IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sebastián A. Zamora Risco
19/05/2025

Outline

1. Executive Summary
2. Introduction
3. Methodology
4. Results
5. Conclusion
6. Appendix

Executive Summary

Summary of methodologies

- Data collection
- Data wrangling
- EDA with Data visualization
- EDA with SQL
- Interactive map with Folium
- Predictive classification analysis

Summary of all results

- EDA results
- Interactive analytics demo in screenshots
- Predictive analysis results

Introduction

- SpaceX advertises **Falcon 9 rocket launches** on its website with a cost of **62 million dollars**; other **providers cost upward of 165 million** dollars each, much of the savings is because **SpaceX can reuse the first stage**. Therefore, **if we can determine if the first stage will land**, **we can determine the cost of a launch**. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Using **publicly available SpaceX data**, **machine learning models** have been developed to **predict whether SpaceX will reuse the first stage.** Thus, the following questions are answered:
  - ❑ What is the relationship between different features and the landing success rate?
  - ❑ Which feature value could improve the landing success rate?
  - ❑ What is the best algorithm to create a model that predict the outcome of the launch?
    - ➢ **Does this model provide a satisfactory prediction of the mission outcome?**

Section 1

Methodology

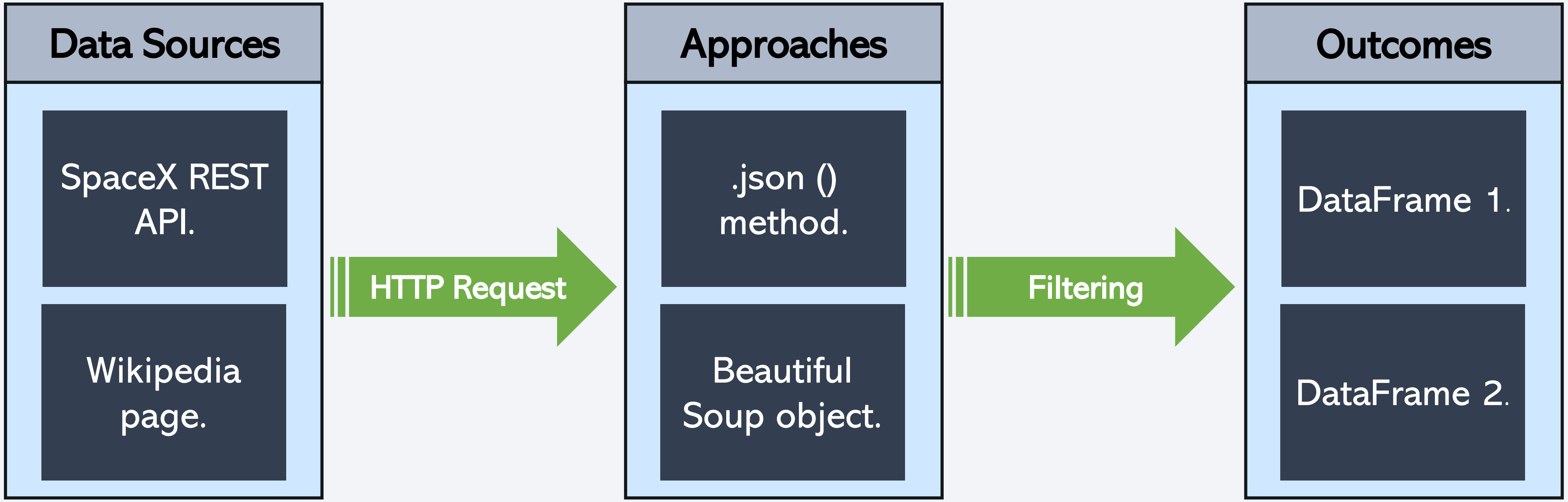
Methodology

Executive Summary

- Data collection methodology:
  - Request data from SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Convert landing outcomes into binary classification labels based on success or failure
- 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 built using data retrieved via **HTTP requests**: from the **SpaceX REST API** and through **web scraping** of a relevant **Wikipedia** page.
  SpaceX data was decoded using the **.json()** method, while Wikipedia data was parsed using a **BeautifulSoup** object.
  The diagram below outline the general data collection process for both approaches.



Data Collection – SpaceX API

GitHub: Data Collection with API

API Request

- ✓ Retrieve rocket launch data from **SpaceX REST API.**

Decode & Preprocess

- ✓ Convert the response using **.json()** and normalize it with **.json_normalize().**

Select and Enrich

- ✓ Select **key features.**
- ✓ Apply **custom functions** to enrich data by making new API requests.

Intermediate DataFrame

- ✓ Create a dictionary and convert it to a DataFrame using **Pandas.**

Final DataFrame

- ✓ Filter for **Falcon 9** launches only.
- ✓ Impute **missing PayloadMass values** using the mean.

Export

- ✓ Export the DataFrame to a **CSV file.**

Data Collection – Scraping

GitHub: Data Collection with Web Scraping

Request HTML Content

- ✓ **Request** the launch data from **Wikipedia static URL.**

Parse & Extract Tables

- ✓ Create a **BeautifulSoup object** and extract all **tables** from the **HTML response.**

Extract Columns Names

- ✓ **Iterate** over **<th>** tags and apply a cleaning function to **extract structured column names.**

Initialize Dictionary

- ✓ Create an **empty dictionary** with extracted **column names as keys.**

Create DataFrame

- ✓ **Loop through table rows** and extract launch records and **convert** the **resulting dictionary** into a **pd DataFrame.**

Export

- ✓ Export the DataFrame to a **CSV file.**

Data Wrangling

GitHub: Data Wrangling

➢ Perform **EDA** to identify patterns in the dataset and **define target labels** for supervised learning models. Landing outcomes are converted into **binary classification labels**: 1 indicates a successful booster landing, and 0 indicates failure.

Load and Inspect Data

- ✓ Read the **CSV** Dataset using **Pandas.**
- ✓ **Display** a few **initial rows** and check column **data types.**
- ✓ Calculate **missing value** percentages.
- ✓ **Count** launches by **launch site** and **orbit.**

Process Landing Outcomes

- ✓ **Count** Outcome values and identify **failed landing types.**
- ✓ **Create a binary classification**: 1 for successful landings, 0 otherwise.
- ✓ Add this as a **new column** in the **dataframe.**

Summarize and Export Data

- ✓ Calculate the **success rate** using the mean of the **binary column.**
- ✓ **Export** the final DataFrame **to a CSV** file.

EDA with Data Visualization

GitHub: EDA with Data Visualization

➢ This analysis explores how different features may influence the landing outcome of the Falcon 9 first stage. By examining patterns and relationships in the data, we can better understand the factors associated with successful landings. These insights are key for optimizing performance and reinforcing the economic advantage of first-stage reusability. The following table summarizes the visualizations created during this stage of the analysis.

Table 1. Summary of Visualizations used in EDA.

Plot Type (Seaborn)	X - Axis	Y - Axis	Grouping variable (Hue)
Catplot	Flight Number	Payload Mass	Launch Outcome (0 = Failure, 1 = Succes)
Catplot	Flight Number	Launch Site	Launch Outcome
Catplot	Payload Mass	Launch Site	Launch Outcome
Barplot	Orbit	Frequency of Launch Outcomes	N/A
Scatterplot	Flight Number	Orbit	Launch Outcome
Scatterplot	Payload Mass	Orbit	Launch Outcome
Lineplot	Year	Succes Rate (Launch Outcome)	N/A

EDA with SQL

GitHub: EDA with SQL

SQL Queries Performed

- Listed the unique names of launch sites used in space missions.
- Retrieved 5 records where the launch site starts with the prefix "CCA".
- Calculated the total payload mass carried by boosters launched by the customer NASA (CRS).
- Computed the average payload mass for boosters of the type "F9 v1.1".
- Listed the date when the first successful landing outcome on a ground pad was achieved.
- Listed booster names that had successful drone ship landings and carried payloads between 4000 and 6000.
- Counted the total number of successful and failed mission outcomes.
- Identified booster versions that carried the maximum payload mass using a subquery.
- Displayed records of failed drone ship landings in 2015, including booster version, launch site, and the launch month.
- Ranked the number of landing outcomes types between 2010-06-04 and 2017-03-20 in descending order.

Build an Interactive Map with Folium

GitHub: Interactive Visual Analytics with Folium

Launch Site Markers and Outcome Classification

- Circular markers were added for all launch sites, each with a popup label displaying the site name.
- Launch outcome markers were visualized using MarkerCluster: Green and Red markers indicate successful landings and failed landings, respectively.

✓ The purpose of these markers is to identify whether launches occurred near the equator and to visually explore outcomes from the same coordinates in an interactive way.

Distance Markers to Points of Interest

- Markers were placed to compute distances from each launch site to nearby points of interest, such as the railway, coastline, highway, and city.
- A MousePosition was added to dynamically display the cursor's latitude and longitude on the map.
- Text markers were used to display the computed distances between launch sites and surrounding locations.
- Lines were drawn to connect each launch site to its corresponding point of interest.

✓ The purpose of these visualizations is to assess whether launch sites are located near potentially hazardous areas.

Build a Dashboard with Plotly Dash

GitHub: Interactive Visual Analytics with Dash App

The app features two main interactive charts: a pie chart and a scatter plot, both updated in real-time from a CSV file.

➢ Dropdown menu: Allows selection of a launch site or all sites combined.

- Pie Chart: Displays success vs. failure for the selected site, or total successful launches per site when "ALL" is selected. Sites can be toggled on/off.

➢ RangeSlider: Filters data by payload mass (kg).

- Scatter Plot: Shows success/failure vs. payload for the selected site(s), colored by Booster Version Category.

➢ Tooltips display detailed information when hovering over data points.

✓ These interactive features were added to enable real-time analysis based on user needs, allowing dynamic visualization tailored to their variables of interest.

Predictive Analysis (Classification)

GitHub: Predictive Analysis with Machine Learning

Data Preparation

- ✓ Create a **NumPy array Y** from the **Class column**.
- ✓ **Standardize X** using **StandardScaler**.
- ✓ Split data into **training/test** sets (80%/20%).

Model and Parameter Definition

- ✓ Create the **model object** (**LogReg**, **SVM**, **Tree**, **KNN**).
- ✓ Define the **hyperparameters** to be tuned for each model

Use GridSearchCV

- ✓ Use the defined models and parameters with **10-fold CV** to find the **best hyperparameters**.

Evaluation on Test Data

- ✓ Compute **accuracy (score)** on the **test set**.
- ✓ Generate and analyze the **confusion matrix**.

Evaluation on Dataset

- ✓ Compute the **overall accuracy** of each model with its best hyperparameter.

Model Selection

- ✓ **Compare all models'** performances and select the one with the **highest accuracy**.

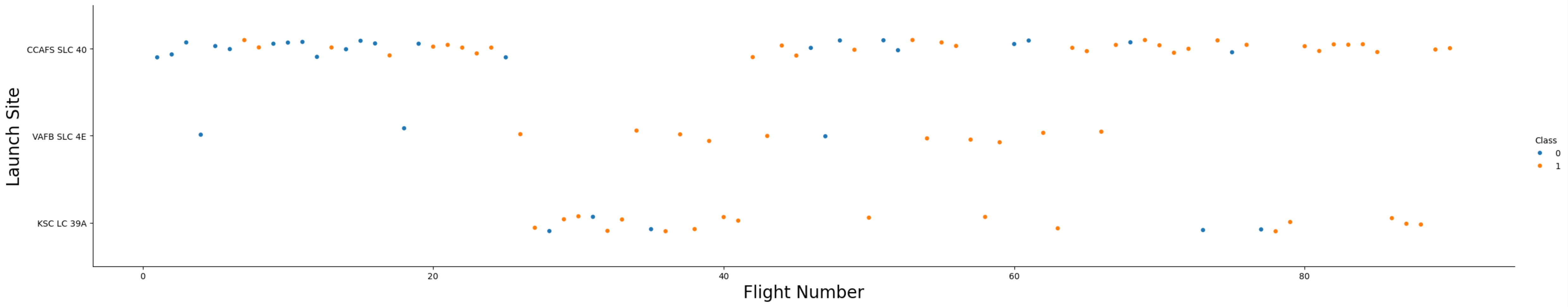
Results

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

Section 2

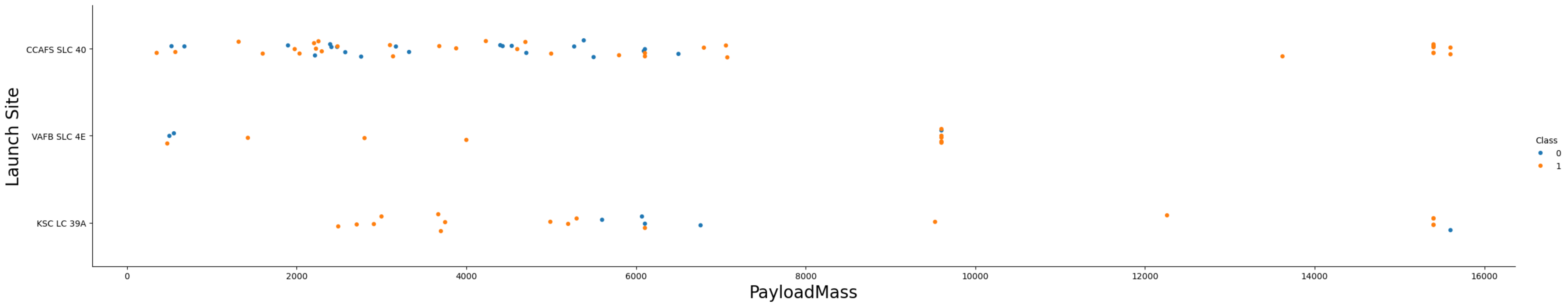
Insights drawn from EDA

Flight Number vs. Launch Site



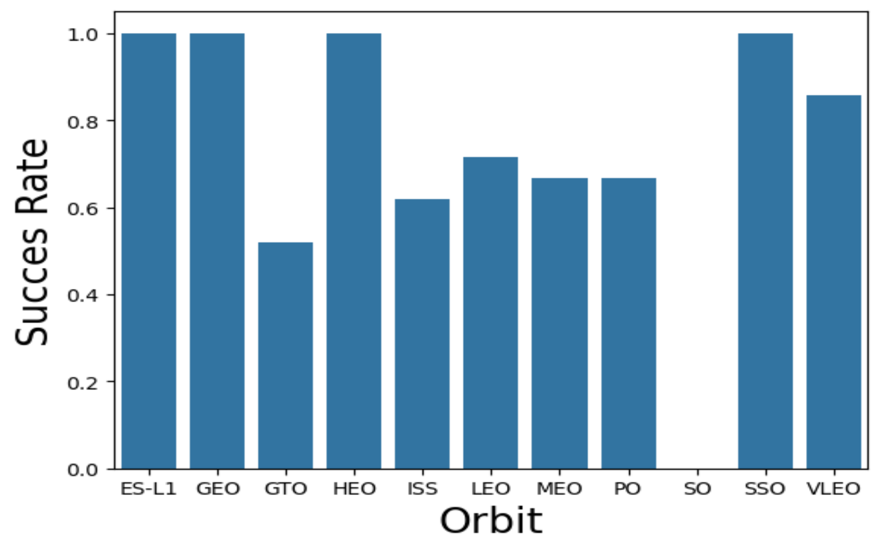
- All launches with a Flight Number above 80 resulted in successful landings.
- At the CCAFS SLC 40, the success rate begins to increase after Flight Number 50.
- VAFB SLC 4E shows the highest success rate for launches with a Flight Number greater than 20.

Payload vs. Launch Site



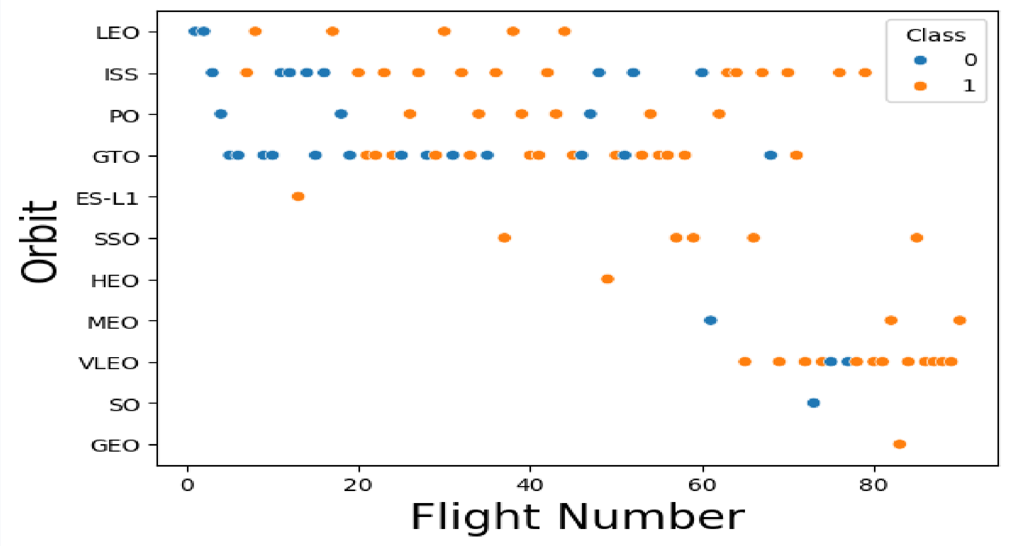
- No clear trend is observed for CCAFS SLC 40 with Payload Mass below 8k; however, launches above 14k show a higher success rate.
- At KSC LC 39A, several failures are observed for launches with Payload Mass around 6k.
- At VAFB SLC 4E, multiple launches with identical Payload Mass appear to be test launches. The success rate suggests that Payload Mass is not the only factor influencing the landing outcome.

Success Rate vs. Orbit Type



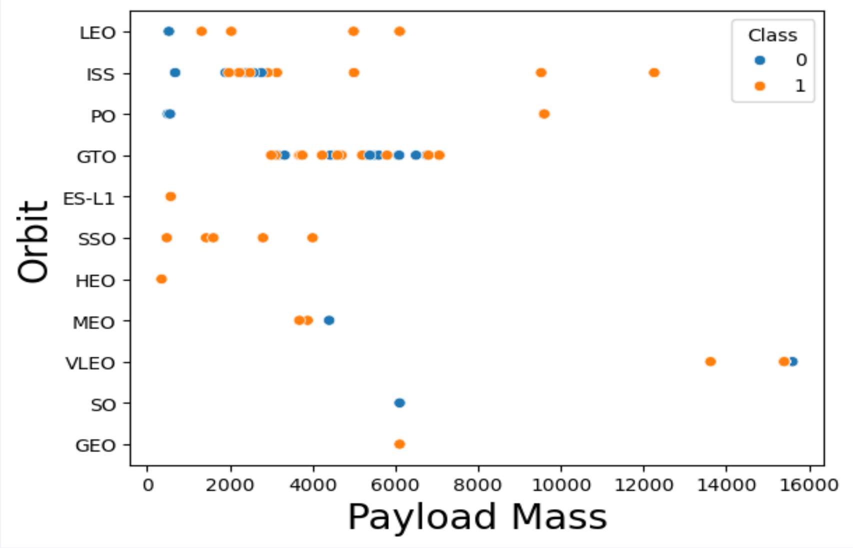
- Launches to orbits GTO, ES-L1, GEO, HEO, and SSO resulted exclusively in successful outcomes.
- No successful landings were recorded for launches to the SO orbit.
- This plot should be complemented with the frequency of launches per orbit to provide proper context.

Flight Number vs. Orbit Type



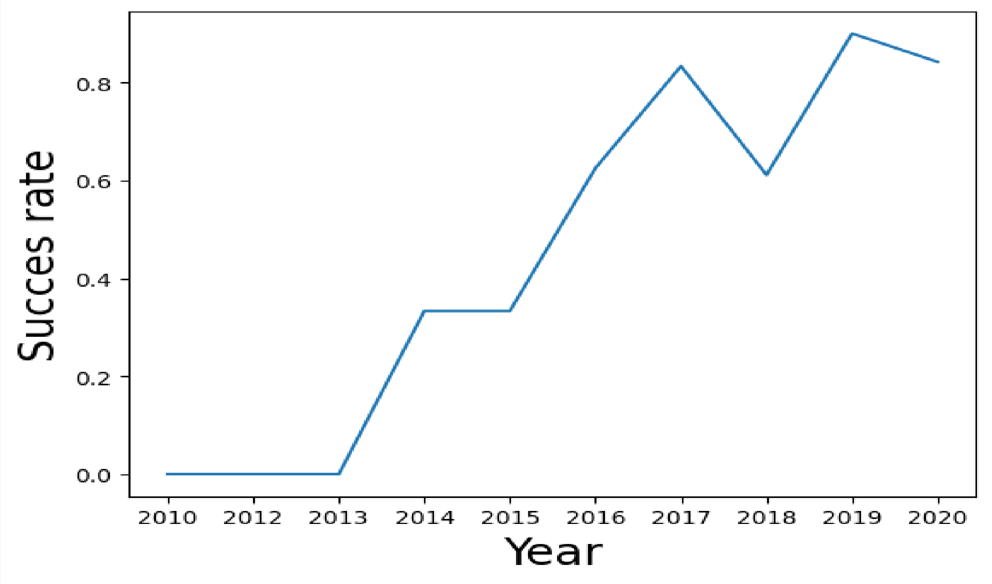
- The VLEO orbit shows a 100% success rate for flight numbers above 80, which stands out given its high number of launches while maintaining this trend.
- A similar behavior is observed for launches to the ISS orbit with flight numbers above 60.
- The GTO orbit does not exhibit a clear trend of success or failure based on flight number.

Payload vs. Orbit Type



- The GTO orbit does not exhibit a clear trend of success or failure based on Payload Mass.
- Launches to the SSO orbit do not appear to be influenced by payload mass, as the success rate remains constant.
- For the same payload mass of 6k, launches to the SO orbit fail, in contrast to successful launches to the GEO orbit.

Launch Success Yearly Trend



- Overall, the success rate has shown an upward trend up to the year 2020.
- Relative declines in success rate are observed in 2018 and 2020. It could be of interest to further investigate the potential causes behind these fluctuations.

All Launch Site Names

```
%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

Listed the unique names of launch sites used in space missions.

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

Retrieved 5 records where the launch site starts with the prefix "CCA".

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

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

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

totalpayloadmass
45596

Calculated the total payload mass carried by boosters launched by the customer NASA (CRS).

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS averagepayloadmass FROM SPACEXTABLE WHERE "Booster_Version" like '%F9 v1.1%';
```

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

averagepayloadmass
2534.6666666666665

Computed the average payload mass for boosters of the type "F9 v1.1".

First Successful Ground Landing Date

```
%sql SELECT MIN("Date") AS firstsuccesful FROM SPACEXTABLE WHERE "Landing_Outcome" like '%Success (ground pad)%';
```

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

firstsuccesful
2015-12-22

Listed the date when the first successful landing outcome on a ground pad was achieved.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT DISTINCT "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

Listed booster names that had successful drone ship landings and carried payloads between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT CASE WHEN TRIM("Mission_Outcome") = 'Success' THEN 'Success' ELSE "Mission_Outcome" END AS Outcome, COUNT(*) AS Total FROM SPACEXTABLE GROUP BY Outcome;
```

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

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

Counted the total number of successful and failed mission outcomes.

Boosters Carried Maximum Payload

```
%sql SELECT "Booster_Version", PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

Booster versions that carried the maximum payload mass.

2015 Launch Records

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

```
 * 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

Displayed records of failed drone ship landings in 2015, including booster version, launch site, and the launch month.

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

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

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

Landing_Outcome	Freq
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

Ranked the number of landing outcomes types between 2010-06-04 and 2017-03-20 in descending order.

Section 3

Launch Sites Proximities Analysis

Launch Site's Location

- ➢ The majority of launches are concentrated in the state of Florida.
- ➢ Overall, launch sites are located relatively close to the equator.
- ➢ All launch sites are situated near coastal areas.

Color - labeled launch outcomes

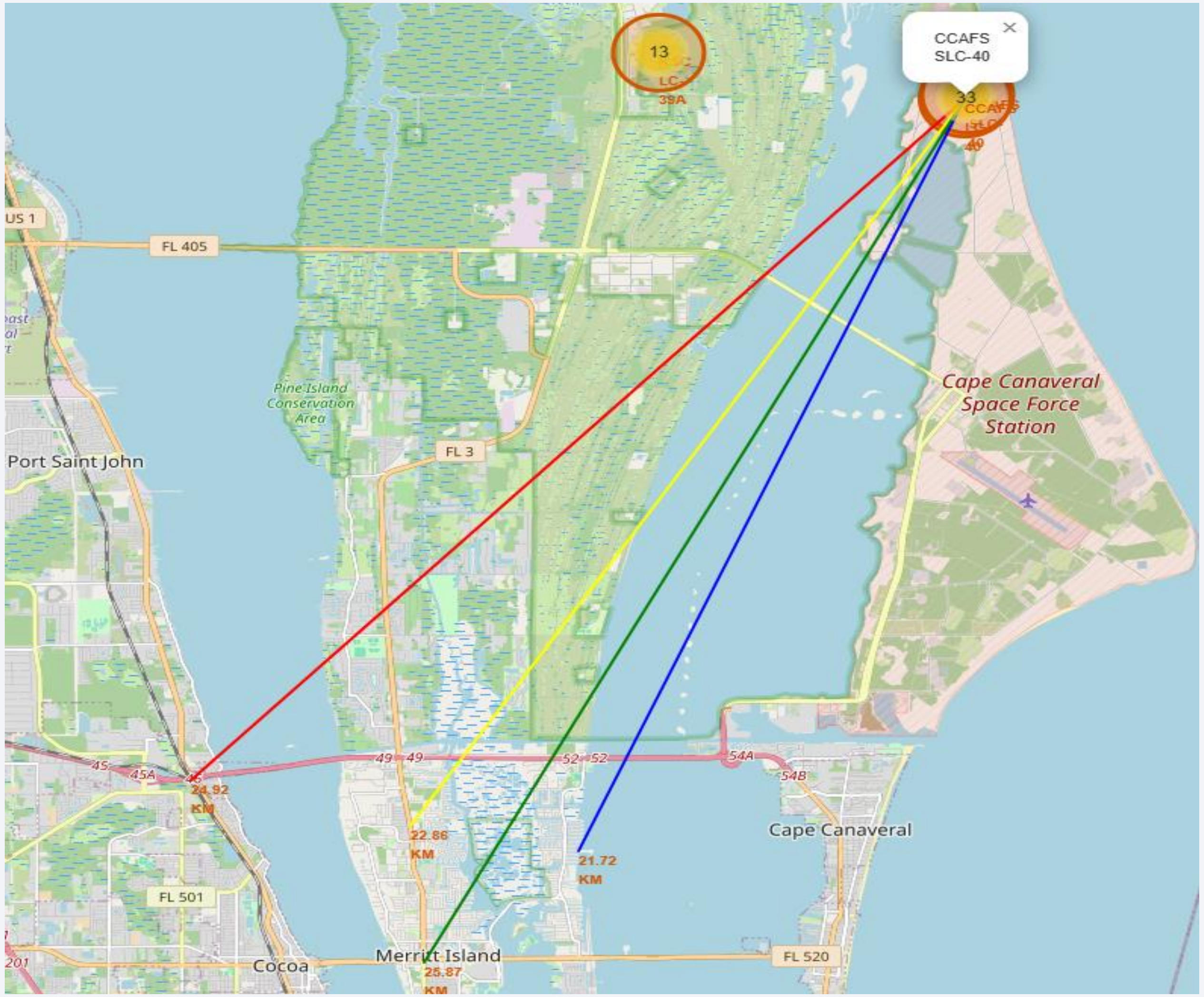
- ➢ This allows for a quick visual assessment of the success rate at each launch site.
- ➢ It can be observed that KSC LC-39A has the highest success rate among them.

Proximity to Launch Site

- ➢ This type of visualization allows for the assessment of distances to key areas of interest, which can help evaluate potential risk zones.
- ➢ The following table summarizes the information presented in this figure.

Table 2. Summary of proximity to interest points.

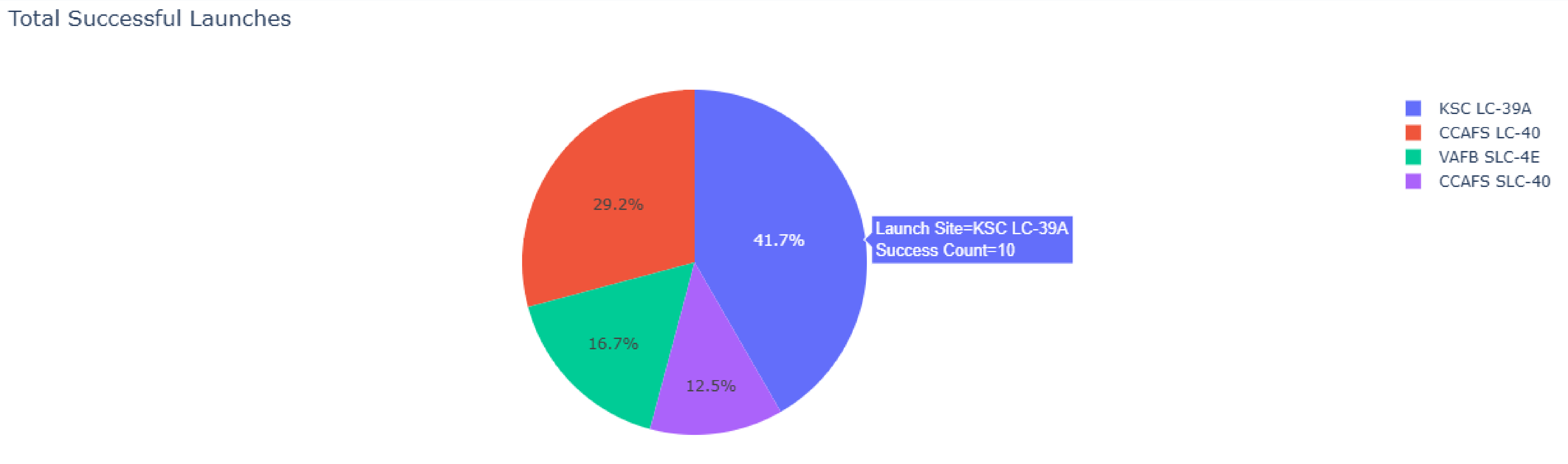
Distance To	Line Color	Distance [km]
Coastline (Banana River)	Blue	21.72
Florida East Coast Railway	Red	24.92
North Courtenay Highway	Yellow	22.86
City (Merritt Island)	Green	25.87



Section 4

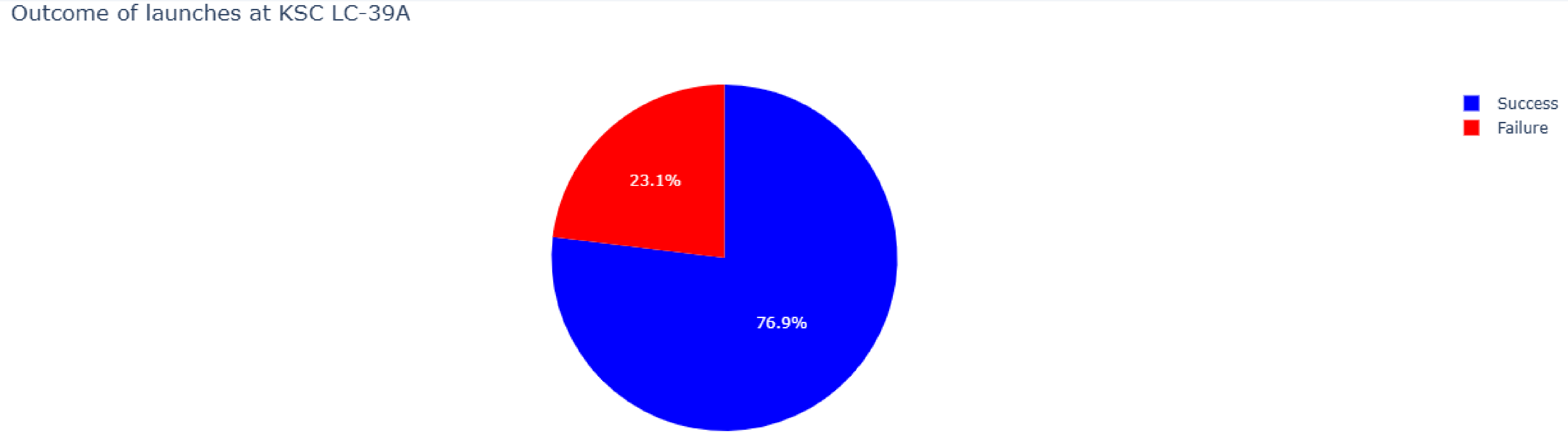
Build a Dashboard with Plotly Dash

Total Successful Launches for All Sites



- KSC LC-39A shows the highest success rate, with a total of 10 successful launches.

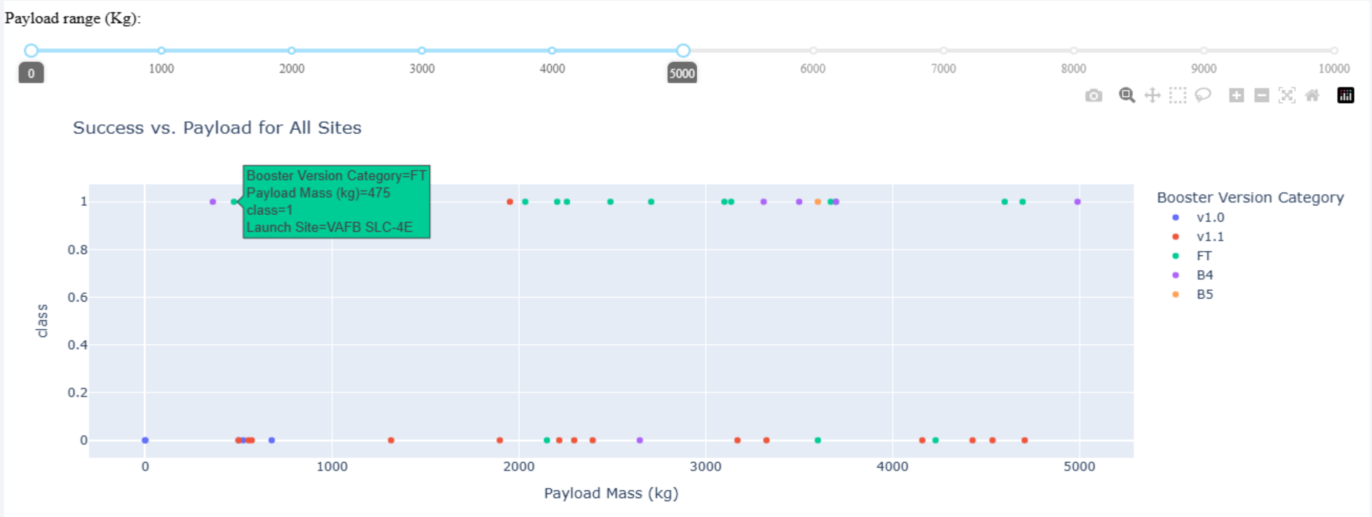
KSC LC – 39A Succes Rate

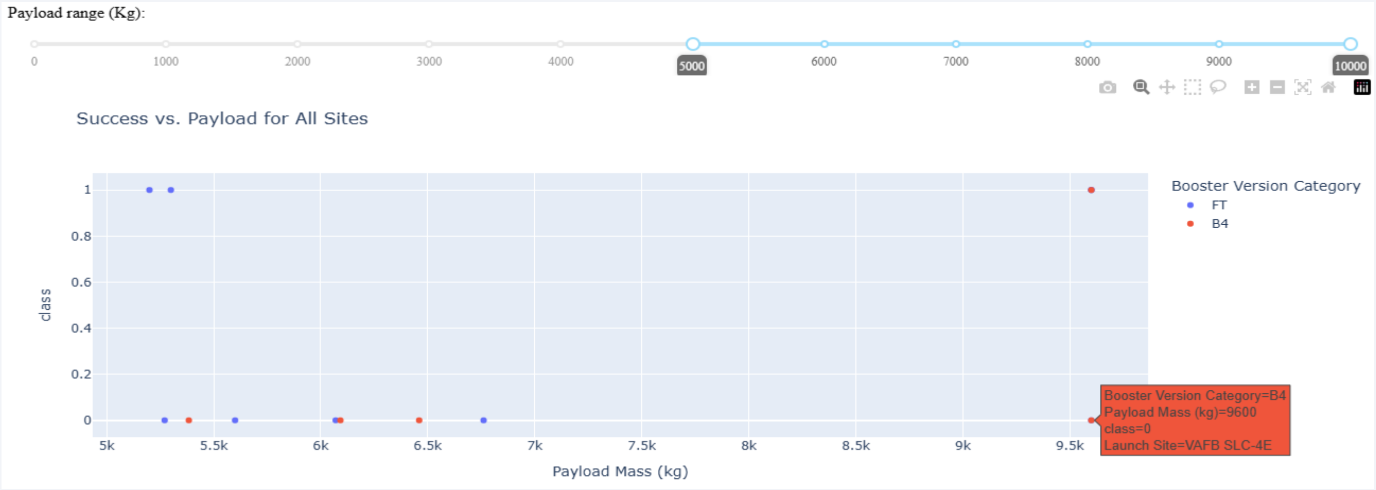


- KSC LC-39A demonstrates a solid success rate, with 10 successful launches and 3 recorded failures.

Success Rate vs. Payload, grouped by Booster Category

- For payloads above 5k [kg], a lower success rate is observed.
- For payloads below 5k [kg], the success rate increases.
- Booster version B4 shows a very low success rate, in contrast to FT, which demonstrates a high success rate.



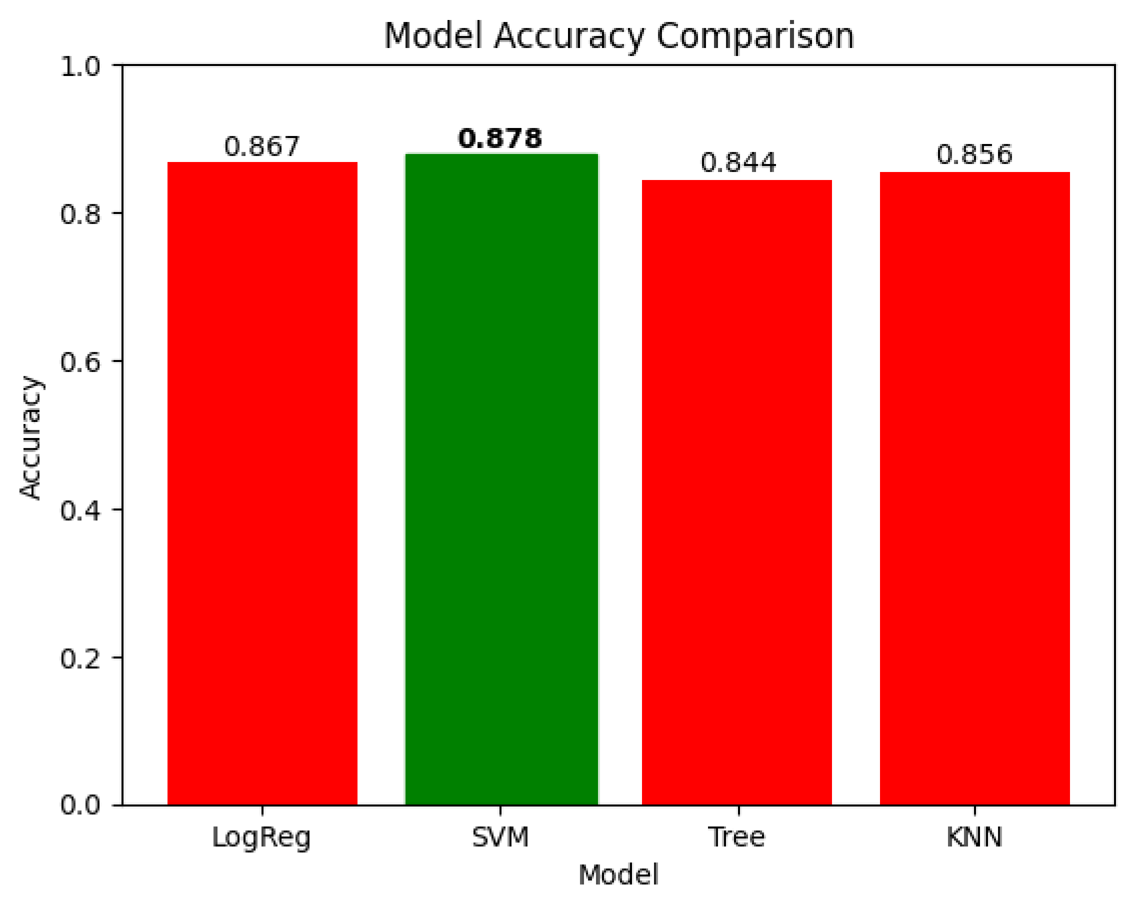


Section 5

Predictive Analysis (Classification)

Classification Accuracy

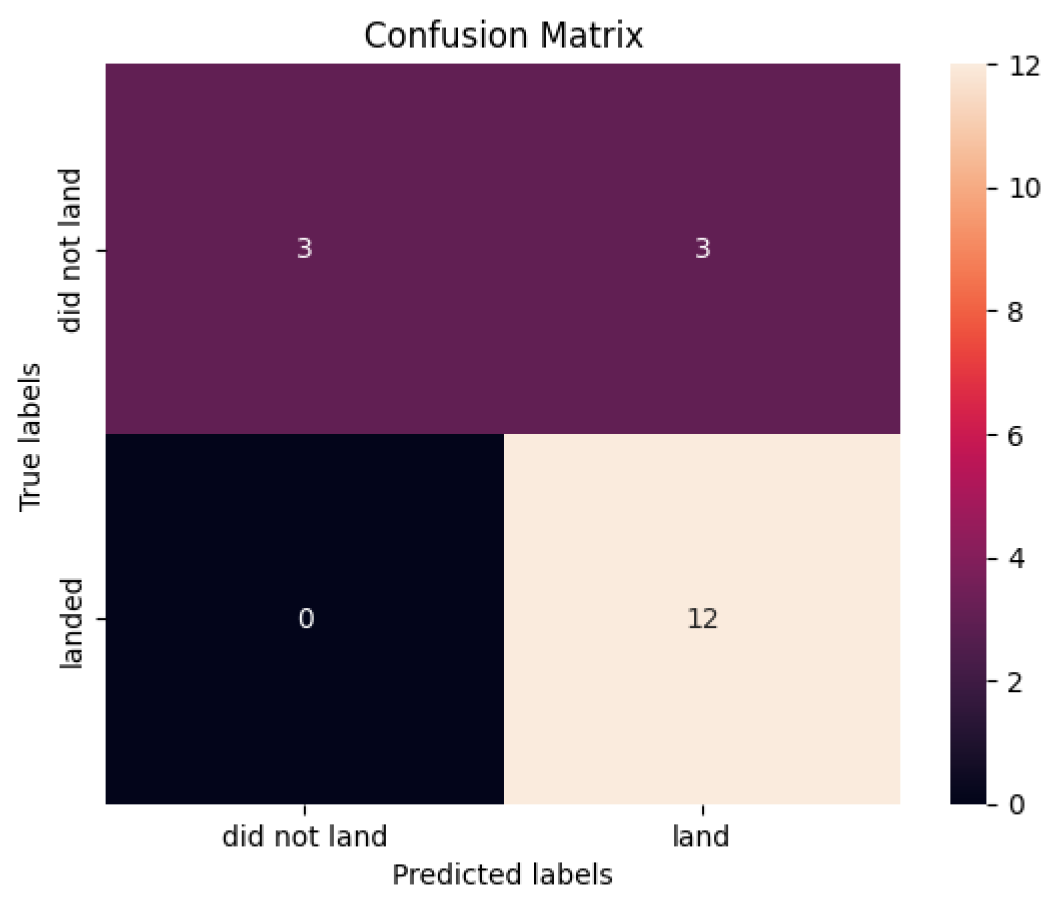
- ➢ The **test data accuracy** was **identical across all evaluated models**.
- ➢ The accuracy for each model was evaluated using the **entire dataset**. Accordingly, the **best-performing** model corresponds to the **Support Vector Machine (SVM)**, as highlighted in the graph.



Confusion Matrix

- Analyzing the **confusion matrix** of the **SVM** model on the **test set**, there are **12 True Positives, 3 True Negatives, 3 False Positives, and 0 False Negatives**, as shown in the figure.

- This indicates that the model **does not confuse failures with successes**; however, it does **misclassify some successes as failures** to a certain extent.



Conclusions

- ✓ The **Support Vector Machine (SVM)** model yielded **the best performance on this dataset.**
- ✓ **Future improvements** could include incorporating **engineered features**, such as the product of two variables (e.g., $A \times B$) or squared terms (e.g., A^2), **to capture potential non-linear relationships**.
- ✓ **All launches to orbits** such as **GTO**, **ES-L1**, **GEO**, **HEO**, and **SSO** resulted in **100% success rates**, indicating high reliability for these mission profiles.
- ✓ A notable **decrease in success rate** was observed **in 2018 and 2020**, which may warrant **further investigation to understand the underlying factors.**
- ✓ **Launch site KSC LC-39A** recorded the **highest success rate**, suggesting favorable operational conditions or infrastructure at this location.
- ✓ The **B4 booster version exhibited a very low success rate**, whereas the **FT version showed consistently high reliability.**

Appendix

- I am currently **enhancing the Dash application** to be packaged as an **executable (.exe),** allowing any user to run it more easily. **The goal** is to make it work with any similar CSV file, **enabling users to explore the data** interactively **and in a more convenient way—without showing the command prompt window and automatically opening the browser** (this has already been implemented).

```python
# Run the app de forma directa
import webbrowser
import threading
if __name__ == '__main__':
    port = 8050
    url = f"http://127.0.0.1:{port}"
    # Abrir el navegador en un hilo separado para que no bloquee el servidor
    threading.Timer( interval: 1, lambda: webbrowser.open(url)).start()
    app.run(port=port)
```

```python
#### lo siguiente es para que el archivo se lea de distintas rutas ...creo
import sys
import os

1 usage
def resource_path(relative_path):
   # algo como:  """ Get absolute path to resource, works for dev and for PyInstaller exe """
    try:
        base_path = sys._MEIPASS  # ruta temporal donde PyInstaller extrae el contenido
    except AttributeError:
        base_path = os.path.abspath(".")  # ruta en modo desarrollo

    return os.path.join(base_path, relative_path)

# Usas así para leer el CSV:
spacex_df = pd.read_csv(resource_path("spacex_launch_dash.csv"))


# Read the airline data into pandas dataframe
#Esto se elimina porque esta arriba de una forma mas chunga spacex_df = pd.read_csv("spacex_launch_dash.csv")
max_payload = spacex_df['Payload Mass (kg)'].max()
min_payload = spacex_df['Payload Mass (kg)'].min()

#cosas auxialires
dropdown_launch_site_options = [{'label': 'All Sites', 'value': 'ALL'}] + [
    {'label': site, 'value': site} for site in spacex_df['Launch Site'].unique()]

# Create a dash application
app = dash.Dash(__name__)

# Create an app layout      #Task 1 add dropdown para site launches (Launch Site)
app.layout = html.Div(children=[html.H1( children: 'SpaceX Launch Records Dashboard',
                                style={'textAlign': 'center', 'color': '#503D36','font-size': 40}),
                                dcc.Dropdown(id='site-dropdown',
```

Appendix

Special thanks to IBM and Coursera for promoting accessible education and continuous learning for everyone. ❤

Thank you!