IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Afif Beji
February 07, 2025

Outline

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

Executive Summary

- **Summary of methodologies:**
  - Data Collection from SpaceX REST APIs and Web Scraping.
  - Data Wrangling (preprocessing: dealing with outliers, missing values, standardization, etc.).
  - Exploratory Data Analysis (EDA) with SQL and Data Visualization.
  - Interactive Visual Analytics and Dashboard with Plotly Dash and Interactive Maps with Folium.
  - Predictive Analysis (Classification) using Basic ML Models.
- **Summary of all results:**
  - Exploratory Data Analysis insights.
  - Interactive Visuals, Dashboard and Maps.
  - Predictive analysis findings _ Best Supervised Classification ML Model performance.

Introduction

- **Project background and context:**
  - SpaceX revolutionized space travel with reusable first stage rockets, especially the Falcon 9.
  - The Falcon 9's first stage is designed to return to Earth for a controlled landing.
  - Historical data on Falcon 9 landings is available through the SpaceX API and via web scraping (Wikipedia).
  - This project examines the data to identify factors influencing successful landings and predict whether SpaceX will reuse the first stage and so determine the approximate price of a launch.
- **Problems we want to find answers:**
  - What factors and features could affect the success or failure of Falcon 9 landings?
  - How have landing success rates evolved over the years?
  - Which launch sites or environmental conditions are most favorable for landings?
  - Can we identify patterns in data that explain landing successes or failures?
  - Can we predict whether a Falcon 9 first stage will successfully land and so estimate the cost of a launch?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Space X REST API & Web Scraping (Wikipedia)
- Perform data wrangling
    - Preprocessing, Cleaning, Handling outliers and missing values, Standardizing data, etc.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Build, tune and evaluate classification models

Data Collection

Start Data Science Project by understanding the overall thematic and problem and identifying the needs for a clear workflow

Problem Understanding

Fetch data using a SpaceX REST API regarding Falcon 9 historical missions. Get a JSON file

Data Collection – SpaceX REST API

Convert the JSON file into a Pandas DataFrame (DF) using json_normalize() function to ensure more fluid data manipulation and analysis.

Data Preparation Convert JSON to a DF

Data Preparation Filter

Filter the DataFrame to only include Falcon 9 launches.

Data Collection – Web Scraping

Gathering relevant data from Wiki pages. Receive an HTML file

Data preparation

Extract relevant information and parse data using BeautifulSoup library

Data preparation Create a DF

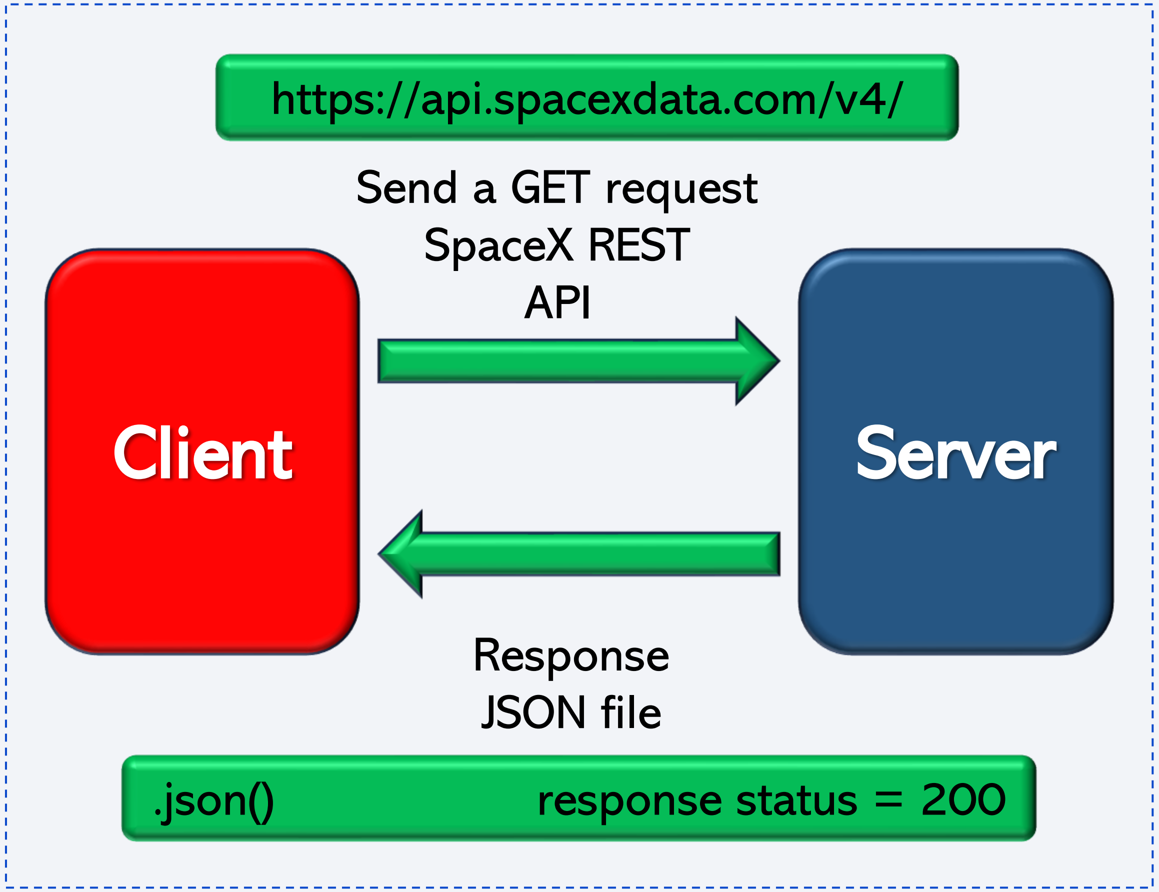
Transform data into a DataFrame (DF)

CSV

The current flowchart outlines principal steps that were followed in the present study for the data collection and wrangling processes. This will ensure an efficient and robust data analysis

Data Collection – SpaceX API

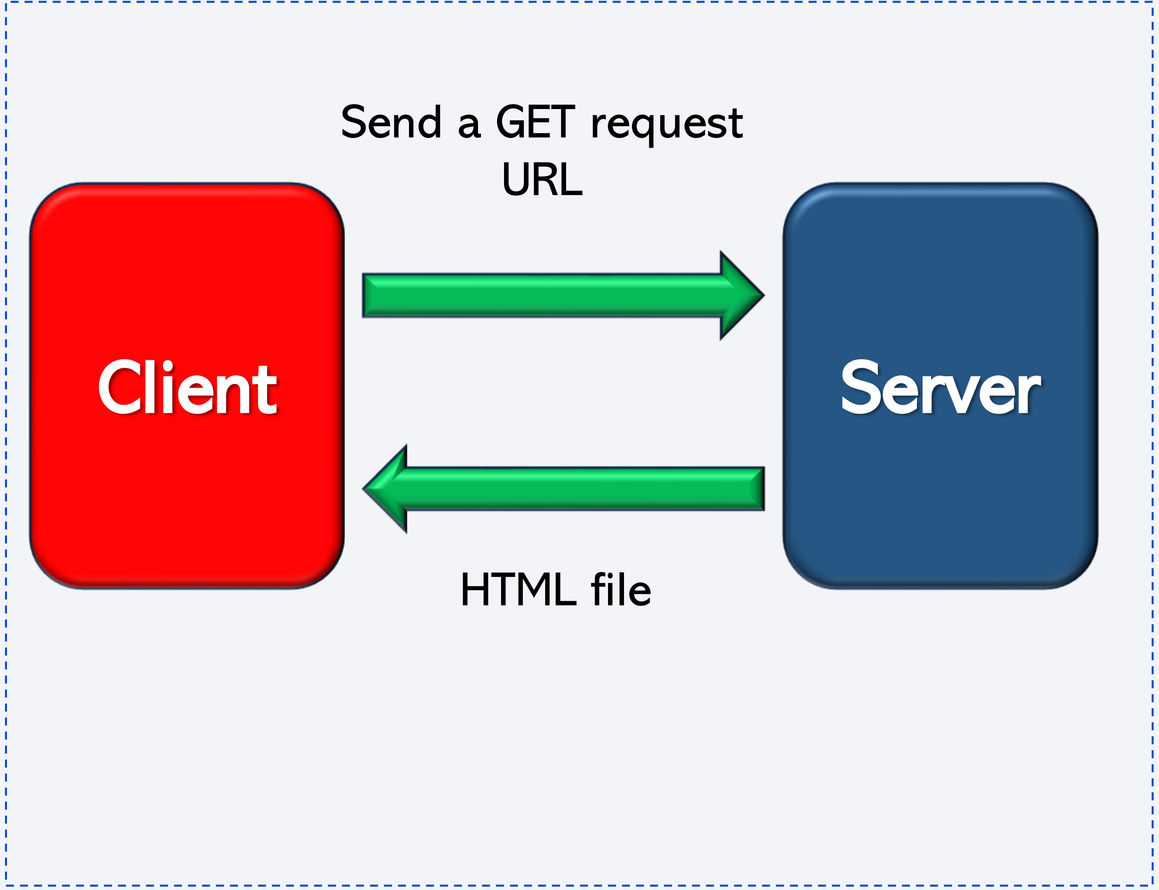
- To gather data from the SpaceX REST API:
- Send a GET request.
- Receive response (JSON file).
- Convert the JSON into a Pandas DataFrame (DF).
- Send multiple endpoints SpaceX REST API to gain more/further information.
- Create a dictionary to store relevant data then create a new DataFrame.
- Filter the DataFrame to only include Falcon 9 launches.
- Export DF to a CSV format file.



Data Collection SpaceX API GitHub

Data Collection - Scraping

- Use Web scraping to retrieve data from Wiki pages:
- Send a GET request.
- Receive response (HTML file).
- Use the Python BeautifulSoup package to scrape HTML tables that contain valuable Falcon 9 launch records.
- Parse data from tables.
- Create a dictionary to keep relevant data.
- Create a Pandas DataFrame.
- Export to a CSV format file.



Data Collection Web Scraping GitHub

Data Wrangling

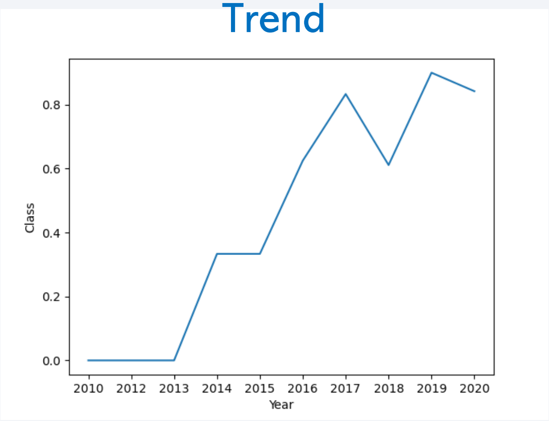
- Data wrangling is also known as data cleaning, data preprocessing or data preparation.
- It involves refining raw data into more relevant useful information.
- It encompasses tasks such as handling outliers and missing values, standardizing and normalizing data to prevent machine misinterpretation or inaccurate analysis during subsequent operations.
- In this study, the main operations can be summarized as:
  - Calculate the number of launches for each launch site.
  - Calculate the number and occurrence for each orbit type.
  - Calculate the number and occurrence of mission outcome of the orbits.
  - Create a landing outcome label from Outcome column.
  - Export the data into a CSV file.

👉 Data Wrangling GitHub

EDA with Data Visualization

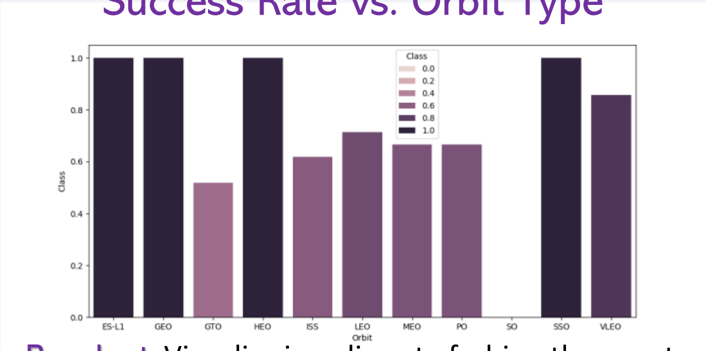
- Exploratory Data Analysis consists of several steps, where we transform data residing in a tabular format into more readable, understandable form. Typically, we create visuals such as (histograms, scatter plots, bar charts, line graphs, and much more). This will help data scientist draw accurate insights and construct clearer ideas about the problem.
- For example, we can represent:

Launch Success Yearly Trend



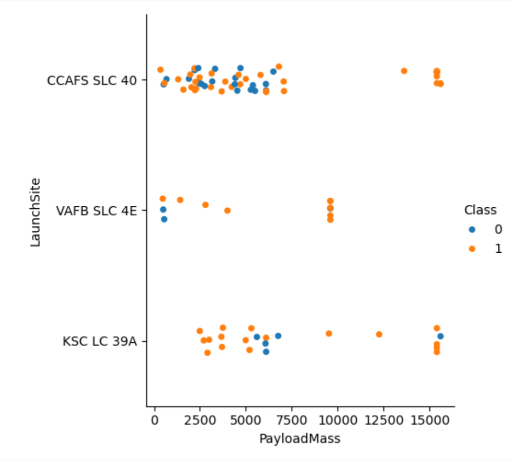
Line graph: Reveals trends and fluctuations over time

Success Rate vs. Orbit Type



Bar chart: Visualize in a discrete fashion the counts or values of different categories to compare them against each other

Payload vs. Launch Site



Scatter plot: Depicts correlation between two variables

Similarly, we used scatter plots to visualize:

- Flight Number vs. Flight Number
- Success Rate vs. Orbit Type
- Flight Number vs. Orbit Type
- Payload vs. Orbit Type

Exploratory Data Analysis (EDA) with Visualization GitHub

EDA with SQL

- SQL queries are crucial for exploring datasets and gaining insights. We performed the following queries:
  - Display the names of unique launch sites in the space mission.
  - Display five records where launch sites begin with the string 'CCA'.
  - Display the total payload mass carried by boosters launched by NASA (CRS).
  - Display the average payload mass carried by booster version F9 v1.1.
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass in the interval [4000, 6000] kg.
  - List the total number of successful and failed mission outcomes.
  - List the names of the booster versions which have carried the maximum payload mass.
  - List the months of 2015, failure landing outcomes in drone ship, booster versions and launch sites.
  - Rank the count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order.

Exploratory Data Analysis (EDA) with SQL GitHub

Build an Interactive Map with Folium

- In order to identify some geographical patterns in launch sites, we created some interactive maps using Python's Folium package.
- The main elements of Folium commands can be summarized in the table below:

Map object	Description / Objective	Use cases	Command
Markers	Add a site's location to a map using its latitude and longitude coordinates, or provide additional information	- Indicate each launch site on the map. - Use clusters to distinguish successful launches (green markers) from failed ones (red markers).	folium.Map() Folium.Icon() folium.map.Marker()
Circles	highlighted circular area shapes with a specified radius and a text label at specific coordinates to emphasize regions of interest and make them easy to locate when zoomed out	- Draw a red circle to mark NASA Johnson Space Center. - Place a black circle to represent each launch site.	folium.Circle()
Lines	Serve to display distance connecting two points or locations on the map	- Create 4 lines representing the distances between selected launch site 'CCAFS SLC-40' and its proximities: coastline, highway, roadway, nearest city.	folium.PolyLine()

Interactive Visual Maps with Folium GitHub

Build a Dashboard with Plotly Dash

- To build interactive web visuals and explore feature statistics, we implemented a web-based interactive Dashboard using Python's Plotly and Dash packages.
- The main objects' descriptions could be summarized in this table:

Object	Objective / Use cases	Command
Dropdown menu	Enable user to either select all sites or a specific launch site: - Select all sites. - Select a single site (e.g., 'CCAFS SLC-40').	dcc.Dropdown()
Pie Chart	Reflects statistics about: - The total count of successful launches for all sites. - Successful launches vs failed ones pertaining to the selected launch site.	px.pie()
Slider	- Adjust a payload mass range by selecting the minimum and maximum values.	dcc.RangeSlider()
Scatter plot	- Explore launch site insights, find relevant data for further analysis and extract information. - Display data points of payload mass vs. success rate for the selected item in the dropdown list.	px.scatter()

[Interactive Visual Analytics and Dashboards with Plotly Dash GitHub]

Predictive Analysis (Classification)

- The different steps to build, evaluate, improve, and found the best performing classification model are:
- Data preparation and preprocessing:
  - Read the dataset.
  - Select feature matrix and target column.
  - Standardize and scale data.
- Model selection
  - Split data into training and testing sets.
  - Select supervised ML models for categorical analysis.
  - Use GridSearchCV technique to fit, tune and find best hyperparameters for each model.
- Model evaluation
  - Calculate the accuracy score for each model on the testing set.
  - Display the confusion matrix.
- Model comparison
  - Compare the chosen ML models accuracies using a bar chart to fix the best one.
  - In this study, the best model was the decision tree model.

Predictive Analysis GitHub

Results

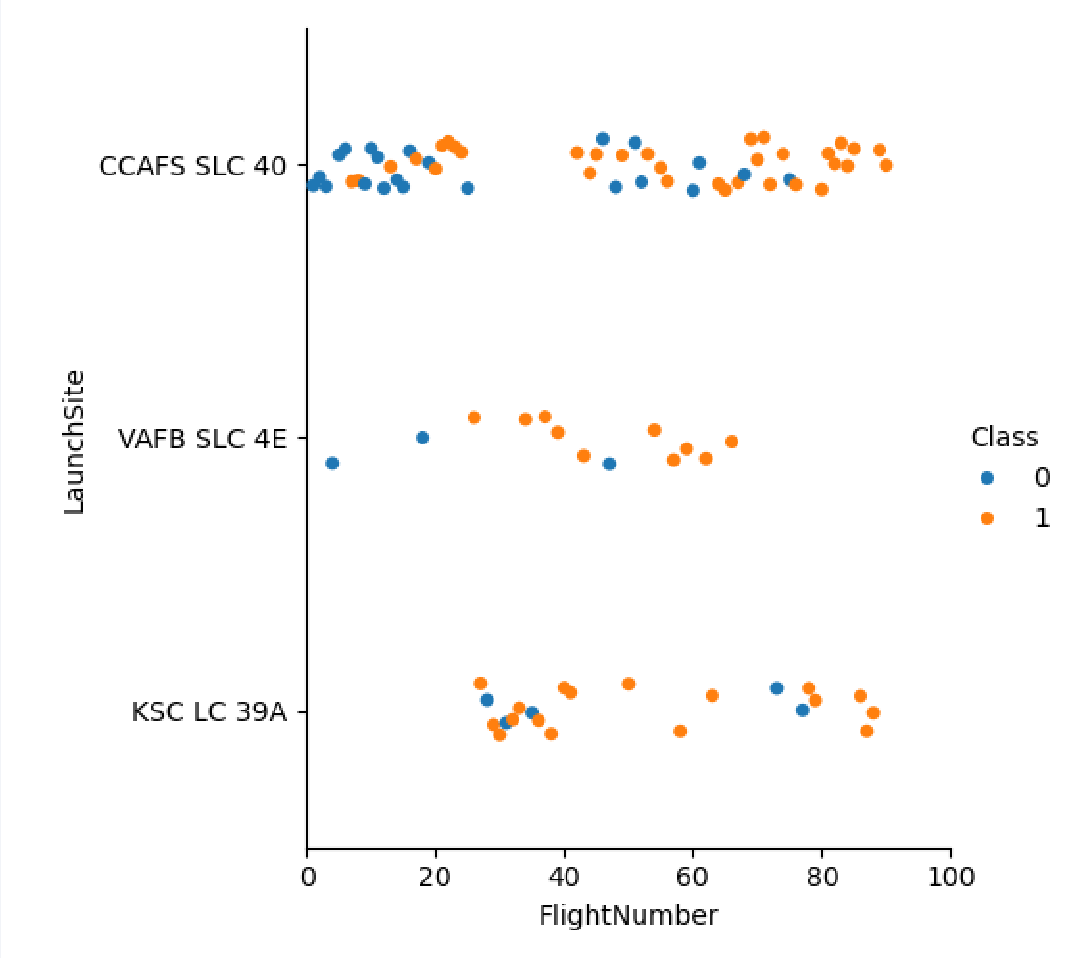
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

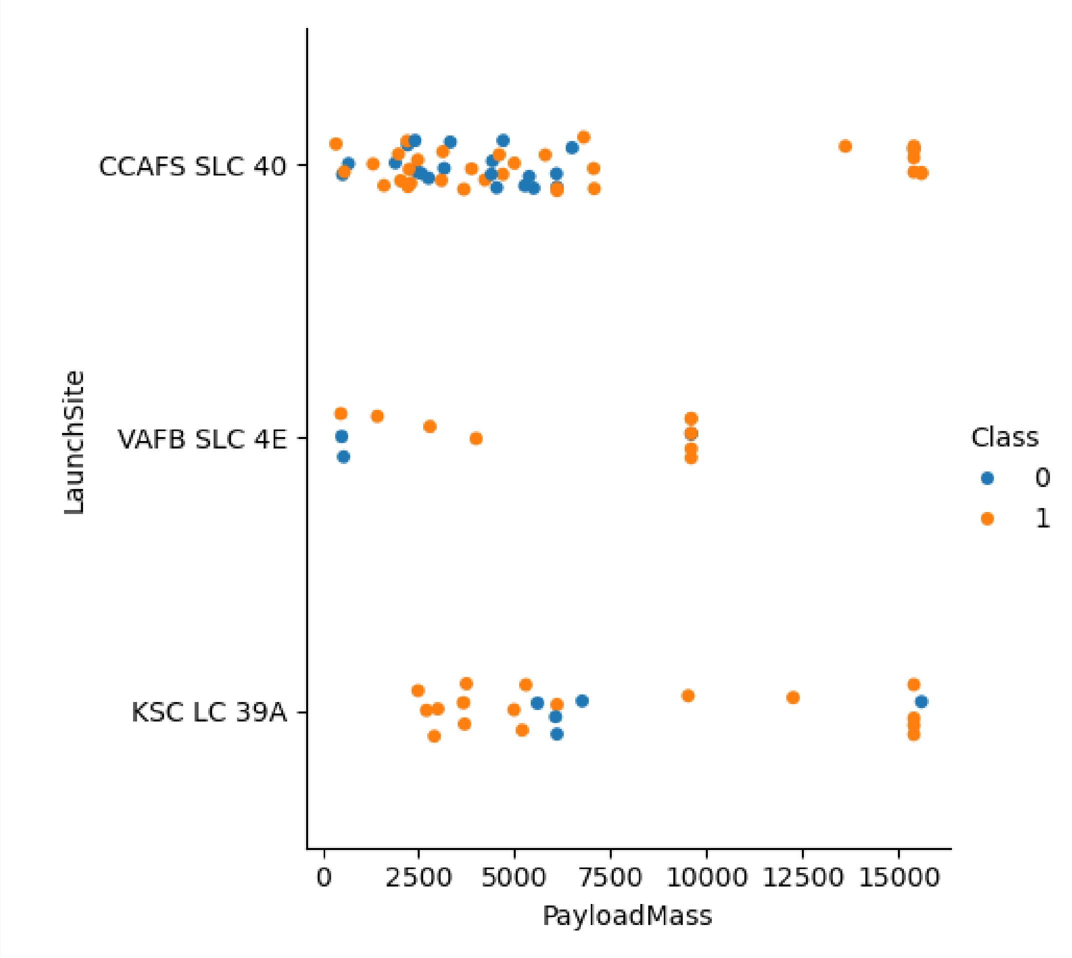
Flight Number vs. Launch Site

- **Early vs later launches:** success rate increases for later launches.
- **For 'CCAFS SLC 40' launch site:** has more launches showing a mix of failures and successes for early flight numbers with more failures than successes → Progressively success trend to increase especially for very high flight numbers.
- **For 'VAFB SLC 4E' launch site:** has fewer launches. Very high success rate except for early launches with more failures.
- **For 'KSC LC 39A' launch site:** In general, has high success rate.



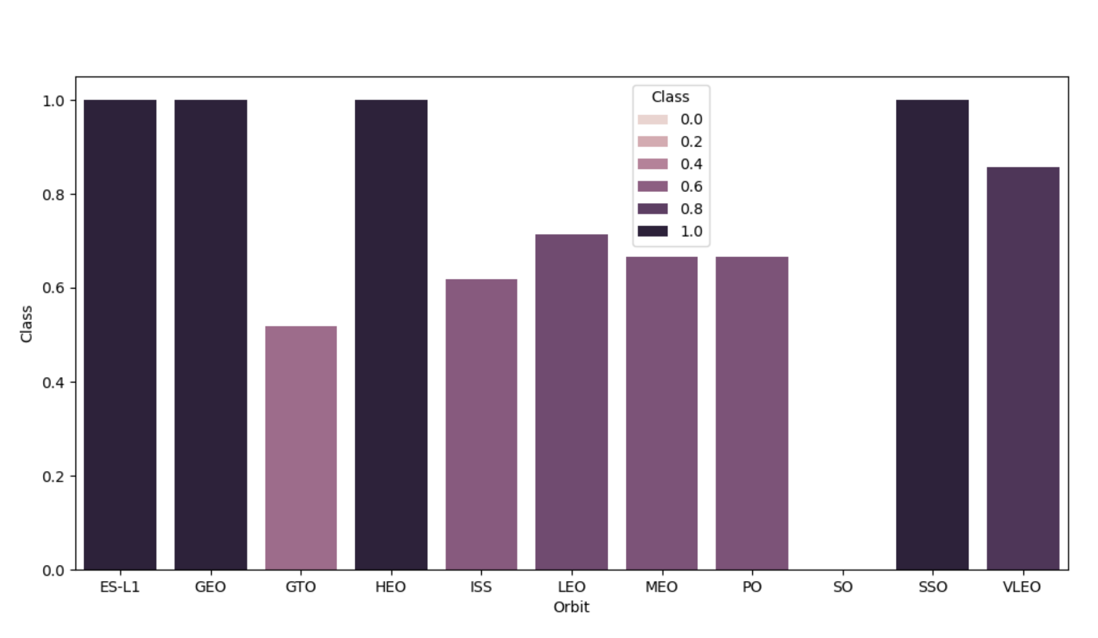
Payload vs. Launch Site

- **Early vs later launches:** success rate is very high for medium and especially heavy payload mass.
- **For 'CCAFS SLC 40' launch site:** has more launches. The figure shows a mix of success/failure rate and only successful launches for heavy payload shuttles.
- **For 'VAFB SLC 4E' launch site:** has fewer launches with the highest success rate. Except for the lightest payload mass shuttles, all the launches were successful.
- **For 'KSC LC 39A' launch site:** overall, it has successful launches except for the range 5000-7000 and for the heaviest payload mass.



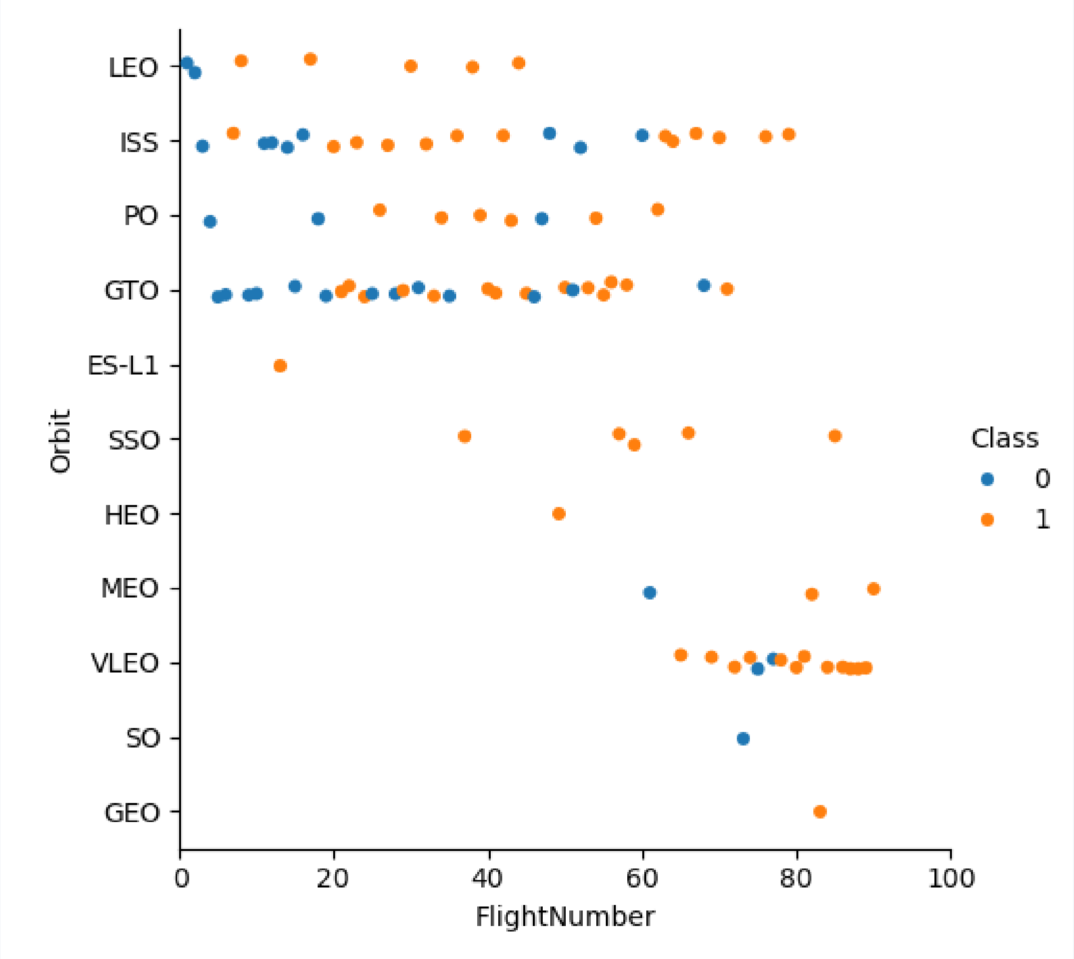
Success Rate vs. Orbit Type

- **‘ES-L1’, ‘GEO’, ’HEO’ and ‘SSO’**: Perfect success rate (100%).
- **‘VLEO’**: High success rate (~85%).
- **‘LEO’, ‘MEO’, ‘PO’ and ‘ISS’**: Moderate success rate (60–70%).
- **‘GTO’**: Average success rate (~50%).
- **‘SO’**: Complete failure. Whereas, we can’t draw a firm stance on this subject due to the limited data, as only one observation is available!



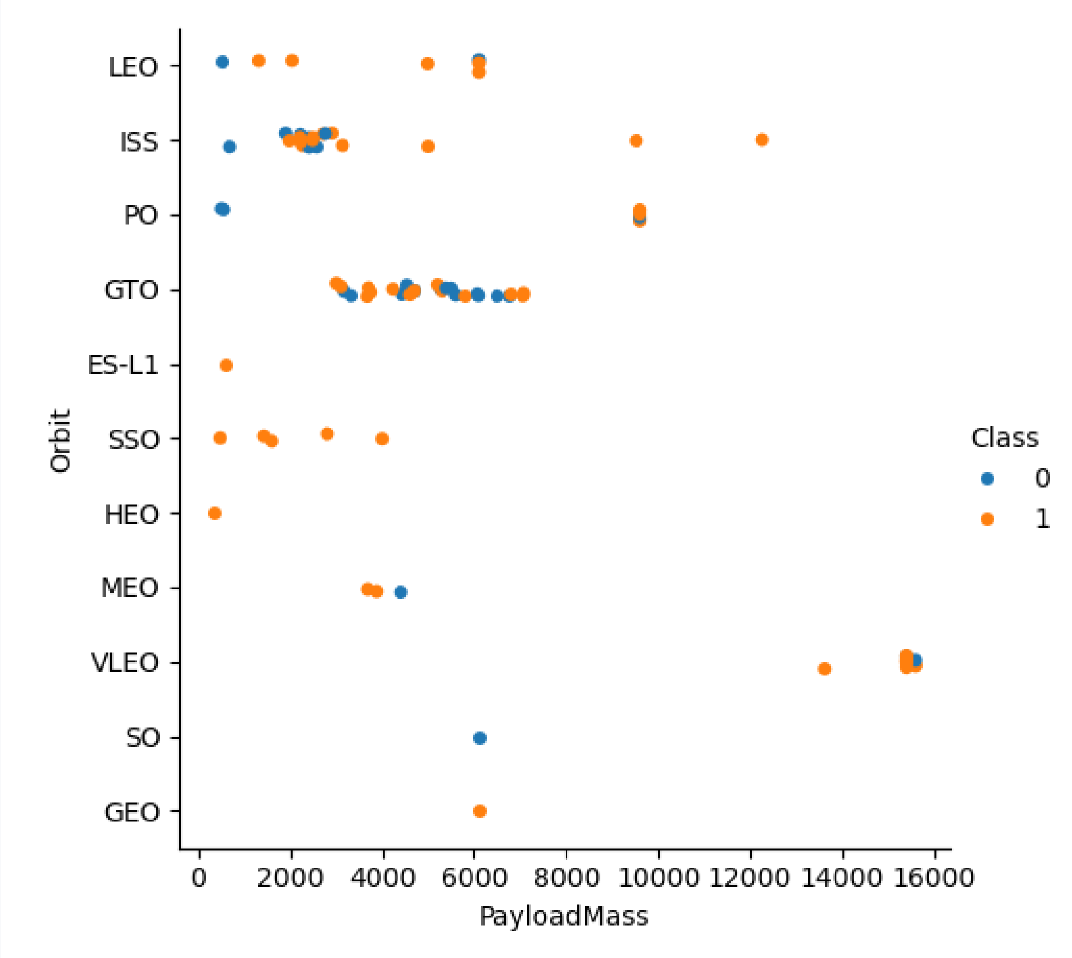
Flight Number vs. Orbit Type

- **‘LEO’**: Apart from lower flight numbers, it has a successful launches. It figures in the first half.
- **‘ISS’, ‘PO’ and ‘GTO’**: Present in a large number of launches, omnipresent for a significant range of flights.
- **‘ISS’ and ‘PO’**: Low success rate for the first launches, followed by a high rate with few failures.
- **‘GTO’**: Starts with multiple failures followed by a mix of success and failures.
- **‘ES-L1’, ‘SSO’, ‘HEO’ and ‘GEO’**: Despite a small number of launches, they have perfect success rate (100%).
- **‘ES-L1’, ‘HEO’ and ‘GEO’**: Each possess a single successful low, medium and high flight number, respectively.
- **‘MEO’ and ‘VLEO’**: They are prominent in later launches with a mixed success rate.
- **‘SO’**: A single failed attempt.



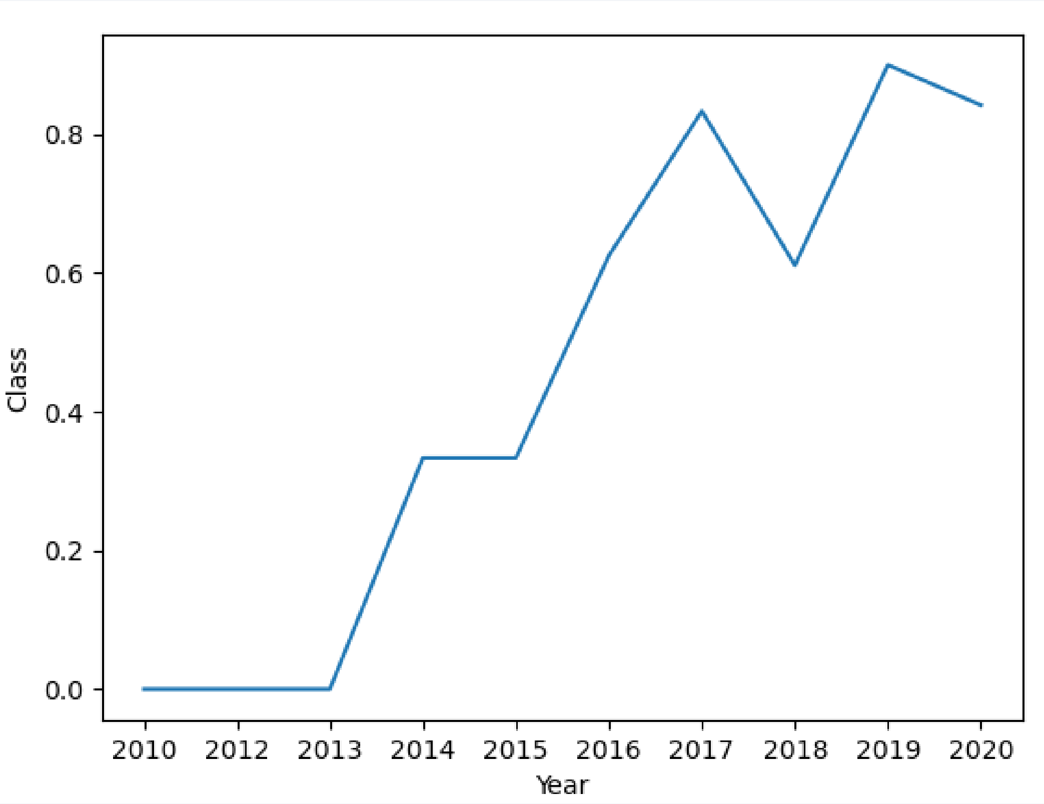
Payload vs. Orbit Type

- **‘LEO’, ‘ISS’ and ‘PO’:** Initially, they show low success rates for lighter payloads but demonstrate considerable improvements as payload mass increases.
- **‘GTO’:** Possess light to moderate payloads with an average success rate.
- **‘VLEO’:** Supports the heaviest missions with a mixture of successes and failures. While,
- **‘ES-L1’ and ‘HEO’:** Associated with the lightest payloads with full success.
- **‘SSO’ and ‘MEO’:** Carry light payload, respectively with perfect and moderate success rates.
- **‘SO’ and ‘GEO’:** Present for the medium payloads, with, one single failed and one successful launch, respectively.



Launch Success Yearly Trend

- 2013 could be considered as a special year, as it witnessed the first successful launch.
- By 2014, the success rate reaches roughly 40%.
- In 2015, the success rate remained unchangeable.
- For the next two years, the steepest successful launches with a high success ratio of 80% reached for the first time.
- In 2018, there was a 20% decrease in the success rate.
- In 2019, the highest performance ever recorded.
- In 2020, there was a slight decrease, bringing the success rate in line with 2017's achievements.



All Launch Site Names

- After running this command:

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

- We obtain the launch sites as follows:

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

- Hence, the four launch sites are:

1. 'CCAFS LC-40'
2. 'CCAFS SLC-40'

3. 'VAFB SLC-4E'

4. 'KSC LC-39A'

Launch Site Names Begin with 'CCA'

- An easy way to find the first 5 records where the launch sites begin with `CCA`, we run:

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

- We get the following table as output:

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

- To filter the data frame so that only records showing boosters from NASA (CRS) / NASA appear, we can run respectively:

```
%sql SELECT * FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)'
```

```
%sql SELECT * FROM SPACEXTABLE WHERE "Customer" LIKE '%NASA%'
```

- To calculate the total payload carried by boosters from NASA (CRS), we run:

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

PAYLOAD_MASS__KG_NASA (CRS)
45596

- To calculate the total payload carried by boosters from NASA, we run:

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

PAYLOAD_MASS__KG_NASA
107010

Average Payload Mass by F9 v1.1

- To calculate the average payload mass carried by booster version F9 v1.1, we run:

```
%%sql
SELECT ROUND(AVG(PAYLOAD_MASS__KG_), 2) AS AVERAGE_PAYLOAD_MASS__KG_
FROM SPACEXTABLE
WHERE "Booster_Version" LIKE '%F9 v1.1%'
```

- The average payload mass carried by booster version F9 v1.1 is around 2534 kg:

AVERAGE_PAYLOAD_MASS__KG_
2534.67

First Successful Ground Landing Date

- The dates of the first successful landing outcome on the ground pad can be found by running:

```
%%sql
SELECT Date FROM SPACEXTABLE
WHERE "Landing_Outcome" LIKE '%Success%'
AND "Landing_Outcome" LIKE '%ground pad%'
ORDER BY Date
LIMIT 1
```

- It was achieved on December 22nd, 2015 at 1:29 UTC. The full record features are shown in the table underneath:

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

Successful Drone Ship Landing with Payload between 4000 and 6000

- The boosters that have successfully landed on drone ship and had a payload mass greater than 4000 but less than 6000 are obtained by running:

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

- The list comprises 4 booster versions as depicted in this table:

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 can be gained by querying:

```
%%sql
SELECT
    CASE
        WHEN Mission_Outcome LIKE '%Success%'
        THEN 'Success'
        ELSE 'Failure'
    END AS "Mission Outcome",
    COUNT(Mission_Outcome) AS TOTAL
FROM SPACEXTABLE
GROUP BY "Mission Outcome";
```

- All missions achieved successfully their goals except for one mission:

Mission Outcome	TOTAL
Failure	1
Success	100

Boosters Carried Maximum Payload

- The List of boosters that have carried the maximum payload mass can be obtained by running:

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

- The table shows a list of 12 distinct booster versions:

#	Booster_Version	PAYLOAD_MASS__KG_
1	F9 B5 B1048.4	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1051.3	15600
4	F9 B5 B1056.4	15600
5	F9 B5 B1048.5	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1049.5	15600
8	F9 B5 B1060.2	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1051.6	15600
11	F9 B5 B1060.3	15600
12	F9 B5 B1049.7	15600

2015 Launch Records

- The failed landing outcomes on drone ship, their booster versions, and launch site names for the year 2015 can be listed after executing:

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

- The result shows two failed missions: one in January and the other in April:

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

- In order to rank the count of landing outcomes between the dates 2010-06-04 and 2017-03-20, in descending order, we can run this query:

```
%%sql
SELECT Date, "Landing_Outcome", COUNT("Landing_Outcome") AS Total
FROM SPACEXTABLE
WHERE Date >= '2010-06-04'
AND Date <= '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY Total DESC
```

- The total count for the corresponding landing outcome is regrouped in this table:

Date	Landing_Outcome	Total
2012-05-22	No attempt	10
2016-04-08	Success (drone ship)	5
2015-01-10	Failure (drone ship)	5
2015-12-22	Success (ground pad)	3
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2010-06-04	Failure (parachute)	2
2015-06-28	Precluded (drone ship)	1

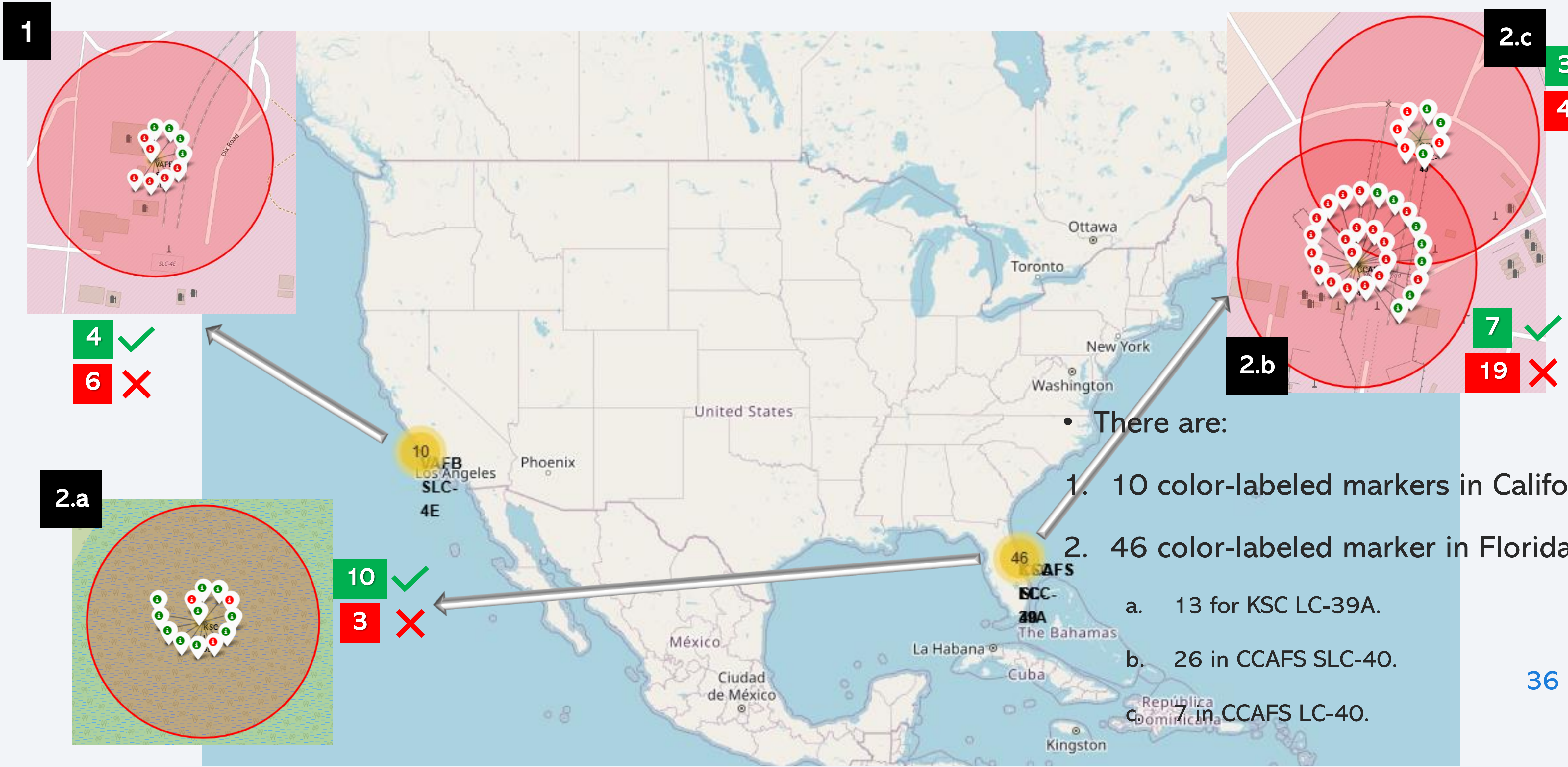
Section 3

Launch Sites Proximities Analysis

Folium Map1 _ All launch sites' locations

- All launch sites are:
  - In proximity to the Equator line → Far away from the planet's axis → Higher inertia → Less energy consumption.
  - In close proximity to the coast → Less risk of falling debris, accidents, explosions, etc.

Folium Map2 _ Color-coded launch records' outcomes

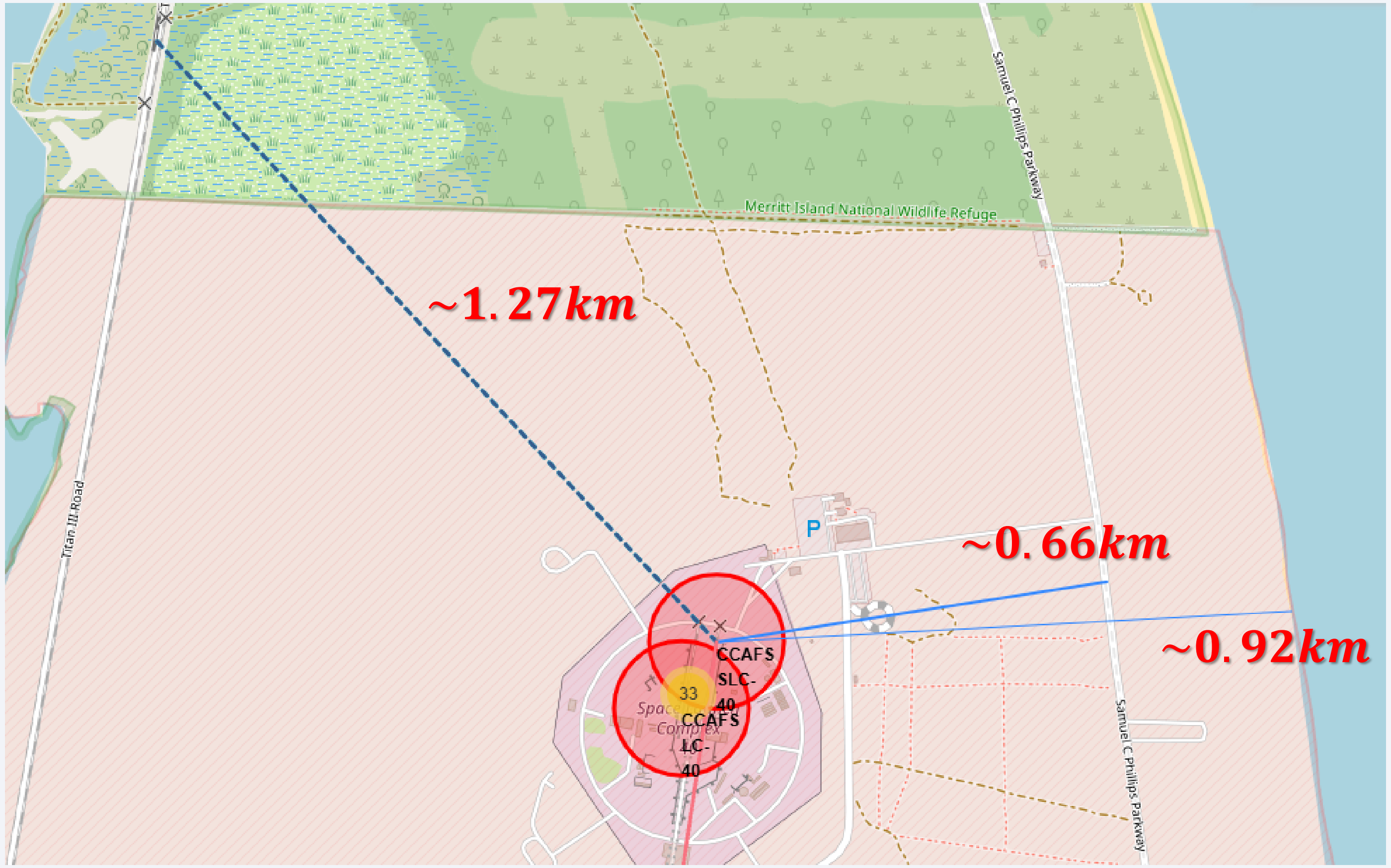


- There are:

1. 10 color-labeled markers in California.
2. 46 color-labeled marker in Florida:
   a. 13 for KSC LC-39A.
   b. 26 in CCAFS SLC-40.
   c. 7 in CCAFS LC-40.

Folium Map3 _ CCAFS SLC-40 Launch site to its proximities

- Situated in just the right place from the nearest city (far away ~18km) and close to the coast → Reduced risk of falling debris, accidents, explosions…
- In proximity to highways and roadways → Ease of supply and delivery.



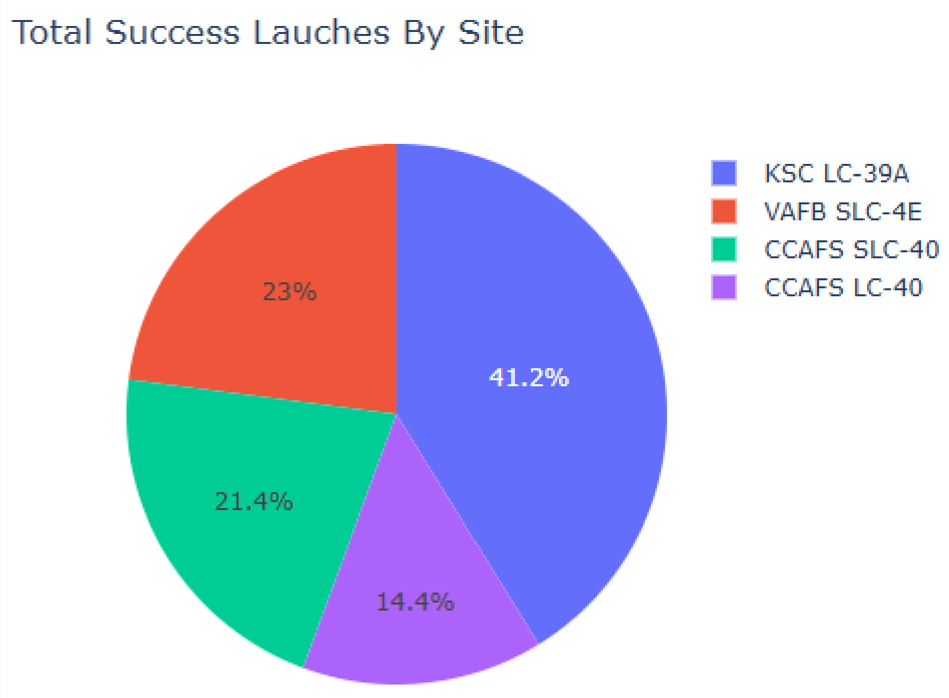
Section 4

Build a Dashboard with Plotly Dash

Total successful launches by site _ Largest successful launches' site

- It is obvious that amid all launch sites, **KSC LC-39A** has the largest successful launches (~ 41.2%). In fact, we can compute it as follows:

$$successful\_launches(KSC\ LC\!-\!39A) = \frac{10}{4+10+3+7} \cong \mathbf{41.2\%}$$

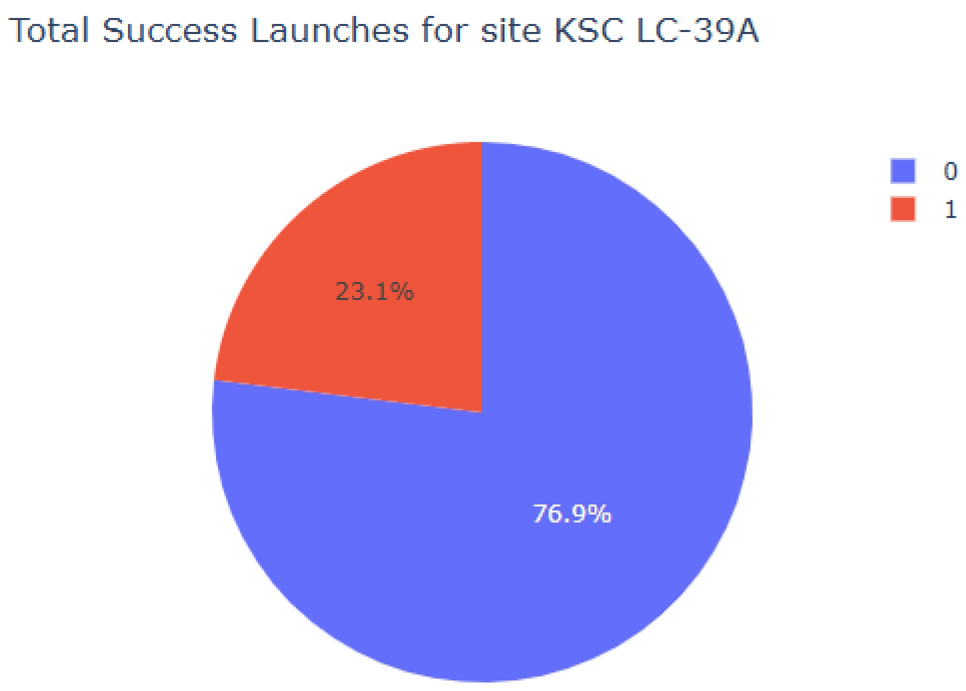


Launch site with the highest launch success rate

- The launch site **KSC LC-39A** can be considered the most successful site with a success rate of 76.9%:

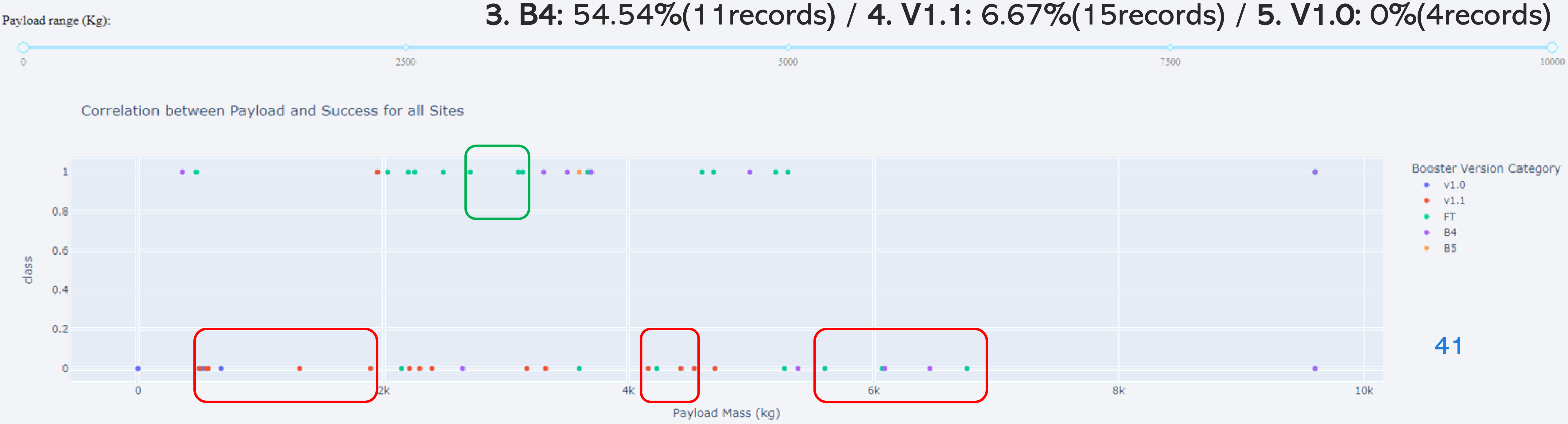
$$success\_rate(KSC\ LC{-}39A) = \frac{10}{10+3} \cong \mathbf{76.9\%}$$

- The success rate of the other sites in decreasing order are outlined below:
  - **CCAFS LC-40:** 73.1%.
  - **VAFB SLC-4E:** 60%.
  - **CCAFS SLC-40:** 57.1%.



Payload vs. launch outcome scatter plot _ all sites

- We notice that the payload ranges show:
- The highest launch success rate : [2700-3150] with 100%.
- The lowest launch success rates are: [500-1900], [4100-5050] and [5500-7000] with 0%.
- The F9 Booster versions can be ranked in decreasing order with respect to their relative launch success rates: **1. B5**: 100%(1record) / **2. FT**: 65.2%(23records) / **3. B4**: 54.54%(11records) / **4. V1.1**: 6.67%(15records) / **5. V1.0**: 0%(4records)



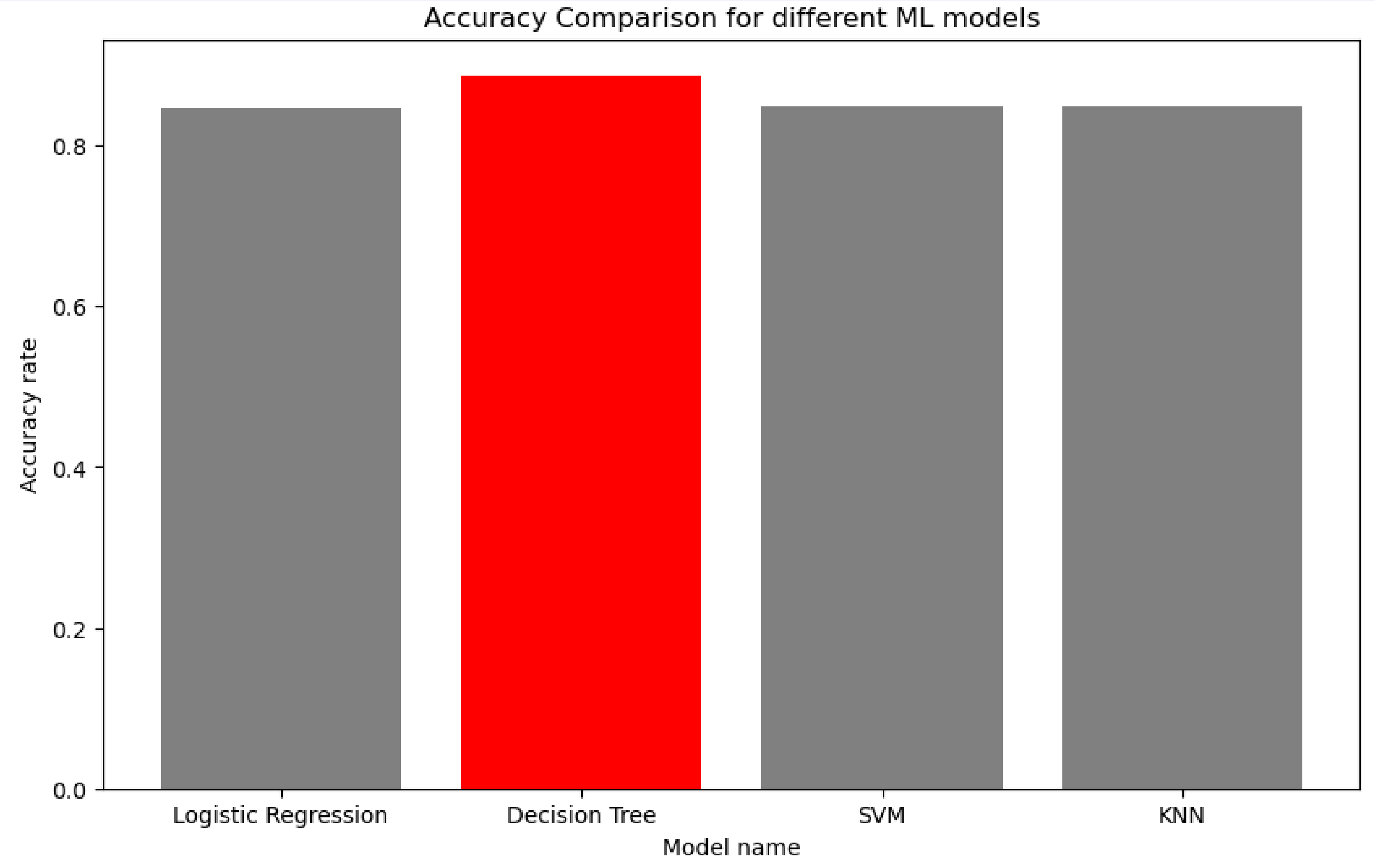
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The following bar chart compares the accuracy for all built classification ML models via a Grid Search Cross Validation approach.

→ The decision Tree model seems to be slightly the best model with the highest classification accuracy ofabout 89%.



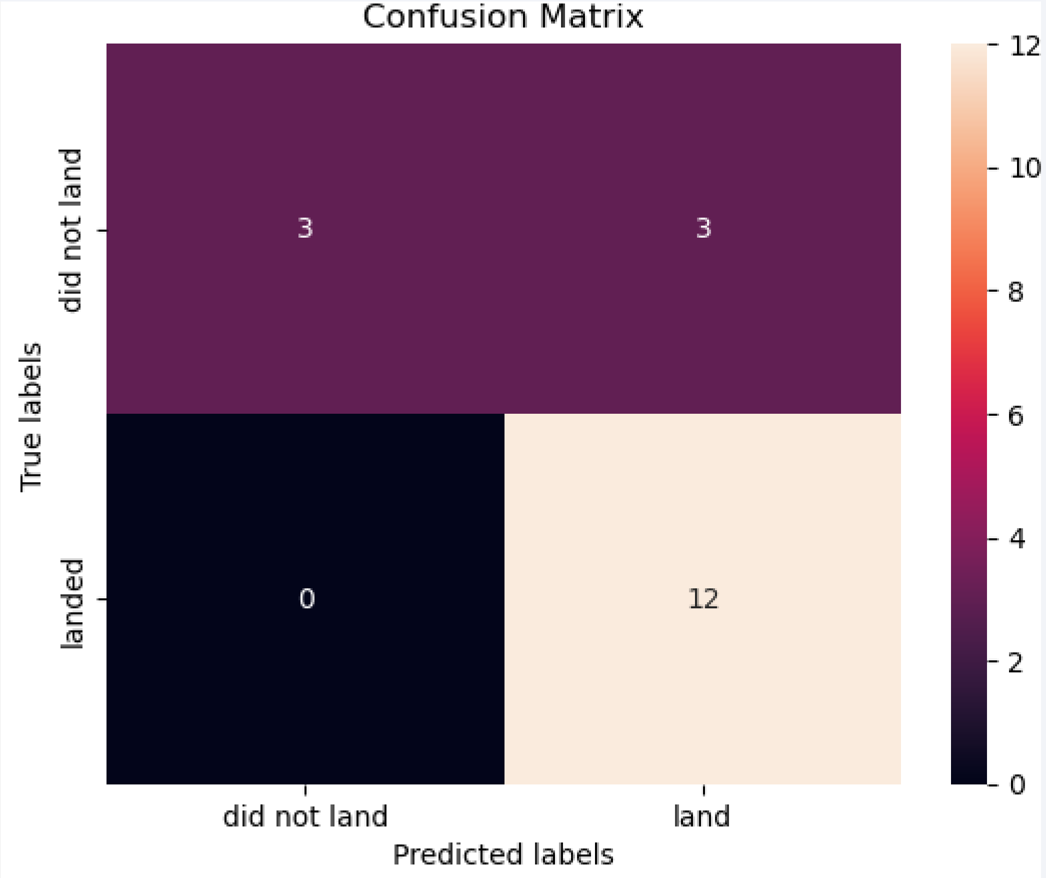
	Model Name	GSCV Accuracy
0	Logistic Regression	0.846
1	Decision Tree	0.888
2	SVM	0.848
3	KNN	0.848

Confusion Matrix

- TP = 12
- TN = 3
- FP = 3
- FN = 0

→ The model correctly identified 12 positive and 3 negative cases. It mistakenly classified 3 negative cases as positive and didn't miss any positive cases.

- Overall, the model missed only 3 values:



Conclusions

- This study has demonstrated how conducting a data science methodology can help data scientist to effectively analyze raw data, reveal meaningful insights and make reliable predictions, particularly in the case of SpaceX's Falcon 9 launches. The main steps encompasses data collection and cleaning, features and insights extraction, patterns exploration, and model construction and evaluation.
- Our main findings include:
  - The KSC LC-39A was the top performing launch site, with the highest success rate of 76.9%.
  - The highest success rate of 100% is reached for ES-L1, Geo, Heo and SSO. The company could invest more in these orbits and reassess the SO orbit launches with one failed launch.
  - The success rate is somewhat highly correlated with payload mass, where heavier payloads showed higher success rates.
  - The Decision Tree model can be considered as the best performing ML model through GridSearchCV with around 89% as accuracy score. Other classification models have roughly 84%.
- By identifying trends in launch sites, and other contributing factors, we were able to draw data-driven conclusions about mission performance.
- This study could be extended by incorporating predictions, tuning models based on user feedback and deploying product seamlessly into a CI/CD pipeline for continuous improvement.

Appendix: Acknowledgements

- I would like to warmly and gratefully thank:
  - All the instructors of this professional certificate course series for their dedication and expertise.
  - Coursera for offering such valuable leaning experience.

Thank you!