SPACE X

IBM DATA SCIENCE CAPSTONE PROJECT

Diego Ignacio Ortiz

CONTENT OUTLINE

EXECUTIVE SUMMARY

- METHODOLOGIES

1. Data Collection with RestAPI and Web Scraping
2. EDA with Data Visualization
3. EDA with SQL
4. Interactive Maps with Folium
5. Interactive Dashboards with Plotly Dash
6. Predictive Analysis (Classification)

- RESULTS

1. Exploratory Data Analysis (EDA)
2. Interactive Analytics
3. Predictive Analytics

INTRODUCTION

Project background and context

This project aims to predict the success of Falcon 9's first stage landing. SpaceX can reuse the first stage, reducing launch costs to 62 million dollars compared to other providers who charge upward of 165 million dollars per launch. By predicting landing success, we can determine launch cost and provide valuable information for competitors.

Problems you want to find answers

What are the key features of a successful or unsuccessful landing for Falcon 9's first stage?

How do different rocket variables impact the success or failure of the landing?

What conditions can SpaceX implement to achieve the highest landing success rate?

METHODOLOGY

01

METHODOLOGY -EXECUTIVE SUMMARY

Data Collection

- SpaceX REST API
- Web Scraping from Wikipedia

Data Wrangling

- Sampling Data
- Dealing with Missing Values
- One Hot Encoding for classification models

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

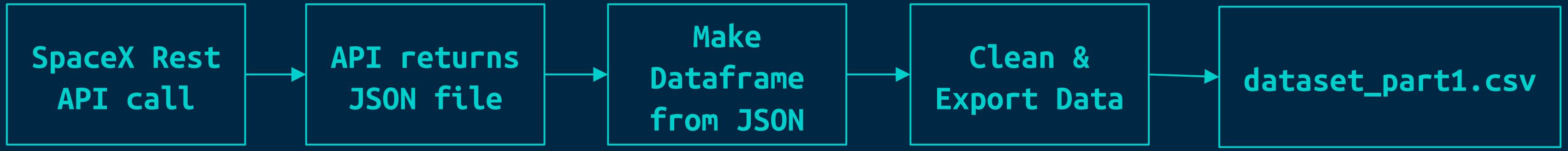
Perform predictive analysis using classification models

DATA COLLECTION

To Collect the data into Datasets we use the SpaceX Rest API

We obain data such as rocket used, payload delivered, launch speecifications, landing specifications and landung outcome.

The Space X Rest API Endpoint URL is api.spacexdata.com/v4/



DATA COLLECTION

We can also use Web Scraping from Wikipedia

We use the url :
https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

Get HTML response from Wikipedia → Extract data with BeautifulSoup → Make Dataframe → Export Data → spacex_web_scraped.csv

METHODOLOGY -EXECUTIVE SUMMARY

Data Collection

- SpaceX REST API
- Web Scraping from Wikipedia

Data Wrangling

- Sampling Data
- Dealing with Missing Values
- One Hot Encoding for classification models

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 with SpaceX
REST API calls

1 .Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url).json()
```

2. Converting Response to a .json file

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3. Apply custom functions to clean data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```
getBoosterVersion(data)
```

4. Assign list to dictionary 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}
```

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

5. Filter dataframe and export to flat file (.csv)

```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

DATA COLLECTION

Web Scrapping from Wikipedia

1 .Getting Response from HTML

```
page = requests.get(static_url)
```

2. Creating BeautifulSoup Object

```
soup = BeautifulSoup(page.text, 'html.parser')
```

3. Finding tables

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

4. Getting column names

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

5. Creation of dictionary

```
launch_dict= dict.fromkeys(column_names)

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


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'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Appending data to keys (refer) to notebook block 12

```
In [12]: extracted_row = 0
         #Extract each table
         for table_number,table in enumerate(
            # get table row
             for rows in table.find_all("tr")
                 #check to see if first table
```

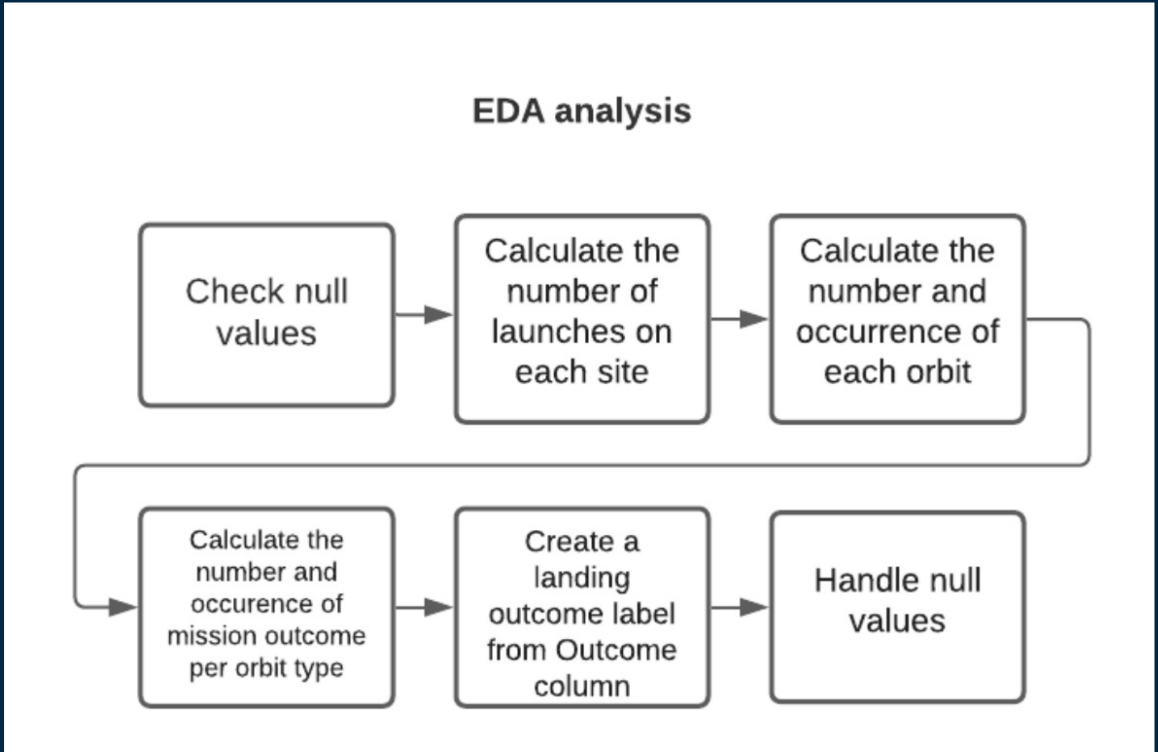
7. Converting dictionary to dataframe

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

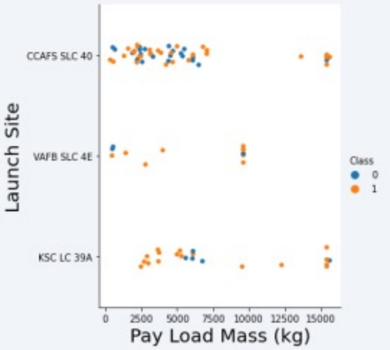
8. Dataframe to .CSV

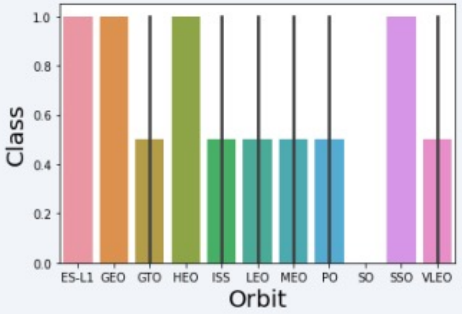
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

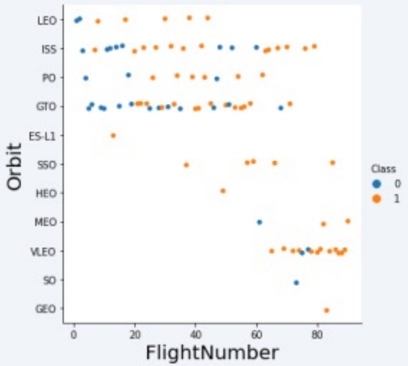
DATA WRANGLING

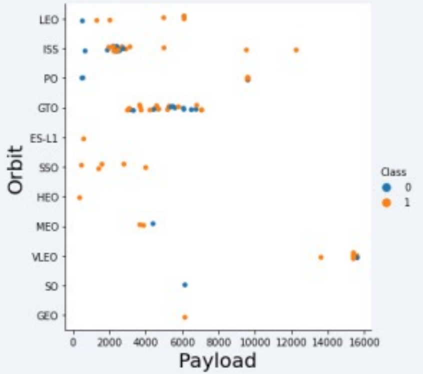


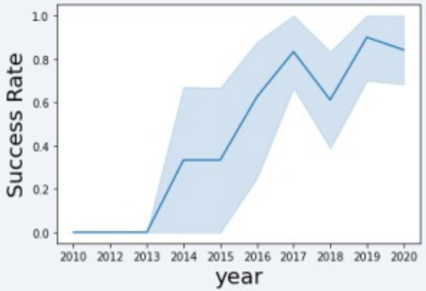
EDA DATA VISUALIZATION









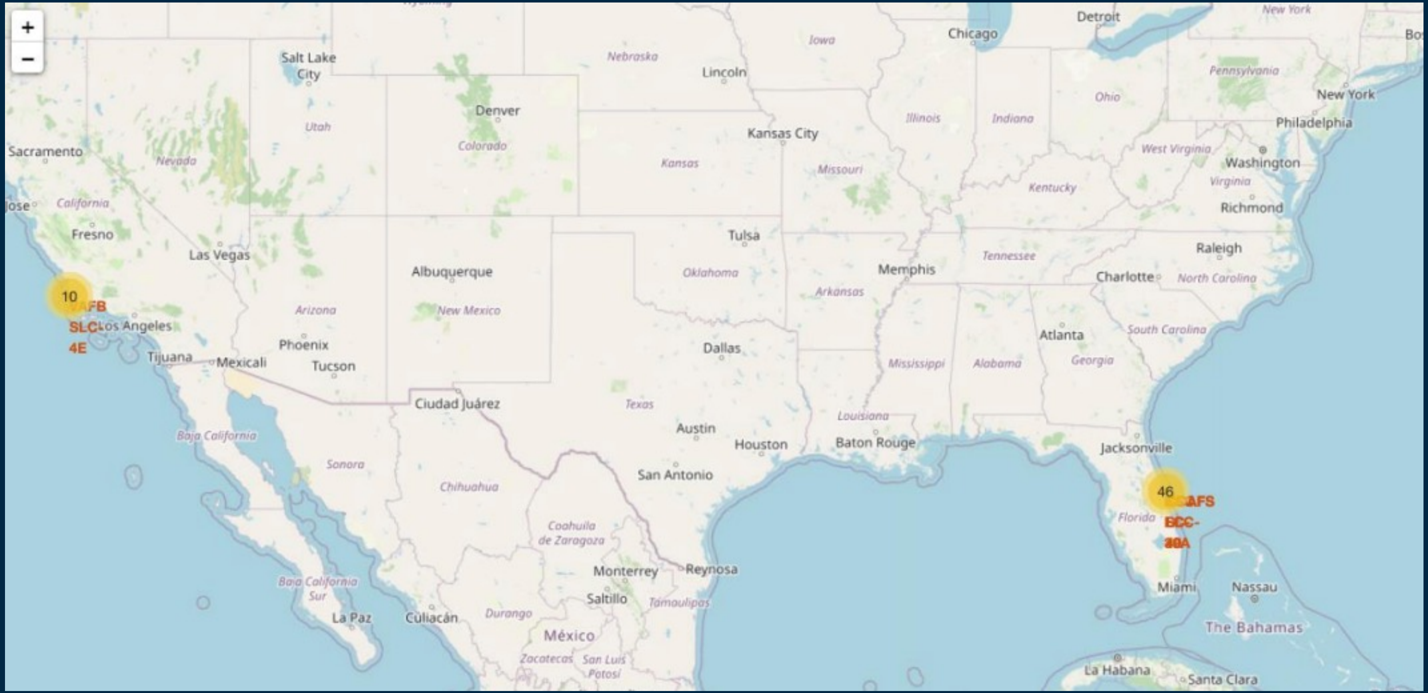


EDA WITH SQL

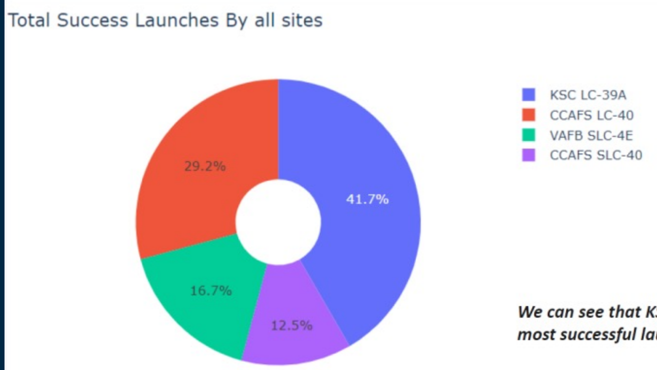
SQL queries performed include:

- 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 drone ship was achieved.
- Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000.
- Listing the total number of successful and failure 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, launch_site for the months in year 2017
- Ranking the count of successful landing_outcomes between the date 2010 06 04 and 2017 03 20 in descendingorder.

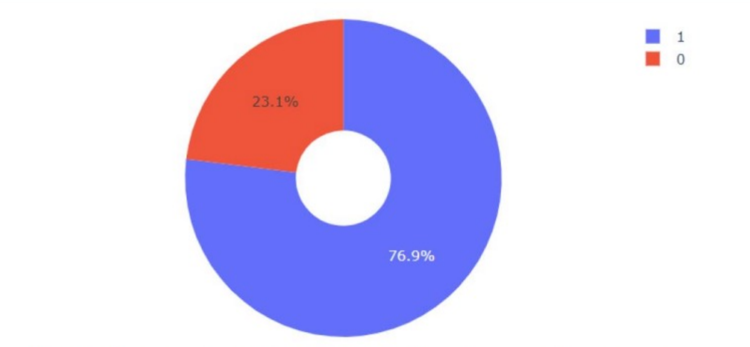
INTERACTVIE MAP WITH FOLIUM



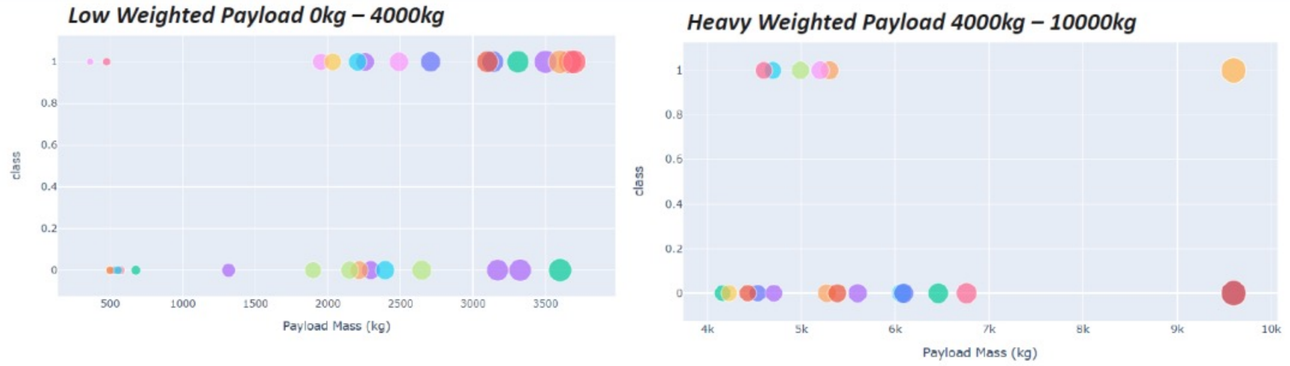
DASHBOARD WITH PLOTLY DASH



We can see that KSC LC-39A had the most successful launches from all the sites



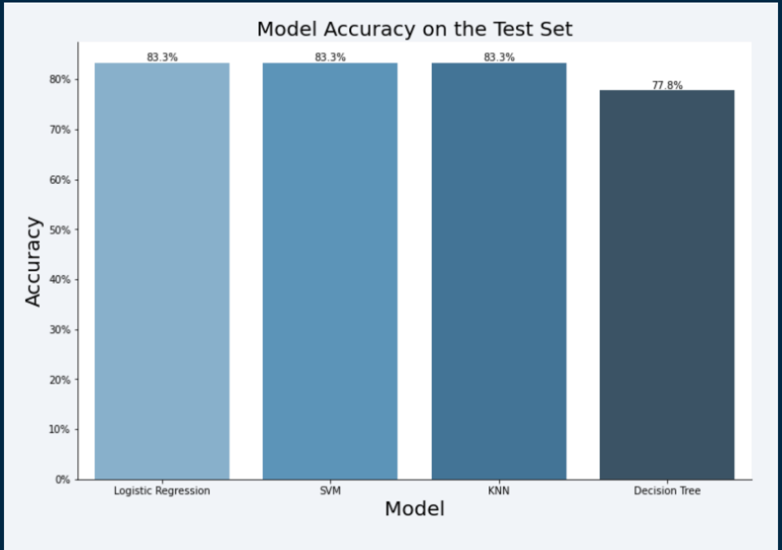
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

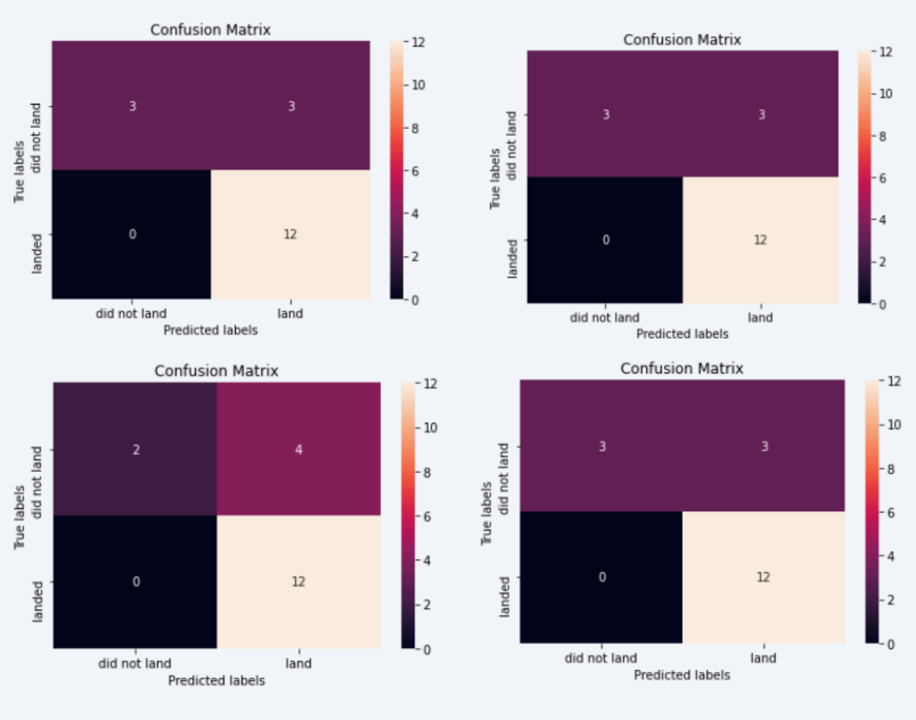


We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

PREDICTIVE ANALYSIS

The SVM, KNN, and Logistic Regression model achieved the highest accuracy at 83.3%, while the SVM performs the best in terms of Area Under the Curve at 0.958.





RESULTS

- The SVM, KNN, and Logistic Regression models are the best in terms of prediction accuracy for this dataset.
- Low weighted payloads perform better than the heavier payloads.
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches.
- KSC LC 39A had the most successful launches from all the sites.
- Orbit GEO,HEO,SSO,ES L1 has the best Success Rate.

CONCLUSIONS

- The SVM, KNN, and Logistic Regression models are the best in terms of prediction accuracy for this dataset.
- Low weighted payloads perform better than the heavier payloads.
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches.
- KSC LC 39A had the most successful launches from all the sites.
- Orbit GEO,HEO,SSO,ES L1 has the best Success Rate

THANKS