# Outline

- Executive Summary

- Introduction

- Methodology

- Results

- Conclusion

- Appendix

# Executive Summary

- Summary of methodologies

  - Data Collection

  - Data Wrangling

  - Exploratory Data Analysis using SQL and data visualization

  - Generating Interactive map on folium

  - Generating dashboard on ploty

  - Generating machine learning models

- Summary of all results

  - The built machine learning models had predicted results with an accuracy of 83.33%.

# Introduction

- Project background and context

  - Many company are presently providing commercial space transport.

  - SpaceX provides the best price currently

  - SpaceX charges 100million dollars less.

  - Lower price is sue to recovery of stage 1 rockets.

- Problems you want to find answers

  - SpaceY wants to compete with SpaceX for commercial space transport.

  - We want to build a machine learning model which can predict the recovery of stage 1 rockets.

Section 1

# Methodology

# Methodology

- Data collection methodology:

    - The required data for this project was collected from 2 sources.

        - Using API calls from SpaceX public API

        - Using web scrapping on wikipedia page.

- Perform data wrangling

    - Cleaning the data for

        - Processing missing data

        - One hot encoding for better processing of the data

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

- Perform interactive visual analytics using Folium and Plotly Dash

- Perform predictive analysis using classification models

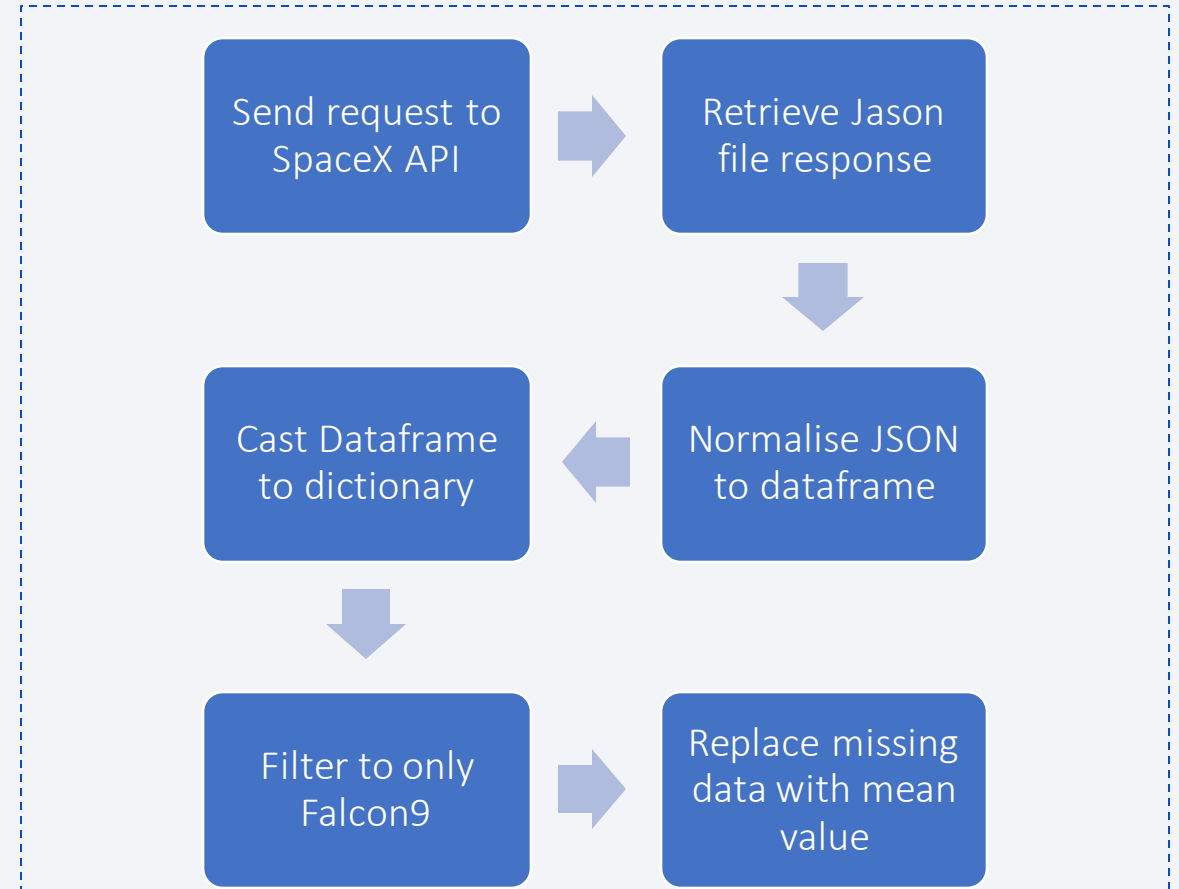    - Using GridSearchCV to find the best parameters to fit/train the model.

# Data Collection

Data for SpaceX flights collected from 2 sources

1) Using API calls and retrieving the data from SpaceX public API.

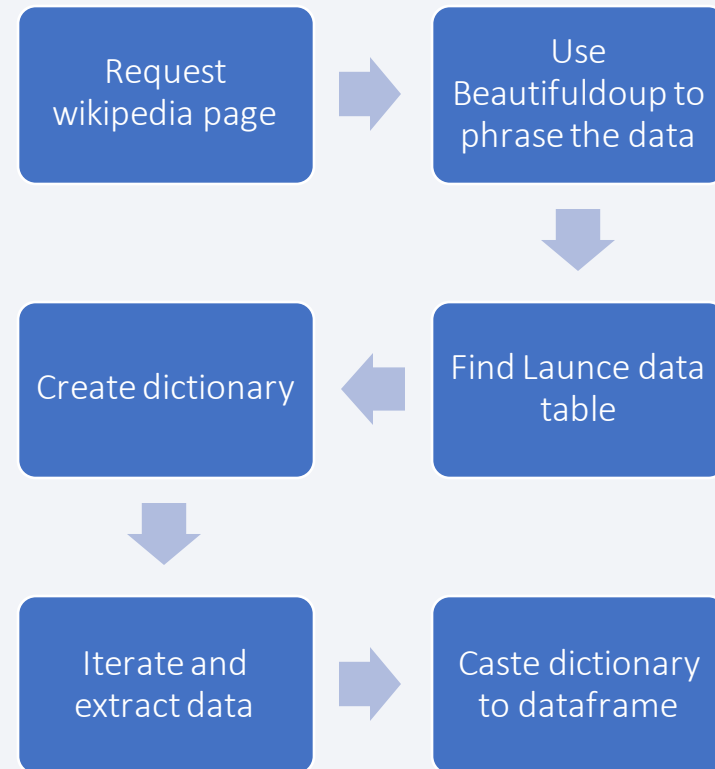2) Using webscraping to retrieve data from wikipedia page.

# Data Collection – SpaceX API

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

| | |
|---|---|
| Send request to SpaceX API | Retrieve Jason file response |
| Cast Dataframe to dictionary | Normalise JSON to dataframe |
| Filter to only Falcon9 | Replace missing data with mean value |

# Data Collection - Scraping

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

```
Request wikipedia page  →  Use Beautifuldoup to phrase the data
                                        ↓
Create dictionary  ←  Find Launce data table
        ↓
Iterate and extract data  →  Caste dictionary to dataframe
```

# Data Wrangling

- Replace successful launch=1 and failure =0

- One hot encoding


- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

# EDA with Data Visualization

- EDA was performed on below variables

Flight number, Payload mass, Launch site, Orbit, Class and year.

- Plots plotted

Flight Numbers vs payload mass, Flight number vs launch site, payload mass vs launch site, Orbit Vs success rate, Flight number vs orbit, success yearly trend.

Scatter, line and bar chart/plot were used to compare relationship between variable to be used to train the machine learning lodels.

- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

# EDA with SQL

Load dataset in IBM DB2

Query using SQL

Quired info regarding launch site, mission outcome, payload sizes, booster version and landing outcome.

- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

# Build an Interactive Map with Folium

- Folium maps mark launch sites successful and unsuccessful landings and proximity to key locations  (railway, highway, coast city.)
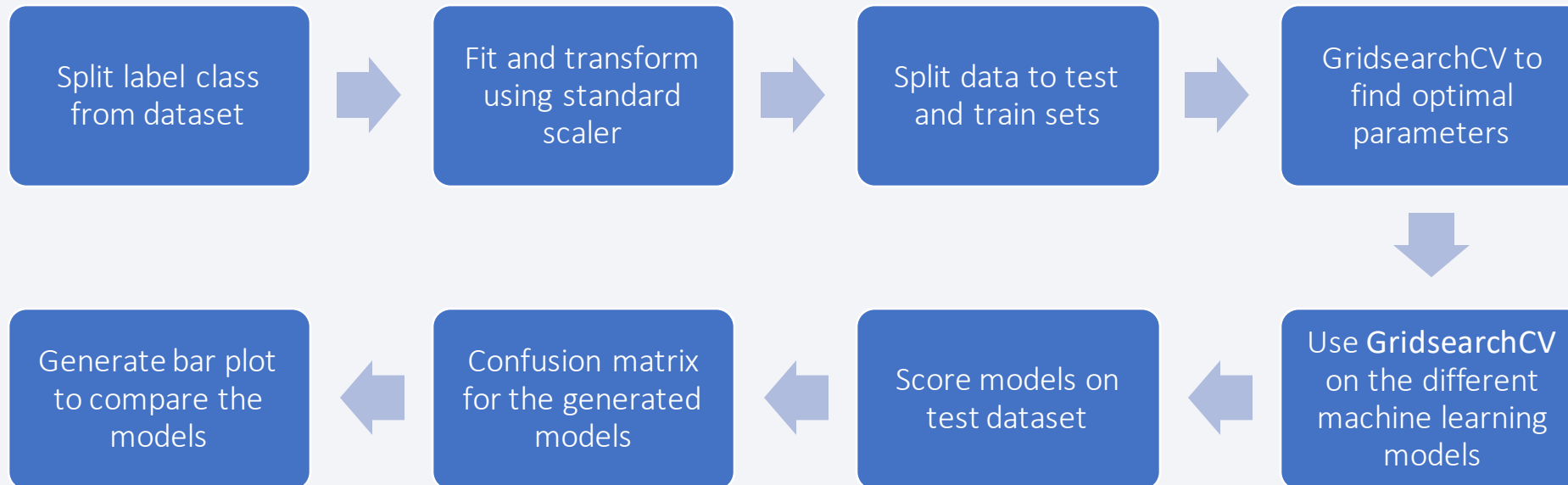
This helps to visualize successful landing relative to location.

- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

# Build a Dashboard with Plotly Dash

- Built dashboards that include pie chart and scatter plot

- Pie charts show distribution of successful launch sites and can show individual site success rate.

- Scatter plot show how success varies across launch sites, payload mass and booster version category.

- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

# Predictive Analysis (Classification)

Split label class from dataset → Fit and transform using standard scaler → Split data to test and train sets → GridsearchCV to find optimal parameters → Use **GridsearchCV** on the different machine learning models → Score models on test dataset → Confusion matrix for the generated models → Generate bar plot to compare the models

- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

# Results
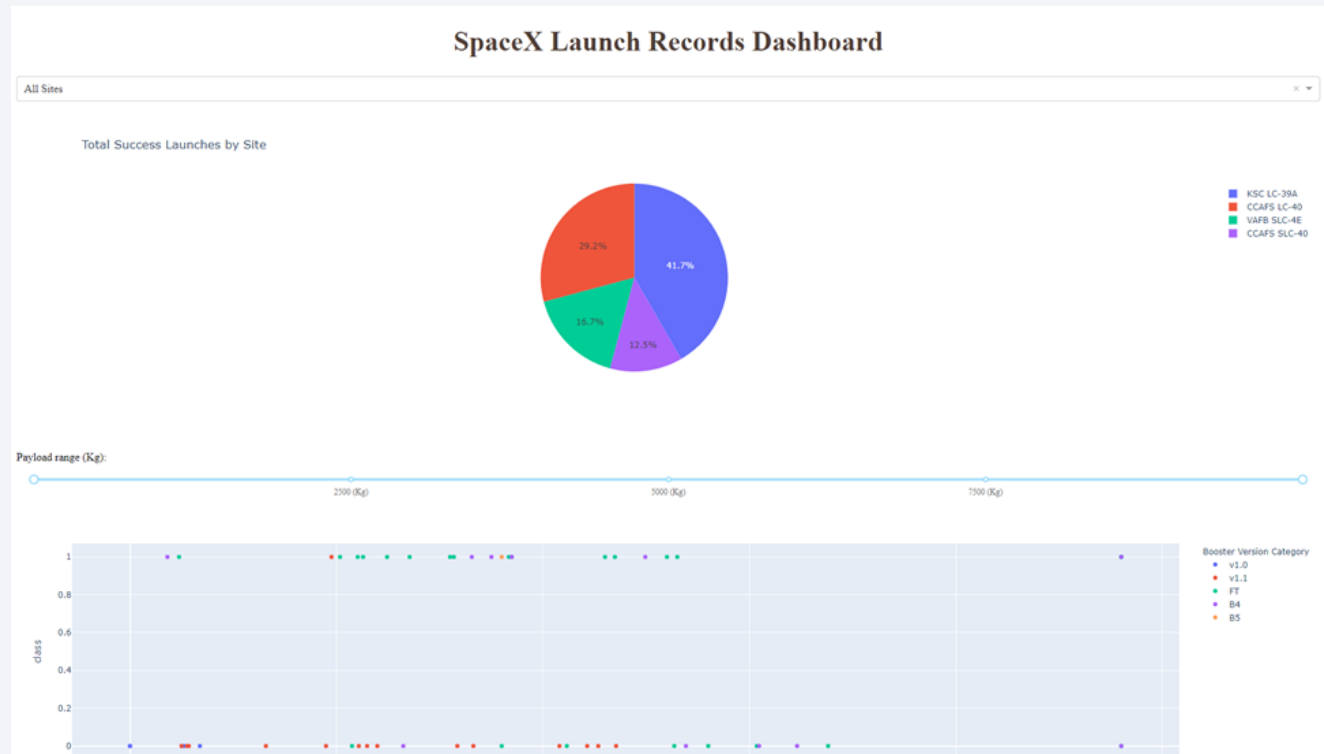


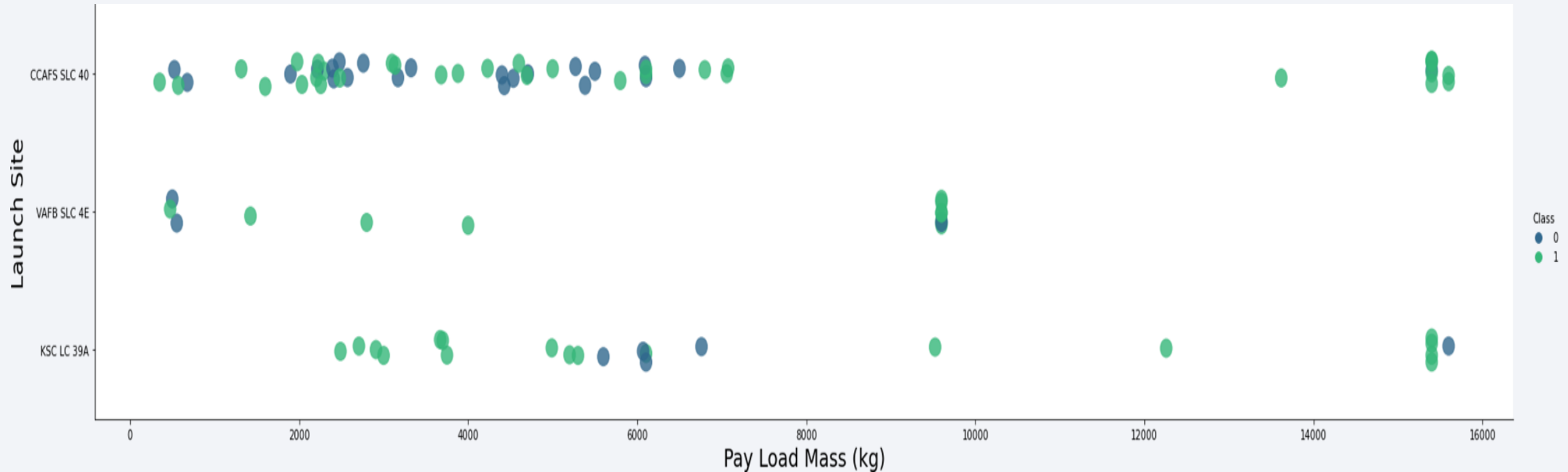- Preview of ploty dashboard

Section 2

# Insights drawn from EDA

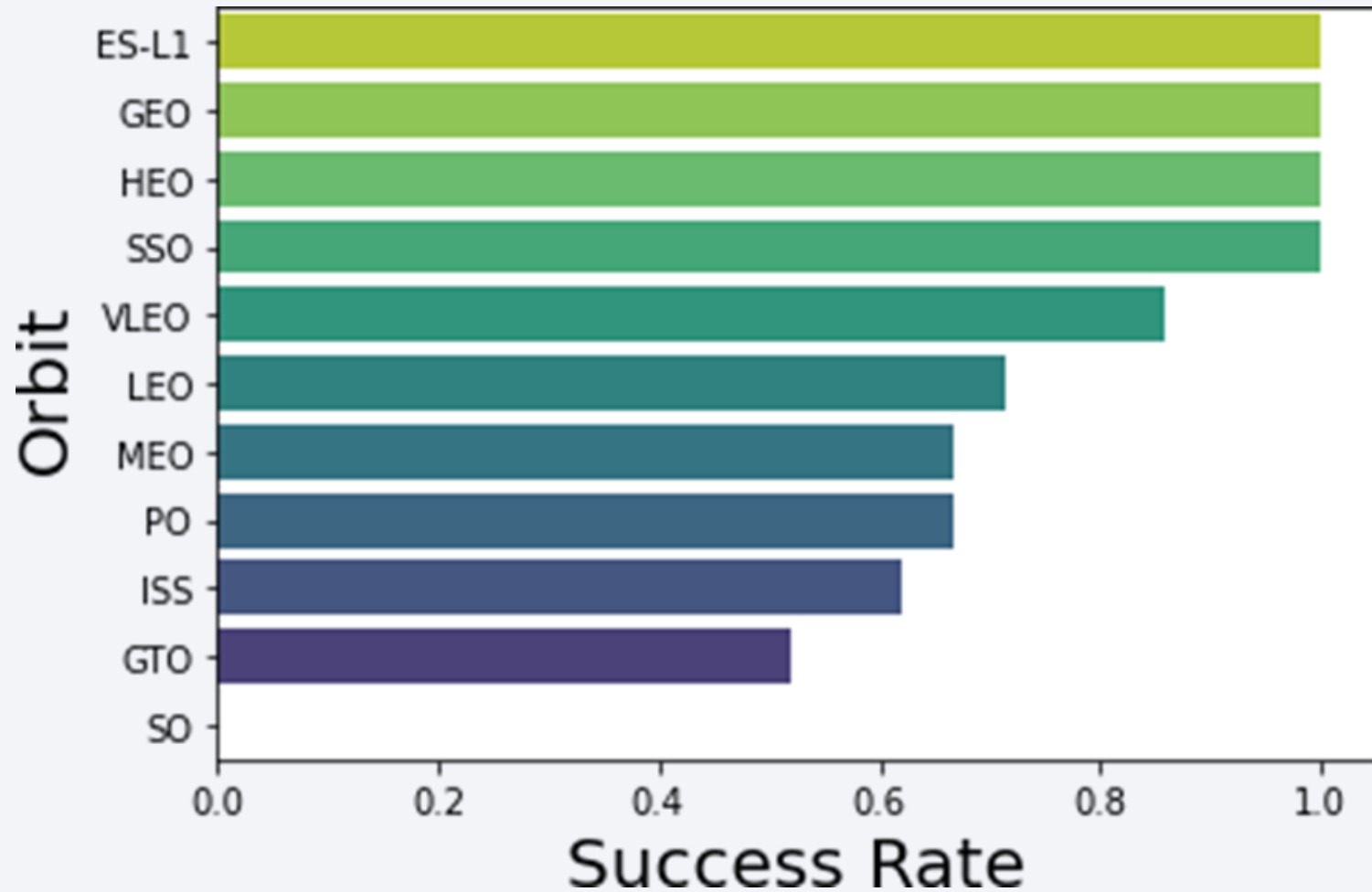# Flight Number vs. Launch Site
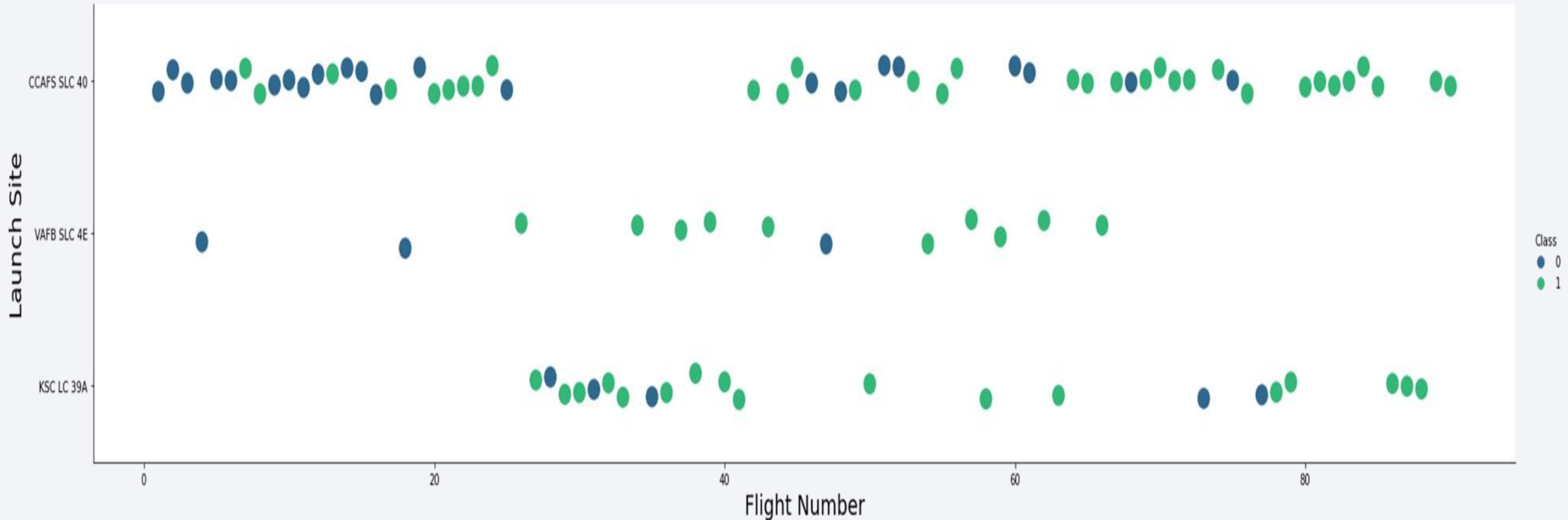


- in the above plot green indicates successful and purple indicates unsuccessful launch
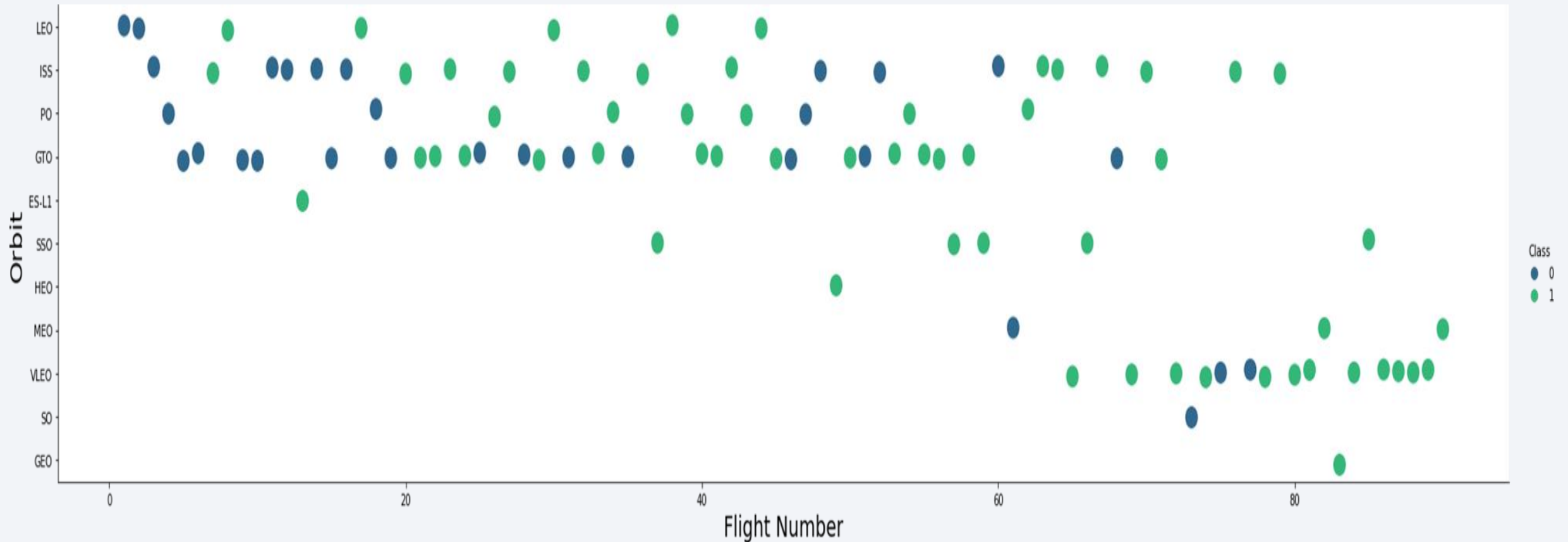
# Payload vs. Launch Site



- in the above plot green indicates successful and purple indicates unsuccessful launch
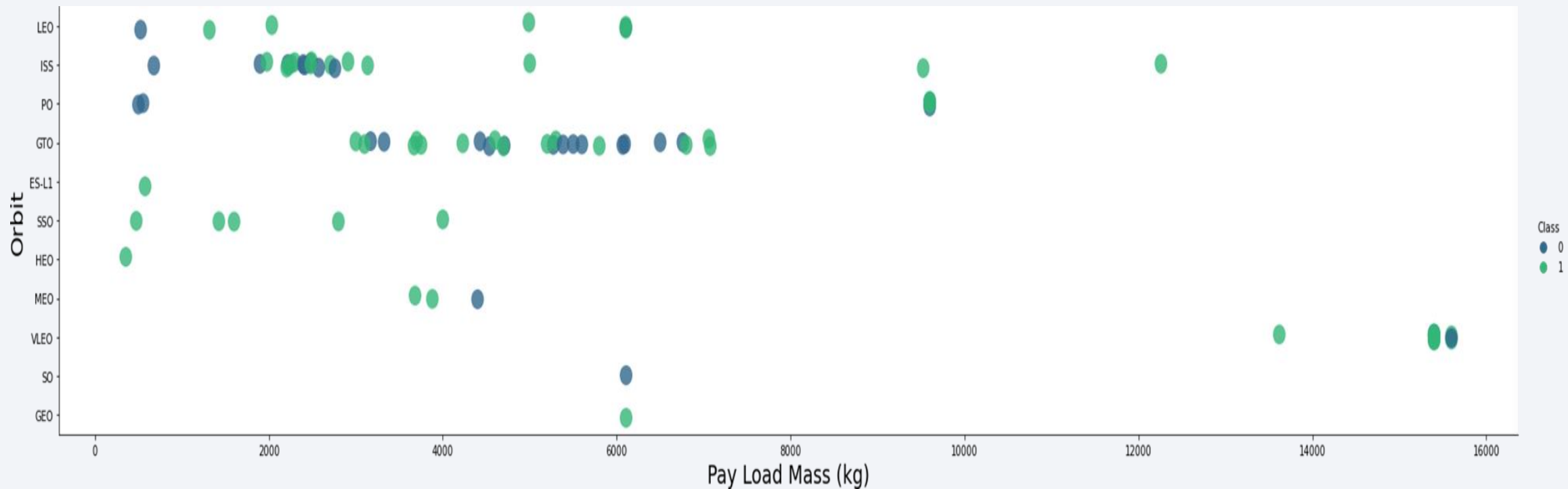
# Success Rate vs. Orbit Type



Success Rate Scale with

0 as 0%

0.6 as 60%

1 as 100%
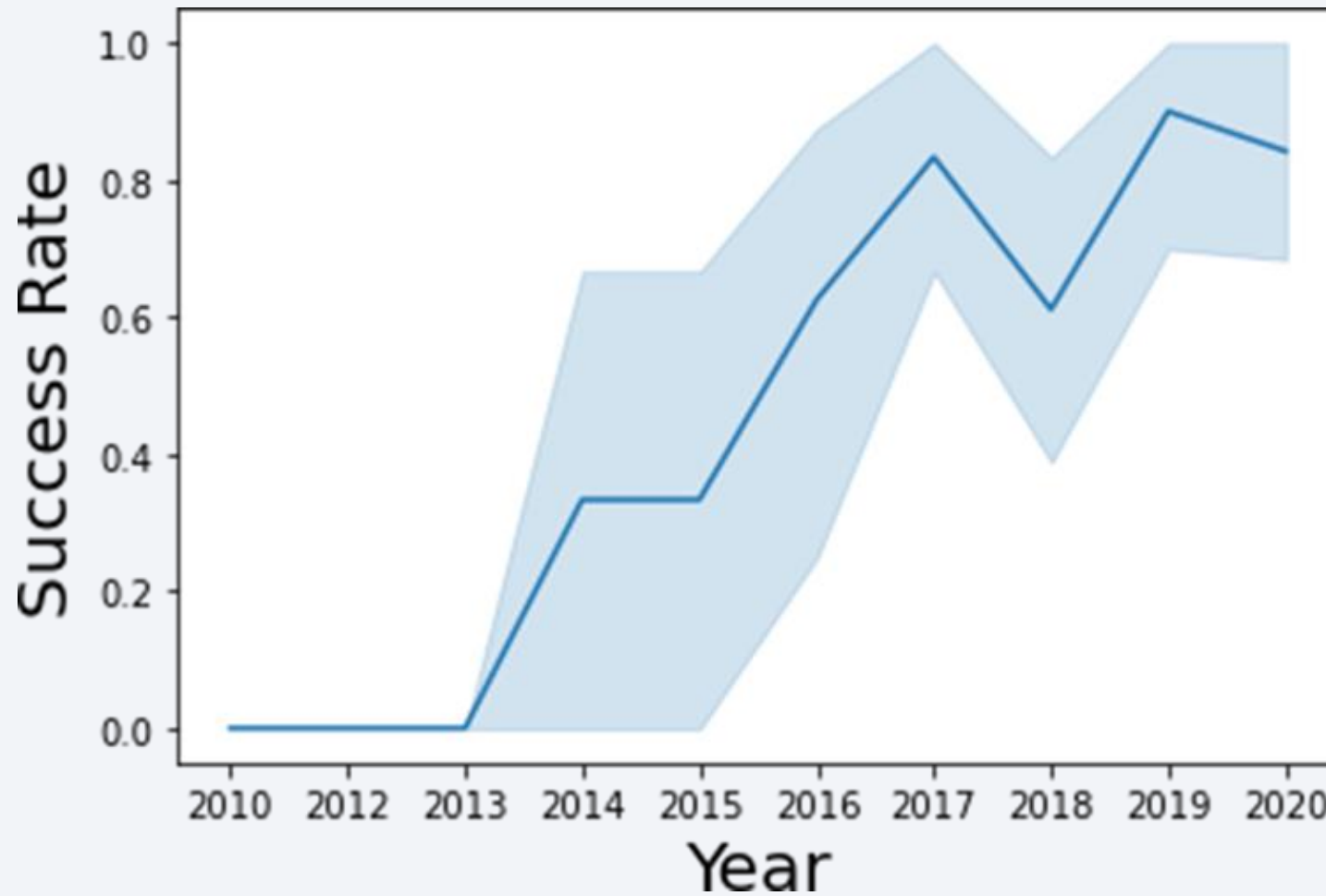
# Flight Number vs. Orbit Type



- Green indicates successful launch; Purple indicates unsuccessful launch.

# Payload vs. Orbit Type



- Green indicates successful launch; Purple indicates unsuccessful launch.

# Launch Success Yearly Trend

# All Launch Site Names

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

         * ibm_db_sa://ftb12020:***@0c77d6f:
        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.

25

# 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:***@0c77d6f2-5da9-48a9-81f8-8(
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.

# Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%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.

# Boosters Carried Maximum Payload

```
%%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.

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

```
%%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
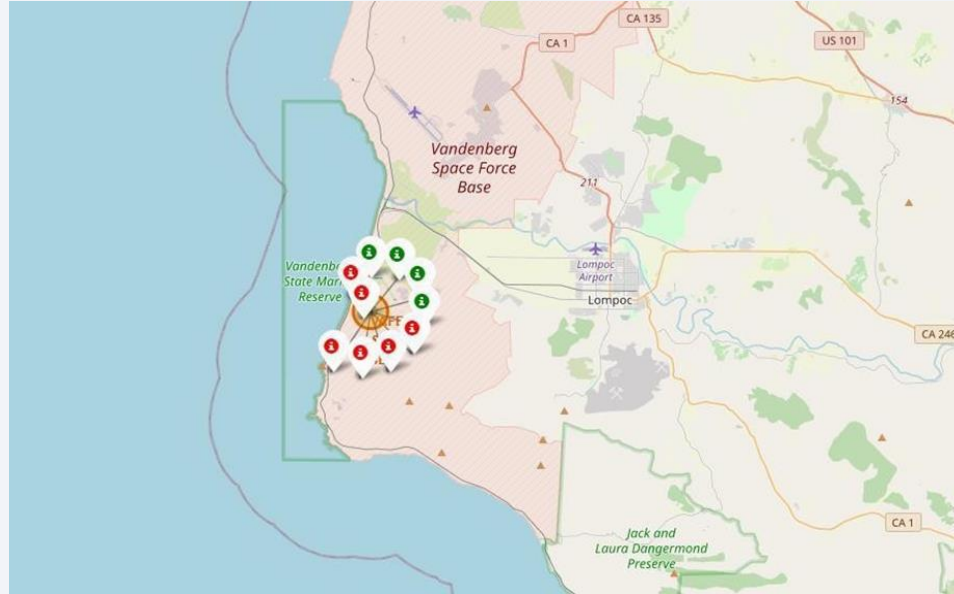
Section 3

# Launch Sites
# Proximities Analysis

# Launch Site Locations
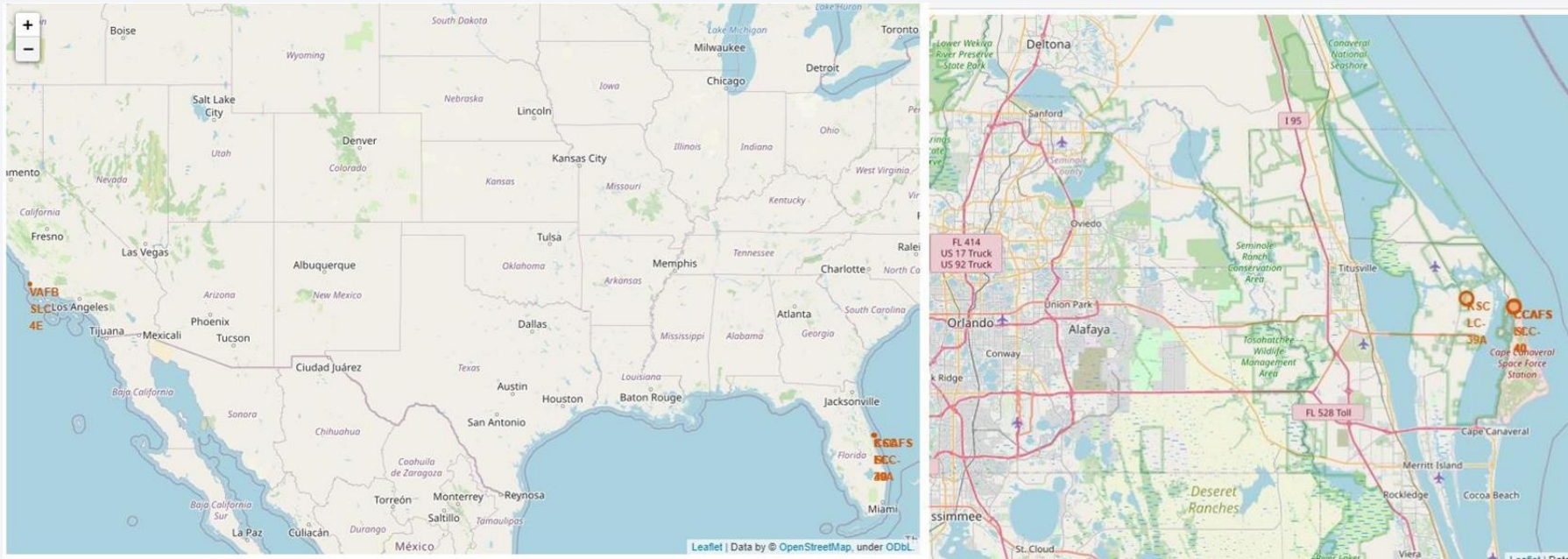


- The left map shows all launch sites relative US map. The right map shows the two Florida launch  sites since they are very close to each other. All launch sites are near the ocean.

# 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.

# Key Location Proximities
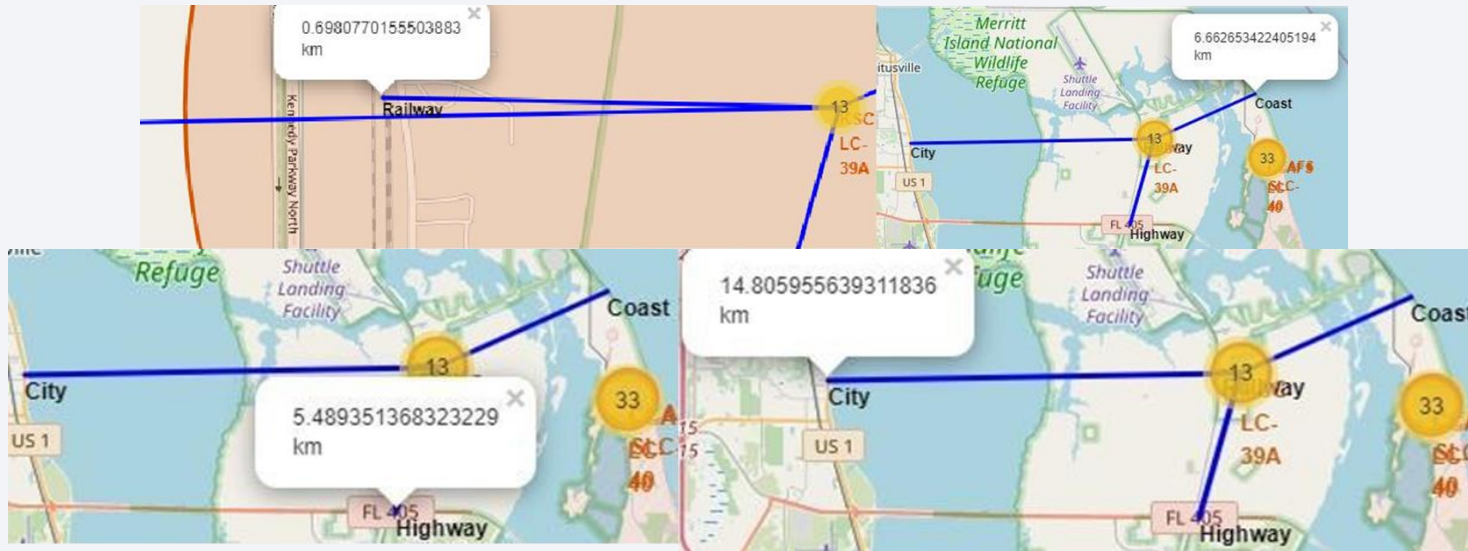


- Using KSC LC-39A as an example, launch sites are very close to railways for large part and supply  transportation. Launch sites are close to highways for human and supply transport.  Launch sites  are also close to coasts and relatively far from cities so that launch failures can land in the sea to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

Section 4

Build a Dashboard
with Plotly Dash

# Successful Launches Across Launch Sites
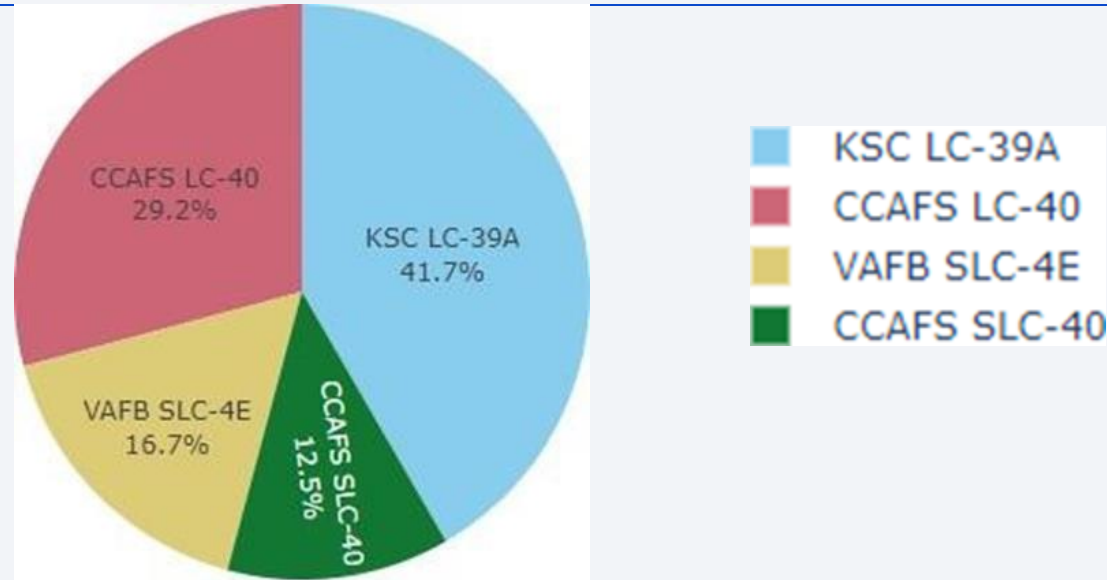


- This is the distribution of successful landings across all launch sites. CCAFS LC-40 is the old name of  CCAFS SLC-40 so CCAFS and KSC have the same amount of successful  landings, but a majority of the  successful  landings where performed before the name change. VAFB has the smallest share of successful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

# Highest Success Rate Launch Site
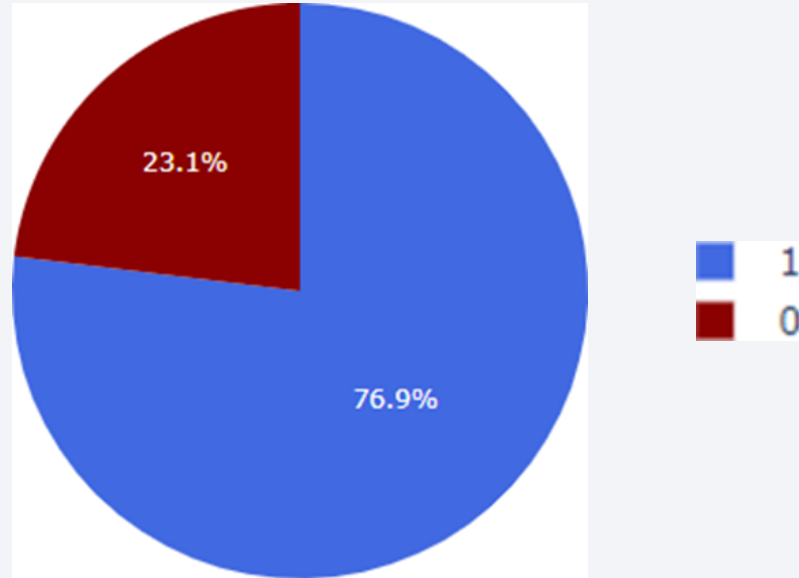


- KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.

# Payload Mass vs. Success vs. Booster  Version Category



- 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
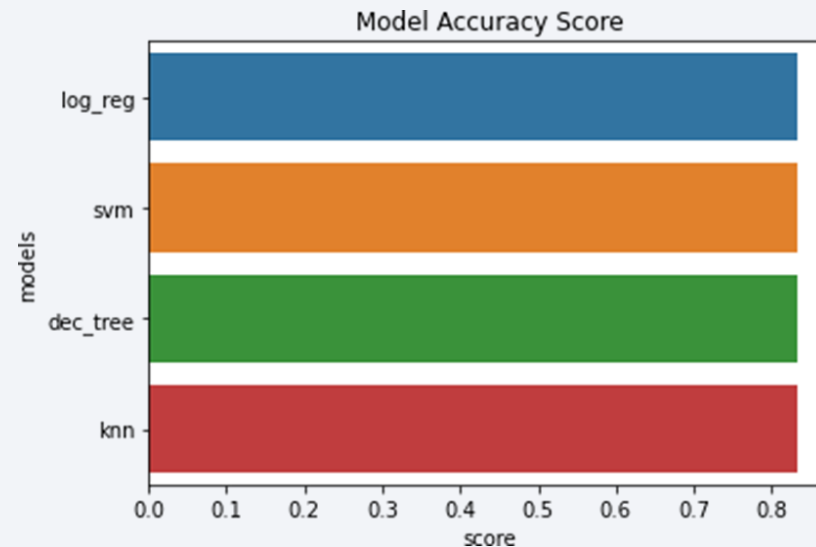


- All models had virtually the same accuracy on the test set at 83.33% accuracy. It should be noted that test size is small at only sample size of 18.

- 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
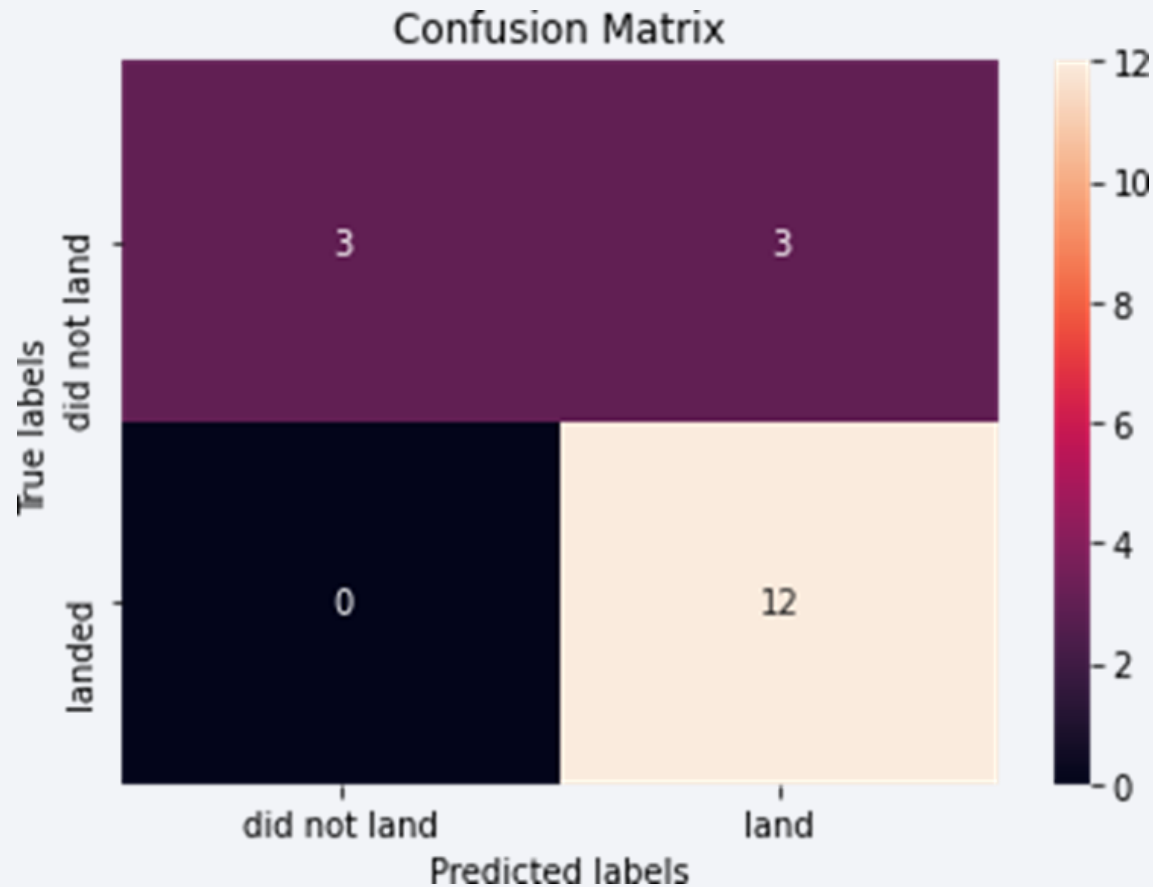


Confusion Matrix

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

- ◦The goal of model is to predict when Stage 1 will successfully land to save ~$100 million USD

- ◦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%

- ◦Allon Mask of SpaceY can use this model to predict with relatively high accuracy whether a  launch will have a successful Stage 1 landing before launch to determine whether the launch  should be made or not

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

# 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

Thank you!