IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pascal Tagne
9/23/2025

Pascal_Tagne capstone Report

Outline

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

Executive Summary

- A systematic **CRISP-DM** methodology was followed, implementing a comprehensive data science pipeline from multi-source data collection to optimized model deployment.
    - Phase 1: Multi-Source Data Collection
    - Phase 2: Data Wrangling & Feature Engineering
    - Phase 3: Data Preprocessing
    - Phase 4: Model Development & Optimization
    - Phase 5: Model Evaluation & Selection
- Summary of all results:
    - Technical Excellence
        - High Accuracy: Exceeded 85% prediction threshold
        - Model Robustness: Consistent performance across validation sets
        - Feature Significance: Statistically significant predictor identification
    - Business Impact
        - Actionable Insights: Clear parameters for mission success optimization
        - Cost-Benefit Justification: Model value exceeds implementation costs
        - Scalability: Framework applicable to future rocket variants

Introduction

- The commercial space industry is undergoing a transformative shift driven by reusable rocket technology. SpaceX's Falcon 9 rocket has emerged as a pioneering platform in this revolution, with its ability to recover and reuse first-stage boosters representing a fundamental breakthrough in cost reduction and operational efficiency. Each successful landing saves approximately $62 million in booster replacement costs, making accurate prediction of landing outcomes a critical business intelligence capability.
- The ability to reliably forecast first-stage landing success has profound implications for the economics of space access. While SpaceX has achieved remarkable landing success rates through engineering excellence, the complex interplay of factors affecting each landing outcome—including payload characteristics, launch parameters, and environmental conditions—creates an ideal scenario for predictive analytics. This project addresses the strategic need to quantify and predict landing success probabilities, enabling data-driven decision-making in mission planning and resource allocation.

Section 1

Methodology

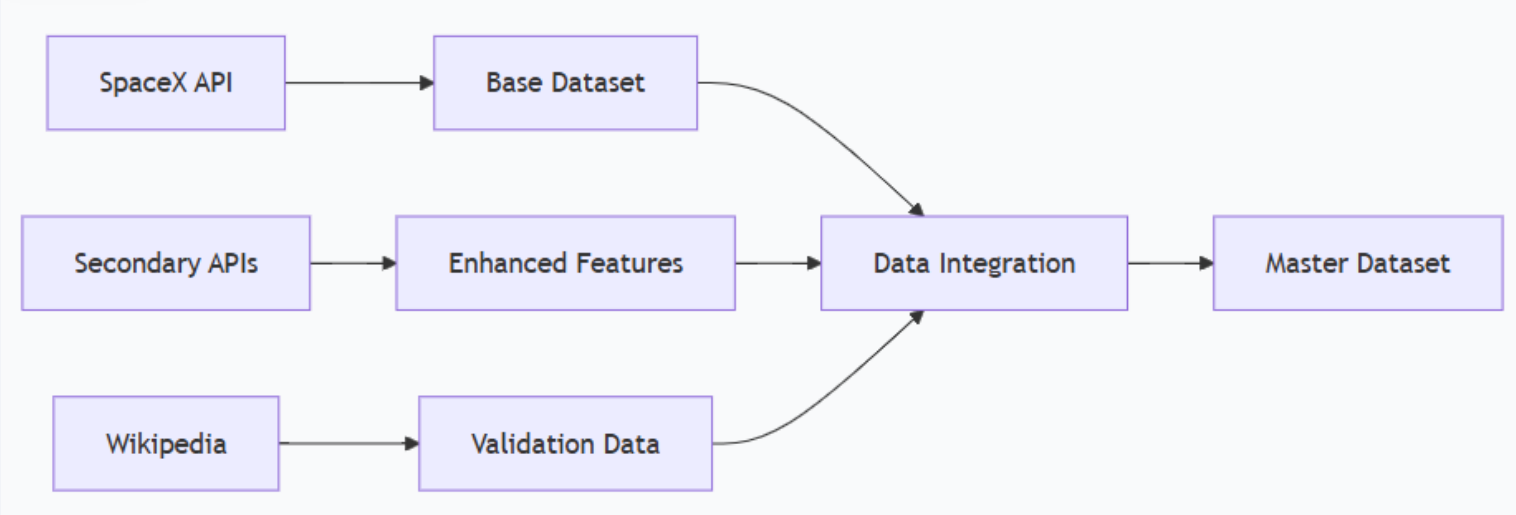
Methodology

Executive Summary

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

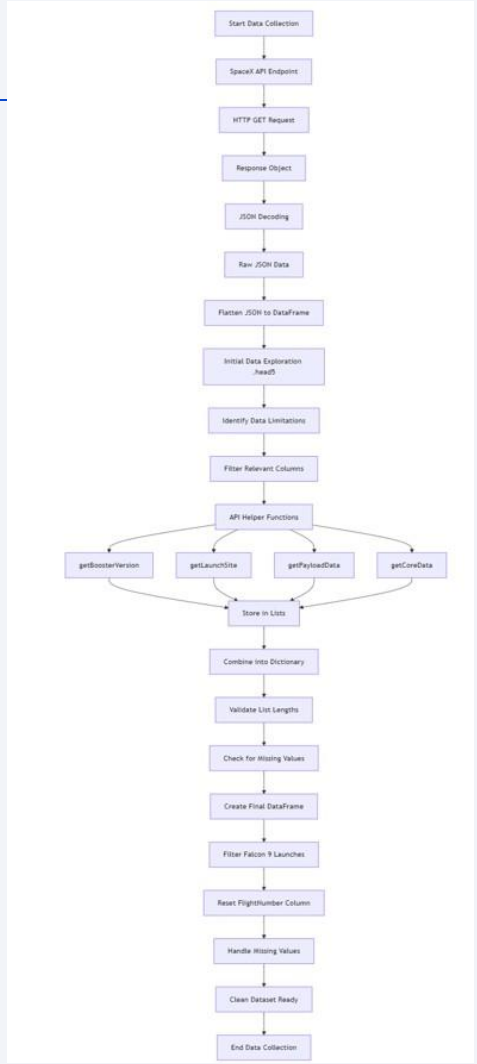
Data Collection

- We used Multi-Source Data Collection Approach:
  - Primary Data Source: SpaceX REST API
  - Data Enhancement: Secondary API Calls
  - Supplementary Data: Web Scraping
  - Data Integration Pipeline
  - Data Cleaning & Filtering Process
  - Final Dataset Structure
  - Quality Assurance Measures



Data Collection – SpaceX API

- Below are the steps we used:
  - HTTP request: requests.get() → response object
  - JSON decoding: response.json() → data variable
  - Data normalization: pd.json_normalize(data) → initial DataFrame
  - Data filtering: Keep only relevant columns
  - Helper functions: getBoosterVersion(), getLaunchSite(), getPayloadData(), getCoreData()
  - Data validation: List length checking, missing values detection
  - DataFrame construction: Dictionary → launch_df
  - Data cleaning: Falcon 9 filtering, FlightNumber reset, missing values handling

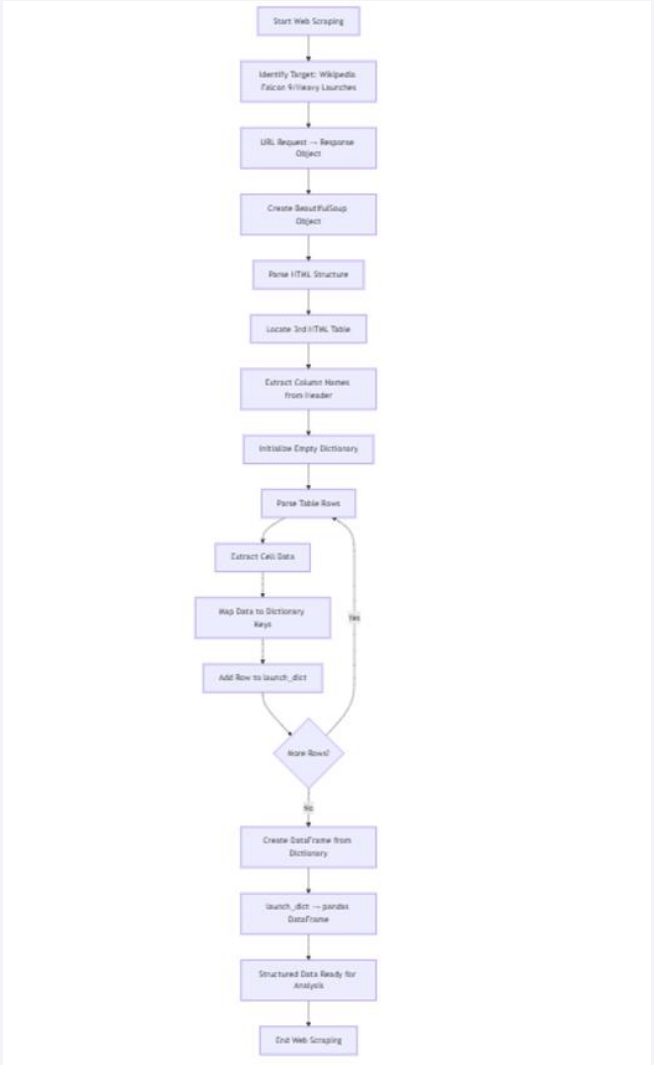


Here is the GitHub URL : https://github.com/ptagne/TestRepo/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection – Scraping.

HTTP Handling: requests.get(url) → Response object
HTML Parsing: BeautifulSoup(response.text, 'html.parser')
Table Navigation: Identify 3rd <table> element
Header Extraction: <th> tags for column names
Data Mapping: Table rows → Dictionary key-value pairs
DataFrame Creation: pd.DataFrame.from_dict(launch_dict

Here is the GitHub URL : https://github.com/ptagne/TestRepo/blob/main/jupyter-labs-webscraping.ipynb



- Data Scraping flowcharts

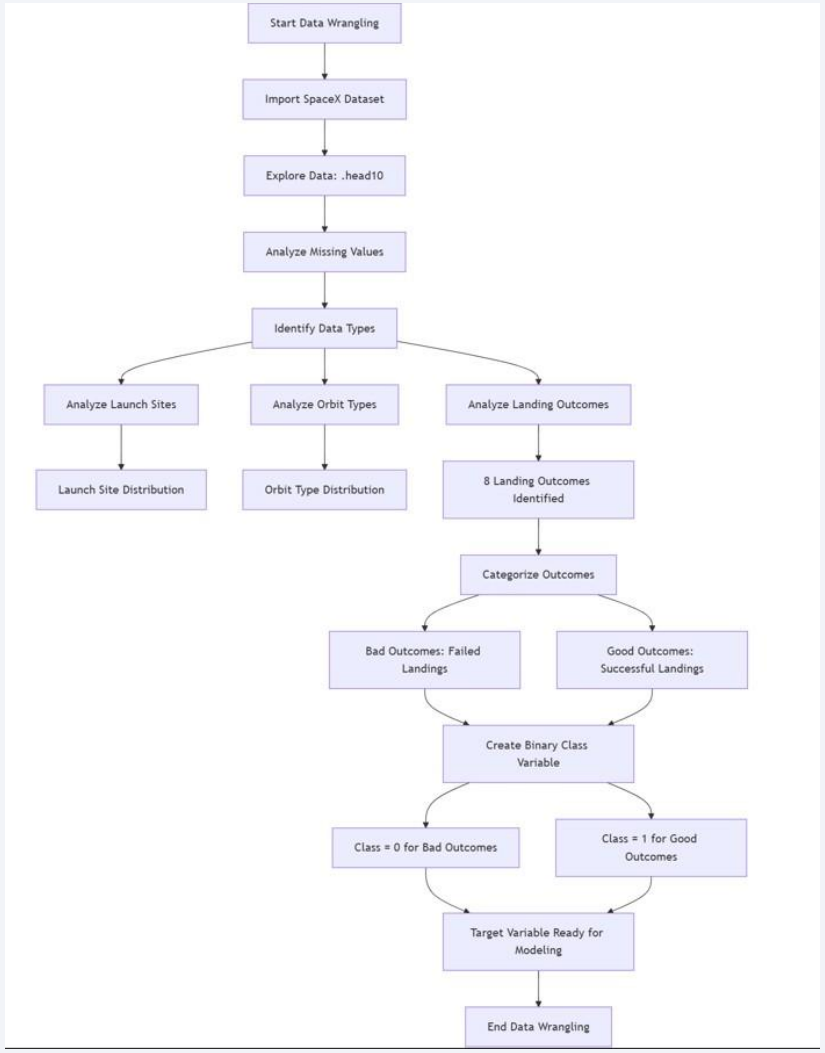
Data Wrangling

- Describe how data were processed:
  - **Data loading**: df = pd.read_csv('Spacex dataset')
  - **Data exploration**: .head(10), .dtypes, .value_counts()
  - **Missing values analysis**: df.isnull().sum()/len(df)*100
  - **Feature characterization**: Numerical vs categorical identification
  - **Launch site analysis**: df['LaunchSite'].value_counts()
  - **Orbit distribution**: df['Orbit'].value_counts()
  - **Outcome categorization**: Landing outcomes classification
  - **Target variable creation**: Binary Class variable for success/failure

Here is the GitHub URL : https://github.com/ptagne/TestRepo/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

Data Wrangling Cont'd.

Data Wrangling Flowcharts



EDA with Data Visualization

- I used scatter and bar plots to visualize relationships between features. I used catplot() type figure because of categorical features.
- Below are list of plots I used:

- Visualize the relationship between Flight Number and Launch Site
- Visualize the relationship between Payload Mass and Launch Site
- Visualize the relationship between success rate of each orbit type
- Visualize the relationship between FlightNumber and Orbit type
- Visualize the relationship between Payload Mass and Orbit type
- Visualize the launch success yearly trend

Here is the GitHub URL: https://github.com/ptagne/TestRepo/blob/main/edadataviz.ipynb

EDA with SQL

- Below are SQL queries I performed:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  - List the total number of successful and failure mission outcomes.

EDA with SQL Cont'd.

- Below are SQL queries I performed:
  - List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.
  - 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.
  - 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.
  - Here is the GitHub URL: https://github.com/ptagne/TestRepo/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

- I first created a folium Map object with folium.Map() named site_map, with an initial center location to be NASA JSC at Houston, Texas.
- I then created a folium circle with folium.Circle() at NASA JSC's coordinate with a popup label showing its name and added to site_map.
- I also created a marker with folium.Marker() to put an icon as a text label on NASA JSC location.
- I created and add folium.Circle and folium.Marker for each **launch site** on the site map. I zoomed and it was a bit crowded.

Build an Interactive Map with Folium Cont'd1.

- I created a MarkerCluster() object named marker_cluster to add launch outcome of each sites and see which sites have high success rates. I then added that to site_map. I used a cluster object for readable map and better user experience for data exploration and analysist. And now from the color-labeled markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rates.
- I used a folium plugin MousePosition() to display the geographic coordinates (lat, long) of the mouse cursor as it moves over the map. That is helpful in identifying exact coordinates on the map and facilitate data exploration and verification.
- I Marked down a point on the closest coastline using MousePosition and calculate the distance between the coastline point and the launch site.
- I also created a marker with distance to a closest city, railway, highway, etc.

Build an Interactive Map with Folium Cont'd2.

- With all these markers and distances calculations, I was able to answer these questions:
  - Are launch sites in close proximity to railways? No.
  - Are launch sites in close proximity to highways? No.
  - Are launch sites in close proximity to coastline? Yes.
  - Do launch sites keep certain distance away from cities? No.

- Here is the GitHub URL : https://github.com/ptagne/TestRepo/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- To my dashboard, I added:
  - All Launch Sites Dropdown and pie chart success rate. Whenever you start the app, you will see a drop down and a pie chart success rate.
  - Once you select a site in the dropdown menu, you will see a pie chart showing that Launch Site success and failed ratio.
  - We added Scatter plot of launch sites vs payload with payload range slider.
- With these we were able to answer questions below:
  - Which site has the largest successful launches? **KSC LC-39A**
  - Which site has the highest launch success rate? **KSC LC-39A**
  - Which payload range(s) has the highest launch success rate? **2490 – 5300 KG**
  - Which payload range(s) has the lowest launch success rate? **6070 – 6761 KG**
  - Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate? **FT**
  - Here is the GitHub URL : https://github.com/ptagne/TestRepo/blob/main/dash_interactivity.py

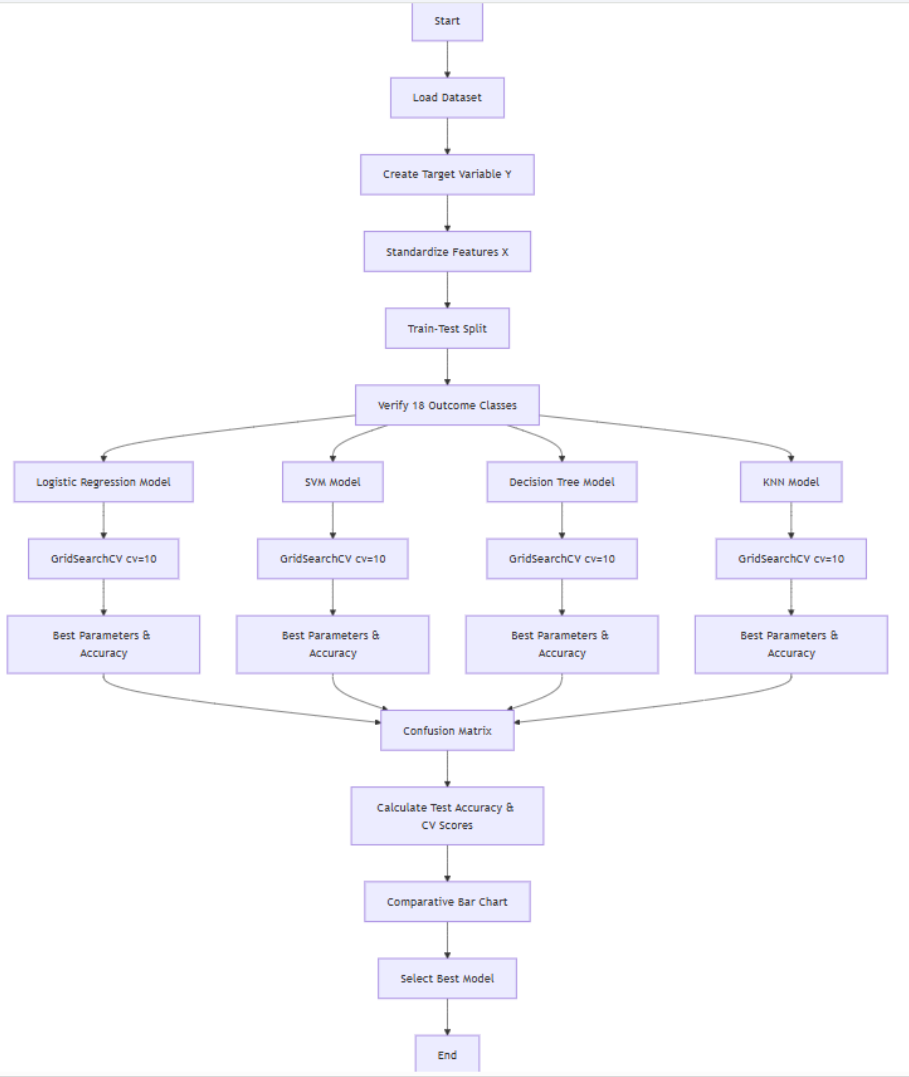
Predictive Analysis (Classification)

- I started by creation a variable Y and assigned the serie df['Class'].to_numpy() to that variable.
- Since the data had categorical features, I standardized the data using an instance of StandardScaler(), fit_transform(X) and reassigned that to X variable.
- I then split data into training and testing using train_test_split().
- I checked the shape of Y and got 18 outcomes.
- I created a logistic regression object named **logreg**, created a GridSearchCV object **logreg_cv** with **cv = 10**. I created a GridSearchCV object to allow me to perform **hyperparameter tuning**. Then used **logreg as estimator** in GridSearchCV. We fit the GridSearchCV object to find the best parameters and the best score (accuracy). We then calculate the accuracy on the test data using the method score(). We obtained the accuracy of test data. Then we looked at the confusion matrix.

Predictive Analysis (Classification) Cont'd1.

- We repeated step 5 with **svm** (Support Vector Machine) by creation SVM object, GridSearchCV object **svm_cv** with **cv = 10**. The **estimator** we used was **svm**. Got best parameters, accuracy, test data accuracy and looked at the confusion matrix.
- We repeated step 5 with a decision tree classifier object named **tree** and GridSearchCV object **tree_cv** with **cv = 10**. The **estimator** we used was **tree**. Got best parameters, accuracy, test data accuracy and looked at the confusion matrix.
- We repeated step 5 with a k nearest neighbors object named **knn** and created a GridSearchCV object **knn_cv** with **cv = 10**. The **estimator** we used was **knn**. Got best parameters, accuracy, test data accuracy and looked at the confusion matrix.
- We finished with the calculation of **test accuracy** and **CV scores** for each model then plotted that on a bar chart to select best model.
- Below is the GitHub URL:
  - https://github.com/ptagne/TestRepo/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Predictive Analysis (Classification) Cont'd2.



Flow Chart of Model prediction process.

Results

- Exploratory data analysis results go from slide 26 to slide 41. Below are slide titles:
  - Flight Number vs. Launch Site scatter plot ; • Payload vs. Launch Site.
  - Success Rate vs. Orbit Type ; • Flight Number vs. Orbit Type
  - Payload vs. Orbit Type ; • Launch Success Yearly Trend
  - All Launch Site Names; • Launch Site Names Begin with 'CCA'
  - Total Payload Mass; • First Successful Ground Landing Date
  - First Successful Ground Landing Date,
  - Successful Drone Ship Landing with Payload between 4000 and 6000
  - Total Number of Successful and Failure Mission Outcomes.
  - Boosters Carried Maximum Payload; • 2015 Launch Records.
  - Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

Results Cont'd 1.

- Interactive analytics demo in screenshots:
  - Slide 49 shows All Launch Sites Dropdown and pie chart success rate.
  - Slide 50 shows Pie Chart of the Launch Site with the highest success ratio.
  - Slide 51 shows Scatter plot of launch sites vs payload with payload range slider

Results Cont'd 2.

```
============================================================
DETAILED MODEL PERFORMANCE SUMMARY
============================================================

Logistic Regression:
  Test Accuracy: 0.8333
  CV Best Score: 0.8607
  Best Parameters: {'C': 0.1, 'penalty': 'l1', 'solver': 'saga'}

SVM:
  Test Accuracy: 0.8333
  CV Best Score: 0.8482
  Best Parameters: {'C': 1, 'degree': 2, 'gamma': 'scale', 'kernel': 'sigmoid'}

Decision Tree:
  Test Accuracy: 0.9444
  CV Best Score: 0.8732
  Best Parameters: {'criterion': 'gini', 'max_depth': 10, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 10}

K-Nearest Neighbors:
  Test Accuracy: 0.8333
  CV Best Score: 0.8482
  Best Parameters: {'algorithm': 'auto', 'n_neighbors': 11, 'p': 1, 'weights': 'uniform'}

🎯 BEST PERFORMING MODEL: Decision Tree (Accuracy: 0.9444)
```

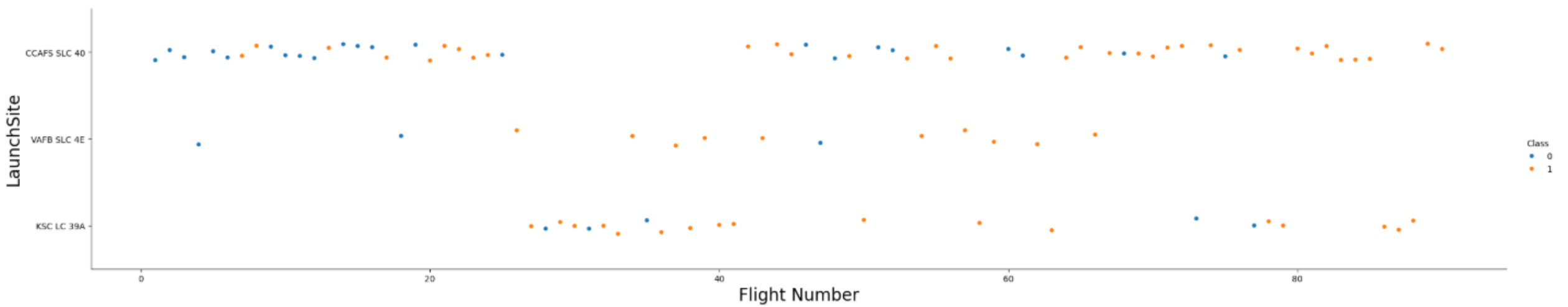
- Predictive analysis results:
  - The method that performed best is DecisionTree.
  - Decision Tree Test Accuracy: 94.44%
  - FP went from 3 to 1.

Section 2

Insights drawn from EDA

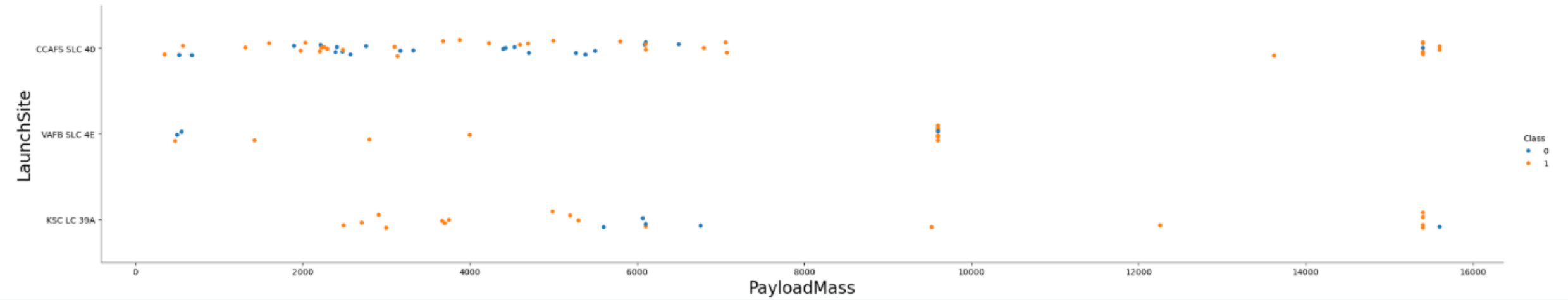
Flight Number vs. Launch Site

- The scatter plot shows that they were few flights from VAFB SLC 4E.
- That more flights were successful (class = 1).
- They were more flights from CCAFS SLC-40.



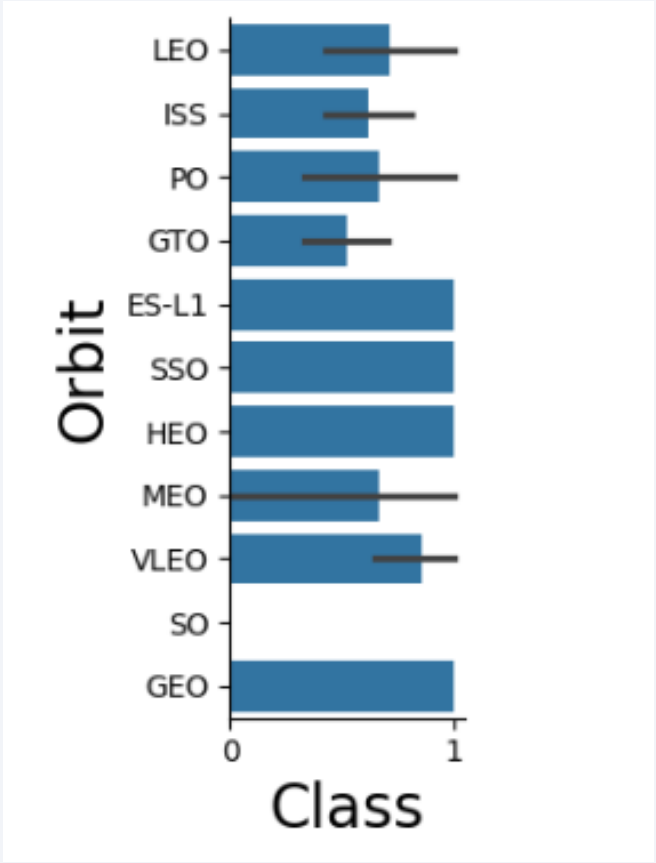
Payload vs. Launch Site

- The scatter plot shows that after 8000 kg of payload they are fewer launches.
- It also shows that VAFB SLC 4E made fewer launches.



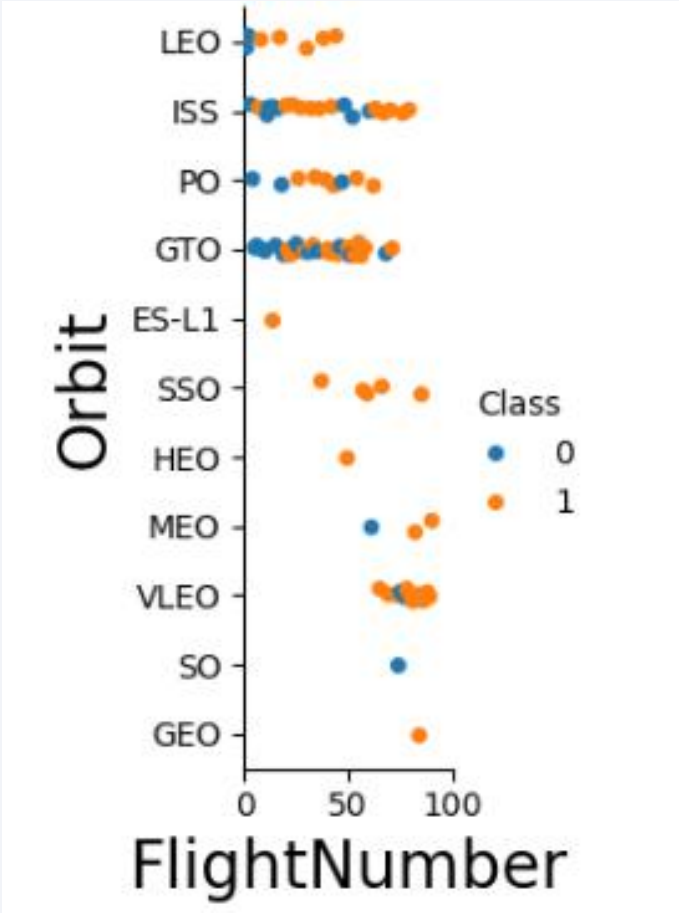
Success Rate vs. Orbit Type

- The bar plot shows that the orbits with the highest success rate are ES-L1, SSO, HEO, and GEO.
- SO has no success.



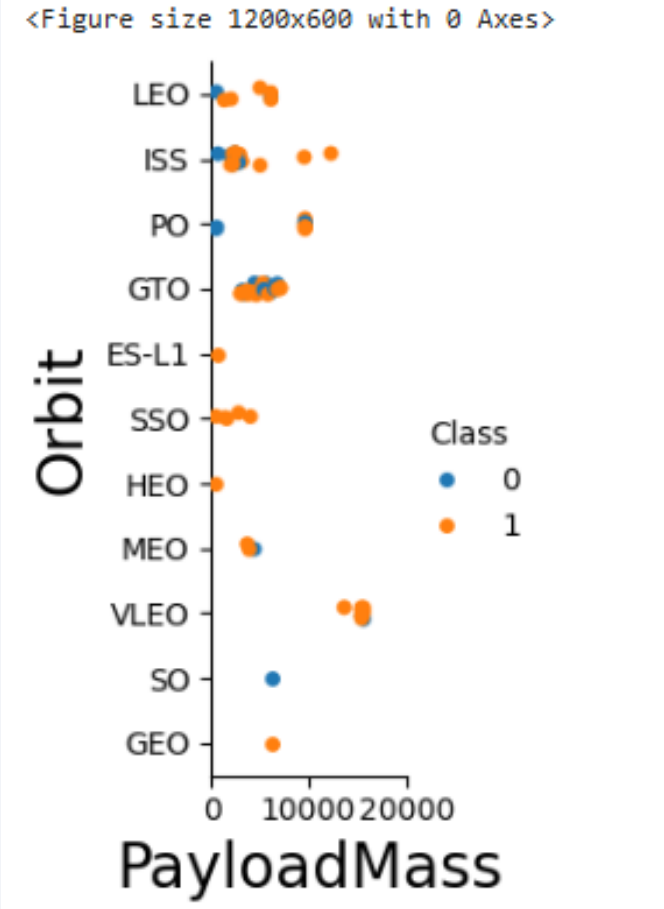
Flight Number vs. Orbit Type

- The scatter plot shows that in the LEO's orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.
- All flights to SSO were successful.



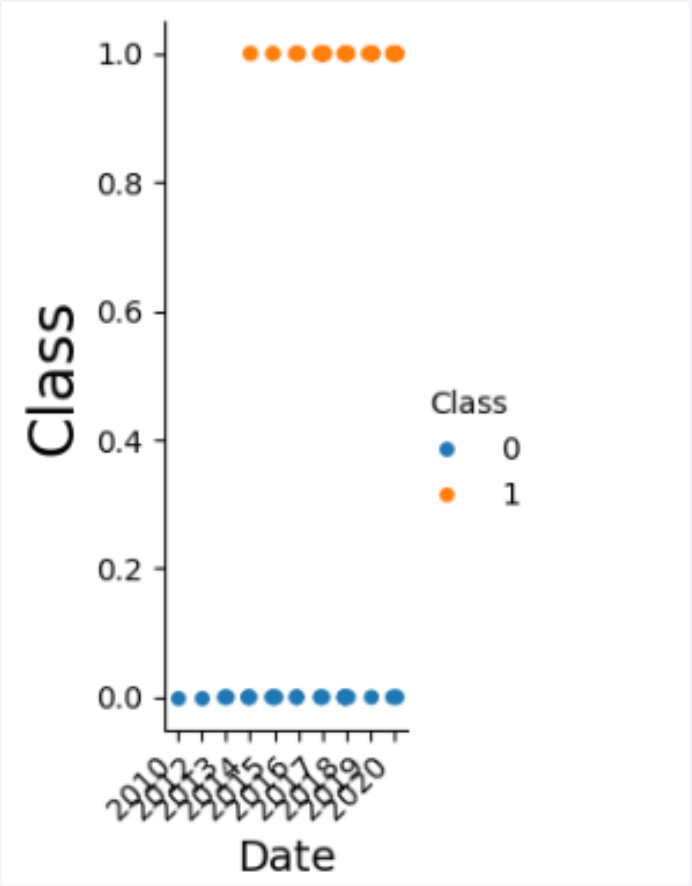
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for LEO and ISS.
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.



Launch Success Yearly Trend

- You can observe that the success rate since 2013 kept increasing till 2020



All Launch Site Names

- The names of the unique launch sites are below.
- I used magic line (%sql) and DISTINCT to get unique names of launch sites.

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

```
[13]: %sql select DISTINCT "Launch_Site" from SPACEXTBL;
```

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

[13]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

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

[11]:
```
%sql select * from SPACEXTBL where "Launch_Site" LIKE "CCA%" LIMIT 5;
```

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

[11]:

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

- I used magic cell, WHERE clause, LIKE and LIMIT to get 5 records where launch site begins with 'CCA'.

Total Payload Mass

- I used select with the WHERE clause = “NASA(CRS)”.
- I did that because I checked DISTINCT customer and found only 1 instance of NASA(CRS).

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

```
[13]: %sql select "Customer", SUM(PAYLOAD_MASS__KG_) as TotalPayloadMass from SPACEXTBL where "Customer" = "NASA (CRS)";
 * sqlite:///my_data1.db
Done.
```

[13]:

Customer	TotalPayloadMass
NASA (CRS)	45596

Average Payload Mass by F9 v1.1

- I used LIKE function with "%F9 v1.1%" to get the average payload mass.
- I used LIKE instead of WHERE because when you do DISTINCT Customer, you will find version like F9 v1.1 B… and F9 v1.1. To include all of those records in the average, I used LIKE on "%F9 v1.1%"

```
[20]: %sql select "Booster_Version", AVG("PAYLOAD_MASS__KG_") as Mean_PayloadMass from SPACEXTBL where "Booster_Version" LIKE "%F9 v1.1%";
 * sqlite:///my_data1.db
Done.
```

[20]:

Booster_Version	Mean_PayloadMass
F9 v1.1 B1003	2534.6666666666665

First Successful Ground Landing Date

- I used MIN() function on “date” feature with the WHERE clause on the Landing_Outcome.

```
[79]: %sql select MIN("Date") from SPACEXTBL where "Landing_Outcome" = "Success (ground pad)" ;
```

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

[79]:

MIN("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- I used select with WHERE clause and BETWEEN to get the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

[22]:
```
%sql select "Booster_Version", "Landing_Outcome", "PAYLOAD_MASS__KG_" as PayLoad_Mass from SPACEXTBL where "Landing_Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

[22]:

Booster_Version	Landing_Outcome	PayLoad_Mass
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

- I first used DISTINCT function to identify 4 groups of Mission_Outcome.
- I then used GROUP BY (“Mission_Outcome”) then count each individual records.

List the total number of successful and failure mission outcomes

[89]:
```
%sql select "Mission_Outcome", COUNT("Mission_Outcome") as Total_Mission_Outcome from SPACEXTBL GROUP BY ("Mission_Outcome");
```

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

[89]:

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

Boosters Carried Maximum Payload

- I used magic cell (%%sql) and subquery to get booster version records that carries max payload.

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

[24]:
```
%%sql SELECT DISTINCT "Booster_Version", "PAYLOAD_MASS__KG_" from SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") from SPACEXTBL WHERE "PAYLOAD_MASS__KG_" IS NOT NULL)
AND "PAYLOAD_MASS__KG_" IS NOT NULL;
```

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

[24]:

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

2015 Launch Records

- I used substr(Date,0,5) to extract Year, the WHERE clause on Landing_Outcome and restrain the records for 2015.

```
[26]: %sql select substr(Date,0,5) as Year, "Booster_Version", "Launch_Site", "Landing_Outcome" from SPACEXTBL where "Landing_Outcome" ="Failure (drone ship)" and  substr(Date,0,5) = "2015";
 * sqlite:///my_data1.db
Done.
```

[26]:

Year	Booster_Version	Launch_Site	Landing_Outcome
2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- I used magic cell (%%sql), RANK() function over COUNT landing_outcomes DESC with the WHERE clause on date between 2010-06-04 and 2017-03-20, ORDER BY landing_outcome count DESC.

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.

[96]:
```
%%sql SELECT "Landing_Outcome", COUNT(*) as Landing_Outcome_count, RANK() OVER (ORDER BY COUNT(*) DESC) as Landing_Outcome_rank FROM SPACEXTBL
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' AND Landing_Outcome IS NOT NULL GROUP BY "Landing_Outcome" ORDER BY Landing_Outcome_count DESC;
```

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

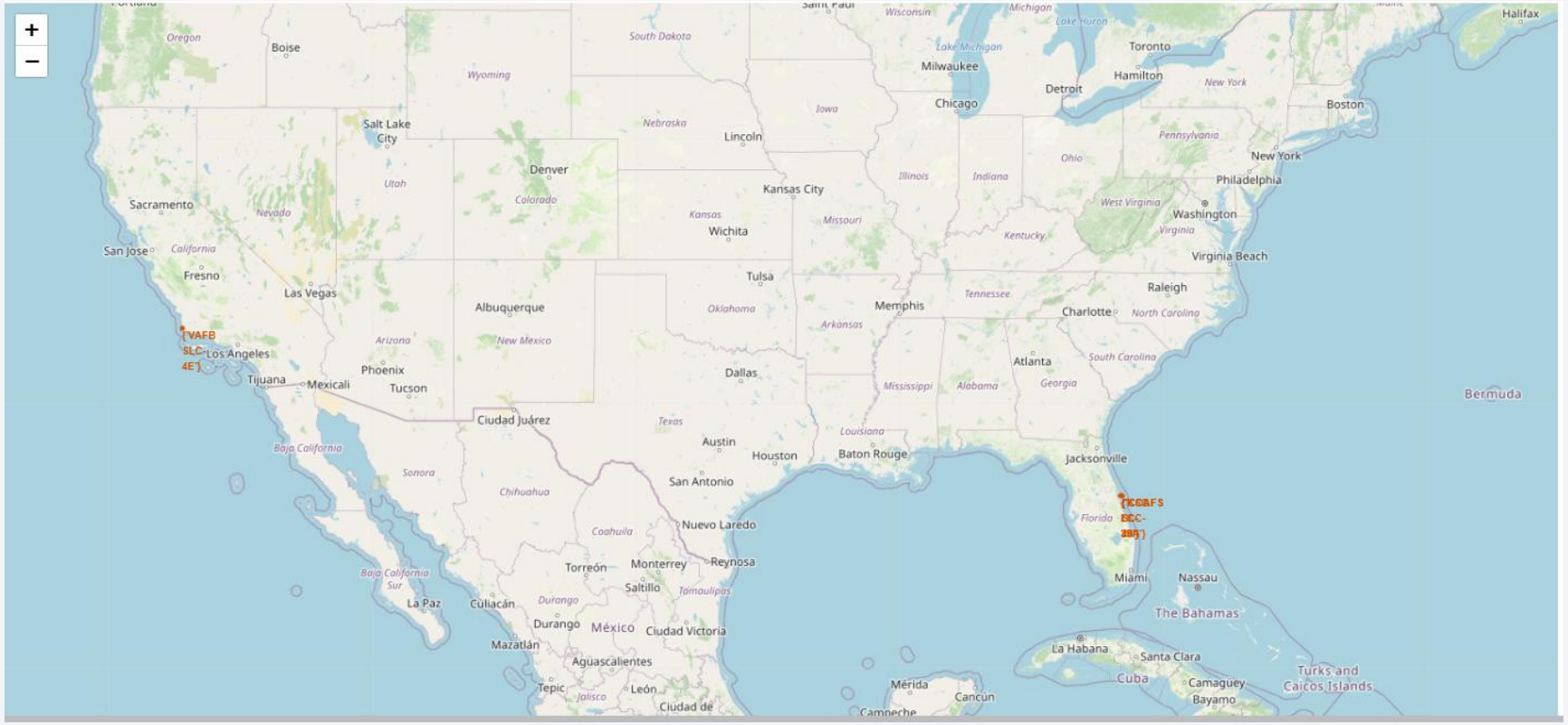
[96]:

Landing_Outcome	Landing_Outcome_count	Landing_Outcome_rank
No attempt	10	1
Success (drone ship)	5	2
Failure (drone ship)	5	2
Success (ground pad)	3	4
Controlled (ocean)	3	4
Uncontrolled (ocean)	2	6
Failure (parachute)	2	6
Precluded (drone ship)	1	8

Section 3

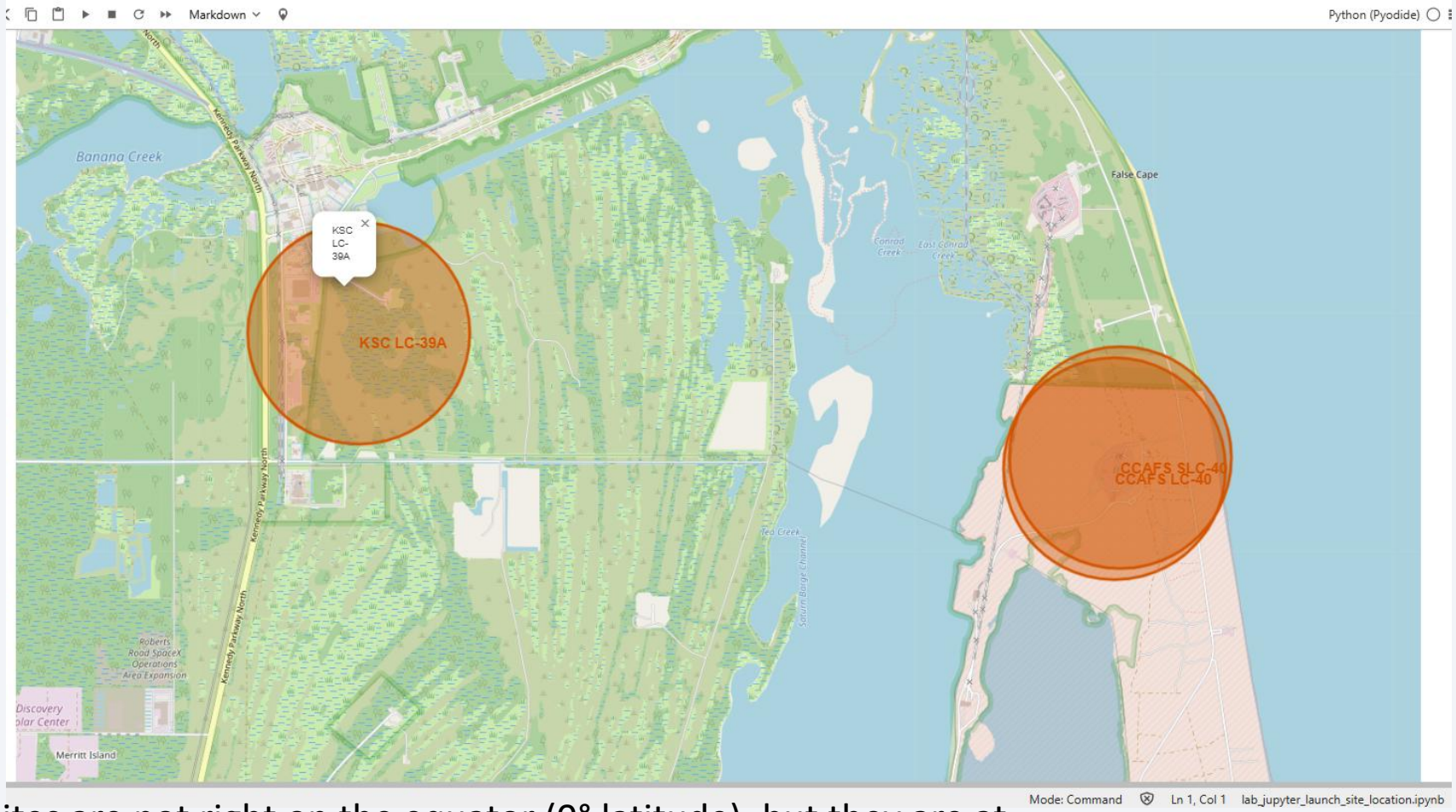
Launch Sites Proximities Analysis

Map of all launch site's location markers.



The map showed launch sites locations. We will zoom to get findings. We also observed that all sites are closer together. We will need markercluster to make it easier to visualize. Please see next slide.

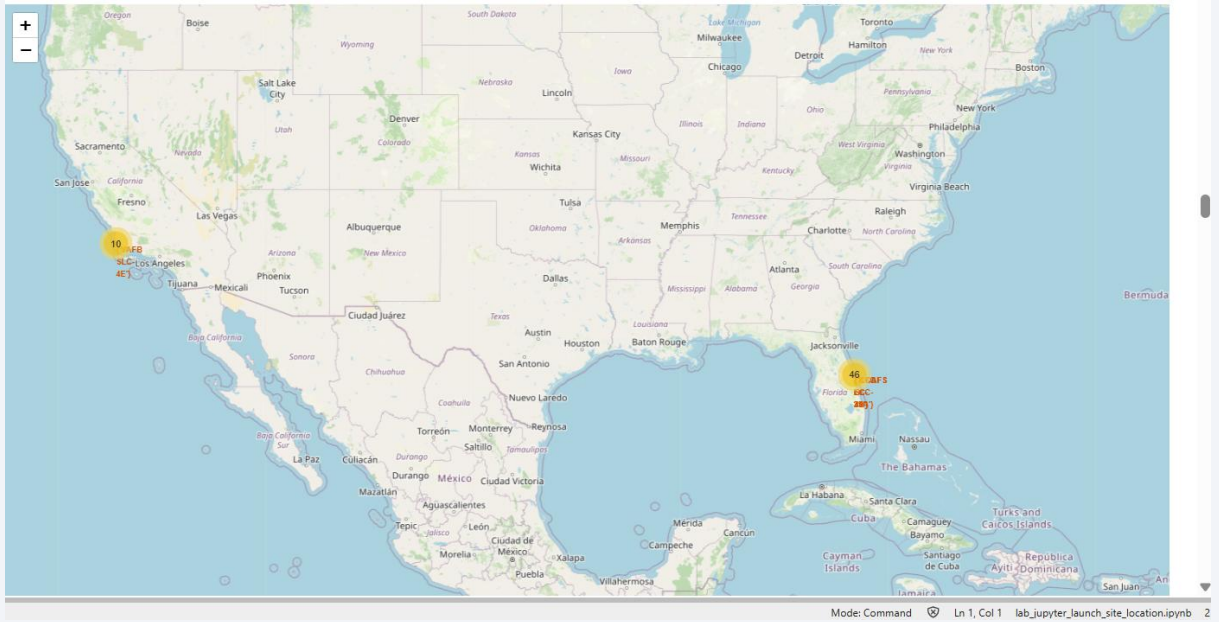
Map of all launch site's location markers cont'd.



1) SpaceX launch sites are not right on the equator (0° latitude), but they are at relatively low latitudes to take advantage of Earth's rotation.
2) Yes, all SpaceX launch sites are in very close proximity to coastlines.

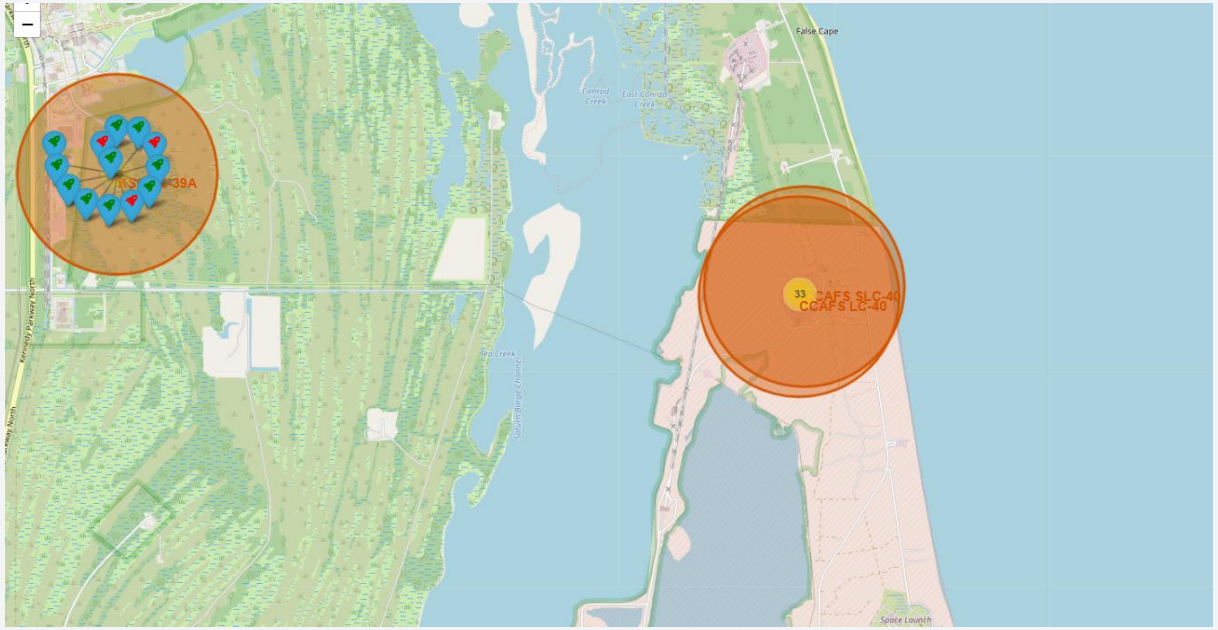
Map of launch sites showing color-labeled launch outcomes.

- This map shows the sites launch. The next chart is the continuation of this site (we zoomed in to have better view).

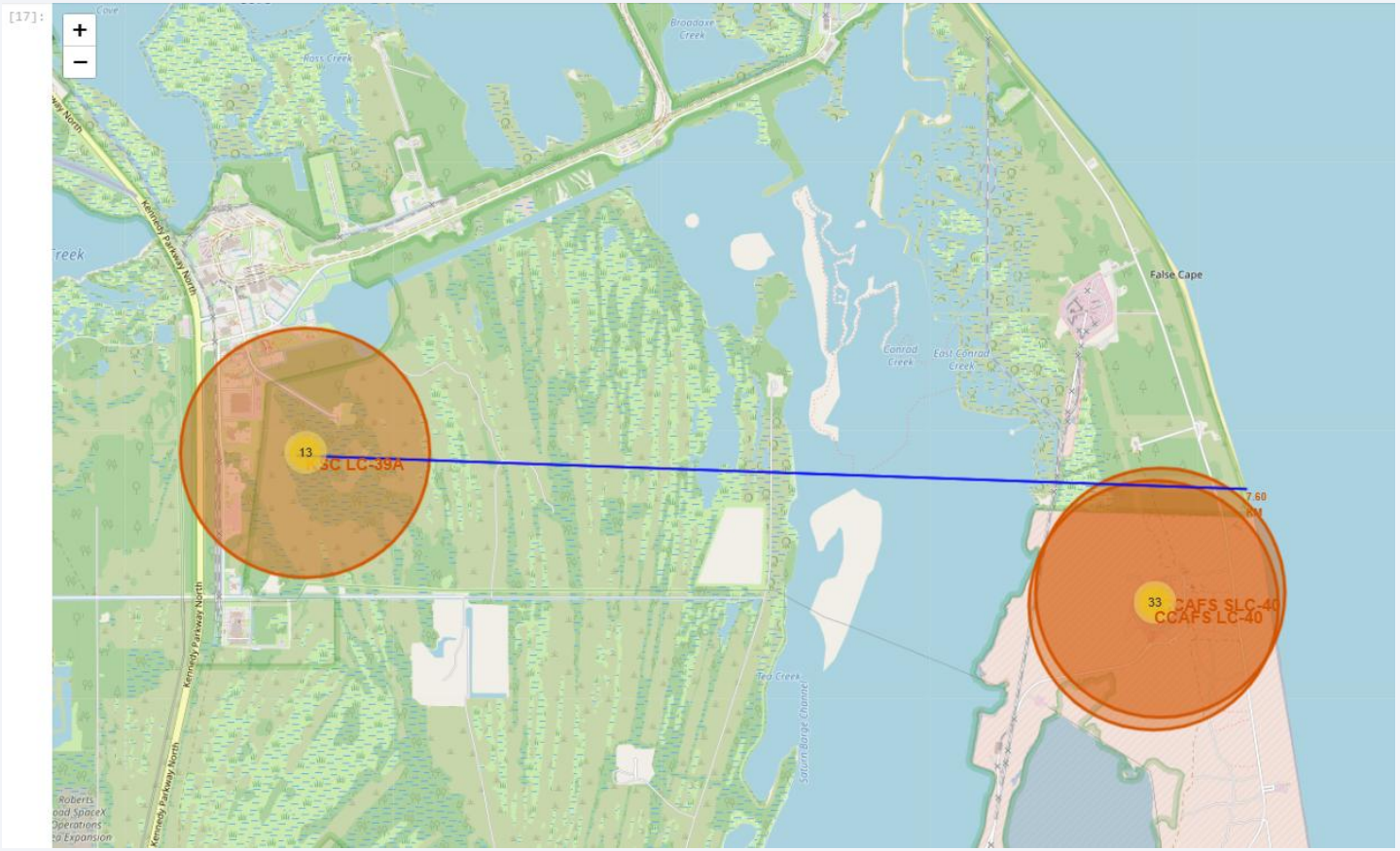


Map of launch sites showing color-labeled launch outcomes Cont'd.

From the color-labeled markers in for KSC LC-39A, we can see that there are 10 success launches and 3 unsuccessful.
From CCAFS SLC-40, we observed 3 success launch and 4 unsuccessful launch.



Map of launch site KSC LC-39A to proximity to coastline.

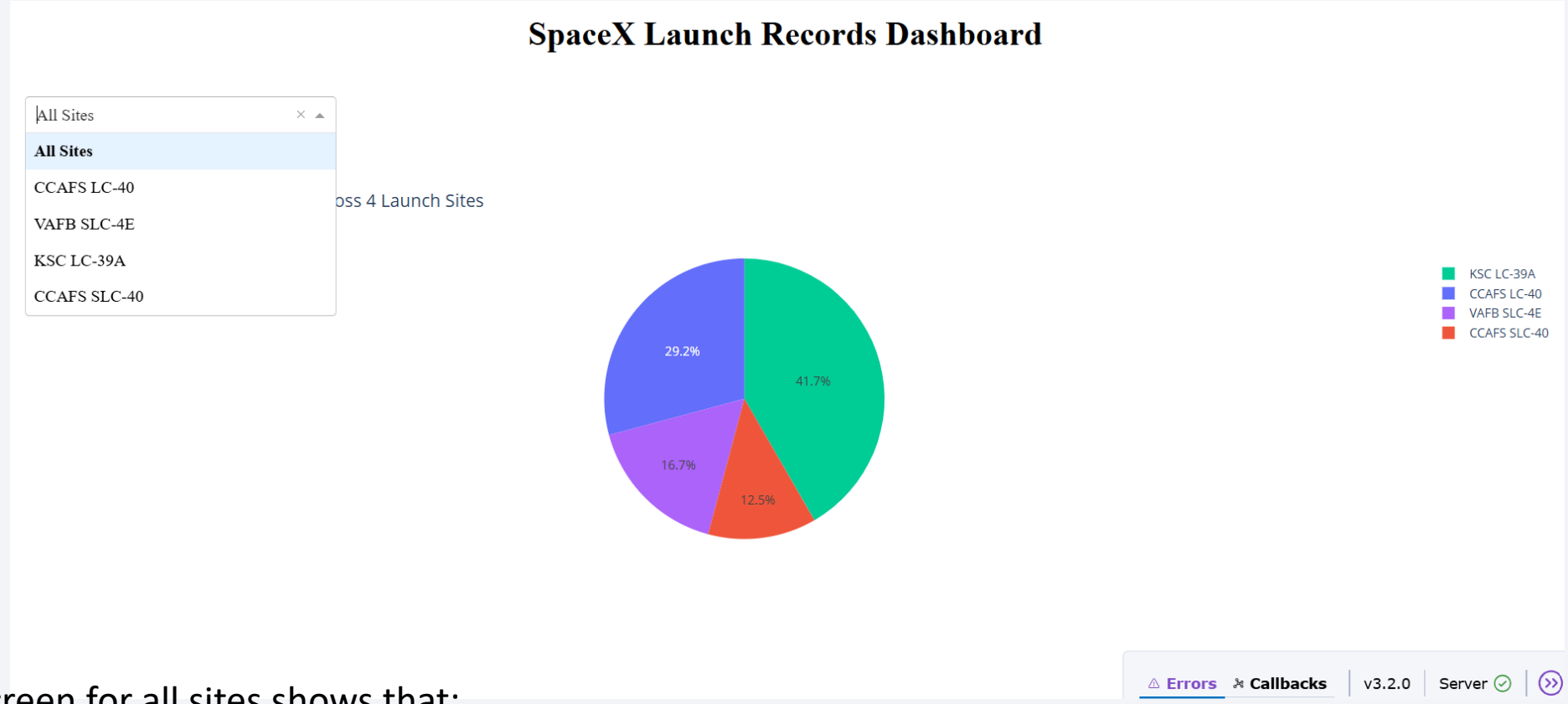


The map showed KSC LC-39A is about **7.6 km** from the coastline.
It also confirm its proximity to the coastline.

Section 4

Build a Dashboard with Plotly Dash

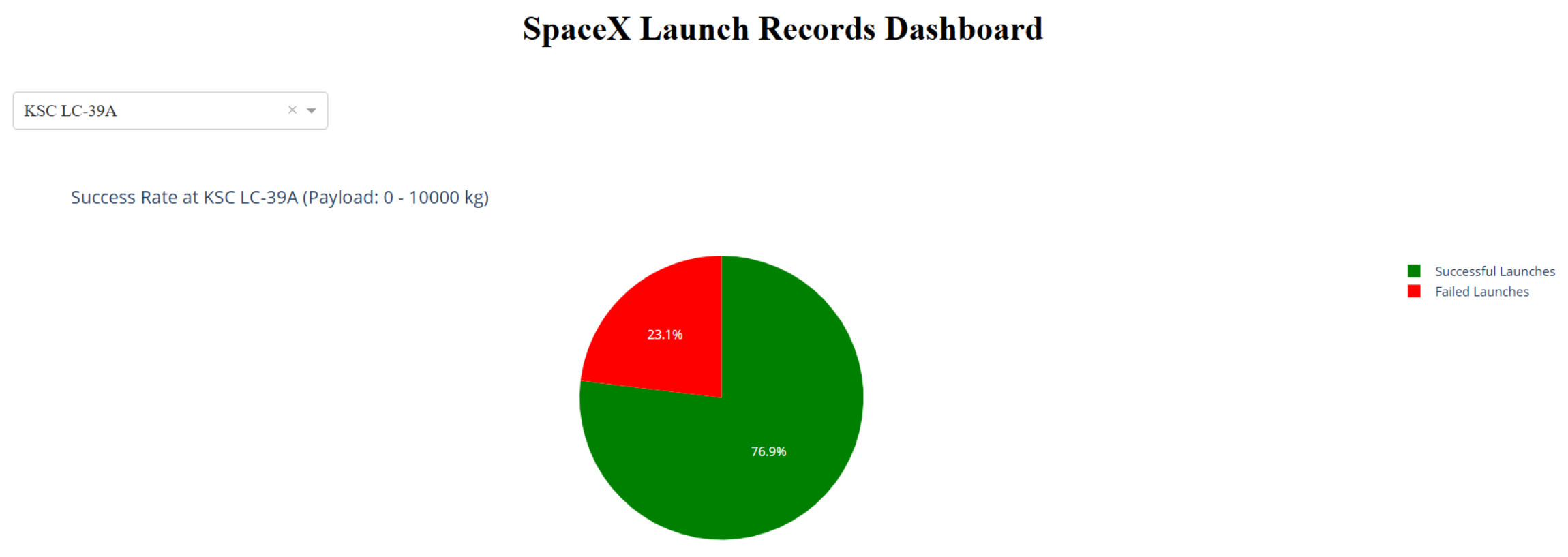
All Launch Sites Dropdown and pie chart success rate



The screen for all sites shows that:

1) The best success launch site is KSC LC-39A with 41.7% success rate.
2) The 2nd best site is CCAFS LC-40 with 29.2% success rate.
3) The third best site is VAFB SLC 4E with 16.7% success rate.
4) The last site is CCAFS SLC-40 with 12.5% success rate.

Pie Chart of the Launch Site with the highest success ratio.



KSC LC-39A launched 13 rockets.
10 launches were successful (79.9%)
3 launches failed (23.1%).

Scatter plot of launch sites vs payload with payload range slider

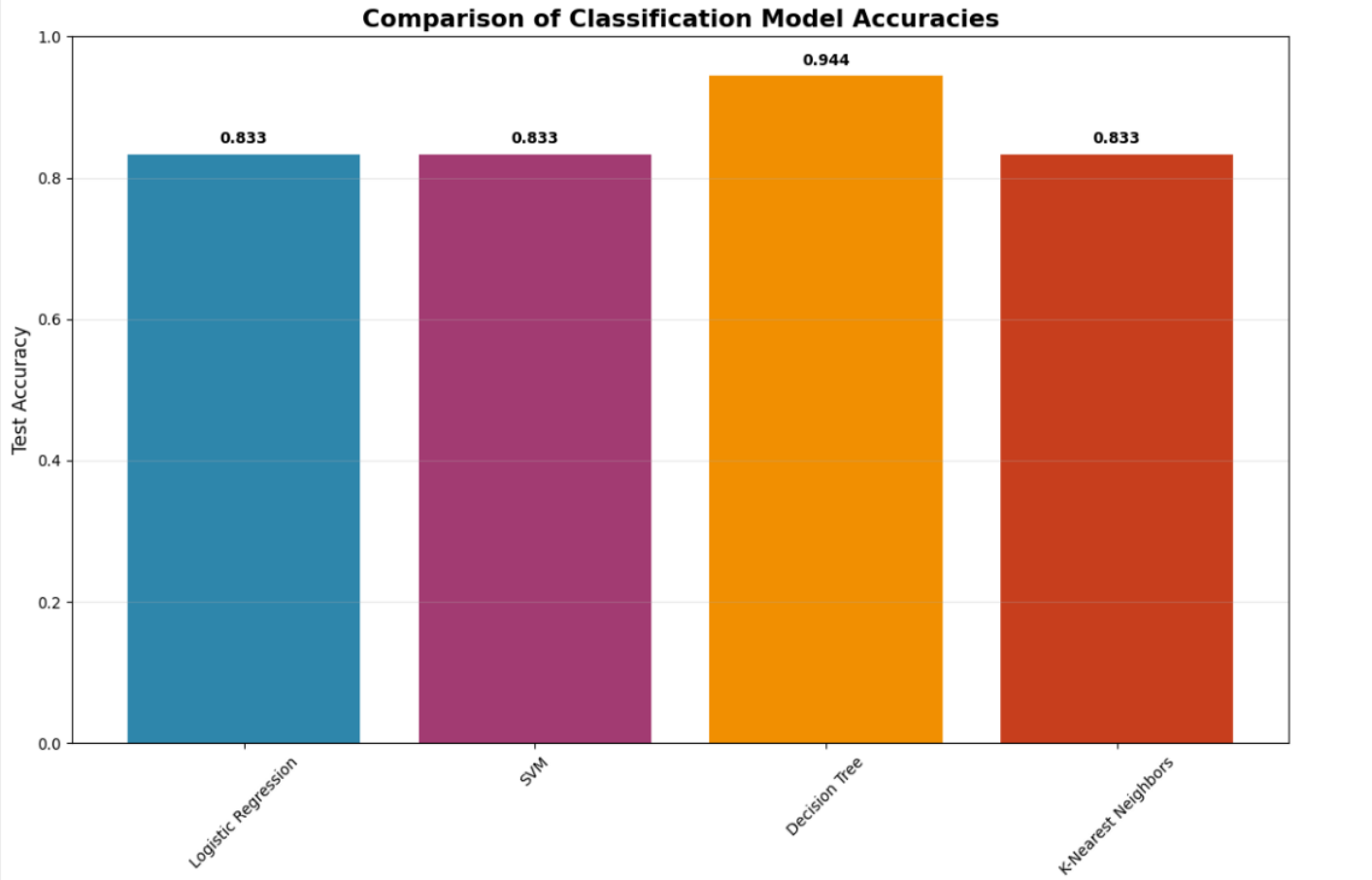
Which site has the largest successful launches? **KSC LC-39A**
Which site has the highest launch success rate? **KSC LC-39A**
Which payload range(s) has the highest launch success rate? **2490 – 5300 KG**
Which payload range(s) has the lowest launch success rate? **6070 – 6761 KG**
Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate? **FT**

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Here is the bar plot of all built model accuracy for all built classification models.
- The bar plot shows that Decision Tree Model has the highest classification accuracy.

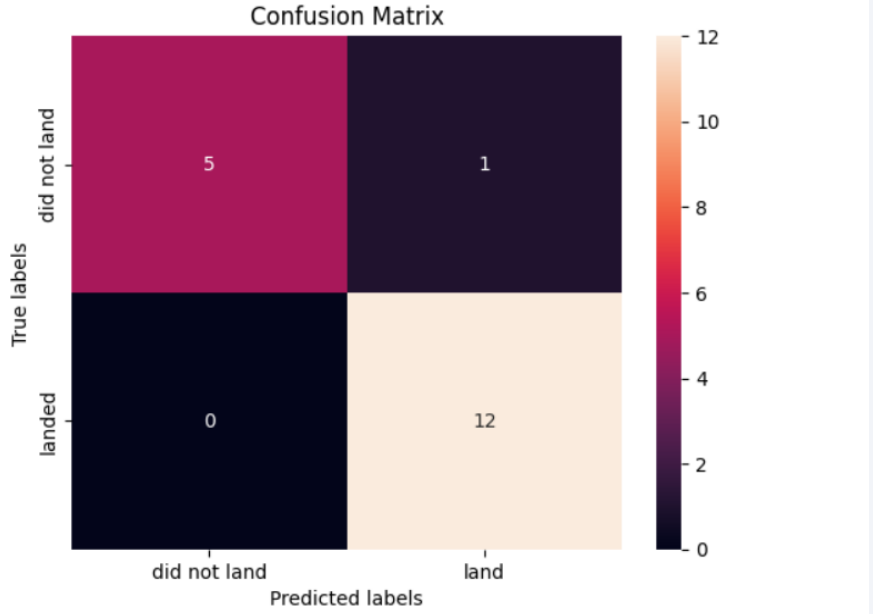


Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation:

The best performing model is the Decision Tree model: False Positive (FP) = 1.

```
In [47]: yhat = tree_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```



Conclusions

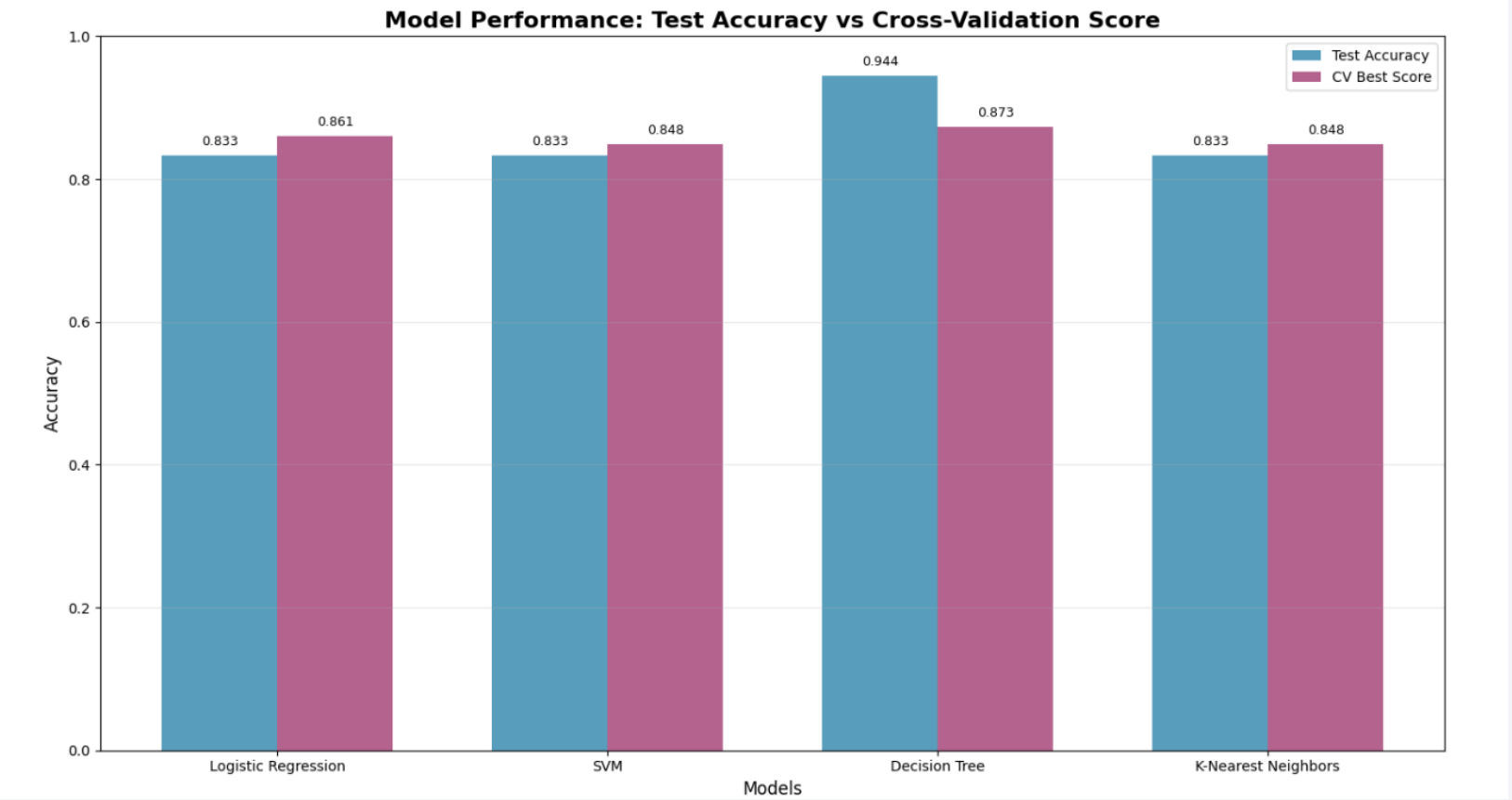
- Point 1) **Data Collection Excellence:** Successfully aggregated data from SpaceX API, secondary endpoints, and Wikipedia scraping.
- Point 2) **Modeling Success:** Achieved exceptional prediction accuracy across multiple algorithms, Identified **Decision Tree** as the most effective predictor with more than **94%** test accuracy, and leveraged GridSearchCV for systematic parameter tuning across all models
- Point 3) T**echnical Implementation:** We used our knowledge to perform the analysis
- Point 4) **Immediate Practical Applications:** Enable SpaceX to assess landing success probability before launch commitments, inform booster recovery strategy decisions, provide quantitative metrics for mission success likelihood, and support decisions on booster refurbishment vs. new construction.
- Point 5) **Limitations and Future Work:**
    - Data Scope was limited to historical launches data; real-time operational factors not included. Feature Availability is dependent on publicly accessible parameters, and temporal Factors like Weather conditions and real-time anomalies not incorporated in our analysis.
    - Advanced Features such as Integrated real-time weather data, mechanical sensor readings, model sophistication like exploring ensemble methods, neural networks, time-series analysis, and operational integration such like developping API for real-time prediction during mission planning will be great.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Appendix 1

Model Performance: Test Accuracy vs Cross-Validation Score.



Appendix 2

Cursor moving on the map and reading location: MoussePosition().

Thank you!