# [Slide](https://drive.google.com/drive/folders/1oJ0Q5v-0s10mjCKSTfPxkag7hnJEp9Uc)

# Chương 1:

## **Cài đặt môi trường và công cụ trên windows Tự học machine learning cơ bản**

* **Anagonda** hoặc cài hẳn **Python** (sử dụng pip để tải các gói thư viện)
* **Anagonda:** mở cmd gõ python > import sklearn
  + Có sẵn môi trường venv nhưng tách biết các package và hệ thống
* Tạo môi trường ảo (Python - còn lệnh tạo của agonda là khác) bằng lệnh:
  + *python -m venv venv*
  + *source venv/Scripts/activate*

*Thoát khỏi môi trường ảo khi cần*

*deactivate*

## **Cài đặt môi trường và công cụ trên linux ubuntu Tự học machine learning**

* Dùng trên linux (coi video)

## **1 jupyter notebook cài đặt và tại sao phải dùng nó**

* Tiện khi chạy từng dòng lệnh > ra kq luôn
* <https://jupyter.org/try-jupyter/lab/>
  + File > Notebook > Pyolite >Select (*shift + enter: để run dòng lệnh – phải chạy từng dòng để không lỗi)*
* Cài đặt Jupyter Notebook trong môi trường Python
  + Mở terminal trong VS Code hoặc cmd, chạy lệnh: *pip install jupyter*
* Hoặc nếu bạn dùng conda: *conda install jupyter*
* Kiểm tra đã cài chưa: *pip show jupyter*

**Sau khi cài đặt xong, bạn có thể chạy Jupyter Notebook độc lập bằng lệnh:**

*jupyter notebook* > http://localhost:8888/tree

## **2 Giới thiệu pytorch và pytorch tensor**

* <https://www.tensorflow.org/guide/tensor?hl=vi>
* <https://docs.pytorch.org/tutorials/beginner/basics/tensorqs_tutorial.html>
* <https://docs.pytorch.org/docs/stable/nn.init.html>
* PyTorch: *Để code đào tạo mạng Nơ-ron*, nổi bật với đồ thị tính toán động và hỗ trợ GPU mạnh mẽ. - *pip3 install torch torchvision*
* Tensor: *đầu vào là mảng nhiều chiều*, cấu trúc dữ liệu đa chiều, tương tự NumPy array, có thể chạy trên CPU hoặc GPU, dùng làm nền tảng cho các phép toán trong PyTorch.
* Trong pytorch chuyển hết thành tensor

## **Làm sao để sử dụng ChatGPT hiệu quả với khoá học**

Chat như người vs người

# Chương 2:

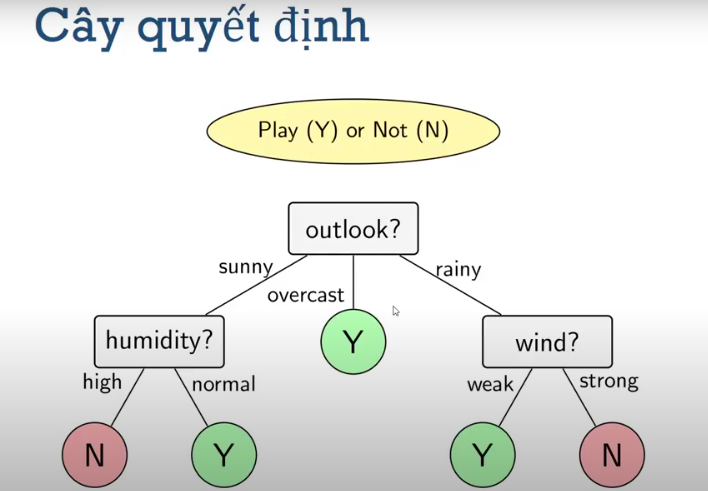
## **Bài 1 Hello world với machine learning Chương trình đầu tiên Tự học machine learning cơ ban**

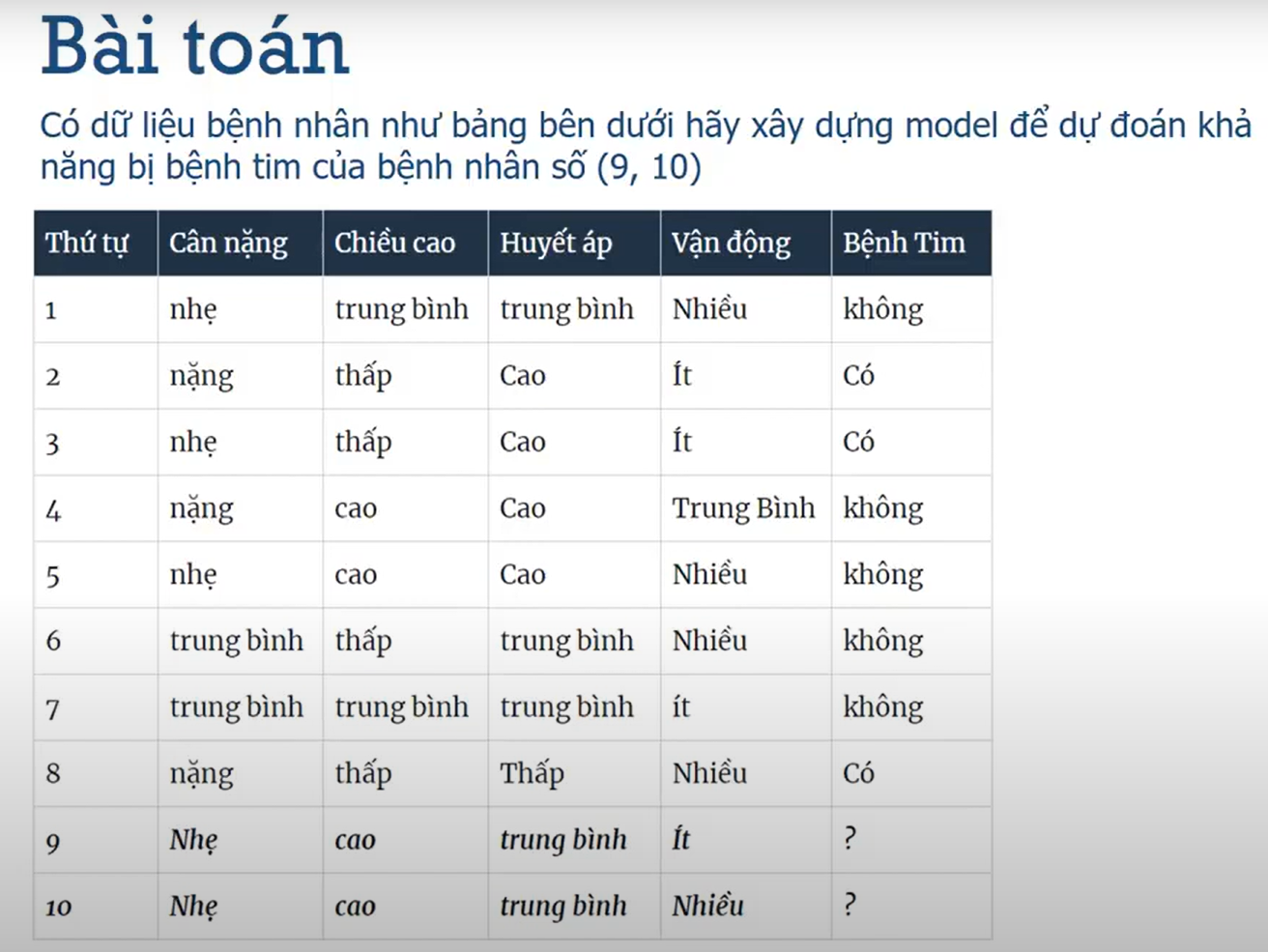
### What is Machine learning?

* Là một nhánh của trí tuệ nhân tạo
* Các thuật toán giúp máy tính có thể học hỏi từ dữ liệu để giải quyết vấn đề cụ thể

### Classification

* **Phân lớp**
* Vd: Khi bạn có một bức ảnh động vật bạn sẽ phân loại nó vào nhóm động vật có vú hay động vật không có vú





dactrung = [

    ['nhe', 'tb', 'tb', 'nhieu'],

    ['nang', 'thap', 'cao', 'it'],

    ['nhe', 'thap', 'cao', 'it'],

    ['nang', 'cao', 'cao', 'tb'],

    ['nhe', 'cao', 'cao', 'nhieu'],

    ['tb', 'thap', 'tb', 'nhieu'],

    ['tb', 'tb', 'tb', 'it'],

    ['nang', 'thap', 'thap', 'nhieu'],

['nhe', 'cao', 'tb', 'it'],

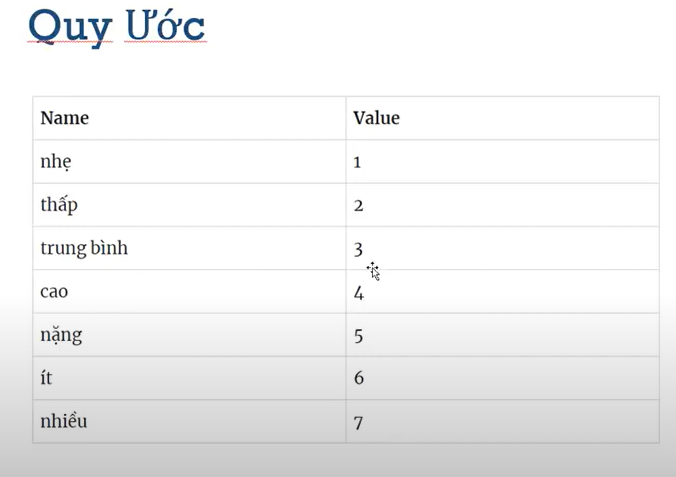
    ['nhe', 'cao', 'tb', 'nhieu'],

]

***(bôi vàng: mình sẽ phải dự đoán)***

“bệnh tym” -> nhãn(label) > Có/Không = 1/0

nhan = [0, 1, 1, 0, 0, 0, 0, 1]



**Chuyển đổi sau khi Quy ước:**

dactrung = [

    [1, 3, 3, 7],

    [5, 2, 4, 6],

    [1, 2, 4, 6],

    [5, 4, 4, 3],

    [1, 4, 4, 7],

    [3, 2, 3, 7],

    [3, 3, 3, 6],

    [5, 2, 2, 7]

]

## **Bài 2 Làm quen với numpy tại sao phải học numpy Tự học machine learning cơ bản.**

* numpy: thư viện tính toán trên mảng
* *pip3 install numpy*

## **Bài 3 Làm quen matplotlib tại sao phải học matplotlib Tự học machine learning cơ bản**

* matplotlib: hiển thị data theo dạng đồ thị, biểu đồ
* *pip3 install matplotlib*

## **Bài 4 Dataset và kiểm tra độ chính xác dự đoán Tự học machine learning cơ bản**

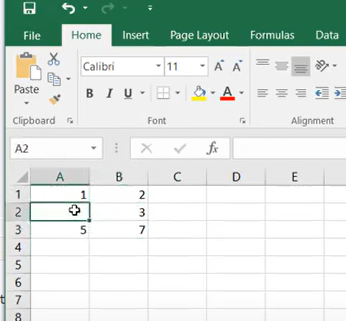
* Dataset: tập data được sưu tầm dưới dạng bảng và được gán nhãn cụ thể > có sẵn trong thư viện sklearn

## **Bài 5 pandas với csv dataset tự học machine learning cơ bản**

* <https://unsw-my.sharepoint.com/personal/z5025758_ad_unsw_edu_au/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fz5025758%5Fad%5Funsw%5Fedu%5Fau%2FDocuments%2FUNSW%2DNB15%20dataset%2FCSV%20Files%2FTraining%20and%20Testing%20Sets&ga=1>
* *pip install pandas:* để đọc file csv

## **Bài 6 Xử lý Missing values trong scikit learn Tự học machine learning cơ bản**

* Data bị thiếu



* Lấy trung bình cột A hoặc xem data nào nhiều thì điền data đó dô ô trống -> vd 1 nhiều thì điền 1
* <https://www.kaggle.com/datasets/ericanacletoribeiro/cicids2017-cleaned-and-preprocessed>
* *Vì dataset này ban đầu chưa có missing value nên tự tạo miss bằng thủ công trong code*

# Chương 3:

## **Bài 7 Linear Regression Hồi quy tuyến tính cơ bản Phần 1 Tự học machine learning cơ bản**

**Hồi quy tuyến tính** là một thuật toán **học có giám sát**, trong đó đầu ra **dự đoán là liên tục** và có **độ dốc không đổi**.

Nó được sử dụng để **dự đoán các giá trị trong một phạm vi liên tục**, (**ví dụ: doanh số, giá cả**) **thay vì cố gắng phân loại chúng thành các danh mục hay nhóm (ví dụ: mèo, chó).**

Có **hai loại** chính:

**Simple regression**

Simple regression giống như một phương trình đường thẳng mà bạn đã từng học ở phổ thông

**y = mx + b**

Trong đó m và b là các biến mà chúng ta cố gắng “học” để với mỗi giá trị input của x chúng ta có một giá trị dự đoán của y đúng nhất có thể.

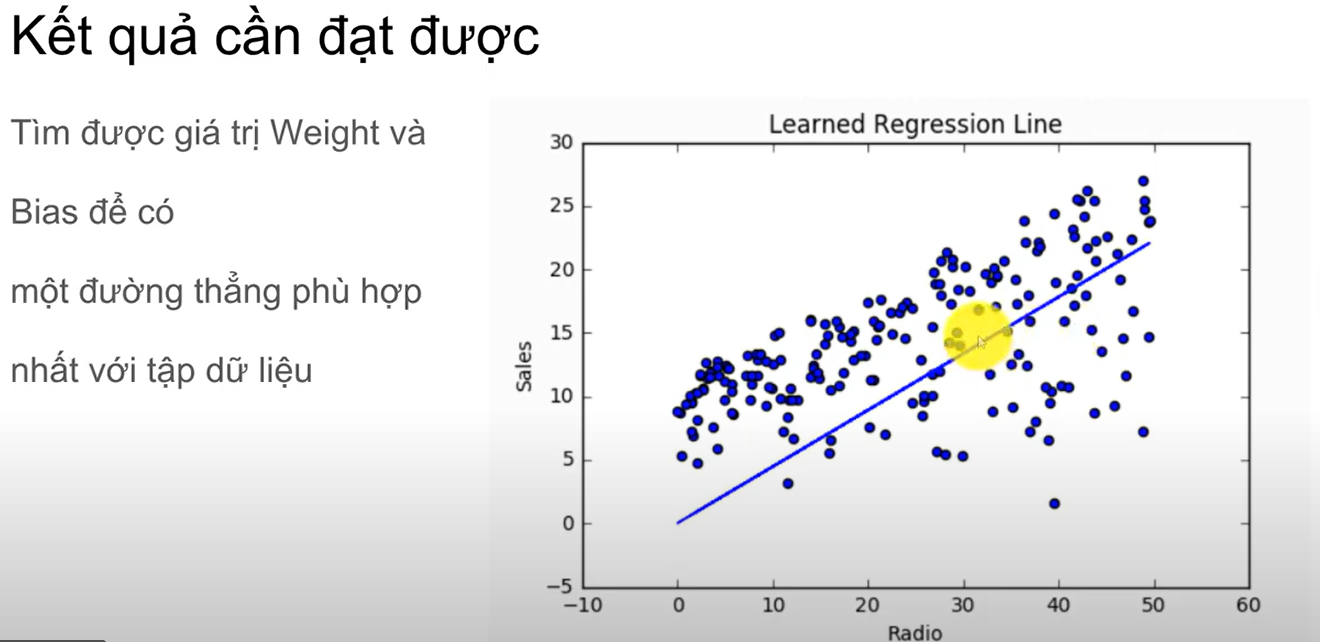
Tạo một dự đoán

Radio: Biến độc lập. Trong machine learning gọi là features (x – đã biết)

Weight: Hệ số của biến độc lập. Trong machine learning gọi là Weight (m)

Bias: giá trị lệch để bù đắp những sai số

**Sales (y - cái cần dự đoán) = Weight. Radio + Bias**



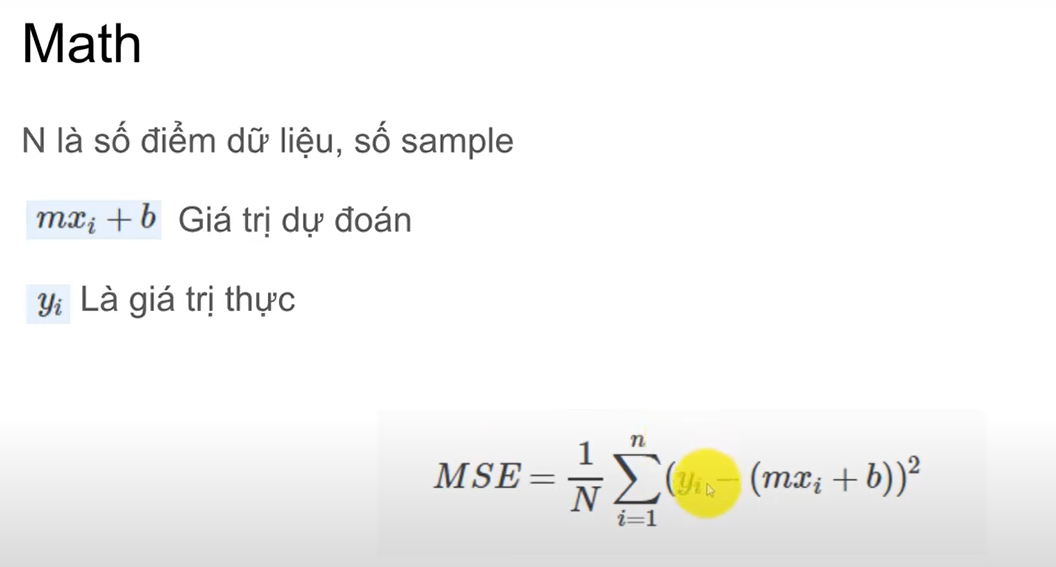
**Cost function**

Chúng ta cần 1 hàm chi phí để tối ưu weight

Hàm lỗi **MSE** (measures squared error) hàm này đo sự sai khác bằng cách lấy **trung bình của bình phương** **giữa** **giá trị dự đoán và giá trị thực tế.**

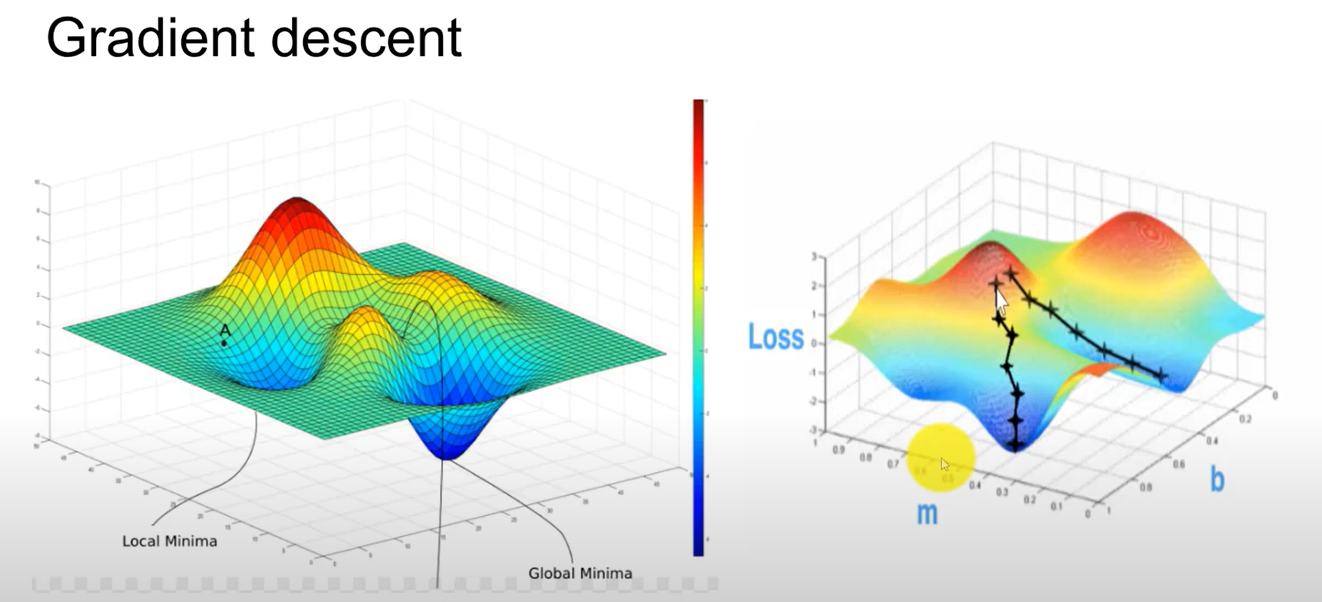
Đầu ra là một số duy nhất thể hiện chi phí của tập các trọng số (weight) hiện tại

Mục tiêu của chúng ta là tối thiểu MSE để tăng độ chính xác hiện tại

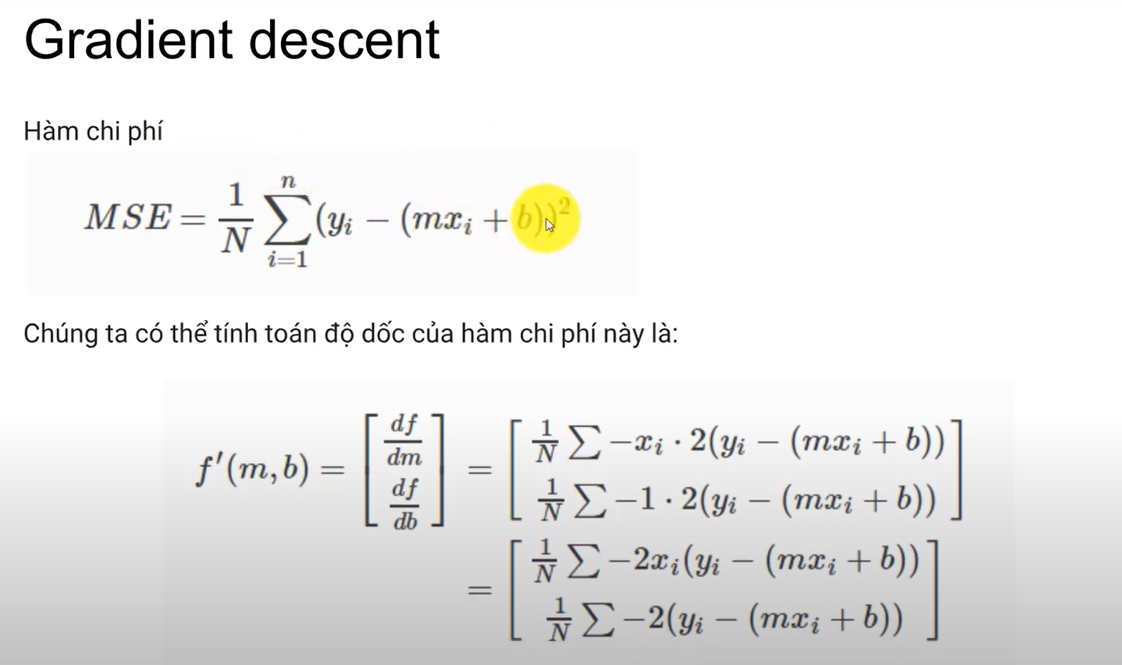


* Hàm để xác định chi phí để cải thiện chi phí đó (nhỏ nhất = tối ưu nhất)

Làm sao để tối ưu nó ?



Tìm điểm local (~ 0, còn global = 0 nhưng mà khó đạt được nếu được thì quá hoàn hảo rồi) bằng cách thả viên biên lăn vào chỗ trũng -> tối ưu



Tính đạo hàm “hàm chi phí”

**Gradient descent**

Để tối ưu hàm chi phí chúng ta cần tính lại weight và bias mới tối ưu hơn.

Chúng ta thực hiện việc tính weight và bias mới bằng cách

lấy weight và bias hiện tại - (tính weight và bias với mỗi điểm dữ liệu rồi lấy trung bình của tất cả các điểm dữ liệu sau đó nhân với learning rate)

learning rate là tốc độ học

**Training**

Đào tạo, training một model là quá trình lặp đi lặp lại cải thiện phương trình dự đoán của bạn bằng cách lặp qua tập dữ liệu nhiều lần.

Update lại weight và bias với **Gradient descent**

Việc training kết thúc khi đạt đến ngưỡng lỗi chấp nhận được hoặc những lần lặp tiếp theo không thể giảm được chi phí

**Multivariable regression**

## **Bài 8 Gradient descent trong hồi quy tuyến tính Linear Regression phần 2 Tự học machine learning**

## **Bài 9 Code chay Linear Regression bằng python Phần 1 Tự học machine learning cơ bản**

## **Bài 10 Code chay Linear Regression bằng python Phần 2 Tự học machine learning cơ bản**

## **Bài 11 Logistic Regression thuật toán hồi quy Logistic phần 1 Tự học machine learning cơ bản**

## **Bài 12 Đạo hàm của hàm sigmoid tính như thế nào Học machine learning cơ bản**

## **Bài 13 Code chay Logistic Regression phần 1 Tự học machine learning cơ bản**

## **Bài 14 Code cost function trong Logistic Regression phần 2 Tự học machine learning cơ bản**

## **Bài 15 code update weight trong Logistic Regression phần 3 tự học machine learning cơ bản**

## **Bài 16 bài tập về nhà và Code Hàm Training Logistic Regression phần 4 tự học machine learn**

## **Bài 17 Phương sai và độ lệch chuẩn trong machine learning Tự học machine learning cơ bản**

## **Bài 18 Overfitting và Underfitting Regularization và cross validation machine learning**

## **Thuật toán bayes Bayes**

## **Phân loại văn bản Bayes text classify**

## **Cây quyết định**