IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Robert Kovachev
May 15th, 2025

Outline

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

Executive Summary

This project aimed to **predict whether the first stage of the Falcon 9 rocket would land successfully** during space tourism missions.

- **Data Collection :** Information was gathered using the SpaceX REST API and web scraping from Wikipedia.
- **Data Cleaning :** Irrelevant data was removed, and key features were selected for modeling.
- **Exploratory Analysis :** Visualizations were created to understand relationships between variables.
- **Interactive Dashboard :** A Plotly Dashboard was built to explore the data visually.
- **Machine Learning Models :** Logistic Regression, SVM, Decision Tree, and KNN models were trained and evaluated.
- **Results :** The best-performing model achieved an accuracy of 83.33% on the test set.

Introduction

The purpose of this project is to **predict if the Falcon 9 first stage, from the SpaceX space travel company, will land successfully**. Becouse if that is the case, the first stage can be reused and costs will be reduced

- The first goal was to **get the data, study it and clean** it so we can stick to the important variables like Launch Site, Flight Number, Payload Mass, Orbit, etcetera
- At the end, the main goal was to **find the best Machine Learning model** that can predict the succusses or failures of every flight
- **GitHub URL** to the full project: https://github.com/rob040404/DS-space-age/tree/main

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected through calls to the SpaceX REST API and web scrapping
- Perform data wrangling
  - Data was processed transforming the information into data frames and cleaning and normalizing the data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Machine Learning models were built and tested in order to see the one that performs most accurately

Data Collection

- **API calls:** data was collected by calling the SpaceX API trough a GET method.
- **Web Scrapping:** we also obtained data of the Falcon 9 and Falcon Heavy Launches by performing web scrapping and extracting the information from the tables of the Wikipedia's HTML.
- **Storage:** When the results were obtained, we stored the data as a CSV file.
- **Importation and transformation:** After thar we imported it to the Jupyter Notebook as a Pandas Data Frame and it was ready to be used.

API calls and web scrapping → Storage into CSV files → Importation of CSV files to Jupyter Notebook → Transformation into Pandas data set

Data Collection – SpaceX API

- Data was collected by calling the **SpaceX API**
- The endpoint was: "https://api.spacexdata.com/v4/launches/past"
- The methos used was **GET**
- The result was received as in a **JSON** format
- Data was filtered by launch site, launch year, payload mass, etc.
- **GitHub** Repository of the whole notebook: https://github.com/rob040404/DS-space-age/blob/main/1_spacex-data-collection-api-v2.ipynb

API call with: **respones.get()**

↓

Success revision with: **response.status_code** (Code 200 obtained)

↓

Normalizing data from JSON to Data Frame : **pd.json_normalize(response.json())**

↓

Extacting only the columns we need: launch site **, launch year, etcetera**

Data Collection - Scraping

- **Web scrapping** was made to the SpaceX Launches Wikipedia page
- **BeautifulSoup** library was used for that purpose
- The response was a **HTML** document from which we extracted the table we needed
- The data from the table was turned into a **Pandas Data Frame**
- **GitHub repository** of the whole Notebook: https://github.com/rob040404/DS-space-age/blob/main/2_webscraping.ipynb

Data Wrangling

- The **CSV file** was imported as a **Pandas Data Frame**
- **Null values** were checked in each attribute and **numerical / categorical** columns were identified
- There were performed calculations on:
  - Number of launches on each site
  - Number and occurrence of each orbit
  - Number and occurrence of mission outcome of the orbits
- Creation of a **landing outcome label** from Outcome column
- **GitHub URL** of the Notebook: https://github.com/rob040404/DS-space-age/blob/main/3_Data%20wrangling-v2.ipynb

Importing the CSV file into a Data set with: **pd.read_csv()**

Analizing the data with methods like: **df.isnull(), df.dtypes**

Calculating several outcomes related to orbits with: df[].count()

Creating another column from the 'Outcome Column' with: **df['Outcome'].apply()**

EDA with Data Visualization

- **Scatter Plot Charts** were used to showcase the relationship between two variables and if they led to a successful landing or not
- **Bar Charts** have also been used to see the success rate for each type of Orbits
- **Line Charts** have been used to track tendencies trough time. To measure the success rate of landings per year
- **GitHub repository** with the full Notebook: https://github.com/rob040404/DS-space-age/blob/main/5_eda-dataviz-v2.ipynb

EDA with SQL

Several queries were performed on the data with **SQL** commands in order to extract valuable information relative to :

- Launch Site names and names that begin with "CCA"
- Total Payload Mass and average Payload Mass by F9v1.1 booster version
- Successful and failed landings
- Launch Records by year
- **GitHub URL** to full Notebook: https://github.com/rob040404/DS-space-age/blob/main/4_eda-sql.ipynb

Build an Interactive Map with Folium

- A **Folium map** was created in order to visualize all information about the launch sites .
- **Folium circles and markers** were added to showcase the different coordinates of the launch sites. And **Marker Cluster** to group spots with little distance between each other.
- **Text labels** were also used to add an textual details near the circles
- **Mouse position** was also added in order to see the coordinates the mouse is over
- **Poly Line** was used to print the distance between two spots on the map.
- **GitHub URL** of the full Notebook: https://github.com/rob040404/DS-space-age/blob/main/6_launch-site-location-v2.ipynb

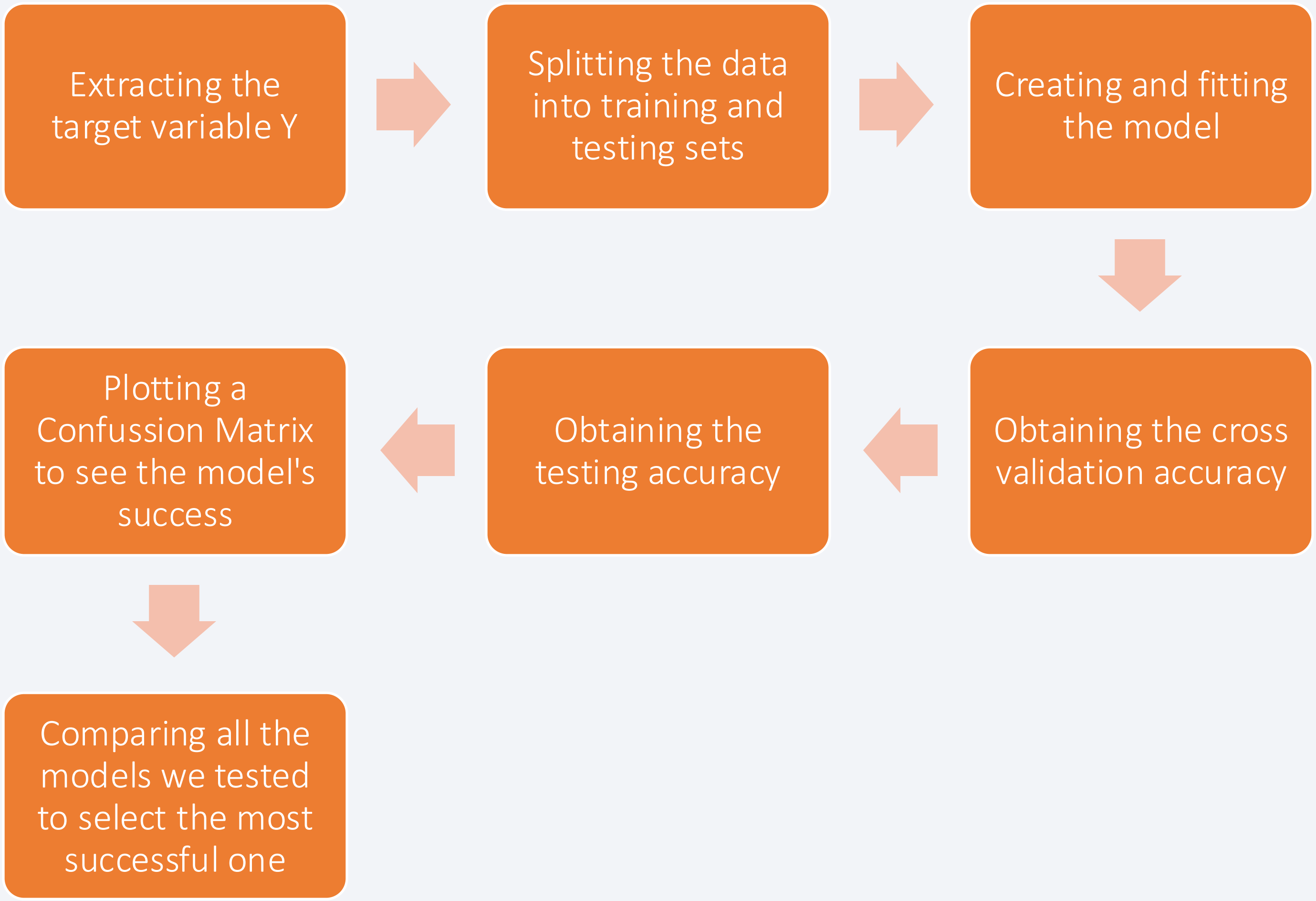
Build a Dashboard with Plotly Dash

We created a **Plotly Dash app** with the purpose to obtain **interactive graphs** that change instantly while we select the data we want to test. Several elements were added

- A **pie chart with a dropdown list** so we can select the Launch Sites and see their success rate.
- We also created **a scatter plot chart with a slider** with the purpose of selecting different ranges of payload mass along with the launch site and see if those were successful or not
- **GitHub URL** of the Plotly Dashboard: https://github.com/rob040404/DS-space-age/blob/main/7_spacex_dash_app.py

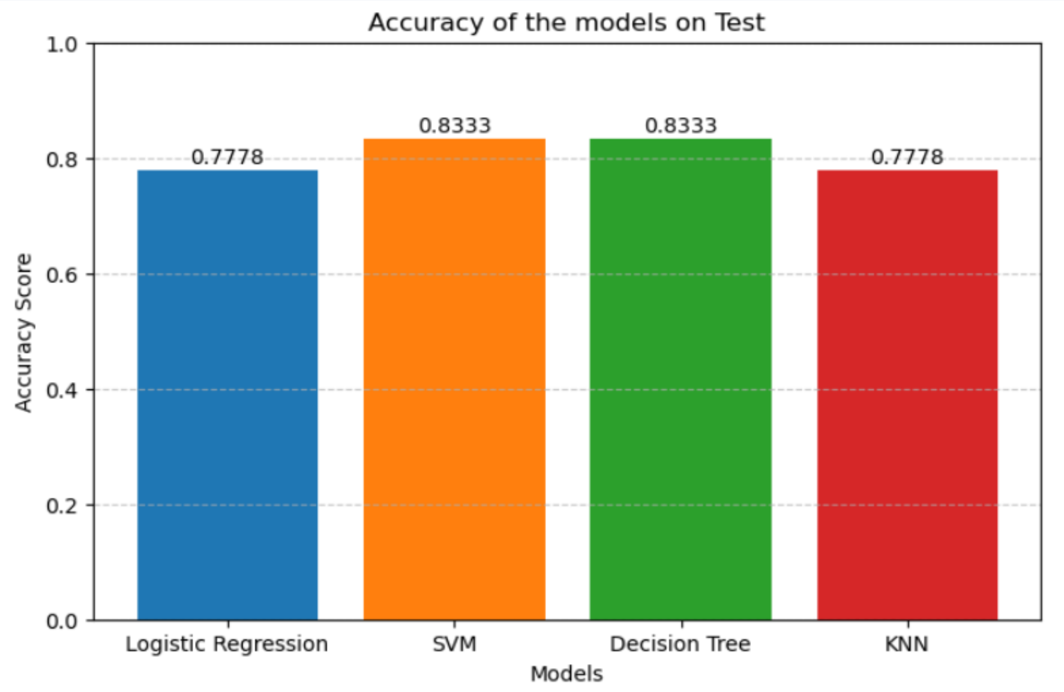
Predictive Analysis (Classification)

- **Several models were trained an tested** to finally see which one performs best.
- **GitHub URL** of the full Notebook: https://github.com/rob040404/DS-space-age/blob/main/8_SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb



Results

- The goal of the project is to **find the best Machine Learning model** to predict if the Falcon 9 first stage will land successfully
- Four models were tested: **Linear Regression, SVM, Decision Tree and KNN**
- The results were measured by the accuracy of all four models and the two that performed best were: **Decision Tree with 83.33% and SVM with 83.33%**
- But **decision tree** obtained more accuracy in the cross training score with 89.11% which is better than the 85% of SVM

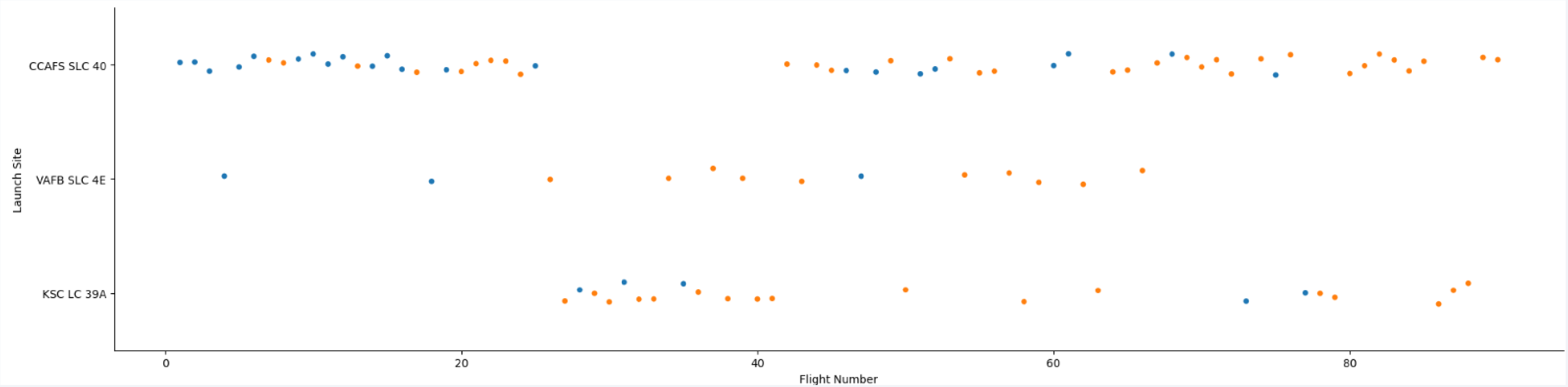


Section 2

Insights drawn from EDA

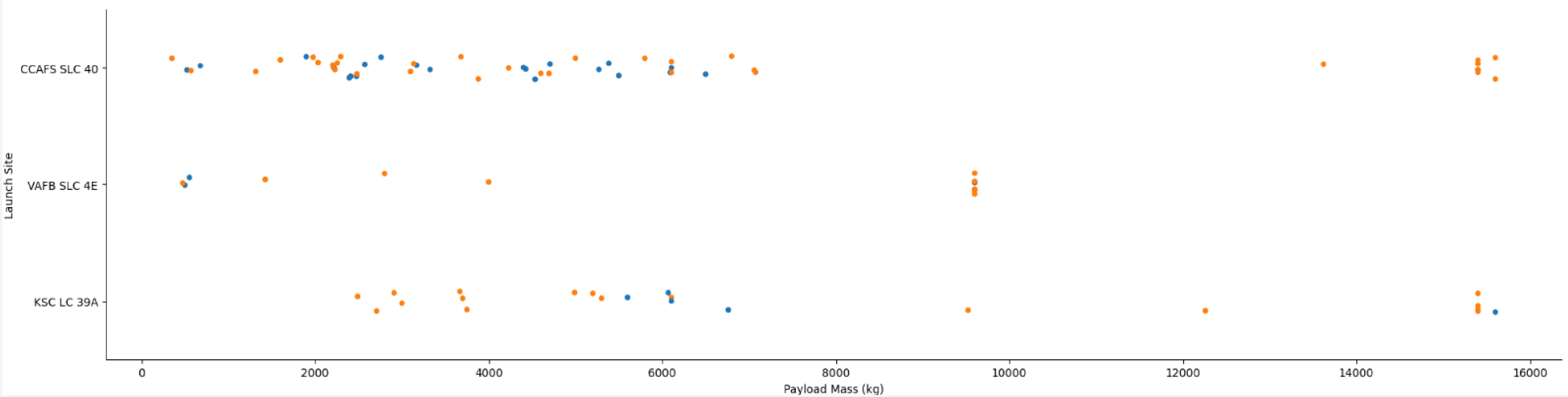
Flight Number vs. Launch Site

- **Flight Number/Launch Site Scatter Plot chart.** It shows us the relationship between the flight number and the launch site in terms of success rate (1=success, 0=failure). The more the recent the flight is there is more probability of successful landing.



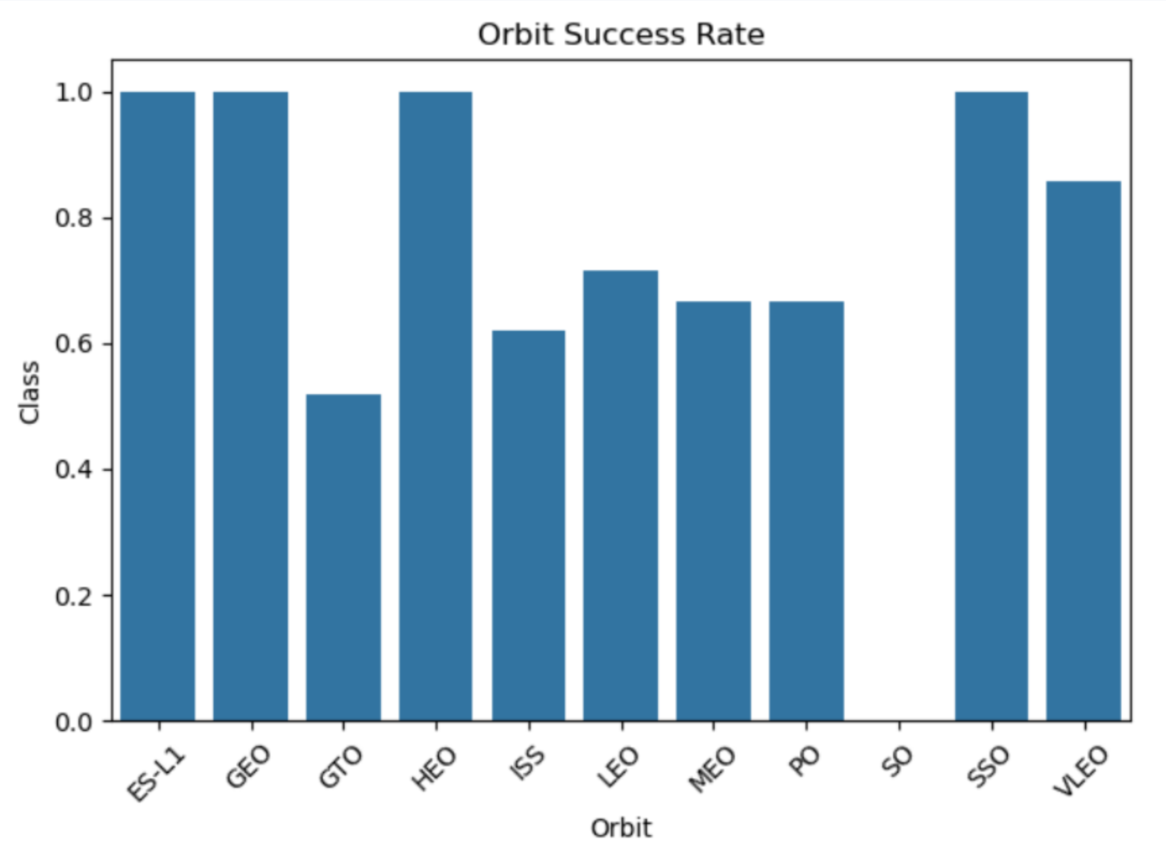
Payload vs. Launch Site

- **Payload Mass/Launch Site scatter plot chart**. It shows us the relationship between the payload and launch site in terms of success rate (1=success, 0=failure). Looks like the heaviest units get more successful landings.



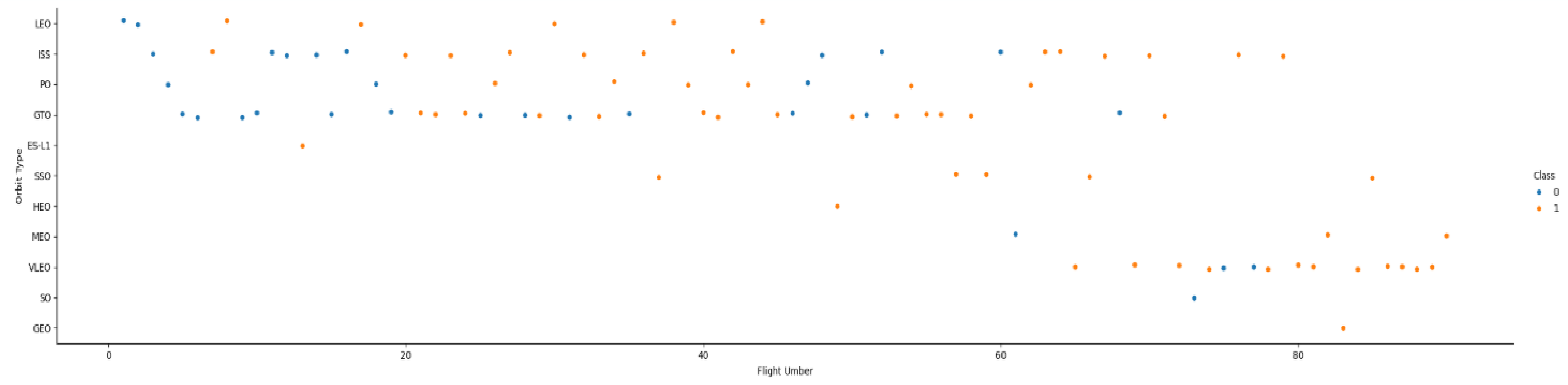
Success Rate vs. Orbit Type

- Orbit Success Rate Bar Plot. "0.0" represents failure and "1.0" represents successful landing. What we see on the plot is de mean for each type of Orbit.



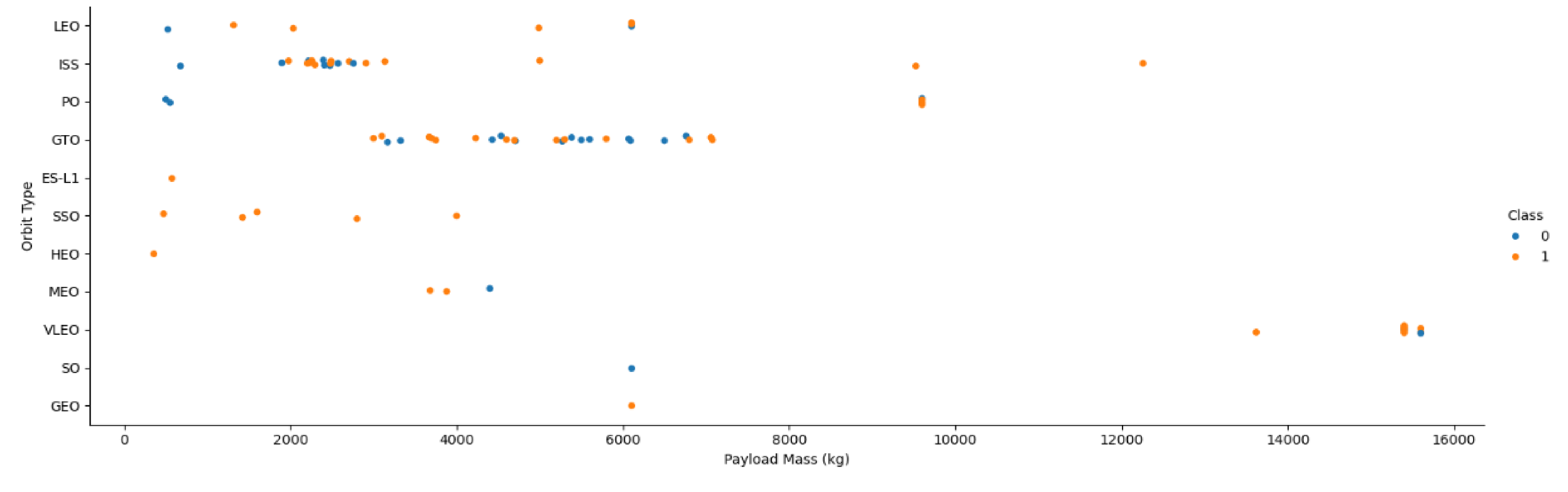
Flight Number vs. Orbit Type

- Flight Number / Orbit Type Scatter Plot Chart. It shows us which orbits were used trough time and how some types are more successful than others.



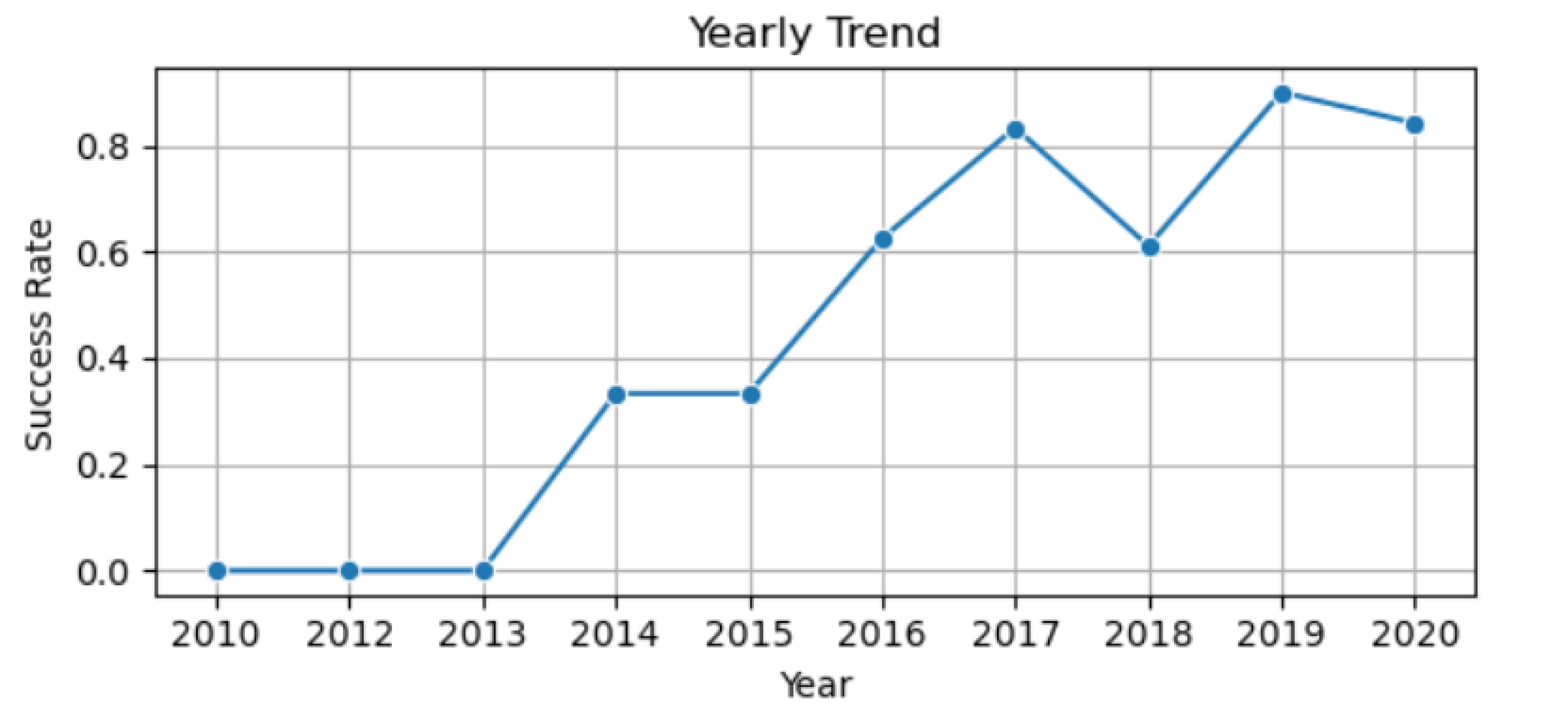
Payload vs. Orbit Type

- Payload Mass / Orbit Type Scatter Plot Chart. It shows us the relationship between the payload, the orbit type and the success



Launch Success Yearly Trend

- Yearly Trend Line Plot. It shows how the successful landing rate has been increasing overtime with little setbacks.



All Launch Site Names

- There are only 4 possible Launch Sites as we can see in the result image
- This result was obtained through an sql command

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

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

Launch Site Names Begin with 'CCA'

- We see a search for names that begin with 'CCA' in the column 'Launch_Site'
- The query was performed with sql again

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

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

- The total payload carried by boosters from NASA was calculated with sql. The result is shown in kilograms

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

SUM(PAYLOAD_MASS__KG_)
48213

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1 result obtained with sql. The result is shown in Kilograms

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- Date of the first successful landing outcome on ground pad. Query and result

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'
```

MIN(Date)
2015-12-22

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

```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome =
'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

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

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes

```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE Landing_Outcome LIKE '%Success%' OR Landing_Outcome LIKE '%Failure%'
```

COUNT(*)
71

```
%sql SELECT COUNT(CASE WHEN Landing_Outcome LIKE '%Success%' THEN 1 END) AS Successful_Missions,
COUNT(CASE WHEN Landing_Outcome LIKE '%Failure%' THEN 1 END) AS Failed_Missions FROM SPACEXTBL;
```

Successful_Missions	Failed_Missions
61	10

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass

```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

Booster_Version	
F9 B5 B1048.4	F9 B5 B1049.5
F9 B5 B1049.4	F9 B5 B1060.2
F9 B5 B1051.3	F9 B5 B1058.3
F9 B5 B1056.4	F9 B5 B1051.6
F9 B5 B1048.5	F9 B5 B1060.3
F9 B5 B1051.4	F9 B5 B1049.7

2015 Launch Records

- List of the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

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

- 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

```
%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;
```

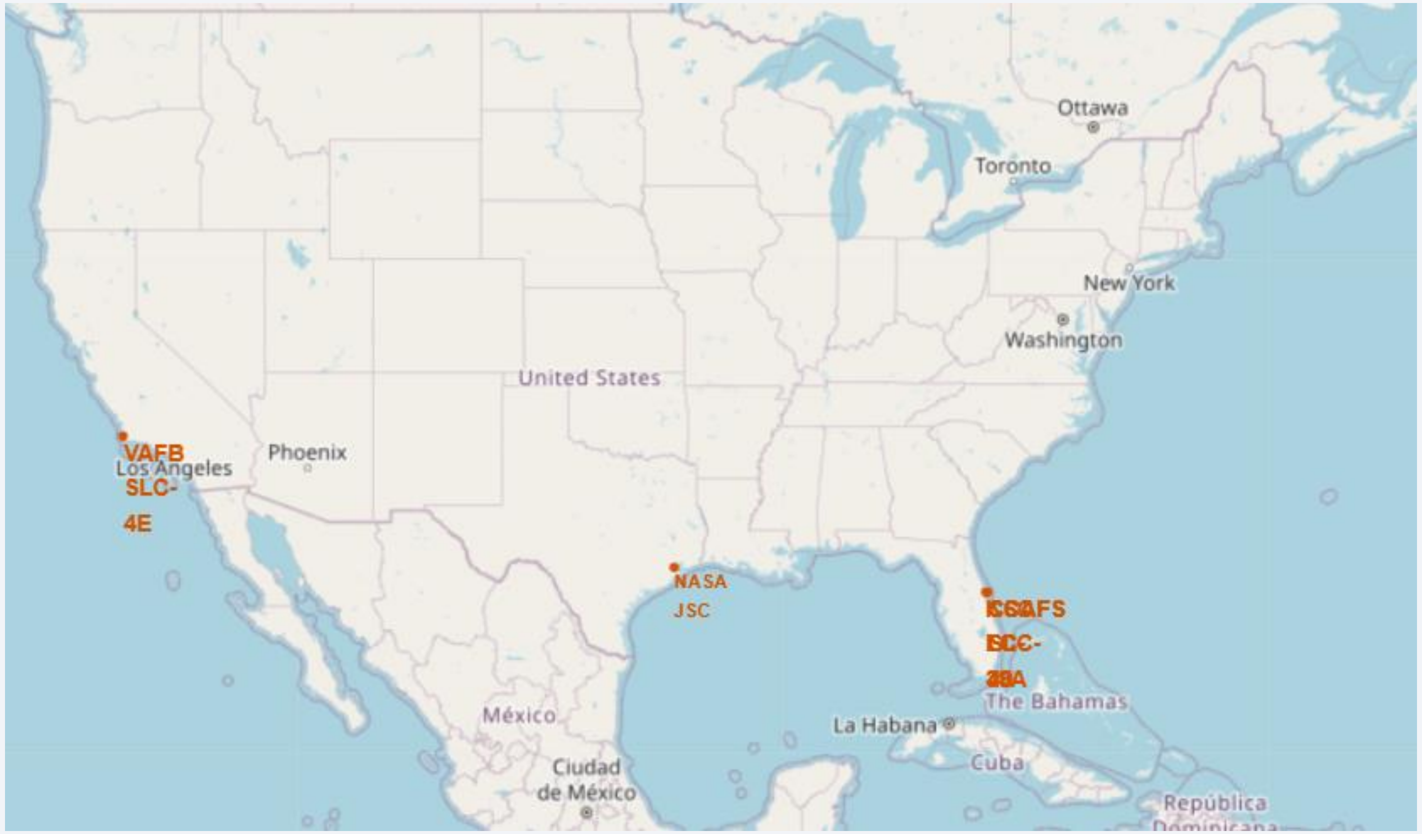
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

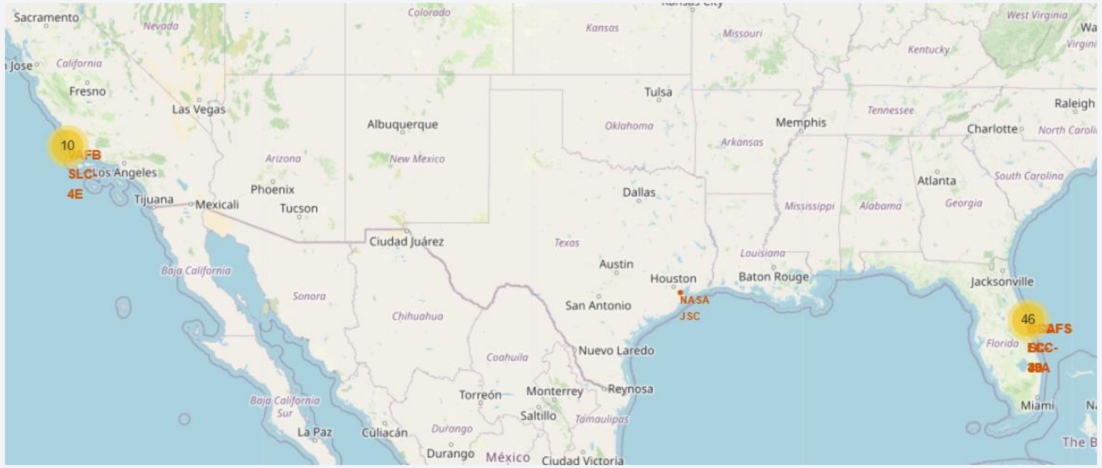
Global Map with all Launch Sites

All Launch Sites are located in Florida and California. In Florida there are three sites that are very close to each other. **Folium library** has been used.



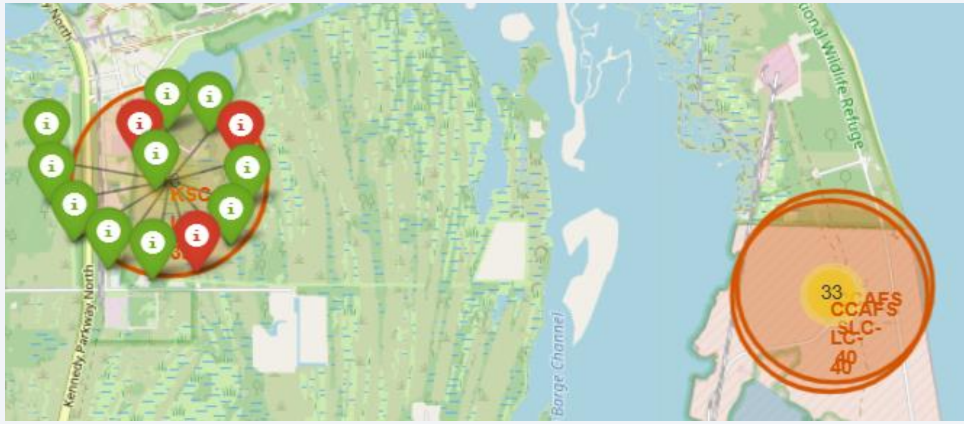
Map of Successful and not Successful Launches

Here we see all the launches by zone and if we zoom or click we can also see the successful and failed ones. **Folium cluster markers** have been used.



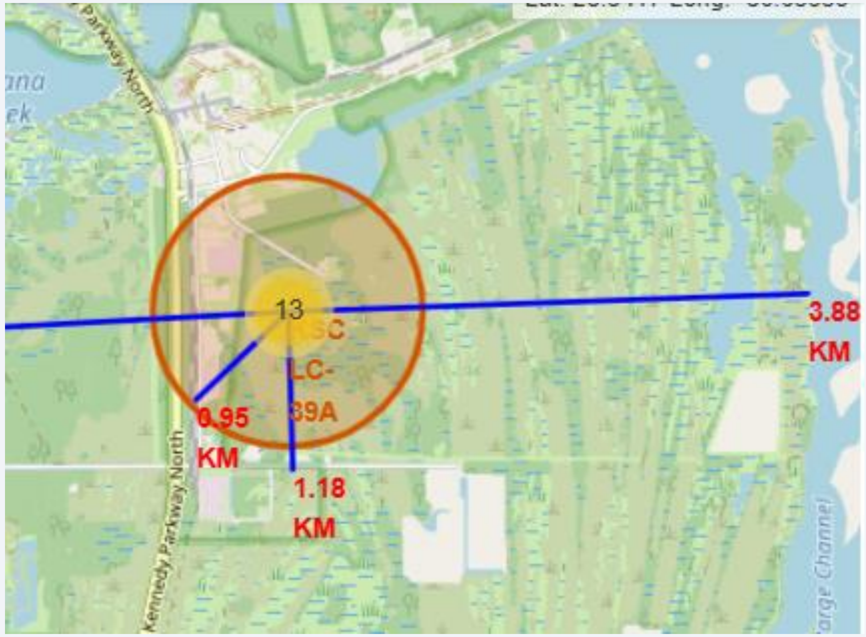
California:

Florida:



Map of the proximities of a Launch Site

Exploration the generated of a launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed. **Folium Poly Lines** were used to represent the distance with a line and **markers** were used to showcase de distance in numerical format.



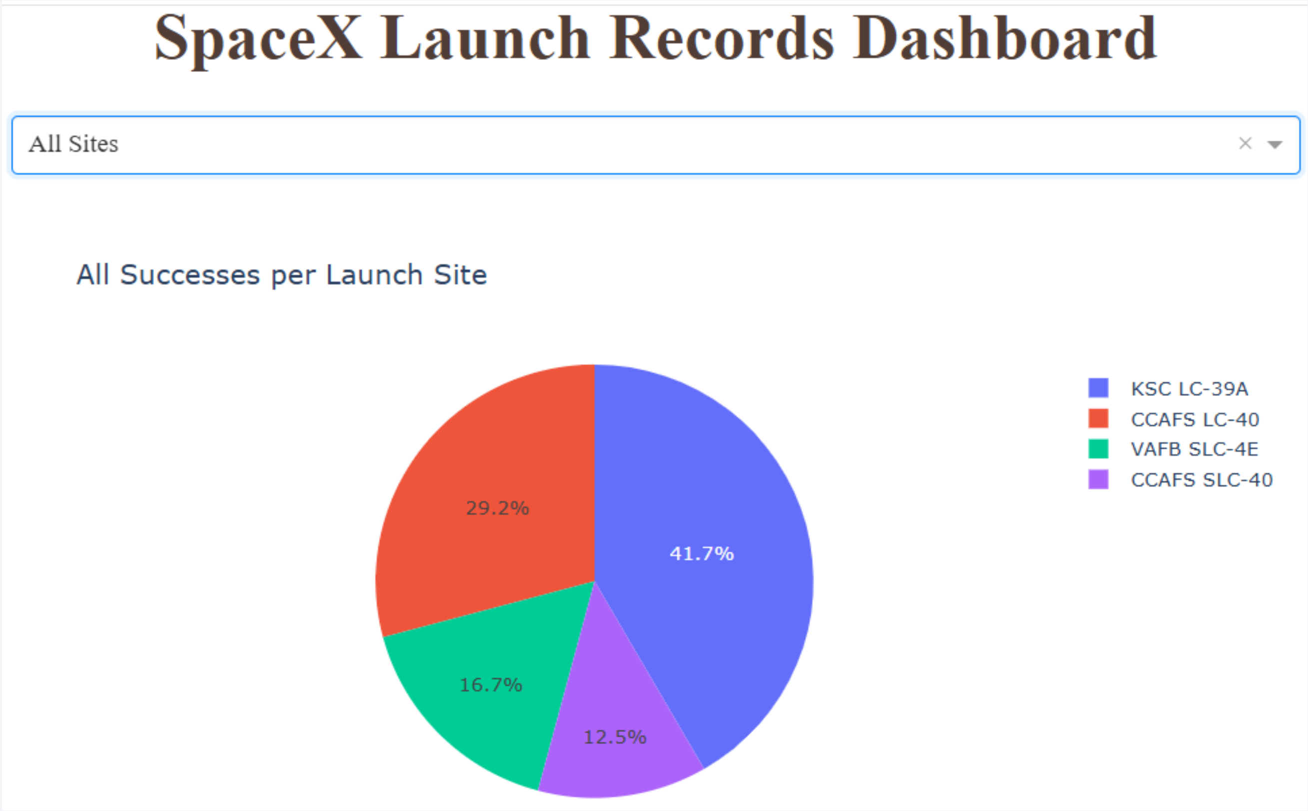
Section 4

Build a Dashboard with Plotly Dash

Pie Chart with the percentage of success for all Launch Sites

Here we see a an **Dash Board app** with an interactive pie chart graph.

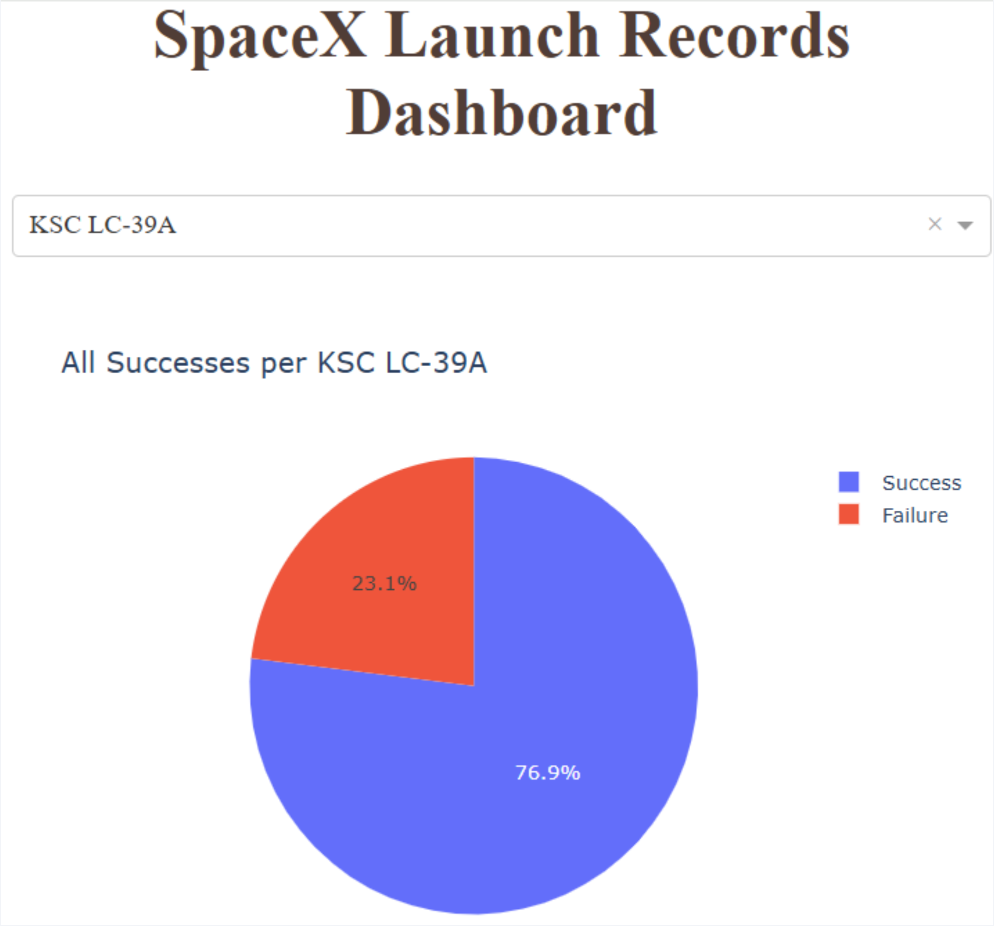
- It shows launch success count for all sites
- There is a dropdown list where we can explore the statistics of all launch sites together or as an individual launch site
- We see **KSC LC-39A** is the one with most success



Pie Chart for the most Successful Launch Site

Interactive Pie Chart the success rate **KSC LC-39A**, which is the most successful Launch Site

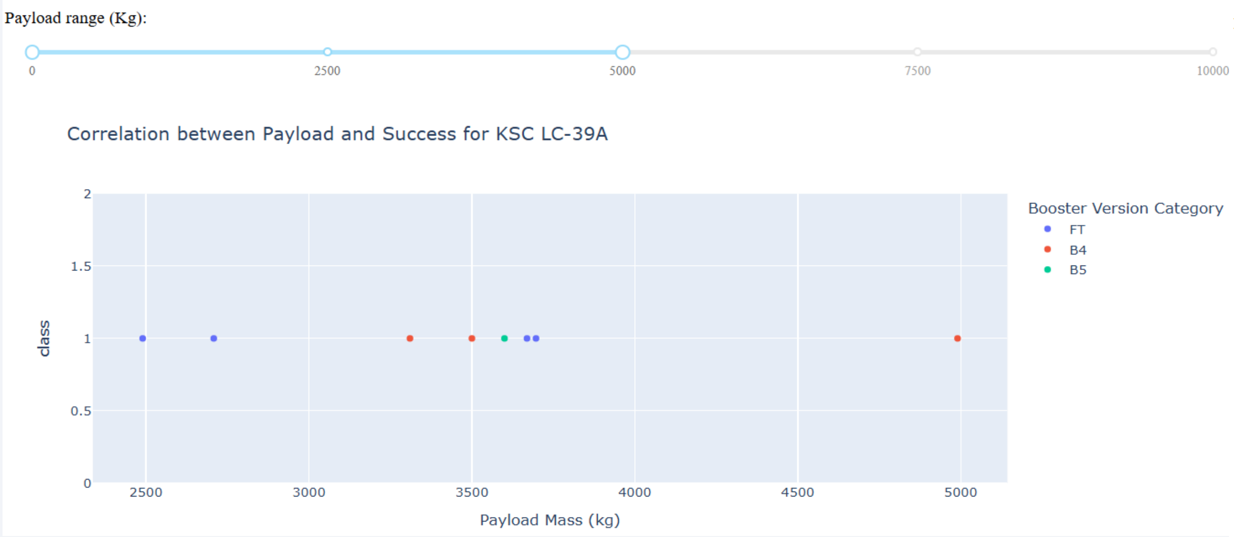
- **76.9%** of the times lands successfully and 23.1% of the time it's a failed landing

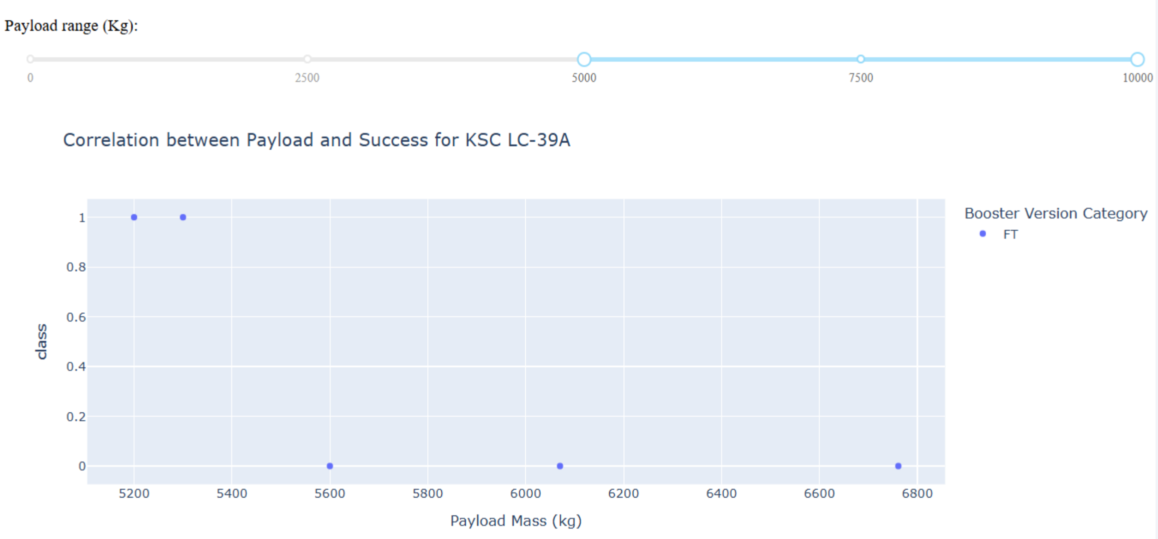


Scatter Plot interactive Chart

An interactive Scatter Plot chart where **the range of the payload mas can be selected** with the slider to see its correlation with the selected Launch Site from the dropdown list.

- We see hay the values change when we try different payloads:



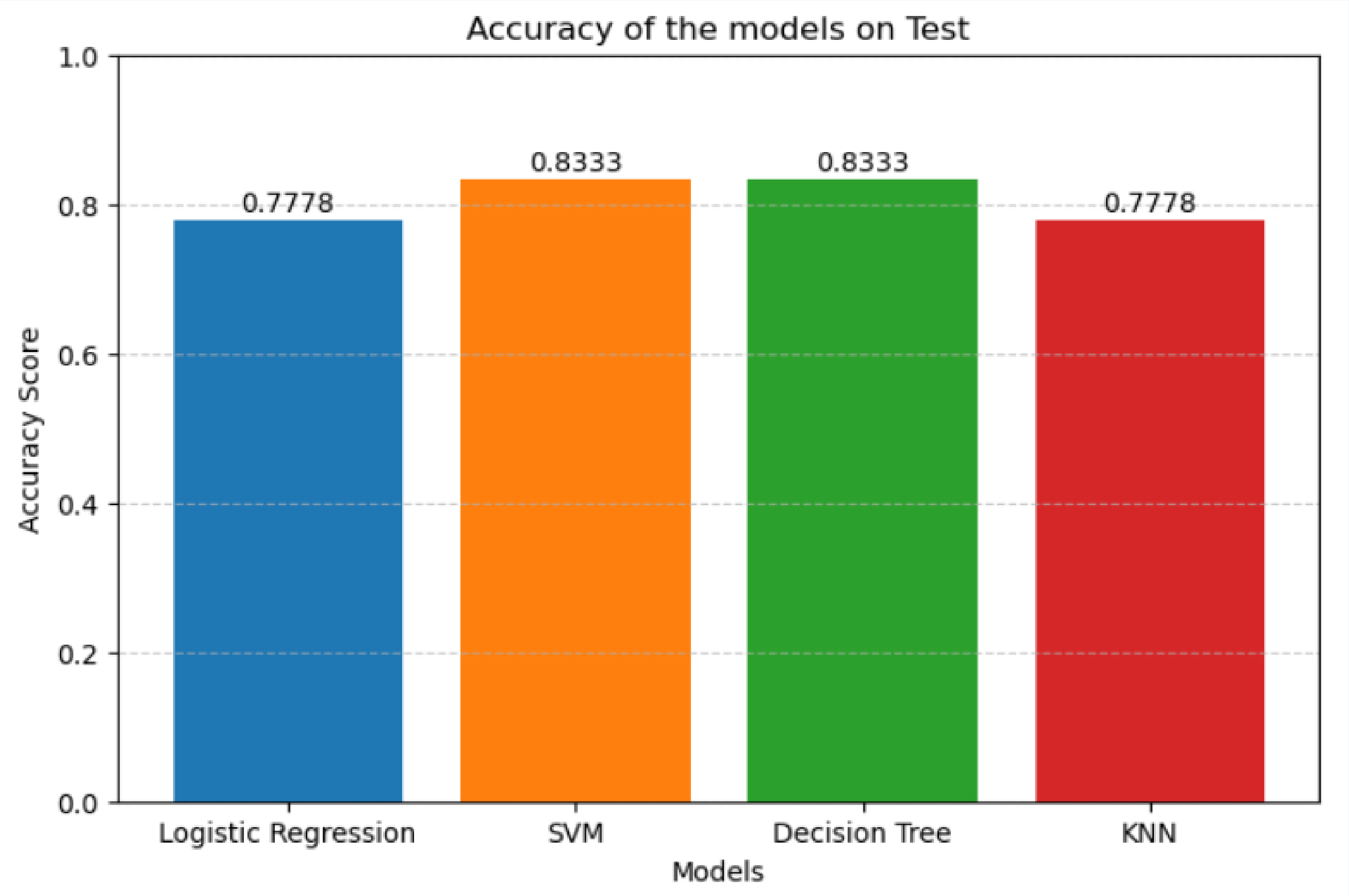


Section 5

Predictive Analysis (Classification)

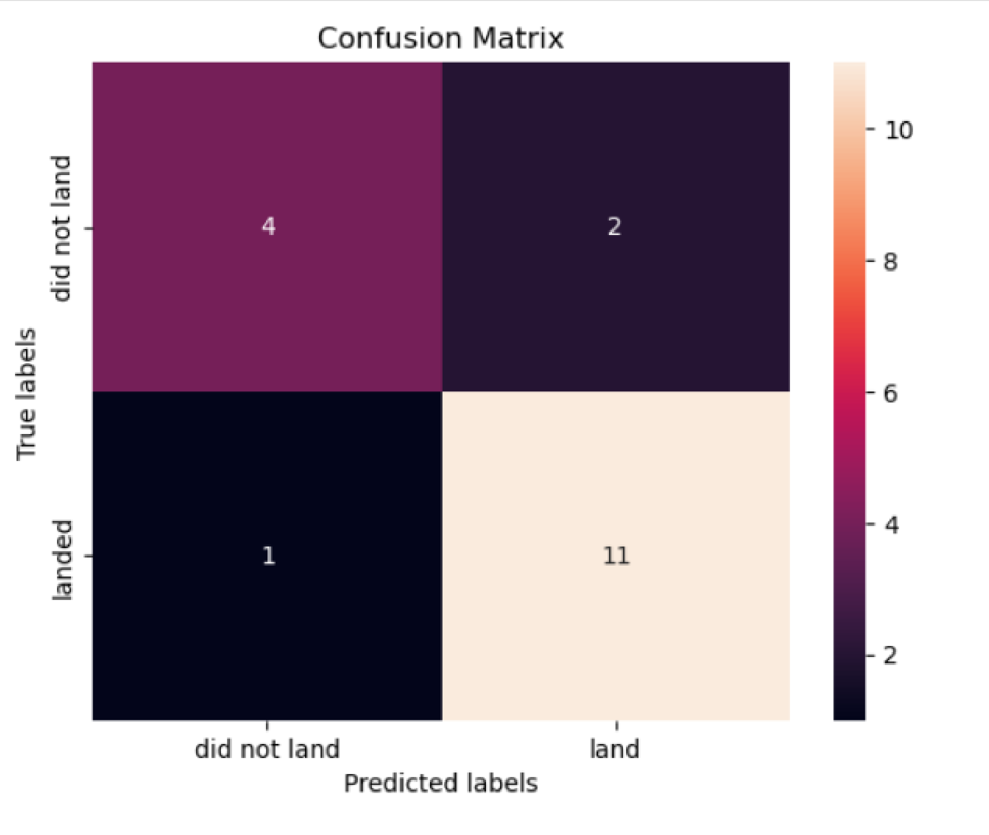
Classification Accuracy

- The bar chart show that there are **two models** that perform better than the others
- SVM and Decision Trees are the winners and there is a tie between them in with this metrics, but **decision trees** performed better with the cross validation score, so maybe it's more stable than the SVM model.
- **GitHub URL** to full Notebook: https://github.com/rob040404/DS-space-age/blob/main/8_SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb



Confusion Matrix

What we see in this matrix is that when we applied the **Decision Tree** (the best performing model) there were 2 false positives and1 false negative.



Conclusions

- Early flights had worse success rate than the latest
- Launch Site "**KSC LC-39A**" is the one with the best success rate
- **Decision Tree and SVM models** have the same accuracy on performing on the test data
- We chose **Decision Tree as the best model** becouse its accuracy on the cross validation was a little better
- Decision Tree predicted 2 false positives and 1 false negative

Thank you!