Presenation on Applied Data Science Capstone

Aishwarya Chennadi

12-Apr

OUTLINE

- Executive Summary
- Introduction
- Data Collection and Data Wrangling
- Exploratory Data Analysis and Interative visual analytics
- Predictive Analysis
- EDA with Visualization Results
- EDA with SQL Results
- Interactive Map with Folium
- Plotly with Dash
- Conclusion

EXECUTIVE SUMMARY

In this presentation we will discuss about Data collection, data wrangling, EDA and its test results performed using SpaceXTable datasheet on Jupyter lab, and generated graphs using Folium and seaborn. We will also concentrate on what is predictive analysis and dashboard generated using plotly dash.

INTRODUCTION

We will focusing on generating results for easy data visualization and data analysis using Plotly, Seaborn, SQL, Dash using some real time datasets executed on labs environment on Jupyter lab.

DASHBOARD

https://github.com/AishwaryaChennadi/DataScience/tree/main

Data Collection and Data Wrangling

Data Collection:

- Data Collection is nothing but collecting data.
- We will use URL to target a specific endpoint of the API to get required data.
- We need to perform a get request using the requests library, which we will use to get the data from the API. This result can be viewed by calling the .json() method.
- Response returned will be in the form of a JSON, specifically a list of JSON objects.
- To convert this JSON to a dataframe, we can use the json_normalize function.

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

- Normalize function will allow us to “normalize” the structured json data into a flat table.

Data Wrangling:

- Data Wrangling is the process of gathering, collecting, and transforming Raw data into another format for better understanding, decision-making, accessing, and analysis in less time. Data Wrangling is also known as Data Munging.

Exploratory Data Analysis and Interative visual analytics

EDA :

- Exploratory Data Analysis is the first step of any data science project.
- This process involves studying, exploring, and visualizing information to derive important insights.
- To find patterns, trends, and relationships in the data, it makes use of statistical tools and visualizations.

Interactive Visual Analytics :

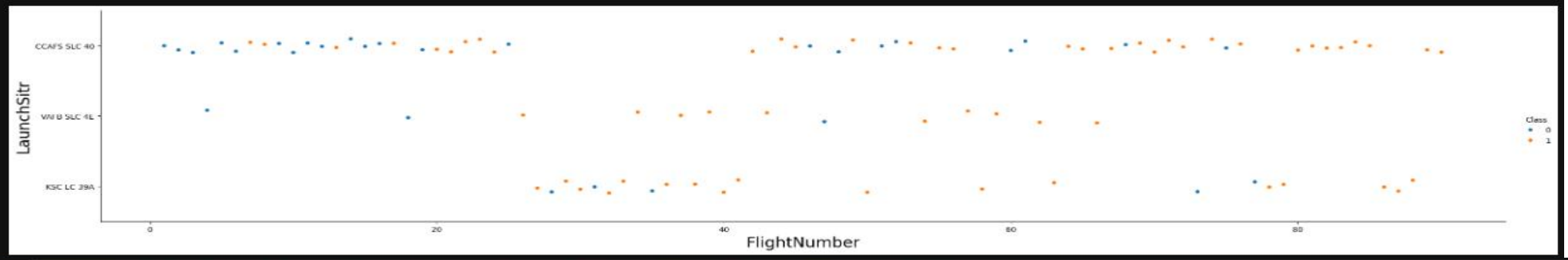
- We use this to build a Dashboard for stakeholders.
- Interactive visual analytics enables users to explore and manipulate data in an interactive and real-time way.
- Common interactions including zoom-in and zoom-out, pan, filter, search, and link.
- With interactive visual analytics, users could find visual patterns faster and more effectively.
- Instead of presenting your findings in static graphs, interactive data visualization, or dashboarding, can always tell a more appealing story.
- We can use **Folium** and **Plotly Dash** to build an interactive map and dashboard to perform interactive visual analytics.

Predictive Analysis

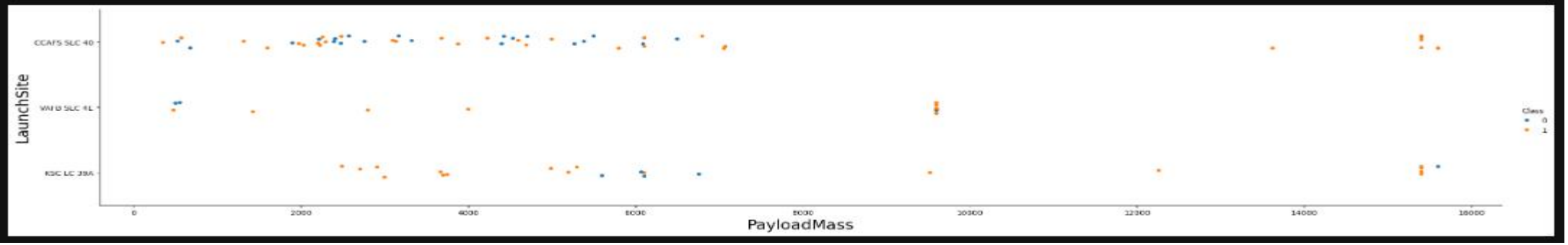
- This involves Preprocessing, allowing us to standardize our data, and Train_test_split, allowing us to split our data into training and testing data.
- We will determine the model with the best accuracy using the training data.
- We will test Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors.

EDA with Visualization Results

[5]:
```
### TASK 1: Visualize the relationship between Flight Number and Launch Site
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("LaunchSitr",fontsize=20)
plt.show()
```

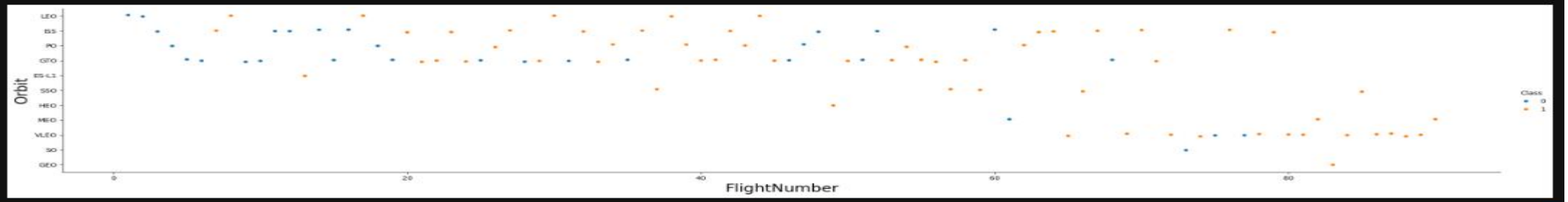


[8]:
```
### TASK 2: Visualize the relationship between Payload and Launch Site
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



EDA with Visualization Results

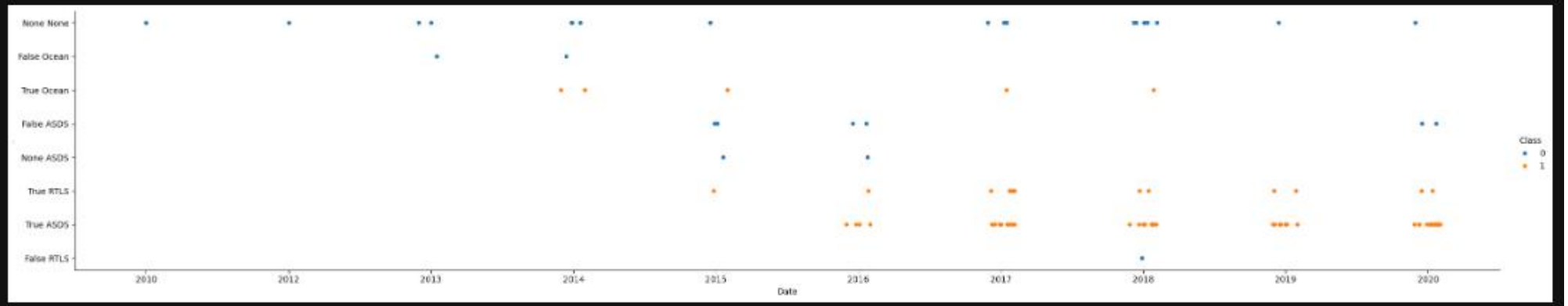
[10]:
```
### TASK  4: Visualize the relationship between FlightNumber and Orbit type
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



[11]:
```
### TASK  5: Visualize the relationship between Payload and Orbit type
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

EDA with Visualization Results

[23]:
```
### TASK  6: Visualize the launch success yearly trend
year=[]
def Extract_year():
    for i in df["Date"]:
        year.append(i.split("-")[0])
    return year
Extract_year()
df['Date'] = year
sns.catplot(y="Outcome", x="Date", hue="Class", data=df, aspect = 5)
plt.xlabel("Date",fontsize=10)
plt.ylabel("Outcome",fontsize=1)
plt.show()
```



You can plot a line chart with x axis to be `Year` and y axis to be average success rate, to get the average launch success trend.

EDA with SQL Results

Note: If the column names are in mixed case enclose it in double quotes For Example "Landing_Outcome"

Task 1

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

[15]:
```
%sql Select DISTINCT Landing_Outcome from SPACEXTABLE
```

```
 * sqlite:///my_data1.db
Done.
```

[15]:

Landing_Outcome
Failure (parachute)
No attempt
Uncontrolled (ocean)
Controlled (ocean)
Failure (drone ship)
Precluded (drone ship)
Success (ground pad)
Success (drone ship)
Success
Failure

Task 2

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

```
%sql Select Launch_Site from SPACEXTABLE where Launch_Site LIKE "CCA%" LIMIT 5
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

EDA with SQL Results

Task 3

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

```
[33]: %sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer="NASA (CRS)"
```

```
 * sqlite:///my_data1.db
Done.
```

[33]:

SUM(PAYLOAD_MASS__KG_)
45596

Task 4

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

```
[36]: %sql Select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = "F9 v1.1"
```

```
 * sqlite:///my_data1.db
Done.
```

[36]:

AVG(PAYLOAD_MASS__KG_)

Task 5

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

Hint:Use min function

```
[38]: %sql Select MIN(DATE) from SPACEXTABLE where Mission_Outcome="Success"
```

```
 * sqlite:///my_data1.db
Done.
```

[38]:

MIN(DATE)
2010-06-04

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

```
[54]: %sql Select Booster_Version from SPACEXTABLE where Mission_Outcome = "Success" AND (PAYLOAD_MASS__KG_ >= "4000") AND (PAYLOAD_MASS__KG_ <= "6000")
```

```
 * sqlite:///my_data1.db
Done.
```

[54]:

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

EDA with SQL Results

Task 7

List the total number of successful and failure mission outcomes

```
[63]: %sql Select Count(*) from SPACEXTABLE where Mission_Outcome = "Success" OR Mission_Outcome = "Failure"
 * sqlite:///my_data1.db
Done.
```

[63]:

Count(*)
98

Task 8

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

```
[77]: %sql Select Booster_Version from SPACEXTABLE WHERE PAYLOAD_MASS__KG_ IN (Select MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
 * sqlite:///my_data1.db
Done.
```

[77]:

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

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.

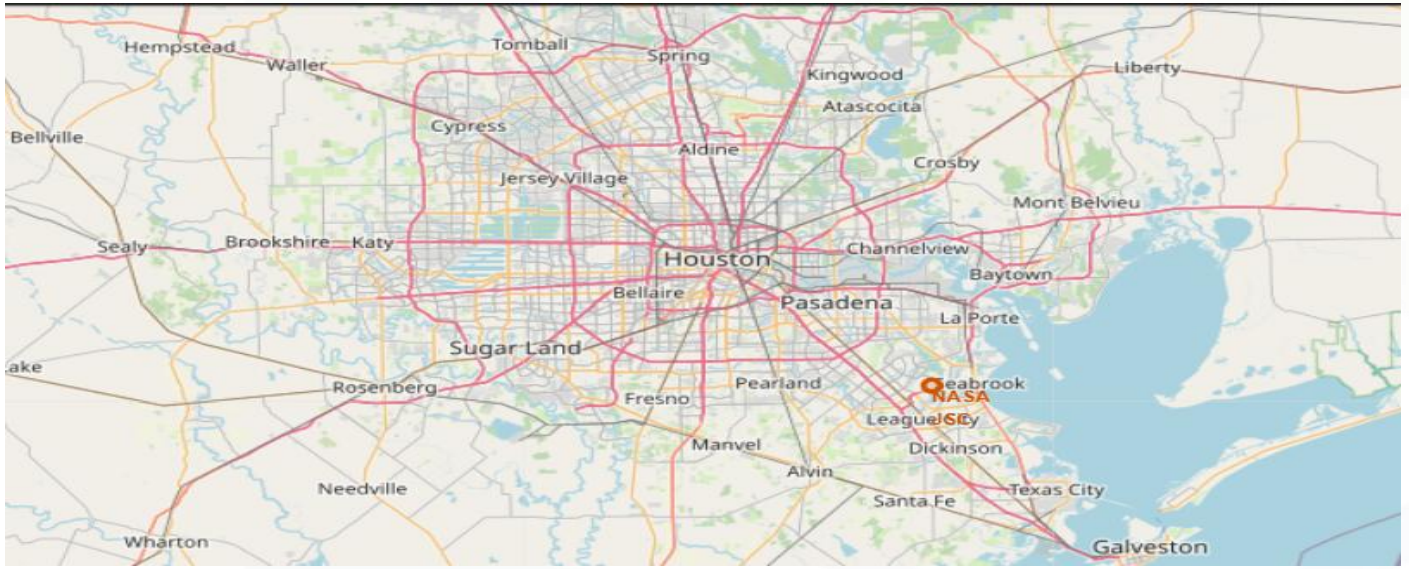
```
[65]: %sql Select count(*) from SPACEXTABLE where Landing_Outcome="Failure (drone ship)" OR Landing_Outcome = "Success(ground pad)" AND Date >= "2016
 * sqlite:///my_data1.db
Done.
```

[65]:

count(*)
5

Interactive Map with Folium

```
[17]: ## Task 1: Mark all launch sites on a map
      import folium
      import pandas as pd
      from folium.plugins import MarkerCluster
      marker_cluster = MarkerCluster().add_to(map)
      marker_cluster = MarkerCluster().add_child(map)
      for loc in locations:
          folium.Marker(location=loc["location"],
                        popup=loc["popup"]).add_to(marker_cluster)
```



PlotLy Dash Results

```
from js import fetch
import io

URL1 = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv"
resp1 = await fetch(URL1)
text1 = io.BytesIO((await resp1.arrayBuffer()).to_py())
data = pd.read_csv(text1)
```

```
data.head()
```

```
URL2 = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_3.csv'
resp2 = await fetch(URL2)
text2 = io.BytesIO((await resp2.arrayBuffer()).to_py())
X = pd.read_csv(text2)
```

CONCLUSION

In this presentation, we discussed about Data collecting, data wrangling, EDA, Interactive visual analysis, Predictive analysis. We are concluding this presentation by theorithal overview and handson results captured from the tab.

THANK YOU