

Python Basics:

How to use Python for data analysis a) Pairplots b) Heatmaps & Co-relation matrices

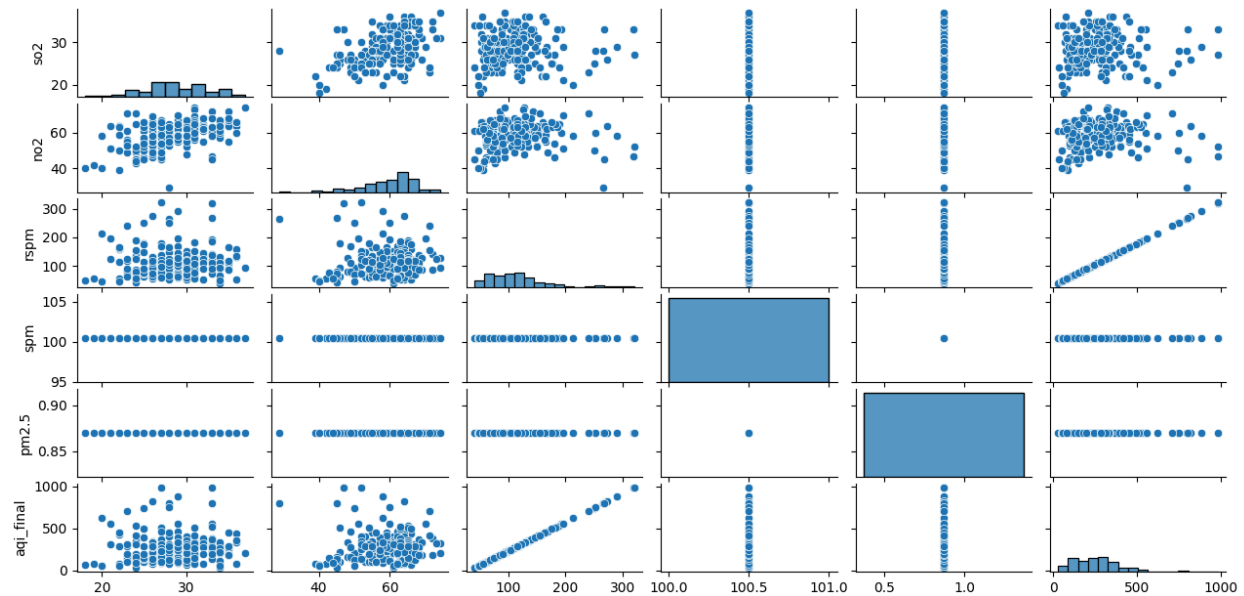
We will try to understand Python examples for dataset analysis. For that we will perform pairplotting, co-relation mapping on the dataset & also get the heatmaps.

Pair plots are utilized for visualizing relationships between multiple variables in a dataset. They offer a comprehensive overview of the data by creating a grid of plots, where each subplot represents the relationship between two variables. We have used the Air-Quality dataset in order to analyse the amount of pollutants and find their association with the corresponding AQI-level.

Simple Pair-plot in Python

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_excel('AQI_Final.xlsx')
data = data.dropna()
sns.pairplot(data)
plt.show()
correlation_matrix = data.corr()
sns.heatmap(
    correlation_matrix,
    annot=True,
    cmap='coolwarm',
    fmt=".2f",
    linewidths=.5
)
X = data[['so2', 'no2', 'rspm', 'spm', 'aqi_final']]
plt.figure(figsize=(8, 6))
plt.title('Correlation Heatmap of pollutants')
plt.show()
```



Here, in this example we have tried to plot the pollutants (SO₂, NO₂, RSPM, SPM, PM_{2.5}) & also the Final AQI where we realize there's similarity between plots of pollutant "RSPM & final_aqi" thus assessing the possibility of a numerical dependency/ relationship with each other.

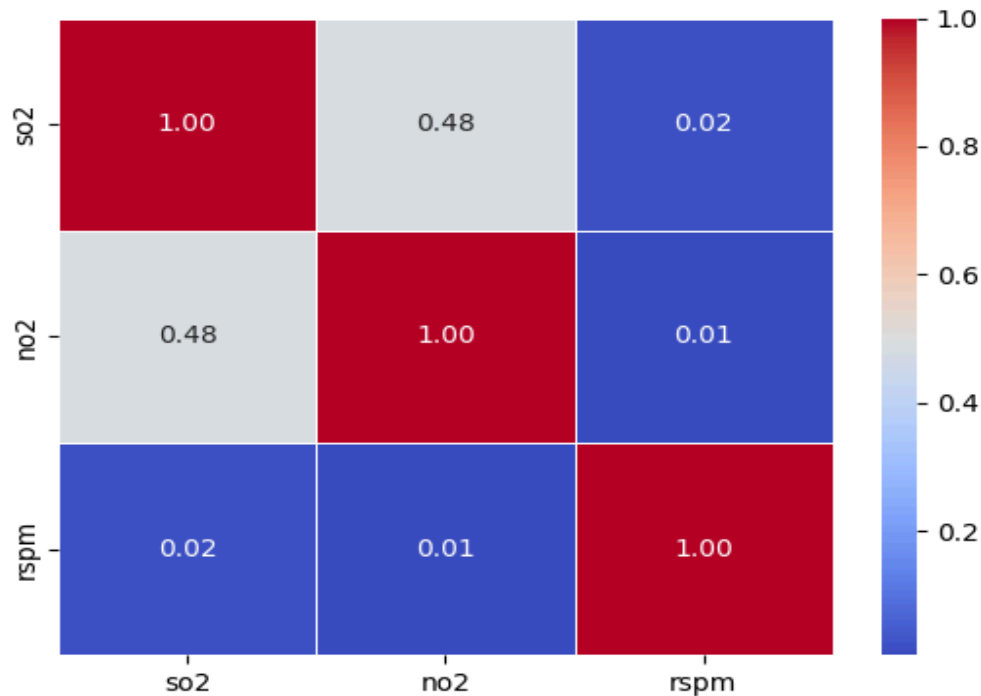
We use correlation analysis next in order to understand the strength and direction of the relationship between variables, and a correlation heatmap to visualize these relationships in a dataset.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

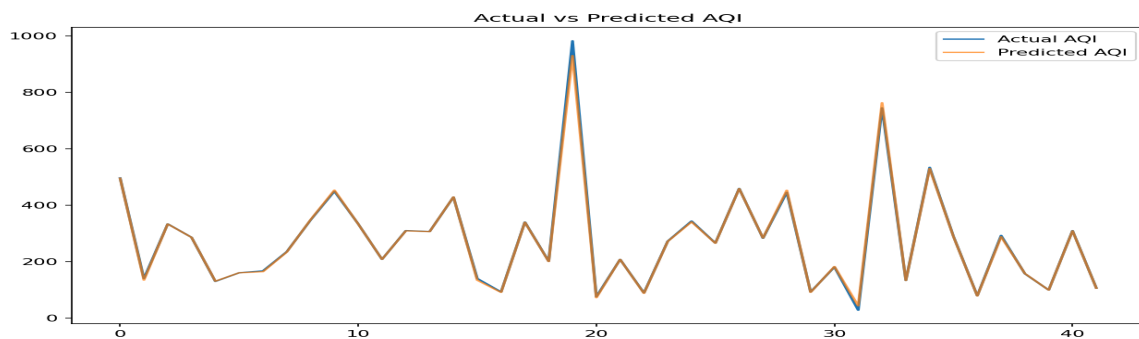
data = pd.read_excel('AQI_Modified.xlsx')
data = data.dropna()
correlation_matrix = data.corr()
sns.heatmap(
    correlation_matrix,
    annot=True,
    cmap='coolwarm',
    fmt=".2f",
    linewidths=.5
)
X = data[['so2', 'no2', 'rspm']]
plt.figure(figsize=(8, 6))
plt.title('Correlation Heatmap of pollutants')
plt.show()
```

Co-relation

Heatmaps make it easy to spot patterns, identify highly correlated or redundant features, and quickly assess multicollinearity, which is crucial for data cleaning and building more effective models.



Regression: Linear regression is a statistical method used to predict a continuous dependent variable i.e target variable based on one or more independent variables it assumes a linear relationship between the dependent & independent variables which means the dependent variable changes proportionally with changes in the independent variables. Here, AQI_Final/ Actual AQI/ Predicted AQI is the dependent variable which has a direct relation with the independent variables (pollutant concentrations SO2/NO2/RSPM/SPM/PM2.5).



```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

data = pd.read_excel('AQI_Final.xlsx')
X = data[['so2', 'no2', 'rspm', 'spm', 'pm2.5']]
y = data['aqi_final']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print("Mean Absolute Error:", mean_absolute_error(y_test, y_pred))
print("Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("R2 Score:", r2_score(y_test, y_pred))
plt.figure(figsize=(10, 6))
plt.plot(y_test.values, label='Actual AQI')
plt.plot(y_pred, label='Predicted AQI', alpha=0.7)
plt.title('Actual vs Predicted AQI')
plt.legend()
plt.show()

```

By using Regression we check whether for the given dataset we are able to predict the AQI given a set of samples we calculate the Mean Absolute Error (MAE), Mean Squared Error (MSE) & the R2 score. The golden plot overlaps with the blue one indicating that for the dataset the actual values versus predicted are equivalent.

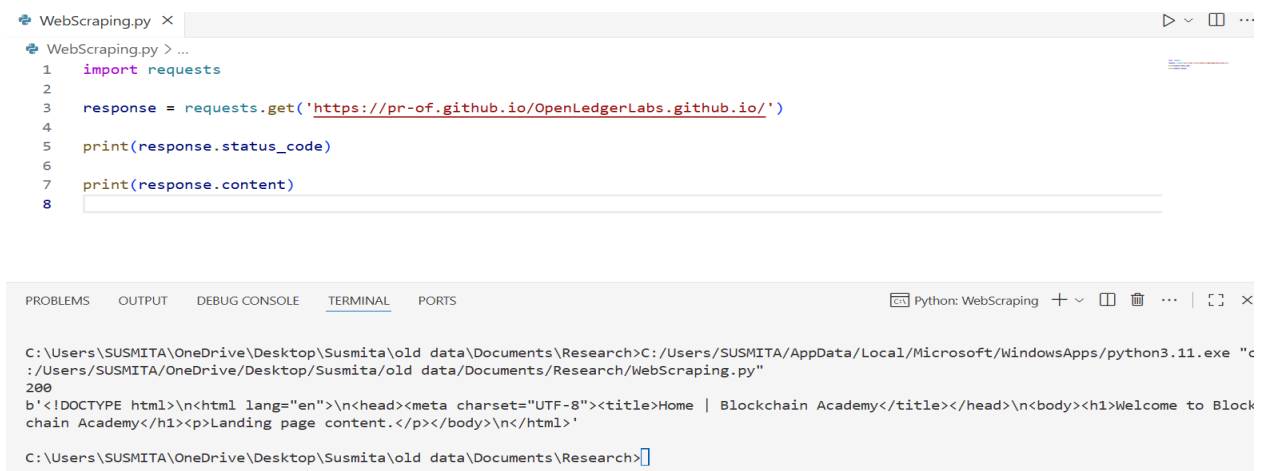
```

C:\Users\SUSMITA\OneDrive\Desktop\Susmita\old data\Documents\Research>C:/Users/SUSMITA/AppData/Local/Microsoft/WindowsApps/
python3.11.exe "c:/Users/SUSMITA/OneDrive/Desktop/Susmita/old data/Documents/Research/airpollutionRegression.py"
Mean Absolute Error: 3.5262857142856543
Mean Squared Error: 84.1346158095215
R2 Score: 0.9975176429301885

```

Python Intermediate:

Web Scraping is required to automate the time-consuming and tedious process of manually extracting data from websites, converting unstructured web content into structured, usable information. This enables businesses and individuals to collect large volumes of data for various purposes, such as market research, competitive analysis, price monitoring, and lead generation, leading to informed decisions and improved efficiency. It is also crucial for fueling AI model training, brand monitoring, and gathering data for journalism and research.



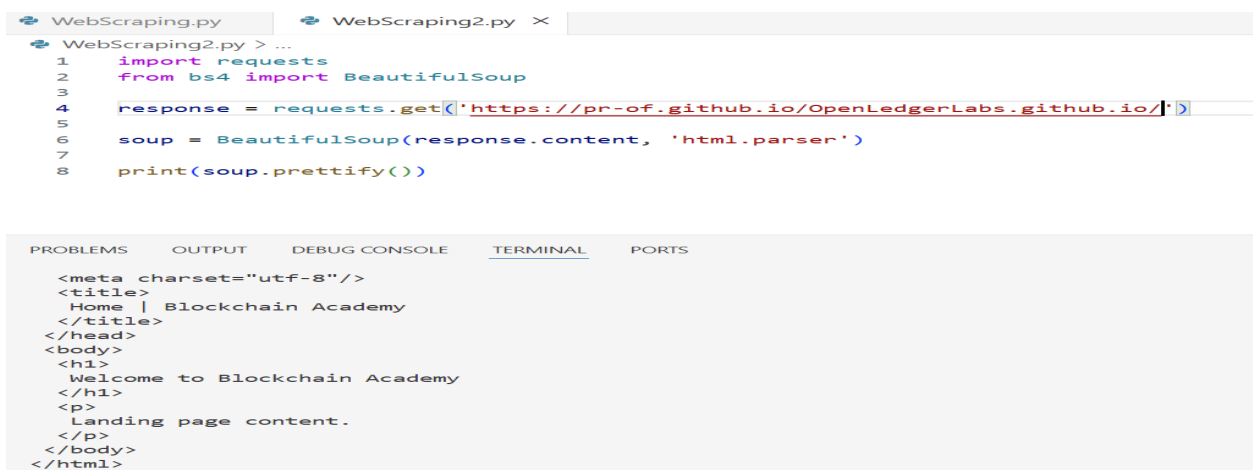
The screenshot shows a Python IDE with a file named 'WebScraping.py'. The code in the editor is as follows:

```
1 import requests
2
3 response = requests.get('https://pr-of.github.io/OpenLedgerLabs.github.io/')
4
5 print(response.status_code)
6
7 print(response.content)
8
```

The terminal output at the bottom shows the execution of the script:

```
C:\Users\SUSMITA\OneDrive\Desktop\Susmita\old_data\Documents\Research>C:/Users/SUSMITA/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/SUSMITA/OneDrive/Desktop/Susmita/old_data/Documents/Research/WebScraping.py"
200
b'<!DOCTYPE html>\n<html lang="en">\n<head><meta charset="UTF-8"><title>Home | Blockchain Academy</title></head>\n<body><h1>Welcome to Blockchain Academy</h1><p>Landing page content.</p></body>\n</html>'
```

Web Scraping using BeautifulSoup



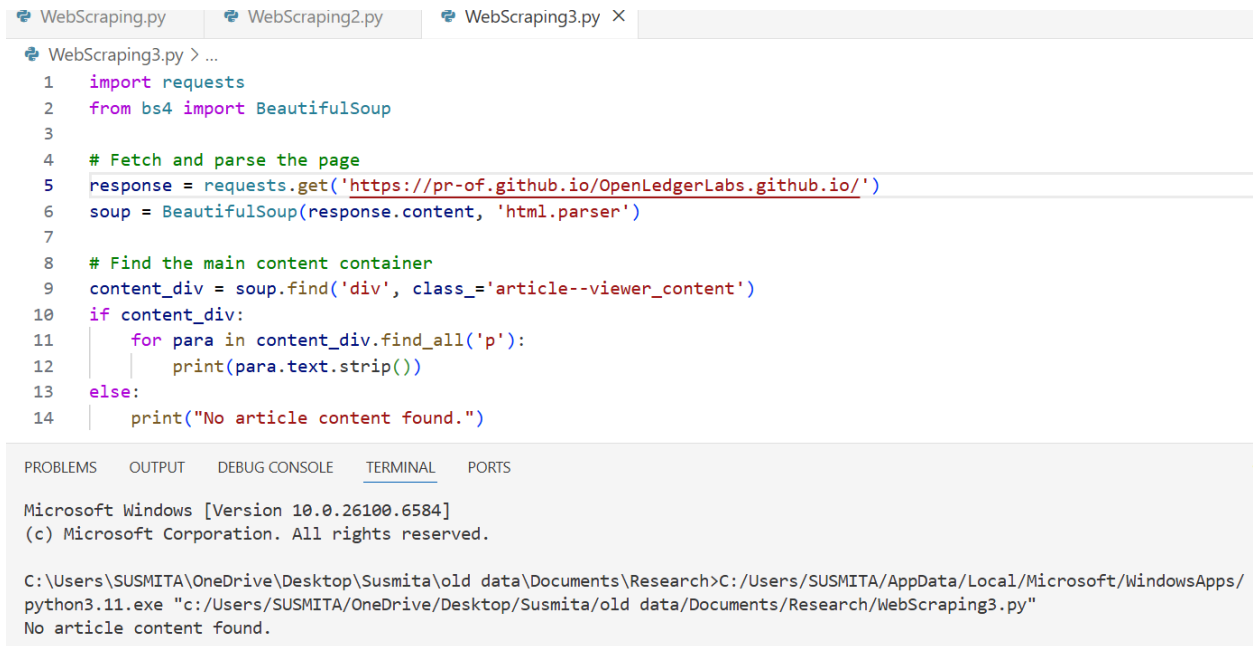
The screenshot shows a Python IDE with two files: 'WebScraping.py' and 'WebScraping2.py'. The code in 'WebScraping2.py' is as follows:

```
1 import requests
2 from bs4 import BeautifulSoup
3
4 response = requests.get('https://pr-of.github.io/OpenLedgerLabs.github.io/')
5
6 soup = BeautifulSoup(response.content, 'html.parser')
7
8 print(soup.prettify())
```

The terminal output at the bottom shows the execution of the script, displaying the HTML content in a prettified format:

```
<meta charset="utf-8"/>
<title>
Home | Blockchain Academy
</title>
</head>
<body>
<h1>
Welcome to Blockchain Academy
</h1>
<p>
Landing page content.
</p>
</body>
</html>
```

Web Scraping using Selenium as a WebDriver



```
WebScraping.py WebScraping2.py WebScraping3.py X
WebScraping3.py > ...
1 import requests
2 from bs4 import BeautifulSoup
3
4 # Fetch and parse the page
5 response = requests.get('https://pr-of.github.io/OpenLedgerLabs.github.io/')
6 soup = BeautifulSoup(response.content, 'html.parser')
7
8 # Find the main content container
9 content_div = soup.find('div', class_='article--viewer_content')
10 if content_div:
11     for para in content_div.find_all('p'):
12         print(para.text.strip())
13 else:
14     print("No article content found.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SUSMITA\OneDrive\Desktop\Susmita\old data\Documents\Research>C:/Users/SUSMITA/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/SUSMITA/OneDrive/Desktop/Susmita/old data/Documents/Research/WebScraping3.py"
No article content found.

Python Projects (working on)

From the futuristic point of view, after performing analysis; its possible to create/ formulate a model which can help us to forecast or predict in the future as to what could be the AQI-level.