



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Michael Norton  
17 April 2020



# Outline

---

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

# Executive Summary

---

- Summary of methodologies
  - **Data Collection** Used both a REST API and Data Scraping to ingest SPACEX data
  - **Data Wrangling** Performed Preprocessing Steps needed to clean data and create Target Class
  - **Exploratory Data Analysis (EDA)** Used Visualization and SQL techniques to gain insights regarding time series trending, geographical mapping representations and the impact of payload, launch site and time on success rates
  - **Dashboard Development** Create a Dashboard using Plotly that allows users to visualize and apply dynamic filters to answer specific questions and draw custom insights
  - **Model Development** Train and Predict Success or Failure of Launches using K Nearest Neighbors, Support Vector Machines, Logistic Regression and Decision Tree algorithms. Use Gridsearch to select best hyperparameters and algorithms.
  - **Predict and Measure** Perform Inference and Assess Model Performance.
- Summary of all results
  - Some Orbits have perfect landing success rates: SSO, HEO, ES-L1 and GEO
  - Launch Success Rates continue to improve over the course of the launch history
  - The best site based on success rate for landing is KSC-LC-39A

# Introduction

---

- Project background

SpaceX competes in providing commercial launches serving satellite launch needs at a highly competitive cost. An individual launch costs \$62 Million while competitors typically charge \$165 Million or more. The main reason for the huge advantage is the ability to recover and reuse the first stage. With this project, we will explore the launch and landing data, build multiple models, compare effectiveness of each model and select a “best” model to predict launch success. The insights and model will support improved launch and landing success and differentiate pricing based on customer preferences for orbit and payloads. .

- Project objectives

- Identify success rates based on multiple criteria including Launch site, landing site, payload, orbit
- Validate the improvement of launches over time
- Provide Insights to our stakeholders through visualizations, a dynamic dashboard and a presentation
- Make a recommendation to support pricing differentiation and continued reduction in launch failure rates.



Section 1

# Methodology

# Methodology

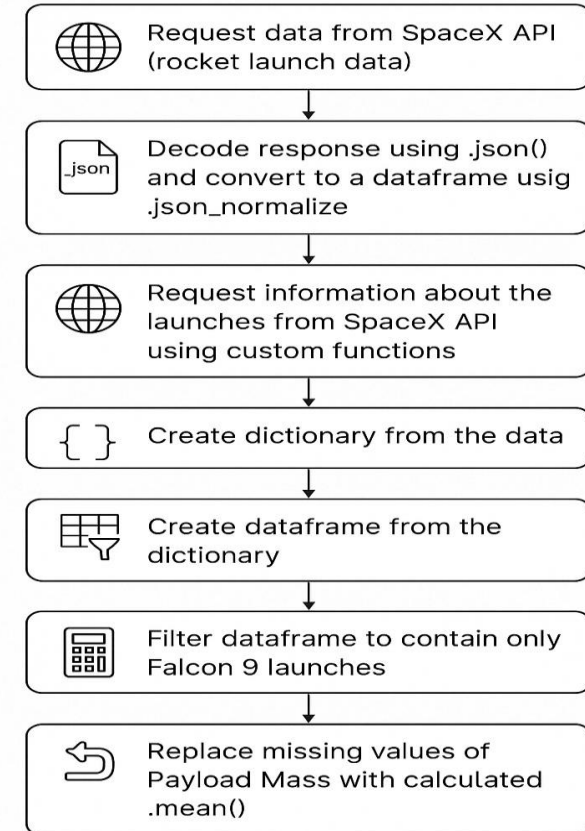
---

- **Data Collection** Used both a REST API and Data Scraping to ingest SPACEX data
- **Data Wrangling** Performed Preprocessing Steps needed to clean data and create Target Class
- **Exploratory Data Analysis (EDA)** Used Visualization and SQL techniques to gain insights regarding time series trending, geographical mapping representations and the impact of payload, launch site and time on success rates
- **Dashboard Development** Create a Dashboard using Plotly that allows users to visualize and apply dynamic filters to answer specific questions and draw custom insights
- **Model Development** Train and Predict Success or Failure of Launches using K Nearest Neighbors, Support Vector Machines, Logistic Regression and Decision Tree algorithms. Use Gridsearch to select best hyperparameters and algorithms.
- **Predict and Measure** Perform Inference and Assess Model Performance.

# Data Collection – SpaceX API

- I used a Get request to access the API and perform the wrangling and formatting steps to create the data frame, clean the data and filter to the relevant Falcon 9 data
- Github Link to code and output for API Collection:  
[https://github.com/mjn82/IBMcaps tone/blob/main/jupyter-labs-spacex-data-collection-api%20\(1\).ipynb](https://github.com/mjn82/IBMcaps tone/blob/main/jupyter-labs-spacex-data-collection-api%20(1).ipynb)

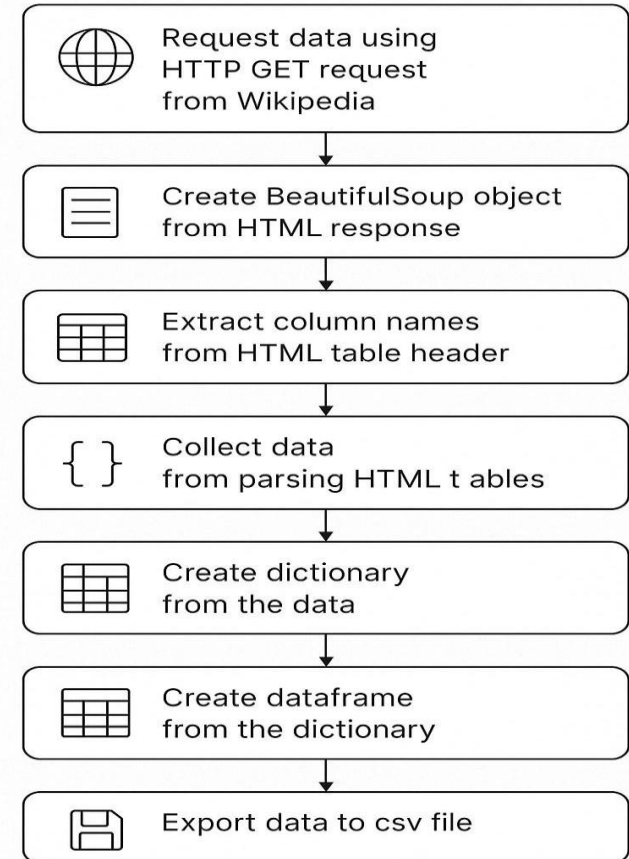
## SpaceX API Data Extraction Workflow



# Data Collection - Scraping

- I used an HTTP Get request and BeautifulSoup to scrape the data from the website and then formatted into a data frame and csv.
- Github Link to code and output for Scraping:  
[https://github.com/mjn82/IBMcapstone/blob/main/jupyter-labs-webscraping%20\(2\).ipynb](https://github.com/mjn82/IBMcapstone/blob/main/jupyter-labs-webscraping%20(2).ipynb)

## Wikipedia Data Extraction Workflow

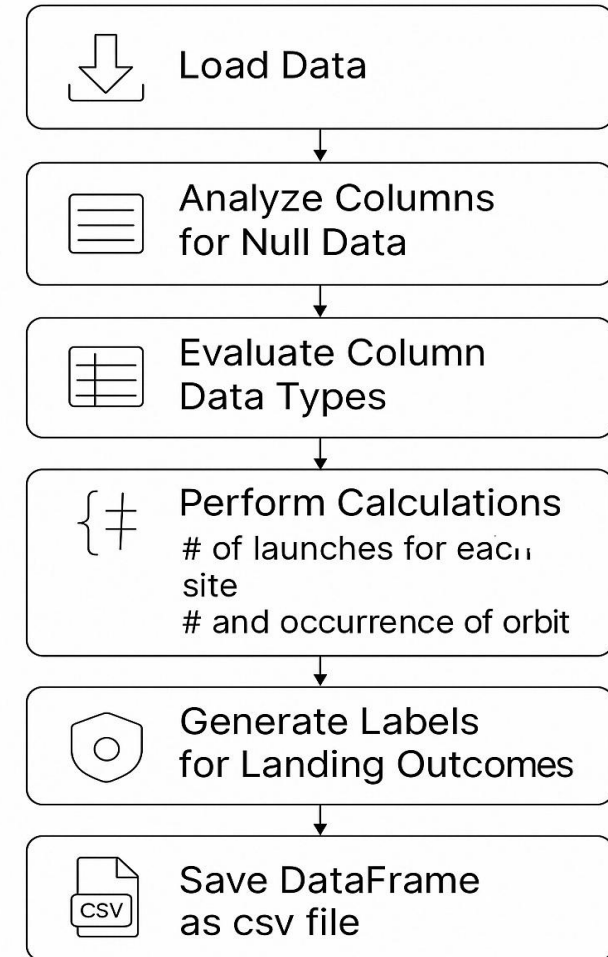




# Data Wrangling






- I evaluated the dataframe for null values (did not treat the one column 'landing pads' which had 29% null values, assessed data types, performed some calculations needed to answer questions and prep for visualizations and then generating my label column for my target variable 'Landing Outcomes')
- Github Link to code and output for Wrangling:  
<https://github.com/mjn82/IBMcapstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb>

## Data Wrangling



# EDA with Data Visualization

- The following chart shows most of the visualizations I generated, I uaws scatter plots for trends and patterns, a bar plot to compare outcomes by category and a line plot to show trends over time.
- Github Link to code and output for Data Visualization:  
<https://github.com/mjn82/IBMcapstone/blob/main/edadataviz.ipynb>

Visualization	Purpose
Flight Number v Launch Site	Detect launch distribution trends by site and over time 
Launch Site v Payload	Explore payload capacity variations across different launch 
Success Rate by Orbit Type	Compare landing success rates among orbit types 
Flight Number v Orbit	Examine how orbit choices have changed over time 
Average Launch Success by Year	Investigate whether different orbit types are associated with different payloads 

# EDA with SQL

---

- Used SQLAlchemy to perform sql against PostgreSQL tables. Below are sample tasks and queries performed.
- Display Names of Launch Sites | `%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;`
- List Total Number of Missions by Outcome

```
%sql SELECT "Mission_Outcome", COUNT(*) AS Total_Missions FROM SPACEXTABLE GROUP BY "Mission_Outcome";
```

- List Date of first successful landing on ground pad

```
%sql SELECT MIN("Date") AS First_Successful_Landing FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';
```

- The entire set of SQL tasks and SQL can be found at this Github :  
[https://github.com/mjn82/IBMcapstone/blob/main/jupyter-labs-eda-sql-coursera\\_sqllite%20\(2\).ipynb](https://github.com/mjn82/IBMcapstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(2).ipynb)

# Build an Interactive Map with Folium

---

- Using the Folium Package, I added various markers overlaying a map of the US with zoom, label and popup features
- Added blue circle at NASA Johnson Space Center's with popup showing its name
- Added red circles at all launch sites coordinates with a similar popup label
- Added green and red colored markers of successful and unsuccessful launches at each launch site to show success rates
- Proximity markers
- Added colored lines to show distance between launch site CCAFS SLC-40 and its proximity to the nearest coastline, railway, highway, and city

# Build a Dashboard with Plotly Dash

---

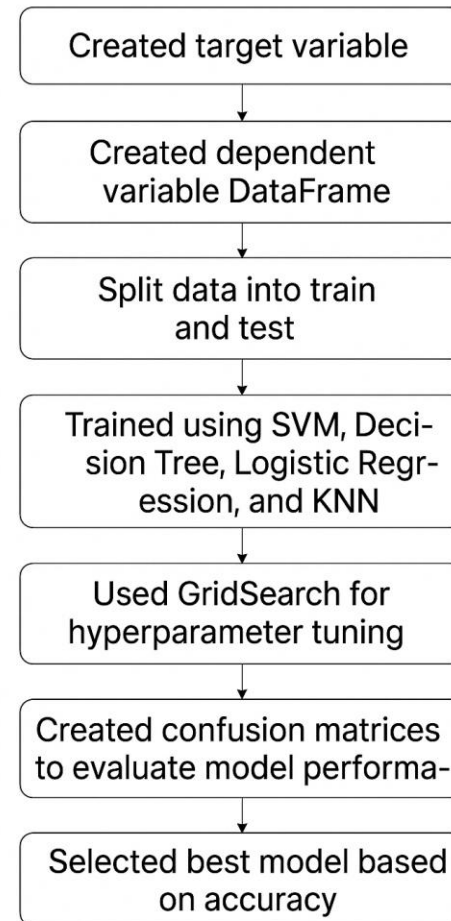
- Using Plotly Dashboard, I created a dashboard with the following features
- A pie chart showing launches success rates
  - Added a dropdown so all of individual sites could be displayed
- A scatter plot showing success rates based on payload
  - Added a slider that allowed filtering of the payload range
  - Applied a color legend to differentiate booster versions
- Github Link to code:  
<https://github.com/mjn82/IBMcapstone/blob/main/spacex-dash-app.py>



# Predictive Analysis (Classification)

- Prepared the target variable and the dataframe with dependent variables, split the data into test and train, trained using SVM, Decision Tree, Logistic Regression and KNN. Used Gridsearch for hyperparameter tuning. Created Confusion Matrices to evaluate model performance . Selected best model based on accuracy.
- Github Link to code and output:  
[https://github.com/mjn82/IBMcapstone/blob/main/SpaceX\\_Machine%20Learning%20Prediction\\_Part\\_5.ipynb](https://github.com/mjn82/IBMcapstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb)

## ML Prediction



# Results Summary Contents

---

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



The background of the slide is an abstract composition. It features a dark blue field on the left side, which transitions into a complex pattern of diagonal streaks in shades of blue, red, and cyan on the right. These streaks have a textured, almost woven appearance. Overlaid on this pattern is a faint, light blue grid that recedes into the distance, creating a sense of depth and perspective.

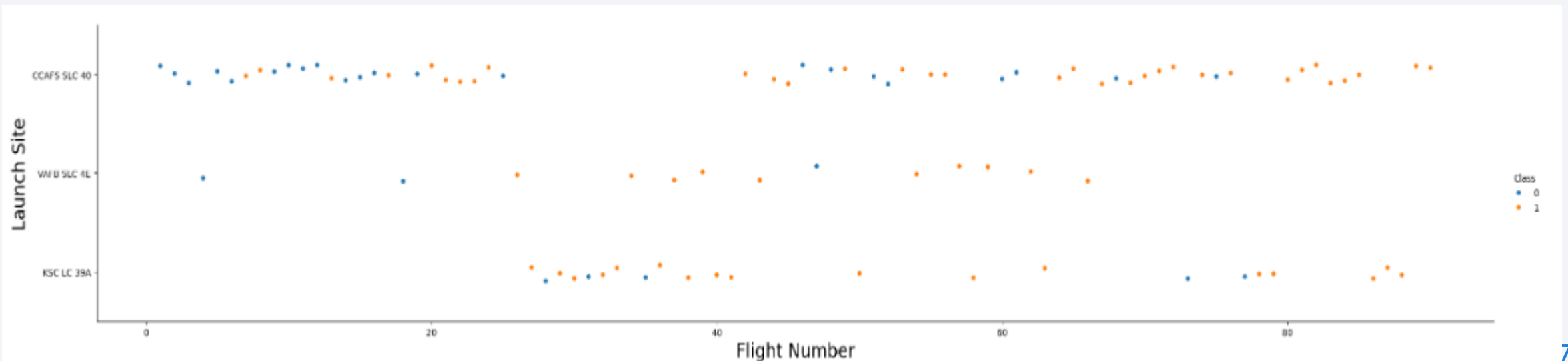
Section 2

# Insights drawn from EDA



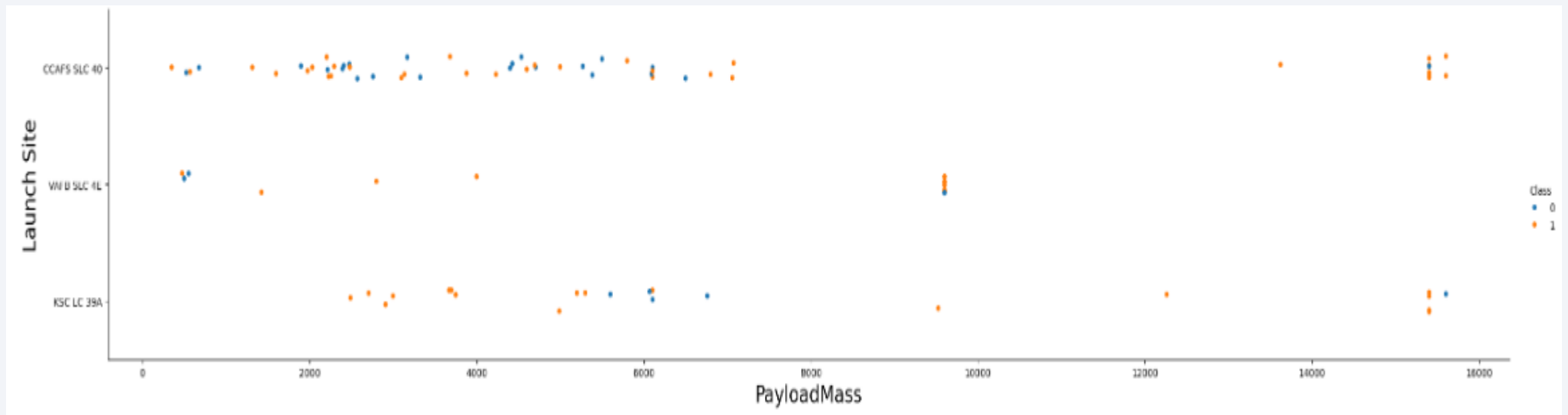
# Flight Number vs. Launch Site

- Success Rates improved significantly over time
- More than half the launches were from CCAFS SLC 40
- CCAFS SLC 40, the first launch site had significant failures in its first launches
- The other two sites may have benefited from the earlier CCAFS SLC 40 launches



# Payload vs. Launch Site

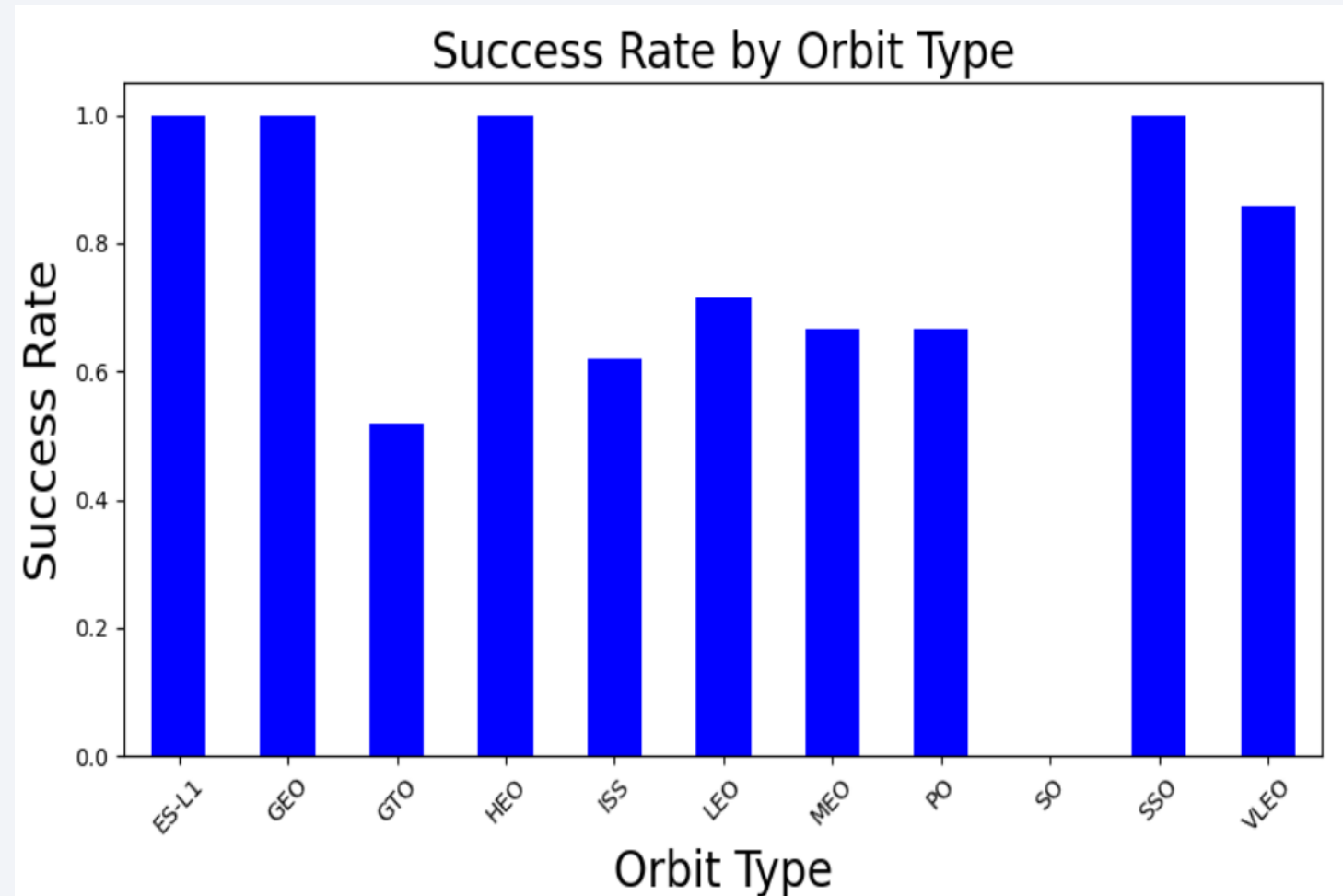
- VAFB SKC 4E has not launched payloads above 10,000 kg
- KSC LC 39A has great success with smaller payloads (< 6,000 kg)
- Larger payloads are more successful than smaller payloads





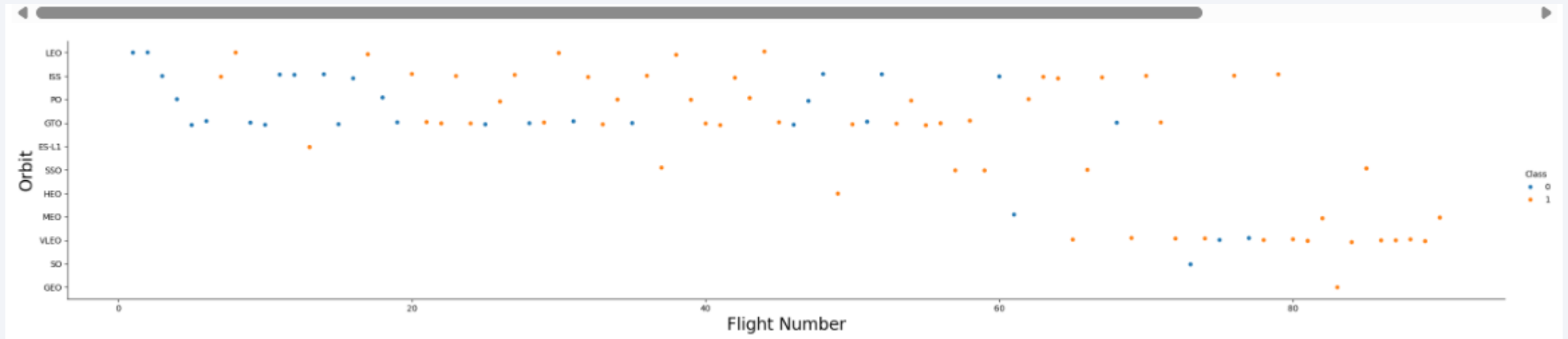
# Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO had perfect success rates
- GTO, ISS, LEO, MEO, PO and VLEO were successful at least half the time.
- There were no successful launches from the SO site



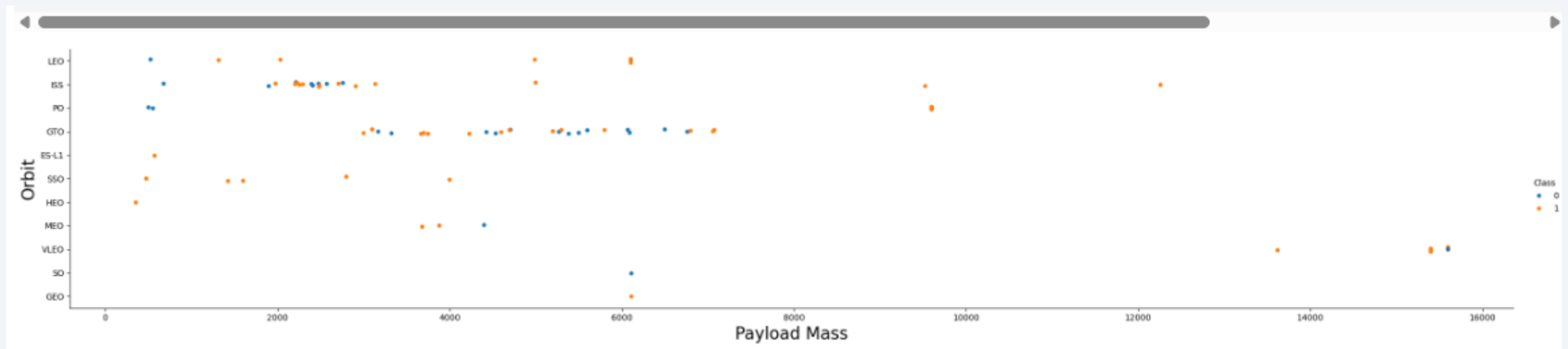
# Flight Number vs. Orbit Type

- This plot shows that landing success improves over time.
- GTO shows less improvement over time while the LEO orbit shows the most improvement over time



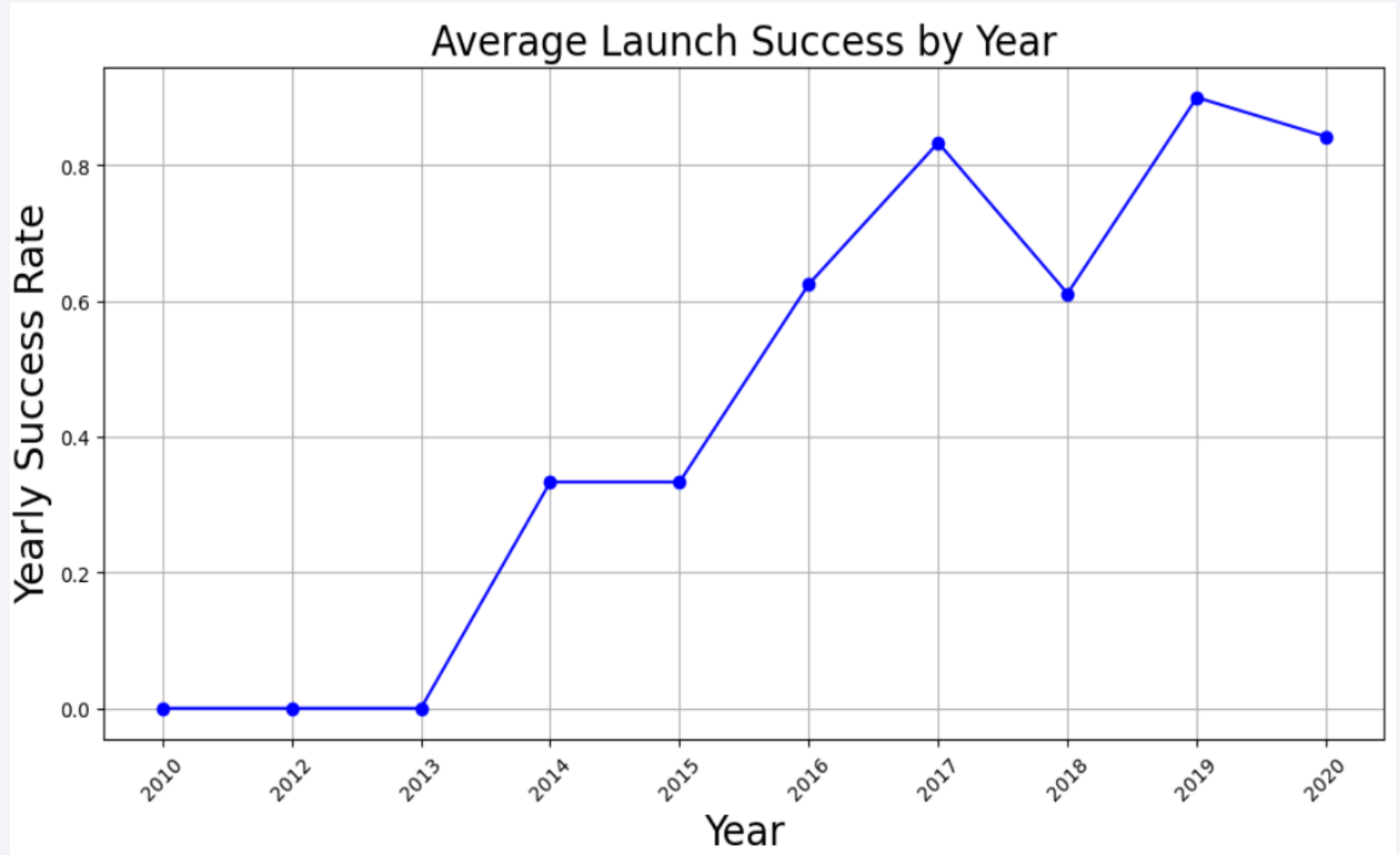
# Payload vs. Orbit Type

- LEO, ISS and PO orbits have the best success rates with the higher payloads
- The GTO orbit has does not show a trend with heavier payloads. It is a mix of success and failure



# Launch Success Yearly Trend

- Success rates generally increase over time
- There was no improvement in the first 3 years of the program
- In 2018 there was a year over year regression in success rates by 20 percent and again a drop in 2020.



# All Launch Site Names

---

- The DISTINCT command in the select statement returns unique strings from the target table and column

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
```

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



# Launch Site Names Begin with 'CCA'

- The condition clause below uses the LIKE command and the LIMIT Command to find just 5 rows that meet the condition including CCA%. If there were strings where CCA% was not just at the beginning of the string I would have used a more complex command.

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

```
%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

# Total Payload Mass

---

- I use the SUM command to aggregate the payload mass and the where condition to apply it only to the string in the customer for NASA
- I name the new attribute total payload mass with the AS command

```
%sql SELECT SUM("Payload_Mass__kg_") AS total_payload_mass FROM SPACEXTABLE WHERE "Customer" LIKE '%NASA (CRS)%';
```

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

<b>total_payload_mass</b>
---------------------------

48213
-------

# Average Payload Mass by F9 v1.1

---

- I use the AVG command to obtain the average the payload mass and the where condition to apply it only to the string in the Booster Version column for F9 v1.1
- I name the new attribute average payload mass with the AS command

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

```
%sql SELECT AVG("Payload_Mass__kg_") AS average_payload_mass FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
```

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

<b>average_payload_mass</b>
2928.4

# First Successful Ground Landing Date

---

- I use the MIN command obtain the earliest date in the “date” column and the where condition to apply it only to rows where the “Landing Outcome” is a success
- I name the new attribute First Successful Landing with the AS command

```
%sql SELECT MIN("Date") AS First_Successful_Landing FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

```
Done.
```

<b>First_Successful_Landing</b>
---------------------------------

2015-12-22
------------

## Successful Drone Ship Landing with Payload between 4000 and 6000

---

- Select used for Booster Version with the WHERE condition for column “Landing Outcome” for success(drone ship) AND the Payload mass in the range of 4000 to 6000

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)'
```

```
AND "Payload_Mass__kg_" > 4000 AND "Payload_Mass__kg_" < 6000;
```

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



# Total Number of Successful and Failure Mission Outcomes

---

- Used the COUNT(\*) command along with the GROUP BY “Mission Outcome” to get a count of each unique string in Mission Outcome
- Named the column for counts as Total\_Missions using the AS command

```
%sql SELECT "Mission_Outcome", COUNT(*) AS Total_Missions FROM SPACEXTABLE GROUP BY "Mission_Outcome".
```

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

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

# Boosters Carried Maximum Payload

- Selecting the Booster Versions using a WHERE condition that uses the MAX command on the payload column to display the Versions that had the maximum payload

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Payload_Mass__kg_"  
= (SELECT MAX("Payload_Mass__kg_") FROM SPACEXTABLE);
```

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

# 2015 Launch Records

---

- Selected rows using the WHERE condition for Landing Outcome is a failure and yer was 2015 using substr(Date, 0,5) displaying Month, Landing Outcome, Booster Version and Launch Site

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

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

- Use Count command to get counts for Landing Outcome and WHERE condition to limit between the date 2010-06-04 and 2017-03-20 and DESC command to make descending order
- New column named Outcome Count using AS command

Landing_Outcome	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

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

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

# Launch Sites Proximities Analysis

# Complete Global View & Launch Site Markers

---

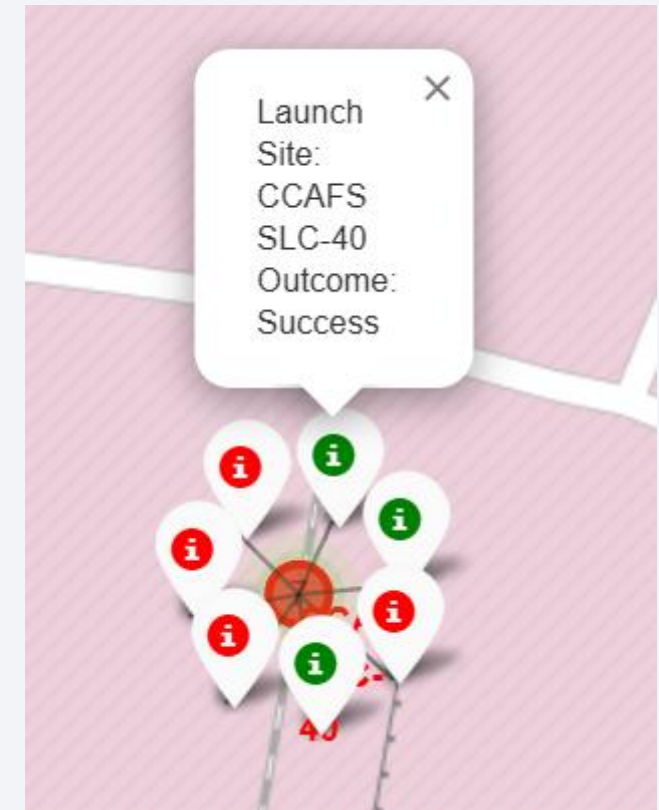
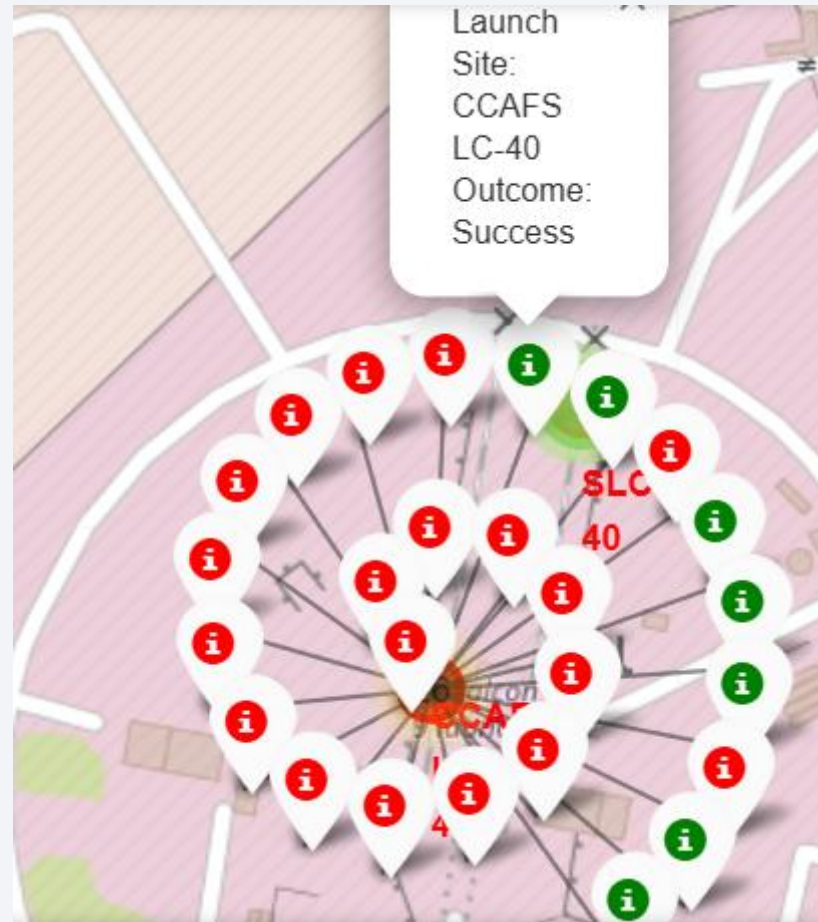
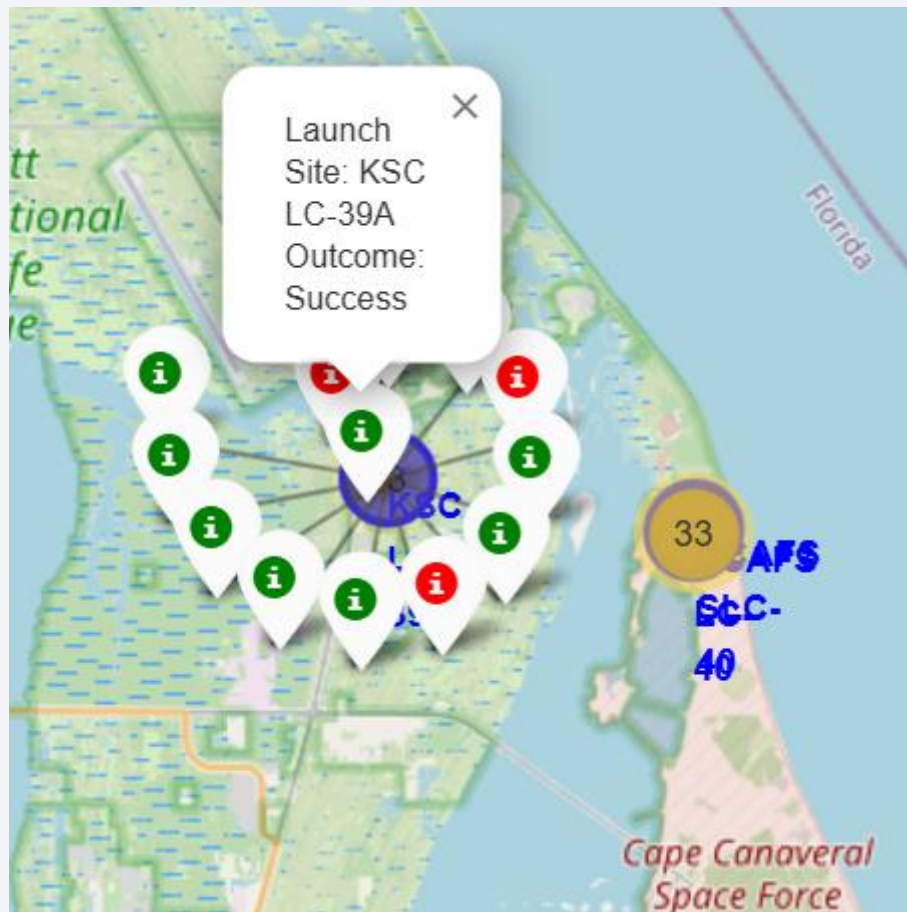


- Launch Sites marked in red and all located in the United States along coastlines and relatively close to the equator



# Florida Launch Sites - Success and Failure by site

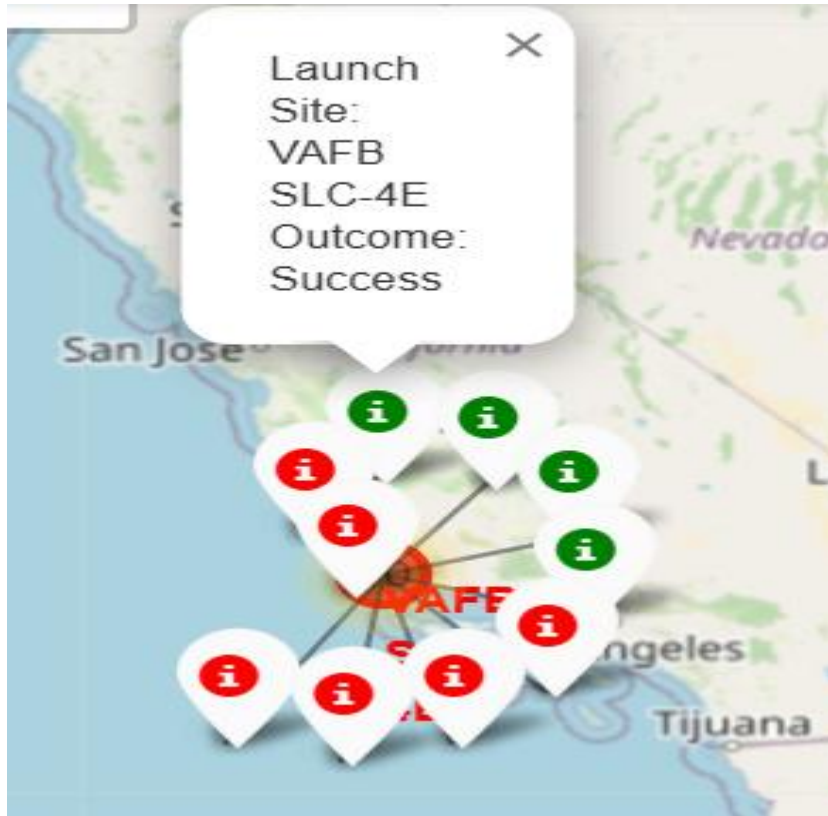
- Success in, Green Failures in Red
- KSC LC-39A has a good success rate while other to sites have more failures than successes





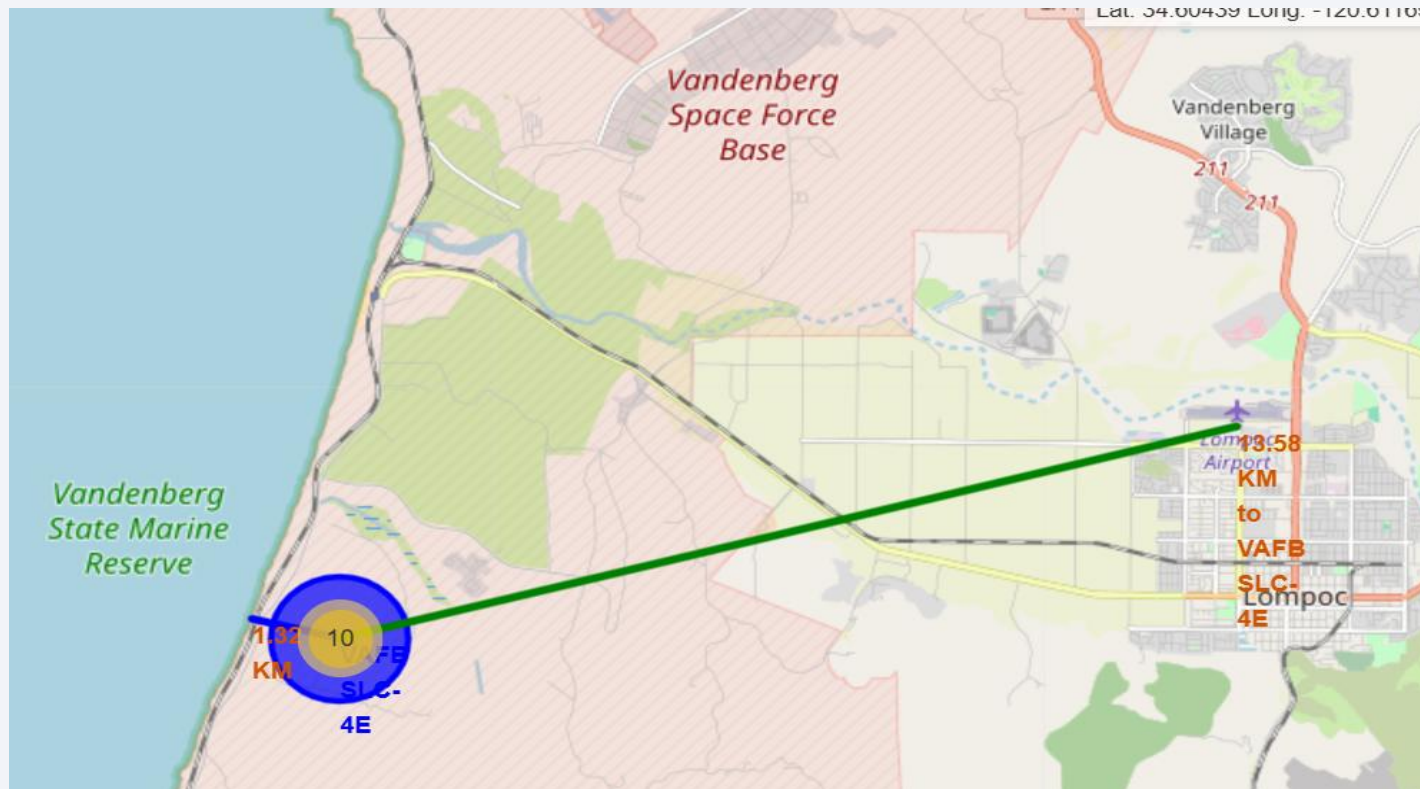
# California Launch Site - Success and Failure by site

- Success in, Green Failures in Red
- VAFB has fewer successes than failures. Note popup appears for any marker clicked on



# Proximity Map for Vandenberg Launch Site

- Vandenberg is 1.3 km to nearest coastline and 13.6 km to nearest airport
- Lines are in Blue and distance in red





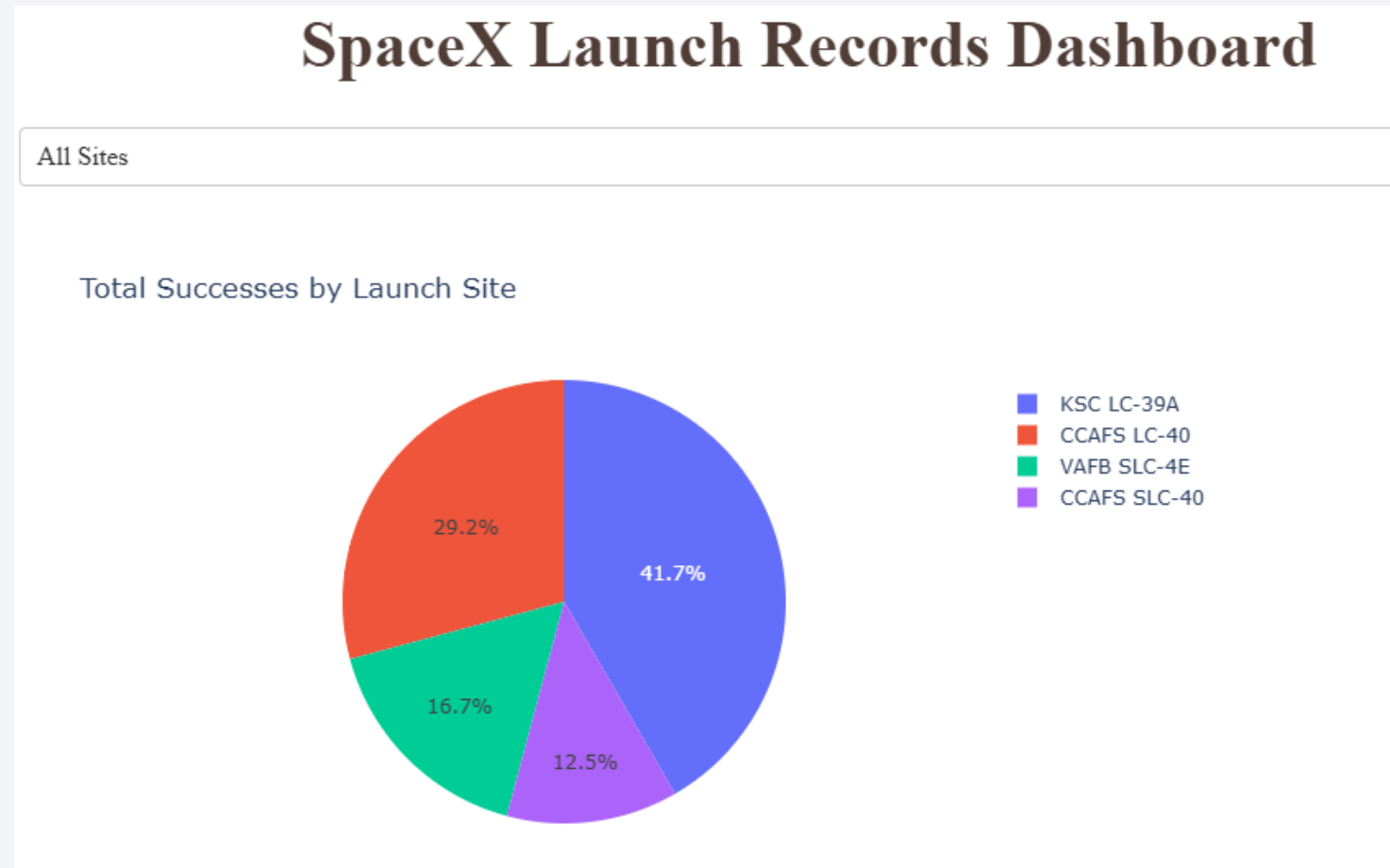
Section 4

# Build a Dashboard with Plotly Dash



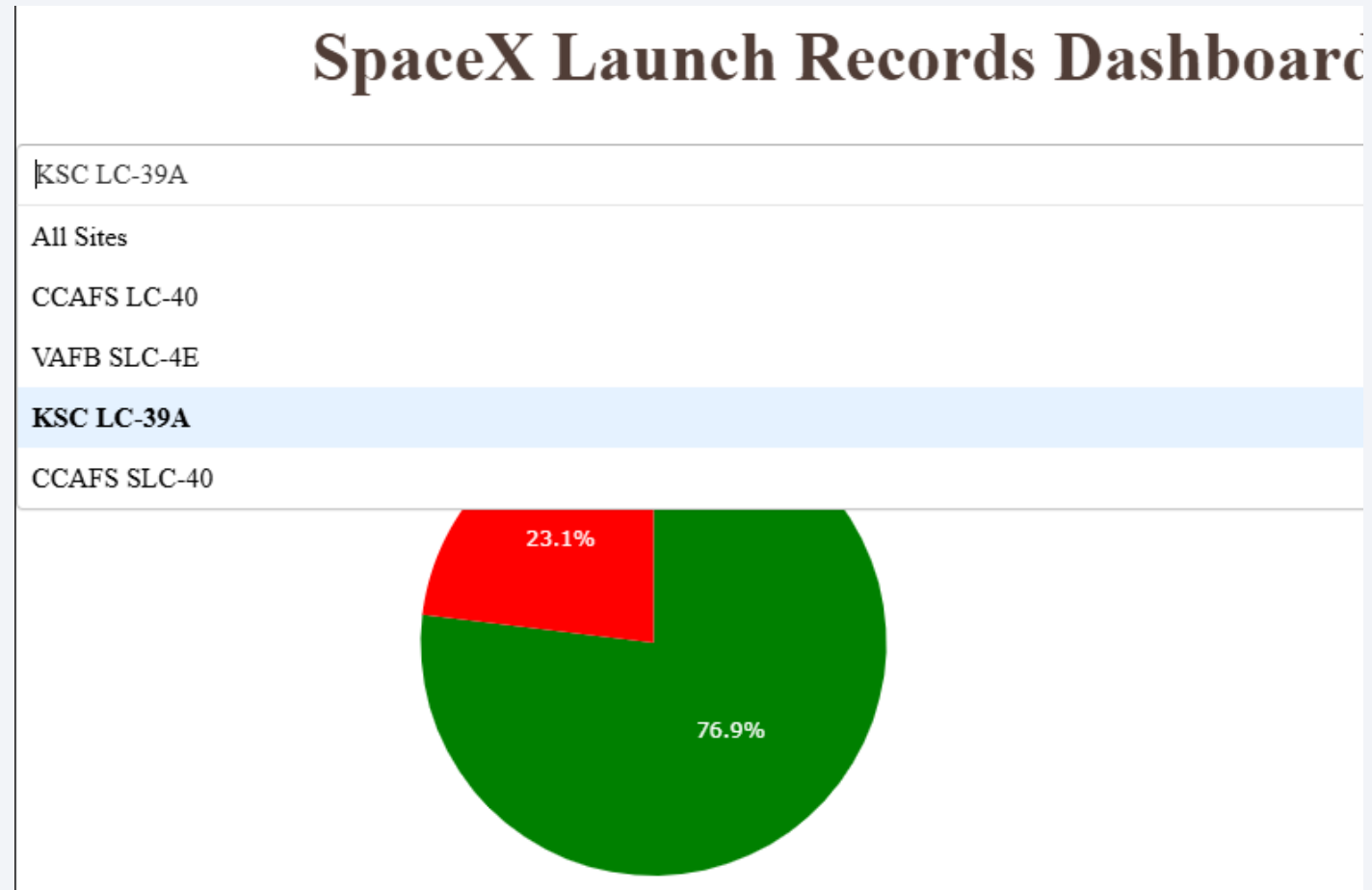
# Launch Success Count - All Sites

- Greatest number of successes was at KSC LC-39A
- Lowest number of successes was at CCAFS SLC-40
- Success count reflects absolute count – NOT the highest ratio of success



# Launch Success Count - Most Successful Site

- KSC LC-39A has best ratio of Successful Launches - 76.9%
- Dropdown included in display so you can see how dropdown selection appears



# Payload v Launch Outcome

- The success rate is highest in the range of 2k – 5.5 kg
- Booster version FT appears to have the best success rate among boosters.



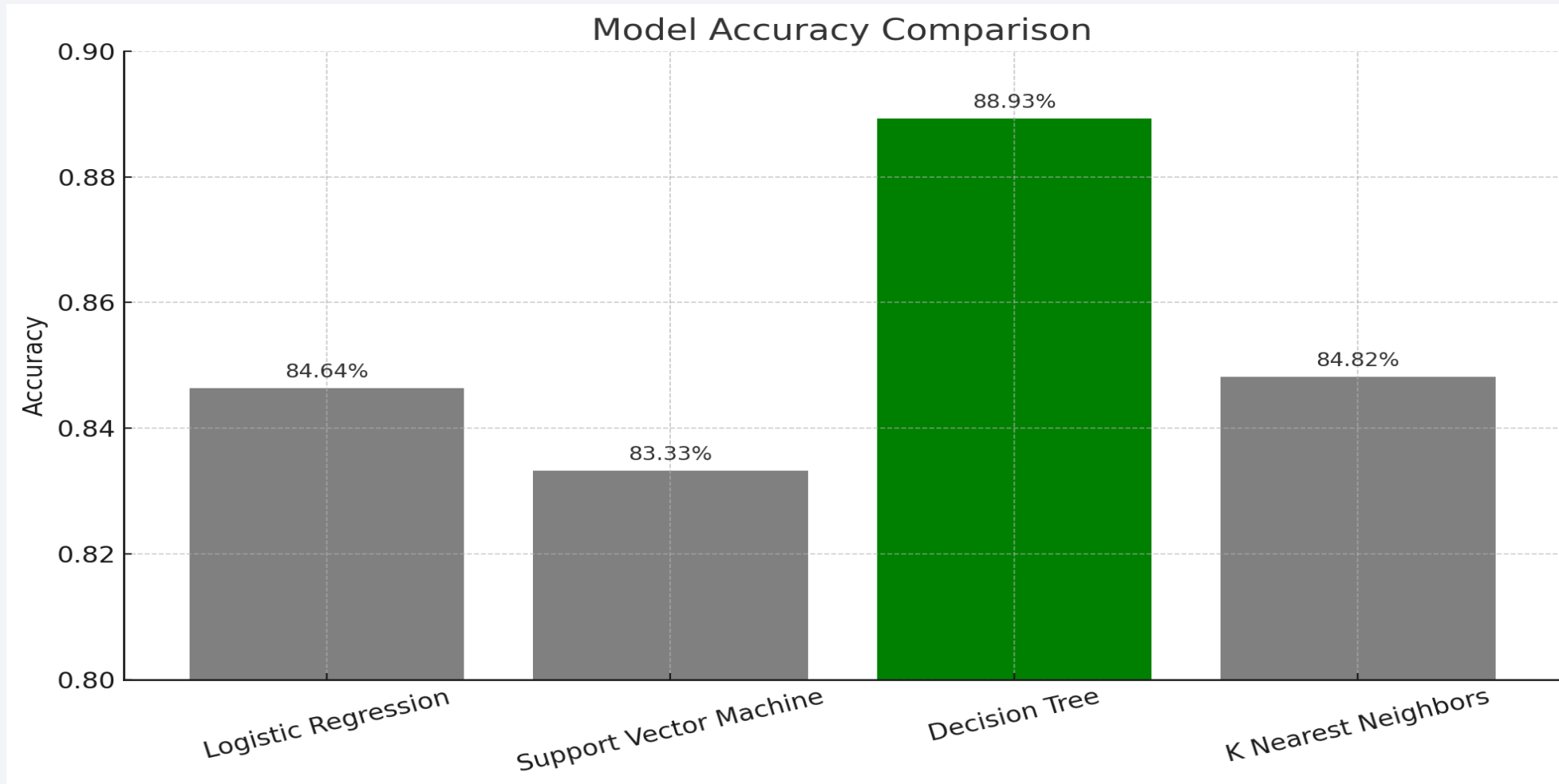
Section 5

# Predictive Analysis (Classification)



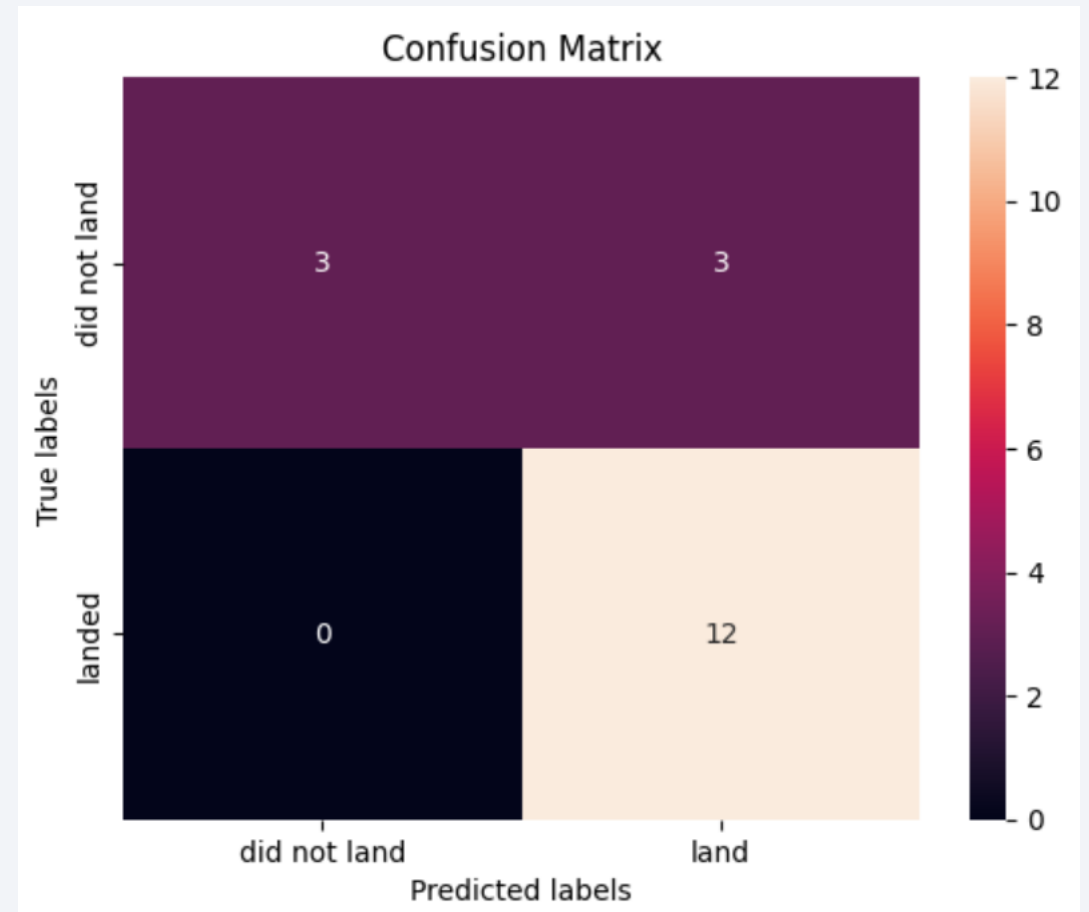
# Classification Accuracy

- The Decision Tree model produced the best results @ 88.93% accuracy



# Confusion Matrix

- Correct Predictions in the confusion matrix appear where True (actual) labels and Predicted Labels Match. So this model predicted all 12 successful landings correctly.
- The model was only able to predict half (3) of the unsuccessful landings correctly.
- This is why accuracy can be misleading as it does a good job overall but is not very good at predicting bad landings correctly
- An F1 would have been a better metric



# Conclusions

---

- Our best model, decision tree is good at predicting successful launches, but has difficulty predicting unsuccessful launches
- Over time, launches became significantly more successful
- Most Launches were from CCAFS SLC 40
- As Payload Mass increased, launches were more successful
- Launches into ES-L1, GEO, HEO, and SSO orbits had no failures
- Launch sites are close to coasts and closer to the equator

Thank you!

