Project 3

ET1550 Introduction to Machine Learning and Artificial Intelligence

Kevin Rasmusson

Lund, 07/19/2022

Part 1

Question 1

Code

```
dataset.describe()
```

Output

	Country_Name	GDP_growth_annual_pct	GDP_per_capita_KUSD	GDP_current_TUSD	GNI_per_capita_KUSD	Exports_of_goods_and_services_pct_of_GDP	Foreign_direct_investment_BUSD	Inflation_consumer_prices_annual_pct	Unemployment_total_pct_of_total_labor_force	Total_tax_and_contribution_rate_pct_of_profit	Life_expectancy_at_birth_total_years
0	Austria	1.0	44.2	0.4	47.5	53.1	-8.0	0.9	5.7	51.7	81.2
1	Belgium	2.0	41.0	0.5	45.6	77.8	-19.5	0.6	8.5	58.4	81.0
2	Bulgaria	4.0	7.1	0.1	7.5	64.0	2.2	-0.1	9.1	27.0	74.6
3	Croatia	2.4	11.8	0.0	13.0	46.4	0.1	-0.5	16.2	20.0	77.3
4	Cyprus	3.2	23.4	0.0	26.0	70.1	29.0	-2.1	14.9	24.0	80.3
5	Czech Republic	5.4	17.8	0.2	18.4	80.6	1.7	0.3	5.0	46.5	78.6
6	Denmark	2.3	53.3	0.3	60.5	55.4	1.9	0.5	6.3	23.9	80.7
7	Estonia	1.8	17.5	0.0	18.7	76.9	-0.7	-0.5	6.2	49.2	77.6
8	Finland	0.5	42.8	0.2	47.2	35.4	16.8	0.2	9.4	37.9	81.5
9	France	1.1	36.7	2.4	41.1	30.6	42.8	0.0	10.3	64.9	82.3
10	Germany	1.5	41.1	3.4	45.8	46.9	62.4	0.5	4.6	48.8	80.6
11	Greece	-0.4	18.1	0.2	20.1	32.2	1.3	-1.7	24.9	49.6	81.0
12	Hungary	3.8	12.7	0.1	13.2	87.6	-5.3	-0.1	6.8	48.2	75.6
13	Ireland	25.2	62.0	0.3	50.4	122.0	237.1	-0.3	9.9	26.0	81.5
14	Italy	0.8	30.2	1.8	33.0	29.7	13.3	0.0	11.9	64.8	82.5
15	Latvia	4.0	13.8	0.0	16.7	60.3	0.8	0.2	9.9	35.9	74.5
16	Lithuania	2.0	14.3	0.0	6.0	68.8	1.0	-0.9	9.1	42.6	74.3
17	Luxembourg	4.3	101.4	0.1	72.5	221.2	12.5	0.5	6.7	20.6	82.3
18	Malta	9.6	24.9	0.0	25.2	154.6	3.6	1.1	5.4	41.5	81.9
19	Netherlands	2.0	45.2	0.8	49.9	82.7	322.6	0.6	6.9	41.0	81.5
20	Poland	4.2	12.6	0.5	13.3	49.1	15.1	-0.9	7.5	40.3	77.5
21	Portugal	1.8	19.3	0.2	20.5	40.6	1.3	0.5	12.4	40.9	81.1
22	Romania	3.0	9.0	0.2	9.6	41.4	4.3	-0.6	6.8	42.0	74.9
23	Slovenia	2.2	20.9	0.0	22.3	77.1	1.7	-0.5	9.0	31.0	80.8
24	Slovak Republic	4.8	16.3	0.1	17.7	92.0	1.5	-0.3	11.5	50.4	76.6
25	Spain	3.8	25.7	1.2	28.4	33.6	23.0	-0.5	22.1	49.8	82.8
26	Sweden	4.5	51.5	0.5	58.4	43.8	10.3	-0.0	7.4	49.1	82.2

Question 2

Code

```
y_dataset = dataset[["Country_Name"]]
X_dataset = dataset.iloc[:, 1:]
```

Output

Dataset separated.

Question 3

Code

```
X_dataset_mean = X_dataset.mean()
X_dataset_std = X_dataset.std()
X_dataset_norm = (X_dataset - X_dataset_mean) / X_dataset_std
```

Output

Dataset normalized.

Question 4

Code

```
pca = PCA(n_components=2)
pca.fit(X_dataset_norm)
```

Output

```
▾        PCA
PCA(n_components=2)
```

Question 5

Code

```
print(pca.components_)
```

Output

```
[[ 2.09769928e-01  4.97345498e-01  1.34095169e-01  4.88851672e-01
   2.97829533e-01  2.71525482e-01  3.26397993e-01 -2.22714435e-01
  -6.13966806e-02  3.59654170e-01]
 [-3.72799823e-01 -4.67068052e-04  5.23711318e-01  1.40692553e-01
  -4.21708884e-01 -6.61906157e-02  9.80983292e-02  1.18597791e-01
   5.18470541e-01  3.03697438e-01]]
```

Comments

According to https://stackoverflow.com/questions/50796024/feature-variable-importance-after-a-pca-analysis, deciphering the development indicators with greatest impact is done by finding the corresponding features with the greatest (absolute) values inside the eigenvectors. For the first eigenvector, this is at indices 1, 3, 8 and 9, corresponding to GDP_per_capita_KUSD, GNI_per_capita_KUSD, Total_tax_and_contribution_rate_pct_of_profit and Life_expectancy_at_birth_total_years respectively. With regards to the second eigenvector, the for most impactful features are located at indices 0, 2, 4 and 8, corresponding to GDP_growth_annual_pct, GDP_current_TUSD, Exports_of_goods_and_services_pct_of_GDP and Total_tax_and_contribution_rate_pct_of_profit.

Question 6

Code

```
total_explained_variance_ratio = np.sum(pca.explained_variance_ratio_)
```

Output

0.5657820657915607

Comments

The number simply reflects the sum of every element in the previously printed array (the array of ratio of variances held by each principal component, namely [0.34336846 0.2224136])

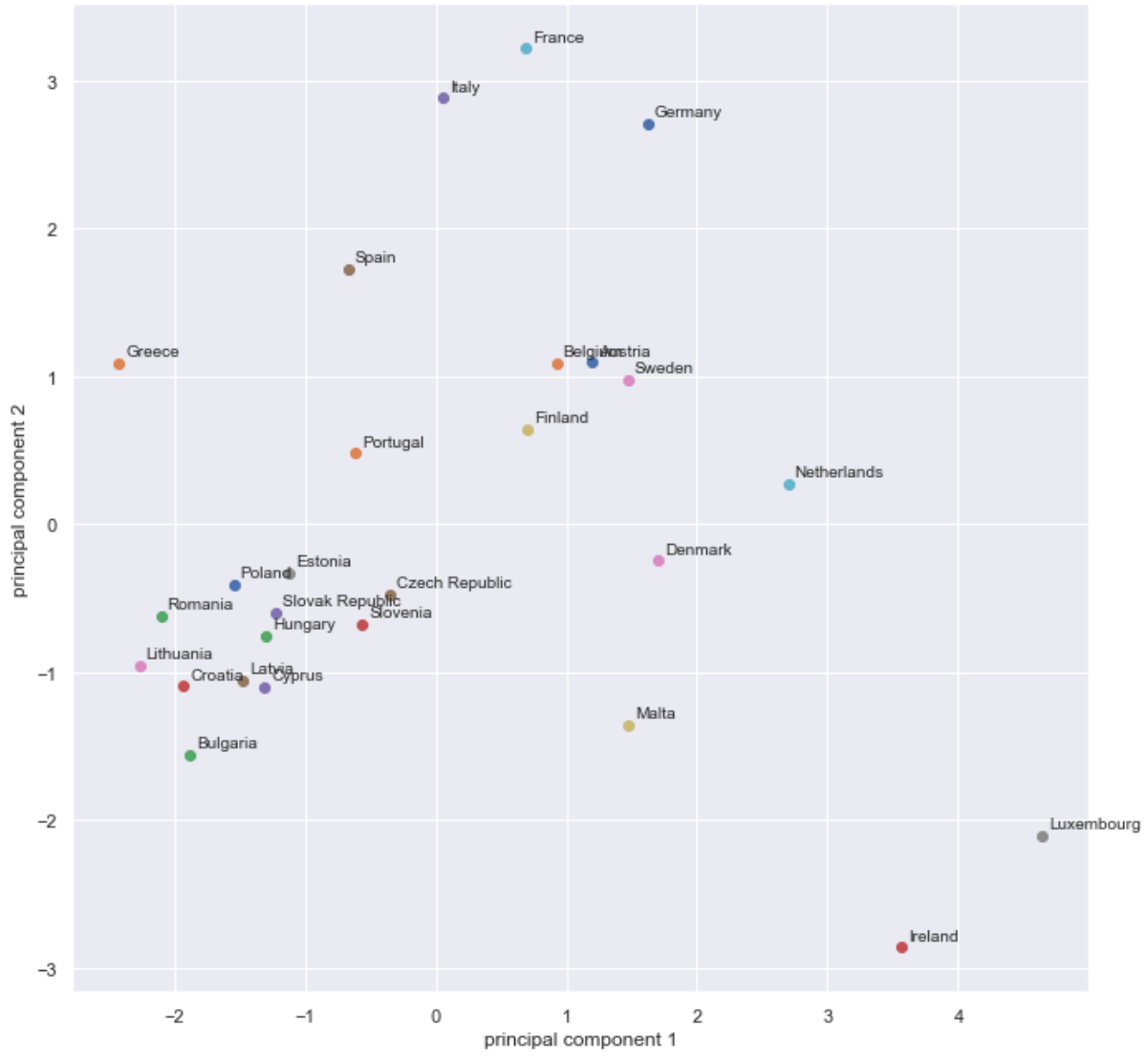
Question 7

Code

```
X_pca = pca.transform(X_dataset_norm)
```

Output
Original shape: (27, 10)
Transformed shape: (27, 2)

Visualization



Part 2

Question 8

Code

```
kmeans = KMeans(n_clusters=10, random_state=0)
kmeans.fit(X_dataset_)
```

Output

```
KMeans(n_clusters=10, random_state=0)
```

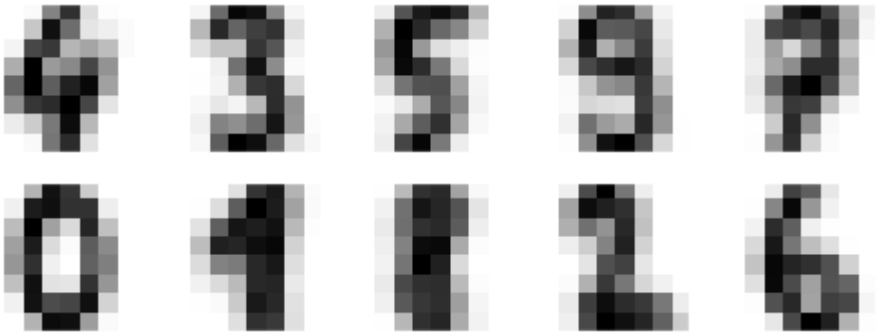
Question 9

Code

```
clusters = kmeans.fit_predict(X_dataset_)
```

Output
(10, 64)

Digit output

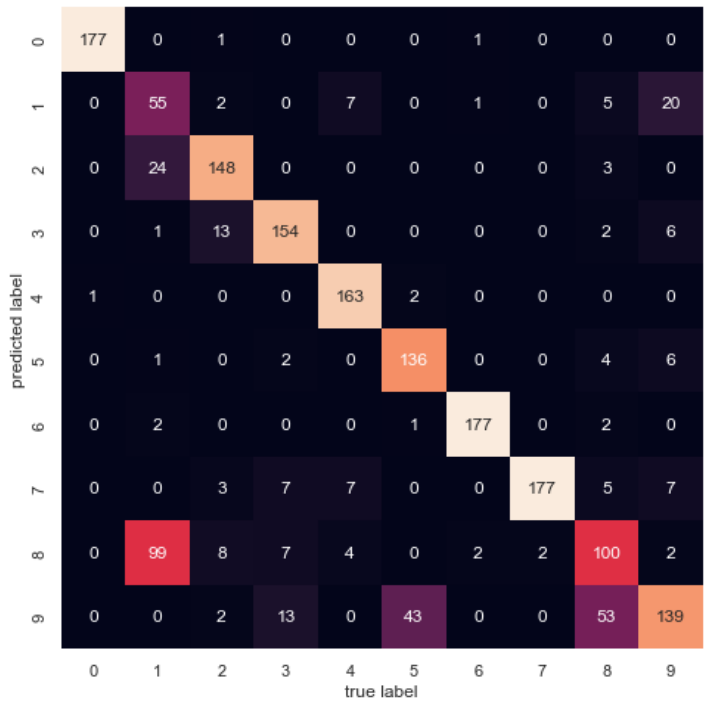


Question 10
Code

```
kmeans_accuracy = accuracy_score(y_dataset_, labels)
```

Output
0.7935447968836951

Question 11
Confusion matrix



Comments

According to the confusion matrix, it seems that the model had most difficulty recognizing the digit 1 (by guessing 8 instead). This could be explained by the low resolution (see question nine, the digit 8 almost looks like a long vertical line (making it a valid guess when presented with a one). Other confusions included guessing the number 9 when presented with either a 5 or an 8. This could also be explained by the similarity of the digits, especially in the low-resolution scenario we operate with.