# Winning Space Race with Data Science

Dany A. DARGHOUTH

September 2023

IBM Developer SKILLS NETWORK

# Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

# Executive Summary

- Summary of methodologies
  - Quantitative and Qualitative analysis of data scraped form available data thanks to the SpaceX API
  - Machine Learning Methods for prediction
- Summary of all results
  - Support Vector Machines and KNN are best suited for the predictions
  - Predictions accuracy of 78%

# Introduction

- Project background and context
  - SpaceY a new Rocket Company founded by Allon Musk, wishes to compete with SpaceX.
- Problems you want to find answers
  - In order to have a better understanding of the competition, the goal of this project is to be able to predic t the cost of a given SpaceX launch aswell as the probability of the company re—using the first stage of the launched rocket.

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
    - Data was scrapped form the available SpaceX API
- Perform data wrangling
    - The retrieved data was then cleaned up, nulls quantitative values were replaced with the mean  and categorical null values were kept.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - The data set was split into training and test data, various model were then trained and tested.

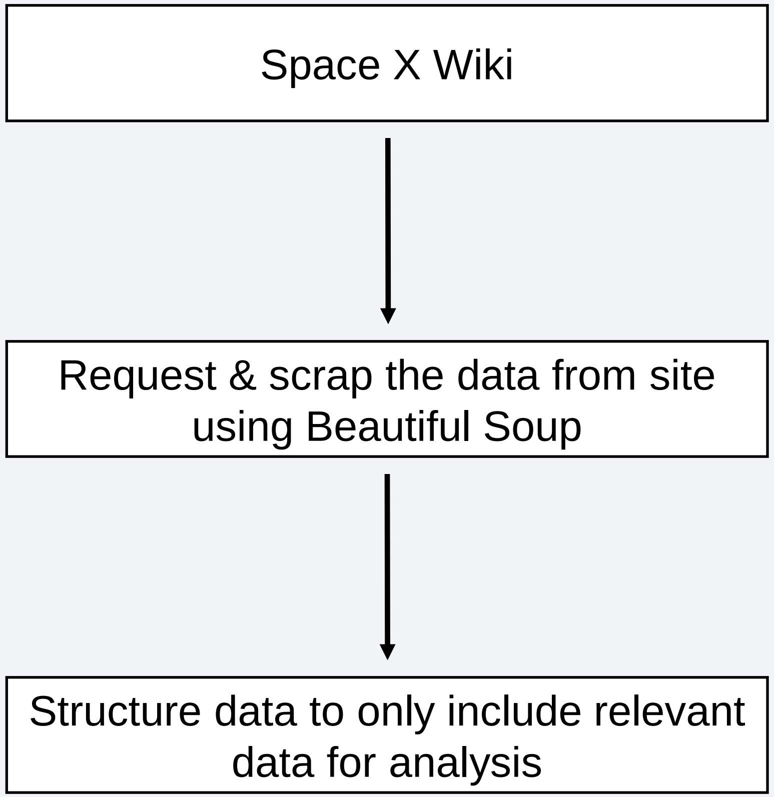
# Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

# Data Collection – SpaceX API

**Github:**

https://github.com/Dany-Drgh/ibm-data_science/blob/main/Capstone%20course/Week%201/Data%20Collection%20API.ipynb

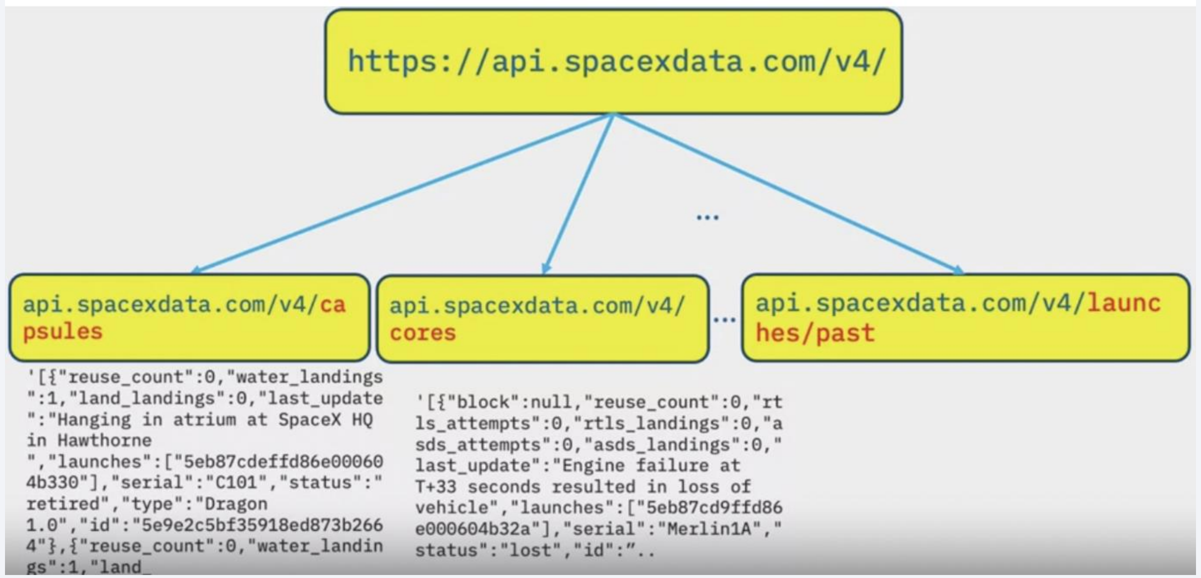


# Data Collection - Scraping

**Collected the data on falcon 9 launches straight from the site**

**Github link:**

https://github.com/Dany-Drgh/ibm-data_science/blob/main/Capstone%20course/Week%201/Data%20Collection%20API.ipynb



# EDA with Data Visualization

**Variety of Charts Plotted**

- Flight No vs Payload Mass
- Flight Number vs Launch Site
- Payload Mass vs Launch Site
- Orbit vs Class
- Flight Number vs Orbit
- PayloadMass vs Orbit
- Data vs Class
- **Github (Both EDA notebooks)**

https://github.com/Dany-Drgh/ibm-data_science/tree/main/Capstone%20course/Week%202

# EDA with SQL

- **%sql** SELECT Launch_Site FROM SPACEXTBL WHERE Launch_Site LIKE ‘%CCA%'
- **%sql** SELECT SUM(PAYLOAD_MASS__KG_) from SPACEXTBL
- **%sql** SELECT AVG(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE Booster_Version = 'F9 v1.1’
- **%sql** SELECT MIN(Landing_Outcome), Date FROM SPACEXTBL
- **%sql** SELECT Booster_Version FROM SPACEXTBL Where Mission_Outcome = 'Success' AND PAYLOAD_MASS__KG_ between 4000 and 6000
- **%sql** SELECT COUNT(*) FROM SPACEXTBL where  Mission_Outcome LIKE '%Success%'
- **%sql** SELECT * FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ > (SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
- **%sql** SELECT substr(Date, 4, 2) as MONTH, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTBL where Landing_Outcome like '%Failure (D%'
- **%sql** SELECT COUNT(Landing_Outcome) from SPACEXTBL where Date Between '04-06-2010' and '20-03-2017' GROUP BY Landing_Outcome LIKE '%SUCCESS%' ORDER BY Landing_Outcome

# Build an Interactive Map with Folium

- Marked all launch sites on map
- Marked all successful/failed launches for each site on the map
- Calculated the distances between a launch site to its proximities
- Added markers around the launch sites because it gives a good ideal of where and why the launches were done here
- Clustered the markers based off whether or not the launch was successful also gave great data insights
- This made it easy to identify which launch sites have relatively high success rates
- https://github.com/Dany-Drgh/ibm-data_science/tree/main/Capstone%20course/Week%203

# Build a Dashboard with Plotly Dash

- In the dashboard, we are looking at the various types of launches and their success and failure rates.
- There are 3 different launch sites we explore

**Github :**

https://github.com/Dany-Drgh/ibm-data_science/blob/main/Capstone%20course/Week%203/spacex_dash_app.py

# Predictive Analysis (Classification)

- Built a classification model using various supervised machine learning classification algorithms

- Most efficient ones appear to be KNN and SVM

- **Github:**

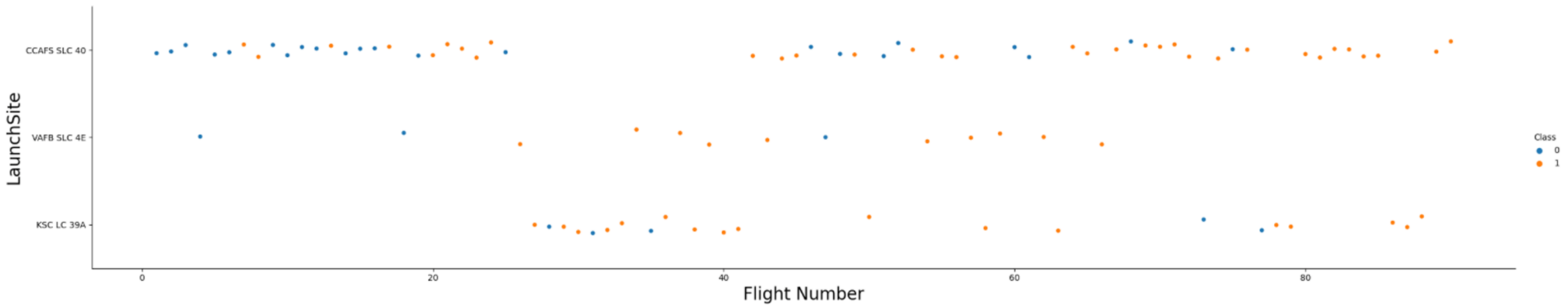
https://github.com/Dany-Drgh/ibm-data_science/blob/main/Capstone%20course/Week%204/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Section 2

# Insights drawn from EDA

# Flight Number vs. Launch Site

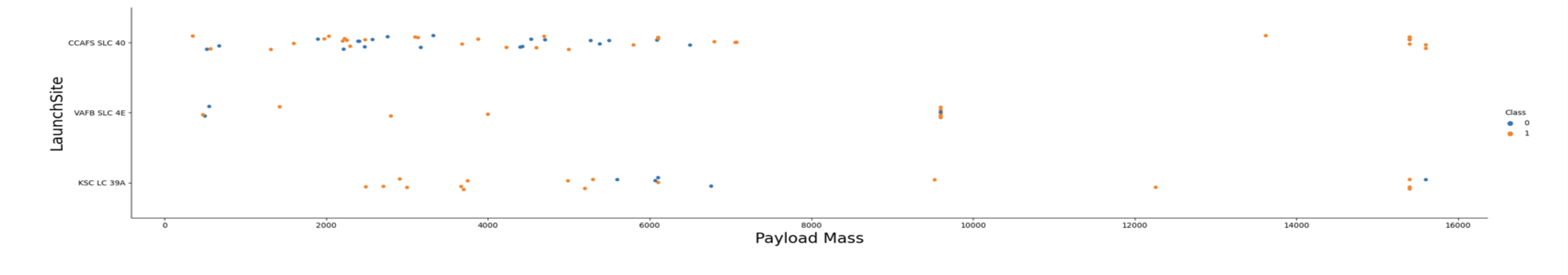
- Launches are done in clusters



# Payload vs. Launch Site

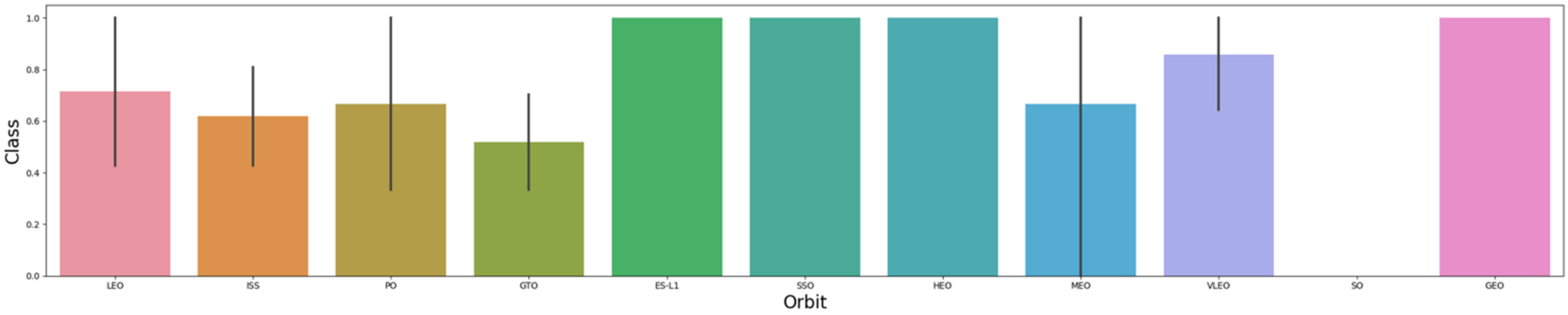
- The higher the payload Mass, the more success the landing.

- A lot of the landings were successful KSC LC 39A when the payload mass was less than 6000



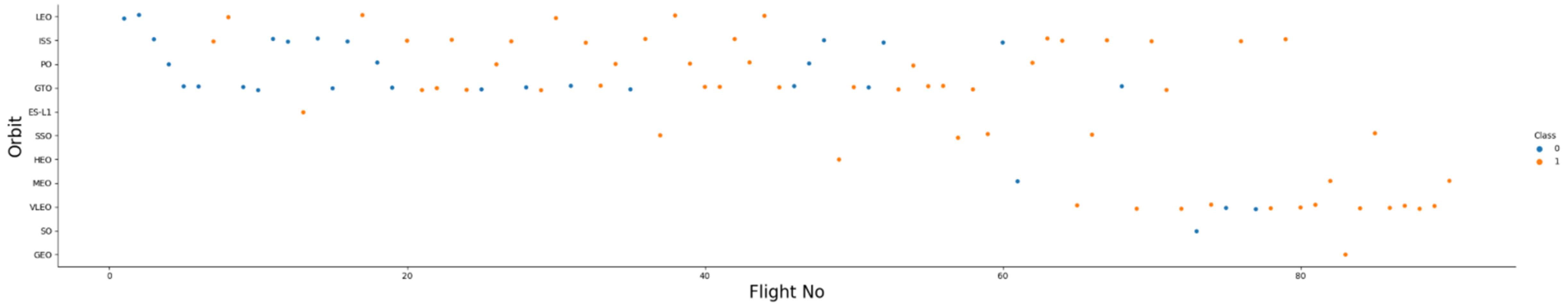
# Success Rate vs. Orbit Type

- ES-L1, SSO and HEO has a perfect success rate



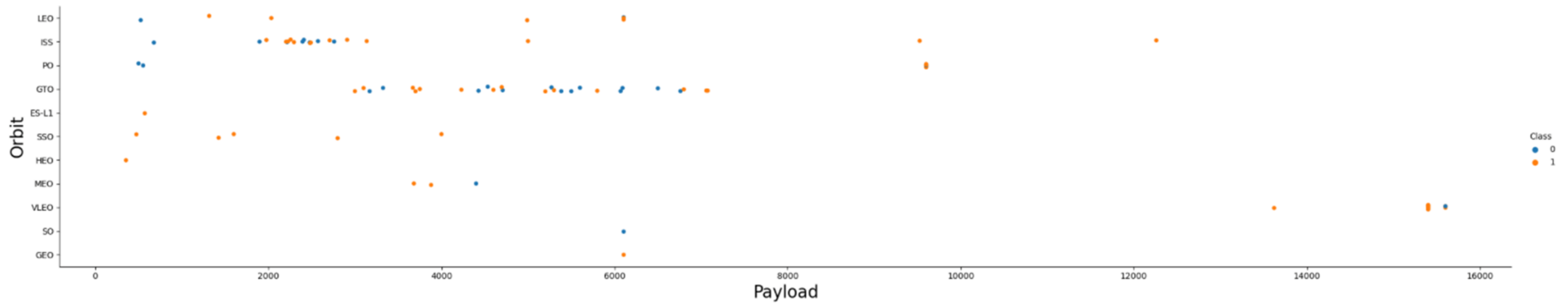
# Flight Number vs. Orbit Type

- Flight numbers over 80 typically have a succesful launch



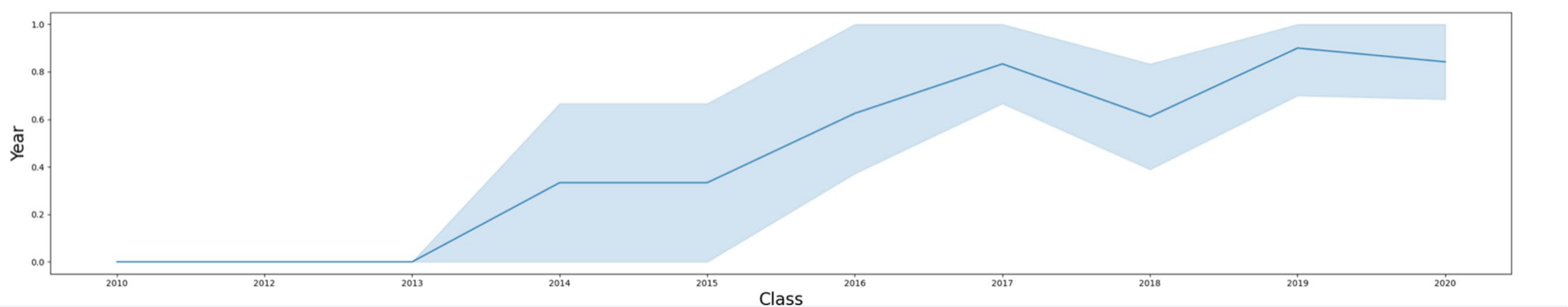
# Payload vs. Orbit Type

- Heavy payloads have more successful landings for Polar, LEO and ISS



# Launch Success Yearly Trend

- Success rate was on a steady increase until 2017
- Then it dipped slightly in 2018, to begin increasing again



# All Launch Site Names

- CCAFS LC-40
- VAFB SLC-4E
- KSC LC-39A
- CCAFS SLC-40

```
%sql select distinct(Launch_Site) from SPACEXTABLE
```

# Launch Site Names Begin with 'CCA'

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 2010-04-06 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 2010-08-12 | 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 |
| 2012-08-10 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 2013-01-03 | 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_in_kg' from SPACEXTABLE where CUSTOMER = 'NASA (CRS)'
```

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

| TOTAL_in_kg |
|---|
| 45596 |

# Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) as 'AVG_in_kg' from SPACEXTABLE where Booster_Version LIKE 'F9 v1.1%'
```

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

| AVG_in_kg |
| --- |
| 2534.6666666666665 |

# First Successful Ground Landing Date

```
%sql select min(Date) as 'Date' from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
```

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

| Date |
|---|
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select Booster_Version from SPACEXTABLE where Landing_Outcome = 'Success (drone ship)' AND payload_mass__kg_ > 4000 AND payload_mass__kg_ < 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 count(*) as 'Total_Success' from SPACEXTABLE where Mission_Outcome = 'Success'
```

Python

* sqlite:///my_data1.db
Done.

| Total_Success |
| --- |
| 98 |

```
%sql select count(*) as 'Total_Failure' from SPACEXTABLE where Mission_Outcome LIKE 'Failure%'
```

Python

* sqlite:///my_data1.db
Done.

| Total_Failure |
| --- |
| 1 |

# Boosters Carried Maximum Payload

```
%sql select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

Python

```
 * 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

```
%sql select strftime('%m',Date) as 'Month', Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE where strftime('%Y',Date) = '2015' AND
Landing_Outcome LIKE 'Failure (drone ship)%'
```

Python

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

| Month | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 10 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

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

```
%sql select Landing_Outcome, count(*) as 'Count' from SPACEXTABLE where Date between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by count(*) desc
```

Python

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

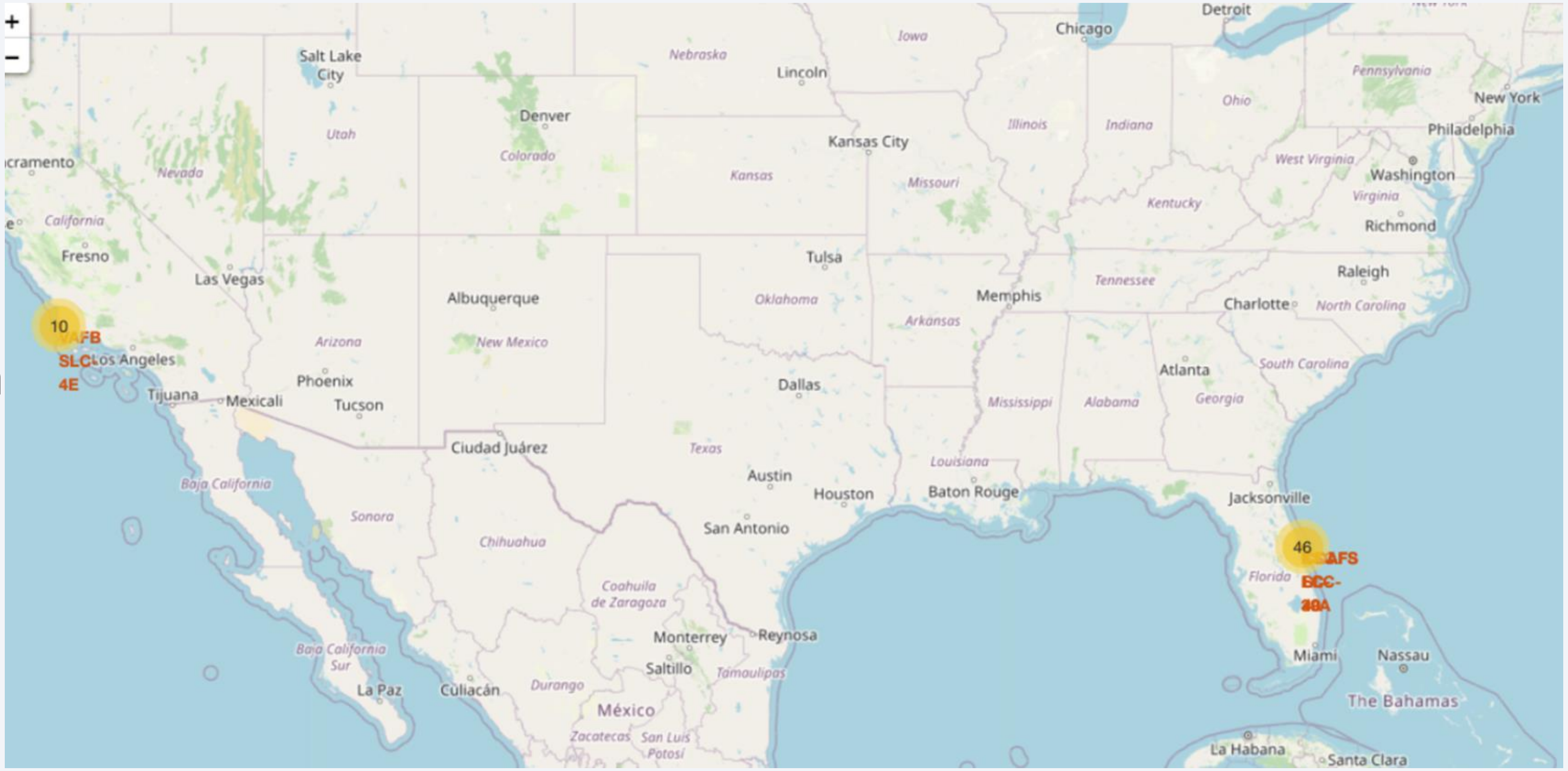
| Landing_Outcome | Count |
|---|---|
| No attempt | 10 |
| Success (ground pad) | 5 |
| Success (drone ship) | 5 |
| Failure (drone ship) | 5 |
| Controlled (ocean) | 3 |
| Uncontrolled (ocean) | 2 |
| Precluded (drone ship) | 1 |
| Failure (parachute) | 1 |

Section 3

# Launch Sites Proximities Analysis

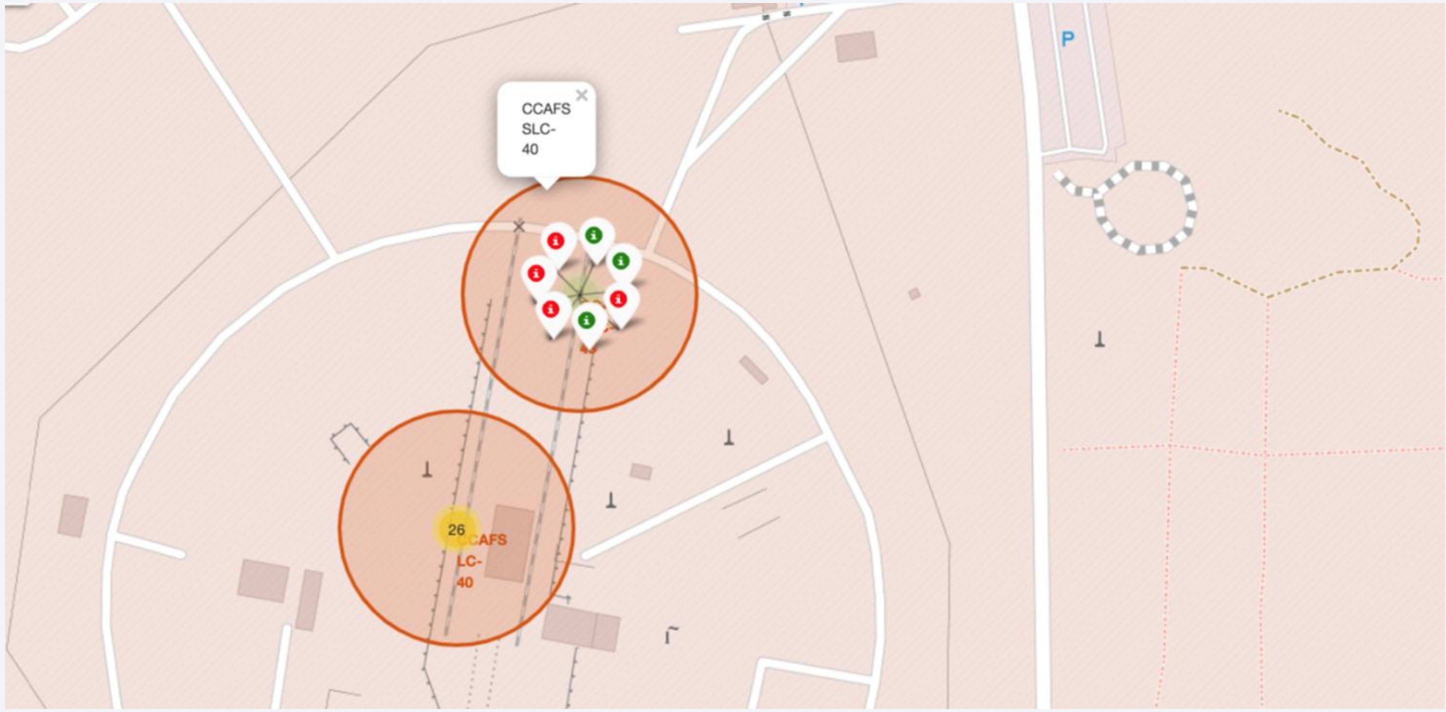
# Launch Sites

Launch sites are by the ocean



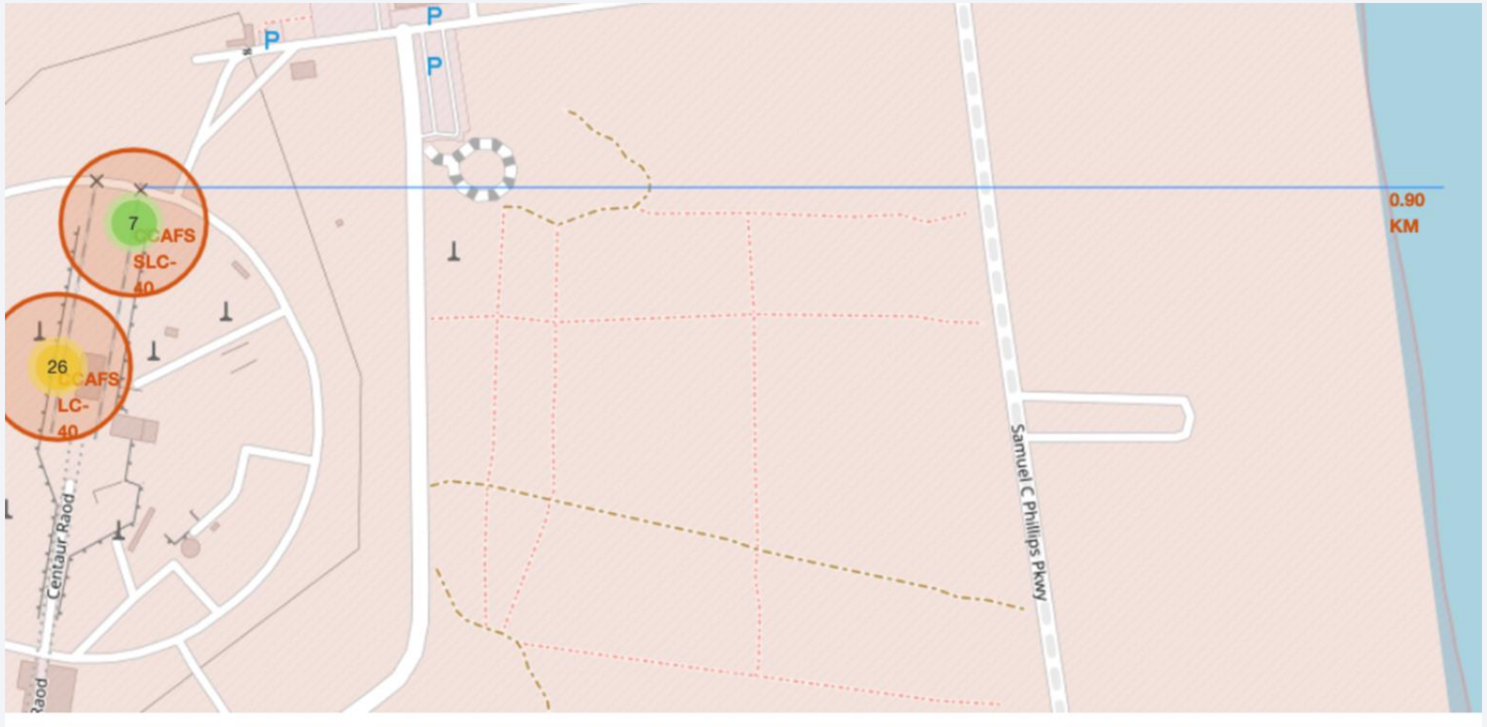
# Launch Site CCFS SLC-40

Red = landing unsuccessful
Green = successful landing



# AFS  Launch Sites

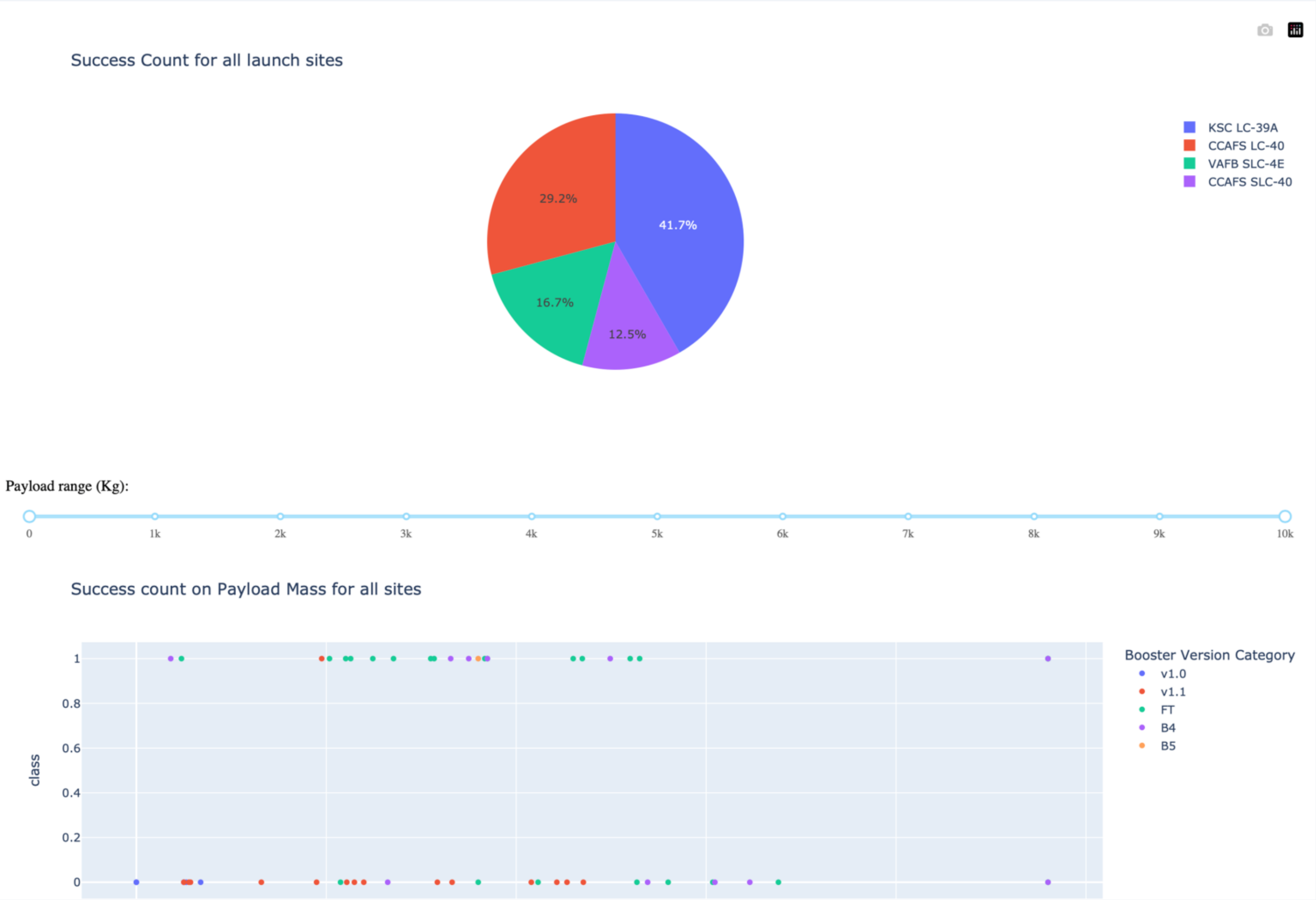
There are 7 landings in AFS SLC- 40
There are 26 landings at AFSLC - 40



Section 4

# Build a Dashboard with Plotly Dash

# Dashboard



Success Count for all launch sites
41.7%
29.2%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40
Payload range (Kg):
0
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
Success count on Payload Mass for all sites
class
1
0.8
0.6
0.4
0.2
0
Booster Version Category
v1.0
v1.1
FT
B4
B5

Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

```python
#find the best model
models = []
models.append(('Logistic Regression', logreg_cv))
models.append(('SVM', svm_cv))
models.append(('Decision Tree', tree_cv))
models.append(('KNN', knn_cv))

# evaluate each model in turn
results = []
names = []
for name, model in models:
    results.append(model.score(X_test, Y_test))
    names.append(name)
tr_split = pd.DataFrame({'Name': names, 'Score': results})

# print the table of scores sorted by descending accuracy
tr_split.sort_values(by='Score', ascending=False)
```

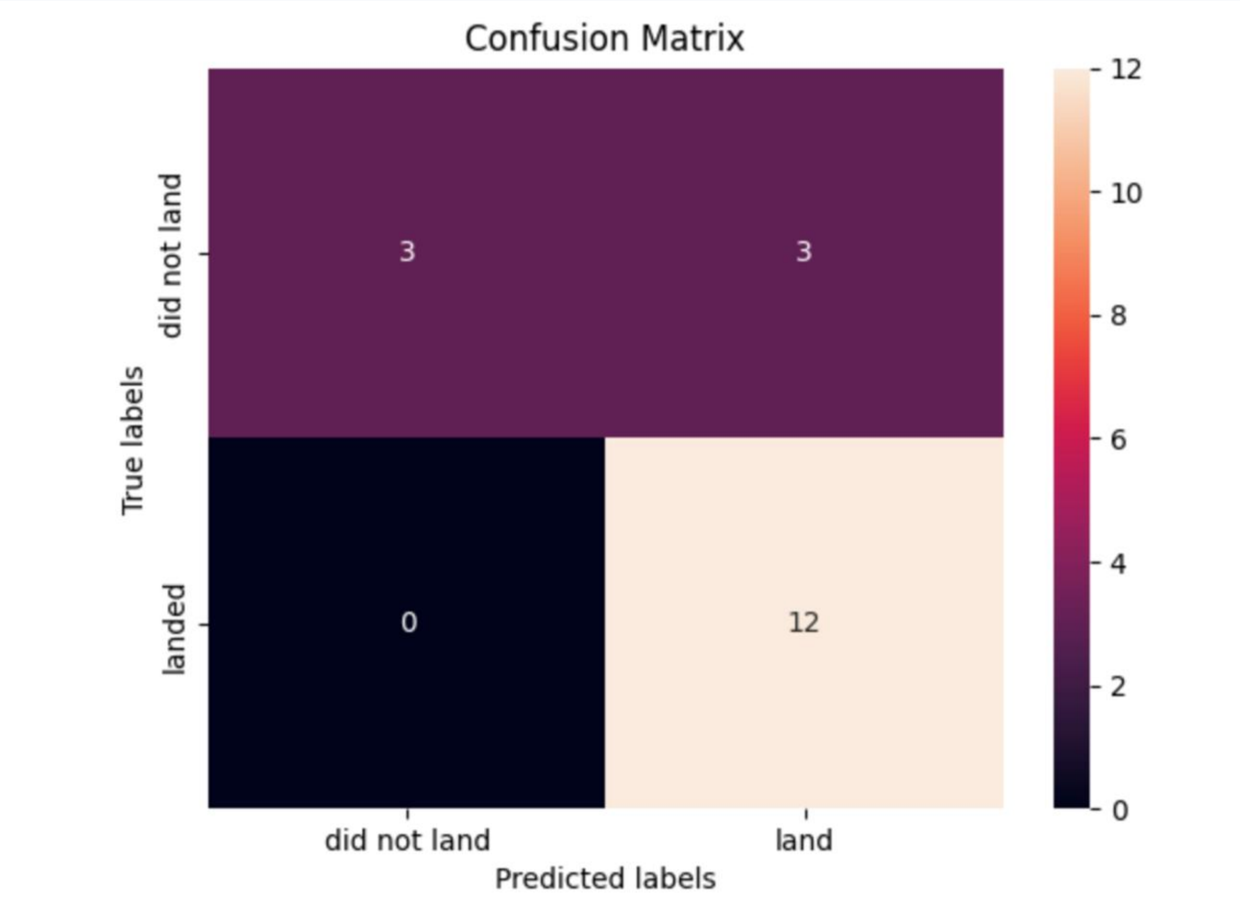
Python

| | Name | Score |
|---|---|---|
| 1 | SVM | 0.777778 |
| 3 | KNN | 0.777778 |
| 0 | Logistic Regression | 0.722222 |
| 2 | Decision Tree | 0.722222 |

Svm is the most Accurate Model

# Confusion Matrix

The only issue here is a small proportion of false positives, it is however the smallest one amongst all models



# Conclusions

- Landing site KSC-LC 39A has the most successful landings
- The SVM model is the best for predicting whether or not a spacecraft will land successfully or not
- Its best to do the positive landings closer to land, so that they are more easily reusable.

# Thank you!