# Space X to the MOON!

By Richard Lee

September 23, 2022

# OUTLINE

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

# EXECUTIVE SUMMARY

## The Methodologies

- Data Collection through API
- Data Collection with Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analytics with Folium
- Machine Learning Prediction

## The Results

- Exploratory Data Analysis result
- Interactive analytics in screenshots
- Predictive Analytics result from Machine Learning Lab

# INTRODUCTION

**SpaceX** is an American spacecraft manufacturer, space launch provider, and a satellite communications corporation. It was founded in 2002 by Elon Musk, with the goal of reducing space transportation costs to enable the colonization of Mars. This project shows all the data I've gathered regarding SpaceX.

# Part 1

## Methodology

# Methodology

## Executive Summary

- Data collection
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

# Data Collection

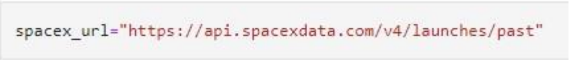
- Data collection is the process of gathering data for use in business decision-making, strategic planning, research and other purposes. It's a crucial part of data analytics applications and research projects: Effective data collection provides the information that's needed to answer questions, analyze business performance or other outcomes, and predict future trends, actions and scenarios.

# Data Collection – SpaceX API

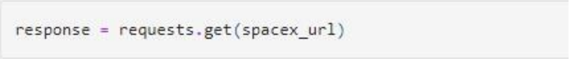
I got the request for the rocket launch data using API, then the JSON result to the data frame, and finally performed data cleaning.

From:

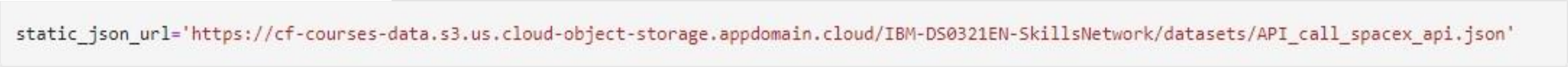
https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/1_Data_Collection_with_API.ipynb



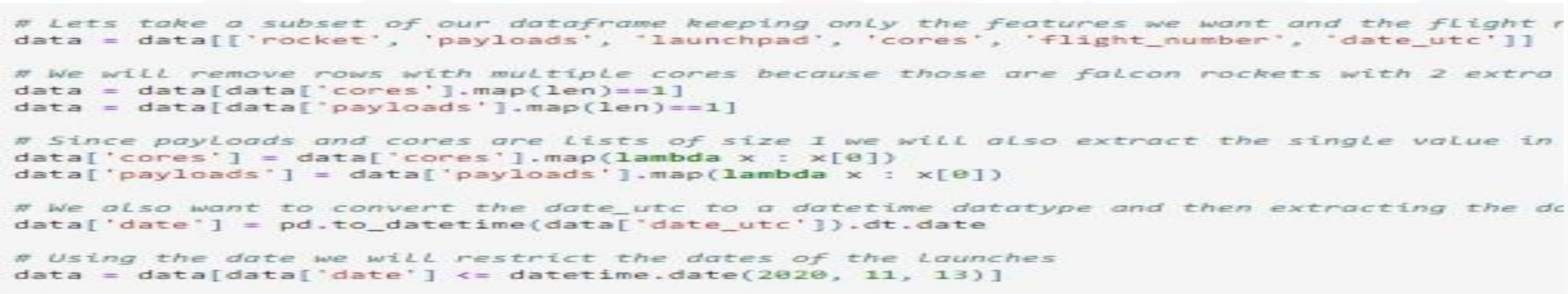
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```



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



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



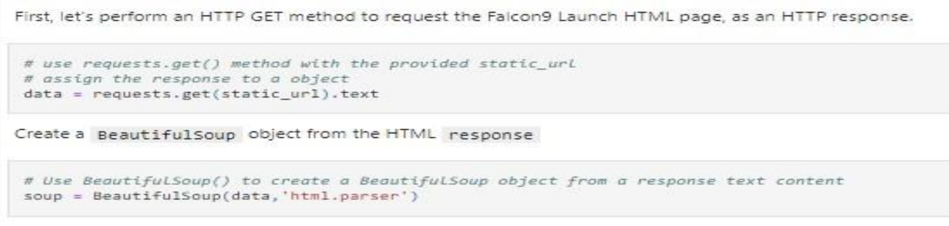
```
# Lets take a subset of our dataframe keeping only the features we want and the flight r
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the da
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

# Data Collection - Scraping

I requested the Falcon9 launch wiki page url, then used BeautifulSoup and finally extracted all the info.

From:

https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/2_Data_Collection_with_Web_Scraping.ipynb

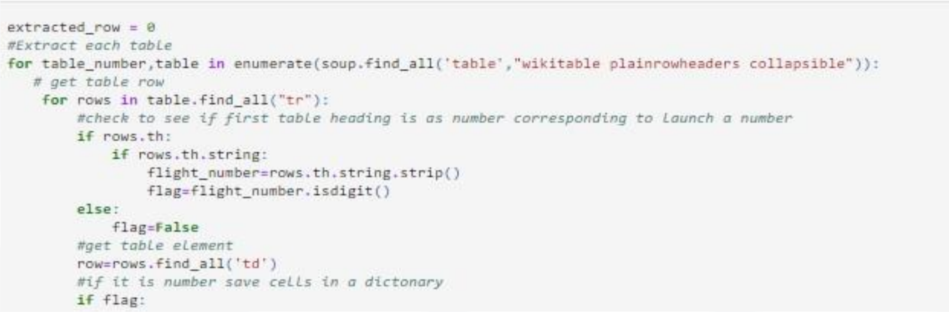


First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# 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')
```

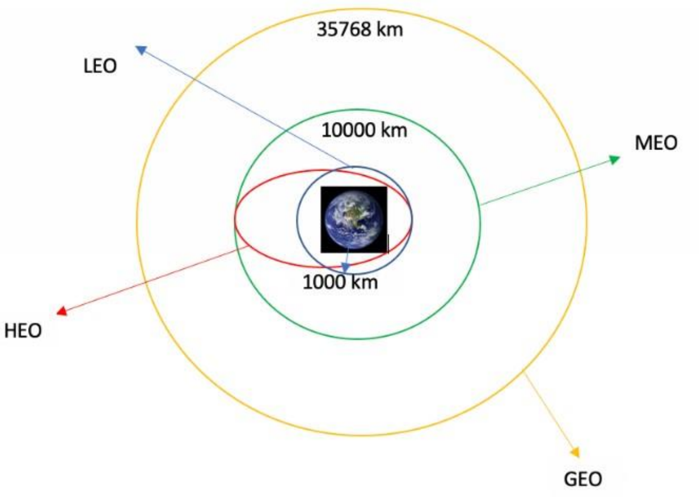


```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
```

# Data Wrangling

From:

https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/3_Data_Wrangling.ipynb



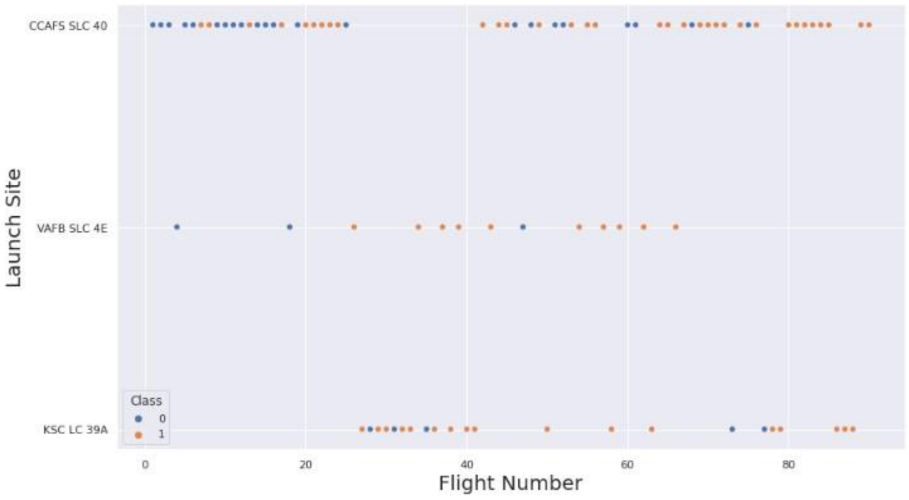
Data wrangling refers to a variety of processes designed to transform raw data into more readily used formats. The exact methods differ from project to project depending on the data you’re leveraging and the goal you’re trying to achieve.

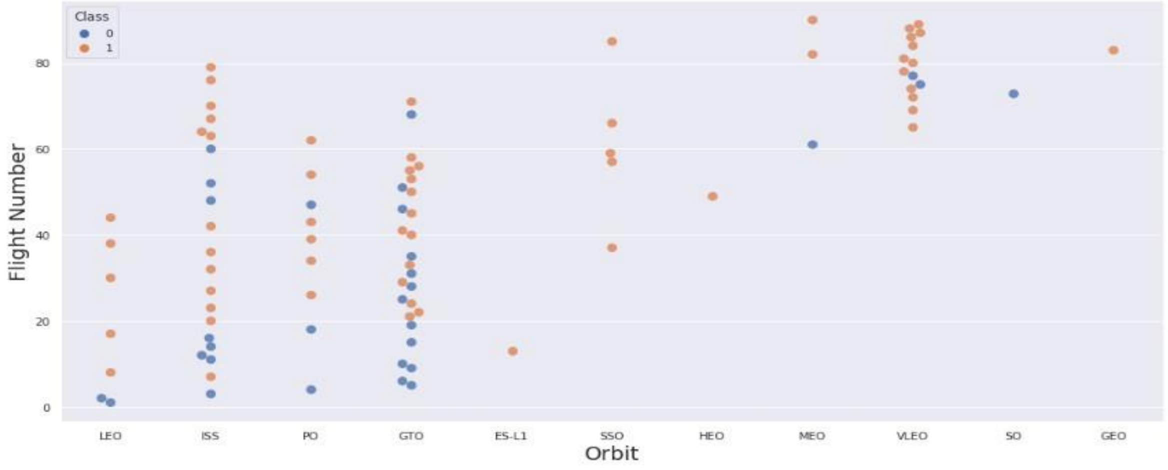
# EDA with Data Visualization

I started by using scatter graph to find the relationship between, Payload and Flight Number, Flight Number and Launch Site, Payload and Launch Site, Flight Number and Orbit Type, and Payload and Orbit Type.

From:

https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/5_Exploratory_Data_Analysis_with_Visualisation_Lab.ipynb



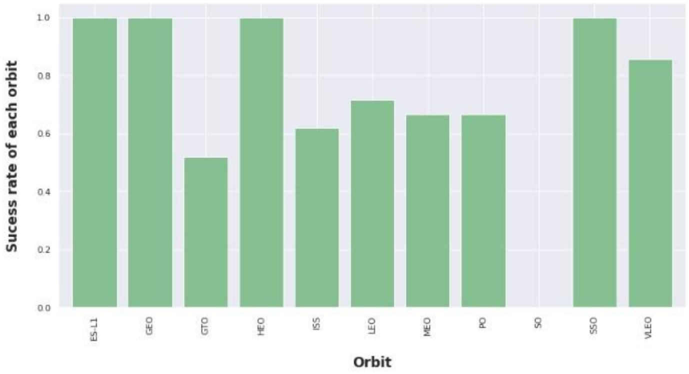


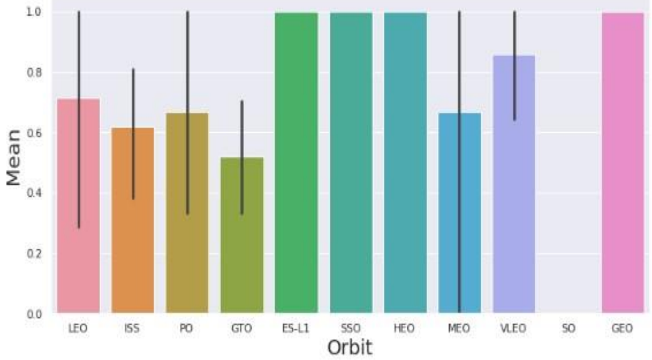
# EDA with Data Visualization

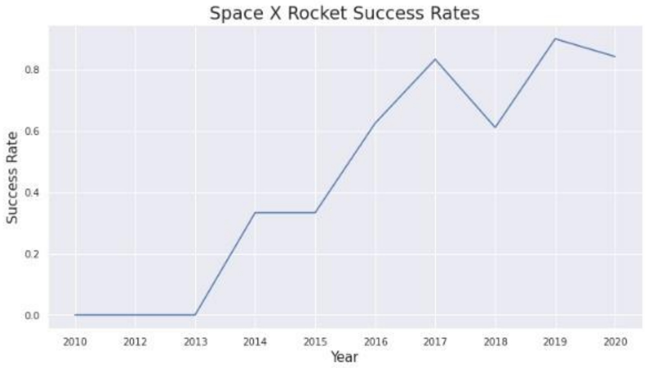
After using scatter plots, I used different visualization tools such like bar graphs and line plots graphs. I used the bar graph to determine which orbits have the highest probability of success. I then used the line graph to show a trend or pattern of the attribute over time .

From:

https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/3_Data_Wrangling.ipynb





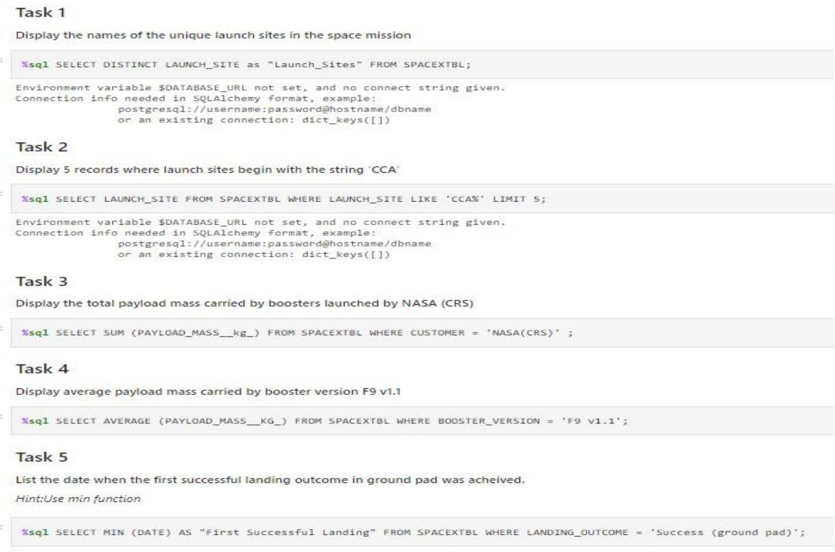


# EDA with SQL

From:

https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/4_Exploratory_Data_Analysis_with_SQL.ipynb

I was able to retrieve useful information using SQL demonstrated with the pictures below:



### Task 1

Display the names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

```
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
```

### Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
%sql SELECT LAUNCH_SITE FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
```

### Task 3

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

```
%sql SELECT SUM (PAYLOAD_MASS__kg_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA(CRS)' ;
```

### Task 4

Display average payload mass carried by booster version F9 v1.1

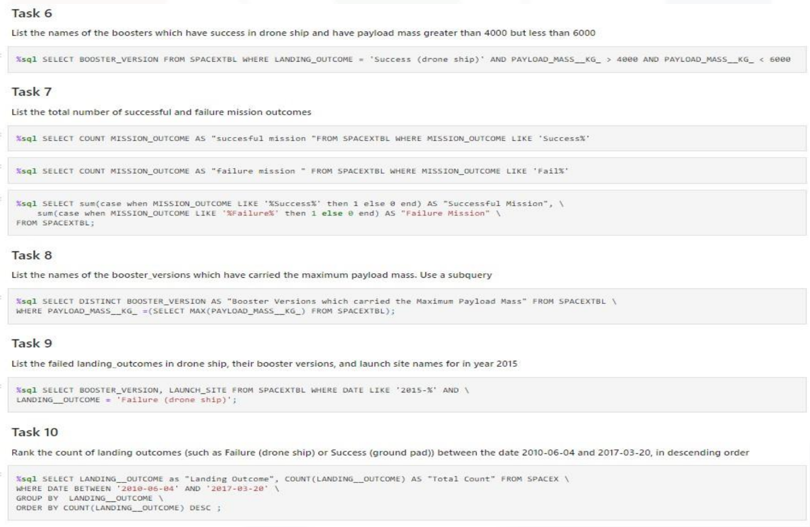
```
%sql SELECT AVERAGE (PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

### Task 5

List the date when the first successful landing outcome in ground pad was acheived.

*Hint:Use min function*

```
%sql SELECT MIN (DATE) AS "First Successful Landing" FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```



### Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

### Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT MISSION_OUTCOME AS "succesful mission "FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%'
```

```
%sql SELECT COUNT MISSION_OUTCOME AS "failure mission " FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Fail%'
```

```
%sql SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful Mission", \
    sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission" \
FROM SPACEXTBL;
```

### Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

### Task 9

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

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

### Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

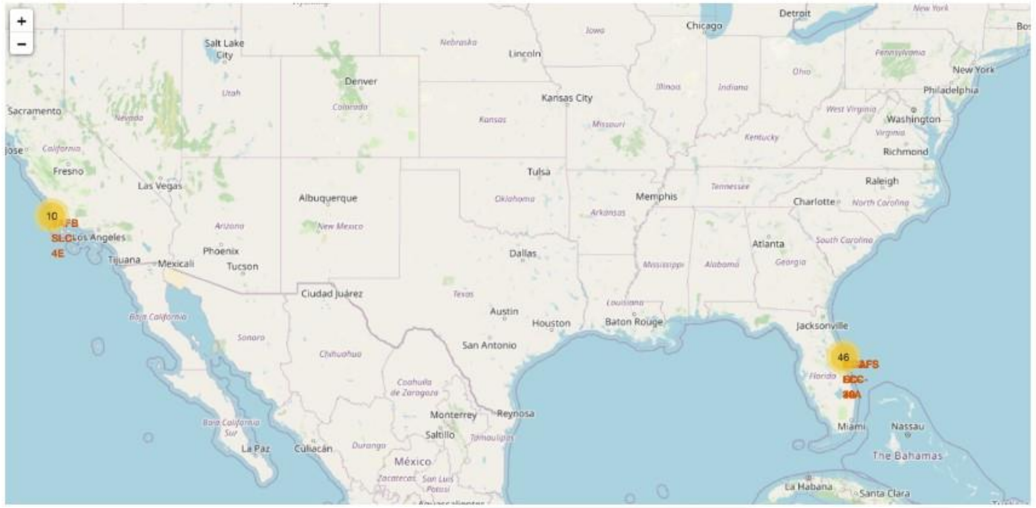
```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

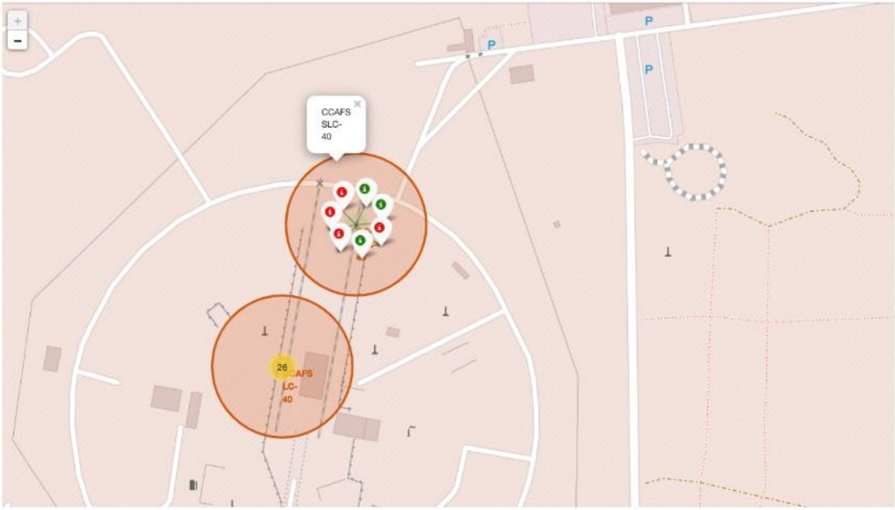
# Build an Interactive Map with Folium

To visualize the launch data into an interactive map I took the latitude and longitude coordinates at each launch site and added a circle marker around each launch site with a label of the name of the launch site. Below are some samples of the of the maps.

From:

https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/6_Interactive_Visual_Analytics_with_Folium.ipynb





# Build a Dashboard with Plotly Dash

I built an interactive dashboard with Plotly dash. I plotted pie charts showing the total launches by a certain sites and then plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg). The following code can be found at:

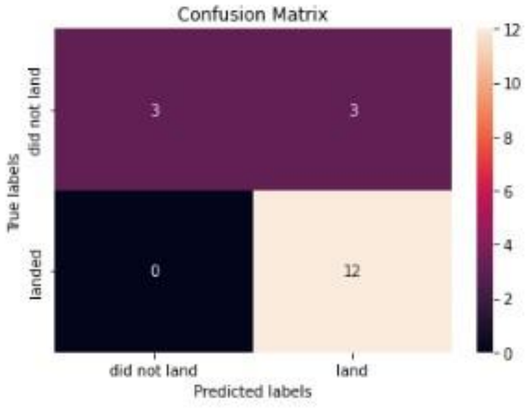
https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/7_plotly_dash_app.py

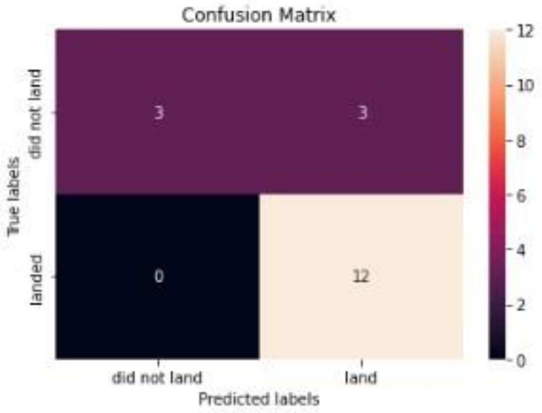
# Predictive Analysis

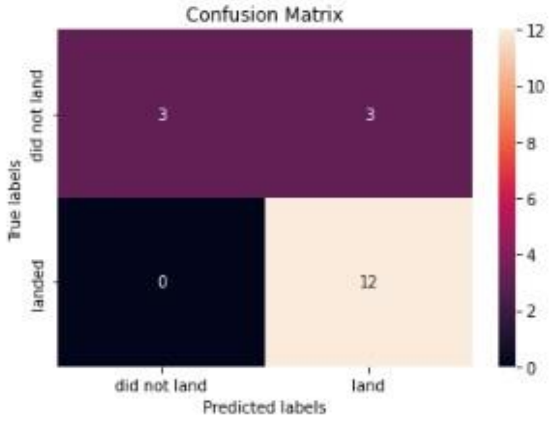
Building the Model, Evaluating the Model, Improving the Model and Find the Best Model. Below are matrixes built from gathering the data.

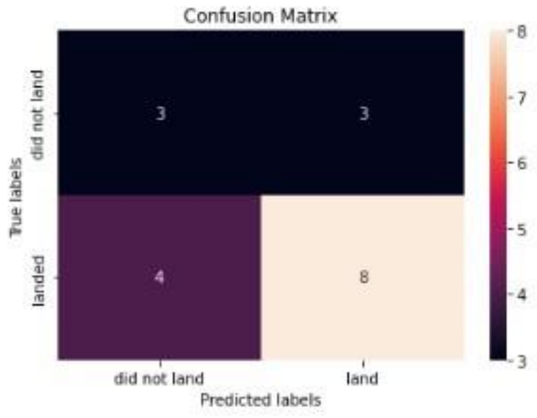
From:

https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/8_notebook_Predictive_Analysis_-_Machine_Learning_Lab.ipynb









## Part 2

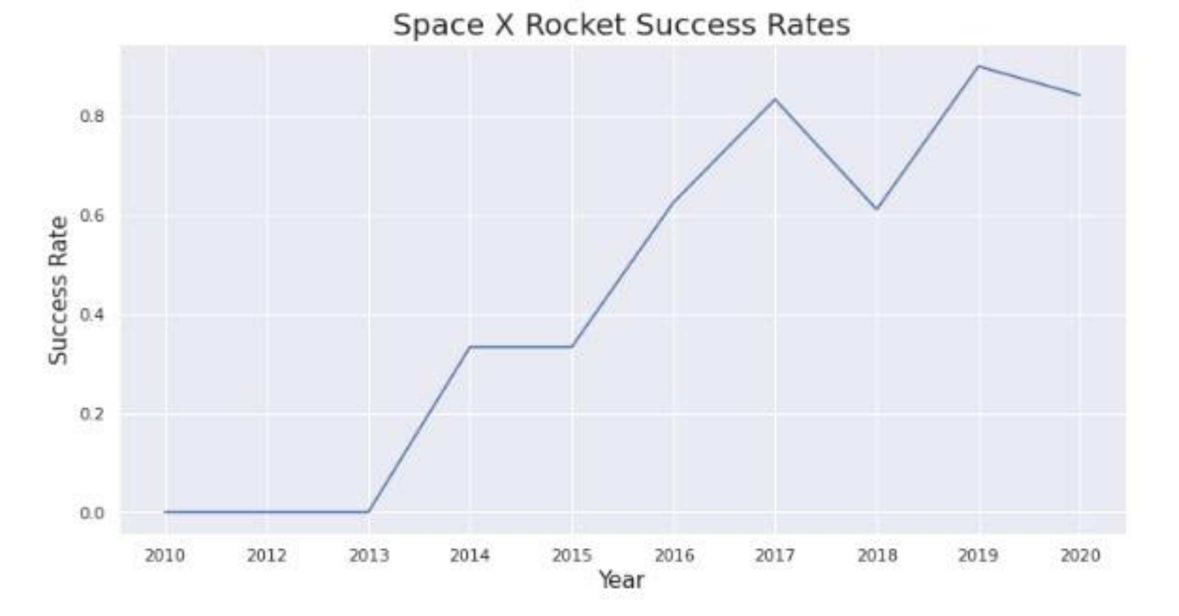
# Insights drawn from EDA

# Rocket Success Rate

The graph below shows the success rate increasing from 2013 till 2020. The success rate

From:

https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/5_Exploratory_Data_Analysis_with_Visualisation_Lab.ipynb



# Part 3

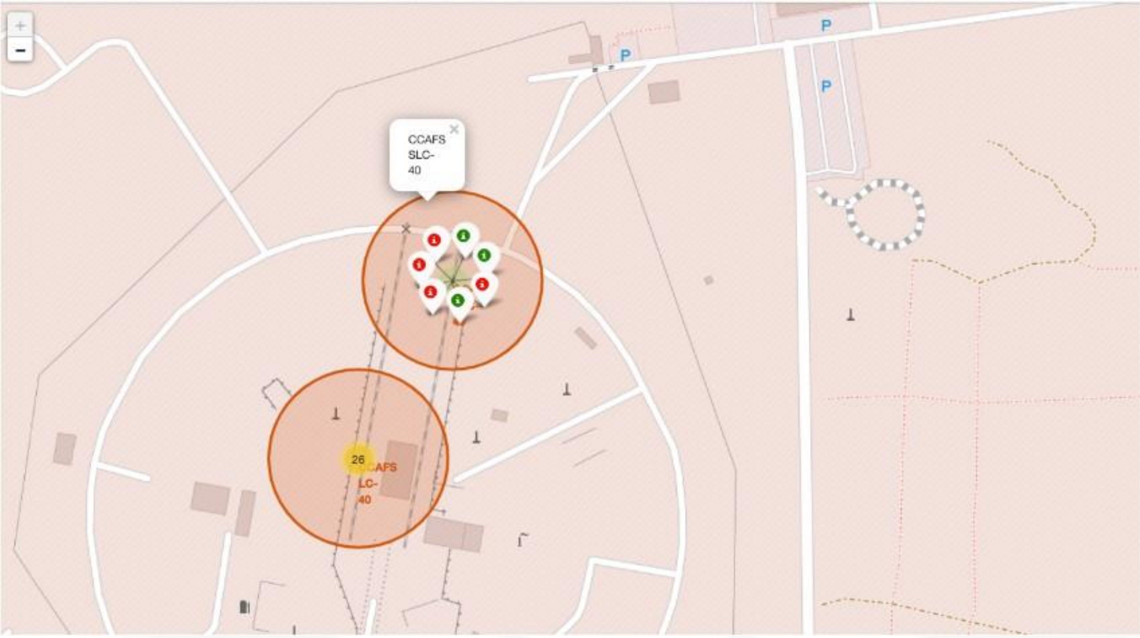
## Launch Sites

# Markers showing launch sites

Below is a picture of a launch site with markers.

From:

https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/6_Interactive_Visual_Analytics_with_Folium.ipynb



# Part 4

plotly

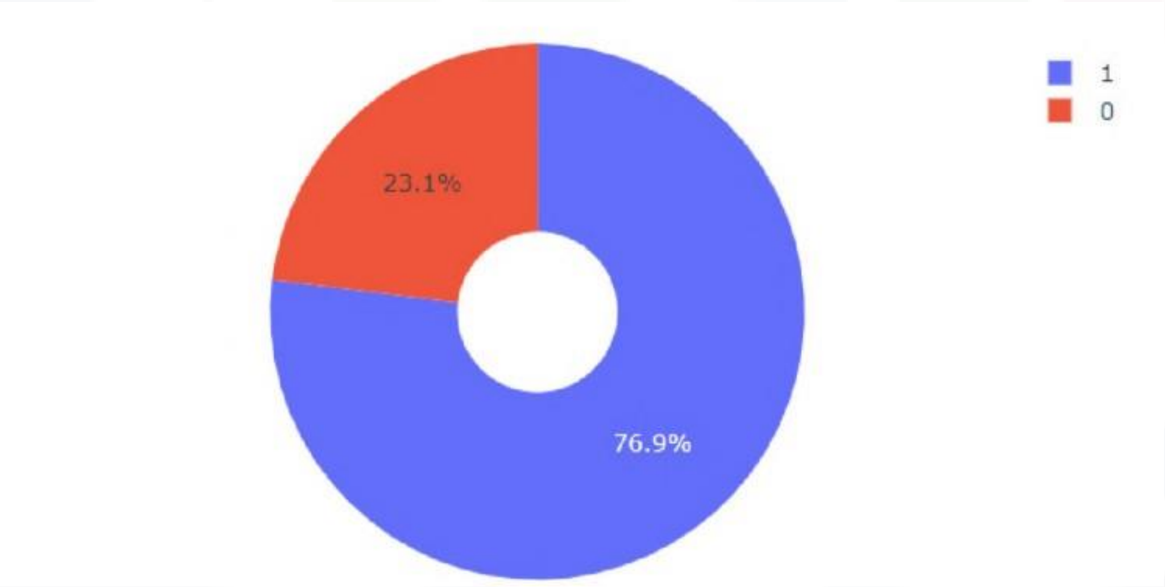
## Plotly Dash

# The highest launch-success ratio

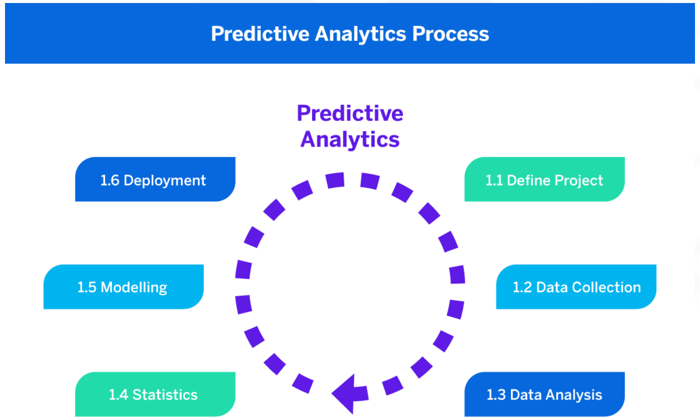
KSC – LC-39A got a 76.9% success rate.

From:

https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/7_plotly_dash_app.py



# Part 5



Predictive Analytics Process

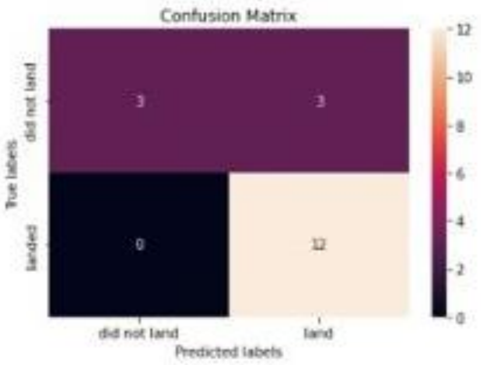
# Predictive Analysis

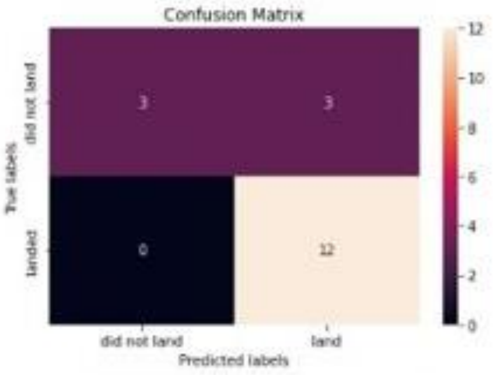
# Confusion Matrix

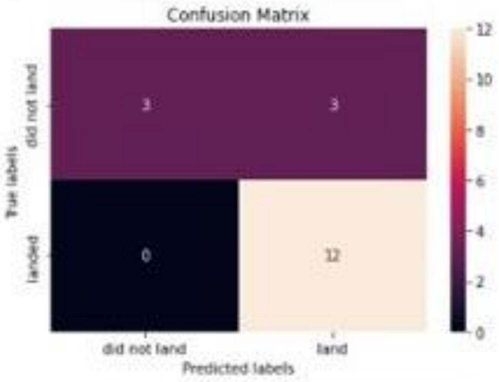
A **confusion matrix** is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one. Below are sample pictures created from the lab.

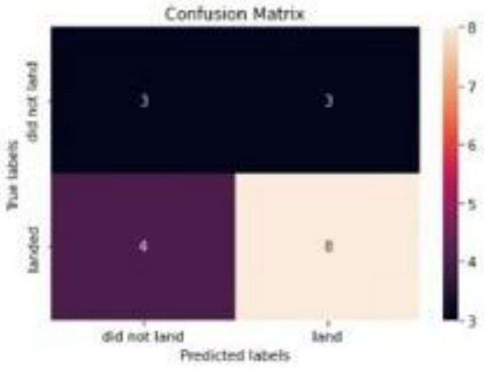
From:

https://github.com/richprogrammer/Applied-Data-Science-Capstone-Project/blob/main/8_notebook_Predictive_Analysis_-_Machine_Learning_Lab.ipynb









# CONCLUSION

- The success rate for SpaceX launches has increased steadily since 2013.
- KSC LC-39A has the most successful launches of any site at 76.9%
- There are ways produce different graphs from the data we retrieve.