CS652: Practical ML and Data Mining
Assignment#5

การทำ dimensionality reductions และจัดกลุ่มด้วย Kmeans

<u>ชุดข้อมูล</u> : เป็นข้อมูลที่ได้จากเครื่องวัด accelerometers และ gyros ของโทรศัพท์มือถือรุ่นหนึ่ง ซึ่งค่าที่วัดได้สามารถบ่งบอกกิจกรรมของผู้ถือโทรศัพท์ขณะนั้นว่ากำลังทำกิจกรรมใด เช่น ยืน นอน นั่ง เดิน บนพื้นราบหรือเดินขึ้นลงบันได เป็นต้น

สมมติว่านักศึกษาไม่ทราบจำนวนของกิจกรรมที่เครื่องวัดสามารถแยกแยะได้ และพยายามจะจัดกลุ่มของกิจกรรมจากค่าที่วัดได้จากโทรศัพท์รุ่นดังกล่าว ด้วยเทคนิคการเรียนรู้ของเครื่องแบบไม่มีผู้สอน เพื่อหาจำนวนกิจกรรมที่เป็นไปได้ด้วยอัลกอริทึม Kmeans

ชุดข้อมูลประกอบด้วย 4 files โดยมีการแบ่งเป็น train และ test data มาให้ สามารถโหลดได้จาก XXX

- data_train.txt
- data_train_labels.txt
- data_test.txt
- data_test_labels.txt

นำข้อมูลที่ได้มาใส่ตัวแปร X_train, y_train และ X_test, y_test โดยตัวแปรแต่ละตัวควรมีขนาด

(7352, 561) (7352,) (2947, 561) (2947,) **ตามลำดับ**

<u>คำสั่ง</u>

1. ให้รวมข้อมูลจาก training กับ test เข้าด้วยกัน โดยรวม X_train กับ X_test เป็น X และรวม y_train กับ y_test เป็น y แล้วแสดงค่าของคลาสที่มีทั้งหมดใน y โดยค่าที่ควรจะได้คือ 1, 2, 3, 4, 5, 6
   ความหมายของ labels เหล่านี้คือ :
   1 – walking
   2 – walking upstairs
   3 – walking downstairs
   4 – sitting
   5 – standing
   6 – laying down

2. ให้ปรับ Scale ของข้อมูลใน X ด้วย StandardScaler

3. ให้ใช้อัลกอริทึม PCA ในการลดขนาดของ dimensions โดยให้เหลือจำนวน components เท่าที่จำเป็น เพื่อให้ได้ค่า variance ที่จำเป็นในการอธิบายข้อมูลที่ปรับ scale แล้วอย่างน้อย 90% ด้วยการใช้พารามิเตอร์ *n_components* และให้ใส่ค่า *random_state = 17*
   ดูวิธีการใส่ค่าของ *n_compnents* ได้จาก
   https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

4. ให้พิมพ์ค่าจำนวน components ที่ต้องใช้เพื่อให้ได้ค่า variance ที่สามารถอธิบายข้อมูลที่ปรับ scale แล้วได้ 90% (Hint ดูได้จากจำนวน column ที่เหลือของ X หลังจากทำ PCA แล้ว)

5. ให้แสดง % ของค่า variance ที่ first principal component สามารถอธิบายได้
   (Hint: ตัวแปร explained_variance_ratio_ ของ pca โดยเอาข้อมูลช่องแรก หรือช่องที่ 0)

6. Visualize ข้อมูลที่ถูก transform ใหม่ด้วยสอง components แรก ด้วยคำสั่ง

```
plt.figure(figsize=(15, 10))
plt.scatter(X_pca[:,0], X_pca[:,1] , c=y, s=20, cmap='viridis');
plt.xlabel("First principal component")
plt.ylabel("Second principal component")
plt.show()
```

7. ใช้ Kmeans ในการทำ clustering โดยใช้ข้อมูลที่ลดขนาด dimensions ด้วย PCA แล้ว (ในที่นี้เราพอรู้จากข้อมูลบ้างแล้วว่าจำนวนกิจกรรมที่มีจริงคือ 6 ตามจำนวน class labels ในชุดข้อมูล จึงควรแบ่งด้วย *n_clusters = 6* แต่ในความเป็นจริงการเรียนรู้แบบไม่มีผู้สอน เราจะไม่รู้จำนวน clusters ล่วงหน้า)
   กำหนดให้ใช้พารามิเตอร์ของ Kmeans ดังนี้

```
n_clusters = n_classes  (number of unique labels of the target class)
n_init = 100
random_state = 17
```

8. Visualize ข้อมูลสอง components แรก แสดงสีของข้อมูลตามคลัสเตอร์ที่ถูกแบ่ง คลัสเตอร์ละหนึ่งสี
   หมายเหตุ ข้อมูลที่ผ่าน PCA แล้วอยู่ในตัวแปร X_PCA

```
plt.figure(figsize=(15, 10))
plt.scatter(X_pca[cluster_labels == 0, 0], X_pca[cluster_labels == 0, 1], s = 100, c = 'red', label = 'walking')
plt.scatter(X_pca[cluster_labels == 1, 0], X_pca[cluster_labels == 1, 1], s = 100, c = 'yellow', label = 'going up the stairs')
plt.scatter(X_pca[cluster_labels == 2, 0], X_pca[cluster_labels == 2, 1], s = 100, c = 'aqua', label = 'going down the stairs')
plt.scatter(X_pca[cluster_labels == 3, 0], X_pca[cluster_labels == 3, 1], s = 100, c = 'violet', label = 'sitting')
plt.scatter(X_pca[cluster_labels == 4, 0], X_pca[cluster_labels == 4, 1], s = 100, c = 'lightgreen', label = 'standing')
plt.scatter(X_pca[cluster_labels == 5, 0], X_pca[cluster_labels == 5, 1], s = 100, c = 'blue', label = 'lying')
plt.title('Cluster of Activities')
plt.xlabel('First principal component')
plt.ylabel('Second principal component')
plt.legend()
plt.show()
```

เปรียบเทียบผลลัพธ์ของการจัดกลุ่มด้วย Kmeans กับ activity จริง ด้วยตาราง โดยรันโค้ดด้านล่างนี้

```
tab = pd.crosstab(y, cluster_labels, margins=True)
tab.index = ['walking', 'going up the stairs',
             'going down the stairs', 'sitting', 'standing', 'lying', 'all']
tab.columns = ['cluster' + str(i + 1) for i in range(6)] + ['all']
tab
```

ซึ่งจะได้ตารางที่มีคอลัมน์เป็นคลัสเตอร์ ส่วนแถวเป็น activity จริง ตัวอย่างของข้อมูลในตารางแสดงดังภาพ

	cluster1	cluster2	cluster3	cluster4	cluster5	cluster6	all
walking	741	0	903	0	0	78	1722
going up the stairs	296	0	1241	0	2	5	1544
going down the stairs	890	0	320	0	0	196	1406
sitting	0	1235	1	91	450	0	1777
standing	0	1344	0	0	562	0	1906
lying	0	53	5	1557	329	0	1944
all	1927	2632	2470	1648	1343	279	10299

จะเห็นได้ว่าในแต่ละ activity class ข้อมูลจะถูกกระจายอยู่ในหลาย clusters ลองหาค่า % สูงสุดของข้อมูลสำหรับแต่ละ activity class ที่ถูกจัดให้อยู่ในคลัสเตอร์ใดคลัสเตอร์หนึ่ง เพื่อดูน้ำหนักการกระจายตัวของคลาสไปตาม cluster

ตัวอย่างเช่น ถ้าคลาส "going down the stairs" ซึ่งมีข้อมูลทั้งหมด 1406 ตัว จะกระจายอยู่ในแต่ละคลัสเตอร์เท่าใด สมมติว่าอยู่ใน

- cluster 1 จำนวน 900
- cluster 3 จำนวน 500
- cluster 6 จำนวน 6

ดังนั้น %ของข้อมูลสูงสุดที่ถูกจัดให้อยู่ในคลัสเตอร์ใดคลัสเตอร์หนึ่งคือ 900/1406 = 0.64

รันโค้ดด้านล่างนี้ เพื่อดูว่าในแต่ละกิจกรรมมีข้อมูลกระจายอยู่เป็นสัดส่วนอย่างไร สังเกตผล

```
pd.Series(tab.iloc[:-1,:-1].max(axis=1).values /
          tab.iloc[:-1,-1].values, index=tab.index[:-1])
```

9. จากผลที่ได้ แสดงว่า Kmeans ไม่ได้แยก activities เป็น 6 กลุ่มได้ดีนัก ให้ลองใช้ elbow method ในการหาจำนวนคลัสเตอร์ที่เหมาะสม แสดงกราฟของ elbow method

10. ทำข้อย่อยต่อไปนี้

10.1 รัน Kmeans ใหม่ด้วยจำนวนคลัสเตอร์ที่ได้จาก Elbow method โดยใช้พารามิเตอร์ชุดเดิม แต่เปลี่ยนแค่ค่า n_clusters เป็นจำนวนคลัสเตอร์ที่ได้จากข้อ 9

10.2 Plot กราฟการกระจายตัวของข้อมูลที่ได้ พร้อม cluster center

10.3 ให้สรุปลักษณะเด่นของคลัสเตอร์ที่ได้ โดยเขียนเป็น Text (จำนวนคลัสเตอร์ที่ได้ วิเคราะห์และอธิบายลักษณะเด่นของแต่ละคลัสเตอร์)

การส่งงาน

- กำหนดส่งงาน 9 เมษายน 2565 ก่อน 23:59 น.
- ตั้งชื่อไฟล์ด้วยเลขทะเบียน ตามด้วยขีดล่างและชื่อการบ้าน เช่น 640961XXXX_ass5.ipynb
- ภายในไฟล์ให้ใช้ Label เพื่อแบ่งงานออกเป็นตอน ๆ ตามที่กำหนด พร้อมทั้งตอบคำถามในแต่ละส่วนด้วยการใช้เซลล์แบบที่เป็น Text
- ส่งงานทาง courses.cs.tu.ac.th