IBM Applied Data Science Capstone

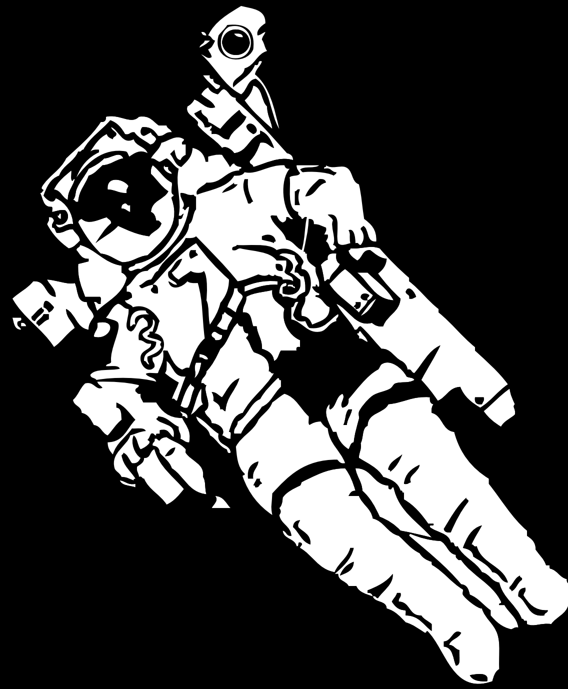
Klaudia Banasiewicz

24.03.2025

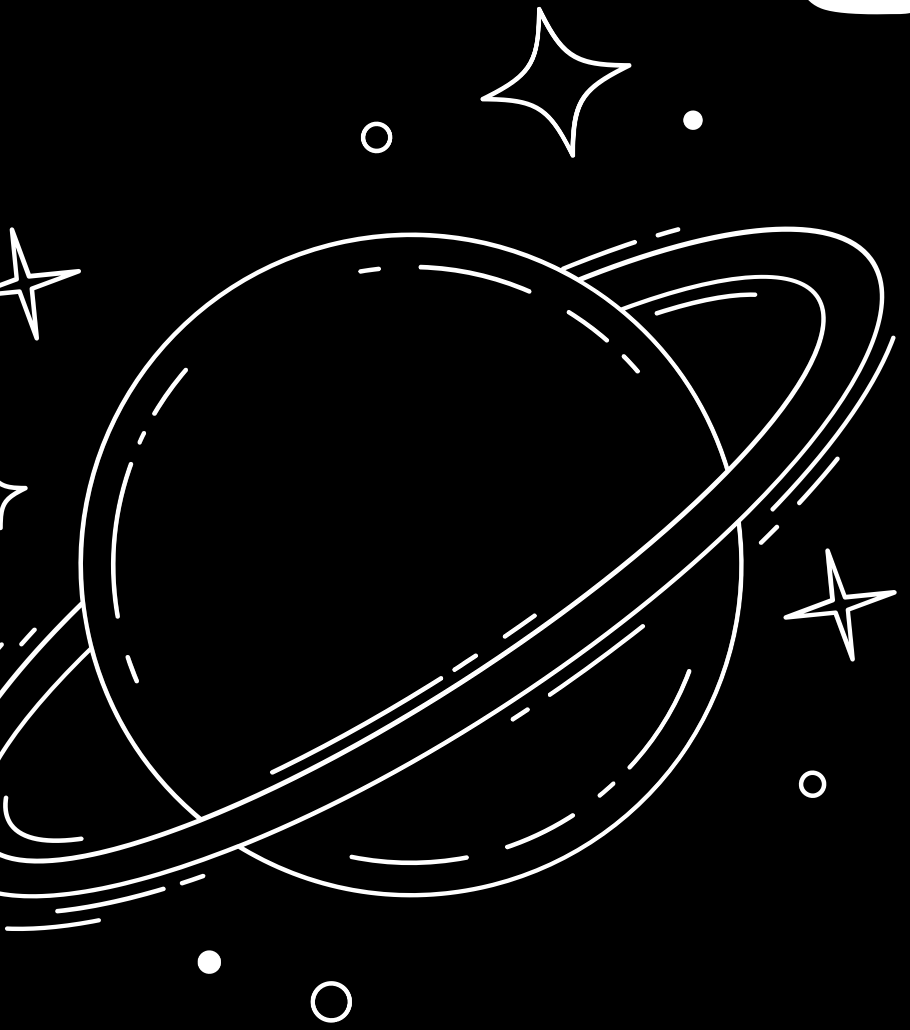
OUTLINE

- Executive Summary
- Introduction
- Methology
- Results
- Conclusion

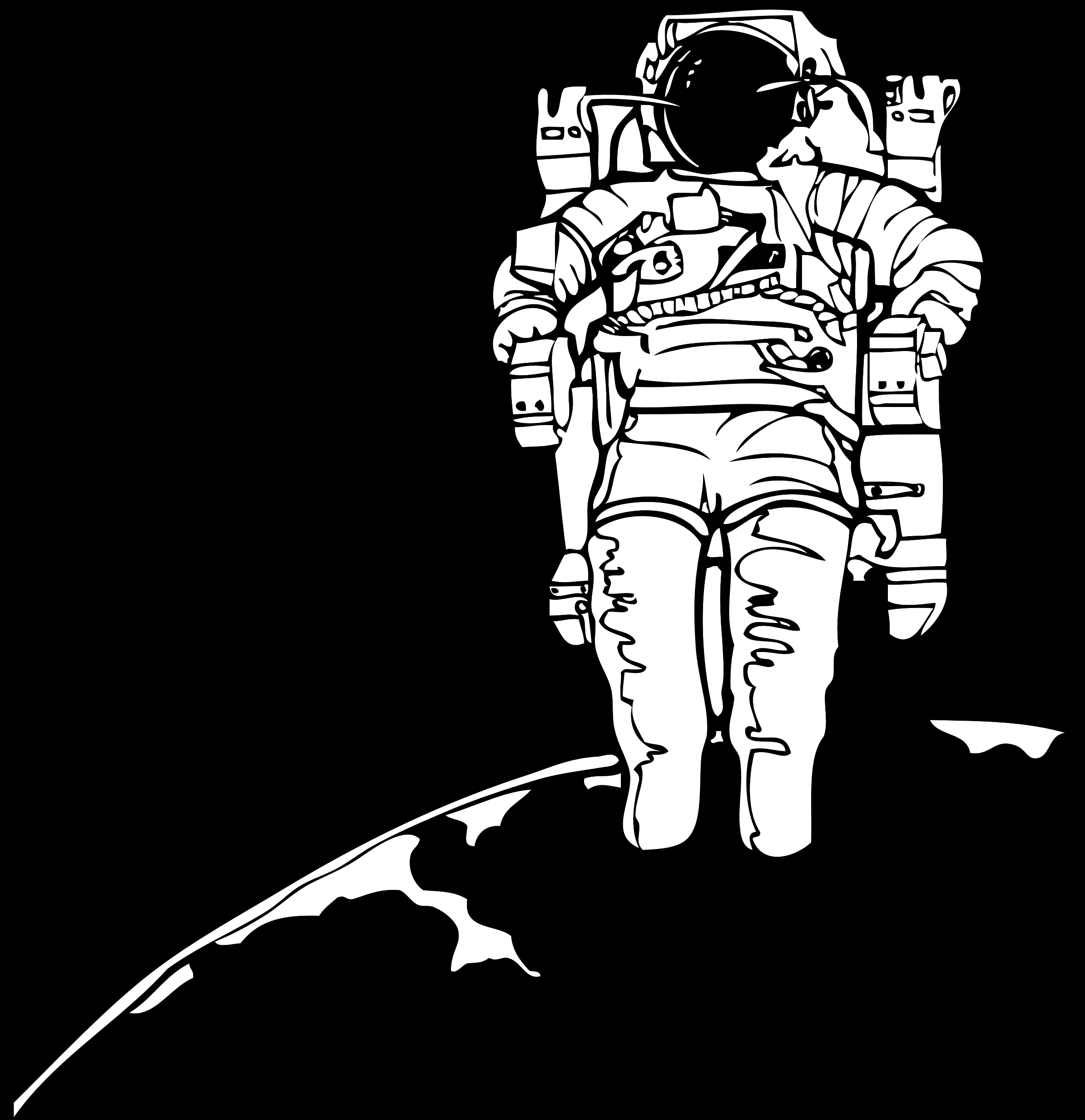
Github: https://github.com/klaudiaban/IBM-Applied-Data-Science-Capstone.git



EXECUTIVE SUMMARY



- Summary of methodologies
  - Data Collection API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis using SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analysis with Folium
  - Interactive Dashboard with Plotly
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis Results
  - Interactive Analytics in Screenshots
  - Predictive Analytics Results



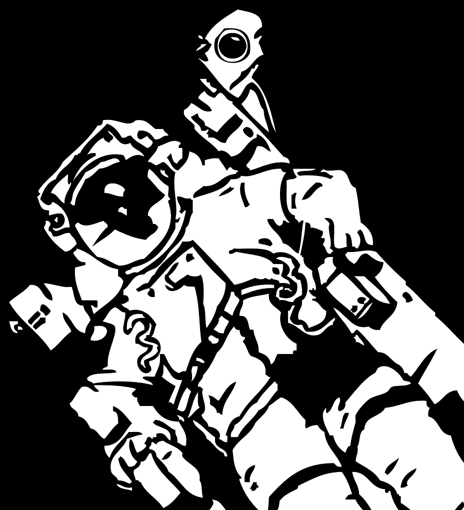
INTRODUCTION

Project background and context

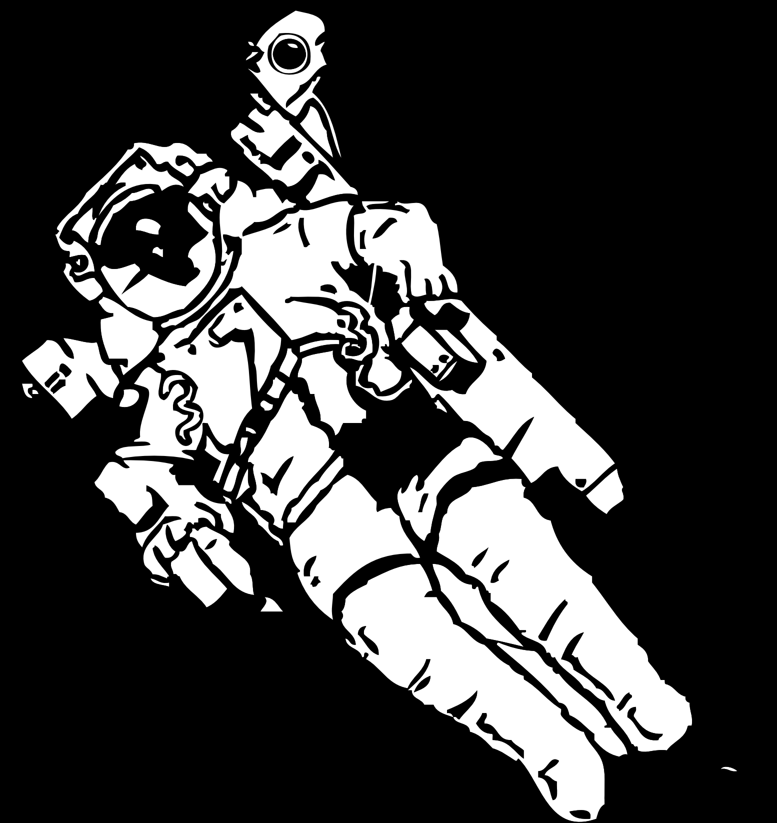
SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. The goal of the project is to predict if the first stage will land succesfully.

Questions to be answered

- What factors determine if the rocket will land succesfully?
- The interaction amongst various features that determine the success rate of a successful landing.
- What operating conditions needs to be in place to ensure a successful landing program?

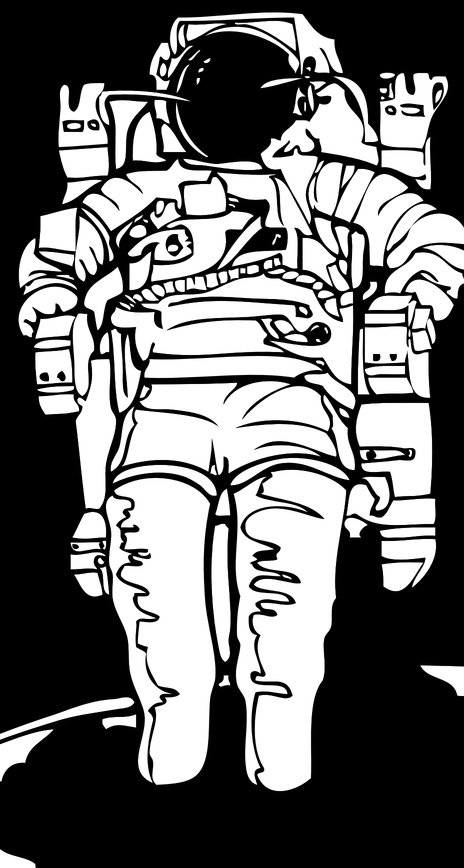


METHODOLOGY



EXECUTIVE SUMMARY

- Data collection methodology:
  - Using SpaceX Rest API
  - Using Web Scrapping from Wikipedia
- Performed data wrangling
  - Filtering the data
  - Dealing with missing values
  - One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models



DATA COLLECTION SPACEX API

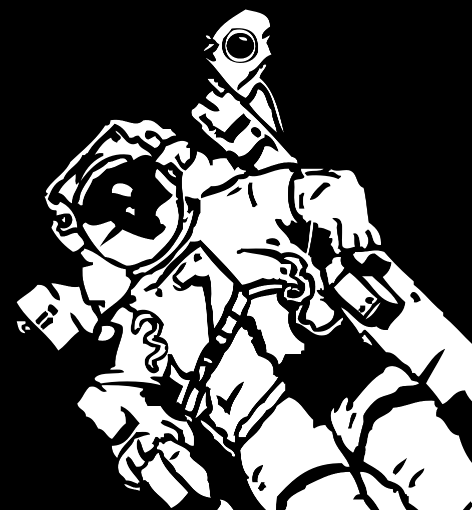
1. Requesting rocket launch data from SpaceX API
2. Decoding the response content using .json() and turning it into a dataframe using .json_normalize()
3. Requesting needed information about the launches from SpaceX API by applying custom functions
4. Constructing data we have obtained into a disctionary
5. Creating a dataframe from the dictionary
6. Filtering the dataframe to only include Falcon 9 launches
7. Replacing missing values of Payload Mass column with calculated .mean() for this column

Github: https://github.com/klaudiaban/IBM-Applied-Data-Science-Capstone/blob/main/1_Data_Collection_API.ipynb

DATA COLLECTION WEB SCRAPING

1. Requesting Falcon 9 launch data from Wikipedia
2. Creating a BeautifulSoup objecy from the HTML response
3. Extracting all column names from the HTML table header
4. Collecting the data by parsing HTML tables
5. Constructing data we have obtained into a dictionary
6. Creating a dataframe from the dictionary

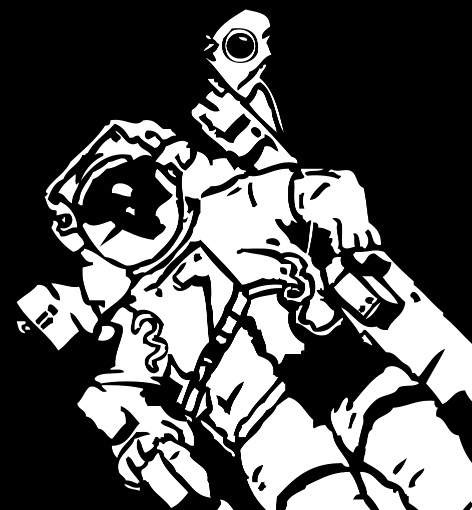
Github: https://github.com/klaudiaban/IBM-Applied-Data-Science-Capstone/blob/main/2_Data_Collection_with_Web_Scraping.ipynb



DATA WRANGLING

1. Calculate the number of launches on each site
2. Calculate the number and occurrence of each orbit
3. Calculate the number and occurrence of mission outcome per orbit type
4. Create a landing outcome label from Outcome column

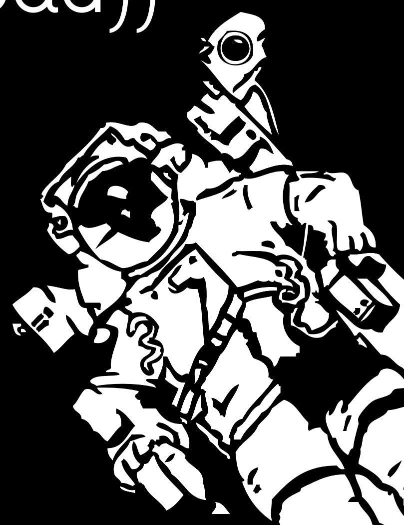
Github: https://github.com/klaudiaban/IBM-Applied-Data-Science-Capstone/blob/main/3_Data_Wrangling.ipynb



EDA WITH SQL

1. Displaying the names of the unique launch sites in the space mission
2. Displaying 5 records where launch sites begin with the string 'CCA'
3. Displaying the total payload mass carried by boosters launches by NASA (CRS)
4. Displaying average payload mass carried by booster version F9 v1.1
5. Listing the date when the first successful landing outcome in ground pad was achieved
6. Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. Listing the total number of successful and failue mission outcomes
8. Listing the names of the booster versions which have carried the maximum payload mass
9. Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015
10. Ranking 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

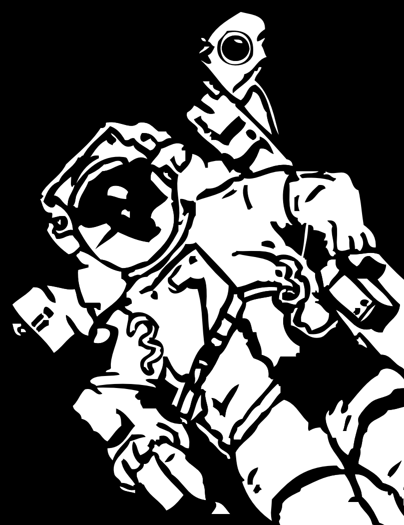
Github: https://github.com/klaudiaban/IBM-Applied-Data-Science-Capstone/blob/main/4_Complete_the_EDA_with_SQL.ipynb



EDA WITH VISUALIZATION

1. VIsualizing the relationship between Flight Number and Launch Site
2. Visualizing the relationship between Payload and Launch Site
3. Visualizing the relationship between success rate of each orbit type
4. Visualizing the relationship between FlightNumber and Orbit type
5. Visualizing the relationship between Payload and Orbit type
6. Visualizing the launch success yearly trend
7. Creating dummy variables to categorical columns
8. Casting all numeric columns to float64

Github: https://github.com/klaudiaban/IBM-Applied-Data-Science-Capstone/blob/main/5_EDA_with_Visualization.ipynb



INTERACIVE VISUAL ANALYSIS WITH FOLIUM

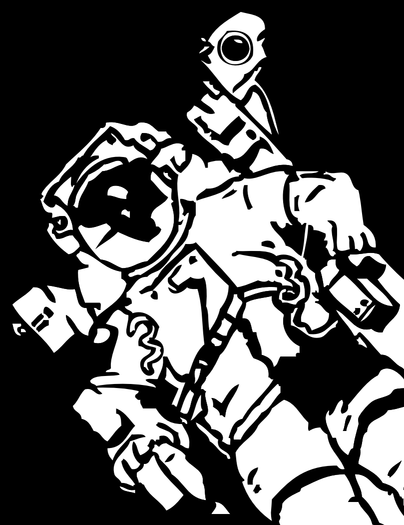
1. Marking all launch sites on a map
2. Marking the success/failed launches for each site on the map
3. Calculating the distances between a launch site to its proximities

Github: https://github.com/klaudiaban/IBM-Applied-Data-Science-Capstone/blob/main/6_Interactive_Visual_Analytics_with_Folium.ipynb

INTERACIVE VISUAL ANALYSIS WITH PLOTLY DASH

1. Adding a dropdown list to enable Launch Site selection
2. Adding a pie chart to show the total successful launches count for all sites
3. Adding a slider to select payload range
4. Adding a scatter chart to show the correlation between payload and launch success
5. Adding a callback function for ‘site-dropdown’ as input, ‘success-pie-chart’ as output
6. Adding a callback function for ‘site-dropdown’ and ‘payload-slider’ as inputs, ‘success-payload-scatter-chart’ as output

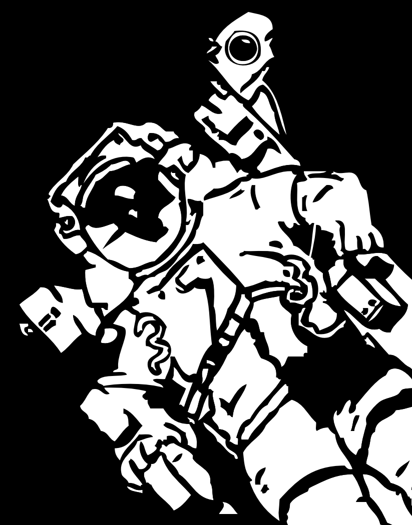
Github: https://github.com/klaudiaban/IBM-Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py



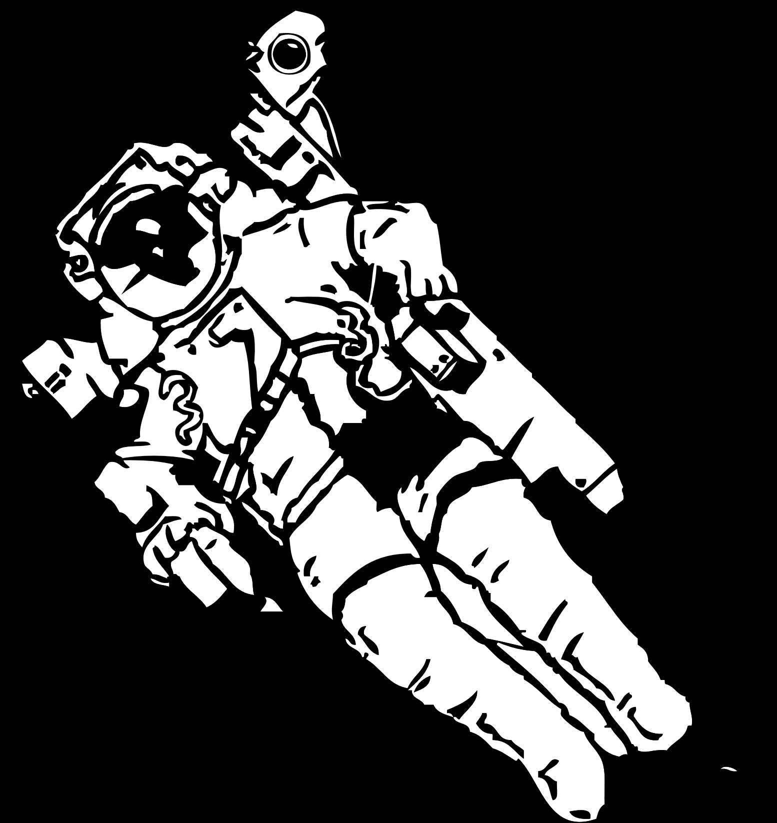
MACHINE LEARNING PREDICTION

1. Creating a NumPy array from the column Class in data, by applying the method to_numpy() then assign it to the variable Y,make sure the output is a Pandas series (only one bracket df['name of column'])
2. Standardizing the data in X then reassign it to the variable X using the provided transform
3. Using the function train_test_split to split the data X and Y into training and test data
4. Creating a logistic regression object then creating a GridSearchCV object logreg_cv with cv = 10
5. Creating a support vector machine object then creating a GridSearchCV object svm_cv with cv = 10
6. Creating a decision tree classifier object then creating a GridSearchCV object tree_cv with cv = 10
7. Creating a decision tree classifier object then creating a GridSearchCV object tree_cv with cv = 10
8. Creating a k nearest neighbors object then creating a GridSearchCV object knn_cv with cv = 10
9. Fiting the object to find the best parameters from the dictionary parameters
10. Calculating the accuracy on the test data using the method score

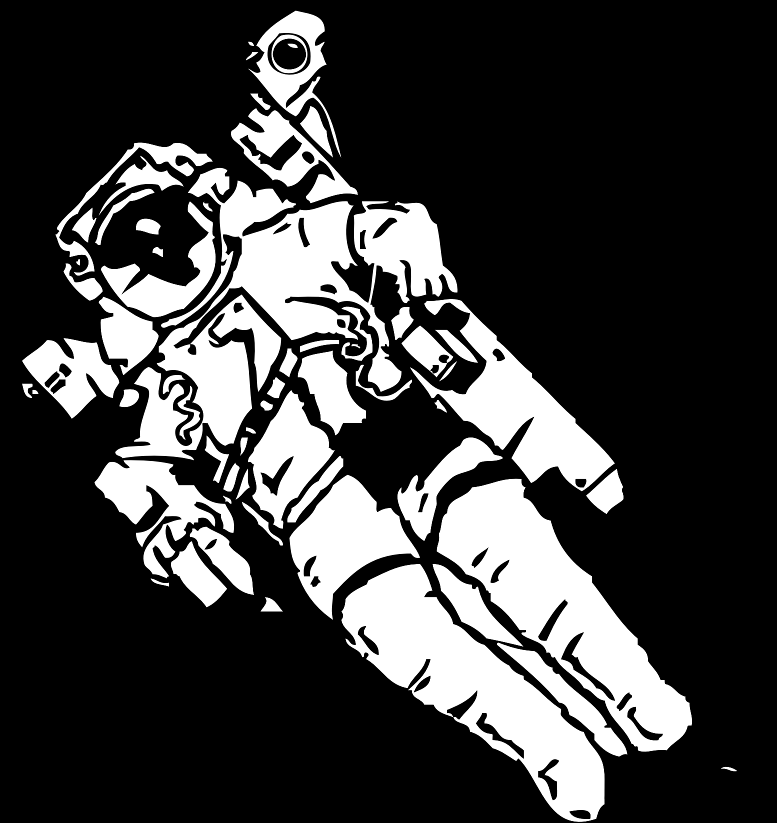
Github: https://github.com/klaudiaban/IBM-Applied-Data-Science-Capstone/blob/main/7_Complete_the_Machine_Learning_Prediction.ipynb



RESULTS



EDA WITH SQL



ALL LAUNCH SITE NAMES

```
%sql select distinct launch_site from SPACEXTBL;
```

Python

```
 * 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 SPACEXTBL 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
2010-06-04	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

TOTAL PAYLOAD MASS

```
%sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXTBL where customer = 'NASA (CRS)';
```

Python

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

total_payload_mass
45596

AVERAGE PAYLOAD MASS BY F9 V1.1

```
%sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXTBL where booster_version like '%F9 v1.1%';
```

Python

* sqlite:///my_data1.db
Done.

average_payload_mass
2534.6666666666665

FIRST SUCCESSFUL GROUND LANDING DATE

```python
%sql select min(date) as first_successful_landing from SPACEXTBL where "Landing_Outcome" = 'Success (ground pad)';
```

Python

 * sqlite:///my_data1.db
Done.

first_successful_landing
2015-12-22

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

```
%sql select booster_version from SPACEXTBL where "Landing_Outcome" = 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000;
```

Python

```
 * 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

```
%sql select mission_outcome, count(*) as total_number from SPACEXTBL group by mission_outcome;
```

Python

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

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

BOOSTERS CARRIED MAXIMUM PAYLOAD

```
%sql select booster_version from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL);
```

Python

* 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

```
%%sql SELECT
    SUBSTR(date, 6, 2) AS month,
    date,
    booster_version,
    launch_site,
    landing_outcome
FROM SPACEXTBL
WHERE landing_outcome = 'Failure (drone ship)'
AND SUBSTR(date, 1, 4) = '2015';
```

Python

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

month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

RANK SUCCESS COUNT BETWEEN 2010-06-04 AND 2017-03-20

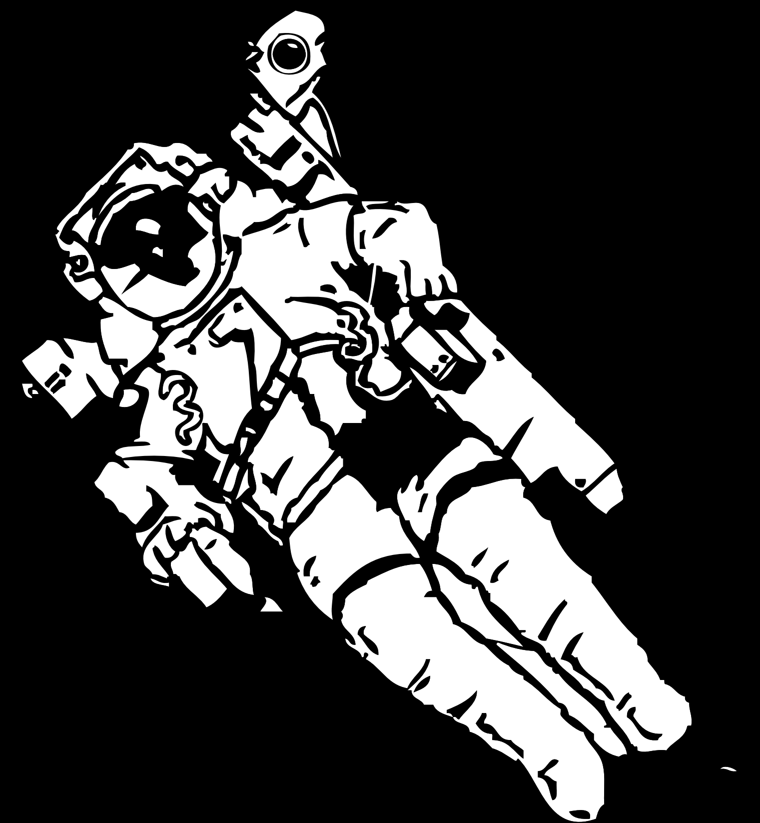
```
%%sql select "Landing_Outcome", count(*) as count_outcomes from SPACEXTBL
      where date between '2010-06-04' and '2017-03-20'
      group by "Landing_Outcome"
      order by count_outcomes desc;
```

Python

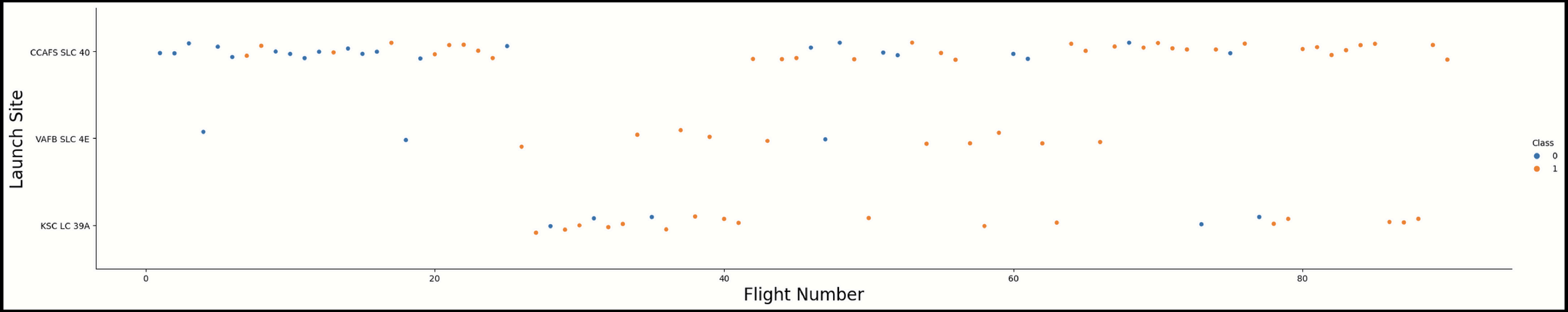
* sqlite:///my_data1.db
Done.

Landing_Outcome	count_outcomes
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

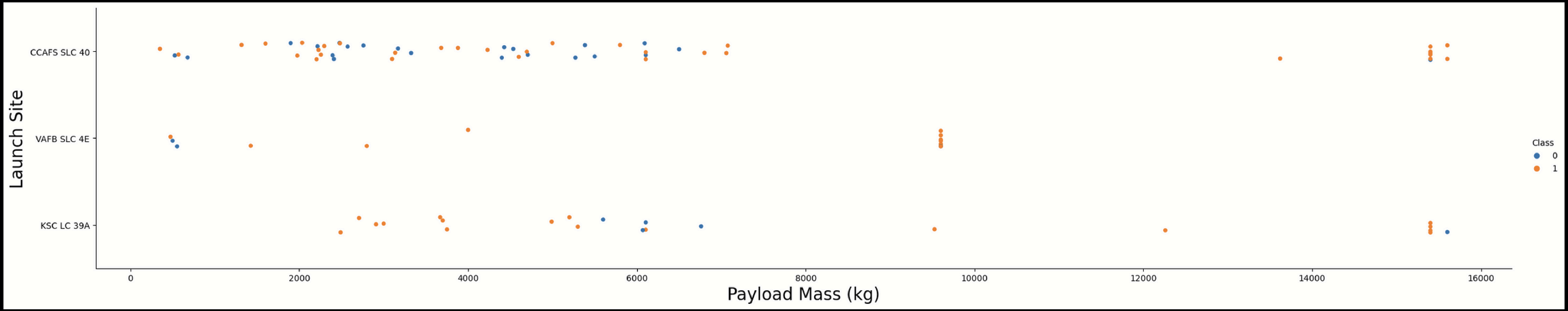
EDA WITH VISUALIZATION



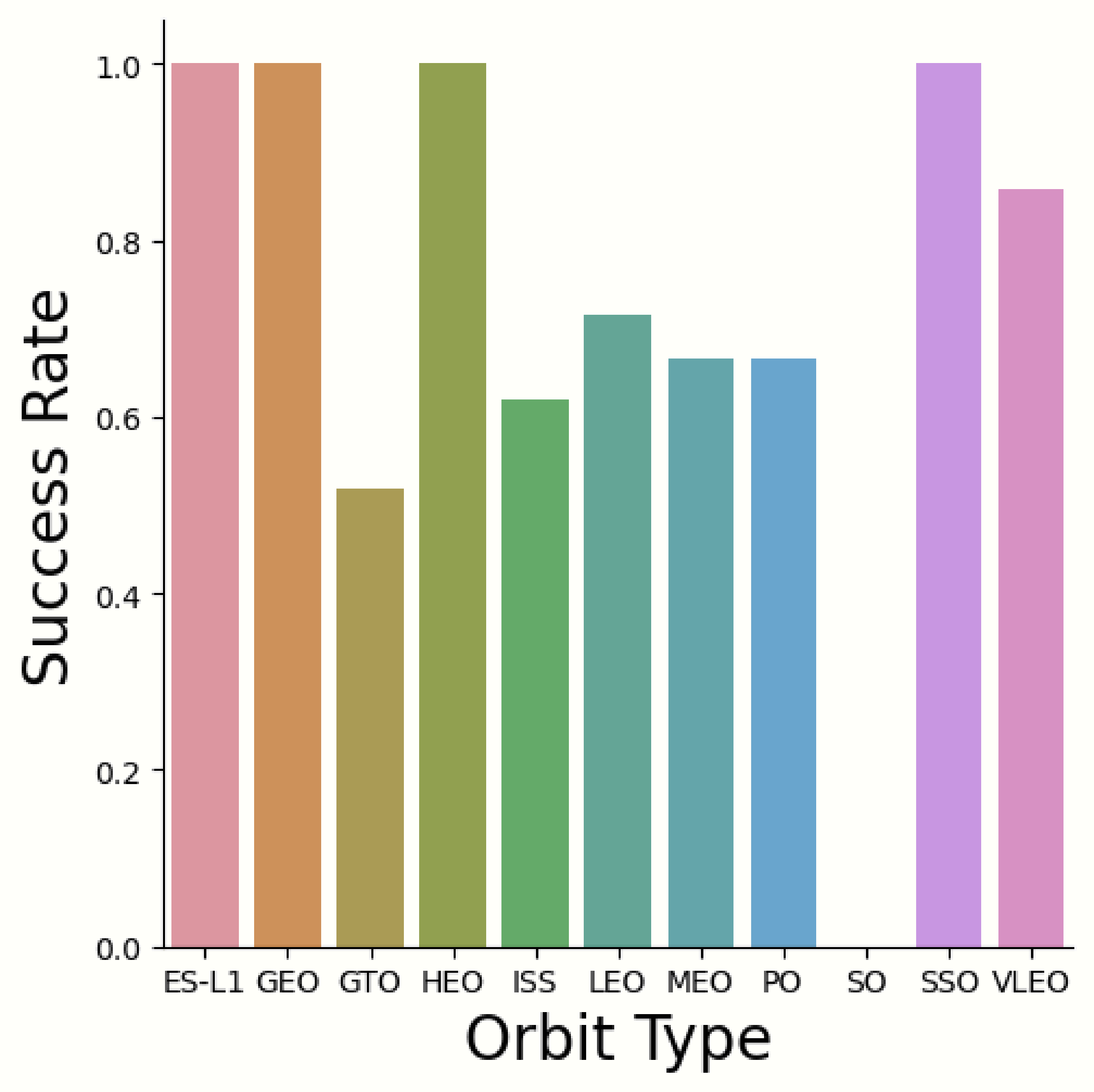
FLIGHT NUMBER VS LAUNCH SITE



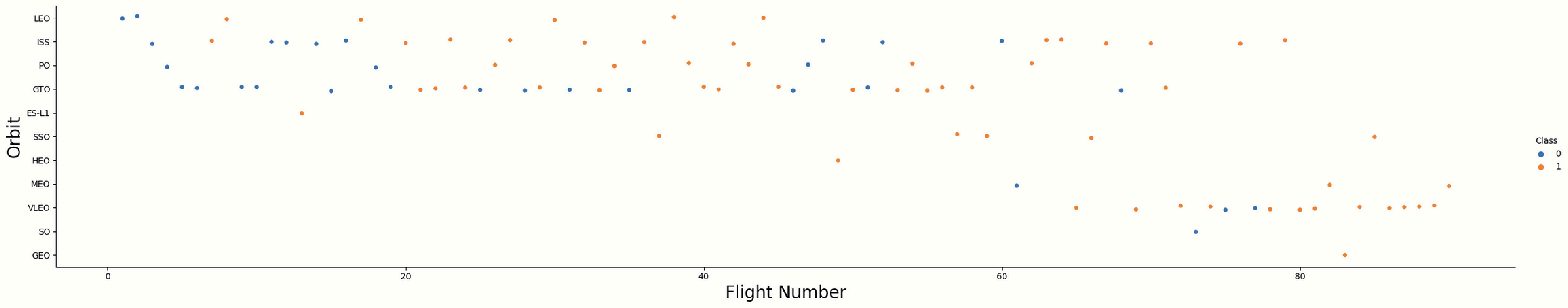
PAYLOAD MASS VS LAUNCH SITE



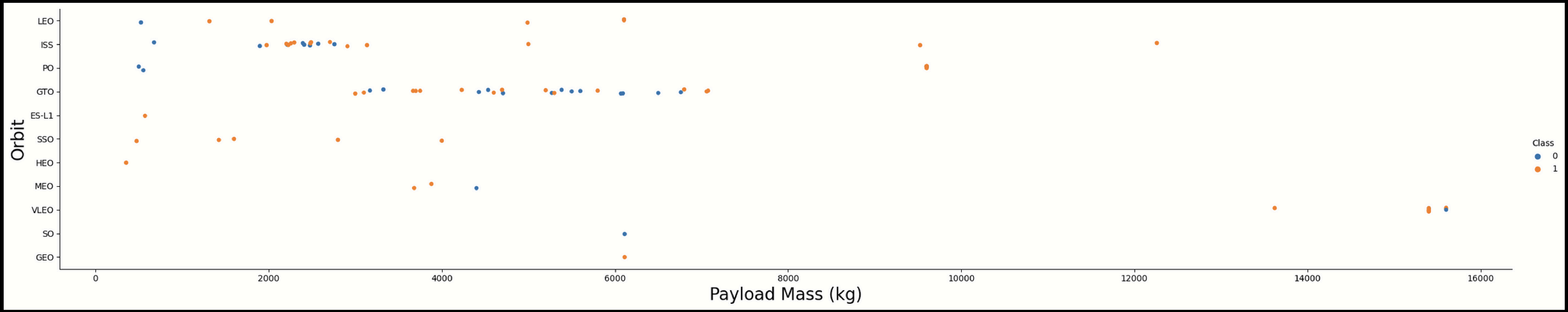
SUCCESS RATE VS ORBIT TYPE



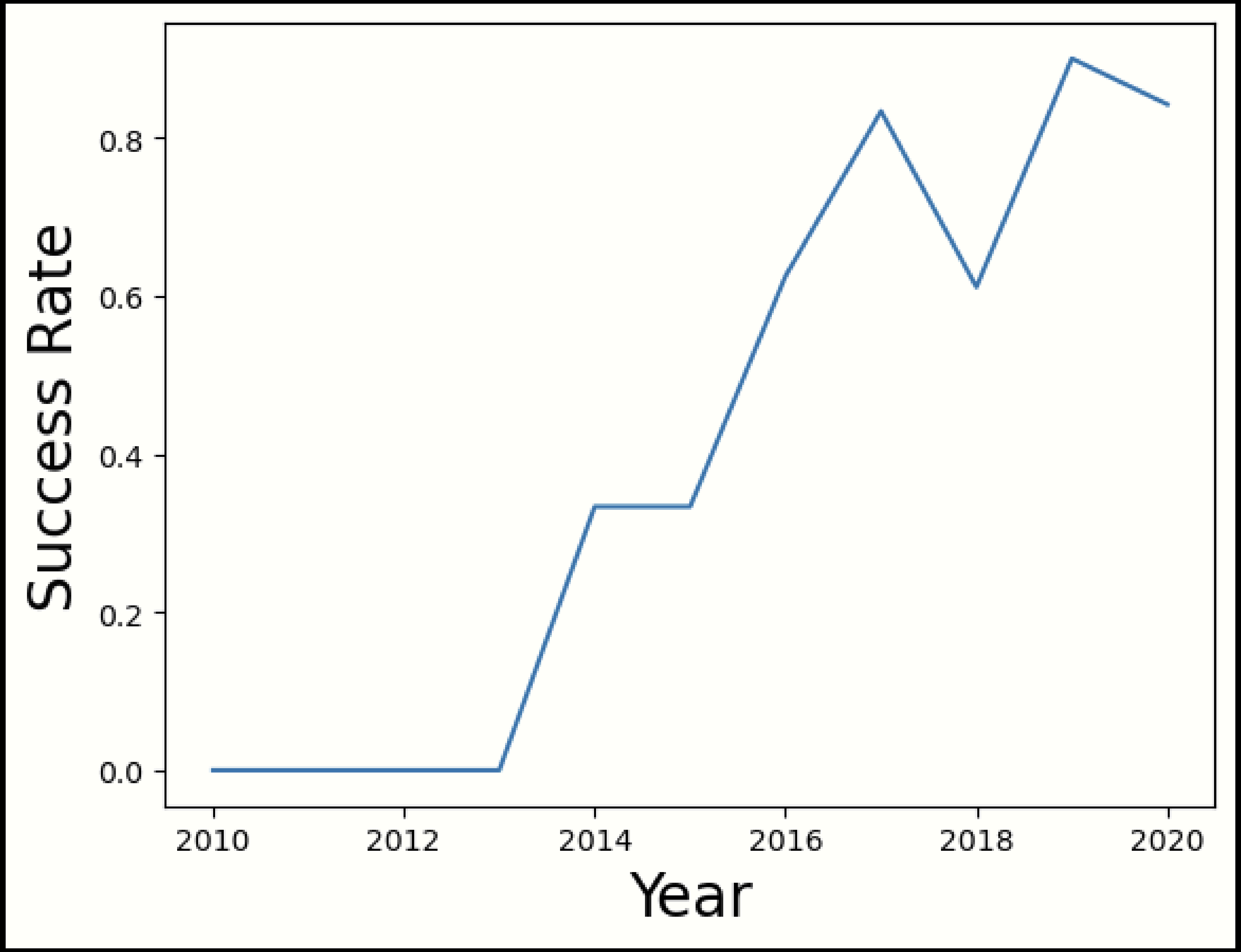
FLIGHT NUMBER VS ORBIT TYPE



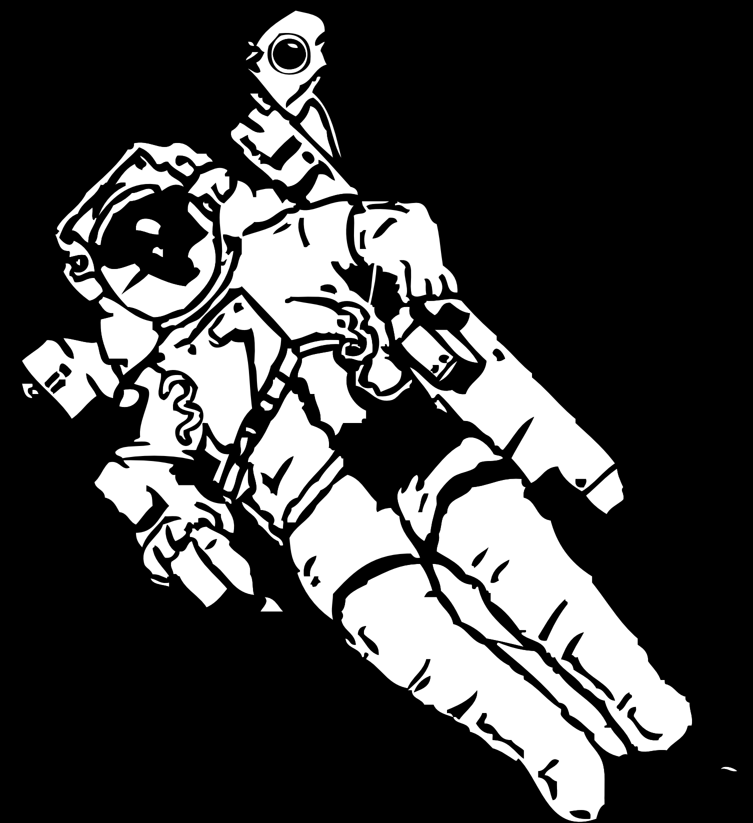
PAYLOAD MASS VS ORBIT TYPE



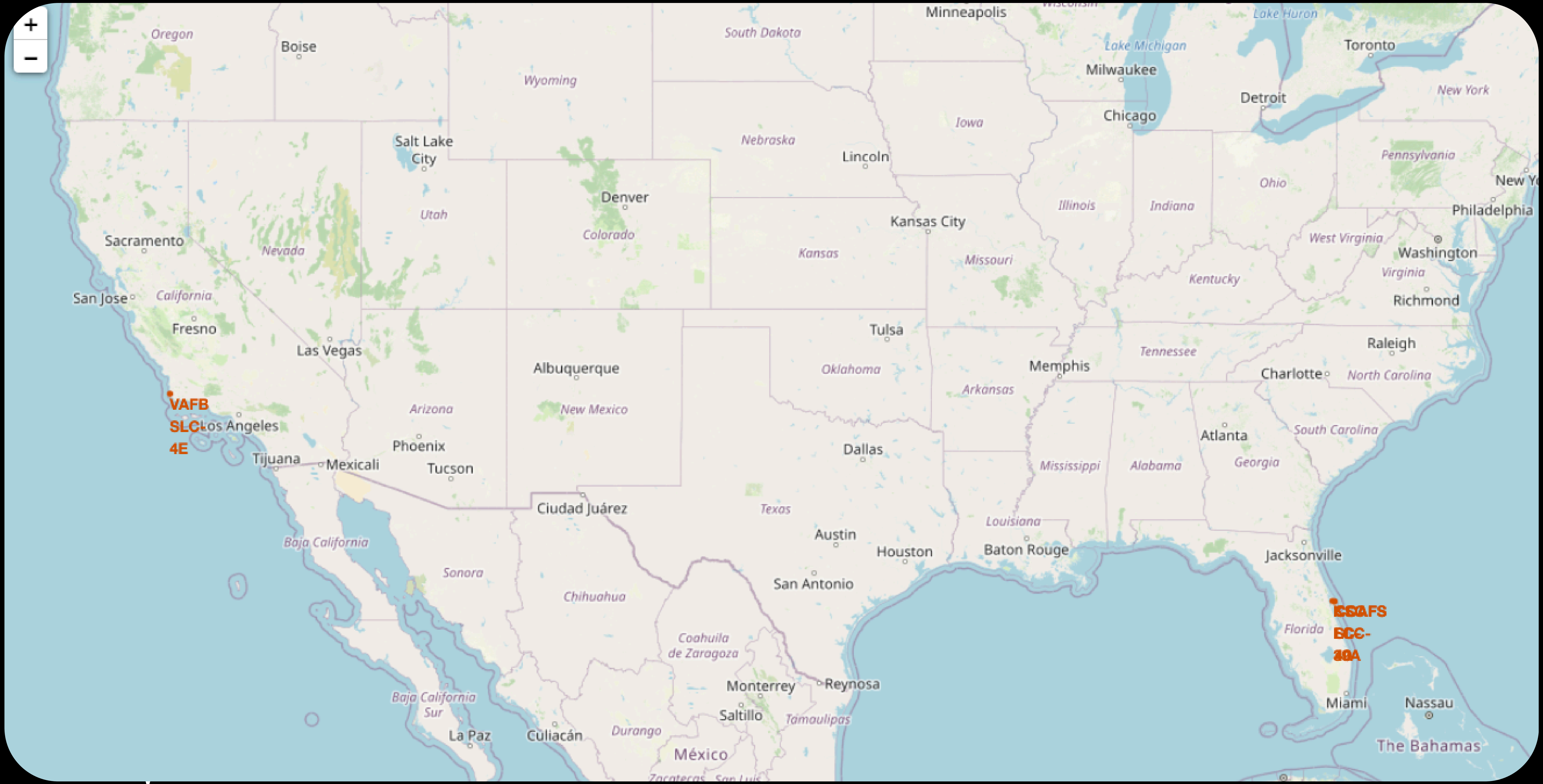
LAUNCH SUCCESS YEARLY TREND



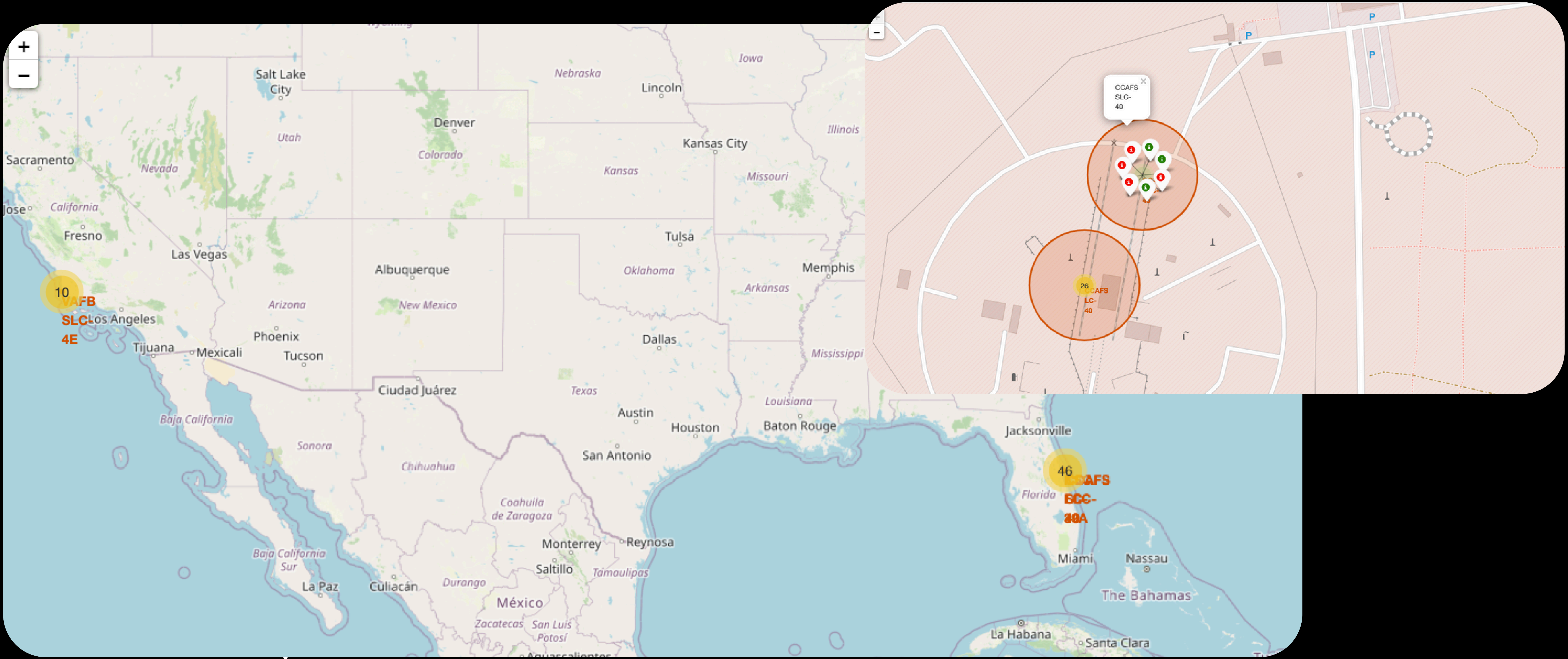
INTERACTIVE ANALYSIS WITH FOLIUM



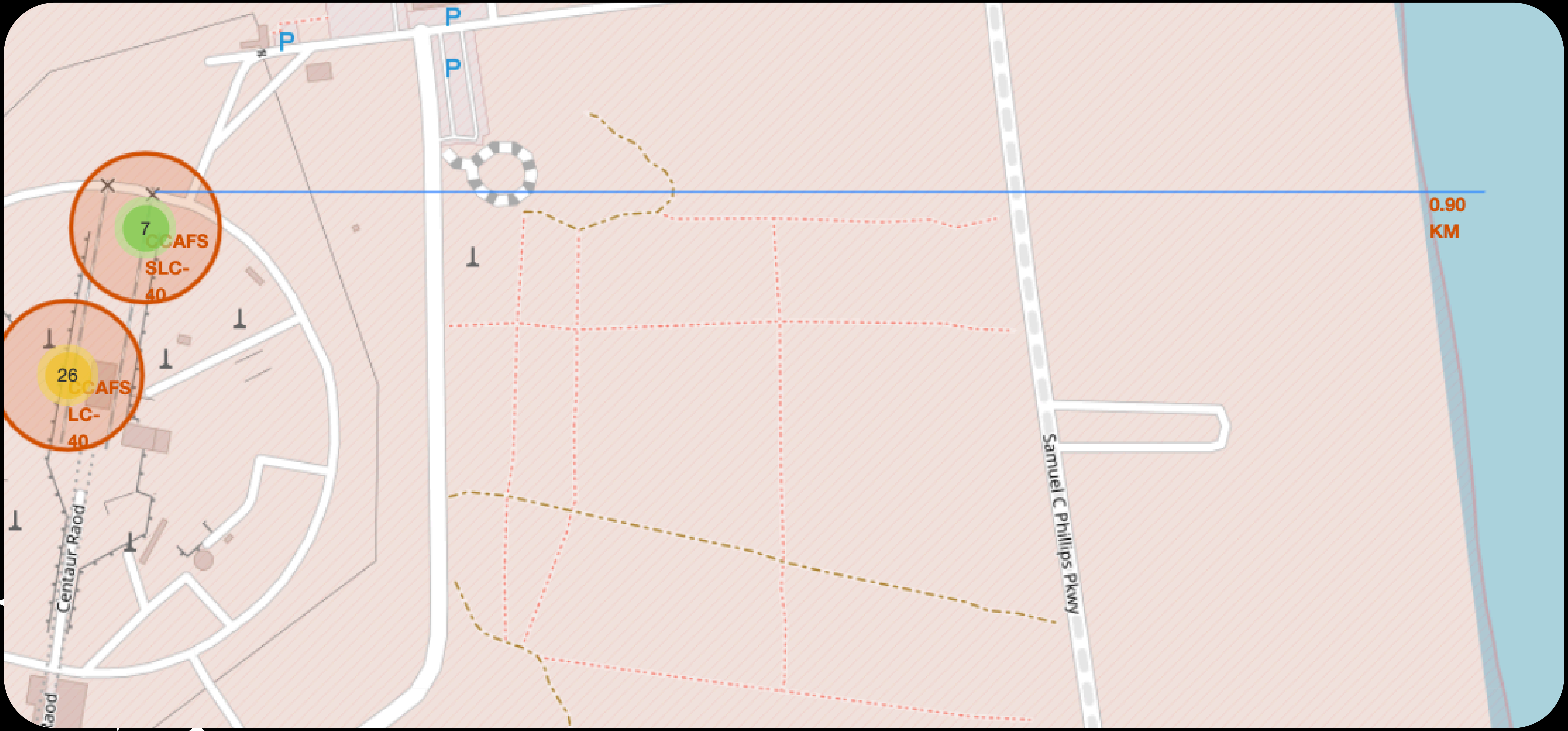
ALL LAUNCH SITES ON A MAP



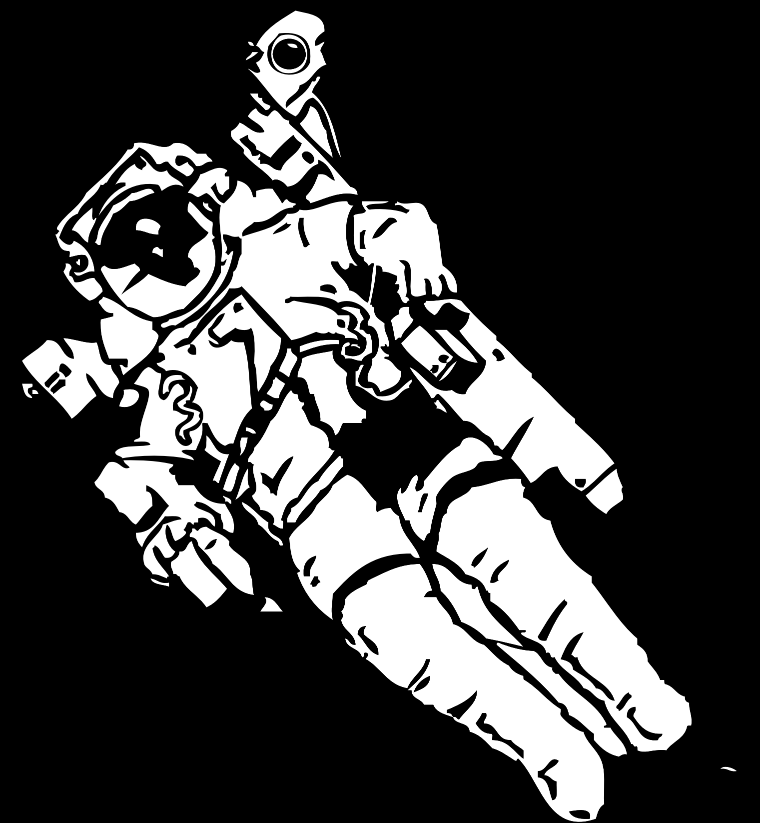
THE SUCCESS/FAILED LAUNCHES FOR EACH SITE ON THE MAP



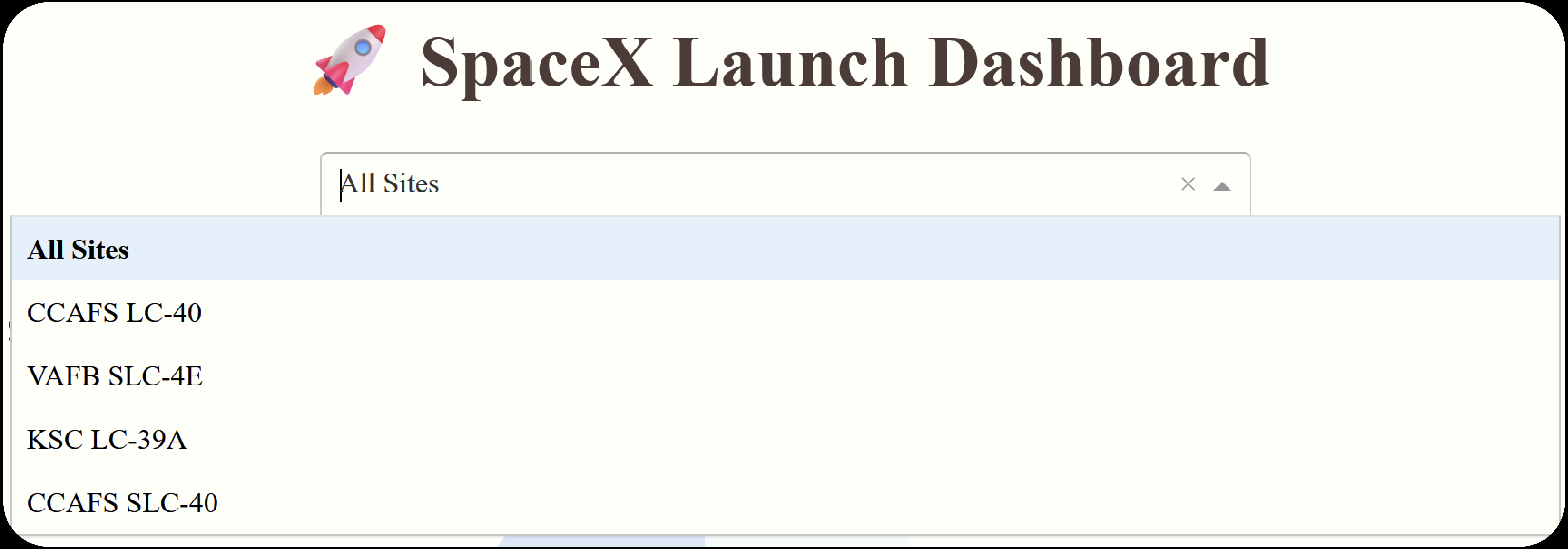
THE DISTANCES BETWEEN A LAUNCH SITE TO ITS PROXIMITIES



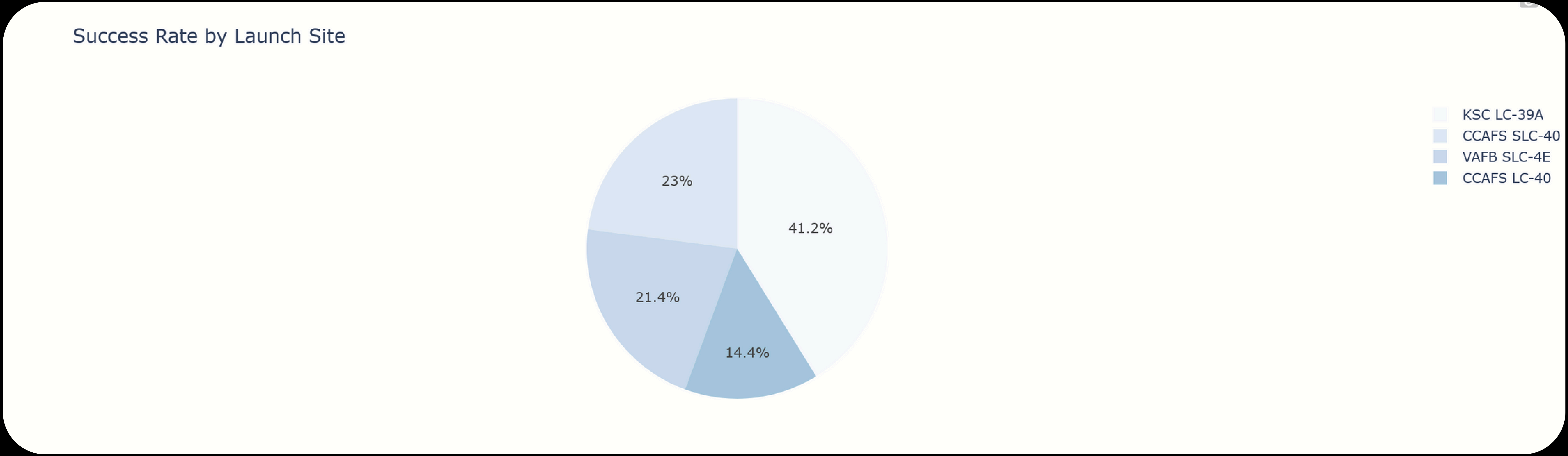
INTERACTIVE ANALYSIS WITH PLOTLY DASH



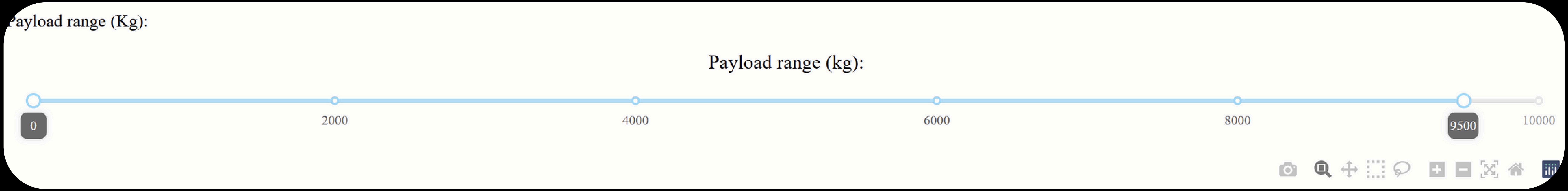
DROPDOWN LIST TO ENABLE LAUNCH SITE SELECTION



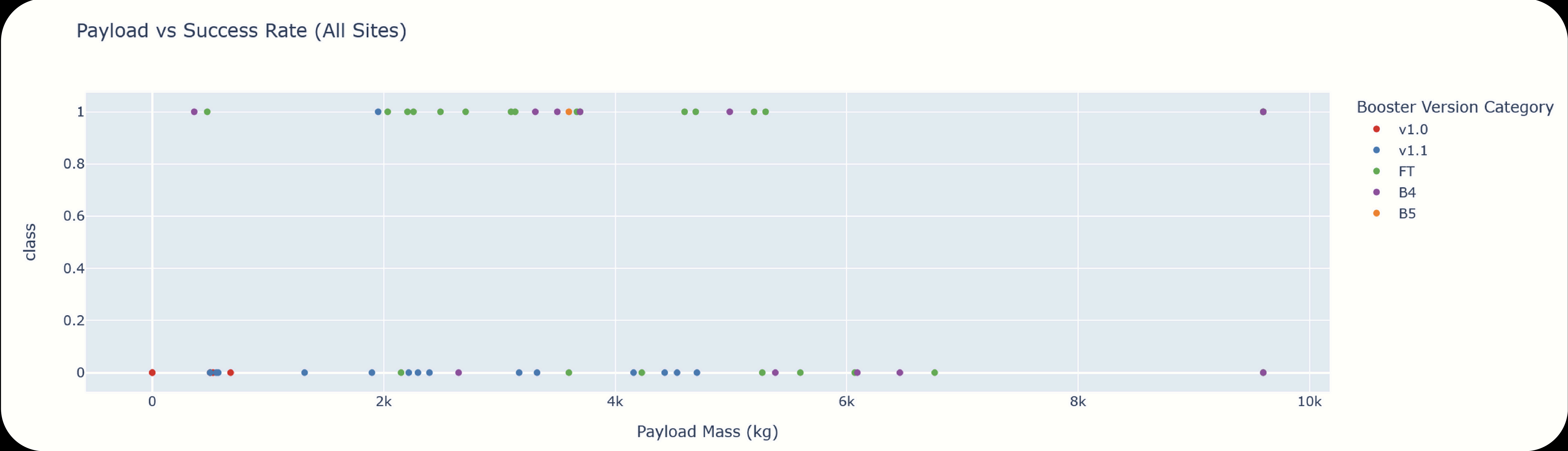
PIE CHART TO SHOW THE TOTAL SUCCESSFUL LAUNCHES COUNT FOR ALL SITES



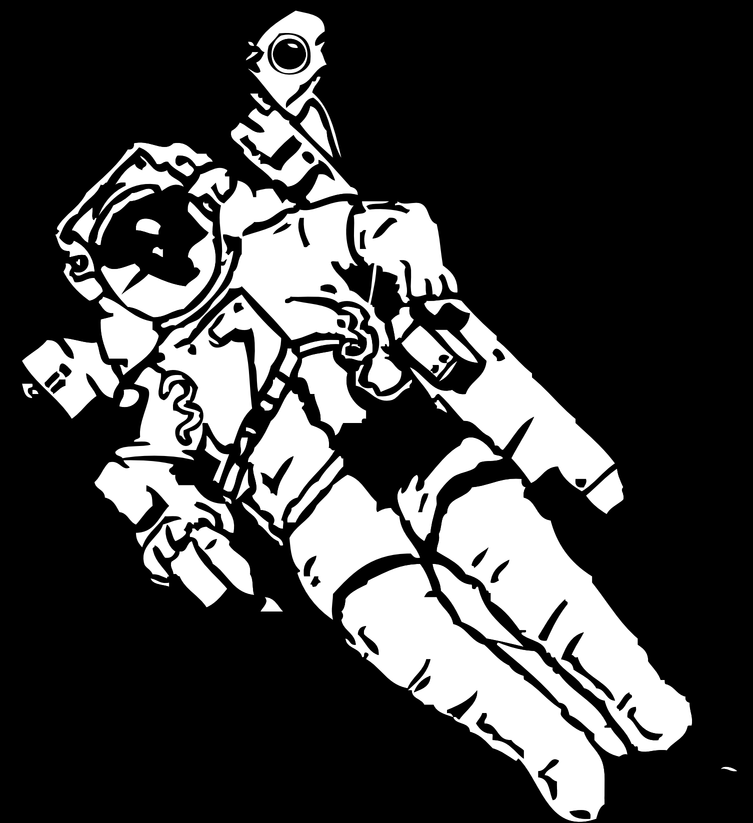
SLIDER TO SELECT PAYLOAD RANGE



SCATTER CHART TO SHOW THE CORRELATION BETWEEN PAYLOAD AND LAUNCH SUCCESS



MACHINE LEARNING PREDICTION



SCORES AND ACCURACY OF THE TEST SET

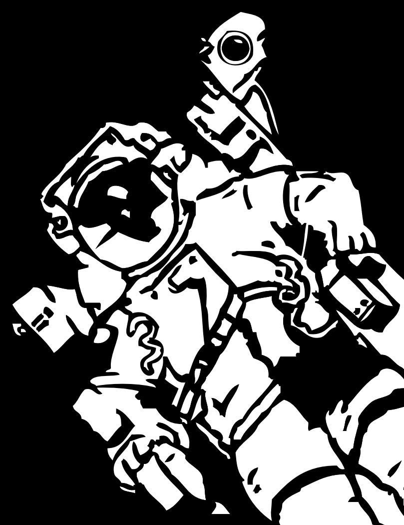
	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

SCORES AND ACCURACY OF THE ENTIRE DATA SET

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.882353	0.819444
F1_Score	0.909091	0.916031	0.937500	0.900763
Accuracy	0.866667	0.877778	0.911111	0.855556

CONCLUSION

- The Decision Tree Model performs best on this dataset.
- Launches carrying lighter payloads achieve higher success rates compared to those with heavier payloads.
- Most launch sites are located near the Equator and are positioned close to coastal areas.
- Over the years, the success rate of launches has steadily increased.
- Among all launch sites, KSC LC-39A boasts the highest success rate.
- Orbits ES-L1, GEO, HEO, and SSO have a perfect 100% success rate.



THANK YOU