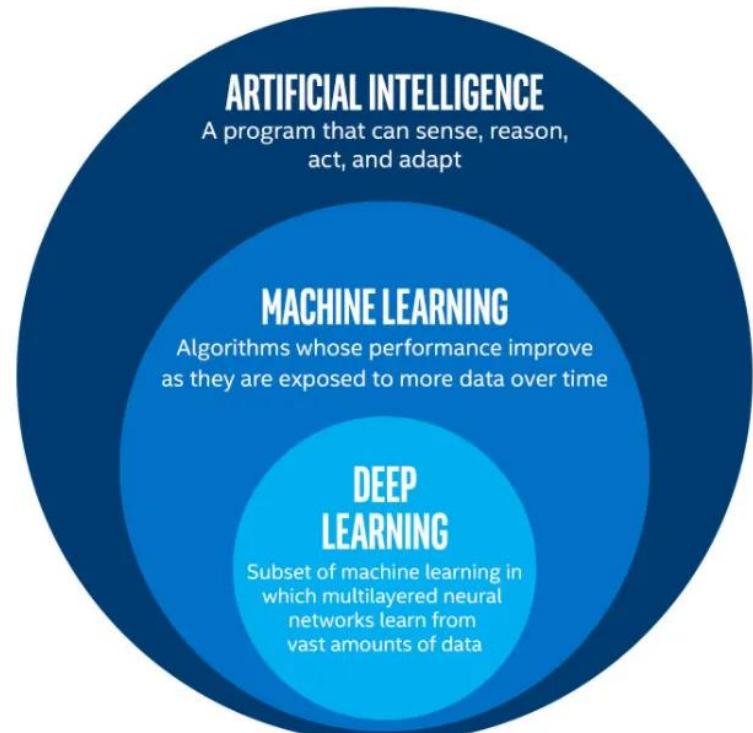


Chap 6 Giới thiệu Machine Learning cơ bản

1. Khái niệm về Machine Learning

- AI: Trí tuệ nhân tạo (AI) là công nghệ cho phép máy tính và máy móc mô phỏng quá trình học hỏi, hiểu biết, giải quyết vấn đề, ra quyết định, sáng tạo và tự chủ của con người.
- ML: Học máy (ML) là một nhánh của khoa học máy tính tập trung vào việc sử dụng dữ liệu và thuật toán để cho phép AI bắt chước cách con người học, từ đó dần dần cải thiện độ chính xác của nó.



[*] <https://ai.stackexchange.com/questions/15859/is-machine-learning-required-for-deep-learning>

1. Khái niệm về Machine Learning

Định nghĩa Machine Learning

“Một chương trình máy tính được gọi là học từ kinh nghiệm E để hoàn thành nhiệm vụ T, với hiệu quả được đo bằng phép đánh giá P, nếu hiệu quả của nó khi thực hiện nhiệm vụ T, khi được đánh giá bởi P, cải thiện theo kinh nghiệm E.”

Tom Mitchell

1. Khái niệm về Machine Learning

- Nhiệm vụ T:
 - Classification
 - Regression
 - Clustering
 - Dimension Reduction
- Phép đánh giá P: đo đạc kết quả của nhiệm vụ T
- Kinh Nghiệm E: Tập dữ liệu
- Hàm mất mát và tham số mô hình
- Feature Vector - Vector đặc trưng: biểu diễn các điểm dữ liệu

2. Phân loại học máy

Supervised Learning - Học Có Giám Sát

- Cho trước bộ dữ liệu (datasets) bao gồm các vector đặc trưng x với nhãn y tương ứng.
- Tìm hàm số biểu diễn mối quan hệ giữa x và y
- Regression - hồi quy
- Classification - phân loại

2. Phân loại học máy

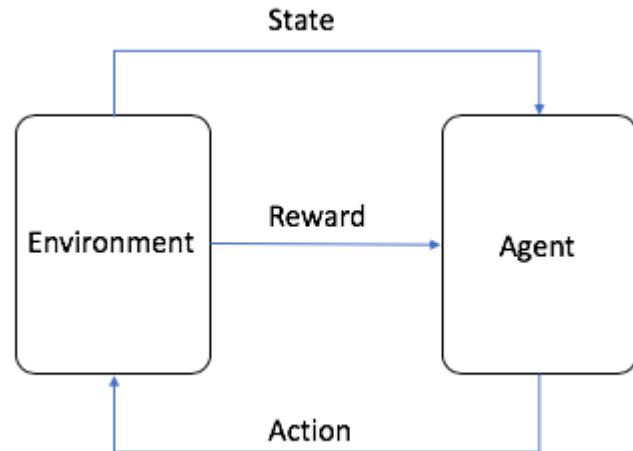
Unsupervised Learning - Học Không Giám Sát

- Cho trước bộ dữ liệu (datasets) bao gồm các vector đặc trưng x mà **không có** nhãn y tương ứng.
- Tìm cấu trúc hoặc giảm số chiều của dữ liệu
- Clustering - Phân cụm
- Dimension Reduction - Giảm chiều dữ liệu

2. Phân loại học máy

Reinforcement Learning - Học Tăng Cường

- Quan sát trạng thái (State) - đưa ra hành động (Action) - tối đa phần thưởng (Reward)



3. Giới thiệu về feature engineering

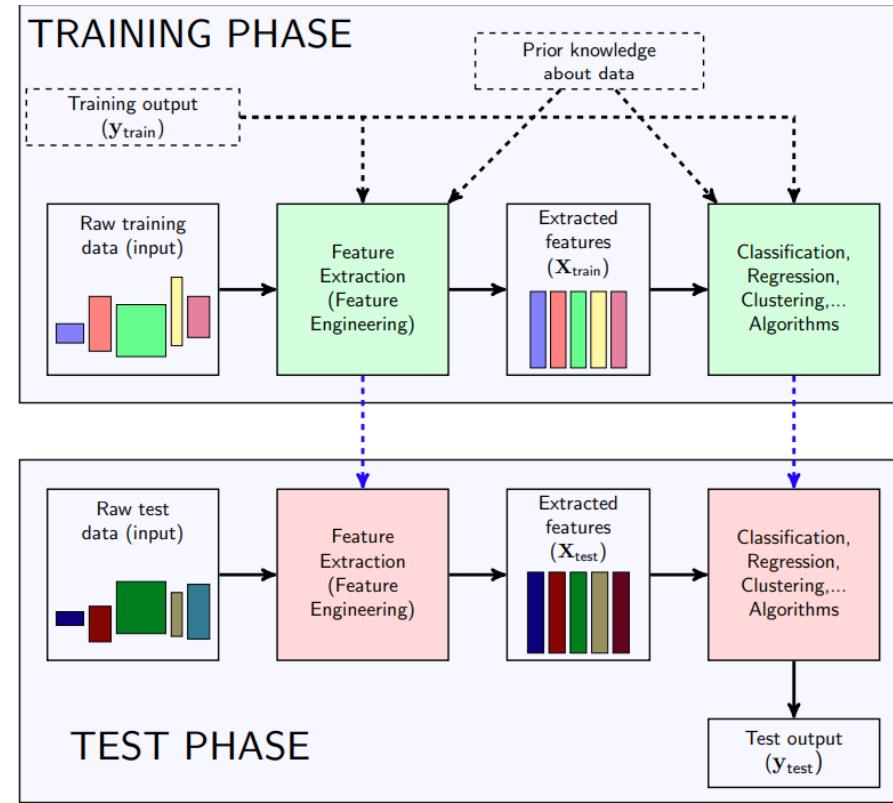
- Mỗi điểm dữ liệu trong các bài toán machine learning thường được biểu diễn bằng một vector được gọi là vector đặc trưng (feature vector).
- Tuy nhiên, trên thực tế, dữ liệu thường ở dạng thô (raw data) với kích thước khác nhau.
- Dữ liệu có thể chứa nhiều nhiễu.
- Thậm chí khi kích thước của các điểm là như nhau, việc lựa chọn, tính toán đặc trưng nào phù hợp cho mỗi bài toán là nhiệm vụ quan trọng trước tiên cần được giải quyết.

3. Giới thiệu về feature engineering

- Loại ra những dữ liệu nhiễu.
- Đưa dữ liệu thô với kích thước, hay số chiều khác nhau về cùng một chuẩn (cùng là các vector hoặc ma trận).
- Dữ liệu chuẩn mới này phải đảm bảo giữ được những thông tin đặc trưng cho dữ liệu thô ban đầu.
- Thiết kế những phép biến đổi để có những đặc trưng phù hợp.
- Quá trình quan trọng này được gọi là trích chọn đặc trưng (feature engineering hay feature extraction).
- Quá trình trích chọn đặc trưng thường yêu cầu kiến thức chuyên môn.

4. Các bước thực hiện một bài toán ML

- Bước 1: Trích chọn đặc trưng/Chuẩn hóa vector đặc trưng
- Bước 2: Xây dựng mô hình
- Bước 3: Kiểm thử
 - Validation
 - Final testing



4. Các bước thực hiện một bài toán ML

Chuẩn hóa vector đặc trưng

- Rescaling:

$$x'_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

- Standardization:

$$x'_i = \frac{x_i - \bar{x}_i}{\sigma_i}$$

- Scaling to unit length:

$$\mathbf{x}' = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$$

4. Các bước thực hiện một bài toán ML

Biến đổi đặc trưng theo hàm

- Trong trường hợp dữ liệu có *phương sai thay đổi* (*heteroscedasticity*) thì chúng ta có thể sử dụng một số hàm biến đổi biến đầu vào để tạo ra những biến có phương sai ổn định và dạng phân phối gần với phân phối chuẩn hơn như logrith, căn bậc 2, căn bậc 3.

Tạo đặc trưng mới

- Trong một mô hình có ít biến đầu vào thì sử dụng biến tương tác có thể giúp tạo ra nhiều biến giải thích mới giúp ích cho mô hình.
- Biến tương tác là những biến kết hợp từ nhiều biến đầu vào chẳng hạn như $x_1x_2, x_1x_2x_3, \dots$. Biến tương tác có thể là tích của hai hoặc nhiều biến.
- Biến bậc cao là những biến được tạo thành từ biến đầu vào bằng cách luỹ thừa với giá trị bậc cao, có thể là bậc 2, 3, ...

4. Các bước thực hiện một bài toán ML

Biểu diễn các biến rời rạc

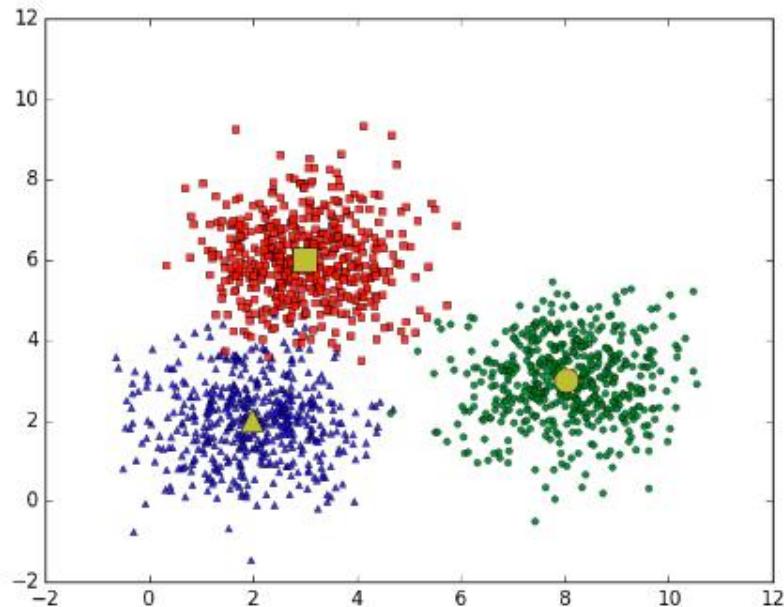
- One-hot encoding

Giá trị	Mã one-hot
"Sài Gòn"	[0, 0, 1]
"Huế"	[0, 1, 0]
"Hà Nội"	[1, 0, 0]

5. Một số thuật toán Unsupervised Learning

5.1 K-means clustering

- Phân chia các điểm dữ liệu thành các cụm có độ tương đồng về đặc trưng.
- Trong thuật toán k-Means chúng ta được cung cấp một tập dữ liệu đầu vào $\{x_1, x_2, \dots, x_n\}$, trong đó $x \in \mathbb{R}^d$ và phân cụm chúng vào những nhóm dữ liệu có tính chất chung. Điểm đặc biệt của tập dữ liệu này là chúng hoàn toàn chưa được gán nhãn. Như vậy k-Means là một thuật toán thuộc lớp các mô hình học không giám sát



Bài toán với 3 clusters.

5. Một số thuật toán Unsupervised Learning

5.1 K-means clustering

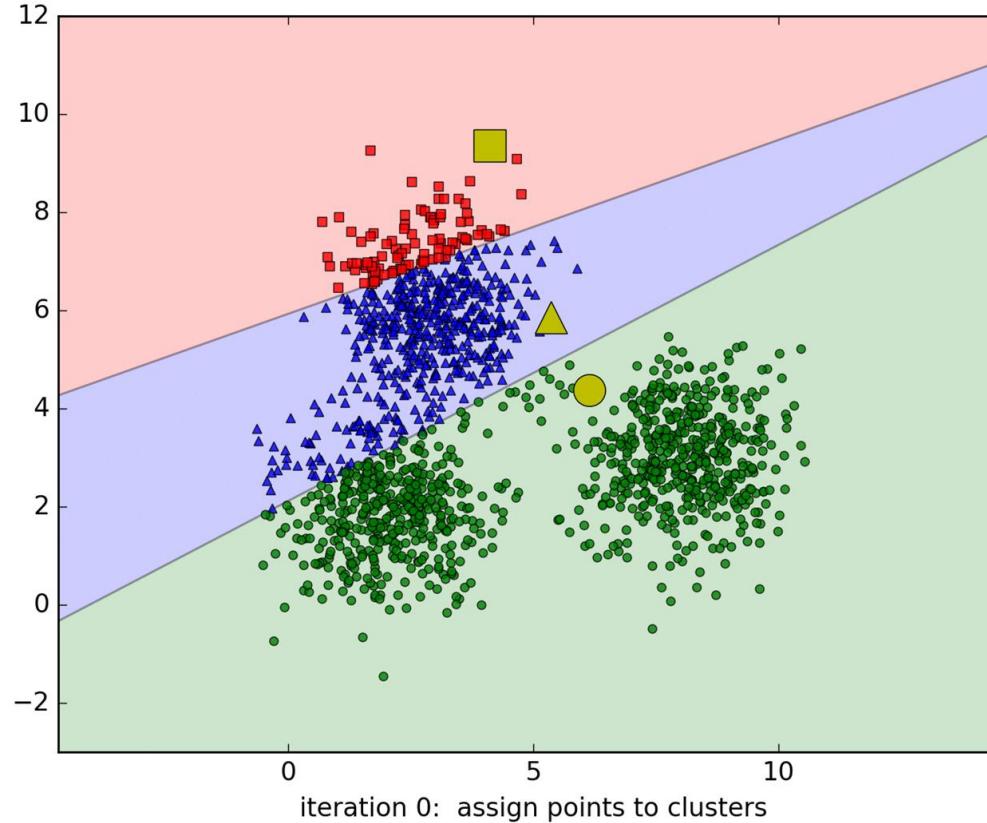
- Trong thuật toán k-Means mỗi cụm dữ liệu được đặc trưng bởi một tâm (centroid). tâm là điểm đại diện nhất cho một cụm và có giá trị bằng trung bình của toàn bộ các quan sát nằm trong cụm.
- Chúng ta sẽ dựa vào khoảng cách từ mỗi quan sát tới các tâm để xác định nhãn cho chúng trùng thuộc về tâm gần nhất.
- Ban đầu thuật toán sẽ khởi tạo ngẫu nhiên một số lượng xác định trước tâm cụm.
- Sau đó tiến hành xác định nhãn cho từng điểm dữ liệu và tiếp tục cập nhật lại tâm cụm.
- Thuật toán sẽ dừng cho tới khi toàn bộ các điểm dữ liệu được phân về đúng cụm hoặc số lượt cập nhật tâm chạm ngưỡng.

5. Một số thuật toán Unsupervised Learning

5.1 K-means clustering

- Tóm tắt thuật toán:
 - **Input:**
 - K: số cluster
 - Bộ dữ liệu: $\{x_1, x_2, \dots, x_n\}$ với $x \in \mathbb{R}^d$
 - **Thuật toán:**
 - 1. Khởi tạo ngẫu nhiên K tâm của các cụm (cluster centroids)
 $\mu_1, \mu_2, \dots, \mu_k$.
 - 2. Lặp lại quá trình cập nhật tâm cụm cho tới khi dừng
 - Xác định nhãn cho từng điểm dữ liệu dựa vào khoảng cách tới tâm cụm
 $c_i = \arg \min_j \|x_i - \mu_j\|_2^2$
 - Tính lại giá trị của từng tâm cụm
$$\mu_j := \frac{\sum_{i=1}^n \mathbf{1}(c_i = j) x_i}{\sum_{i=1}^n \mathbf{1}(c_i = j)}$$

5. Một số thuật toán Unsupervised Learning



5. Một số thuật toán Unsupervised Learning

5.1 K-means clustering

- Hàm biến dạng

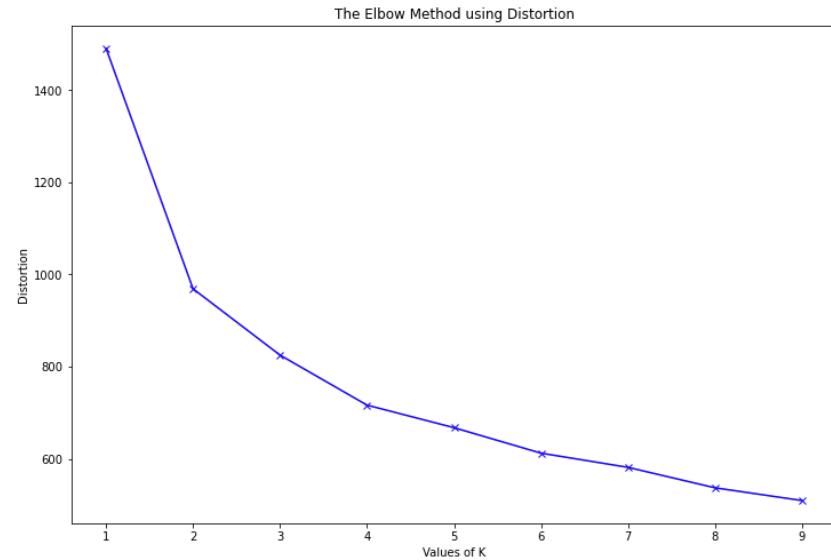
$$\mathcal{L}(\mathbf{c}, \mu) = \sum_{i=1}^n \sum_{j=1}^k \|\mathbf{x}_i - \mathbf{1}(c_i = j)\mu_j\|_2^2$$

- Trong đó $\mathbf{c} = (c_1, c_2, \dots, c_n)$ là các nhãn của toàn bộ các điểm dữ liệu được dự đoán bởi thuật toán.
- $\mathbf{1}(c_i = j)\mu_j$ chính là tâm cụm mà quan sát x_i được phân bổ về.
- Hàm biến dạng luôn giảm sau mỗi vòng lặp. Mặt khác hàm biến dạng bị chặn dưới bởi 0 nên là một chuỗi hội tụ. Tức là sau một hữu hạn bước thì thuật toán k-Means sẽ dừng.

5. Một số thuật toán Unsupervised Learning

5.1 K-means clustering

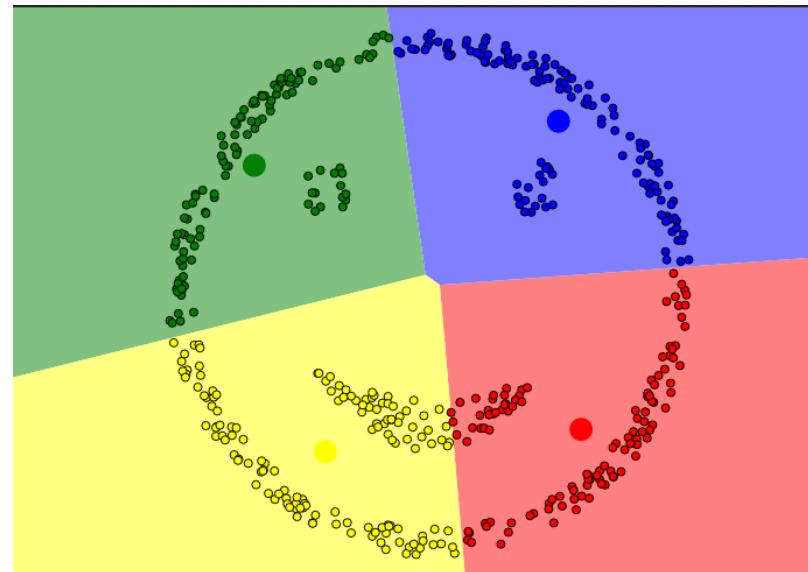
- Chọn số cụm bằng phương pháp elbow
 - Điểm *khuy tay* là điểm mà ở đó tốc độ suy giảm của *hàm biến dạng* sẽ thay đổi nhiều nhất.
 - Phương pháp Elbow là một phương pháp thường được sử dụng để lựa chọn số lượng cụm phân chia hợp lý dựa trên biểu đồ.
 - Tuy nhiên có một số trường hợp chúng ta sẽ không dễ dàng phát hiện vị trí của Elbow, đặc biệt là đối với những bộ dữ liệu mà quy luật phân cụm không thực sự dễ dàng được phát hiện.



5. Một số thuật toán Unsupervised Learning

5.1 K-means clustering

- Hạn chế của thuật toán
 - Phải xác định trước số cụm
 - Nhạy cảm với điều kiện khởi tạo
 - Trong nhiều trường hợp dữ liệu phân bố phức tạp, thuật toán sẽ khó hội tụ.
 - Nhạy cảm với outliers
 - Yêu cầu lớn về bộ nhớ.



5. Một số thuật toán Unsupervised Learning

5.1 K-means clustering

- Chúng ta có thể dễ dàng cài đặt thuật toán K-means thông qua class KMeans của thư viện sklearn.cluster

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=2, random_state=0)
kmeans.fit(dataset)
print('kmeans labels of 5 observation: ', kmeans.labels_[:5])
print('kmeans centroids: \n', kmeans.cluster_centers_)
```

```
kmeans labels of 5 observation: [0 0 0 1 0]
kmeans centroids:
 [[ 4.58876493 -3.13006162]
 [-4.99023469  0.44409831]]
```

Dự báo cụm cho một quan sát mới:

```
x = np.array([[5, 1]])
kmeans.predict(x)
```

```
array([0], dtype=int32)
```

5. Một số thuật toán Unsupervised Learning

5.2 Dimensionality Reduction - Giảm chiều dữ liệu - PCA

- Mục đích: Những bộ dữ liệu lớn thường tiêu tốn nhiều bộ nhớ lưu trữ và thời gian huấn luyện. Do đó khi đối mặt với những bộ dữ liệu kích thước lớn chúng ta thường tìm cách giảm chiều dữ liệu từ không gian cao chiều (*high dimensionality*) xuống không gian thấp chiều (*low dimensionality*) mà vẫn giữ được những đặc trưng chính của dữ liệu nhưng tiết kiệm được chi phí huấn luyện và dự báo.

5. Một số thuật toán Unsupervised Learning

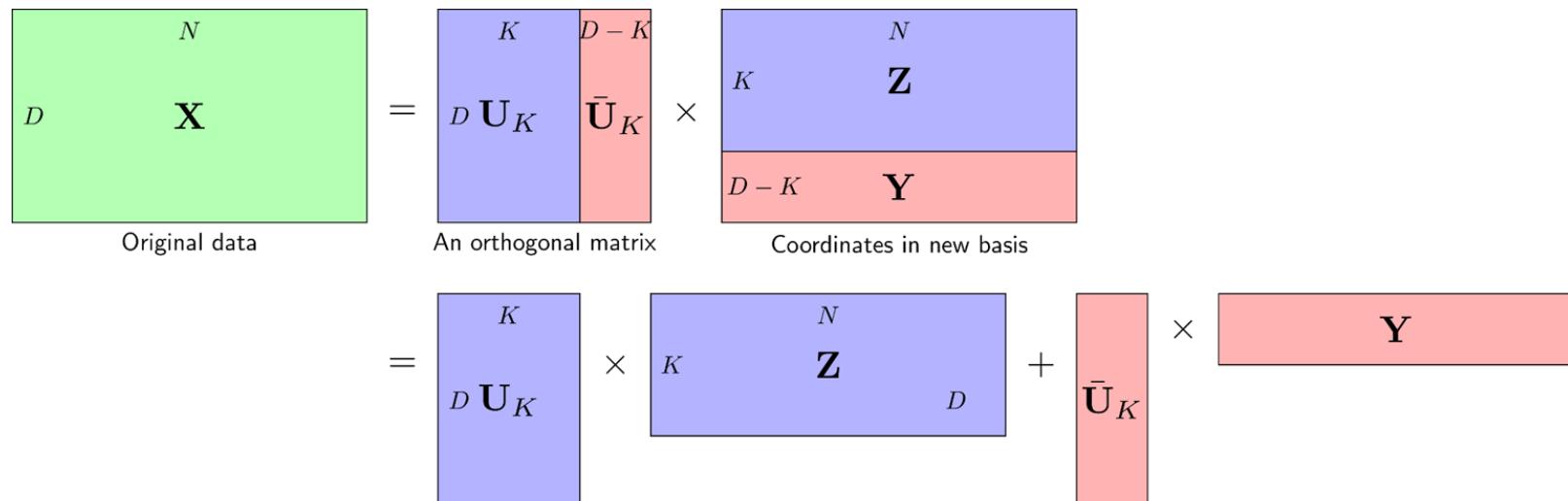
5.2 Dimensionality Reduction - Giảm chiều dữ liệu - PCA

- PCA là viết tắt của cụm từ *principal component analysis*. Thuật ngữ Tiếng Việt còn gọi là *phân tích thành phần chính*.
- Đây là một phương pháp giảm chiều dữ liệu (*dimensionality reduction*) tương đối hiệu quả dựa trên phép phân tích suy biến (*singular decomposition*) mà ở đó chúng ta sẽ chiếu các điểm dữ liệu trong không gian cao chiều xuống một số ít những véc tơ thành phần chính trong không gian thấp chiều mà đồng thời vẫn bảo toàn tối đa độ biến động của dữ liệu sau biến đổi.
- Ưu điểm của PCA đó là sử dụng tất cả các biến đầu vào nên phương pháp này không bỏ sót những biến quan trọng.

5. Một số thuật toán Unsupervised Learning

5.2 Dimensionality Reduction - Giảm chiều dữ liệu - PCA

- PCA chính là phương pháp đi tìm một hệ cơ sở mới sao cho thông tin của dữ liệu chủ yếu tập trung ở một vài toạ độ, phần còn lại chỉ mang một lượng nhỏ thông tin.



[*]<https://machinelearningcoban.com/2017/06/15/pca/#3-principal-component-analysis>

5. Một số thuật toán Unsupervised Learning

5.2 Dimensionality Reduction - Giảm chiều dữ liệu - PCA

- Thuật toán:

- 1. Tính vector kỳ vọng của toàn bộ dữ liệu

$$\mathbf{x} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n$$

- 2. Trừ mỗi điểm dữ liệu với vector kỳ vọng của toàn bộ dữ liệu

$$\hat{\mathbf{x}}_n = \mathbf{x}_n - \mathbf{x}$$

- 3. Tính ma trận hiệp phương sai - covariance matrix

$$\mathbf{S} = \frac{1}{N} \hat{\mathbf{x}} \hat{\mathbf{x}}^T$$

5. Một số thuật toán Unsupervised Learning

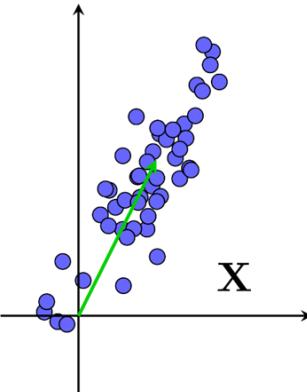
5.2 Dimensionality Reduction - Giảm chiều dữ liệu - PCA

- Thuật toán:
 - 4. Tính các giá trị riêng và vector riêng có norm bằng 1 của ma trận hiệp phương sai, sắp xếp theo thứ tự giảm dần của giá trị riêng.
 - 5. Chọn K vector riêng ứng với K giá trị riêng lớn nhất để xây dựng ma trận U_K có các cột tạo thành một hệ trực giao. K vectors này được gọi là các thành phần chính, tạo thành một không gian con gần với phân bố của dữ liệu ban đầu.
 - 6. Chiếu dữ liệu ban đầu theo không gian con tìm được
 - 7. Dữ liệu mới là tọa độ của các điểm dữ liệu trên không gian mới

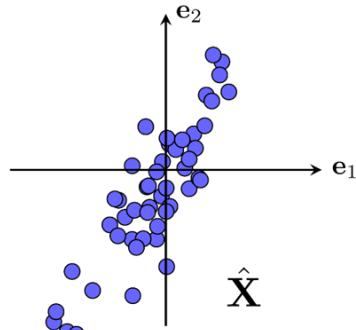
$$\mathbf{Z} = \mathbf{U}_K^T \hat{\mathbf{X}}$$

PCA procedure

1. Find mean vector



2. Subtract mean

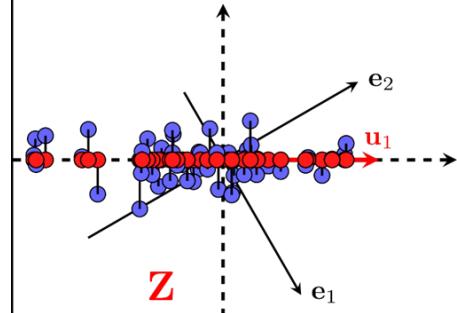


3. Compute covariance matrix:

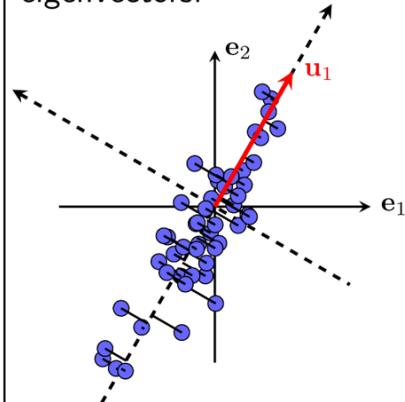
$$\mathbf{S} = \frac{1}{N} \hat{\mathbf{X}} \hat{\mathbf{X}}^T$$

4. Computer eigenvalues and eigenvectors of \mathbf{S} :
 $(\lambda_1, \mathbf{u}_1), \dots, (\lambda_D, \mathbf{u}_D)$
Remember the orthonormality of \mathbf{u}_i .

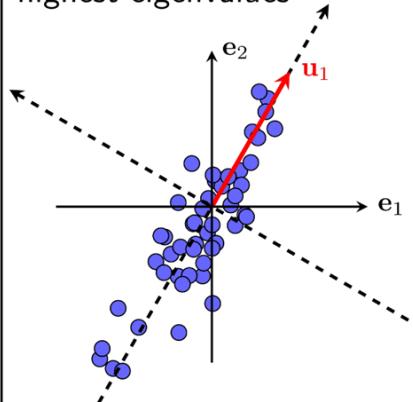
7. Obtain projected points in low dimension.



6. Project data to selected eigenvectors.



5. Pick K eigenvectors w. highest eigenvalues



```
from sklearn import datasets
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

iris = datasets.load_iris()

X = iris.data
y = iris.target
target_names = iris.target_names

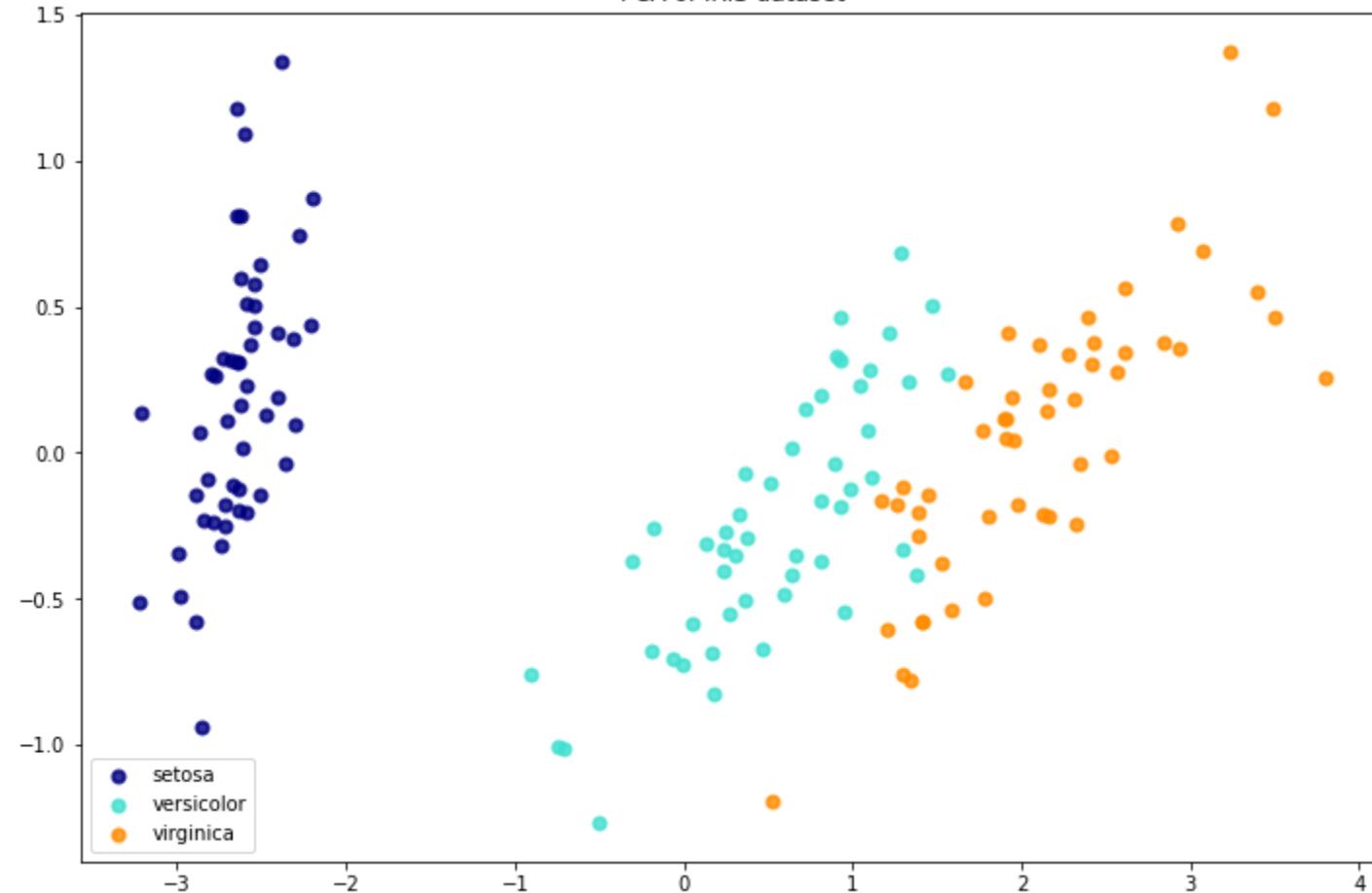
pca = PCA(n_components=2)
X_r = pca.fit(X).transform(X)

plt.figure(figsize=(12, 8))
colors = ['navy', 'turquoise', 'darkorange']
lw = 2

for color, i, target_name in zip(colors, [0, 1, 2], target_names):
    plt.scatter(X_r[y == i, 0], X_r[y == i, 1], color=color, alpha=.8, lw=lw,
                label=target_name)
plt.legend(loc='best', shadow=False, scatterpoints=1)
plt.title('PCA of IRIS dataset')

plt.show()
```

PCA of IRIS dataset



6. Một số thuật toán Supervised Learning

6.1 Hồi quy tuyến tính - Linear Regression

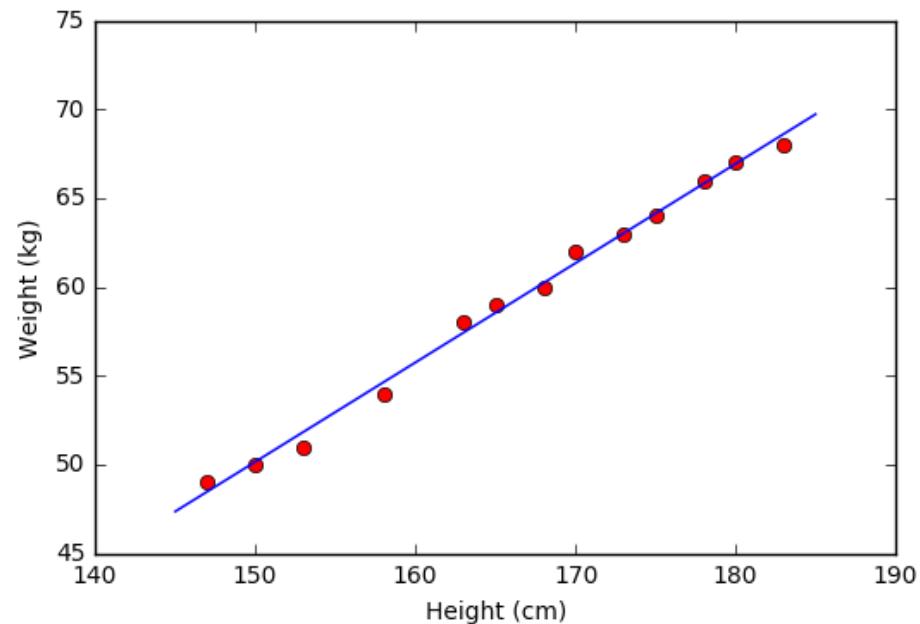
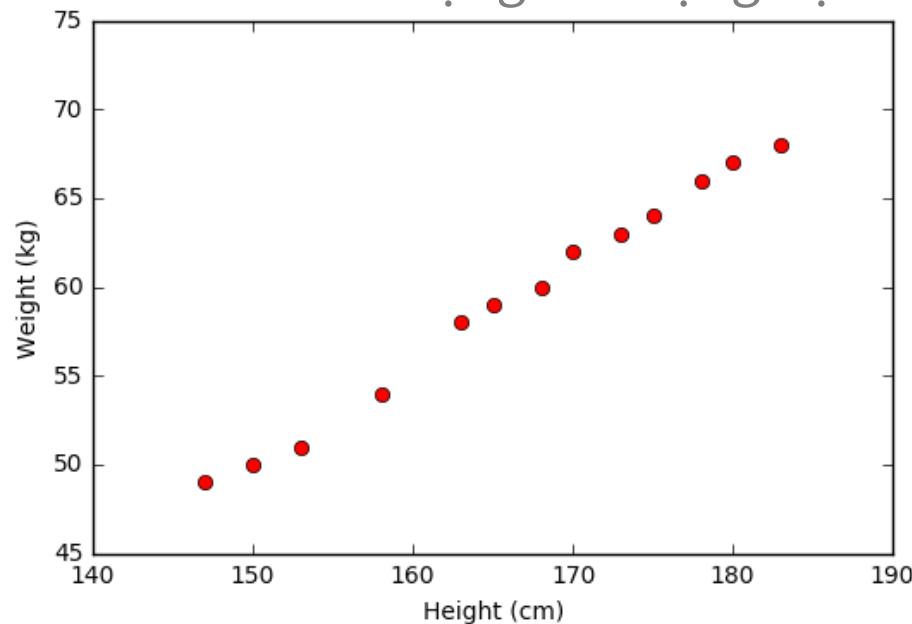
- Ví dụ:
 - Ước lượng cân nặng dựa vào chiều cao

Chiều cao (cm)	Cân nặng (kg)	Chiều cao (cm)	Cân nặng (kg)
147	49	168	60
150	50	170	72
153	51	173	63
155	52	175	64
158	54	178	66
160	56	180	67
163	58	183	68
165	59		

6. Một số thuật toán Supervised Learning

6.1 Hồi quy tuyến tính - Linear Regression

- Ví dụ:
 - Ước lượng cân nặng dựa vào chiều cao



6. Một số thuật toán Supervised Learning

6.1 Hồi quy tuyến tính - Linear Regression

- Hồi quy tuyến tính với một biến
 - Model:

$$\hat{y} = \mathbf{f}(x) = \theta_0 + \theta_1 x \quad (1)$$

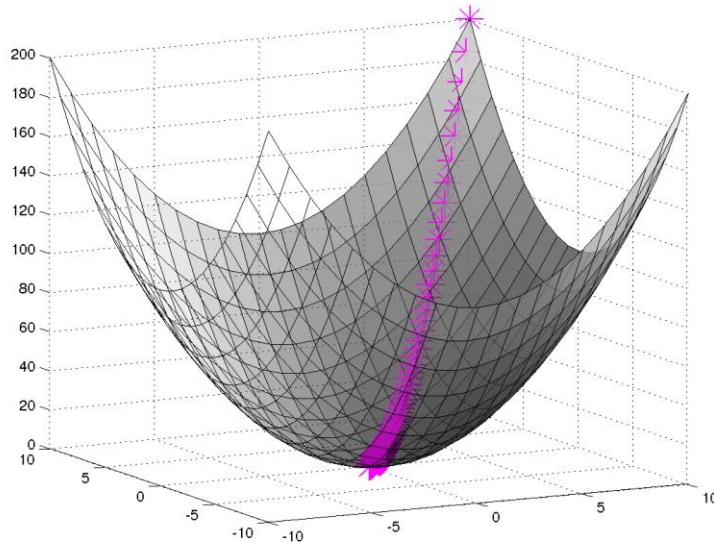
- Loss function - MSE:

$$\mathbf{J}(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (\hat{y}^i - y^i)^2 \quad (2)$$

6. Một số thuật toán Supervised Learning

6.1 Hồi quy tuyến tính - Linear Regression

- Hồi quy tuyến tính với một biến
 - Gradient descent:



$$\theta_j = \theta_j - \alpha \frac{\partial \mathbf{J}(\theta_0, \theta_1)}{\partial \theta_j} \quad (3)$$

where : $j = 0, 1$

6. Một số thuật toán Supervised Learning

6.1 Hồi quy tuyến tính - Linear Regression

- Tổng quát:

$$\vec{\theta} = \vec{\theta} - \alpha \frac{\partial \mathbf{J}(\vec{\theta})}{\partial \vec{\theta}} \quad (4)$$

6. Một số thuật toán Supervised Learning

6.1 Hồi quy tuyến tính - Linear Regression

- Phương trình chuẩn - Normal Equation

$$\vec{\theta} = (X^T X)^{-1} X^T Y \quad (4)$$

6. Một số thuật toán Supervised Learning

6.1 Hồi quy tuyến tính - Linear Regression

$$\vec{x}^i = \begin{bmatrix} 1 \\ x_1^i \\ \dots \\ x_n^i \end{bmatrix} \quad \vec{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \dots \\ \theta_n \end{bmatrix} \quad Y = \begin{bmatrix} y^1 \\ y^2 \\ \dots \\ y^m \end{bmatrix}$$

$$X = \begin{bmatrix} x^{1T} \\ x^{2T} \\ \dots \\ x^{mT} \end{bmatrix} = \begin{bmatrix} 1 & x_1^1 \dots x_n^1 \\ 1 & x_1^2 \dots x_n^2 \\ \dots & \dots \\ 1 & x_1^m \dots x_n^m \end{bmatrix}_{m \times (n+1)}$$

6. Một số thuật toán Supervised Learning

6.1 Hồi quy tuyến tính - Linear Regression

<u>Gradient Descent</u>	<u>Normal Equation</u>
Cần tinh chỉnh learning rate	Không cần learning rate
Tìm nghiệm bằng quá trình lặp	Không cần lặp
$O(kN^2)$	$O(N^3)$
Hiệu quả đối với không gian nhiều chiều	Không hiệu quả đối với dữ liệu lớn và nhiều chiều
	$(X^T X)$ có thể không tồn tại nghịch đảo

6. Một số thuật toán Supervised Learning

6.1 Hồi quy tuyến tính - Linear Regression

- Có thể mở rộng ra hồi quy bậc cao

$$X = [x_1, x_2] \rightarrow \tilde{X} = [x_1, x_2, x_1^2, x_2^2, \sin(x_1), \cos(x_2), x_1 x_2]$$

- Các hạn chế của Linear Regression:
 - Nhạy cảm với nhiễu.
 - >> cần tiến xử lý loại bỏ outlier
 - Không biểu diễn được các mối quan hệ phức tạp
 - Lựa chọn và biến đổi đặc trưng yêu cầu kiến thức chuyên môn

6. Một số thuật toán Supervised Learning

6.1 Hồi quy tuyến tính - Linear Regression

Example

6. Một số thuật toán Supervised Learning

6.1 Logistic Regression

- Model:

$$y \in \{0, 1\}$$

$$f(x) = P(y = 1|x; \theta) = g(\theta^T x)$$

- Hàm $g()$ được gọi là hàm kích hoạt - activation function
 - Sigmoid
 - Softmax
 - tanh

6. Một số thuật toán Supervised Learning

6.1 Logistic Regression

- Loss Function - Cross Entropy:

$$\begin{aligned}\mathbf{J}(\theta) &= -\frac{1}{m} \sum_{i=1}^m [y^i \log(f(x^i)) + (1 - y^i) \log(1 - f(x^i))] \\ &= -\frac{1}{m} \sum_{i=1}^m \sum_{j=1}^c y_j^i \log(P_j^i)\end{aligned}$$

6. Một số thuật toán Supervised Learning

6.1 Logistic Regression

- Các biến thể của Gradient Descent:
 - Mini-Batch Gradient Descent
 - Stochastic Gradient Descent - SGD
 - Momentum
 - Nesterov Accelerated Gradient
 - AdaGrad
 - AdaDelta
 - RMSProp
 - Adam

6. Một số thuật toán Supervised Learning

6.1 Logistic Regression

- Example

7. Lựa chọn đặc trưng

- Vét cạn - Exhaustive Search
- Stepwise Regression
 - Forward selection
 - Backward elimination
- Quan sát độ tương quan giữa các biến và nhãn
 - Độ tương quan cao -> giữa các biến có tương tác mạnh

SVM

- **Siêu phẳng (hyperplane)** là một khái niệm hình học tổng quát:
 - Trong 2D (2 chiều): siêu phẳng là **một đường thẳng**.
 - Trong 3D (3 chiều): siêu phẳng là **một mặt phẳng**.
 - Trong n chiều: siêu phẳng là **một mặt có $n-1$ chiều**.
-  Trong bài toán phân loại, siêu phẳng là **đường/mặt phân chia các lớp**.