**MLP 303x: Classification**

1. **Giới thiệu về Classification**

* Hiểu về bài toán Classification
* Hiểu về Supervised learning
* Một số thuật toán dùng trong Classification
* Sự khác nhau giữa Classification và Regression

1. **Các thuật toán cơ bản trong Classification**
   1. **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:

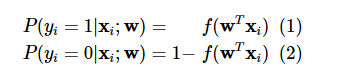
- Là hàm liên tục, có giá trị thực và trong khoảng 0-1

- 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

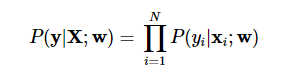
Xác suất để y nhận giá trị tương ứng 1, 0 khi ta biết Xi và tham số w.



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 lên, ta thu được:



Ta đi tìm max của hàm này, đồng nghĩa với việc tìm min của hàm log và coi đây là hàm mất mát của mô hình.

* Sử dụng Sklearn để thực hiện fit model
* Phát hiện và xử lý Overfitting trong Classification
* Overconfident predictions due to overfitting
* L2 Regularized cho Logistic Regression
* Sparse logistic regression
  1. **SVM**
* Giới thiệu về Large Margin Classification

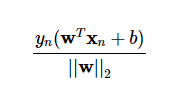
SVM – Support vector machine là thuật toán đi tìm một đường thẳng/ mặt phẳng phân chia hai class trong không gian sao cho đáp ứng tiêu chí khoảng cách giữa hai điểm gần đường phân chia nhất của hai lớp là như nhau và lớn nhất, khoảng cách này được gọi là Margin- biên, nên đồng thời SVM cũng có tên gọi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

* Giới thiệu về Kernels
* SVM và cách mà model được huấn luyện

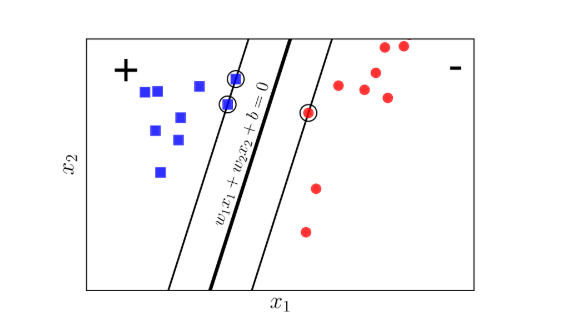
Đầu tiên ta giả sử có một đường thẳng phân chia hai lớp như sau:

WTx+b = w1x1+w2x2+b=0

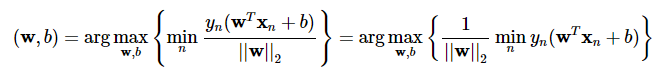
Với một cặp (xn,yn) bất kỳ, ta có thể tính khoảng cách từ điểm đó tới đường thẳng phân chia là:



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



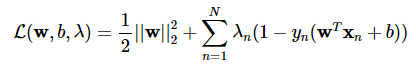
Bài toán SVM ta sẽ đi tìm w, b sao cho hàm sau đạt giá trị lớn nhất, tức là việc phân chia rạch ròi nhất giữa hai lớp:



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.

* Sử dụng Sklearn để thực hành với SVM
  1. **Naïve Bayes**
* Hiểu về lý thuyết Bayes, cách sử dụng naïve Bayes classifiers

Ta xuất phát từ bài toán có K lớp cần phân loại vào, khi đó ta có một record xi, ta muốn biết xác suất để record đó rơi vào lớp k- xác suất có điều kiện P(yi=k|xi) từ đó ta có thể tìm được class mà điểm dữ liệu có khả năng xảy ra cao nhất và phân vào lớp đó.

Và xác suất có điều kiện trên khó tính trực tiếp nên ta sử dụng Bayes để có thể xử lý:

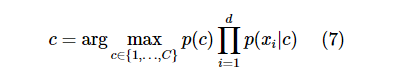
P(yi=k|xi)= P(x|k)\*P(k)/P(x)

P(x) thì không phụ thuộc vào c nên ta có thể coi là hằng số (Evidence)

P(c) là xác suất của class c trong tất cả class, ta chỉ việc count số lần xuất hiện của c trong training data rồi chia cho tổng số bản ghi là tìm ra

P(x|k) – Phương pháp này giả định các chiều x là độc lập với nhau nên P(x|k)= P(x1, x2…|k)= tích các xác suất của các x với điều kiện k. Vì giả định này rất chặt nên còn được gọi với cái tên Naïve.

Ta tìm ra được lớp c của một bản ghi dựa trên:



Ứng với mỗi phân phối của Xi ta sẽ có cách tính xác suất khác nhau, một số phân phối thường gặp như: Gaussian, Multinormial, Bernoulli.

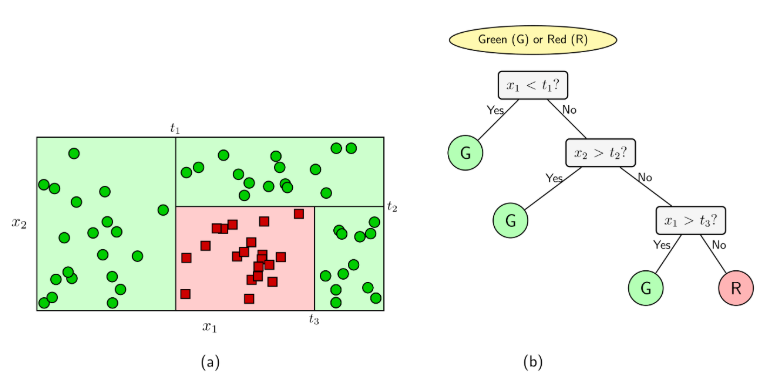
* Generative model
* Sử dụng Naïve Bayes cho phân loại text
* Cách Naïve Bayes được huấn luyện, thực hành với Sklearn
  1. **Neural network**
* Giới thiệu về Neural network, cách áp dụng neural network
* Các activation function là một trong những nội dung quan trọng nhất của NN, Các activation function đóng vai trò là nhận các giá trị từ các node phía trước, sau đó sẽ đưa ra quyết định trên các giá trị này (Đánh giá thông tin đầu vào) (<https://blog.vietanhdev.com/posts/2019-09-23-cac-ham-kich-hoat-activation-function-trong-neural-networks/>)
* Cost Function và cách thuật toán Backpropagation hoạt động

(<https://towardsdatascience.com/how-does-back-propagation-in-artificial-neural-networks-work-c7cad873ea7>)

* Gradient descent trong Neural network
* Feed Forward Neural network

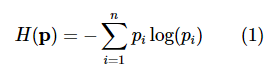
(<https://nttuan8.com/bai-3-neural-network/>)

* Sử dụng Sklearn để implement, training và predict với FFN (2 Layers)- Đối với yêu cầu bắt buộc của khóa học thì em phải nắm được ý nghĩa của những Parameter căn bản nhất (<https://towardsdatascience.com/neural-networks-from-scratch-easy-vs-hard-b26ddc2e89c7>) (<https://towardsdatascience.com/neural-networks-parameters-hyperparameters-and-optimization-strategies-3f0842fac0a5>)
  1. **Decision tree**
* 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.

* Decision tree với biến liên tục
* Sử dụng Sklearn để implement, train, predict với Decision tree
* Overfitting trong Decision tree, sử dụng Early stopping để loại bỏ Overfitting
* Prunning Decision Trees

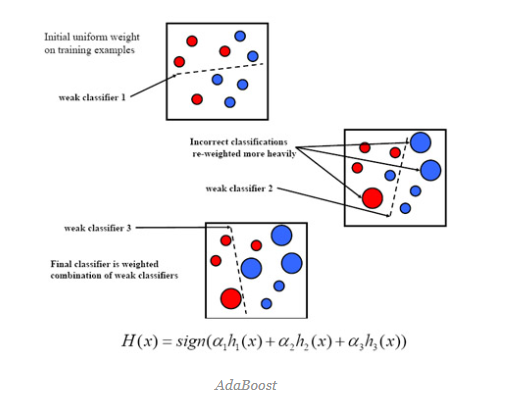
(<https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb>)

https://blog.vietnamlab.vn/decistion-tree/

* 1. **Ada-boost**
* Thuật toán học tăng cường Boosting trong các bài toán phân loại - Ada-boost

Boosting là thuật toán học từ dữ liệu bằng cách xây dựng nhiều thuật toán học cùng lúc (ví dụ như cây quyết định) và kết hợp chúng lại. Mục đích là để có một cụm hoặc một nhóm các weak learner sau đó kết hợp chúng lại để tạo ra một strong learner duy nhất.

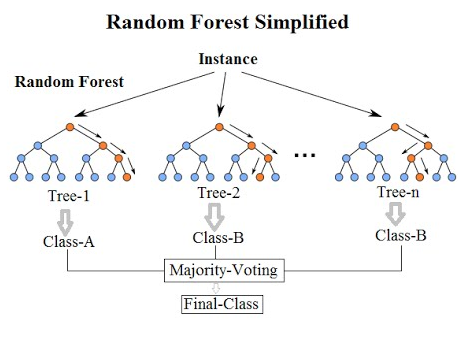
* Cách áp dụng Adaboost



Adaboost là thuật toán được xây dựng bằng cách kết hợp nhiều thuật toán cùng một lúc, phối hợp với nhau để cho ra kết quả tốt nhất:

Sau mỗi phiên thì sẽ lấy ra các learner với độ chính xác cao nhất, những lớp bị phân sai sẽ được đánh trọng số cao lên để thực hiện ở phiên sau H(x) là trọng số.

* Sự hội tụ và overfitting trong boosting
* Implement, train, predict Gradient Boosting với Sklearn
  1. **Random Forest**
* Giới thiệu về Random Forest



Tree-based: Sử dụng rất nhiều các cây quyết định sau đó dùng cơ chế voting để có thể chọn ra kết quả cuối cùng.

* Implement, train, predict Random Forest với Sklearn

1. **Evaluation, Handling Huge Dataset, Machine Learning System Design**
   1. **Evaluation**

* Precision, recall, F1 Score, AUCROC, Confusion metrics
* Precision – recall tradeoff
  1. **Huge dataset**
* Scale ML to huge datasets
* Scale ML with stochastic gradient, cách mà Stochastic gradient thực thi
* Fitting models from streaming data
  1. **Machine Learning System**
* Error analysis
* Error metrics for Skewed Classes
* Tradeoff Precision - recall

|  |  |
| --- | --- |
| **IO** | **Outcomes** |
| MLP303x\_O1 | Know what classification problems are in Machine Learning |
| MLP303x\_O2 | List real word classification problem examples and applications |
| MLP303x\_O3 | List some common classification algorithms |
| MLP303x\_O4 | Have intuition about Linear Classifiers & Logistic Regression model and know how the model is learned using gradient ascent |
| MLP303x\_O5 | Implement, train, predict for Logistic Regression using Scikit-learn |
| MLP303x\_O6 | Implement Logistic Regression from scratch using Python, Numpy |
| MLP303x\_O7 | Practice Logistic Regression with real word dataset |
| MLP303x\_O8 | Have knowledge to handle overfitting, regularization in Logistic Regression |
| MLP303x\_O9 | Practice about overfitting, regularization in Logistic Regression |
| MLP303x\_O10 | Have intuition about Decision Tree and know how the model is learned |
| MLP303x\_O11 | Implement, train, predict for Decision Tree using Scikit-learn |
| MLP303x\_O12 | Implement Binary Decision Tree from scratch using Python, Numpy |
| MLP303x\_O13 | Practice Decision Tree with real word dataset |
| MLP303x\_O14 | Have knowledge about methods for preventing overfitting in Decision Tree |
| MLP303x\_O15 | Have intuition about Boosting, Adaboost and how the model is learned |
| MLP303x\_O16 | Implement, train, predict for GradientBoost using scikit-learn |
| MLP303x\_O17 | Implement Boosting Python, Numpy |
| MLP303x\_O18 | Practice Boosting with real word dataset |
| MLP303x\_O19 | Have knowledge to prevent overfitting in Boosting |
| MLP303x\_O20 | Have intuition about SVM and how the model is learned |
| MLP303x\_O21 | Have intuition and knowledge about SVM kernel method |
| MLP303x\_O22 | Implement, train, predict for SVM using Scikit-learn |
| MLP303x\_O23 | Practice SVM with real word dataset |
| MLP303x\_O24 | Have intuition about Naive Bayes and know how the model is trained |
| MLP303x\_O25 | Implement, train, predict for Naive Bayes using Scikit-learn |
| MLP303x\_O26 | Practice Naive Bayes with real word dataset |
| MLP303x\_O27 | Have intuition and knowledge about Random Forest and how the model is trained |
| MLP303x\_O28 | Implement, train, predict for Random Forest using scikit-learn |
| MLP303x\_O29 | Practice Random Forest with real word dataset |
| MLP303x\_O30 | Have intuition about Feed Forward Neural Network |
| MLP303x\_O31 | Know how Gradient descent is used when training Neural Network |
| MLP303x\_O32 | Know how the Backpropagation algorithm works |
| MLP303x\_O33 | Implement, train, predict for Feed Forward Neural Network using scikit-learn |
| MLP303x\_O34 | Implement 2 layers Neural Network from scratch |
| MLP303x\_O35 | Practice Feed Forward Neural Network with real word dataset |
| MLP303x\_O36 | Have knowledge about precision and recall metric |
| MLP303x\_O37 | Have knowledge about trade-off between precision and recall |
| MLP303x\_O38 | Practice precision and recall with a Logistic Regression classifier |
| MLP303x\_O39 | Know who the stochastic gradient descent (ascent) algorithm works |
| MLP303x\_O40 | Practice stochastic gradient descent with a Logistic Regression classifier |
| MLP303x\_O41 | Practice all classification algorithms for a real world problem |