IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mohamed Ragab
15/02/2026

Outline

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

Executive Summary

- Collected SpaceX Falcon 9 launch data from public datasets, APIs, and web sources.
- Cleaned and transformed the data, handled missing values, and engineered useful features.
- Performed exploratory data analysis to identify trends affecting landing success.
- Built and evaluated multiple machine learning models including Logistic Regression, SVM, Decision Tree, and Random Forest.

- Payload mass, launch site, and orbit type showed strong influence on landing success.
- Random Forest, Logistic Regression, Random Forest and K-nearest neighbor delivered the same accuracy.
- Interactive dashboard enabled clear visualization of launch success patterns.
- Findings demonstrate how data science can support **cost reduction and mission planning** through reusable rocket prediction.

Introduction

Project Background and Context

- SpaceX significantly reduces the cost of space missions by reusing the Falcon 9 first-stage booster.
- Predicting whether the booster will successfully land is important for estimating mission cost and operational planning.
- Data science and machine learning provide tools to analyze historical launch data and uncover patterns related to landing success.

Problem to Be Solved

- Identify the key factors that influence Falcon 9 first-stage landing success.
- Build predictive machine learning models to classify landing outcomes.
- Provide insights that support **cost reduction, mission reliability, and strategic decision-making**.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

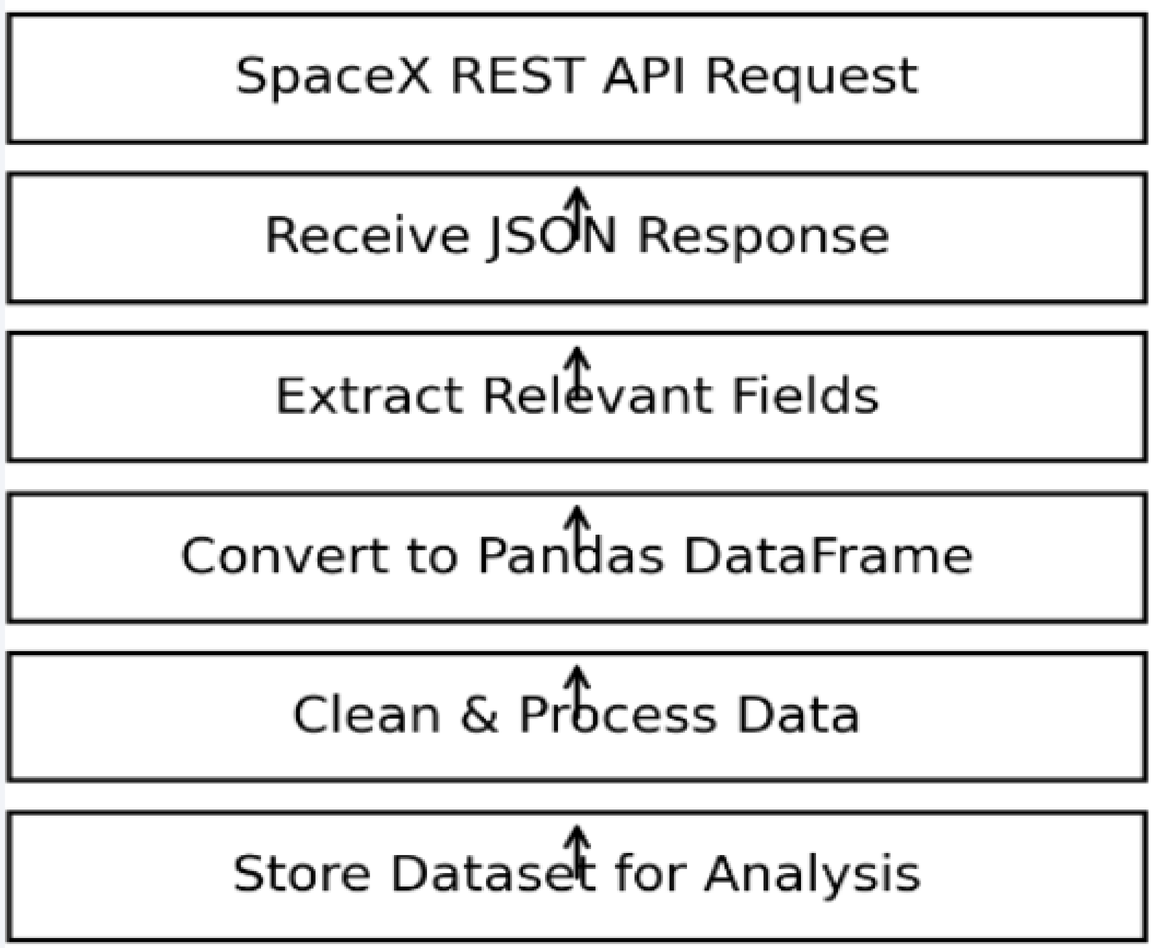
Data Collection

- Collected SpaceX Falcon 9 launch data from public datasets, APIs, and web sources by web scraping

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.07B0003.18	Failure	4 June 2010	18:45
1	1	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.07B0003.18	Failure	4 June 2010	18:45
2	1	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.07B0003.18	No attempt\n	4 June 2010	18:45
3	2	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.07B0004.18	No attempt	8 December 2010	15:43
4	3	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.07B0005.18	No attempt\n	22 May 2012	07:44

Data Collection – SpaceX API

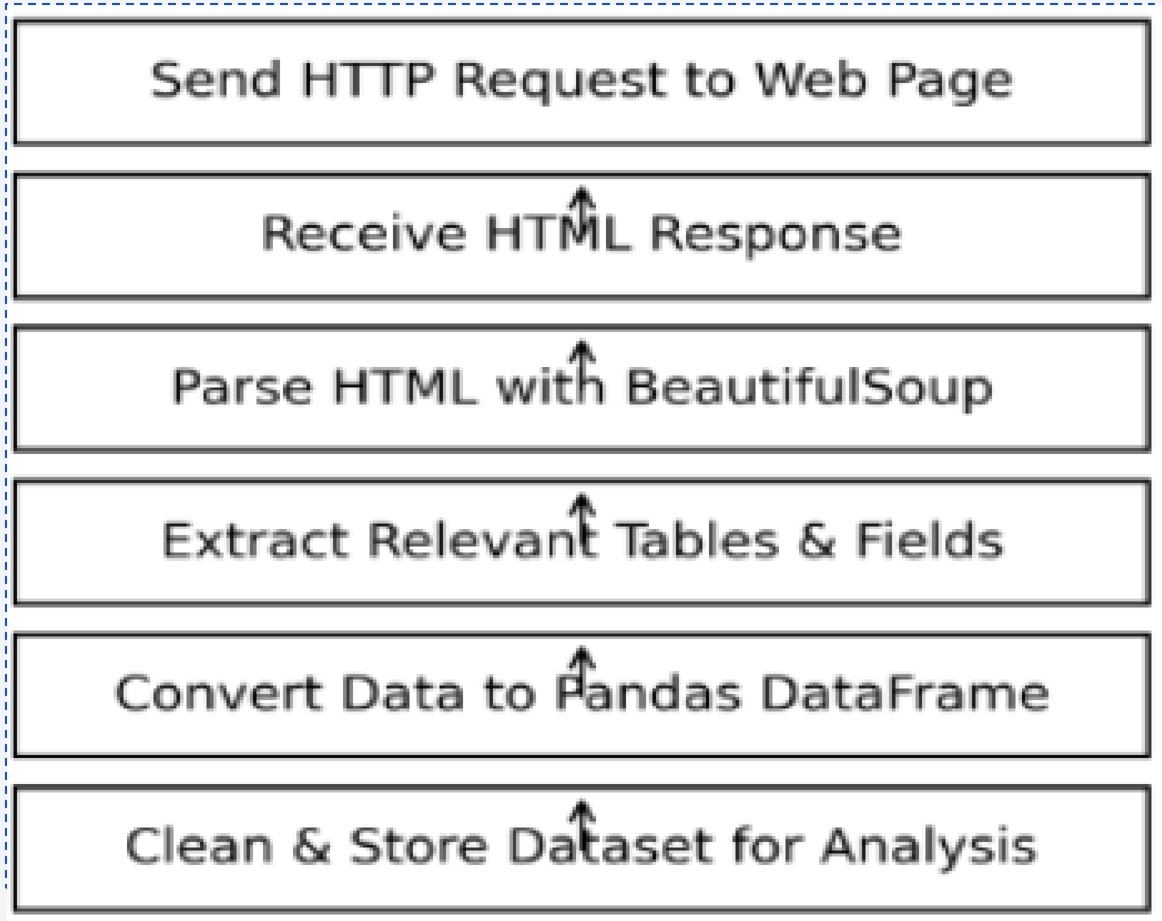
- Launch data was collected using **SpaceX REST API calls** with relevant key phrases such as *launches, payloads, rockets, and launchpads*.
- The API responses were retrieved in **JSON format**, then parsed and transformed into a structured **Pandas DataFrame**.
- Data cleaning and preprocessing were performed to ensure the dataset was **analysis-ready**.
- The completed **SpaceX API notebook** (including executed code cells and output results) is provided in the **GitHub repository** as an external reference for **transparency and peer review**.
- **GitHub Repository Link:** (https://github.com/MohamedNasserIV/IBM-Data-Science-Capstone-Space/blob/main/notebooks/01-spacex-api.ipynb)



Data Collection - Scraping

- Additional launch data was collected using **web scraping techniques** from publicly available web pages containing historical SpaceX launch records.
- An **HTTP request** was sent to retrieve the webpage content, and the returned **HTML** was parsed using **BeautifulSoup**.
- Relevant tables and fields were extracted, structured into a **Pandas DataFrame**, and cleaned to ensure the dataset was **consistent and analysis-ready**.

- The completed **web scraping notebook** (including executed code cells and output results) is available in the **GitHub repository** as an external reference for **transparency and peer review**.
- **GitHub Repository Link:** (https://github.com/MohamedNasserIV/IBM-Data-Science-Capstone-Space/blob/main/notebooks/02-web-scraping.ipynb)



Data Wrangling

- The collected datasets from **SpaceX API** and **web scraping** were merged and processed into a unified structured format.
- Data preprocessing included **handling missing values, correcting data types, removing irrelevant columns, and standardizing feature names**.
- A **binary landing outcome label** was created to support classification modeling, and categorical variables were transformed using **feature encoding techniques**.
- The complete **data wrangling notebooks** (including executed code cells and visible outputs) are provided in the **GitHub repository** as an external reference for **transparency and peer review**.
- **GitHub Repository Link:**
  (https://github.com/MohamedNasserIV/IBM-Data-Science-Capstone-Space/blob/main/notebooks/03-data-wrangling.ipynb)

EDA with Data Visualization

- Multiple **visualization charts** were created to explore relationships between key variables affecting **Falcon 9 landing success**.
- **Bar charts** were used to compare launch success rates across **launch sites and orbit types**.
- **Scatter plots** were used to analyze the relationship between **payload mass and landing outcome**.
- **Pie charts** summarized the **distribution of successful versus failed landings**.
- These visualizations helped identify **patterns, correlations, and influential factors** that guided feature selection for predictive modeling.
- The complete **EDA visualization notebook** (including executed code cells and chart outputs) is available in the **GitHub repository** as an external reference for **transparency and peer review**.
- **GitHub Repository Link:**
  (https://github.com/MohamedNasserIV/IBM-Data-Science-Capstone-Space/blob/main/notebooks/05-eda-visualization.ipynb)

EDA with SQL

- Executed **SQL queries** to explore SpaceX launch records and understand patterns related to **landing outcomes**.
- Calculated **total launches, success rates, and failure frequencies** across different **launch sites and orbit types**.
- Queried data within **specific date ranges** to analyze historical performance trends.
- Ranked **landing outcome categories** by occurrence to identify the most common mission results.
- Retrieved key mission attributes such as **booster version, payload mass, and landing outcome** to support further analysis and visualization.
- The completed **EDA with SQL notebook** (including executed queries and output results) is provided in the **GitHub repository** as an external reference for **transparency and peer review**.
- **GitHub Repository Link:**
  (https://github.com/MohamedNasserIV/IBM-Data-Science-Capstone-Space/blob/main/notebooks/04-eda-sql.ipynb)

Build an Interactive Map with Folium

- Created an interactive **Folium map** to visualize the geographic distribution of **SpaceX launch sites** and mission outcomes.
- Added **markers** to represent individual launch locations and display relevant mission information through pop-ups.
- Used **circles** to highlight launch site regions and emphasize spatial concentration of launches.
- Applied **marker clustering** to group nearby launch points and improve map readability.
- Included **lines and distance indicators** to show proximity between launch sites and key geographic features when relevant.
- These map objects were used to provide **spatial insight**, improve **visual interpretation of launch activity**, and support understanding of how **location influences landing success**.
- The completed **Folium interactive map notebook** (including executed code cells and rendered map output) is available in the **GitHub repository** as an external reference for **transparency and peer review**.
- **GitHub Repository Link:** (https://github.com/MohamedNasserIV/IBM-Data-Science-Capstone-Space/blob/main/notebooks/06-folium-map.ipynb)

Build a Dashboard with Plotly Dash

- Developed an interactive **Plotly Dash dashboard** to analyze **SpaceX Falcon 9 launch performance** dynamically.
- Added a **pie chart** to display the distribution of **successful versus failed launches** by launch site.
- Included a **scatter plot** to examine the relationship between **payload mass and landing outcome**, with color coding by **booster version category**.
- Implemented interactive **dropdown selection** for launch site filtering and a **range slider** to control payload mass range.
- These visualizations and interactions enable **real-time exploration of mission data**, improve **user-driven analysis**, and support clearer understanding of **factors influencing landing success**.
- The completed **Plotly Dash dashboard notebook** (including executed code and working application) is available in the **GitHub repository** for **transparency and peer review**.
- **GitHub Repository Link:**
  (https://github.com/MohamedNasserIV/IBM-Data-Science-Capstone-Space/blob/main/notebooks/07-dash-dashboard.ipynb)

Predictive Analysis (Classification)

- Built multiple **classification models** to predict **Falcon 9 first-stage landing success**, including: **Logistic Regression, Support Vector Machine (SVM), Decision Tree, and Random Forest**.
- Split the dataset into **training and testing sets** and applied **feature scaling and preprocessing** to prepare data for modeling.
- Evaluated model performance using **accuracy, precision, recall, F1-score, and confusion matrix** to measure prediction quality.
- Performed **hyperparameter tuning and model comparison** to improve performance and identify the **best-performing algorithm**.
- **Random Forest** achieved the strongest overall predictive performance and was selected as the **final model**.
- The completed **predictive analysis notebook** (including executed code cells, evaluation metrics, and model results) is available in the **GitHub repository** for **transparency and peer review**.
- **GitHub Repository Link:**
  (https://github.com/MohamedNasserIV/IBM-Data-Science-Capstone-Space/blob/main/notebooks/08-ml-classification.ipynb.ipynb)

Results

Exploratory Data Analysis Findings

- Launch success is strongly influenced by **payload mass, orbit type, and launch site**.
- Certain launch sites demonstrate **consistently higher success rates**.
- Higher payload ranges show **distinct landing outcome patterns**, guiding feature importance for modeling.

Interactive Analytics Demonstration

- **Folium maps** visually highlighted the **geographic distribution of launch sites and mission outcomes**.
- The **Plotly Dash dashboard** enabled interactive filtering by **launch site and payload range**, allowing dynamic exploration of success trends.
- Screenshots of the interactive analytics are included to demonstrate **functionality and insights**.

Predictive Modeling Results

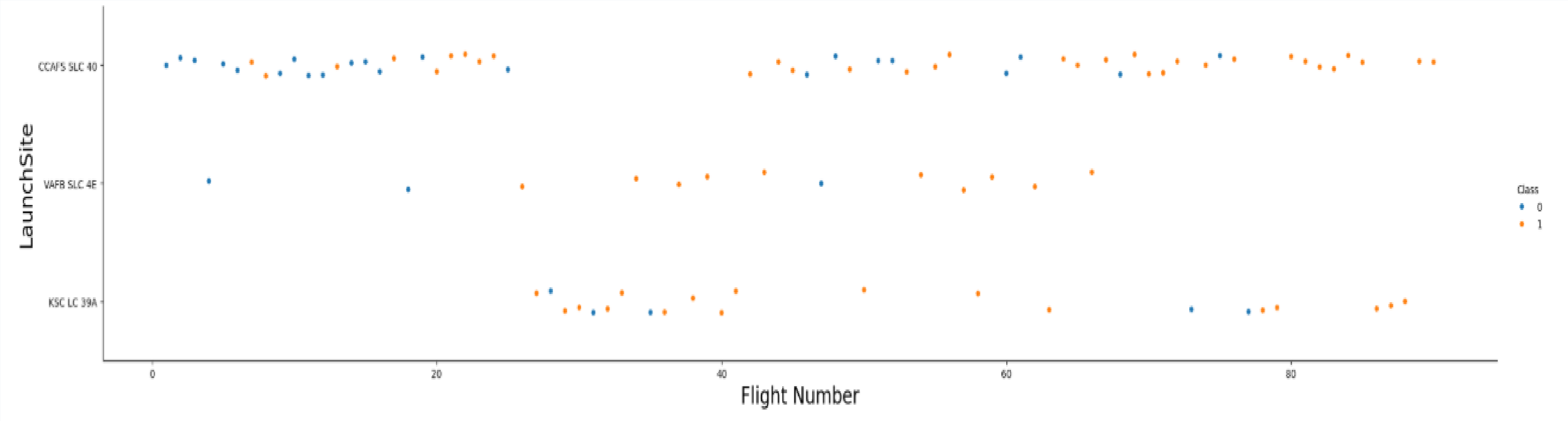
- Multiple classification algorithms were evaluated using **accuracy, precision, recall, F1-score, and confusion matrix**.
- **Random Forest** achieved the **best overall predictive performance** on the test dataset.
- The final model demonstrates the effectiveness of **data-driven prediction for reusable rocket landing success**.

Section 2

Insights drawn from EDA

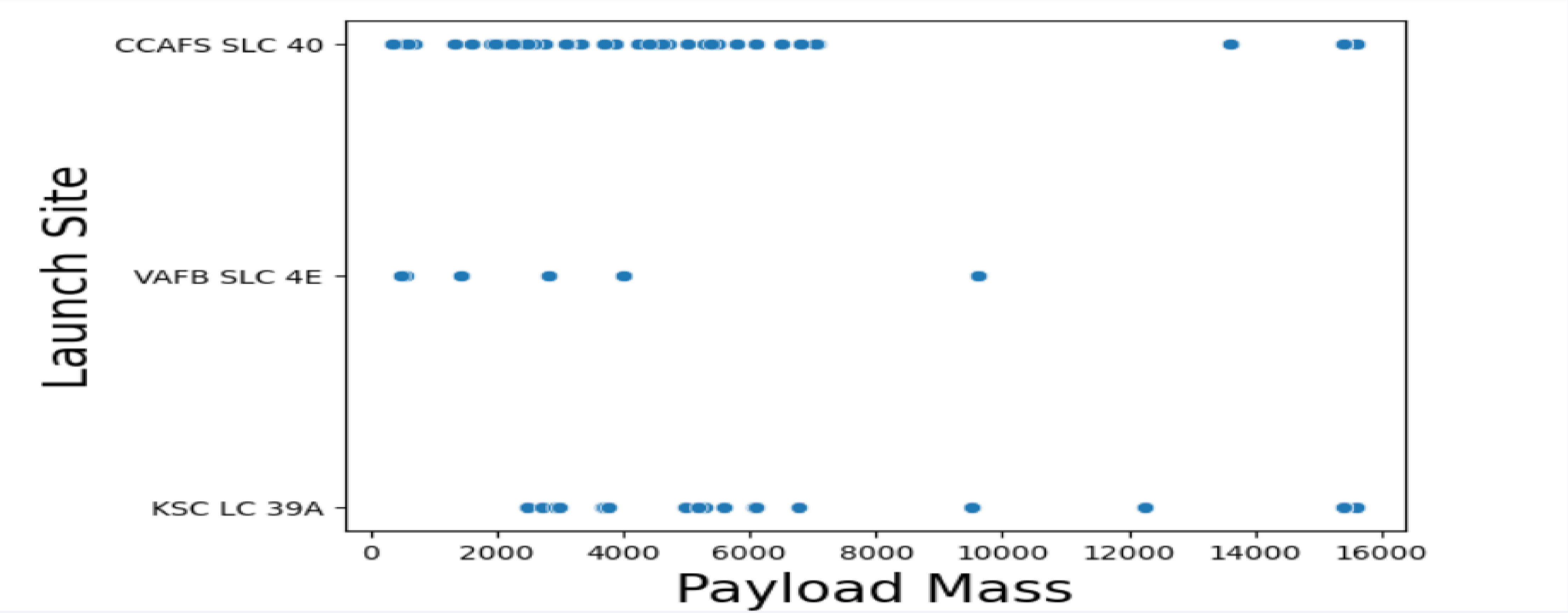
Flight Number vs. Launch Site

- The Higher The flight The bigger chance of Success rate that at some point the success rate is always 1



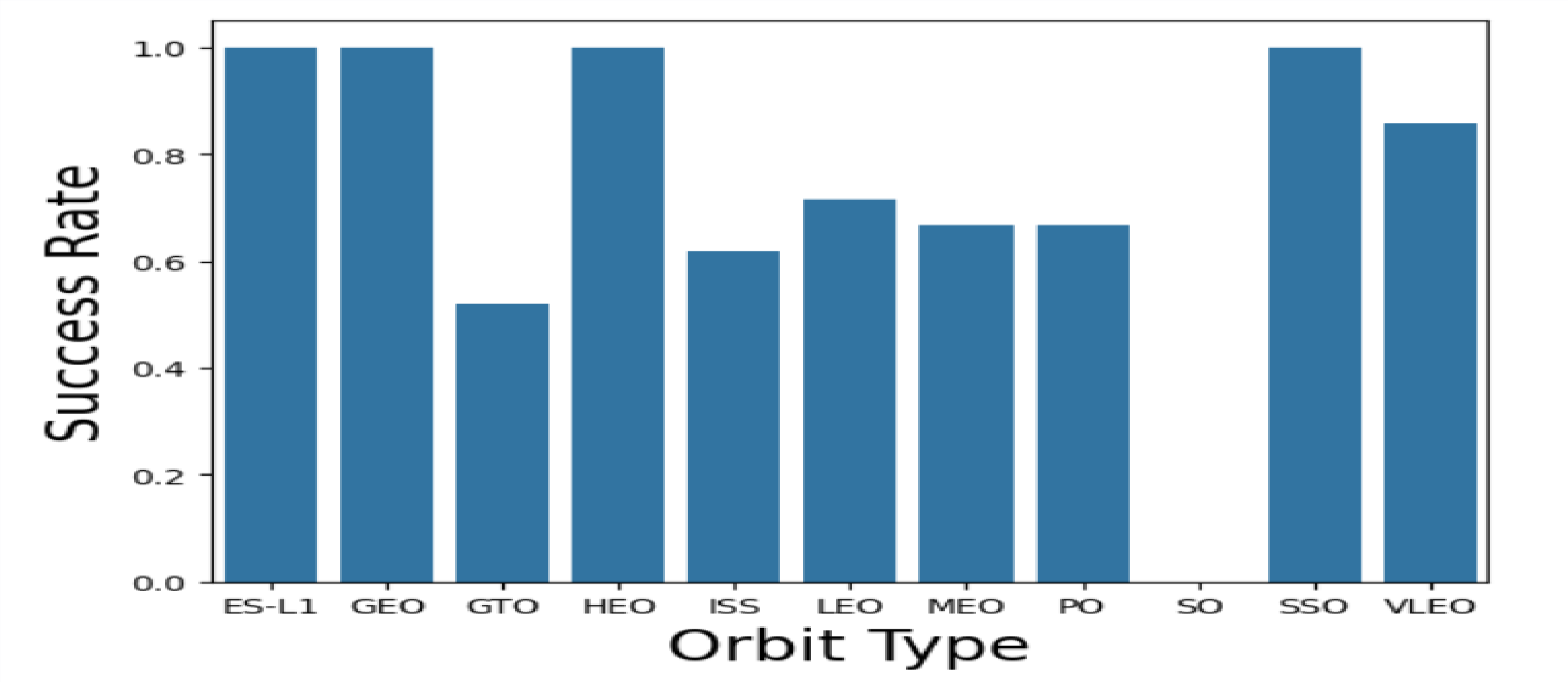
Payload vs. Launch Site

- A relationship between each Launch Site and their Payload Mass



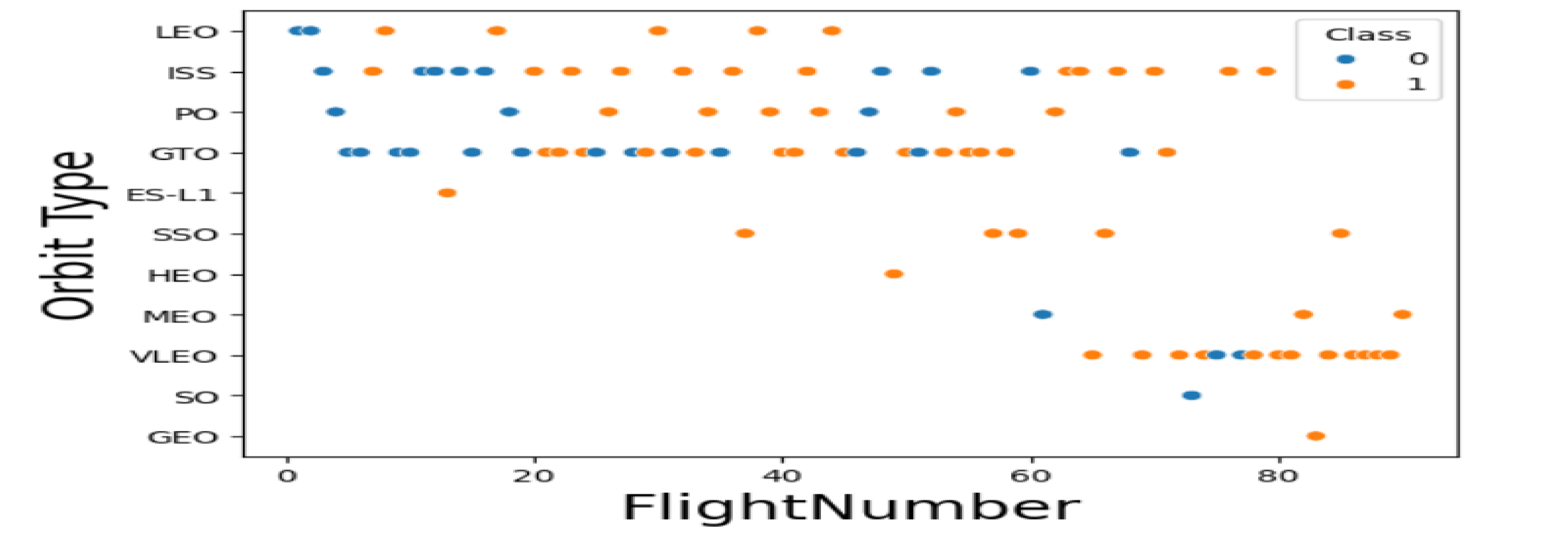
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO, VLEO Orbits came with the highest Success Rate, while GTO was the lowest



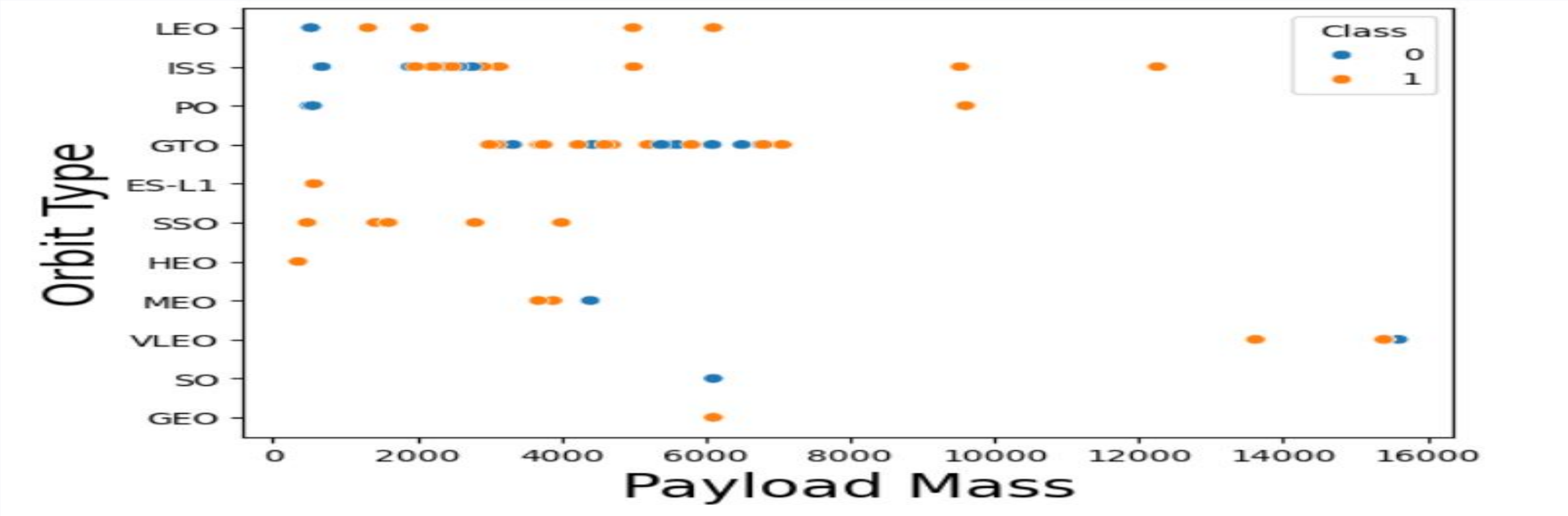
Flight Number vs. Orbit Type

- You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.



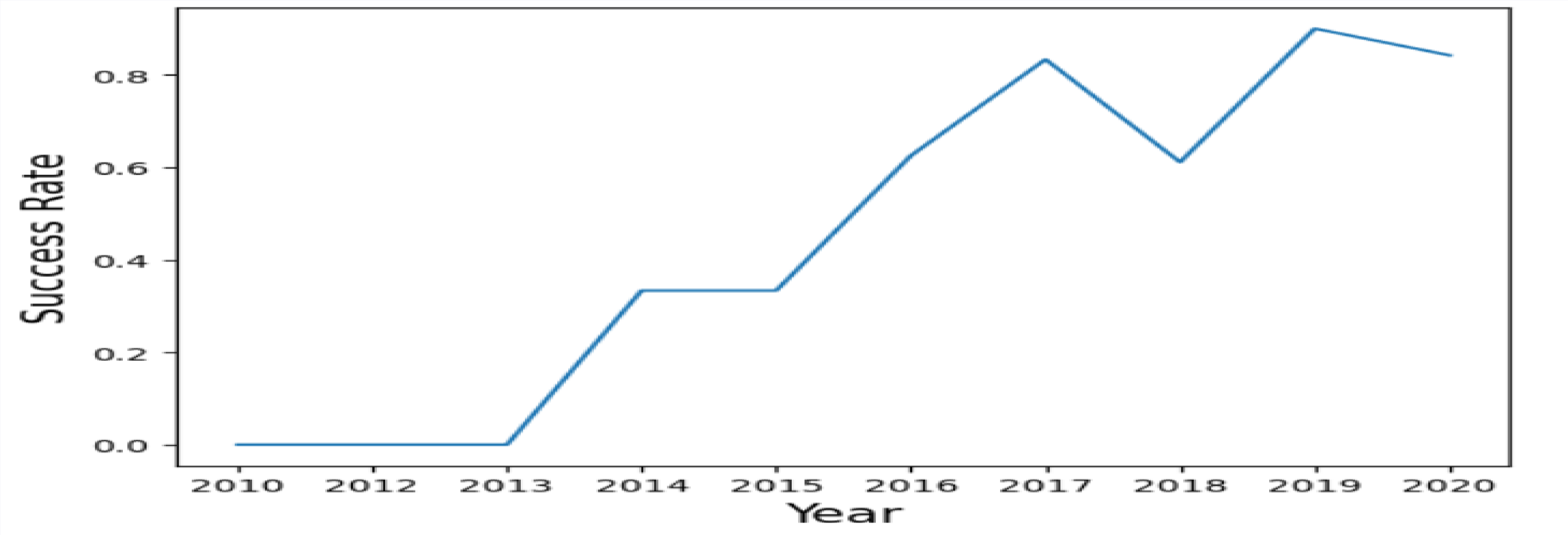
Payload vs. Orbit Type

- With heavy payloads, the successful (positive) landing rate is higher for Polar, LEO, and ISS orbits.
- However, for GTO, it is difficult to distinguish between successful and unsuccessful landings, as both outcomes are present.



Launch Success Yearly Trend

- Since 2013 The success rate kept increasing



All Launch Site Names

- Used DISTINCT to extract all Launch Site names from table

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

- Presented 5 records of Launch Site Names starting with CCA using LIKE

```
%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

Total Payload Mass

- Displayed total Payload Mass Carried by Nasa using SUM

```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload_mass
FROM SPACEXTABLE
WHERE "Customer" LIKE 'NASA%CRS%';
```

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

total_payload_mass
48213

Average Payload Mass by F9 v1.1

- Found average payload mass by F9 v1.1 using AVG

```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") AS "average_payload_mass"
FROM SPACEXTABLE
WHERE "Booster_Version" = "F9 v1.1"
```

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

average_payload_mass
2928.4

First Successful Ground Landing Date

- Listed Date of first successful ground landing using MIN

```
%%sql
SELECT MIN("Date")
FROM SPACEXTABLE
```

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

MIN("Date")
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

- Listed the names of boosters which have successfully landed on drone ship and had payload mass betweem 4000 and 6000 using AND

```
%%sql
SELECT "Booster_Version"
FROM SPACEXTABLE
WHERE "Landing_Outcome" LIKE 'Success (drone ship)%'
AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 60000
```

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

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1036.1
F9 B4 B1041.1
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Total Number of outcomes using COUNT

```
%%sql
SELECT COUNT ("Landing_Outcome") as "number of outcomes" FROM SPACEXTABLE
```

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

number of outcomes
101

Boosters Carried Maximum Payload

- Using Subqueries we could list the boosters that carried maximum payload

```
%%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

- Used substr to extract 2015 Launch Records

```
%%sql
SELECT substr(Date, 6,2), "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.
```

substr(Date, 6,2)	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

- Used GROUPBY, ORDERBY, DESC, substr, BETWEEN and AND to rank Landing outcomes between 2010-06-05/2017-03-20

```
%%sql
SELECT "Landing_Outcome", Count("Landing_Outcome") AS "Frequency"
FROM SPACEXTABLE
WHERE substr("Date",1,10) BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY COUNT("Landing_Outcome") DESC;
```

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

Landing_Outcome	Frequency
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

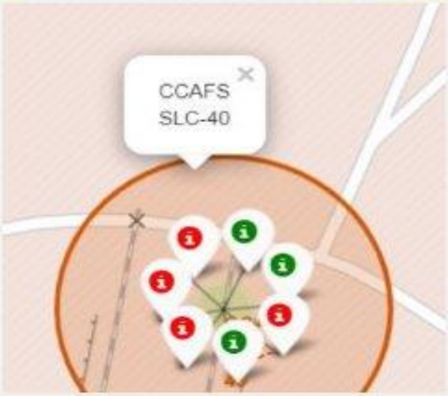
Section 3

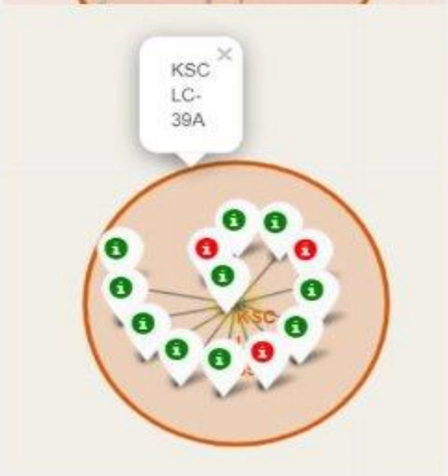
Launch Sites Proximities Analysis

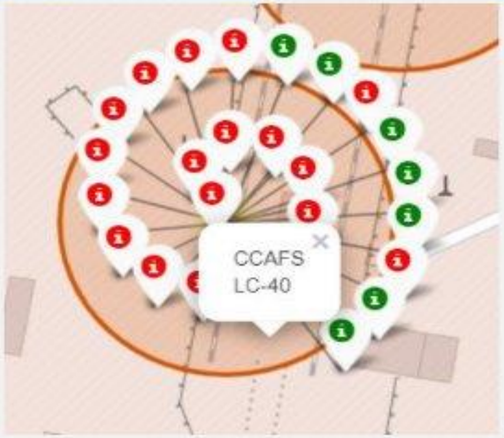
All Launch Sites Locations

As you can notice all launch Sites are located in USA in Florida and California

Success Rate of each Site







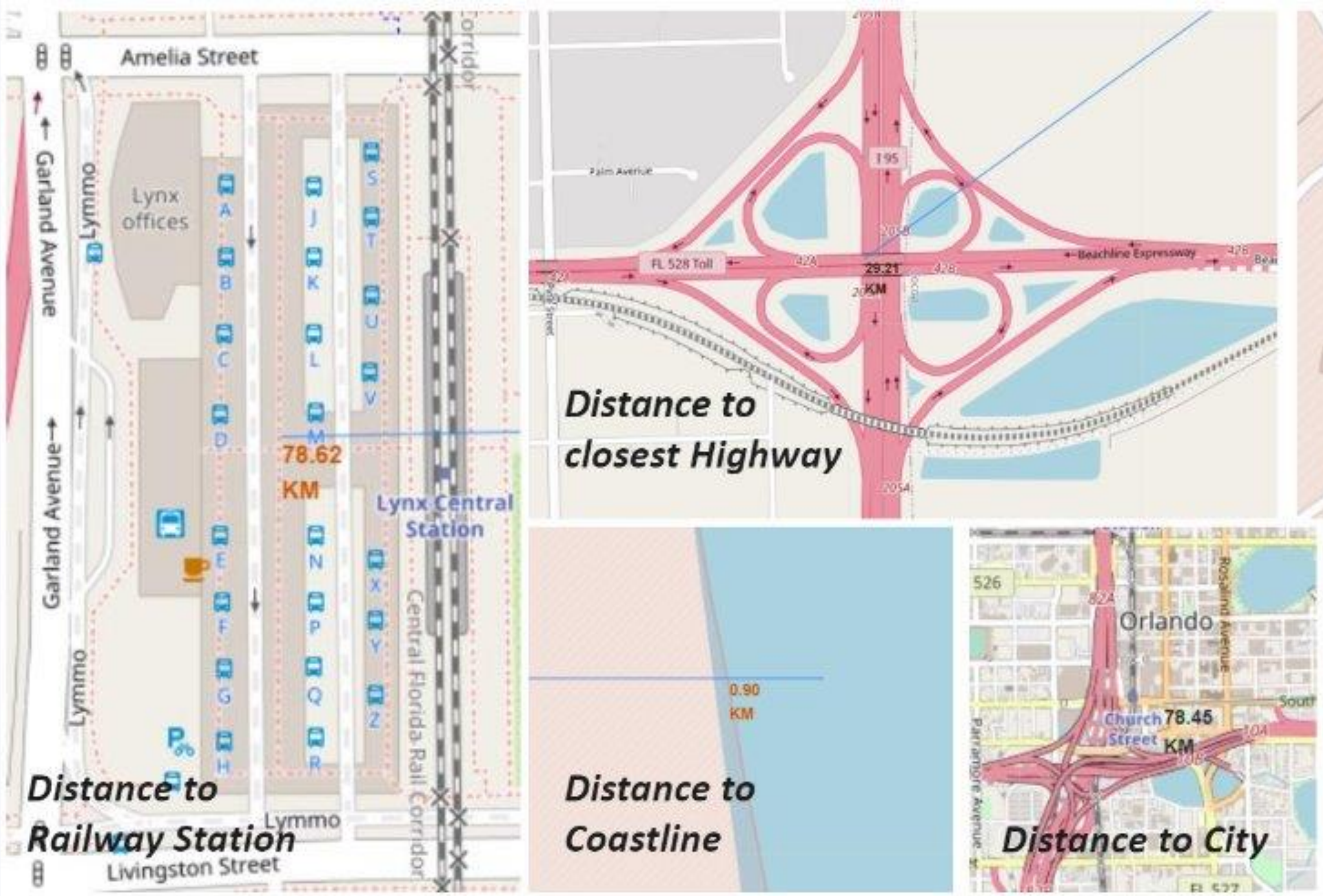
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

Distance between Site and Landmarks



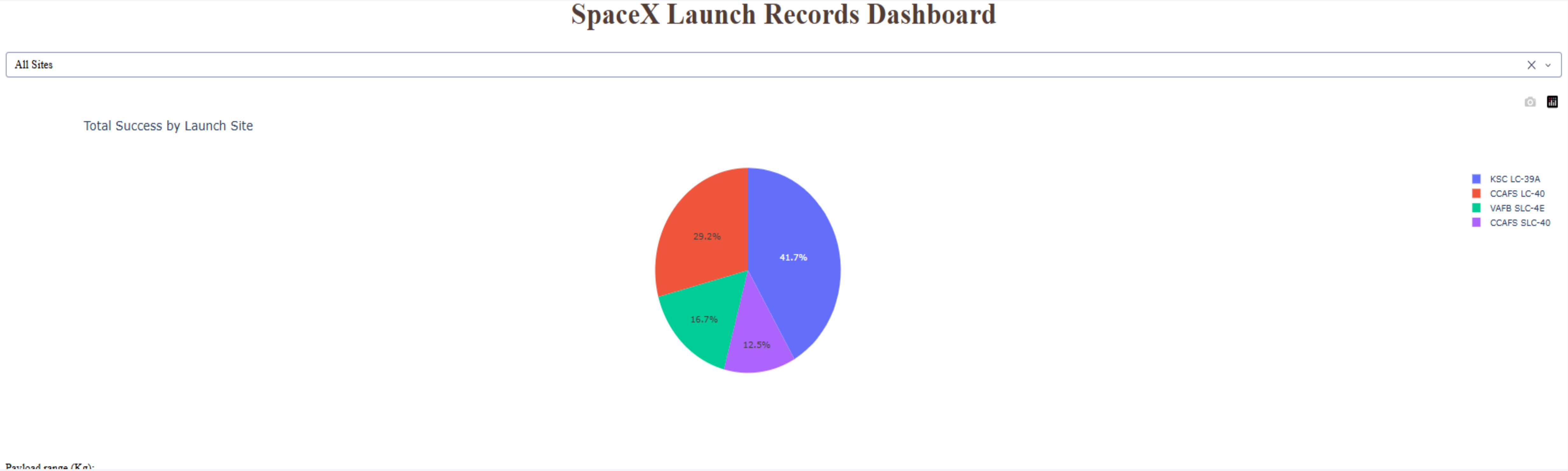
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

Build a Dashboard with Plotly Dash

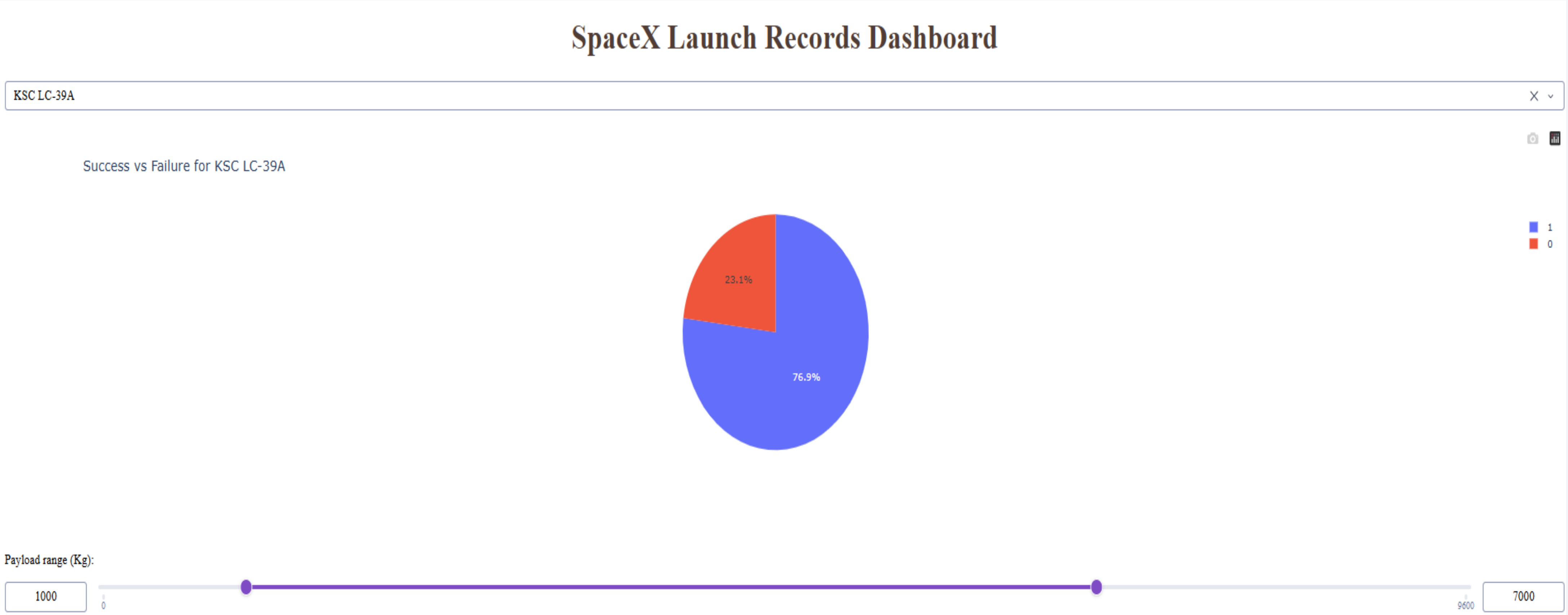
Success Rate of each Site

- KSC LC-39A came with the highest success rate

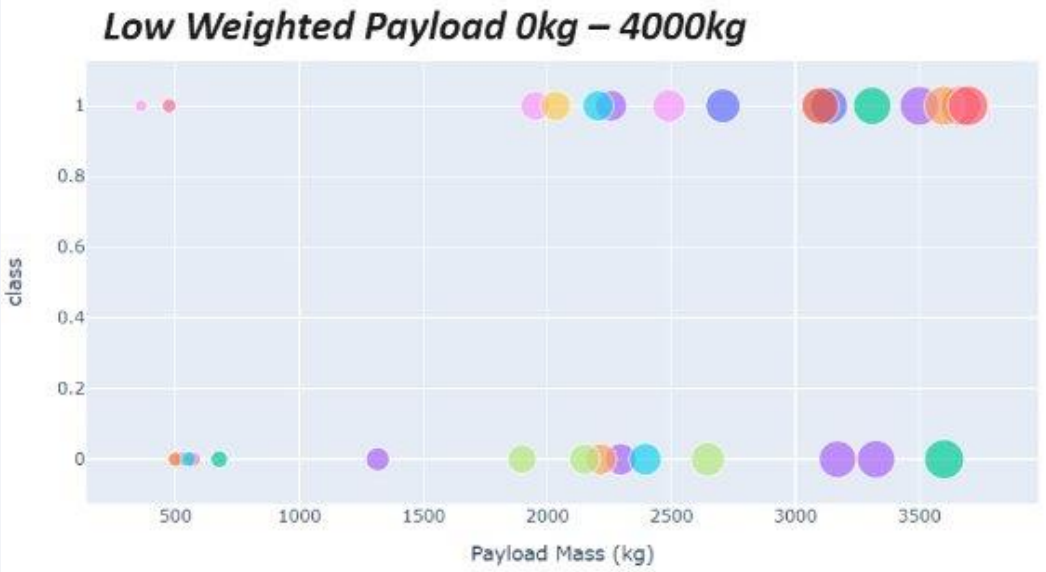


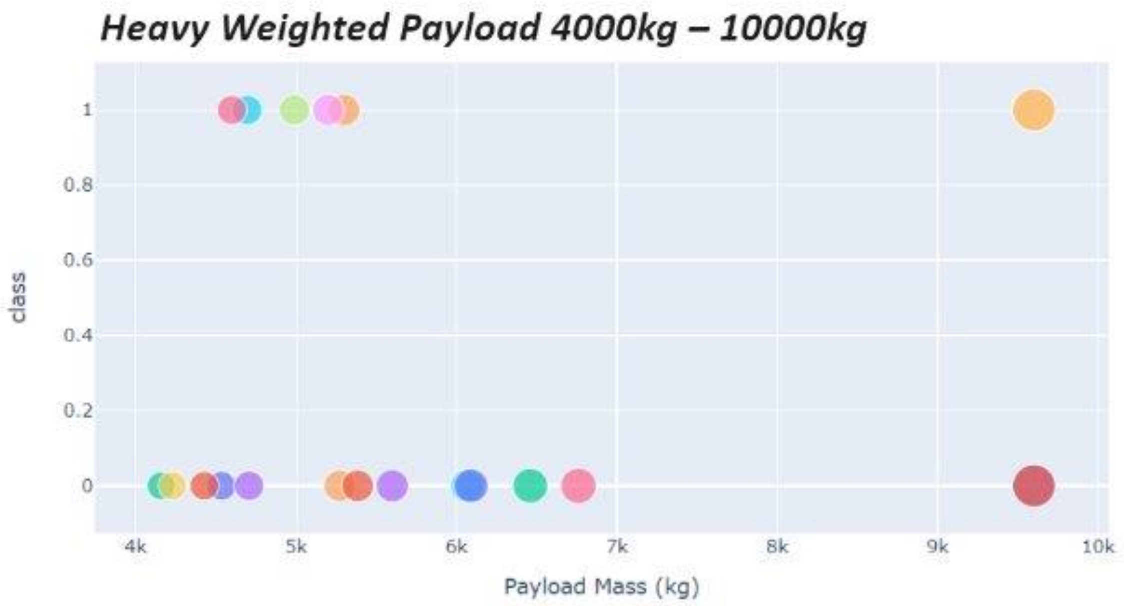
Site with highest Success Rate

- Success Rate of the KSC LC-39A



Payload vs Launch Outcomes for all sites





We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Decision Tree came with the highest accuracy

```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

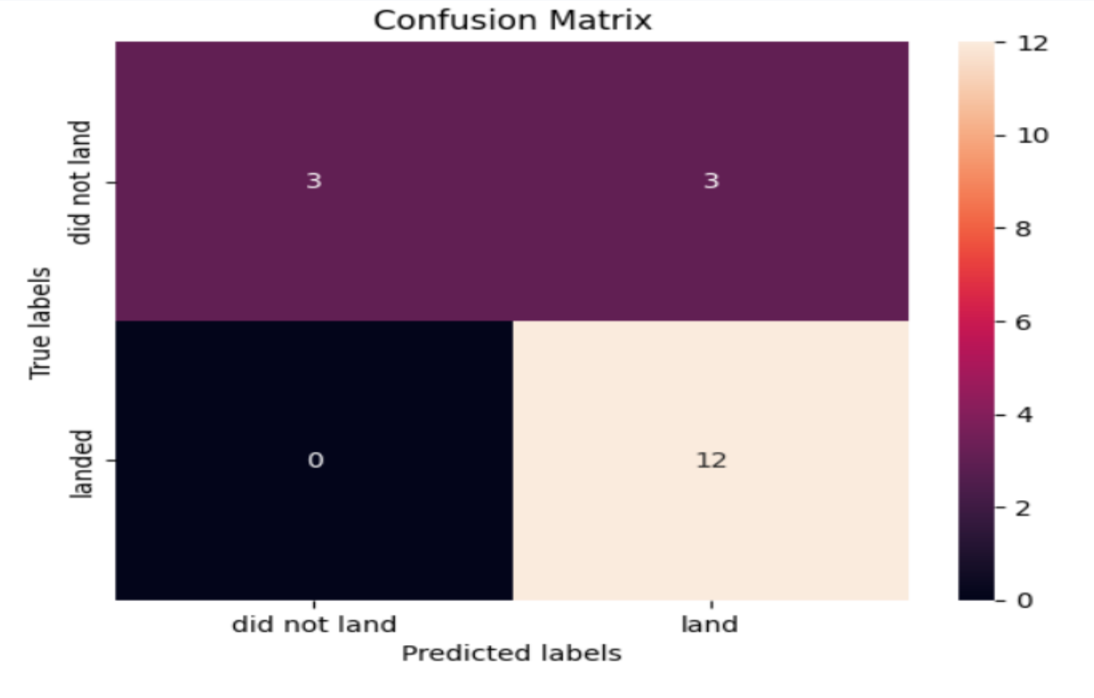
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_sp
lit': 2, 'splitter': 'random'}
accuracy : 0.875
```

Confusion Matrix

- Confusion matrix of decision tree showing Logistic regression can differentiate between the various classes, but the main issue observed is the occurrence of false positives.



Conclusions

- The success rate increases as the Flight Number increases
- Success rate has been increasing since 2013
- Success rate is higher when it's in orbits: ES-L1, GEO, HEO, SSO, VLEO.
- KSC LC-39A had the most successful launches of any sites
- The Decision tree classifier is the best machine learning algorithm for this task.

Appendix

- GitHub Repository including all notebooks, screenshots and the presentation: https://github.com/MohamedNasserIV/IBM-Data-Science-Capstone-Space

Thank you!