MLP 302X - ML Regression

Mục tiêu khóa học:

- Hồi quy tuyến tính đơn giản

- Bias Variance tradeoff

- Linear regression with ban (Lasso vs Ridge regression)

- Evaluate, dataset. (Phương pháp đánh giá được customize theo bài toán như nào: Định nghĩa lại trọng số cho từng cụm giá trị)

# Regression Overview

## 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:

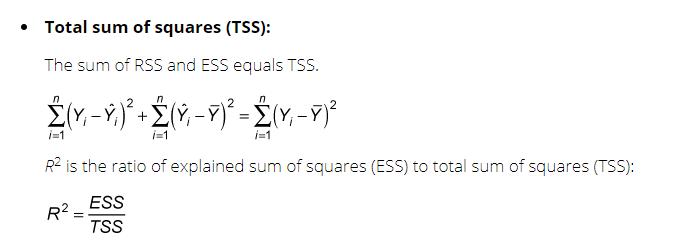
* Analytical solution (normal equation)
* Optimization solution (gradient descent)

# Regression Algorithm

## 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.

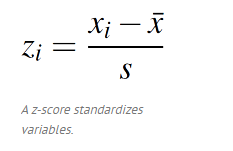
* Hàm cập nhật weight.

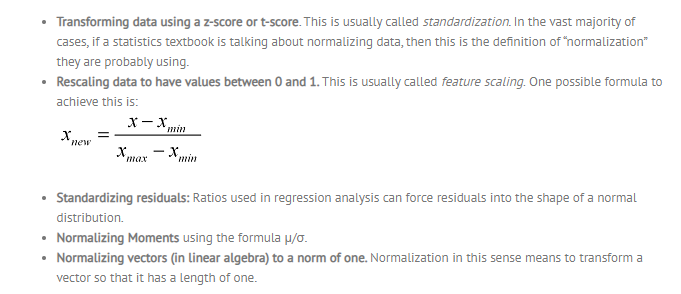
Ta cập nhật weight theo từng giá trị đạo hàm mà ta tìm được ở bước trên.

* Single/multivariable gradient descent and updated weights.

Ta làm tương tự với hàm Linear regression.

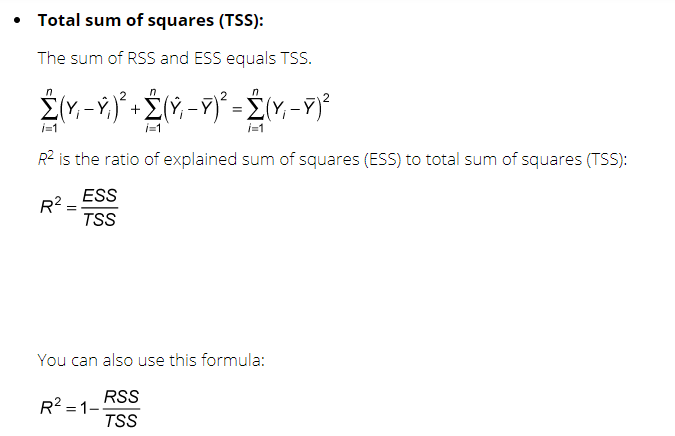
* Normalization: Min/max scaler, normalized function trong sklearn.





## Performance assessment metric

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



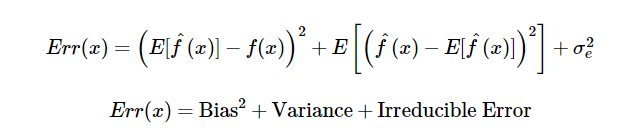
## Overfitting

* 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.

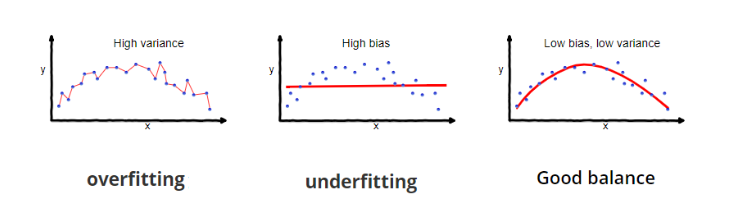
* Variance

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>

## 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

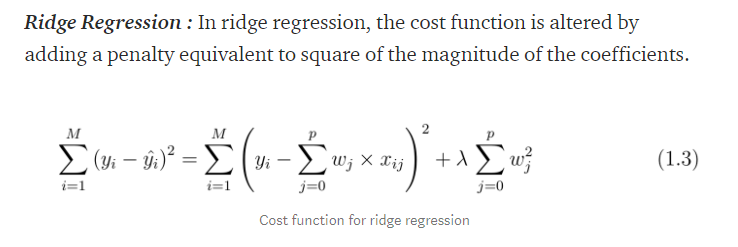
ví dụ: hệ số là 0,198. Cứ tăng 1% trong biến độc lập, biến phụ thuộc tăng 0,198%

# Regularization

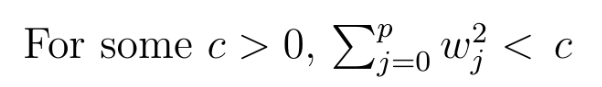
Đối với Regression ta có thể giải thích theo hướng Probabilistic model: <https://stats.stackexchange.com/questions/367485/map-estimation-as-regularisation-of-mle>

## L2 regularization

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



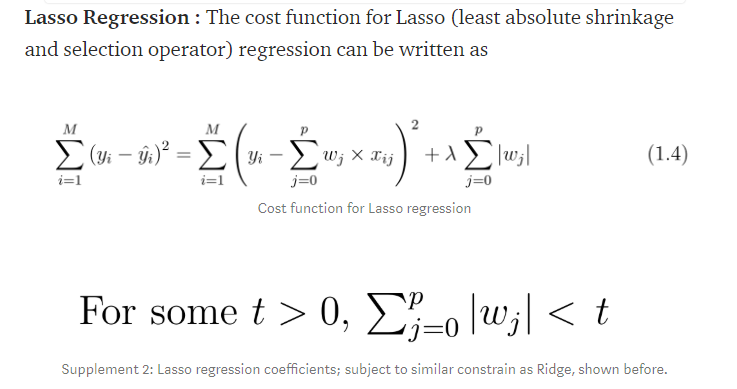
Nó chính là hàm Lagrange của Loss function với constraint cho các hệ số nhằm làm giảm sự biến động mạnh, không ổn định trong mô hình. Giá trị của lamda càng gần 0 thì hệ số thu được sẽ càng gần với Linear Model:



* Mục đích sử dụng L2 pelnaty: Reduce overfitting.

## L1 regularization

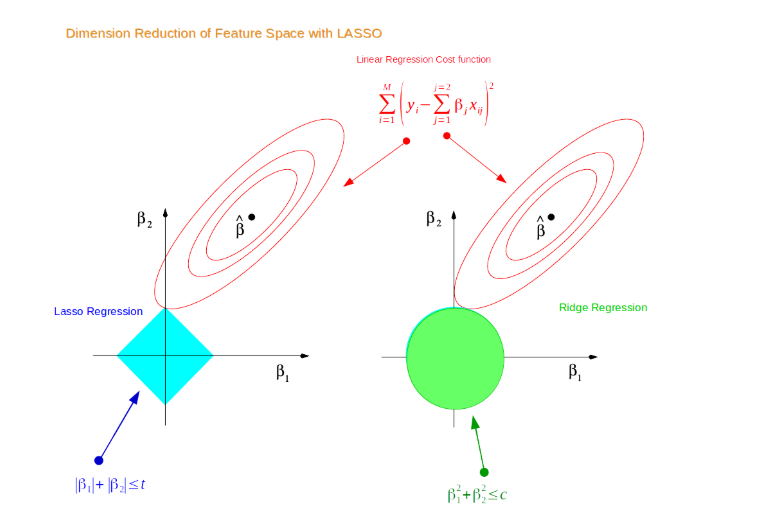
* L1 pelnaty: Công thức, hàm loss và đạo hàm của hàm loss.
* Lasso regression.



* 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:



Ta có nhiều cách để giải thích tại sao Lasso lại giúp ta Feature selection còn Ridge thì không. Ví dụ với: Y=Bx+e

  
Với Beta>0, ta lấy đạo hàm được:



# 

# Khi ta tăng lamda lên thì B có thể tiến về 0, làm tương tự có thể chứng minh được điều này không khả thi với Ridge

# Regression on sklearn

* Gọi model: Các model khác nhau. Parameters, attribute và method của từng model.
* Sử dụng model: Training/validation/prediction.
* Finetuning: Parameters tuning, CV tuning, feature selection tuning.
* Lưu model.

# KNN Regression

## KNN basic

* Instance-base learning or lazy learning: Khái niệm và tính chất.
* Distance function.
* KNN for classification.

## K-NN advance

* Kernel Regression.

https://towardsdatascience.com/kernel-regression-made-easy-to-understand-86caf2d2b844

<https://towardsdatascience.com/regression-kernel-and-nearest-neighbor-approach-6e27e5e955e7>

* Weighted average.
* How to choose lambda.
* Gaussian Kernel.
* Gaussian Variance.
* k-NN with big data: High dimensions, data scarcity, computational complexity.

Ta đi tính weight theo density cho từng điểm dữ liệu sau đó dựa vào weight và actual value ta đi tính cho giá trị cần dự báo.