IBM Data Science Capstone Project – Space X

Full Project URL

Ayoub Ouakani 28/01/2022

Executive Summary	• Presentation of the entire process and life cycle of the project
Introduction	• Explaining the tasks
Methodology	• Technics and méthodes used to work with the data
Insights Drawn from EDA	• Local insights related to lunch sites using EDA
Lunch Sites Proximities Analysis	• Studying the georgraphy of the lunching sites with folium
Predictive Analysis	• Selecting the right Classifier to fit the task
Final Global Insights	• Concluding the accurate results

Executive Summary

- Summary of methodologies :

- Data collection
- Data wrangling
- EDA with data visualization
- EDA with SQL
- Building an interactive map with Folium
- Predictive analysis (Classification)

- Summary of all results :

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

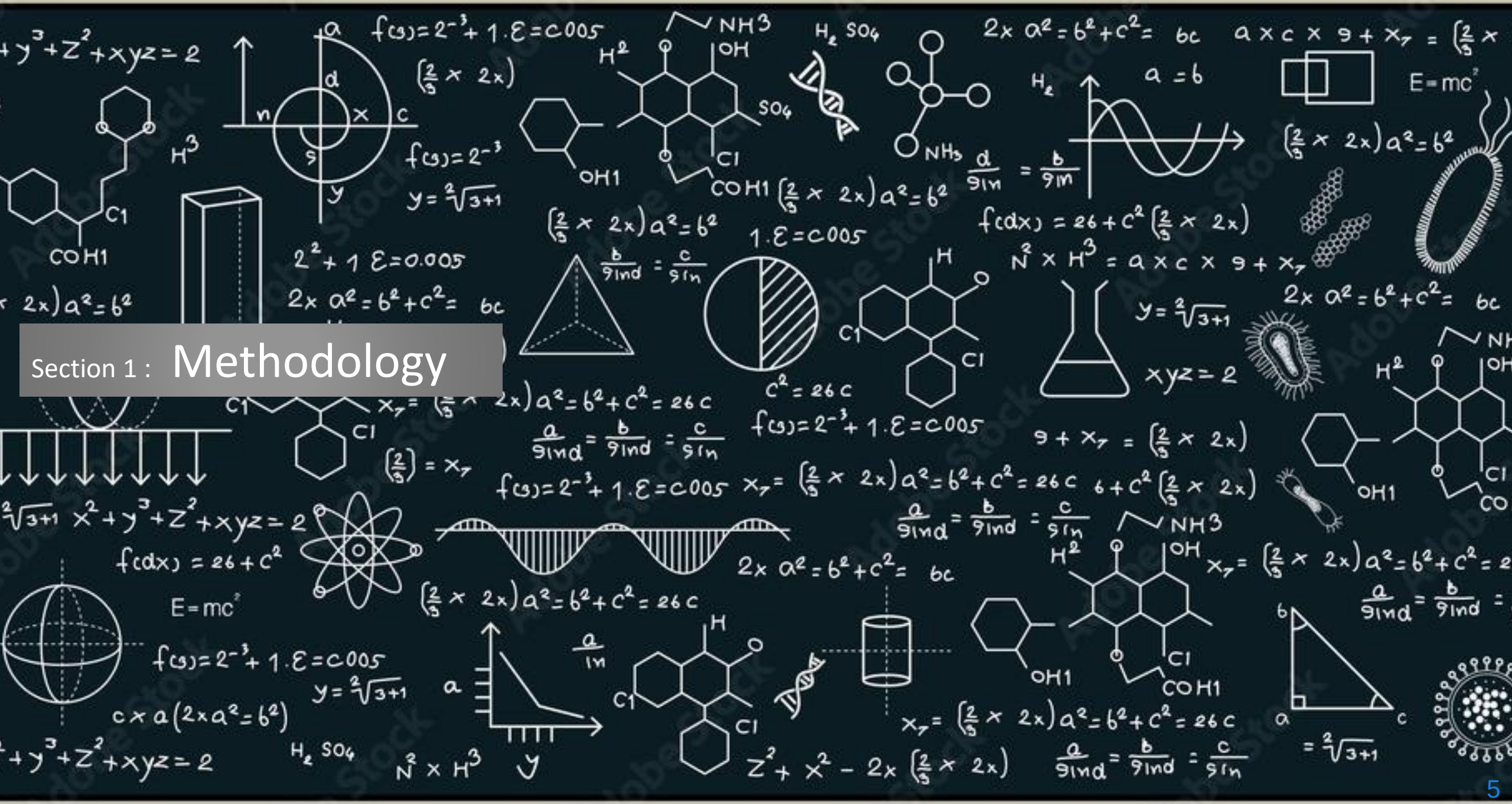
Introduction

• Project background and context :

We predicted if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

• Common problems that needed solving :

- What variables influences if the rocket will land successfully?
- The correlation between certain rocket variables will impact in determining the class rate of a successful landing.
- What are the optimal conditions does SpaceX need to achieve the best results and ensure the best rocket success landing rate.



Section 1 : Methodology

Methodology

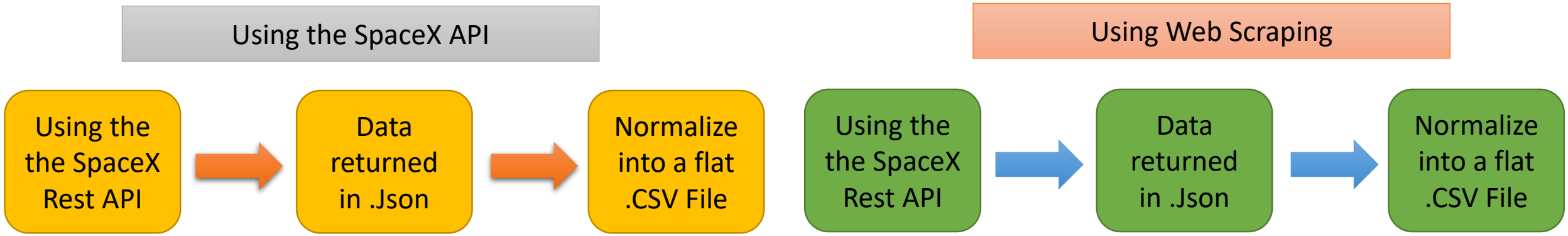
Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - (Web Scraping) from wikipedia
- Perform data wrangling
  - One Hot Encoding for catégorical Data and dropping irrelevant variables
- Perform exploratory data analysis (EDA) using visualization and SQL:
  - Plotting : Scatter Graphs, Bar Graphs to show relationships between variables to show patterns of data.
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection Road Map

The following datasets were collected by the following methods :

- We used SpaceX launch open to public data that is gathered through the REST API.
- The API delivered to us data about launches, the reusability of the rocket, payload carried, launching sites specifications, landing specifications, and landing class (outcome)
- Our goal is to use this data to establish a predictive model about whether the SpaceX falcon 9 rocket will land or not
- Another method through which data was obtained for the Falcon 9 Launch data is web scraping (done on Wikipedia) using the python library BeautifulSoup.



Data Collection – SpaceX API

Getting Response From API :

```
In [28]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [29]: response = requests.get(spacex_url)
```

Convert .JSON Response to a PD Dataframe:

```
In [34]: # Use json_normalize meethod to convert the json result into a dataframe
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

Clean Data with Customized Fonctions + Filter to get only Falcon9 :

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

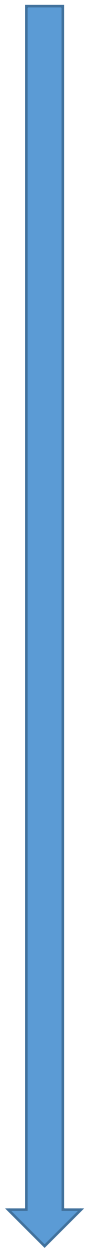
```
In [58]: # Hint data['BoosterVersion']!='Falcon 1'
data_launch['BoosterVersion' ] != 'Falcon 9'
data_falcon9 = data_launch
data_falcon9.shape
```

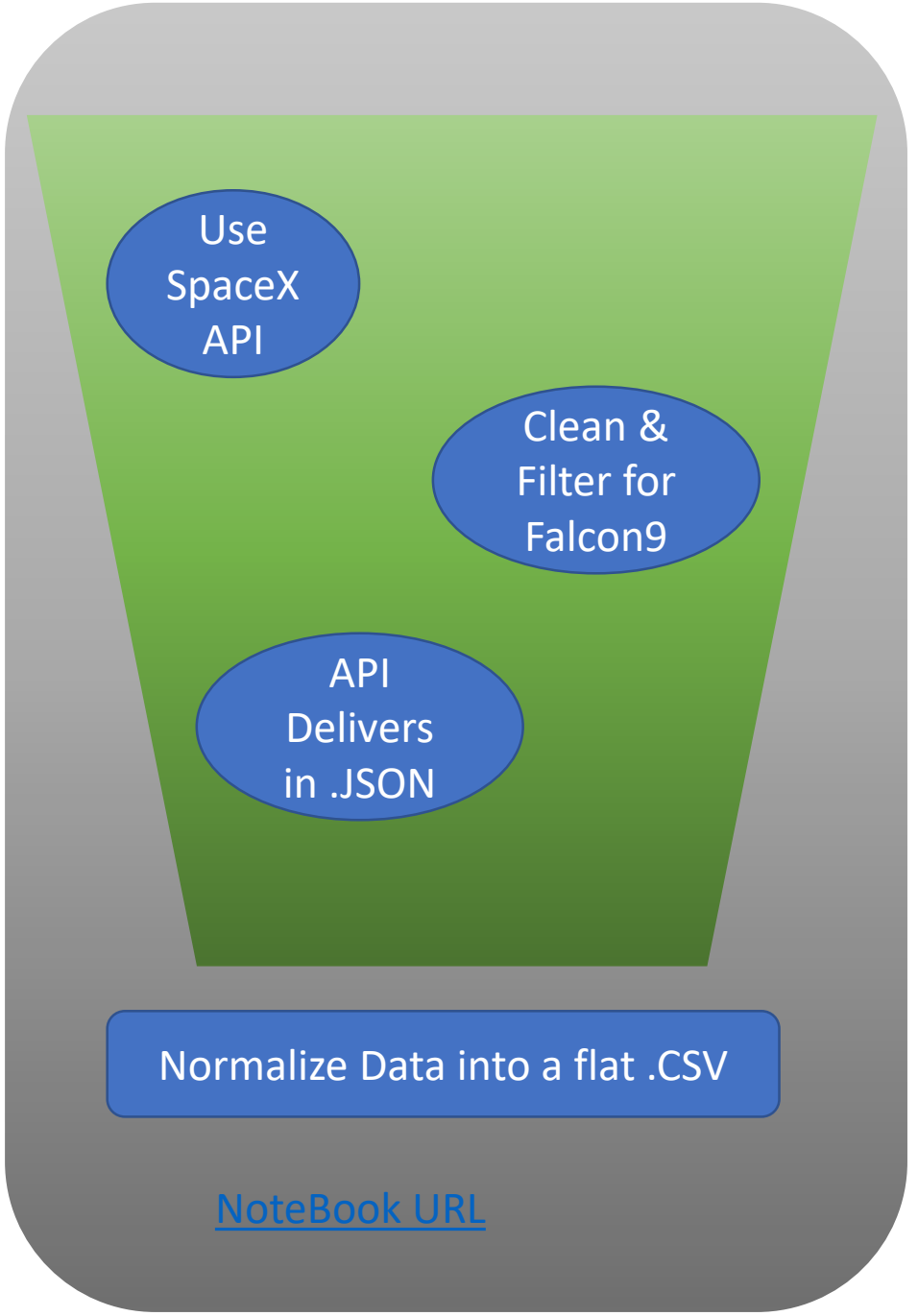
```
Out[58]: (94, 17)
```

Replace Empty Value with mean + Export :

```
# Calculate the mean value of PayloadMass column
mean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].fillna(mean)
```

Export .CSV





Data Collection – Scraping

Getting Response From HTML :

```
In [24]: # use requests.get() method with the provided static_url
         response = requests.get(static_url)
```

From Beautiful Soup Object to Finding Tables:

```
In [40]: # Use BeautifulSoup() to create a BeautifulSoup object
         soup = BeautifulSoup(response.text, "html.parser")
```

```
In [42]: # Use the find_all function in the BeautifulSoup object
         html_tables = soup.find_all('table')
```

Getting Columns Names and append Data to keys (Notebook Bloc 15) :

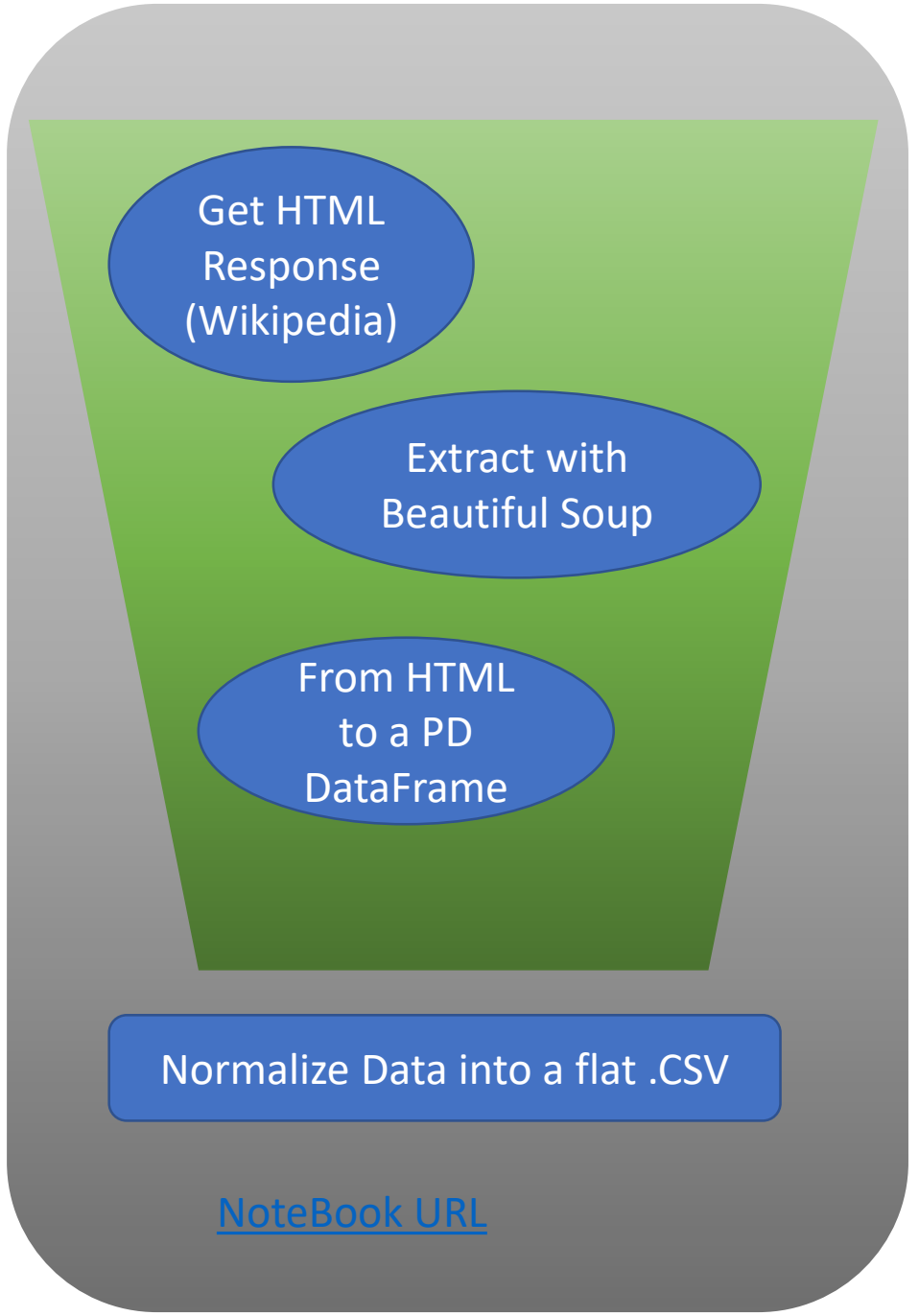
```
In [11]: column_names = []
         column_names = []
         temp = soup.find_all('th')
         for x in range(len(temp)):
             try:
              name = extract_column_from_header(temp[x])
              if (name is not None and len(name) > 0):
                 column_names.append(name)
             except:
              pass
```

```
In [15]: 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
```

From Dictionary to DataFrame to .CSV :

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

```
In [17]: 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, True 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.
In this lab we will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

Process

Performing Exploratory Data Analysis (EDA) on the Dataset

Calculate number of lunches per site

Calculate the nuber and frequency per orbit

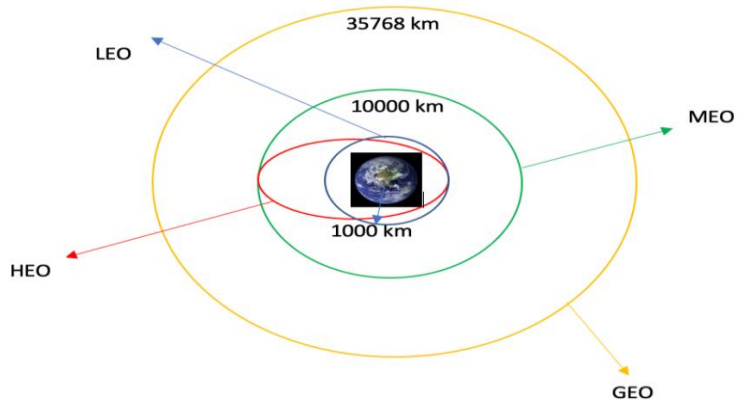
Number and frequency of outcome per orbit

Export in .CSV

Labeling Outcome from Outcome Column

Succes rate per landing

Each Lunch is meant for a certain Orbit



EDA with Data Visualization

Scatter Plots Drawn :

Flight Number VS. Payload Mass

Flight Number VS. Launch Site

Payload VS. Launch Site

Orbit VS. Flight Number

Payload VS. Orbit Type

Orbit VS. Payload Mass

(Scatter Plot explain the correlation between two variables)

Bar Graph Drawn :

Sucess rate Vs Orbit

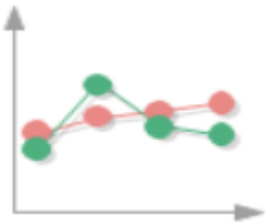
(A bar diagram makes it easy to compare sets of data between different groups at a glance. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes. Bar charts can also show big changes in data over time.

Line Graph Drawn :

Success rate VS Year

(Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded)

NoteBook URL



EDA with SQL

Performed SQL queries to gather information about the dataset.

For example of some questions we were asked about the data we needed information about. Which 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 ‘CCA’
- 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 ground pad was achieved.
- Listing the names of the boosters which have success in drone ship 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. (Subquery)
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

NoteBook URL

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 dataframe launch_outcomes(failures, successes) to classes 0 and 1 with Green and Red markers on the map in a MarkerCluster()

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 distance to landmarks

Example of some trends in which the Launch Site is situated in.

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

NoteBook URL

Build a Dashboard with Plotly Dash

The dashboard is built with Flask and Dash web framework.

Graphs

- Pie Chart showing the total launches by a certain site/all sites
- display relative proportions of multiple classes of data.
- size of the circle can be made proportional to the total

quantity it represents

Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions

- It shows the relationship between two variables.
- It is the best method to show you a non-linear pattern.
- The range of data flow, i.e. maximum and minimum value, can be determined.
- Observation and reading are straightforward.

Predictive Analysis (Classification)

BUILDING MODEL:

- Load our dataset into NumPy and Pandas
- Transform Data + Standarize
- 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 (SVM, Decision Tree, KNN, Logistic Regression)
- 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

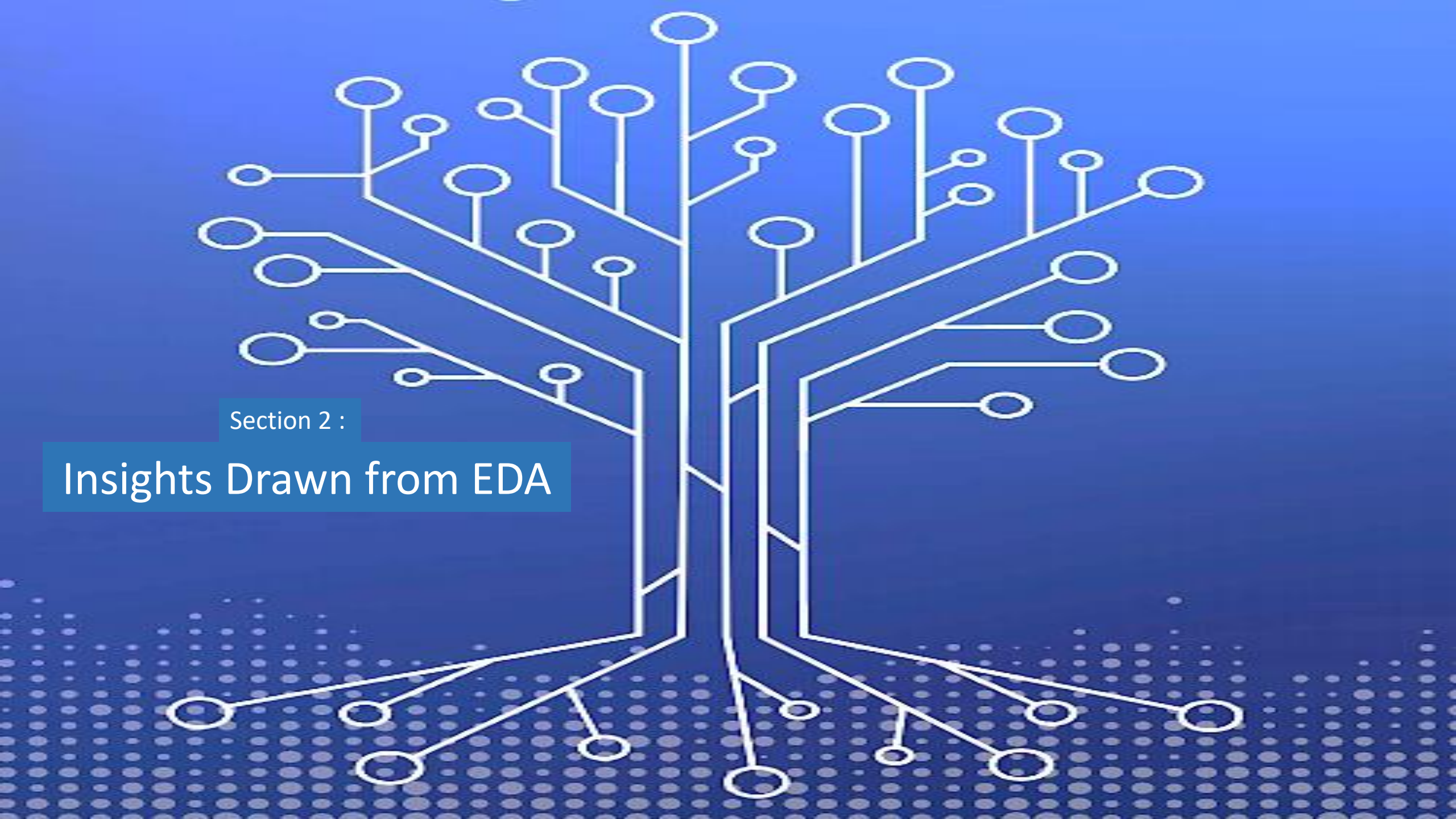
FINDING THE BEST PERFORMING CLASSIFICATION MODEL

- 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.

NoteBook URL

Results

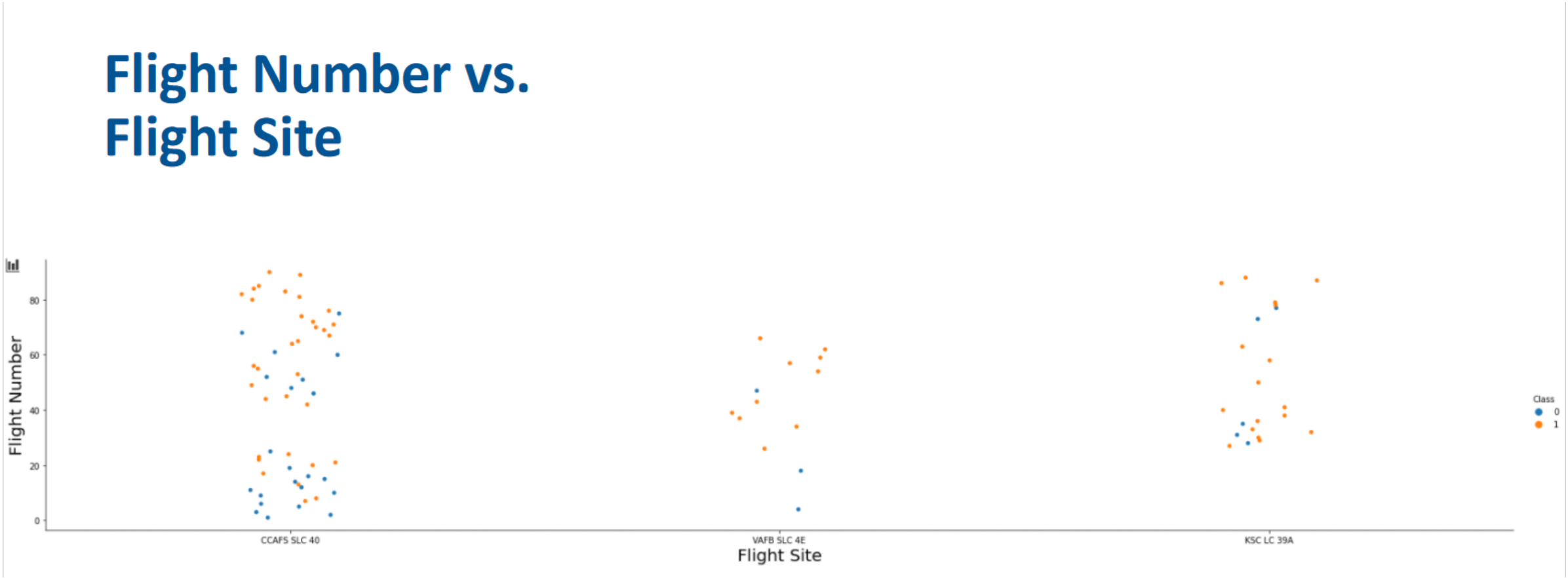
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2 :

Insights Drawn from EDA

Flight Number vs. Launch Site



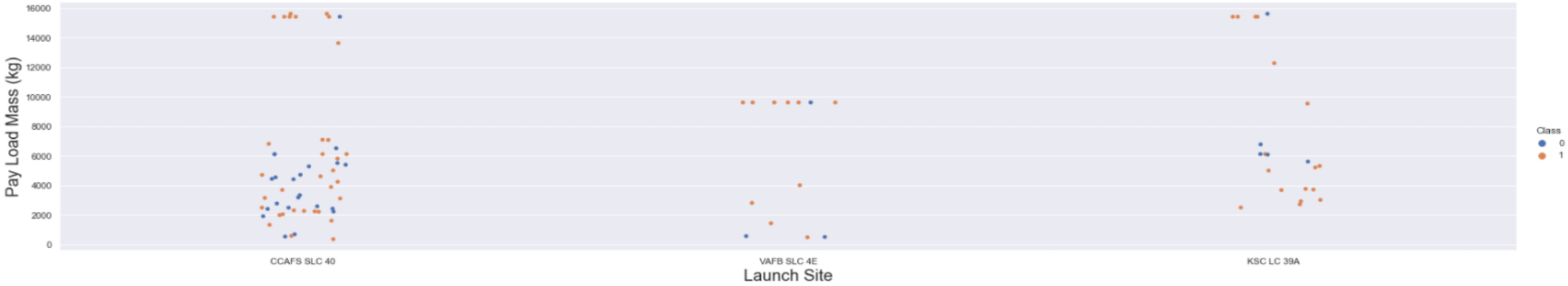
Conclusion :

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

NoteBook URL

Payload vs. Launch Site

Payload Mass vs. Launch Site

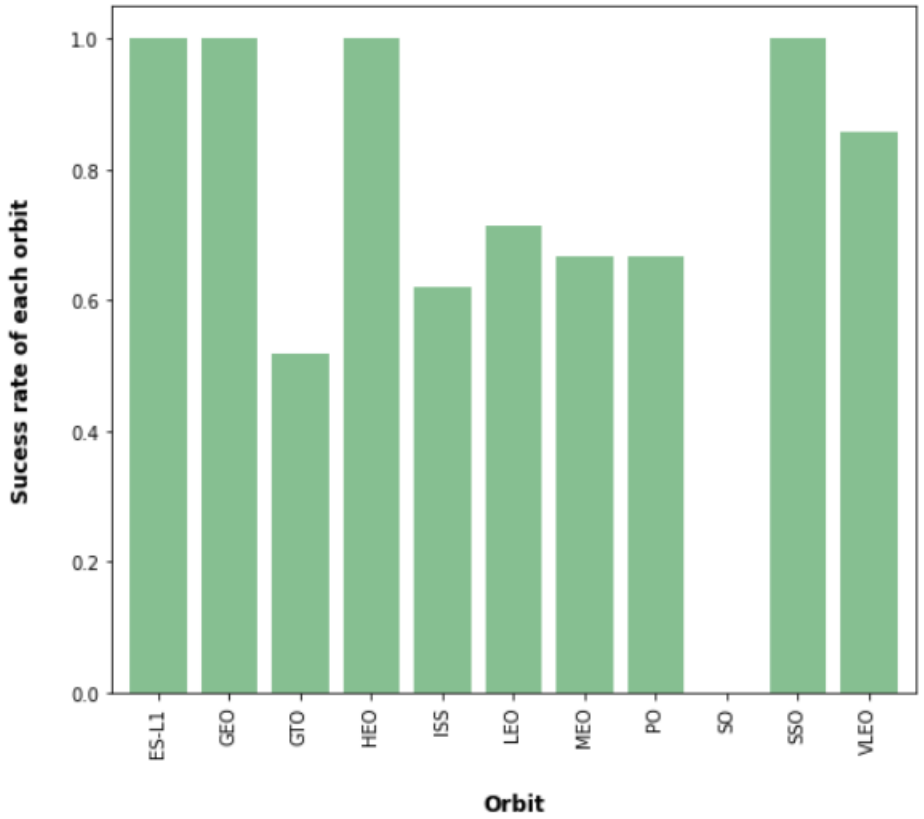


Conclusion :

There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependant on Pay Load Mass for a success launch.

NoteBook URL

Success Rate vs. Orbit Type

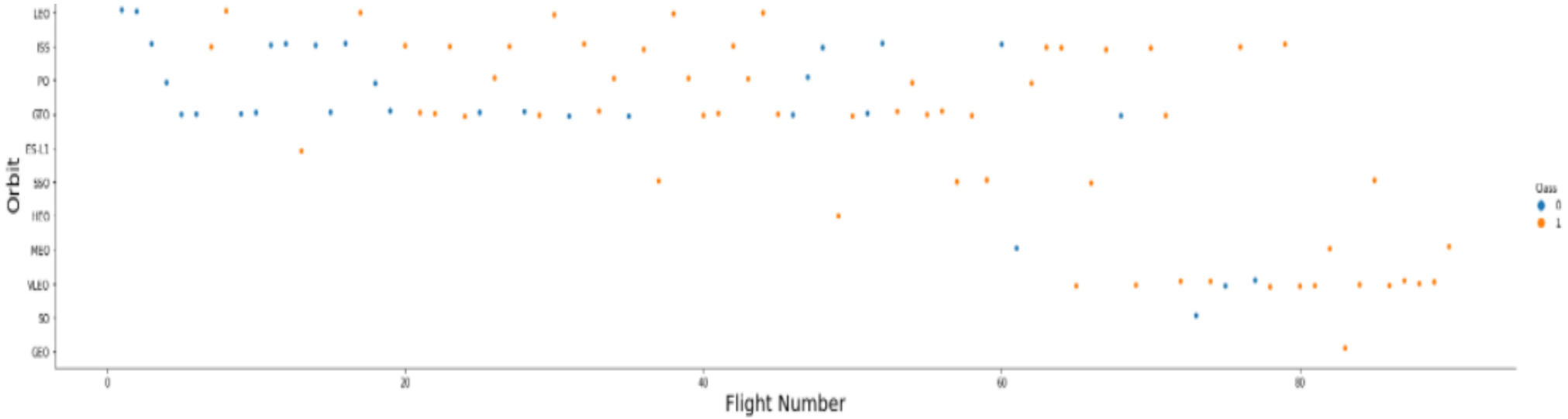


Conclusion :

Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

NoteBook URL

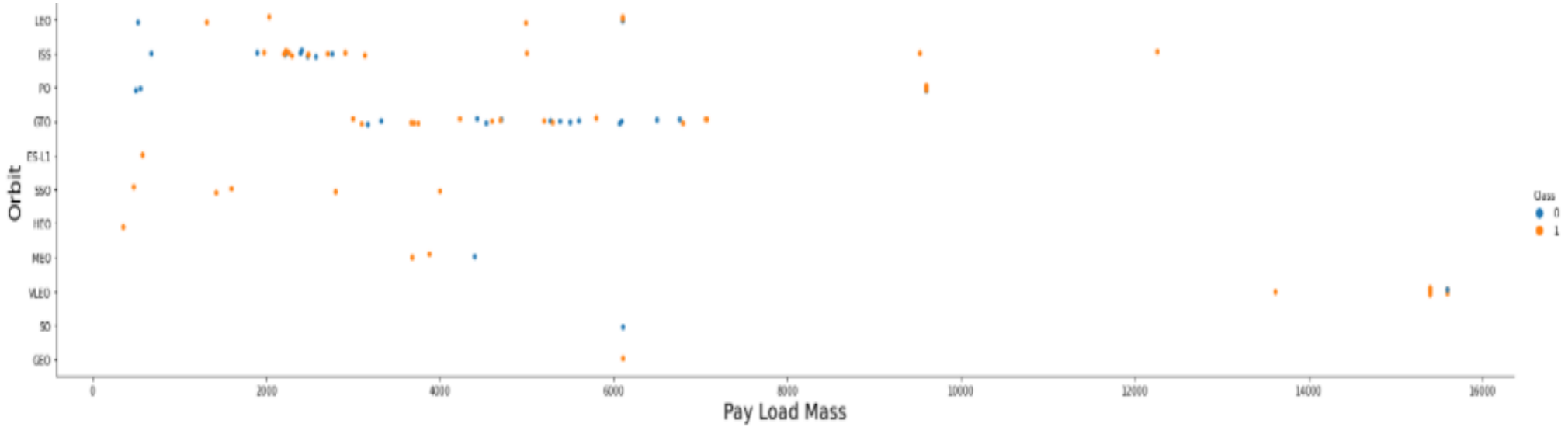
Flight Number vs. Orbit Type



Conclusion :
You Can see that for Leo Orbit the more flights are taken the better the class becomes
but the the rest there seem to be no correlation between the success of flights and the number of flights

NoteBook URL

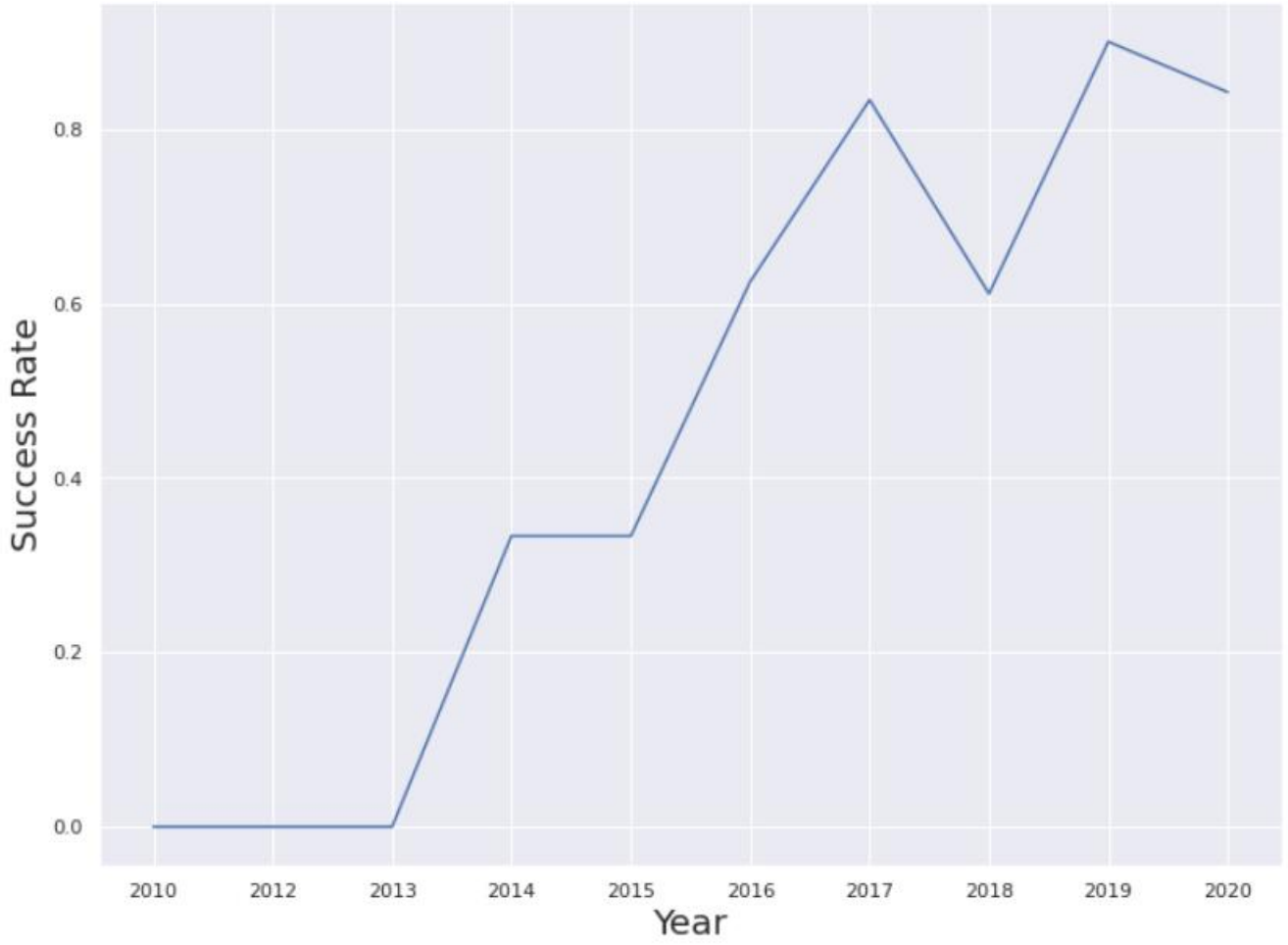
Payload vs. Orbit Type



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

NoteBook URL

Launch Success Yearly Trend



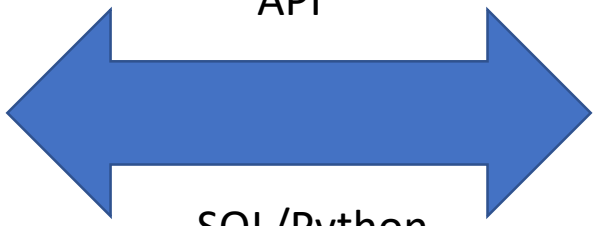
Conclusion :

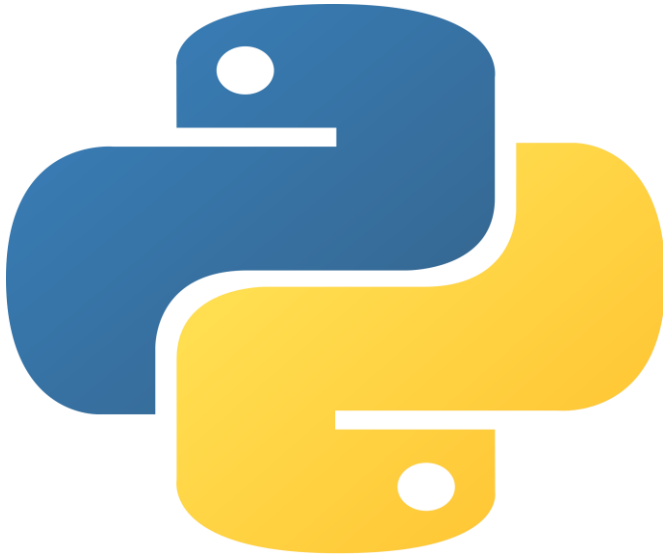
It appears that the success rate is increating with every year, except on 2018 but it got back up again later

NoteBook URL

EDA with SQL

DB2 DataBase on Cloud





Python Notebook on IBM Cloud Park for Data

NoteBook URL

All Launch Site Names

SQL QUERY (ScreenShot) :

```
In [21]: %sql select DISTINCT launch_site from SPACEX;
 * ibm_db_sa://mhg49466:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

Out[21]:

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 the table SpaceX

NoteBook URL

Launch Site Names Begin with 'CCA'

SQL QUERY (ScreenShot) :

```
In [24]: %sql select * from SPACEX where launch_site like 'CCA%' LIMIT 5
 * ibm_db_sa://mhg49466:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

Out[24]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

QUERY Explination :

Using the word LIMIT 5 in the query means that it will only show 5 records from the top of SpaceX table and LIKE keyword has a wild card with the words ‘CCA%’ the percentage in the end suggests that the Launch_Site name must start with CCA.

NoteBook URL

Total Payload Mass

SQL QUERY (ScreenShot) :

```
In [25]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEX where customer = 'NASA (CRS)'
 * ibm_db_sa://mhg49466:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
Out[25]:
```

1
45596

QUERY Explination :

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 NASA (CRS)

NoteBook URL

Average Payload Mass by F9 v1.1

SQL QUERY (ScreenShot) :

```
In [26]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEX where booster_version like 'F9 v1.1%'
 * ibm_db_sa://mhg49466:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
Out[26]:
```

1
2534

QUERY Explanation :

Using the function AVG works out the average in the column PAYLOAD_MASS_KG_
The WHERE clause filters the dataset to only perform calculations on Booster_version F9 v1.1

NoteBook URL

First Successful Ground Landing Date

SQL QUERY (ScreenShot) :

```
In [28]: %sql select min(date) from SPACEX where landing__outcome = 'Success (ground pad)'
 * ibm_db_sa://mhg49466:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
Out[28]:
1
2015-12-22
```

QUERY Explanation :

Using the function MIN works out the minimum date in the column Date
The WHERE clause filters the dataset to only perform
calculations on Landing_Outcome Success (drone ship)

NoteBook URL

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL QUERY (ScreenShot) :

```
In [30]: %sql select booster_version from SPACEX where landing__outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
 * ibm_db_sa://mhg49466:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

Out[30]:

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
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_ > 4000 AND Payload_MASS_KG_ < 6000

NoteBook URL

Total Number of Successful and Failure Mission Outcomes

SQL QUERY (ScreenShot) :

```
%sql select mission_outcome, count(*) as COUNT from SPACEX group by mission_outcome
```

```
 * ibm_db_sa://mhg49466:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

2]:

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

QUERY Explanation :

??

NoteBook URL

Boosters Carried Maximum Payload

SQL QUERY (ScreenShot) :

```
In [37]: %sql select DISTINCT booster_version, payload_mass__kg_ from SPACEX where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEX) ORDER BY payload_mass__kg_ DESC
 * ibm_db_sa://mhg49466:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

Out[37]:

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

QUERY Explanation :

Using the word DISTINCT in the query means that it will only show Unique values in the Booster_Version column from the table SpaceX

NoteBook URL

2015 Launch Records

SQL QUERY (ScreenShot) :

```
In [44]: %sql Select landing__outcome, booster_version, launch_site from SPACEX where date like '2015%' and landing__outcome = 'Failure (drone ship)'
 * ibm_db_sa://mhg49466:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

Out[44]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

QUERY Explanation :
We Used the percentage symbole wild cart to retrieve all date starting with 2015
And operator adds a consecutive query to the initial Where Condition

NoteBook URL

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

SQL QUERY (ScreenShot) :

```
In [46]: %sql select landing__outcome, count(*) as COUNT from (select * from SPACEX where date between '2010-06-04' and '2017-03-20') group by landing__outcome order by COUNT desc
```

```
 * ibm_db_sa://mhg49466:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

Out[46]:

landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

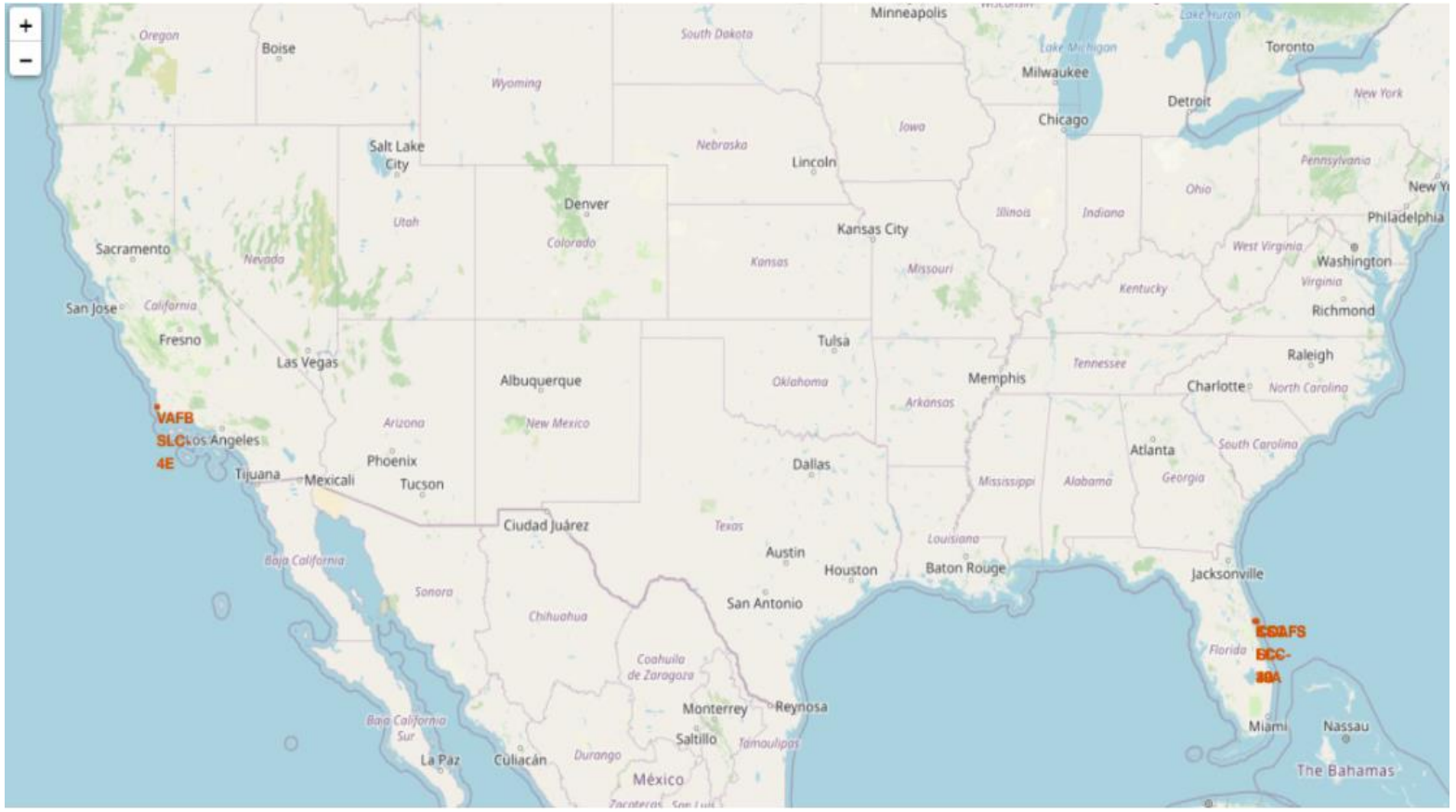
QUERY Explanation :
Function COUNT counts records in column
WHERE filters data
LIKE (wildcard)
AND (conditions)
Group by (Grouping)
Order By (Ordering)

NoteBook URL

Section 3 :

Lunch Sites Proximities Analysis

All launch sites global map markers

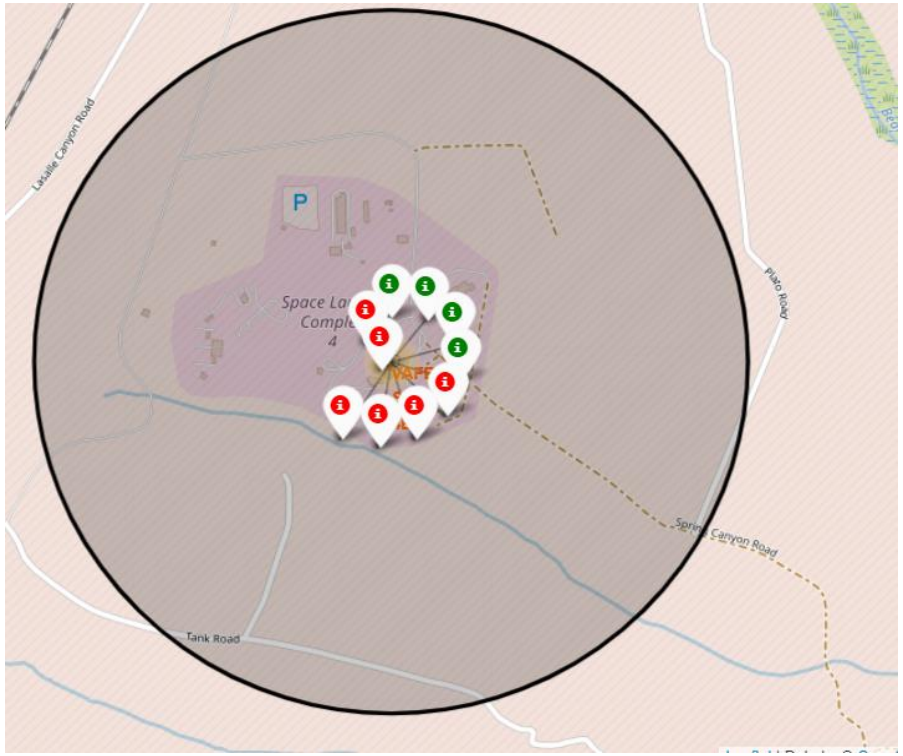


We clearly can observe that the lunching sites are distributed on the Florida east coast and california's west coast

NoteBook URL

Colour Labelled Markers

California's Lunching sites



Florida's Lunching sites

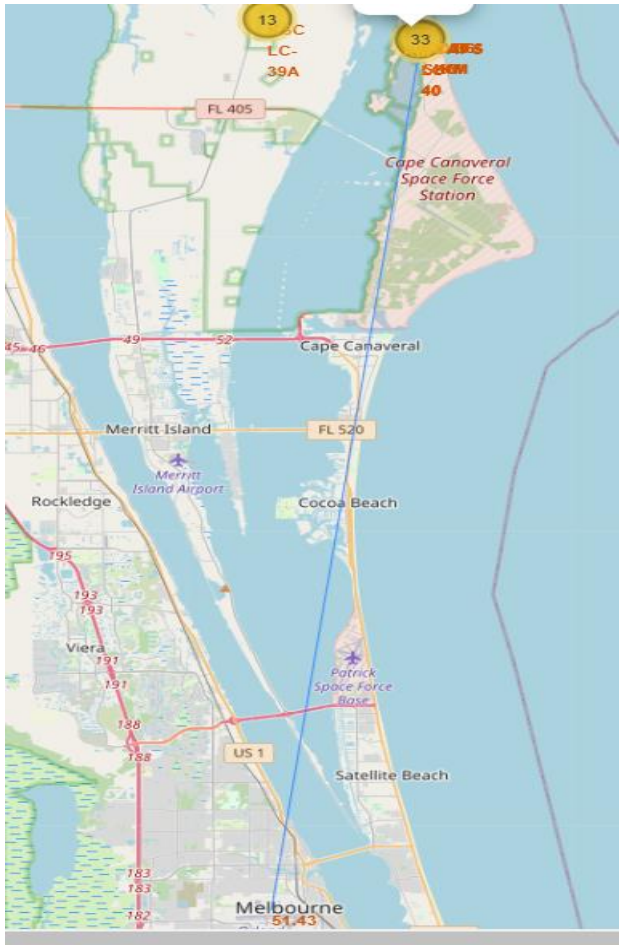
Green Markers show succesful lunches

Red Markers show failed lunches

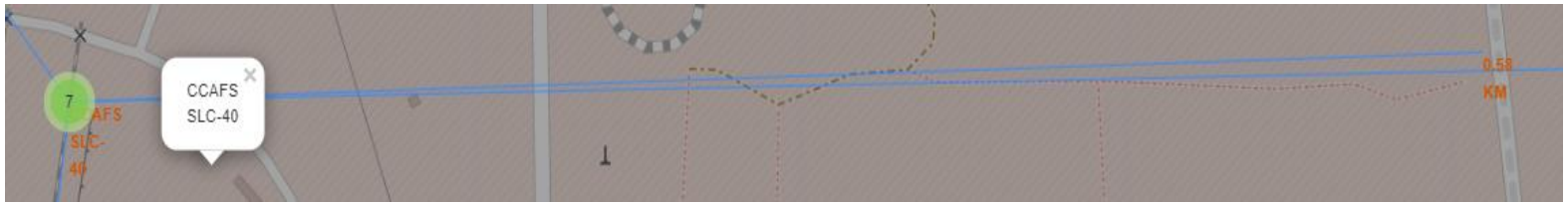
NoteBook URL

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

Distance to City 51,43 Km

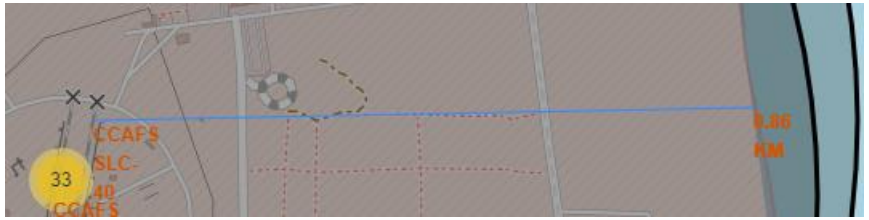


Distance to Highway 0,58 Km



Distance to Railway 1,28 Km

Distance to Coast 0,86 Km



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

NoteBook URL

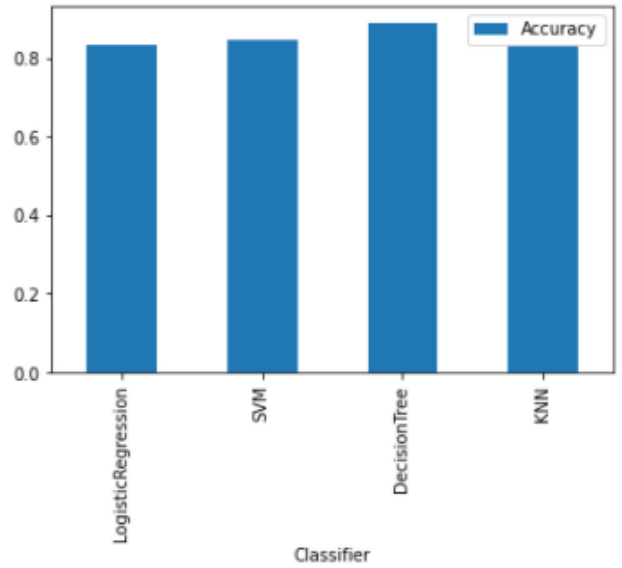
Section 4 :

Predictive Analysis

Classification Accuracy

```
In [43]: ax = df_barchart.plot.bar(x='Classifier', y='Accuracy')
         ax
```

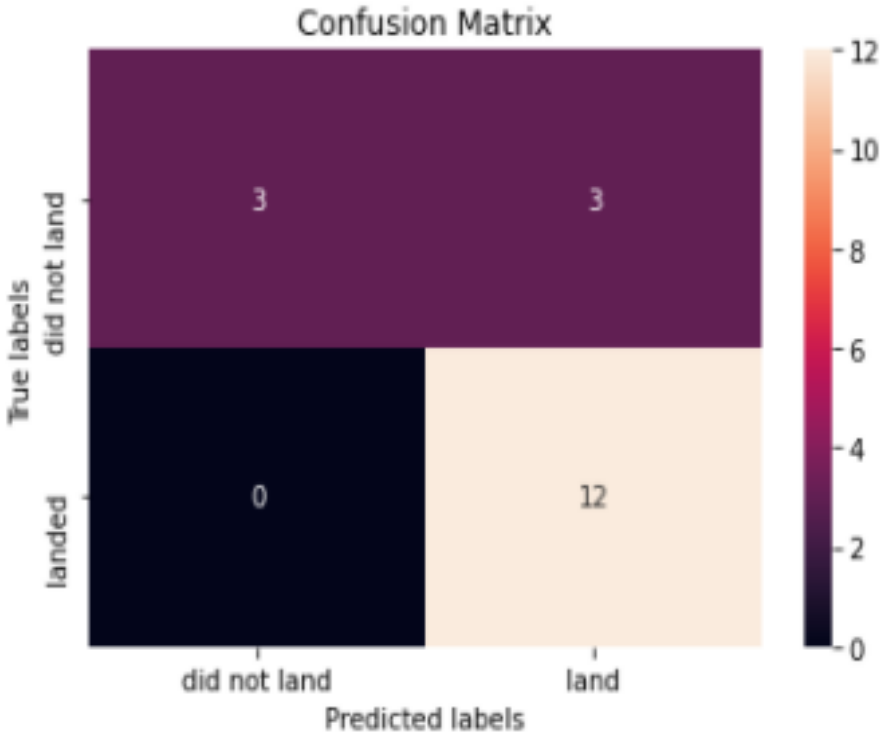
```
Out[43]: <AxesSubplot:xlabel='Classifier'>
```



It is obvious that the Decistion Tree model has the highest accuracy of (0.8892) wich fits the data better

NoteBook URL

Confusion Matrix



The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.

NoteBook URL

Conclusions

Insights :

- Point 1 : It appears that the success rate is positively correlated with the number of flights experimented
- Point 2 : after the year of 2013 the success rate got better with each flight
- Point 3 : Certain lunching sites give the best success rate : ES-L1, GEO, HEO, SSO, VLEO
- Point 4 : The decision tree model fits the data better and appears to be the best model to use for this certain task

Thank you!