IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alex Van Buren
11/10/2023

Outline

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

Executive Summary

- Gathered data from SpaceX API and web scraping
- Explored the data using visualizations and SQL queries
- Explored Launch locations using folium
  - Warm coastal locations with good infrastructure are ideal
- Made Plotly Dashboard to explore each site's success with varying payloads and booster versions
- Made machine learning classification models to predict landing outcomes
  - Models predicted accurately ~83.3%

Introduction

- Many Companies are launching into Space
- SpaceX is inexpensive because they can reuse rockets
- Use Publicly available data from SpaceX launches
- Use ML to predict if a launch will be successful

Section 1

Methodology

Methodology

Executive Summary

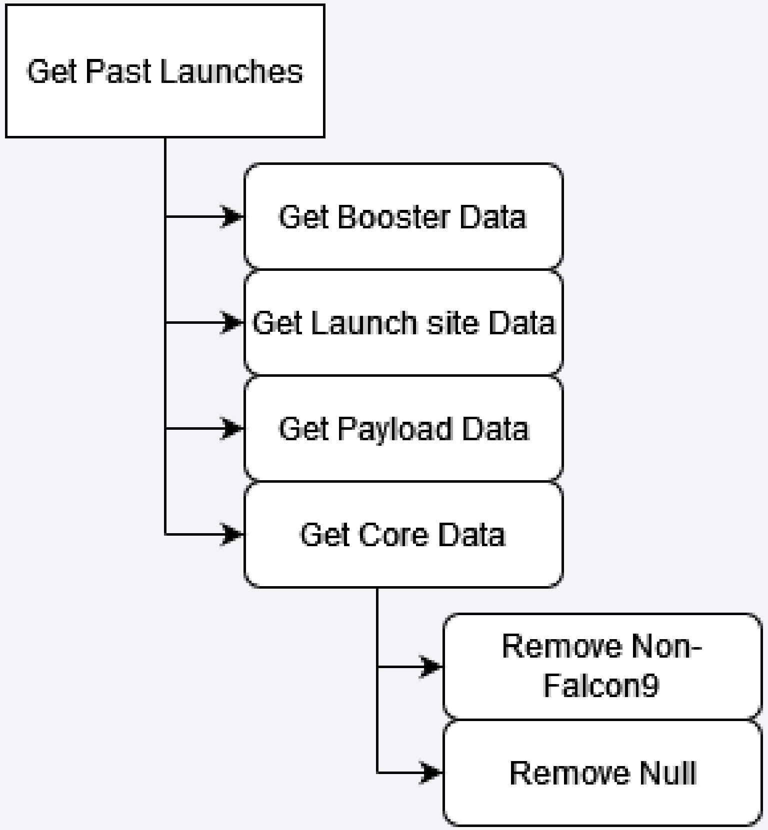
- Data collection methodology:
  - Collect data by using SpaceX API
  - Scrape available data from Wikipedia
- Perform data wrangling
  - Change outcome values from various types to 1 for success and 0 for failure
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Use Grid Search with parameters to find best parameters for each model

Data Collection

- Two methods for data collection were used
  1. SpaceX web API
  2. Web Scrapping from Wikipedia

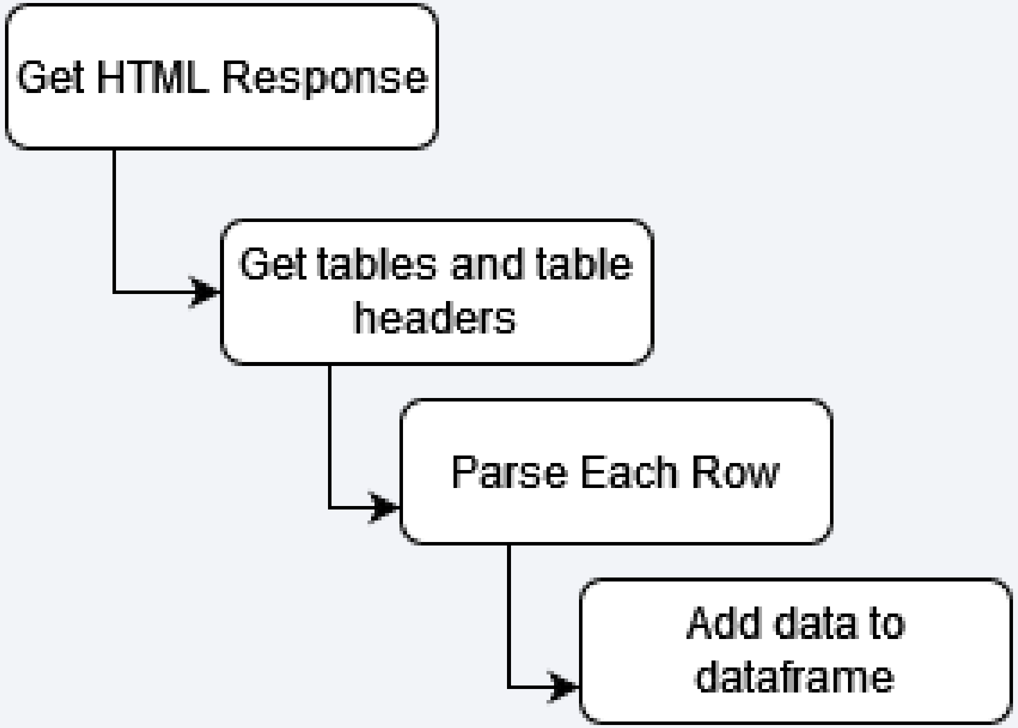
Data Collection – SpaceX API

- Use SpaceX API requests to get JSON of SpaceX launches
- Some of the data returned is just id strings
- Use those id's in further API calls to get the data for Payload, BoosterVersion, LaunchSites, and Core
- Remove non Falcon9 launches and null data
- https://github.com/Alex-Van-Buren/Applied_Data_Science_Capstone/blob/main/01-jupyter-labs-spacex-data-collection-api.ipynb



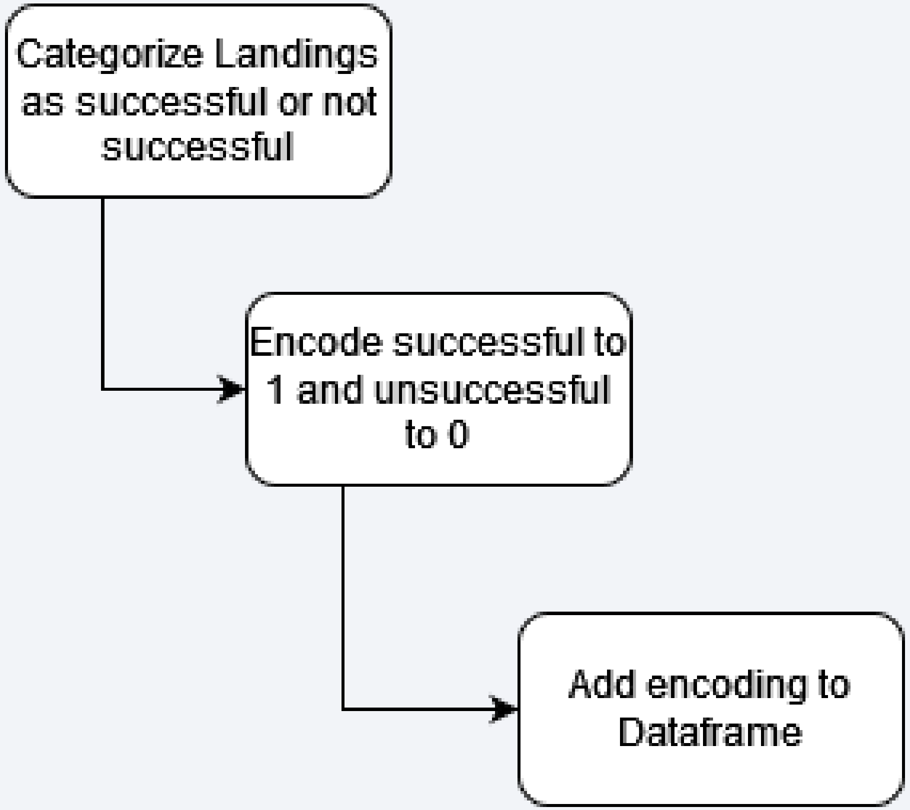
Data Collection - Scraping

- Get HTML response
- Get tables and table headers
- Parse through each row of the table
- Put the data in a pandas dataframe
- https://github.com/Alex-Van-Buren/Applied_Data_Science_Capstone/blob/main/02-jupyter-labs-webscraping.ipynb



Data Wrangling

- Landing Outcomes were categorized as successful or unsuccessful
- Successful were encoded to 1 while unsuccessful were encoded to 0
- The new encodings were added to the dataframe under column name Class
- https://github.com/Alex-Van-Buren/Applied_Data_Science_Capstone/blob/main/03-labs-jupyter-spacex-Data%20wrangling.ipynb



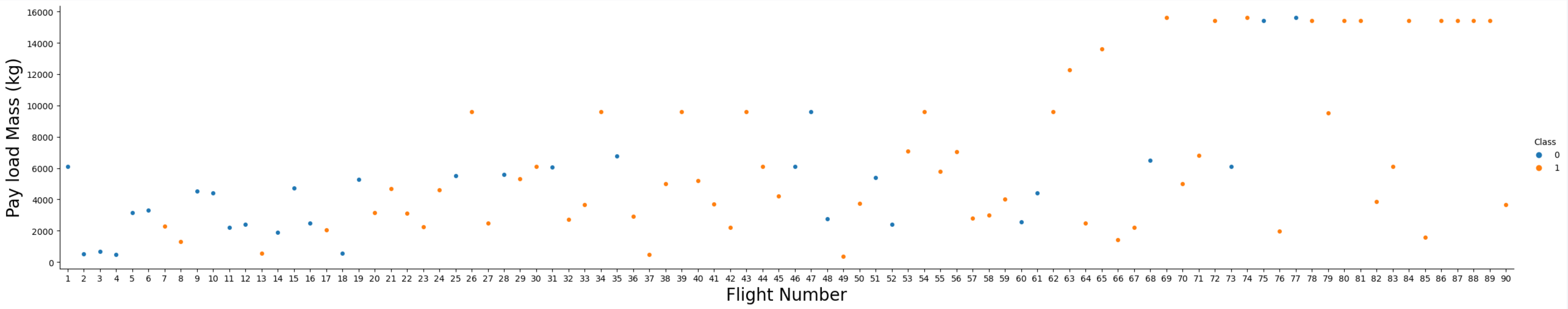
Section 2

Insights drawn from EDA

EDA with Data Visualization

- Various Graphs were made exploring the landing success based on factors such as payload, launch site, Orbit
- Flight number was also explored as the company had more success with more experience (higher flight number)
- https://github.com/Alex-Van-Buren/Applied_Data_Science_Capstone/blob/main/05-jupyter-labs-eda-dataviz.ipynb

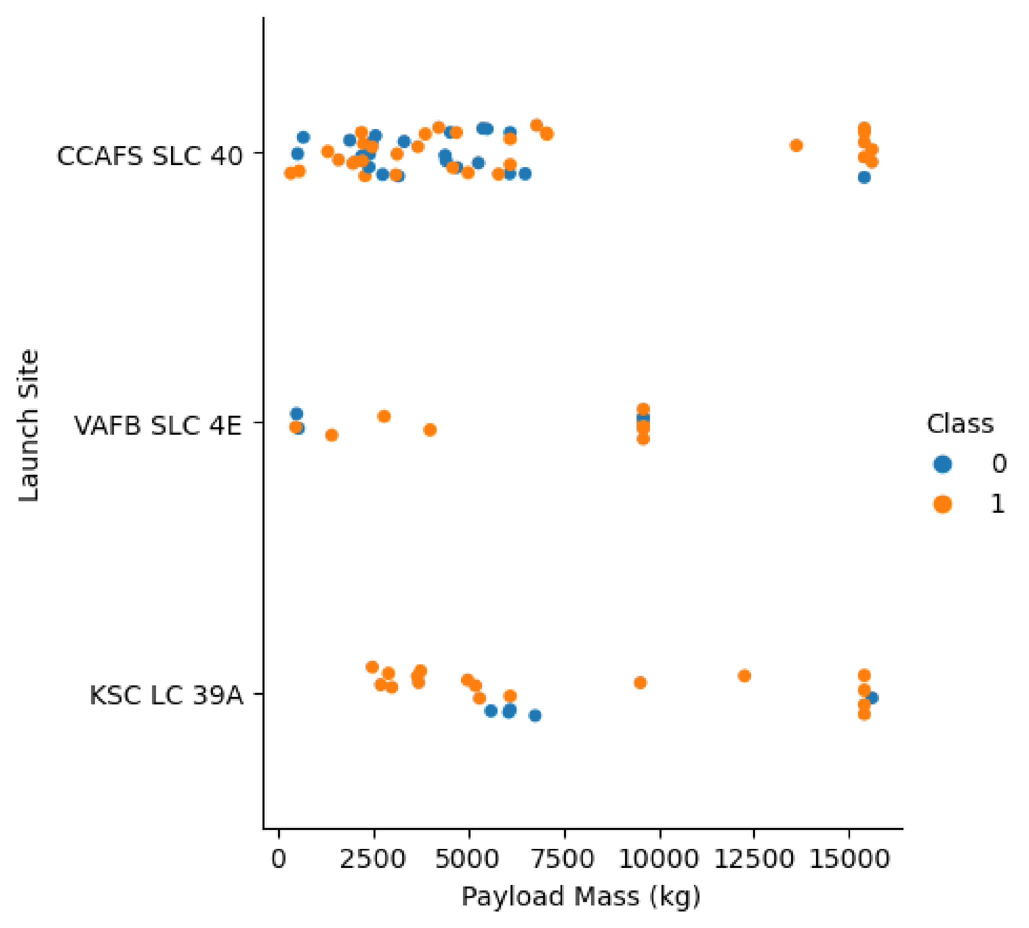
Flight Number vs. Launch Site



- Class of 0 indicates a failed landing, while class 1 is a success
- We can see that successes increase with flight number
- We can see that payload also increases with flight number

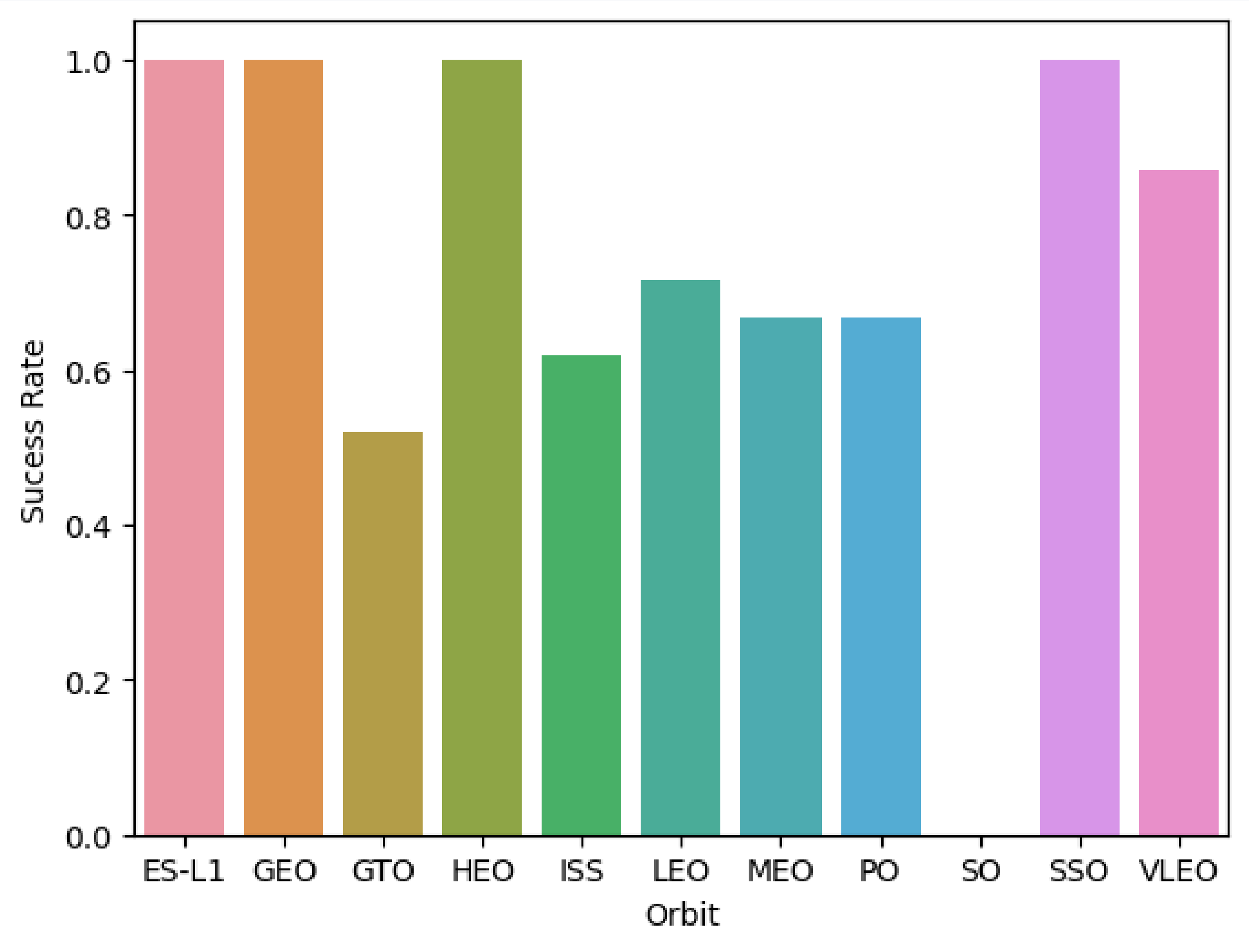
Payload vs. Launch Site

- VAFB SLC 4E site did not exceed 10000kg payload
- Higher payload flights tended to have higher success rates
- CCAFS SLC 40 launches with payload under 10000kg had high failure



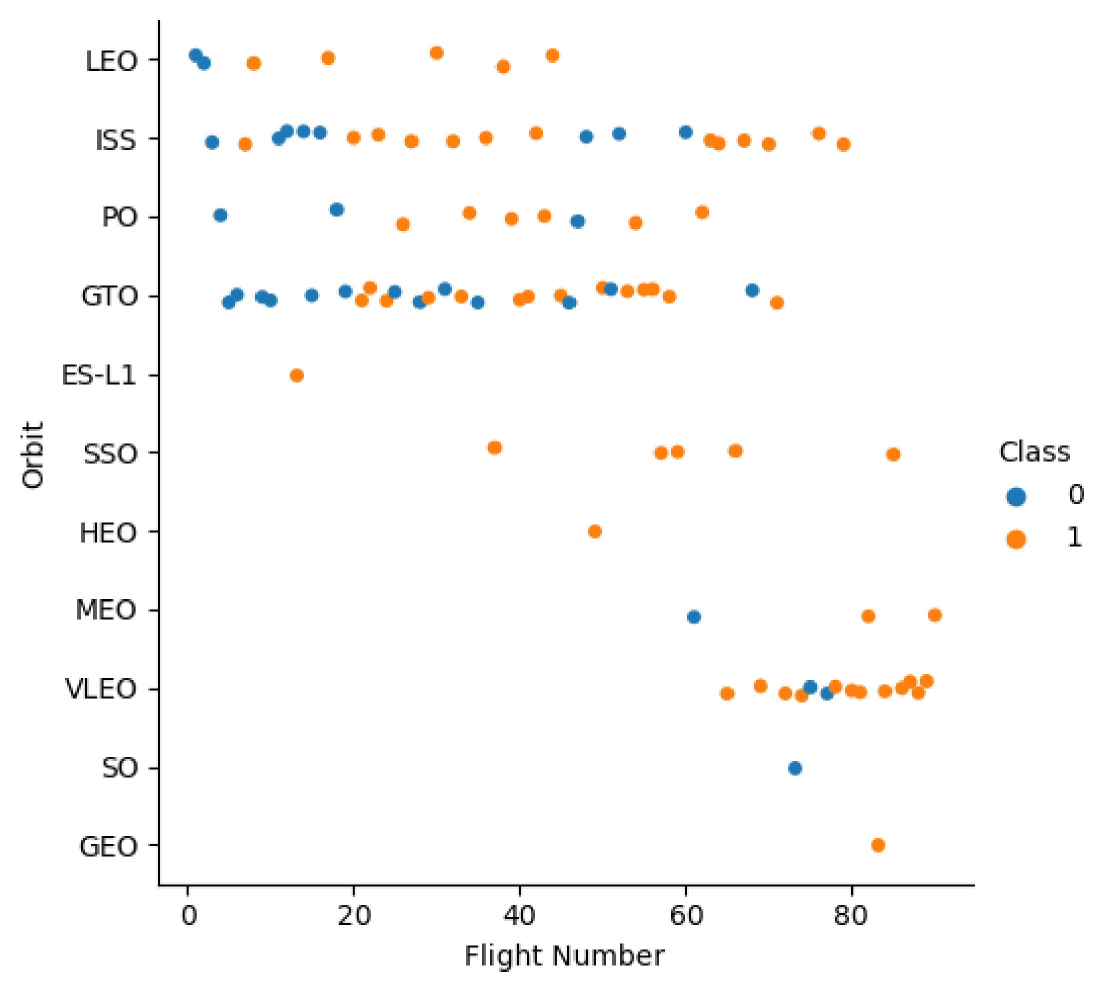
Success Rate vs. Orbit Type

- SSO had 100% success rate
- ES-L1, GEO, HEO also had 100% success rate but only 1 data point
- SO had 0% success rate with only 1 data point

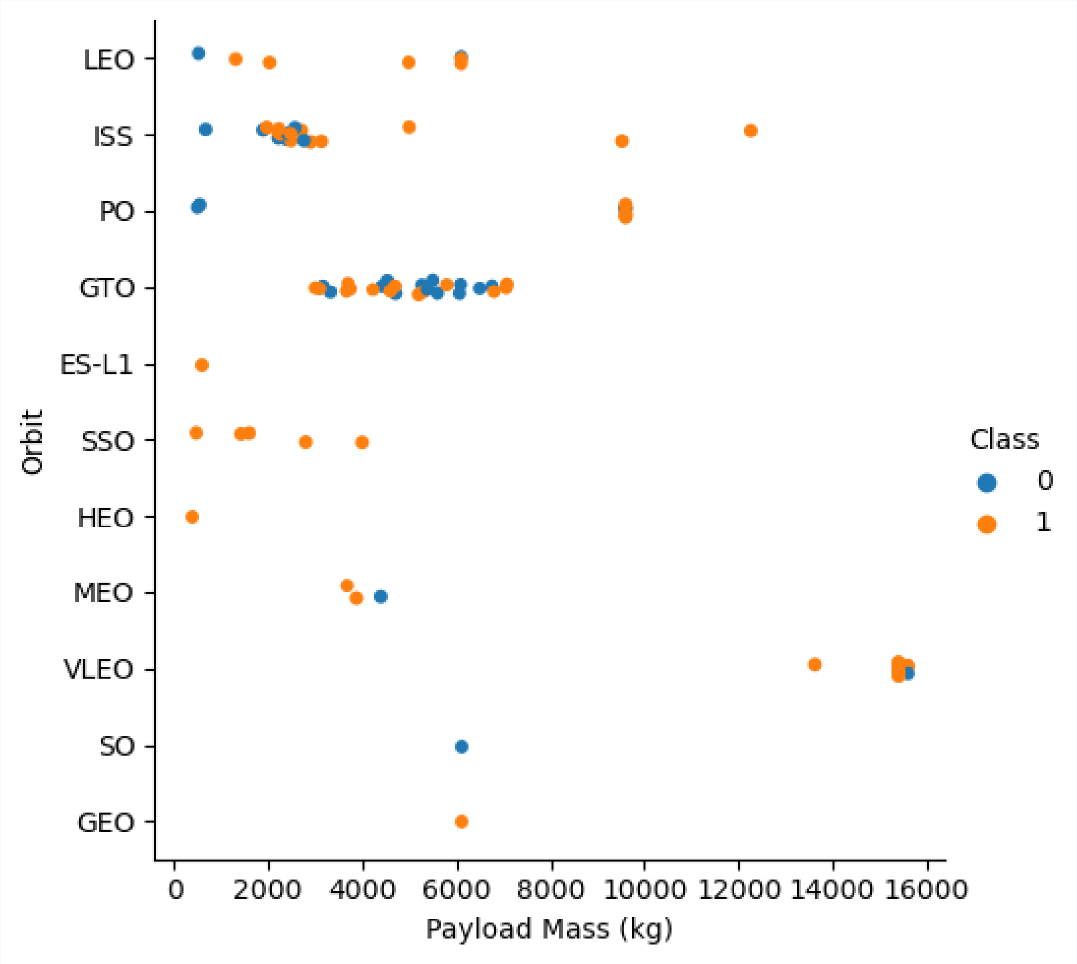


Flight Number vs. Orbit Type

- Most Early flights were ISS, PO, GTO, or LEO
- VLEO's higher success rate(with decent sample size) could be explained by later flights (company has more practice)

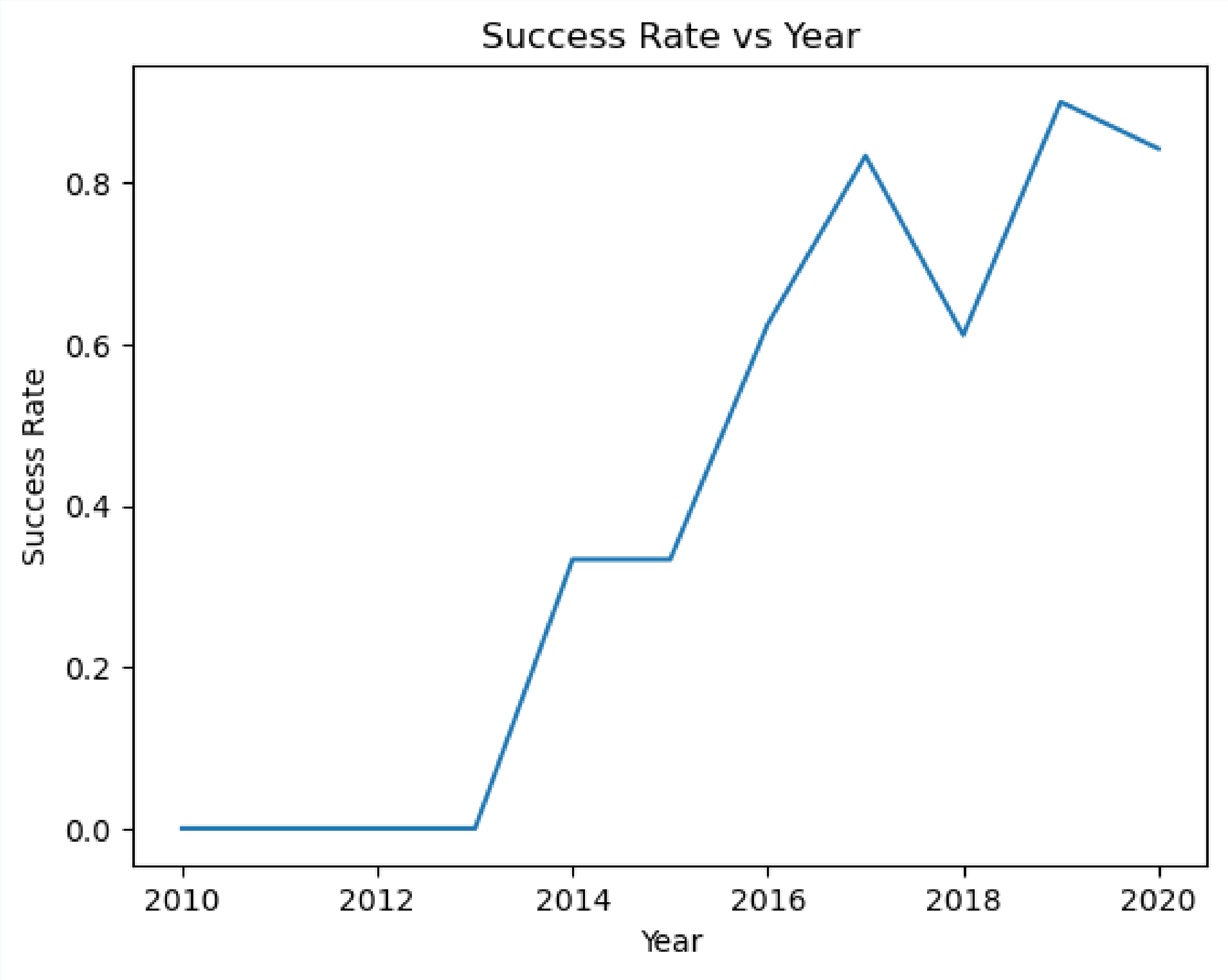


Payload vs. Orbit Type



- Orbits seem to have a payload range
- GTO 3000-8000kg
- VLEO >12000kg
- Most ISS payloads between 2000 and 4000kg

Launch Success Yearly Trend



- General upward trend
- 2018 saw a significant dip in success compared to 2017 and 2019

EDA with SQL

- Various SQL queries about the launch sites, payloads, and dates were explored
- SQL queries are great for parsing through the data and getting simple calculations such as sums and averages.
- https://github.com/Alex-Van-Buren/Applied_Data_Science_Capstone/blob/main/04-jupyter-labs-eda-sql-coursera_sqllite.ipynb

All Launch Site Names

- Used distinct to get the unique launch sites
- This database had 1 more launch site than the previous data.
- CCAFS LC-40 was not present in the previous graphics

```
%%sql
SELECT DISTINCT "Launch_Site" from SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

- Found 5 records where launch sites begin with `CCA`
- Used Like operator to get records that start with CCA and limit to only get 5

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

Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Used Sum operator to total up the mass

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE "Customer" == 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Used AVG operator to get the average mass

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE "Booster_Version" == 'F9 v1.1'
```

[29]

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Used MIN function on 'Date' to get first ground pad success

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

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

MIN("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Between operator used to get correct payload mass

```
%%sql
SELECT Booster_Version FROM SPACEXTABLE WHERE "Landing_Outcome" == 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

```
 * 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

- Totals found by using Count and Group by.
- Unsure why two different Success groups. (Different white space maybe)

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

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

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Used subquery to find max payload which subsequently selected the booster versions.

```
%%sql
SELECT "Booster_version" FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ == (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

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

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Date was string with year at the beginning, so I selected for year by using like operator.
- Sqlite doesn't support month names so I displayed month numbers

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

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

Month	Year	Booster_Version	Launch_Site	Landing_Outcome
10	2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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 date 2010-06-04 and 2017-03-20, in descending order
- Used between operator with date.
- Grouped by landing outcome ordered by the count.
- Can see that not attempting to land was the highest outcome

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

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

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

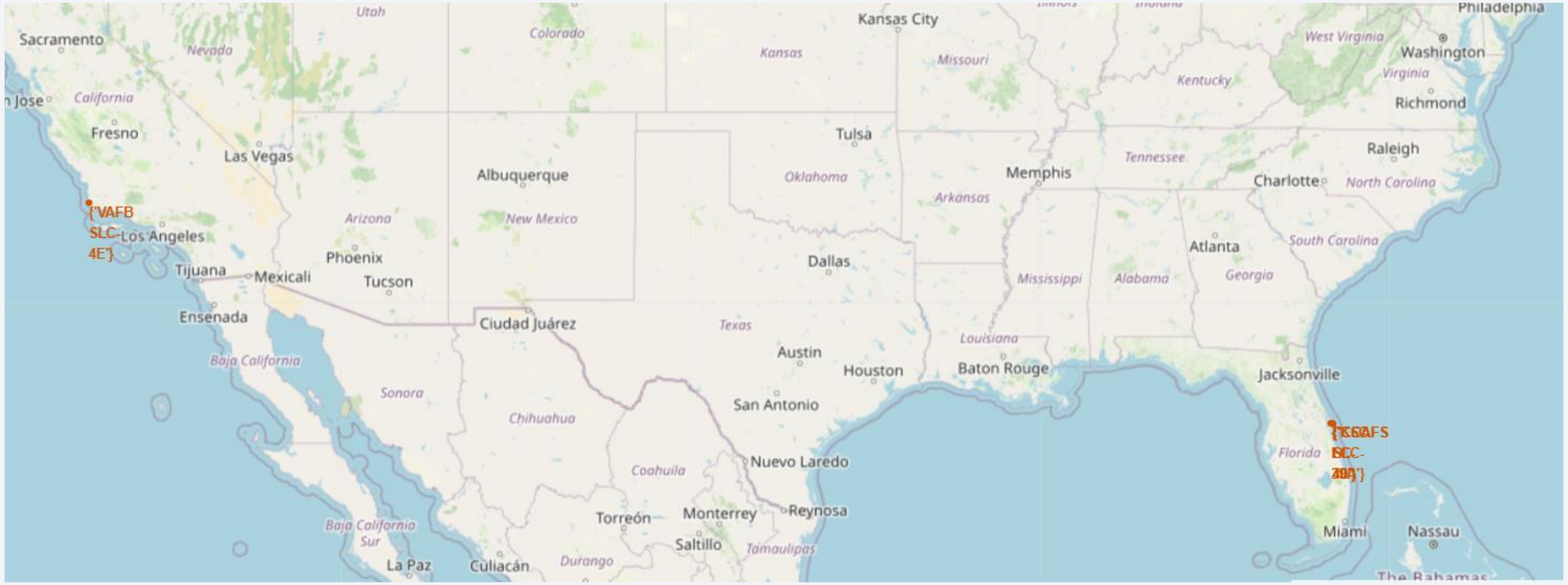
Section 3

Launch Sites Proximities Analysis

Build an Interactive Map with Folium

- Used Folium to explore the launch sites
- Each launch was marked on each site with green for success and red for unsuccessful
- Each site was also examined for its proximity to important landmarks and infrastructure
- https://github.com/Alex-Van-Buren/Applied_Data_Science_Capstone/blob/main/06-lab_jupyter_launch_site_location.ipynb

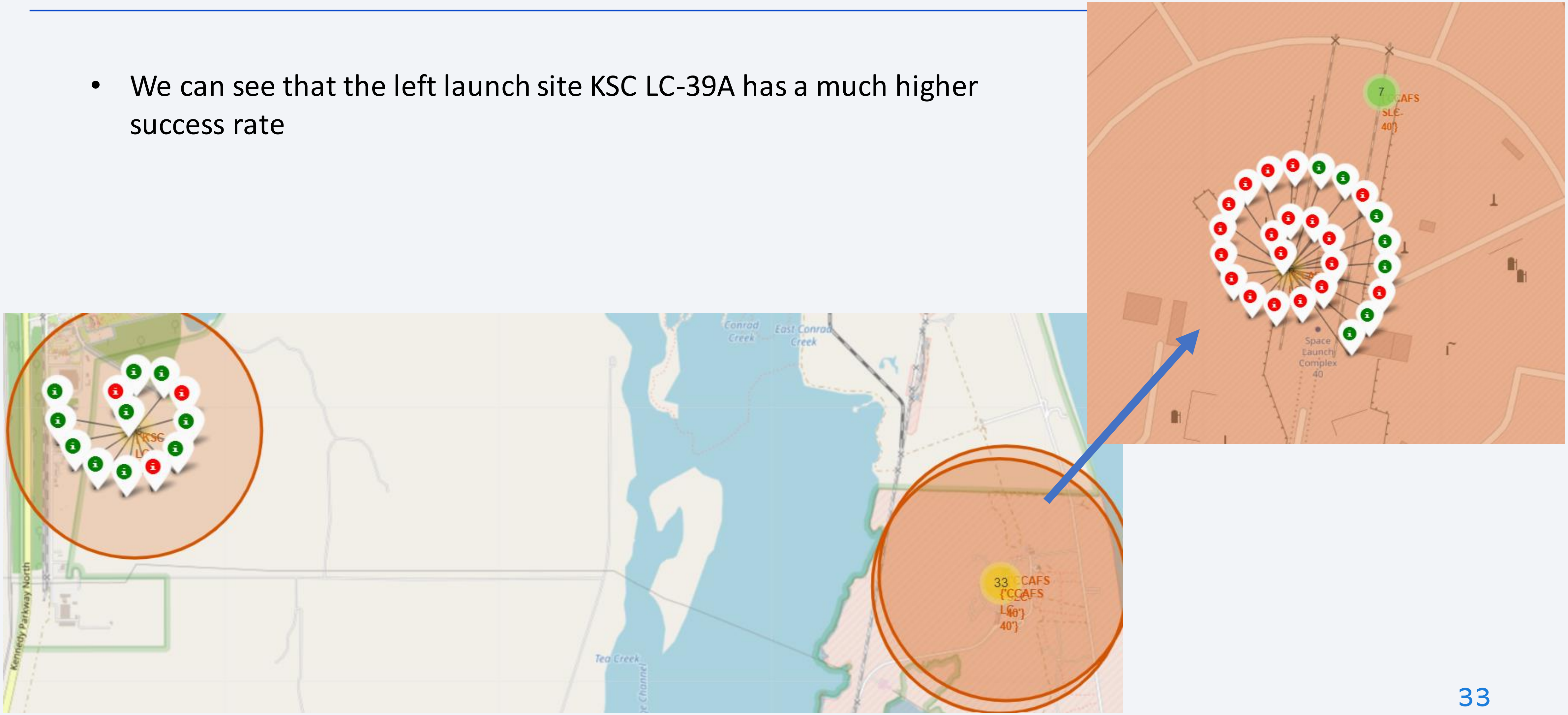
Site Locations Map



- 3 sites are situated in Florida while 1 is in California
- All sites are in Southern USA. (Warmer weather may be better)
- All sites are along the coast.

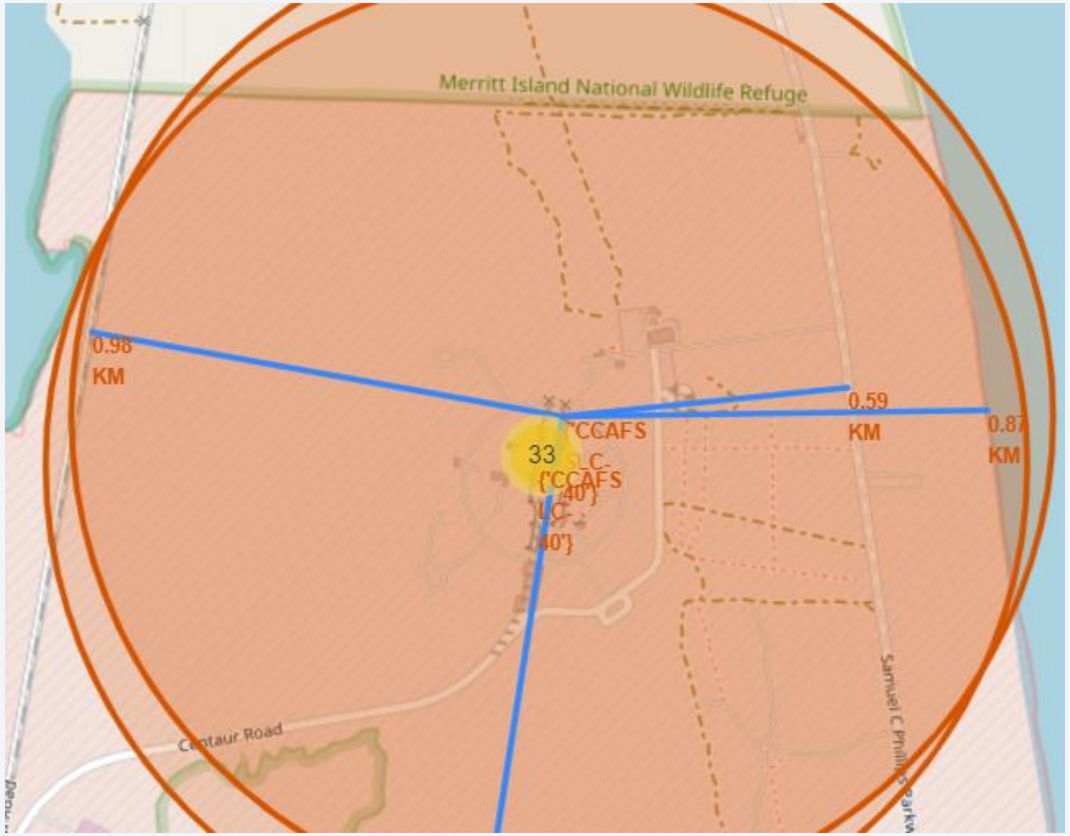
Site Launches

- We can see that the left launch site KSC LC-39A has a much higher success rate



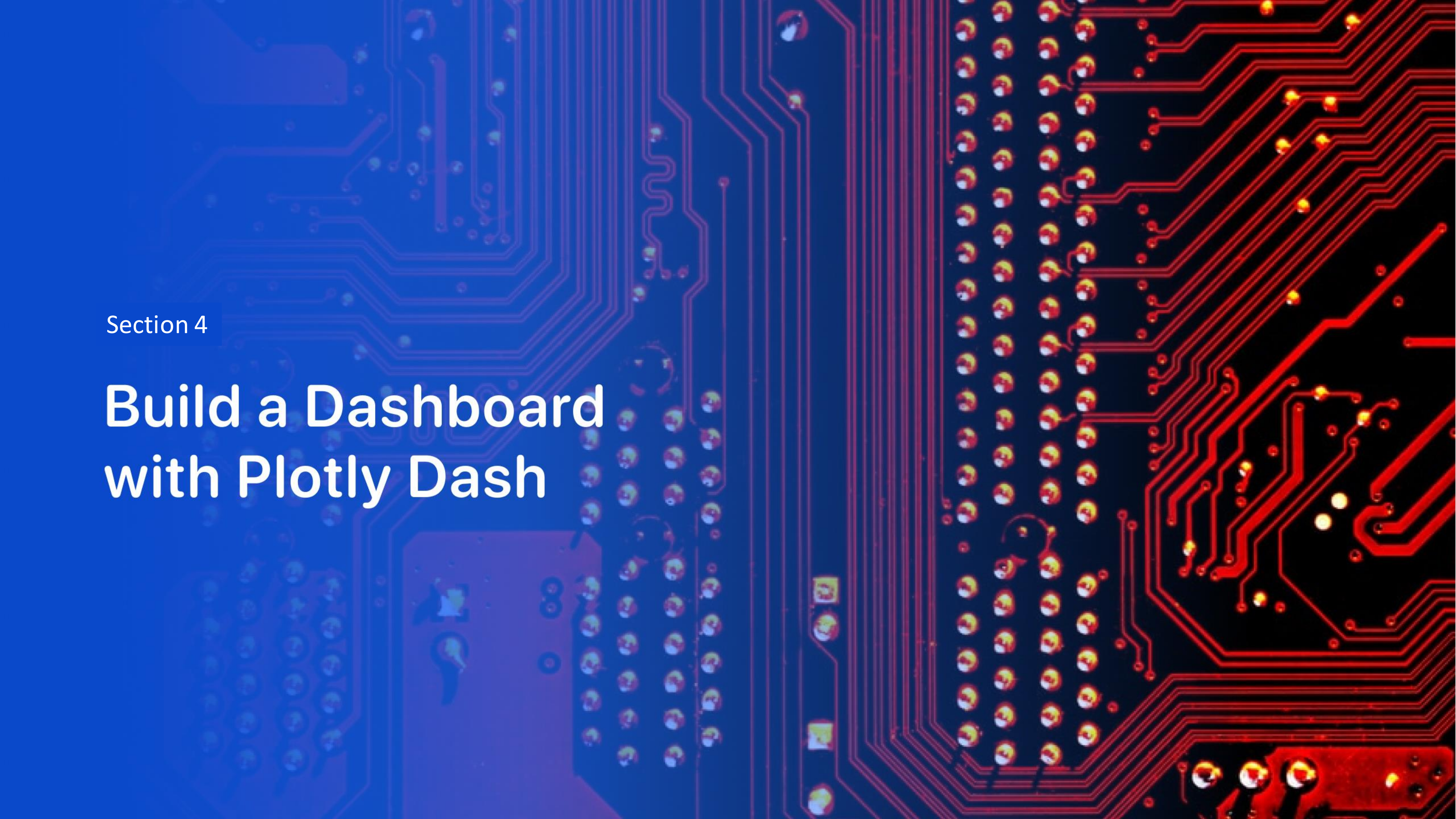
Land Marks and Infrastructure

- Distance to Ocean, Rail, and Highway are all relative close. (Under 1km)
- Distance to nearest City was much farther. (~18km)
- Rail and Hwy provide good access to get supplies.
- Nearby Ocean and far away city are good for safety



Section 4

Build a Dashboard with Plotly Dash

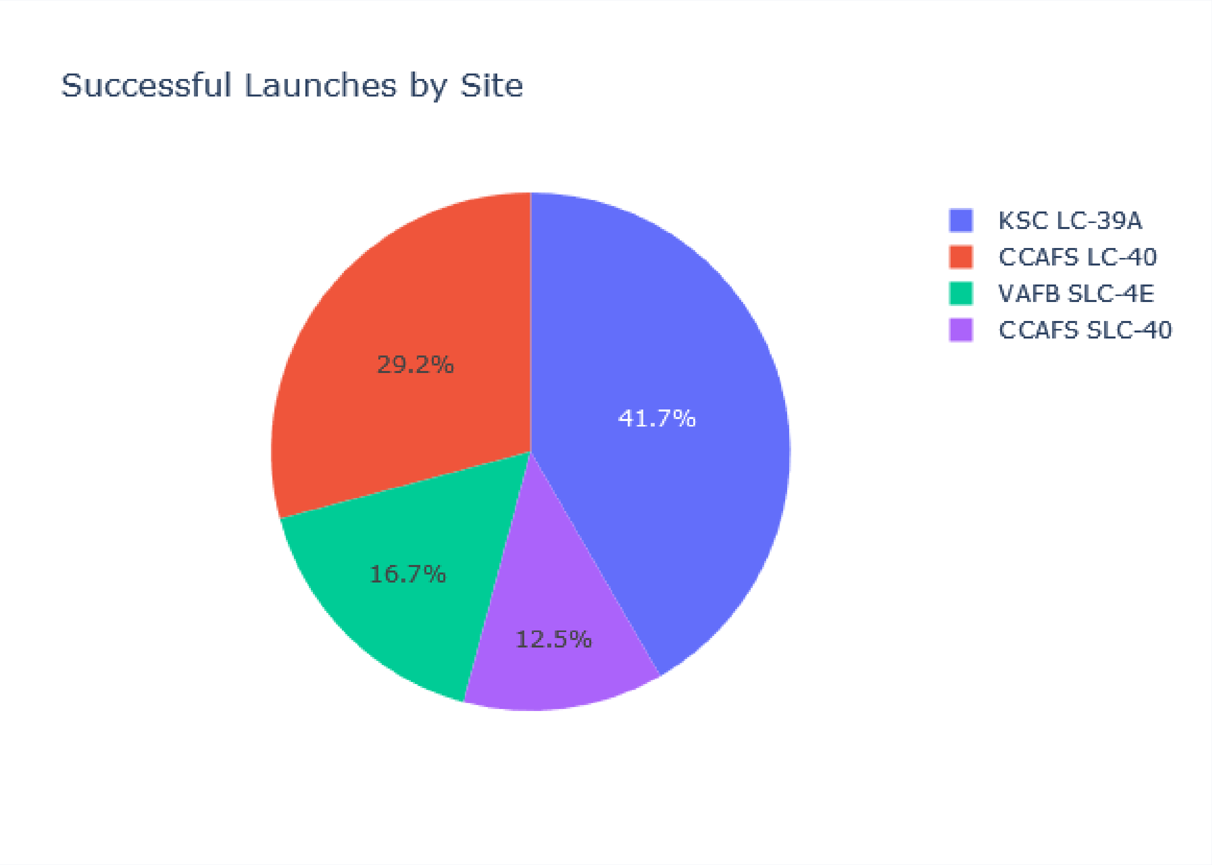


Build a Dashboard with Plotly Dash

- Used Plotly Dash to make an interactive dashboard
- It was made to explore each site's success.
- It also displays the payloads and booster versions used at each site
- https://github.com/Alex-Van-Buren/Applied_Data_Science_Capstone/blob/main/07_spacex_dash_app.py

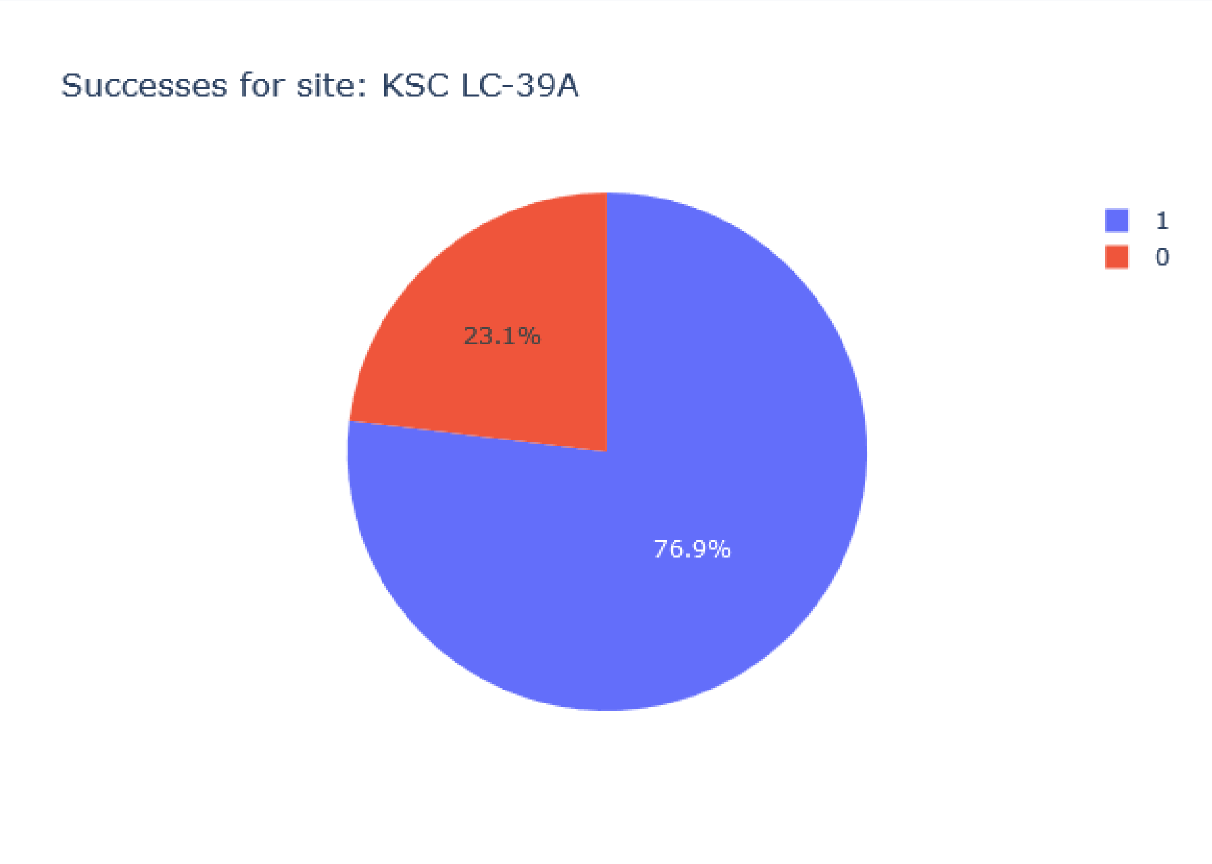
All Sites Success Count

- KSC LC-39A contributed most successful launches
- CCAFS SLC-40 had least successful launches

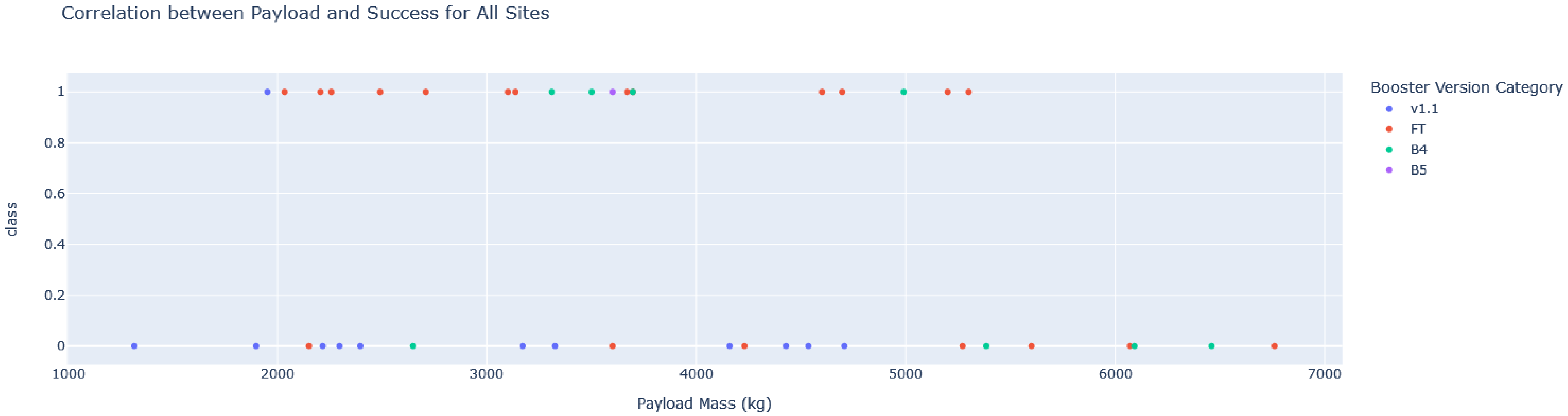


Highest Success Rate

- KSC LC 39-A also had the highest success rate at 76.9%
- Had 10 successful launches



Payload and Booster Version Success



- FT Booster Version between 2000 and 4000kg has high success rate
- V1.1 success rate is very low

Section 5

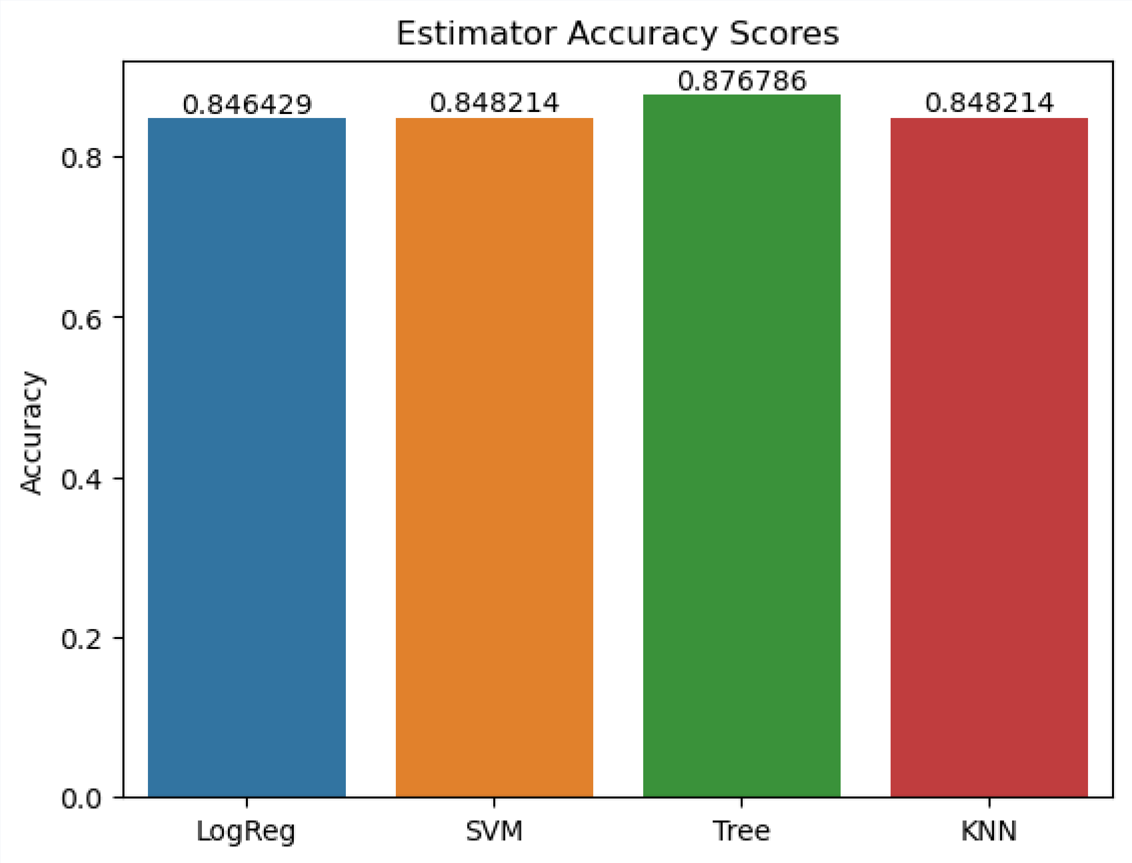
Predictive Analysis (Classification)

Predictive Analysis (Classification)

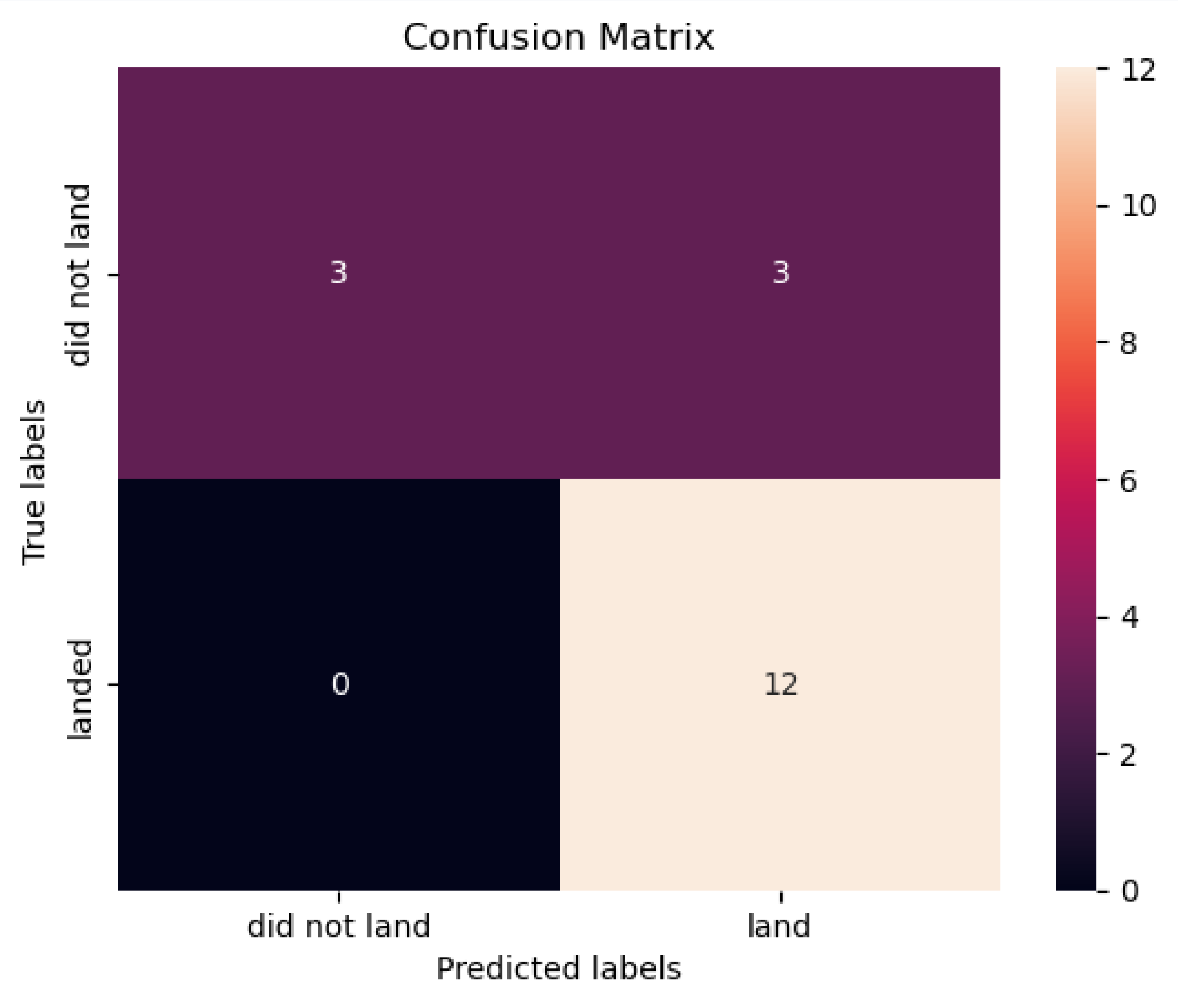
- 4 different classifiers were tested for predicting successful launches.
- Logistic Regression, Support Vector Machines (SVM), Decision Tree Classifiers, and K nearest neighbors (KNN)
- First all of the data was scaled and split into training and testing sets
- Grid Search was used to test multiple parameter groups for each classifier
- https://github.com/Alex-Van-Buren/Applied_Data_Science_Capstone/blob/main/08_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Classification Accuracy

- All the models performed about the same.
- The Decision Tree Classifier had slightly higher accuracy for the training data



Confusion Matrix



- Each confusion matrix for the test data on each estimator looked exactly like this one.
- 15 correctly labeled points and 3 false positives where the rocket was predicted to land but did not.

Classification Results

- The classifiers used were nearly indistinguishable.
- More data would likely be needed to identify a clear best classifier
- The training accuracy for each method was between .84 and .88
- The testing data accuracy was slightly lower at 0.833

Conclusions

- It's likely that a new company would struggle to land rockets successfully right away, increasing early costs
- A new company in space technology would want to be located near the coast
- Predicting landing outcomes for rockets using machine learning seems to work fairly well

Appendix

- All files can be found at:
- https://github.com/Alex-Van-Buren/Applied_Data_Science_Capstone/tree/main

Thank you!