**Lesson 2: Regression**

## Khái niệm về Regression

Như đã nói ở buổi 1 , Regression là thuật toán Supervised của Machine Learning. Trong đó, chúng ta đã biết output, và chúng ta muốn dạy cho máy tính tự chạy ra được cái output đó :

Chúng ta có công thức của bài toán học có giám sát :

Y = f(x)

Với Y là output ( chúng ta đã biết )

x là dữ liệu đầu vào ( chúng ta đã có )

hàm f chính là thuật toán để dạy cho máy tính.

* Thuật toán Regression dùng để giải quyết các bài toán tìm y, mà y là một giá trị số thực dựa vào các dữ liệu đã có, ví dụ: dự đoán giá nhà dựa vào số phòng và diện tích, dự đoán huyết áp dựa vào lượng đường huyết và cân nặng , dự đoán giá xe dựa vào số năm đã sử dụng và số km đã đi, ...

Các bước cơ bản để giải quyết bài toán

* Xác định bài toán (output, benefit) \*Building model is not the end goal\*
* Lấy dữ liệu và Data preprocessing (xử lý missing value, …)
* Exploratory data analysis (EDA)
* Feature Scaling
* Split to train set and test set (train set 80%, test set 20%)
* Building and Training model
* Predicting the Test Results
* Evaluating the Model Performance

## Linear Regression

Linear Regression là một thuật toán Regression đơn giản dùng để dự đoán một giá trị y dựa trên giá trị x đã có.

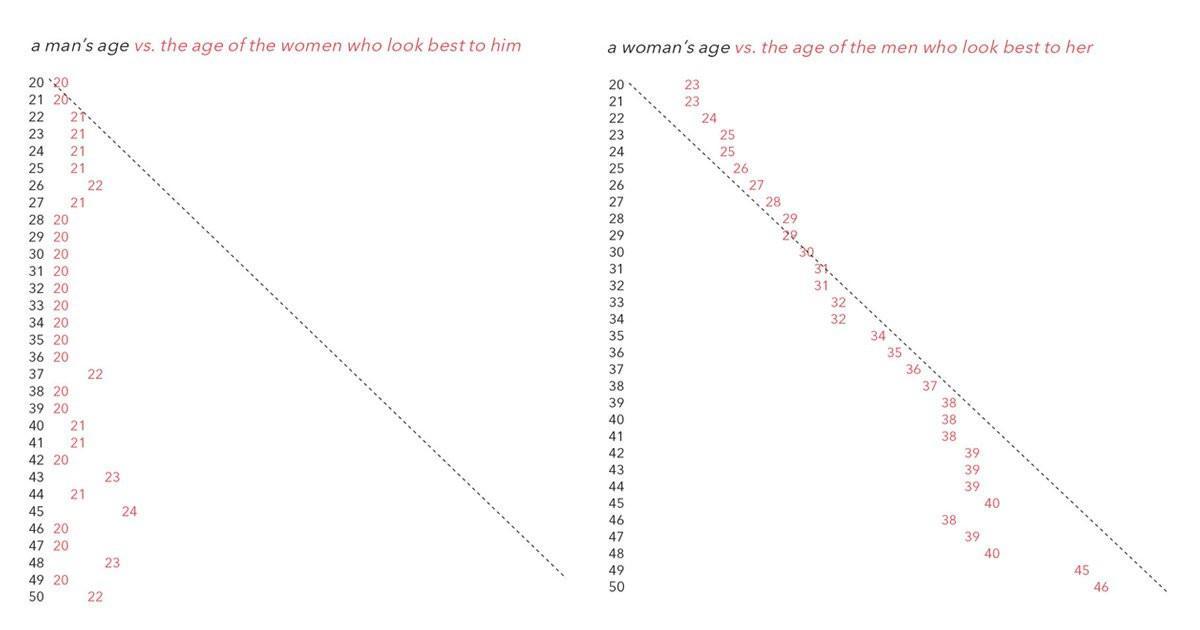
Linear Regression có dạng là một phương trình đường thẳng :

Y = ax + b ( với b là sai số , a là độ dốc , và x là giá trị đã biết , b là sai số )

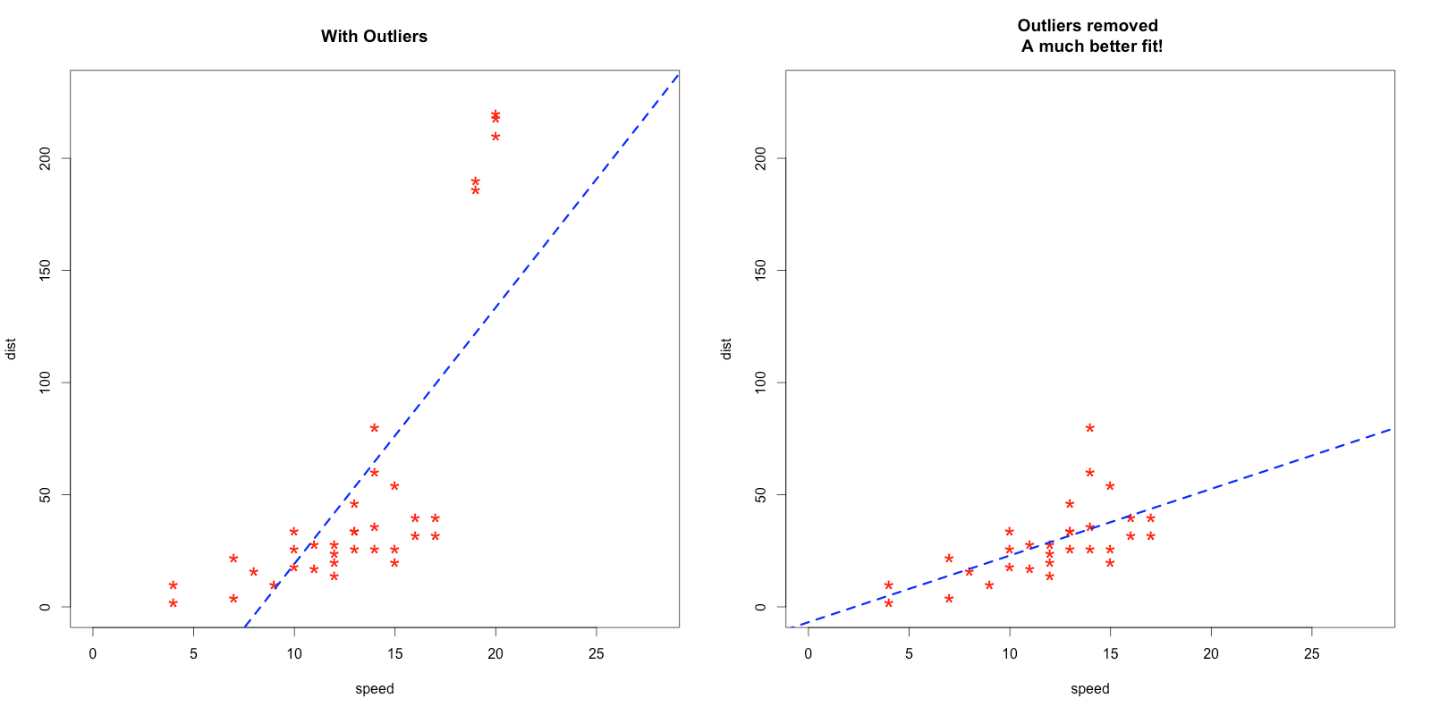
=> Ta có 1 tập dữ liệu gồm n điểm, ta giả định có 1 mối quan hệ tương đối tuyến tính (linear) giữa biến x và y, bài toán là cần tìm đường thẳng nào mà đi sát nhất với tát cả n điểm dữ liệu đó

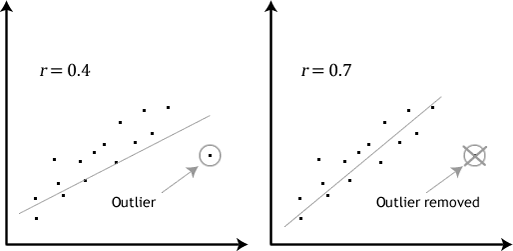
## Thực hành : Google Colab [Link](https://colab.research.google.com/drive/1lL-2IFEuJxh4RKf4QMjJPsQ8XoZBjwo5?usp=sharing)

Lưu ý : Correlation & Causation.



Outlier





DecisionSkills Articles

**Loss function:**

Sẽ giải thích kĩ hơn ở phần sau của bài giảng. Xem phần cuối tài liệu này.

## Multi Variate Linear Regression

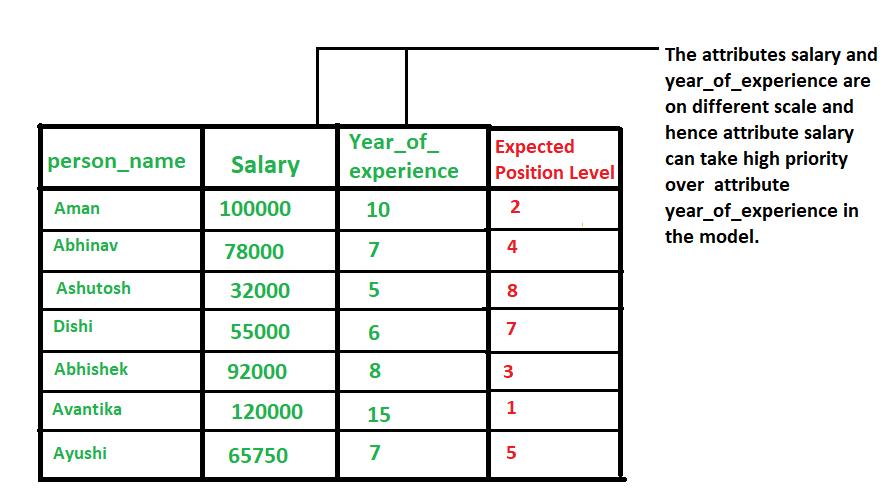
Tương tự như Linear Regression, Multi Variate Linear Regression là thuật toán Regression dùng để dự đoán một giá trị y dựa trên nhiều giá trị x, x1 , x2 đã có.

Ví dụ : Dự đoán giá nhà có x1 mét vuông , x2 phòng ngủ và khoảng cách đến trung tâm thành phố là x3.

Y = a1x1 + a2x2 + a3x3 + ... + anxn + b

Notes : Vì Multi Variate là có nhiều biến, nên đương nhiên hàm Y của chúng ta giờ đây sẽ có nhiều chiều , vì vậy không thể vẽ bằng 2D được nữa, nếu muốn vẽ tượng trưng bằng 2D, chúng ta phải tiến hành thu giảm chiều dữ liệu bằng những phương pháp như PCA, hay những phương pháp unsupervised khác.

Lưu ý : Data normalization & Standardization



The two most discussed scaling methods are **Normalization** and **Standardization**.

***Normalization***typically means rescales the values into a range of [0,1].

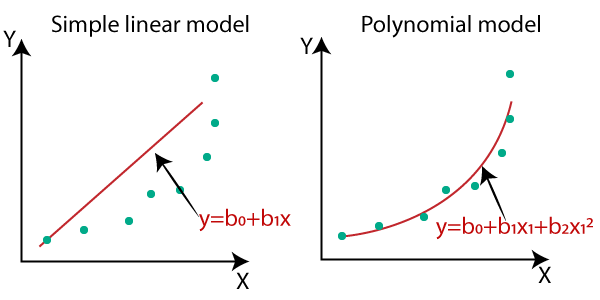
***Standardization***typically means rescales data to have a mean of 0 and a standard deviation of 1 (unit variance)

**Feature Scaling** : Đưa 2 biến có giá trị khác nhau về cùng 1 hệ quy chiếu . Ví dụ : Nhiệt độ (\*C) và USD. Có thể thấy 30\*C và 3,000,000 USD không cùng 1 hệ quy chiếu, nên nếu chúng ta vẽ, hoặc dự đoán mà không “scale” chúng về 1 hệ quy chiếu thì mô hình chúng ta sẽ có độ chính xác không cao.

## Non Linear Regression

Khái niệm :

Non Linear Regression là những thuật toán Regression dùng để dự đoán những bài toán mà Linear Regression không thể dự đoán được , hoặc dự đoán với kết quả chính xác rất thấp. Vì đồ thị hàm số y bây giờ không còn là một đường thẳng nữa ( Non Linear).



## Polynominal Regression

Khái niệm:

Polynominal Regression là Regression với đường thẳng bây giờ là ở dạng y = a\*x^n + b

## Thực hành : Google Colab [Link](https://colab.research.google.com/drive/1lL-2IFEuJxh4RKf4QMjJPsQ8XoZBjwo5?usp=sharing)

## Model Evaluation

Đánh giá : **Overfit** & **Underfit**

Diagram

Description automatically generated

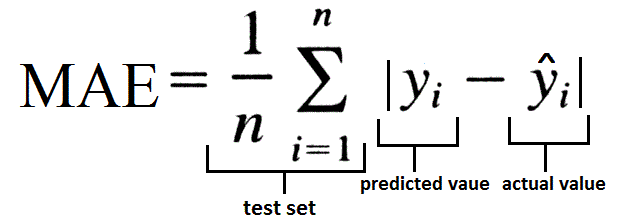
Nguồn : Machine Learning cơ bản , Vũ Hữu Tiệp

Mô hình tốt : Mô hình chạy và cho kết quả khá khớp ở cả 2 bộ dữ liệu huấn luyện & test

Mô hình quá khớp ( overfit) : Mô hình chạy và cho kết quả quá khớp ở bộ dữ liệu huấn luyện, nhưng lại không tốt ở bộ dữ liệu test.

Mô hình chưa khớp ( underfit): Mô hình chạy và cho kết quả không khớp ở cả 2 bộ dữ liệu train & test.

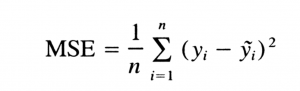
Mean Absolute Error



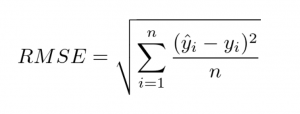
là trung bình của giá trị tuyệt đối các sai số, dùng để đo lường sự khác biệt giữa hai biến dự đoán và biến thật.

Mean Square Error

là trung bình của bình phương các sai số, tức là sự khác biệt giữa các ước lượng và những gì được đánh giá , nó càng dần về 0 thì càng có đủ độ tin cậy chứng tỏ mô hình ít bị sai số nhất



Ngoài ra còn có RMSE là căn bậc 2 của MSE



## Model Improvement

Cách cải thiện Mô hình :

* 1. **Feature Scaling**

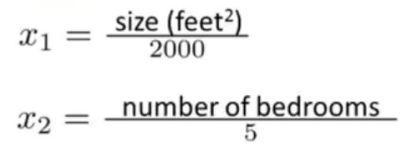
feature scaling giúp cho giá trị của các variables được đồng dạng với nhau hơn.

Ví dụ :

X1 : diện tích ( 0 – 200 m vuông )

X2 : số phòng ( 1 – 5 )

Như vậy khoảng cách của x1 là từ 0 đến tận 200 trong khi x2 là chỉ từ 1 đến 5. Điều này làm cho việc chạy Gradient Descent lẫn performance của model rất khó khăn. Để có được 2 feature tốt hơn. Ta lấy các giá trị feature chia cho số có giá trị tuyệt đối lớn nhất.



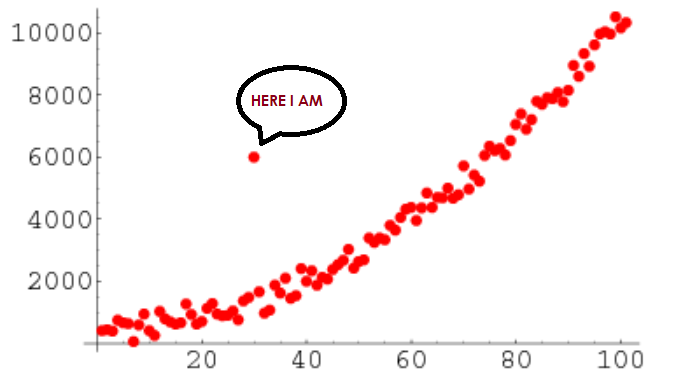
Như vậy ta luôn thu gọn được các giá trị của feature nằm trong khoảng từ -1 đến 1.

Diagram

Description automatically generated

* 1. **Loại bỏ outlier**

Loại bỏ những giá trị quá khác biệt sẽ giúp cho model hoạt động tốt hơn.

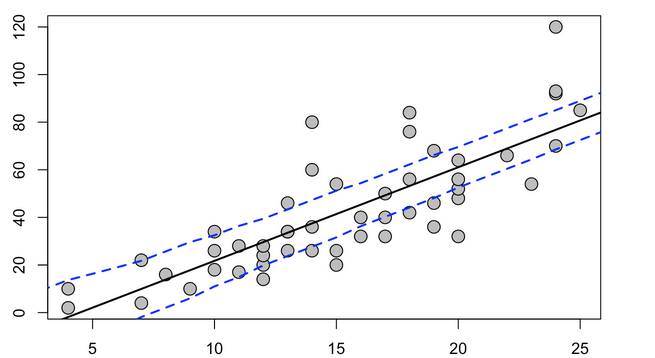


* 1. **Hàm mất mát (Loss fuction)**

Hàm mất mát ( loss function) và Gradient Descent

Loss function là một hàm số thể hiện sự chênh lệch giữa đại lượng 2 đại lượng y , y được dự đoán (tiên anh ký hiệu là ŷ, gọi là “y hat”) và y đúng. Trong mọi bài toán supervised learning, mục tiêu của ta luôn bao gồm giảm thiểu sự chênh lệch giữa y dự đoán và y đúng. Trong trường hợp lý tưởng y= ŷ, loss function sẽ trả về giá trị cực tiểu bằng 0

* Cần biết loss function để tìm ra hàm số fit tốt nhất các features.



Có 2 dạng bài supervised learning cơ bản :

Khi y là một số thực dao động trong khoảng (−∞,∞), ta được một bài toán **regression**. Ví dụ như ta cần tiên đoán giá nhà , giá cổ phiếu, giá xăng, giá vàng.

Khi yyy là một đại lượng rời rạc chỉ nhận giá trị trong một tập label hữu hạn rời rạc nào đó, ta được bài toán **classification**. Ví dụ, khi ta cần nhận dạng tên một người từ một tấm ảnh chân dung, cho dù có hàng tỉ cái tên trên thế giới thì tập hợp này vẫn là rời rạc hữu hạn. Vì thế, bài toán này vẫn được quy vào dạng classification. Ví dụ bài toán dự đoán kết quả ("có" hoặc "không", "đúng" hoặc "sai", "positive" hoặc "negative", ...). ta chuyển đổi tập label thành tập {−1,+1}\{-1, +1\}{−1,+1} (-1 nghĩa là "không", +1 nghĩa là "có").

Vì loss function đo đạc chênh lệch giữa *y actual*  và y predicted ( yhat)

*L*(*y*^​,*y*)=*y*^​ − *y*

Tuy nhiên hàm này lại không thỏa mãn tính chất không âm của một loss function. Ta có thể sửa nó lại một chút để thỏa mãn tính chất này. Ví dụ như lấy giá trị tuyệt đối của hiệu:

*L*(*y*^​,*y*)=∣*y*^​−*y*∣

Loss function này không âm nhưng lại không thuận tiện trong việc cực tiểu hóa, bởi vì đạo hàm của nó không liên tục (nhớ là đạo hàm của f(x)=∣x∣ bị đứt quãng tại x=0) ( cần phải đạo hàm để tìm cực tiểu) và thường các phương pháp cực tiểu hóa hàm số thông dụng đòi hỏi phải tính được đạo hàm. Một cách khác đó là lấy bình phương của hiệu:

*L*(*y*^​,*y*)=1/2​(*y*^​−*y*)2

Khi tính đạo hàm theo y^, ta được ∇L = 1/2×2×(y^−y)=y^−y

Có đồ thị đường thẳng : y = ax + b / hoặc ℎ𝜃 𝐱 : 𝑦 = 𝜃0 + 𝜃1𝑥1

*L*(𝜃0​,𝜃1)=1/2​(*y*^​−*y*)2

* Tìm 𝜃0 , 𝜃1 = min *L*(𝜃0​,𝜃1)

*L*(*y*^​,*y*)=1/2​(*y*^​−*y*)2 . Mà ½ là hằng số nên người ta sẽ chú ý vào (y^ - y)^2 và còn hay được gọi là square loss.

Chart, scatter chart

Description automatically generated

* Gradient Descent là sự biến thiên ( đạo hàm ) của các điểm trên parabol y^ - y.

Thuật toán Gradient Descent dùng để tìm ra điểm mà hàm số đạt [cực tiểu](https://vi.wikipedia.org/wiki/C%E1%BB%B1c_tr%E1%BB%8B_c%E1%BB%A7a_h%C3%A0m_s%E1%BB%91)

Xem thêm Loss Function : [Link](https://www.youtube.com/watch?v=yt7fzvwfWHs)

**Warm up** : Một nhà leo núi tìm đường xuống thung lũng

Chart, line chart

Description automatically generated

**Best case :**

Chart, line chart

Description automatically generated

**Reality :**

A picture containing necklet

Description automatically generated

## Thực hành : Google Colab [Link](https://colab.research.google.com/drive/1lL-2IFEuJxh4RKf4QMjJPsQ8XoZBjwo5?usp=sharing)