
# IBM Data Science Capstone Project

PREFERANCE CHIHUMURA

31 MARCH 2022

IBM Developer

SKILLS NETWORK

# OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

# EXECUTIVE SUMMARY

- Summary of Methodologies.
- Data Collection
- Data Wrangling
- EDA with Data Visualization.
- EDA with SQL.
- Build an interactive map with Folium
- Building a Dashboard with Plotly Dash.
- Classification With Machine Learning

# INTRODUCTION

**BACKGROUNG AND CONTEXT.**

We predicted that the landing stage for Falcon 9 rocket will land successfully. According to Space X's website the rocket will be launched with a cost of US62 million dollars, other providers will have costs up to US165 million dollars. The huge difference is because Space X can reuse its first stage.

## PROBLEMS TO BE ADDRESSED.

-What factors can affect the successful landing of the rocket?

-The best conditions to ensure the successful landing of the rocket.

# METHODOLOGY

- Data Collection Methodology.
  - Space X Rest API.
  - (Web Scrapping) from Wikipedia
- Performed data wrangling (Transforming data for Machine Learning)
- •One Hot Encoding data fields for Machine Learning and dropping irrelevant columns
- •Performed exploratory data analysis (EDA) using visualization and SQL
  - -Plotting : Scatter Graphs, Bar Graphs to show relationships between variables to show patterns of data.
- •Performed interactive visual analytics using Folium and Plotly Dash
- •Performed predictive analysis using classification models
  - •How to build, tune, evaluate classification models

# METHODOLOGOY.

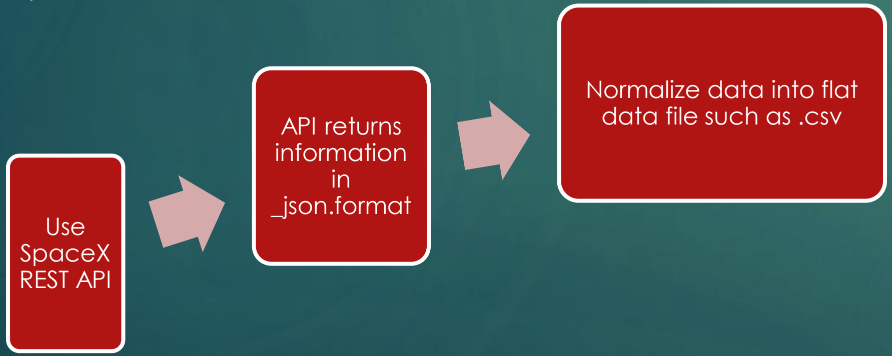



IBM Developer

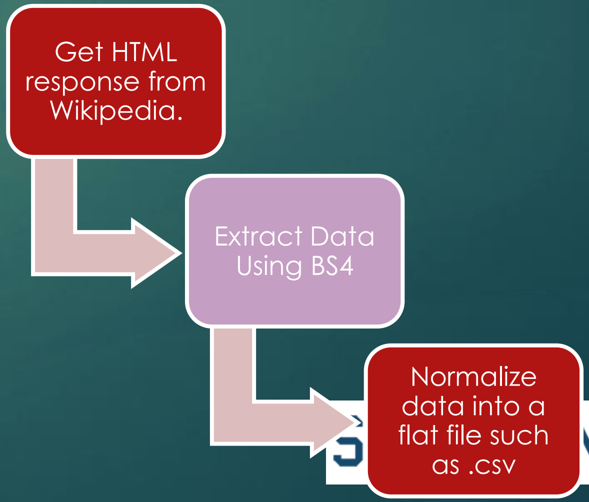
SKILLS NETWORK

# METHODOLOGY.

- The Dataset was collected by:
  - Using  SpaceX launch data gathered by the SpaceX Rest API.
  - The API will give us data about launches, and information about rockets used, payload mass, landing specifications and landing outcomes.
- The motive is to find out if the SpaceX rocket will land or not.
- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/.
- •Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using BeautifulSoup4.
- SPACEX API

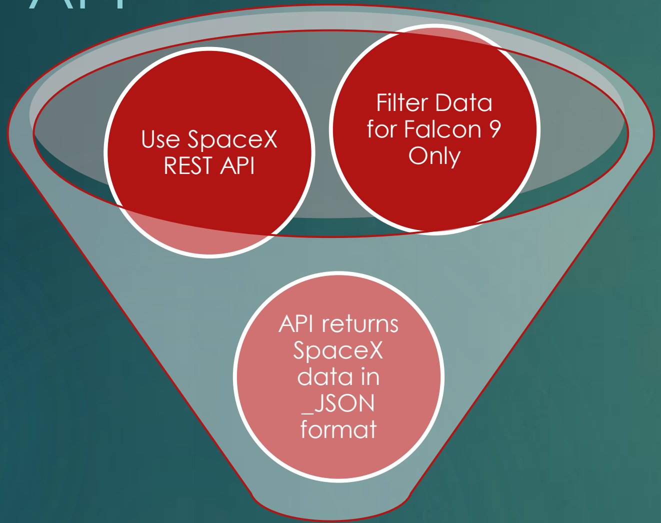



WEB SCRAPPING.



# SPACEX API- FLOW CHART

## Data Collection- SpaceX API




Normalize Data into a flat file such as .csv

## 1.Getting Response from API.

```
def getPayloadData(data):
    for load in data['payloads']:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

## 2. Converting Response to a .json file.

```
# Use json_normalize meethod to convert the json result into a dataframe
from pandas import DataFrame
from pandas import json_normalize
import json


data =pd.json_normalize(response.json())
data
```

## 3. Data Cleaning.

```
getLaunchSite(data):
getBoosterVersion(data):
getPayloadData(data):
getCoreData(data)
```

4. Assigning list to Dict then Dataframe.

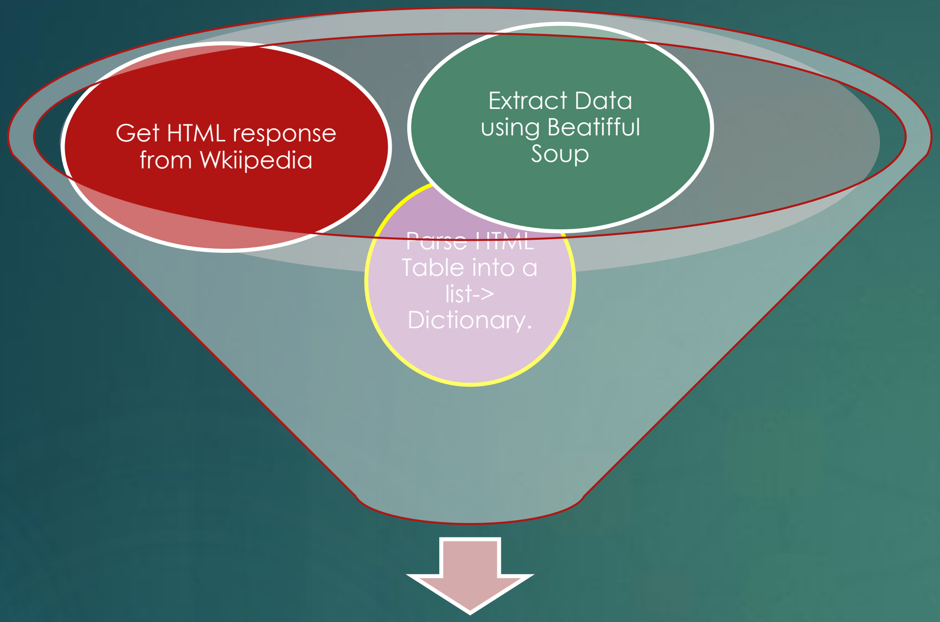
```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Filter the Dataframe.

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data_launch[data_launch.BoosterVersion !='Falcon 1']
len(data_falcon9['BoosterVersion'])
```

[Link to My Github Notebook.]

# DATA COLLECTION -WEBSCRAPPING.




Normalize Data into flat data file such as .csv.

[Link to My Github Notebook.]

- . 1Getting Response from HTML.

```
r= requests.get(static_url)

print(r.text)
```

2. Creating Beautiful Soup Object.

```
doc= BeautifulSoup(r.text, 'html.parser')
print(doc.prettify())
```

3.Finding Tables.

```
html_tables= soup.find_all('table')
```

4. Getting Column Names.

```
for row in columns_table:
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
column_names
```

5.Create A Dictionary.

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant 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']=[]
```

[Link to My Github Notebook]

6. Append data to keys.

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

7. Converting Dictionary to DataFrame.

```
df=pd.DataFrame({key: pd.Series(value) for key, value in launch_dict.items()})
df.head()
```

8.DataFrame to .csv

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

# DATA WRANGLING.

- INTRODUCTION.
- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example,T rue Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean .True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.
- We mainly convert those outcomes into Training Labels with1means the booster successfully landed0means it was unsuccessful.

PROCESS.

Perform Exploratory Data Analysis EDA on Dataset.

Calculate the number of launch sites at each site.

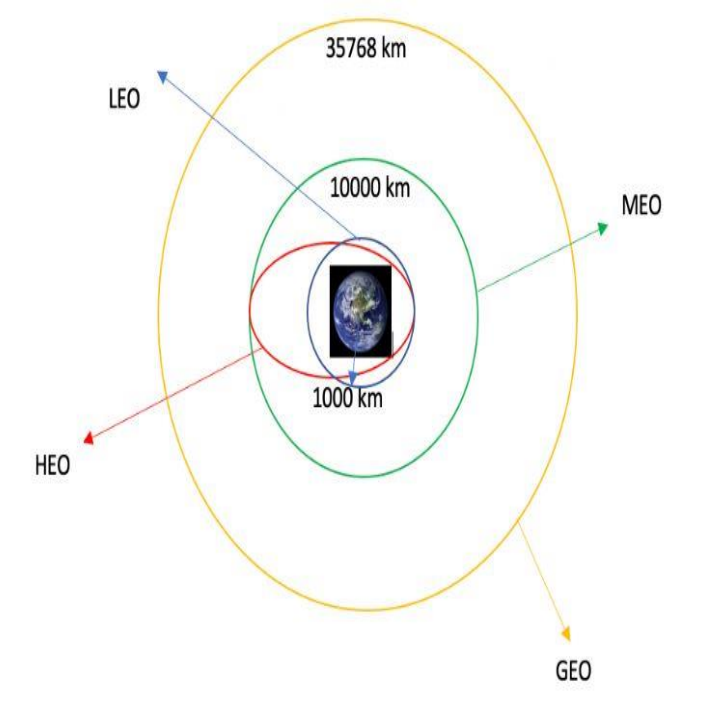
Calculate the number and occurrence of each orbit.

Calculate the number & occurrence of mission outcome per orbit type.

Export dataset as .csv

Create a landing outcome label from Outcome column.

Work out success rate for every landing in dataset.




[Link to My Github Notebook]

IBM Developer

SKILLS NETWORK

## EDA With SQL.

- **Performed SQL queries to gather information about the dataset.**
- **For example, we were asked some questions about the data we needed information about. We are using SQL queries to get the answers in the dataset :**
- **•Displaying the names of the unique launch sites in the space mission**
- **•Displaying 5 records where launch sites begin with the string 'KSC'**
- **•Displaying the total payload mass carried by boosters launched by NASA (CRS)**
- **•Displaying average payload mass carried by booster version F9 v1.1**
- **•Listing the date where the successful landing outcome in the drone ship was achieved.**
- **•Listing the names of the boosters which have success in the ground pad and have payload mass greater than 4000 but less than 6000**
- **•Listing the total number of successful and failed mission outcomes**
- **•Listing the names of the booster versions which have carried the maximum payload mass.**
- **•Listing the records which will display the month names, successful landing outcomes in-ground pad, booster versions, the launch site for the months in the year 2017**
- **•Ranking the count of successful landing outcomes between the dates 2010-06-04 and 2017-03-20 in descending order.**

Link to Github Notebook

Build An Interactive Map with Folium.

**To visualize the Launch Data into an interactive map.** We 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.*
**We assigned the data frame launch outcomes(failures, successes) to *classes 0 and 1***with Green and Red markers on the map in a MarkerCluster()
**Using Haversine's formula we calculated the distance** from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. **Lines are** drawn on the map to measure the distance to landmarks
**Example of some trends in which the Launch Site is situated.**
•Are launch sites in close proximity to railways? No
•Are launch sites in close proximity to highways? No
•Are launch sites in close proximity to the coastline? Yes
•Do launch sites keep a certain distance away from cities? Yes

Link to My Github Notebook..

# CLASSIFICATION BY MACHINE LEARNING.

- **BUILDING MODEL**
- •Load our dataset into NumPy and Pandas
- •Transform Data
- •Split our data into training and test data sets
- •Check how many test samples we have
- •Decide which type of machine learning algorithms we want to use
- •Set our parameters and algorithms to GridSearchCV
- •Fit our datasets into the GridSearchCV objects and train our dataset.

- **EVALUATING MODEL**
- •Check accuracy for each model
- •Get tuned hyperparameters for each type of algorithms
- •Plot Confusion Matrix

- **IMPROVING MODEL**
- •Feature Engineering
- •Algorithm Tuning

- **FINDING THE BEST PERFORMING CLASSIFICATION MODEL**
- •The model with the best accuracy score wins the best performing model
- •In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook.

Link To My GitHub NoteBook.


- Exploratory data analysis results

- Predictive analysis results

- Interactive analytics demo in screenshots

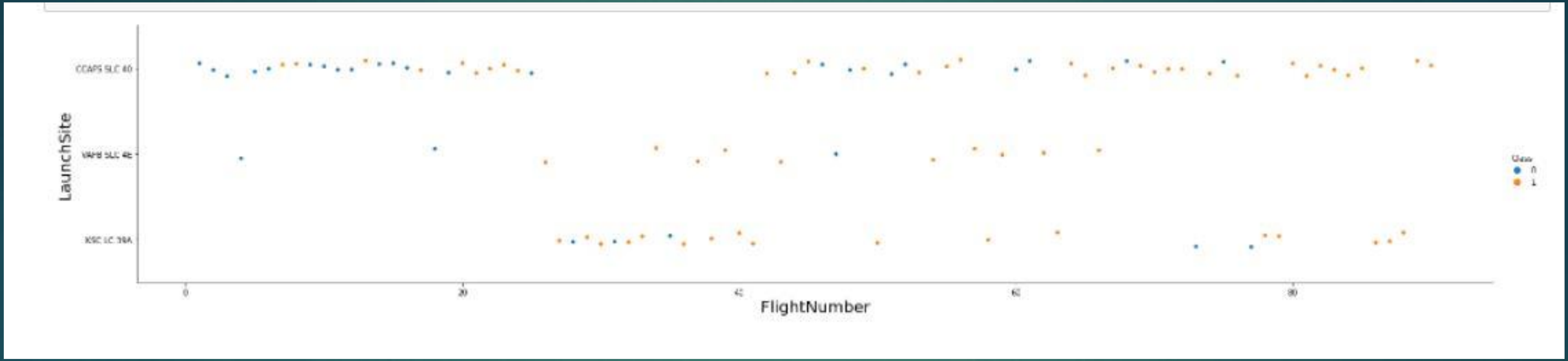
EDA With Visualisation


IBM Developer

SKILLS NETWORK

# EDA With Data Visualisation.

## Flight Numbers vs Flight Site

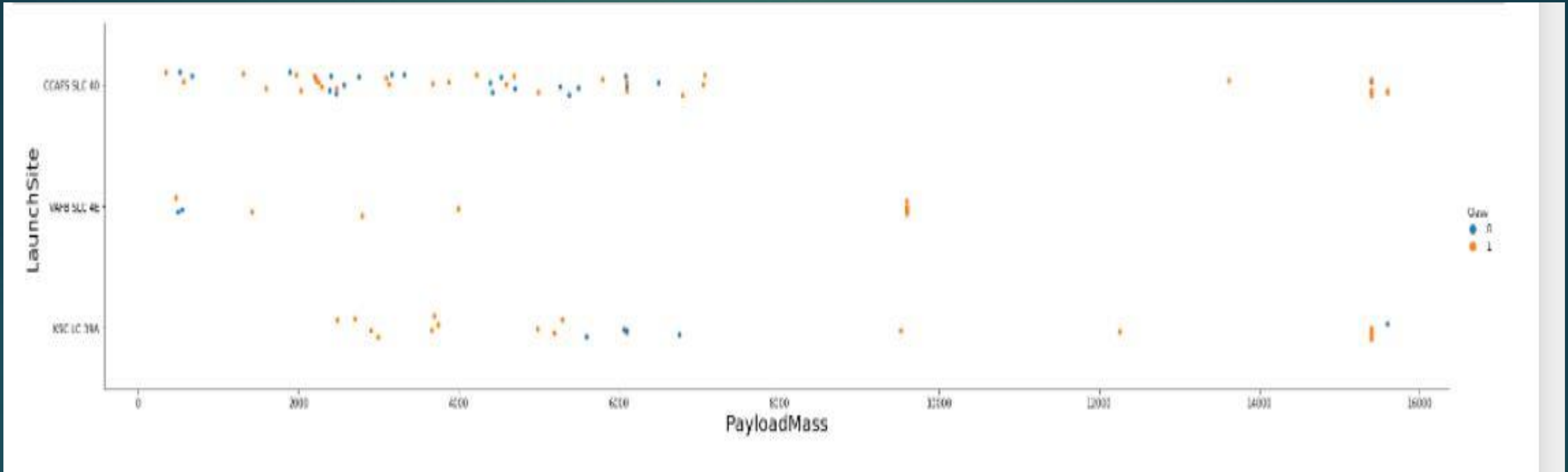



- ➢ The more amount of flights at a launch site the greater the success rate at the launch site.

IBM Developer

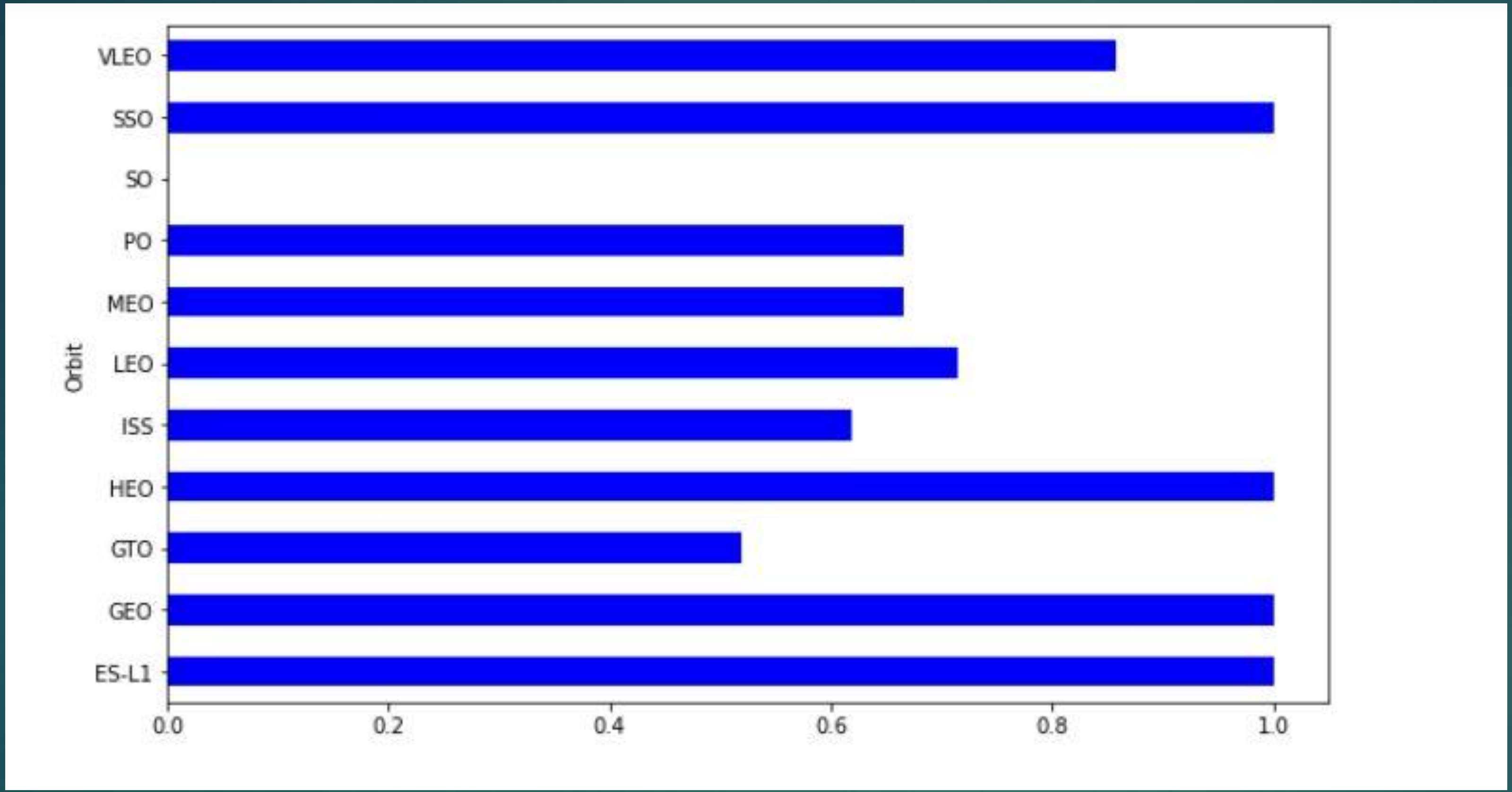
SKILLS NETWORK

# PAYLOAD MASS vs LAUNCH SITE



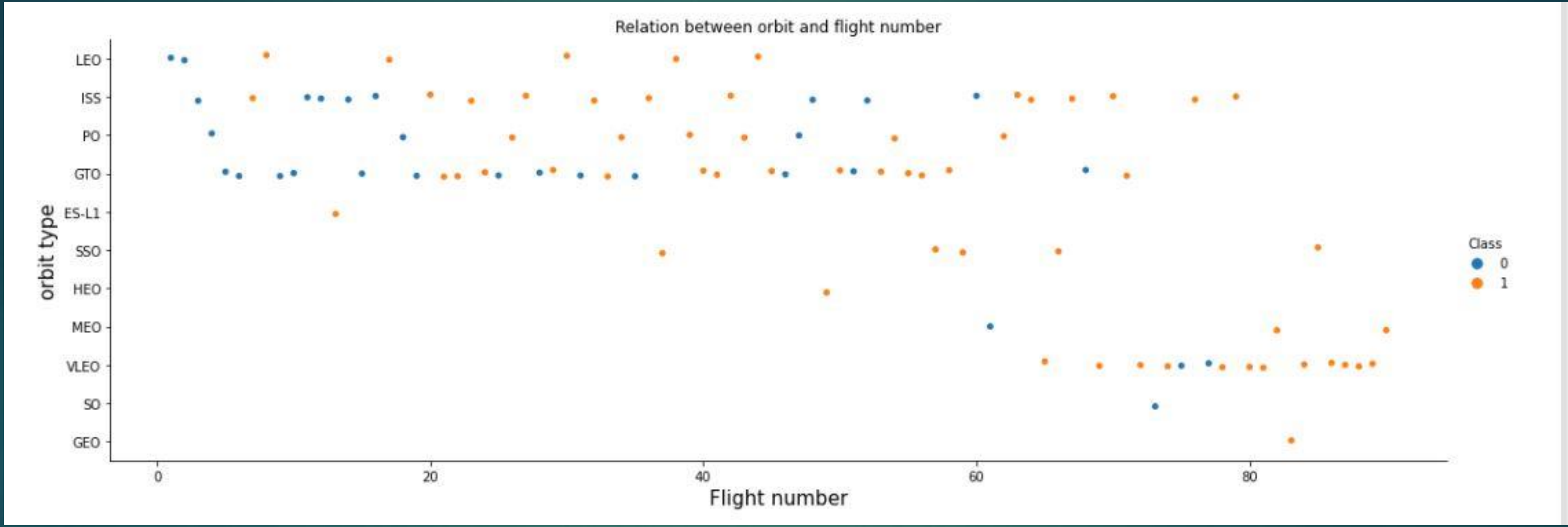

- The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket.
- There is not quite a clear pattern to be found using this visualization to decide if the Launch Site is dependent on Pay Load Mass for a successful launch.

# SUCCESS RATE VS ORBIT TYPE.



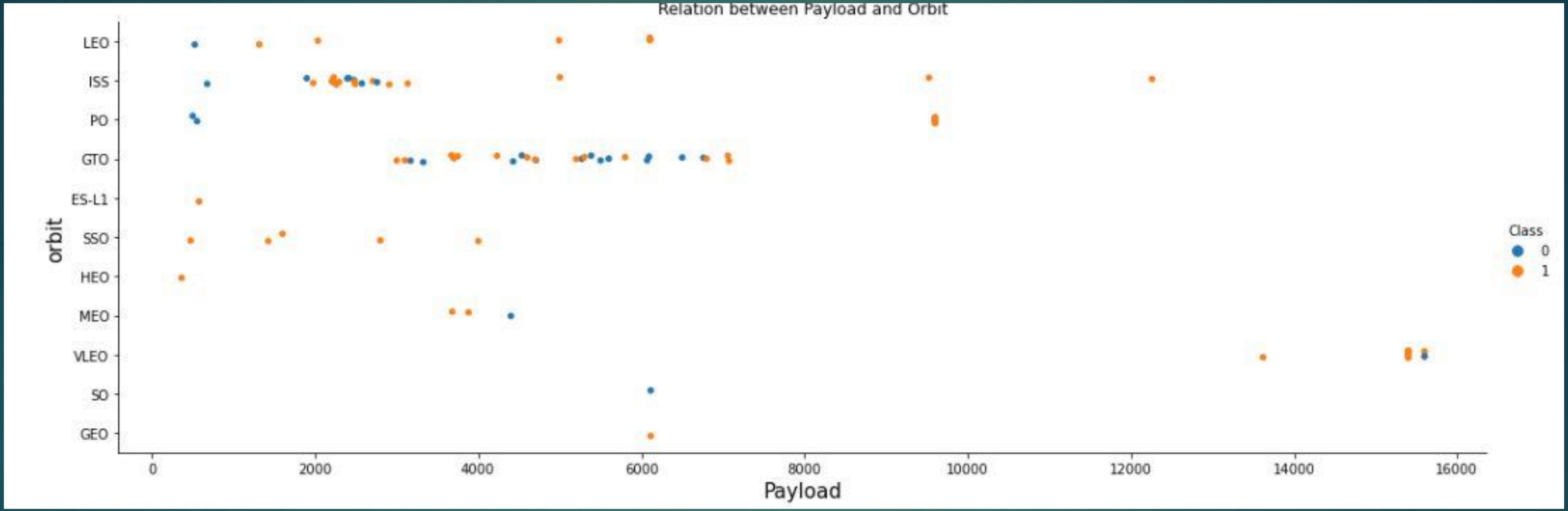

- Orbit SSO, HEO, GEO, ES-L1 had the highest rate of success.

# ORBIT vs FLIGHT NUMBER.



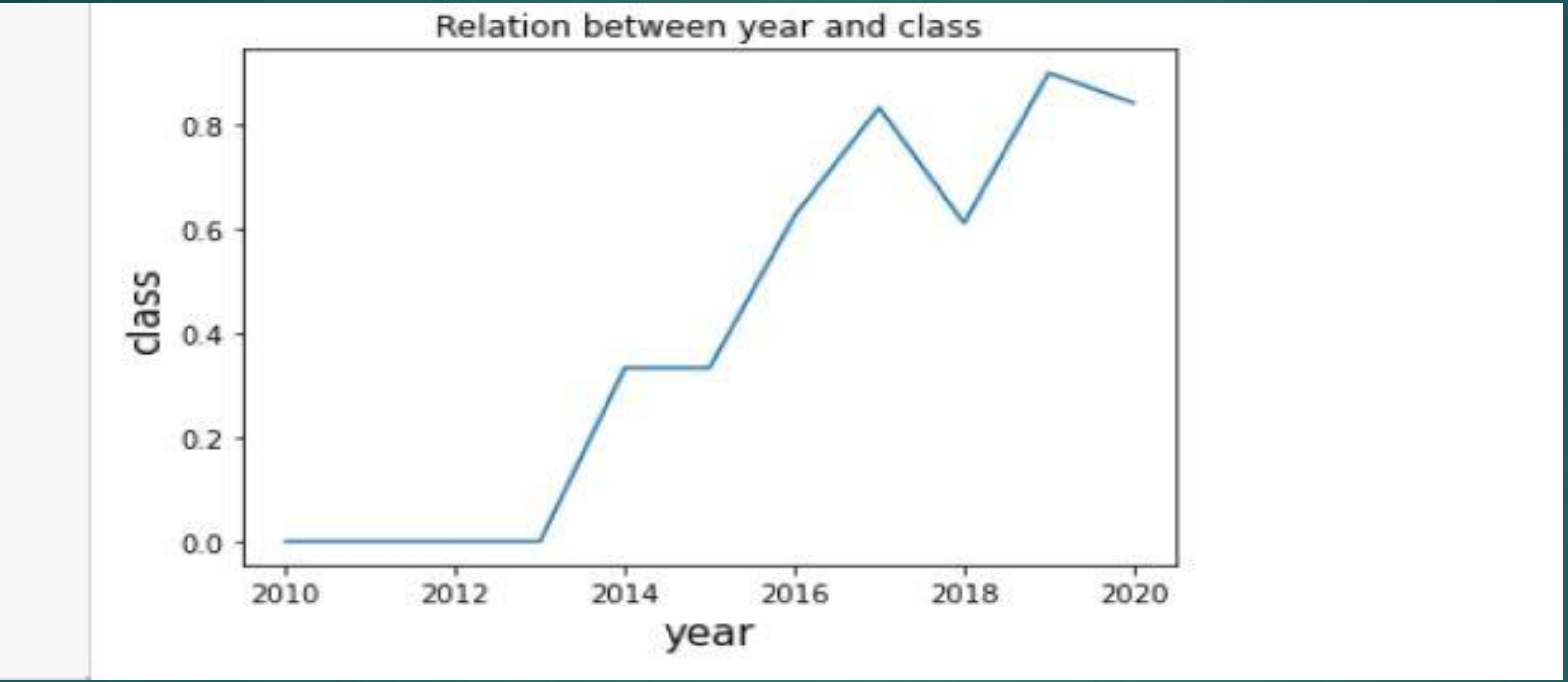

- In the LEO Orbit, the number of flights the more success is those flights, however, there is no correlation in the GTO Orbit,

# PAYLOAD vs ORBIT TYPE.




- You should observe that Heavy payloads have a negative influence on GTO orbits and a positive on GTO and Polar LEO (ISS) orbits.

# LAUNCH SUCCESS YEARLY TREND.




- Due to different technology such as  technology advancement, the success rate kept increasing from 2013 to 2020.

IBM Developer

SKILLS NETWORK

EDA **With** SQL



[Link to My Gitthub Notebook.](#)

IBM Developer

SKILLS NETWORK

## UNIQUE LAUNCH SITES.

%sql select Distinct launch_site from SpaceX.

| launch_site |
| --- |
| CCAFS LC-40 |
| CCAFS SLC-40 |
| KSC LC-39A |
| VAFB SLC-4E |

**QUERY EXPLAINATION**
Using the word ***DISTINCT*** in the query means that it will only show Unique values in the ***Launch_Site*** column from ***SpaceX***

# LAUNCH SITES THAT START WITH "CCA'

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

In [27]:
```
%sql select launch_site from spacex where launch_site like 'CCA%'limit 5;
```

```
 * ibm_db_sa://jhl98678:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

Out[27]:

| launch_site |
|---|
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |

**QUERY EXPLANATION**

Using the word ***Launch_site*** from the table to select all launch sites.

Use a delimitor "***CCA%-LIMIT 5",*** to show only 5 sites.

Total Payload Mass by Customer (NASA).

```
%sql select sum(payload_mass__kg_) from spacex where customer = 'NASA (CRS)';

 * ibm_db_sa://jhl98678:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

| 1 |
|---|
| 22007 |

QUERY EXPLANATION.
Using the function ***SUM*** summates the total in the column ***PAYLOAD_MASS_KG.***

The ***WHERE*** clause filters the dataset to only perform calculations on ***CUSTOMER.***

# Average Payload carried by Booster version F9 v1.1

```
%%sql
select "BOOSTER_VERSION", avg(payload_mass__kg_) as average_payload_mass__kg_ from spacex
group by "BOOSTER_VERSION"
```

```
 * ibm_db_sa://jhl98678:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

| booster_version | average_payload_mass__kg_ |
|---|---|
| F9 B4 B1039.2 | 2647 |
| F9 B4 B1040.2 | 5384 |
| F9 B4 B1040.1 | 4990 |
| F9 B4 B1041.1 | 9600 |
| F9 B4 B1043.1 | 5000 |
| F9 B4 B1044 | 6092 |

QUERY EXPLANATION.

Using ***Average*** function works out the average in the column.

The Date a successful landing was achieved.

```
%sql select min(DATE) from spacex where landing__outcome = 'Success (ground pad)'

 * ibm_db_sa://jhl98678:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

| 1 |
|---|
| 2017-01-05 |

QUERY EXPLANATION.

Using the ***MIN*** function works out the date in the column ***Date.***

The ***Where*** clause filters the dataset to only perform calculations.

Successful drone ship landing with payload between 4000 & 6000.

***SQL QUERY.***

%sql select booster_version from spacex where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000.

| booster_version |
| --- |
| F9 FT B1022 |
| F9 FT B1031.2 |

QUERY EXPLANATION.

Selecting only ***Booster_Version***
The ***where clause*** filters the dataset to ***Landing_Outcome = Success (drone ship)***
The ***AND*** clause specifies additional filter conditions
***Payload_MASS_KG_ between 4000 & 6000***

## LANDING OUTCOME.

```
%sql select COUNT(MISSION_OUTCOME) as MISSION_OUTCOME from spacex group by MISSION_OUTCOME;
```

```
 * ibm_db_sa://jhl98678:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

| mission_outcome |
|---|
| 44 |
| 1 |

## QUERY EXPLANATION.

We use the ***SELECT*** function, to filter ***mission outcomes*** and then group using the ***Mission Outcomes***

## BOOSTERS THAT CARRIED MAXIMUM PAYLOAD.

```
%sql select booster_version from spacex where payload_mass__kg_ = (select max(payload_mass__kg_) from spacex);
```

```
 * ibm_db_sa://jhl98678:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

| booster_version |
|---|
| F9 B5 B1048.4 |
| F9 B5 B1049.4 |
| F9 B5 B1049.5 |
| F9 B5 B1060.2 |
| F9 B5 B1058.3 |

QUERY EXPLANATION.

Use the Select function to select Booster Version.
Using the Max function to select the maximum payload in the Payload Mass Column

## Failed Landing Outcomes, Droneships, Booster Versions, and LaunchSite Names for the Year 2015.

SQL QUERY.

%sql select BOOSTER_VERSION, launch_site, year(DATE) from spacex WHERE extract(YEAR FROM DATE) = '2015' and LANDING__OUTCOME = 'Failure (drone ship)'

```
%sql select BOOSTER_VERSION, launch_site, year(DATE) from spacex WHERE extract(YEAR FROM DATE) = '2015' and LANDING__OUTCOME = 'F
```

```
 * ibm_db_sa://jhl98678:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

| booster_version | launch_site | 3 |
|---|---|---|
| F9 v1.1 B1012 | CCAFS LC-40 | 2015 |

# Successful Landing Outcomes Between 2010/06/04 and 2017/03/20.

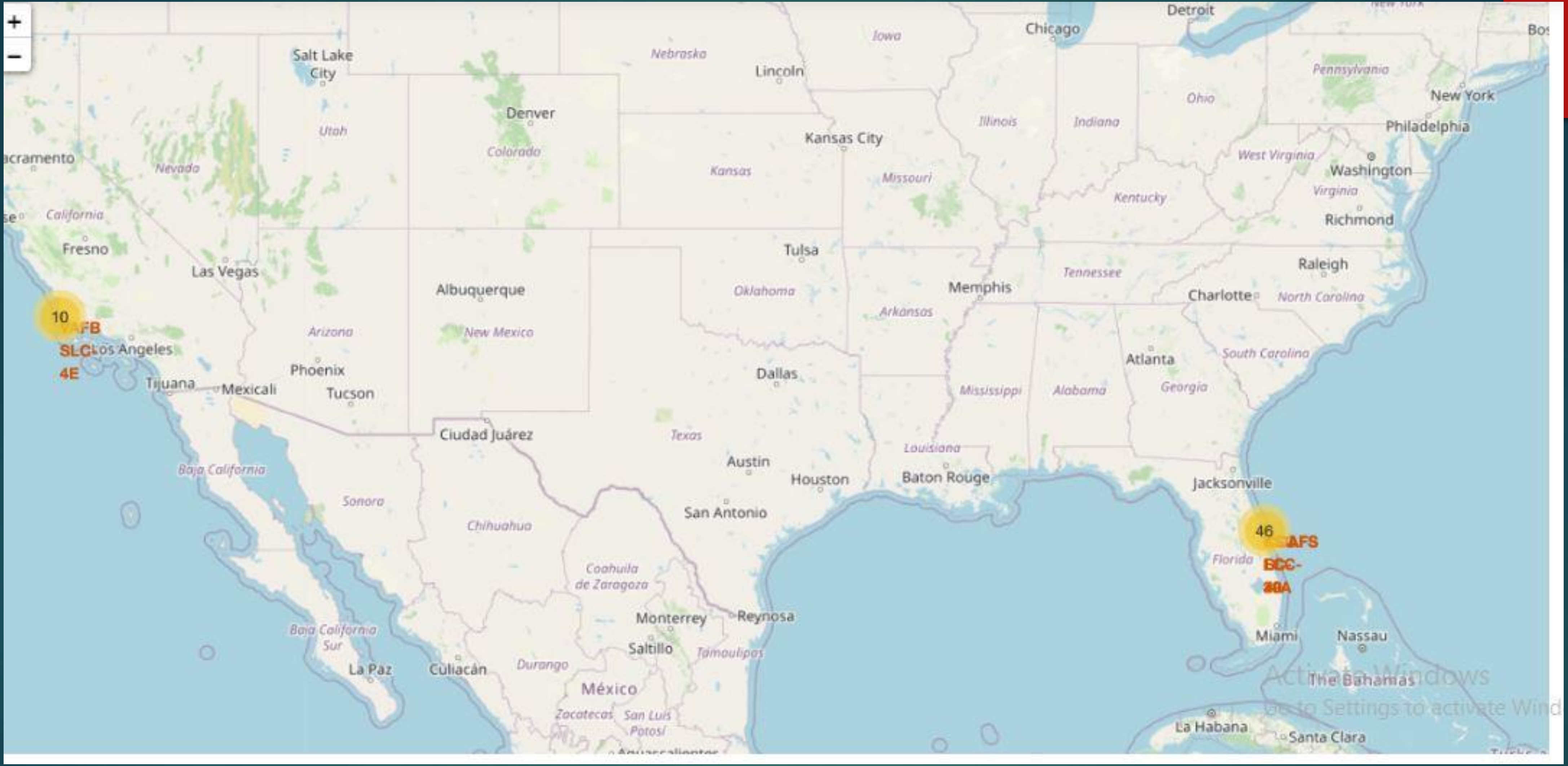
```
%sql select landing__outcome, count(landing__outcome) as total from spacex \
where landing__outcome like 'Success%' and DATE between '2010-06-04' and '2017-03-20' \
group by landing__outcome \
order by count(landing__outcome) desc;
```

```
 * ibm_db_sa://jhl98678:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

| landing__outcome | total |
|---|---|
| Success (drone ship) | 2 |
| Success (ground pad) | 2 |

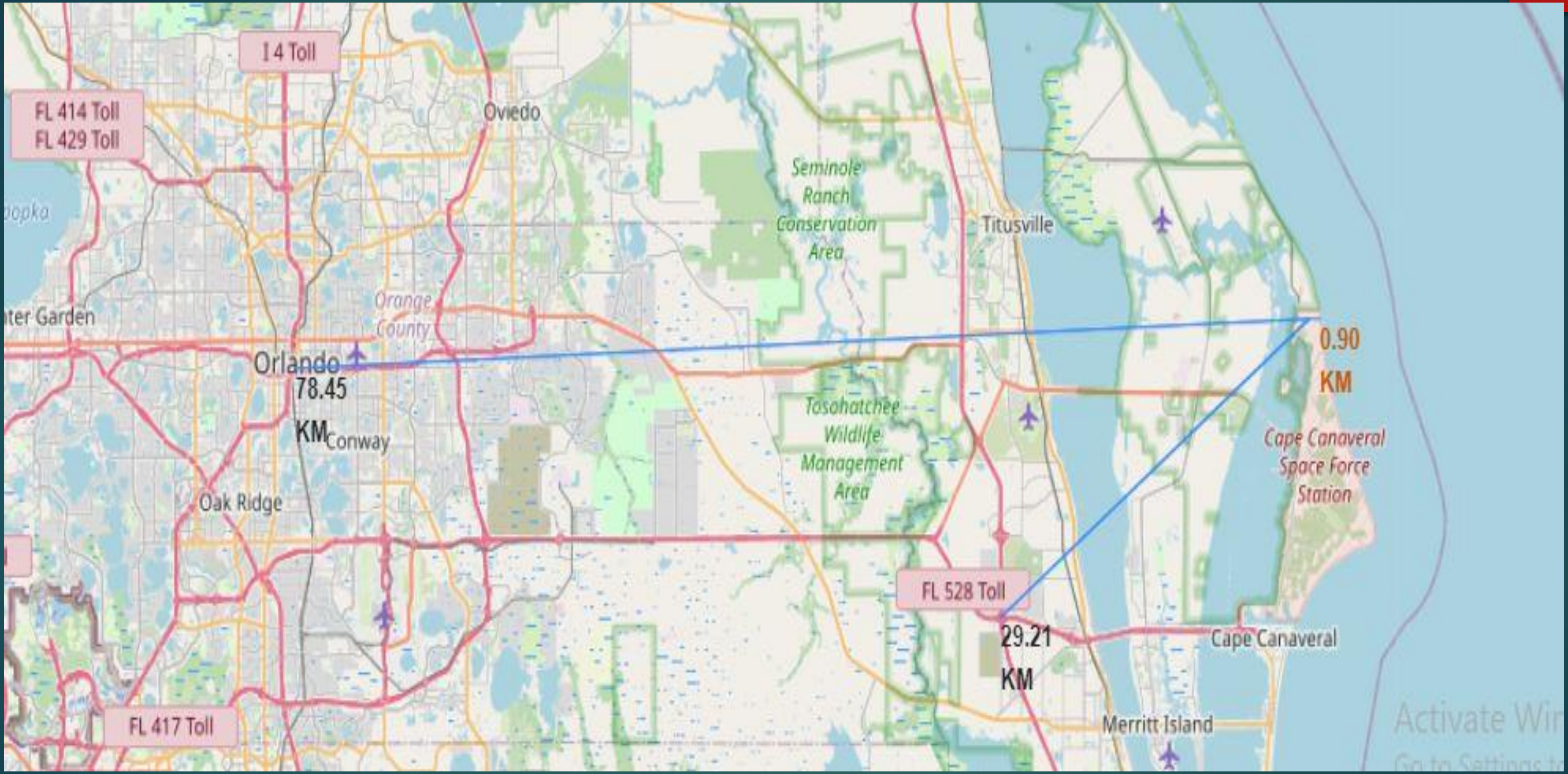
# INTERACTIVE MAP WITH FOLIUM

[Link to My Github Notebook]

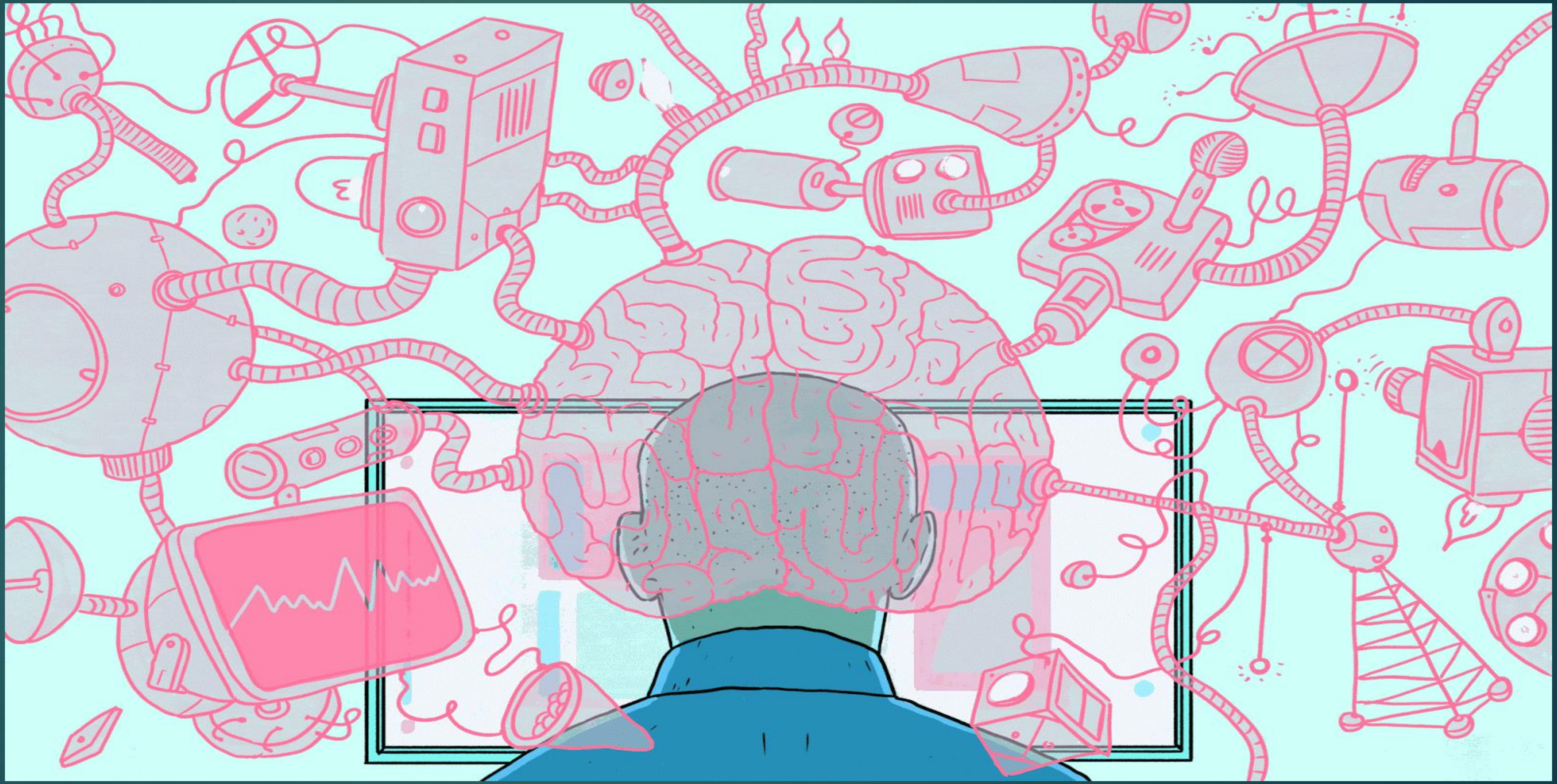



SpaceX launch sites are located on the East & West Coast of the United States of America.

Working out Launch Sites distance to landmarks to find trends with Haversine formula using CCAFS-SLC-40 as a reference.



# PREDICTIVE ANALYSIS (Classification)

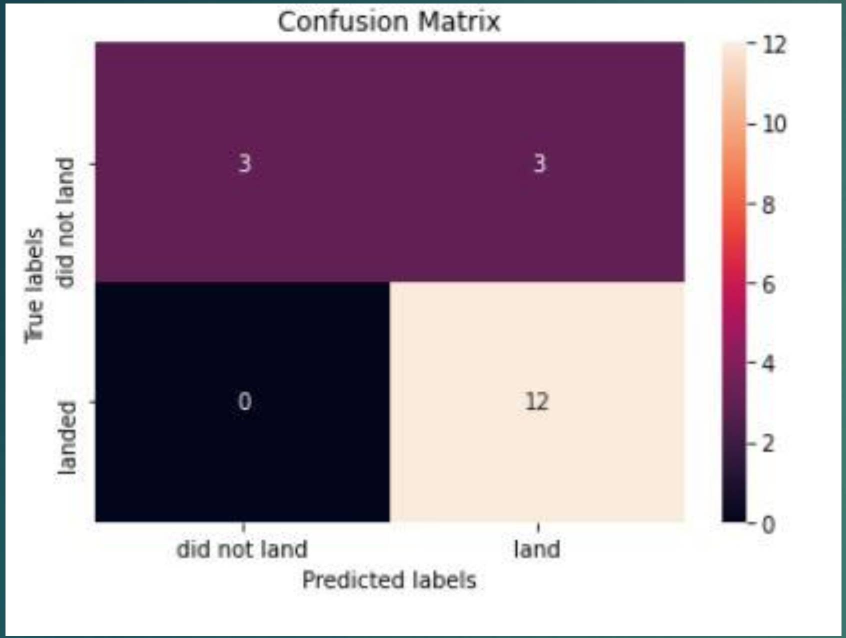


[Link to My Github Notebook.]

IBM Developer

SKILLS NETWORK

After selecting the best hyperparameters for the decision tree classifier using the validation data, we achieved 83.33% accuracy on the test data.



# CONCLUSION.

- •The Tree Classifier Algorithm is the best for Machine Learning for this dataset
- •Low weighted payloads perform better than the heavier payloads
- •The success rates for SpaceX launches are directly proportional time in years they will eventually perfect the launches
- •We can see that KSC LC-39A had the most successful launches from all the sites
- •Orbit GEO, HEO, SSO, ES-L1 has the Best Success Rate

# PROGRAMMING LANGUAGE TRENDS

Current Year

<Bar chart of top 5 programming languages for the current year goes here.>

Next Year

< Bar chart of top 5 programming languages for the next year goes here.>

# PROGRAMMING LANGUAGE TRENDS - FINDINGS & IMPLICATIONS

Findings

- Finding 1
- Finding 2
- Finding 3

Implications

- Implication 1
- Implication 2
- Implication 3

# DATABASE TRENDS

Current Year

< Bar chart of top 5 databases for the current year goes here >

Next Year

< Bar chart of top 5 databases for the next year goes here.>

IBM Developer

SKILLS NETWORK

# DATABASE TRENDS - FINDINGS & IMPLICATIONS

Findings

- Finding 1
- Finding 2
- Finding 3

Implications

- Implication 1
- Implication 2
- Implication 3

# DASHBOARD

<The permanent link of the read-only view of the Cognos dashboard goes here.>

# DASHBOARD TAB 1

Screenshot of dashboard tab 1 goes here

# DASHBOARD TAB 2

Screenshot of dashboard tab 2 goes here

IBM Developer SKILLS NETWORK

# DASHBOARD TAB 3

Screenshot of dashboard tab 3 goes here

# DISCUSSION

# OVERALL FINDINGS & IMPLICATIONS

Findings

- Finding 1
- Finding 2
- Finding 3

Implications

- Implication 1
- Implication 2
- Implication 3

# CONCLUSION

- Point 1
- Point 2
- Point 3
- Point 4

# APPENDIX

- Include any relevant additional charts, or tables that you may have created during the analysis phase.

IBM Developer

SKILLS NETWORK

# GITHUB JOB POSTINGS

In Module 1 you have collected the job postings data using GitHub API in a file named “github-job-postings.xlsx”. Present that data using a bar chart here. Order the bar chart in the descending order of number of job postings.

# POPULAR LANGUAGES

In Module 1 you have collected the job postings data using web scraping in a file named “popular-languages.csv”. Present that data using a bar chart here. Order the bar chart in the descending order of salary.