

T1

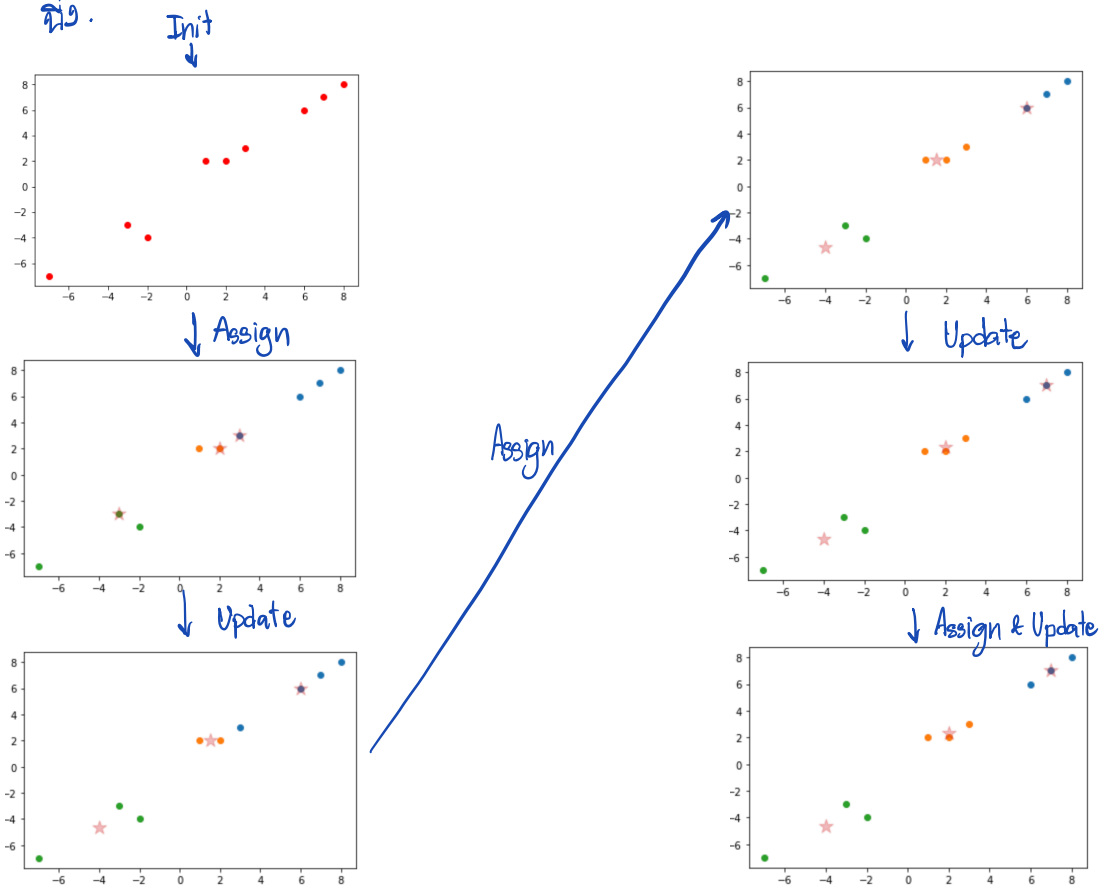
สามารถ Implement ได้ที่ Clary K-Mean

T1. If the starting points are (3,3), (2,2), and (-3,-3). Describe each assign and update step. What are the points assigned? What are the updated centroids? You may do this calculation by hand or write a program to do it.

ธนพงศ์ ช่างสกุล 6231374221.

เมื่อทำการทำการ Assign แต่ละจุดภายในกราฟ จะพบว่าแต่ละจุดจะถูก Cluster ไปตามจุด Centroid ที่ใกล้ที่สุด จากนั้นเมื่อทำการ Update centroid จุดใหม่จะเกิดจากค่า mean ของจุดทั้งหมดภายใน Cluster ซึ่งจะเป็นจุด Centroid ไม่ได้เป็นจุดที่ถูกกำหนดมาตั้งแต่แรกจะอยู่ระหว่างจุดเหล่านั้น

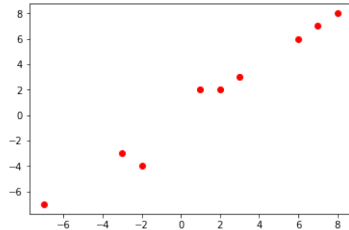
สุดท้ายเมื่อทำการ Assign และ Update ทั้งหมดซ้ำไปประมาณ 2 รอบ จุด Centroid เริ่มนิ่ง.



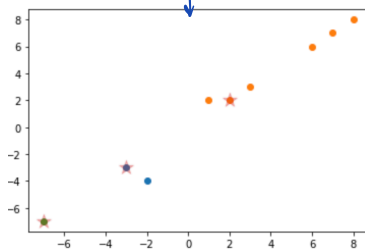
T2. If the starting points are $(-3,-3)$, $(2,2)$, and $(-7,-7)$, what happens?

ลักษณะการทำ Clustering จะเหมือนขั้นตอนหนึ่งเกี่ยวกับจุด Starting points แต่การ Assign และ Update เพียงแค่ 1 จุดจุด Centroid ก็เริ่มแล้ว.

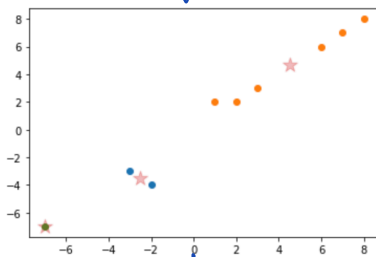
↓ Init



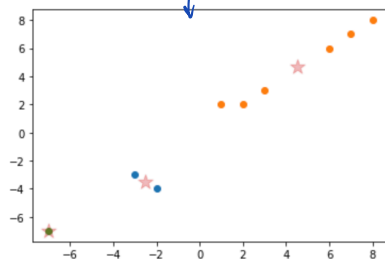
↓ Assign



↓ Update



↓ Assign & Update



T3. Between the two starting set of points in the previous two questions, which one do you think is better? How would you measure the 'goodness' quality of a set of starting points?

In general, it is important to try different sets of starting points when doing k-means.

หาพิจารณาประสิทธิภาพของจุดเริ่มต้น Starting point ด้วยการวัด

Fraction of explained variance (FEV) ที่คำนวณถึงอัตราส่วนระหว่าง Between class variance

และ All-data variance นามว่า Starting point จุดที่หนึ่ง มีค่า FEV ที่มากกว่าจุดที่สอง :

ซึ่งสอดคล้องกับการเปลี่ยนแปลงด้วยประมาณจากจุดต่อมตาม ก็เช่นกัน คือ: ใช้ข้อใด

Starting point ที่ให้การ Clusters ที่แบ่งกลุ่ม และกลุ่มที่มีขนาดเท่ากันทั้งในทั่วจำนวน

ผล: วัตุมีขนาดเท่า: Cluster.

T4. What is the median age of the training set? You can easily modify the age in the dataframe by

```
train["Age"] = train["Age"].fillna(train["Age"].median())
```

Note that you need to modify the code above a bit to fill with mode() because mode() returns a series rather than a single value.

```
1 train['Age'].median()
28.0

[21] 1 train["Age"] = train["Age"].fillna(train["Age"].mode()[0])
    2 train["Age"].head()

0    22.0
1    38.0
2    26.0
3    35.0
4    35.0
Name: Age, dtype: float64
```

T5. Some fields like 'Embarked' are categorical. They need to be converted to numbers first. We will represent S with 0, C with 1, and Q with 2. What is the mode of Embarked? Fill the missing values with the mode. You can set the value of Embarked easily with the following command.

```
train.loc[train["Embarked"] == "S", "Embarked"] = 0
```

Do the same for Sex.

```
[22] 1 mode = train['Embarked'].mode()
    2 mode[0]

'S'

1 train["Embarked"] = train["Embarked"].fillna(mode[0])
2 train["Embarked"].head()

0    S
1    C
2    S
3    S
4    S
Name: Embarked, dtype: object
```

```
1 train.loc[train["Embarked"] == "S", "Embarked"] = 0
2 train.loc[train["Embarked"] == "C", "Embarked"] = 1
3 train.loc[train["Embarked"] == "Q", "Embarked"] = 2
4 train.head()
```

```
1 train.loc[train["Sex"] == "male", "Sex"] = 0
2 train.loc[train["Sex"] == "female", "Sex"] = 1
3 train.head()
```

T6. Write a **logistic regression classifier using gradient descent** as learned in class. Use PClass, Sex, Age, and Embarked as input features. You can extract the features from Pandas to Numpy by

```
data = np.array(train[["PClass", "Sex", "Age", "Embarked"]].values)
```

Check the datatype of each values in data, does it make sense? You can force the data to be of any datatype by using the command

ตาม Clons Logistic_regression และ Preprocessor ใน Code .

T7. Submit a screenshot of your submission (with the scores). Upload your code to courseville.

export.csv

just now by WatweQwee

[add submission details](#)

0.77272

T8. Try adding some higher order features to your training (x_1^2, x_1x_2, \dots). Does this model has better **accuracy on the training set**? How does it perform on the **test set**?

เพื่อเติม Order 9 ที่เหลือ Feature ทุก Feature แล้วพบว่า Accuracy ใน Training set เพิ่มขึ้นเพียง 1% 0.001 และ เที่ยบใน Test set จากคะแนนใน Kaggle พบว่าได้ทำคะแนน จึงอาจสรุปได้ว่า การเพิ่ม Order ภายใน Model ไม่ได้ช่วยประสิทธิภาพในการ Learning เลย.

```
1 model11.accuracy(), model12.accuracy()
(0.7800224466891134, 0.7811447811447811)
```

export2.csv

a few seconds ago by WatweQwee

[add submission details](#)

0.77272

T9. What happens if you reduce the amount of features to just Sex and Age?

พบว่า Model มีประสิทธิภาพที่ต่ำลงเล็กน้อยจากผลของ Accuracy และ คะแนน Test set ใน Kaggle

```
1 model13.accuracy()
0.7777777777777778
```

export3.csv

34 minutes ago by WatweQwee

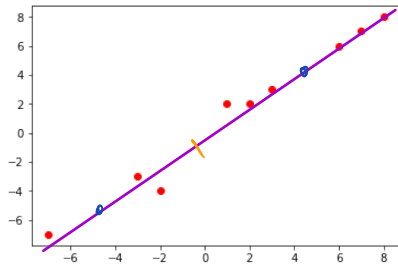
[add submission details](#)

0.74401

OT1. What would be the best K for this question? Describe your reasoning.

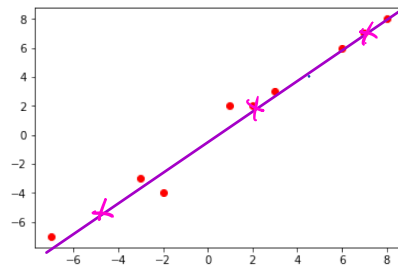
$K=2$

พิจารณา ลักขณ: ข้อมูลเป็นลักขณ: เส้นตรง โดยมีกรแบ่งกลุ่มบนเส้นตรง.

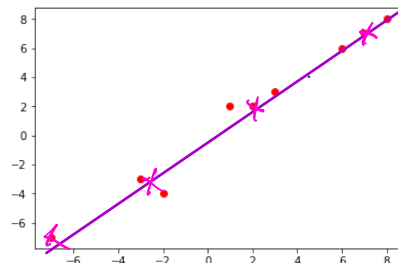


ซึ่งเราสามารถทำ Cluster บนเส้นตรงนี้ได้.

ถ้าเราแบ่ง $K=2$ ข้อมูลจะถูกแบ่งเป็น 2 พังแนวนอน โดยฟังก์ชันจะมีสามจุด จุด Centroid ดังกล่าวจะอยู่ที่ประมาณจุดสี่ห้าเงินตามรูป และเส้นแบ่งจะทว่าสอง Cluster จะเป็นสี่เหลี่ยม. ซึ่งจากสมมติฐานของ K Mean ที่ว่าแต่ละ Cluster จะมีสี่เหลี่ยมที่ใกล้เคียงกัน



แต่ $K=3$ ถ้าตามข้อ T1 จะพบว่า จุด Centroid จะอยู่ที่จุดสี่เหลี่ยม ซึ่งจะพบว่า รัศมีของ 1/10 จะจะไม่เท่ากัน. ดังนั้น $K=3$ อาจไม่ดีที่สุด



แต่ถ้า $K=4$ มันจะซับซ้อนโดยจุดหนึ่งของจุด Centroid เป็น Data point ซึ่งอาจกล่าวได้ว่า K มากเกินไป

OT 4 Let $A \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{m \times n}$.

$$\nabla_A \text{tr} AB = B^T$$

$$\text{tr} AB = \sum_{i=1}^n \sum_{j=1}^m A_{ij} B_{ji}$$

$$\nabla_A \text{tr} AB = \begin{bmatrix} \frac{\partial \text{tr} AB}{\partial A_{11}} & \dots & \frac{\partial \text{tr} AB}{\partial A_{1m}} \\ \vdots & & \vdots \\ \frac{\partial \text{tr} AB}{\partial A_{n1}} & \dots & \frac{\partial \text{tr} AB}{\partial A_{nm}} \end{bmatrix}$$

$$= \begin{bmatrix} B_{11} & \dots & B_{m1} \\ \vdots & & \vdots \\ B_{1n} & \dots & B_{mn} \end{bmatrix}$$

$$= B^T$$

OT 5 $\nabla_{A^T} f(A) = (\nabla_A f(A))^T$

Let $A \in \mathbb{R}^{n \times m}$

$$\nabla_{A^T} f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}^T} & \dots & \frac{\partial f}{\partial A_{1n}^T} \\ \vdots & & \vdots \\ \frac{\partial f}{\partial A_{m1}^T} & \dots & \frac{\partial f}{\partial A_{mn}^T} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \dots & \frac{\partial f}{\partial A_{n1}} \\ \vdots & & \vdots \\ \frac{\partial f}{\partial A_{1m}} & \dots & \frac{\partial f}{\partial A_{nm}} \end{bmatrix}$$

1 2 3

2 1 2 3

3 1 3 2

$$= \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \dots & \frac{\partial f}{\partial A_{1m}} \\ \vdots & & \vdots \\ \frac{\partial f}{\partial A_{n1}} & \dots & \frac{\partial f}{\partial A_{nm}} \end{bmatrix}^T$$

$$= (\nabla_A f(A))^T$$