APPLIED DATA SCIENCE PROJECT

KEHINDE ADELAKIN

CONTENT

- Executive Summary
- Introduction
- Methodology
- Conclusion

EXECUTIVE SUMMARY

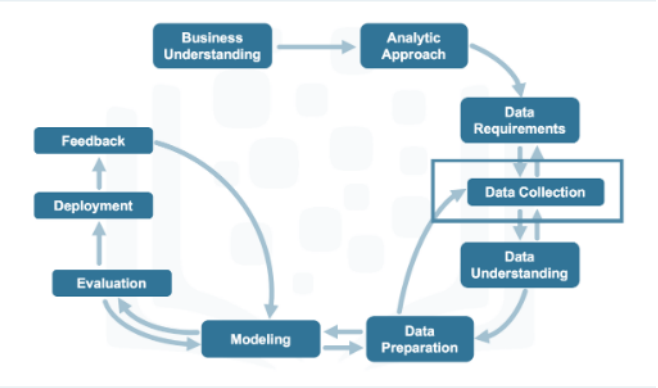
- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

INTRODUCTION

SpaceX is a ground-breaking corporation that has completely changed the space sector by providing rocket launches, notably Falcon 9, for as little as $62 million, compared to other suppliers that charge as much as $165 million for each launch. The majority of these savings are attributable to SpaceX's brilliant concept to re-land the rocket after the first stage of the launch so that it can be used on a later flight. The price will drop considerably more if this practice is repeated. The objective of this project, as a data scientist for a firm that competes with SpaceX, is to develop the machine learning pipeline to forecast the first stage landing result in the future. Finding the appropriate amount to bid against SpaceX for a rocket launch is dependent on this research.

DATA COLLECTION AND DATA WRANGLING METHODOLOGY

Methodology



Data collection methodology:

- Data Gathering: Collect data using the SpaceX REST API and web scraping from Wikipedia.
- Data Wrangling: Clean and prepare the data, handling missing values and outliers.
- Data Processing: Utilize one-hot encoding to transform categorical features into numerical format.
- Exploratory Data Analysis (EDA): Explore data patterns and relationships using visualization tools like matplotlib and SQL queries.
- Interactive Visual Analytics: Employ Folium and Plotly Dash for interactive visualizations to gain deeper insights.
- Model Building: Split the data into training and testing sets, select appropriate classification algorithms (e.g., Logistic Regression, Decision Trees), and train the models.
- Model Tuning: Fine-tune model hyperparameters using techniques like GridSearchCV or RandomizedSearchCV to improve performance.
- Model Evaluation: Evaluate models using metrics such as accuracy, precision, recall, and confusion matrices to assess predictive capabilities.
- Iterative Improvement: Iterate on the model building and tuning process based on evaluation results to enhance model accuracy and stability.

DATA COLLECTION AND DATA WRANGLING METHODOLOGY

Data Collection

Data collection refers to the systematic process of gathering and measuring information about specific variables within a defined system. This process allows for answering pertinent questions and assessing outcomes effectively. In this context, the dataset under consideration was obtained through the SpaceX REST API and web scraping from Wikipedia.

Initiate data collection with a GET request to the REST API. Decode the JSON response and convert it into a pandas DataFrame using json_normalize(). Clean the data, handle missing values, and perform necessary preprocessing.

We'll use BeautifulSoup for web scraping to extract launch records as an HTML table. Then, parse and convert the table into a pandas DataFrame for analysis.

DATA COLLECTION AND DATA WRANGLING METHODOLOGY

Data Collection – Scrapping

Get request for data

Use pandas to convert result to dataframe

Performed data cleaning and filling the missing value

Create a BeautifulSoup from the HTML response

Extract all column/variable names

Create Data Frame

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

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

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

```
data = requests.get(static_url).text
```

```
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()
```

```
headings = []
for key,values in dict(launch_dict).items():
    if key not in headings:
        headings.append(key)
    if values is None:
        del launch_dict[key]


def pad_dict_list(dict_list, padel):
    lmax = 0
    for lname in dict_list.keys():
        lmax = max(lmax, len(dict_list[lname]))
    for lname in dict_list.keys():
        ll = len(dict_list[lname])
        if  ll <lmax:
            dict_list[lname] += [padel] * (lmax - ll)
    return dict_list

pad_dict_list(launch_dict,0)

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

DATA COLLECTION AND DATA WRANGLING METHODOLOGY

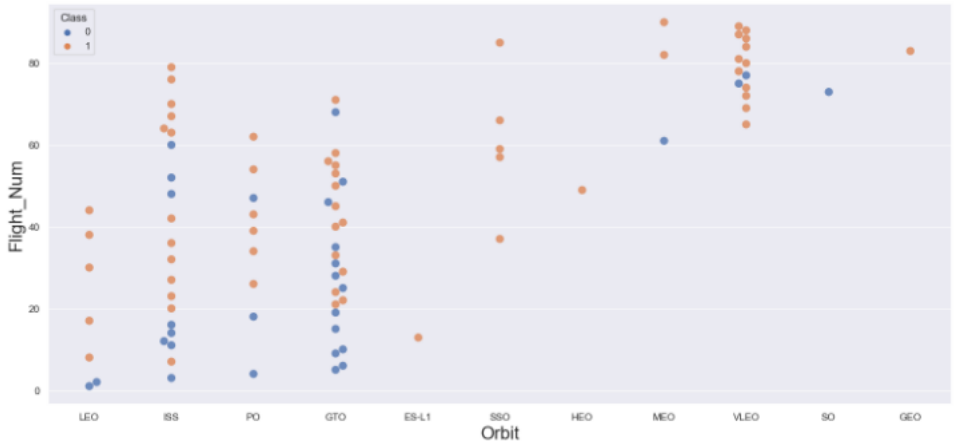
Data Collection – Wrangling

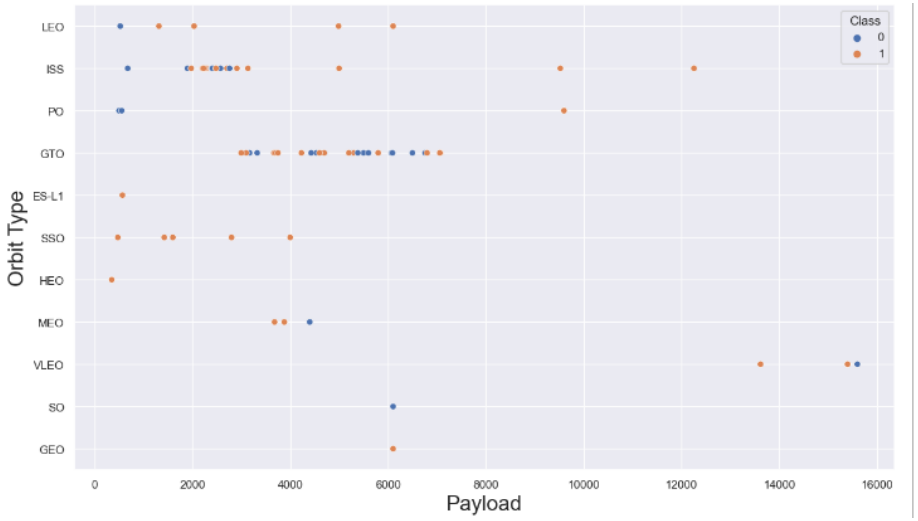
Data wrangling involves cleaning and organizing messy datasets for efficient access and Exploratory Data Analysis (EDA).

First, we'll determine the number of launches at each site and analyze the frequency of mission outcomes per orbit type.

Next, we'll generate a landing outcome label based on the outcome column, facilitating analysis, visualization, and machine learning tasks. Finally, we'll export the processed data to a CSV file.

EDA WITH DATA VISUALIZATION





Scatter chart is one of the easiest way to interpret the relationship between the attributes.

We used this chart to find the relationship between the attributes such as between:

- Flight Number and Orbit Type.
- Payload and Orbit Site.

EDA WITH SQL

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

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

```
%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;
```

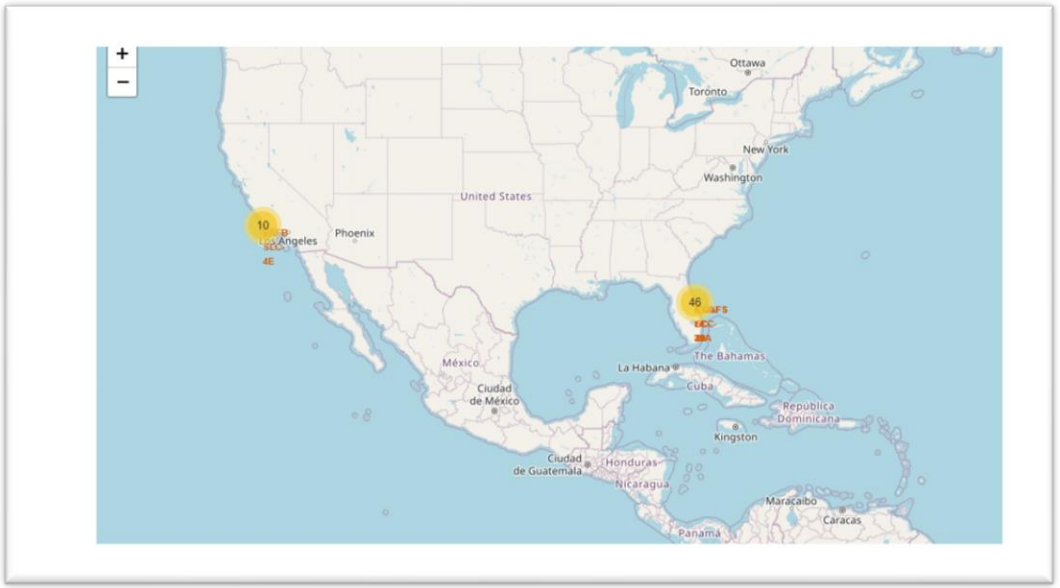
```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

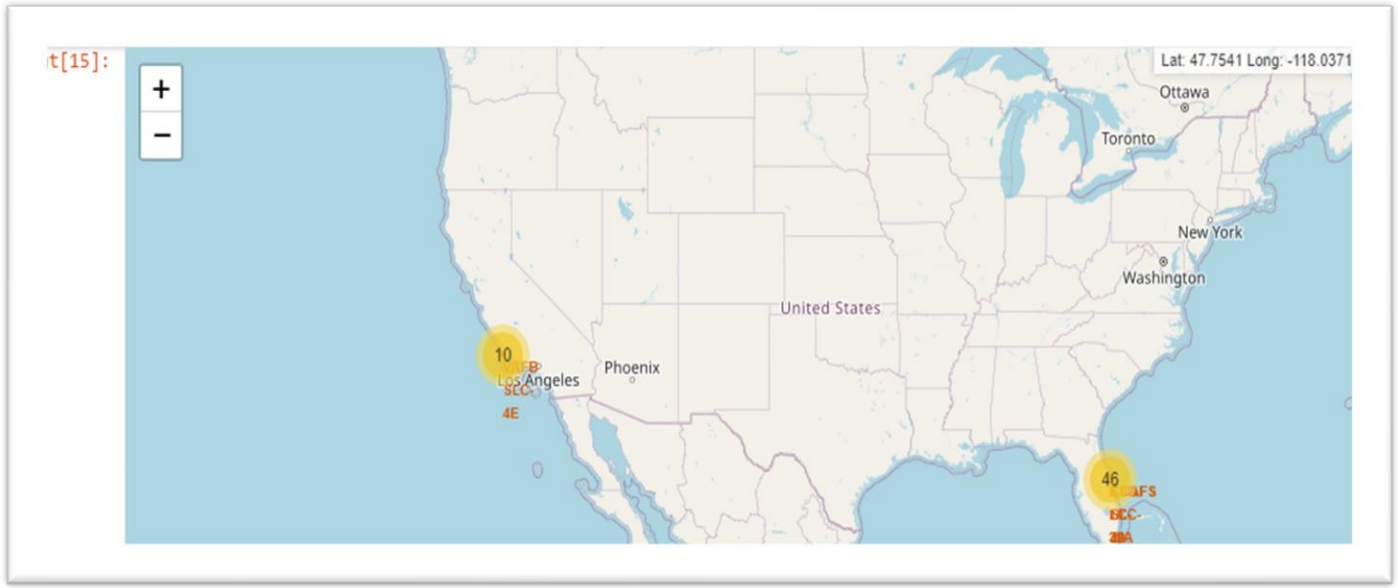
We ran a number of SQL queries to gain a deeper comprehension of the dataset, such as:

- Displaying the names of the launch sites
- Listing the date when the first successful landing outcome in ground pad was achieved
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster_versions which have carried the maximum payload mass

BUILD AN INTERACTIVE MAP WITH FOLIUM

To create an interactive map using the launch data as a graphic. We placed a circle marker with the name of the launch location labeled around each launch site after obtaining the latitude and longitude coordinates at each launch site.

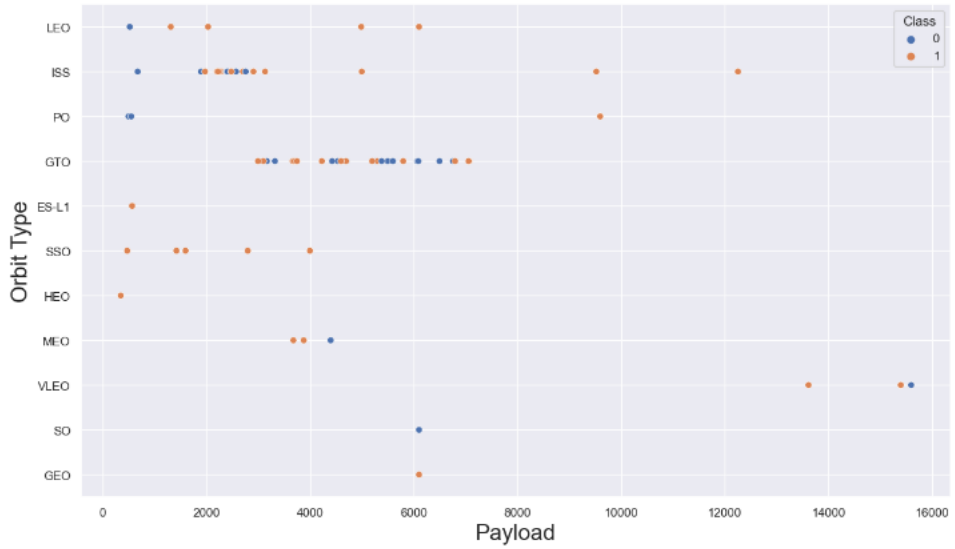




BUILD A DASHBOARD WITH PLOTLY DASH

Using Plotly Dash, we created an interactive dashboard that lets the user manipulate the data however they see fit..

- We created scatter plot that displayed Flight Number and Orbit Type.
- We also created a scatter plot that shows the Payload and Orbit Site



PREDICTIVE ANALYSIS (CLASSIFICATION)

Constructing the Model:

- Import the dataset into Pandas and NumPy
- Split the data into training and test datasets after transforming it.
- Select the machine learning type to employ,
- Then adjust GridSearchCV's settings and methods to fit the dataset.

Assessing the Model:

- Verify each model's accuracy;
- Adjust the hyperparameters for each kind of method.
- Draw the matrix of misunderstanding.

Enhancing the Model:

- Apply Algorithm Tuning and Feature Engineering

Locate the most suitable Model:

- The model that performs the best will be the one with the highest accuracy score.

CONCLUSION

We can then draw the following conclusion:

- From 2013 onwards, SpaceX's launch success rate has improved in direct proportion to the number of years until 2020, when it is expected to achieve launch perfection.
- With a success percentage of 100% and multiple occurrences, SSO orbits have the highest rate.

THANK YOU