Unsupervised ML part II : Introduction to Clustering and Principle Component Analysis in Python

อ.ดร.ปัญญนัท อ้นพงษ์

ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร

aonpong_p@su.ac.th

Outline

- **Part II introduction**
- Introduction to Clustering
- Dimensionality Reduction
  - PCA (Principal Component Analysis)
    - แนวคิดของ PCA
    - กระบวนการ
    - รายละเอียดของผลลัพธ์
    - เมื่อข้อมูลมีมิติมากขึ้น
  - เทคนิค Dimensionality Reduction อื่น ๆ ที่น่าสนใจ

Part II introduction

- สัดส่วนคะแนน

Final	40%
- สอบปลายภาค Lecture	19% ?
- สอบปลายภาค Lab	19% ?
- เข้าเรียน	2% ?
Assignment	10%

Part II introduction

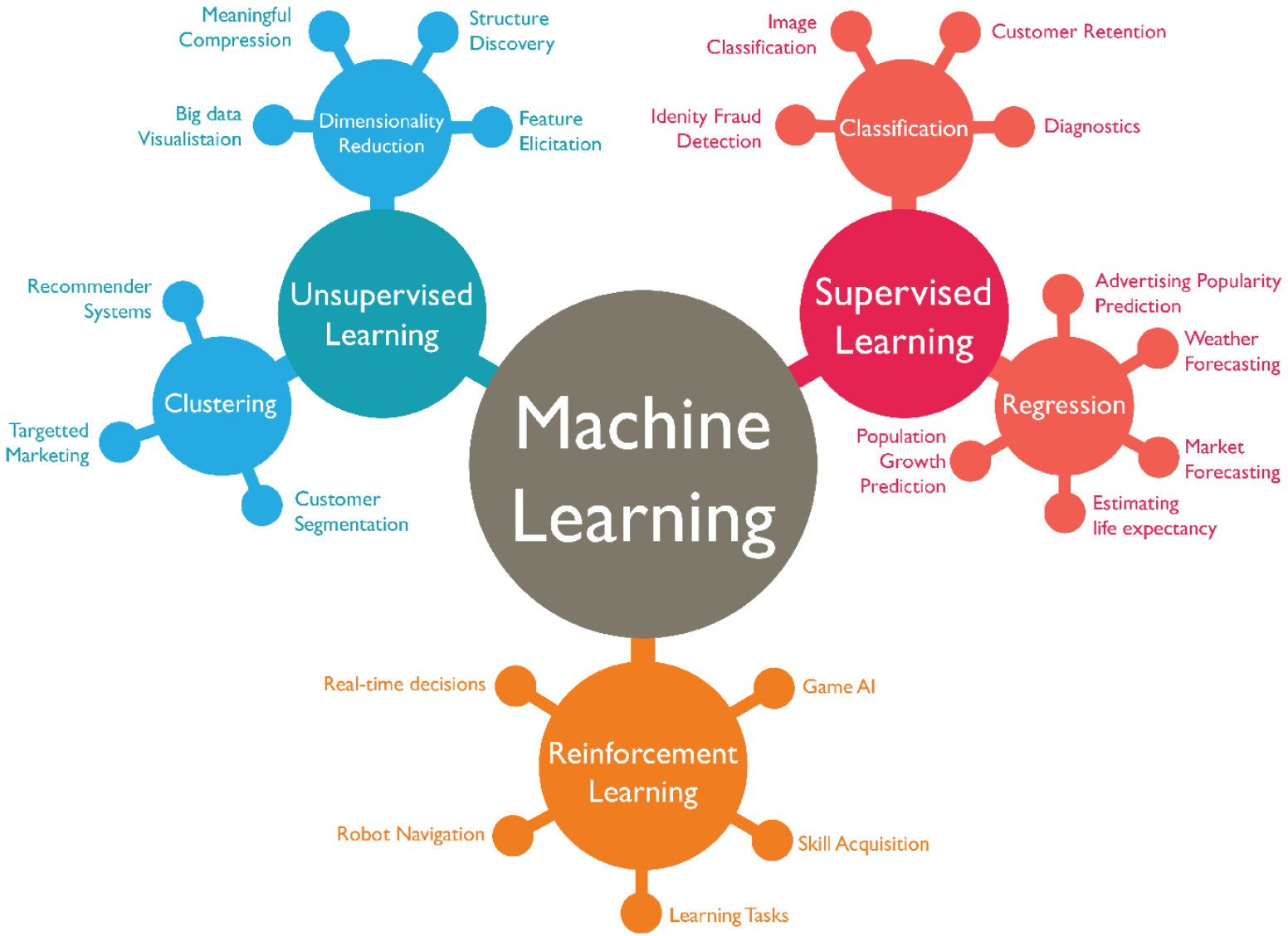
- แผนการเรียน

ชีต	วันที่ (จันทร์)	เนื้อหา	วันที่ (พุธ)	เนื้อหา
1	จ. 30 มกราคม 2566	การแบ่งกลุ่มแบบต่าง ๆ (Clustering) Principle Component Analysis (PCA)	พ. 1 กุมภาพันธ์ 2566	ประกาศ Assignment ติดตั้ง Anaconda/PyCharm PCA in Python
2	จ. 6 กุมภาพันธ์ 2566	การแบ่งกลุ่มด้วย k-Means Clustering 1	พ. 8 กุมภาพันธ์ 2566	k-Means in Python
3	จ. 13 กุมภาพันธ์ 2566	การแบ่งกลุ่มด้วย k-Means Clustering 2	พ. 15 กุมภาพันธ์ 2566	k-Means in Python ติดตามความคืบหน้า
4	จ. 20 กุมภาพันธ์ 2566	การแบ่งกลุ่มลำดับชั้น (Hierarchical) 1	พ. 22 กุมภาพันธ์ 2566	Hierarchical in Python
5	จ. 27 กุมภาพันธ์ 2566	การแบ่งกลุ่มลำดับชั้น (Hierarchical) 2	พ. 1 มีนาคม 2566	Hierarchical in Python ติดตามความคืบหน้า
6	จ. 6 มีนาคม 2566	วันหยุดราชการ	พ. 8 มีนาคม 2566	Gaussian Mixture Model (lect. ชดเชย)
7	จ. 13 มีนาคม 2566	การประยุกต์ใช้กับปัญหา	พ. 15 มีนาคม 2566	Presentation

Outline

- Part II introduction
- **Introduction to Clustering**
- Dimensionality Reduction
  - PCA (Principal Component Analysis)
    - แนวคิดของ PCA
    - กระบวนการ
    - รายละเอียดของผลลัพธ์
    - เมื่อข้อมูลมีมิติมากขึ้น
  - เทคนิค Dimensionality Reduction อื่น ๆ ที่น่าสนใจ

Introduction to Clustering



Introduction to Clustering

สิ่งที่มีนิยามใกล้เคียงกับ Clustering คือ Classification

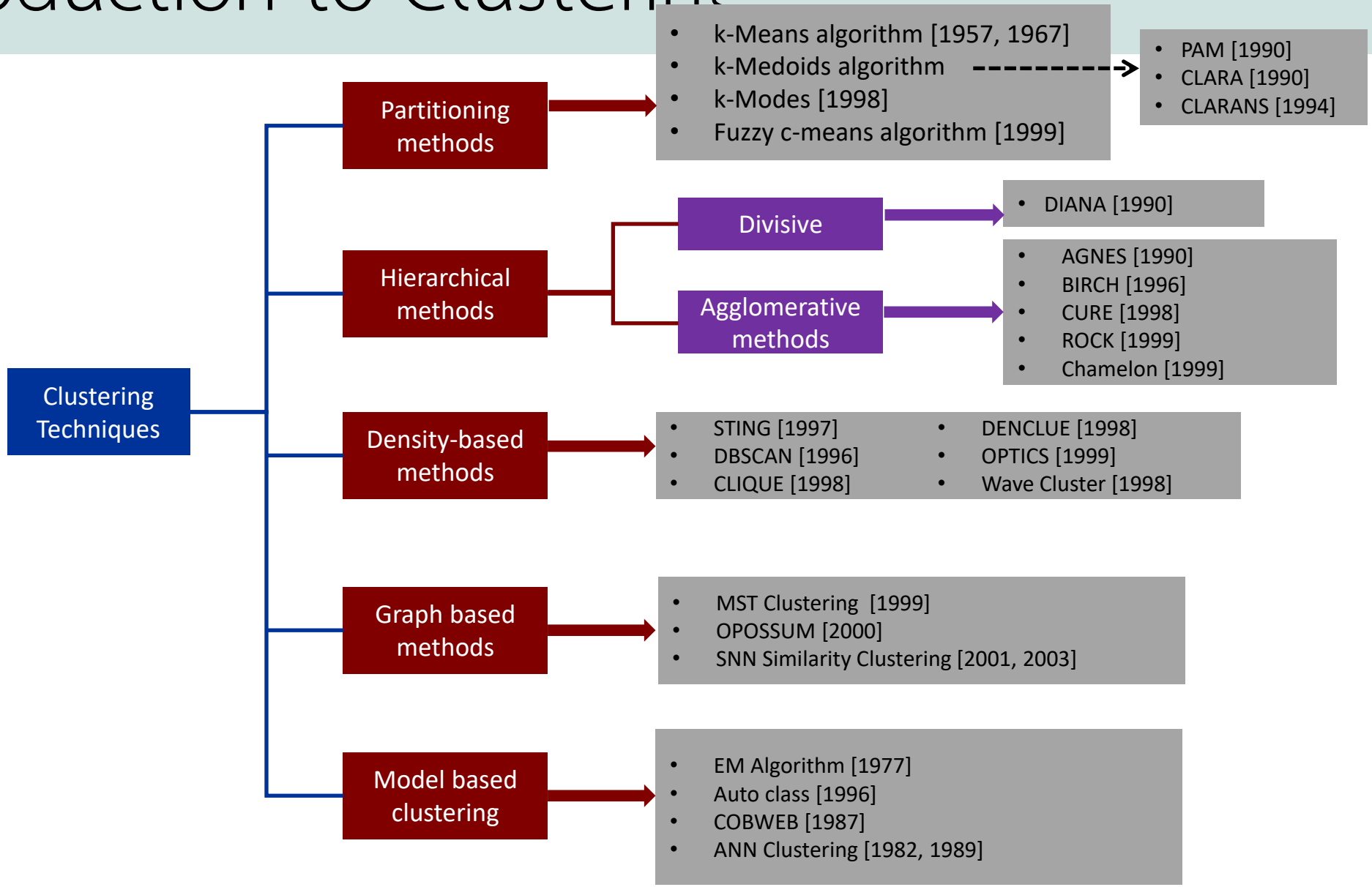
- Classification (การจำแนก) สิ่งที่ระบบต้องรู้เมื่อทำการเทรน ประกอบไปด้วยข้อมูลอัตลักษณ์ของวัตถุและ Class Label ของข้อมูล เช่น
  - ❖มีปีก มีขนมันเงา กระดูกเป็นรูพรุน น้ำหนักเบา มีสองขา **กำหนด Class label คือ นก**
  - ❖ไม่มีขา มีเกล็ด น้ำหนักเบา **กำหนด Class label คือ งู**
  - ดังนั้น นักศึกษาคิดว่า ถ้ามีวัตถุชิ้นหนึ่ง มีสองขา น้ำหนักเบา มีขนมันเงา วัตถุชิ้นนี้น่าจะเป็นสิ่งใด? (งู / นก)

Introduction to Clustering

- Clustering (การจัดกลุ่ม) สิ่งที่ระบบต้องรู้ ประกอบไปด้วยข้อมูลอัตลักษณ์ของวัตถุแต่ไม่จำเป็นต้องรู้ Class Label ของข้อมูล เช่น
  - ❖มีปีก มีขนมันเงา กระดูกเป็นรูพรุน น้ำหนักเบา มีสองขา
  - ❖ไม่มีขา มีเกล็ด น้ำหนักเบา

  - ดังนั้น นักศึกษาคิดว่า ถ้ามีวัตถุชิ้นหนึ่ง มีสองขา น้ำหนักเบา มีขนมันเงา วัตถุชิ้นนี้น่าจะเป็นสิ่งใด?
    - กรณีนี้ เราจะระบุไม่ได้ว่าวัตถุชิ้นนี้คืออะไร แต่เราจะรู้ว่ามันน่าจะเป็นวัตถุที่อยู่กลุ่มเดียวกับตัวไหน

Introduction to Clustering



Introduction to Clustering

ในการเรียนครึ่งเทอมหลังนี้ เราจะศึกษาเกี่ยวกับการเรียนรู้แบบ Unsupervised เพียงบางส่วน โดยจะครอบคลุมเนื้อหาดังนี้

- Partitioning
  - k-Means algorithm
  - PAM (k-Medoids algorithm)
- Hierarchical
  - DIANA (divisive algorithm)
  - AGNES
  - ROCK

  } (Agglomerative algorithm)
- Density – Based
  - DBSCAN

Introduction to Clustering

Hierarchical algorithms: แบ่งแบบใช้ผลจากการแบ่งครั้งก่อนหน้าในการแบ่งครั้งถัดไป

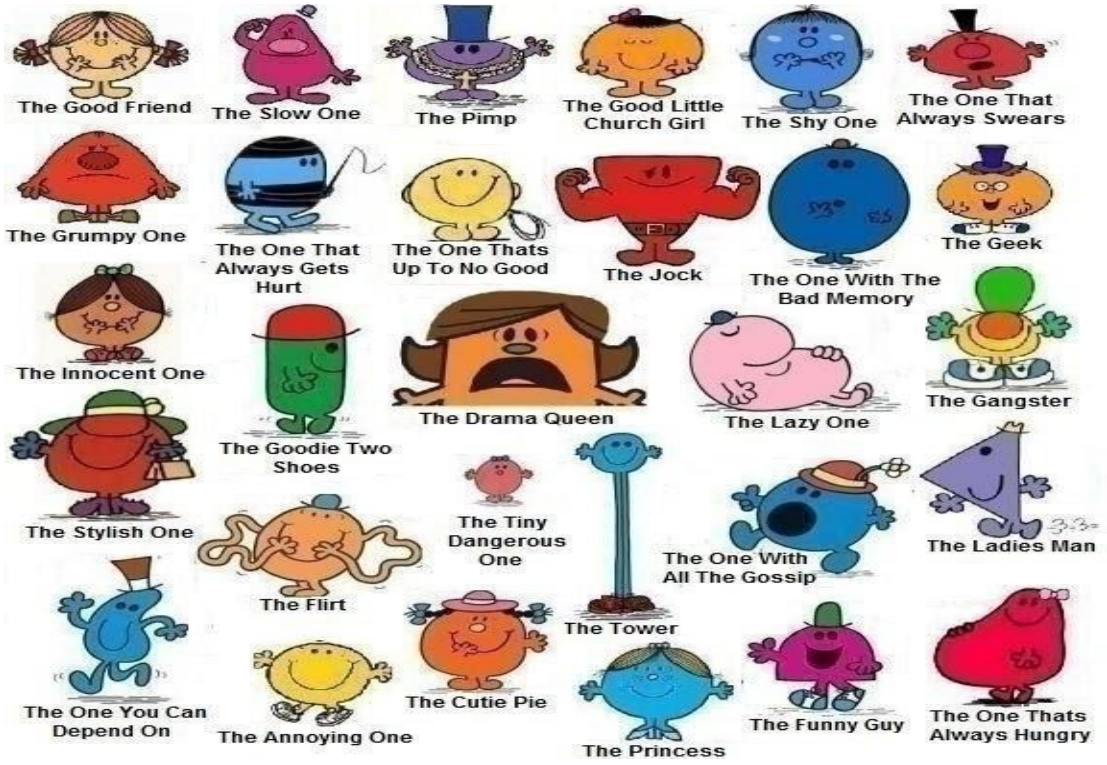
- Agglomerative ("bottom-up"): แบ่งจากก้อนเล็ก ๆ แล้วค่อย ๆ จับกลุ่มโดยนำตัวที่มีความคล้ายกันที่สุดมารวมเป็นกลุ่มเดียวกัน แล้วค่อย ๆ เพิ่มตัวที่พิจารณาเข้าไปในกระบวนการเรื่อย ๆ
- Divisive ("top-down"): แบ่งจาก 1 ก้อนใหญ่เป็นก้อนเล็ก ๆ และเพิ่มการแบ่งกลุ่มต่อไปเรื่อย ๆ โดยใช้ผลจากการแบ่งครั้งก่อนหน้าเป็นต้นกำเนิด

Partitional clustering: แบ่งแบบแบ่งทุกกลุ่มออกจากกันในครั้งเดียว มีหลายเทคนิค เช่น

- K-means and derivatives
- Fuzzy c-means clustering
- QT clustering algorithm

Introduction to Clustering

- **Classification** เป็น Machine Learning ประเภท **Supervised**
- **Clustering** เป็น Machine Learning ประเภท **Unsupervised** (Unsupervised Classification)



Outline

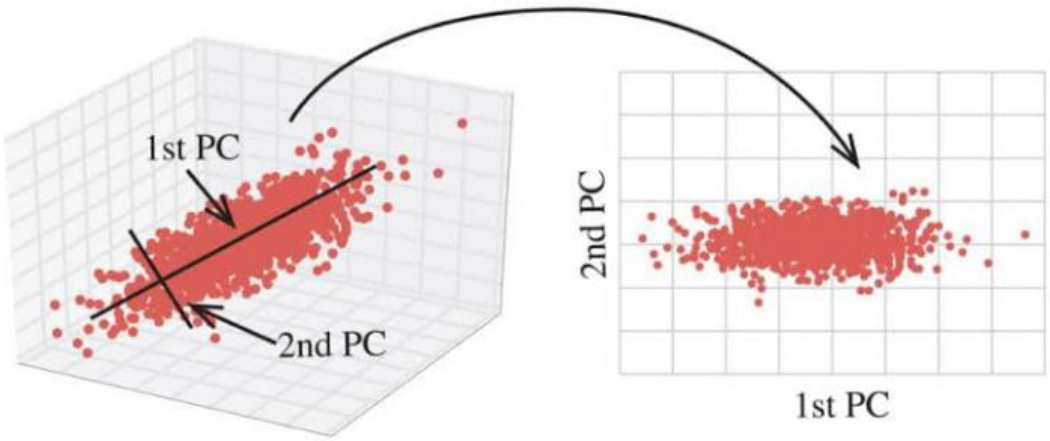
- Part II introduction
- Introduction to Clustering
- **Dimensionality Reduction**
  - **PCA (Principal Component Analysis)**
    - **แนวคิดของ PCA**
    - กระบวนการ
    - รายละเอียดของผลลัพธ์
    - เมื่อข้อมูลมีมิติมากขึ้น
  - เทคนิค Dimensionality Reduction อื่น ๆ ที่น่าสนใจ

Main Idea of PCA

- PCA แปลเป็นภาษาไทยตรงตัวว่า การวิเคราะห์องค์ประกอบหลัก
- เรามองว่าข้อมูลทั้งหมดมีองค์ประกอบหลายอย่างรวมกันอยู่
- แต่เราทราบกันดีว่า องค์ประกอบหลายตัวไม่ได้เกี่ยวข้องกับงานที่เราจะทำ
  - เช่นการใช้ข้อมูลที่มีฟีเจอร์ 100 ตัว การที่จะบอกว่าฟีเจอร์ทั้งหมดเกี่ยวข้องกับสิ่งที่เราต้องการพยากรณ์แทบจะเป็นไปไม่ได้เลย
- การทำ PCA จึงเป็นการหาความสัมพันธ์ของข้อมูลภายในด้วยกันเอง และให้ความสัมพันธ์กับฟีเจอร์ที่มีความสัมพันธ์สูงกว่า

Main Idea of PCA

- ถ้าเป็นแบบนั้น ทำไม PCA จึงไม่ใช่ Feature Selection
  - เหตุผลหลักของการทำ PCA คือการลดมิติข้อมูล โดยการลดจำนวนฟีเจอร์ (คล้าย Feature Selection)
  - แต่การลดฟีเจอร์ของ PCA นั้นไม่ใช่การตัดออก แต่เราจะแปลงสภาพ (Transform) ข้อมูลให้อยู่ในรูปแบบอื่นแทน โดยเราจะใส่ใจความสัมพันธ์ของข้อมูลกับงานที่จะทำด้วย
  - นั่นหมายความว่าฟีเจอร์ส่วนใหญ่จะยังคงมีตัวตนแฝงอยู่ในข้อมูลด้วย ไม่ได้ถูกตัดออกไปแบบ Feature Selection อย่างไรก็ตาม ฟีเจอร์ที่มีความสัมพันธ์กับงานต่ำจะแสดงออกน้อยกว่าฟีเจอร์ที่มีความสัมพันธ์สูง



Main Idea of PCA

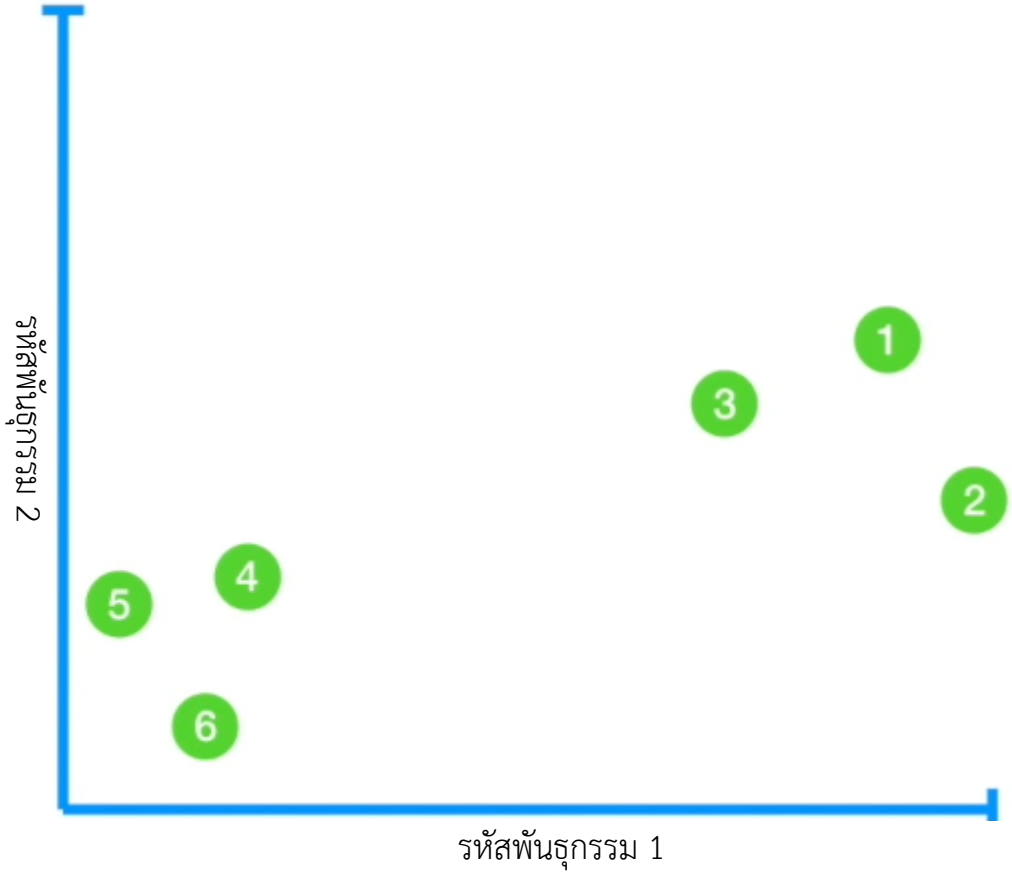
- สมมติข้อมูลที่จะใช้เป็นดังนี้ (ตัวอย่างจาก Statquest by JoshStamer)

	หนู 1	หนู 2	หนู 3	หนู 4	หนู 5	หนู 6	...
รหัสพันธุกรรม 1	10	11	8	3	2	1	...
รหัสพันธุกรรม 2	6	4	5	3	2.8	1	...

Main Idea of PCA

- นำข้อมูลในตารางมาพล็อตกราฟ

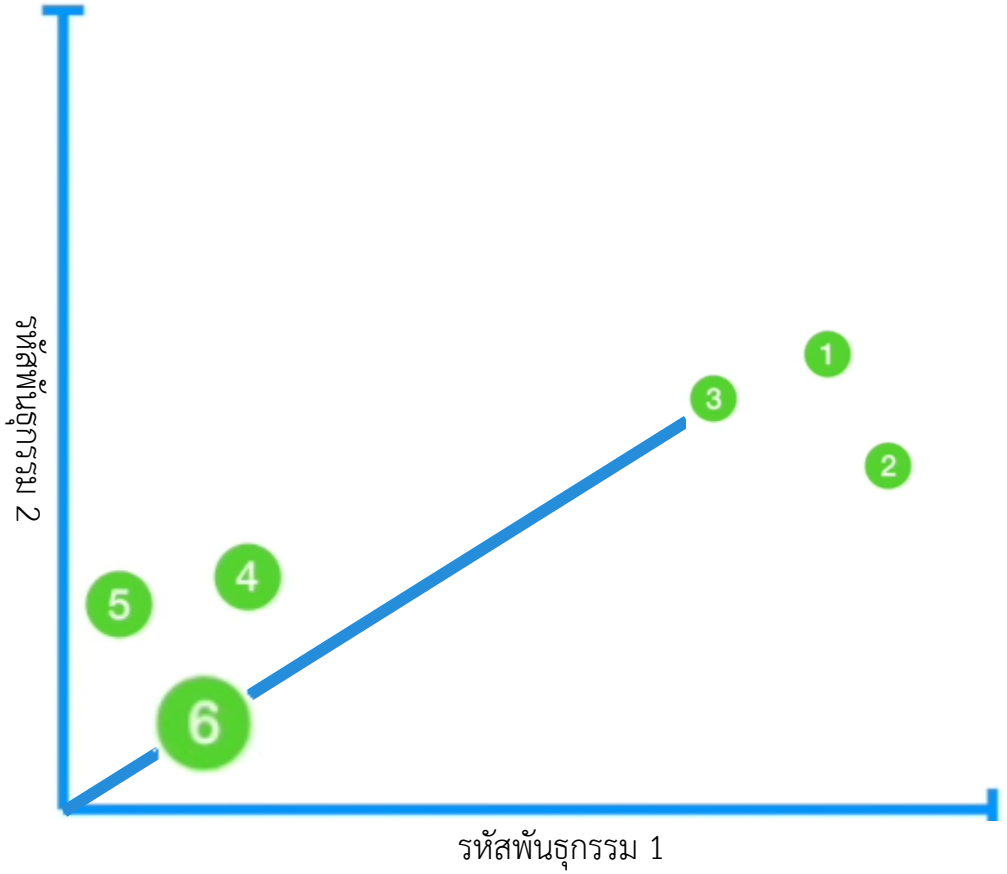
	หนู 1	หนู 2	หนู 3	หนู 4	หนู 5	หนู 6	...
รหัสพันธุกรรม 1	10	11	8	3	2	1	...
รหัสพันธุกรรม 2	6	4	5	3	2.8	1	...



Main Idea of PCA

- ถ้าเพิ่มข้อมูลอีก 1 ฟีเจอร์ กราฟที่วาดจะกลายเป็น 3 มิติ

	หนู 1	หนู 2	หนู 3	หนู 4	หนู 5	หนู 6	...
รหัสพันธุกรรม 1	10	11	8	3	2	1	...
รหัสพันธุกรรม 2	6	4	5	3	2.8	1	...
รหัสพันธุกรรม 3	12	9	10	2.5	1.3	2	...



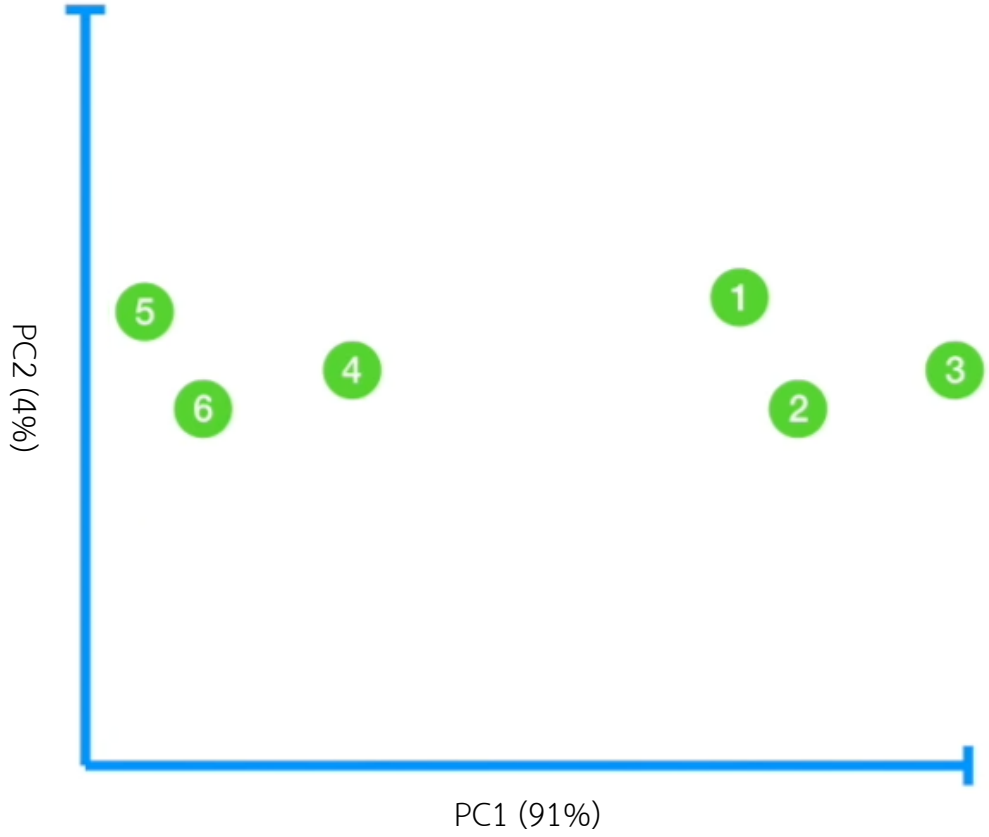
Main Idea of PCA

- ถ้าเพิ่มข้อมูลเป็น 4 ฟีเจอร์ เมื่อนำมาพล็อตกราฟ ในทางทฤษฎี เราจะได้กราฟ 4 มิติ แต่ในทางปฏิบัติ เราไม่มีความสามารถจะพล็อตกราฟแบบนั้นออกมาให้เห็นได้

	หนู 1	หนู 2	หนู 3	หนู 4	หนู 5	หนู 6	...
รหัสพันธุกรรม 1	10	11	8	3	2	1	...
รหัสพันธุกรรม 2	6	4	5	3	2.8	1	...
รหัสพันธุกรรม 3	12	9	10	2.5	1.3	2	...
รหัสพันธุกรรม 4	5	7	6	2	4	7	...

Main Idea of PCA

- แต่การแสดงผลข้อมูลทั้งหมดออกมาในคราวเดียวสามารถทำได้ผ่าน PCA
- PCA สามารถลดข้อมูล 4 มิติ เหลือ 2 มิติ โดยมีข้อมูลทั้งหมดเป็นส่วนประกอบ ไม่ใช่ตัดทิ้ง เพียงเท่านี้ เราก็สามารถพล็อตกราฟข้อมูลรหัสพันธุกรรมทั้ง 4 ได้
- นอกจากนี้ PCA ยังบอกเราได้ด้วยว่าข้อมูลตัวไหนที่สำคัญต่อการทำนายหรือจัดกลุ่ม



Outline

- Part II introduction
- Introduction to Clustering
- **Dimensionality Reduction**
  - **PCA (Principal Component Analysis)**
    - แนวคิดของ PCA
    - **กระบวนการ**
    - รายละเอียดของผลลัพธ์
    - เมื่อข้อมูลมีมิติมากขึ้น
  - เทคนิค Dimensionality Reduction อื่น ๆ ที่น่าสนใจ

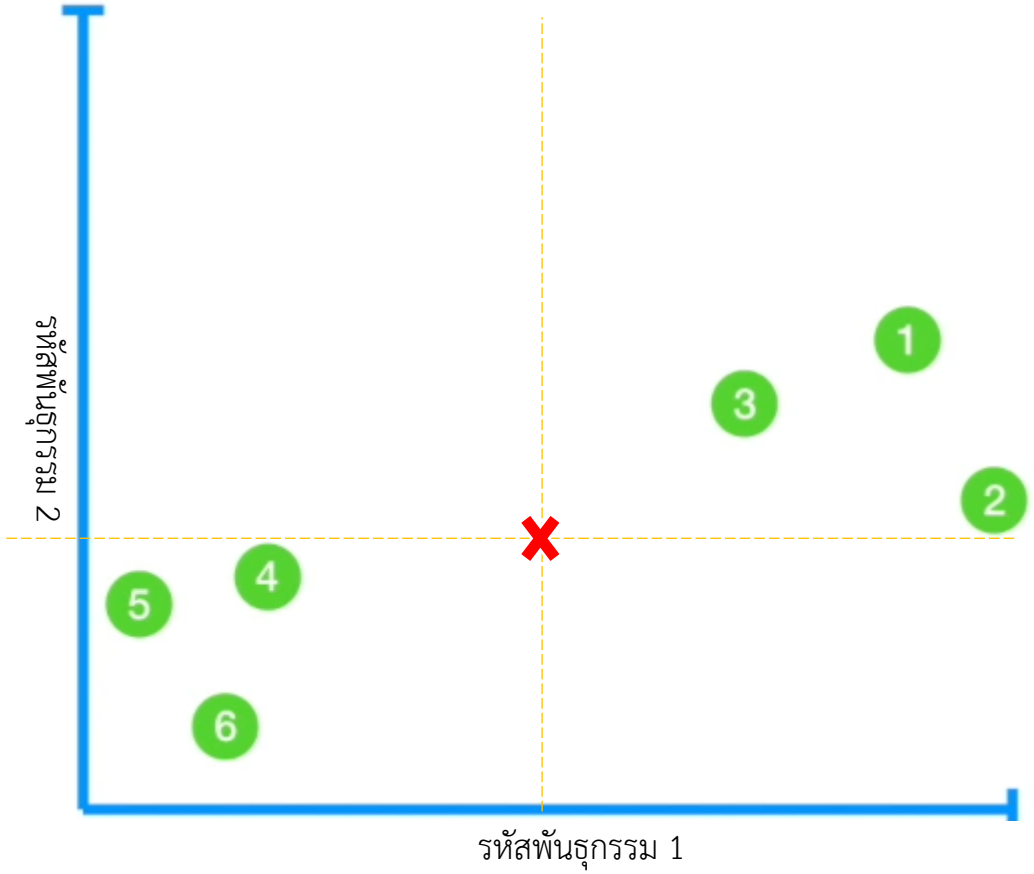
กระบวนการ PCA

ในขั้นตอนตัวอย่าง จะขออธิบายแบบง่าย ๆ ก่อน โดยเริ่มจากตัวอย่างที่มีเจอร์แค่ 2 ตัว

	หนู 1	หนู 2	หนู 3	หนู 4	หนู 5	หนู 6	...
รหัสพันธุกรรม 1	10	11	8	3	2	1	...
รหัสพันธุกรรม 2	6	4	5	3	2.8	1	...

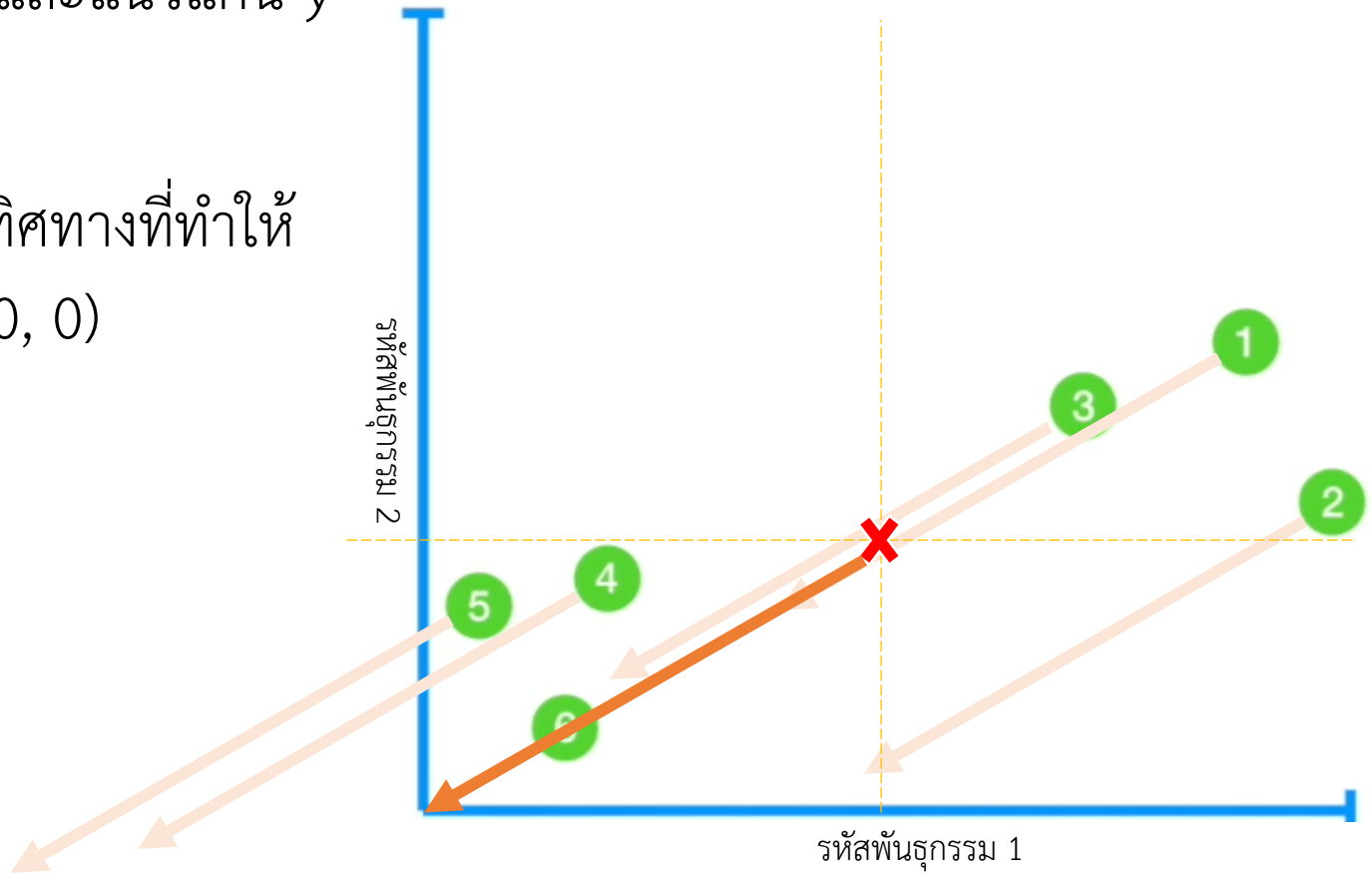
ขั้นตอนวิธี PCA

1. หาค่าเฉลี่ยของข้อมูลในแนวแกน x และแนวแกน y
2. หาจุดตัดของค่าเฉลี่ยทั้งสองแกนนั้น



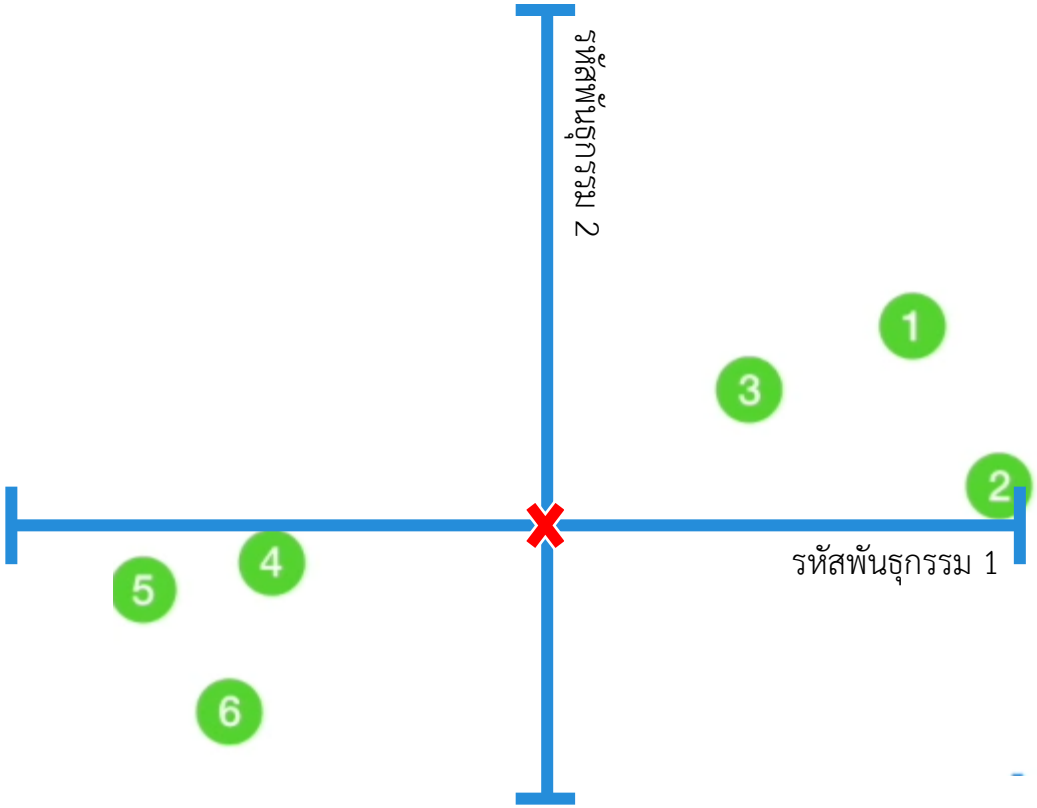
ขั้นตอนวิธี PCA

1. หาค่าเฉลี่ยของข้อมูลในแนวแกน x และแนวแกน y
2. หาจุดตัดของค่าเฉลี่ยทั้งสองแกนนั้น
3. เลื่อนข้อมูลทั้งหมดพร้อมๆกันไปในทิศทางที่ทำให้จุดตัด (กากบาทสีแดง) ไปอยู่ที่จุด (0, 0)



ขั้นตอนวิธี PCA

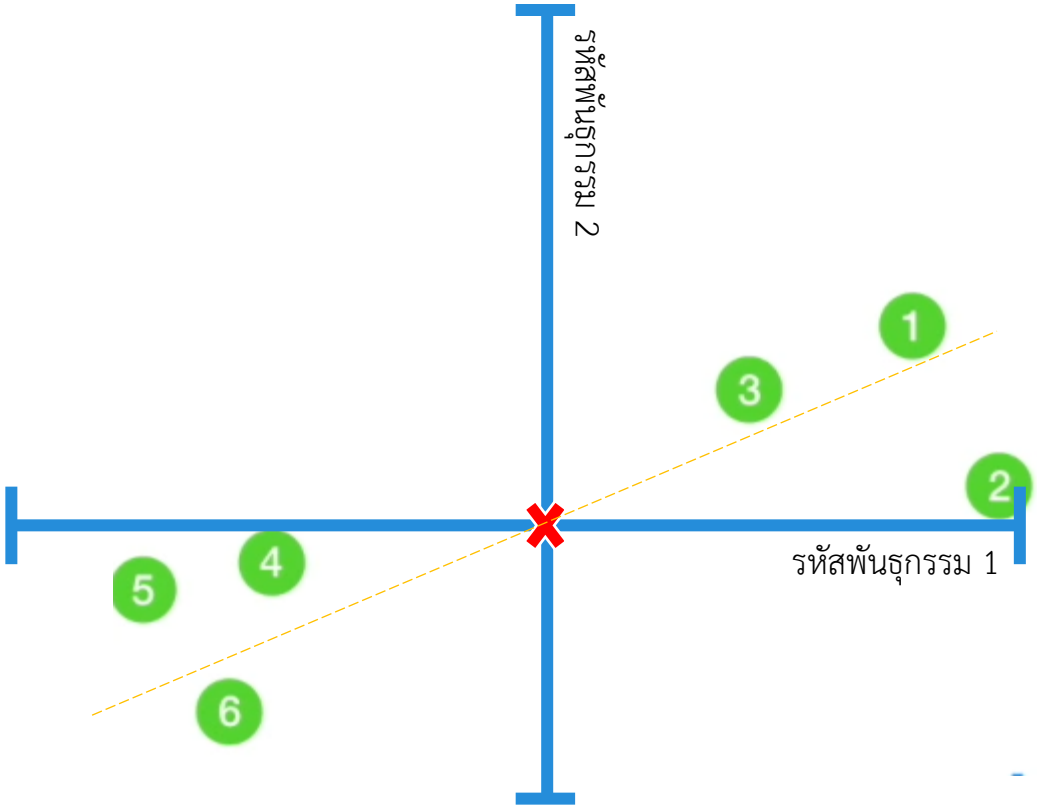
1. หาค่าเฉลี่ยของข้อมูลในแนวแกน x และแนวแกน y
2. หาจุดตัดของค่าเฉลี่ยทั้งสองแกนนั้น
3. เลื่อนข้อมูลทั้งหมดพร้อมๆกันไปในทิศทางที่ทำให้จุดตัด (กากบาทสีแดง) ไปอยู่ที่จุด (0, 0)



ขั้นตอนวิธี PCA

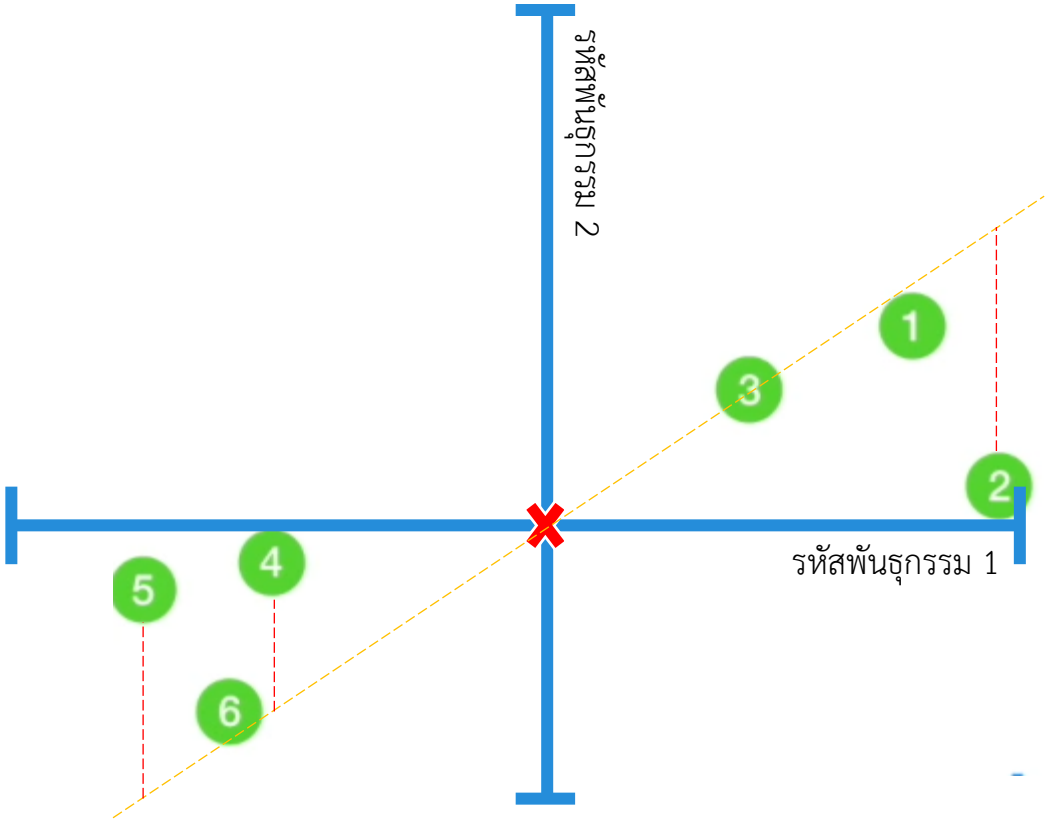
1. หาค่าเฉลี่ยของข้อมูลในแนวแกน x และแนวแกน y
2. หาจุดตัดของค่าเฉลี่ยทั้งสองแกนนั้น
3. เลื่อนข้อมูลทั้งหมดพร้อมๆกันไปในทิศทางที่ทำให้จุดตัด (กากบาทสีแดง) ไปอยู่ที่จุด (0, 0)
4. วาดเส้นตรงผ่านจุด (0, 0) โดยสุ่ม Slope และค่อยๆ หา slope ที่จะฟิตข้อมูลทั้งหมดได้มากที่สุด

- ขั้นตอนและวิธีคิดส่วนนี้**ไม่เหมือน linear regression**



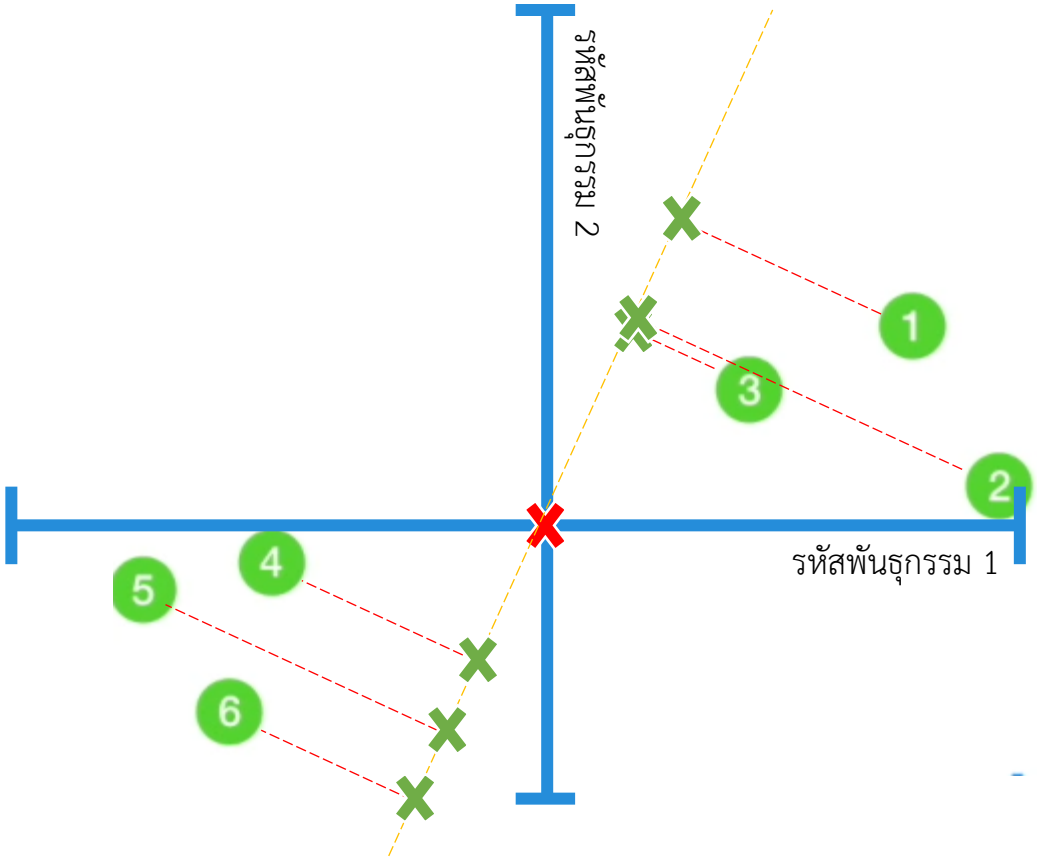
ขั้นตอนวิธี PCA

- ใน linear regression เราจะหา residual เพื่อหาความผิดพลาดในแนวแกนเดียวเท่านั้น



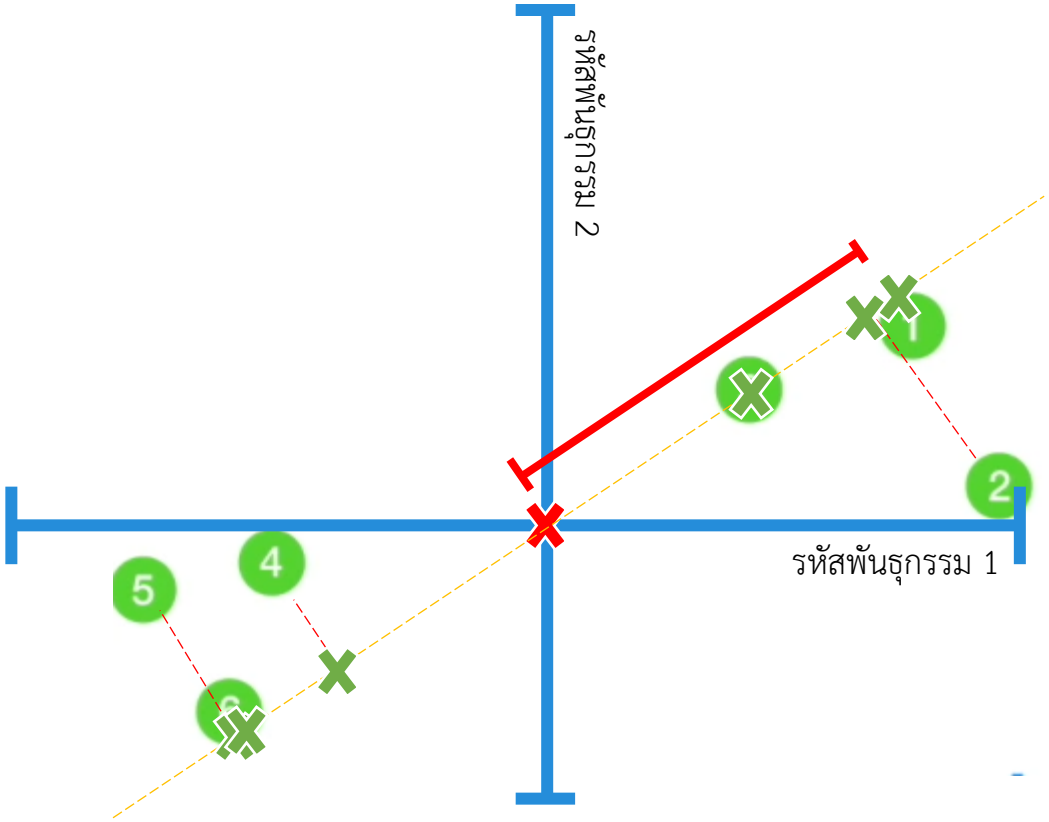
ขั้นตอนวิธี PCA

- ใน linear regression เราจะหา residual เพื่อหาความผิดพลาดในแนวแกนเดียวเท่านั้น
- **แต่ใน PCA เราจะ project แบบตั้งฉาก**



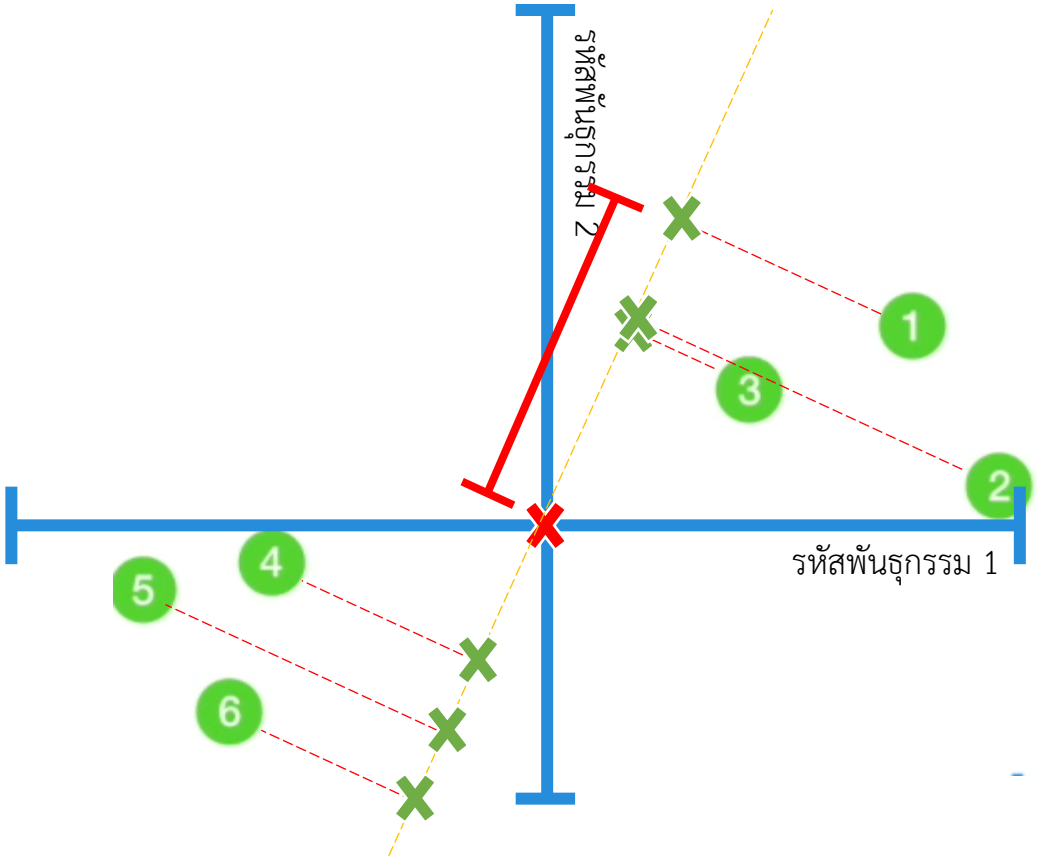
ขั้นตอนวิธี PCA

- ใน linear regression เราจะหา residual เพื่อหาความผิดพลาดในแนวแกนเดียวเท่านั้น
- **แต่ใน PCA เราจะ project แบบตั้งฉาก**
- ที่จริงแล้วเราไม่ได้ต้องการได้เส้นที่ใกล้กับจุดที่สุด แต่เราต้องการได้เส้นที่ทำให้จุดทุกจุดห่างจากจุด (0, 0) มากที่สุดต่างหาก



ขั้นตอนวิธี PCA

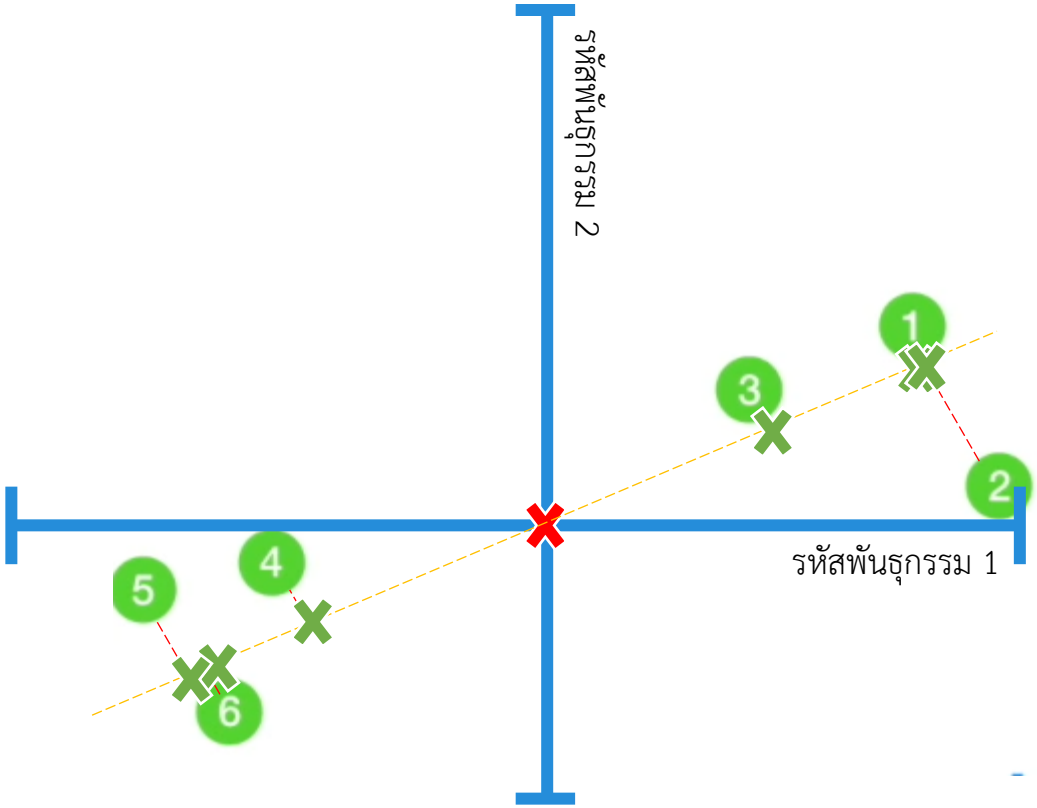
- ใน linear regression เราจะหา residual เพื่อหาความผิดพลาดในแนวแกนเดียวเท่านั้น
- **แต่ใน PCA เราจะ project แบบตั้งฉาก**
  - ที่จริงแล้วเราไม่ได้ต้องการได้เส้นที่ใกล้กับจุดที่สุด แต่เราต้องการได้เส้นที่ทำให้จุดทุกจุดห่างจากจุด (0, 0) มากที่สุดต่างหาก
  - ในเชิงสถิติ เราจะใช้ slope ที่ทำให้ผลรวมของกำลังสองของระยะนี้ (จาก (0, 0) ถึงจุดกากบาทสีเขียวแต่ละจุด) น้อยที่สุดที่หาได้ เราเรียกค่านี้ว่า Sum of Squared distances หรือ SS(distances)



ขั้นตอนวิธี PCA

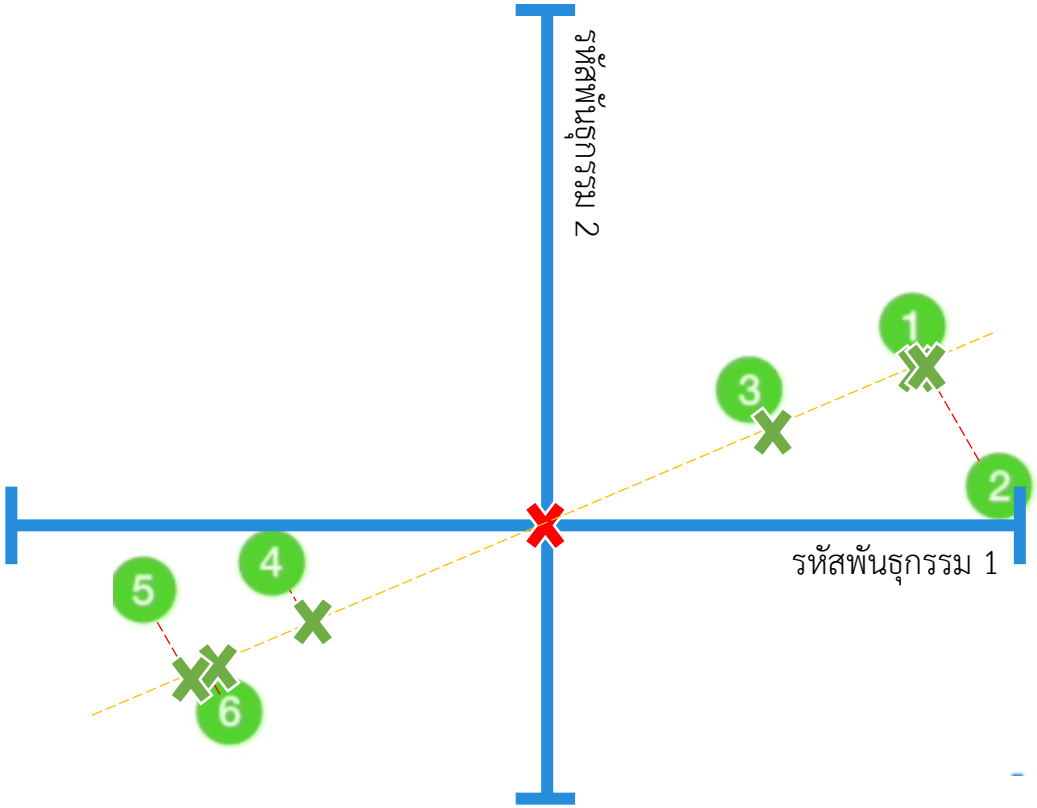
- ใน linear regression เราจะหา residual เพื่อหาความผิดพลาดในแนวแกนเดียวเท่านั้น
- **แต่ใน PCA เราจะ project แบบตั้งฉาก**
- ที่จริงแล้วเราไม่ได้ต้องการได้เส้นที่ใกล้กับจุดที่สุด แต่เราต้องการได้เส้นที่ทำให้จุดทุกจุดห่างจากจุด (0, 0) มากที่สุดต่างหาก
- สุดท้ายเรามาจบที่ slope ดังรูป

(Biggest SS(distance))



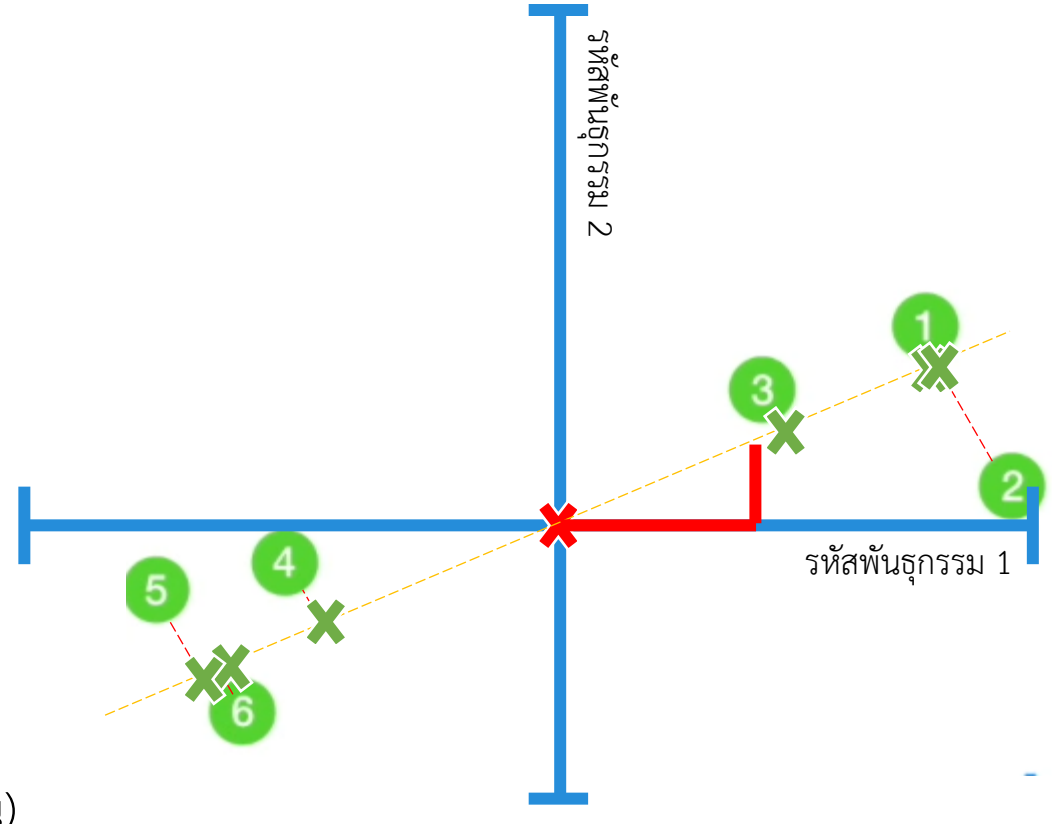
ขั้นตอนวิธี PCA

1. หาค่าเฉลี่ยของข้อมูลในแนวแกน x และแนวแกน y
2. หาจุดตัดของค่าเฉลี่ยทั้งสองแกนนั้น
3. เลื่อนข้อมูลทั้งหมดพร้อมๆกันไปในทิศทางที่ทำให้จุดตัด (กากบาทสีแดง) ไปอยู่ที่จุด (0, 0)
4. วาดเส้นตรงผ่านจุด (0, 0) โดยสุ่ม Slope และค่อยๆ หา slope ที่จะฟิตข้อมูลทั้งหมดได้มากที่สุด
5. ตีเส้นฉากและ Project จุดต่างๆ ลงบนเส้นตรงที่หาได้จากข้อ 4



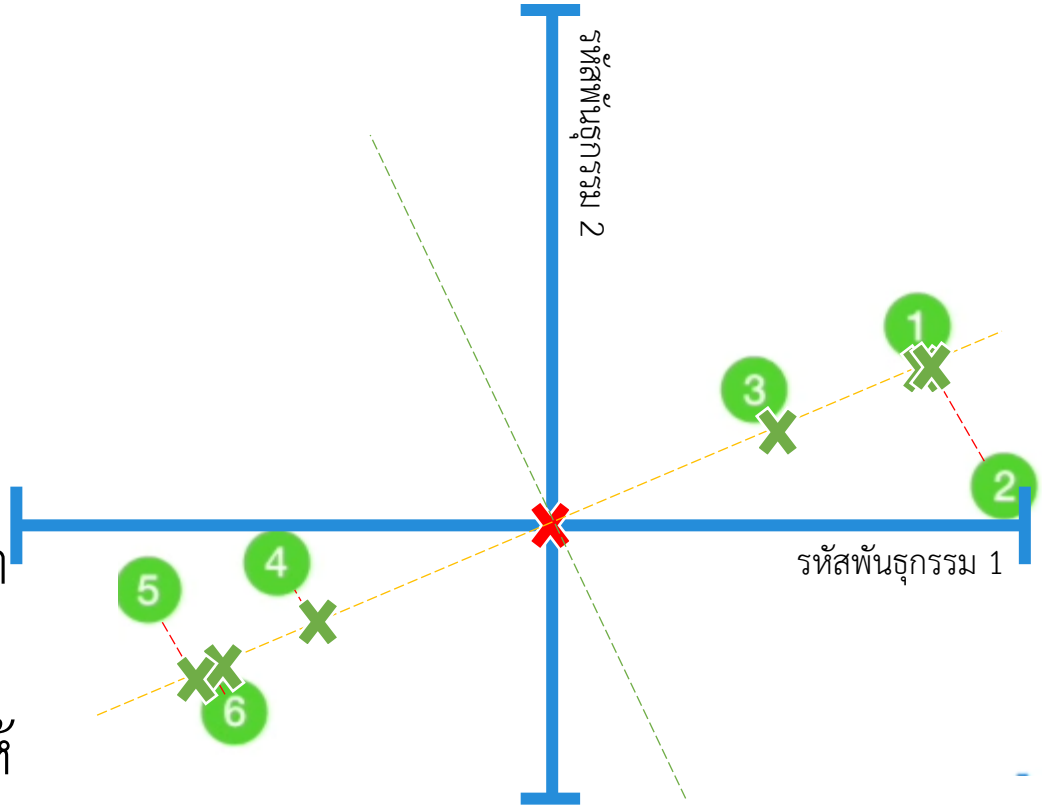
ขั้นตอนวิธี PCA

- เส้นสีเหลืองที่มีกากบาทสีเขียวอยู่บนเส้นนี้ เรียกว่า Principal Component 1 (PC1)
- Slope ของเส้นนี้ (สมมติว่าเป็น 0.25) จะบอกเราว่าข้อมูลไหนเป็นส่วนประกอบของ PC1 เท่าไหร่ เหมือนการผสมค็อกเทล
  - การที่ slope เป็น 0.25 ก็หมายความว่า PC1 ประกอบด้วย รหัสพันธุกรรม 1 จำนวน 4 ส่วน และรหัสพันธุกรรม 2 จำนวน 1 ส่วน
  - สัดส่วนดังกล่าวนี้ (0.25 หรือ 1:4) เรียกว่า Linear Combination
  - **<u>ตัวอย่างโจทย์</u>** “PC1 เป็น Linear Combination ของ A1, A4” หมายความว่าอย่างไร? นักศึกษาตอบได้มั้ย
  - สัดส่วนนี้ยังสามารถบอกเราได้ว่า รหัสพันธุกรรม 1 มีความสำคัญต่อการระบุตัวตนของหนูได้มากกว่ารหัสพันธุกรรม 2 (ไม่ได้บอกว่าจะทำนายอะไรได้ดีหรือไม่ แต่หนูแต่ละตัวมีรหัสพันธุกรรมนี้แตกต่างกัน)



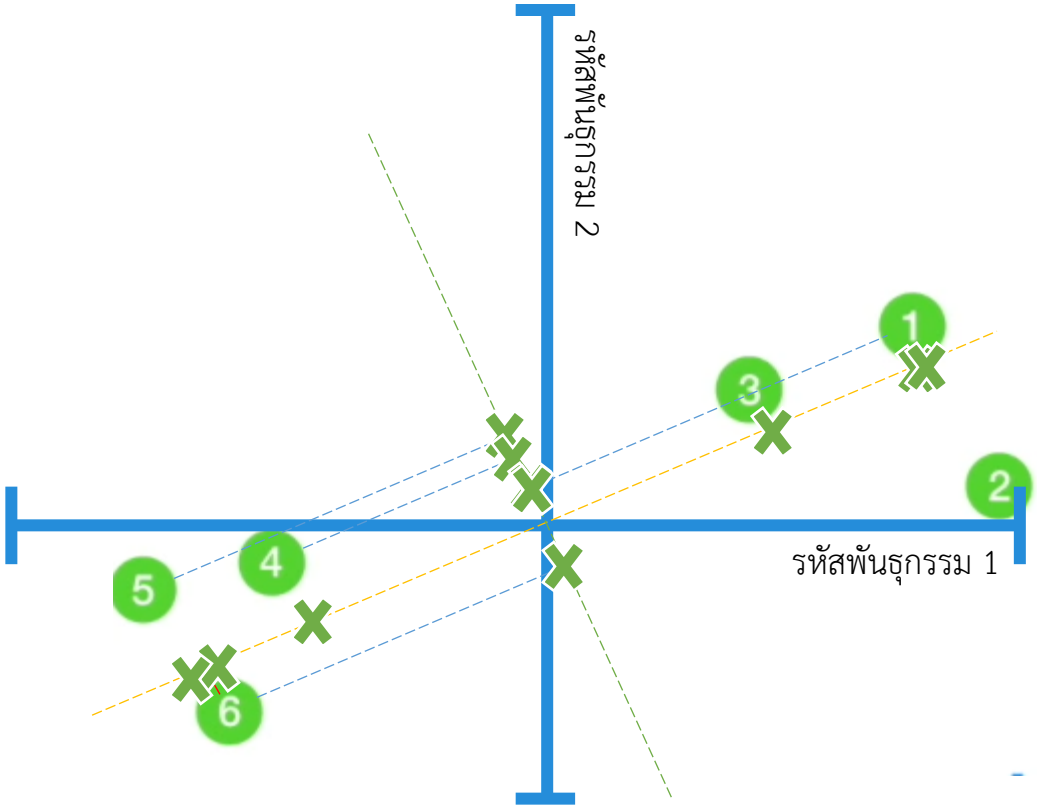
ขั้นตอนวิธี PCA

1. หาค่าเฉลี่ยของข้อมูลในแนวแกน x และแนวแกน y
2. หาจุดตัดของค่าเฉลี่ยทั้งสองแกนนั้น
3. เลื่อนข้อมูลทั้งหมดพร้อมๆกันไปในทิศทางที่ทำให้จุดตัด (กากบาทสีแดง) ไปอยู่ที่จุด (0, 0)
4. วาดเส้นตรงผ่านจุด (0, 0) โดยสุ่ม Slope และค่อยๆ หา slope ที่จะฟิตข้อมูลทั้งหมดได้มากที่สุด
5. ตีเส้นฉากและ Project จุดต่างๆ ลงบนเส้นตรงที่หาได้จากข้อ 4
6. เมื่อได้ PC1 แล้ว ทำการหา PC2 โดยการวางเส้นให้ตั้งฉากกับ PC1



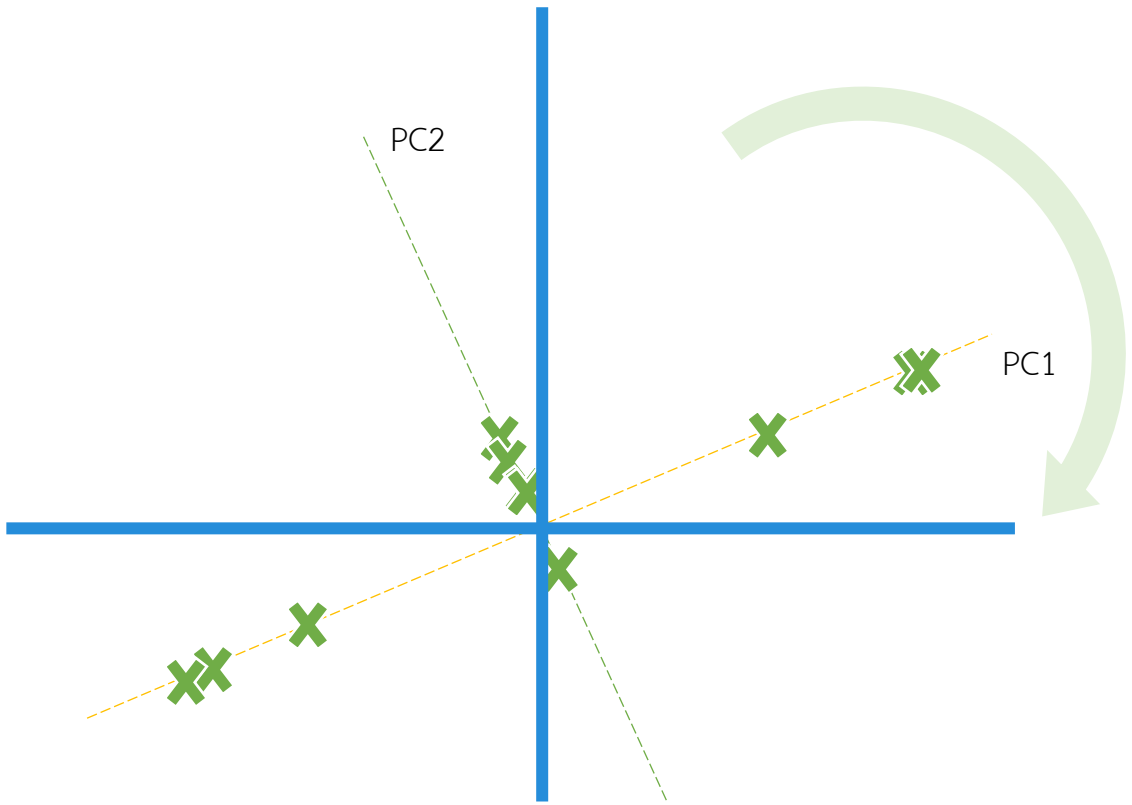
ขั้นตอนวิธี PCA

1. หาค่าเฉลี่ยของข้อมูลในแนวแกน x และแนวแกน y
2. หาจุดตัดของค่าเฉลี่ยทั้งสองแกนนั้น
3. เลื่อนข้อมูลทั้งหมดพร้อมๆกันไปในทิศทางที่ทำให้จุดตัด (กากบาทสีแดง) ไปอยู่ที่จุด (0, 0)
4. วาดเส้นตรงผ่านจุด (0, 0) โดยสุ่ม Slope และค่อยๆ หา slope ที่จะฟิตข้อมูลทั้งหมดได้มากที่สุด
5. ตีเส้นฉากและ Project จุดต่างๆ ลงบนเส้นตรงที่หาได้จากข้อ 4
6. เมื่อได้ PC1 แล้ว ทำการหา PC2 โดยการวางเส้นให้ตั้งฉากกับ PC1
7. หาเส้นฉากและ Project จุดต่างๆ ลงบนเส้น PC2



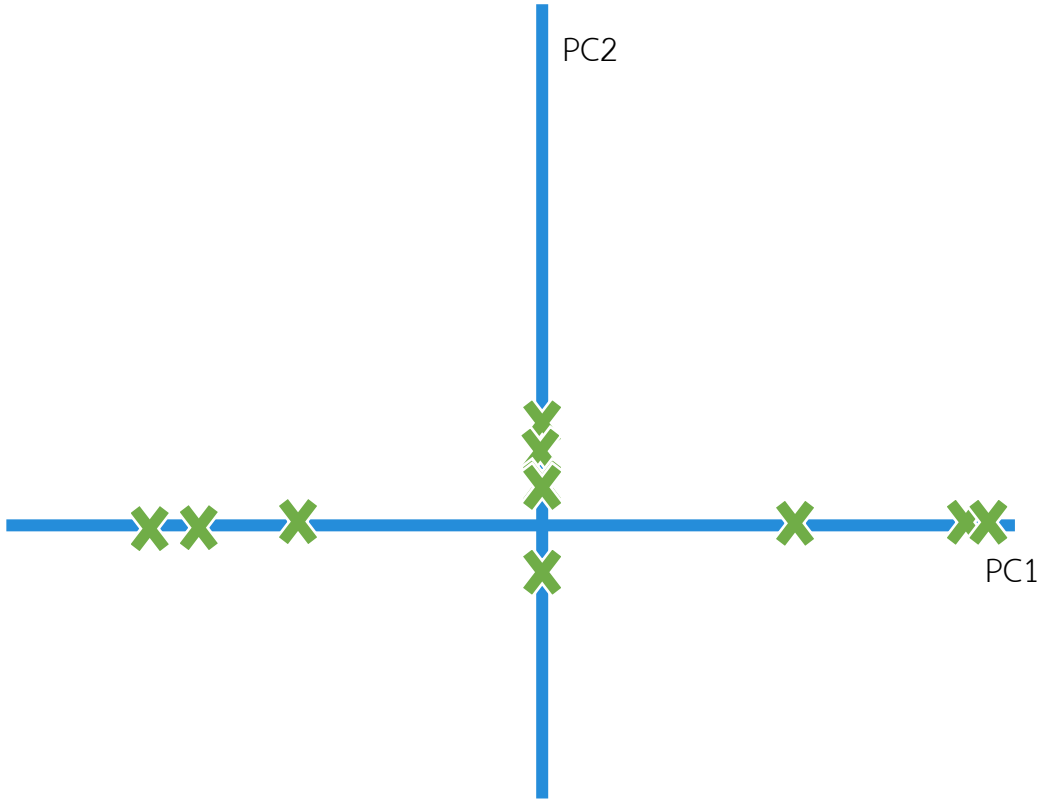
ขั้นตอนวิธี PCA

1. หาค่าเฉลี่ยของข้อมูลในแนวแกน x และแนวแกน y
2. หาจุดตัดของค่าเฉลี่ยทั้งสองแกนนั้น
3. เลื่อนข้อมูลทั้งหมดพร้อมๆกันไปในทิศทางที่ทำให้จุดตัด (กากบาทสีแดง) ไปอยู่ที่จุด (0, 0)
4. วาดเส้นตรงผ่านจุด (0, 0) โดยสุ่ม Slope และค่อยๆ หา slope ที่จะฟิตข้อมูลทั้งหมดได้มากที่สุด
5. ตีเส้นฉากและ Project จุดต่างๆ ลงบนเส้นตรงที่หาได้จากข้อ 4
6. เมื่อได้ PC1 แล้ว ทำการหา PC2 โดยการวางเส้นให้ตั้งฉากกับ PC1
7. หาเส้นฉากและ Project จุดต่างๆ ลงบนเส้น PC2
8. หมุนแกนทั้ง PC1 และ PC2 ให้ตรง x, y



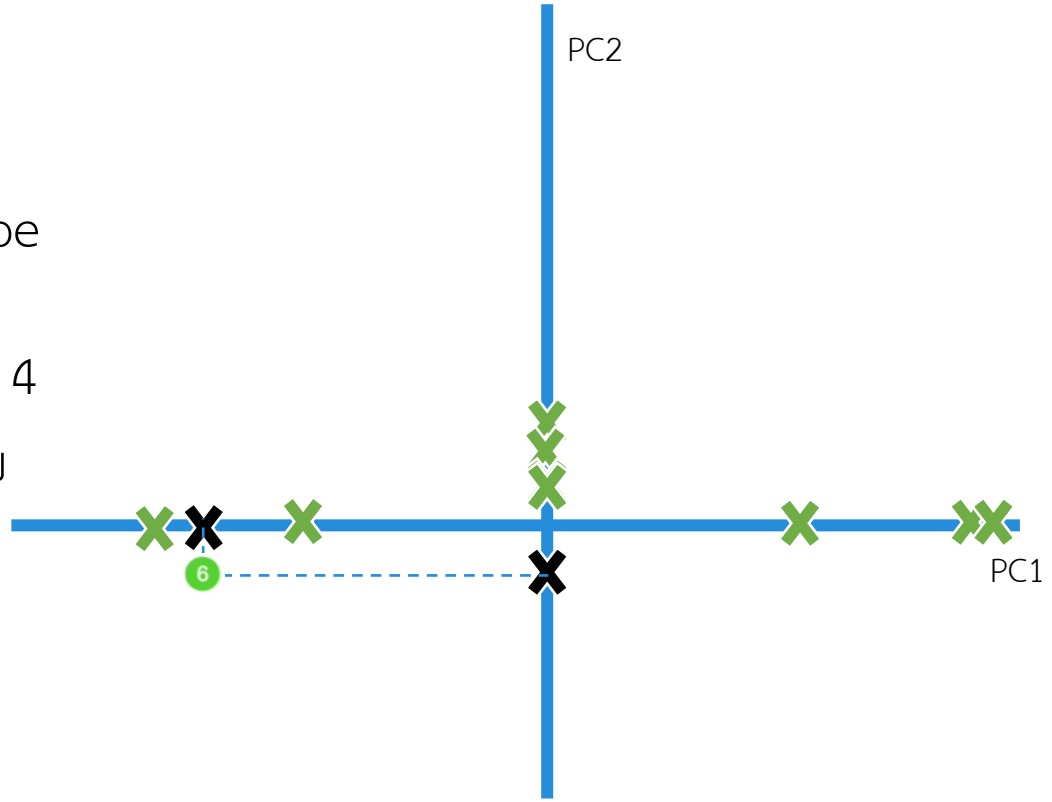
ขั้นตอนวิธี PCA

1. หาค่าเฉลี่ยของข้อมูลในแนวแกน x และแนวแกน y
2. หาจุดตัดของค่าเฉลี่ยทั้งสองแกนนั้น
3. เลื่อนข้อมูลทั้งหมดพร้อมๆกันไปในทิศทางที่ทำให้จุดตัด (กากบาทสีแดง) ไปอยู่ที่จุด (0, 0)
4. วาดเส้นตรงผ่านจุด (0, 0) โดยสุ่ม Slope และค่อยๆ หา slope ที่จะฟิตข้อมูลทั้งหมดได้มากที่สุด
5. ตีเส้นฉากและ Project จุดต่างๆ ลงบนเส้นตรงที่หาได้จากข้อ 4
6. เมื่อได้ PC1 แล้ว ทำการหา PC2 โดยการวางเส้นให้ตั้งฉากกับ PC1
7. หาเส้นฉากและ Project จุดต่างๆ ลงบนเส้น PC2
8. หมุนแกนทั้ง PC1 และ PC2 ให้ตรง x, y



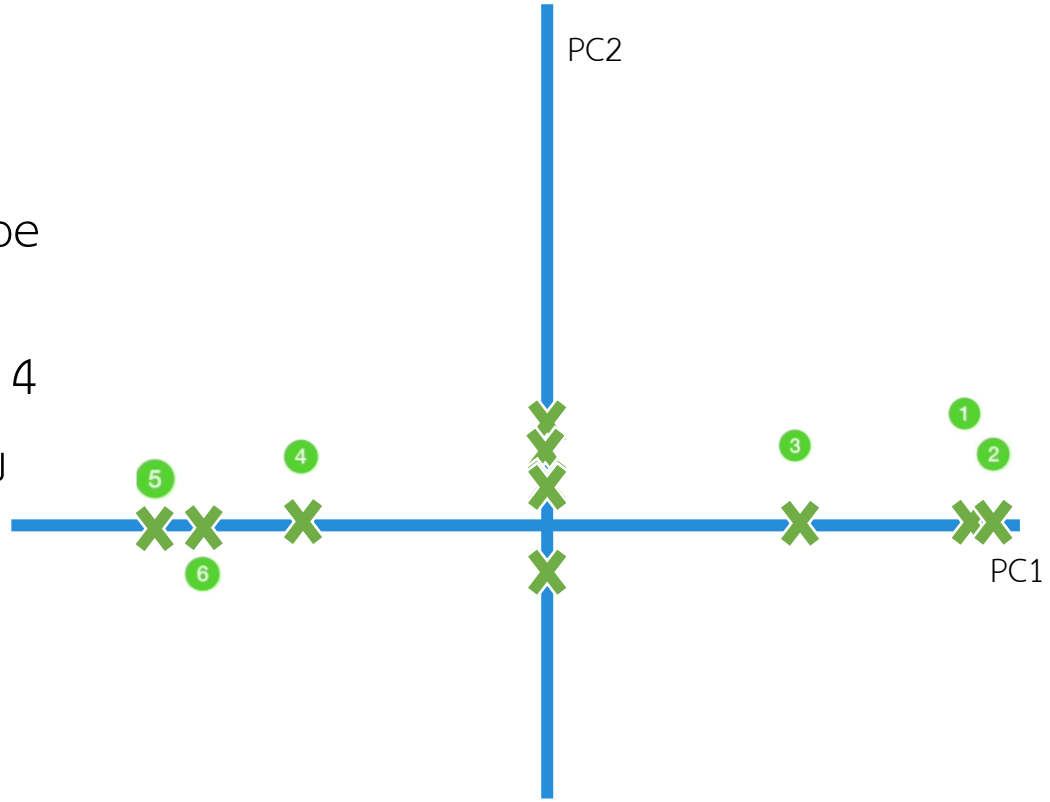
ขั้นตอนวิธี PCA

1. หาค่าเฉลี่ยของข้อมูลในแนวแกน x และแนวแกน y
2. หาจุดตัดของค่าเฉลี่ยทั้งสองแกนนั้น
3. เลื่อนข้อมูลทั้งหมดพร้อมๆกันไปในทิศทางที่ทำให้จุดตัด (กากบาทสีแดง) ไปอยู่ที่จุด (0, 0)
4. วาดเส้นตรงผ่านจุด (0, 0) โดยสุ่ม Slope และค่อยๆ หา slope ที่จะฟิตข้อมูลทั้งหมดได้มากที่สุด
5. ตีเส้นฉากและ Project จุดต่างๆ ลงบนเส้นตรงที่หาได้จากข้อ 4
6. เมื่อได้ PC1 แล้ว ทำการหา PC2 โดยการวางเส้นให้ตั้งฉากกับ PC1
7. หาเส้นฉากและ Project จุดต่างๆ ลงบนเส้น PC2
8. หมุนแกนทั้ง PC1 และ PC2 ให้ตรง x, y
9. Plot ตามค่าคู่อันดับของ Sample แต่ละตัว



ขั้นตอนวิธี PCA

1. หาค่าเฉลี่ยของข้อมูลในแนวแกน x และแนวแกน y
2. หาจุดตัดของค่าเฉลี่ยทั้งสองแกนนั้น
3. เลื่อนข้อมูลทั้งหมดพร้อมๆกันไปในทิศทางที่ทำให้จุดตัด (กากบาทสีแดง) ไปอยู่ที่จุด (0, 0)
4. วาดเส้นตรงผ่านจุด (0, 0) โดยสุ่ม Slope และค่อยๆ หา slope ที่จะฟิตข้อมูลทั้งหมดได้มากที่สุด
5. ตีเส้นฉากและ Project จุดต่างๆ ลงบนเส้นตรงที่หาได้จากข้อ 4
6. เมื่อได้ PC1 แล้ว ทำการหา PC2 โดยการวางเส้นให้ตั้งฉากกับ PC1
7. หาเส้นฉากและ Project จุดต่างๆ ลงบนเส้น PC2
8. หมุนแกนทั้ง PC1 และ PC2 ให้ตรง x, y
9. Plot ตามค่าคู่อันดับของ Sample แต่ละตัว



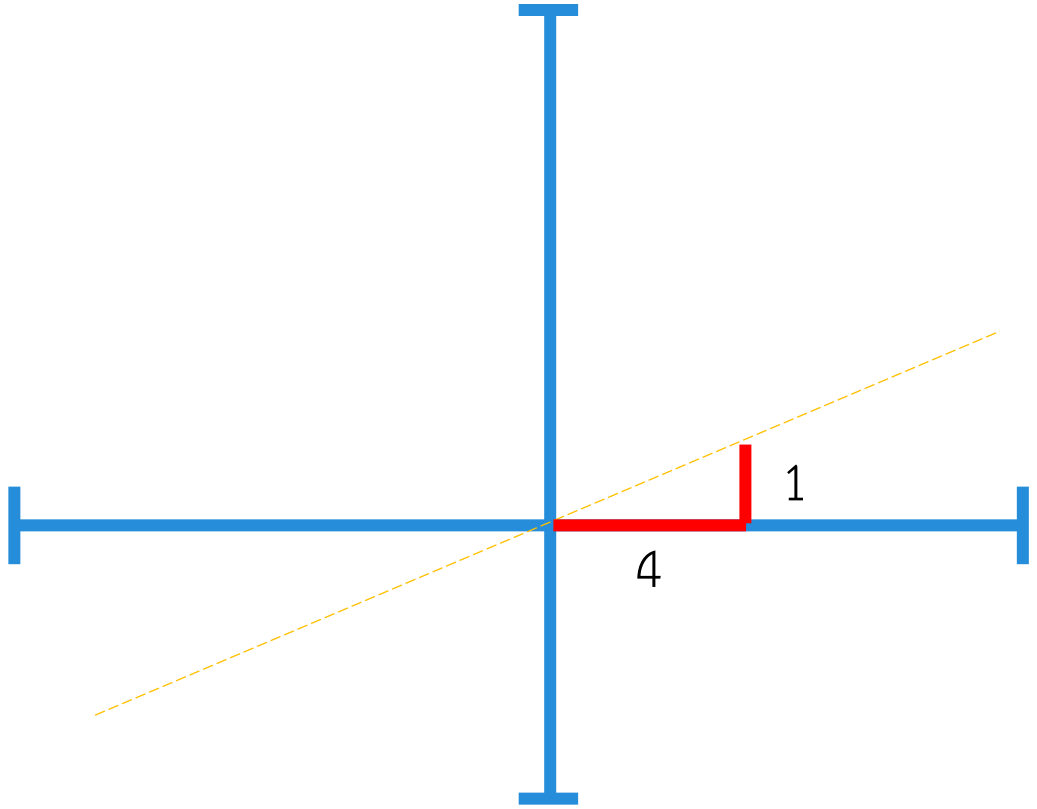
Outline

- Part II introduction
- Introduction to Clustering
- **Dimensionality Reduction**
  - **PCA (Principal Component Analysis)**
    - แนวคิดของ PCA
    - กระบวนการ
    - **รายละเอียดของผลลัพธ์**
    - เมื่อข้อมูลมีมิติมากขึ้น
  - เทคนิค Dimensionality Reduction อื่น ๆ ที่น่าสนใจ

รายละเอียดของผลลัพธ์

Eigen Value

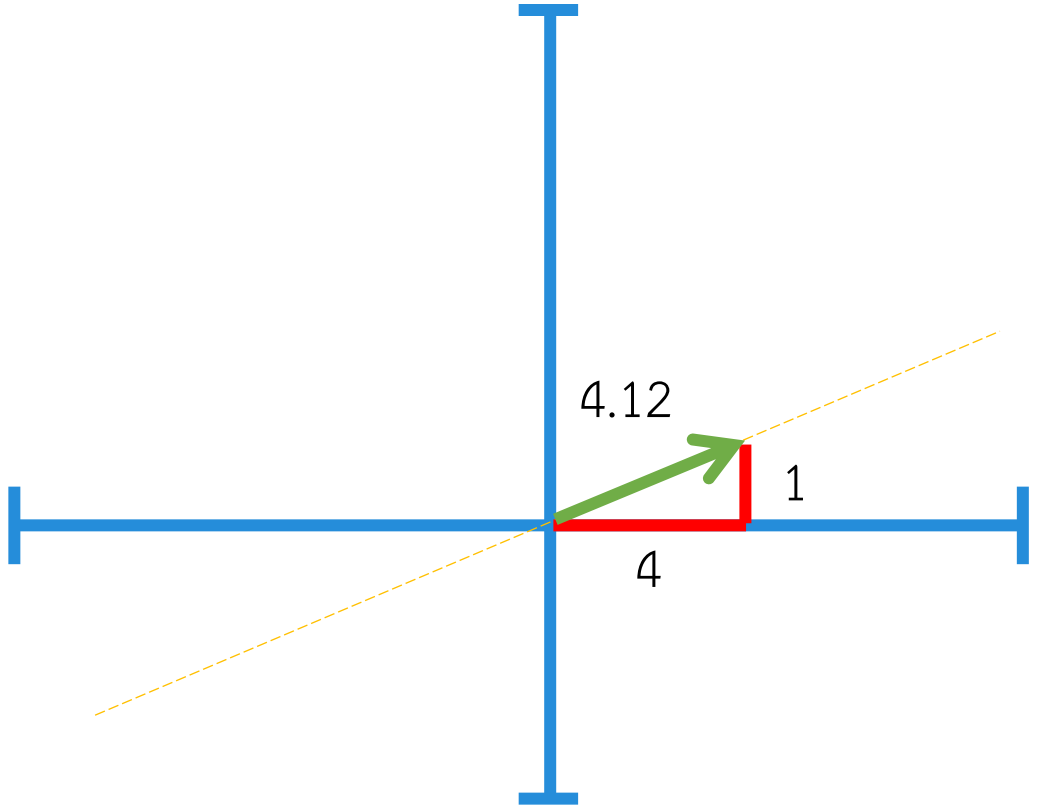
- ย้อนกลับไปตอนที่เรายังไม่ได้หมุนกราฟ และพิจารณาเฉพาะ PC1
- ในขั้นนี้เราระบุ Slope เป็น 0.25 ดังนั้นระยะของเส้นสีแดงควรเป็นดังภาพ



รายละเอียดของผลลัพธ์

Eigenvector

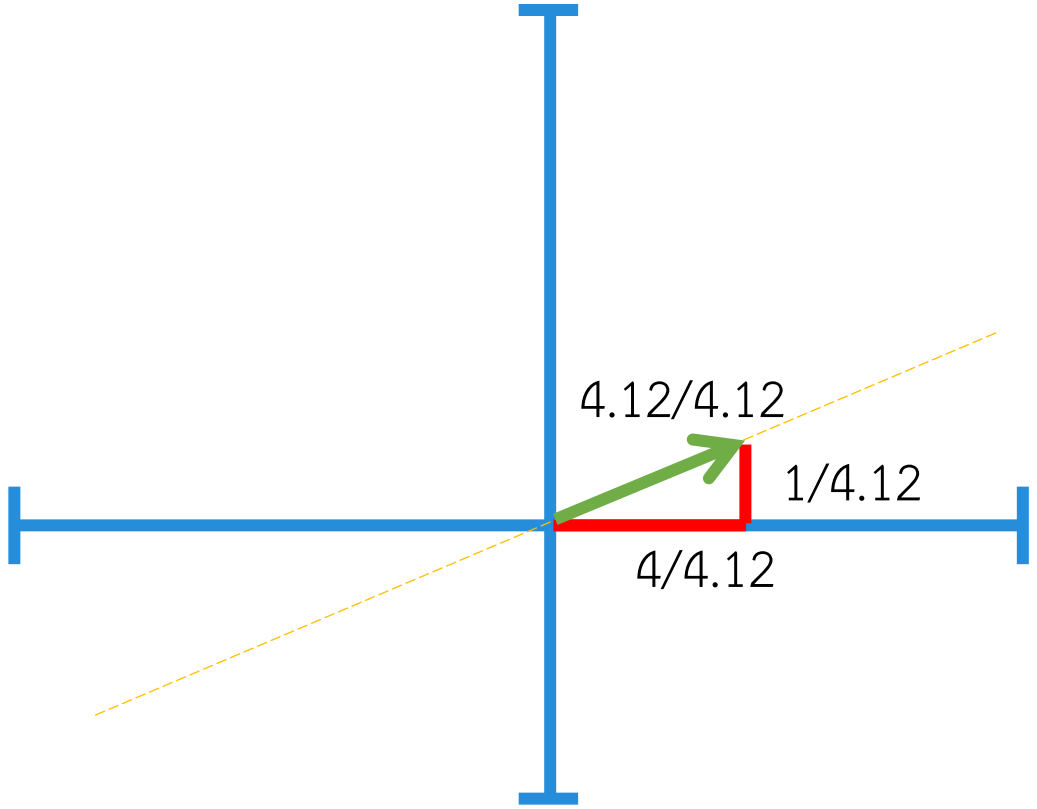
- ย้อนกลับไปตอนที่เรายังไม่ได้หมุนกราฟ และพิจารณาเฉพาะ PC1
- ในขั้นนี้เราระบุ Slope เป็น 0.25 ดังนั้นระยะของเส้นสีแดงควรเป็นดังภาพ
- เมื่อคำนวณด้วยทฤษฎีบทพีธากอรัส เราจะได้เวคเตอร์เส้นสีเขียว มีขนาด 4.12



รายละเอียดของผลลัพธ์

Eigenvector

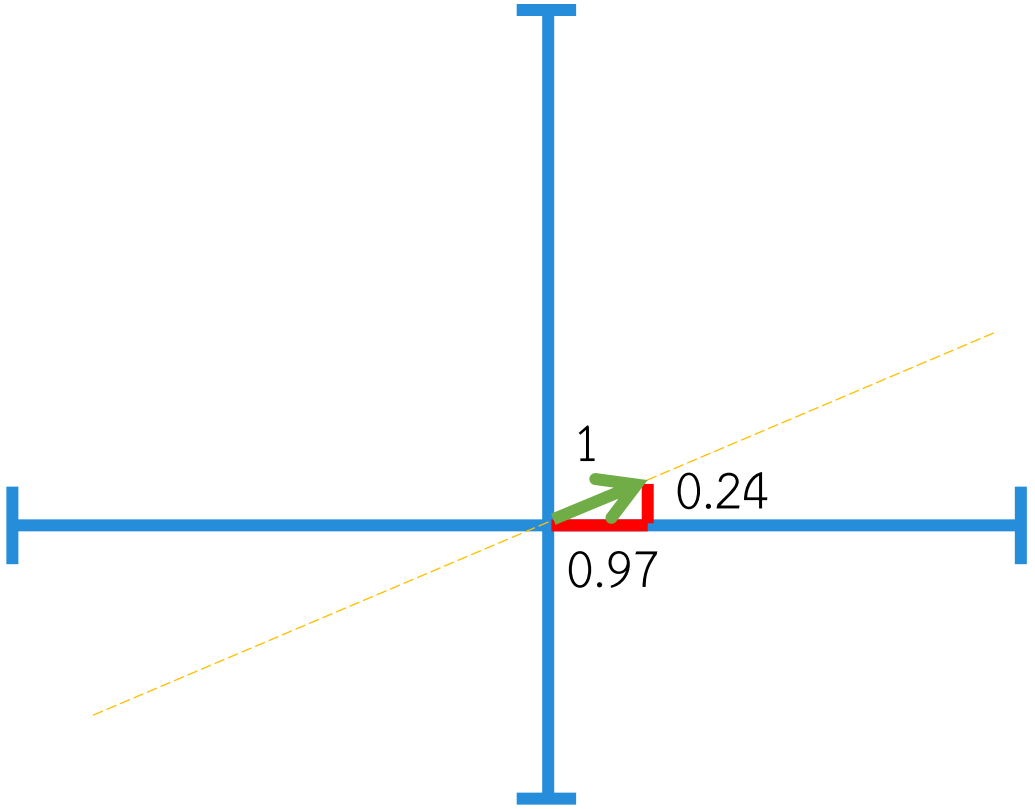
- ย้อนกลับไปตอนที่เรายังไม่ได้หมุนกราฟ และพิจารณาเฉพาะ PC1
- ในขั้นนี้เราระบุ Slope เป็น 0.25 ดังนั้นระยะของเส้นสีแดงควรเป็นดังภาพ
- เมื่อคำนวณด้วยทฤษฎีบทพีธากอรัส เราจะได้เวคเตอร์เส้นสีเขียว มีขนาด 4.12
- แต่เวลาเราพูดในเชิงวิชาการ เราจะพูดถึงเวคเตอร์นี้ในขนาด 1 หน่วย เราจึงต้องปรับสเกลทุกค่าลง โดยหาร 4.12 ทั้ง 3 ค่า



รายละเอียดของผลลัพธ์

Eigenvector

- จากผลดังภาพ สามารถอธิบายใหม่ได้ว่า
  - การสร้าง PC1 เกิดจากการผสมกันของรหัสพันธุกรรม 1 จำนวน 0.97 ส่วน และรหัสพันธุกรรม 2 จำนวน 0.24 ส่วน
  - เราเรียกสัดส่วนของรหัสพันธุกรรมจาก Eigenvector นี้ว่า “Loading Score”

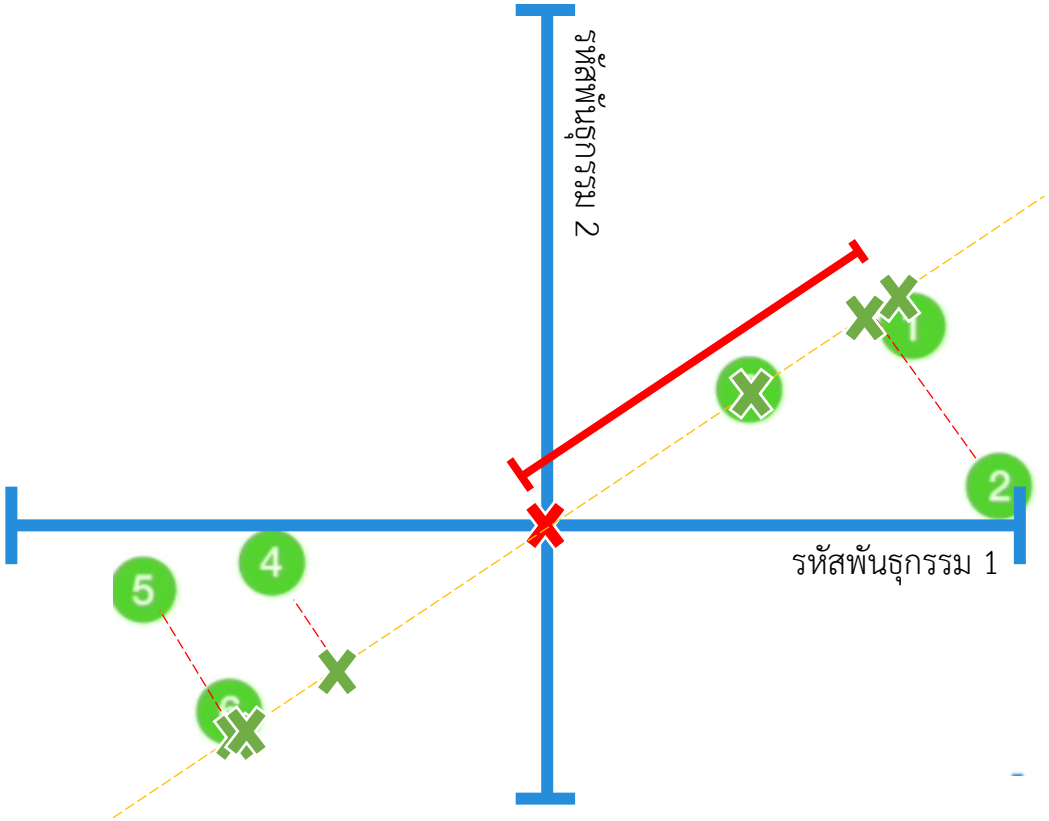


รายละเอียดของผลลัพธ์

Variation

- ยังจำ SS(distances) ได้มั้ย (หน้า 19)
- การหา Variation ของแต่ละแกน

$$Variation = \frac{SS(distances)}{n-1}$$



รายละเอียดของผลลัพธ์

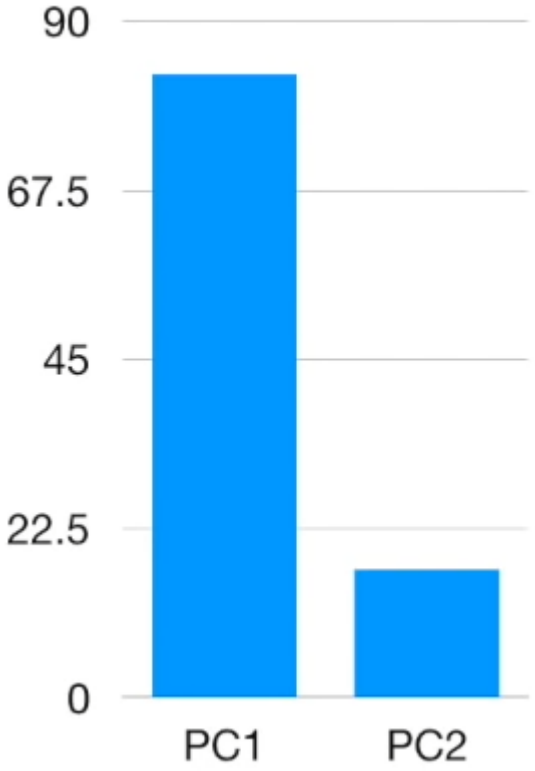
Percent Variation

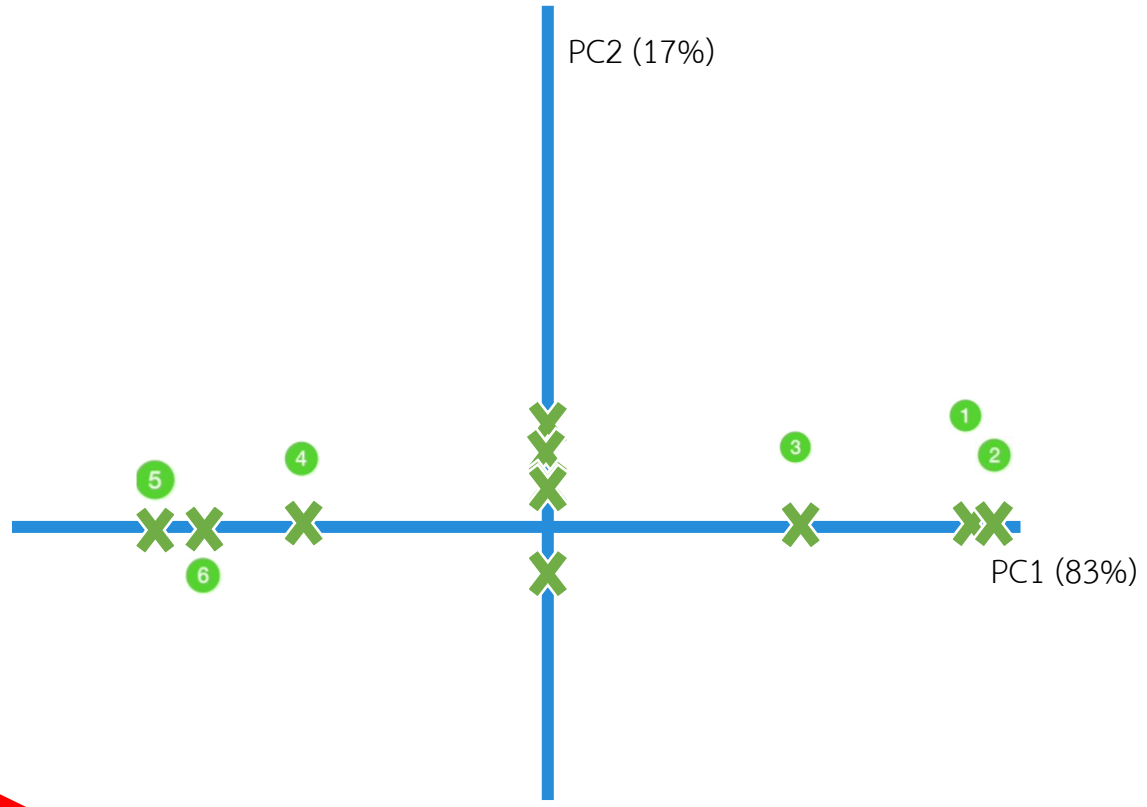
- เมื่อหา Variation ของแต่ละแกนได้แล้ว ถ้าต้องการหา Percent Variation เราก็นำค่า Variation ของแกนนั้น หารด้วย Variation ทั้งหมดที่มีและคูณ 100 ดังสมการ

$$\%Variation = \frac{Variation_{observe}}{Variation_{Total}} \times 100$$

- สมมติ แกน PC1 คำนวณได้ค่า variation 15 และ แกน PC2 คำนวณได้ค่า variation 3 ดังนั้น
  - Percent variation ของ PC1 คือ (15/18)x100 = 83%
  - Percent variation ของ PC2 คือ (3/18)x100 = 17%

รายละเอียดของผลลัพธ์





ชื่อเฉพาะของกราฟแท่งที่แสดง %variation คือ **“Scree Plot”**

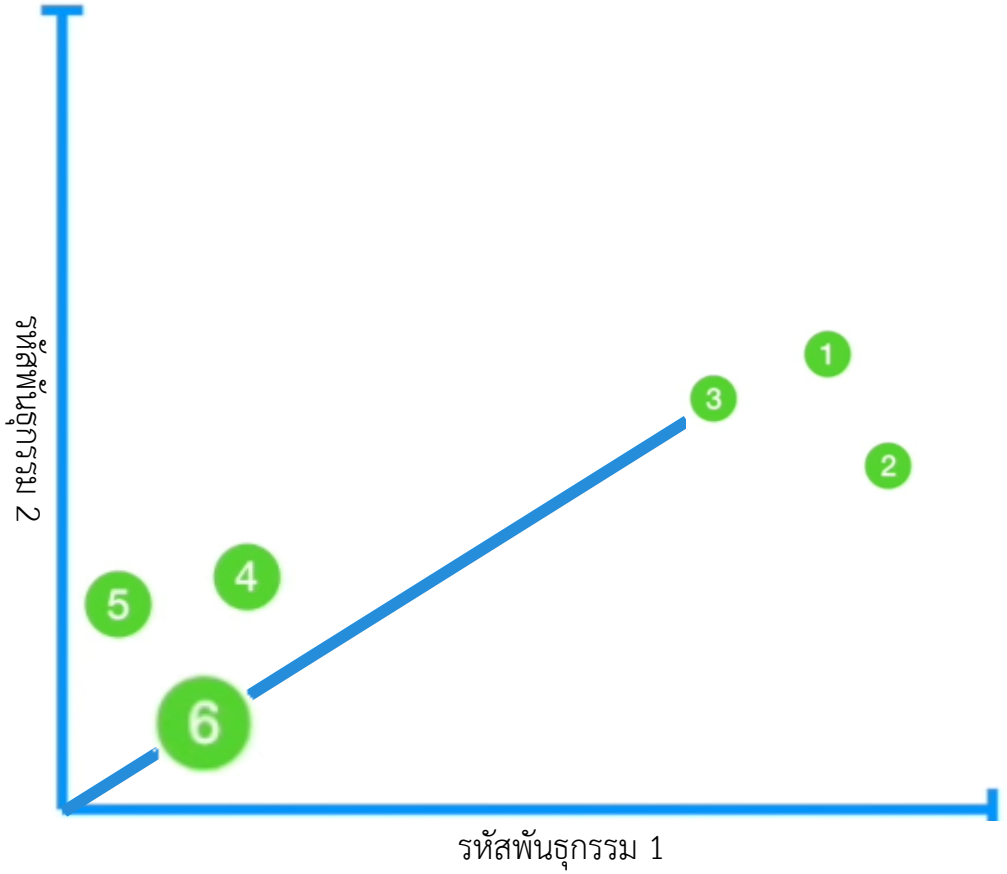
Outline

- Part II introduction
- Introduction to Clustering
- **Dimensionality Reduction**
  - **PCA (Principal Component Analysis)**
    - แนวคิดของ PCA
    - กระบวนการ
    - รายละเอียดของผลลัพธ์
    - **เมื่อข้อมูลมีมิติมากขึ้น**
  - เทคนิค Dimensionality Reduction อื่น ๆ ที่น่าสนใจ

เมื่อข้อมูลมีมิติมากขึ้น

- วิธีการก็เหมือนเดิมแทบทั้งหมด
- เพียงแต่กราฟจะมากกว่า 2 มิติ
- ขอยกตัวอย่างเป็น 3 มิติก่อน

	หนู 1	หนู 2	หนู 3	หนู 4	หนู 5	หนู 6	...
รหัสพันธุกรรม 1	10	11	8	3	2	1	...
รหัสพันธุกรรม 2	6	4	5	3	2.8	1	...
รหัสพันธุกรรม 3	12	9	10	2.5	1.3	2	...



เมื่อข้อมูลมีมิติมากขึ้น

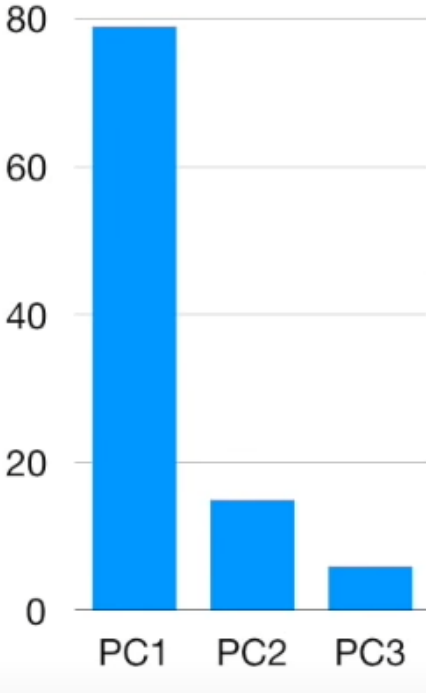
- วิธีการก็เหมือนเดิมแทบทั้งหมด
  - ดึงข้อมูลมาไว้ตรงกลางของกราฟ
  - หา PC1 โดยวาดเส้นตรงให้ฟิตกับข้อมูลที่สุด (แกนไหนองศาไหนก็ได้)
  - หา PC2 โดยให้เส้นตั้งฉากกับ PC1 แต่เนื่องจากเป็นกราฟ 3 มิติ เราจะต้องหาเส้นที่ฟิตกับข้อมูลมากที่สุดเฉพาะในแกนที่ตั้งฉากกับ PC1
  - หา PC3 โดยวาดให้ตั้งฉากกับทั้ง PC1 และ PC2
  - Projection ข้อมูลลงสู่แกนทั้ง 3
  - เอียงแกนทั้ง 3 ให้อยู่ในระนาบ x, y, z และพล็อตกราฟตาม coordinate ใหม่
- ถ้ามีมิติมากกว่านี้จะวาดกราฟไม่ได้ แต่ยังคงดำเนินการตามระบบคณิตศาสตร์ได้

เมื่อข้อมูลมีมิติมากขึ้น

- วิธีการก็เหมือนเดิมแทบทั้งหมด
  - ดึงข้อมูลมาไว้ตรงกลางของกราฟ
  - หา PC1 โดยวาดเส้นตรงให้ฟิตกับข้อมูลที่สุด (แกนไหนองศาไหนก็ได้)
  - หา PC2 โดยให้เส้นตั้งฉากกับ PC1 แต่เนื่องจากเป็นกราฟ 3 มิติ เราจะต้องหาเส้นที่ฟิตกับข้อมูลมากที่สุดเฉพาะในแกนที่ตั้งฉากกับ PC1
  - หา PC3 โดยวาดให้ตั้งฉากกับทั้ง PC1 และ PC2
  - Projection ข้อมูลลงสู่แกนทั้ง 3
  - เอียงแกนทั้ง 3 ให้อยู่ในระนาบ x, y, z และพล็อตกราฟตาม coordinate ใหม่
- ถ้ามีมิติมากกว่านี้จะวาดกราฟไม่ได้ แต่ยังคงดำเนินการตามระบบคณิตศาสตร์ได้

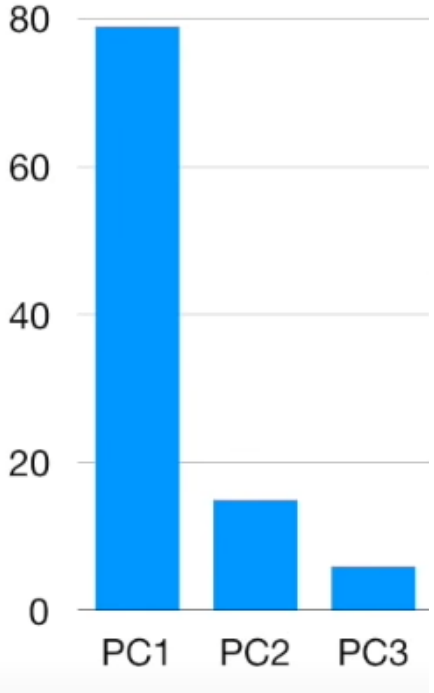
เมื่อข้อมูลมีมิติมากขึ้น

- ถ้าอยากลดมิติ ทำอย่างไร?
- การลดมิติ จะต้องอ้างถึง percent variation หรือดู Scree plot
- จากภาพ PC1 มีความสำคัญสูงสุด ตามมาด้วย PC2 และ PC3
  - ตามภาพคือ เพียง PC1 และ PC2 ก็สามารถอธิบาย variation ของข้อมูลได้มากกว่า 90%
- สมมติว่าต้องการสร้างกราฟ 2 มิติ ของข้อมูลนี้



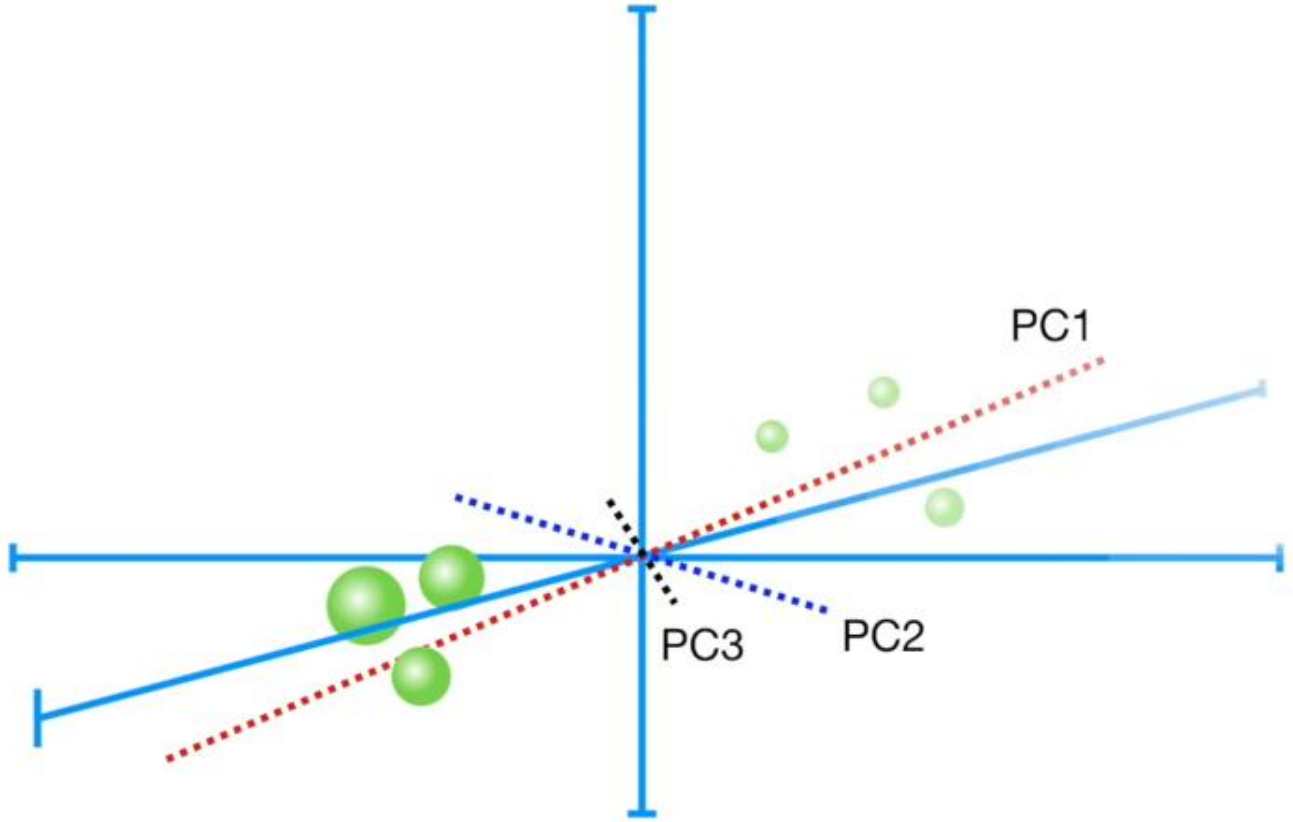
เมื่อข้อมูลมีมิติมากขึ้น

- ถ้าอยากลดมิติ ทำอย่างไร?
- การลดมิติ จะต้องอ้างถึง percent variation หรือดู Scree plot
- จากภาพ PC1 มีความสำคัญสูงสุด ตามมาด้วย PC2 และ PC3
  - ตามภาพคือ เพียง PC1 และ PC2 ก็สามารถอธิบาย variation ของข้อมูลได้มากกว่า 90%
- สมมติว่าต้องการสร้างกราฟ 2 มิติ ของข้อมูลนี้



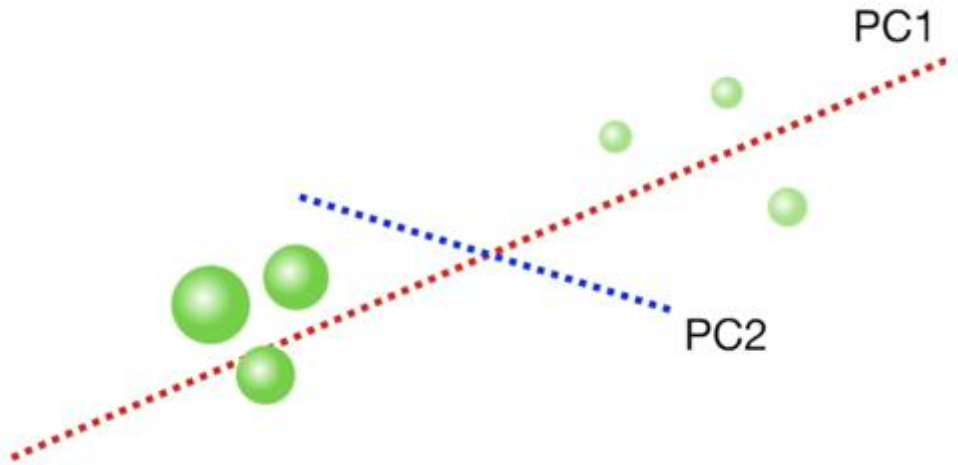
เมื่อข้อมูลมีมิติมากขึ้น

1. ตั้งต้นที่กราฟ 3 มิติ



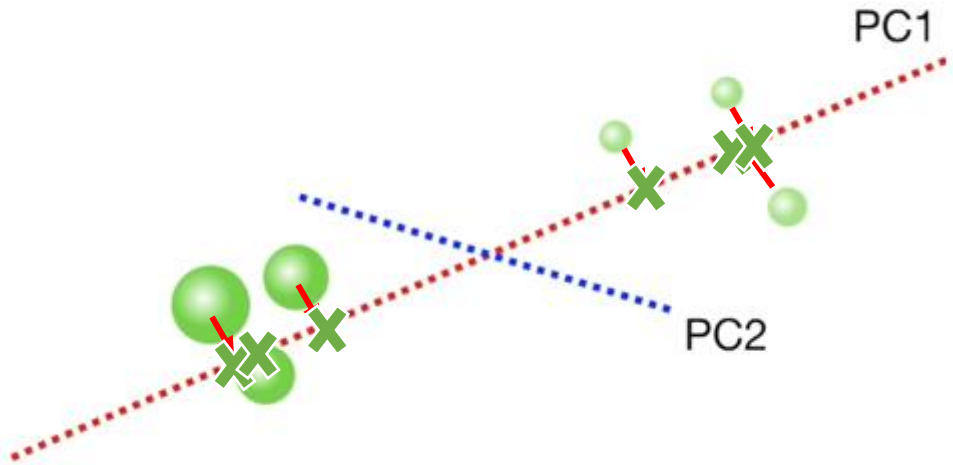
เมื่อข้อมูลมีมิติมากขึ้น

1. ตั้งต้นที่กราฟ 3 มิติ
2. ลบแกนที่ไม่ต้องการออก ในที่นี้คือ PC3



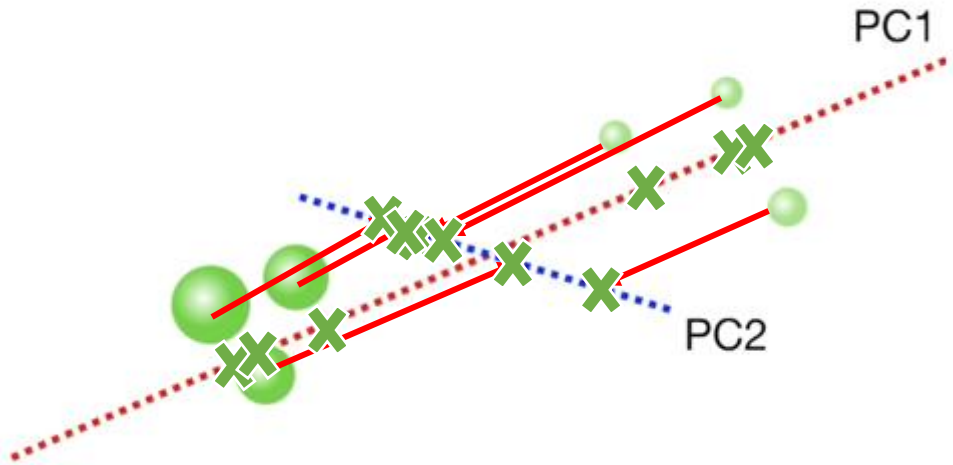
เมื่อข้อมูลมีมิติมากขึ้น

1. ตั้งต้นที่กราฟ 3 มิติ
2. ลบแกนที่ไม่ต้องการออก ในที่นี้คือ PC3
3. Project ข้อมูลลงบนแกน PC1



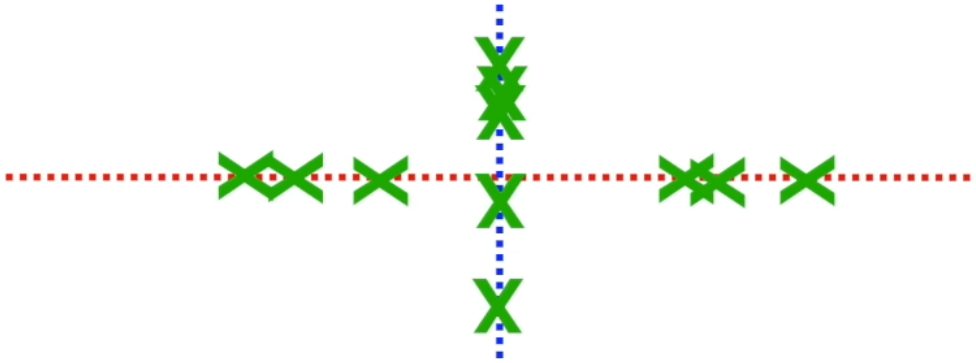
เมื่อข้อมูลมีมิติมากขึ้น

1. ตั้งต้นที่กราฟ 3 มิติ
2. ลบแกนที่ไม่ต้องการออก ในที่นี้คือ PC3
3. Project ข้อมูลลงบนแกน PC1
4. Project ข้อมูลลงบนแกน PC2



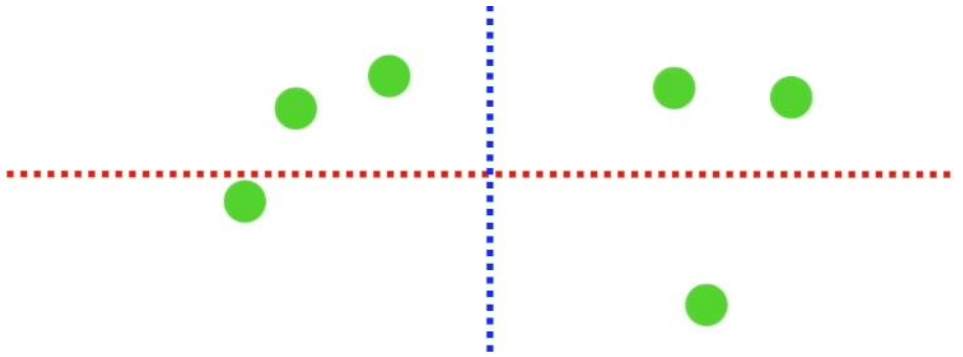
เมื่อข้อมูลมีมิติมากขึ้น

1. ตั้งต้นที่กราฟ 3 มิติ
2. ลบแกนที่ไม่ต้องการออก ในที่นี้คือ PC3
3. Project ข้อมูลลงบนแกน PC1
4. Project ข้อมูลลงบนแกน PC2
5. หมุนแกนขึ้นมาให้ตรง



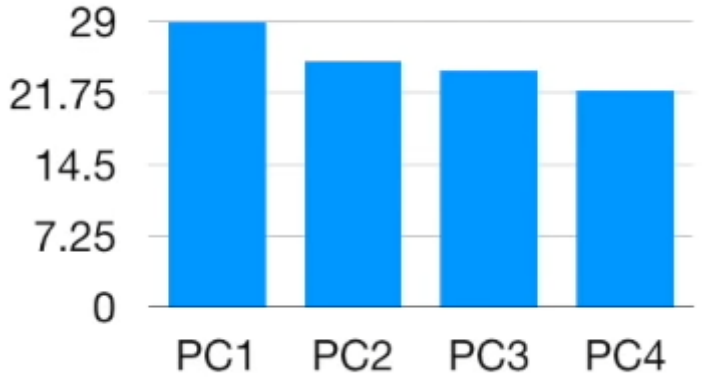
เมื่อข้อมูลมีมิติมากขึ้น

1. ตั้งต้นที่กราฟ 3 มิติ
2. ลบแกนที่ไม่ต้องการออก ในที่นี้คือ PC3
3. Project ข้อมูลลงบนแกน PC1
4. Project ข้อมูลลงบนแกน PC2
5. หมุนแกนขึ้นมาให้ตรง
6. ลงจุดตาม coordinate



เมื่อข้อมูลมีมิติมากขึ้น

- บ่อยครั้ง เราจะเจอ Scree plot แบบนี้



- หมายความว่า แค่ PC1 หรือ PC2 ก็ยังไม่สามารถระบุความแน่ชัดของตัวอย่างได้
- แต่แม้จะมีความไม่แน่ชัดดังตัวอย่าง แต่ก็ยังสามารถลดมิติลง และนำมาใช้เพื่อวิเคราะห์หากลุ่มก้อนหรือคลัสเตอร์ ได้

ตัวอย่างโค้ด

class `sklearn.decomposition.`**PCA**(*n_components=None, *, copy=True, whiten=False, svd_solver='auto', tol=0.0, iterated_power='auto', n_oversamples=10, power_iteration_normalizer='auto', random_state=None*)

Parameters::

n_components : *int, float or 'mle', default=None*

Number of components to keep. if n_components is not set all components are kept:

```
n_components == min(n_samples, n_features)
```

If `n_components == 'mle'` and `svd_solver == 'full'`, Minka's MLE is used to guess the dimension. Use of `n_components == 'mle'` will interpret `svd_solver == 'auto'` as `svd_solver == 'full'`.

If `0 < n_components < 1` and `svd_solver == 'full'`, select the number of components such that the amount of variance that needs to be explained is greater than the percentage specified by n_components.

If `svd_solver == 'arpack'`, the number of components must be strictly less than the minimum of n_features and n_samples.

Hence, the None case results in:

```
n_components == min(n_samples, n_features) - 1
```

ตัวอย่างโค้ด

class `sklearn.decomposition.`**PCA**(*n_components=None, *, copy=True, whiten=False, svd_solver='auto', tol=0.0, iterated_power='auto', n_oversamples=10, power_iteration_normalizer='auto', random_state=None*)

Attributes::

components_ : *ndarray of shape (n_components, n_features)*
Principal axes in feature space, representing the directions of maximum variance in the data. Equivalently, the right singular vectors of the centered input data, parallel to its eigenvectors. The components are sorted by `explained_variance_`.

explained_variance_ : *ndarray of shape (n_components,)*
The amount of variance explained by each of the selected components. The variance estimation uses `n_samples - 1` degrees of freedom.

Equal to n_components largest eigenvalues of the covariance matrix of X.

New in version 0.18.

explained_variance_ratio_ : *ndarray of shape (n_components,)*
Percentage of variance explained by each of the selected components.

If `n_components` is not set then all components are stored and the sum of the ratios is equal to 1.0.

singular_values_ : *ndarray of shape (n_components,)*
The singular values corresponding to each of the selected components. The singular values are equal to the 2-norms of the `n_components` variables in the lower-dimensional space.

New in version 0.19.

ตัวอย่างโค้ด

class `sklearn.decomposition.`**PCA**(*n_components=None, *, copy=True, whiten=False, svd_solver='auto', tol=0.0, iterated_power='auto', n_oversamples=10, power_iteration_normalizer='auto', random_state=None*)

n_components_ : *int*

The estimated number of components. When n_components is set to 'mle' or a number between 0 and 1 (with svd_solver == 'full') this number is estimated from input data. Otherwise it equals the parameter n_components, or the lesser value of n_features and n_samples if n_components is None.

n_features_ : *int*

Number of features in the training data.

n_samples_ : *int*

Number of samples in the training data.

ตัวอย่างโค้ด

```
>>> import numpy as np
>>> from sklearn.decomposition import PCA
>>> X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
>>> pca = PCA(n_components=2)
>>> pca.fit(X)
PCA(n_components=2)
>>> print(pca.explained_variance_ratio_)
[0.9924... 0.0075...]
>>> print(pca.singular_values_)
[6.30061... 0.54980...]
```

```
>>> pca = PCA(n_components=2, svd_solver='full')
>>> pca.fit(X)
PCA(n_components=2, svd_solver='full')
>>> print(pca.explained_variance_ratio_)
[0.9924... 0.00755...]
>>> print(pca.singular_values_)
[6.30061... 0.54980...]
```

```
>>> pca = PCA(n_components=1, svd_solver='arpack')
>>> pca.fit(X)
PCA(n_components=1, svd_solver='arpack')
>>> print(pca.explained_variance_ratio_)
[0.99244...]
>>> print(pca.singular_values_)
[6.30061...]
```

ตัวอย่างโค้ด

`fit_transform`(*X, y=None*) [source]

Fit the model with X and apply the dimensionality reduction on X.

Parameters::	**X : *array-like of shape (n_samples, n_features)*** Training data, where `n_samples` is the number of samples and `n_features` is the number of features. **y : *Ignored*** Ignored.
Returns::	**X_new : *ndarray of shape (n_samples, n_components)*** Transformed values.

`get_feature_names_out`(*input_features=None*) [source]

Get output feature names for transformation.

Parameters::	**input_features : *array-like of str or None, default=None*** Only used to validate feature names with the names seen in `fit`.
Returns::	**feature_names_out : *ndarray of str objects*** Transformed feature names.

Outline

- Part II introduction
- Introduction to Clustering
- Dimensionality Reduction
  - PCA (Principal Component Analysis)
    - แนวคิดของ PCA
    - กระบวนการ
    - รายละเอียดของผลลัพธ์
    - เมื่อข้อมูลมีมิติมากขึ้น
  - **เทคนิค Dimensionality Reduction อื่น ๆ ที่น่าสนใจ**

เทคนิค Dimensionality Reduction อื่นๆ

- t-distributed stochastic neighbor embedding (T-SNE)
  - sklearn.manifold.TSNE
- Random projections
  - sklearn.random_projection
- Feature agglomeration
  - sklearn.cluster.FeatureAgglomeration

Conclusion

- Part II introduction
- Introduction to Clustering
- Dimensionality Reduction
  - PCA (Principal Component Analysis)
    - แนวคิดของ PCA
    - กระบวนการ
    - รายละเอียดของผลลัพธ์
    - เมื่อข้อมูลมีมิติมากขึ้น
  - เทคนิค Dimensionality Reduction อื่น ๆ ที่น่าสนใจ