IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Lwasampijja Baker
18th May 2025

Outline

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

Executive Summary

1. Summary of methodologies

- Data Collection API
- Data Collection with Web Scraping
- Data Wrangling
- Exploratory Data Analysis with Data Visualization
- Exploratory Data Analysis with SQL
- Interactive Visual Analytics with Folium
- Predictive Analysis with Machine Learning

2. Summary of all results

- Exploratory Data Analysis
- Interactive Analytics in Screenshots
- Predictive Analytics Results

Introduction

SpaceX advertises Falcon 9 rocket launches at a cost of $62 million—significantly lower than other providers, who charge upwards of $165 million per launch. A key reason for this cost advantage is SpaceX's ability to reuse the rocket's first stage.

Therefore, the success of a first-stage landing is a critical determinant of launch cost-efficiency. Accurately predicting whether the first stage will land successfully can provide valuable insights for potential competitors looking to bid against SpaceX in the commercial launch market.

The primary goal of this project is to develop a machine learning pipeline capable of predicting the success of Falcon 9 first-stage landings based on launch parameters and conditions.

Section 1

Methodology

Methodology

Executive Summary

- Data was collected from two main sources:
  - SpaceX API: https://api.spacexdata.com/v4/rockets/
  - Web scraping from Wikipedia: List of Falcon 9 and Falcon Heavy launches
- Performed data wrangling to clean and preprocess the dataset.
- Enriched the data by creating a landing outcome label based on outcome data after analyzing and summarizing features.
- Applied one-hot encoding to convert categorical features into a usable format for machine learning.
- Conducted exploratory data analysis (EDA) using data visualizations and SQL queries to uncover patterns and trends.
- Created interactive visual analytics using Folium and Plotly Dash to explore spatial and temporal aspects of the data.
- Built and evaluated predictive models using classification algorithms to forecast landing success.
- Performed hyperparameter tuning to optimize model performance.

Data Collection

Sources: SpaceX API and Wikipedia launch history page.

Methods: Retrieved structured data via API requests; used web scraping (Python: BeautifulSoup, requests) for tabular data from Wikipedia.

Integration: Merged both sources using common fields like launch date and mission ID.

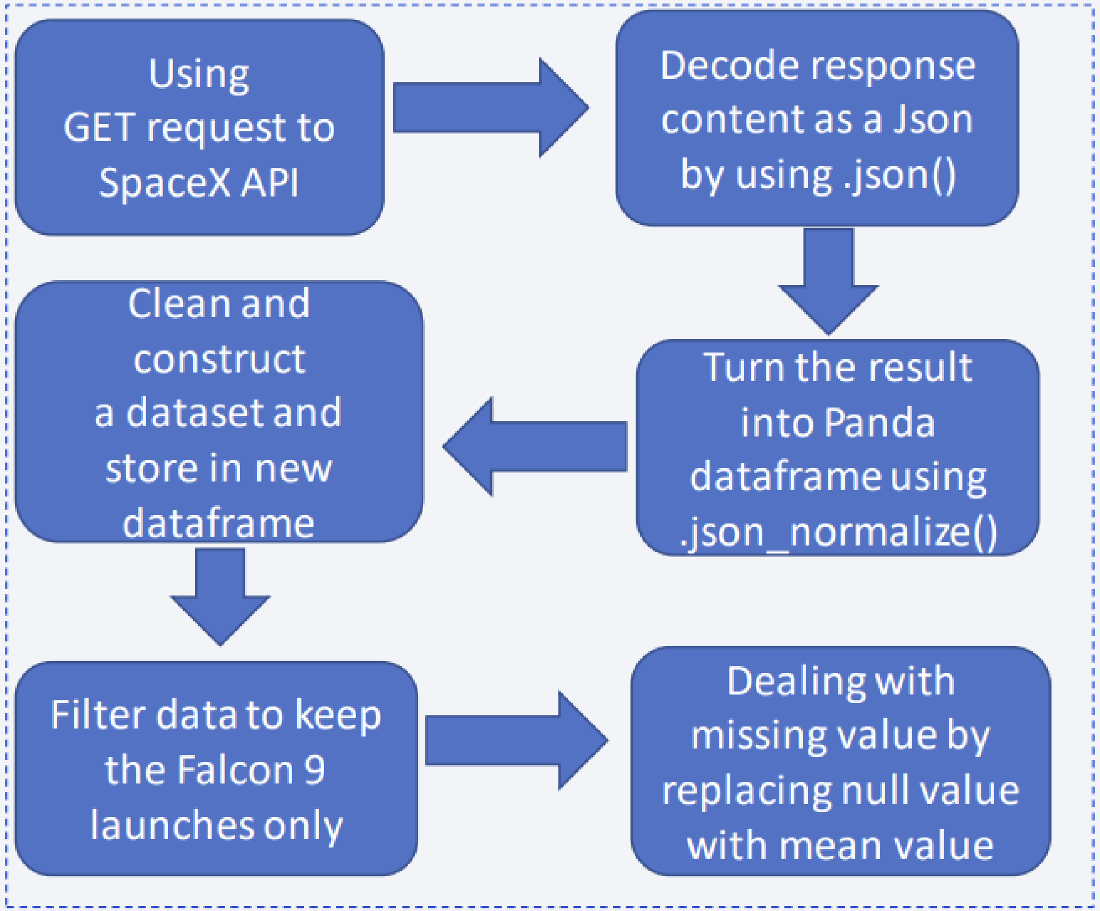
Cleaning & Validation: Ensured consistency, removed duplicates, and enriched records with outcome labels.

Output: Final dataset saved in CSV format, ready for analysis.

Data Collection – SpaceX API

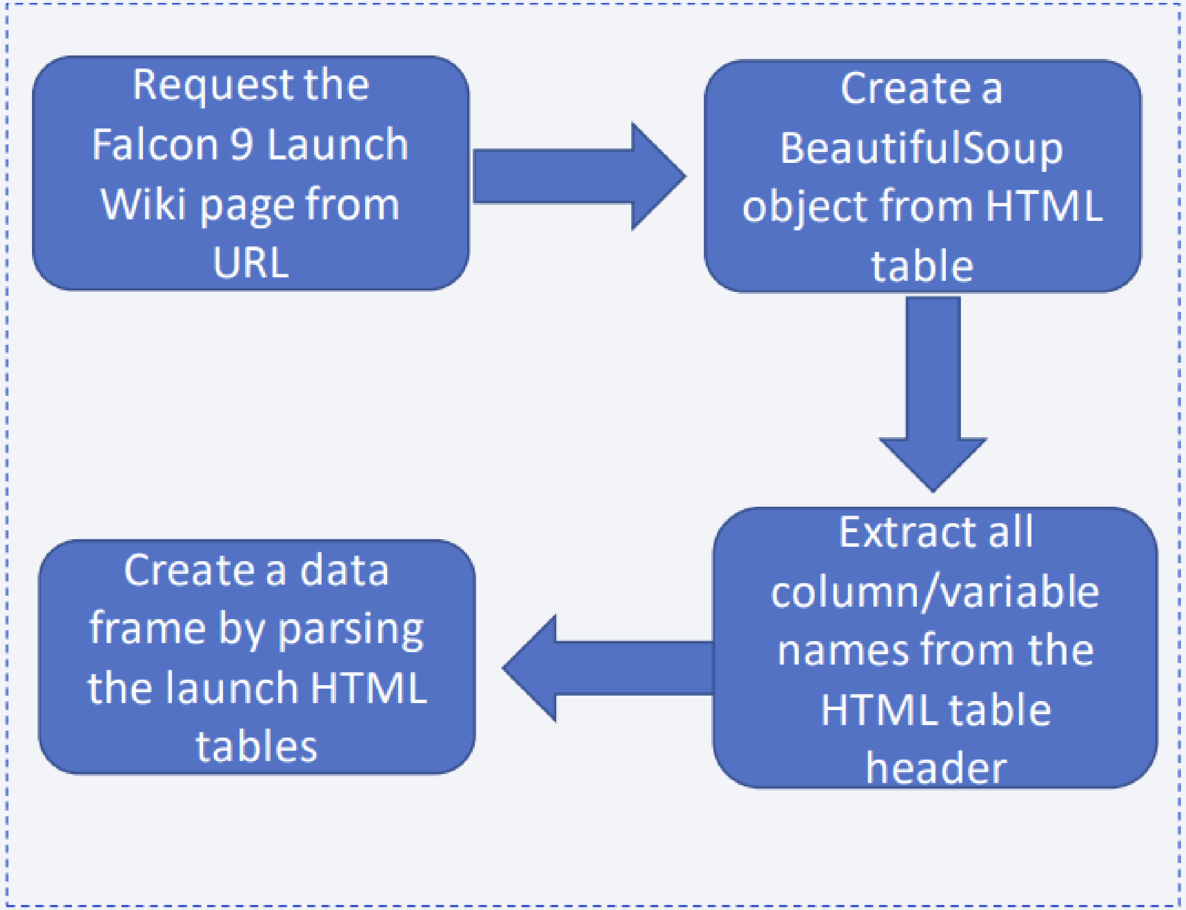
- SpaceX offers a public API from where data can be obtained and then used;
- This API was used according to the flowchart beside and then data is persisted.

- https://github.com/baker371/capstone/blob/main/jupyter-labs-webscraping.ipynb



Data Collection - Scraping

- Data from SpaceX launches can also be obtained from Wikipedia;
- Data are downloaded from Wikipedia according to the flowchart and then persisted.
- https://github.com/baker371/capstone/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Exploratory data analysis was conducted and determined the training labels.
- I calculated the number of launches at each site, and the number and occurrence of each orbits.
- I created landing outcome label from outcome column and exported the results to csv.
- https://github.com/baker371/capstone/blob/main/labs-jupyter-spacex-Data%20wrangling-v2.ipynb

EDA with Data Visualization

- I explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
- https://github.com/baker371/capstone/blob/main/jupyter-labs-eda-dataviz-v2.ipynb

EDA with SQL

- To explore data, scatterplots and bar plots were used to visualize the relationship between pair of features including
- The name of unique launch sites in the space mission.
- The total payload mass carried by boosters launched by NASA (CRS)
- The average payload mass carried by booster version F9 V1.1
- The total number of successful and failure mission outcomes
- The failed landing outcomes in drone ship, their booster version and launch site names.
- https://github.com/baker371/capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Markers, circles, lines and marker clusters were used with Folium Maps
- Markers indicate points like launch sites;
- Circles indicate highlighted areas around specific coordinates, like NASA Johnson Space Center;
- Marker clusters indicates groups of events in each coordinate, like launches in a launch site; and
- Lines are used to indicate distances between two coordinates.
- https://github.com/baker371/capstone/blob/main/lab_jupyter_launch_site_location_v2.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- https://github.com/baker371/capstone/blob/main/plotly_dash.py

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- https://github.com/baker371/capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5_v1.ipynb

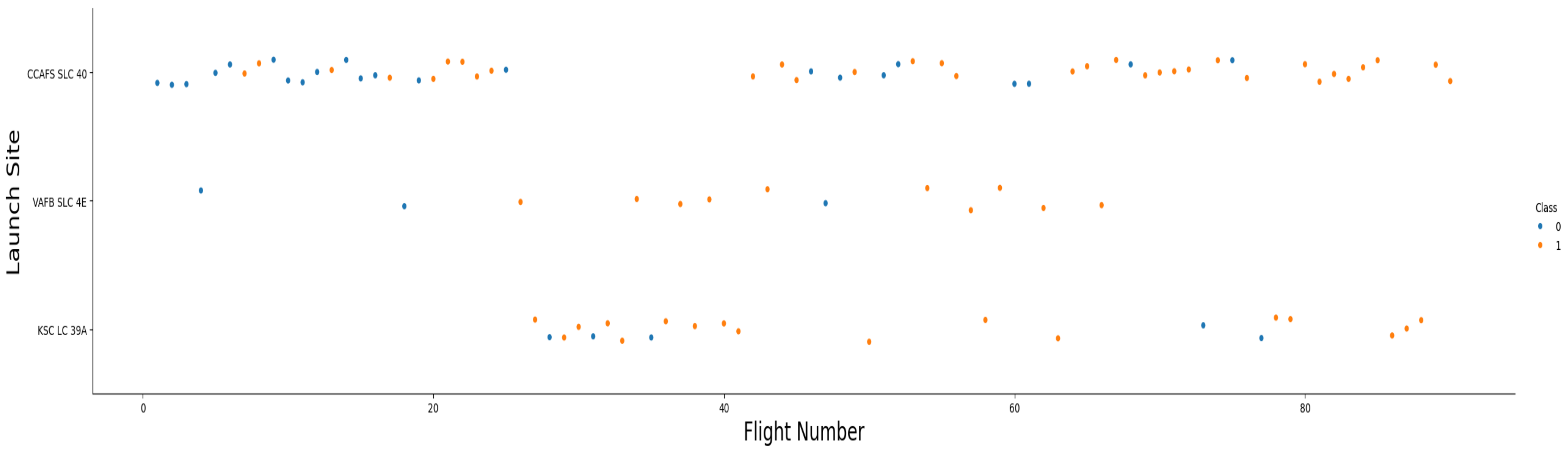
Results

- Exploratory data analysis results:
- Space X uses 4 different launch sites;
- The first launches were done to Space X itself and NASA;
- The average payload of F9 v1.1 booster is 2,928 kg;
- The first success landing outcome happened in 2015 fiver year after the first launch;
- Many Falcon 9 booster versions were successful at landing in drone ships having payload above the average;
- Almost 100% of mission outcomes were successful;
- Two booster versions failed at landing in drone ships in 2015: F9 v1.1 B1012 and F9 v1.1 B1015;
- The number of landing outcomes became as better as years passed.

Section 2

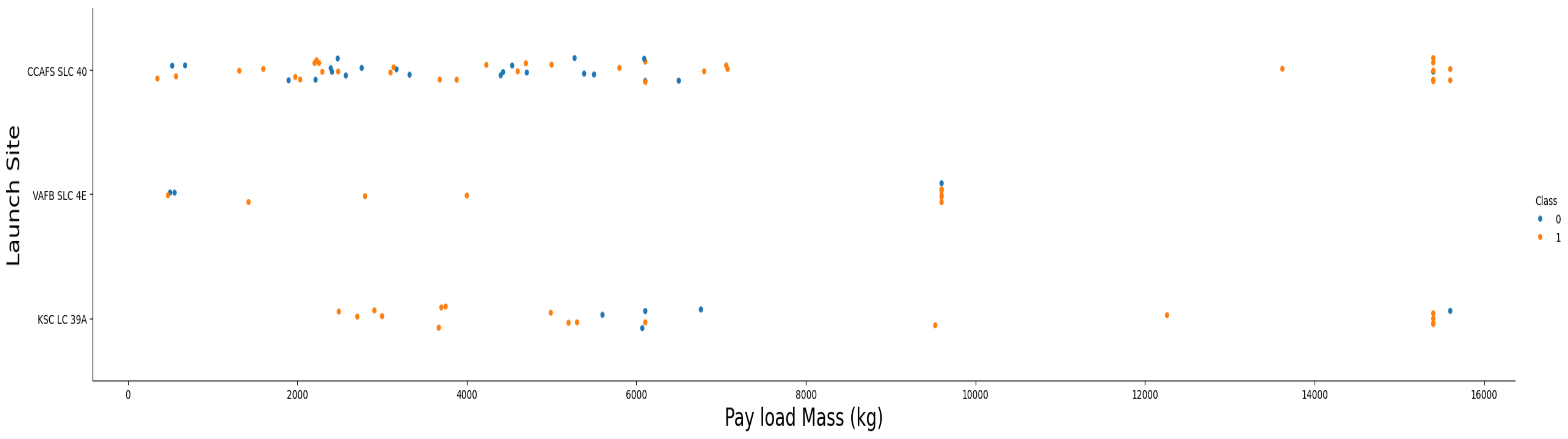
Insights drawn from EDA

Flight Number vs. Launch Site



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

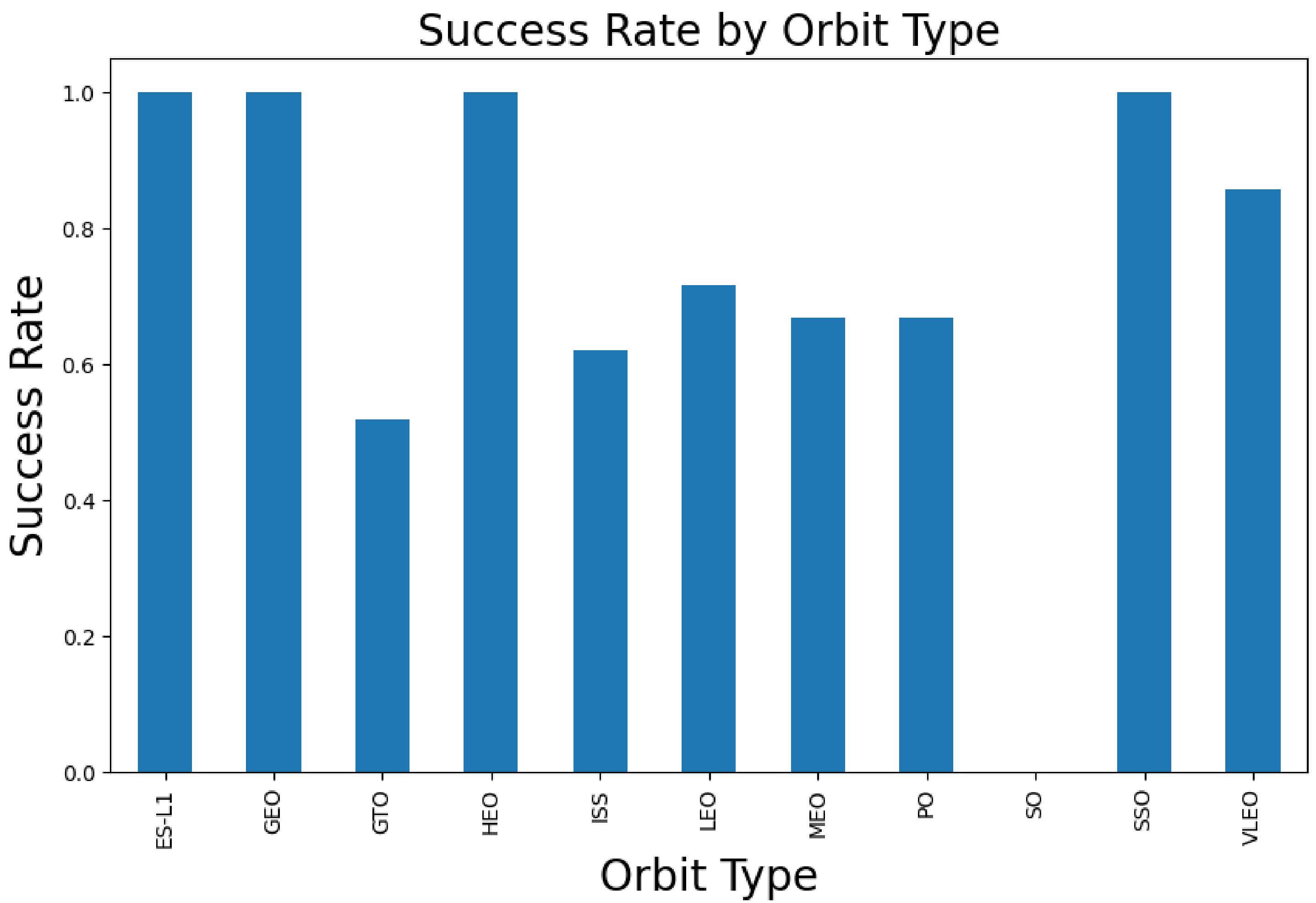
Payload vs. Launch Site



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

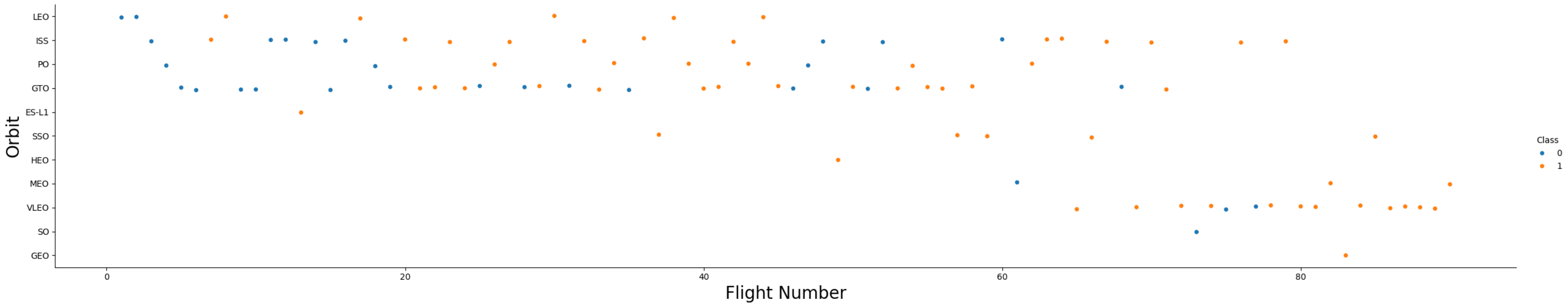
Success Rate vs. Orbit Type

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



Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe 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.

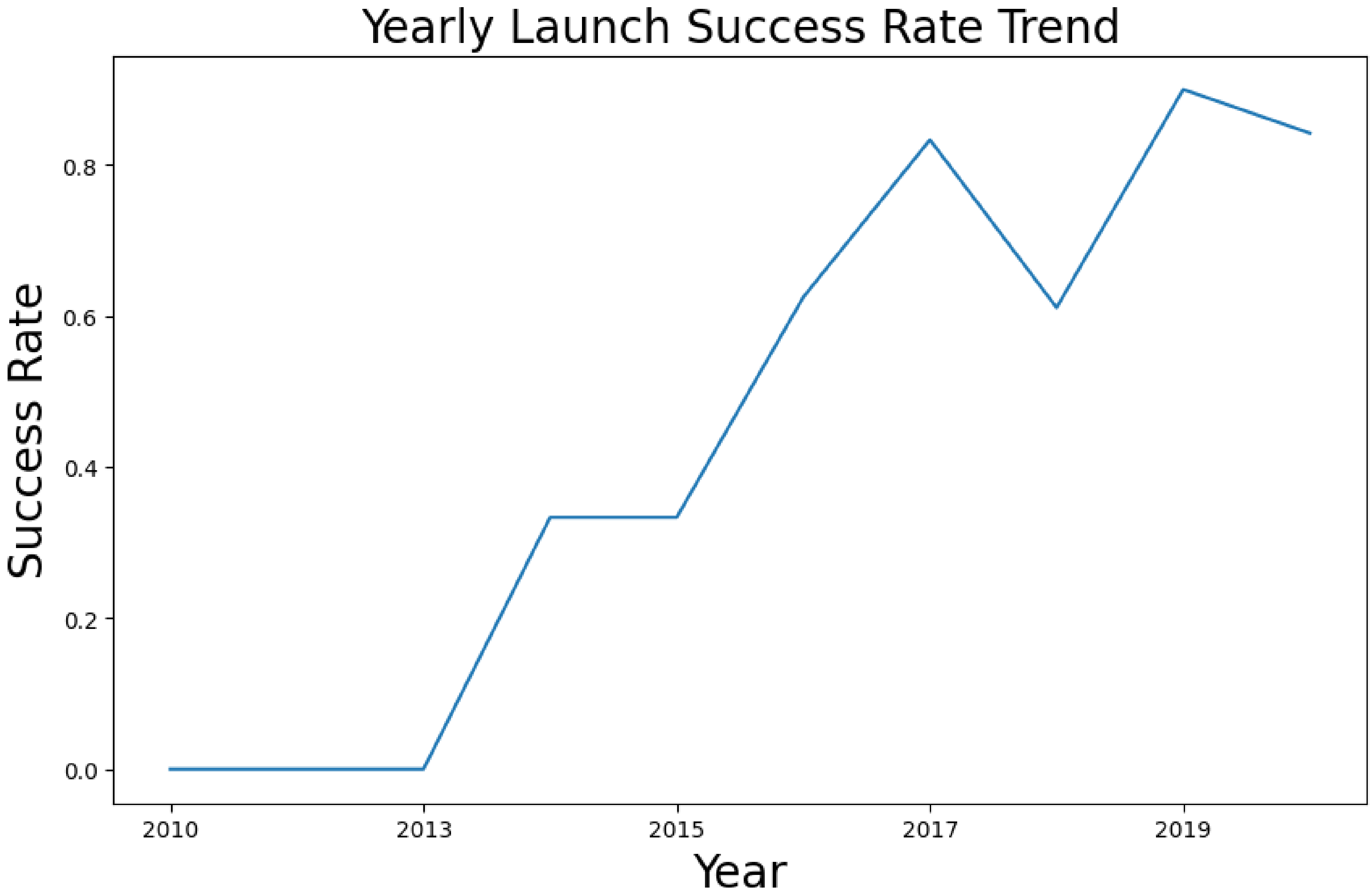


Payload vs. Orbit Type

- 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 rat since 2013 kept on increasing till 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 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
```

```
 * 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

- We used the query above to display 5 records where launch sites begin with \`CCA\`

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below

```
[ ] %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

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

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

AVG(PAYLOAD_MASS__KG_)
2928.4

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

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 SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)'
```

```
 * 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 FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

```
 * 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

- We used count and group by.

```
[ ] %sql SELECT Mission_Outcome, COUNT(*) FROM SPACEXTABLE GROUP BY Mission_Outcome
```

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

Mission_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 FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

```
 * 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

- 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, "Landing_Outcome", Booster_Version, Launch_Site FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Failure (drone ship)' AND substr(Date, 0, 5) = '2015'

 * 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 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 to 2010-03-20.
- 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 OutcomeCount FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing_Outcome" ORDER BY OutcomeCount DESC
```

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

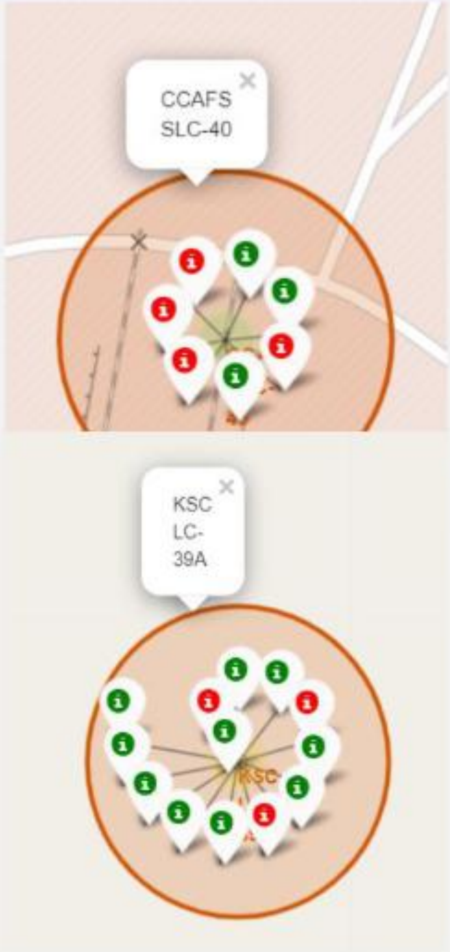
Landing_Outcome	OutcomeCount
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 4

Build a Dashboard with Plotly Dash

All launch sites global map markers

Markers showing launch sites with color labels

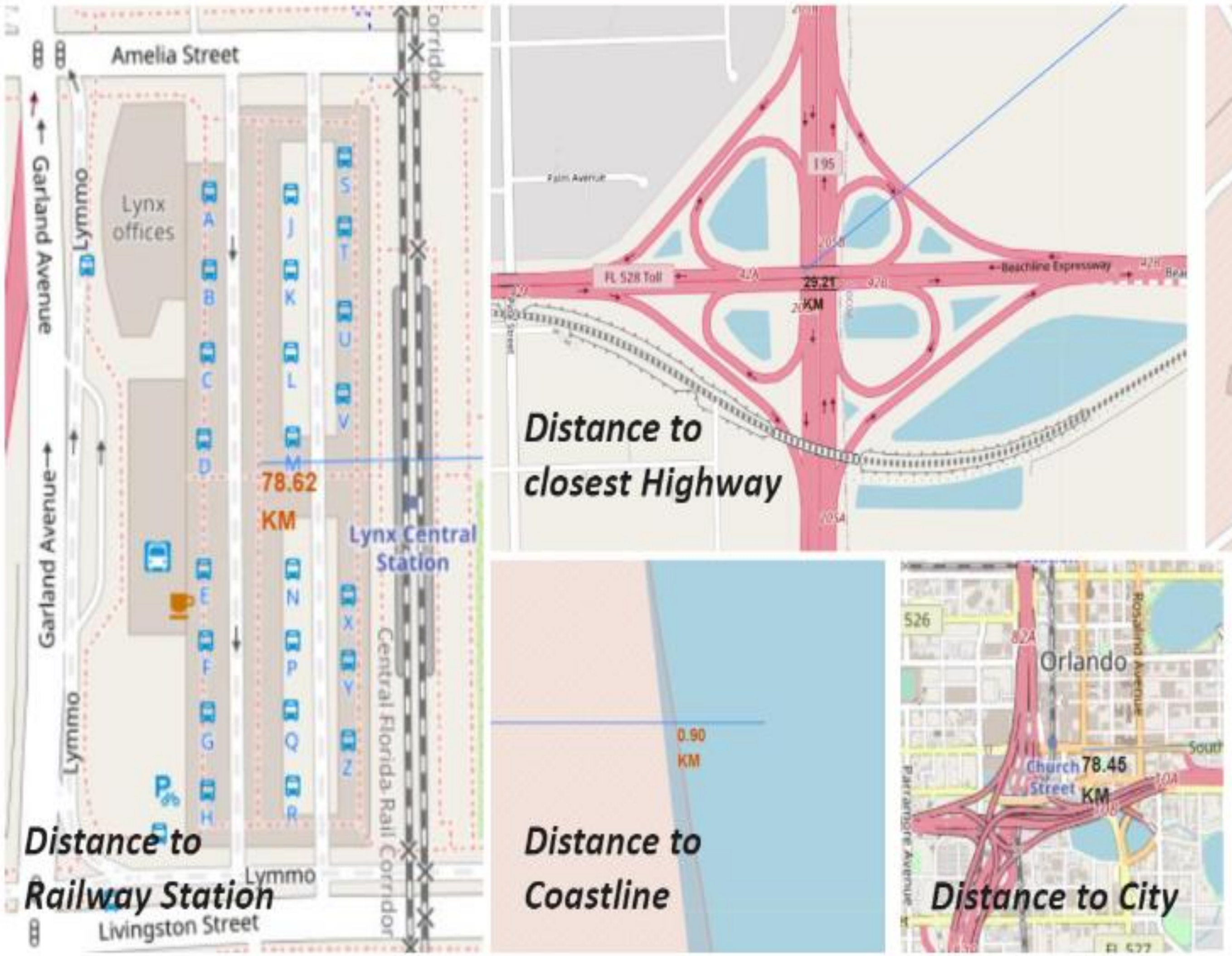


Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

Launch Site distance to landmarks

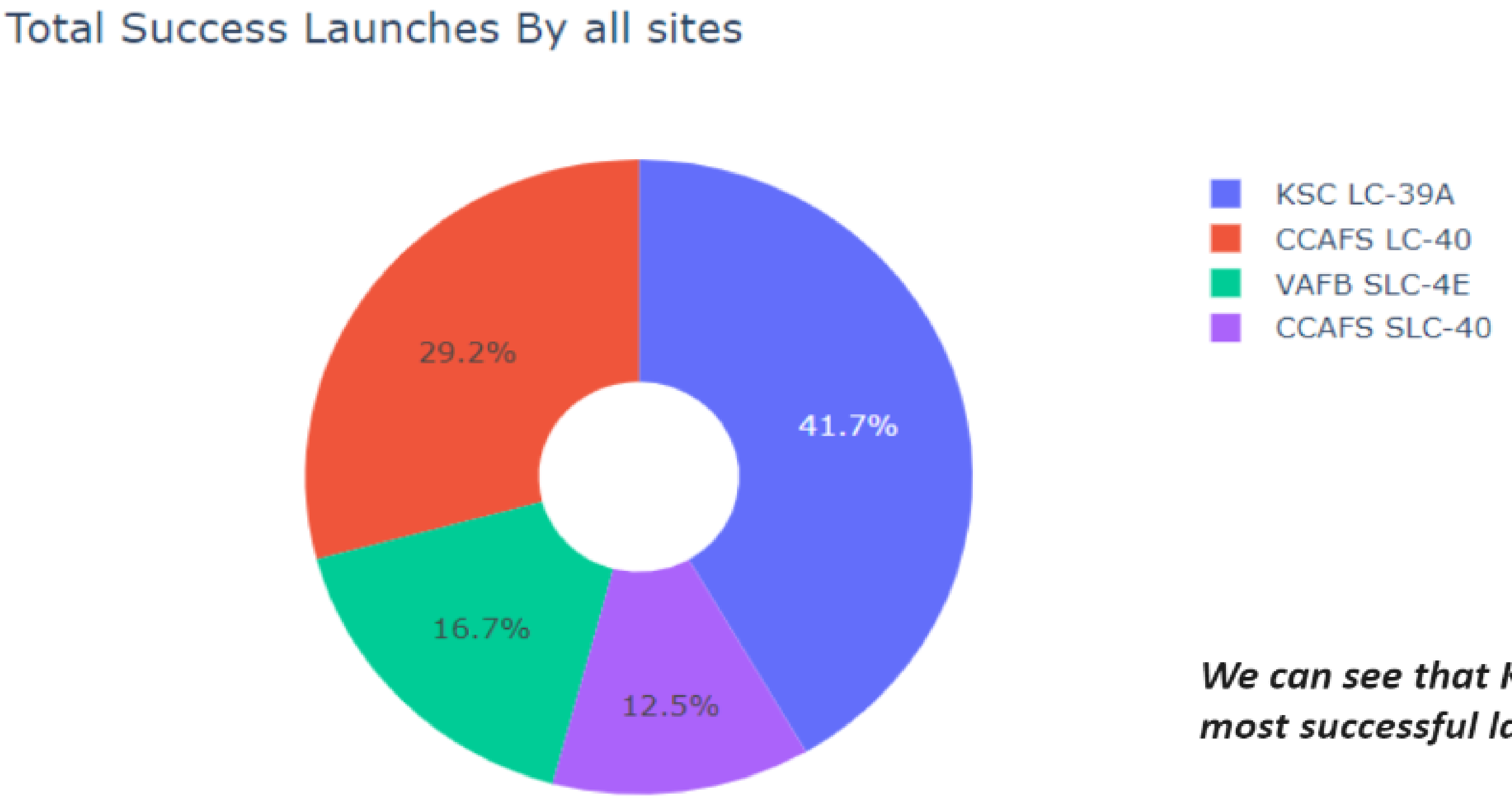


•Are launch sites in close proximity to railways? No
•Are launch sites in close proximity to highways? No
•Are launch sites in close proximity to coastline? Yes
•Do launch sites keep certain distance away from cities? Yes

Section 4

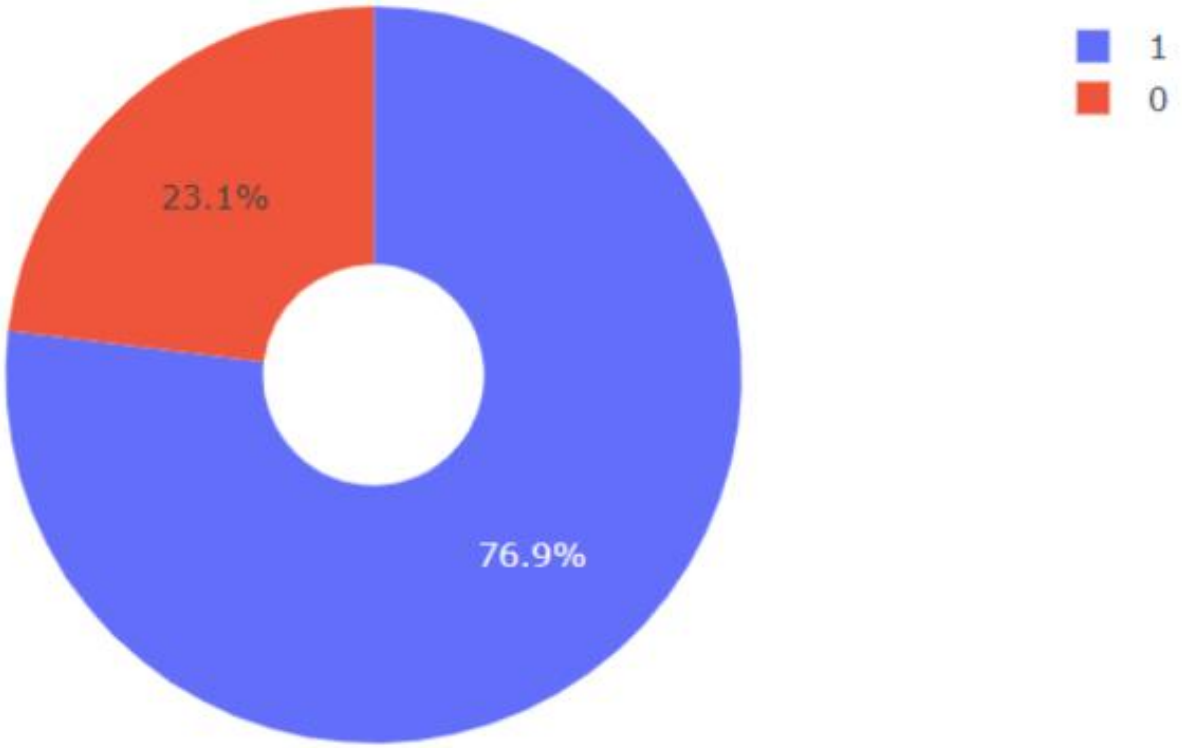
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



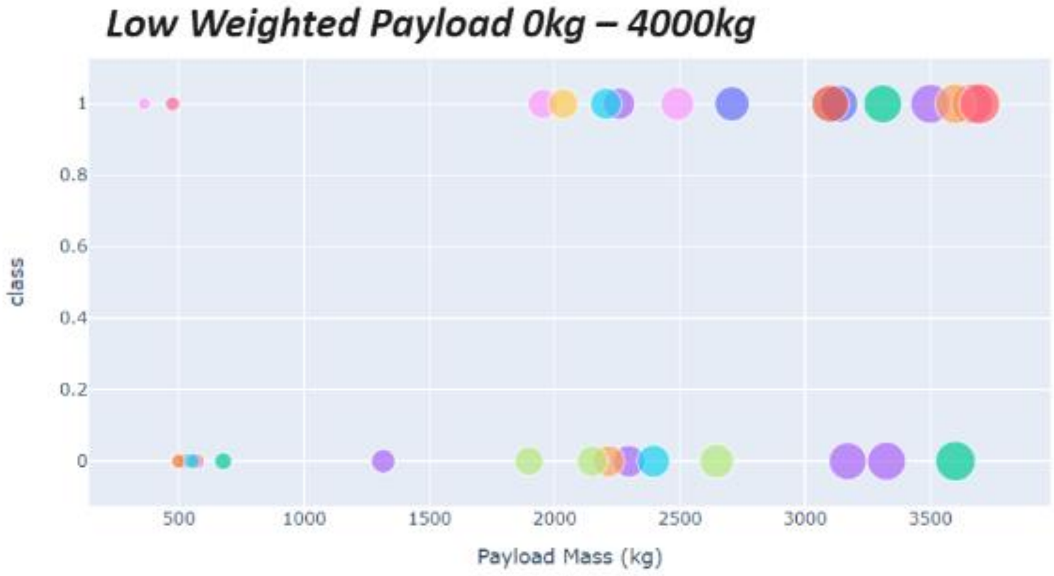
We can see that KSC LC-39A had the most successful launches from all the sites

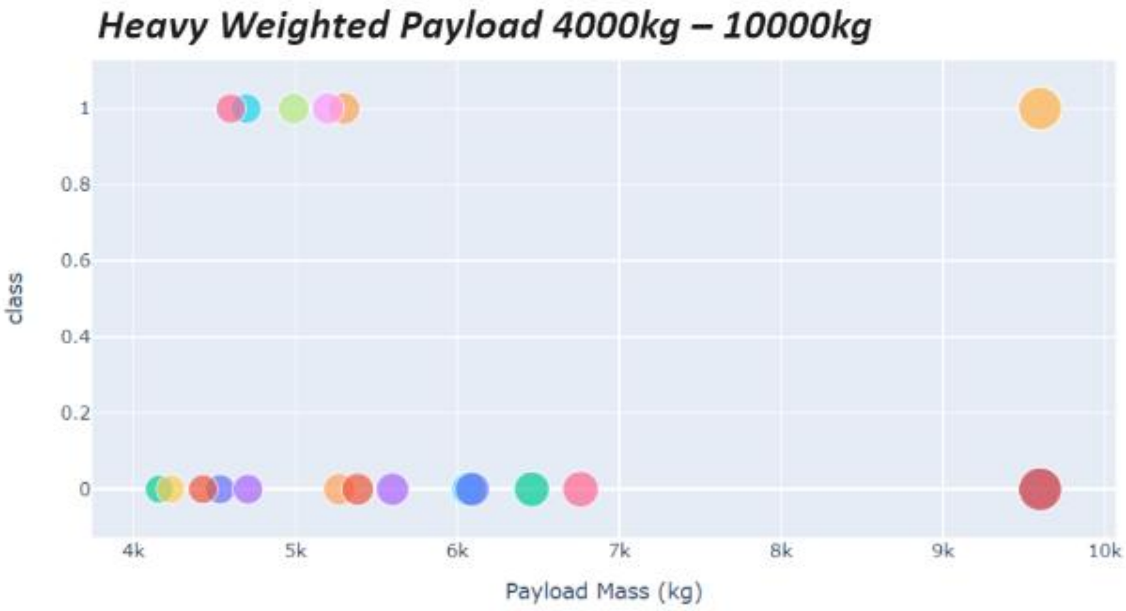
Pie chart showing the Launch site with the highest launch success ratio



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 rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy

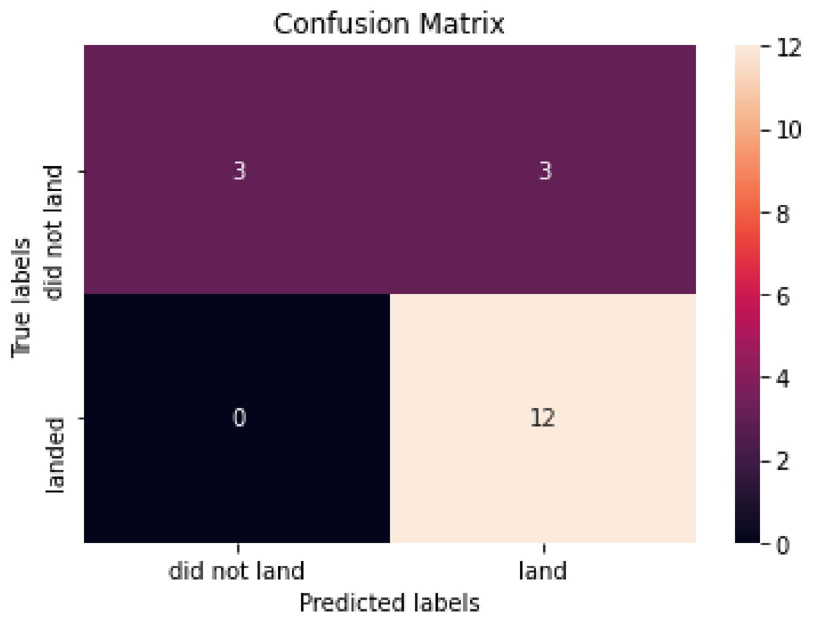
```
[ ]  scores = {'Logistic Regression': logreg_cv.score(X_test, Y_test),
               'SVM': svm_cv.score(X_test, Y_test),
               'Decision Tree': tree_cv.score(X_test, Y_test),
               'KNN': knn_cv.score(X_test, Y_test)}

     best_method = max(scores, key=scores.get)
     print(f"The method with the best accuracy on the test data is {best_method} with an accuracy of {scores[best_method]:.4f}")
```

```
The method with the best accuracy on the test data is Logistic Regression with an accuracy of 0.8333
```

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 .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.