



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Luís Fortes  
24 of July 2023



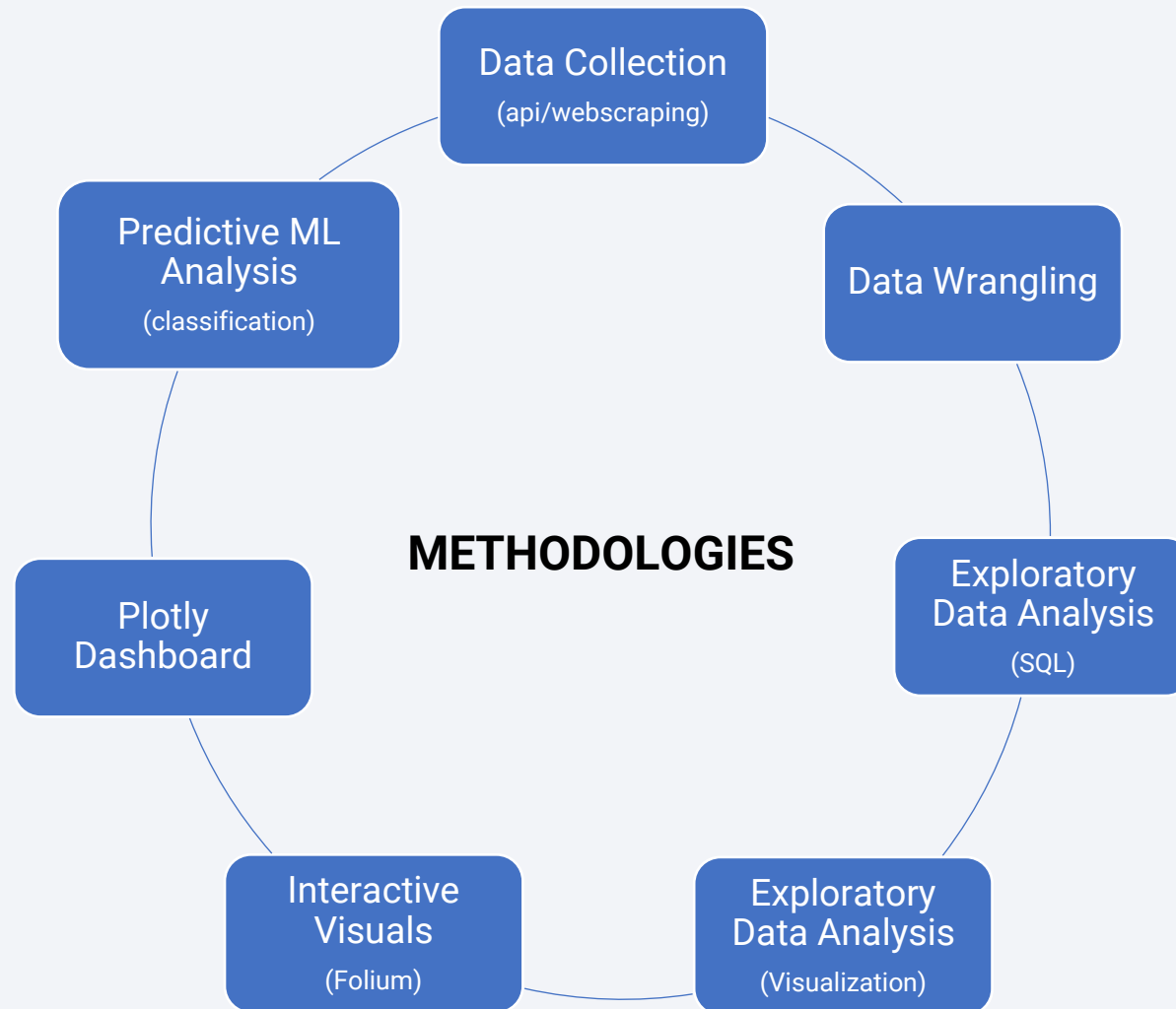
# Outline

---

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

# Executive Summary

---



# Executive Summary

---

## RESULTS



Exploratory Data Analysis



Interactive Analytics



Predictive Analytics

# Introduction

---



## Predicting Falcon 9 First Stage Landing Success

One of the **key innovations introduced by SpaceX** is the **reusable rocket** technology, particularly exemplified by the Falcon 9 rocket. By reusing the first stage of the Falcon 9 rocket, SpaceX has been able to drastically **reduce the cost** of launching payloads into space, making space missions **more accessible** and **cost-effective**.

The main objective of this data science project is to **build a predictive model** that can determine the likelihood of the Falcon 9 first stage landing successfully during rocket launches and this determine the cost of a launch.



Section 1



# Methodology

# Methodology

---

## Executive Summary

### **Data collection methodology:**

-  SpaceX API
-  Web scraping from Wikipedia

### **Perform data wrangling**

-  One-Hot Encoding

### **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

---

🚀 Data was collected through:

🚀 SpaceX API

🚀 Data collected via get request

🚀 Decoded the data as JSON ( *.json() method* ) and transformed it into a dataframe using built in pandas method *normalize ( .json\_normalize() )*.

🚀 Web Scraping

🚀 Scrapped the SpaceX data from Wikipedia , using a python the python library *BeautifulSoup*. Gathered the response as HTML tables, parsed the tables and converted it into a pandas dataframe.



# Data Collection – SpaceX API

<https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/jupyter-labs-spacex-data-collection-api.ipynb>

---

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

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

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

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

```
# Use json_normalize meethod to convert the json result into a dataframe  
data = pd.json_normalize(response.json())
```

# Data Collection – Scraping

<https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/jupyter-labs-webscraping.ipynb>

To keep the lab tasks consistent, you will be asked to scrape the data from a snapshot of the `List of Falcon 9 and Falcon Heavy launches` Wikipage updated on `9th June 2021`

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, 'html.parser')
```

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

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

We will create an empty dictionary with keys from the extracted column names in the previous task. Later, this dictionary will be converted into a Pandas dataframe

```
launch_dict = dict.fromkeys(column_names)

# Remove an irrelevant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster'] = []
launch_dict['Booster landing'] = []
launch_dict['Date'] = []
launch_dict['Time'] = []
```

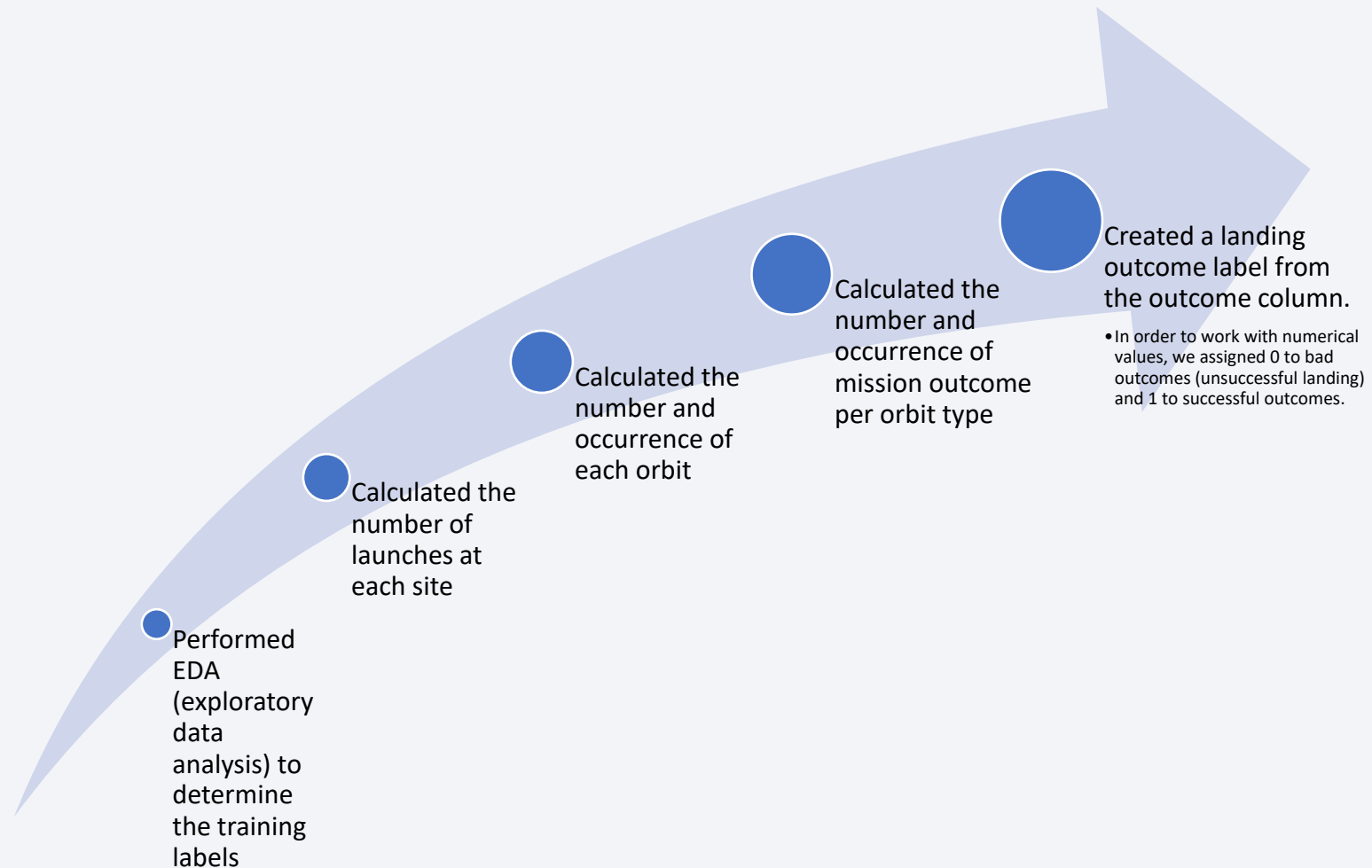
After you have fill in the parsed launch record values into `launch_dict`, you can create a dataframe from it.

```
df = pd.DataFrame(launch_dict)
df
```

# Data Wrangling

[https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/labs-jupyter-spacex-data\\_wrangling\\_jupyterlite.jupyterlite.ipynb](https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb)

---

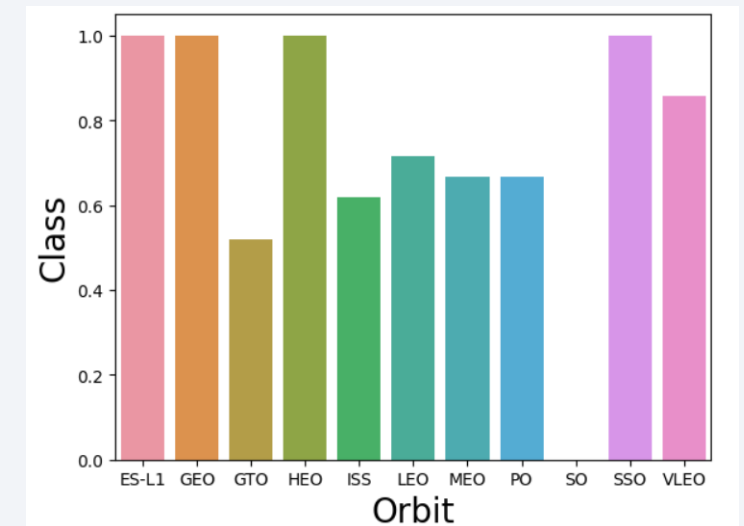
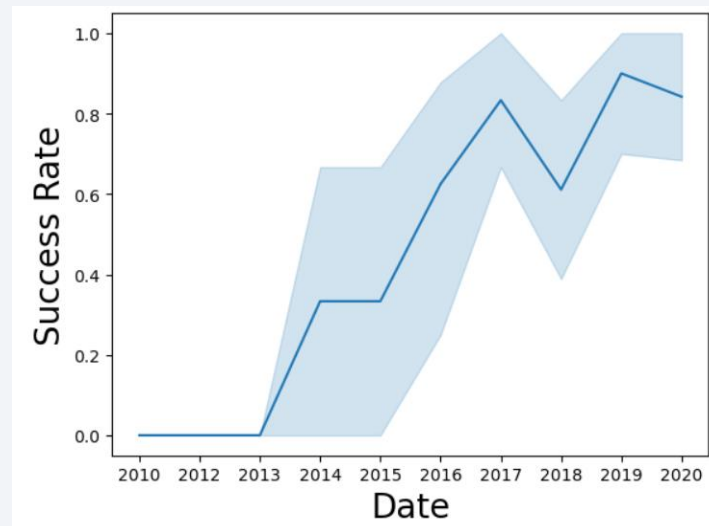
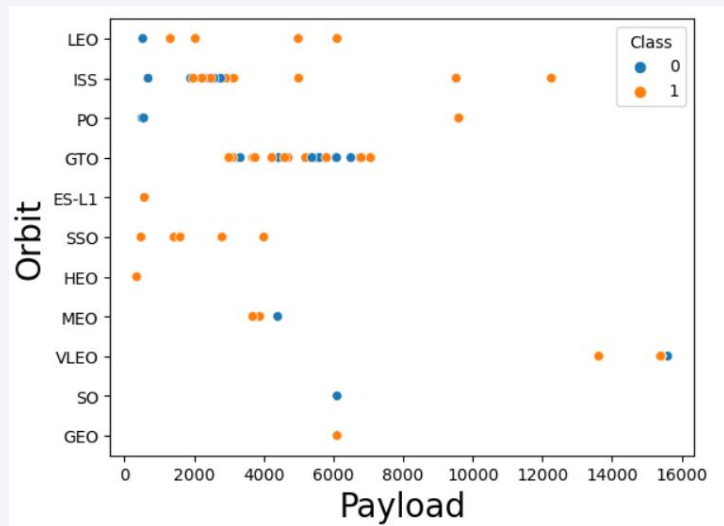


# EDA with Data Visualization

<https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb>

## 🚀 Data Visualization conducted

- 🚀 Scatter plot relationships between: Flight Number & Launch Site, Launch Site & Payload, Flight Number & Orbit Type, Payload & Orbit Type (only one example was provided below)
- 🚀 Line chart to visualize the yearly trend of the launch success rate
- 🚀 Bar chart to visualize the relationship between the success rate of each orbit type



# EDA with SQL

[https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/jupyter-labs-eda-sql-edx\\_sqlite.ipynb](https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/jupyter-labs-eda-sql-edx_sqlite.ipynb)

---

🚀 **SpaceX dataset was loaded into a SQL database**

🚀 **Applied SQL queries to get insights from the dataset**

- 🚀 Display the names of the unique launch sites in the space mission,
- 🚀 Display the total payload mass carried by boosters launched by NASA (CRS),
- 🚀 Display average payload mass carried by booster version F9 v1.1,
- 🚀 Total number of successful and failure mission outcomes,
- 🚀 Names of the booster versions which have carried the maximum payload mass,
- 🚀 Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)).

# Build an Interactive Map with Folium

[https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/lab\\_jupyter\\_launch\\_site\\_location.jupyterlite.ipynb](https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/lab_jupyter_launch_site_location.jupyterlite.ipynb)

---

## 🚀 Components added to the Folium Map

- 🚀 Marked launch sites and added map objects such as highlighted circles, color labeled markers and lines to have a visual impact of the success or failure of the launches in each site.
- 🚀 Assigned the feature launch outcomes to our class (successful = 1, failed = 0)
- 🚀 Calculated distances between launch sites and its proximities (railways, highways, coastlines), in order to answer the following questions:
  - 🚀 Are launch sites in close proximity to railways?
  - 🚀 Are launch sites in close proximity to highways?
  - 🚀 Are launch sites in close proximity to coastline?
  - 🚀 Do launch sites keep certain distance away from cities?



# Build a Dashboard with Plotly Dash

[https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/spacex\\_dash\\_app.py](https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/spacex_dash_app.py)

---

## 🚀 The following components were added to the interactive plotly dashboard:

### 🚀 Functionalities

- 🚀 Added a dropdown list to enable Launch Site selection
- 🚀 Add a slider to select payload range

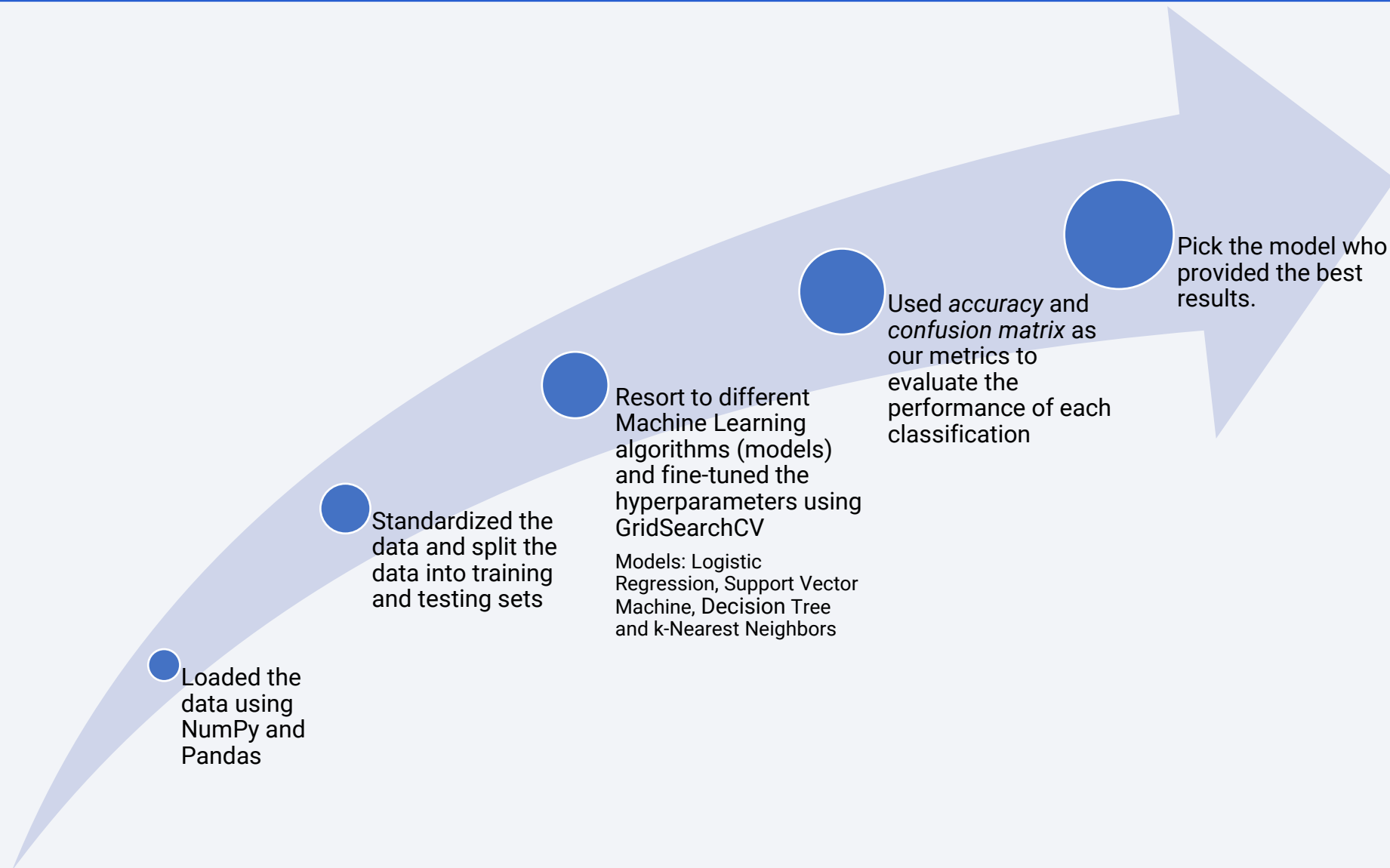
### 🚀 Plots

- 🚀 Pie charts to show the total successful launches count for *all sites or selected site* on the dropdown list
- 🚀 Scatter charts to show the correlation between payload mass and launch success

# Predictive Analysis (Classification)

[https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/SpaceX\\_Machine\\_Learning\\_Prediction\\_Part\\_5.jupyterlite.ipynb](https://github.com/0xfortes/IBM-Data-Science/blob/main/Capstone%20SpaceX/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb)

---



# Results

---

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



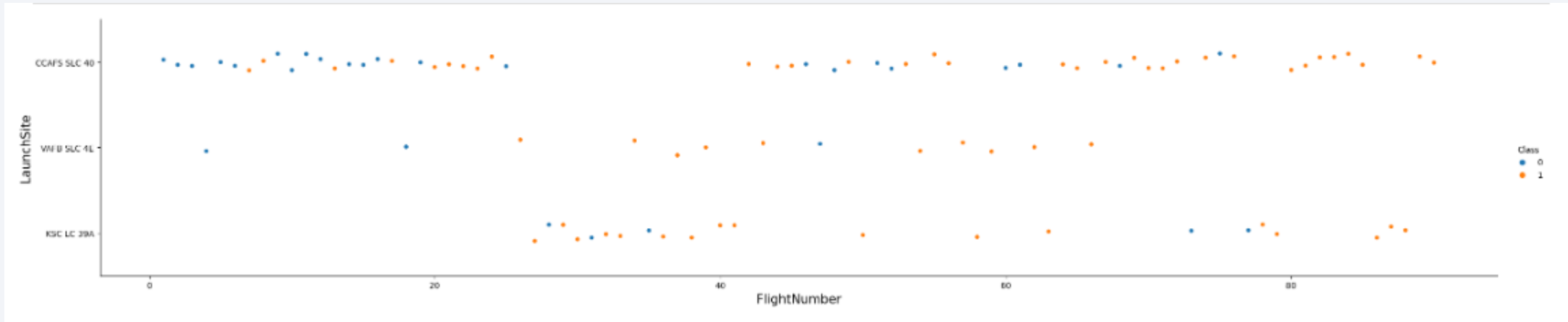
The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of red and cyan. A faint, light blue grid pattern is also visible, particularly in the lower half of the image. The overall effect is dynamic and technological.

Section 2

# Insights drawn from EDA



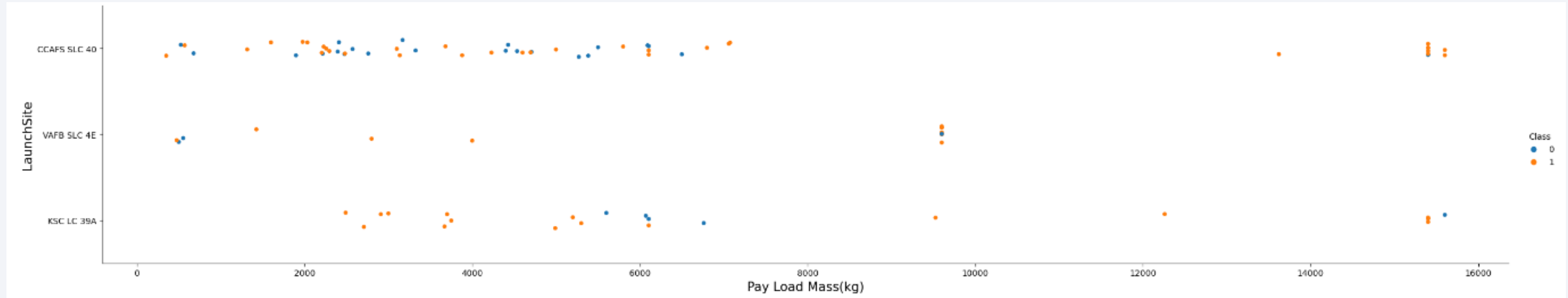
# Flight Number vs. Launch Site



A launch site with a higher number of flights tends to have a higher success rate



# Payload vs. Launch Site

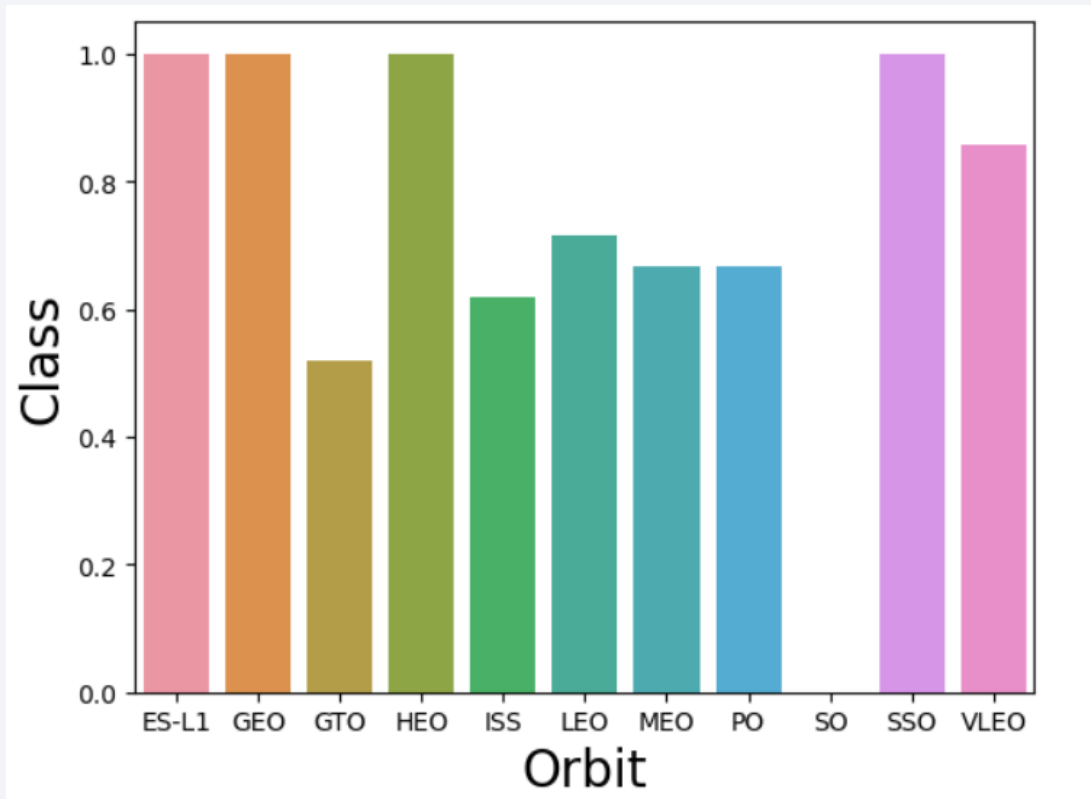


- **CCAFS SLC 40**
  - The heavier the payload mass, higher the success rate for the rocket launch
- **VAFB SLC 4E**
  - There's no success rate launching from this site using payloads heavier than 10000 Kg
- **KSC LC 35A**
  - Success rate is higher when using payloads with a mass between 2000 and 4000 Kg



# Success Rate vs. Orbit Type

---

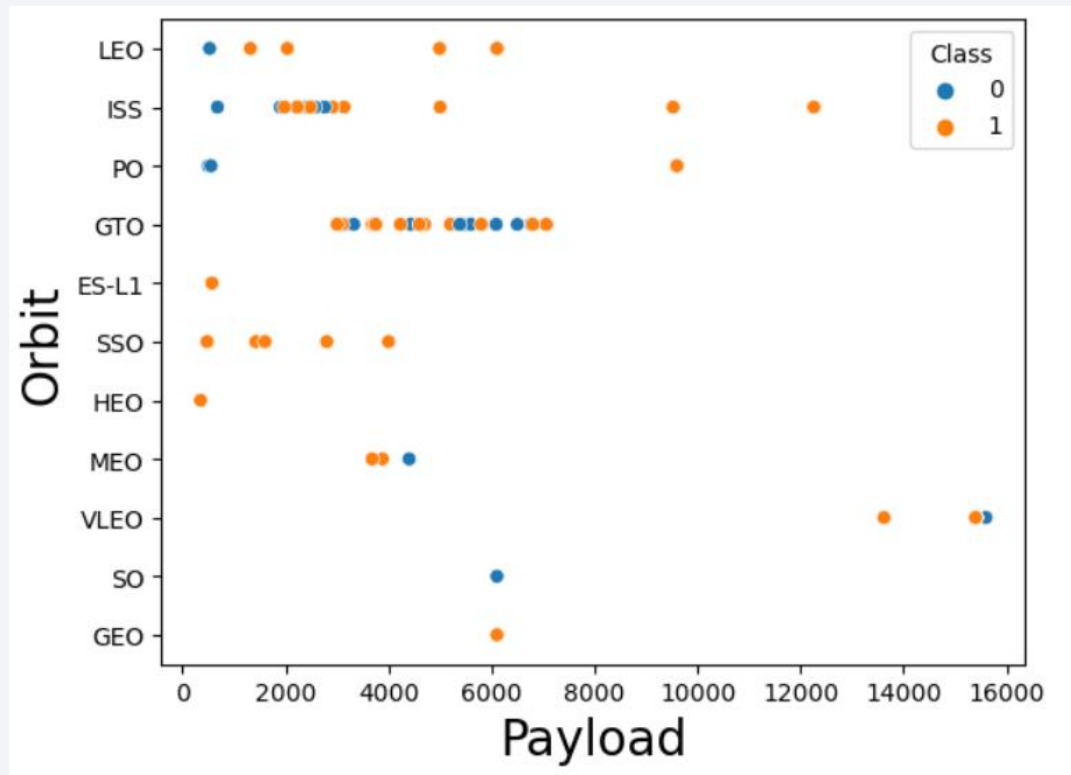


**Orbits with the higher success rate**

🚀 ES-L1, GEO, HEO, SSO, VLEO



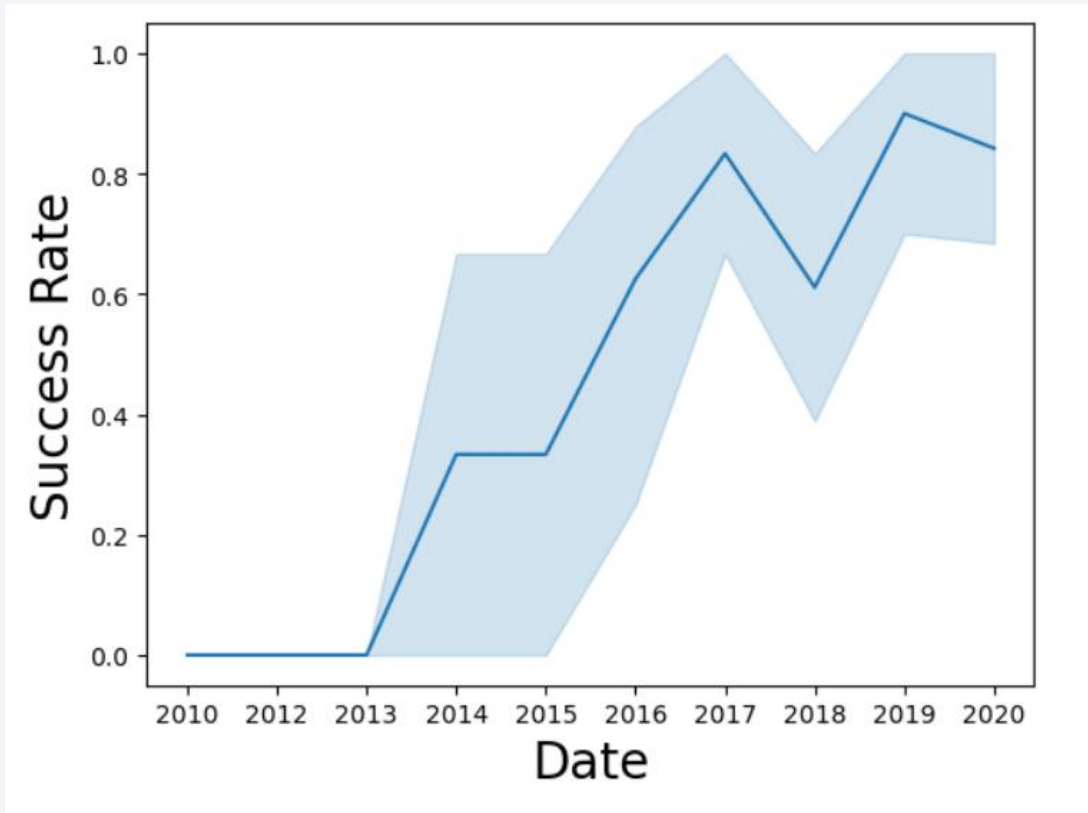
# Payload vs. Orbit Type



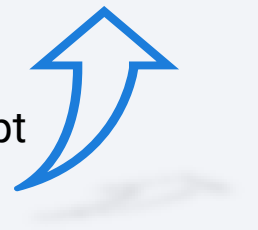
With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS. However, for GTO we cannot distinguish this well as both positive landing rate and negative landing are both present.

# Launch Success Yearly Trend

---



The success rate since 2013 kept increasing till 2020



# All Launch Site Names

---

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

```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL
```

The **DISTINCT** keyword in the SQL statement returns ONLY the launch sites from the SpaceX dataset

# Launch Site Names Begin with 'KSC'

---

We limit our query to search and retrieve only 5 records from the data set, where the launch site name begins with “KSC”

```
%sql SELECT * FROM SPACEXTBL WHERE (LAUNCH_SITE) LIKE 'KSC%' LIMIT 5;
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490.0	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600.0	GTO	EchoStar	Success	No attempt
2017	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300.0	GTO	SES	Success	Success (drone ship)
2017	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300.0	LEO	NRO	Success	Success (ground pad)
2017	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070.0	GTO	Inmarsat	Success	No attempt



# Total Payload Mass

---

Total payload mass (kg) carried by boosters launched by NASA (CRS)

```
%sql SELECT sum(PAYLOAD_MASS_KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'
```

sum(PAYLOAD_MASS_KG_)
45596.0

# Average Payload Mass by F9 v1.1

---

Average payload mass (kg) carried by booster version F9 v1.1

```
%sql SELECT avg(PAYLOAD_MASS_KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'
```

avg(PAYLOAD_MASS_KG_)
2928.4

# First Successful Ground Landing Date

---

The first successful landing outcome on drone ship was on 04/08/2016

```
%sql SELECT min(date) FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)'
```

<b>min(date)</b>
04/08/2016

Retrieve the earliest date (min(date)) from the SpaceX dataset when the first successful landing outcome in a drone ship was achieved.

# Successful Drone Ship Landing with Payload between 4000 and 6000

---

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS_KG_ BETWEEN 4000 AND 6000
```

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Name of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 (**BETWEEN** clause allow us to select intervals of data)

# Total Number of Successful and Failure Mission Outcomes

---

```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS COUNT FROM SPACEXTBL GROUP BY MISSION_OUTCOME
```

Mission_Outcome	COUNT
None	0
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

# Boosters Carried Maximum Payload

---

Names of the booster which have carried the maximum payload mass from the SpaceX dataset

```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS_KG_ = (SELECT MAX(PAYLOAD_MASS_KG_) FROM SPACEXTBL)
```

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

Firstly, we filter the dataset to include only the data related to the booster version. Afterward, we utilize a query to retrieve the maximum value from the "payload\_mass\_kg" column.



# 2017 Launch Records

---

```
%sql SELECT substr(Date, 4, 2) as 'Month', substr(Date, 7, 4) as 'Year', LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE Year='2017' AND LANDING_OUTCOME = 'Success (ground pad)'
```

Month	Year	Landing_Outcome	Booster_Version	Launch_Site
02	2017	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
01	2017	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
03	2017	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
08	2017	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
07	2017	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
12	2017	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

Records which will display the month names, successful landing outcomes in ground pad ,booster versions, launch site for the months in year 2017

Since SQLite does not support month names. We need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2017' for year.

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

---

```
%sql SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS COUNT FROM SPACEXTBL GROUP BY LANDING_OUTCOME HAVING DATE BETWEEN '04-06-2010' AND '20-03-2017' ORDER BY '%DESC'
```

Landing_Outcome	COUNT
Controlled (ocean)	5
Failure	3
Failure (parachute)	2
No attempt	1
Success (drone ship)	14

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

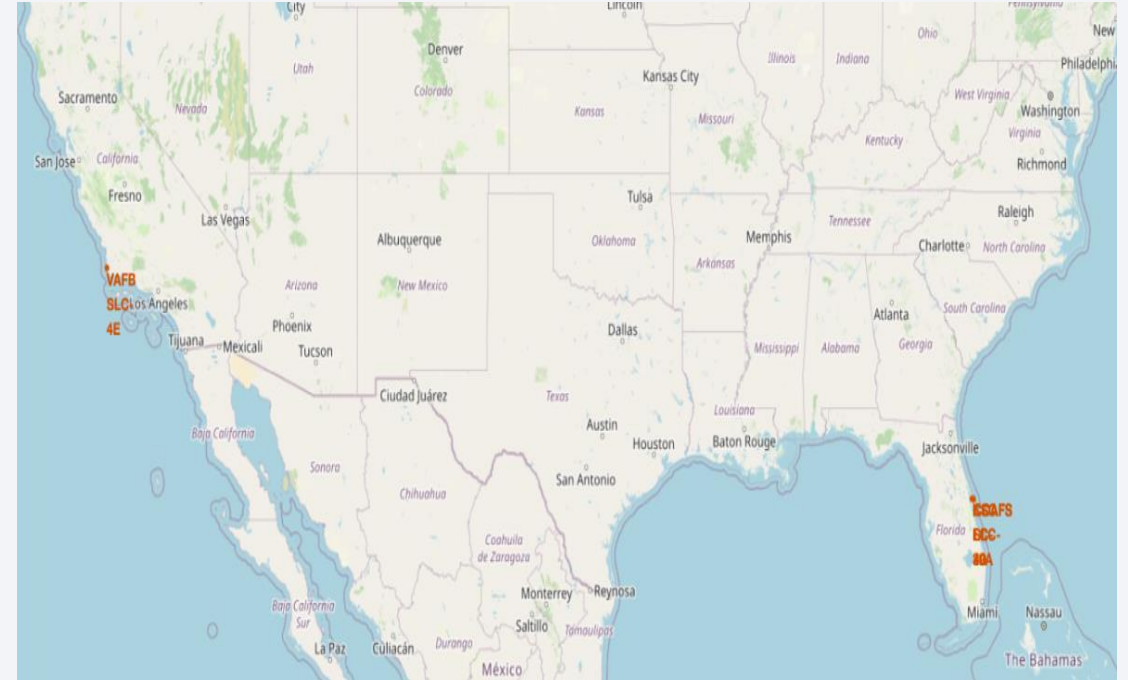
A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

# Launch Sites Proximities Analysis

# SpaceX marked launch locations

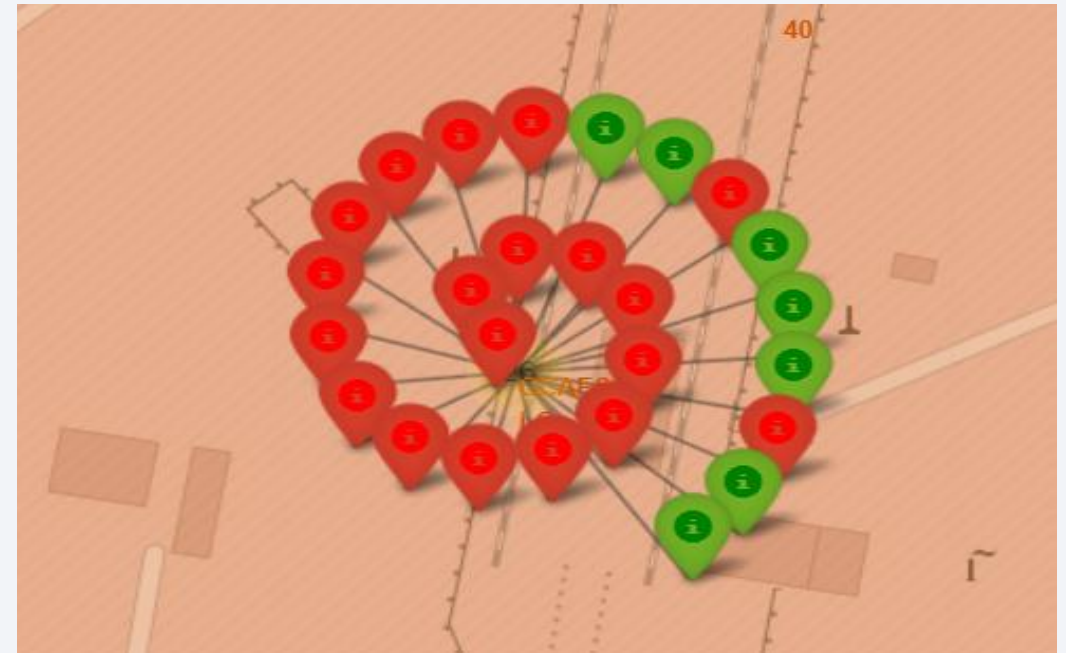
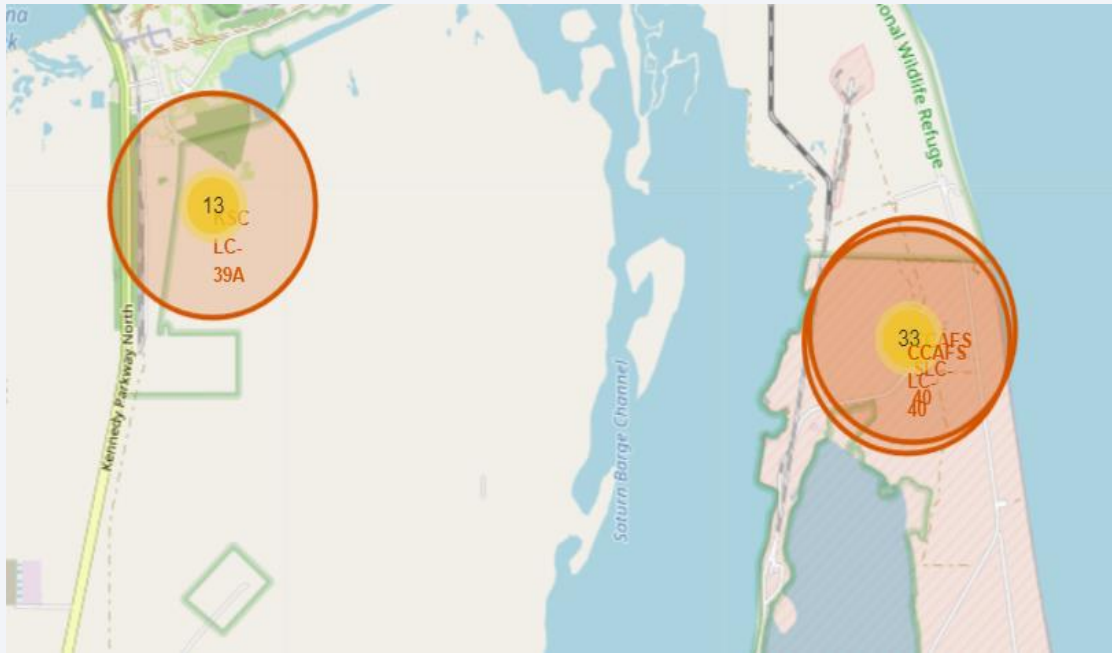
---



We have incorporated both a marker (*folium.Marker*) and a circle (*folium.Circle*) to enable easy identification of each location on the world map.

# Launch Outcome Markers by location

---



We created Marker clusters because it can be a good way to simplify a map containing many markers having the same coordinate.

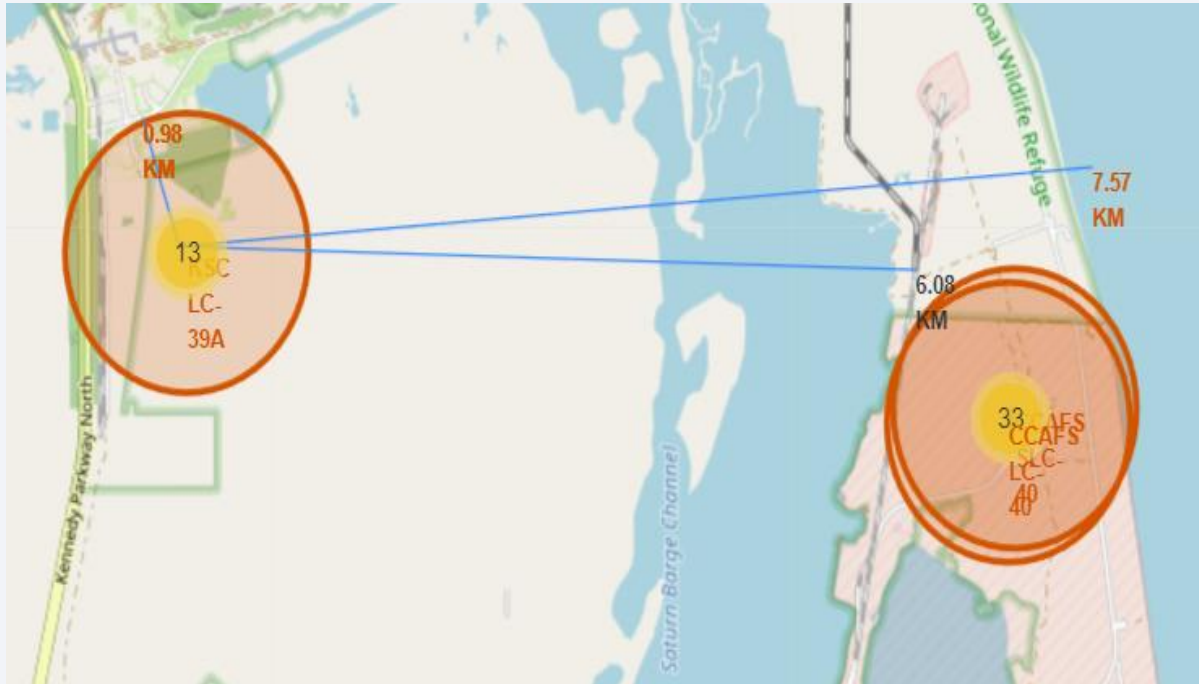
We also added color markers in order to identify the success/fail of the launches in a specific site

(Green = Successful, Red = Failed)



# Launch Site Proximities

---



1. Searched for the closest railway, highway and coastline,
2. Marked down their coordinates,
3. Draw a line (*PolyLine*) from each of the closest point of interest to the launch site, using the coordinates,
4. Calculated the distance of each point to the launch site.

These steps will allow us to answer the following questions:

- Are launch sites in close proximity to railways?
- Are launch sites in close proximity to highways?
- Are launch sites in close proximity to coastline?
- Do launch sites keep certain distance away from cities?



Section 4

# Build a Dashboard with Plotly Dash

# The success rate of SpaceX launches at each site

---

Success Count for all launch sites



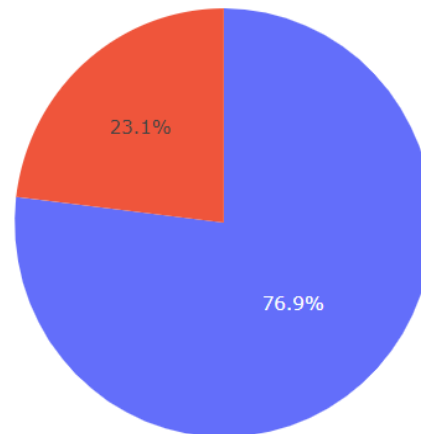
KSC LC-39A had the highest number of successful launches among all the sites



# Total Launches for KSC LC-39A

---

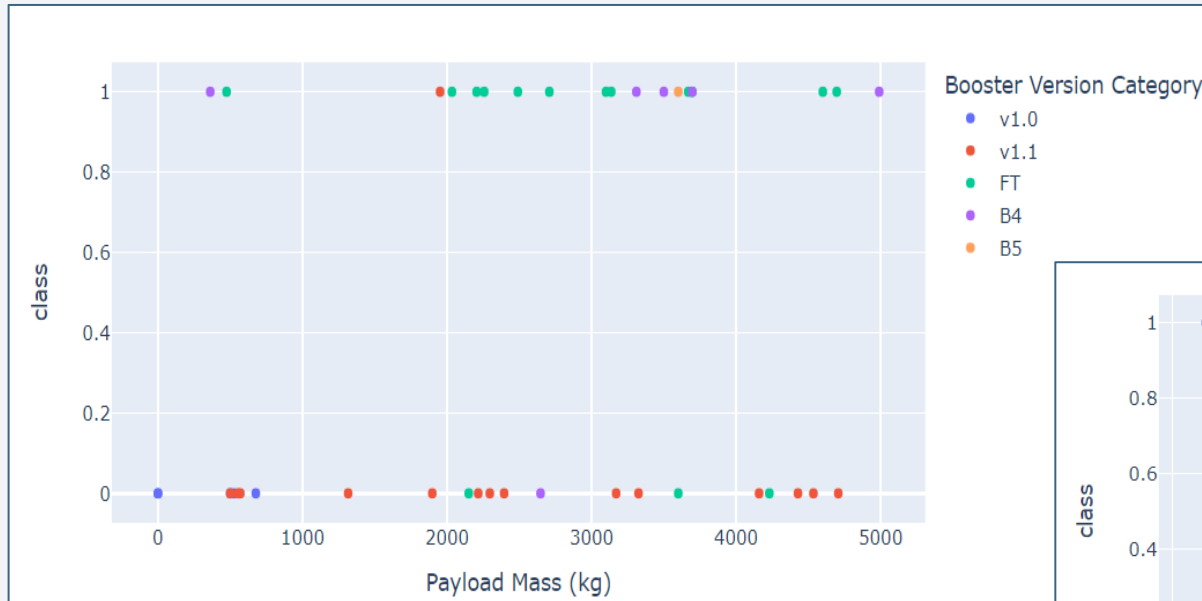
Total Success Launches for site KSC LC-39A



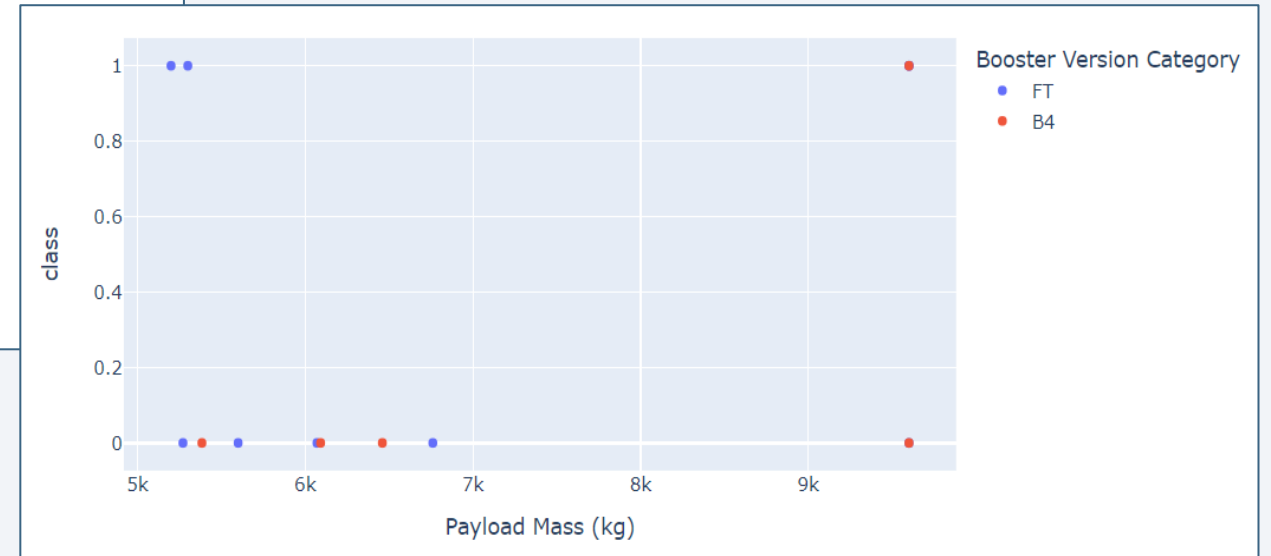
**The success rate for KSC LC-39A site is 76,9%**

■ 1  
■ 0

# Correlation between payload and launch success



*Payload mass 0 – 5000 Kg*



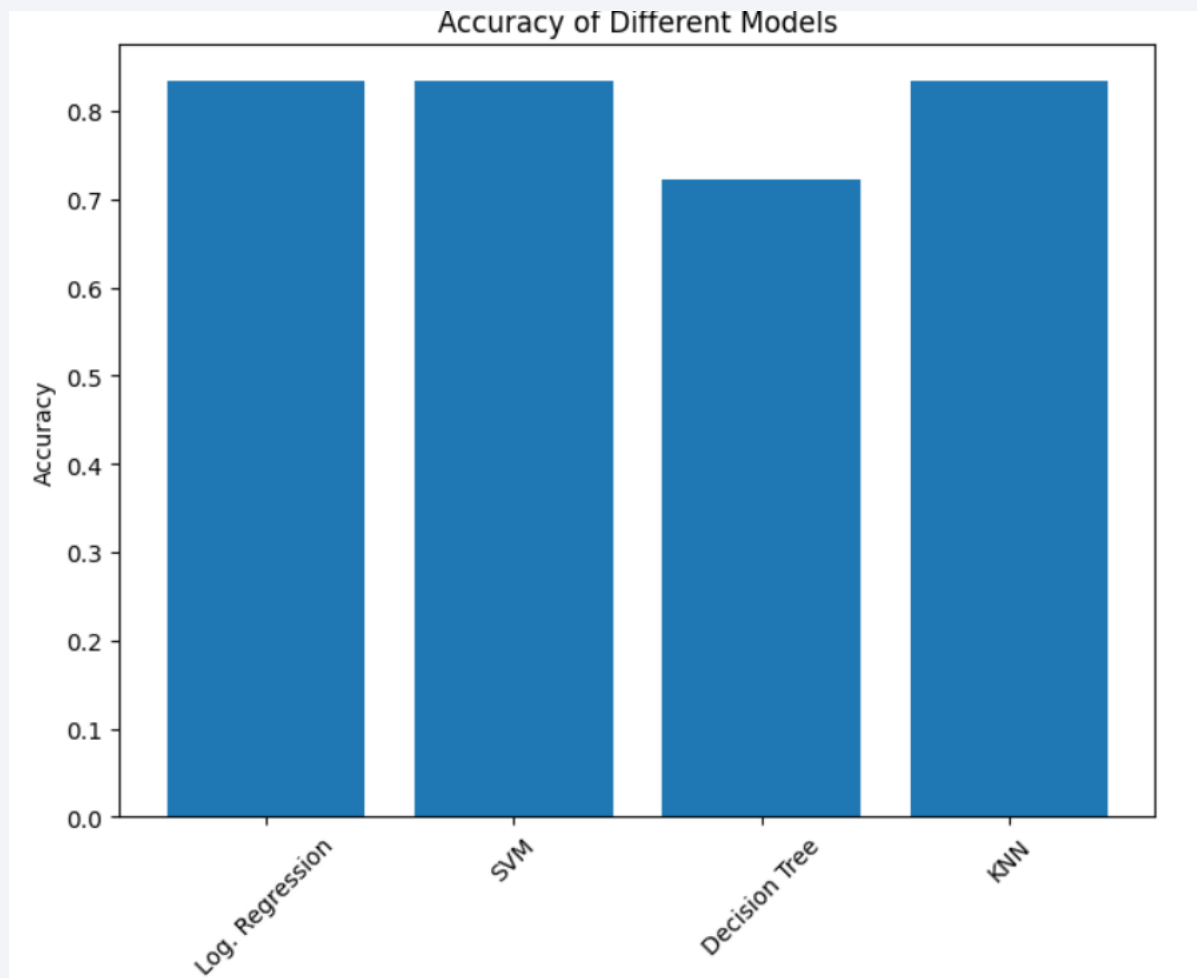
*Payload mass 5000 – 10000 Kg*

For lower weight payloads, the success rate is significantly higher compared to higher weight payloads

Section 5

# Predictive Analysis (Classification)

# Classification Accuracy



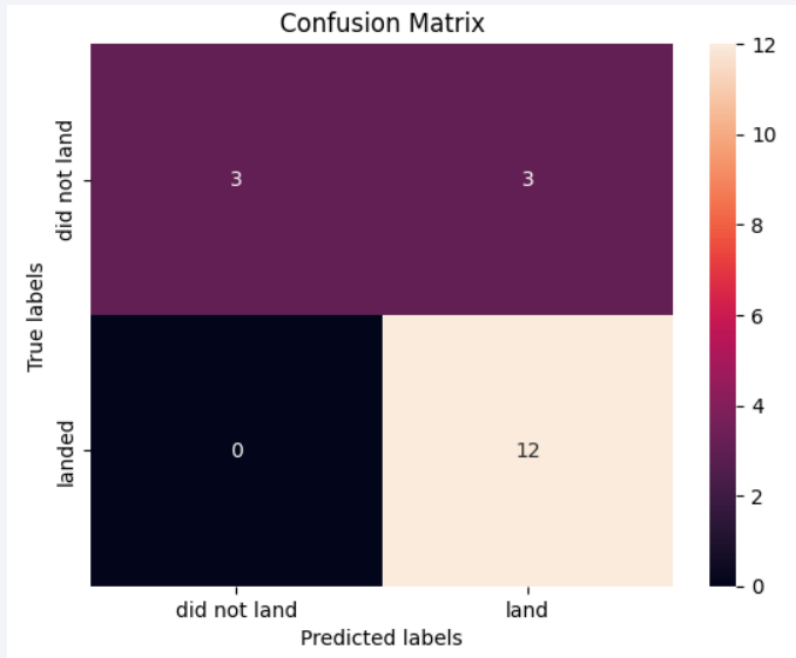
```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print('Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K neardsdt neighbors method:', knn_cv.score(X_test, Y_test))
```

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.7222222222222222
Accuracy for K neardsdt neighbors method: 0.8333333333333334
```

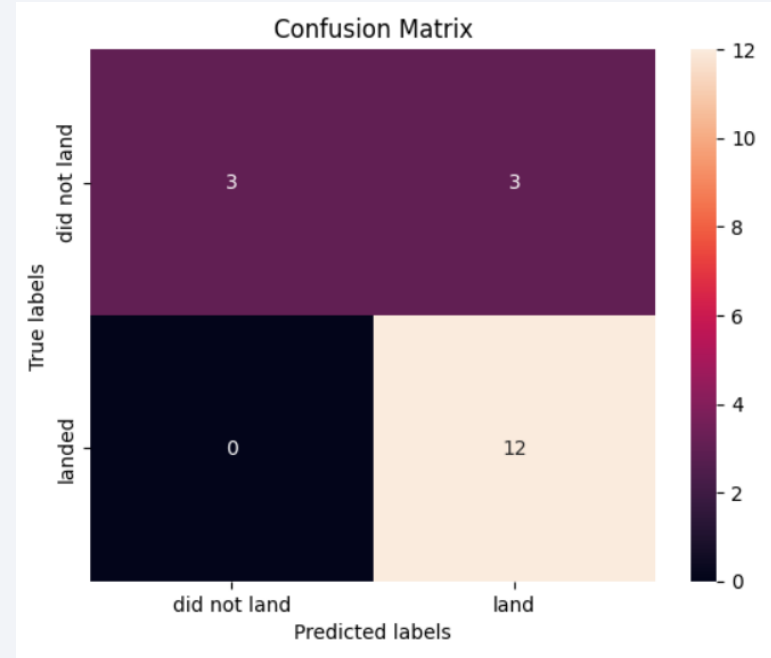
**Best performing models with similar accuracy of 83,3%**

- Logistic Regression
- SVM
- KNN

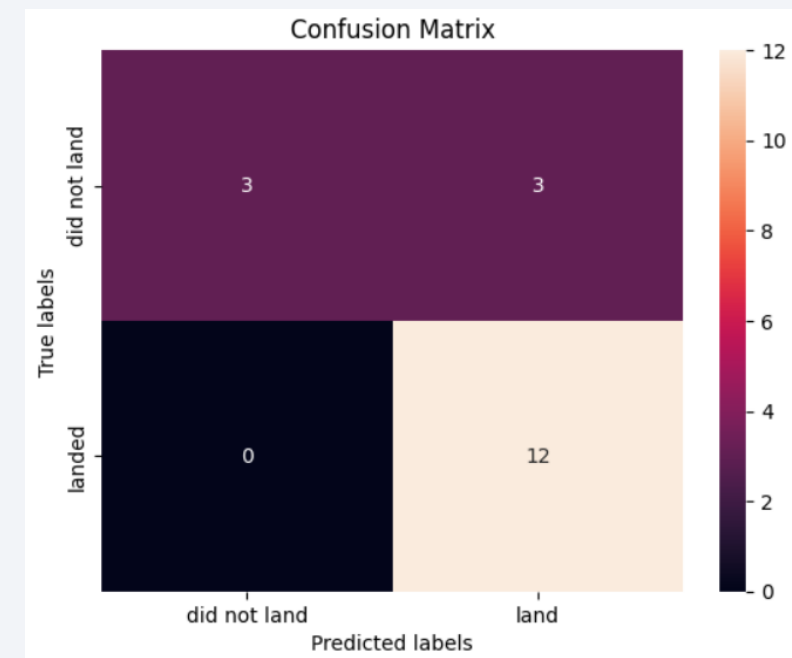
# Confusion Matrix



Logistic Regression



SVM



KNN

We achieved a high accuracy with 15 out of 18 observations correctly classified (83.33%). The model displayed robust performance in correctly identifying successful rocket landings (True Positive = 12) while making a few misclassifications for unsuccessful landings (False Positive = 3).

# Conclusions

---

- The success rates for SpaceX launches are proportional to the number of years they have been conducting launches.
- Orbits, such as ES-L1, GEO, HEO, SSO, and VLEO, exhibit higher success rates in SpaceX launches.
- KSC LC 39A stands out as the site with the highest number of successful launches
- Lighter payloads tend to have a higher success rate, leading to more successful launches
- Logistic regression, SVM, and KNN machine learning algorithms demonstrate to have the best accuracy when applied to the SpaceX dataset.



Thank you!

