Personal Note - Week 05: Support Vector Machine & Naive Bayes

A. Support Vector Machine

1. **Tổng quan**
    - SVM là thuật toán được dùng nhiều trong ML (cả về **phân lớp lẫn hồi quy SVR)**
    - Mục tiêu:
        - Tìm ra một mặt **siêu phẳng** trong không gian N chiều (ứng với N đặc trưng), chia dữ liệu thành hai phần tương ứng với lớp của chúng.
        - Tìm ra siêu phẳng có **margin (lề)** rộng nhất (tức là khoảng cách tới các điểm của hai lớp là lớn nhất).
    - Số chiều trong siêu phẳng phụ thuộc vào số đặc trưng.
2. **Vector hỗ trợ**
    - **Trong không gian vector,** một điểm được coi là **một vector** từ gốc toạ độ tới điểm đó
    - Các điểm dữ liệu **nằm trên** hoặc **gần nhất** với siêu phẳng được gọi là **support vectors**
    - Các điểm này được sử dụng để **tối ưu hoá margin.** Nếu xoá các điểm này, vị trí của siêu phẳng sẽ thay đổi.
    - **Support Vectors** phải **cách đều** siêu phẳng
3. **Dữ liệu không thể phân chia tuyến tính**

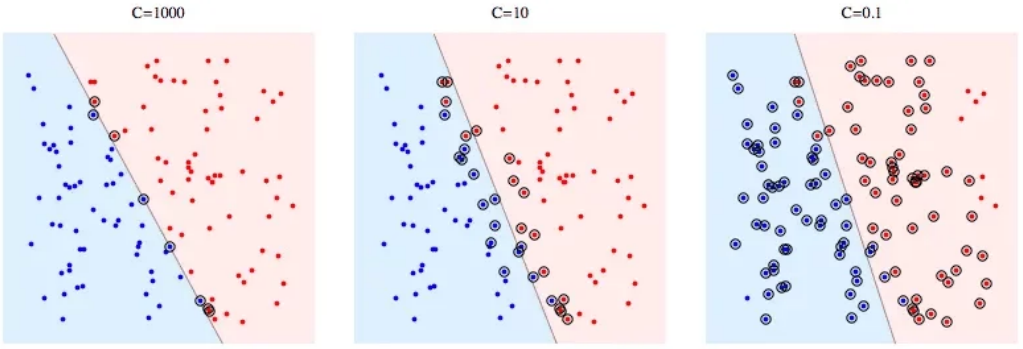
- Đối với các dữ liệu không thể phân chia tuyến tình, ta không thể thực hiện vẽ một siêu phẳng để phân tách các điểm dữ liệu.
- Khi này, ta có hai giải pháp là **Soft Margin** và **Kernel tricks**

4. **Soft Margin**

- Là thuật toán cho phép SVM **mắc một số lỗi nhất định** và **giữ cho margin càng rộng càng tốt** để các điểm khác vẫn có thể được **phân loại chính xác**.

→ Cân bằng giữa **phân loại sai** và **tối đa hoá lề**

- Có hai kiểu phân loại có thể xảy ra:
  - Dữ liệu nằm ở đúng bên nhưng phạm vào lề
  - Dữ liệu nằm ở sai bên
- Điều chỉnh **Soft margin** thông qua **mức độ chấp nhận lỗi (C)**
  - Là một **siêu tham số** trong SVM
  - Nó được xem như một tham số phạt trong mô hình
  - **C càng lớn** → SVM càng bị phạt năng khi thực hiện phân loại sai. Do đó, **margin** càng hẹp và càng ít support vector được sử dụng.
    - Mô hình sẽ cố gắng giảm thiểu lỗi phân loại.
    - Tuy nhiên, dễ gặp tình trạng **overfitting**
  - **C nhỏ**: Mô hình sẽ cố gắng tối ưu hóa lề rộng hơn và chấp nhận nhiều lỗi phân loại hơn.



→ **Soft margin** phù hợp với các loại dữ liệu có ngoại lệ (outlier) hoặc dữ liệu không hoàn toàn tách biệt rõ ràng theo tuyến tính

5. **Kernel trick**

- Là một hàm ánh xạ dữ liệu từ **không gian ít chiều** sang **không gian nhiều chiều hơn**

→ Tìm được một siêu phẳng phân tách dữ liệu

- Ý tưởng: Khi dữ liệu trong không gian ban đầu (input space) không thể phân tách tuyến tính, **Kernel Trick** ánh xạ dữ liệu vào một không gian đặc trưng mới (feature space) với số chiều cao hơn. Trong không gian này, dữ liệu có thể trở thành **tuyến tính** và có thể được phân tách bằng một siêu phẳng.
- Các loại kernel phổ biến:
  - **Tuyến tính - Linear Kernel:** Sử dụng khi **dữ liệu** có thể **phân tách tuyến tính.**
  - **Đa thức - Polynomial kernel:** Ánh xạ dữ liệu vào không gian đa thức, phù hợp với **dữ liệu có tính phi tuyến tính** nhưng **có xu hướng** được **mô hình hoá** tốt **bởi hàm đa thức**
  - **RBF - Radial Basis Function:** Phổ biến nhất với **dữ liệu phi tuyến tính**. Kernel này đo độ tương đồng giữa hai điểm dựa trên khoảng cách giữa chúng
  - **Sigmoid kernel:** Liên quan đến **hàm kích hoạt sigmoid** trong neuron network, có thể được sử dụng trong **các bài toán phi tuyến tính.**

→ Khi sử dụng mô hình SVM, có thể sử dụng GridSearch để tìm kiếm (fine-tune) các hyper-parameters tốt nhất cho dữ liệu.

B. Naive Bayes

1. **Tổng quan**
   - Là thuật toán phân loại thuộc nhóm Supervised Learning dựa trên định lý Bayes
   - Với giả định rằng các đặc trưng (features) của dữ liệu **là độc lập với nhau** (nghĩa là mỗi đặc trưng đóng góp vào việc dự đoán kết quả mà không phụ thuộc vào các đặc trưng khác).
   - **Naive Bayes** cho kết quả tốt trong nhiều bài toán, đặc biệt là **phân loại văn bản: phân loại email, phân loại văn bản theo chủ đề, và phân tích tình cảm**
2. **Công thức**

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

3. **Cách hoạt động của Naive Bayes trong bài toán Classification**
    - Với mỗi lớp C_k, tính xác suất có điều kiện của từng đặc điểm x_i với lớp đó: $P(x_i|C_k)$.
    - Tính xác suất kết hợp: $P(C_k|x_1, x_2, \ldots, x_n) = P(C_k) \prod P(x_i|C_k)$, theo nguyên lý Bayes.
    - Chọn lớp có xác suất lớn nhất làm nhãn dự đoán.