**Lab 6.2 KMeans Clustering (3/6/2022)**

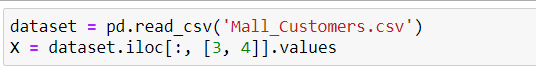
**ລະຫັດນັກສຶກສາ: 205Q0010.19**

**ຊື່ ແລະ ນາມສະກຸນ: ທ້າວ ນູຊົ່ວ ເຮີ**

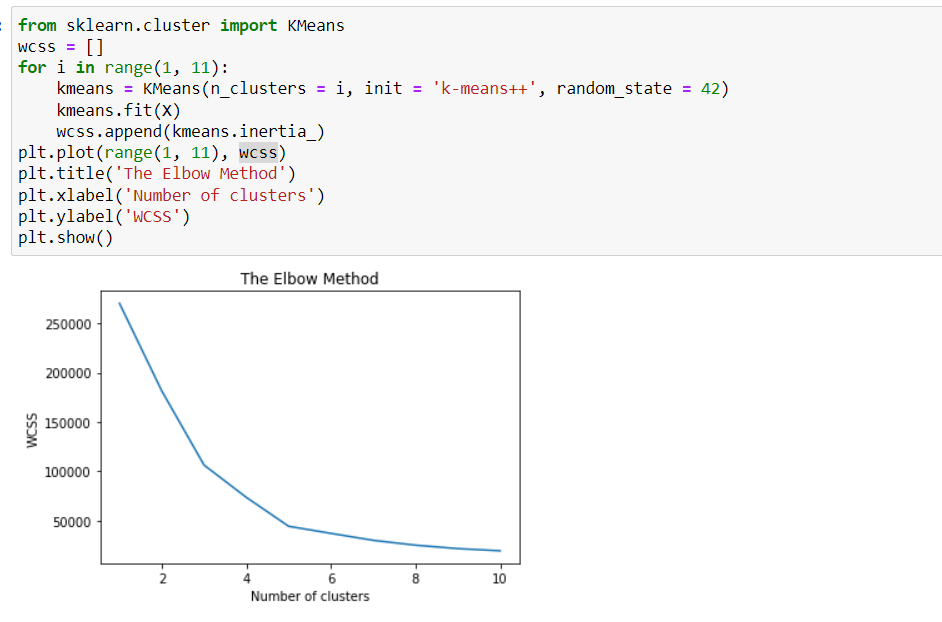
ຈົ່ງຕອບຄຳຖາມຕໍ່ໄປນີ້ໃຫ້ສຳເລັດດ້ວຍການນຳໃຊ້ຄຳສັ່ງຂອງ Python:

**ພາກທີ 1**

* 1. ຈາກຊຸດຂໍ້ມູນ Mall\_Customers.csv. ຈົ່ງທຳການເລືອກຄຸນລັກສະນະ (Features): Age ແລະ Annual Income (k$).



* 1. ຈົ່ງທຳການກຳນົດຄ່າຂອງ K ດ້ວຍການນຳໃຊ້ sklearn.cluster ແລະ Kmeans ເພື່ອຄິດໄລ່ຄ່າ wcss ພ້ອມ ແຕ້ມເສັ້ນສະແດງຂອງ Elbow Method.



1.3. ຈົ່ງສ້າງແບບຈຳຮອງ Kmeans ເພື່ອແບ່ງກຸ່ມຊຸດຂໍ້ມູນດ້ວຍຕາມຄ່າຂອງ K

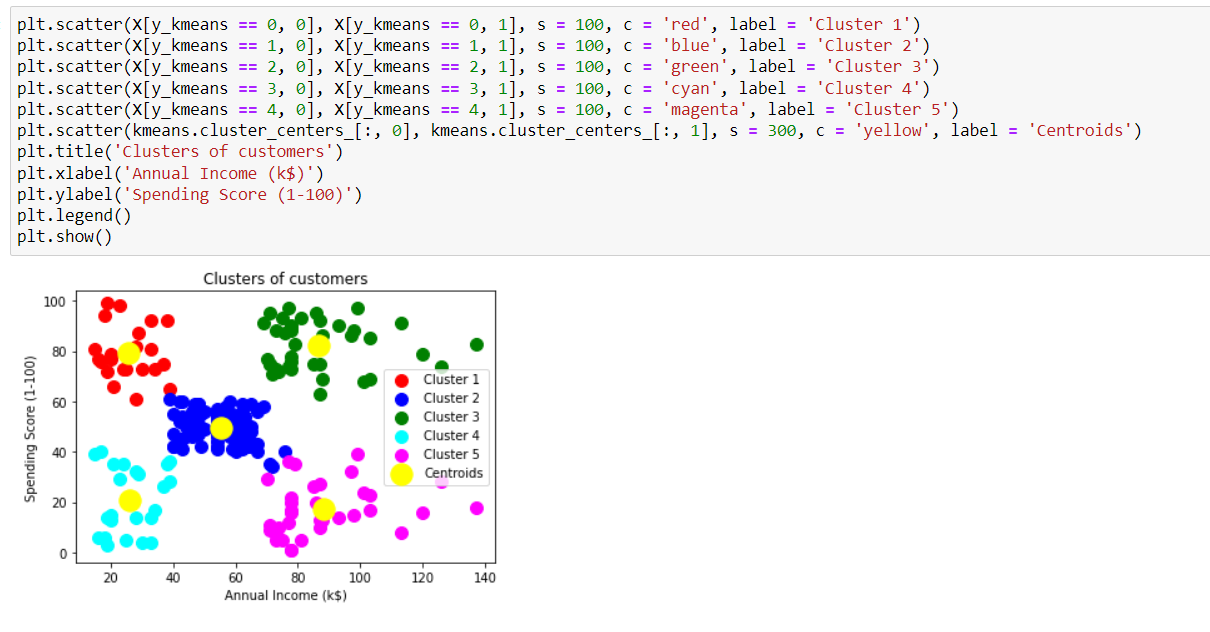
ໃນຂໍ້1.2.



* 1. ຈົ່ງແບ່ງກຸ່ມຊຸດຂໍ້ມູນໃນຂໍ້ທີ 1.1 ດ້ວຍແບບຈຳຮອງ (Algorithm) ຂໍ້ 1.3.



1.5. ຈົ່ງທຳການສຳຫຼວດ (visualizing) ກຸ່ມຂໍ້ມູນທີ່ຖືກແບ່ງດ້ວຍ scatter plot ພ້ອມອະທິບາຍຜົນ.



ແມ່ນກຸ່ມທີ່ມີອາຍຸສູຸງແຕ່ມີລາຍໄດ້ຂະຫນາດປານການຫາສູງ

ແມ່ນກຸ່ມທີ່ມີລາຍໄດ້ນ້ອຍໄປຫາຫຼາຍໃນໄວອາຍຸຍັງນ້ອຍ

ແມ່ນກຸ່ມຄົນທີມີອາຍຸລຳດັບການແຕ່ມີລາຍໄດ້ຂະຫນາດນ້ອຍໄປຫາຫຼາຍ

ແມ່ນກຸ່ມຄົນທີ່ມີລາຍໄດ້ຫຼາຍ ໃນໄວອາຍຸນ້ອຍ

ແມ່ນກຸ່ມຄົນທີ່ມີລາຍໄດ້ນ້ອຍ ໃນໄວອາຍຸປານການ

**ພາກທີ 2**

2.1 ຈາກຊຸດຂໍ້ມູນ Mall\_Customers.csv. ຈົ່ງທຳການເລືອກຄຸນລັກສະນະ (Features): Age, Annual Income(k$) ແລະ Spending Score (1-100).

col = ['Age', 'Annual Income (k$)','Spending Score (1-100)']

2.2. ຈົ່ງທຳການກະກຽມຂໍ້ມູນດ້ວຍ preprocessing.StandardScaler() ຂອງຄຸນລັກສະນະຂອງຂໍ້ມູນໃນຂໍ້ 2.1.

from sklearn import preprocessing

scaler = preprocessing.StandardScaler() # create StandardScaler instance

z=scaler.fit\_transform(dataset[col]) # calc z-score

z[:5].round(4) # 2D numpy array

array([[-1.4246, -1.739 , -0.4348],

[-1.281 , -1.739 , 1.1957],

[-1.3528, -1.7008, -1.7159],

[-1.1375, -1.7008, 1.0404],

[-0.5634, -1.6627, -0.396 ]])

2.3 ຈົ່ງທຳການກຳນົດຄ່າຂອງ K ດ້ວຍການນຳໃຊ້ sklearn.cluster ແລະ Kmeans ເພື່ອຄິດໄລ່ຄ່າ wcss ພ້ອມ ແຕ້ມເສັ້ນສະແດງຂອງ Elbow Method.

wcss\_z = []

for i in range(1, 11):

    kmeans = KMeans(n\_clusters = i, init = 'k-means++', random\_state = 42)

    kmeans.fit(X)

    wcss\_z.append(kmeans.inertia\_)

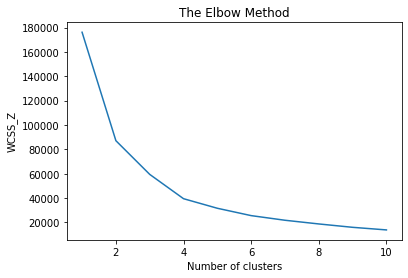
plt.plot(range(1, 11), wcss\_z)

plt.title('The Elbow Method')

plt.xlabel('Number of clusters')

plt.ylabel('WCSS\_Z')

plt.show()



2.4. ຈົ່ງສ້າງແບບຈຳຮອງ Kmeans ເພື່ອແບ່ງກຸ່ມຊຸດຂໍ້ມູນດ້ວຍຕາມຄ່າຂອງ K ໃນຂໍ້2.3.

kmeans\_z = KMeans(n\_clusters = 4, init = 'k-means++', random\_state = 42)

2.5. ຈົ່ງແບ່ງກຸ່ມຊຸດຂໍ້ມູນໃນຂໍ້ທີ 2.2 ດ້ວຍແບບຈຳຮອງໃນຂໍ້ 2.4.

z\_kmeans = kmeans\_z.fit\_predict(X)

2.5. ຈົ່ງຊອກຫາເມັດກາງ cluster\_centers\_ ຂອງການແບ່ງກຸ່ມຊຸດຂໍ້ມູນ.

kmeans\_z.cluster\_centers\_.round(4)

array([[ 31.9589, 72.9589],

[ 55.8148, 51.7778],

[ 39. , 106.5 ],

[ 30.3469, 29.2653]])

2.6. ຈົ່ງສະແດງນຳເບີກຸ່ມ (labels\_) ໃສ່ໃນຊຸດຂໍ້ມູນໃນຮູບແບບ dataframe.

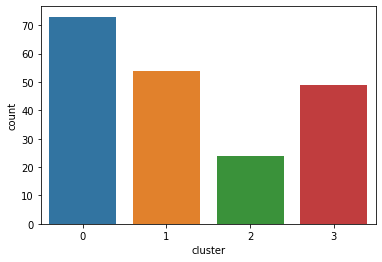
dataset['cluster']=kmeans\_z.labels\_

dataset.head()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CustomerID | Gener | Age | Annual Income(K$) | Spending Score(1-100) |
| 1 | Male | 19 | 15 | 39 |
| 2 | Male | 21 | 15 | 39 |
| 3 | Fmale | 20 | 16 | 6 |
| 5 | Fmale | 23 | 16 | 77 |

2.7. ຈົ່ງສະແດງກຸ່ມຂໍ້ມູນດ້ວຍ sns.countplot(), ພ້ອມອະທິບາຍຜົນ.

sns.countplot(x='cluster', data=dataset);



* ໃນຮູບຈະສະແດງໃຫ້ເປັນເຖິງການຂະຫຍາຍຕົວຂອງກຸ່ມວ່າເປັນຊະນິດທີມີຈຳນວນລາຍຮັບຫຼາຍຫຼື່ນ້ອງແລະ ຈັດຢູ່ໃນລຳດັບໃດ

2.8. ຈົ່ງສະແດງກຸ່ມຂໍ້ມູນດ້ວຍ sns.violinplot(), ພ້ອມອະທິບາຍຜົນ.

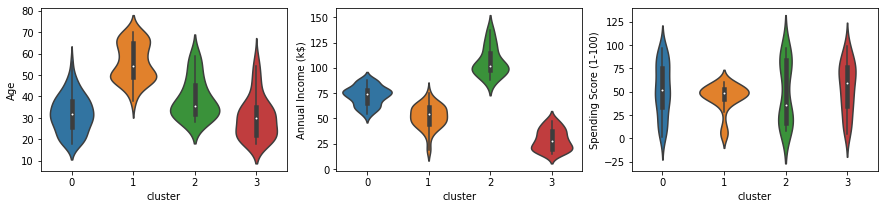
cols=['Age', 'Annual Income (k$)', 'Spending Score (1-100)']

fig, ax = plt.subplots(nrows=1, ncols=3, figsize=(15, 3))

ax=ax.ravel()

for i, col in enumerate(cols):

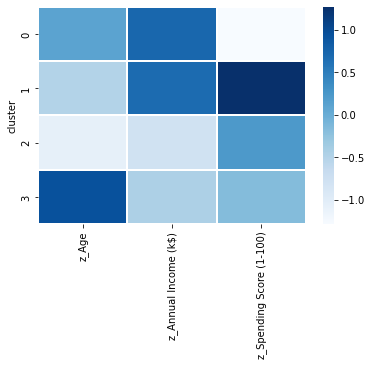
    sns.violinplot(x='cluster', y=col, data=dataset, ax=ax[i])



* ໃນຮູບທີ1 : ແມ່ນຈະສະແດງເຖິງອາຍຸ
* ໃນຮູບທີ່2 : ແມ່ນຈະສະແດງເຖິງລາຍໄດ້ຂອງແຕ່ລະກຸ່ມ
* ໃນຮູບທີ່3 : ແມ່ນຈະສະແດງເຖິງຄະ​ແນນ​ການ​ໃຊ້​ຈ່າຍແຕ່ລະກຸ່ມ​

2.9. ຈົ່ງສະແດງກຸ່ມຂໍ້ມູນດ້ວຍ sns.heatmap(), ພ້ອມອະທິບາຍຜົນ.

sns.heatmap(dx.groupby('cluster').median(), cmap="Blues", linewidths=1);



ໃນຮຸບນີ້ແມ່ນຈະສະແດງແຕ່ລະແຖວໂດຍລະອຽບເຊີ່ງວ່າຖ້າແຖວໃດມີສີທີ່ແຈ້ງແມ່ນກຸ່ມທີ່ມີຈຳນວນຫຼາຍໃນກຸ່ມນັ້ນຕາມລຳດັບລົງມາ