



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Name: Rafael Mondragón

Date: 06/22/2024



# Outline

---

- Executive Summary ..... 3
- Introduction ..... 4
- Methodology ..... 5-33
- Results ..... 34-45
- Conclusion ..... 46
- Appendix ..... 47

# Executive Summary

---

- Summary of Methodologies

The whole project was based on the use of Python libraries to perform the different stages of the Data Science Project. Thus, Python was used from the data collection with *requests* and *BeautifulSoup* and data wrangling with *Pandas*, all the way to ML models with *scikit-learn* and dashboards with *Plotly Dash*, showing how powerful it is as Data Science tool.

- Summary of all Results

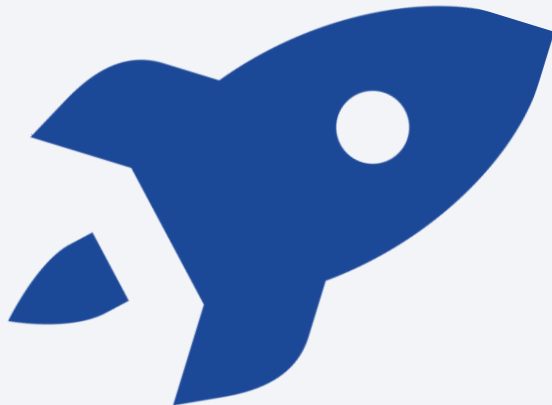
The final results were based on the creation of ML models through which we can evaluate their accuracy to predict if the landing of a launch will be successful or not. To do this task, we pass through the stages of EDA and data visualization in order to get a clearer understanding of the relationships between features and the expected outcome, so we can transmit this insights to our ML models.

# Introduction

---

- Project background and context

Commercial space voyages have become a reality, so the race between companies to domain the market is at its peak. Among the many competitors like Virgin Galactic, Rocket Lab, Blue Origin, etc.; it seems that the one that has positioned itself best is SpaceX. With their Falcon 9 rocket, SpaceX, has achieved to create reusable first stages, which allows them to lower the cost of their launches to 62 million dollars, compared to the 165 million of other companies.



- Problems you want to find answers

As part of the new rocket company *SpaceY*, we would like to analyze the success that SpaceX has achieved. Therefore, we will analyze the different factors of a rocket launch (location, booster version, payload mass, orbit, etc.), to determine which are the most influential in a successful outcome, so that the first stage can be reused.



Section 1

# Methodology

# Methodology

---

## Executive Summary

- Data collection methodology.
  - The information was collected from two main sources, the SpaceX API and the Wikipedia page for “Falcon 9 and Falcon Heavy Launches Records”. For the first one, the “requests” library was used to obtain the information from the different endpoints. Subsequently, for the Wikipedia page, the BeautifulSoup library was used to obtain the Falcon 9 records by its HTML table. Finally, the Pandas library was used to save both sources of information as Pandas DataFrames.
- Perform data wrangling.
  - After collecting the data, the Pandas library was used to wrangle the data. In this way, we obtained some information about what each feature represented, the datatypes, missing values, etc. At the end, we created a column named “Class” which summarizes the outcome of the landing in a binary value (0-failed, 1-success), with the objective of making it easier to use in our visualizations and predictions.

# Methodology

---

## Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL.
  - Using the sqlite3 library, we perform some EDA to get a clearer view of the values and obtained some useful insights like unique values for Launch Site, Booster Version, Mission & Landing Outcomes. Additionally, seaborn and matplotlib libraries were used to create different graphs that help to visualize the correlation between features.
- Perform interactive visual analytics using Folium and Plotly Dash.
  - The folium library was utilized to create an interactive map, in which the different launch sites and their respective outcomes were displayed. Likewise, an interactive dashboard was generated with Plotly Dash, to show the impact of the different Launch Sites and Booster Versions in the outcome.

# Methodology

---

## Executive Summary

- Perform predictive analysis using classification models.
  - By taking advantage of the Scikit Learn library, we created and fitted, 4 ML models, based on the next algorithms: Logistic Regression, Support Vector Machine, Decision Tree Classifier and K Nearest neighbors. We then used the Grid Search technique to obtained the best hyperparameters for each model, and evaluate their accuracy to determine which was the best one.



# Data Collection – SpaceX API

---

Requests library

- Use requests library to obtain the data in JSON format.

JSON\_normalize

- With Pandas library use method `json_normalize` to transform the JSON info into a Pandas DataFrame.

Endpoints

- With the different endpoints, use the request library to complete the information from the DataFrame,

Falcon 9

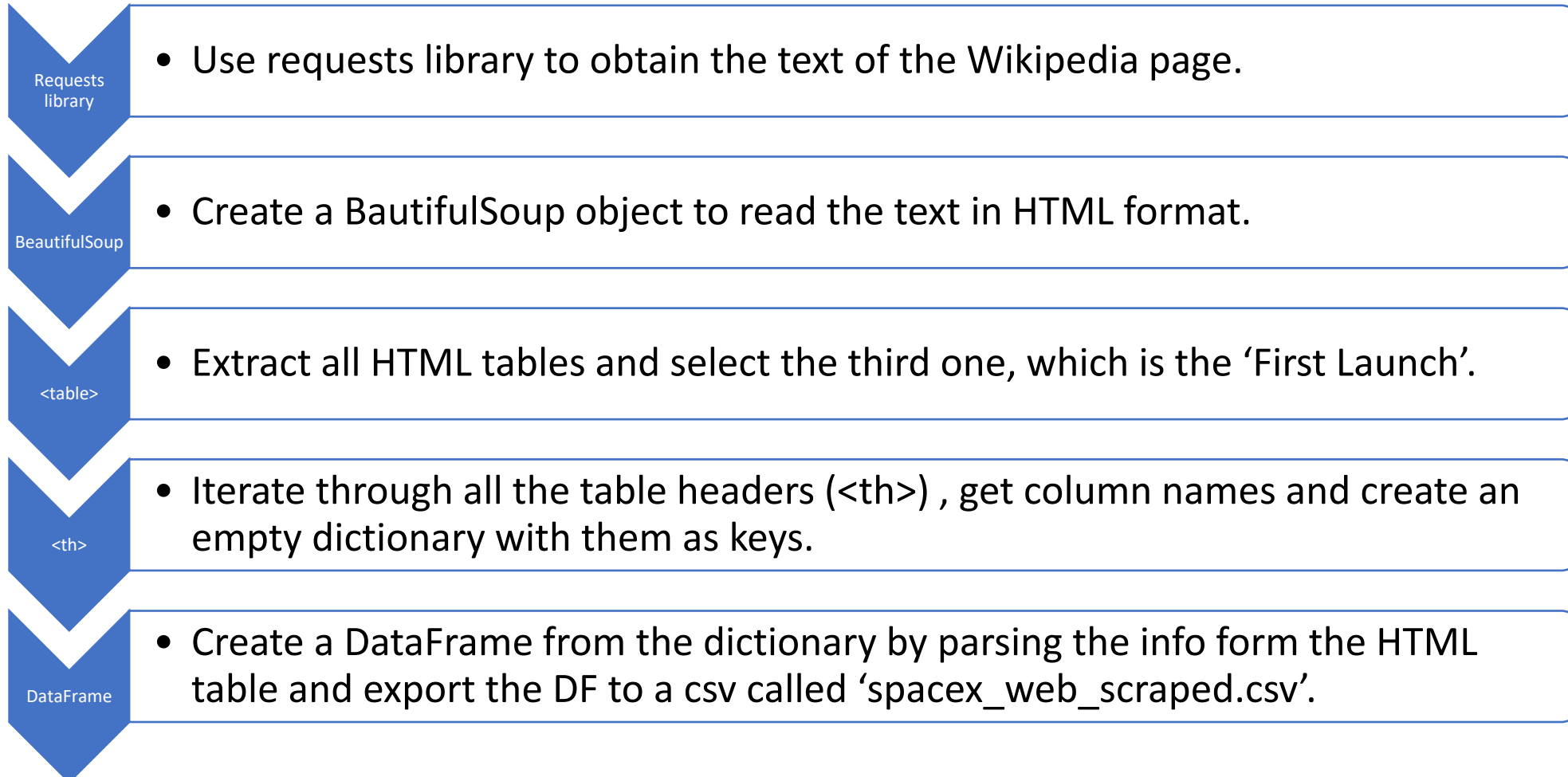
- Filter DF's information to only handle the Falcon 9 Booster Version.

Null Values

- Handle Payload Mass null values by replacing them with the mean and export the DF to a csv called 'dataset\_part\_1.csv'.

GitHub URL: [https://github.com/MonDrachen/DataScience\\_SpaceX/blob/main/jupyter-labs-spacex-data-collection-api.ipynb](https://github.com/MonDrachen/DataScience_SpaceX/blob/main/jupyter-labs-spacex-data-collection-api.ipynb)

# Data Collection – Scraping



GitHub URL: [https://github.com/MonDrachen/DataScience\\_SpaceX/blob/main/jupyter-labs-webscraping.ipynb](https://github.com/MonDrachen/DataScience_SpaceX/blob/main/jupyter-labs-webscraping.ipynb)

# Data Wrangling

---


Read 'dataset\_part\_1.csv' file as a pandas DF.



Explore DF, get percentage of null values, data types and check value counts of features.



Create a new column named 'Class', representing landing outcomes as a binary value (0:failed, 1:success).



Export the new DF as a csv file named 'dataset\_part\_2.csv'.

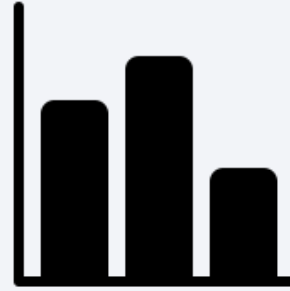
# EDA with Data Visualization

---



## Scatter Plot

- Show correlation between variables (Payload mass, Launch site, Orbit, etc).



## Bar Graph

- Compare the success rate of different orbits.



## Line Chart

- Show the evolution of success rate through the years.

# EDA with SQL

---

- In order to get a clearer understanding of our data, we performed several queries that helped to refine the information wanted. Some of the operations were:

## LIKE

- Used to search a specific pattern in a value.

## DISTINCT

- It was utilized to determine all the unique values that a feature has.

## SUM/AVG/COUNT

- Employed to create a short statistical analysis of the features

## MIN/MAX

- Allows to get insights from the range that a feature handles.



# Build an Interactive Map with Folium

---

- The main elements used in the interactive map were:

## Circle

- To identify where the different launch sites were located.

## Marker

- To show labels with the name of the different launch sites.

## Cluster and icon

- Create a set of the individual launches in a same site, differentiating if the outcome was a success or failure.

## Line

- To show the distance between launch sites and relevant locations, such as roads, coasts, cities, etc.

## Popup

- To present additional information when clicking a certain location.

GitHub URL:

[https://nbviewer.org/github/MonDrachen/DataScience\\_SpaceX/blob/main/lab\\_jupyter\\_launch\\_site\\_location.ipynb](https://nbviewer.org/github/MonDrachen/DataScience_SpaceX/blob/main/lab_jupyter_launch_site_location.ipynb)

# Build a Dashboard with Plotly Dash

---

## Graphs

- Pie Chart: To show distribution of successful launches by site / percentage of successful-failed outcomes per site.
- Scatter Plot: To show correlation between Payload Mass and Outcome, while checking the influence of Booster version.

## Interactions

- Dropdown: To select launch site of interest (or all sites) that will be displayed in the graphs.
- Range Slider: To adjust the range in which the scatter plot will show the Payload Mass (x-axis).

# Predictive Analysis (Classification)

---

Read 'dataset\_part\_2.csv' file as a pandas DF.




Separate the features (X) from the target (Y) and normalized the features with a *Standard Scaler*.



Train-test split with test\_size=0.2.



Define parameters for each ML algorithm (logistic regression, SVM, KNN, Decision Trees) and initialize a Grid Search object with each of them.



Fit the train data, calculate predictions with test data and evaluate each model with *score* function and *confusion matrix*. Compare and determine best fit.

GitHub URL:

[https://github.com/MonDrachen/DataScience\\_SpaceX/blob/main/SpaceX\\_Machine\\_Learning\\_Prediction\\_Part\\_5.jupyterlite.ipynb](https://github.com/MonDrachen/DataScience_SpaceX/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb)



The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of red and cyan. A faint, light blue grid pattern is also visible, particularly in the lower-left quadrant. The overall effect is dynamic and technological.

Section 2

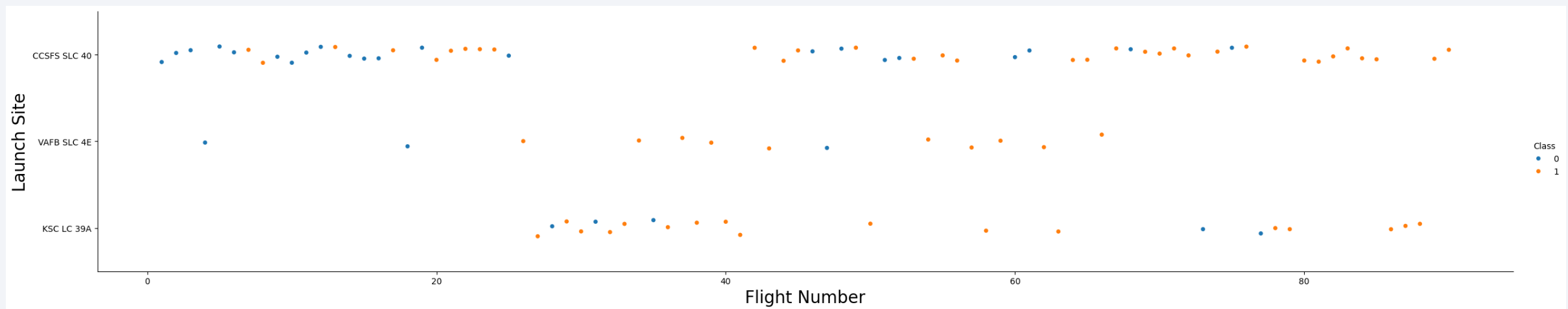
# Insights drawn from EDA



# Flight Number vs. Launch Site

---

- The majority of the launches were performed in CCSFS SLC 40.
- The last launches had a much better outcome compared to the first ones.
- VAFB SLC-4E and KSC LC-39A had a better success ratio than CCSFS SLC 40.

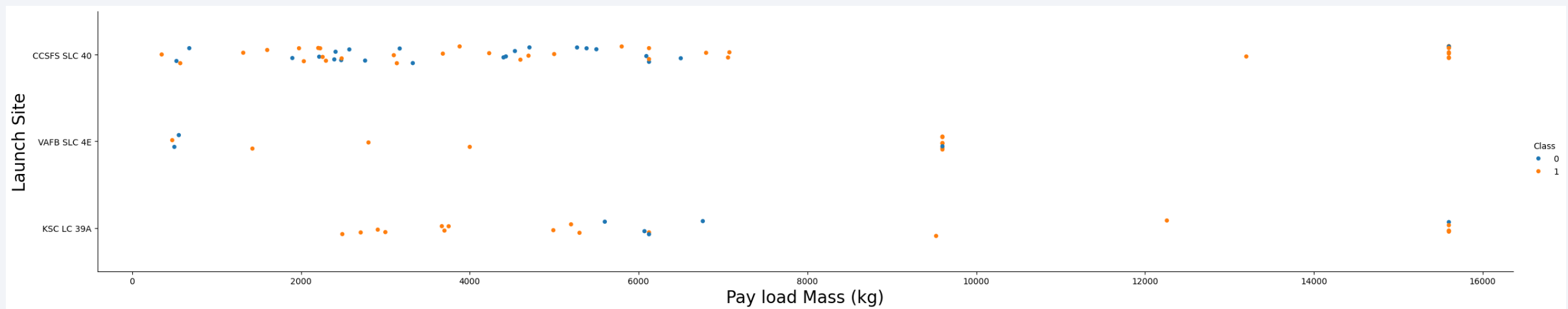




# Payload vs. Launch Site

---

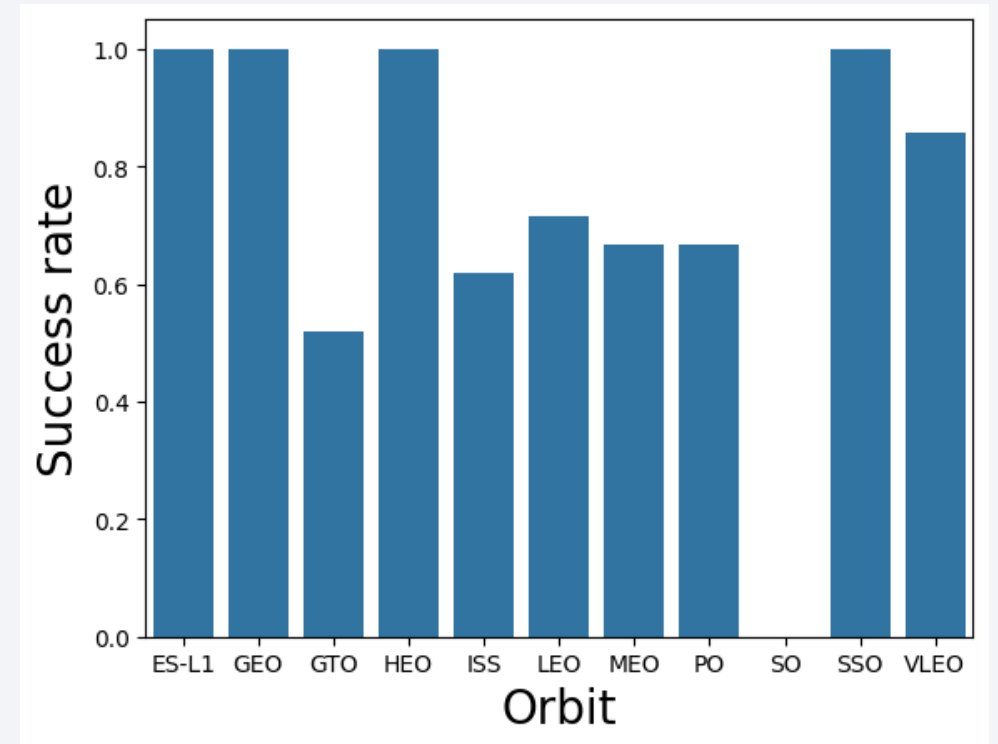
- The distribution of launches is mostly found in the range from 0 to 7000 kg of payload mass.
- The majority of the launches with a payload mass greater than 8000 kg, had a successful outcome.



# Success Rate vs. Orbit Type

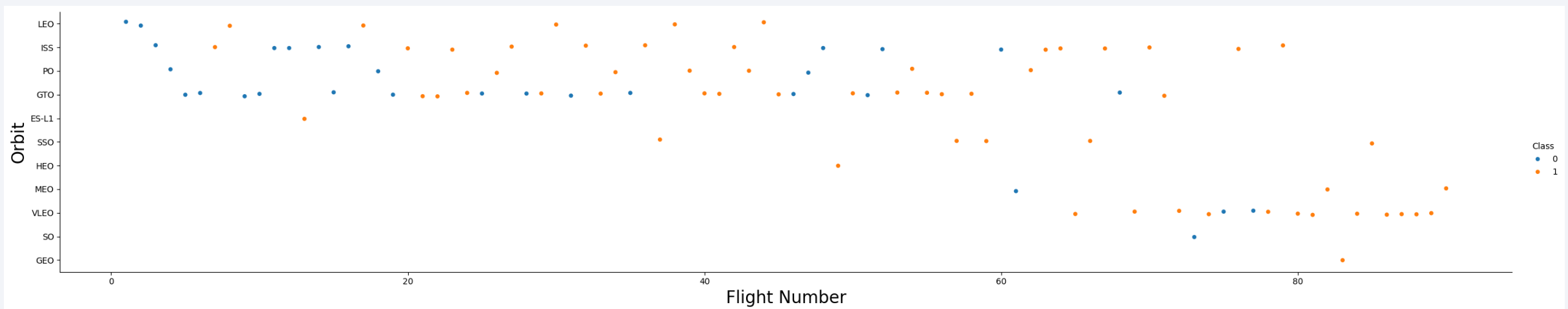
---

- The orbits ES-L1, GEO, HEO and SSO were the ones with the highest success rate (approx. 100%).
- SO orbit had a success rate of 0%.
- Almost all of the orbits had a success rate greater than 50%.



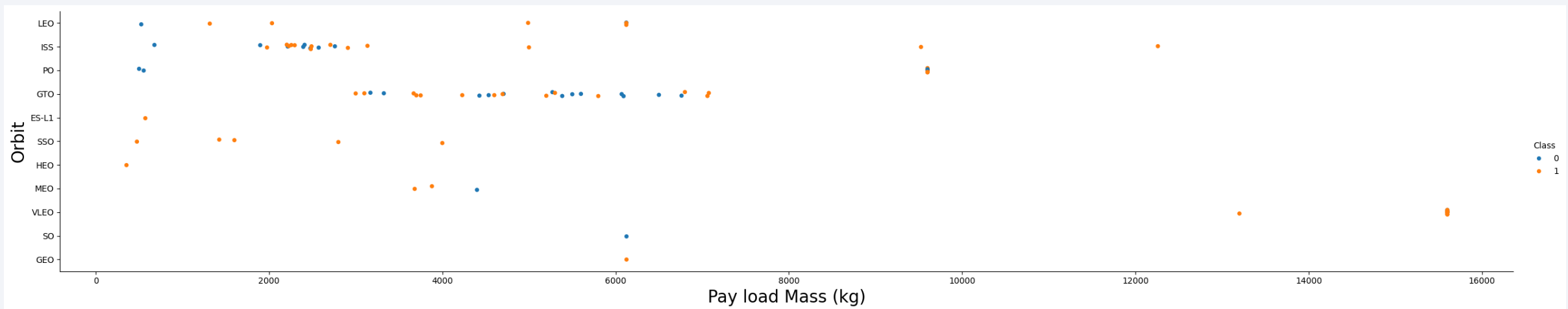
# Flight Number vs. Orbit Type

- The major part of the last flights were done in the VLEO orbit.
- Almost all of the flights were done in the orbits: LEO, ISS, PO, GTO.
- We can confirm that the ES-L1, SSO, GEO, and HEO orbits had a 100% accuracy. Although we have to take into account that their number of launches is significantly smaller than the ones of LEO, ISS, PO and GTO.



# Payload vs. Orbit Type

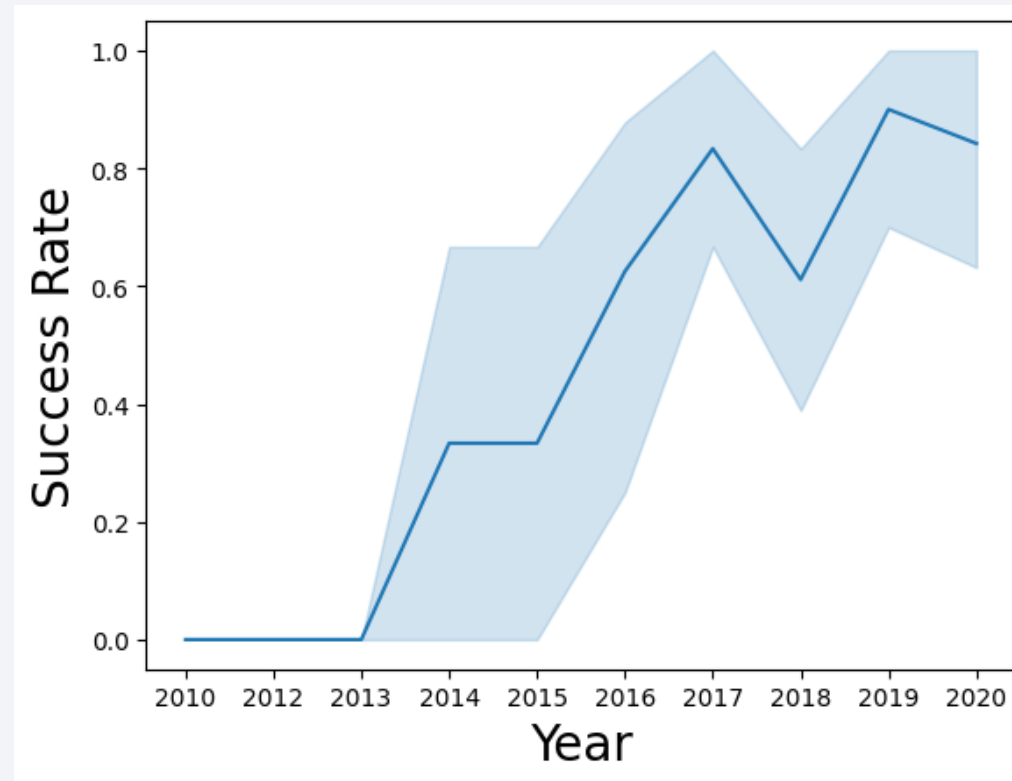
- The payload mass of orbit GTO is the one that is the most shrinked, it ranges from approx. 3,000kg to 7,000kg. Also, the major part of the ISS payload mass is in a range from 2,000kg to 4,000kg.
- LEO, SO and PO are the ones with the longest distributions.



# Launch Success Yearly Trend

---

- The first three years, the success rate was 0%.
- From 2013 to 2017, the success rate improved or stayed in the same value as the previous year.
- 2018 had a drop of almost 20% of success rate, but in 2019 it was recovered.
- After hitting its peak in 2019 (almost 90%), in 2020 the success rate dropped to approx. 80%.





# All Launch Site Names

---

- The DISTINCT keyword allows to retrieve the unique values of a column.
- There are 4 different Launch sites: CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, CCAFS SLC-40.

```
%sql select DISTINCT Launch_Site from SPACE_TABLE;
```

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

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

# Launch Site Names Begin with 'CCA'

- The LIKE keyword allows to find specific patterns (sub-strings) within a value, while the LIMIT keyword retrieves the specified # of records.
- The first 5 records that fulfill the query belong to the CCAFS LC-40.

```
%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 SUM keyword retrieves the total of a certain column. Combined with the where clause, we can specify that we are only interested in the rows in which NASA is the customer.
- The total payload mass of all records in which the customer was NASA is of 45,596 kg.

```
%sql select SUM(PAYLOAD_MASS_KG_) from SPACEXTABLE WHERE Customer = 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS_KG_)
45596

# Average Payload Mass by F9 v1.1

---

- The AVG keyword retrieves the mean of a certain column. Combined with the LIKE keyword, we can specify that we are only interested in the rows which have a Booster Version of the F9 v1.1.
- The average payload mass of all records which have a a Booster Version of the F9 v1.1 is 2534.65 kg.

```
%sql select AVG(PAYLOAD_MASS_KG_) from SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

```
Done.
```

<u>AVG(PAYLOAD_MASS_KG_)</u>
------------------------------

2534.6666666666665
--------------------

# First Successful Ground Landing Date

---

- By utilizing the MIN keyword in the DATE column, we can retrieve the earliest date. Combined with the LIKE keyword, we can specify that we are only interested in the successful ground pad outcomes.
- The earliest successful landing done on a ground pad dates back to 22 Dec, 2015.

```
%sql select MIN(Date) from SPACEXTABLE WHERE Landing_Outcome LIKE '%Success (ground pad)%'
```

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

```
Done.
```

MIN(Date)
-----------

2015-12-22
------------



## Successful Drone Ship Landing with Payload between 4000 and 6000

---

- By using the BETWEEN keyword we can specify a range of values for a column. Also, in this query the AND keyword is used to specify the conditions of the WHERE clause. Which are a successful landing on drone ship and a payload mass in the range of 4,000 to 6,000 kg.

```
%osql select Booster_Version from SPACEXTABLE WHERE Landing_Outcome LIKE '%Success (drone ship)%' AND PAYLOAD_MASS__KG_ BETWEEN 4000 and 5999;
```

```
* 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

---

- By using the COUNT keyword we can count the number of times that a certain value happens in the column.
- In the result, we can see that there was only 1 failure and 100 successful missions.

```
%sql SELECT MIssion_Outcome, COUNT(Mission_Outcome) FROM SPACEXTABLE GROUP BY Mission_Outcome;
```

\* sqlite:///my\_data1.db  
Done.

Mission_Outcome	COUNT(Mission_Outcome)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

# Boosters Carried Maximum Payload

---

- To obtain the booster versions with the highest payload mass, we had to create a subquery in which we obtained the max payload; once we had that value, we can use it in the main query in the where clause. We can see that the results are different versions of the F9 B5 B10XX.X

```
%sql select Booster_Version from SPACESTABLE WHERE PAYLOAD_MASS_KG_ = (SELECT MAX(PAYLOAD_MASS_KG_) FROM SPACESTABLE);
```

```
* 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

---

- Similar to the LIKE keyword, the SUBSTR() function is used to obtain a sub-string, however, instead of searching a certain value, it searches an initial and ending index. With this, we were able to obtain the month and year from the DATE column. Finally, the where clause specifies the year to be 2015 and a landing outcome of failure on drone ship.

```
%sql SELECT substr(Date,6,2) as Month, substr(Date,0,5) as Year, Booster_Version, Landing_Outcome, Launch_Site FROM SPACEXTABLE WHERE substr(Date,0,5) = '2015' and Landing_Outcome = 'Failure (drone ship)';
```

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

Month	Year	Booster_Version	Landing_Outcome	Launch_Site
01	2015	F9 v1.1 B1012	Failure (drone ship)	CCAFS LC-40
04	2015	F9 v1.1 B1015	Failure (drone ship)	CCAFS LC-40

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

---

- In this query, we counted the landing outcomes, between '2020-06-04' and '2017-03-20', by using the BETWEEN and COUNT keywords. Then we ordered the result by the landing outcome count, using the ORDER BY clause in descending order.

```
%sql SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' and '2017-03-20' GROUP BY Landing_Outcome ORDER BY 2 DESC;
```

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

Landing_Outcome	COUNT(Landing_Outcome)
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

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

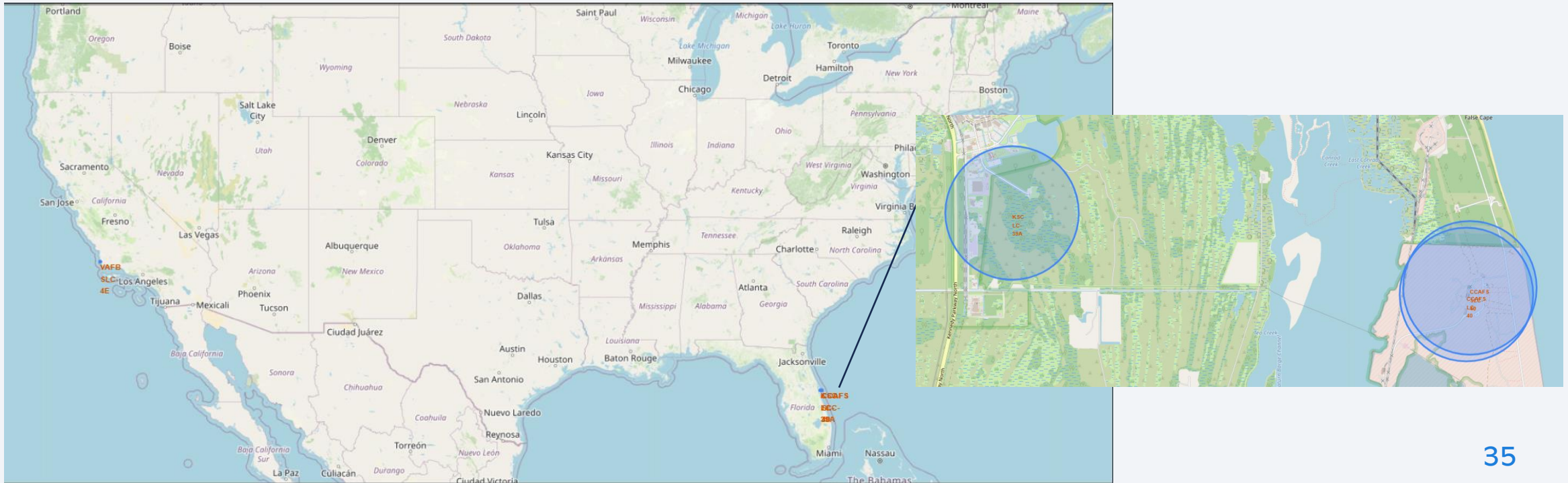
Section 3

# Launch Sites Proximities Analysis



# Launch Sites Map

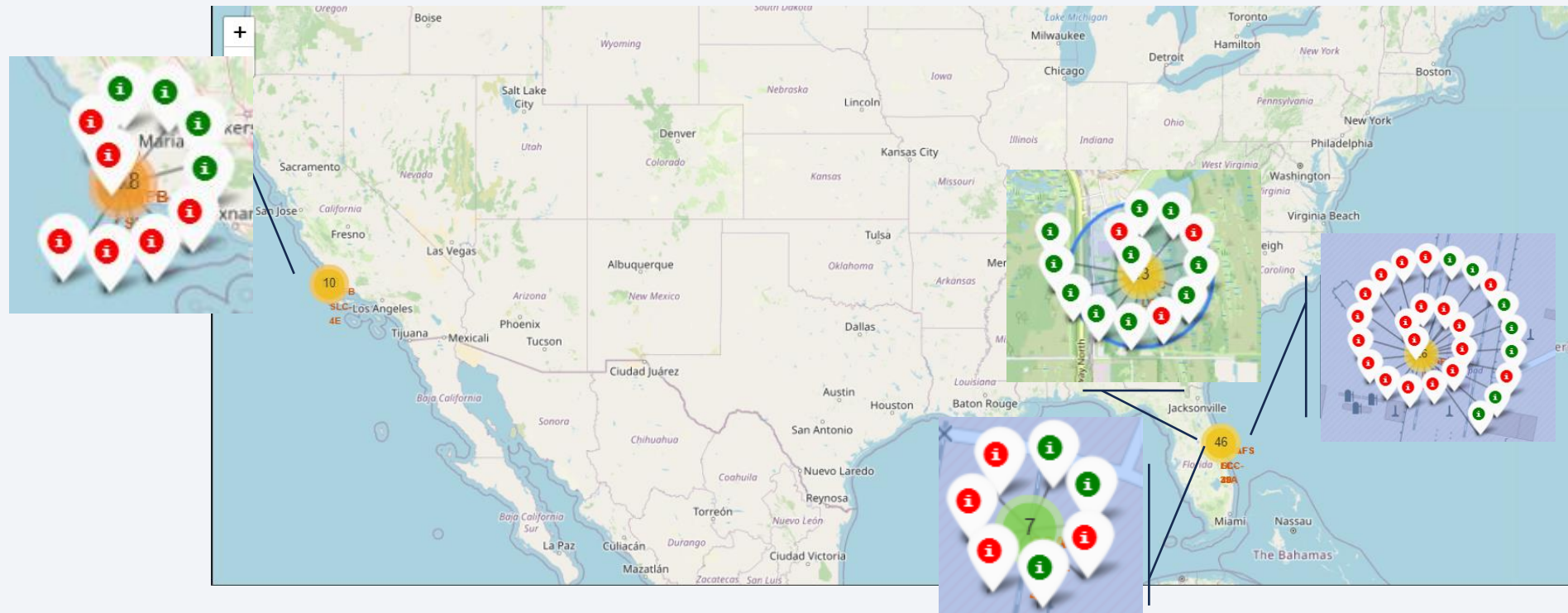
- The blue circles indicate the location of the sites, while the orange markers are the labels with their respective names.
- There is only one launch site in the west coast.
- The three launch sites in the east coast are very close, so we have to zoom in to distinguish them.





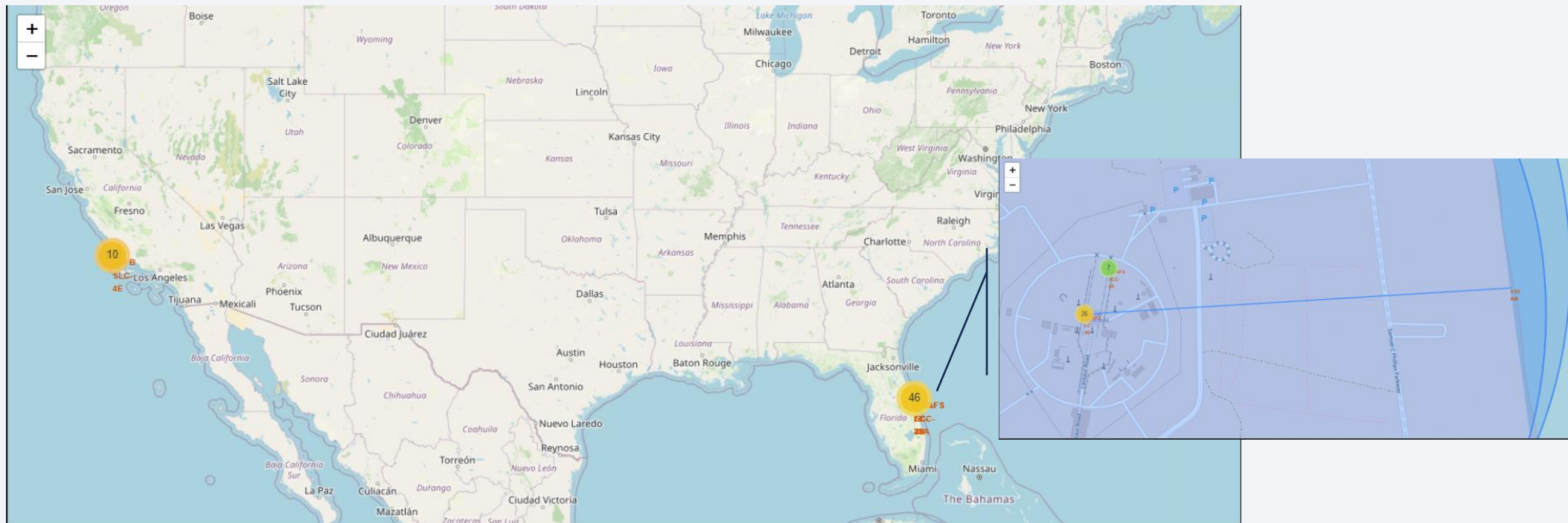
# Launch Records per Site Map

- The clusters allow to visualize the total of launches in a certain part, and when we zoom in we can visualize each site with their respective outcomes (green icon for success and red for failure).
- The KSC LC-39A is the launch site with the best success rate.
- The CCAFS LC-40 is the launch site with the most launches but the worst success rate.



# Nearest Coast Map

- The blue line indicates the distance from the CCAFS LC-40 to the coast (0.93km). This was calculated by using the MousePosition class, with which we obtained the coordinates for the specific point. Then, we calculate the distance and used a marker and a line to show them in the map.







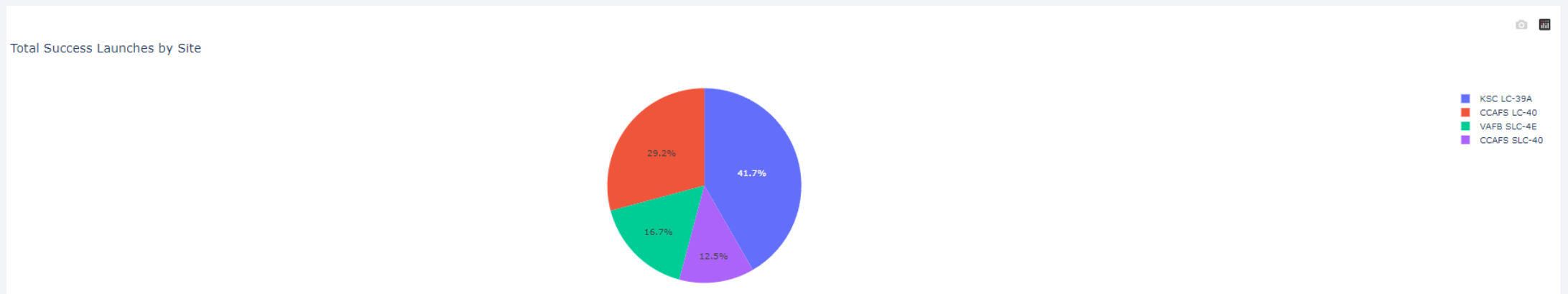
Section 4

# Build a Dashboard with Plotly Dash

# Distribution of Successful Launches by Site

---

- KSC LC-39A has almost half of the successful launches.
- CCAFS LC-40 and VAFB SLC-4E have a similar percentage being this, less than 20%.
- 7 out of 10 of the successful launches were done either in KSC LC-39A or CCAFS LC-40.



# Launch Outcomes in KSC LC-39A

---

- 3 out of every 4 launches have a successful outcome in the KSC LC-39A launch site.
- By reviewing the other graphs, we realize that KSC LC-39A was the only launch site with a positive outcome (>50% positive outcome).

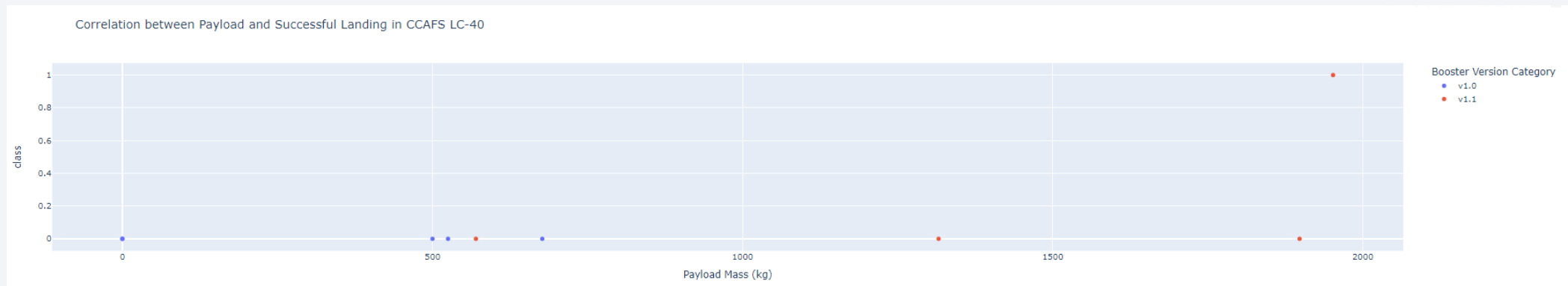
Launch Outcomes in KSC LC-39A



# CCAFS LC-40 Payload vs Landing Graph

---

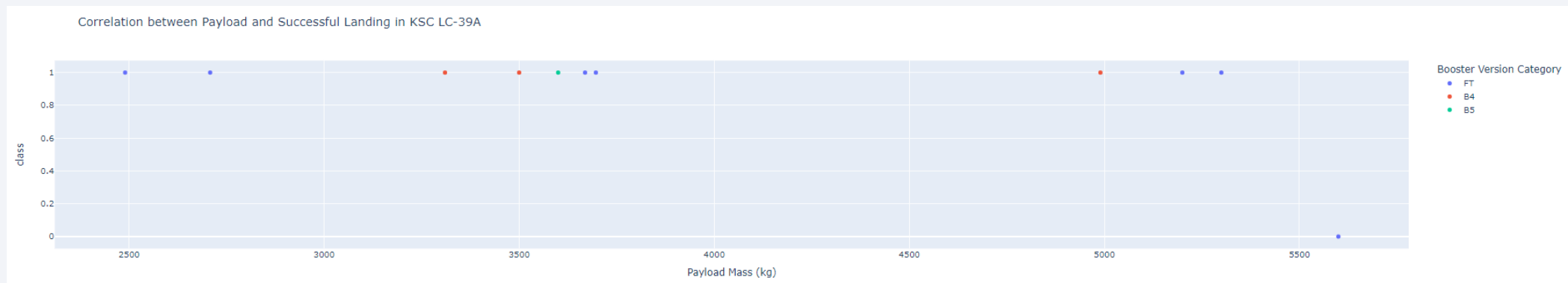
- In the range from 0 to 2,000 kg of payload mass, CCAFS LC-40 has a deficient performance with almost all of its landings being a failure, regardless of the Booster version used.





# KSC LC-39A Payload vs Landing Graph

- In the range from 2,000 to 6,000 kg of payload mass, KSC LC-39A has a great performance with almost all of its landings being a success, regardless of the Booster version used.



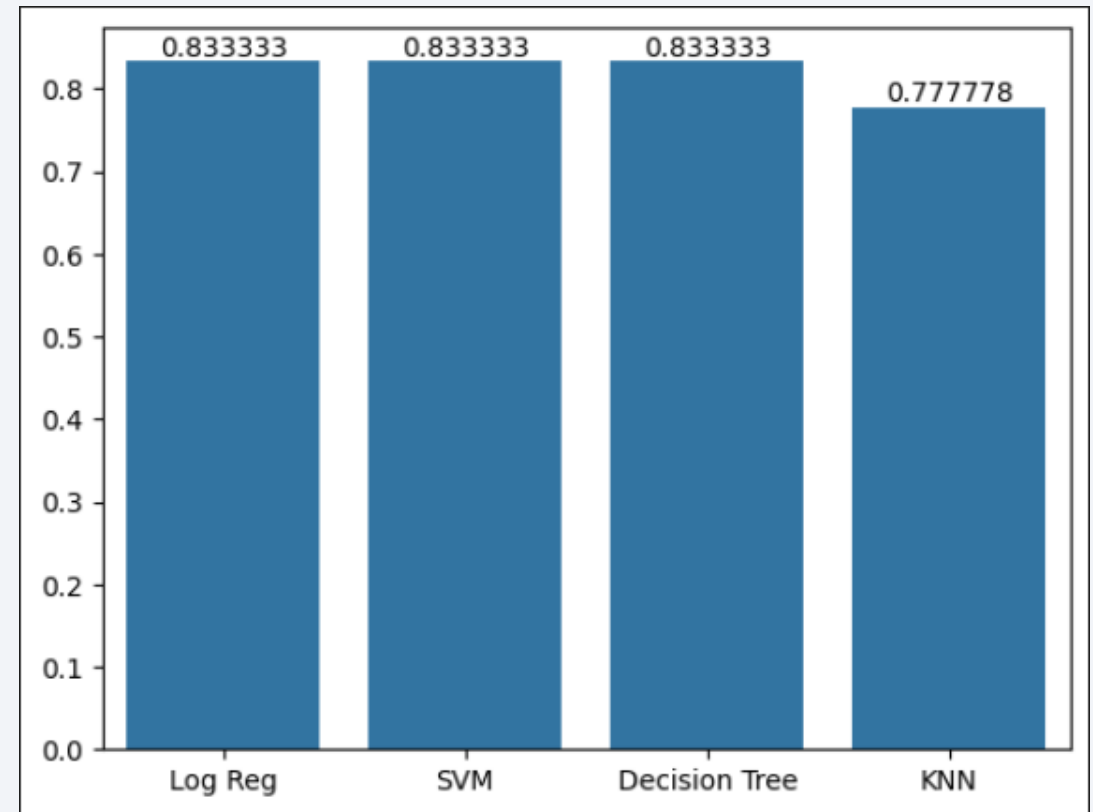
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

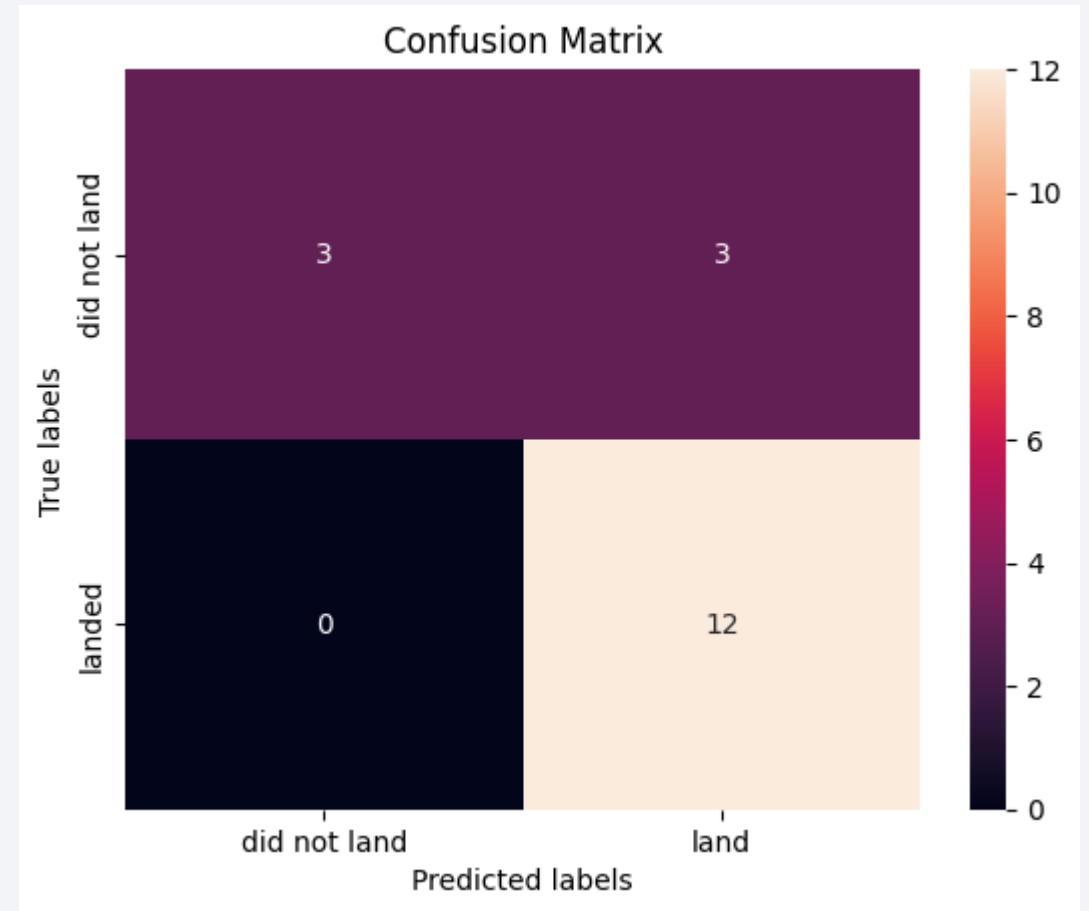
---

- From the graph, we can visualize that the Logistic Regression, the Support Vector Machine and the Decision Tree algorithm had the best performance when calculating the accuracy in the test set; so in theory, each one of them represents a good option. Additionally, we can't discard the KNN model, since its performance is not so far away from the others.



# Confusion Matrix

- The Logistic Regression, the SVM and the Decision Tree Classifier had the same confusion matrix. Here, we can appreciate that the successful landing outcomes were predicted with a 100% accuracy, while the failed ones had a 50% accuracy.
- Out of the 15 predicted successful landings, 3 of them were not correct; so we can state that when a success is predicted, it has 80% accuracy of being true.

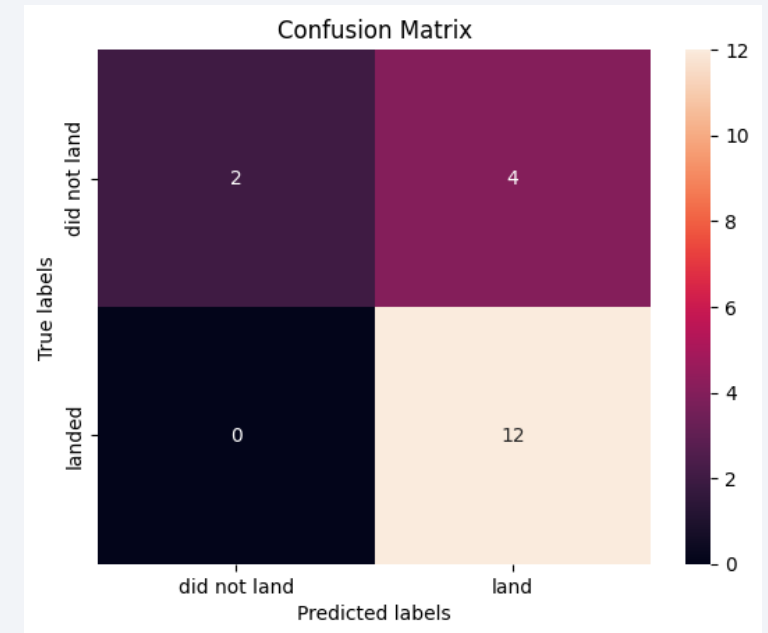
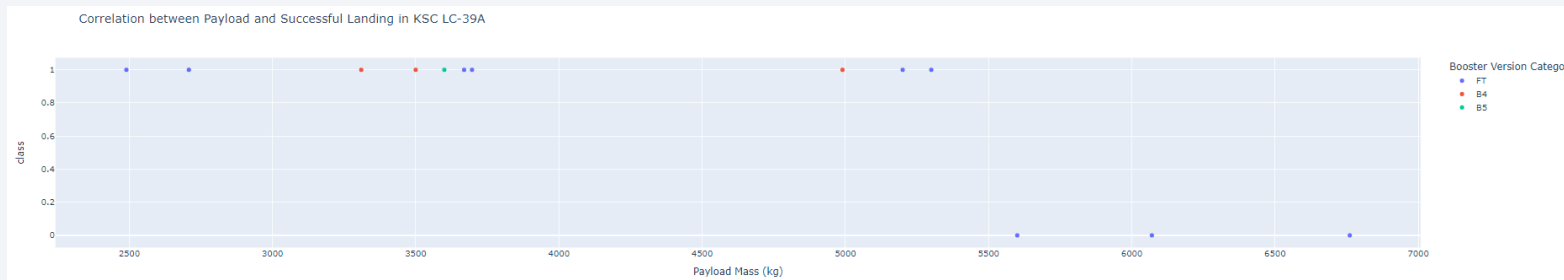
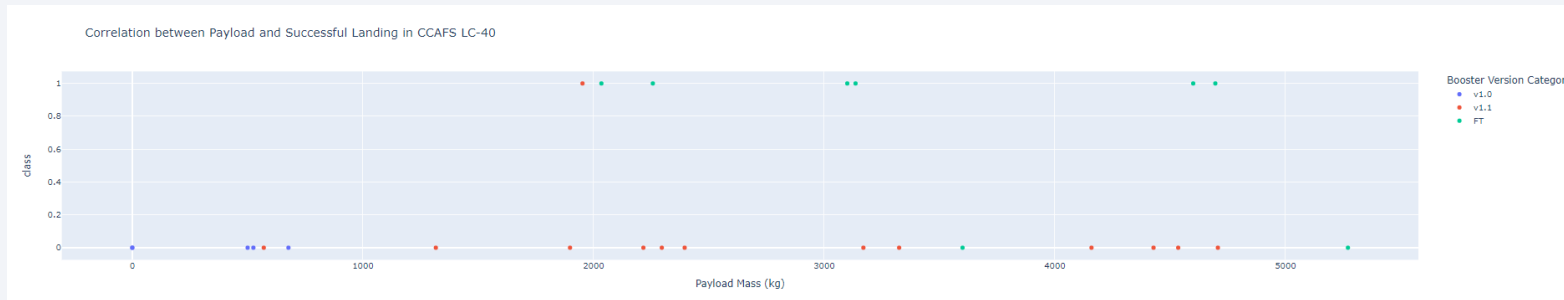


# Conclusions

---

- The similarity in the performance of the ML models can be due to the fact that the test size is very small (only 18 records), so in order to get a better understanding of which is the best, we will need to increment the number of records.
- The KSC LC-39A launch site is the one that performs the best in success rate terms.
- The overall trend of successful landings has had an almost continuous increment through the last 10 years, so it is expected that this trend continues in the next years, reaching a percentage of almost 90%.
- In terms of the booster version, we can state that there is not a clear influence over the landing outcome, unlike orbit, launch site and payload mass.

# Appendix



KNN Confusion Matrix



Thank you!

