WINNING SPACE RACE

WITH DATA SCIENCE

METEHAN BATI

OUTLINE

EXECUTIVE SUMMARY

- This report outlines the analysis of SpaceX data initially stored in a JSON file, which was then imported into a pandas DataFrame using Python. The dataset underwent rigorous cleaning and pre-processing to handle missing values, outliers, and inconsistencies.
- After the data wrangling phase, exploratory data analysis (EDA) was carried out using a combination of visualizations and SQL queries to extract valuable insights.
- The analysis culminated in the development of interactive visual analytics tools, leveraging Folium and Plotly Dash, to facilitate deeper exploration and effective presentation of the findings.

ENGAGING THE AUDIENCE

The commercial space industry, led by companies like SpaceX, is undergoing a transformative shift with innovations like reusable rocket technology. Our project aims to explore the feasibility of competing with SpaceX by analyzing publicly available data and leveraging data science techniques.

Problems you want to find answers

Can we predict SpaceX's success in recovering and reusing the first stage of its Falcon 9 rockets?

What factors influence SpaceX's decision to reuse or discard the first stage?

How does SpaceX's cost-saving strategy impact rocket launch pricing?

What challenges and opportunities exist for a new entrant like Space Y in competing with SpaceX?

METHODOLOGY

SECTION 1

METHODOLOGY

- 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 COLLECTIONOVERVIEW

The data collection process involved a hybrid approach, combining API requests from Space X's public API with web scraping from a table within Space X's Wikipedia entry. The API provided information such as FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, and various other parameters. Meanwhile, the web scraping extracted data such as Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, and Time. Subsequent slides will illustrate the flowcharts depicting the distinct processes for API data collection and web scraping.

DATA COLLECTION – SPACEX API

- Import Libraries and Define Auxiliary Functions
- Request and parse the SpaceX launch data using the GET request
- Filter the dataframe to only include Falcon 9 launches
- Dealing with Missing Values

https://github.com/metehanbati/IBM_Final_Project-/blob/main/jupyter-labs-webscraping.ipynb

DATA COLLECTION - SCRAPING

Request the Falcon9 Launch Wiki page from its URL

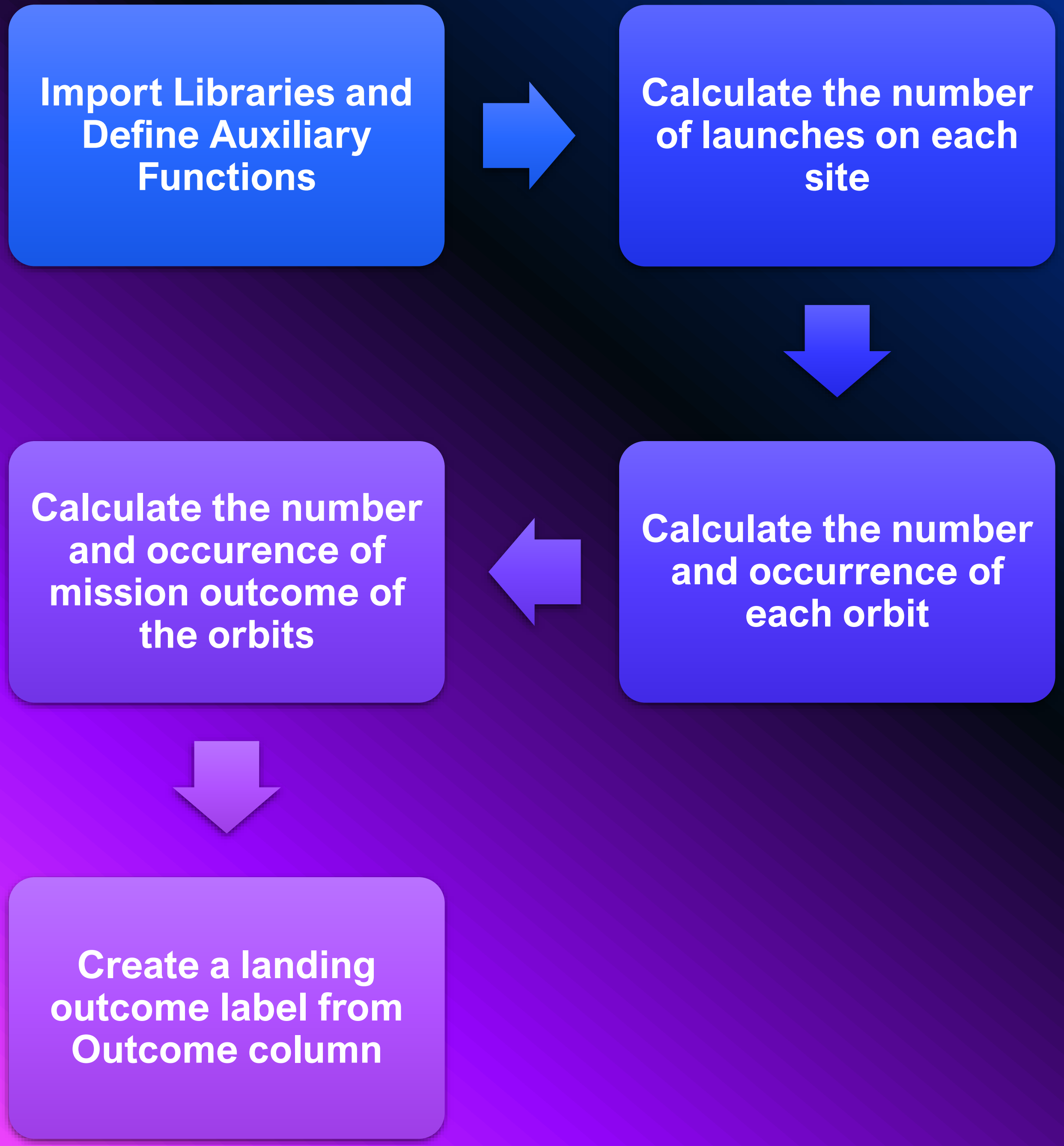
↓

Extract all column/variable names from the HTML table header

↓

Create a data frame by parsing the launch HTML tables

https://github.com/metehanbati/IBM_Final_Project-/blob/main/jupyter-labs-webscraping.ipynb



DATA COLLECTION – SCRAPINGDATA WRANGLING

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

EDA WITH DATA VISUALIZATION

- I utilized scatter charts, line charts, and bar charts in my analysis:
- Scatter chart: This type of chart is ideal for exploring relationships between variables. By plotting data points, it enabled me to identify correlations and trends effectively.
- Line chart: Line charts are particularly useful for visualizing trends and changes over time. They allowed me to observe patterns and fluctuations in the data, providing valuable insights into temporal dynamics.
- Bar chart: I employed bar charts to compare categorical or grouped data. This visual representation facilitated the comparison of values across different categories, enhancing the understanding of relative differences and distributions.
- https://github.com/metehanbati/IBM_Final_Project-/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

FINAL TIPS & TAKEAWAYS

- Consistent rehearsal
  - Strengthen your familiarity
- Refine delivery style
  - Pacing, tone, and emphasis
- Timing and transitions
  - Aim for seamless, professional delivery
- Practice audience
  - Enlist colleagues to listen & provide feedback

Seek feedback

Reflect on performance

Explore new techniques

Set personal goals

Iterate and adapt

BUILD AN INTERACTIVE MAP WITH FOLIUM

- **Markers**: Markers were added to pinpoint specific locations on the map, such as significant landmarks, cities, or data collection points. These markers provide visual cues and help users identify and navigate to specific locations of interest easily.
- **Circles**: Circles were used to represent areas of interest or influence, such as the radius around a particular location. These circles may denote regions of coverage, proximity, or impact and provide a visual representation of spatial relationships within the data.
- **Lines**: Lines were added to connect points of interest or illustrate routes, pathways, or connections between different locations. These lines help users visualize spatial relationships and understand the connectivity between various geographical features or data points.
- https://github.com/metehanbati/IBM_Final_Project-/blob/main/lab_jupyter_launch_site_location.jupyterlite.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 successrates.
- 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.

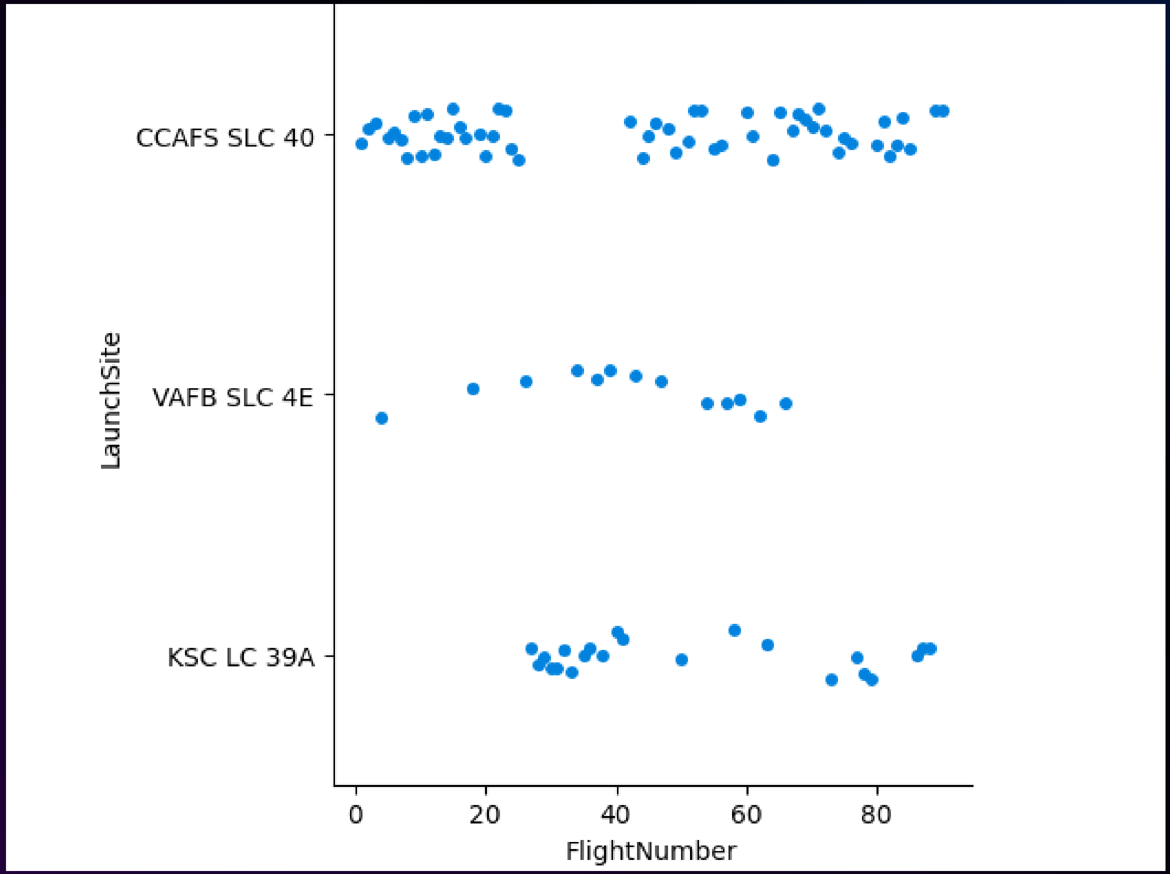
PREDICTIVE ANALYSIS (CLASSIFICATION)

- **Split label column 'Class' from dataset**
- **Fit and Transform features using Standart Scaler**
- **Train_test_split Data**
- GridSearchCV to find optimal parameters
- **Use GridSearchCV on LogReg, SVM, Decision Tree and KNN models**
- **Score Models on split test set**
- Confusion Matrix for all models
- Barplot to compare scores fo models

RESULTS

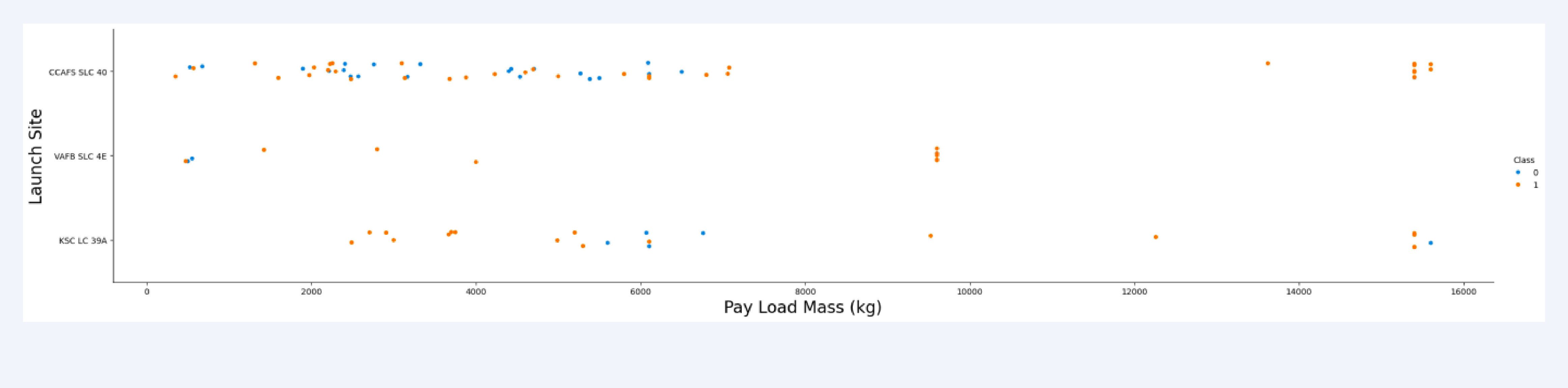
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



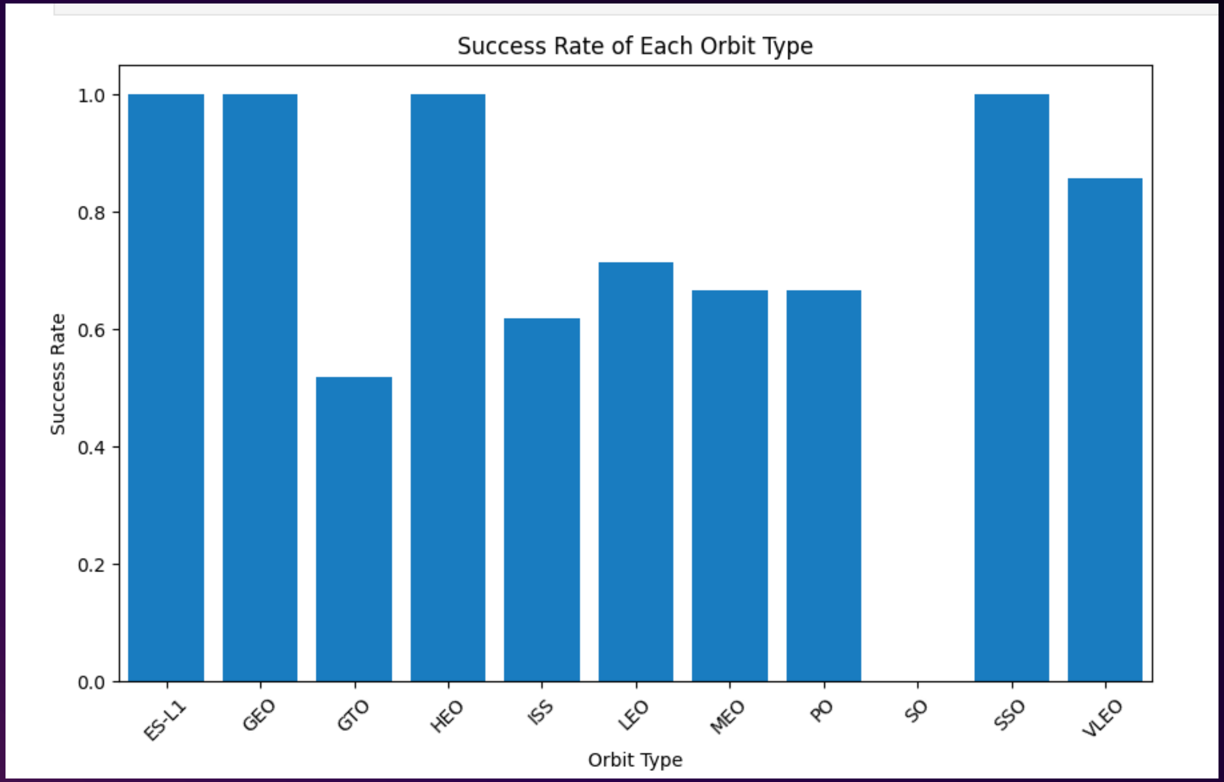
PAYLOAD VS. LAUNCH SITE

The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



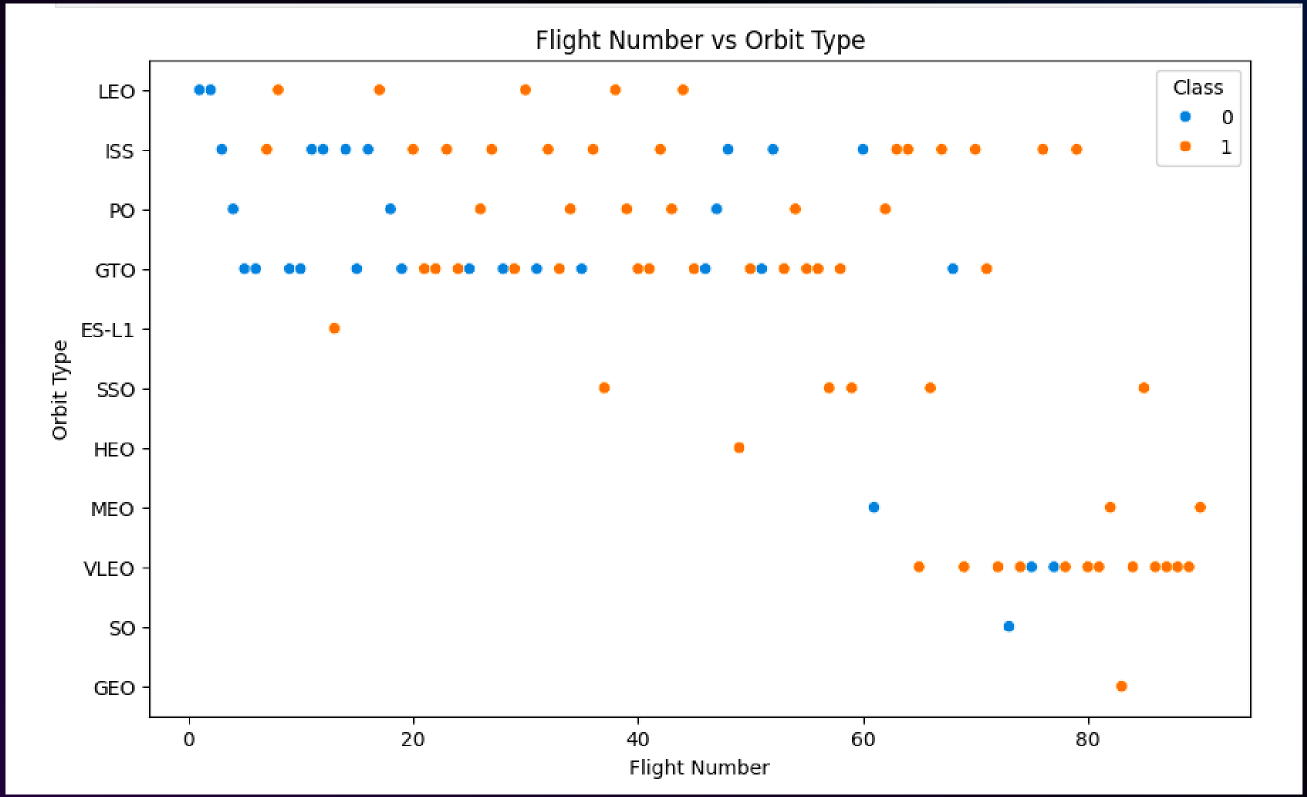
SUCCESS RATE VS. ORBIT TYPE

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO has the most success rate.



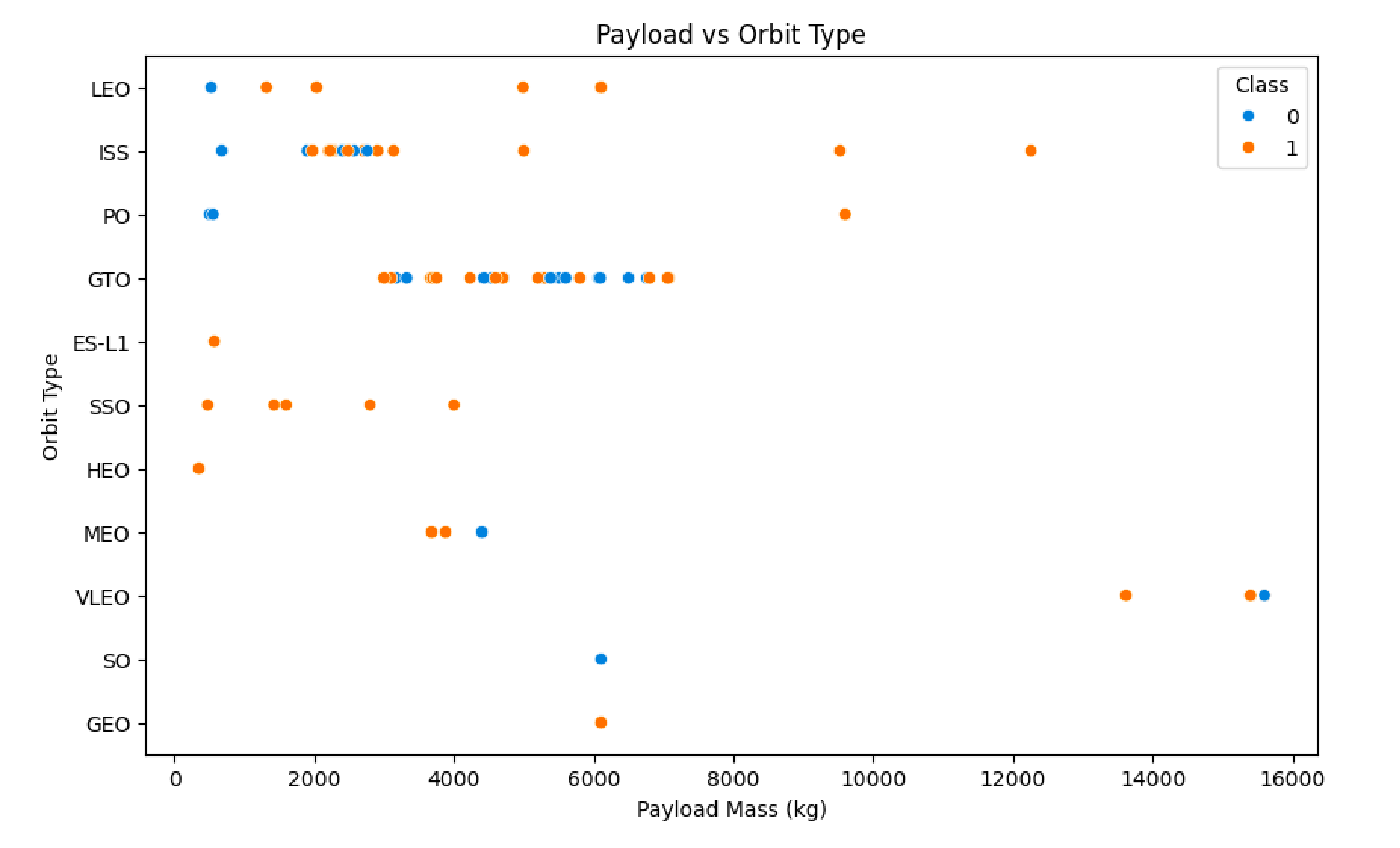
FLIGHT NUMBER VS. ORBIT TYPE

The plot shows the Flight Number vs. Orbit Type. We observed that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



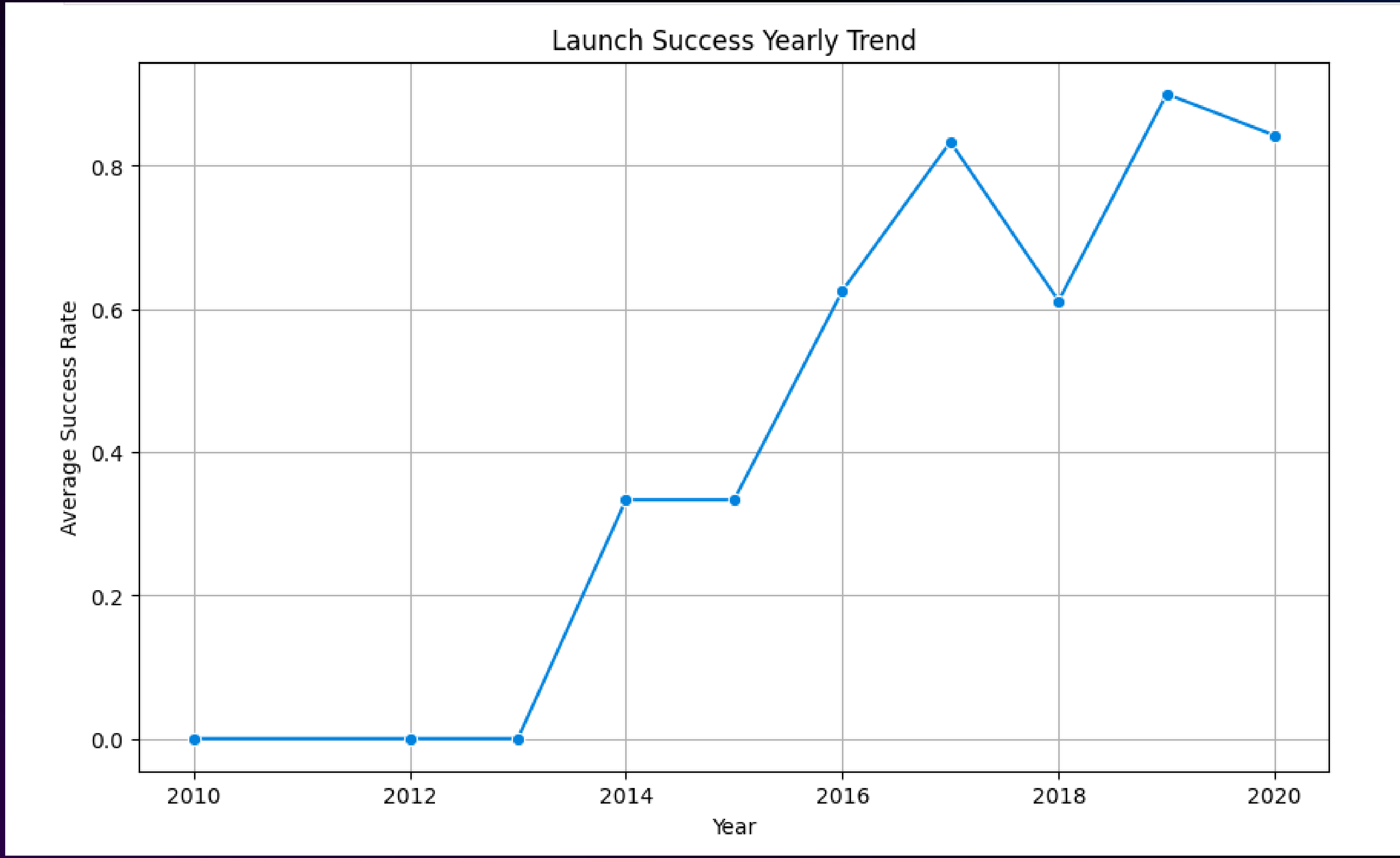
PAYLOAD VS. ORBIT TYPE

- We can observe that with heavy payloads, the successful landing are more for PO, LEO, and ISS orbits.



LAUNCH SUCCESS YEARLY TREND

- From the plot, we can observe that success rate since 2013 kept on increasing until 2020.



ALL LAUNCH SITE NAMES

We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

```
%%sql
SELECT DISTINCT Launch_Site
FROM SPACEXTBL;
```

```
 * 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'

- We used the query above to display 5 records where launch sites begin with ‘CCA’.

```
%%sql
SELECT Launch_Site
FROM SPACEXTBL
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5;
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

TOTAL PAYLOAD MASS

We calculated the total payload carried by boosters from NASA as 99980 using the query below:

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) as TOTAL
FROM SPACEXTBL
WHERE Customer LIKE 'NASA%'
```

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

TOTAL
99980

AVERAGE PAYLOAD MASS BY F9 V1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2534.66666.

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS Average
FROM SPACEXTBL
WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

Average
2534.6666666666665

FIRST SUCCESSFUL GROUND LANDING DATE

- • We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015.

```
%%sql
SELECT MIN(DATE)
FROM SPACEXTBL
WHERE Landing_Outcome LIKE 'Success%'
```

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

MIN(DATE)
2015-12-22

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

- We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000.

```
%%sql
SELECT Booster_Version, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 v1.1	4535
F9 v1.1 B1011	4428
F9 v1.1 B1014	4159
F9 v1.1 B1016	4707
F9 FT B1020	5271
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1030	5600
F9 FT B1021.2	5300
F9 FT B1032.1	5300
F9 B4 B1040.1	4990
F9 FT B1031.2	5200
F9 B4 B1043.1	5000
F9 FT B1032.2	4230
F9 B4 B1040.2	5384
F9 B5 B1046.2	5800
F9 B5 B1047.2	5300
F9 B5B1054	4400
F9 B5 B1048.3	4850
F9 B5 B1051.2	4200
F9 B5B1060.1	4311
F9 B5 B1058.2	5500
F9 B5B1062.1	4311

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

- We used GROUP BY to filter MissionOutcome was a success or a failure.

```
%%sql
SELECT Mission_Outcome, COUNT(*) AS Outcome_Count
FROM SPACEXTBL
GROUP BY Mission_Outcome;
```

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

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

BOOSTERS CARRIED MAXIMUM PAYLOAD

- We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

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

```
 * 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

- We used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

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

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

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

RANK LANDING OUTCOMES BETWEEN 2010-06-04 AND 2017-03-20

- We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 and 2017- 03-20.
- We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

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

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

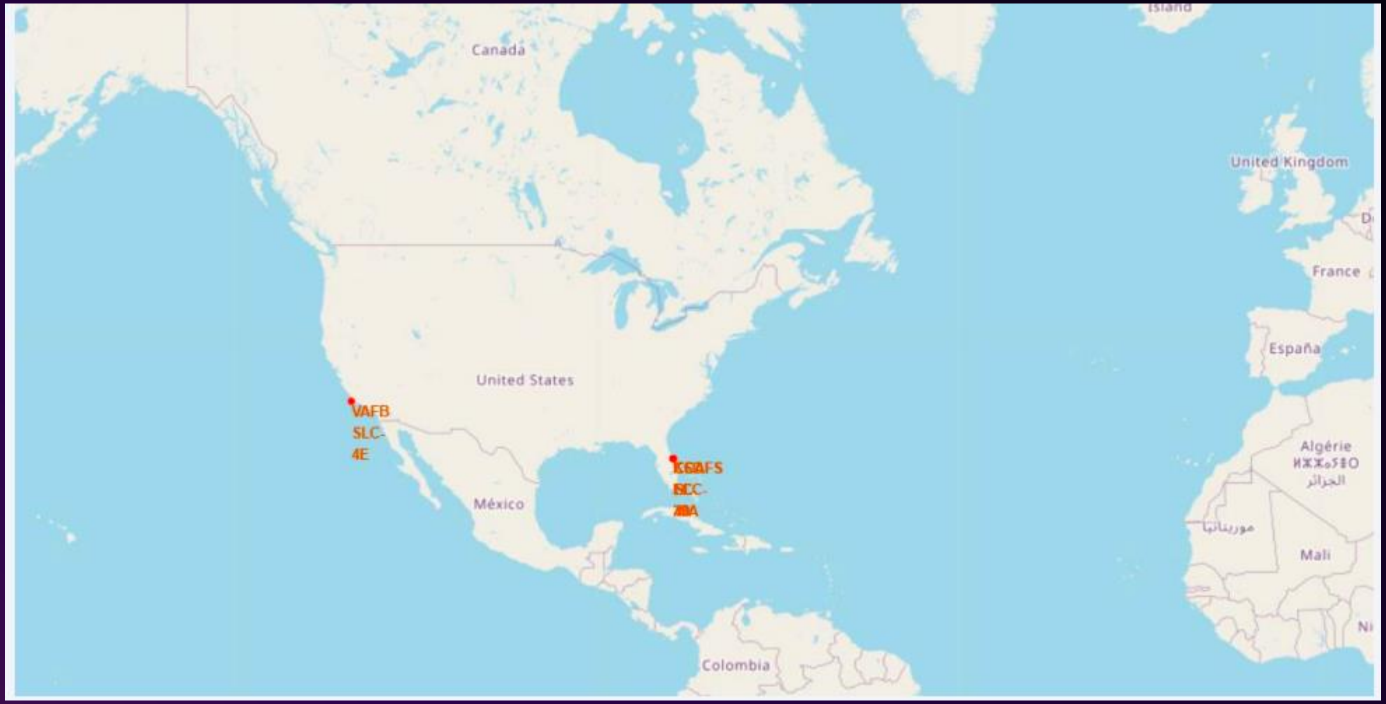
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

LAUNCH SITES PROXIMITIES ANALYSIS

SECTION 3

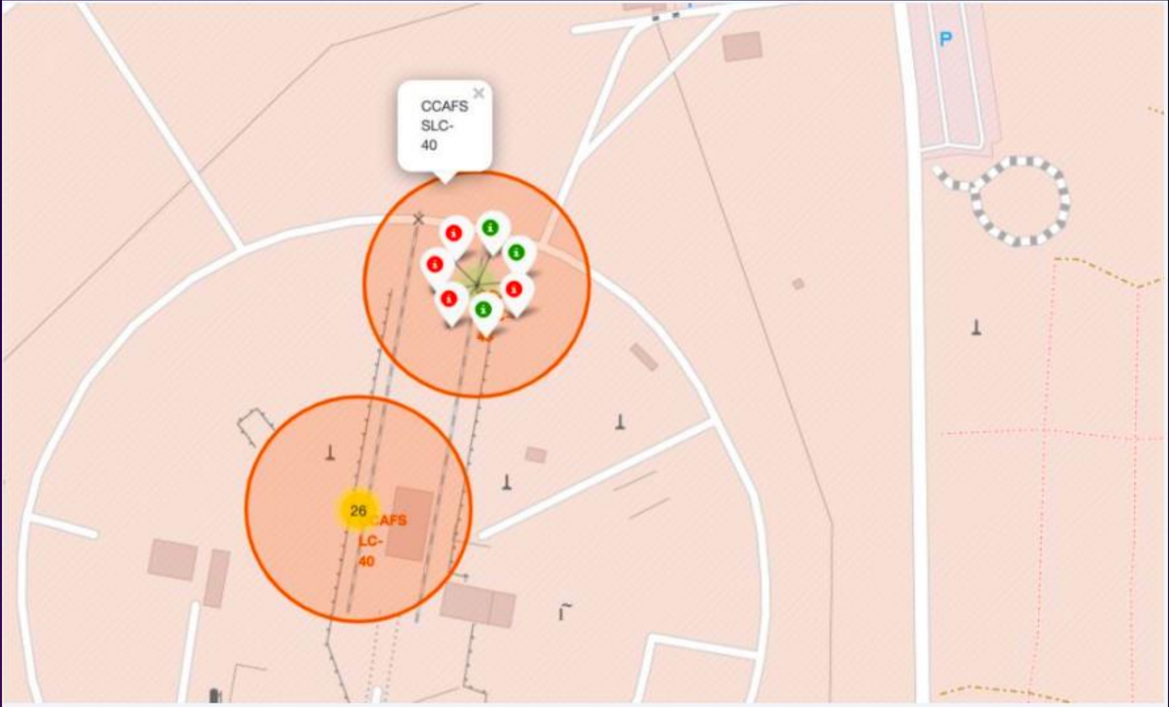
ALL LAUNCH SITES GLOBAL MAP MARKERS

- We can see that the SpaceX launch sites are in the United States of America Florida and California.



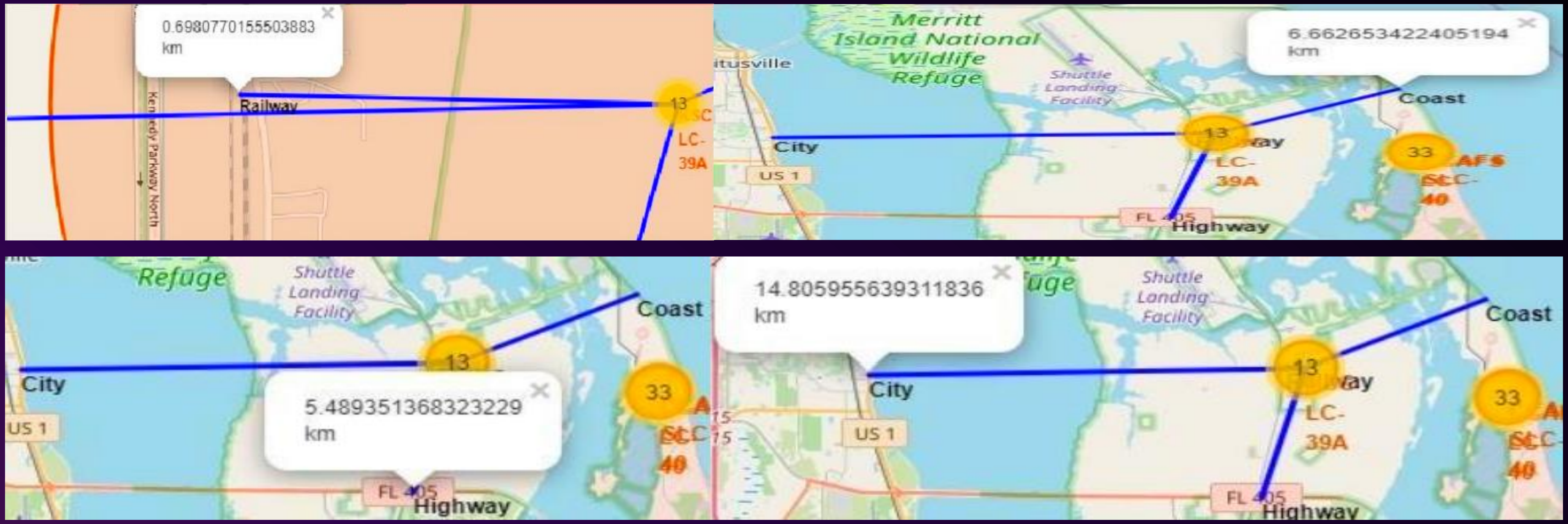
MARKERS SHOWING LAUNCH SITES WITH COLOR LABELS

- Green Marker shows successful launches and Red Marker shows Failures.



LAUNCH SITE DISTANCE TO LANDMARKS

- 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 rocketsfalling on densely populated areas

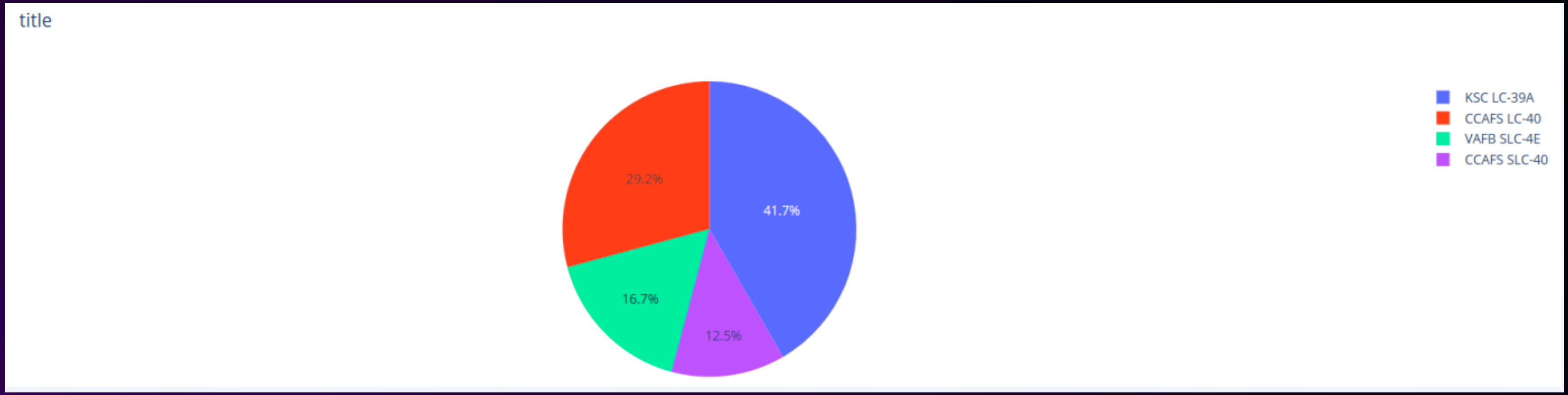


BUILD A DASHBOARD WITH PLOTLY DASH

SECTION 4

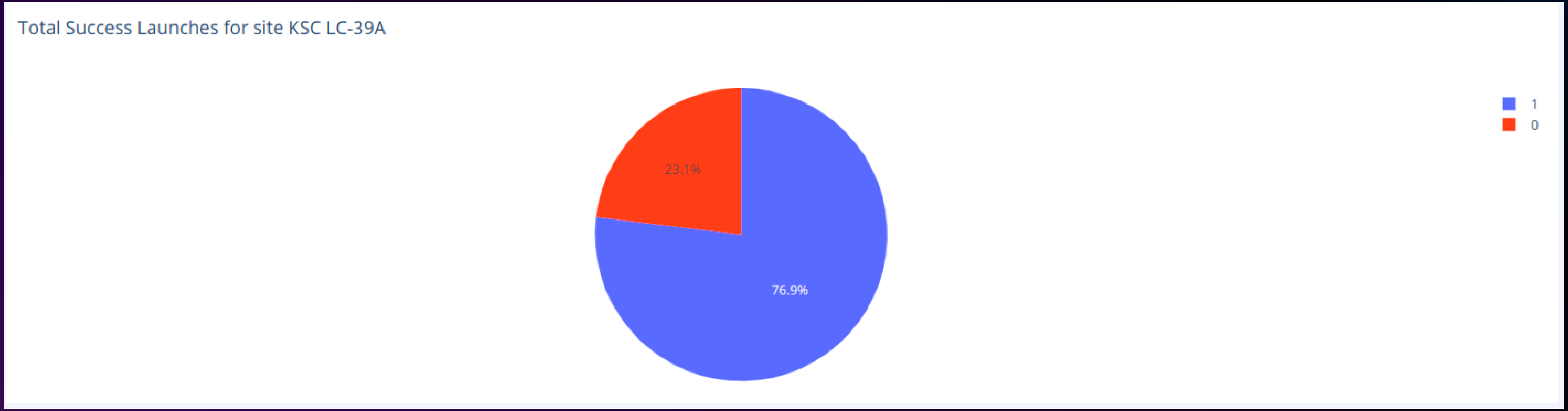
PIE CHART SHOWING THE SUCCESS PERCENTAGE ACHIEVED BY EACH LAUNCH SITE

- We can see that KSC LC-39A had the most successful launches from all the sites.



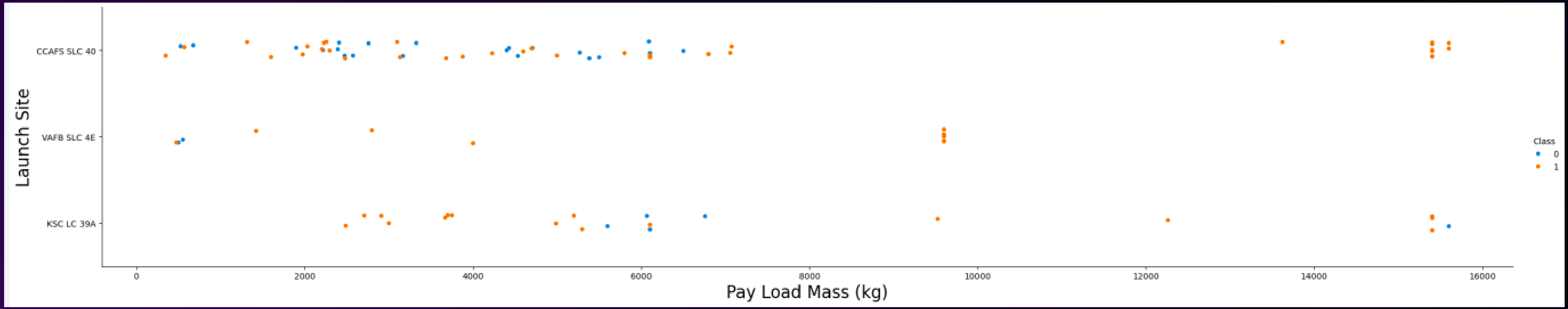
PIE CHART SHOWING THE LAUNCH SITE WITH THE HIGHEST LAUNCH SUCCESS RATIO

KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate.



SCATTER PLOT OF PAYLOAD VS LAUNCH OUTCOME FOR ALL SITES, WITH DIFFERENT PAYLOAD SELECTED IN THE RANGE SLIDER

- We can see the success rate for low weighted payloads is higher than the heavy weighted payloads.

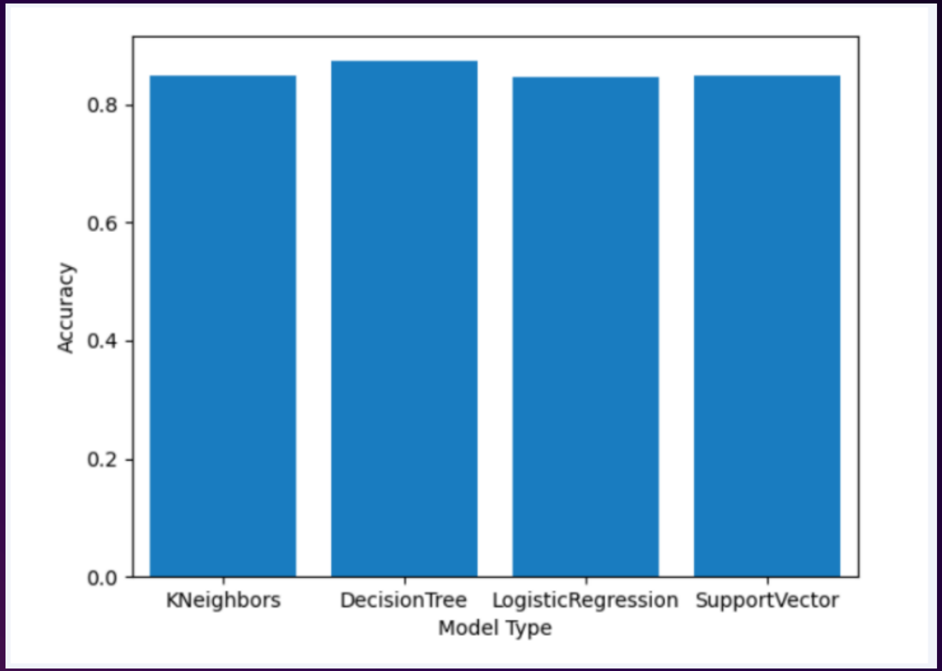


PREDICTIVE ANALYSIS (CLASSIFICATION

SECTION 5

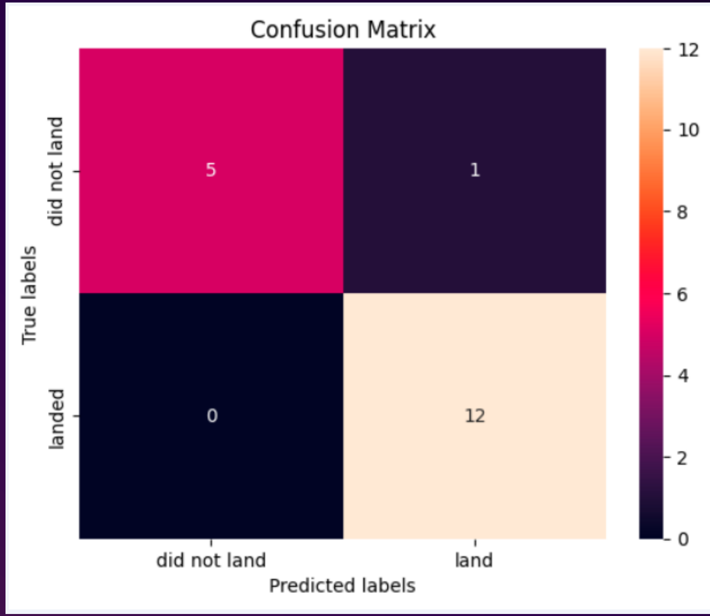
CLASSIFICATION ACCURACY

- From the right bar chart, we can see that the decision tree classifier is the model with the highest classification accuracy.



CONFUSION MATRIX

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives, for example, unsuccessful landing marked as successful landing by the classifier.



CONCLUSIONS

- Based on our analysis, several conclusions can be drawn:
- There appears to be a correlation between the number of flights conducted at a launch site and the success rate of launches at that site, with higher flight volumes correlating with greater success rates.
- The success rate of launches exhibited an upward trend from 2013 to 2020.
- Orbits such as ES-L1, GEO, HEO, SSO, and VLEO demonstrated the highest success rates.
- Among all launch sites, KSC LC-39A boasted the highest number of successful launches.
- Based on our findings, the Decision Tree Classifier emerged as the most suitable machine learning algorithm for this particular task.

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

Metehan Bati

Github.com/metehanbati