**DSP303x- Machine Learning for Data Science**

Module 1 - What is Machine Learning?

* + - Lesson 1 - Introduction to Machine Learning

Module 2 – Regression

Định nghĩa về regression

* Regression giải quyết bài toán nào và các vấn đề tương ứng?

Regression là phương pháp tìm mối quan hệ giữa một tập các biến độc lập (giải thích- features) và biến phụ thuộc Y. Hay nói cách khác là giải thích cho sự thay đổi của Y bằng các giá trị của biến x.

Ta tìm sự giải thích của từng biến độc lập lên sự thay đổi của Y bằng cách lấy đạo hàm theo biến độc lập đó để tìm ra beta tương ứng.

* Có gì khác biệt giữa regression và classification.

Regression là khi biến phụ thuộc (y) của bài toán là giá trị liên tục còn khi biến Y là giá trị rời rạc thì sẽ là bài toán classification.

* Ứng dụng của regression.

Regression được áp dụng vô cùng rộng rãi trong các doanh nghiệp do nhu cầu dự báo, dự đoạn để có kế hoạch và sự chuẩn bị tốt cho các tình huống đó:

Ví dụ:

Các bài toán khác nhau của Regression:

* Linear regression.
* Logistic regression.
* Multivariable regression.
* Polynomial regression. (đa thức, các bậc cao hơn bậc 1)

Mô hình hóa bài toán Regression và Regression dưới góc nhìn toán học

* Mô hình hóa bài toán Regression

Chọn biến x bằng correlation matrix như nào?

Với các biến dummy thì ta xử lý thế nào: Embed encoding

Sau khi chọn dược biến thì ta cần xây dựng một model xung quanh các biến đó và fit nó.

* Ý nghĩa và góc nhìn toán học của bài toán Regression.

Ta đưa hệ linear equation về dạng Y=bx với X là ma trận cỡ m\*n (M hàng n cột) và B là ma trận có kích thước n\*1 và Y là ma trận với kích thước m\*1.

Sau khi đưa về hệ trên ta cần xử lý thêm các bước như sau:

- Đưa hệ các vector cột thành tập các vector độc lập tuyến tính (Chính là việc loại bỏ các cột có tương quan mạnh với nhau hay biểu diễn được cho nhau) đây chính là idea của việc giảm chiều dữ liệu

- Sau đó nếu hệ quá phức tạp ta có thể sử dụng Matrix decomposition để tách thành tích hai ma trận giúp cho việc xử lý dễ dàng hơn

- Sau đó ta sẽ fit hệ trên và giải bình thường.

Các cách chính để giải quyết bài toán linear regression:

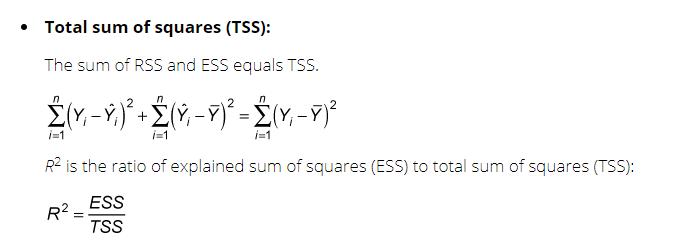
* Analytical solution (normal equation)
* Optimization solution (gradient descent)

# 2 Regression Algorithm

# 2.1 Optimization problems

* Định nghĩa của hàm loss (objective) function:

RSS: Residual sum squared: Tổng bình phương phần dư hay khoảng cách eculidean giữa predicted value vs actual value.



* Công thức hàm loss và đạo hàm của hàm loss.
* Mục đích tối ưu hàm objective function: Min/max.

Ta cần tìm một mô hình giải thích tốt nhất cho những dữ liệu mà ta có, khi đưa sang vấn đề toán học có thể hiểu được là ta cần tìm bộ tham số Beta sao cho giá trị dự đoạn là gần sát nhất so với giá trị thực tế.

* Gradient descent/ascent: Định nghĩa và ý tưởng của phương pháp.

Do việc tìm cực trị hàm số bằng Chain rules gặp khó khăn ở việc tính toán và đạo hàm và thường khó để có thể sử dụng vào các bài toán thực tiễn nên ta phải đi giải bài toán xấp xỉ, phần này thì ta cần lưu ý tới điều kiện dừng của thuật toán: Số vòng lặp tối đa hoặc giá trị đạo hàm đủ nhỏ (gần với 0) để ta có thể dừng thuật toán.

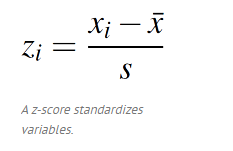
* Hàm cập nhật weight.

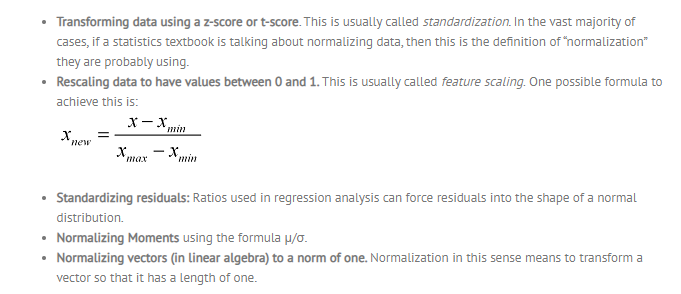
Ta cập nhật weight theo từng giá trị đạo hàm mà ta tìm được ở bước trên.

* Single/multivariable gradient descent and updated weights.

Ta làm tương tự với hàm Linear regression.

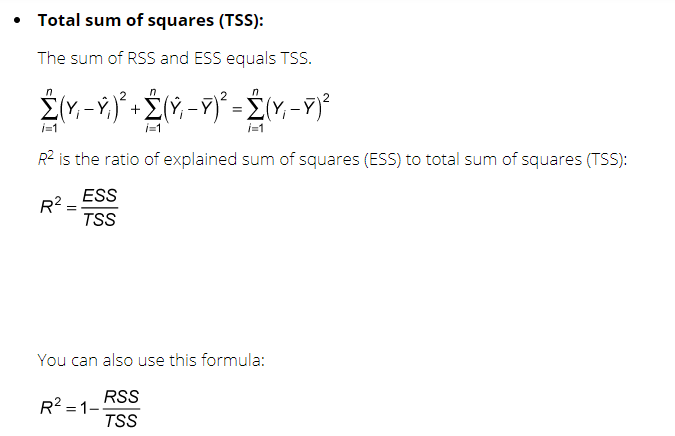
* Normalization: Min/max scaler, normalized function trong sklearn.





## 2.2 Performance assessment metric

* RSS: Định nghĩa và cách sử dụng.
* MSE Định nghĩa và cách sử dụng.
* Train/Validation/Test splitting.
* Cross-validation.



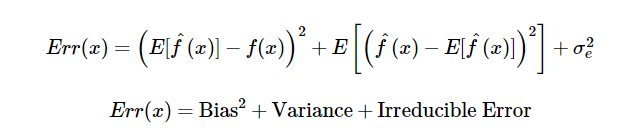
## 2.3 Overfitting

* Bias

Bias là sai lệch giữa giá trị dự đoán và giá trị chính xác của bộ dữ liệu. Mô hình bị Bias đồng nghĩa với việc nó đơn giản hóa và có sự sai lệch cao trong cả tập train và test.

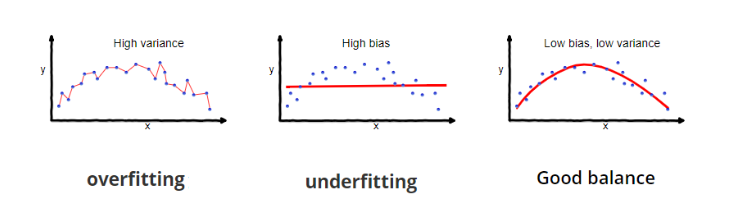
* Variance

Variance là mô hình tập trung vào việc giải thích sự phân bố của các data point được cho trước nên thường có hiệu quả rất tốt đối với Tập train nhưng rất tệ với tập test.



Trong Error của model gồm hai phần Variance và Bias ta có thể làm giảm được.

* Bias/Variance tradeoff



Ta buộc phải chọn xem cân đối như nào để mô hình không bị underffiting và overfitting. Nếu mô hình quá đơn giản thì có thể có độ lệch cao và phương sai thấp và ngược lại khi mô hình phức tạp

* How to avoid bias/variance?

(Theo em thì model hồi quy tuyến tính có xảy ra bias khi mà chúng ta tiếp cận theo phương pháp đã nêu ở trên không?

Câu trả lời là không vì liner là dạng đường thẳng, không bao giờ xảy ra tình huống uốn lượn để fit qua các điểm như trong overfit thường thấy.)

<https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229>

## 2.4 Cách đọc kết quả của mô hình hồi quy với các transformation thường gặp:

*TH1: Chỉ có biến phụ thuộc được Log- transformation*

trường hợp 1 giả sử model mình tìm ra là y= 2x thì khi x tăng lên 1 đơn vị y sẽ tăng lên 200%

*TH2: Chỉ có log-transform với biến x, biến Y không*

Ta chia hệ số cho 100. Nó cho biết rằng mức tăng 1% trong biến độc lập sẽ tăng (hoặc giảm) biến phụ thuộc theo đơn vị (hệ số / 100).

Ví dụ: hệ số là 0,198 thì 0,198 / 100 = 0,00198. Cứ tăng 1% trong biến độc lập, biến phụ thuộc của chúng ta tăng khoảng 0,002 đơn vị

*TH3: Cả biến Y và x đều log-transform*

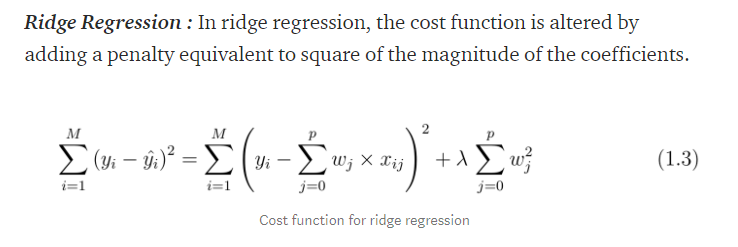
thì trường hợp này hệ số sẽ thể hiện mức tăng % của y khi tăng 1% của x

ví dụ: hệ số là 0,198. Cứ tăng 1% trong biến độc lập, biến phụ thuộc tăng 0,198%

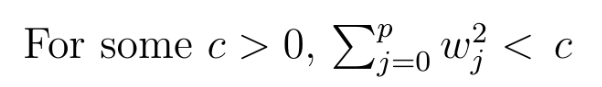
## 2.5 Regularization

## 2.5.1. L2 regularization

* L2 pelnaty: Công thức, hàm loss và đạo hàm của hàm loss.
* Ridge regression.



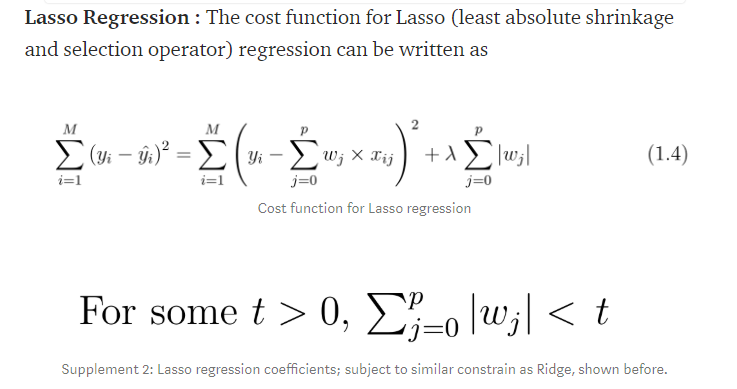
Nó chính là hàm Lagrange của Loss function với constraint cho các hệ số nhằm làm giảm sự biến động mạnh, không ổn định trong mô hình. Giá trị của lamda càng gần 0 thì hệ số thu được sẽ càng gần với Linear Model:



* Mục đích sử dụng L2 pelnaty: Reduce overfitting.

## 2.5.2 L1 regularization

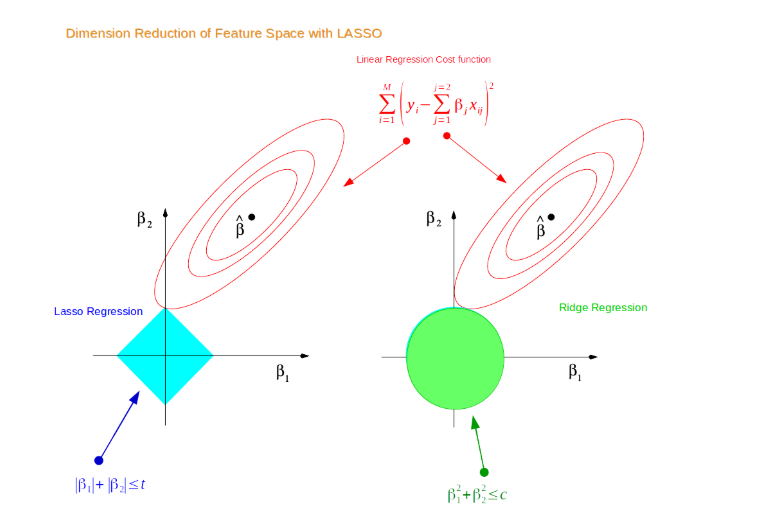
* L1 pelnaty: Công thức, hàm loss và đạo hàm của hàm loss.
* Lasso regression.



* Mục đích sử dụng L1 pelnaty: Reduce overfitting, feature selection.

Cả hai đều dùng để giảm overfit vì nó sẽ control độ lớn của các hệ số, giúp cho mô hình có bộ tham số nhỏ, giảm phức tạp và không bị biến động mạnh hay nhạy cảm với các outlier của dữ liệu

Tuy nhiên Lasso còn dùng để feature selection, ta có thể theo dõi ở đây:



Module 3 - Classification

**3.1 Bài toán Classification**

- Hiểu về bài toán Classification

- Hiểu về Supervised learning

- Một số thuật toán dùng trong Classification

**3.2 Logistic Regression**

Giới thiệu về Logistic Regression và Gradient Descent

Output của Logistic Regression có thể biểu diễn như sau:



Theta được gọi là hàm activation, đối với regression thì là f(x)=x. Đối với Logistic regression thì hàm activation cần thỏa mãn một số tính chất:

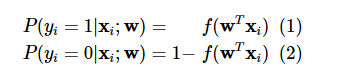
- Là hàm liên tục, có giá trị thực và trong khoảng 0-1

- Càng cách xa đường phân chia ở giữa càng thể hiện sự phân lớp rõ ràng (gần 0 hoặc gần 1)

- Có đạo hàm ở mọi điểm để có thể dễ dàng tìm cực trị

Hiểu về xác suất của class:

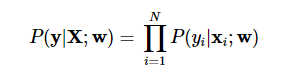
- Xác suất để y nhận giá trị tương ứng 1, 0 khi ta biết Xi và tham số w.



- Ta đi tìm w thỏa mãn:



- Khi giả sử các bản ghi độc lập với nhau, ta có xác suất cho cả bộ dữ liệu như sau:



- Và để dễ dàng trong việc tính đạo hàm, và đặc tính của hàm Log- transformation nên ta lấy log của function trên, và lấy dấu âm ta thu được:

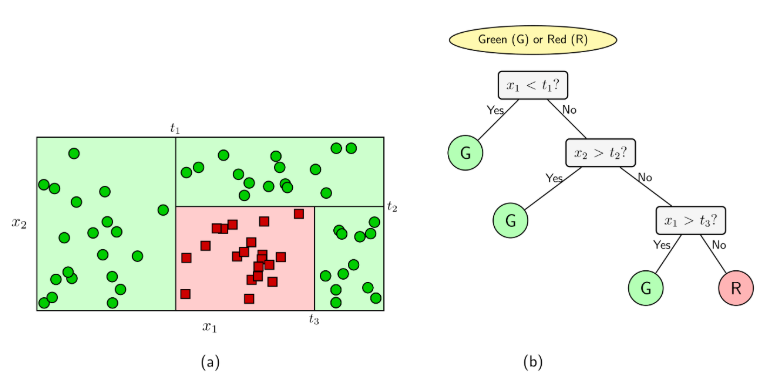


- Ta đi tìm min của hàm này, đồng nghĩa với việc tìm max của hàm log và coi đây là hàm mục tiêu của mô hình. Việc đi tìm W thỏa mãn điều kiện trên tương tự như với Linear Regression.

Ta có thể nhận ra rằng Regulation có thể áp dụng tương tự như với Regression.

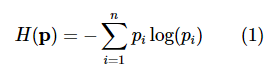
**3.3 Decision Trees**

- Giới thiệu về Decision tree



- Cách mà Decision tree thực thi:

Tại mỗi điểm ta có hàm Entropy thể hiện



- Hàm số này sẽ cho giá trị thấp nhất nếu dữ liệu trong mỗi child node nằm trong cùng một class (tinh khiết nhất), và cho giá trị cao nếu mỗi child node có chứa dữ liệu thuộc nhiều class khác nhau.

- Overfitting trong Decision tree, sử dụng Early stopping để loại bỏ Overfitting

**(**[**https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb**](https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb)**)**

**3.4 Support Vector Machine**

Giới thiệu về Large Margin Classification

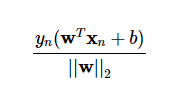
- SVM – Support vector machine là thuật toán đi tìm một đường thẳng/ mặt phẳng phân chia hai class trong không gian sao cho đáp ứng tiêu chí khoảng cách giữa hai điểm gần đường phân chia nhất của hai lớp là như nhau và lớn nhất, khoảng cách này được gọi là Margin- biên, nên đồng thời SVM cũng có tên gọi là Large Margin Classification.

- SVM và cách mà model được huấn luyện:

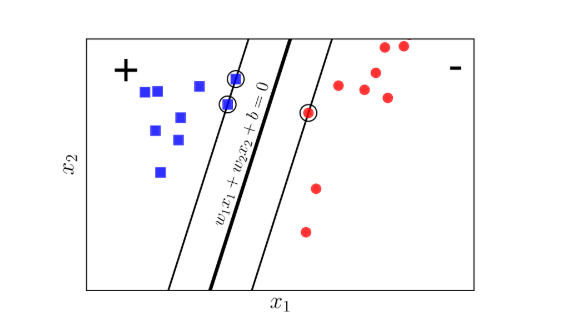
- Đầu tiên ta giả sử có một đường thẳng phân chia hai lớp như sau:

WTx+b = w1x1+w2x2+b=0

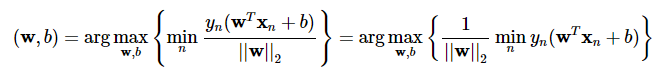
- Với một cặp (xn,yn) bất kỳ, ta có thể tính khoảng cách từ điểm đó tới đường thẳng phân chia là:



- Giá trị min của khoảng cách đó chính là Margin trong định nghĩa ở trên.



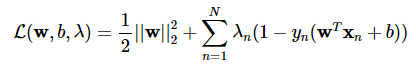
- Bài toán SVM ta sẽ đi tìm w, b sao cho hàm sau đạt giá trị lớn nhất, tức là việc phân chia rạch ròi nhất giữa hai lớp:



- Nếu ta thay vector hệ số w bởi kw và b bởi kb trong đó k là một hằng số dương thì mặt phân chia không thay đổi, tức khoảng cách từ từng điểm đến mặt phân chia không đổi, tức margin không đổi. Dựa trên tính chất này, ta có thể giả sử tử số= 1

- Nên ta có: tử số >=1 với mọi n

- Ta tìm được hàm Lagrange:



- Và đi tìm w, b để tìm giá trị cực trị cho hàm Lagrange này

- SVM Kernel method: Việc sử dụng trực tiếp SVM đôi khi là khó khăn với các feature ban đầu, kernel là phương pháp để chiếu dữ liệu đó sang một mặt phẳng khác để có thể dễ dàng tìm được đường phân lớp.

Module 4 - Clustering

**4.1 K-Means Clustering**

Cách thuật toán K-Means hoạt động:

- Xác định số cụm K bằng Elbow method, số cụm tăng dần, khi nào sai số ứng với số cụm hội tụ ta sẽ dừng vì bài toán cần cân đối giữa số lượng cụm cần phân và sai số khi phân cụm

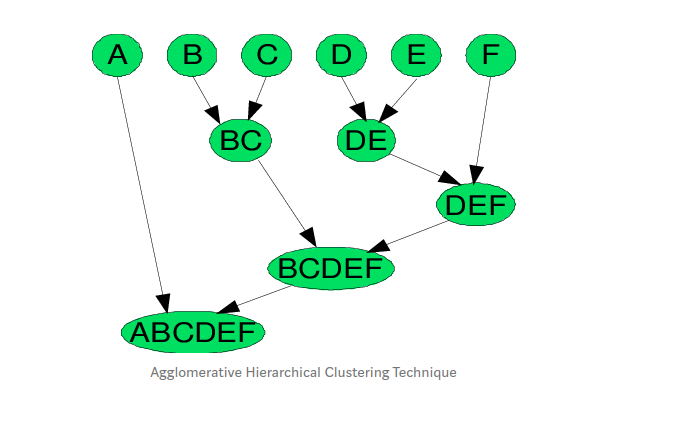
- Với số cụm là K, việc phân cụm sẽ tiến hành qua các bước.

- khởi tạo k điểm centroid ứng với K cụm

- Phân các datapoints vào k cụm theo tiêu chí datapoint gần với điểm nào nhất sẽ đưa về cụm đó

- Sau khi phân xong sẽ cập nhật lại centroid của cụm sao cho trung bình sai số từ centroid tối ưu tới tất cả các điểm là nhỏ nhất.

**4.2 Hierarchical Clustering**



- Agglomerative Hierarchical clustering là một thuật toán phân cụm cũng đánh giá sự tương đồng dựa trên khoảng cách giữa các cụm. Nhìn vào sơ đồ trên có thể thấy qua các step, các cụm gần nhau sẽ được gom lại với nhau cho tới khi dừng thuật toán.

- Divisive Hierarchical clustering: Divisive- phân tách từ 1 cụm duy nhất ban đầu (giả định) ra thành các cụm nhỏ khác nhau, phần này liên quan tới việc đánh giá mức độ tương đồng của các cụm, có rất nhiều phương pháp đánh giá, ví dụ chọn hai điểm ở gần với boundary của mỗi cụm rồi tính khoảng cách tới điểm tương tự ở cụm kia hoặc dùng điểm trung tâm (Nhưng sẽ gặp khó khăn với cụm ko có dạng elip và gặp khó khi noise cao…) https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec

**4.3 DBSCAN**

DBSCAN là thuật toán phân cụm dựa trên việc xác định các vùng thỏa mãn điều kiện về mức độ dày đặc và coi các vùng thưa thớt là vùng nhiễu để loại bỏ đi.

- Xác định điểm lõi, giống như centroid

- Tìm vùng cho điểm lõi

- Loại bỏ các vùng thưa

Hai parameter quan trọng của thuật toán là :

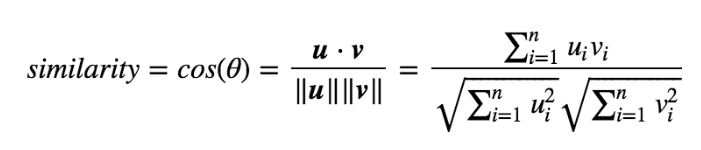
- eps: Ngưỡng để xác định hai điểm lân cận, cùng 1 cụm

- minpoints: số điểm tối thiểu được định nghĩa thành 1 vùng, Từ minpoints và éps ta sẽ xác định được Density (Số điểm thỏa mãn thực tế hai điều kiện trên để tạo thành 1 vùng)

Link chi tiết (<https://medium.com/@shritamkumarmund.98/how-dbscan-algorithm-works-2b5bef80fb3>)

Module 5 - Recommendation system

Điểm chung của hai thuật toán trình bày duois đây đều dựa trên mức độ tương đồng, nếu hai vector (đại diện cho hai đối tượng) có kích cỡ như nhau ta có thể dùng dot product để có thể tính toán sự tương đồng này. Hoặc có thể dùng theo phương pháp:

Hiểu rất đơn giản mức độ tương đồng giữa hai vector u,v được đo bằng góc giữa hai vector đó.

**5.1 Content-based Recommendation Systems**

Recomendation dựa trên mức độ tương đồng giữa các nội dung, sản phẩm mà một user đã sử dụng. Giả sử mình đã từng dùng một số sản phẩm có đặc tính được ghi nhận bởi các features của sản phẩm đó, khi đó sẽ gợi ý các sản phẩm có mức độ tương đồng với sản phẩm mà được dùng trước đó.

**5.2 Collaborative Filtering**

Đầu tiên thuật toán đánh giá mức độ tương đồng giữa hai đối tượng dựa trên các sản phẩm, nội dung mà hai đối tượng đã sử dụng. Sau đó nếu A và B có mức độ tương đồng cao thì sẽ gợi ý cho các sản phẩm của người này dùng mà người kia chưa dùng.

**Outcomes object of DSP303x**

|  |  |
| --- | --- |
| **IO** | **Outcomes** |
| DSP303x\_O1 | Outline the definition of Machine Learning |
| DSP303x\_O2 | Label the basics of Regression and its application |
| DSP303x\_O3 | Manipulate the Linear Regression on a dataset for estimation |
| DSP303x\_O4 | Interpret some advance Regressions algorithms:  - Multiple linear Regression.  - Non-linear Regression |
| DSP303x\_O5 | Define different Classification methods and its applications |
| DSP303x\_O6 | Manipulate K-Nearest Neighbours |
| DSP303x\_O7 | Comprehensively explore Logistic Regression:  - Definition.  - Training algorithm.  - Logistic Regression vs Linear Regression |
| DSP303x\_O8 | Interpret some advance Classifications algorithms: Decision Trees & Support Vector machine |
| DSP303x\_O9 | Manipulate K-means |
| DSP303x\_O10 | Comprehensively explore k-Means:  - Definition.  - Similarity vs Dissimilarity.  - Algorithm explanation.  - k-Means ' accuracy |
| DSP303x\_O11 | Interpret some advance Clustering algorithms:  - Hierarchical Clustering.  - DBSCAN - Density-based Clustering |
| DSP303x\_O12 | Classify different types of Recommendations System and its applications |
| DSP303x\_O13 | Interpret the most common Recommendation System algorithm:  Content- based Recommendation Engines |
| DSP303x\_O14 | Comprehensively explore Collaborative Filtering:  - User-based  - Item-based  - Similarity weights |
| DSP303x\_O15 | Get acquainted with the Python libraries for Machine Learning:  - Numpy  - Sklearn  - Pandas  - Matplotlib |
| DSP303x\_O16 | Clarify evaluation methods and metrics in different problems of machine learning (regression, classification, clustering) |
| DSP303x\_O17 | Outline the definition of Ensemble Learning and some typical Ensemble methods |