`

Journal Report on CRISP-DM Stage -2 for Data Mining and Machine Learning CC7184

Student ID: 22048235 **Rahul Mehta**

Title: ***Examining the joblessness patterns in London and UK***

Abstract

The analysis of quarterly data of UK and London aims at apply data mining methodologies of 30 years from the year 1992 to 2022. The dataset comprises of various variables which constitute the London Unemployment data in terms of percentage and in thousands (000s), UK unemployment data in terms of % and in thousands (000s) and Percentage gap of UK and London Data along with the quarterly data from year 1992 to 2022. The data set is cleansed by removing redundant and inconsistent values. The primary objective of cleaning data is to obtain the fruitful and accurate information which would finally help in achieving the desired results.

An ample number of data mining techniques have been implemented to get the desired result for instance regression, decision trees and clustering. The result too focuses on the 2008 world economic crisis due to severe recession. The research discovers the patterns which could be helpful for future understanding of reducing unemployment.

Introduction

Literature Review:

An examination of previous research, reports, and other sources of data and information on the topic of unemployment in the area is required for a literature study of London and UK unemployment stats. The goal of the literature review is to find patterns, developments, gaps, and hinders in the body of current research and to offer novel ideas that can direct additional research or governance. The research further emphasises how complicated and complex unemployment exists, and the way it can have a significant negative social, economic, and emotional effect on individuals as well as communities.

Although certain investigations point out the potential positive effects of proactive labour market measures, such as training, education, and creation of employment initiatives, and others maintain that long-term unemployment can result in social exclusion, destitution, and psychological disorders. There are also going on disparities in unemployment rates amongst certain social groups and regional destinations, with residents of particular regions, particularly young people, women, ethnic minorities, and those who are female, being more likely to be unemployed than those who are male.

Therefore, the examination of the literature on London and UK data on unemployment presents a rich and complex perspective on the dynamics of job markets in that region and points out its importance of evidence-based policymaking to address unemployment challenges and foster equitable and environmentally friendly growth.

Methods:

A variety of data on jobs and layoffs in the UK have been published by the Office for National Statistics. This data provides statistics on the number of people with jobs, the number of persons lacking jobs, and the unemployment rate. A quarterly survey of the labour force, or the LFS, is used for collecting the data. The Labour Force Study (LFS) is a household survey that captures data on people's employment status, occupations, and hours worked. The ONS additionally publishes data on job openings and other labour force factors

`

Main Findings:

Unemployment rates have fluctuated all through time in London and the UK nationally. The UK experienced a period of prosperity and declining rates of unemployment between 2010 and 2020. The COVID-19 pandemic in 2020, however, contributed for an enormous spike in employment in the UK, with the unemployment rate reaching its highest point of 5.1% in January 2021. In general, London has had a higher unemployment rate than the UK average, which is indicative of the more diverse and cutthroat nature of the labour market in the capital city. London's jobless rate was greater than the UK average in the years before to the epidemic but had been cautiously dropping.

The city's serve market, however, was severely affected by the pandemic, and its jobless rate increased considerably in 2020. The deviations in rates of unemployment by demographic classifications, involving a higher percentage of unemployed people among adolescents, those from ethnic minorities, and those with less schooling.

Methodology

Cleaning of information, the organization, and the growth into a format suitable for analysis or predictive modelling is referred to as data preparation, also known as data preprocessing. The method includes an assortment of steps, which includes integrating the data, transforming data, and data reduction.

Data integration is the procedure of combining knowledge from various records, involving databases, spreadsheets, and text files. When data is kept in various locations or formats, this step is essential.

Data transformation means putting the data into a standard form so that algorithmic methods for machine learning may readily analyse or use it. This step could encompass implementing features, scaling, encoded and data the normalisation process.

Data reduction: By taking off redundant characteristics or samples from the information set, data reduction involves decreasing the size of the dataset. Taking this step might reduce the risk of overfitting and raise the success rate of machine learning algorithm design.

Data cleaning is the beginning dataset needs to have mistakes redundant information, discrepancies, and missing or irrelevant data abolished or rectified. This phase is essential for making sure the uniformity and precision of the data.
Modelling techniques are methods for creating computational or statistical models that represent everyday events or events. These mathematical representations can be used to imitate scenarios, render projections, or identify a system's underlying mechanisms.

Following are some of the modeling techniques:-

1. Decision trees: Decision trees stands for the selections and probable results deploying a tree-like model. In prediction and classification difficulties, it is often utilised.
2. Clustering: By applying the clustering approach, identical points of information can be put collectively according to similar features or characteristics. It is frequently used to relate for problems with learning that is unsupervised.
3. Neural networks are an algorithm for recognizing patterns and forecast based on massive data sets that are inspired by the structure and operation of the human brain.

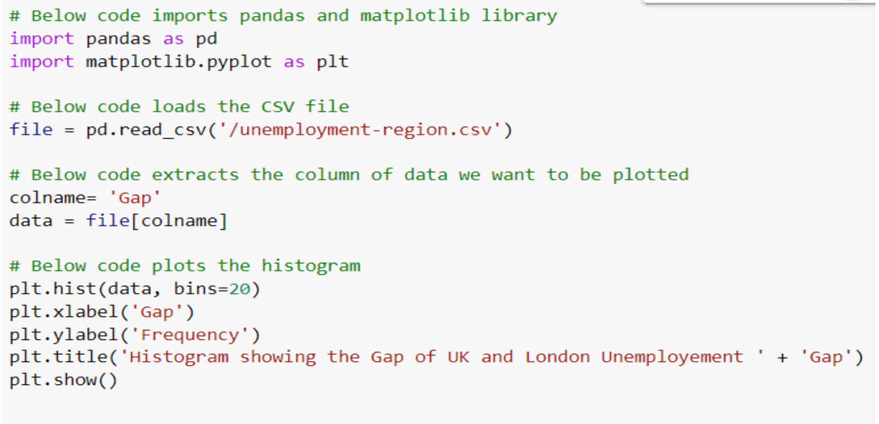
`

Results and discussion:

As far as the dataset is concerned, It has been seen that there is massive unemployment in UK and London altogether during the world recession in 2008 and in 2020 during the Covid Pandemic. The data will be analysed by using Histograms, Scatter plot, K-Means and Principal Component analysis to discover some patterns which would further help the government to boost the economy by reducing the unemployment rate. The ultimate goal is to find the likeliness of UK and London Data along with their separate trends.

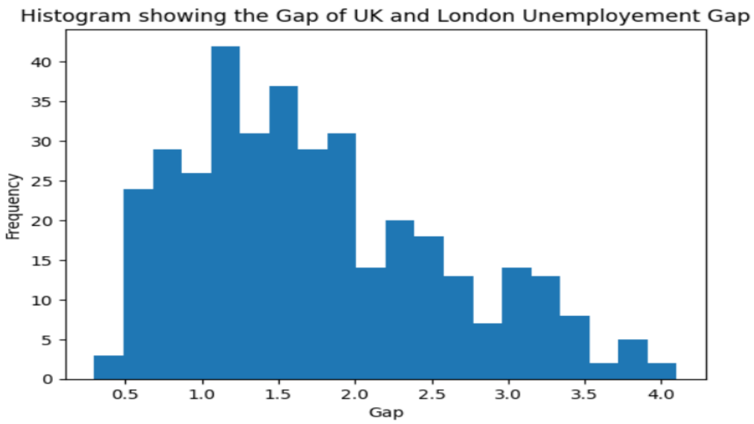
Creating Histogram by taking Percentage Gap as the main column

Source Code



```
# Below code imports pandas and matplotlib library
import pandas as pd
import matplotlib.pyplot as plt

# Below code loads the CSV file
file = pd.read_csv('/unemployment-region.csv')

# Below code extracts the column of data we want to be plotted
colname= 'Gap'
data = file[colname]

# Below code plots the histogram
plt.hist(data, bins=20)
plt.xlabel('Gap')
plt.ylabel('Frequency')
plt.title('Histogram showing the Gap of UK and London Unemployement ' + 'Gap')
plt.show()
```

Output



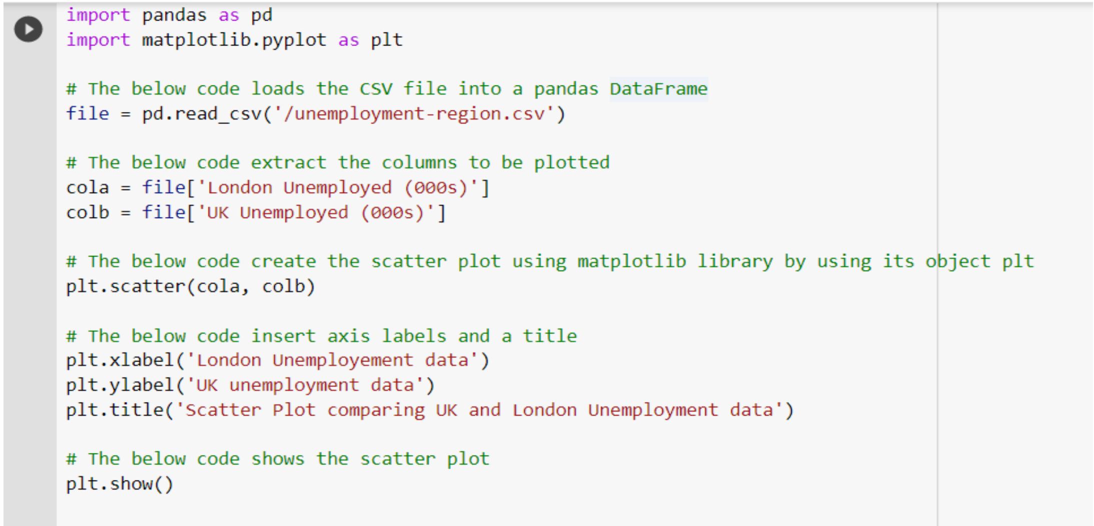
Code Description:

- The above code imports 2 libraries **pandas** and **matplotlib.pyplopt** Then objects were created for the library to fetch their inside built in functions. A variable named **file** is created which in which the csv file that was uploaded on Google collab is being read. Then another variable **colname** is created which is going to store the details of the Gap column from the csv file. Now with the object **plt** of matplotlib.pyplot library, we are going to set the title, x-axis, y-axis and finally shows the histogram by using **plt.hist (data, bins=20)** and **plt.show ( )** function.

`

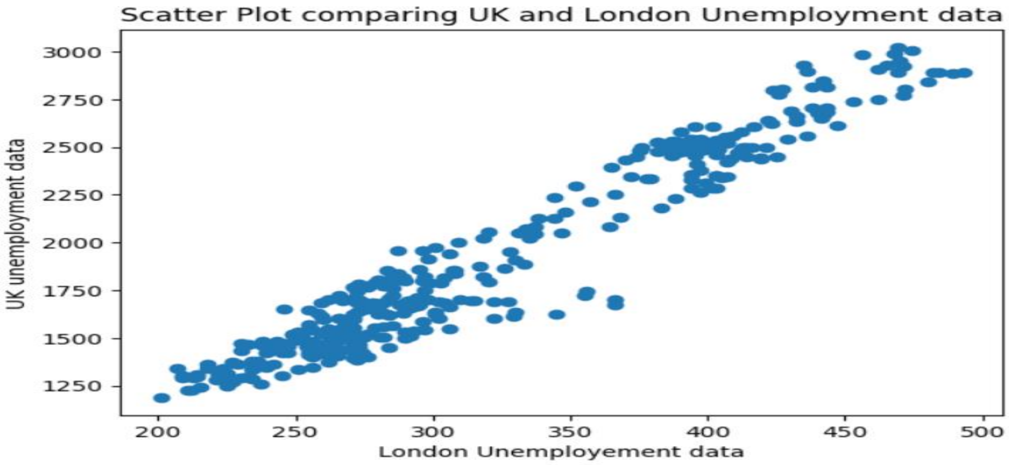
Creating scatter plot for by taking UK unemployed (000s) and London unemployed (000s) as the main columns

Source code



```
import pandas as pd
import matplotlib.pyplot as plt

# The below code loads the CSV file into a pandas DataFrame
file = pd.read_csv('/unemployment-region.csv')

# The below code extract the columns to be plotted
cola = file['London Unemployed (000s)']
colb = file['UK Unemployed (000s)']

# The below code create the scatter plot using matplotlib library by using its object plt
plt.scatter(cola, colb)

# The below code insert axis labels and a title
plt.xlabel('London Unemployement data')
plt.ylabel('UK unemployment data')
plt.title('Scatter Plot comparing UK and London Unemployment data')

# The below code shows the scatter plot
plt.show()
```

Output



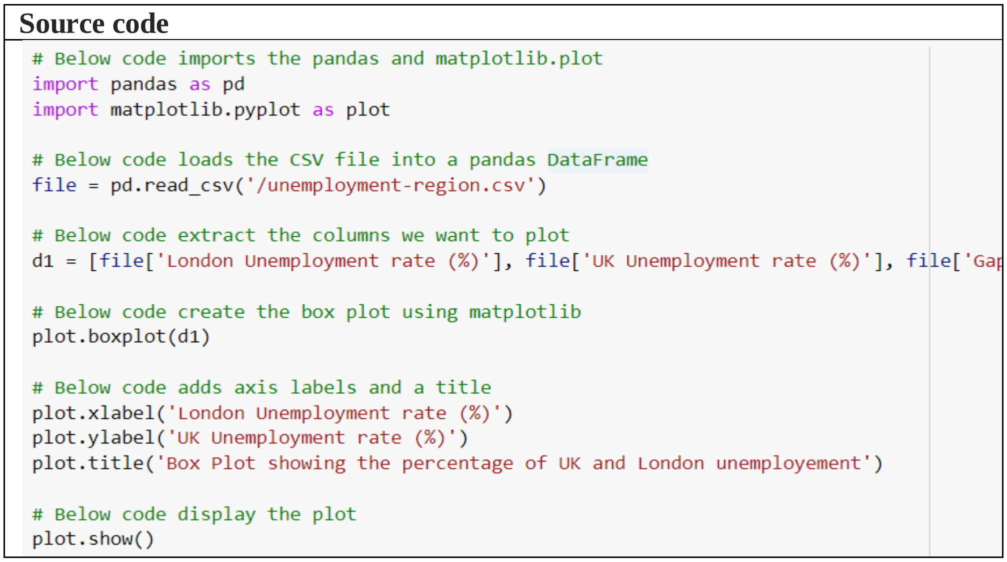
Code description:

The above code describes the creation of UK unemployed and London Unemployed data scatter chart. The first and the foremost thing is to import the **pandas** library and creating its object **pd** and the another library **matplotlib.pyplot** with its object **plt.** Secondly, another variable **file** is created which is reading the csv file which was uploaded in Google collab project. In the next step, two variables are created **cola** which stores the values of London unemployed (000s) and **colb** which store the values of UK unemployed (000s). Then we use scatter chart to print values of **cola** and **colb**. Then we plot the data using labeling x axis, y axis and title and finally using **plt.show ( )** to display the scatter chart.

`

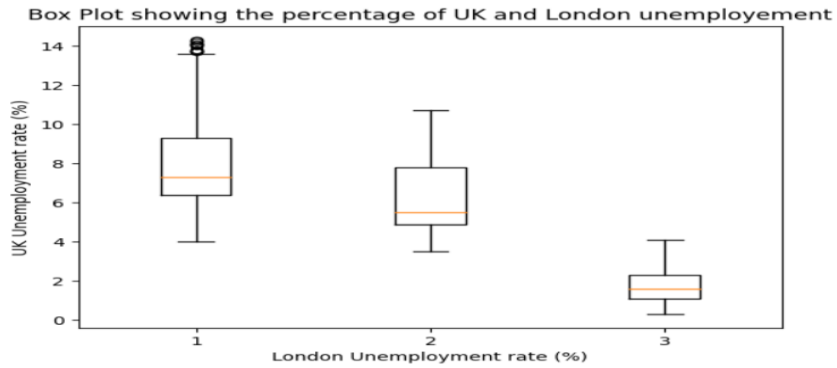
Creating Box plot for the data for the UK and London Unemployment data in terms of percentage

Source code



```
# Below code imports the pandas and matplotlib.plot
import pandas as pd
import matplotlib.pyplot as plot

# Below code loads the CSV file into a pandas DataFrame
file = pd.read_csv('/unemployment-region.csv')

# Below code extract the columns we want to plot
d1 = [file['London Unemployment rate (%)'], file['UK Unemployment rate (%)'], file['Gap

# Below code create the box plot using matplotlib
plot.boxplot(d1)

# Below code adds axis labels and a title
plot.xlabel('London Unemployment rate (%)')
plot.ylabel('UK Unemployment rate (%)')
plot.title('Box Plot showing the percentage of UK and London unemployement')

# Below code display the plot
plot.show()
```

Output



Code description:

Here we are creating the box plot of the given csv data which was uploaded on Google collab. Firstly the two libraries **pandas** and **matplotlib.pyplot**. Then a variable file is created which reads the csv file. Now in another variable **d1,** we are string the data of three columns: **London Unemployment rate (%), UK Unemployment rate (%)** and Gap. Then we are creating the boxplot from the variable **d1**. After that plotting of x axis, y axis and the title showing the title. Then we **plot.show ( )** to display the box plot.

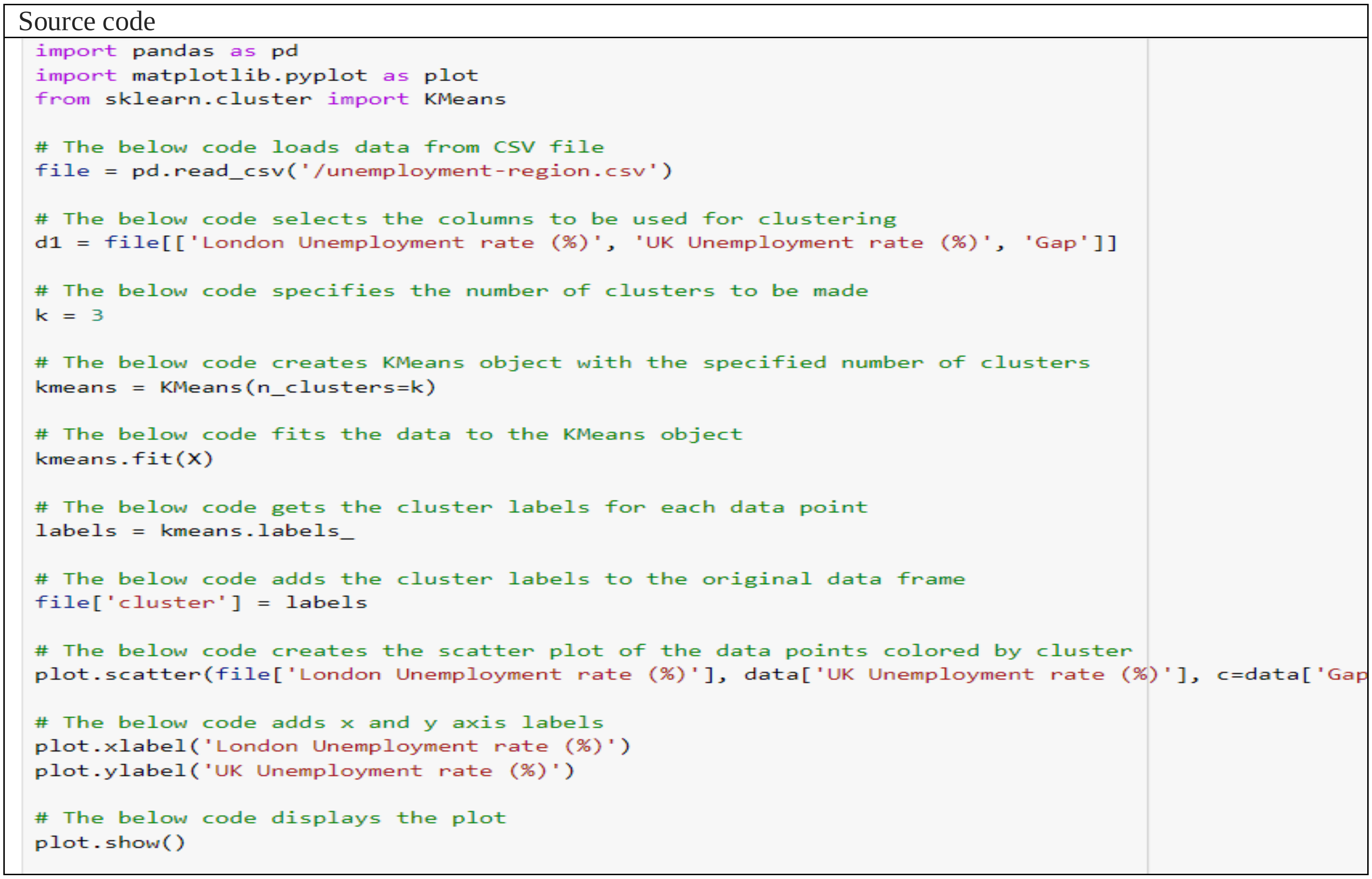
K-Means

A appreciated unsupervised machine learning algorithm for clustering or grouping data is K-means. An array of observations (or data points) is to be grouped into k groups (or clusters) in accordance to their similarity.
The technique chooses k beginning centroids (cluster centres) at random, where k is the desired number of clusters. The nearest the centre of mass. is then provided with each data point, and that centroid is updated to represent the average of all the data points that are provided to it. Until the centroids stop moving clearly or an established number of iterations has been accomplished reached, this process is continued.

Segmentation of customers, segmenting pictures, and recognising anomalies are just a few instances that illustrate the many uses for the straightforward and efficient algorithm K-means. It has to adhere to some restrictions, though, including an assumption of comparable cluster sizes and spherical clusters as well as its sensitivity to the initial the centre of mass.

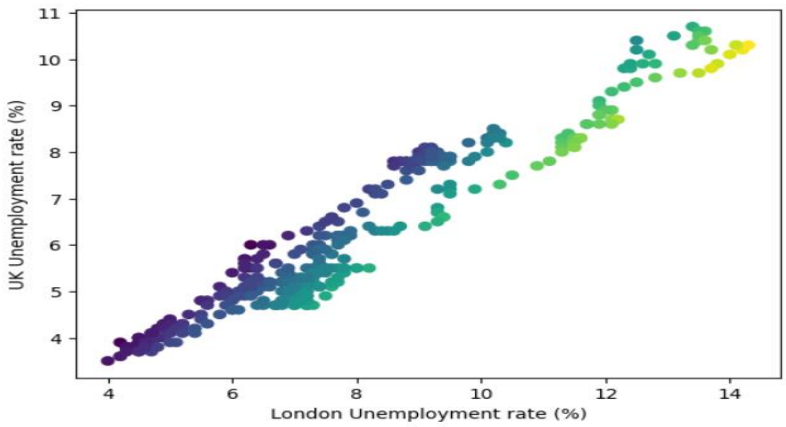
`

Source code



```
import pandas as pd
import matplotlib.pyplot as plot
from sklearn.cluster import KMeans

# The below code loads data from CSV file
file = pd.read_csv('/unemployment-region.csv')

# The below code selects the columns to be used for clustering
d1 = file[['London Unemployment rate (%)', 'UK Unemployment rate (%)', 'Gap']]

# The below code specifies the number of clusters to be made
k = 3

# The below code creates KMeans object with the specified number of clusters
kmeans = KMeans(n_clusters=k)

# The below code fits the data to the KMeans object
kmeans.fit(X)

# The below code gets the cluster labels for each data point
labels = kmeans.labels_

# The below code adds the cluster labels to the original data frame
file['cluster'] = labels

# The below code creates the scatter plot of the data points colored by cluster
plot.scatter(file['London Unemployment rate (%)'], data['UK Unemployment rate (%)'], c=data['Gap

# The below code adds x and y axis labels
plot.xlabel('London Unemployment rate (%)')
plot.ylabel('UK Unemployment rate (%)')

# The below code displays the plot
plot.show()
```

Output



Code Description:

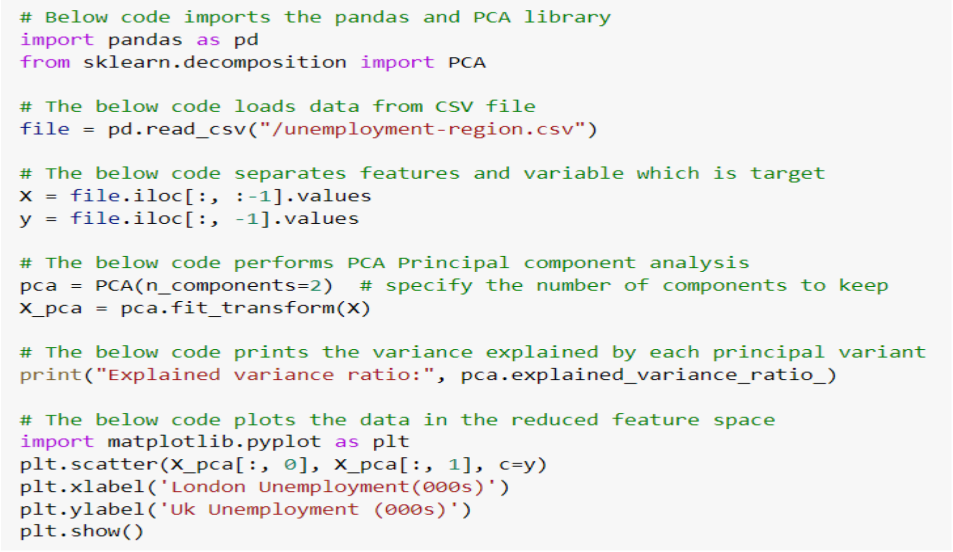
The above code imports pandas, **matplotlib** and **Kmeans** library. Then a variable file is created which is reading the csv file that is uploaded on Google collab with the name unemployement-region.csv. After that in d1, we are storing the information of the three variables London Unemployment rate %, UK unemployment rate % and Gap. Then in another variable k we are specifying the number of clusters to be made which are basically 3. Then we have to create an object of the cluster. Then we are fitting the values of the variable. Then we are setting the scatter plot by using x-labels, y-labels and finally printing it by using **plot.show ( ).**

`

Principal Component analysis:

PCA which stands for Principal Component Analysis is an unsupervised machine learning methodology as far as the data mining is concerned. The primary objective is to eradicate the anomalies while keeping an ample amount of the data. PCA technique converts the initial data into freshly coordinate system as far as the principal components are concerned. PCA allows getting a clear picture of the data and for diminishing the noise. It further enables the data scientist to find the most appropriate variables. In nutshell, PCA helps to extract the patterns which eventually help to understand data in a more appropriate manner.

Source code :



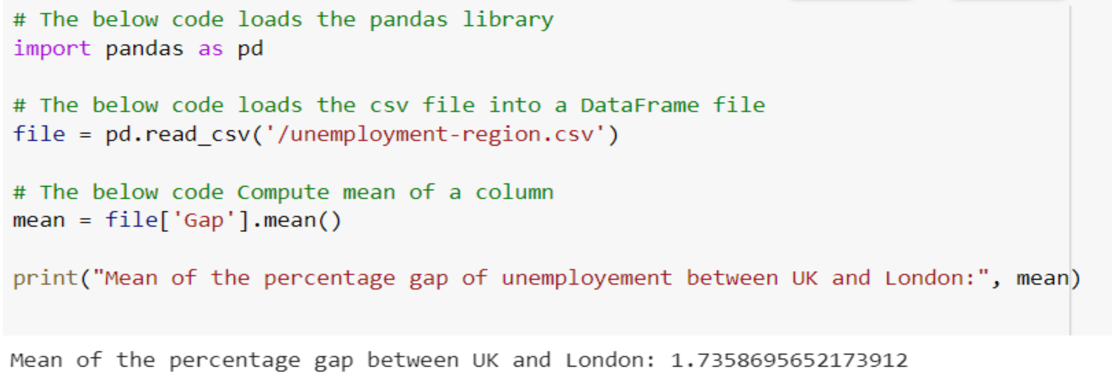
```
# Below code imports the pandas and PCA library
import pandas as pd
from sklearn.decomposition import PCA

# The below code loads data from csv file
file = pd.read_csv("/unemployment-region.csv")

# The below code separates features and variable which is target
X = file.iloc[:, :-1].values
y = file.iloc[:, -1].values

# The below code performs PCA Principal component analysis
pca = PCA(n_components=2)  # specify the number of components to keep
X_pca = pca.fit_transform(X)

# The below code prints the variance explained by each principal variant
print("Explained variance ratio:", pca.explained_variance_ratio_)

# The below code plots the data in the reduced feature space
import matplotlib.pyplot as plt
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y)
plt.xlabel('London Unemployment(000s)')
plt.ylabel('Uk Unemployment (000s)')
plt.show()
```

Code description:

The very first thing which is immensely important is to import the required libraries which is pandas and PCA. Secondly, a variable file is created which is storing the data which being read from the csv file unemployment-region.csv that is being stored in Google Collab. This eventually separates the feature variable X and the target variable Y. Then the PCA model fitted to the standard feature by invoking **pca.fit.transform ( )** on a feature variable X. The values which are based on principal component are stored in a dataframe with the target variable. Finally by using the xlabel and ylabel, the data is being plotted.

Finding the mean for the percentage unemployment gap of London and UK

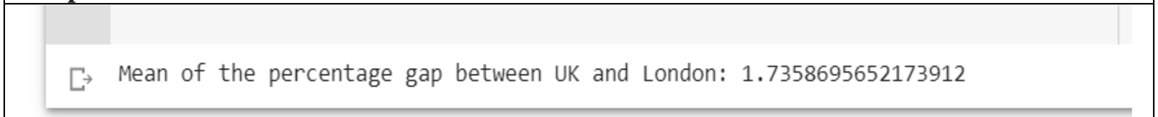
Source Code



```
# The below code loads the pandas library
import pandas as pd

# The below code loads the csv file into a DataFrame file
file = pd.read_csv('/unemployment-region.csv')

# The below code Compute mean of a column
mean = file['Gap'].mean()

print("Mean of the percentage gap of unemployement between UK and London:", mean)
```

```
Mean of the percentage gap between UK and London: 1.7358695652173912
```

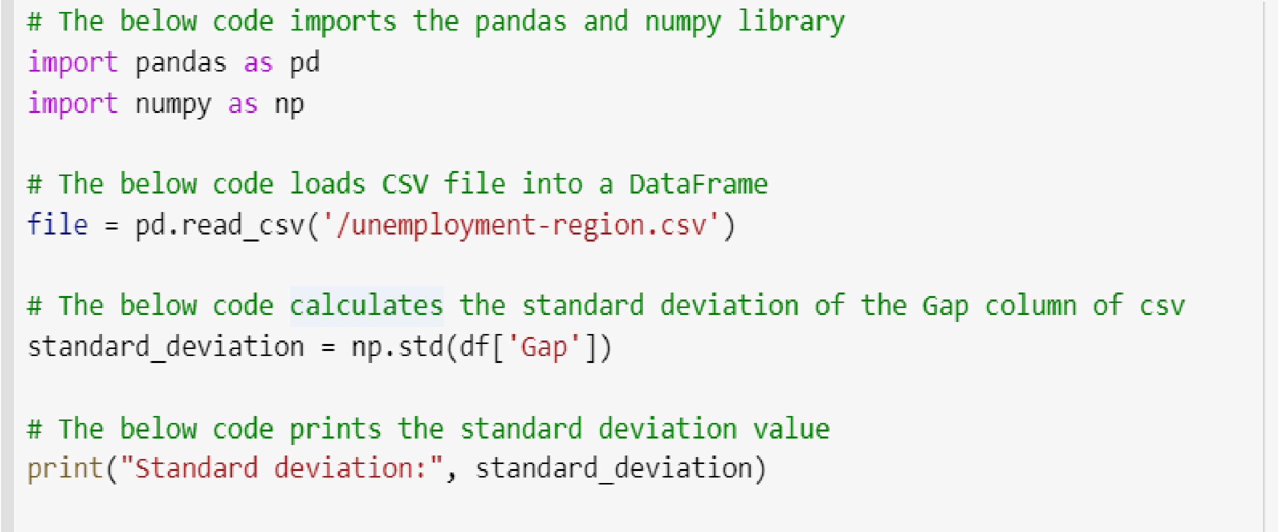
Output



```
Mean of the percentage gap between UK and London: 1.7358695652173912
```

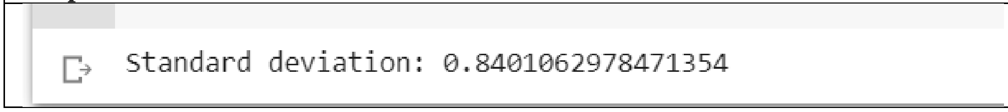
Code description:
First of all pandas library has been imported and then a variable name file is created which is storing the csv file uploaded on Google collab. Then **mean ( )** has been applied on the **Gap** column of csv file which will eventually print the mean of the **Gap** column which is 1.735

Source Code



```
# The below code imports the pandas and numpy library
import pandas as pd
import numpy as np

# The below code loads CSV file into a DataFrame
file = pd.read_csv('/unemployment-region.csv')

# The below code calculates the standard deviation of the Gap column of csv
standard_deviation = np.std(df['Gap'])

# The below code prints the standard deviation value
print("Standard deviation:", standard_deviation)
```

Output



```
Standard deviation: 0.8401062978471354
```

Code description:
The above mention code import the pandas and numpy library and creating its objectives pd and np respectively.
Then a variable file is created which is reading the data from the csv file uploaded in Google Collab project. Now with numpy object np, a function np.std function is being invoked which is finding the standard deviation of the Gap column of the data set. Finally in the output the value of the standard deviation is being printed which is 0.840106.

Conclusions and perspectives

As far as the project and technical report is concerned, the overall objective is not to create super quality models but the models who deliver high quality patterns and information. When we evaluate the data set with different models, we would be able to see how the data set acts as fitted into different models. The takeaway that should be derived from the dataset analysis is that from 1992 to 2022, the unemployment GAP is decreasing massively but the government and the society need to take more measures to reduce this gap as much as possible to strengthen the economy and the nation.

`

Link to Google Colab project

https://colab.research.google.com/drive/1Fg0Vh_TvbvjEu3ruV_wLqo20Fvi1ZHap?usp=sharing

References

https://data.london.gov.uk/dataset/unemployment-rate-region (csv data fetched from this link)

https://www.geeksforgeeks.org/data-mining/