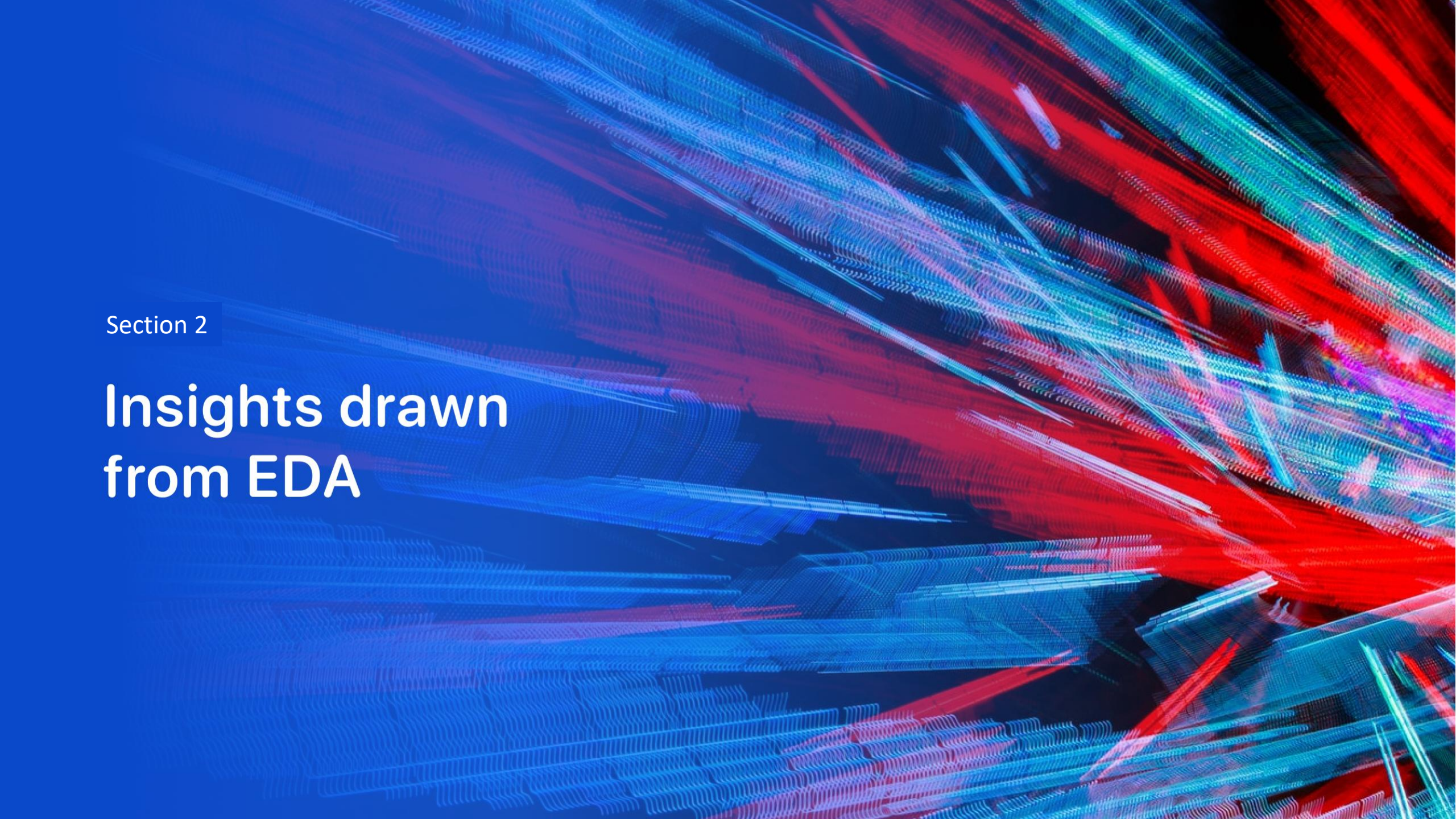
GitHub link to code: https://github.com/whitingjim/Applied-Data-Science-Capstone/blob/main/Applied%20Data%20Science%20Capstone%20Hands-on%20Lab%2C%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.ipynb

Predictive Analysis (Classification)

GitHub link to code: https://github.com/whitingjim/Applied-Data-Science-Capstone/blob/main/SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb

- Following models used: SVM, Classification Trees (CART), Logistic Regression, K Nearest Neighbors (KNN)
- Class (Y) variable created with binary outcomes (successful or unsuccessful) using 1 or 0
- Standardize predictor variables (X) via preprocessing.StandardScaler
- Use train_test_split method to split X and Y data into training and test data sets (80% train, 20% test)
- Create model object for each model and GridSearch CV object to find optimal parameters (i.e., model settings)
- Calculate accuracy of model, consider confusion matrix as to how well model performs (true positive, false negatives, etc.)




Section 2
Insights drawn from EDA

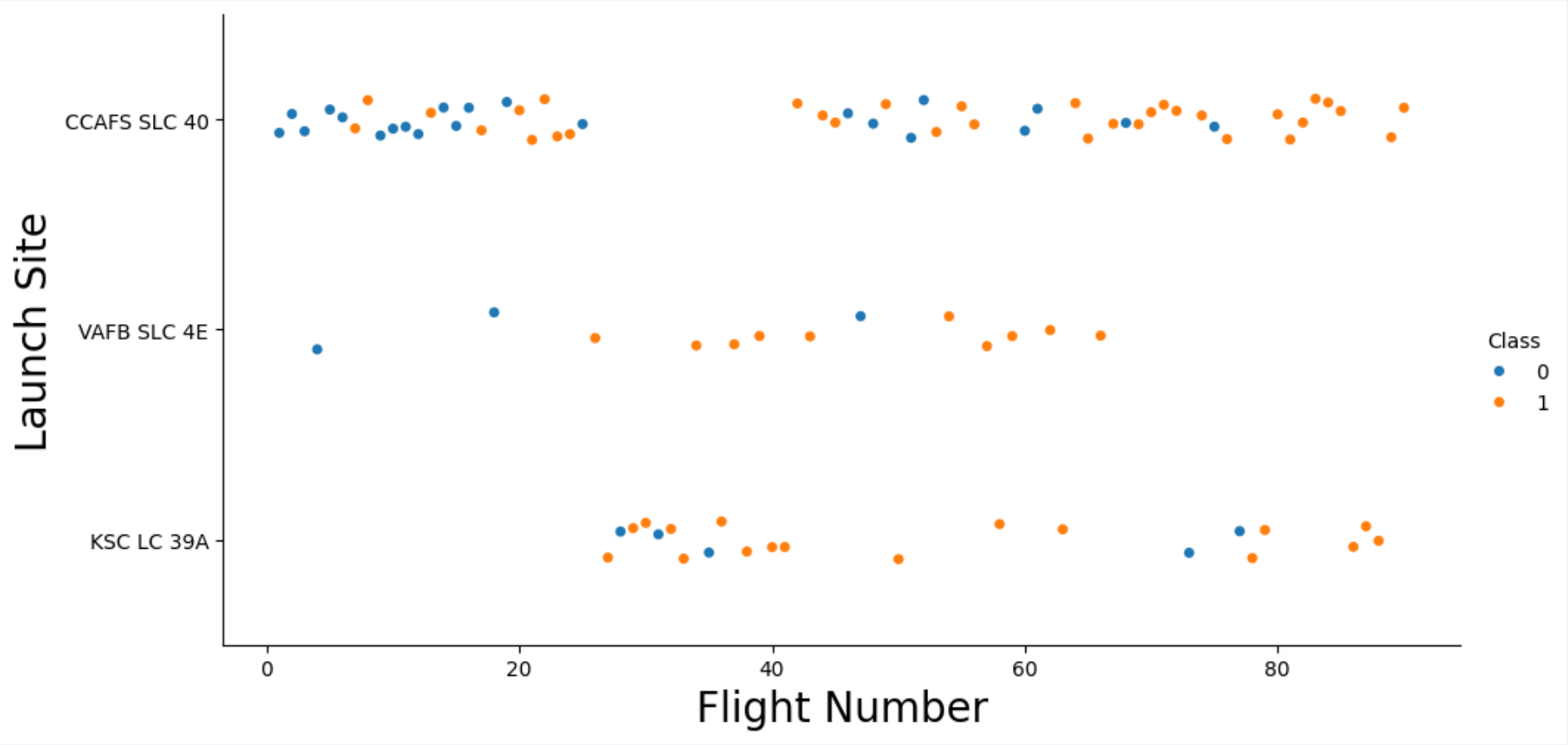
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Launch site acronyms

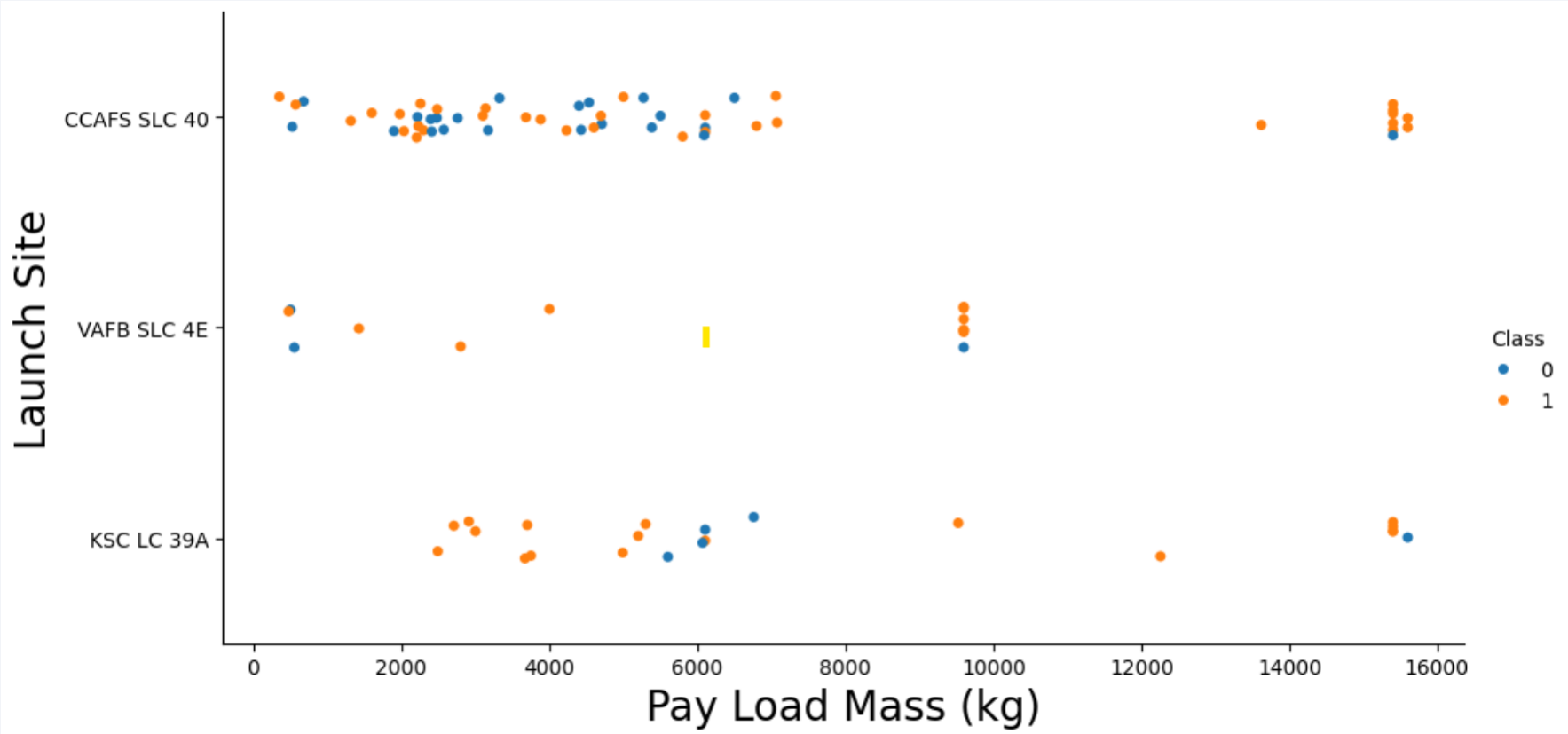
Acronym	Site Name	Location	Reference
CCAFS SLC 40	Cape Canaveral Air [Space] Force Station	Cape Canaveral, Florida	https://en.wikipedia.org/wiki/Cape_Canaveral_Space_Force_Station
KSC LC 39A	Kennedy Space Center	Merritt Island, Florida	https://en.wikipedia.org/wiki/Kennedy_Space_Center
VAFB SLC 4E	Vandenberg Air [Space] Force Base	Santa Barbara County, California	https://en.wikipedia.org/wiki/Vandenberg_Space_Force_Base

Flight Number vs. Launch Site



- High density of early launches, and later, at Cape Canaveral
- Cape Canaveral seems to have been a “test bed” for launches for attempted landings, after which more missions at Vandenberg and Kennedy Space Center
- Most launches at Cape Canaveral, followed by Kennedy Space Center

Payload vs. Launch Site



- Strong concentration of early lower-mass-payload launches, and successful and unsuccessful landing testing, occurred at Cape Canaveral
- Mid-level payload mass launches and Vandenberg and Kennedy Space Center
- High mass launches tended to be more at Cape Canaveral and Kennedy Space Center

Primer for Orbit Types

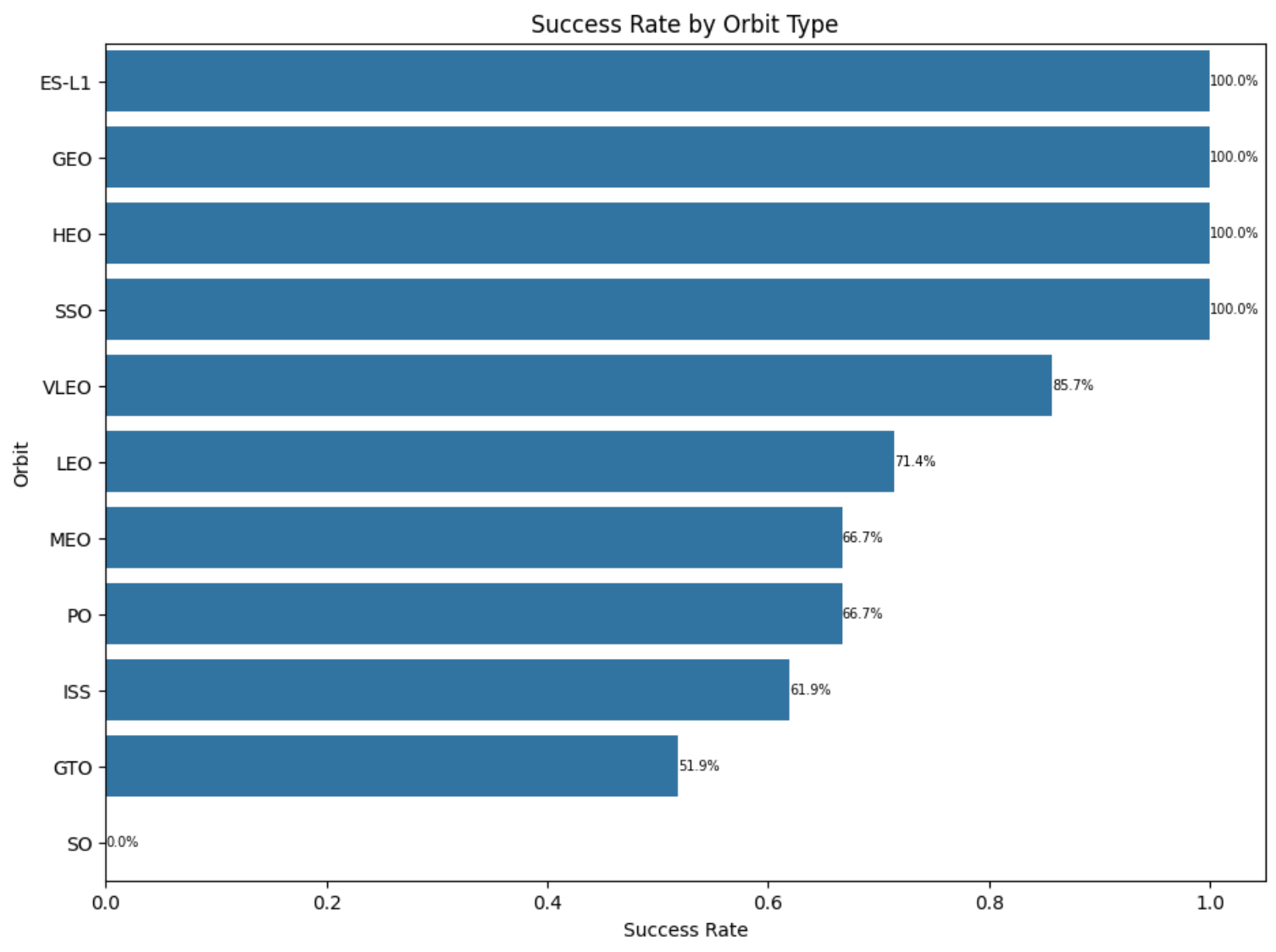
Orbit Type	Typical Altitude	Notes
ES-L1 (Earth-Sun Lagrange Point 1)	~1,500,000 km	Gravitational balance point between Earth and Sun
HEO (Highly Elliptical Orbit)	~500 km to 40,000+ km	Highly stretched, varies greatly during orbit
GEO (Geostationary Earth Orbit)	~35,786 km	Fixed over one spot at the equator
GTO (Geostationary Transfer Orbit)	~200 km to 35,786 km	Elliptical transfer orbit toward GEO
MEO (Medium Earth Orbit)	~2,000 km to 35,786 km	Includes GPS satellites (~20,200 km)
SSO (Sun-Synchronous Orbit)	~600–800 km	A type of polar LEO
PO (Polar Orbit)	~200–1,000 km	Polar LEO range
LEO (Low Earth Orbit)	~160–2,000 km	Broad range; many satellites operate here
ISS (International Space Station Orbit)	~400 km	Specific orbit within LEO
VLEO (Very Low Earth Orbit)	~160–300 km	Just barely above atmosphere
SO (Surface Orbit)	~0 km (hypothetical)	Would skim the surface; not practically possible

Primary Sources:

NASA Earth Observatory. (n.d.). *Catalog of Earth Satellite Orbits*. Accessible at https://earthobservatory.nasa.gov/features/OrbitsCatalog

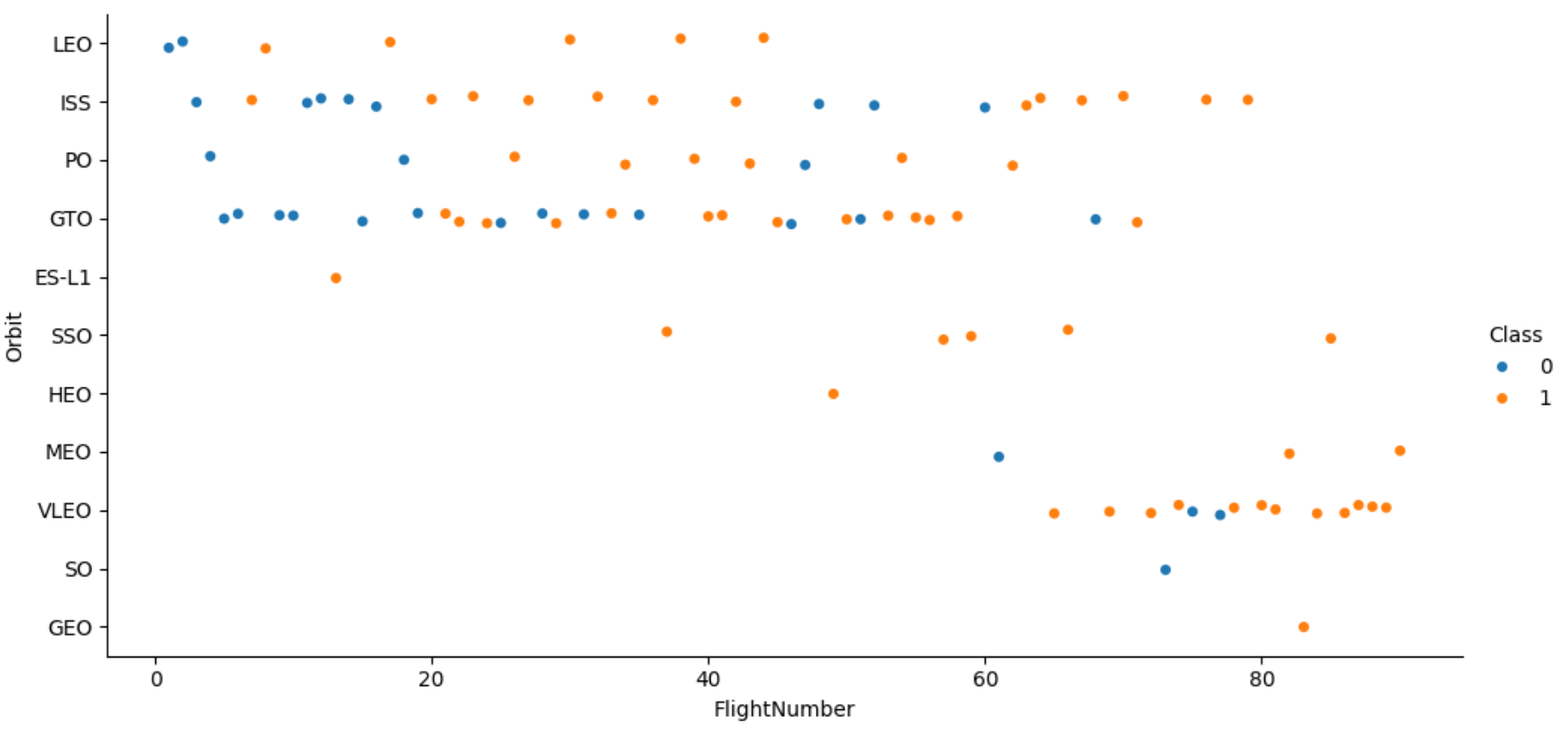
European Space Agency (ESA)
European Space Agency. (n.d.). *Types of Orbits*. Available at https://www.esa.int/Enabling_Support/Space_Transportation/Types_of_orbits

Success Rate vs. Orbit Type



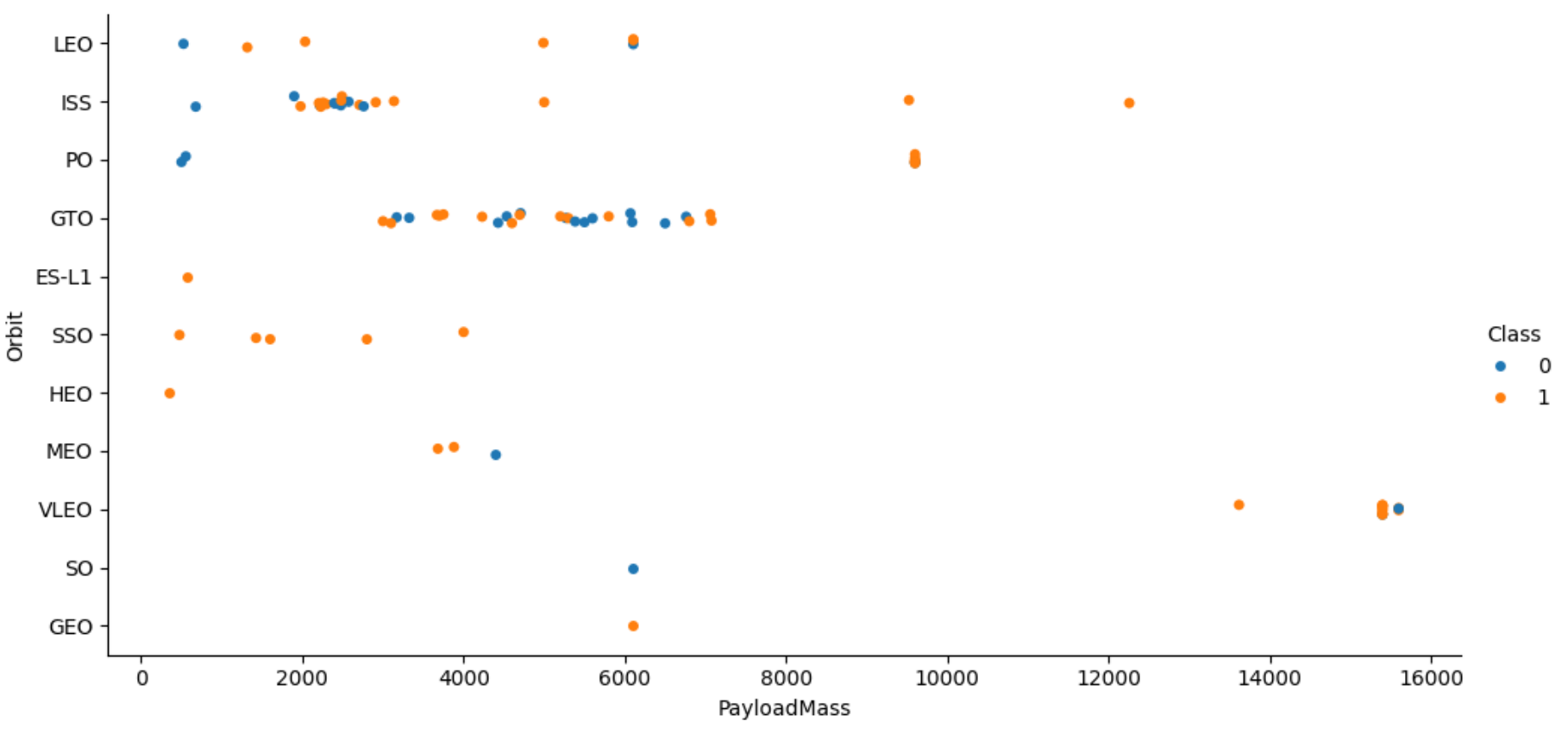
- Highest success rates with ES-L1 (high altitude), Geo-Stationary (over equator), Highly Elliptical (widely varying altitude) and Sun Synchronous orbits
- Lowest success rates (early on) were International Space Station and Geo-Stationary Transfer orbits

Flight Number vs. Orbit Type



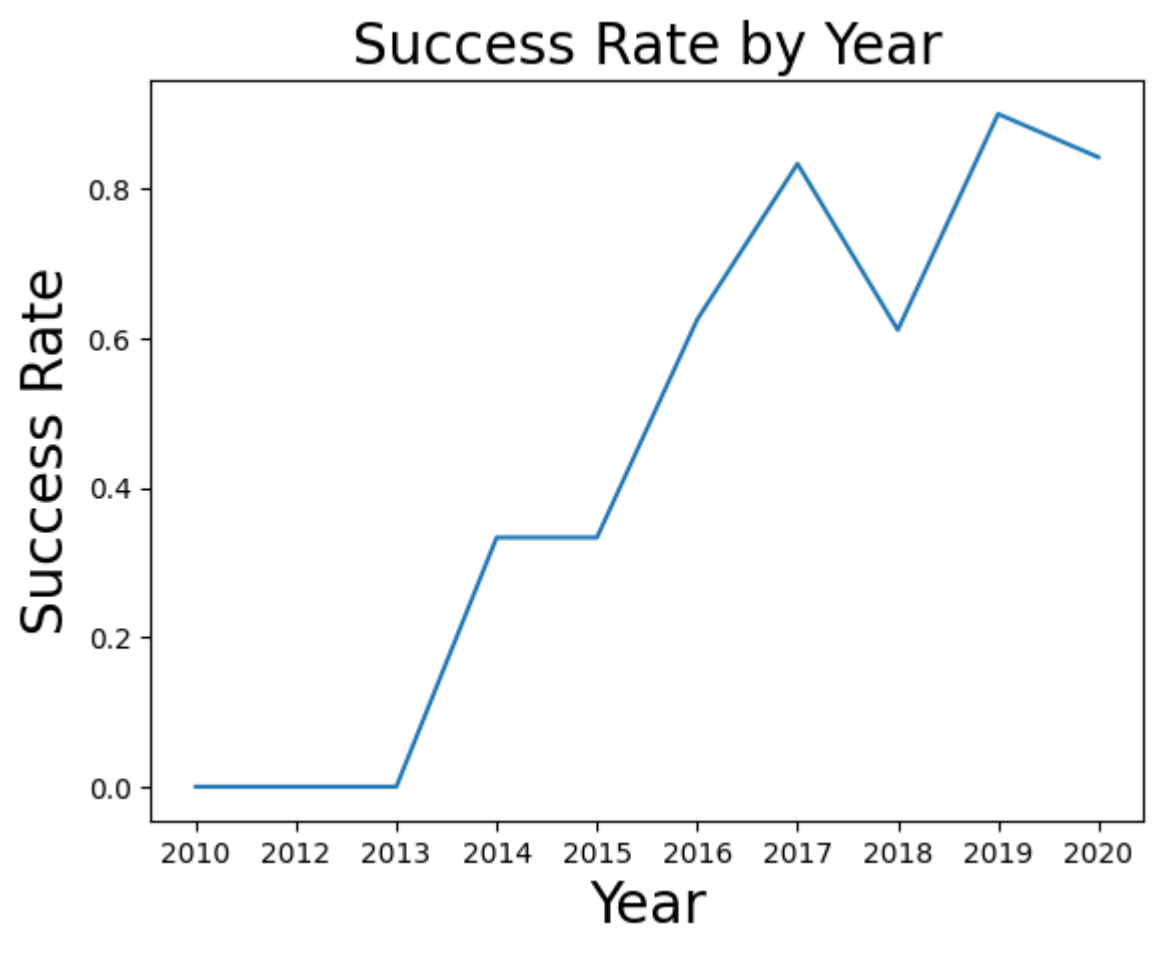
- At first glance, for Low Earth Orbit (LEO), it appears that successful landings are related to the number of flights (as success improves over flight number…
- This kind of pattern seems much less clear with Geostationary Transfer Orbit (GTO)

Payload vs. Orbit Type



- Most missions had a payload from 2000 to 7000 KG
- Heavier payloads appear to have a higher successful landing rate… although, these would have occurred after much testing with lower-mass payloads

Launch Success Yearly Trend



- For first 3 years, the overt aspects of landing success rate were very modest, but apparently lessons were being learned…
- … because there was a large (nearly 40%) step change in 2013
- Similar pattern emerged during 2015, with high (~40%, again) success rate gain from 2015-2017

All Launch Site Names

- All unique (non-duplicative) launch sites identify via SQL query
- SQL Magic, SELECT DISTINCT statement used to identify information
- Four sites observed below in the SQL output

```
# display the names of the unique Launch_Site
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

[13]

Pytho

```
 * 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'

- Assignment was to display 5 records where launch sites begin with `CCA`
- SQL Magic, all rows (*) , LIKE (to identify similar), and LIMIT number of records used

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

```
# display 5 records where Launch Site begins with the string 'CCA'
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 10;
```

Python

```
 * 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

Total Payload Mass

- Calculated total payload carried by boosters from NASA (CRS) via SQL, resulting in a total of 45,596 KG
- SQL Magic, SELECT the SUM of the Mass for NASA as a customer

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

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

Python

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculated the average payload mass carried by booster version Falcon 9 (F9) v1.1
- The average payload was 2928.4 KG for the Falcon 9 version 1.1
- Used SQL Magic and SQL for selecting for desired booster version, calculating average

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

```
# Display the average payload mass carried by booster version 'F9 v1.1'
%sql SELECT Booster_Version, AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1' GROUP BY Booster_Version;
```

23] Python

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

Booster_Version	AVG(PAYLOAD_MASS__KG_)
F9 v1.1	2928.4

First Successful Ground Landing Date

- First successful ground landing was on December 22, 2022
- Result using SQL Magic, to query (SELECT) the minimum date with a successful ground landing (as opposed to other successful landing types, e.g., at sea, on drone ship at sea, etc.).

```
# List the date when the first successful landing outcome in ground pad occurred
%sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)' AND Landing_Outcome IS NOT NULL;
```

[30] Python

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

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000 KG

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000, were:
- F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2
- SQL Magic, SELECT statement targeting “Success (drone ship)” and within-parameter Mass used to achieve result

```
[31]  # List the names of the boosters which have successs in drone ship landing and have a payload mass between 4000, but less than 6000 kg
      %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
                                                                                                                                  Python
 * 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

- Calculated total number of successful and failure mission outcomes
- Approximately 100 successful mission outcomes (as opposed to booster landing outcomes, which are different)
- SQL Magic, aggregation of count derived from Mission_Outcome variable

List the total number of successful and failure mission outcomes

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

[32] Python

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

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

Boosters Carried Maximum Payload

```
# List the names of the booster versions which have carried the maximum payload mass
%sql SELECT Booster_Version, MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE GROUP BY Booster_Version ORDER BY MAX(PAYLOAD_MASS__KG_) DESC LIMIT 15;
```

[26] ✓ 0.0s

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

Booster_Version	MAX(PAYLOAD_MASS__KG_)
F9 B5 B1060.3	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1056.4	15600
F9 B5 B1051.6	15600
F9 B5 B1051.4	15600
F9 B5 B1051.3	15600
F9 B5 B1049.7	15600
F9 B5 B1049.5	15600
F9 B5 B1049.4	15600
F9 B5 B1048.5	15600
F9 B5 B1048.4	15600
F9 B5 B1049.6	15440
F9 B5 B1059.3	15410
F9 B5 B1051.5	14932

- Listed names of boosters which have carried the maximum payload mass.
- Numerous (12) boosters observed that carried 15.6 KG payload.

2015 Launch Records

- Listed failed landing outcomes at drone ships, their booster versions, and launch site names for the year 2015 – there were a total of two (2) such landings.
- SQL Magic used, query calls for chronological numeric month names, booster versions and landing outcomes that pertain to failed landings on drone ships (not other places), AND the desired year.

```
# List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
%sql SELECT strftime('%m', Date) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE Landing_Outcome = 'Failure (drone ship)' AND Date LIKE '2015%' ORDER BY Month;
```

[23] ✓ 0.0s Python

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

Month	Landing_Outcome	Booster_Version	Launch_Site
01	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 (such as Failure (drone ship) or Success (ground pad)) between the dates 2010-06-04 and 2017-03-20, in descending order.
- During this period, the top three (3) landing outcomes were: No attempt [at landing], Success (drone ship), Failure (drone ship).

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

[24] ✓ 0.0s Python

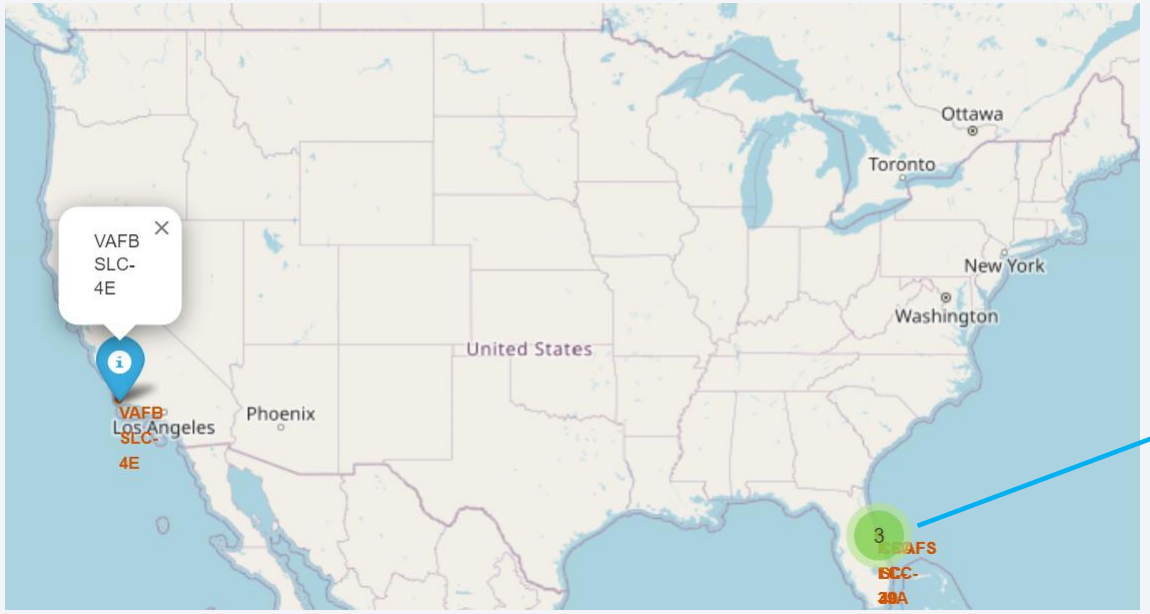
```
 * sqlite:///my_data1.db
Done.
```

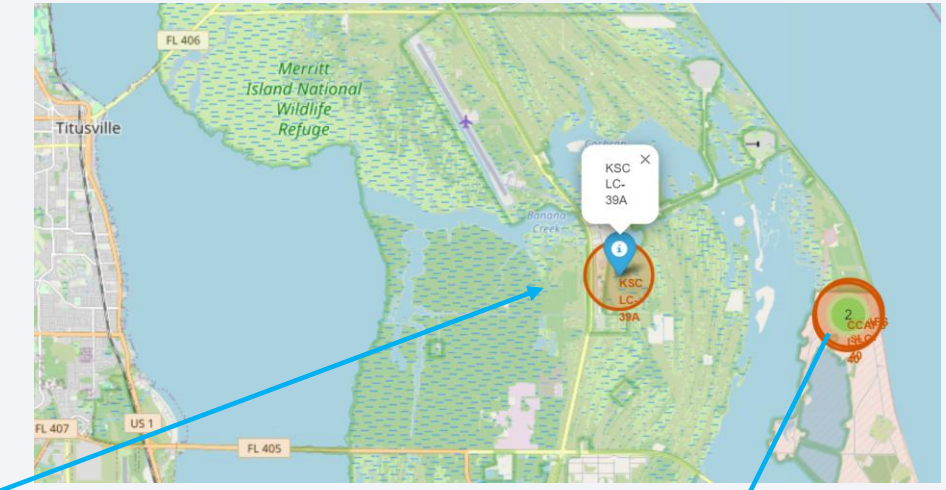
Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

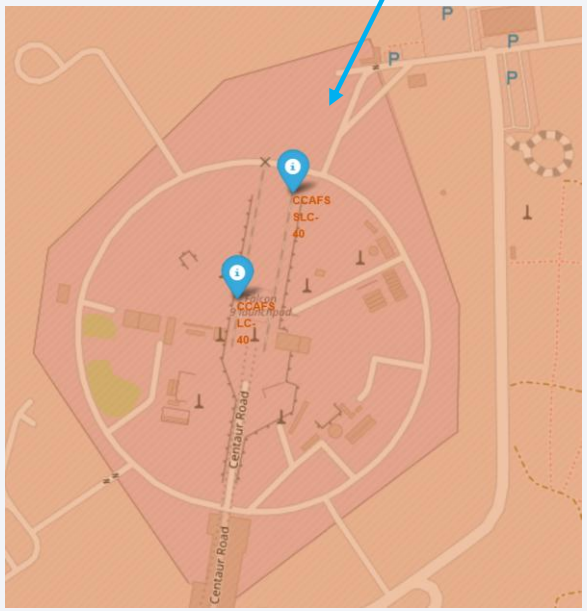
Section 3

Launch Sites Proximities Analysis

Folium map – including screen shots for all four launch sites



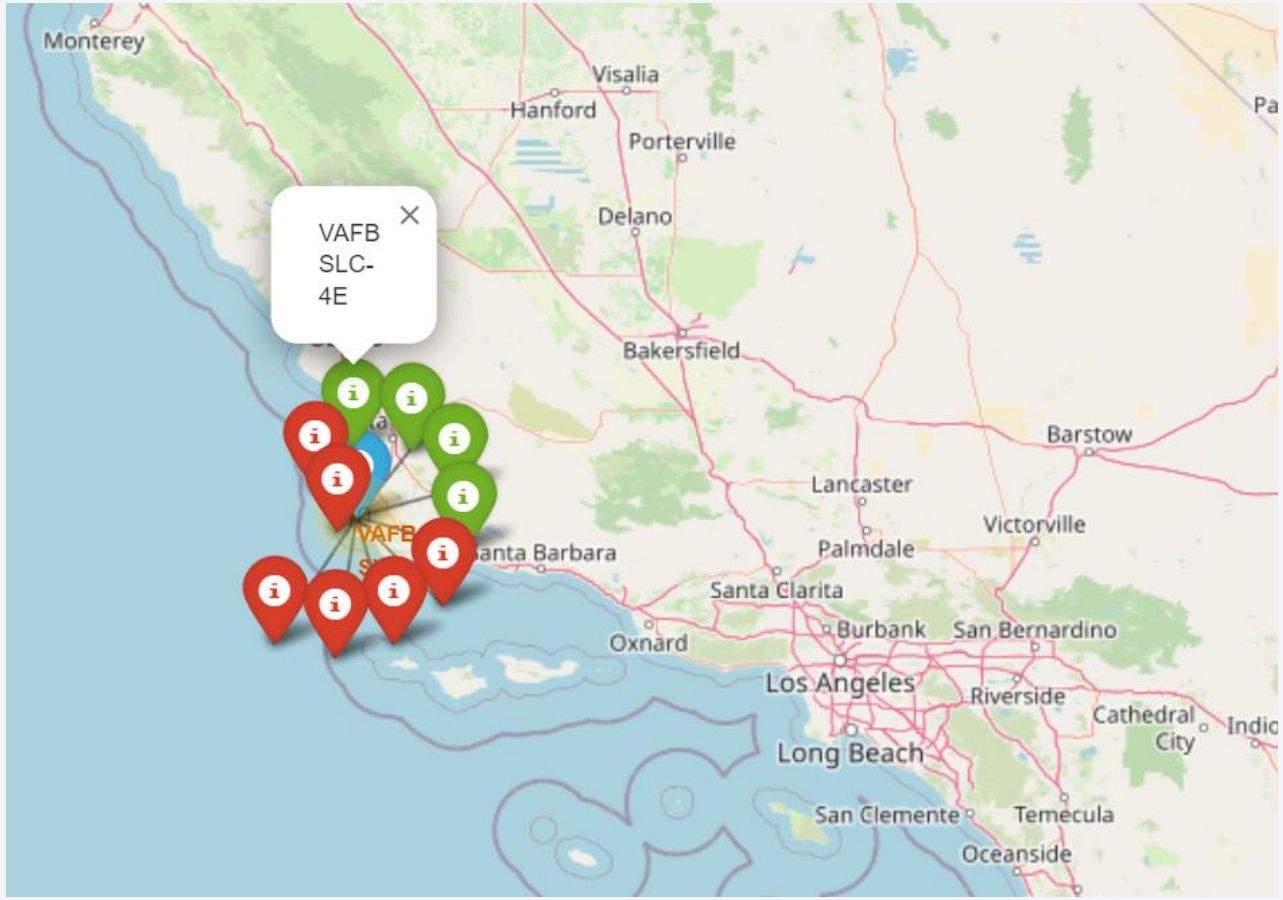




- Interactive map allows for exploration of launch sites.
- One site at VAFB in California, while other three are in Florida at the Kennedy Space Center and Cape Canaveral.

Folium Map - use of interactive markers to indicate landing success and failure

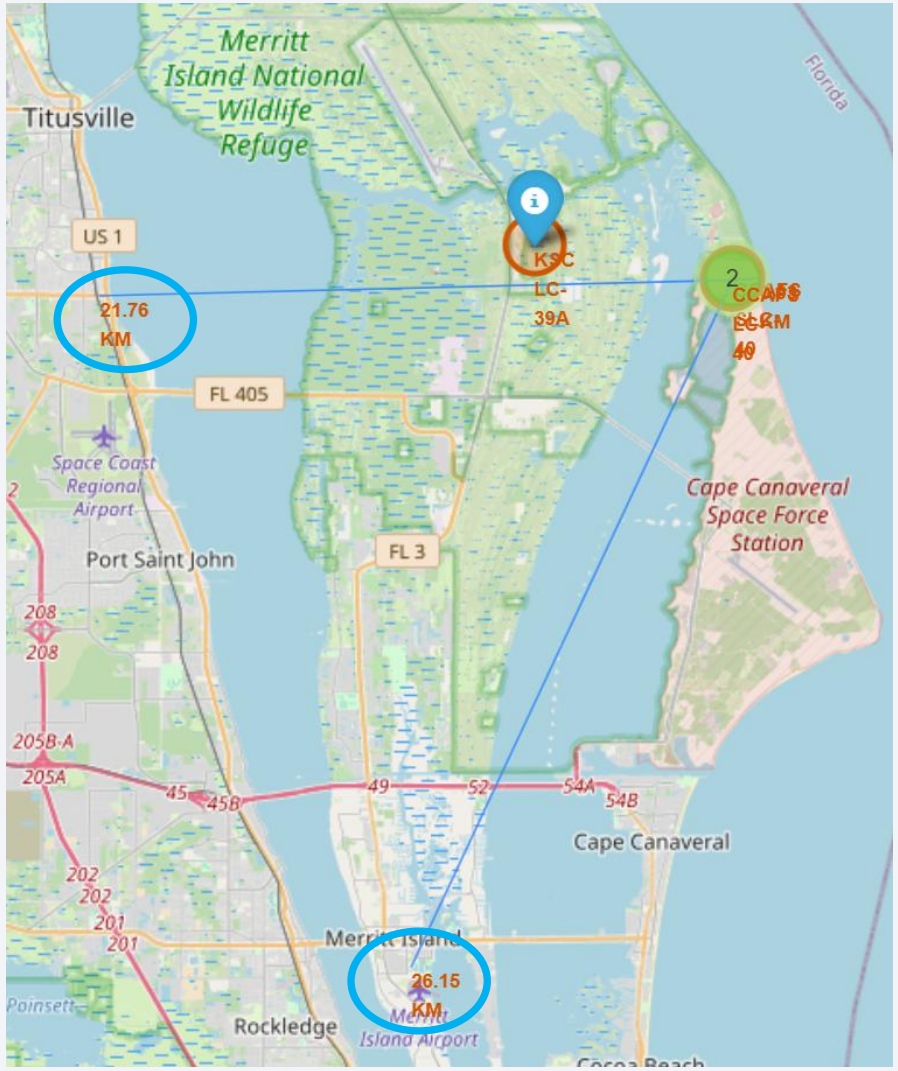
Example 1: Vandenberg Air Force [Space] Base (VAFB)



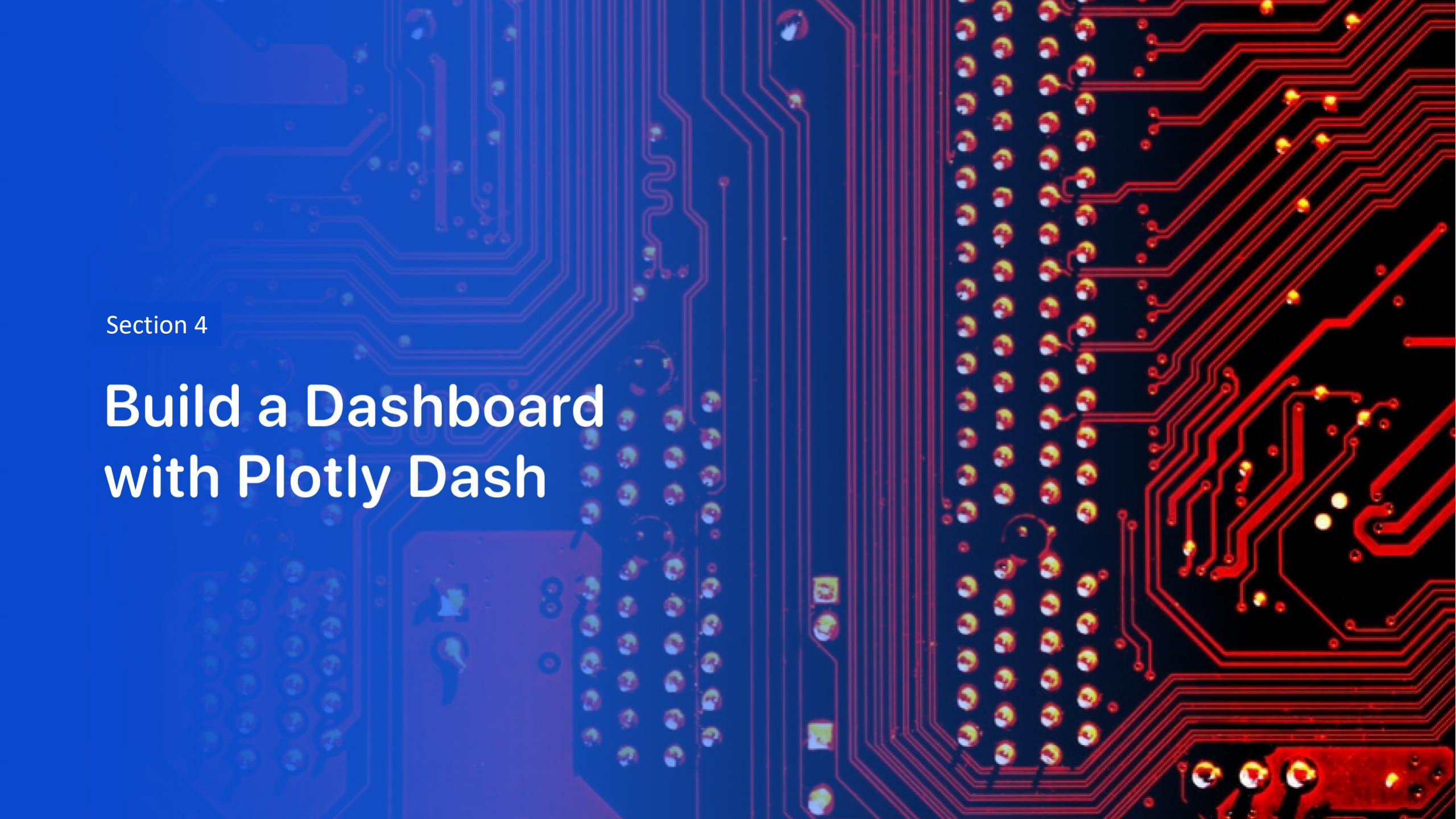
Example 2: Kennedy Space Center (KSC)

Interactive map contains clickable elements which show the launch pad where each successful (green) and unsuccessful (red) landing attempt was made

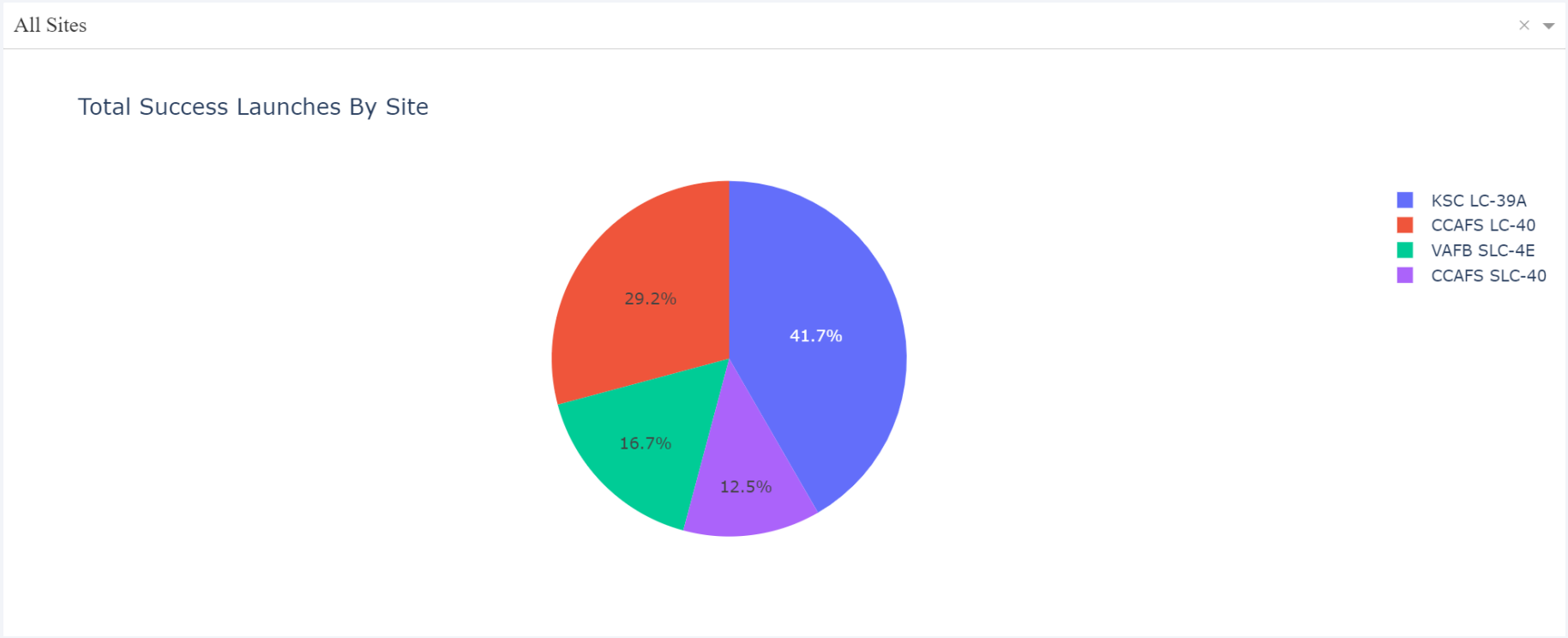
Folium map – demonstration of auto-calculation feature



- Interactive map includes lines that automatically calculate the distance to the nearest highway and airport
- Cape Canaveral approximately 22 KM (~13.5 miles) from nearest major highway…
- And, about 26 KM (~16 miles) from the nearest airport

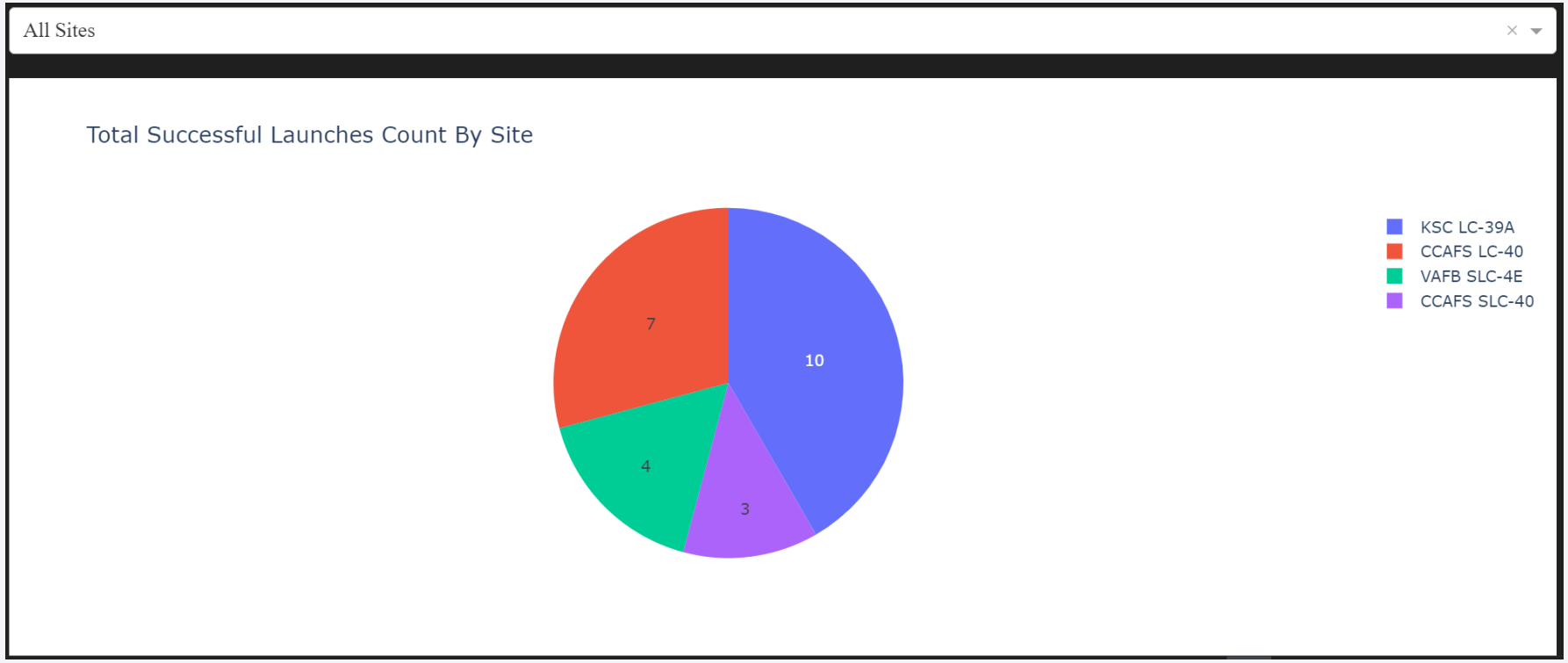

Section 4
Build a Dashboard with Plotly Dash

SpaceX Launch Dashboard – All Sites Success Rate



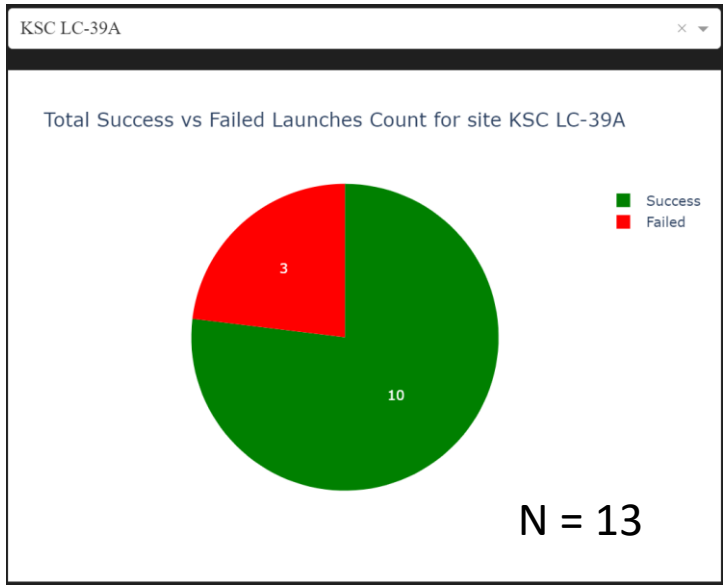
- Nearly 2/3 of the launch successful launch rate occurred at Kennedy Space Center LC-39A (41.7%) and Cape Canaveral LC-40 (29.2%)

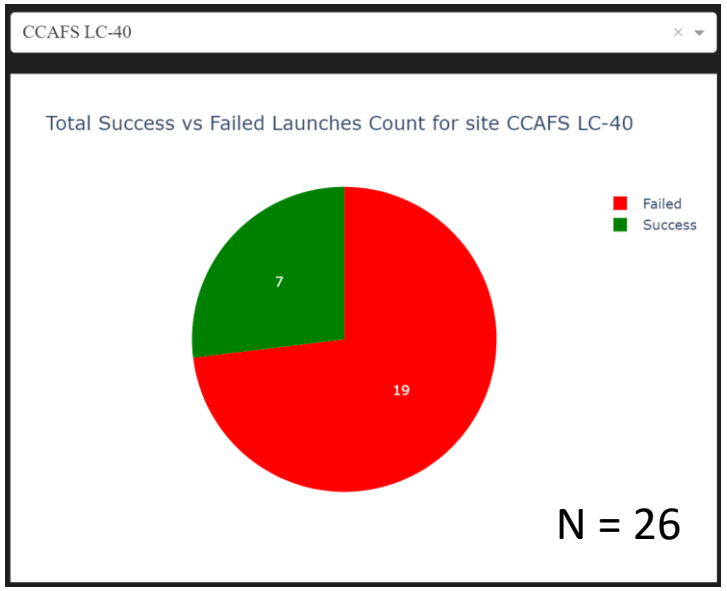
SpaceX Launch Dashboard – All Sites Success Counts

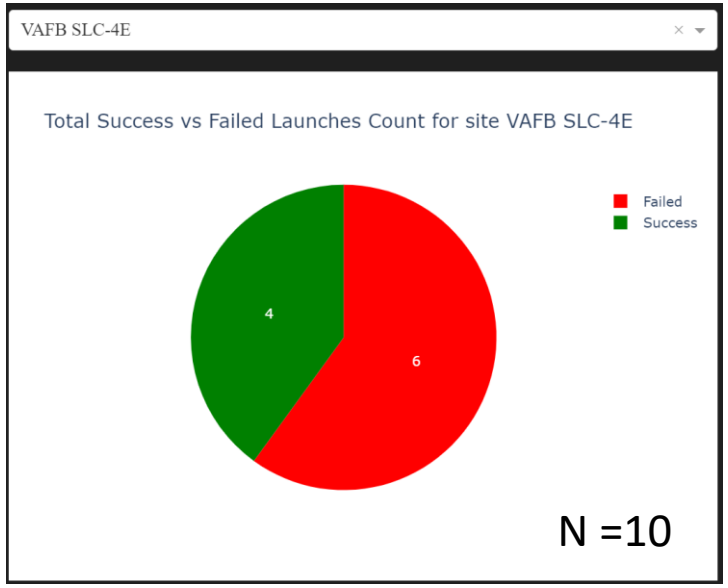


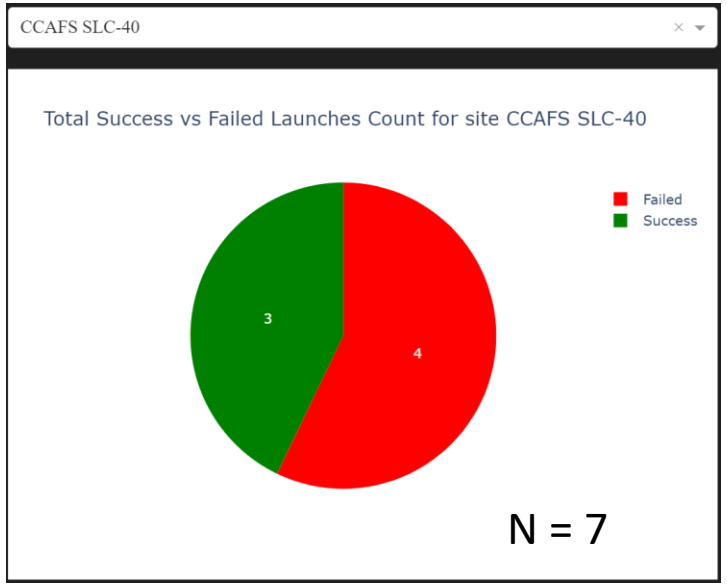
- Kennedy Space Center (KSC) site had the highest count (10) of successful launches
- … followed by Cape Canaveral site LC-40 with (7) successful launches

SpaceX Launch Dashboard – Success vs. Failed Counts by Launch Site



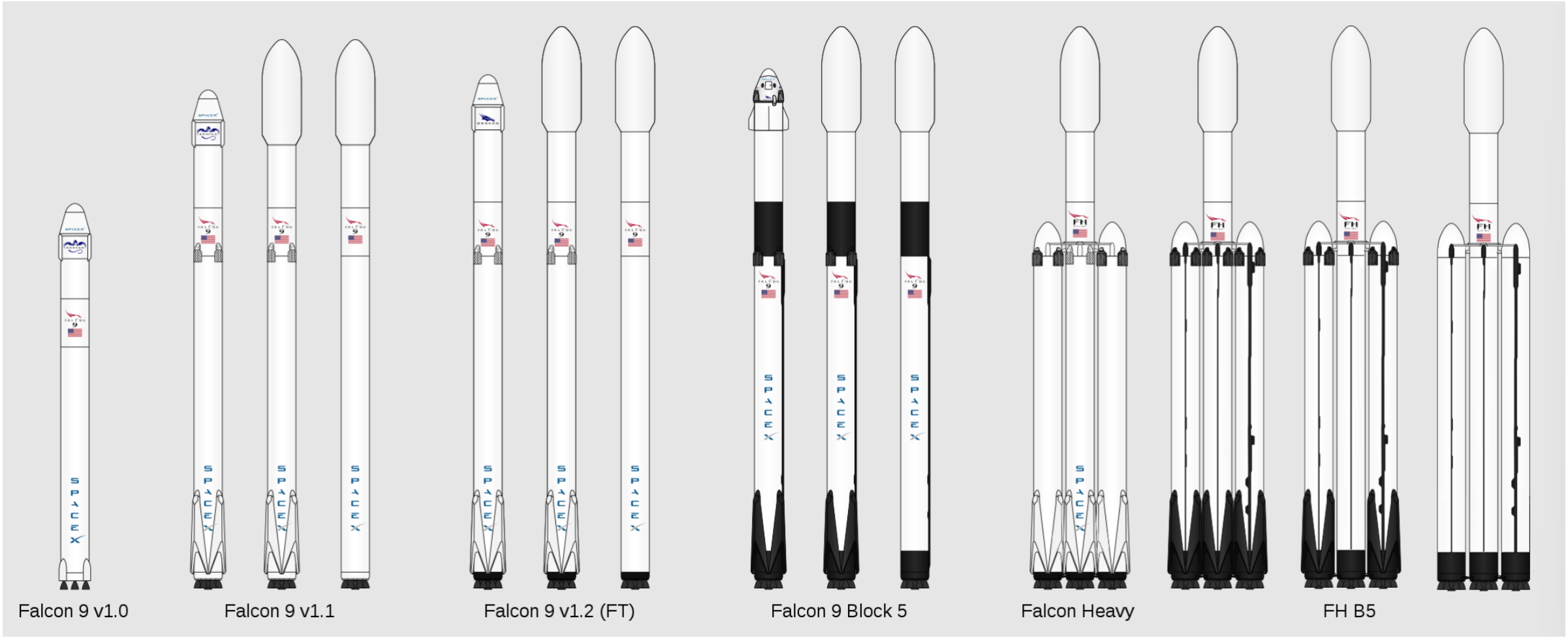






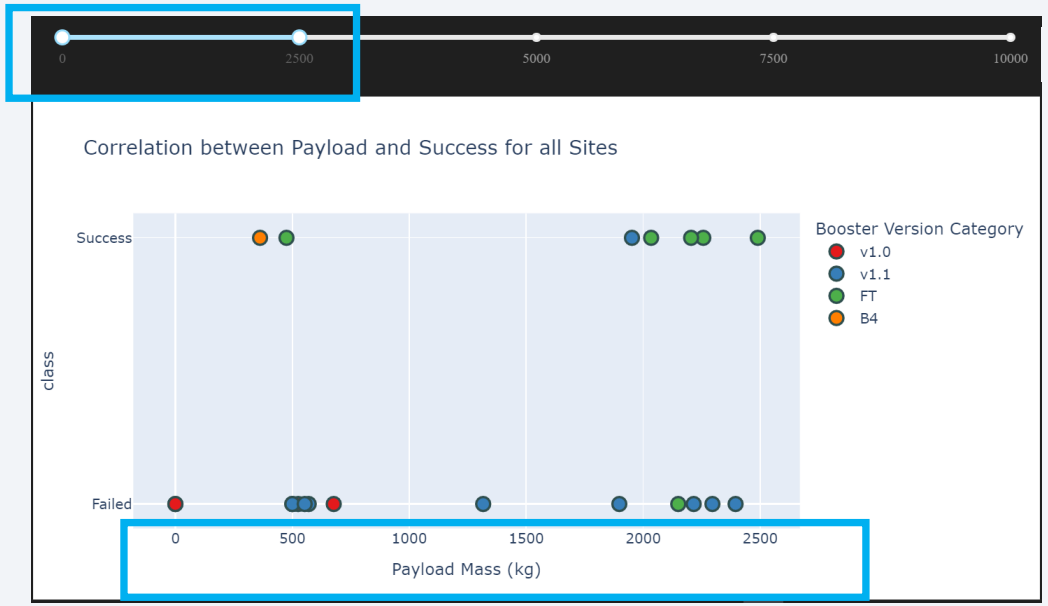
- Kennedy Space Center (KSC) site had highest success rate… **however, it had relatively fewer launches** (13)…
- … Cape Canaveral LC-40 site had most launches in the aggregate (26) – i.e., twice as many as Kennedy Space Center
- See the effect of booster versions on the success counts and rates in next few slides

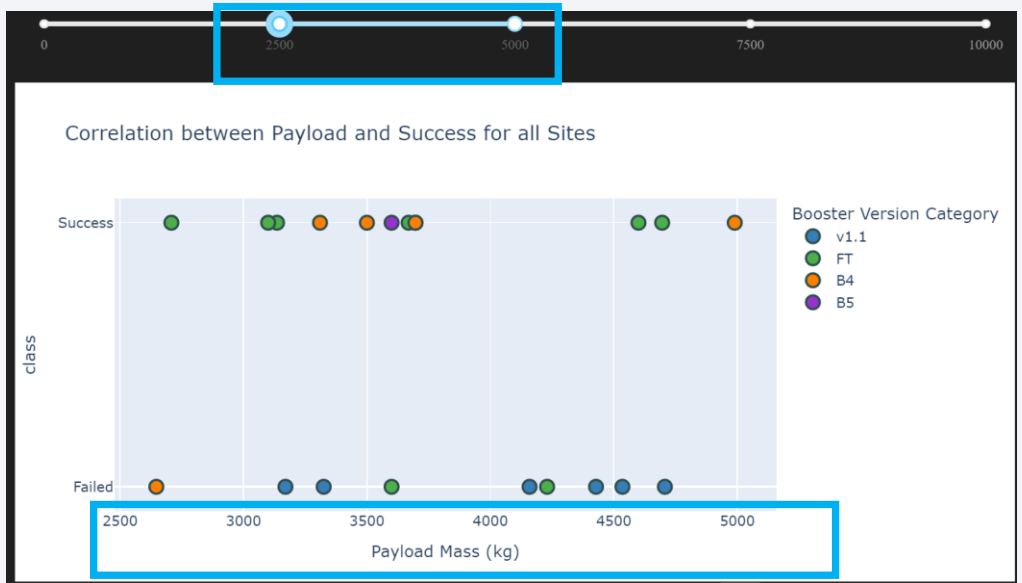
Evolution of Falcon 9 and Falcon Heavy rockets



https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches#/media/File:F9_and_Heavy_visu.png

Correlation between payload mass and booster version, across all sites (1/2)





- For payload masses from 0 to 2500 KG, the FT booster version was most successful…

- … And, when considering payload mass between 2500 and 5000 KG, the FT version continued to best, but only slightly better than B4 booster version.

https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Correlation between payload mass and booster version, across all sites (2/2)

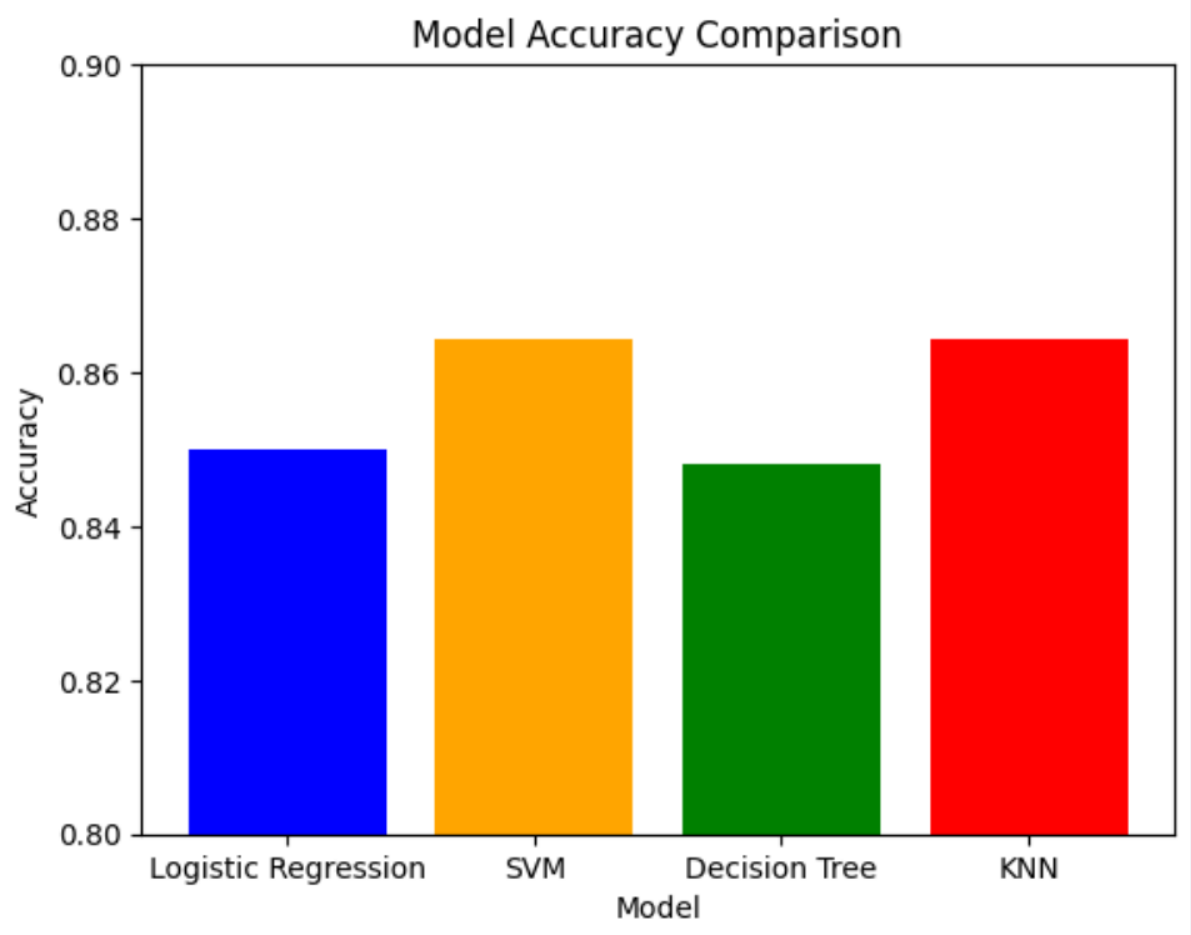
- The relative success of FT version continued, except at highest recorded mass of 9600 KG where the B4 version was successful

https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Section 5

Predictive Analysis (Classification)

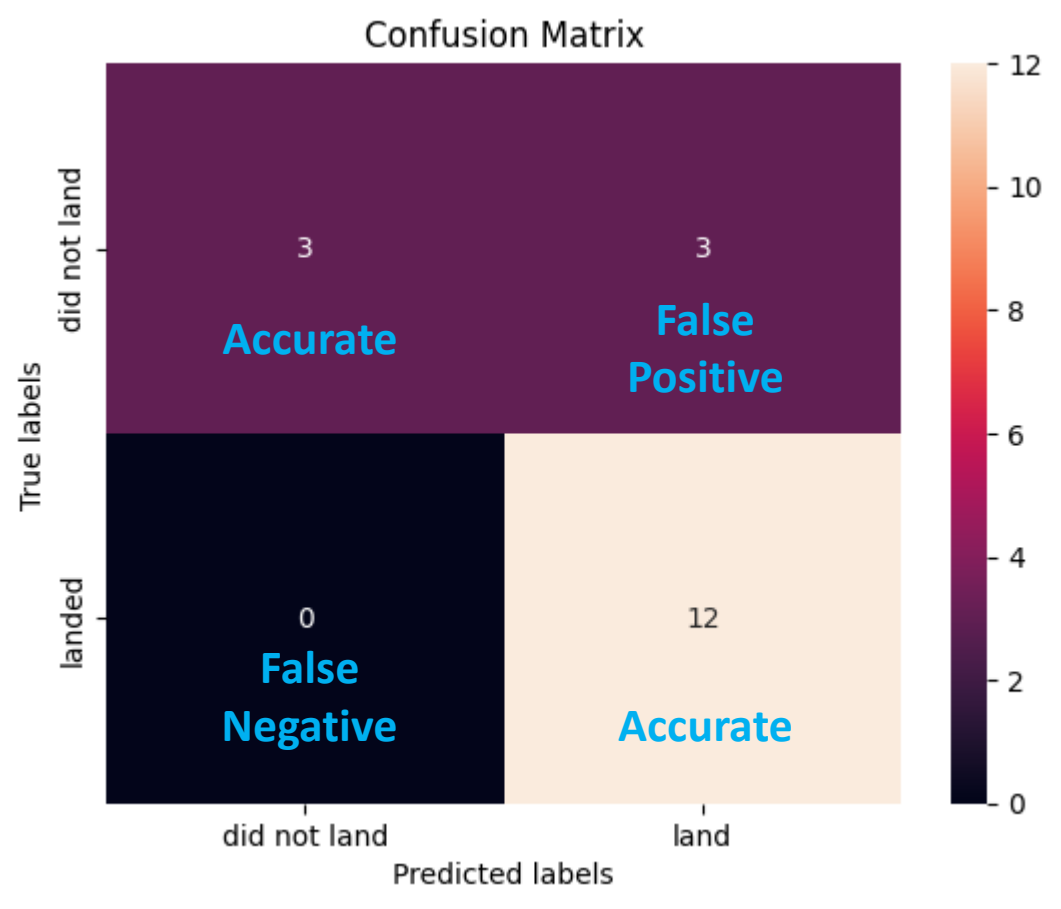
Classification Accuracy – predictive models have similar accuracy



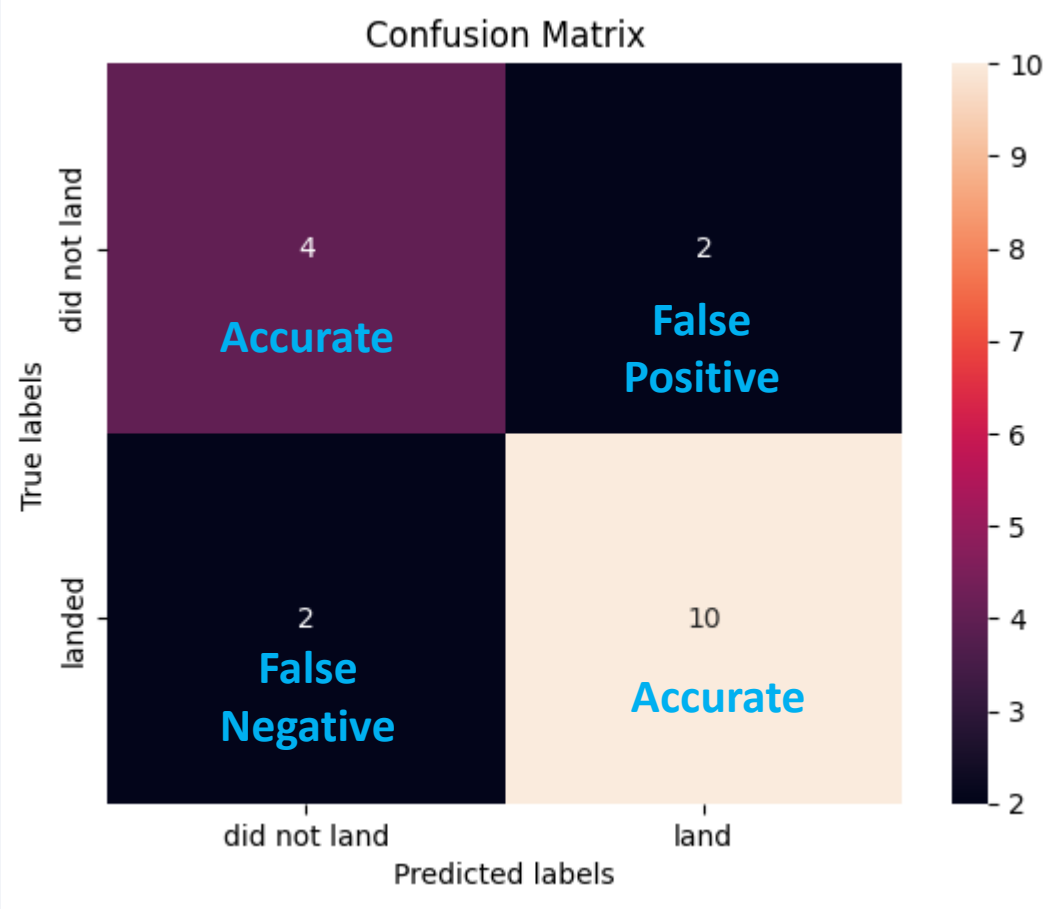
- Both Support vector model (SVM) and K Nearest Neighbors, were best performing classification model in terms of accuracy, each at ~86.4%
- Logistic regression (85.0% accuracy) and decision tree (84.8% accuracy) models performed similarly
- Besides accuracy, these models have tradeoffs in terms of what they do well, and not so well
- Stratification used to help ensure the same amount of target variable (land, did not land) was used in training and test sets
- GridSearchSVM used for tuning each model to ensure most optimal hyperparameters, for sake of increasing accuracy

Confusion Matrixes (1/2)

***Support Vector model (SVM) – 86.4% accuracy**



****K Nearest Neighbor (KNN) – 86.4 % accuracy**

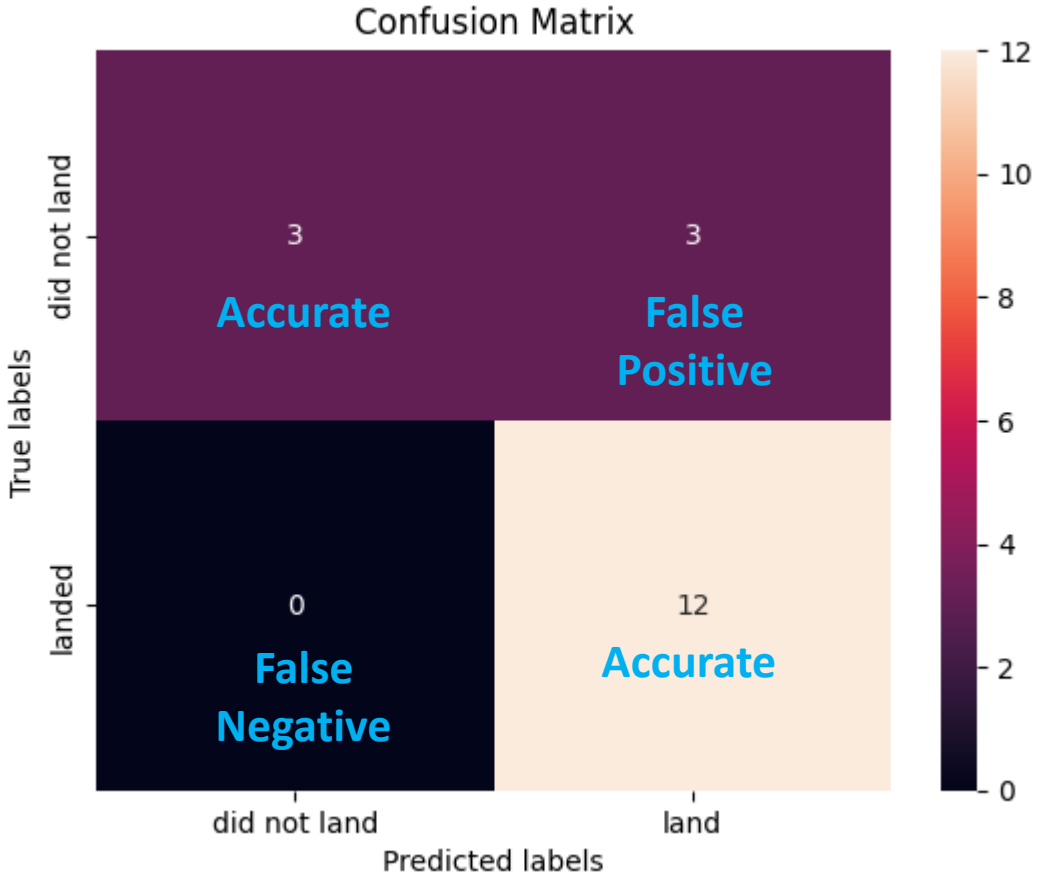


*Confusion matrix for support vector machine classifier (scikit-learn 1.6.0, Pedregosa et al., 2011), tuned via 10-fold GridSearchCV over C, gamma, and kernel type (linear, rbf, poly, sigmoid). Model selection based on mean validation accuracy.

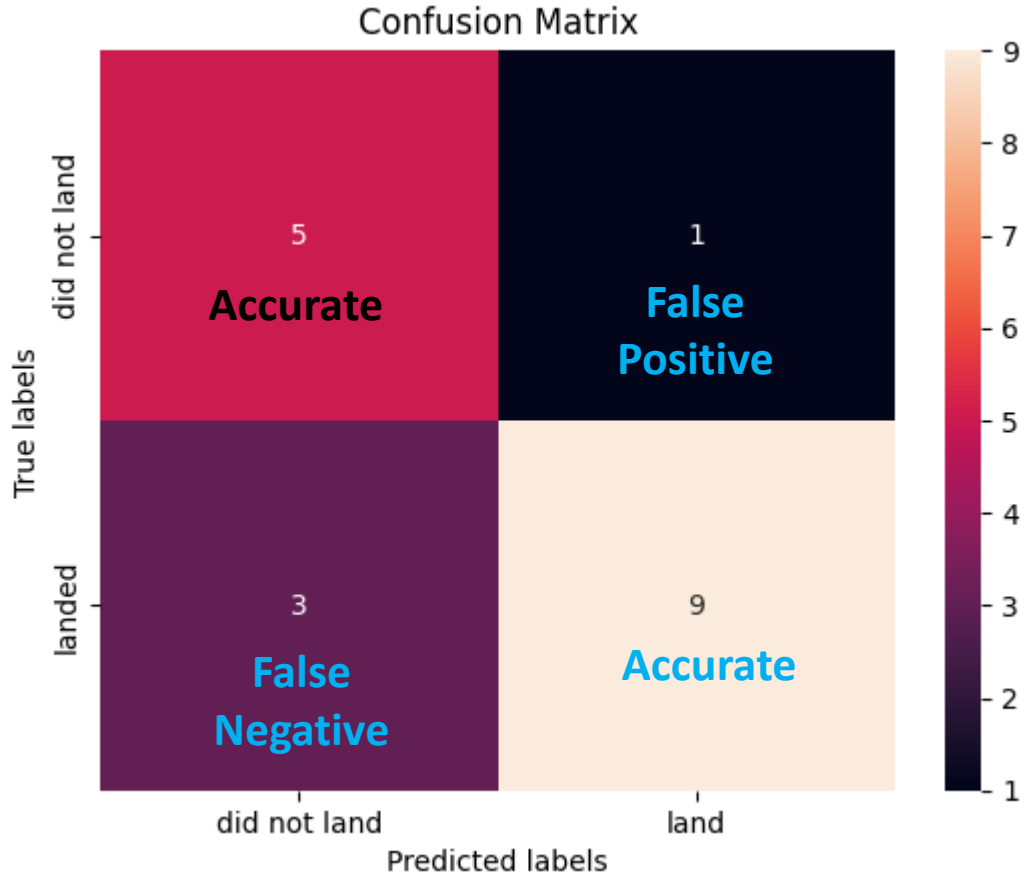
Confusion matrix for Knearest neighbors classifier (scikit-learn 1.6.0, Pedregosa et al., 2011), tuned via 10-fold GridSearchCV over n_neighbors, distance metric (p), and search algorithm. Model selected by mean cross-validation accuracy.

Confusion Matrix (2/2)

Logistic regression model – 85.0% accuracy



****Decision Tree model – 84.8% accuracy**



*Confusion matrix for logistic regression classifier (scikit-learn 1.6.0, Pedregosa et al., 2011), tuned via 10-fold GridSearchCV over regularization strength C. Model trained with L2 penalty and lbfgs solver; selected using mean cross-validation accuracy.

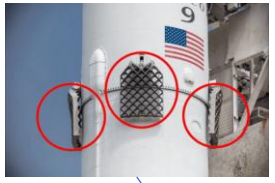
**Confusion matrix for CART classifier (scikit-learn 1.6.0, Pedregosa et al., 2011), using 10-fold cross-validation with grid search over criterion and max_depth. Model selected via mean validation accuracy.

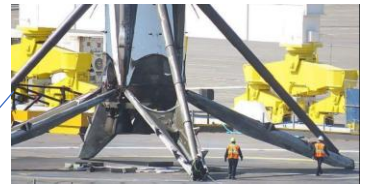
Confusion Matrix – model performance tradeoffs

Model	Model Summary	Implication in predicting booster crash costs (i.e., booster re-use helps keep costs, pricing lower)
SVM	Highest overall accuracy (~86.4%), but results in 3 false positives (predicts landing, where a crash occurred).	• Accurate, but underestimates crash risk. • Relying on model's landing predictions, will sometimes result in errantly offering lower prices due to unpredicted booster losses.
KNN	High accuracy (~86.4%), but evenly makes mistakes (2 false positives, 2 false negatives).	• Accurate, but unpredictable mix of errors. • Could result in pricing noise based on under charging or overcharging for budgeting purposes.
Logistic regression	Little less accurate (~85.0%) than SVM, but results in similar performance (3 false positives), and errors are "too optimistic".	• Same as SVM -- the risk is offering prices that are too low, due to 3 unpredicted crashes. • Possible to experiment with the decision threshold to try an improve model.
Decision Tree	Lower accuracy (~84.8%), but conservative with "more pessimistic" errors (3 false negatives) i.e., more likely to predict some crashes that end up being landings.	• Little lower accuracy, but most conservative model for financial assumptions. • Fewer costly "surprise" booster crashes. • Using this conservative model for planning is more likely to provide a financial cushion.

Decision Tree results: proper functioning of legs, number of booster reuses, grid fins and have strongest predictive effects

Grid fins





Landing legs

Booster recovery…

… for refurbishment and later re-use


```
Top features by importance:
 Legs_True          0.7170
ReusedCount         0.1935
GridFins_True       0.0895
Serial_B1032        0.0000
Serial_B1041        0.0000
Serial_B1040        0.0000
Serial_B1039        0.0000
Serial_B1038        0.0000
Serial_B1037        0.0000
Serial_B1036        0.0000
dtype: float64
```

Serial numbers are for various rocket boosters, which no predictive effect in this model (CART decision Tree)

Footnote 1: Node “value” shows the class counts at that split; feature-importance scores are aggregated over all splits as measures of total impurity reduction, so the numbers won’t directly match.
Footnote 2: : All split thresholds (the “<=” values) are in standardized units (z-score cut-offs), since the features were scaled (mean 0, σ 1) before training.

Conclusions

- Baseline established on historical data indicates approximately 2/3 (66.67%) of booster recovery outcomes (based on exploratory data analysis); the SVM model was most (86.4%) accurate, whereas Decision Tree model of most practical use (84.8%) -- providing a 20 percent lift over the naïve [non-predictive] historical data view.
- Each recovered booster can reduce per-launch cost by $25 to $40 million USD (conservatively), before refurbishment; this is enough margin to offer 25% customer discounts and still reinvest in R&D.
- Present data model gaps:
  - Missing weather and live rocket telemetry data
  - Public scraped records can be prone to noise
  - Need real-time feedback from rockets to adjust for potential model drift
- Deploy a first version model using decision tree model, integrate data to bridge data gaps, to capture booster re-use advantages based on learnings from launches.

- *Agan, T. (2013, April 25). *What SpaceX can teach us about cost innovation.* Harvard Business Review. Retrieved May 7, 2025, from https://hbr.org/2013/04/what-spacex-can-teach-us-about
- *de Selding, P. B. (2016, April 25). *SpaceX's reusable Falcon 9: What are the real cost savings for customers?* SpaceNews. Retrieved May 7, 2025, from https://spacenews.com/spacexs-reusable-falcon-9-what-are-the-real-cost-savings-for-customers/
- *Ars Technica. (2024, December 2). *SpaceX has set all kinds of records with its Falcon 9 rocket this year.* Retrieved May 7, 2025, from https://arstechnica.com/space/2024/12/spacex-has-set-all-kinds-of-records-with-its-falcon-9-rocket-this-year/
- *DuePublico. (n.d.). Ariane 6 estimated launch cost: ≈ $230 million (more than double Falcon 9). *DuePublico*. Retrieved May 7, 2025, from: https://duepublico2.uni-due.de/servlets/MCRFileNodeServlet/duepublico_derivate_00082409/Diss_Soennichsen.pdf

Thank you!