DATA SCIENCE CAPSTONE PROJECT

YONGHOON LEE
2024-03-03

OUTLINE

Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

EXECUTIVE SUMMARY 1/2

Summary of Methodologies

- Data collection from SpaceX API and SpaceX Wikipedia page
- Landing outcome data into 'class' column as binary format
- Exploratory Data Analysis with data visualization and SQL
- Launch Sites Locations Analysis with Folium
- Dashboard with Plotly Dash
- One hot encoding in order to convert categorical values into numerical format
- Apply StandardScaler and GridSearchCV to find best parameters for machine learning models
- Visualize accuracy scores

EXECUTIVE SUMMARY 2/2

Summary of all results

- Machine learning models were trained and evaluated as below:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree Classifier
  - K-Nearest Neighbor
- Decision Tree resulted in the highest accuracy on training
  - However, all generated accuracy rate of 83.33% on testing
- Larger dataset would be required for better accuracy of each model

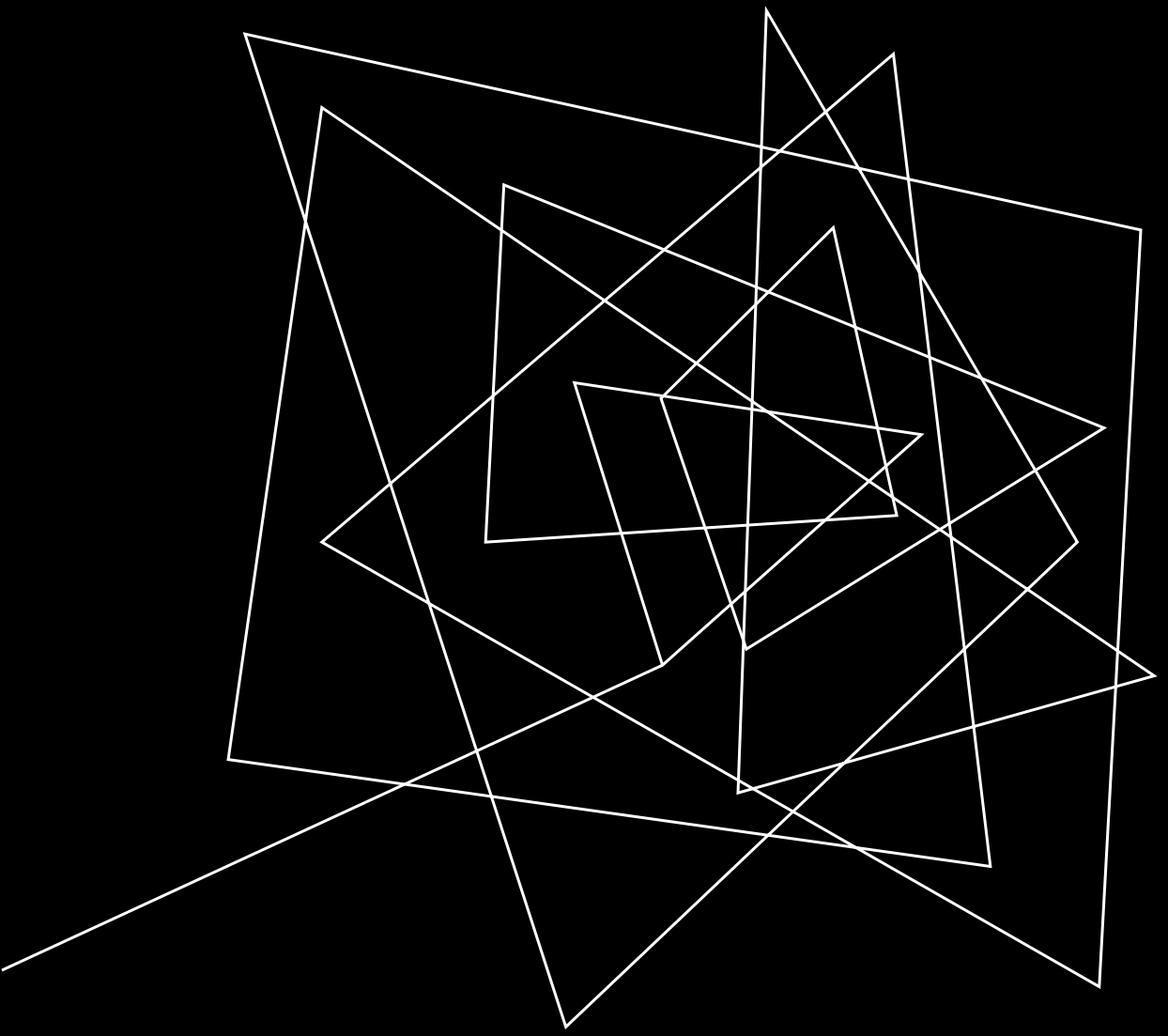
INTRODUCTION

Background:

- The commercial space age is here
  - Companies are making space travel affordable
- Space X Falcon 9 costs $62 million for each launch
  - Competitors cost upward of $165 million for each launch
- Much of the savings is because SpaceX can reuse the first stage

Problem:

- Space Y would like to compete with Space X
- If we can determine if the first stage will land, we can determine the cost of a launch.
  - This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.



SECTION 1

METHODOLOGY

METHODOLOGY

Executive Summary

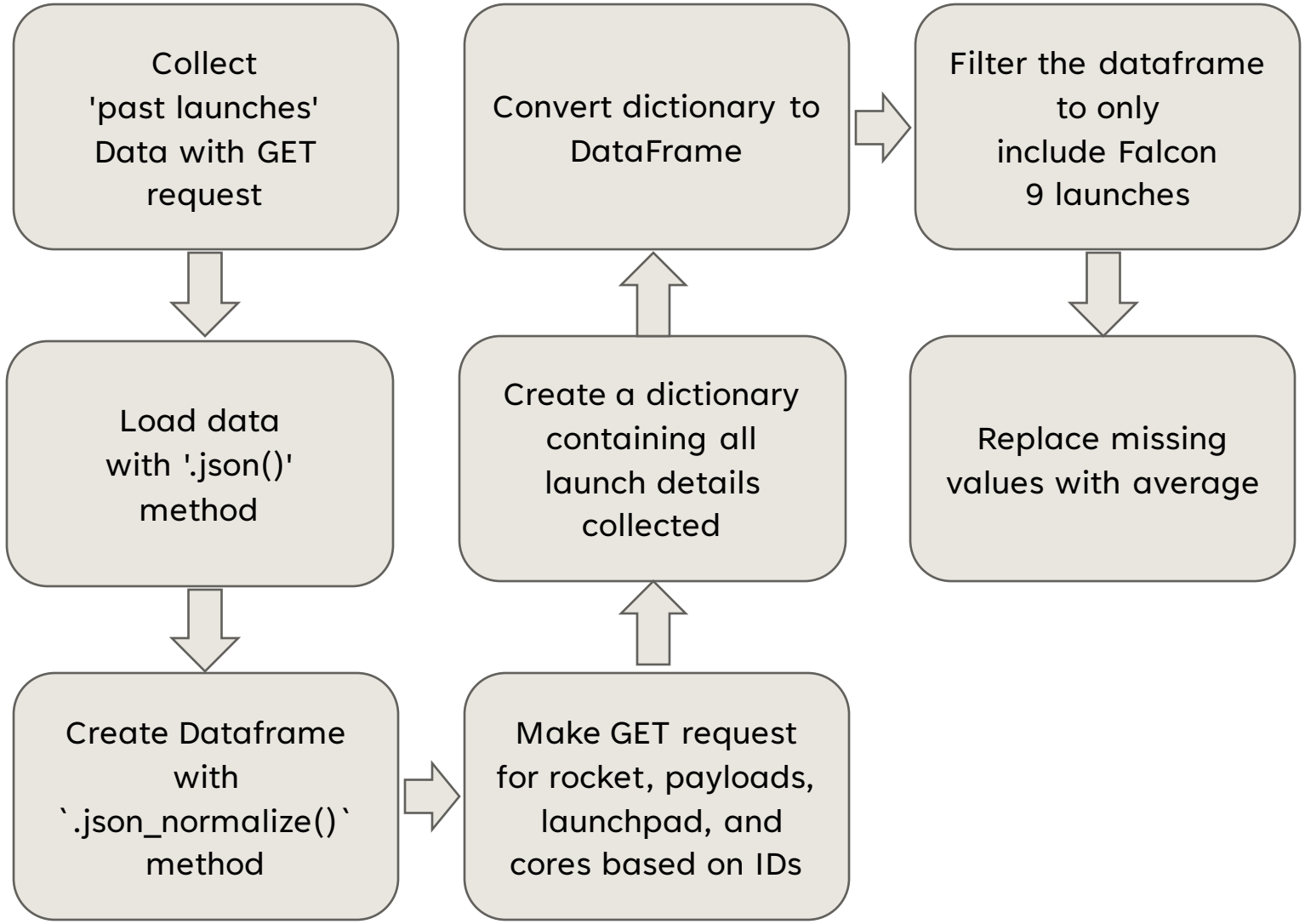
- Data collection methodology:
  - Request to the SpaceX API
  - Webscraping
- Perform data wrangling
  - Create 'class' column for landing outcome in binary format
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Linear Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbor
  - Apply GridSearchCV to find the best parameter

DATA COLLECTION

- Data collection processed with 2 methods:
  - Extract data from Space X API
    - https://api.spacexdata.com/
    - 5 endpoints used to collect data and generate dataframe with detail
  - Webscraping data from Wikipedia
    - List of Falcon 9 and Falcon Heavy launches
    - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Data columns from SpaceX API:
  - 'FlightNumber', 'Date', 'BoosterVersion', 'PayloadMass', 'Orbit', 'LaunchSite', 'Outcome', 'Flights', 'GridFins', 'Reused', 'Legs', 'LandingPad', 'Block', 'ReusedCount', 'Serial', 'Longitude', 'Latitude'
- Data Columns from Wikipedia:
  - 'Flight No.', 'Launch site', 'Payload', 'Payload mass', 'Orbit', 'Customer', 'Launch outcome', 'Version Booster', 'Booster landing', 'Date', 'Time'

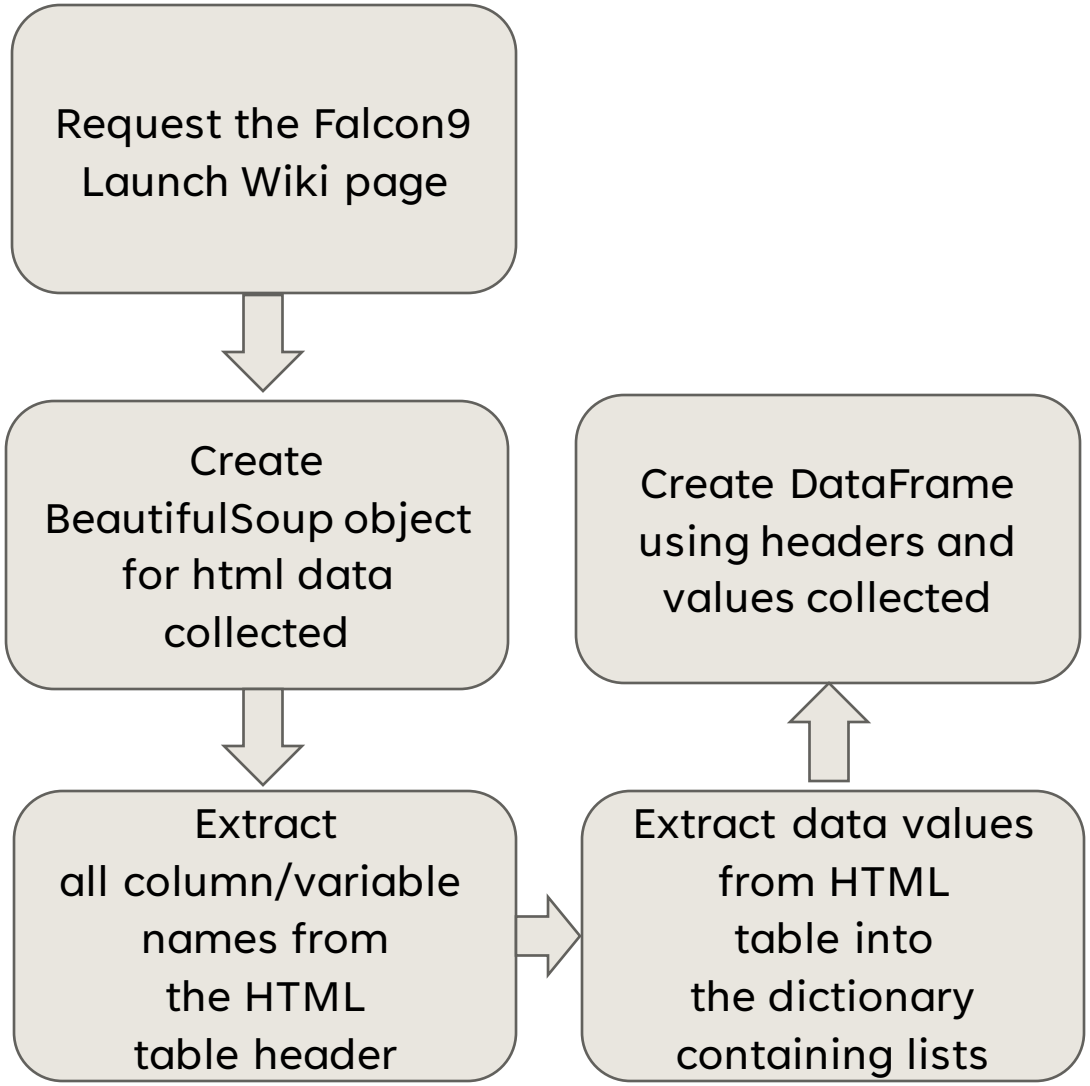
DATA COLLECTION – SPACEX API

Github: https://github.com/ylhoony/ibm-ds-course/blob/main/10_Applied_DS_Capstone/01_data-collection-api.ipynb



DATA COLLECTION – SCRAPING

Github: https://github.com/ylhoony/ibm-ds-course/blob/main/10_Applied_DS_Capstone/02_webscraping.ipynb



DATA WRANGLING

- Load data
- Review missing values and data types
- Review items below using '.value_counts()'
  - the number of launches on each site
  - the number and occurrence of each orbit
  - the number and occurrence of mission outcome of the orbits
- Create a landing outcome label from Outcome column
  - Create 'Class' column for landing success in binary format
- Github: https://github.com/ylhoony/ibm-ds-course/blob/main/10_Applied_DS_Capstone/03_data-wrangling.ipynb

EDA WITH DATA VISUALIZATION

- Exploratory Data Analysis with visualization is performed to see the relationship between variables
- EDA conducted through visualization for:
  - Flight Number and Payload Mass – Scatter plot
  - Flight Number and Launch Site – Category plot (default Strip plot)
  - Payload and Launch Site – Scatter plot
  - Success rate of each orbit type – Bar chart
  - Flight Number and Orbit type – Scatter plot
  - Payload and Orbit type – Scatter plot
  - Launch success yearly trend – Line plot
- Converted categorical data into numerical representations
  - Used 'one-hot-encoding' with '.get_dummies()' method

- Github: https://github.com/ylhoony/ibm-ds-course/blob/main/10_Applied_DS_Capstone/04_eda-dataviz.ipynb

EDA WITH SQL

- Conducted to:
  - Understand the Spacex DataSet
  - Load the dataset into the corresponding table in a Db2 database or SQLite
  - Execute SQL queries to answer assignment questions
    - Regarding Launch site, Payload, Mission outcome, Landing outcome

- Github: https://github.com/ylhoony/ibm-ds-course/blob/main/10_Applied_DS_Capstone/05_eda-sql.ipynb

BUILD INTERACTIVE MAP WITH FOLIUM

- Create Folium map object with methods below:
  - .Map() - to create Folium map object
  - .Circle() - to add highlighted circle area
  - .Marker() - to add marker to the map
  - .MarkerCluster() - to add marker cluster displaying total number of markers
  - .MousePosition() - to get coordinate for a mouse over a point on the map
  - .PolyLine() - to add lines between points on the map
- Folium map marks all launch sites with landing outcomes, as well as a proximity to key locations such as coast, railway, highway, and city
- It gives us the information on the map intuitively where launch sites are located with landing outcomes in respective site

- Github: https://github.com/ylhoony/ibm-ds-course/blob/main/10_Applied_DS_Capstone/06_interactive-map-folium.ipynb
- Use this link if you do not see the maps on Github: https://nbviewer.org/github/ylhoony/ibm-ds-course/blob/main/10_Applied_DS_Capstone/06_interactive-map-folium.ipynb

BUILD DASHBOARD WITH PLOTLY DASH

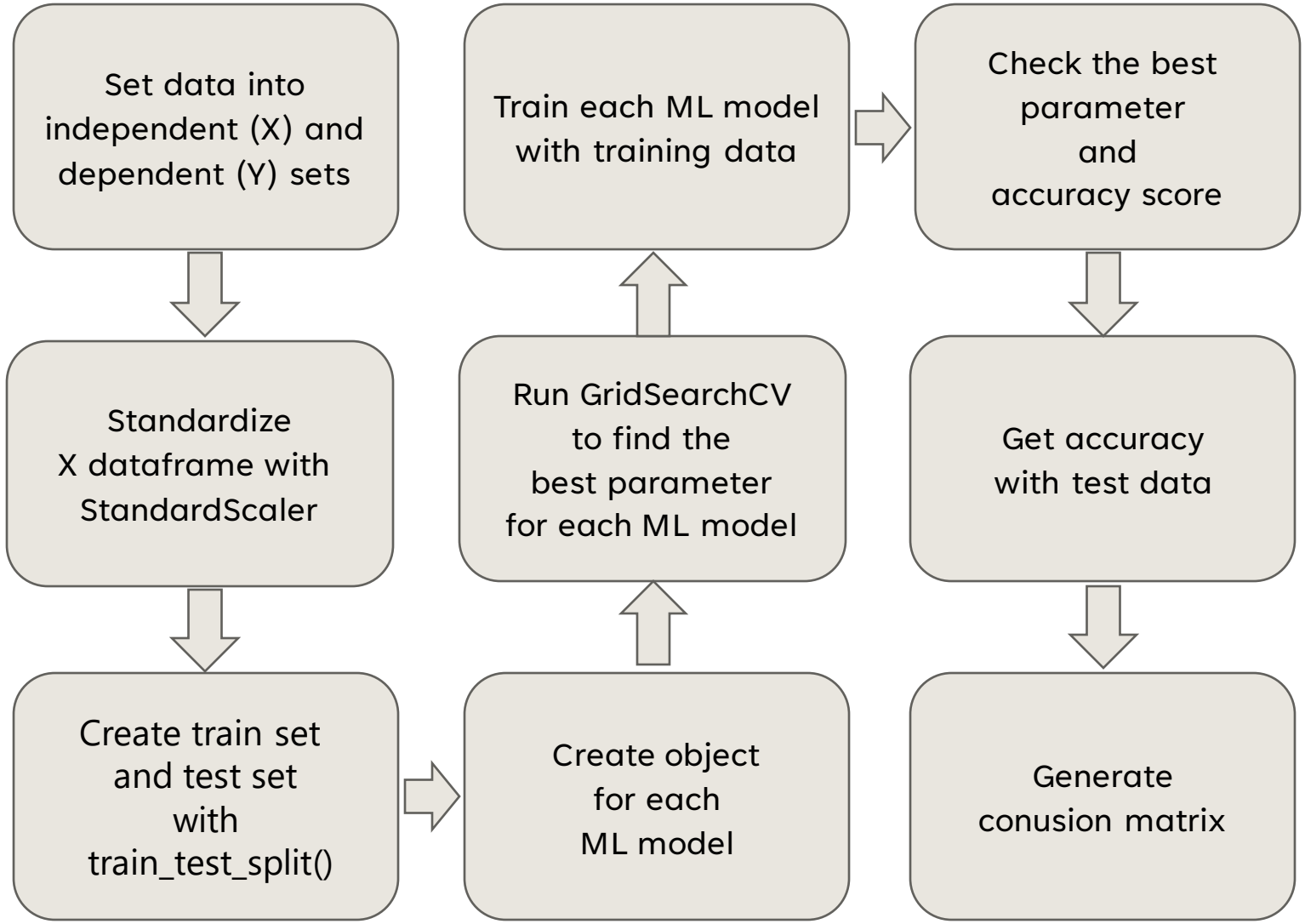
- Dashboard displays a pie chart and a scatter plot.
- Pie chart shows
  - Proportion of successful landings across all launch sites
  - Or proportions of landing success/failure at each launch site
- Scatter plot
  - Takes two inputs: Launch Sites and Payload Mess range
  - Returns the distribution of landing success/failure by payload mess range selected with booster version category in different color

- Github: https://github.com/ylhoony/ibm-ds-course/blob/main/10_Applied_DS_Capstone/07_dashboard-dash-app.py

PREDICTIVE ANALYSIS (CLASSIFICATION)

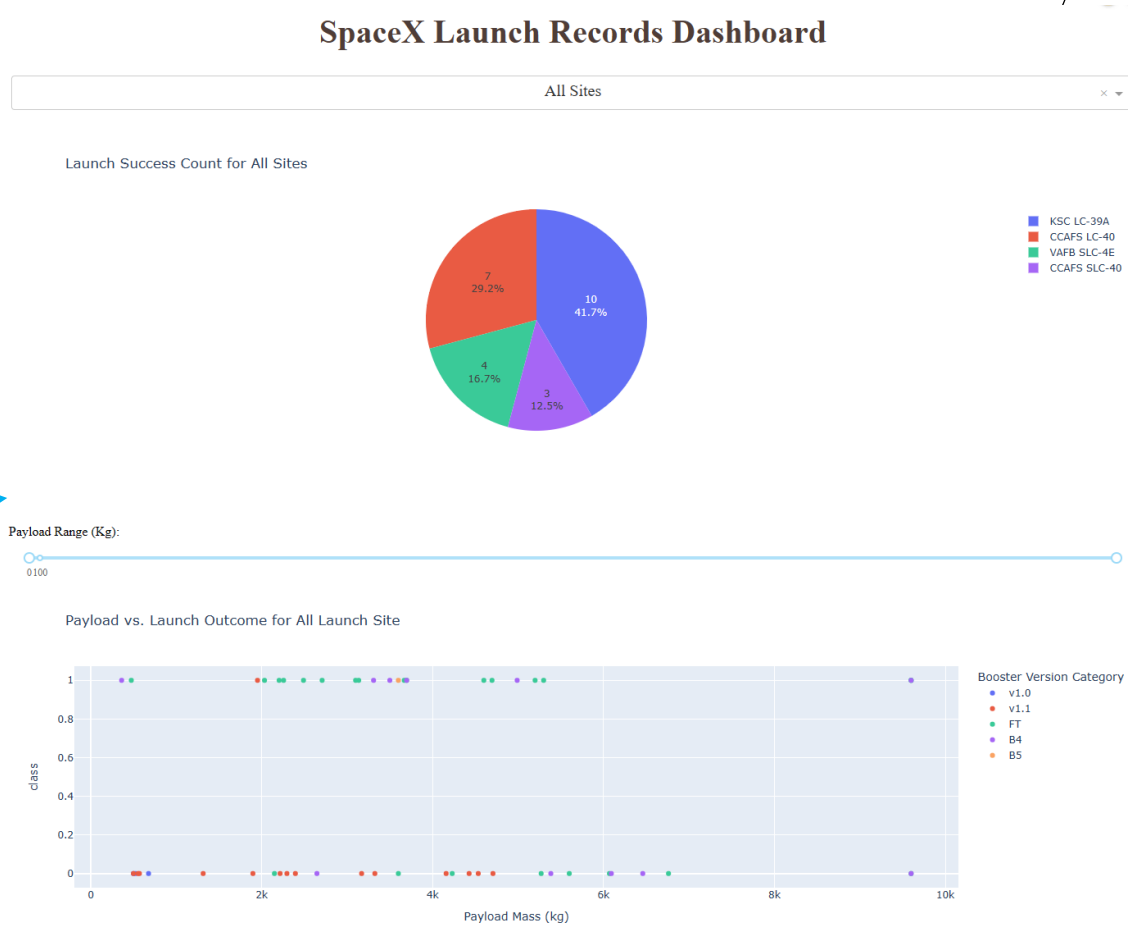
- ML techniques implemented
  - Linear Regression,
  - Support Vector Machine,
  - Decision Tree
  - K-Nearest Neighbor
- CV = 10

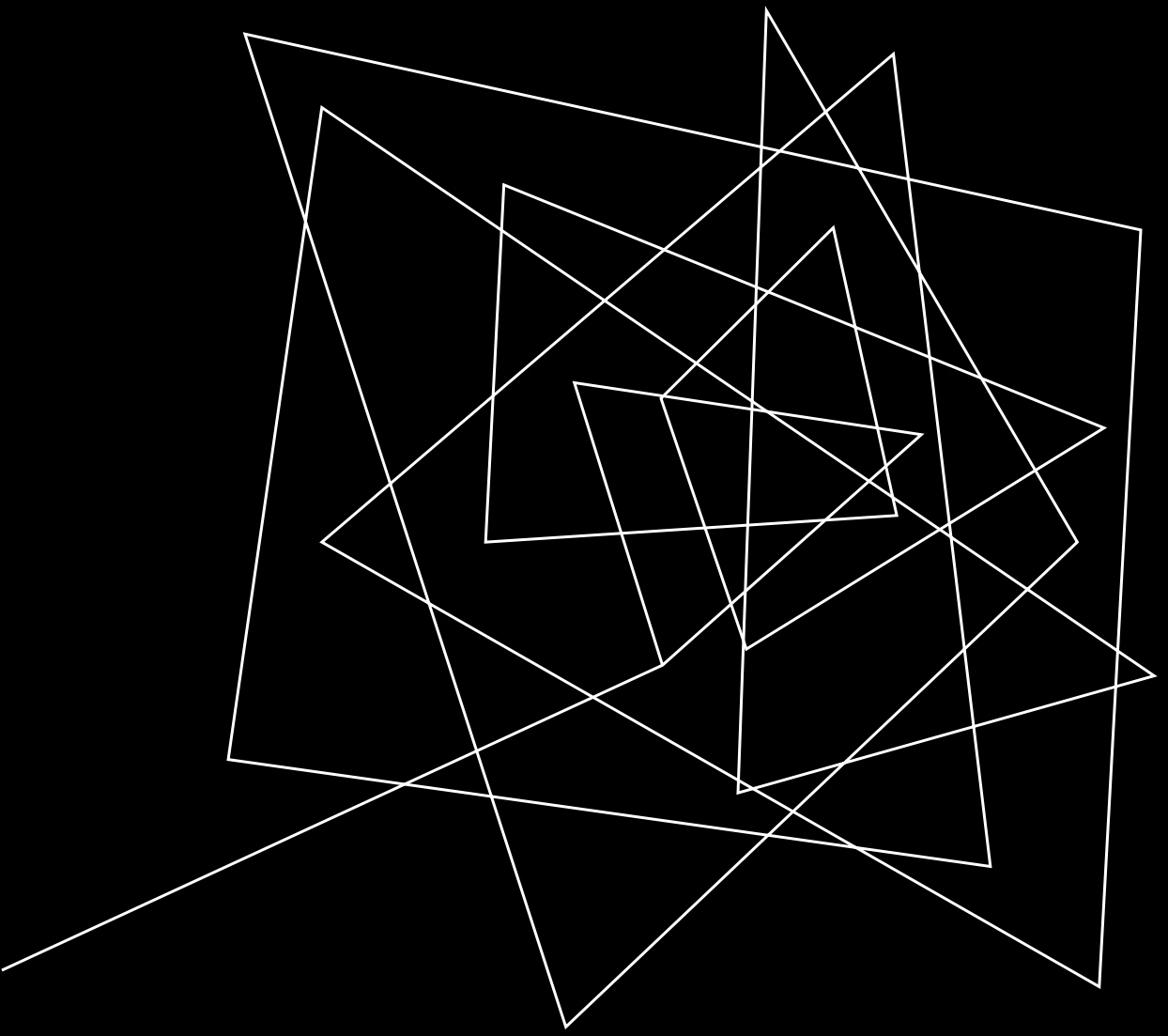
Github: https://github.com/ylhoony/ibm-ds-course/blob/main/10_Applied_DS_Capstone/08_machine-learning-predictive-analysis.ipynb



RESULT

- Launch success rate improved over time to about 80% from the beginning
- Most of launches are with payload below 8,000kg
- Mission success rate is about 99% while landing outcome has lower success rate
- Dashboard in Plotly Dash screenshot
- The accuracy of ML models is all same at 83.33%



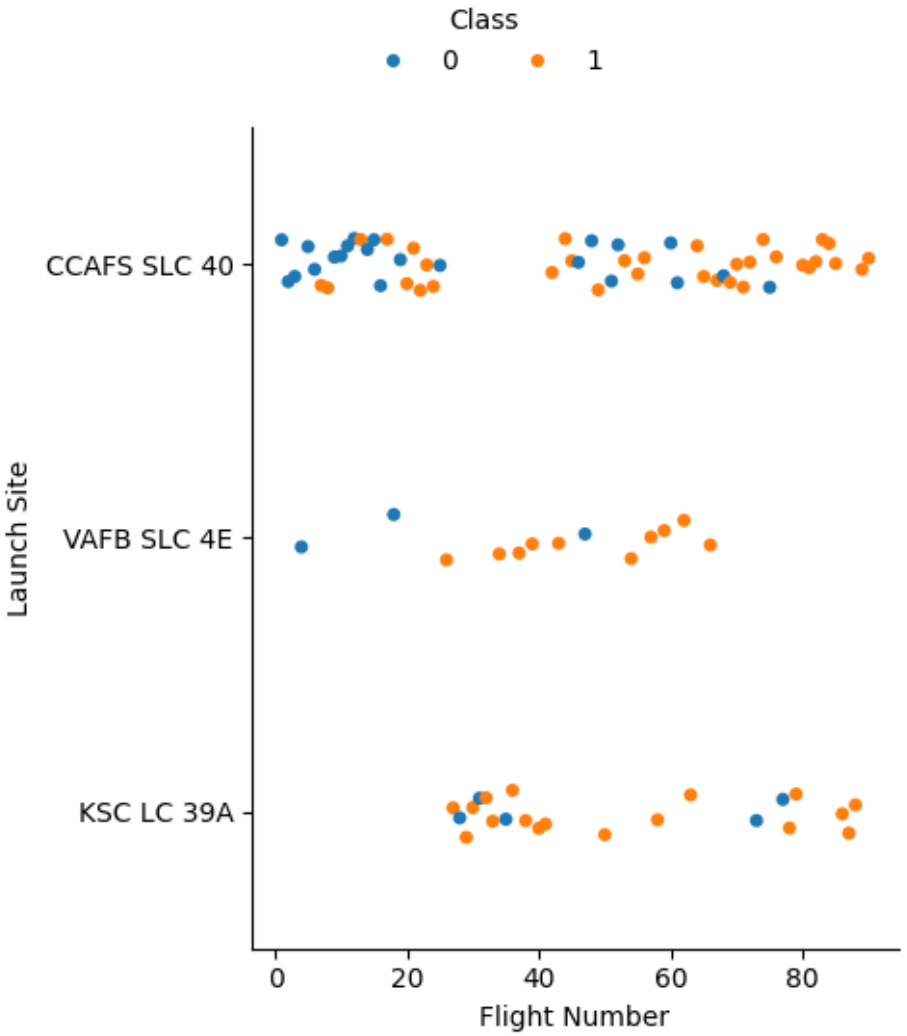


SECTION 2

INSIGHTS DRAWN FROM EDA (EXPLORATORY DATA ANALYSIS)

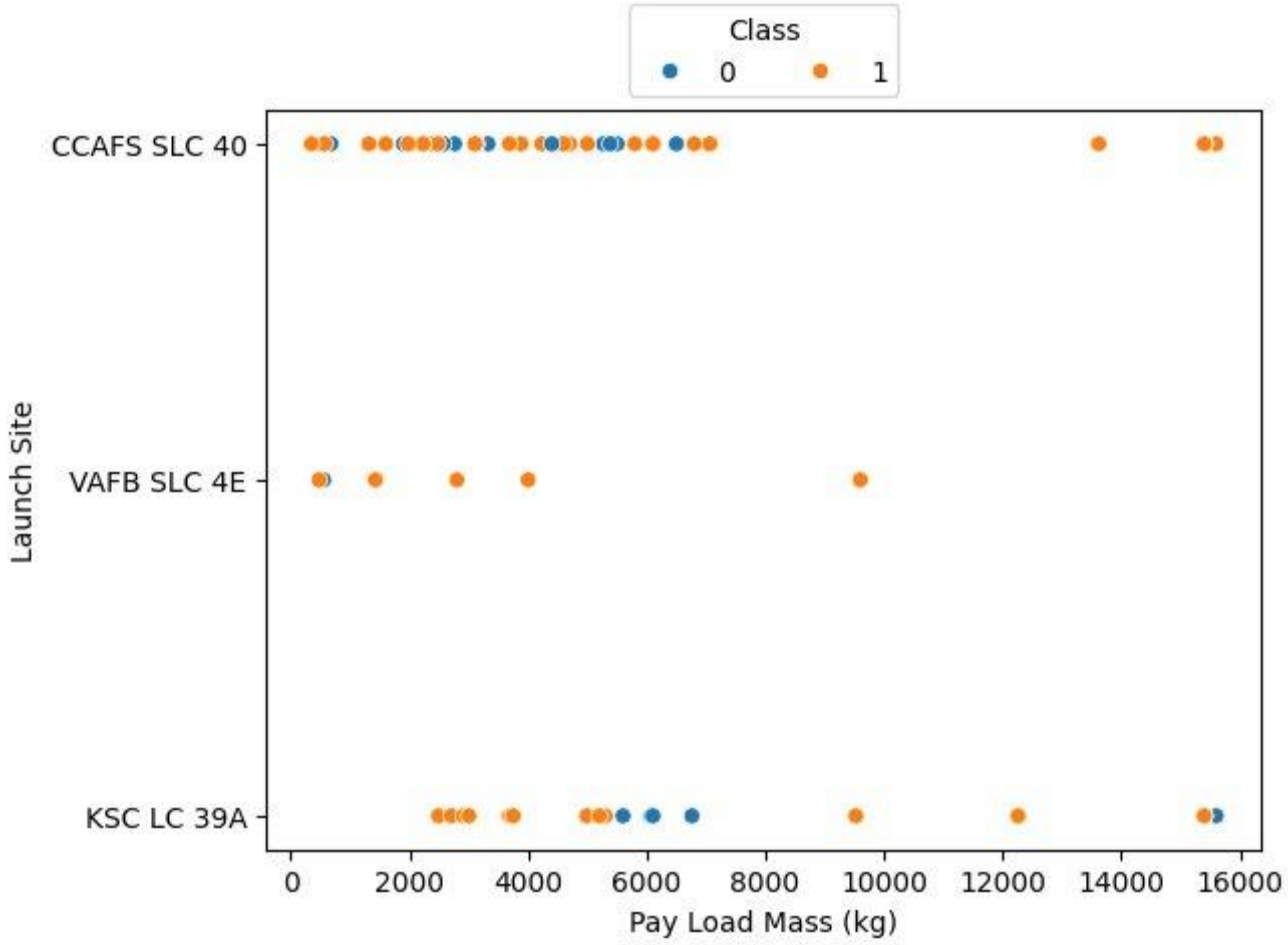
FLIGHT NUMBER VS. LAUNCH SITE

- 'CCAFS SLC 40' has the highest number of launching, but it did not operate flight number between around 25 and early 40. It has only 60% of success rate, but can see the improvement over time.
- 'VAFB SLC 4E' has the least number of launches, but it has not launched since flight number around 70. It has 77% of success rate, but the number of launches is low.
- 'KSC LC 39A' started launching around 25. A lot of launches were implemented till early 40s of flight number since the start. It also has 77% of launches with a fair number of launches.



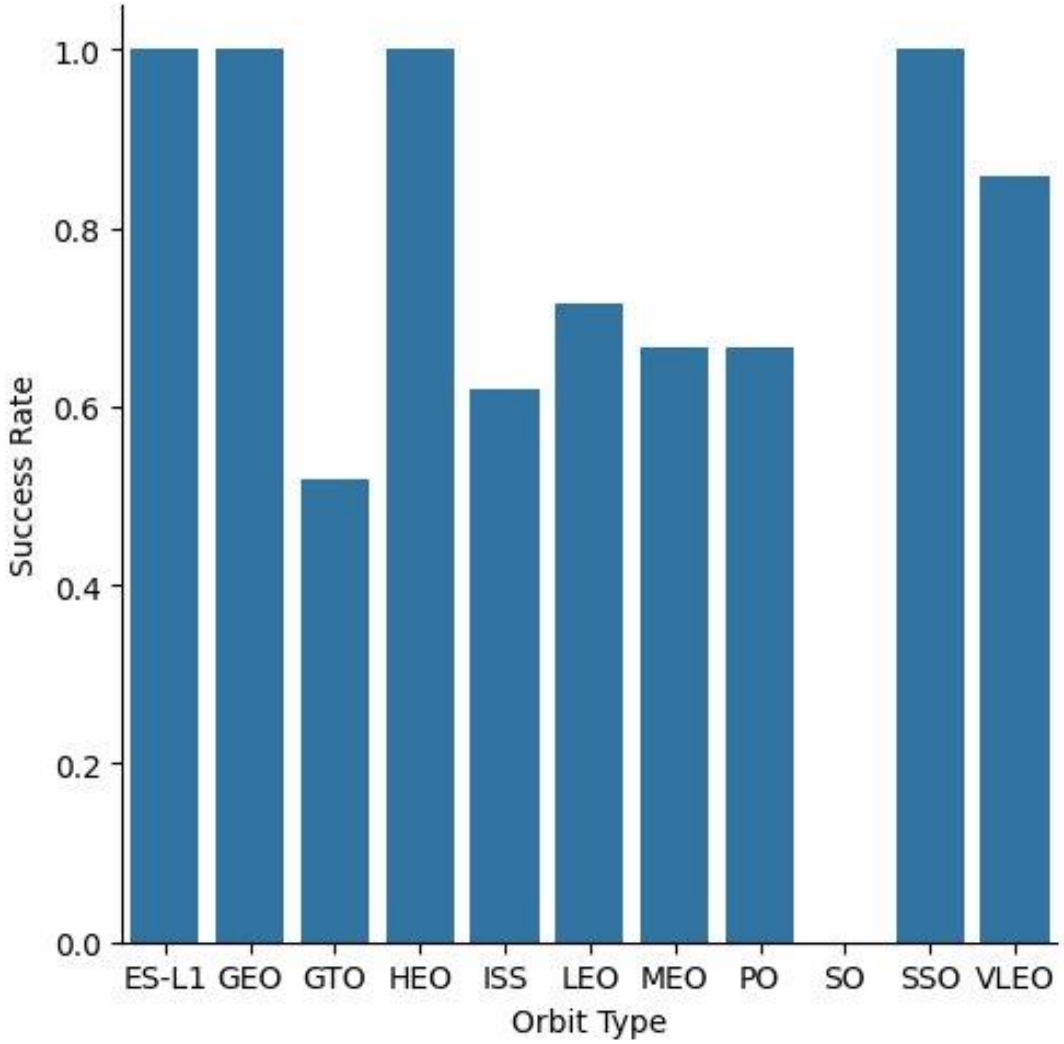
PAYLOAD VS. LAUNCH SITE

- For the VAFB-SLC launch site, there are no rockets launched for heavy payload mass(greater than 10000).
- Most of launches are with payload below 8000



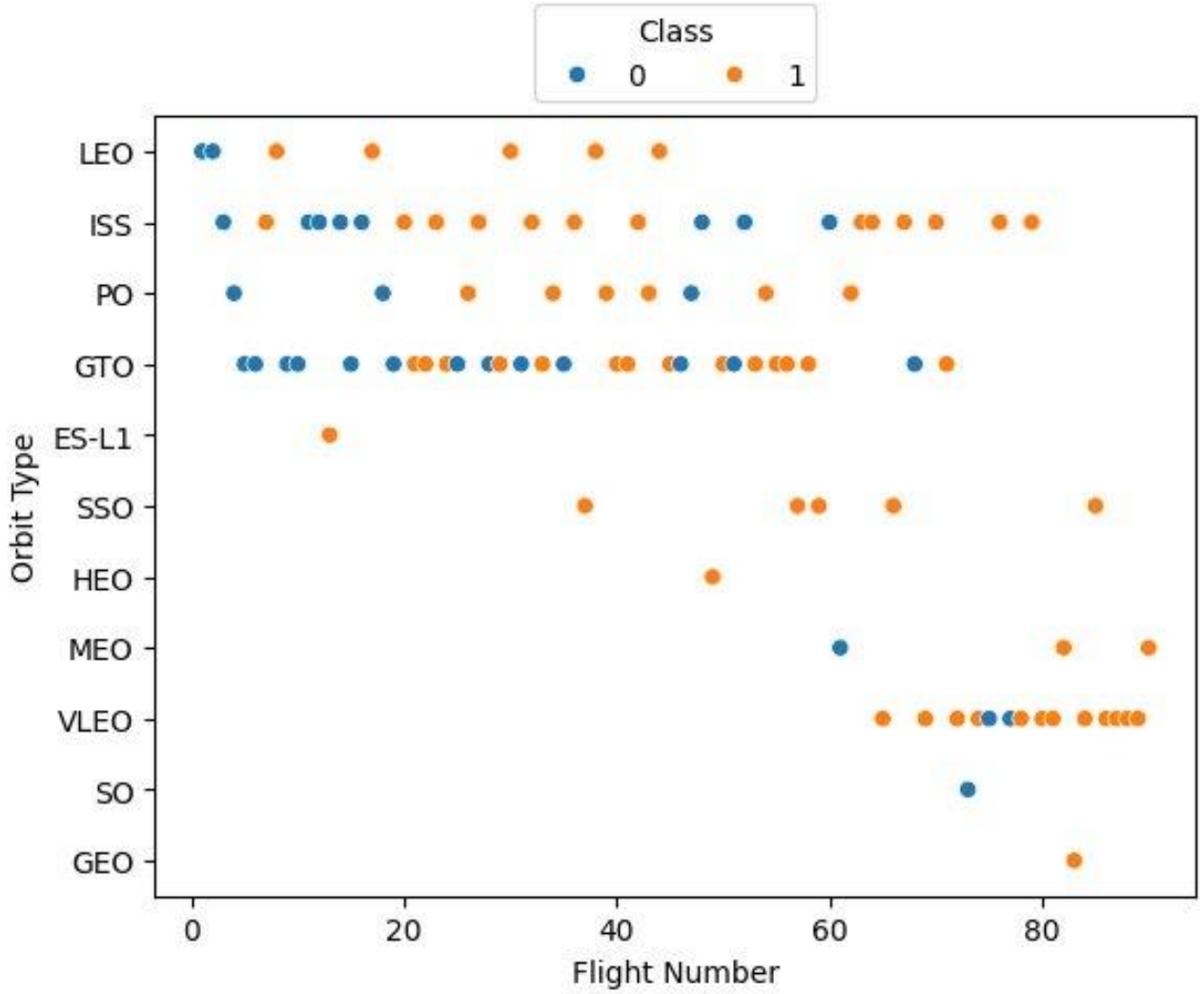
SUCCESS RATE VS. ORBIT TYPE

- Orbit type 'SO' has no success for launch.
- Orbit type 'ES-L1', 'GEO', 'HEO', and 'SSO' have succeeded 100% of launches.
- Orbit type 'VLEO' has over 80% of launch success.
- The rest of orbit types have success rate between 50% and 80%.



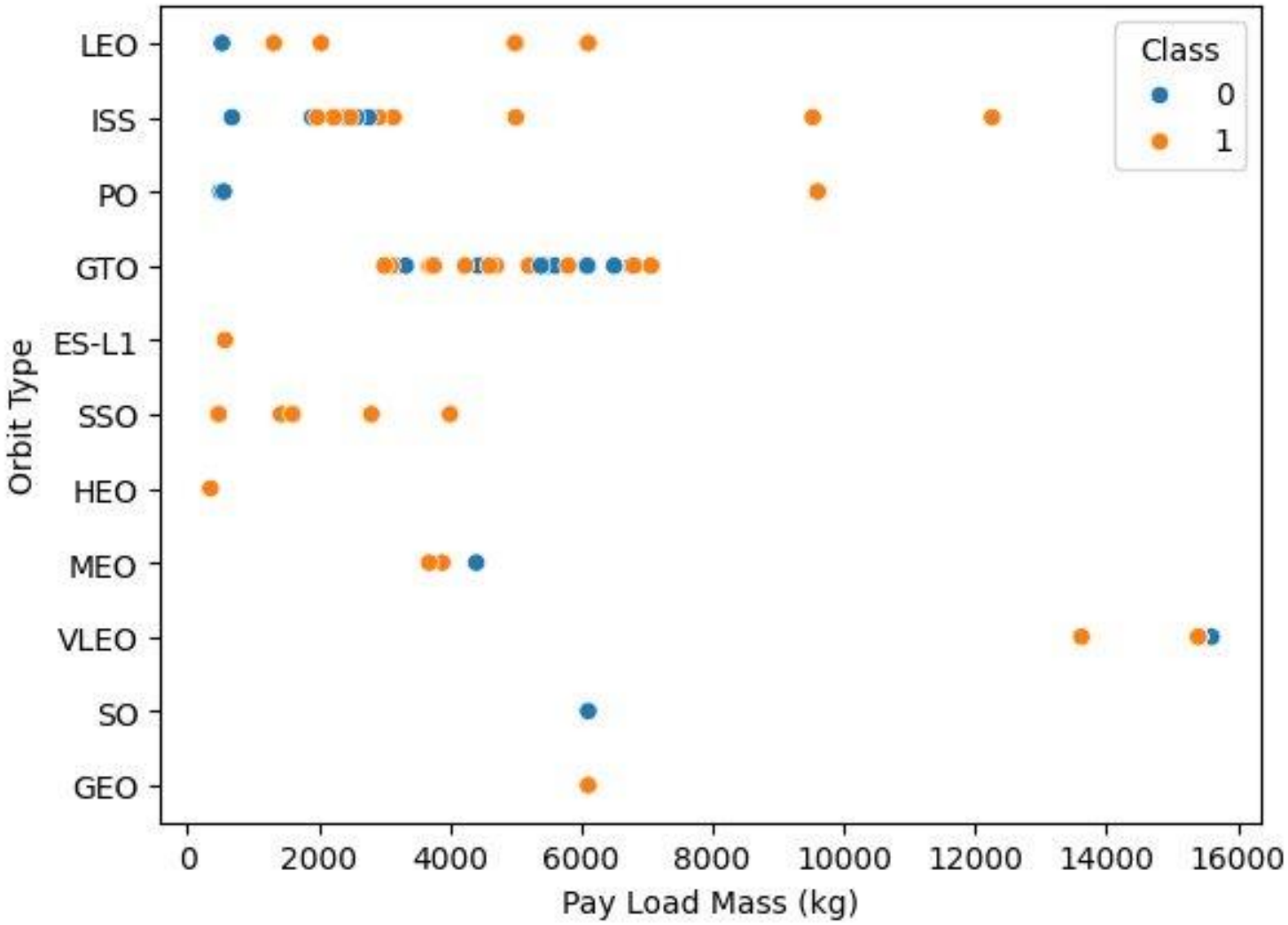
FLIGHT NUMBER VS. ORBIT TYPE

- In the LEO orbit the Success appears related to the number of flights.
- On the other hand, there seems to be no relationship between flight number when in GTO orbit.



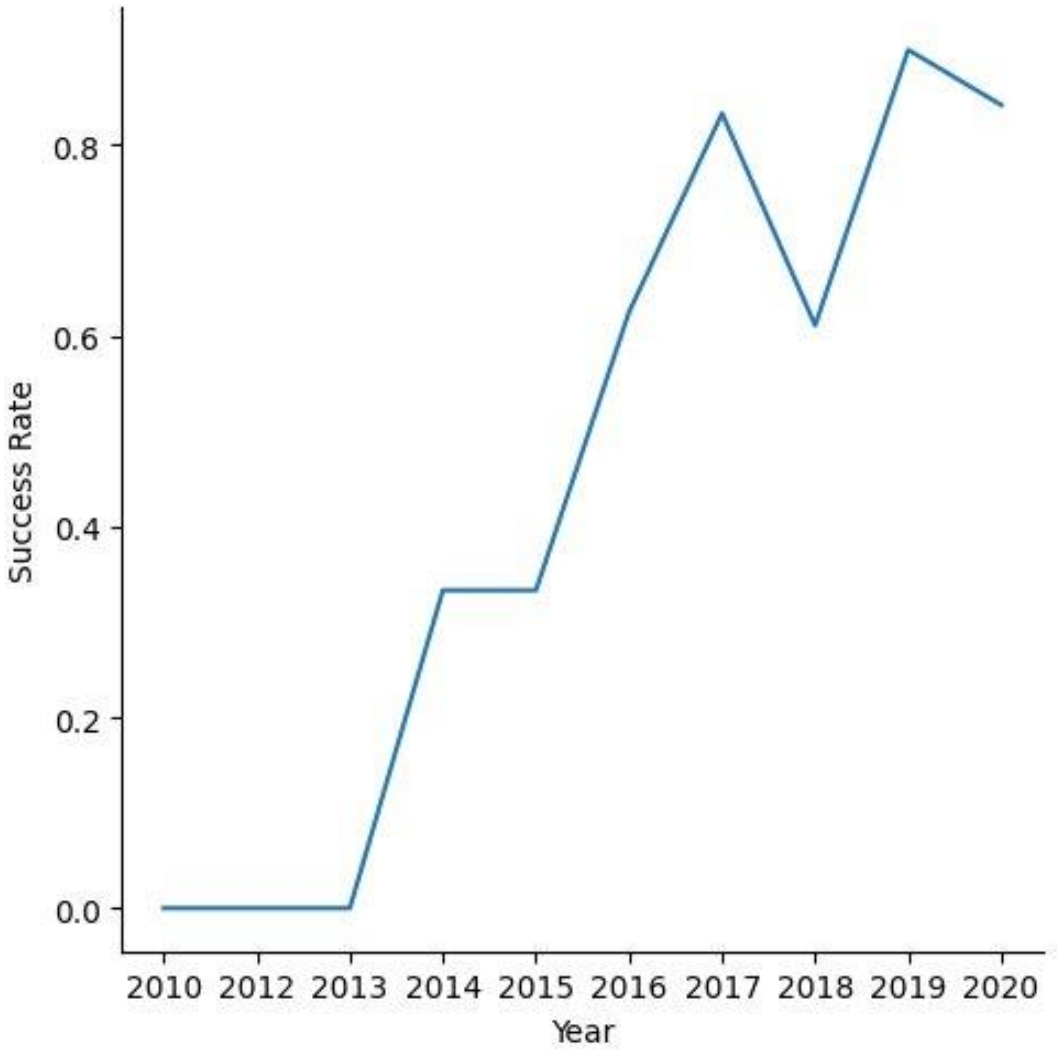
PAYLOAD VS. ORBIT TYPE

- With heavy payloads, the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However, for GTO we cannot distinguish this well as both successful and unsuccessful landings are both here.



LAUNCH SUCCESS YEARLY TREND

- The success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.



ALL LAUNCH SITE NAMES

- Query to find the unique launch site names
- Returns 4 unique launch sites in database
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40

```
%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'

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

- The first five records with launch sites beginning with 'CCA' from database
- Used 'LIMIT' in the query to get the certain number of records

TOTAL PAYLOAD MASS

- This query is to get the total payload mass carried by boosters launched by NASA (CRS)
- Returns 45,596 kg total

```
%%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

- Display average payload mass carried by booster version F9 v1.1
- Returns average 2928.4 kg

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

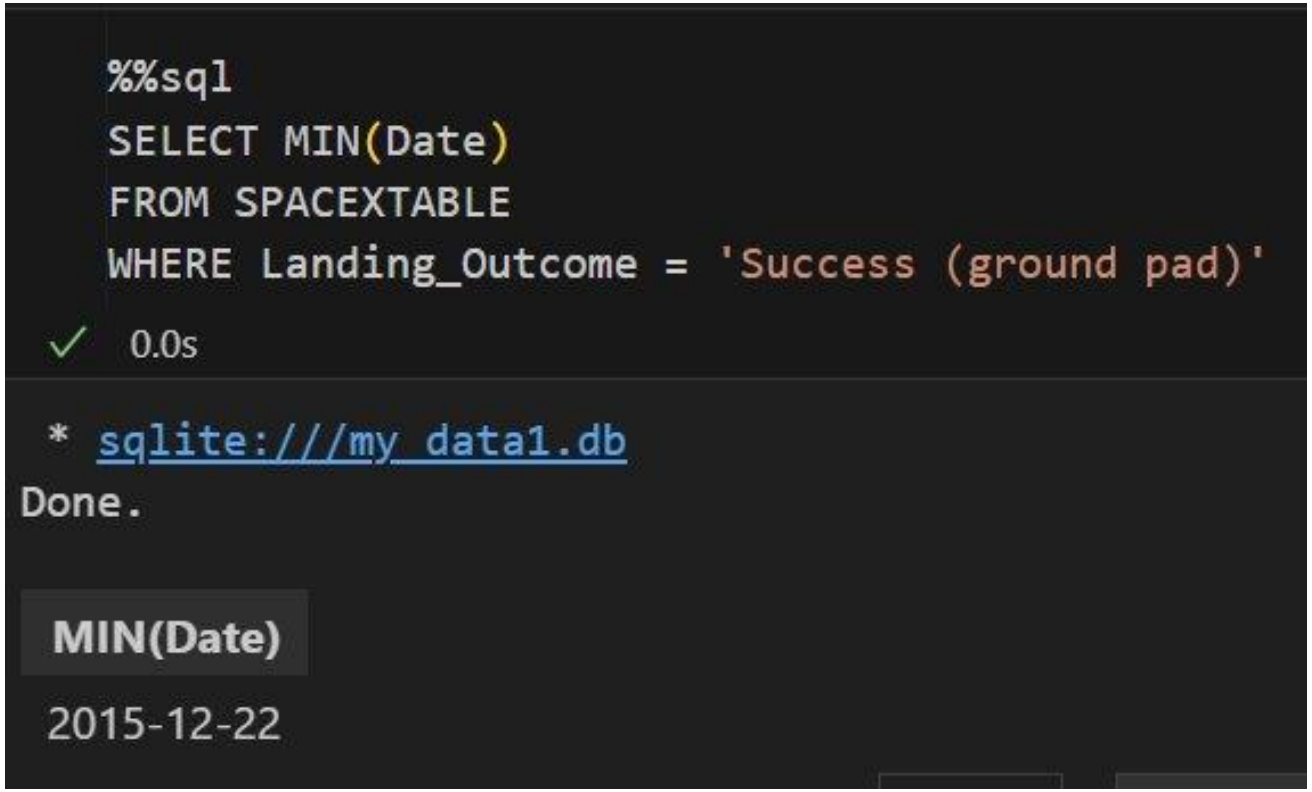
✓ 0.0s

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

AVG(PAYLOAD_MASS__KG_)
2928.4

FIRST SUCCESSFUL GROUND LANDING DATE

- Query to get the date when the first successful landing outcome in ground pad was achieved
- Used 'MIN()' considering that datetime data type is continuous value
- Returns 2015-12-22



SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

- Returns four unique booster versions that had successful drone ship landing and payload mass greater than 4000 but less than 6000

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

✓ 0.0s

```
 * 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

- Returns count of each mission outcome
- SpaceX successfully achieved the mission at around 99%.
  - Only 1 mission failure
  - 1 success with unclear payload status
- Landing failure does not mean mission failure

```
%%sql
SELECT Mission_Outcome, COUNT(Mission_Outcome) AS Count
FROM SPACEXTABLE
GROUP BY Mission_Outcome
```

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

Mission_Outcome	Count
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

BOOSTERS CARRIED MAXIMUM PAYLOAD

- Returns names of the booster versions which have carried the maximum payload mass
- All of them returned 15,600 kg
- Names of booster version are similar – F9 B5 B10xx.x

```
%%sql
SELECT Booster_Version, PAYLOAD_MASS__KG_
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = ( SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE )
```

✓ 0.0s

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

2015
LAUNCH RECORDS

- Returns
  - Month
  - Failure landing outcomes in drone ship
  - Booster versions
  - Launch site for the months in year 2015.
- 2 records returned

```
%%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'
```

✓ 0.0s

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

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 OUTCOME BETWEEN 2010-06-04 AND 2017-03-20

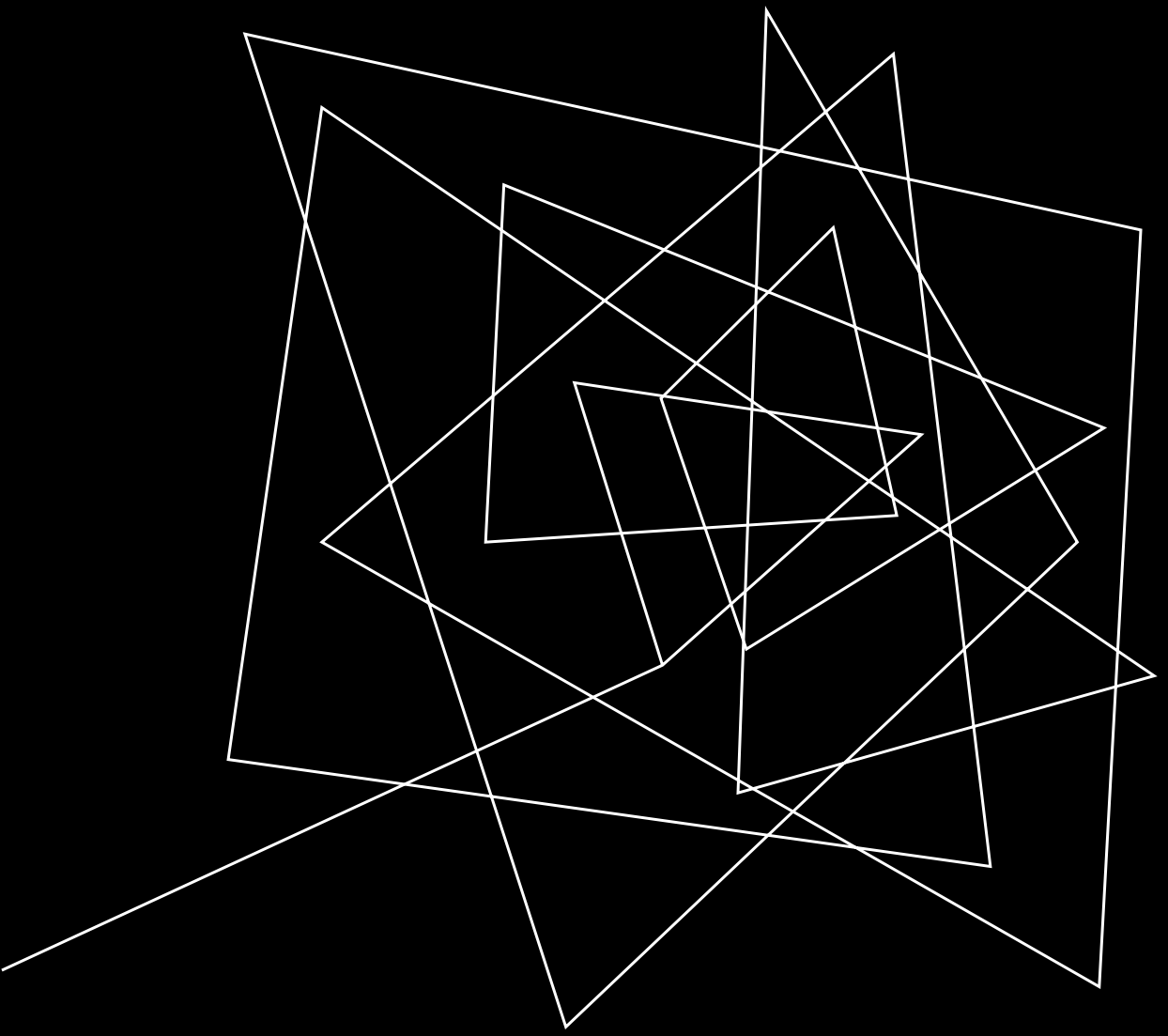
- Returns the count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order
- 10 launches had 'No attempt' for landing
- Total 8 success from 5 drone ship and 3 ground pad
- Total 7 failures from 5 drone ships and 2 parachutes

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

✓ 0.0s

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

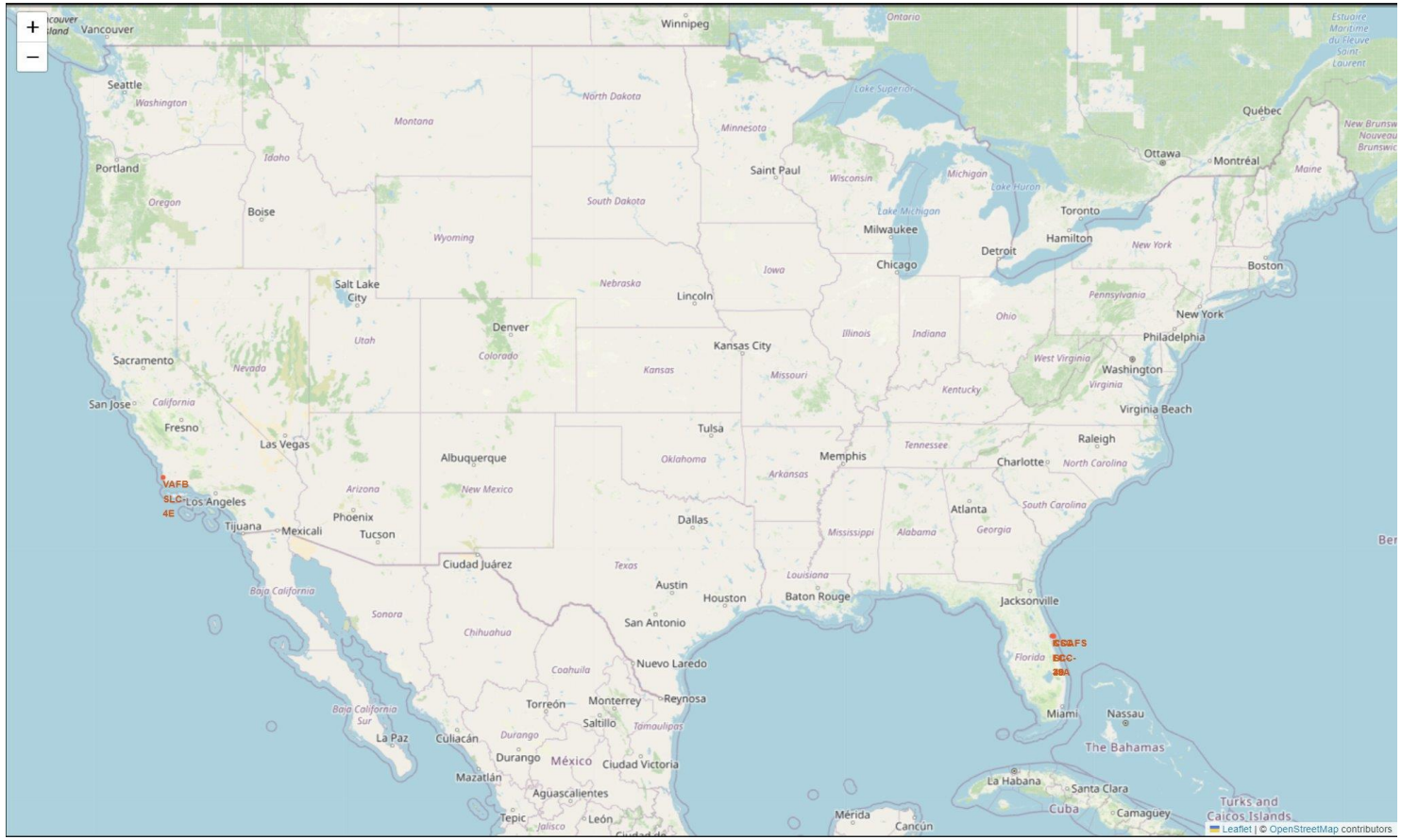
Landing_Outcome	Total
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1



SECTION 3

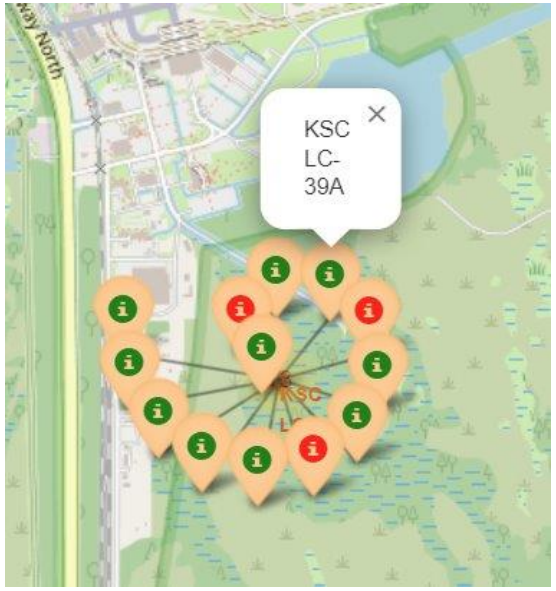
LAUNCH SITES PROXIMITIES ANALYSIS

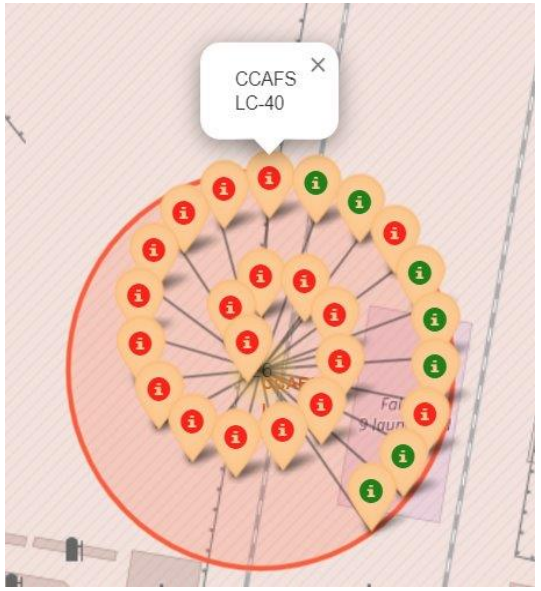
LAUNCH SITE LOCATIONS

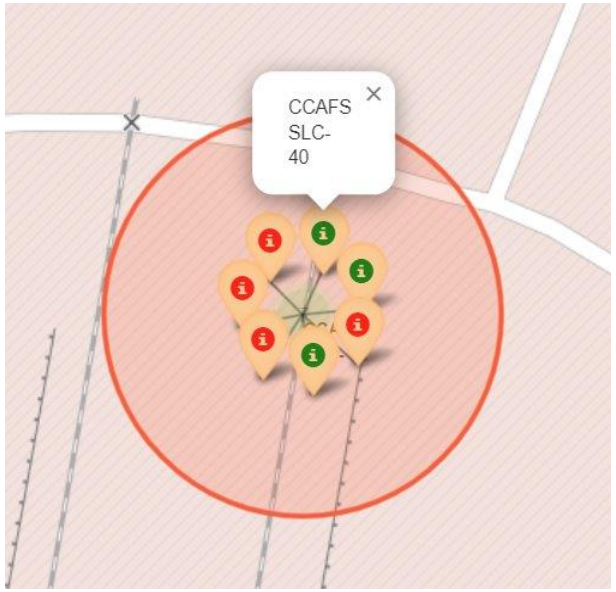


- All launch sites are close to the Equator line and the coast.
- However, they are away from the cities.
- 3 sites on the east coast are close to each other

LAUNCH SITE WITH COLOR-LABELED LAUNCH OUTCOMES

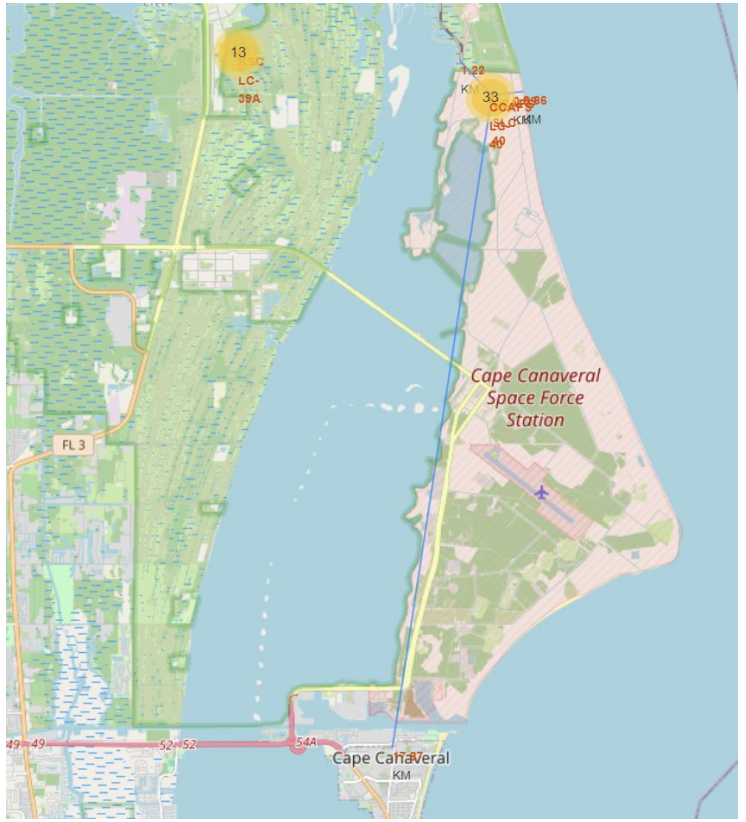




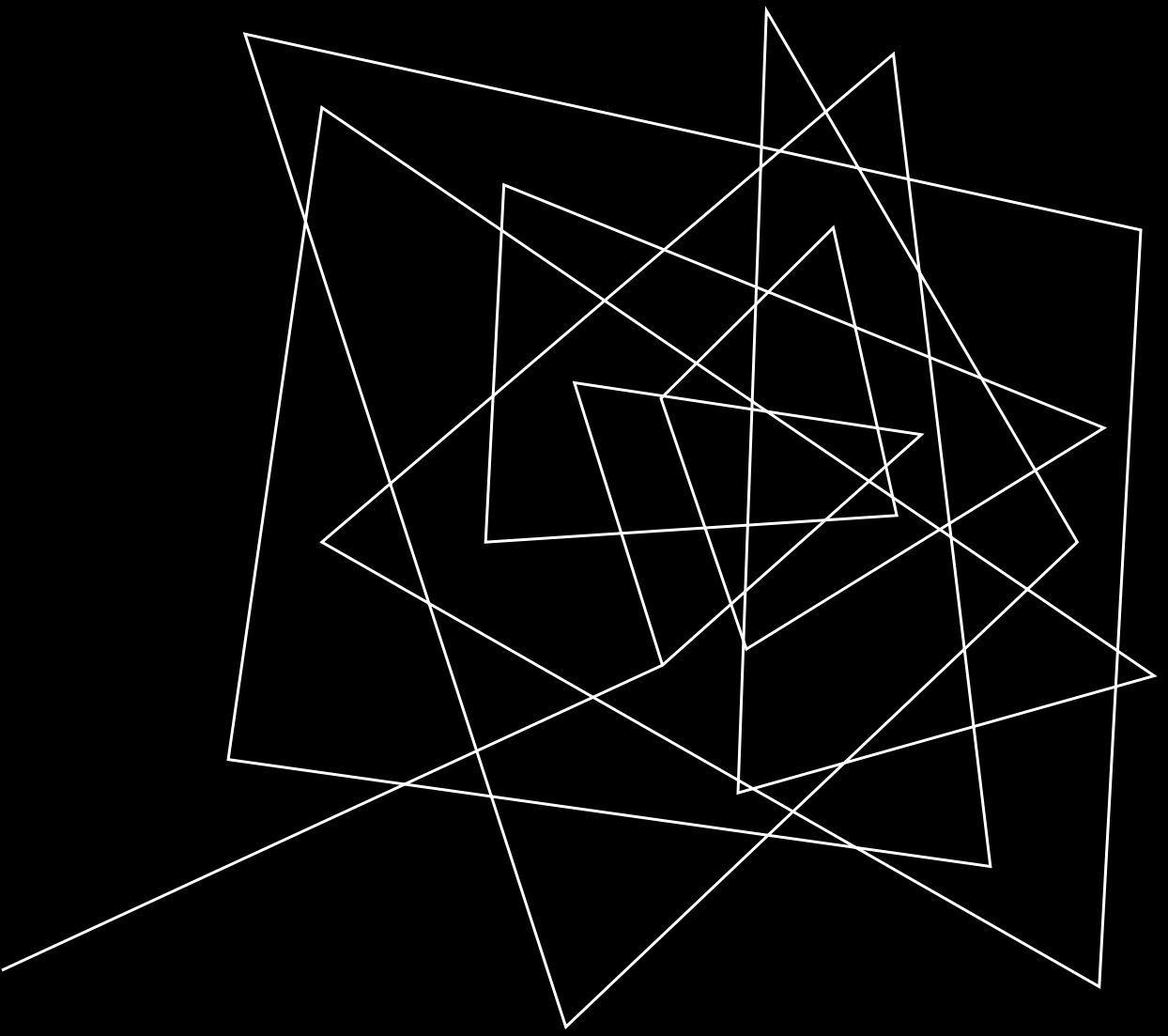


- Once you click cluster on Folium map, it displays the icons with colors for landing outcomes
- As you see images above, each site will show successful landing in green, and failure in red
- For example, VAFB SLC-4E shows 4 successful landings and 6 failed landings

LAUNCH SITE TO ITS PROXIMITIES



- Distance from CCAFS SLC-40 to key locations is as below:
  - 0.86 km to coast
  - 0.59 km to highway
  - 1.22 km to railway
  - 17.87 km to Cape Canaveral city
- It would be close to the coast but further away from the city due to the safety concerns.
- It could be close to highway and railway for transportation.

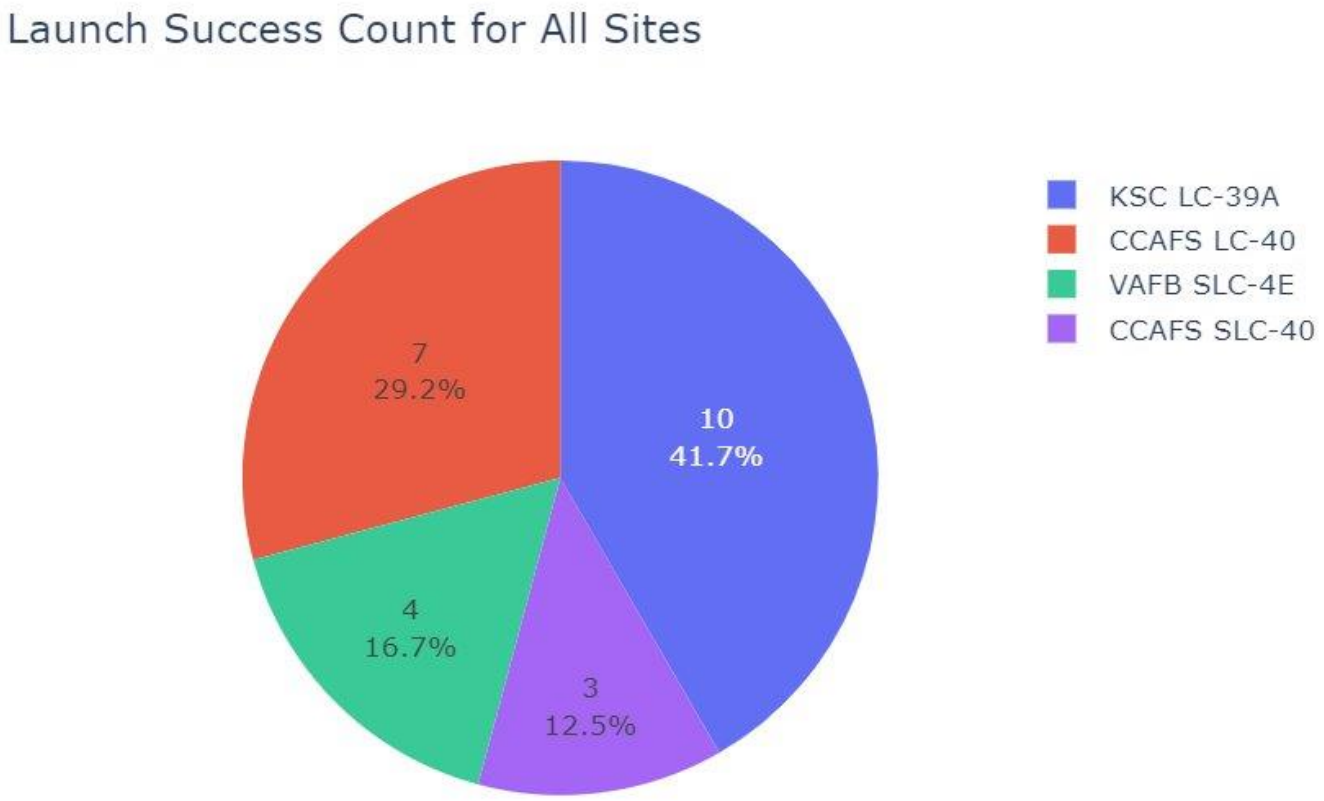


SECTION 4

BUILD A DASHBOARD WITH PLOTLY DASH

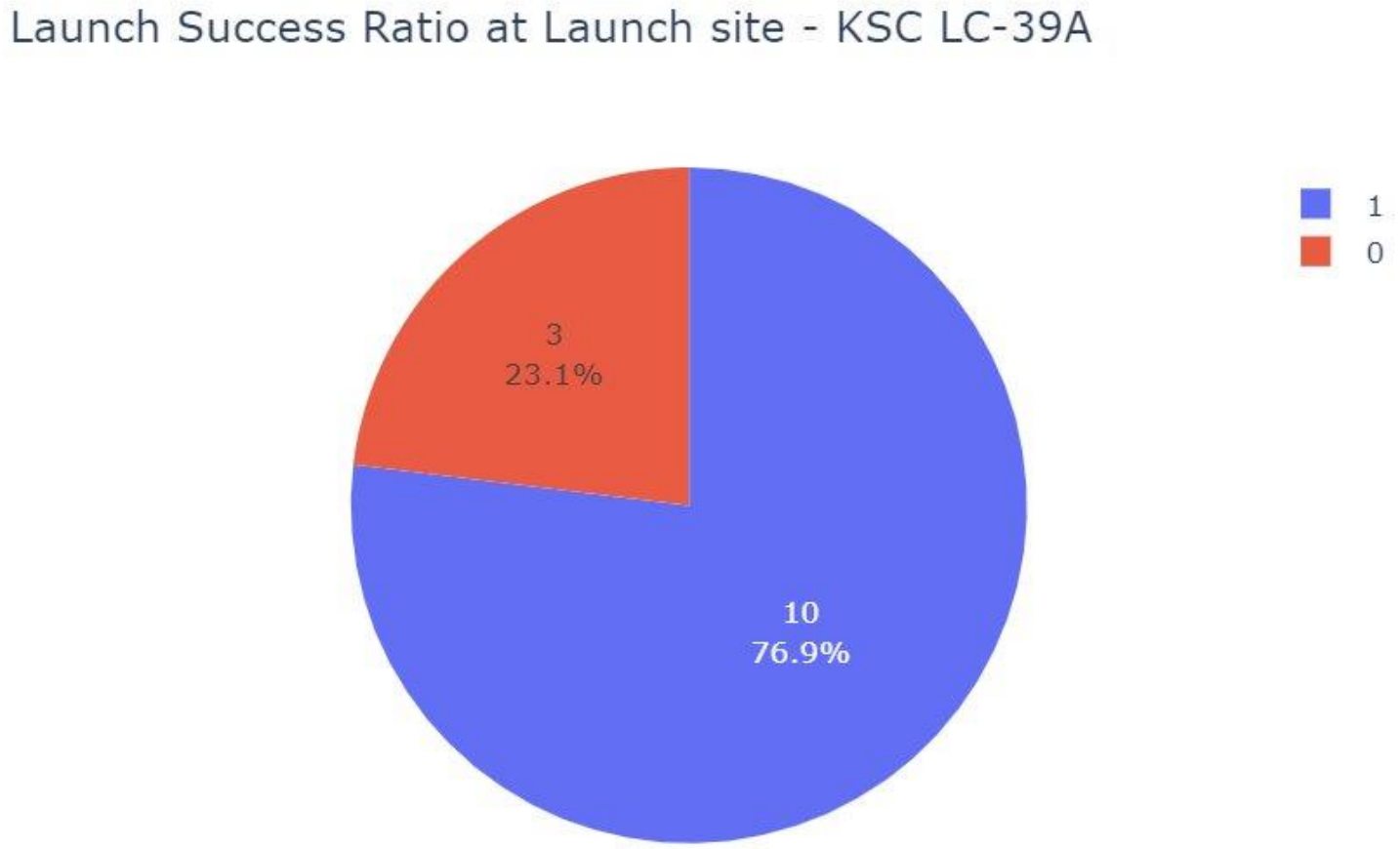
LAUNCH SUCCESS COUNT FOR ALL SITES

- It shows the proportion of launch success at all sites.
- KSC LC-39A has the most success in launching at 41.7% with 10 successes.
- In the meantime, CCAFS SLC-40 has the least success at 12.5% with only 3 succeses



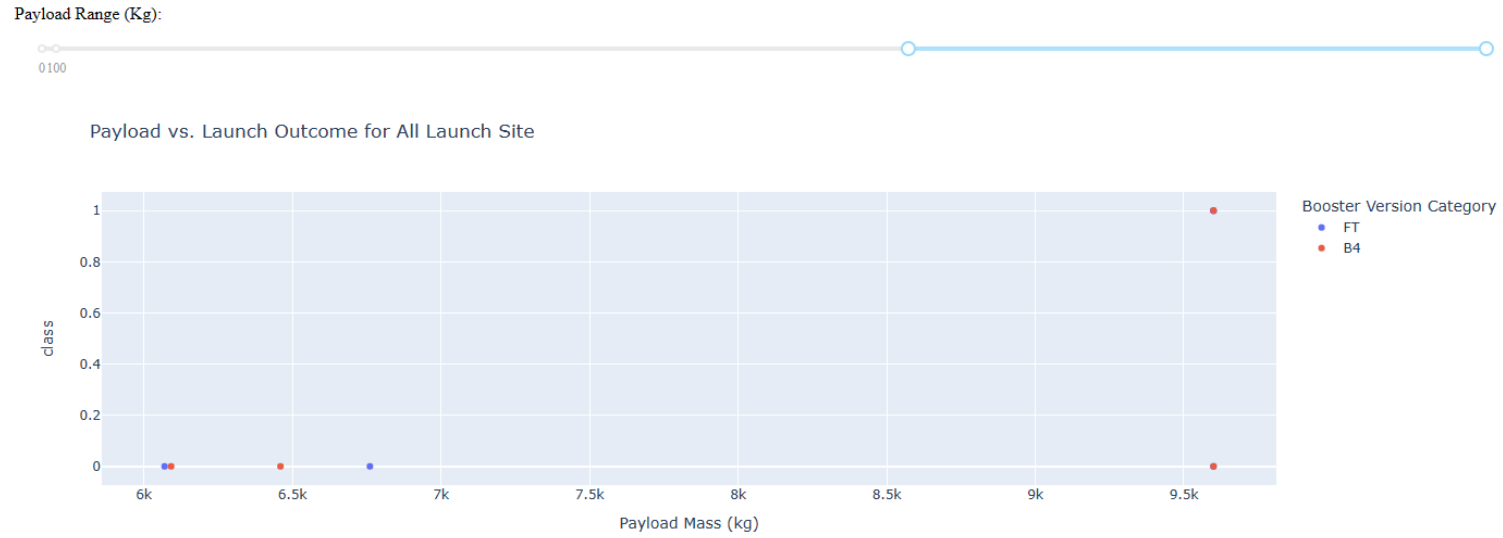
LAUNCH SITE WITH HIGHEST LAUNCH SUCCESS

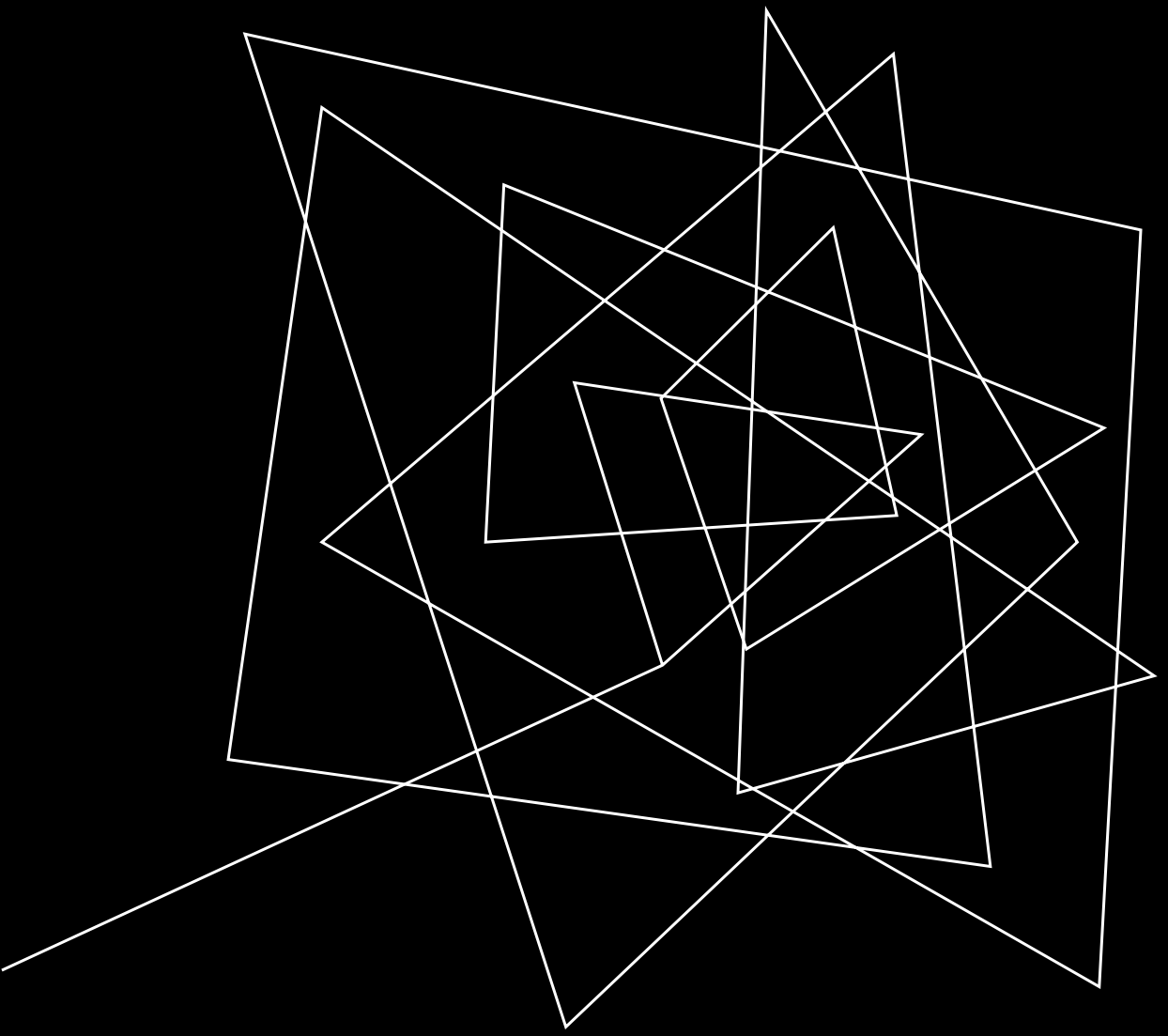
- Launch site KSC LC-39A has the highest number of success when comparing with other sites with 10 successful landings and 3 failed landings.
- About 77% of launches from this site successfully landed.



PAYLOAD VS. LAUNCH OUTCOME

- In Dashboard, payload range selector ranges from 0 to 10,000 kg only while actual data has maximum payload mass of 15,600 kg.
- Booster Version Category 'FT' has the most success.
- Booster Version Category 'B4' has success with the highest payload mass.
- Most of the success occurred below 6,000kg payload mass.
- There are not many launches with payload over 6,000 kg.



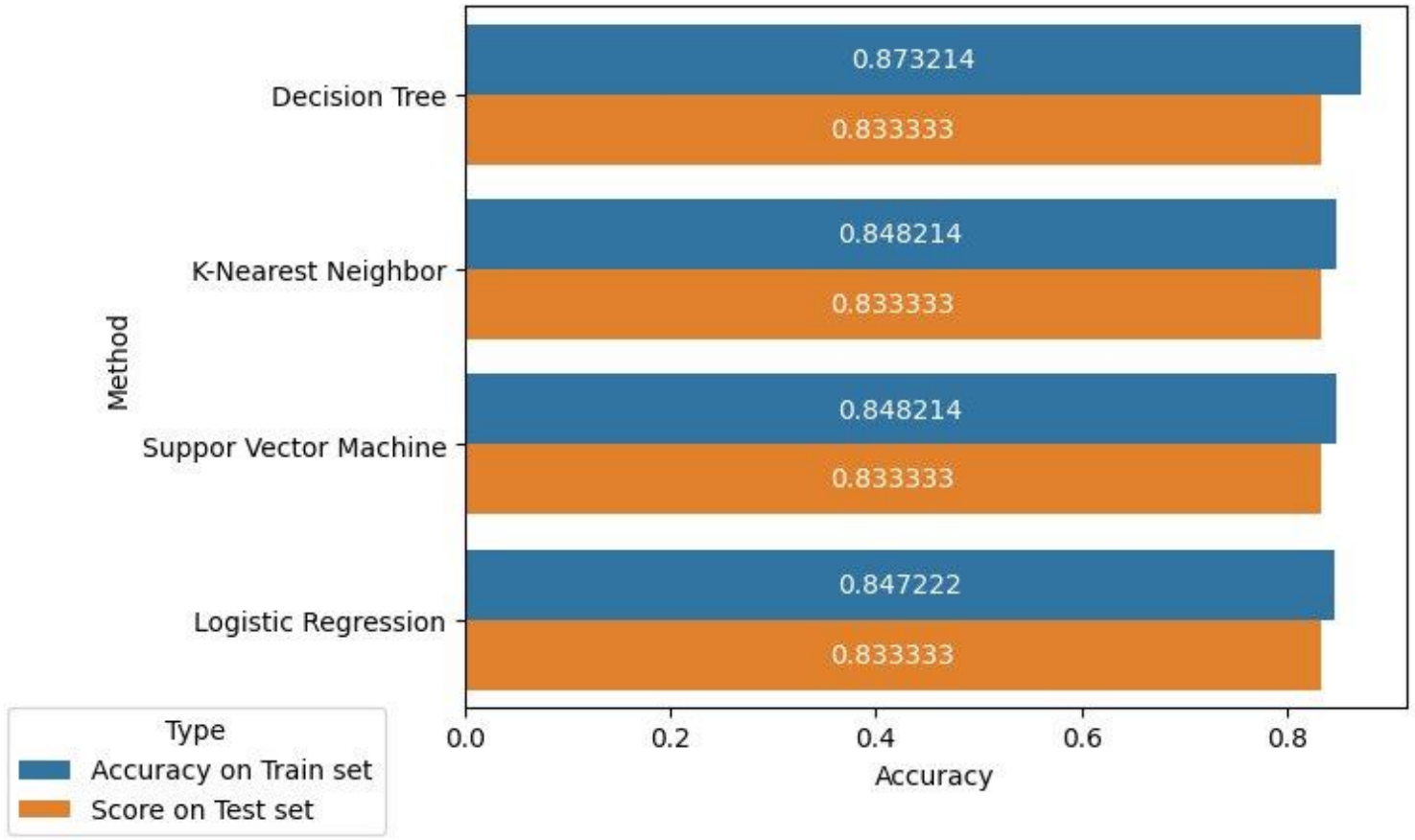


SECTION 5

PREDICTIVE ANALYSIS (CLASSIFICATION)

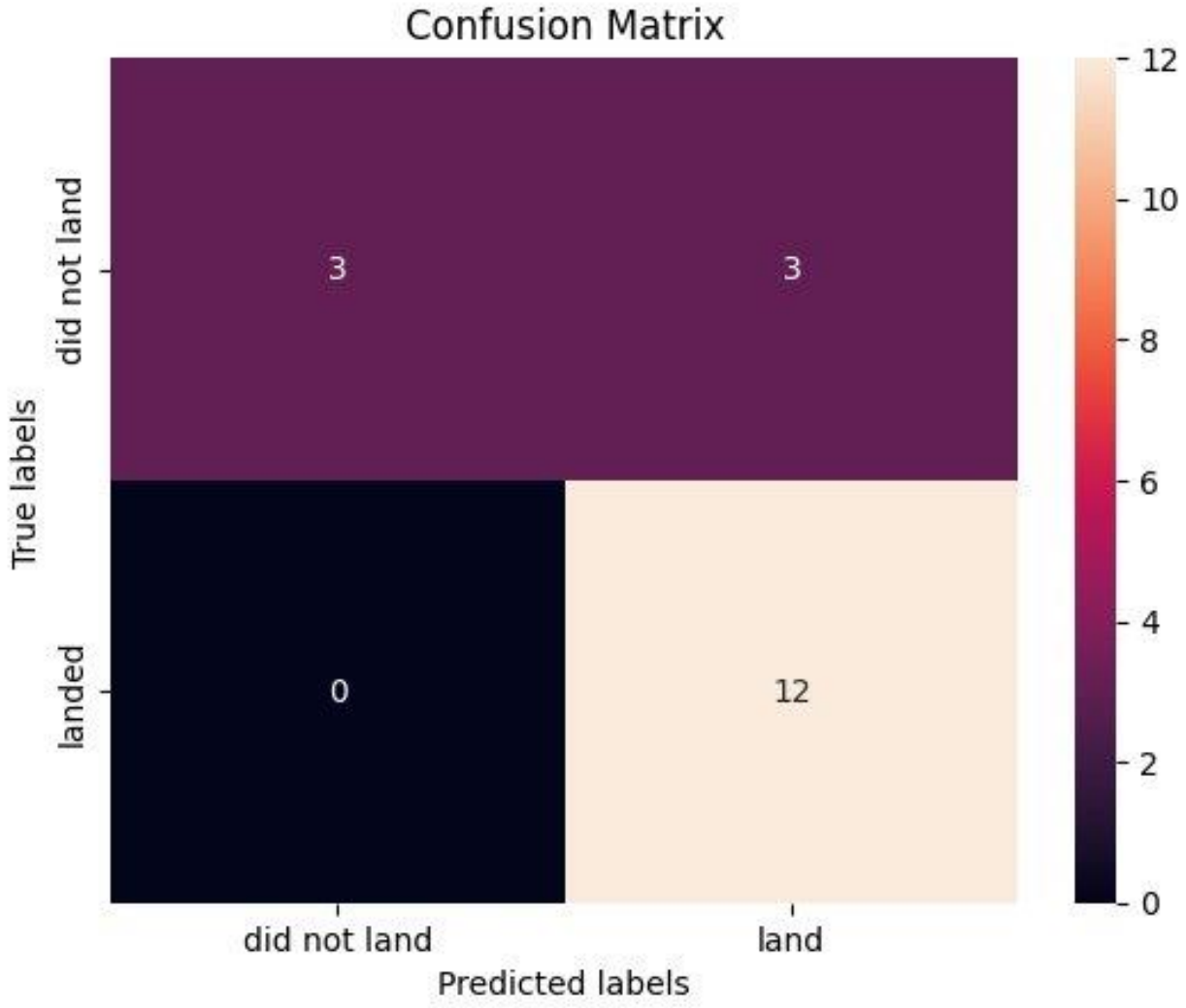
CLASSIFICATION ACCURACY

- For training data set, Decision Tree resulted in the highest score.
- However, with test data set, all models resulted in the same accuracy at 83.33%.
- Size of test dataset is small at only 18. It may not represent the population properly.
- It is likely that larger dataset is required to ensure the model is robust.



CONFUSION MATRIX

- All ML models generated the same confusion matrix since all resulted in the same accuracy score on the test dataset.
- The models predicted all 12 landings correctly as true labeled landing without failure.
- From 6 unsuccessful landings in True labels, model predicted 3 unsuccessful landings properly and 3 successful landings as false positives.
- Model requires adjustment to decrease false positives in a way that increases overall score.



CONCLUSIONS

Back to Problems:

- Space Y that would like to compete with SpaceX founded by Billionaire industrialist Allon Musk.
- Train a machine learning model and use public information to predict if SpaceX will reuse the first stage.
- If we can determine if the first stage will land, we can determine the cost of a launch to bid against SpaceX for a rocket launch.

Conclusions:

- Based on the available data, machine learning model is generated with an accuracy of 83.33%
- Space Y can use the model to predict the success of the first stage landing at reasonable accuracy
- It is recommended that we train machine learning models with a larger set of data to improve accuracy

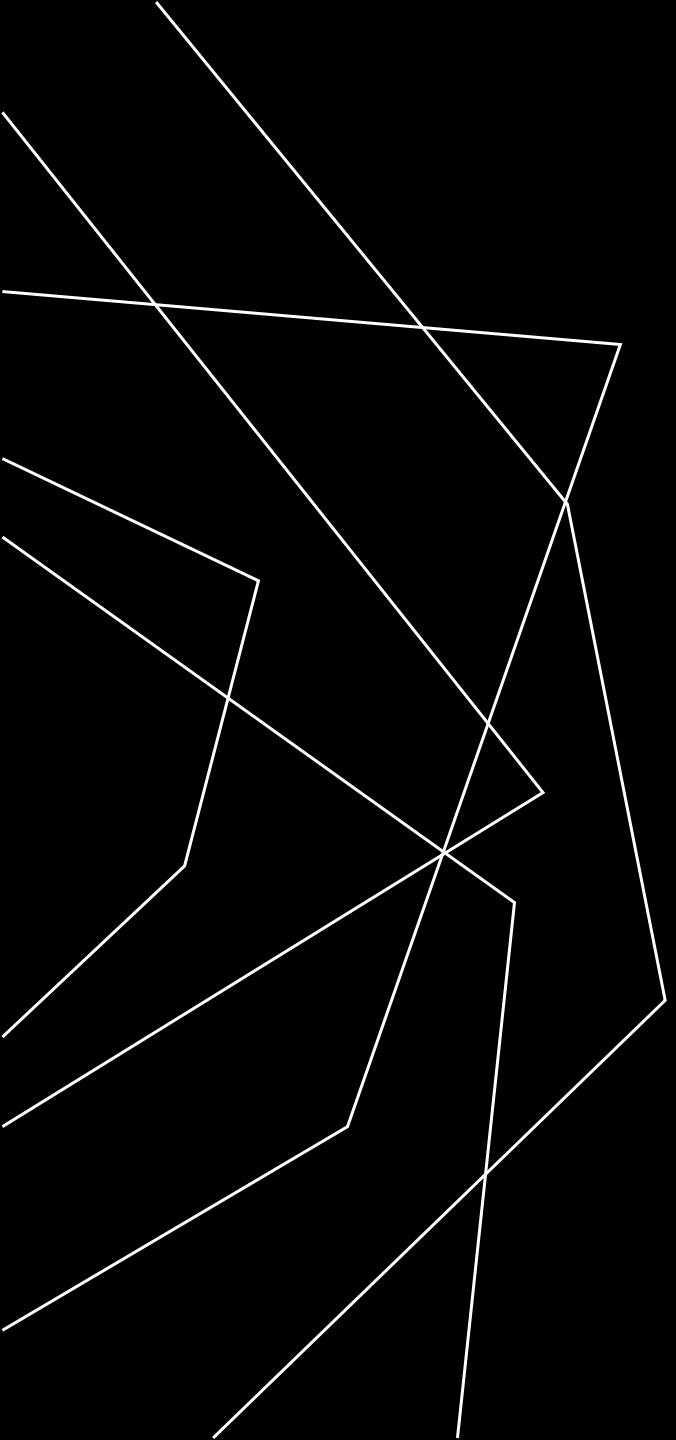
APPENDIX

- Github: [ibm-ds-course/10_Applied_DS_Capstone at main · ylhoony/ibm-ds-course (github.com)]
  - PDF Presentation file: [ibm-ds-course/10_Applied_DS_Capstone/Data_Science_Capstone_Project.pdf at main · ylhoony/ibm-ds-course (github.com)]
- Plotly Dash: Pie chart upate_traces() - [Pie traces in Python (plotly.com)]
- Seaboarn: Move Legend - **[seaborn.move_legend — seaborn 0.13.2 documentation (pydata.org)]**
- Distance between launch sites

```
# Calculate distance between launch sites
import itertools

launch_site_list = list(launch_sites_df['Launch Site'])
combinations = list(itertools.combinations(launch_site_list, 2))

for ls1, ls2 in combinations:
    ls1_coord = tuple(launch_sites_df[launch_sites_df['Launch Site']==ls1].values.flatten())
    ls2_coord = tuple(launch_sites_df[launch_sites_df['Launch Site']==ls2].values.flatten())

    distance = calculate_distance(ls1_coord[1], ls1_coord[2], ls2_coord[1], ls2_coord[2])
    print(f"Distance between {ls1_coord[0]} and {ls2_coord[0]} is {distance:.2f} KM.")
```

✓ 0.0s

```
Distance between CCAFS LC-40 and CCAFS SLC-40 is 0.11 KM.
Distance between CCAFS LC-40 and KSC LC-39A is 6.90 KM.
Distance between CCAFS LC-40 and VAFB SLC-4E is 3827.04 KM.
Distance between CCAFS SLC-40 and KSC LC-39A is 6.94 KM.
Distance between CCAFS SLC-40 and VAFB SLC-4E is 3827.06 KM.
Distance between KSC LC-39A and VAFB SLC-4E is 3820.25 KM.
```



THANK YOU!