IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Promise Nwabueze Igbojionu
29th June 2024

Outline

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

Methodology Summary

1. Objective: The objective of this project was to determine the cost of SpaceX Falcon 9 launches by predicting if the first stage will be successfully recovered for reuse. Several tools were used for analysis and visualization

2. Data Collection: Historical data was collected from the SpaceX Falcon 9 launch through API calls and web scrapings, including outcomes of first stage recovery, mission parameters, and some cost information.

3. Dashboard Development: Interactive dashboards was created to visualize how each feature from the data collected would give predictive insights immediately for Space Y's strategic decision-making.

4. Machine Learning Model: Train a predictive model using machine learning algorithms to forecast the likelihood of first stage reuse based on collected data.

Result Summary

In the analysis of SpaceX Falcon 9 launches, historical data indicates successful recovery of the first stage in majority of missions based on improvement in key factors such as payload, orbit and other requirements. Additionally, a machine learning model was developed and achieved an accuracy of 83.33% demonstrating its effectiveness in predicting the first stage reuse. In general, Booster version FT category had the most successful launches regardless of payload mass while V1.1 was the least successful

Introduction

- **Project background and context**

The project focuses on analyzing SpaceX Falcon 9 rocket launches, exploring the feasibility and implications of first stage recovery. SpaceX has revolutionized space travel with its ability to recover and reuse the first stage of Falcon 9 rockets, significantly reducing launch costs compared to traditional methods. This project also focuses on using machine learning models to predict whether the first stage will land successfully which is vital to planning, budgeting and the commercialization and accessibility of space.

- **Problems to Address**

The project seeks to tackle three main questions:

- What is SpaceX Falcon 9 first stage recovery success rates and key factors that influence these rates?
- How can we generate actionable insights for decision-making in the space industry?
- What are the cost Estimation for launches for Space Y based on predicted first stage recovery success?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

The SpaceX REST API is called with using the requests library - .get()method.

The response object was turned into a pandas dataframe using the .json_normalize() method.

Important features were extracted such as Booster Version, Orbit, Payload Mass etc.

A new dataframe object with headers and rows was constructed with the extracted details and missing values were handled.
[SpaceX-API Calls GIthub Link]()

Install and import libraries → Call API → Create Data-Frame with response → Extract Details → Add headers(Columns) → Filter to have only Falcon 9 launches → Handle missing data → Store data in a CSV format

Data Collection - Scraping

- Install requests library and import
- Wikipedia page was scraped for tables using the .get() method
- Column headers were extracted from tables with predefined functions
- Dataframe created with the extracted Data.
- Github - Webscraping link

Scraped Wikipedia page for HTML Tables → Parse response with BeautifulSoup and extracted column names from table headers → Created dictionary using table headers as column headers → Created dataframe from the dictionary → Saved in a CSV file

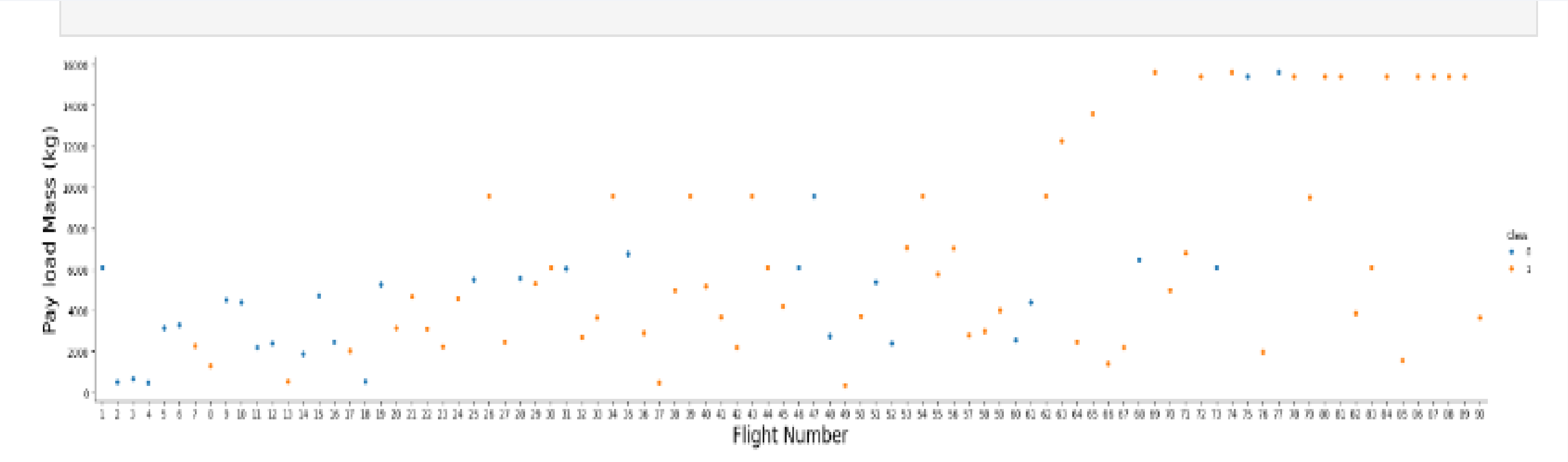
Data Wrangling

- Description of data pre-processing
  - Using the .value_counts() method, the number of launches for each launch site, orbit and landing outcome was counted.
  - Each attribute or column with missing values were identified, and the percentage of missing data were calculated
  - The data type for each attribute was identified
  - The landing outcome were categorized (dummy-coded) into 1 and o for successful and unsuccessful outcomes

Launch site count → Attributes with missing values identified → Data types of each column was shown → Landing outcomes dummy-coded to 0 or 1 depending on outcome

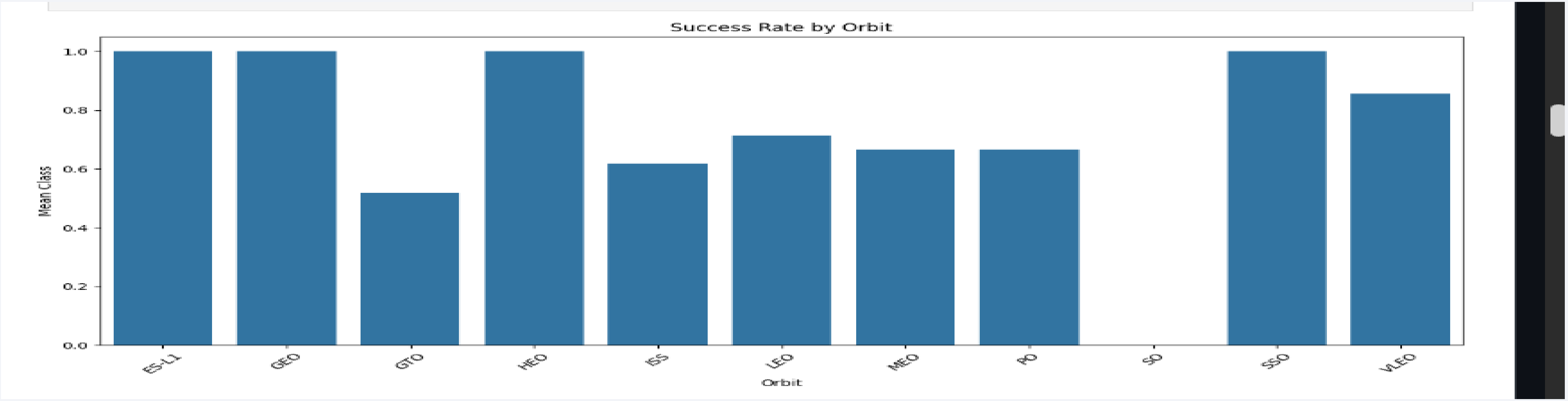
- [Github - Data Wrangling]

EDA with Data Visualization



➢This chart shows the relationship between flight number and payload mass and how it translated to mission outcome. From the chart, we can see that as flight number increased with a steady increase in payload mass, the chances for a successful landing increased.

EDA with Data Visualization



➢This chart shows the relationship between the launches made to different orbits and the success rate calculated using the mean. From the chart, we can see that launches made to ES-L1, GEO, HEO and SSO were all successful and the other orbits had some unsuccessful mission outcomes.

- [Github link - EDA with charts]

EDA with SQL

- Some SQL queries performed includes:
  - Unique launch sites - %sql SELECT DISTINCT “Launch_Site” FROM SPACEXTABLE;
  - Records of launch sites beginning with ‘CCA’ - %sql SELECT * FROM SPACEXTABLE WHERE “Launch_Site LIKE ‘CCA%’ LIMIT 5;
  - Total payload mass launched by NASA (CRS) - %sql SELECT SUM("PAYLOAD_MASS__KG_") AS "TOTAL PayLoad Mass" FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)’;
  - Average payload mass carried by booster version F9 v1.1 - %sql SELECT AVG("PAYLOAD_MASS__KG_") AS "Average PayLoad Mass" FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1’;
  - Dates of first successful landing outcome in ground pad - %sql SELECT * FROM SPACEXTABLE WHERE "Date" = (SELECT MIN("Date") FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)’);
- Github link - EDA with SQL

Build an Interactive Map with Folium

- Summary of map objects such as markers, circles, lines, etc. added to a folium map
  - A map of the NASA Space station was created, with circles and markers indicating launch sites. The outcome of each launch—red for failure and green for success—is also displayed for these sites.
- Reason for these objects
  - These objects were added to help us determine whether location played a role in the success or failure of the landings. We can also observe that launch sites are situated far away from infrastructure and cities.
- Github link - Launch site locations with folium

Build a Dashboard with Plotly Dash

- Features of the interactive dashboard
  - Dropdown of all launch sites from which specific launch site or all options can be selected
  - Pie chart that outputs data based on dropdown selection
  - Range slider that displays a range of payload mass that user can select
  - Scatter plot showing changes in mission outcome according to the payload mass selected
- Reason for the plots and interactions
  - These interactive features helps us to see at a glance the mission outcomes of specific launch site or the ratio of impact from all launch sites put together
  - They also help us to see at a glance how each payload mass can affect mission outcome and help us decide the payload mass that has the greatest impact just from visuals
  - Github Link - Dash app

Predictive Analysis (Classification)

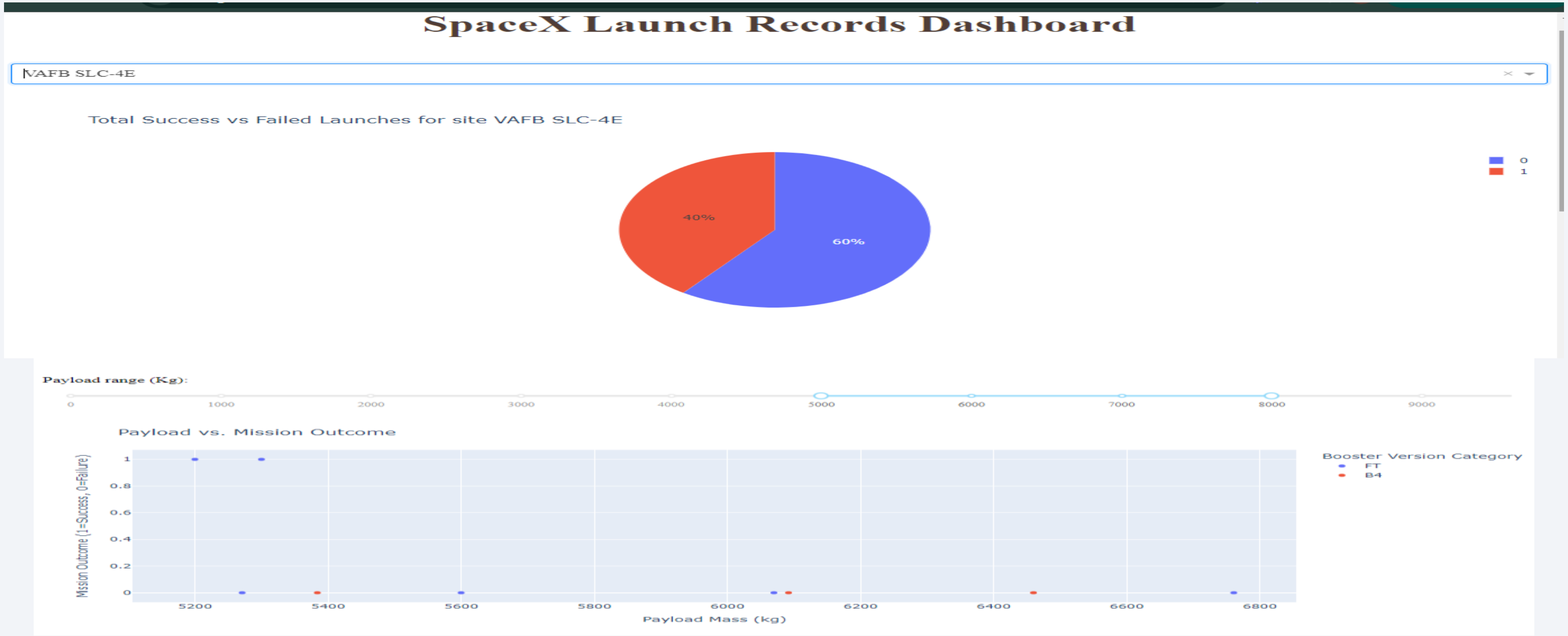
- Classification model summary
  - The data was split into training and testing datasets and then fitted with Logistic Regression, Support Vector Machine, Decision Tree, and k-Nearest Neighbor models
  - The Decision Tree Model performed the best
- You need present your model development process using key phrases and flowchart
  - Split the data using train-test/split method → created a classification model object and parameters dictionary → Created a GridSearch Object with the parameters and object created and fitting the object to get the best performing parameters and accuracy → apply the best parameters on the test set plot and the confusion matrix
  - GitHub link - Machine Learning Prediction

Results

- Exploratory data analysis results
  - Launches with payload mass in KG greater than 10,000 were likely to result in failure
  - ES-L1, GEO, HEO and SSO were orbits with the best success rates.
- Interactive analytics demo in screenshots – Next page
- Predictive analysis results
  - The best classifier is: Decision Tree
  - Best score: 0.8767857142857143
  - Best parameters: {'criterion': 'gini', 'max_depth': 12, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}

Results

- Interactive analytics demo in screenshots

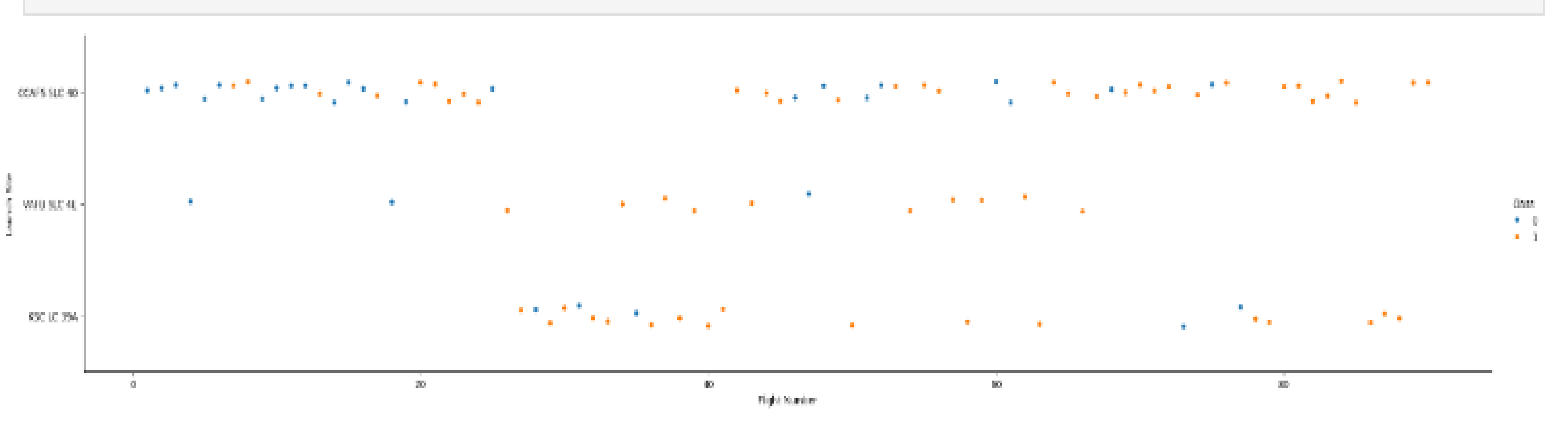


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

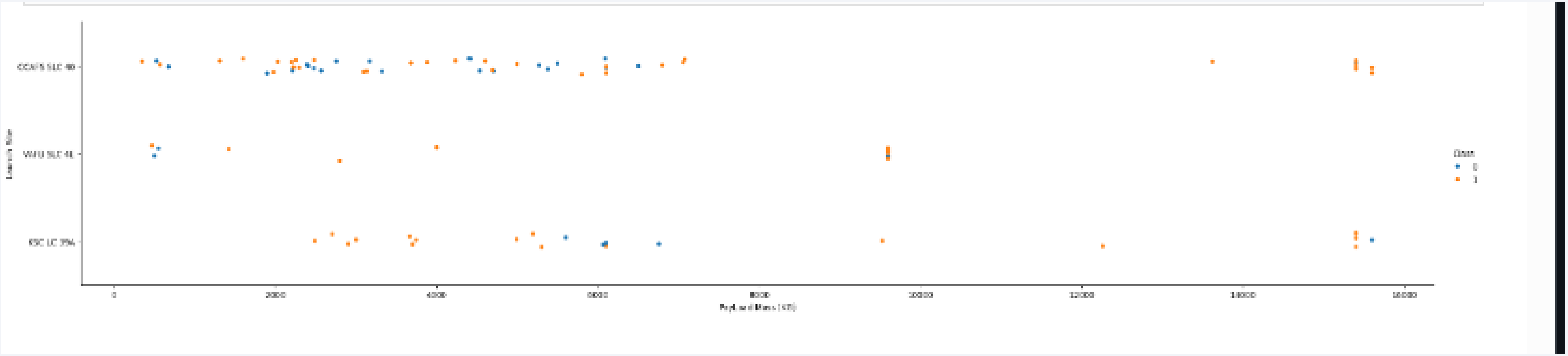
- Scatter plot of Flight Number vs. Launch Site



- The chart indicates that CFAFS SLC-40 achieved more successful outcomes after the 20th launch attempt. In contrast, other launch sites had minimal or no attempts below the 20th flight. The improved success rate above the 20th attempt by the other launch sites may be attributed to error corrections based on lessons learned from CFAFS SLC-40's earlier unsuccessful launches.

Payload vs. Launch Site

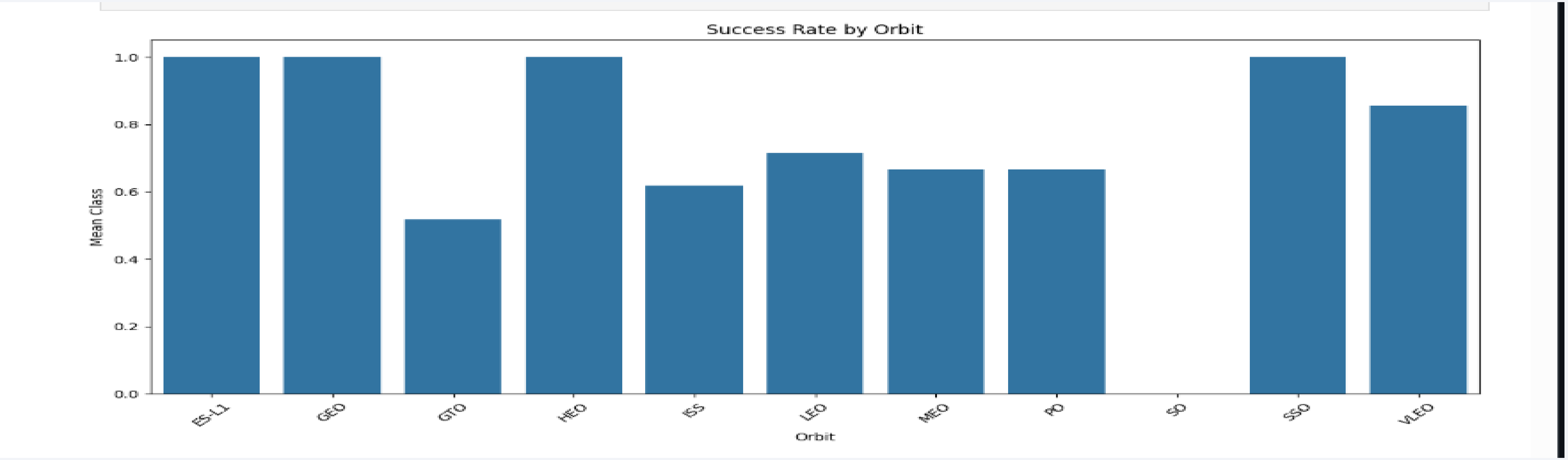
- Scatter plot of Payload vs. Launch Site



- Payload masses greater than 10,000 kg decrease the chances of success, regardless of the launch site.

Success Rate vs. Orbit Type

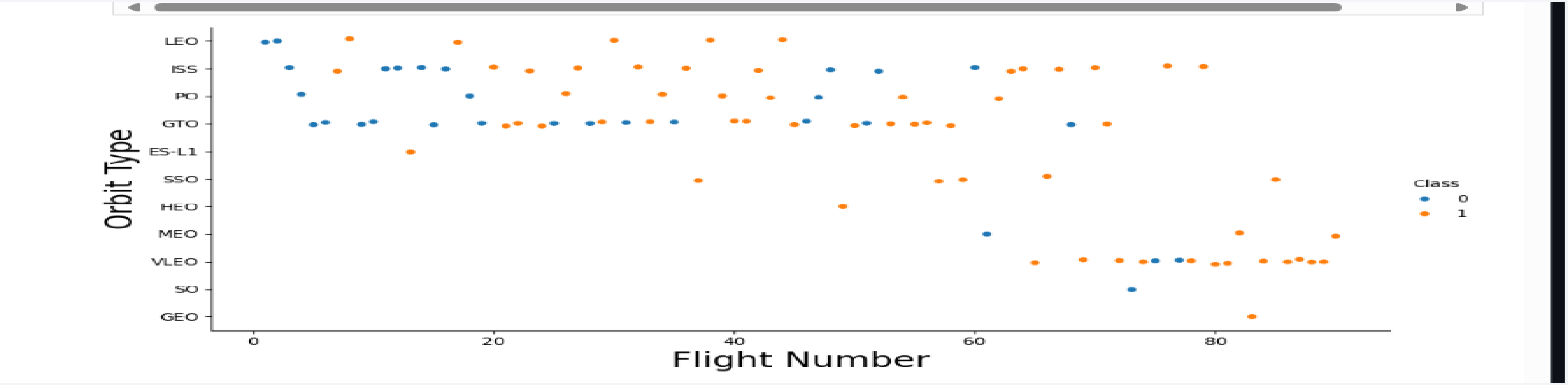
- Bar chart for the success rate of each orbit type



- . From the chart, we can see that launches made to ES-L1, GEO, HEO and SSO were all successful and the other orbits had some unsuccessful mission outcomes with GTO having at least 50% and SO not having any success rate.

Flight Number vs. Orbit Type

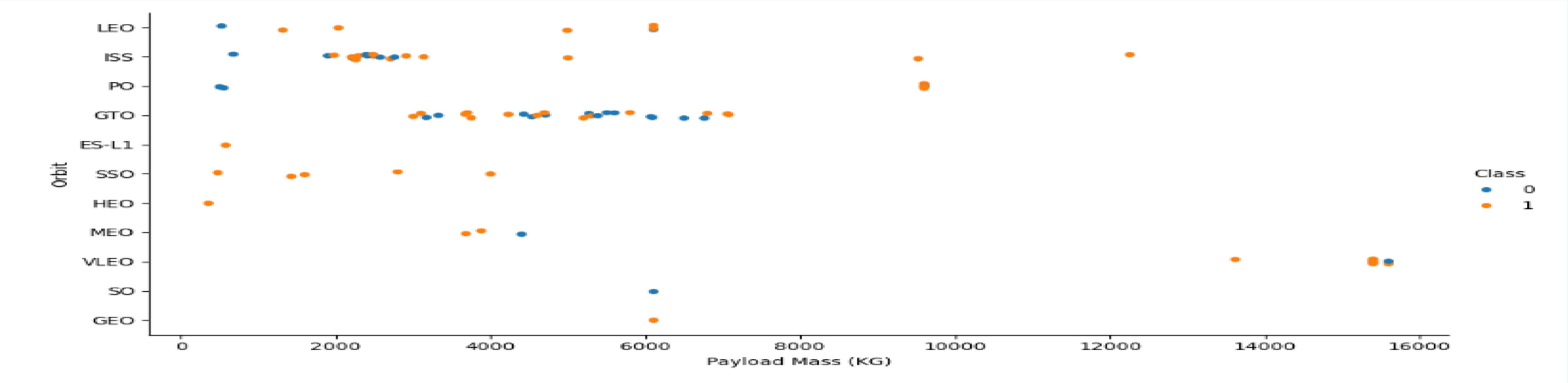
- Scatter point of Flight number vs. Orbit type



For some orbits, as flight number increased, the success rate increased while for others like the GTO orbit, they seem not to be a relationship with flight number. Orbits like VLEO made its first attempt above the 60th mark and had quite a number of significant successful outcomes at other attempts.

Payload vs. Orbit Type

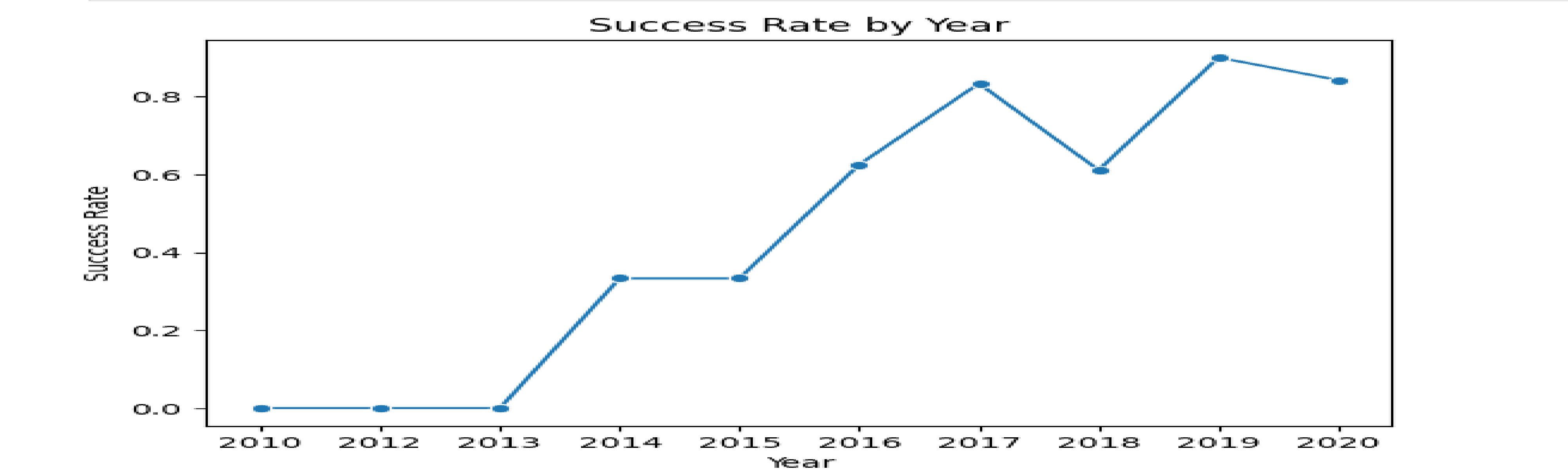
- Scatter point of payload vs. orbit type



- SSO seems to show successful outcomes with all attempts even as the payload increased but all below 4000kg. Heavy payload produced more successful outcomes for Polar, Leo and ISS. However, GTO cannot distinguish how payload affects it’s mission outcomes.

Launch Success Yearly Trend

- Line chart of yearly average success rate



- There was a steady increase in success rates from 2013 till 2020. The year 2019 recorded the highest success rates at about 90%

All Launch Site Names

- Find the names of the unique launch sites

Done.

Out[13]:

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

- %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
- This query selects unique launch sites from the spacextable

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

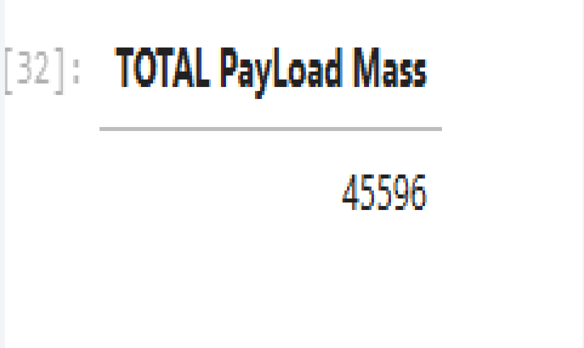
16]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Lar
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Fai
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	Fai
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	

- %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
This query selects all columns from the spacex table where the launch site name starts with ‘CCA’ and outputs only the first five rows.

Total Payload Mass

- Total payload carried by boosters from NASA



[32]:

TOTAL PayLoad Mass
45596

- %sql SELECT SUM("PAYLOAD_MASS__KG_") AS "TOTAL PayLoad Mass" FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)’;
- This query calculates the total of all payload mass where the customer column has a row input of ‘NASA (CRS)’.
  QUERY RESULT: The total payload mass for ‘NASA (CRS)’ is **45596 KG**

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1

Average PayLoad Mass
2928.4

- %sql SELECT AVG("PAYLOAD_MASS__KG_") AS "Average PayLoad Mass" FROM SPACEXTABLE WHERE "Booster_Version" = 'F9 v1.1';
- The query above calculates the average of the payload mass for row inputs that have booster version of F9 v1.1 and then outputs it with column header ‘Average Payload Mass’
- QUERY RESULT: The average payload mass for Booster version F9 v1.1 is **2928.4 KG**

First Successful Ground Landing Date

- Dates of the first successful landing outcome on ground pad

Done.

56]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landi
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Suc

- Query: %sql SELECT * FROM SPACEXTABLE WHERE "Date" = (SELECT MIN("Date") FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)’);
- Query Result:
  - 2015-12-22
  - The first successful ground landing happened on the 22nd of December 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

55]:

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- Query: %sql SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;

- Query Result:
  - The booster versions above have successful drone ship landings with payloads between 4000kg and 6000kg

Done.

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

Total Number of Successful and Failure Mission Outcomes

- Query: %sql SELECT "Mission_Outcome", COUNT(*) AS "Total Count" FROM SPACEXTABLE GROUP BY TRIM("Mission_Outcome");

- Query Result: The figure shows the total number of successful and failure outcomes.

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

Boosters Carried Maximum Payload

- Query: %sql SELECT DISTINCT("Booster_Version") FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE);
- Query Result: The figure shows the booster versions that have carried the maximum payload.

Done.

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

Months in 2015 that have failed landing outcome in drone ships

- Query: %sql SELECT substr(Date, 6, 2) AS Month, "Booster_Version", "Launch_Site", "Landing_Outcome" FROM SPACEXTABLE WHERE substr(Date, 0, 5) = '2015' AND "Landing_Outcome" LIKE '%Failure%drone ship%’;

  Query Result: The figure shows the month, booster version, launch sit and landing outcome of failed landing in drone ships in the year 2015

Landing_Outcome	count(Landing_Outcome)
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

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

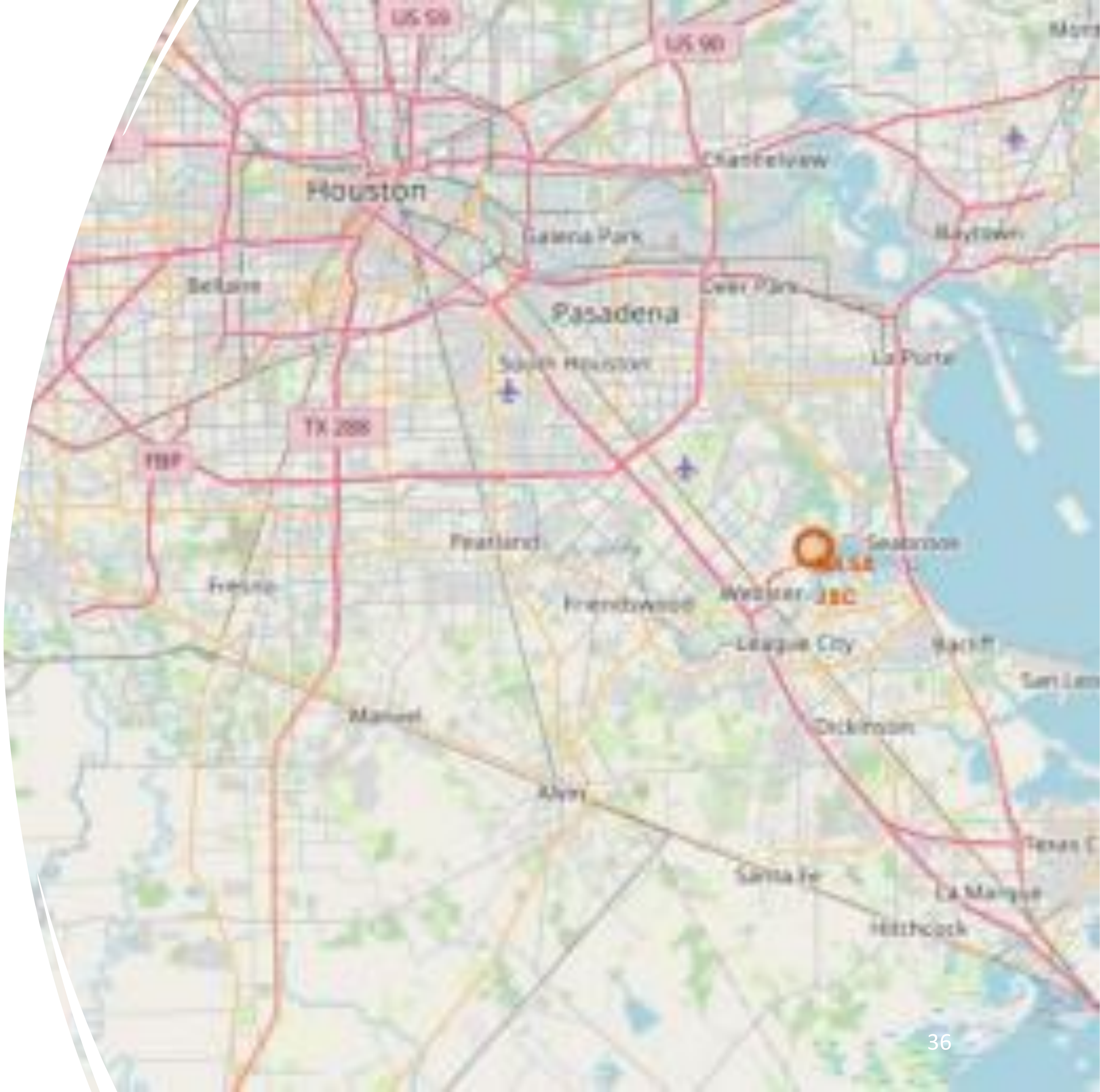
- Rank of count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- Query: %sql SELECT DISTINCT("Landing_Outcome"), COUNT(*) AS "Total Count" FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20’ GROUP BY "Landing_Outcome“ ORDER BY "Total Count" DESC;
- There were 10 records with no attempt

Section 3

Launch Sites Proximities Analysis

Map showing the NASA Johnson Space Center Houston, Texax.

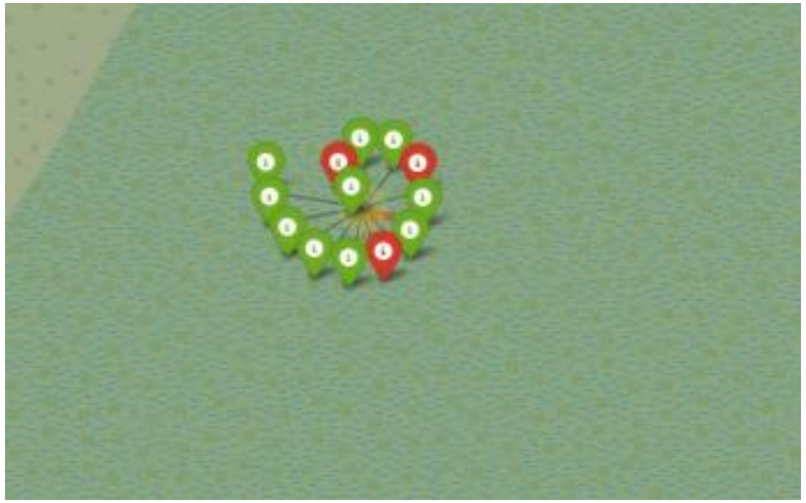
- The Nasa Station is located in a city that has major roads and airport leading to other cities/towns



Map showing the Launch Sites

Launch sites are in close proximity to the coast

Map showing the KSC LC-39A launch site



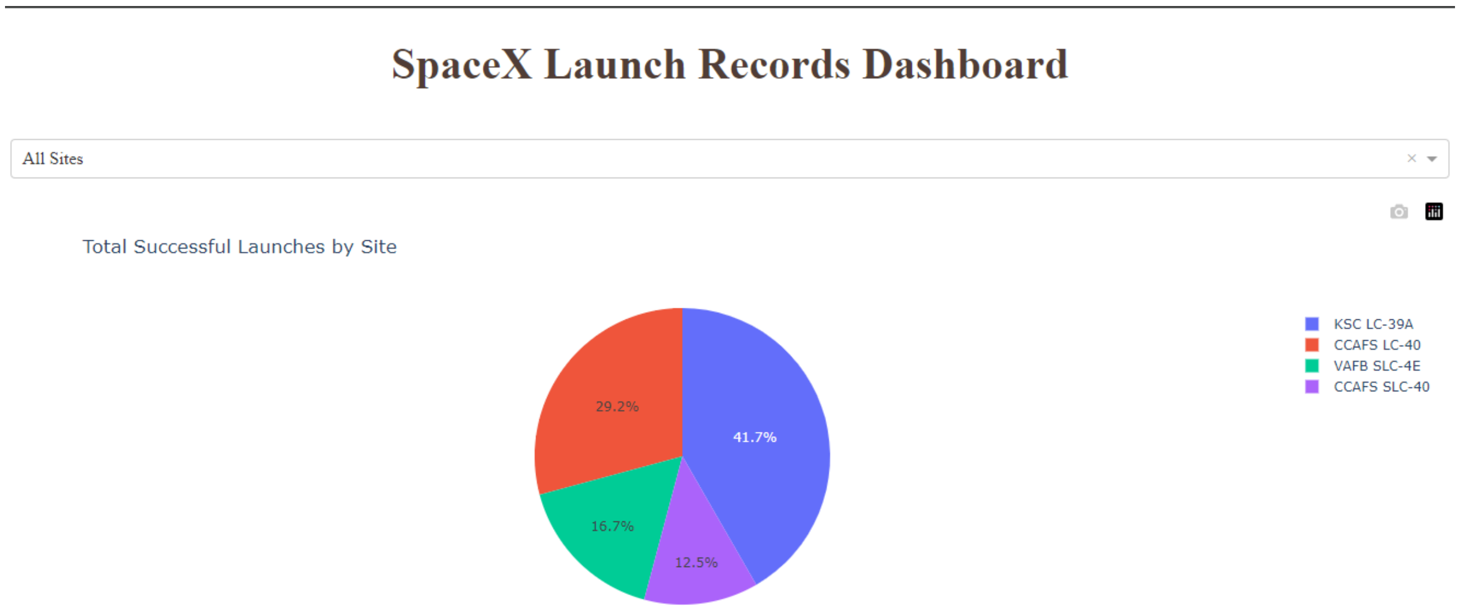
- The green markers indicate successful launches while the red markers show failures

Section 4

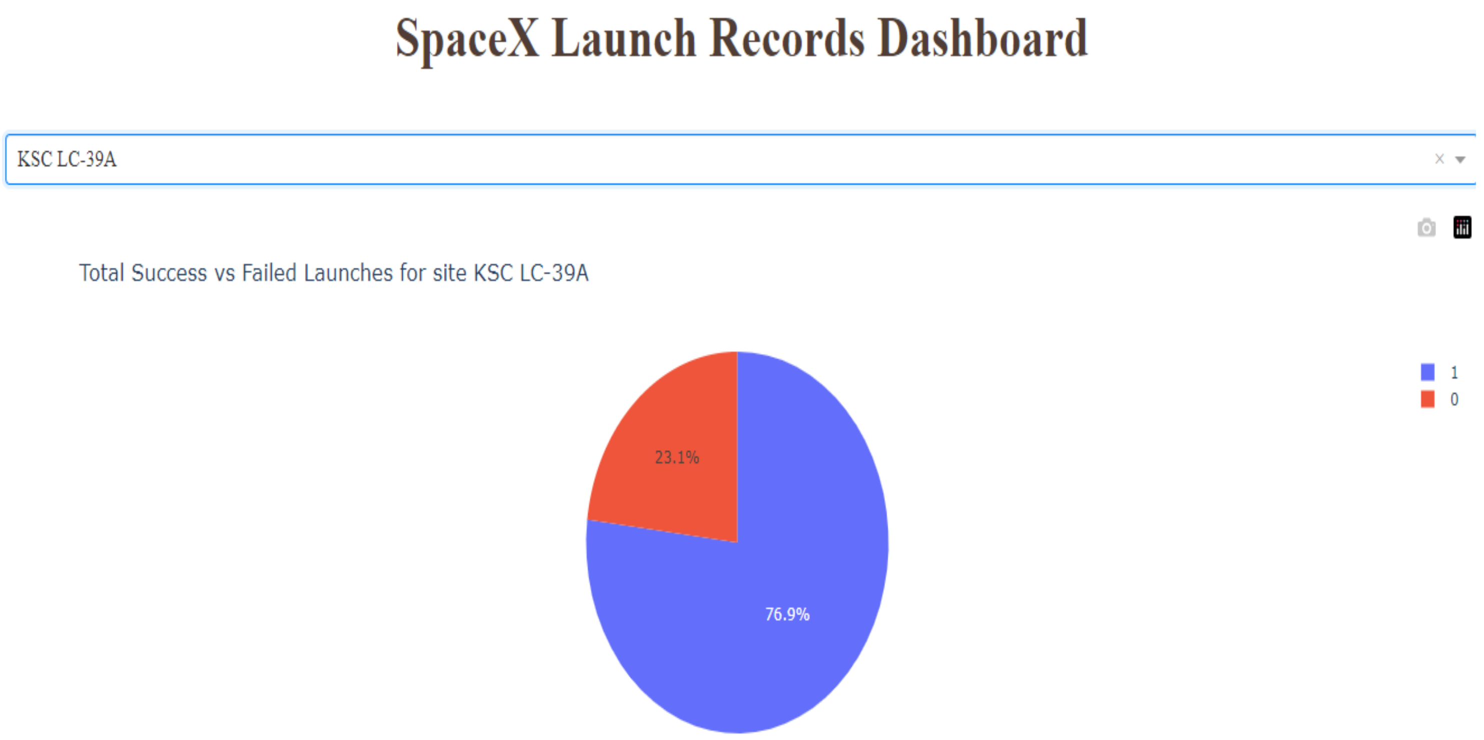
Build a Dashboard with Plotly Dash

Interactive Dashboard of Pie Chart showing successful launches by launch sites

- The chart shows the percentage of all successful launches contributed by each site. KSC LC-39A contributed the most successful launches with a percentage of **41.7%**



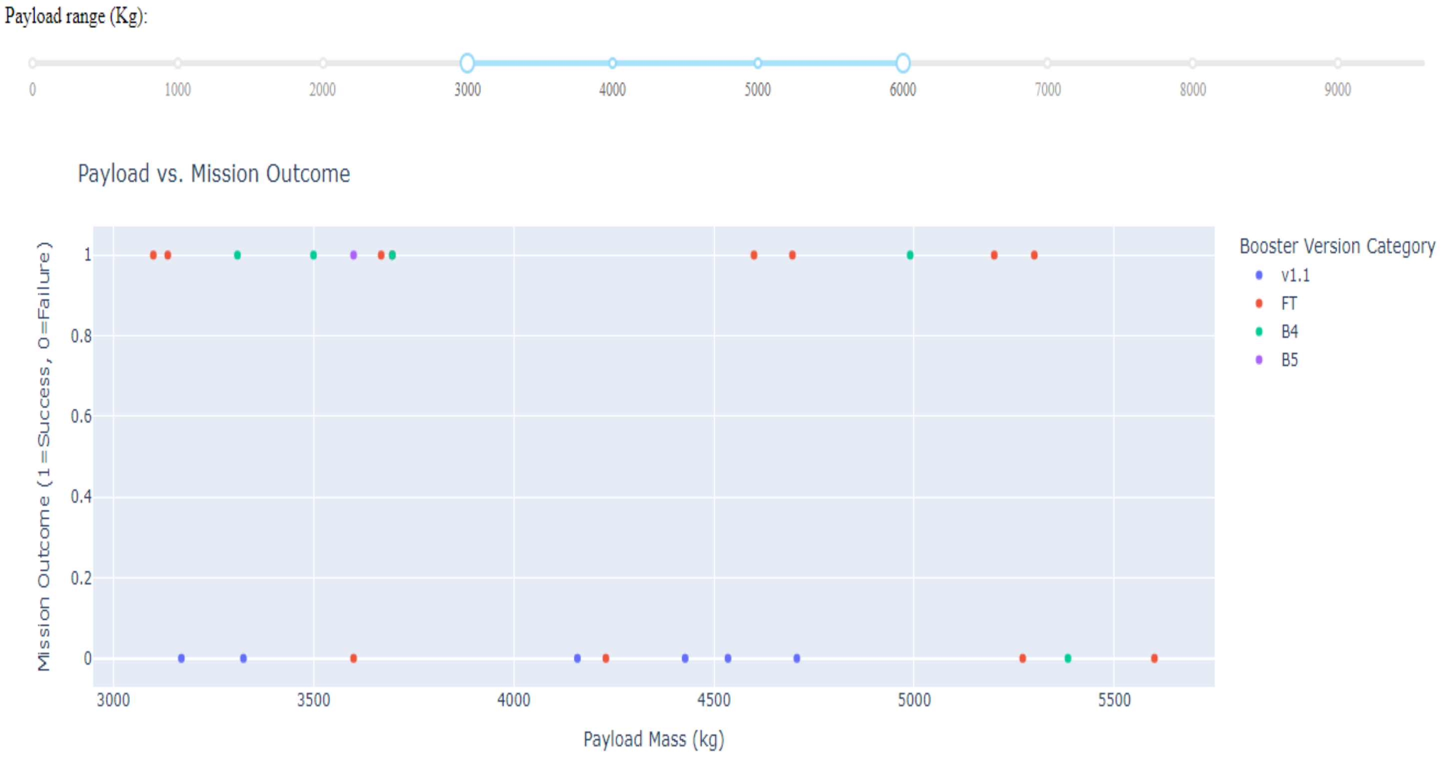
Dashboard Screenshot of the KSC LC-39A Launch Site



- The KSC LC-39A is ranked the highest successful launch site for mission outcomes. It has a percentage of 23.1% failure rate and 76.9% success rate

Dashboard screenshot of payload mass vs launch outcome for booster version in all sites

- For all sites, and within a payload range of 3000kg to 6000kg, booster version FT have more successful mission outcome than unsuccessful ones. Booster version v1.1 seems to only have unsuccessful mission outcome within this range

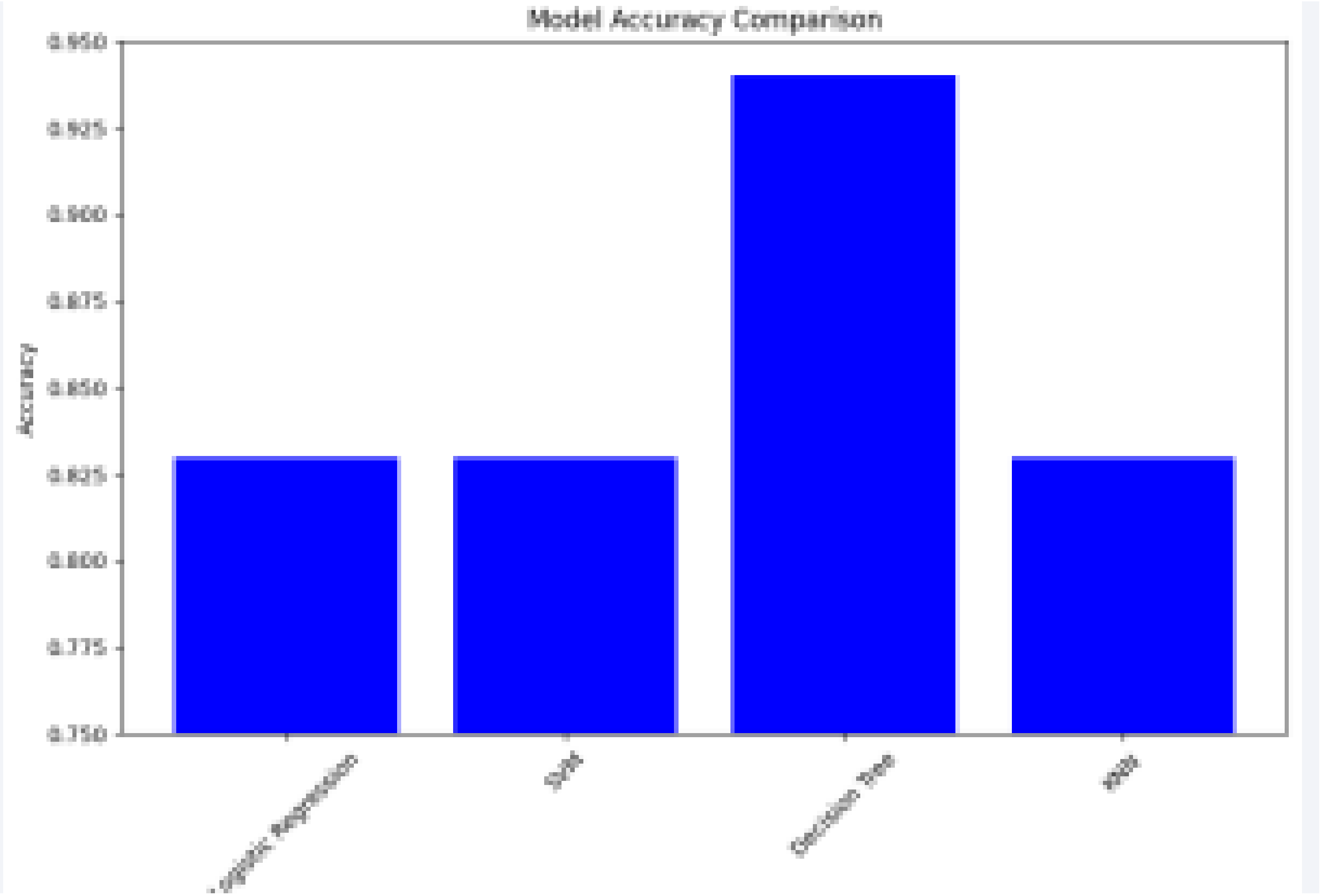


Section 5

Predictive Analysis (Classification)

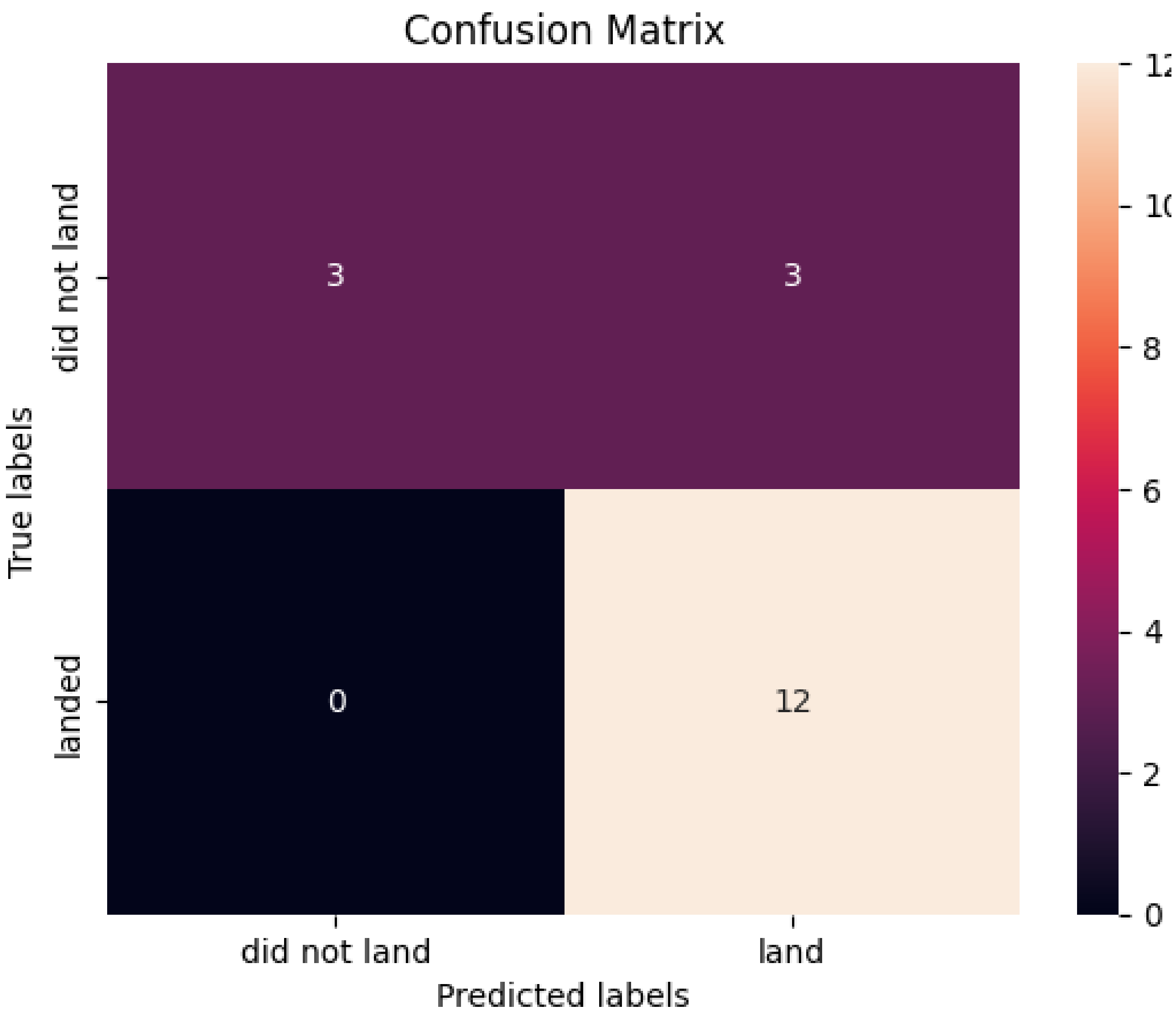
Classification Accuracy

- The decision Tree model performs the best. All other models have equal performance



Confusion Matrix

In the confusion matrix, we have the test data predict 3 true negative outcomes, 0 false negative outcome, 3 false positives and 12 true positives.



Conclusions

- The Decision Tree is the best model to use for predictions of future mission outcomes
- The KSC LC-39A launch site is the most probable place to have a successful mission outcome also based on other factors like the payload
- Payload mass is best kept within a range of 4000kg-6000kg
- The FT booster version is the leading booster for successful outcomes

Appendix

Find the method performs best:

```
data = {
    'KNN': {'best_score': knn_cv.best_score_, 'best_params': knn_cv.best_params_},
    'Decision Tree': {'best_score': tree_cv.best_score_, 'best_params': tree_cv.best_params_},
    'SVM': {'best_score': svm_cv.best_score_, 'best_params': svm_cv.best_params_},
    'Logistics Regression': {'best_score': logreg_cv.best_score_, 'best_params': logreg_cv.best_params_}
}

# Find the classifier with the highest best score
best_classifier = None
best_score = -float('inf')

for clas, result in data.items():
    if result['best_score'] > best_score:
        best_score = result['best_score']
        best_classifier = clas

# Print the best classifier and its details
print(f"The best classifier is: {best_classifier}")
print(f"Best score: {best_score}")
print(f"Best parameters: {data[best_classifier]['best_params']}")
```

```
The best classifier is: Decision Tree
Best score: 0.8767857142857143
Best parameters: {'criterion': 'gini', 'max_depth': 12, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samp
les_split': 5, 'splitter': 'random'}
```

Thank you!