IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

João Pinto
April 26th, 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling and Processing
  - Exploratory Data Analysis
  - Interactive Visual Analytics
  - Predicitive Modeling
- Summary of all results
  - Insights from EDA;
  - Visualization Impact
  - Best Performing Model

Introduction

- This project is part of the IBM Data Science Professional Certificate program, designed to develop practical skills in data analysis, data visualization, and machine learning. The project focuses on applying real-world data science techniques using live data from SpaceX's public API
- Key Problems to Solve

  Which factors most influence SpaceX launch success?

  Can we predict future mission outcomes using machine learning?

  Which launch sites, rockets, and payloads show the highest success rates?

  Which classification model provides the best predictions?

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

- Datasets were collected by identifying relevant and credible sources, extracting data via web scraping, APIs, or direct downloads, assessing data quality (checking for missing values and duplicates), and securely storing the cleaned data for further analysis.
- data collection process using flowcharts

[Identify Data Sources]

↓

[Extract Data (Scraping / API / Download)]

↓

[Assess Data Quality (Missing Values, Duplicates)]

↓

[Store Data (Organized and Secured)]

Data Collection – SpaceX API

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[9]:  static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_ap
```

We should see that the request was successfull with the 200 status response code

```
[10]: response=requests.get(static_json_url)
```

```
[11]: response.status_code
```

```
[11]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
[12]: # Use json_normalize meethod to convert the json result into a dataframe

data = response.json()
df = pd.json_normalize(data)
```

Using the dataframe `data` print the first 5 rows

```
[14]: # Get the head of the dataframe
print(df.head())
```

```
       static_fire_date_utc  static_fire_date_unix    tbd    net  window  \
0  2006-03-17T00:00:00.000Z           1.142554e+09  False  False     0.0
1                      None                    NaN  False  False     0.0
2                      None                    NaN  False  False     0.0
```

Github: https://github.com/joaogpinto04/IBM-Data-Science/blob/7cc9ce7a817e6cdafd91712cb4c882362b5a1ca7/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please che
external reference link towards the end of this lab

]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
html_tables = soup.find_all('table')
# Assign the result to a list called `html_tables`
print(f"Número de tabelas encontradas: {len(html_tables)}")
```

```
Número de tabelas encontradas: 25
```

Starting from the third table is our target table contains the actual launch records.

]:
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

GitHub: https://github.com/joaogpinto04/IBM-Data-Science/blob/7cc9ce7a817e6cdafd91712cb4c882362b5a1ca7/jupyter-labs-webscraping.ipynb

Data Wrangling

- Data was collected vir Rest API

[Initial Data Inspection]

↓

[Data Cleaning (Missing Values, Duplicates, Standardization)]

↓

[Feature Engineering (New Features, Encoding)]

↓

[Data Integration (Merging Datasets)]

↓

[Data Filtering (Selecting Relevant Features)]

Github: https://github.com/joaogpinto04/IBM-Data-Science/blob/2e7b3d8f455e0cd7657f6b8ec0a515a7a7ced7af/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- During the EDA phase, several charts were created to understand patterns, relationships, and trends within the SpaceX dataset:

  Charts Used: Histograms; Scatter Plots; Box Plots; Heatmaps; Pie Charts;

Github: https://github.com/joaogpinto04/IBM-Data-Science/blob/b17331853aedf18f33502b808970fe76802806e4/edadataviz.ipynb

EDA with SQL

- The SQL queries that I performed were SUM, AVG, DROP, MIN,GROUPBY,WHERE…
- GitHub URL: https://github.com/joaogpinto04/IBM-Data-Science/blob/fe9bb8f92270b5ca7bbff5df6f96377583588857/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Map Objects Created:

- **Markers**:
- Placed at each launch site location to indicate site names and basic information (e.g., site name, success rates).
- **Circle Markers**:
- Added to visually represent launch site activity. The size and color of the circles indicated the number of launches or success rates, making it easy to compare sites at a glance.
- **Popups**:
- Attached to each marker to provide more detailed information when clicked (e.g., total launches, success rate, most recent mission).
- **Polylines**:
- (Optional if used) Used to connect launch sites to payload destination points, helping to illustrate flight paths.

Why These Objects Were Added:

- **Markers and Popups**:
- To provide a clear, interactive way to view detailed information about each launch site without cluttering the map.
- **Circle Markers**:
- To quickly visualize and compare the activity levels and success rates across different launch sites in a meaningful way.

GitHub: https://github.com/joaogpinto04/IBM-Data-Science/blob/4a3ee0264fa828af8898d50247c349288e0c5774/lab_jupyter_launch_site_location.ipynb

Predictive Analysis (Classification)

Key Phrases:

- **Model Selection**:
- Chose several classification algorithms — Logistic Regression, SVM, Decision Tree, and k-Nearest Neighbors (k-NN).
- **Model Training**:
- Trained each model using the training data split from the full dataset.
- **Hyperparameter Tuning**:
- Used **GridSearchCV** with cross-validation (cv=10) to find the best hyperparameters for each model.
- **Model Evaluation**:
- Evaluated models based on their accuracy on validation and test datasets, ensuring fair comparison.
- **Model Comparison**:
- Compared test accuracies to select the best performing model.

GitHub URL: https://github.com/joaogpinto04/IBM-Data-Science/blob/6b06ceb8c50cc906cf81ba62dbdf7e83b8f4e3b9/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

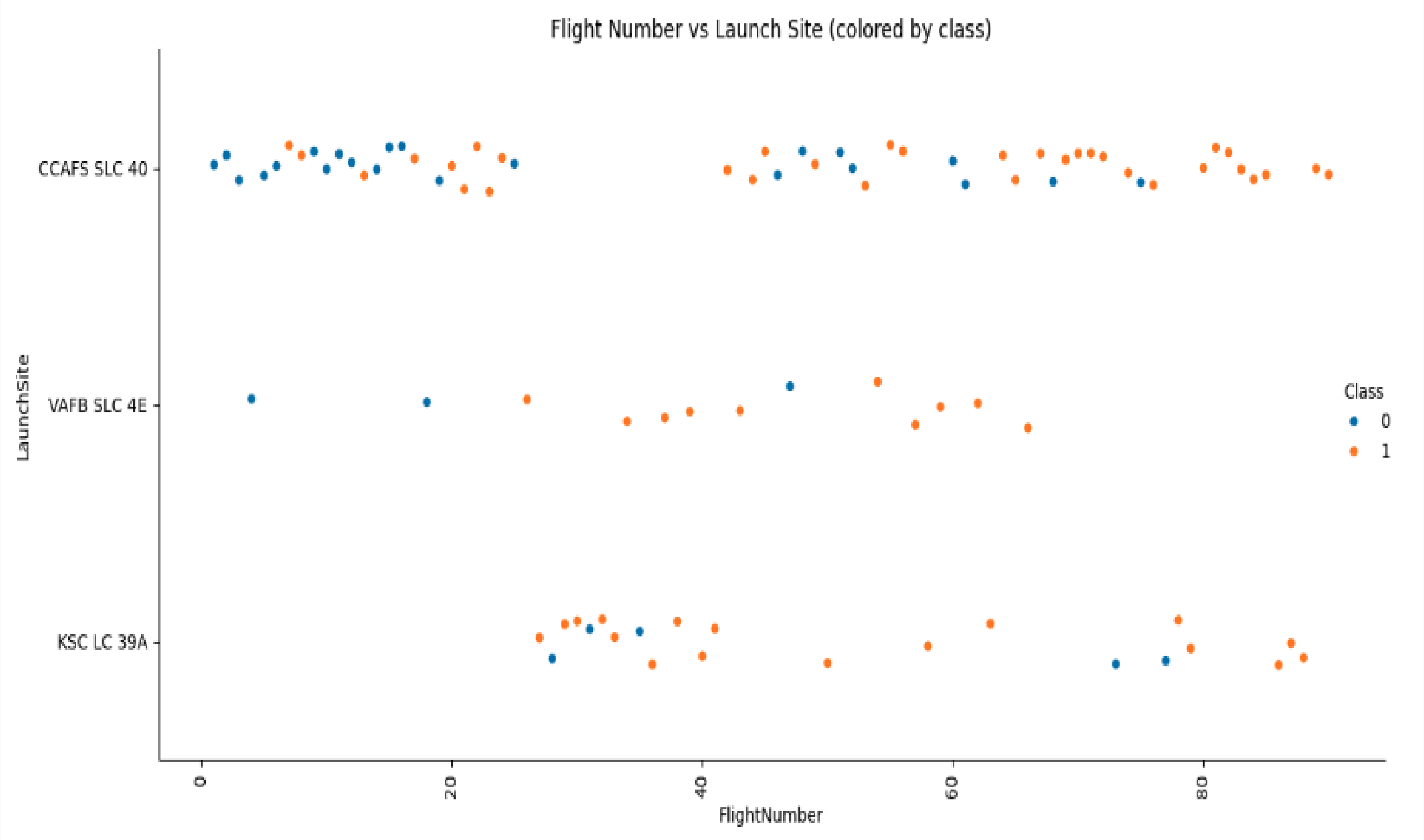
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

Insights:

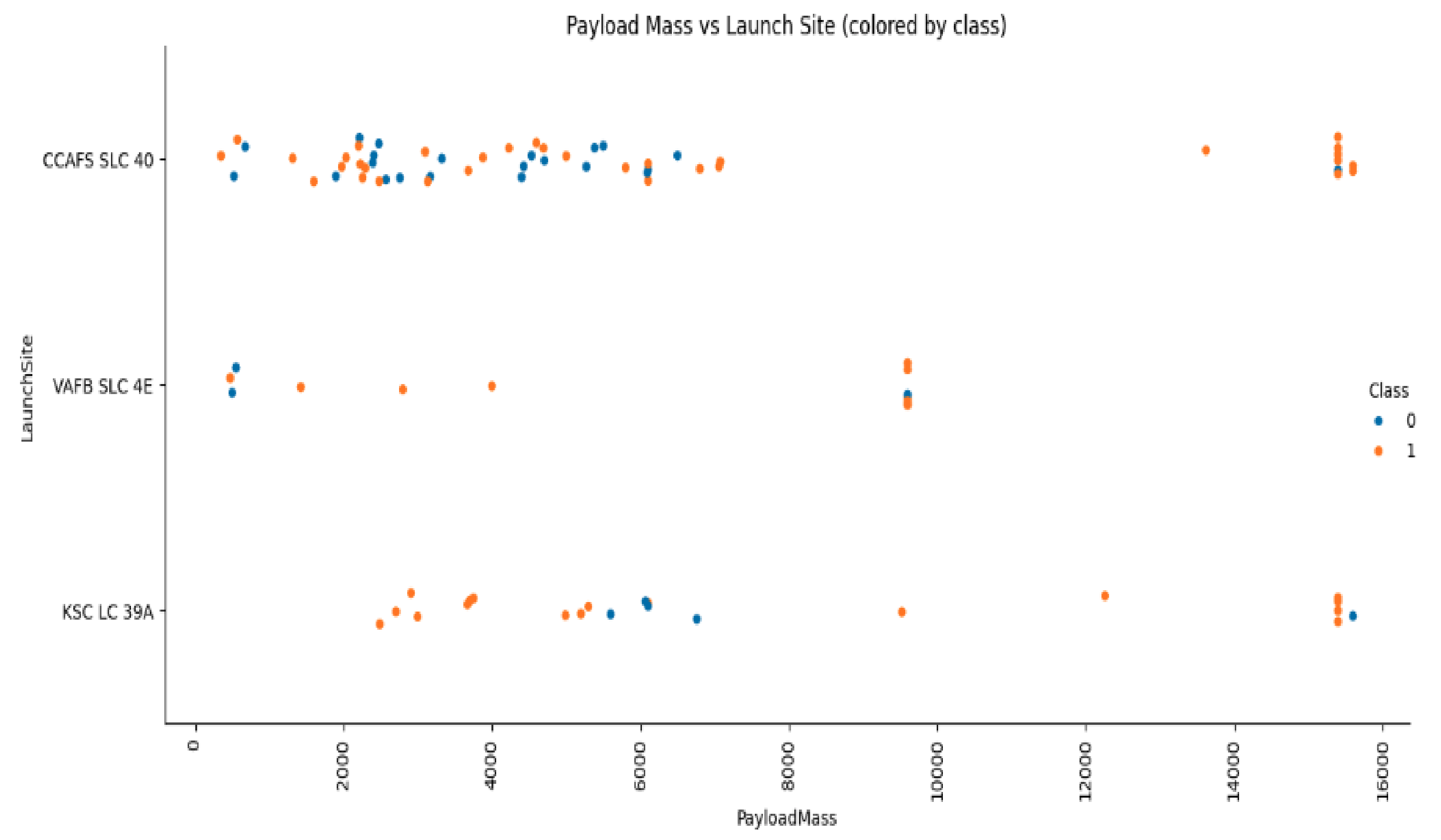
- Over time (as the flight number increases), we observe an increase in the number of **successful launches** (orange points) across different launch sites.
- **CCAFS SLC 40** appears to have the highest number of launches, followed by **KSC LC 39A** and **VAFB SLC 4E**.
- **Early flight numbers** at some sites show more **failures** (more blue dots), suggesting improvements in reliability as the number of missions increased.



Payload vs. Launch Site

Insights:

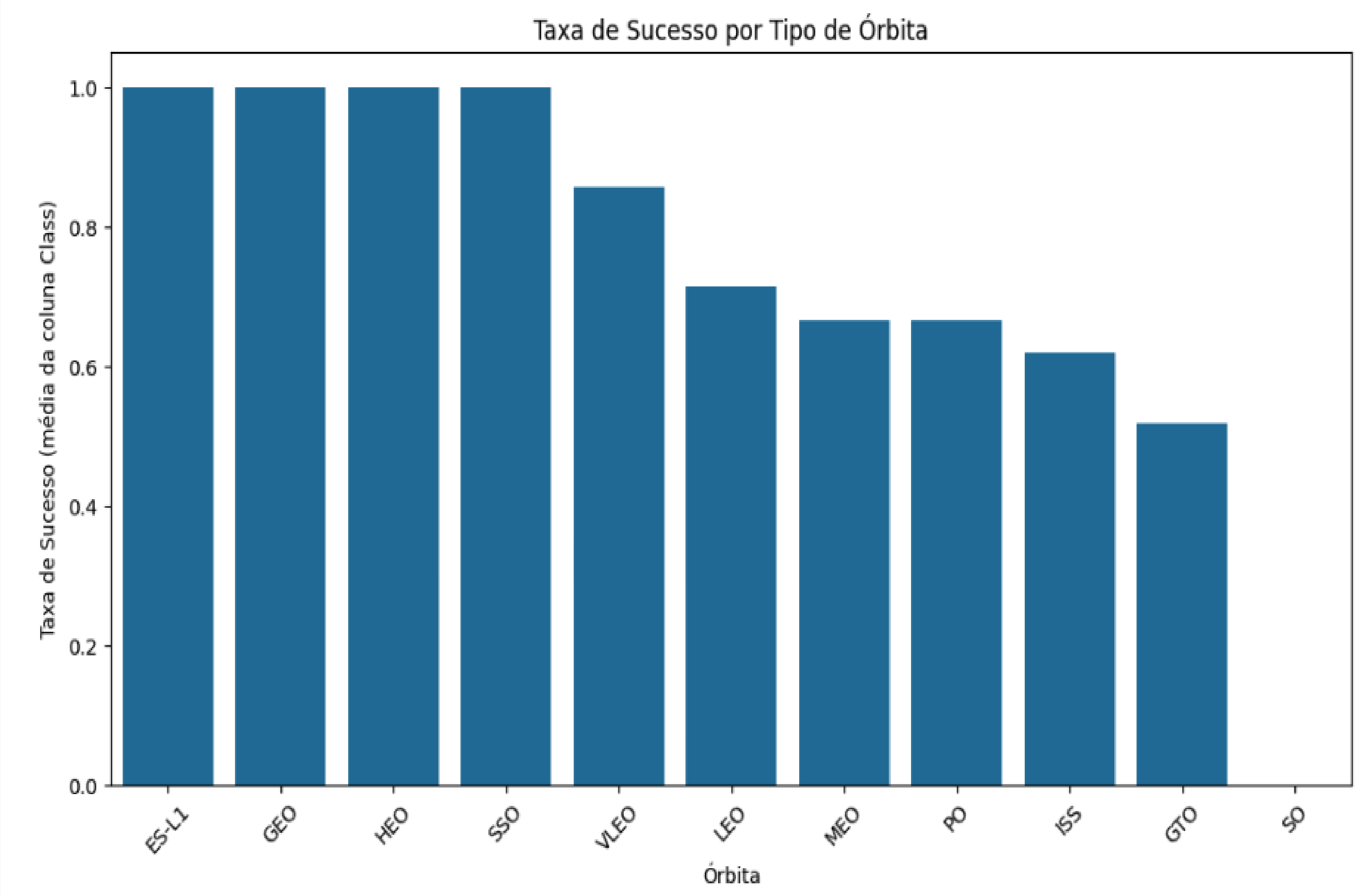
- Over time (as the flight number increases), we observe an increase in the number of **successful launches** (orange points) across different launch sites.
- **CCAFS SLC 40** appears to have the highest number of launches, followed by **KSC LC 39A** and **VAFB SLC 4E**.
- **Early flight numbers** at some sites show more **failures** (more blue dots), suggesting improvements in reliability as the number of missions increased.



Success Rate vs. Orbit Type

Insights:

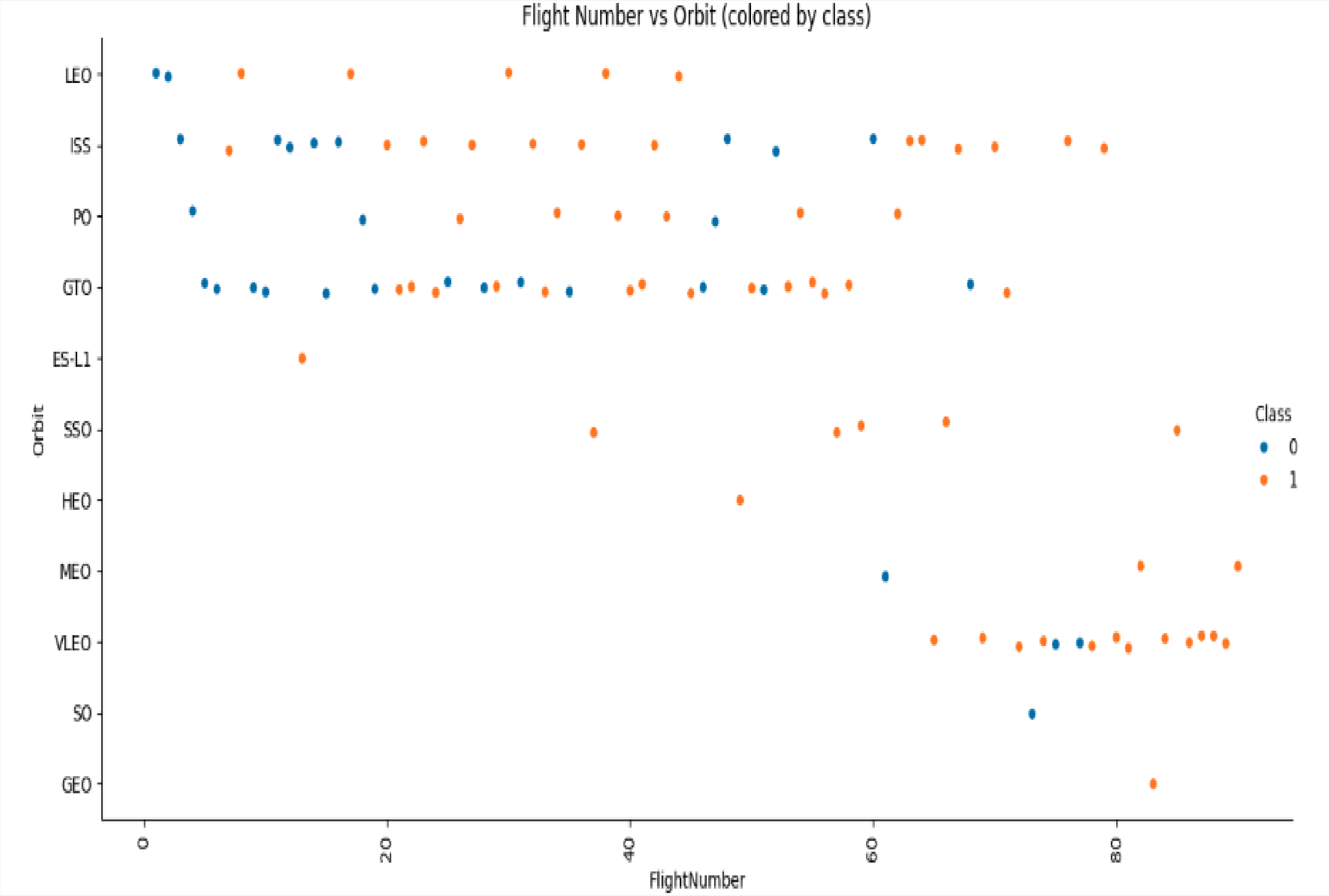
- Certain orbit types such as **ES-L1, GEO, HEO, and SSO** show a **100% success rate**.
- **VLEO** (Very Low Earth Orbit) and **LEO** (Low Earth Orbit) have slightly lower but still relatively high success rates.
- **GTO** (Geostationary Transfer Orbit) and **SO** (possibly "Sub-Orbital" or an uncommon category) have the **lowest success rates** among the listed orbit types.
- This suggests that missions aiming for higher or more complex orbits (like GTO) may involve greater technical challenges leading to lower success percentages.



Flight Number vs. Orbit Type

Insights:

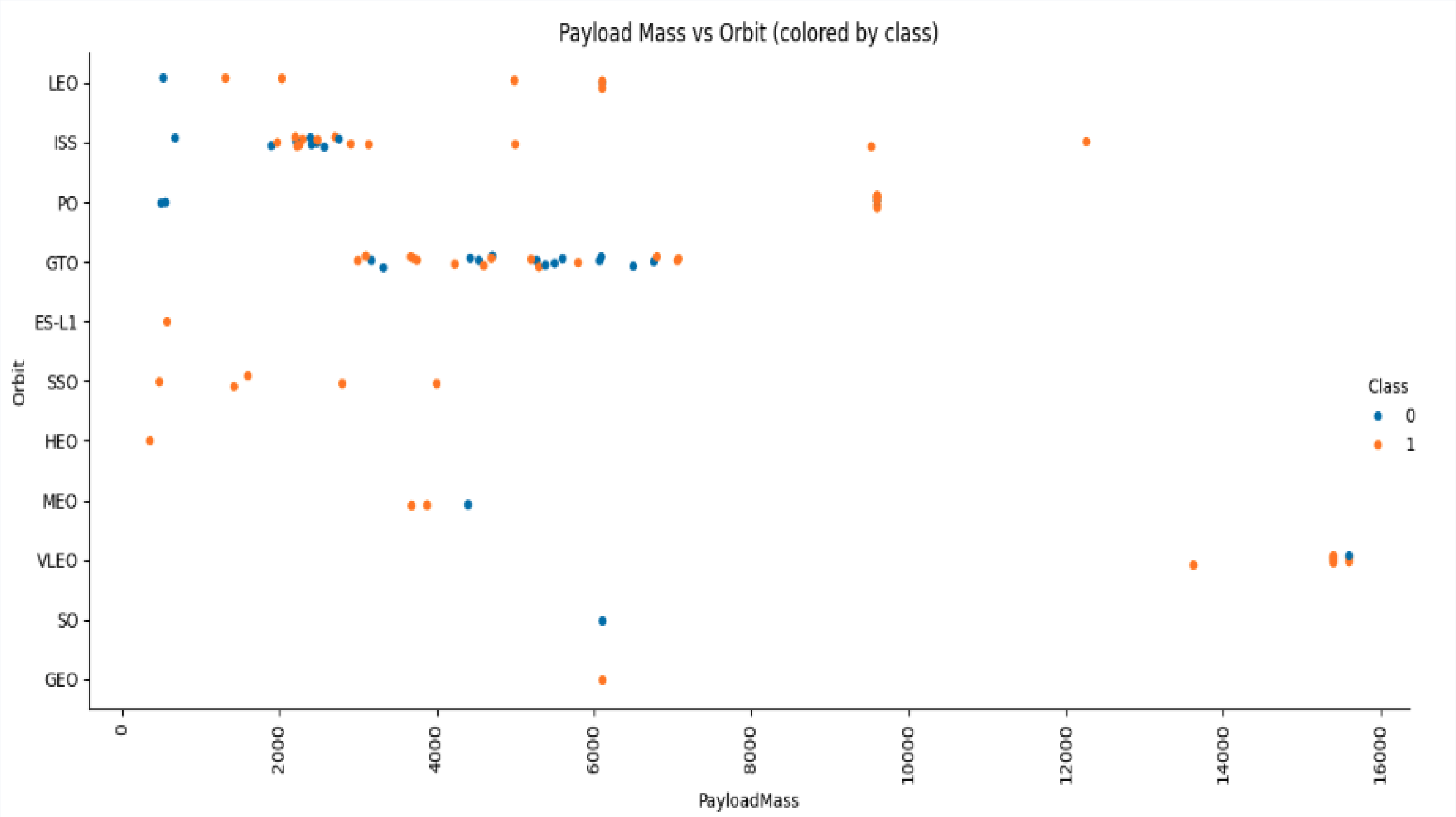
- **Early flight numbers** (left side) show more **failures** (blue dots), especially for challenging orbits like **GTO** and **LEO**.
- **Later flights** (higher flight numbers) show **more consistent success** (more orange dots), indicating SpaceX's growing reliability over time.
- Some orbits, like **ES-L1** and **SSO**, are almost exclusively associated with successful missions.
- **VLEO** missions, which cluster towards higher flight numbers, show a mixed performance, reflecting the challenges of low orbit missions.



Payload vs. Orbit Type

Insights:

- For orbits like **ISS**, **PO**, and **GTO**, the payload mass tends to be higher, with a significant number of missions around **4000–6000 kg** and even beyond **10000 kg**.
- **Success rates** (orange dots) remain high across most payload ranges, but some **failures** (blue dots) occur particularly with mid-range payloads (~4000–6000 kg), especially for **GTO** missions.
- Missions targeting **GEO**, **VLEO**, and **SO** show varied success depending on the payload mass, but the number of failures is more visible compared to other orbits.

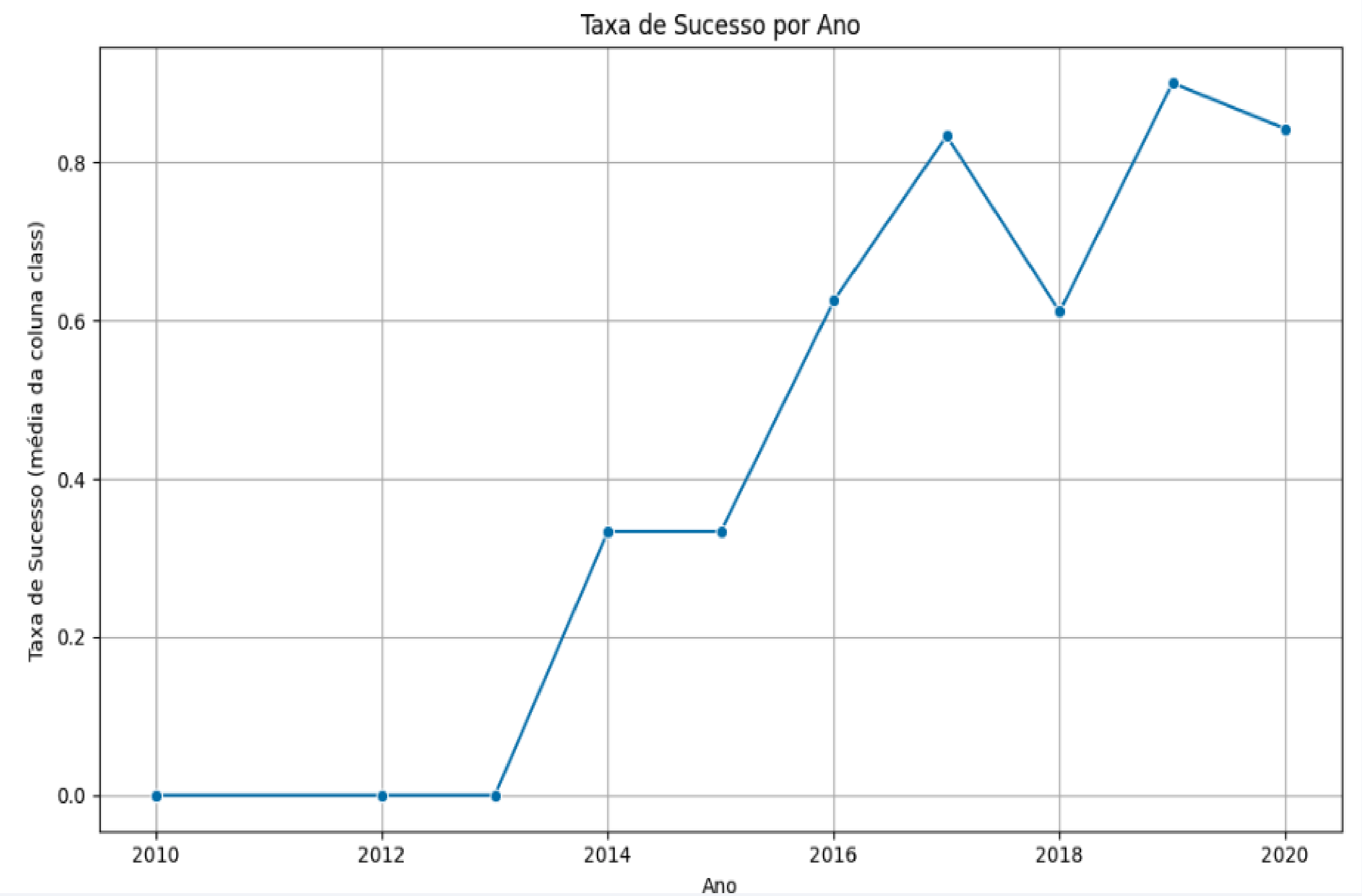


Launch Success Yearly Trend

Insights:

- From **2010 to 2013**, the success rate was **0**, indicating no successful launches in those years.
- A noticeable improvement began in **2014**, with the success rate steadily increasing.
- Significant success was achieved from **2016 onwards**, with rates above **60%**.
- **2019** marked the peak success rate, reaching close to **90%**.
- Although there was a slight drop in **2020**, the success rate remained consistently high.

- This trend clearly shows SpaceX's **technological improvement and operational maturity** over the years.



All Launch Site Names

- Names of the unique launch sites

```
features= df[['LaunchSite']]
features.head()
```

	LaunchSite
0	CCAFS SLC 40
1	CCAFS SLC 40
2	CCAFS SLC 40
3	VAFB SLC 4E
4	CCAFS SLC 40

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

```
[ ]  1 %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

- the total payload carried by boosters from NASA was 48.213 Kg

```
%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Customer" like 'NASA (CRS)%'
```

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

SUM(PAYLOAD_MASS__KG_)
48213

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 = 2534.7Kg

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where "Booster_Version" LIKE 'F9 v1.1%'

 * sqlite:///my_data1.db
Done.

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665
```

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad

```
[ ]  1 %sql SELECT MIN(Date) AS first_successful_landing_date FROM SPACEXTABLE  WHERE landing_outcome = 'Success (ground pad)';
```

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

first_successful_landing_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The 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 SPACEXTABLE WHERE landing_outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_
```

```
 * 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

- The total number of successful and failure mission outcomes

```
%sql SELECT mission_outcome, COUNT(*) AS total_count FROM SPACEXTABLE WHERE  mission_outcome IN ('Success', 'Failure (in flight)') GROUP BY mis
```

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

Mission_Outcome	total_count
Failure (in flight)	1
Success	98

Boosters Carried Maximum Payload

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

```
%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

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

```
[ ] 1 %sql SELECT CASE WHEN substr(Date, 6, 2) = '01' THEN 'January' WHEN substr(Date, 6, 2) = '02' THEN 'February' WHEN substr(Date, 6, 2) = '03' TH
    2

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

month_name	Booster_Version	Launch_Site	Landing_Outcome
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- Rank 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 SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing_outc
```

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

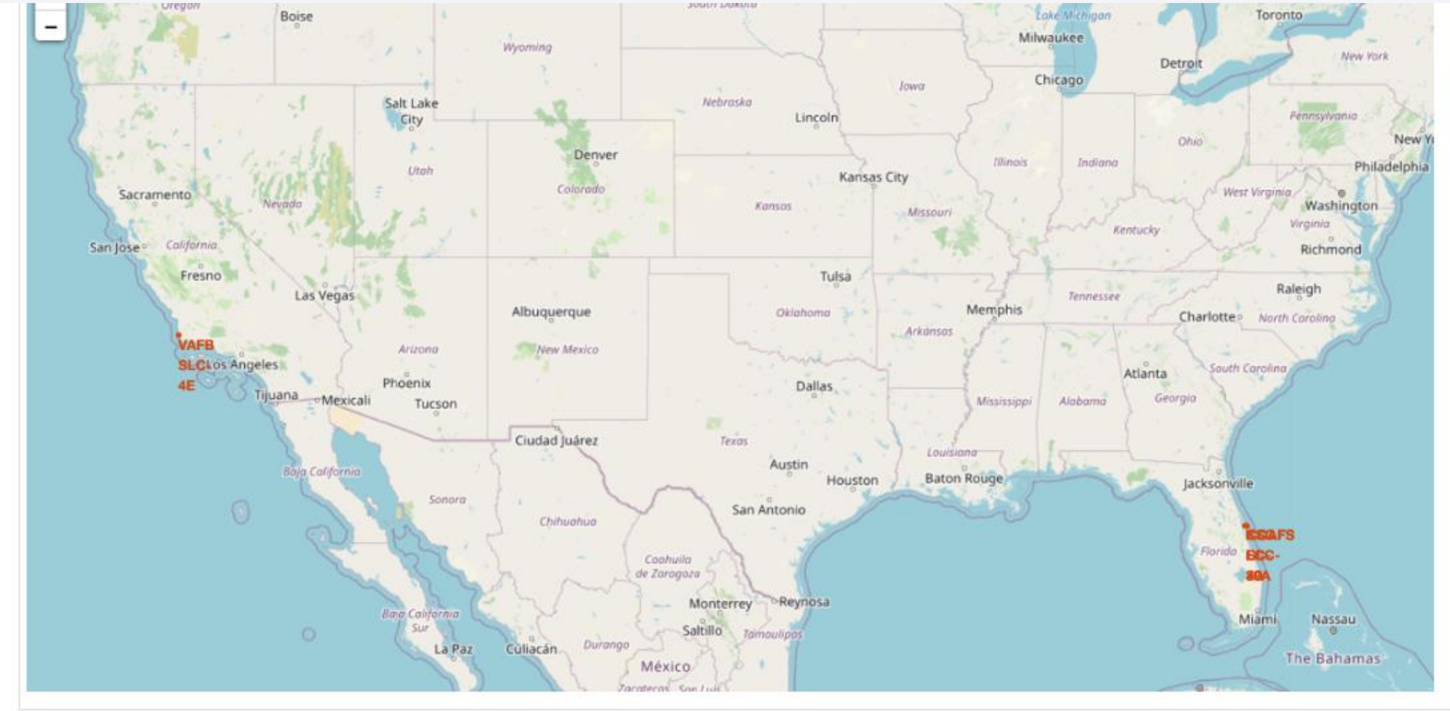
Landing_Outcome	outcome_count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

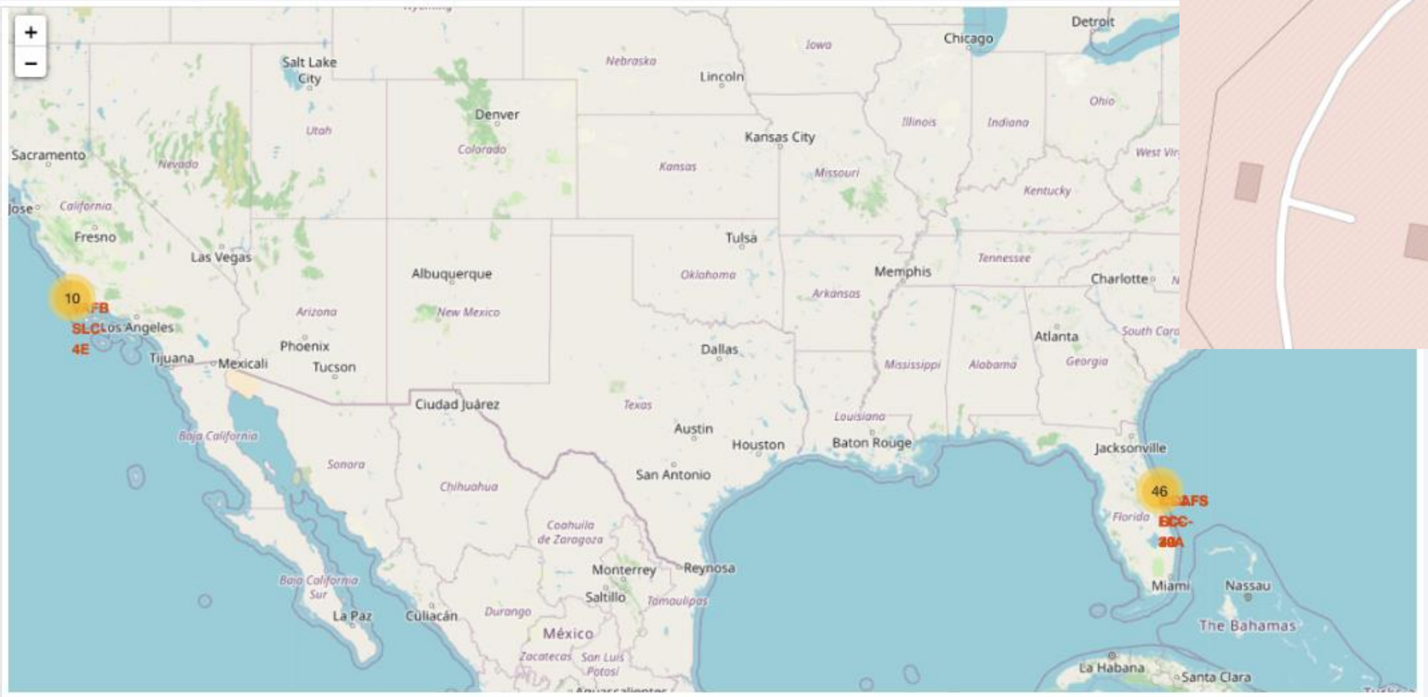
Launch Sites Proximities Analysis

USA Launch sites

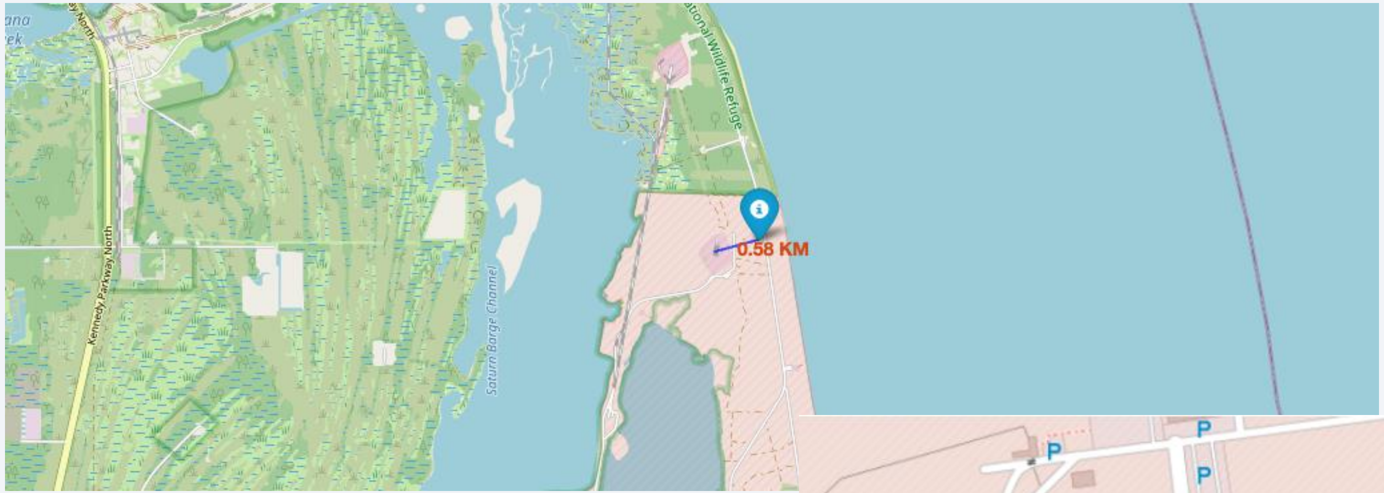
- USA launch sites

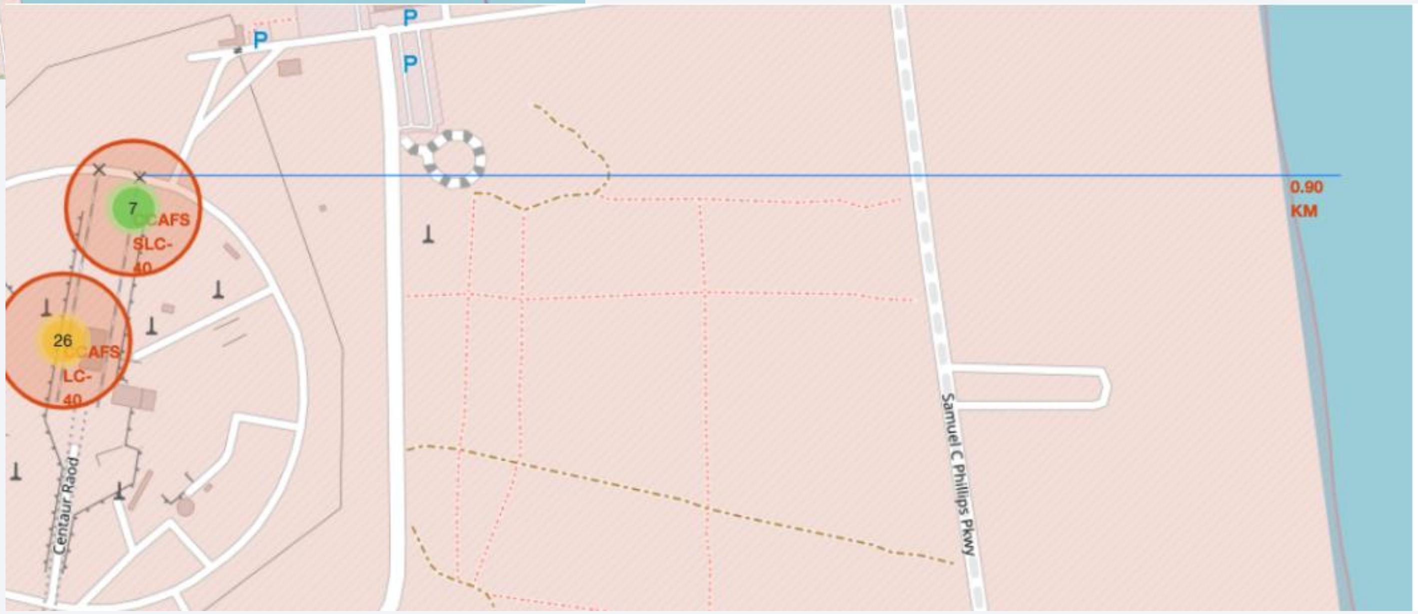


Color-labeled launch outcomes



Facilites close by launch sites

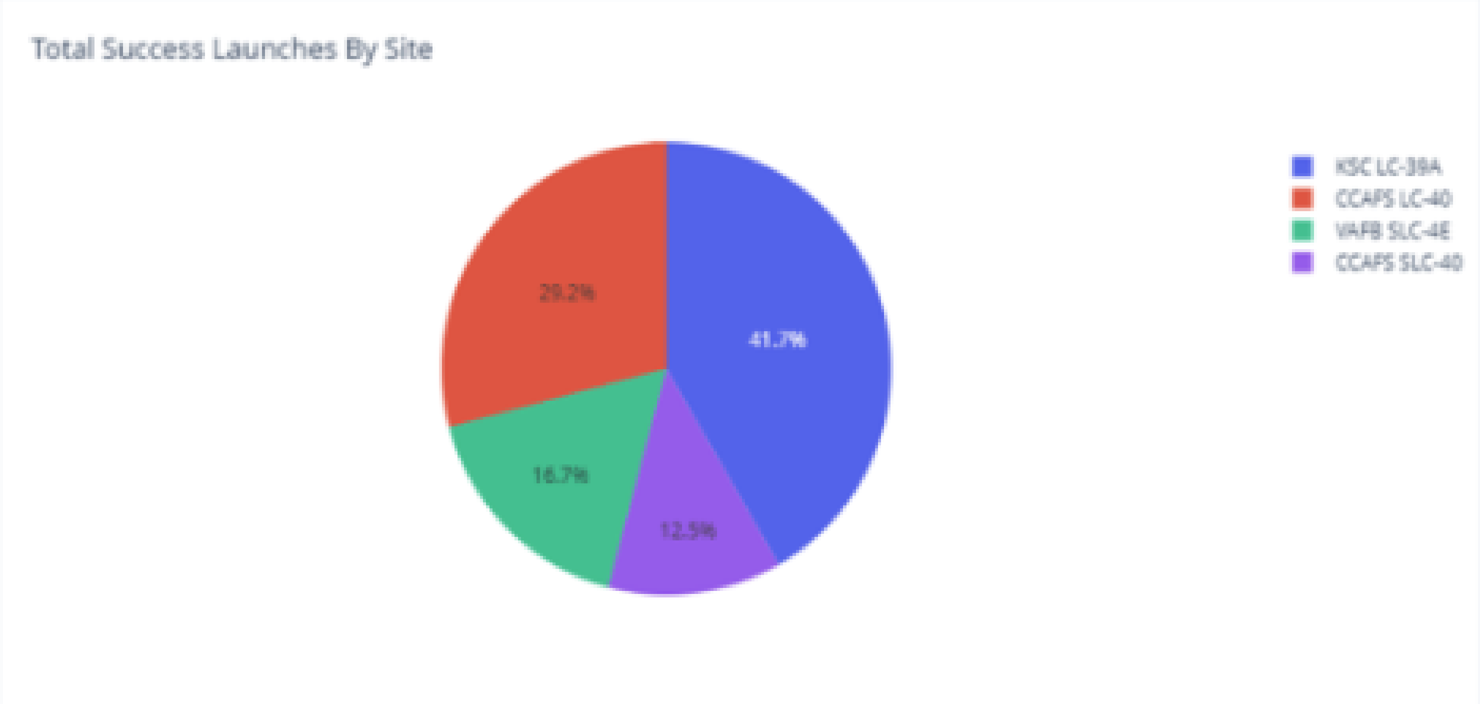




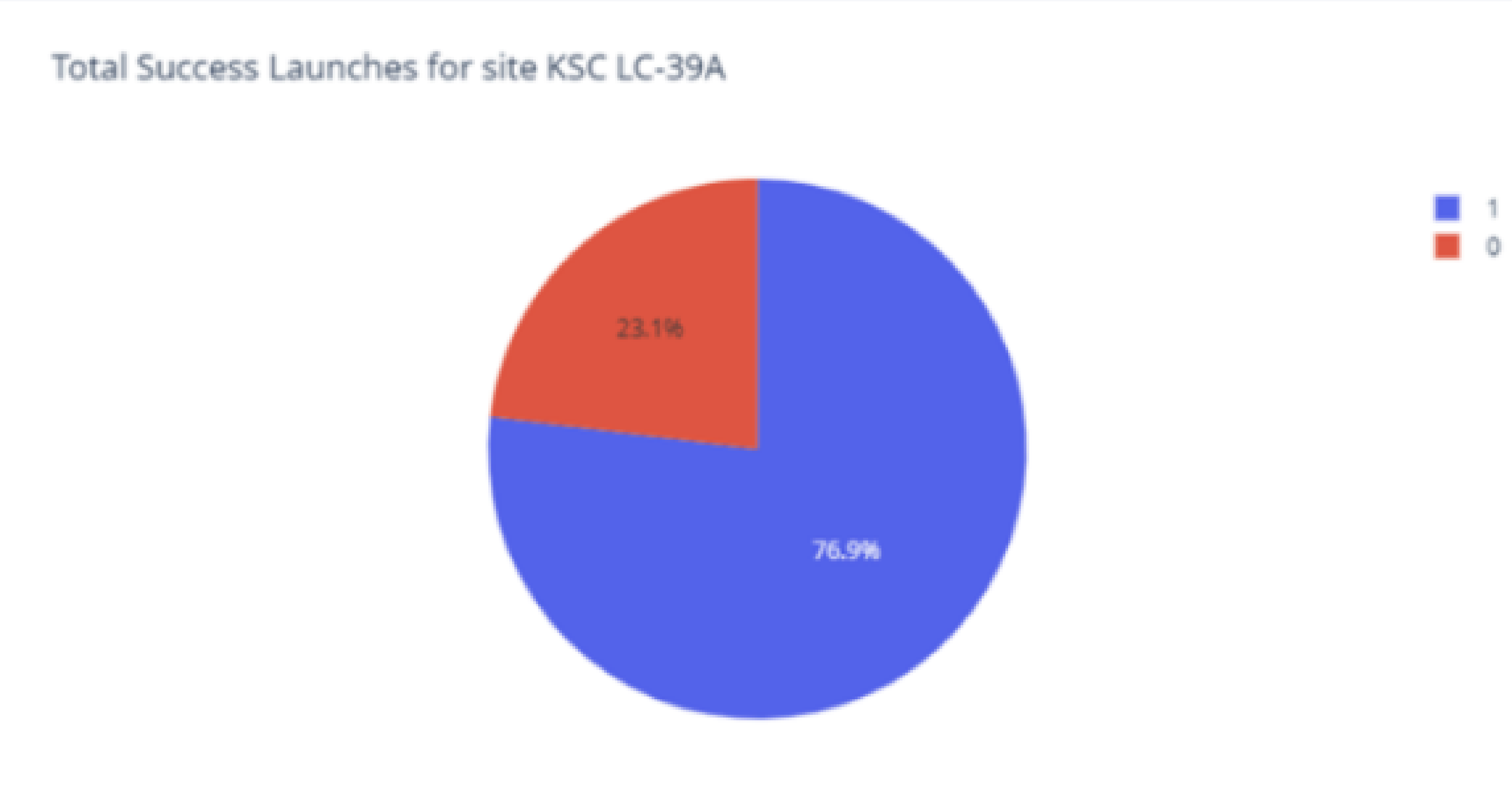
Section 4

Build a Dashboard with Plotly Dash

Total Launch Success for All Sites



Highest launch Success Ratio

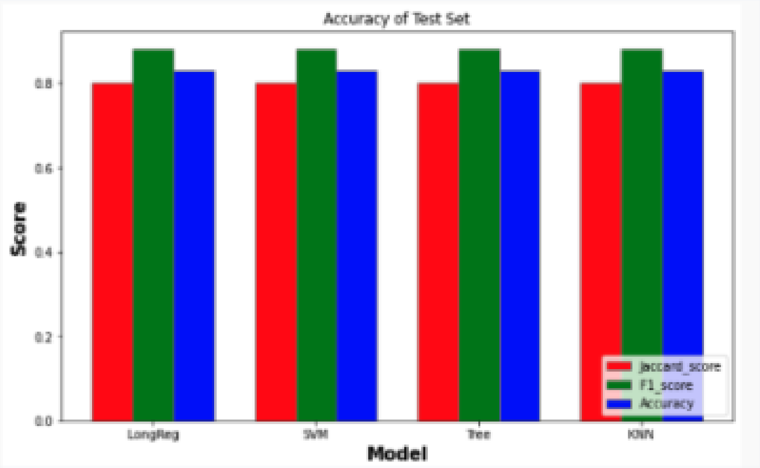


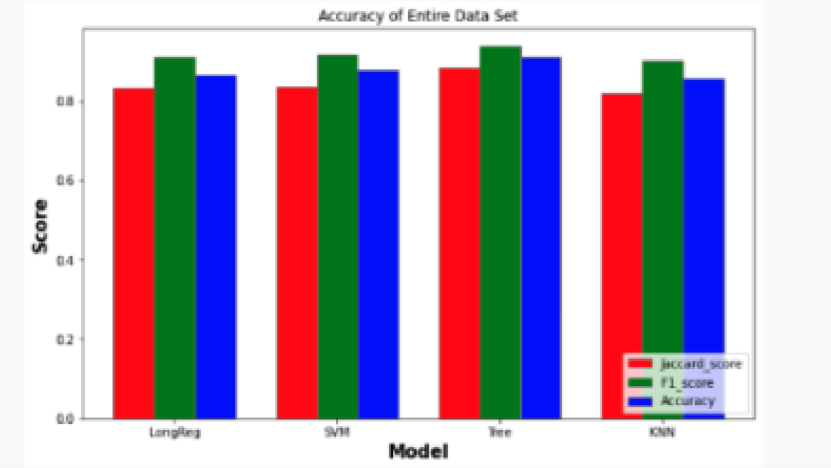
Payload vs Launch Outcome for All Sites

Section 5

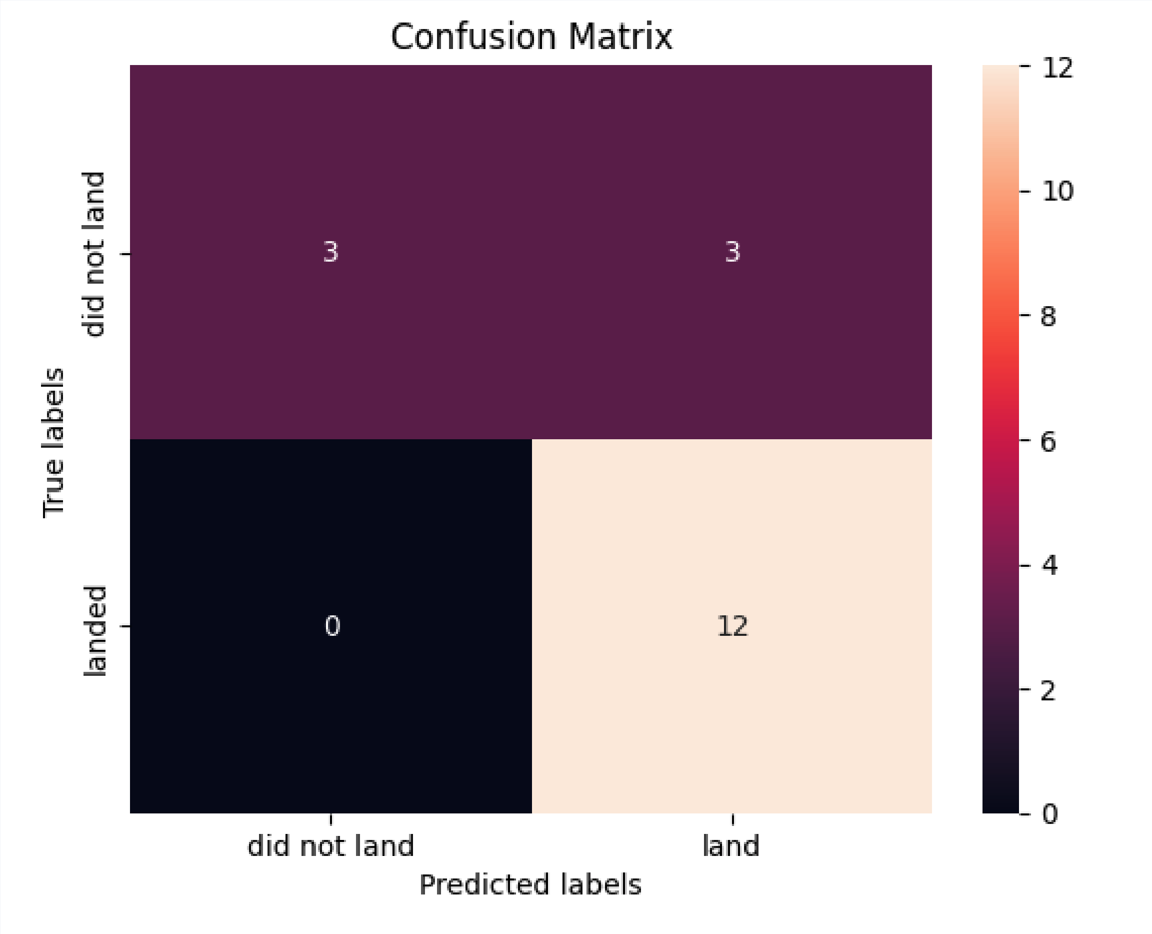
Predictive Analysis (Classification)

Classification Accuracy





Confusion Matrix



Conclusions

- Success rate for the rocket launches increased after 2013.
- Launch site KSC LC-39A has the highest success rate

Appendix

- GitHub URL for project https://github.com/joaogpinto04/IBM-Data-Science

Thank you!