# Outline

- Executive Summary

- Introduction

- Methodology

- Results

- Conclusion

- Appendix

# Executive Summary

- For this project, data was collected from the SpaceX API and the SpaceX Wikipedia page. The "class" label column was created, classifying successful landings. The data was explored using SQL, visualization, Folium maps, and dashboards. Relevant columns were collected for use as features. All categorical variables were converted to binary using one-hot encoding. The data was standardized, and GridSearchCV was used to find the best parameters for the machine learning models. The accuracy of all models was visualized.

- Four machine learning models were generated: logistic regression, support vector machine, decision tree classifier, and K-nearest neighbors. All yielded similar results with an accuracy of approximately 83.33%. All models overestimated successful landings. More data is needed to better determine the accuracy of the models.

# Introduction

## Project background

- Commercial Space Age is Here

- Space X has best pricing ($62 million vs. $165 million USD)

- Largely due to ability to recover part of rocket (Stage 1)

- Space Y wants to compete with Space X

## Problems

- Space Y tasks us to train a machine learning model to predict successful Stage 1 recovery

Section 1

# Methodology

# Methodology

Executive Summary

- Data collection methodology:

  - Combined data from SpaceX public API and SpaceX Wikipedia page

- Perform data wrangling

  - Classifying true landings as successful and unsuccessful otherwise

- Perform exploratory data analysis (EDA) using visualization and SQL

- Perform interactive visual analytics using Folium and Plotly Dash

- Perform predictive analysis using classification models

  - Tuned models using GridSearchCV

# Data Collection

Data collection process involved a combination of API requests from Space X public API and web scraping data from a table in Space X's Wikipedia entry.

The next slide will show the flowchart of data collection from API and the one after will show the flowchart of data collection from webscraping.

Space X API Data Columns:

FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins,

Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude
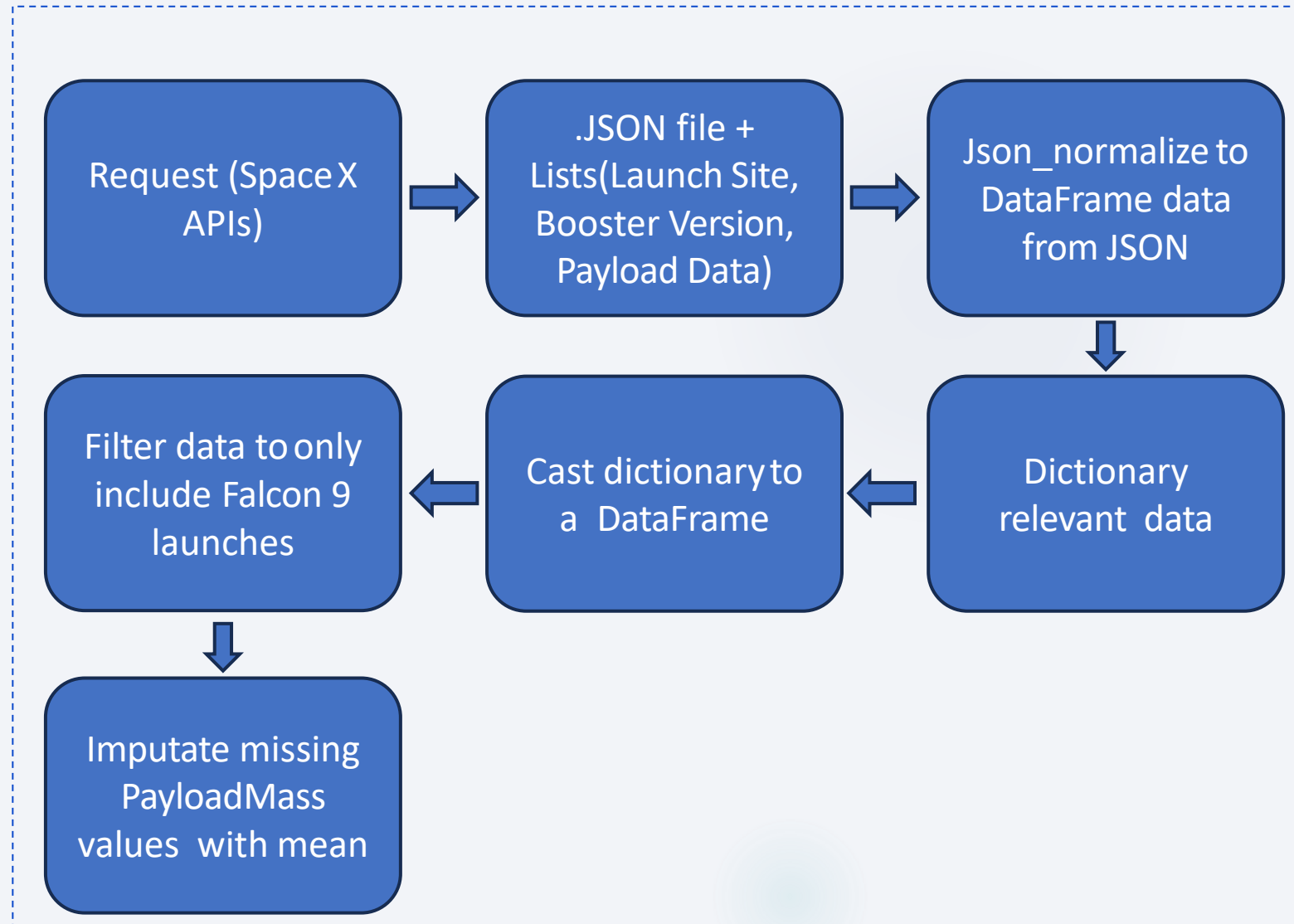
Wikipedia Webscrape Data Columns:

Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time.
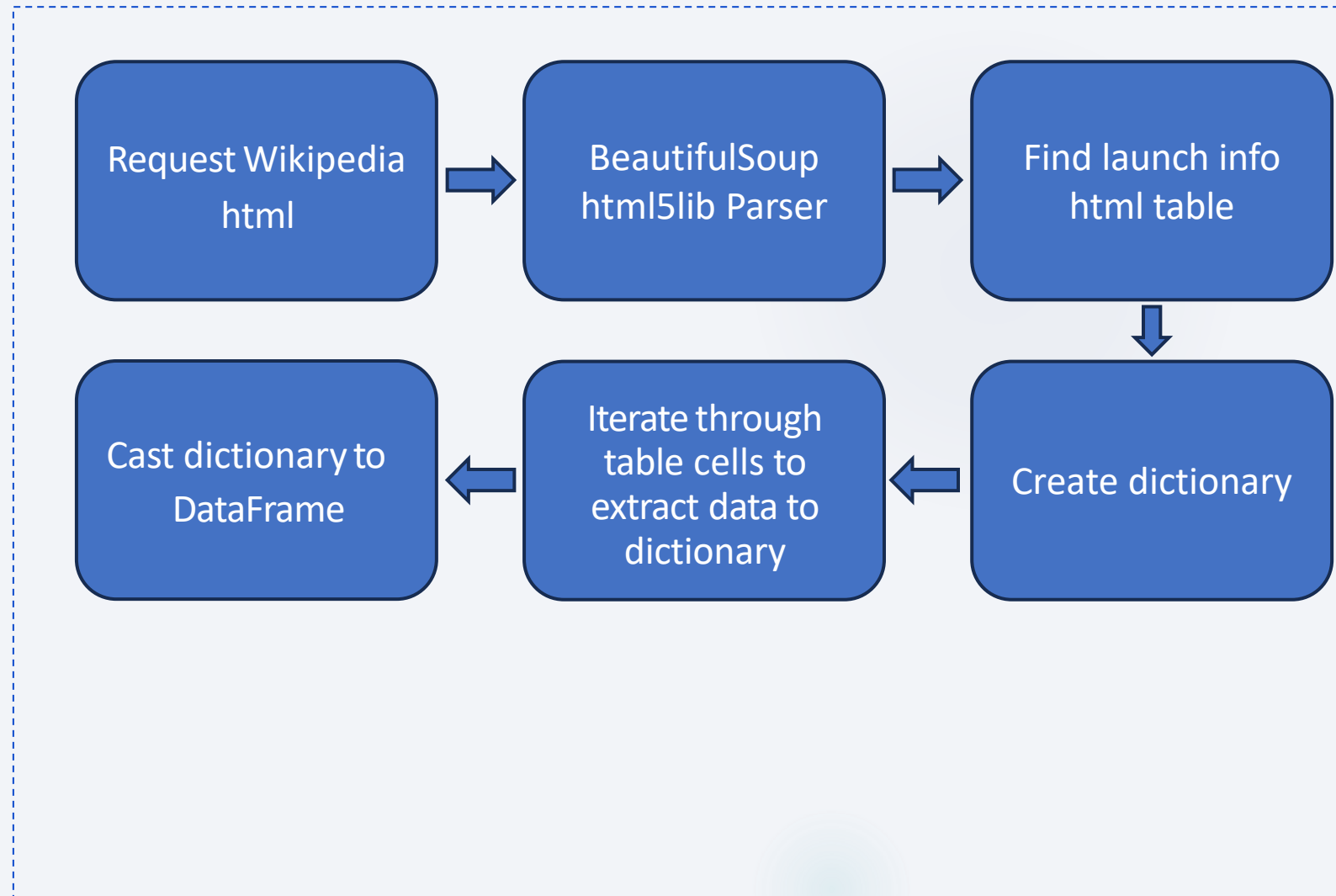
# Data Collection – SpaceX API

▶ GitHub URL

https://github.com/nyrko/Prueba_3_
Data_Science/blob/main/jupyter-
labs-spacex-data-collection-api.ipynb

```
Request (SpaceX
APIs)
```
→
```
.JSON file +
Lists(Launch Site,
Booster Version,
Payload Data)
```
→
```
Json_normalize to
DataFrame data
from JSON
```
↓
```
Filter data to only
include Falcon 9
launches
```
←
```
Cast dictionary to
a DataFrame
```
←
```
Dictionary
relevant data
```
↓
```
Imputate missing
PayloadMass
values with mean
```

# Data Collection - Scraping

▶ GitHub URL

https://github.com/nyrko/Prueba_3_Data_Science/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

```
Request Wikipedia html → BeautifulSoup html5lib Parser → Find launch info html table
                                                                    ↓
Cast dictionary to DataFrame ← Iterate through table cells to extract data to dictionary ← Create dictionary
```

# Data Wrangling

▶ **Data processing**

Create a training label with landing outcomes where successful = 1 & failure = 0.

Outcome column has two components: 'Mission Outcome' 'Landing Location'

New training label column 'class' with a value of 1 if 'Mission Outcome' is True and 0

otherwise.  Value Mapping:

True ASDS, True RTLS, & True Ocean – set to -> 1

None None, False ASDS, None ASDS, False Ocean, False RTLS – set to -> 0

▶ **GitHub URL**

https://github.com/nyrko/Prueba_3_Data_Science/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

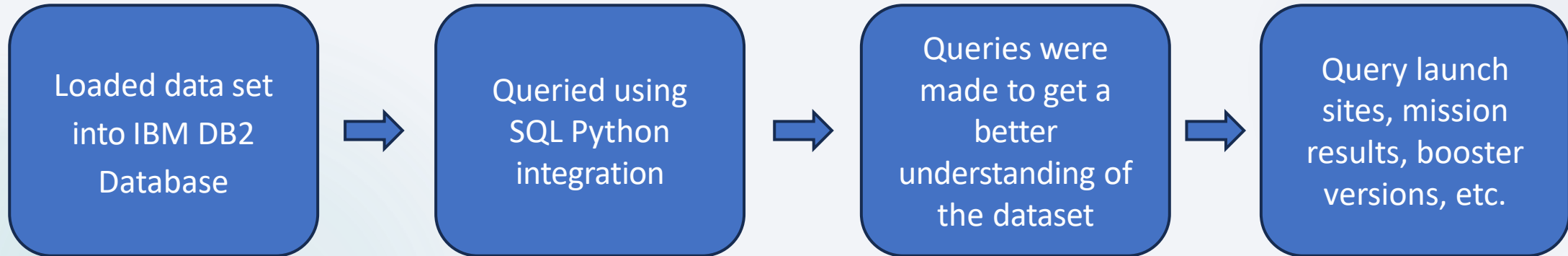# EDA with Data Visualization

▶ Graphics used

Data exploration for the corresponding analysis was performed using the variables Flight Number, Payload Mass, Launch Site, Orbit, Class, and Year.

Graphs used: Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit vs. Success Rate, Flight Number vs. Orbit, Payload vs. Orbit, and Annual Success Trend.Scatter plots, line graphs, and bar graphs were used to compare the relationships between variables and determine if a relationship existed for use in training the machine learning model.

▶ GitHub URL

https://github.com/nyrko/Prueba_3_Data_Science/blob/main/EDA_visualizacion.ipynb

# EDA with SQL

▶ SQL query summary

| Loaded data set into IBM DB2 Database | → | Queried using SQL Python integration | → | Queries were made to get a better understanding of the dataset | → | Query launch sites, mission results, booster versions, etc. |

▶ GitHub URL

https://github.com/nyrko/Prueba_3_Data_Science/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

# Build an Interactive Map with Folium

▶ Folium maps mark Launch Sites, successful and unsuccessful landings, and a proximity example to key locations: Railway, Highway, Coast, and City.

▶ This allows us to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.

▶ GitHub URL

https://github.com/nyrko/Prueba_3_Data_Science/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

# Build a Dashboard with Plotly Dash

▶ Dashboard includes a pie chart and a scatter plot.

Pie chart can be selected to show distribution of successful landings across all launch sites and  can be selected to show individual launch site success rates.

Scatter plot takes two inputs: All sites or individual site and payload mass on a slider between 0  and 10000 kg.

The pie chart is used to visualize launch site success rate.

The scatter plot can help us see how success varies across launch sites, payload mass, and booster version category.

▶ GitHub URL

https://github.com/nyrko/Prueba_3_Data_Science/blob/main/spacex_dash_app.py

# Predictive Analysis (Classification)

▶ Model development

```
Split label column 'Class' from dataset → Fit and transform features using Standar Scaler → Train test split data → GridSearchCV (cv=10) to find optimal parameters
                                                                                                                                         ↓
Barplot to compare scores of models ← Confusion Matrix for all models ← Score models on Split test set ← Use GridSearchCV on LogReg, SVM, Decision Tree, and KNN models
```
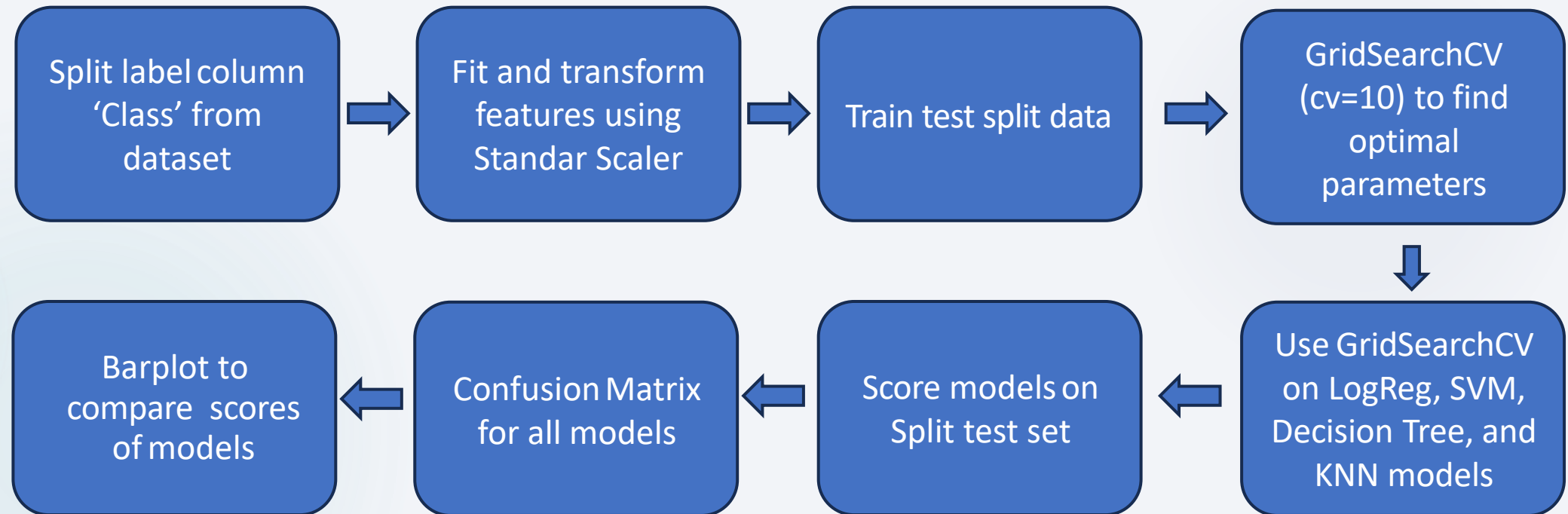
▶ GitHub URL

https://github.com/nyrko/Prueba_3_Data_Science/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb
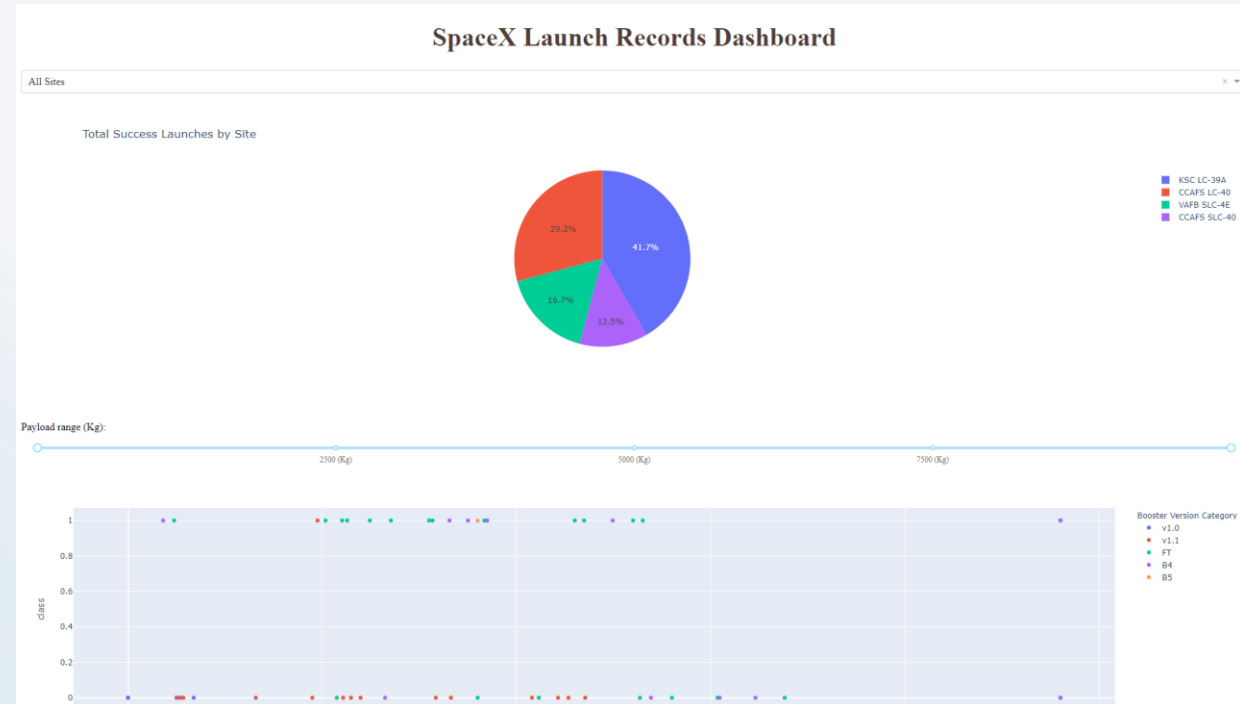
# Results

This is a preview of the Plotly dashboard. The following sides will show the results of EDA with  visualization, EDA with SQL, Interactive Map with Folium, and finally the results of our model with  about 83% accuracy.

Section 2

# Insights drawn from EDA

# Flight Number vs. Launch Site
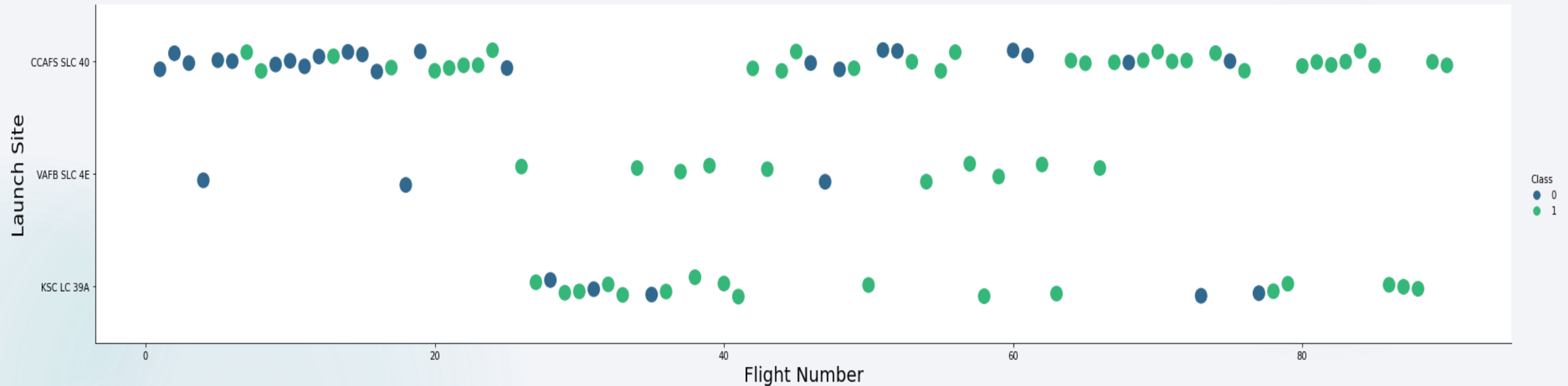
Green indicates successful launch; Purple indicates unsuccessful launch.

Graphic suggests an increase in success rate over time (indicated in Flight Number).
Likely a big breakthrough around flight 20 which significantly increased success rate.
CCAFS appears to be the main launch site as it has the most volume.
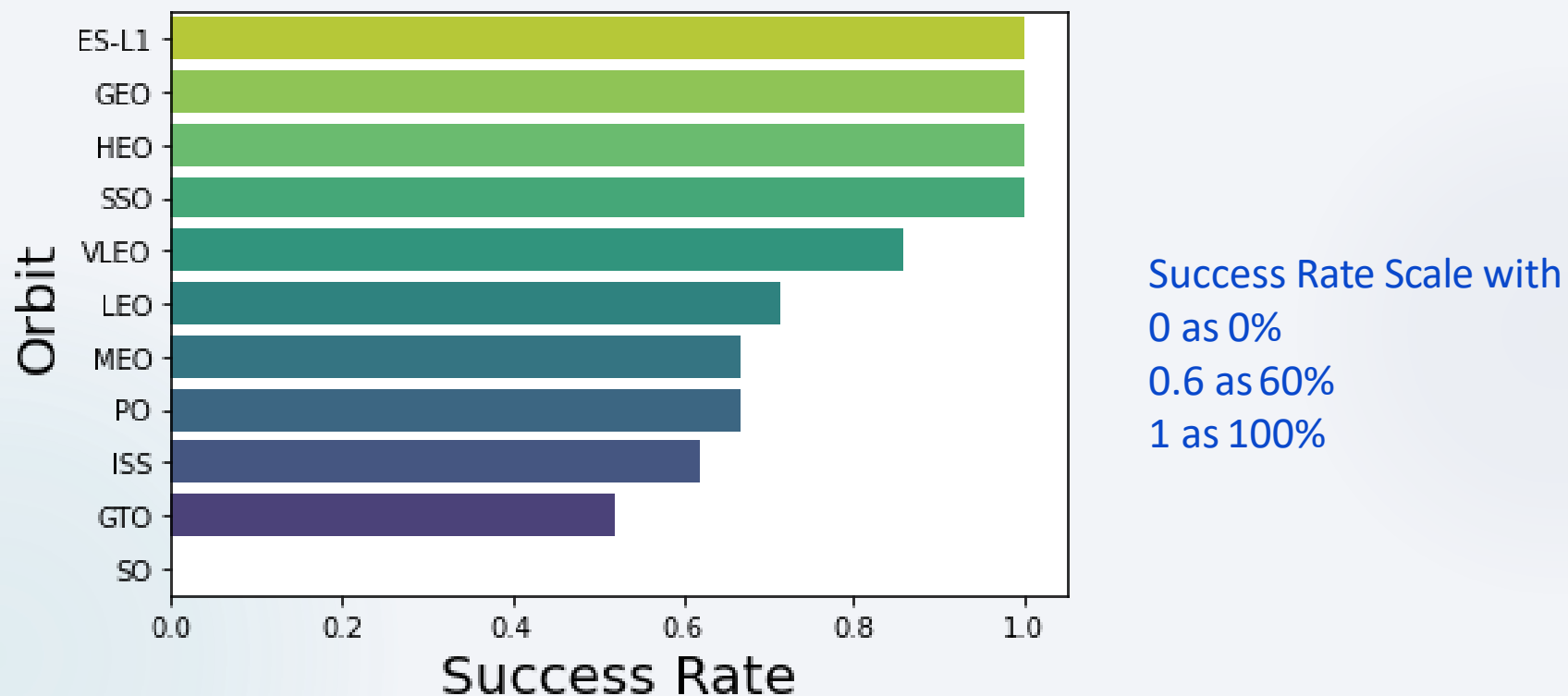
# Payload vs. Launch Site
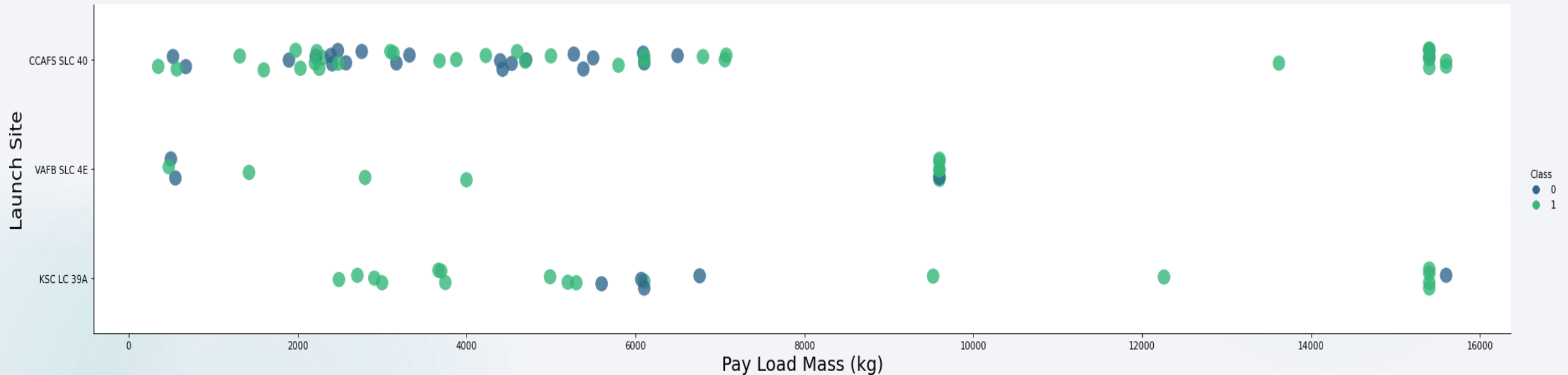
Green indicates successful launch; Purple indicates unsuccessful launch.

Payload mass appears to fall mostly between 0-6000 kg.

Different launch sites also seem to use different payload mass.

# Success Rate vs. Orbit Type

Success Rate Scale with
0 as 0%
0.6 as 60%
1 as 100%

ES-L1 (1), GEO (1), HEO (1) have 100% success rate (sample sizes in parenthesis)
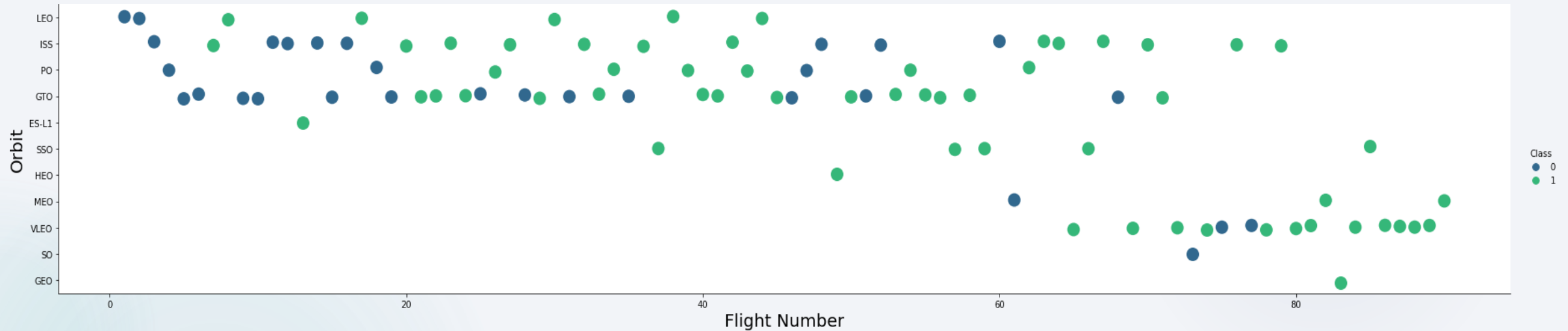
SSO (5) has 100% success rate

VLEO (14) has decent success rate and attempts

SO (1) has 0% success rate

GTO (27) has the around 50% success rate but largest sample

# Flight Number vs. Orbit Type



Green indicates successful launch; Purple indicates unsuccessful launch.
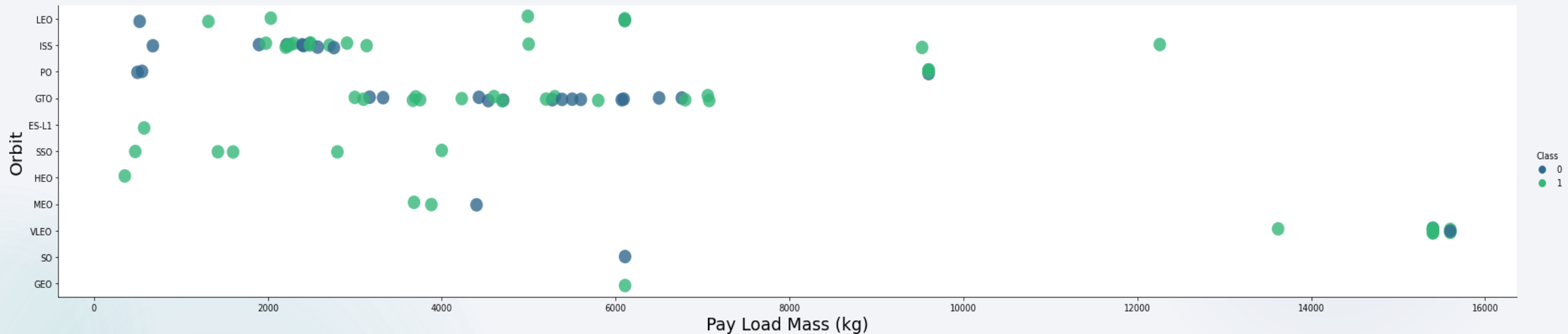
Launch Orbit preferences changed over Flight Number.

Launch Outcome seems to correlate with this preference.

SpaceX started with LEO orbits which saw moderate success LEO and returned to VLEO in recent launches

SpaceX appears to perform better in lower orbits or Sun-synchronous orbits

# Payload vs. Orbit Type

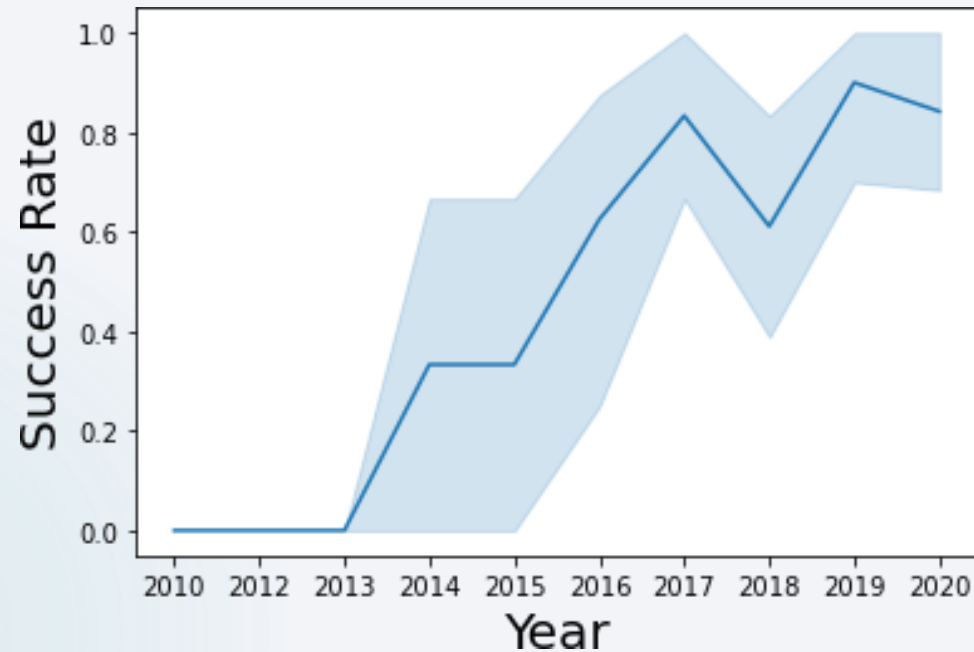Green indicates successful launch; Purple indicates unsuccessful launch.

Payload mass seems to correlate with orbit

LEO and SSO seem to have relatively low payload mass

The other most successful orbit VLEO only has payload mass values in the higher end of the range

# Launch Success Yearly Trend

95% confidence interval
(light blue shading)

Success generally increases over time since 2013 with a slight dip in 2018

Success in recent years at around 80%

# All Launch Site Names

```
In [4]: %%sql
        SELECT UNIQUE LAUNCH_SITE
        FROM SPACEXDATASET;

         * ibm_db_sa://ftb12020:***@0c77d6f2
        Done.
```

Out[4]:

| launch_site |
|---|
| CCAFS LC-40 |
| CCAFS SLC-40 |
| CCAFSSLC-40 |
| KSC LC-39A |
| VAFB SLC-4E |

Query unique launch site names from database.

CCAFS SLC-40 and CCAFSSLC-40 likely all represent the same
launch site with data entry errors.

CCAFS LC-40 was the previous name.

Likely only 3 unique launch_site

values:  CCAFS SLC-40, KSC LC-39A,

VAFB SLC-4E

# Launch Site Names Begin with 'CCA'

```
In [5]: %%sql
        SELECT *
        FROM SPACEXDATASET
        WHERE LAUNCH_SITE LIKE 'CCA%'
        LIMIT 5;
```

 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.

Out[5]:

| 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 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 2012-10-08 | 00:35:00 | 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 |

First five entries  in database with  Launch Site name  beginning with  CCA.

# Total Payload Mass

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)';
```

 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.

| sum_payload_mass_kg |
|---|
| 45596 |

This query sums the total payload  mass in kg where NASA was the  customer.

CRS stands for Commercial Resupply Services which indicates  that these payloads were sent to  the International Space Station  (ISS).

# Average Payload Mass by F9 v1.1

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE booster_version = 'F9 v1.1'
```

 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.

| avg_payload_mass_kg |
| --- |
| 2928 |

This query calculates the  average payload mass or  launches which used  booster version F9 v1.1

Average payload mass of F9 1.1 is on the low end of  our payload mass range

# First Successful Ground Landing Date

```
%%sql
SELECT MIN(DATE) AS FIRST_SUCCESS
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (ground pad)';
```

 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81

Done.

| first_success |
| --- |
| 2015-12-22 |

This query returns the first successful ground pad landing  date.

First ground pad landing wasn't
until the end of 2015.

Successful landings in general appear starting 2014.

```
%%sql
SELECT booster_version
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN 4001 AND 5999;
```

 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.database
Done.

| booster_version |
|---|
| F9 FT B1022 |
| F9 FT B1026 |
| F9 FT B1021.2 |
| F9 FT B1031.2 |

This query returns the four  booster versions that had  successful drone ship landings  and a payload mass between  4000 and 6000 noninclusively.

# Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT mission_outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
GROUP BY mission_outcome;
```

 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-
Done.

| mission_outcome | no_outcome |
| --- | --- |
| Failure (in flight) | 1 |
| Success | 99 |
| Success (payload status unclear) | 1 |

This query returns a count of each mission outcome.

SpaceX appears to achieve its mission outcome nearly 99% of the  time.

This means that most of the landing
failures are intended.

Interestingly, one launch has an unclear payload status and unfortunately one failed in flight.

```
%%sql
SELECT booster_version, PAYLOAD_MASS__KG_
FROM SPACEXDATASET
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET);
```

 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1
Done.

| 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 |

This query returns the booster versions that carried the highest payload mass of 15600 kg.

These booster versions are very similar and all are of the F9 B5 B10xx.x variety.

This likely indicates payload mass correlates with the booster version that is used.

# 2015 Launch Records

```
%%sql
SELECT MONTHNAME(DATE) AS MONTH, landing__outcome, booster_version, PAYLOAD_MASS__KG_, launch_site
FROM SPACEXDATASET
WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

* ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.app
Done.

| MONTH | landing__outcome | booster_version | payload_mass__kg_ | launch_site |
|---|---|---|---|---|
| January | Failure (drone ship) | F9 v1.1 B1012 | 2395 | CCAFS LC-40 |
| April | Failure (drone ship) | F9 v1.1 B1015 | 1898 | CCAFS LC-40 |

This query returns the Month, Landing  Outcome, Booster Version, Payload  Mass (kg), and Launch site of 2015  launches where stage 1 failed to land  on a drone ship.

There were two such occurrences.

```
%%sql
SELECT landing__outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
WHERE landing__outcome LIKE 'Succes%' AND DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY no_outcome DESC;
```

 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg
Done.

| landing__outcome | no_outcome |
|---|---|
| Success (drone ship) | 5 |
| Success (ground pad) | 3 |

This query returns a list of successful landings  and between 2010-06-04 and 2017-03-20 inclusively.

There are two types of successful landing  outcomes: drone ship and ground pad landings.

There were 8 successful landings in total  during this time period

Section 3

# Launch Sites Proximities Analysis

# Launch site location
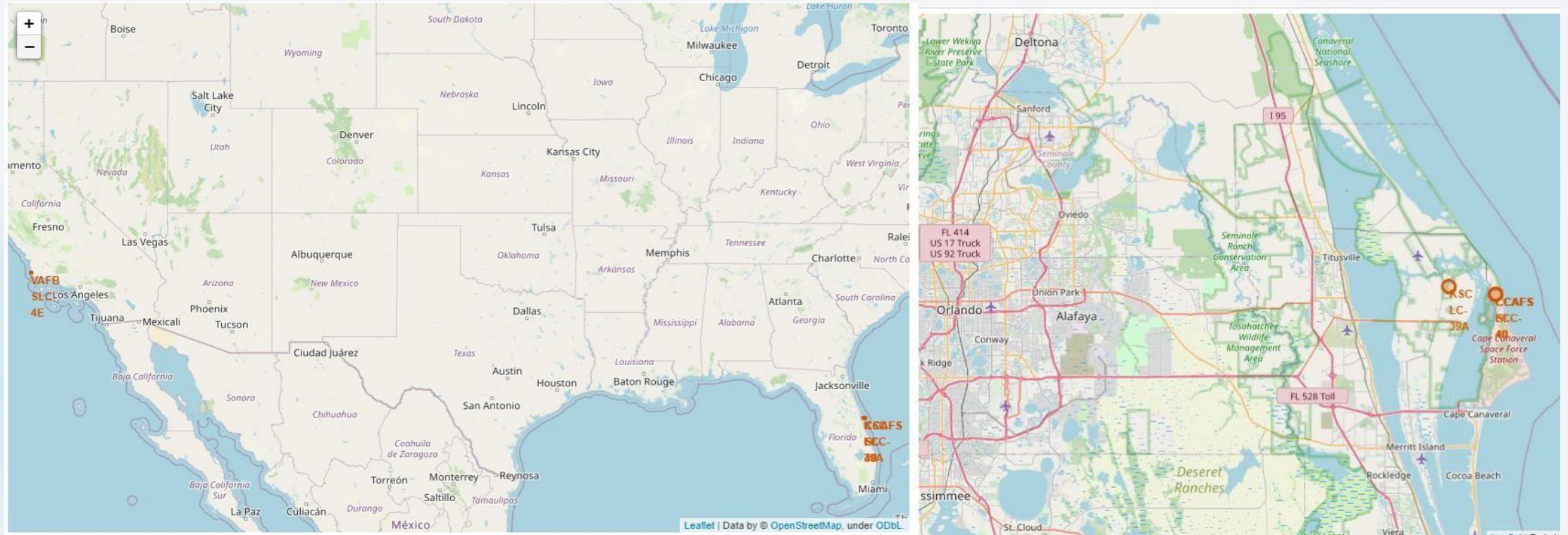
The left map shows all launch sites relative US map. The right map shows the two Florida launch  sites since they are very close to each other. All launch sites are near the ocean.
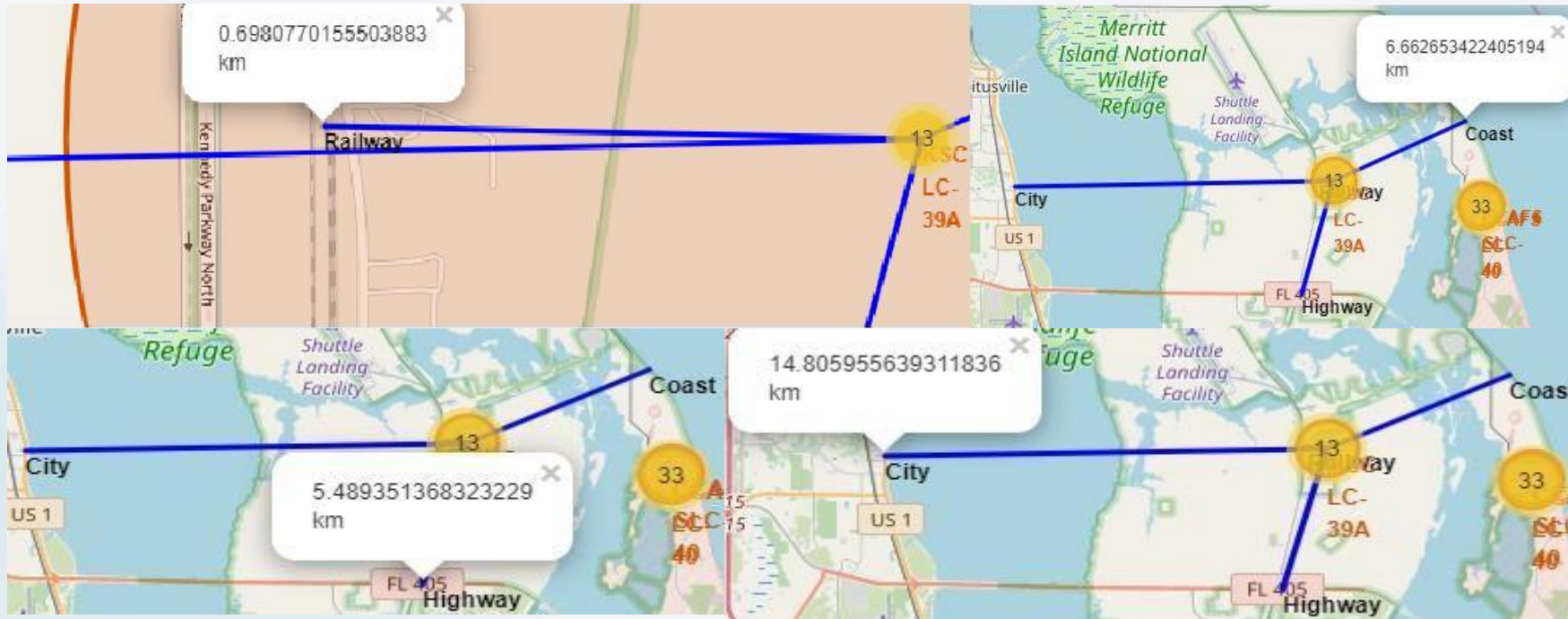
# Color-Coded Launch Markers
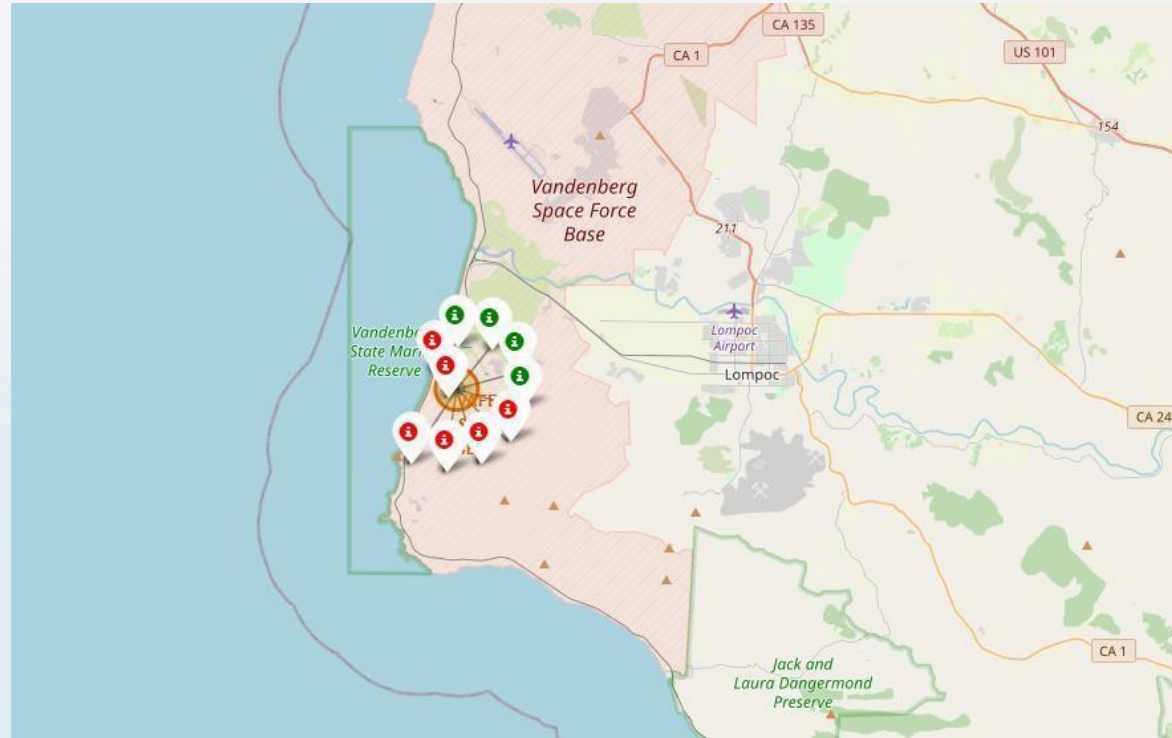
Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed landing (red icon). In this example VAFB SLC-4E shows 4 successful landings and 6 failed landings.

Using KSC LC-39A as an example, launch sites are very close to railways for large part and supply transportation. Launch sites are close to highways for human and supply transport. Launch sites are also close to coasts and relatively far from cities so that launch failures can land in the sea to avoid rockets falling on densely populated areas.
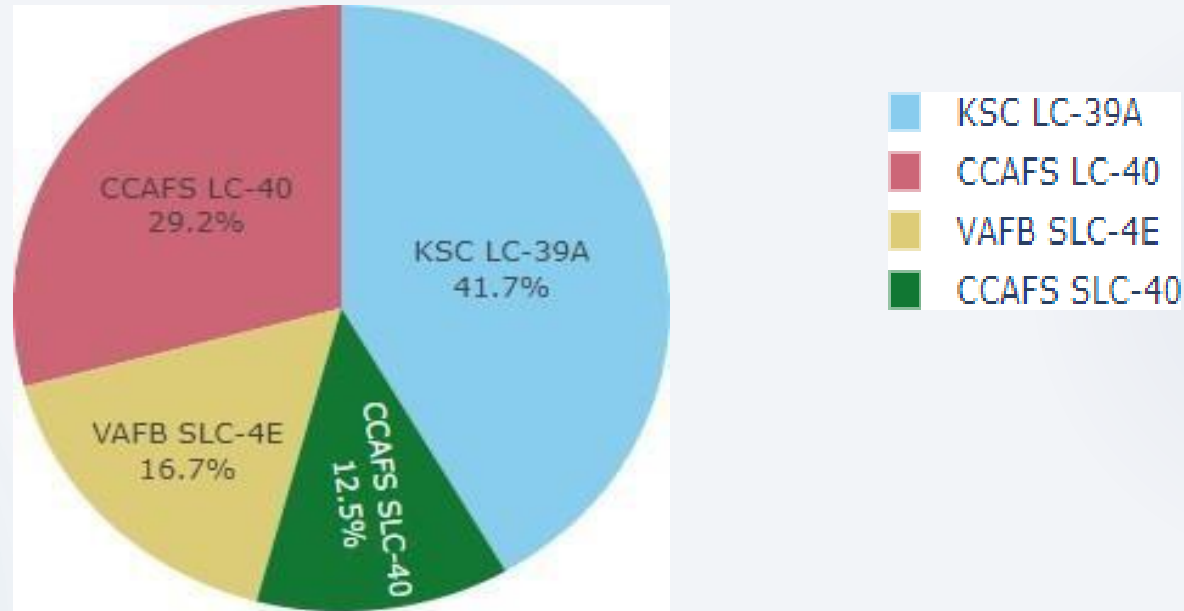
Section 4

# Build a Dashboard with Plotly Dash

# Successful Launches Across Lauch Sites



This is the distribution of successful landings across all launch sites. CCAFS LC-40 is the old name of CCAFS SLC-40 so CCAFS and KSC have the same amount of successful landings, but a majority of the successful landings where performed before the name change. VAFB has the smallest share of successful landings. This may be due to smaller sample and increase in difficulty of launching in the west coast.

# Highest Success Rate Launch Site

KSC LC-39A Success Rate (blue=success)
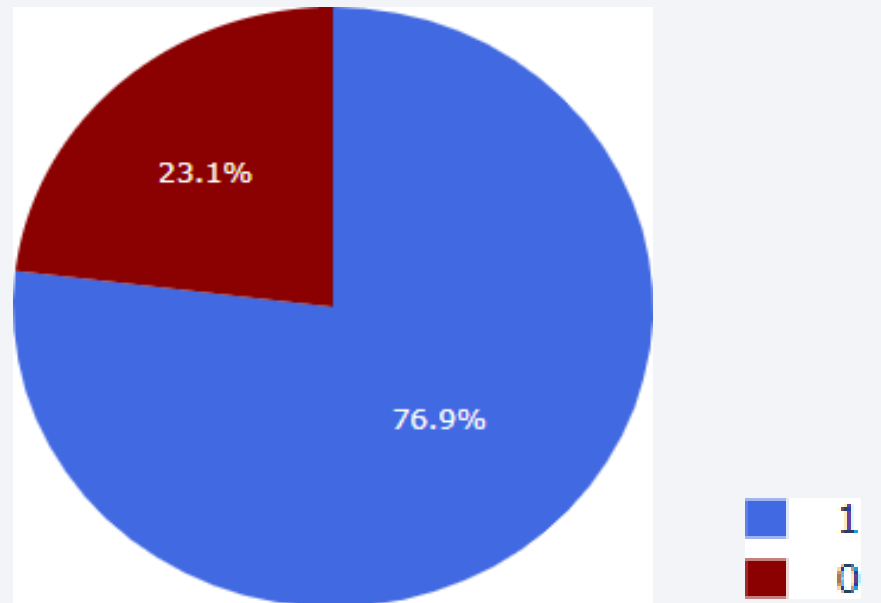
KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.

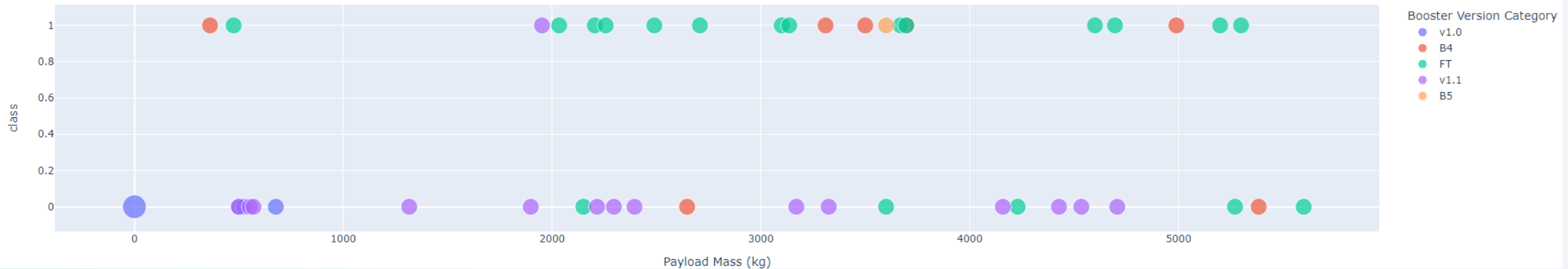# Playload vs. Launch Outcome

Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600. Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size. In this particular range of 0-6000, interestingly there are two failed landings with payloads of zero kg.
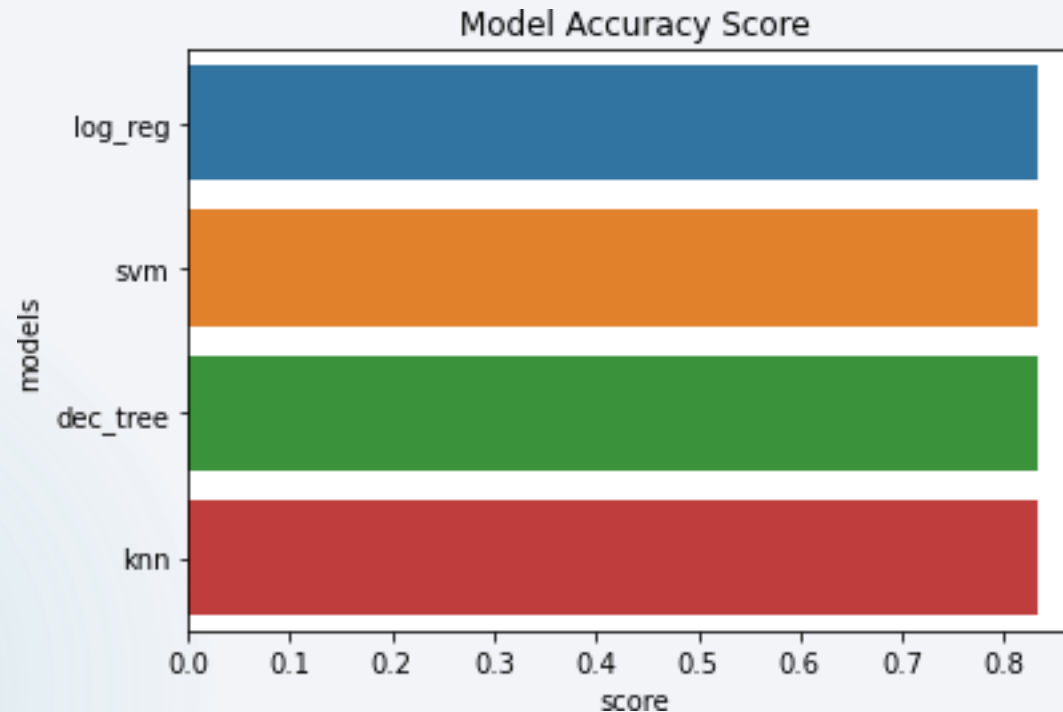
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

All models had virtually the same accuracy on the test set at 83.33% accuracy.

It should be noted that test size is small at only sample size of 18.

This can cause large variance in accuracy results, such as those in Decision Tree Classifier model in repeated runs.

We likely need more data to determine the best model.

# Confusion Matrix

Correct predictions are on a diagonal from top left to bottom right.

Since all models performed the same for the test set, the confusion matrix is the same across all models. The models predicted 12 successful landings when the true label was successful landing.

The models predicted 3 unsuccessful landings when the true label was unsuccessful landing.
The models predicted 3 successful landings when the true label was unsuccessful landings (false positives).
Our models over predict successful landings.

# Conclusions

▶ Our task: to develop a machine learning model for Space Y who wants to bid against SpaceX

▶ Used data from a public SpaceX API and web scraping SpaceX Wikipedia page

▶ Created data labels and stored data into a DB2 SQL database

▶ Created a dashboard for visualization

▶ We created a machine learning model with an accuracy of 83%

▶ If possible more data should be collected to better determine the best machine learning model  and improve accuracy

# Appendix

Thank you!