ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA TOÁN - CƠ - TIN HỌC

BÁO CÁO CUỐI KÌ

Tên đề tài: CHẨN ĐOÁN TÌNH TRẠNG KHỐI U CỦA BỆNH NHÂN MẮC BỆNH UNG THƯ VÚ

Nhóm thực hiện: Nhóm 9

MACHINE LEARNING

Hà Nội - 04/2025

Báo Cáo Cuối Kỳ

Machine Learning

Giảng viên hướng dẫn: TS. Cao Văn Chung

Sinh viên thực hiện: Nguyễn Thành Trung - 22001672

Nguyễn Thị Ánh - 22000070

Nguyễn Tiến Đạt - 22000081

Ngày 18 tháng 04 năm 2025

Tóm tắt nội dung

Dự án này tập trung vào việc ứng dụng các mô hình học máy để phân loại ung thư vú (lành tính - Benign hoặc ác tính - Malignant) dựa trên bộ dữ liệu Breast Cancer Wisconsin (Diagnostic) – một tập dữ liệu kinh điển, thường được sử dụng để đánh giá hiệu suất các thuật toán phân loại. Với đặc điểm dữ liệu có khả năng tách tuyến tính, dự án triển khai các phương pháp tiền xử lý và giảm chiều để tối ưu hóa hiệu quả phân loại.

Phần đầu tiên trình bày cơ sở lý thuyết về các kỹ thuật học máy, bao gồm chuẩn hóa dữ liệu, giảm chiều bằng PCA và LDA, cũng như các mô hình phân loại như K-Nearest Neighbors (KNN), Logistic Regression,... Phạm vi nghiên cứu bao gồm các giai đoạn từ tiền xử lý dữ liệu, giảm chiều, đến huấn luyện và đánh giá mô hình. Các bước thực nghiệm được mô tả chi tiết, bao gồm chuẩn hóa dữ liệu, áp dụng PCA (giữ 95% phương sai), LDA, và huấn luyện mô hình KNN (k=23), Logistic Regression...

Kết quả thực nghiệm cho thấy cả hai mô hình đạt độ chính xác tổng thể trên 90%, chứng minh hiệu quả của các phương pháp học máy trong bài toán phân loại ung thư vú. Báo cáo nhấn mạnh khả năng áp dụng các mô hình này trong chẩn đoán y khoa, hỗ trợ bác sĩ đưa ra quyết định chính xác hơn. Kết luận khẳng định tiềm năng của các kỹ thuật học máy trong việc xử lý dữ liệu y tế và mở ra hướng nghiên cứu mới về tối ưu hóa các thuật toán phân loại.

Mục lục

Chương 1

Lý do chọn đề tài

Ung thư vú là một trong những căn bệnh ung thư phổ biến và gây tử vong cao nhất ở phụ nữ trên toàn thế giới. Theo Tổ chức Y tế Thế giới, mỗi năm có hàng triệu ca mắc mới và hàng trăm nghìn ca tử vong do bệnh này. Tuy nhiên, nếu được phát hiện và điều trị sớm, khả năng sống sót có thể đạt trên 90%. Vì vậy, việc chẩn đoán sớm, chính xác và kịp thời đóng vai trò then chốt trong việc nâng cao hiệu quả điều trị và kéo dài tuổi thọ cho bệnh nhân. Hiện nay, các kỹ thuật học máy (machine learning) ngày càng được ứng dụng rộng rãi trong lĩnh vực y tế, đặc biệt là trong chẩn đoán hình ảnh, phân tích dữ liệu lâm sàng và hỗ trợ quyết định điều trị. Học máy có khả năng khai thác các mẫu dữ liệu phức tạp mà con người khó nhận ra, từ đó đưa ra dự đoán có độ chính xác cao và nhất quán. Trong bối cảnh dữ liệu y tế ngày càng phong phú, đây là công cụ mạnh mẽ để hỗ trợ bác sĩ đưa ra quyết định nhanh chóng và đáng tin cậy hơn.

Đề tài này sử dụng bộ dữ liệu Breast Cancer Wisconsin (Diagnostic), một tập dữ liệu kinh điển chứa các đặc trưng hình thái học tế bào, rất phù hợp cho bài toán phân loại khối u lành tính hoặc ác tính. Thông qua việc triển khai các bước tiền xử lý dữ liệu, áp dụng các kỹ thuật giảm chiều (PCA, LDA), xây dựng và đánh giá nhiều mô hình học máy khác nhau, dự án không chỉ giúp người học nâng cao kiến thức thực tiễn mà còn mở rộng hiểu biết về cách học máy góp phần vào công cuộc chăm sóc sức khỏe.

Chọn đề tài này không chỉ vì tính thực tiễn và học thuật, mà còn vì mong muốn góp phần vào việc phát triển các giải pháp công nghệ y tế – một lĩnh vực vừa giàu tiềm năng nghiên cứu, vừa mang ý nghĩa nhân văn sâu sắc.

Chương 2

Tiền xử lý dữ liệu

Tổng quan về DataFrame `df`

Kiểm tra số hàng và cột: `df.shape`, tên cột và kiểu dữ liệu: `df.dtypes`, số lượng giá trị không thiếu: `df.count()`, dung lượng bộ nhớ: `df.info()`.

Kiểm tra giá trị thiếu

```
df.isna().sum()
```

Kiểm tra số lượng giá trị thiếu (`NaN`) trong từng cột. `df.isna()` trả về DataFrame với `True` cho phần tử thiếu, `False` cho phần tử không thiếu. `sum()` tổng hợp số lượng `True`.

Thống kê biến mục tiêu

```
df['diagnosis'].value_counts()
```

Đếm số mẫu có nhãn `'M'` (ác tính) và `'B'` (lành tính) trong cột `diagnosis`.

Mã hóa nhãn bằng Label Encoding

```
df['diagnosis'].replace({"B": "0", "M": "1"}, inplace=True)
```

Chuyển nhãn từ dạng chữ cái sang số: `"B"` $\rightarrow$ `"0"` (lành tính), `"M"` $\rightarrow$ `"1"` (ác tính).

Trực quan hóa dữ liệu bằng Seaborn

```
sns.catplot(x='diagnosis', data=df, kind='count')
```

Biểu diễn số lượng mẫu mỗi loại bằng biểu đồ cột.

Loại bỏ cột không cần thiết

```
df.drop(['id', 'Unnamed: 32'], axis=1, inplace=True)
```

Xóa các cột không cần thiết, đặc biệt là `Unnamed: 32` có thể là cột rỗng (không xác định).

Tách biến đầu vào và đầu ra

```
X = df.drop(['diagnosis'], axis=1)
y = df['diagnosis']
```

Tách `X` là dữ liệu đầu vào, `y` là biến mục tiêu.

Chia tập dữ liệu

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.3,
    random_state=42,
    stratify=y
)
```

Chia tập dữ liệu với 30% dữ liệu làm tập kiểm tra, giữ nguyên tỷ lệ phân bố nhãn giữa hai tập.

Chuẩn hóa dữ liệu

Việc chuẩn hóa rất quan trọng trong học máy, đặc biệt với các mô hình nhạy cảm với đơn vị đo như SVM, KNN, Logistic Regression.
`StandardScaler` từ `scikit-learn` chuẩn hóa dữ liệu theo phân phối chuẩn với trung bình 0 và độ lệch chuẩn 1:

$$x_{\text{scaled}} = \frac{x - \mu}{\sigma}$$

Trong đó:

- x là giá trị dữ liệu ban đầu
- μ là trung bình đặc trưng: $\mu = \frac{x_1 + x_2 + \cdots + x_N}{N}$
- σ là độ lệch chuẩn: $\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$

Chương 3

Giảm chiều và trực quan hóa dữ liệu

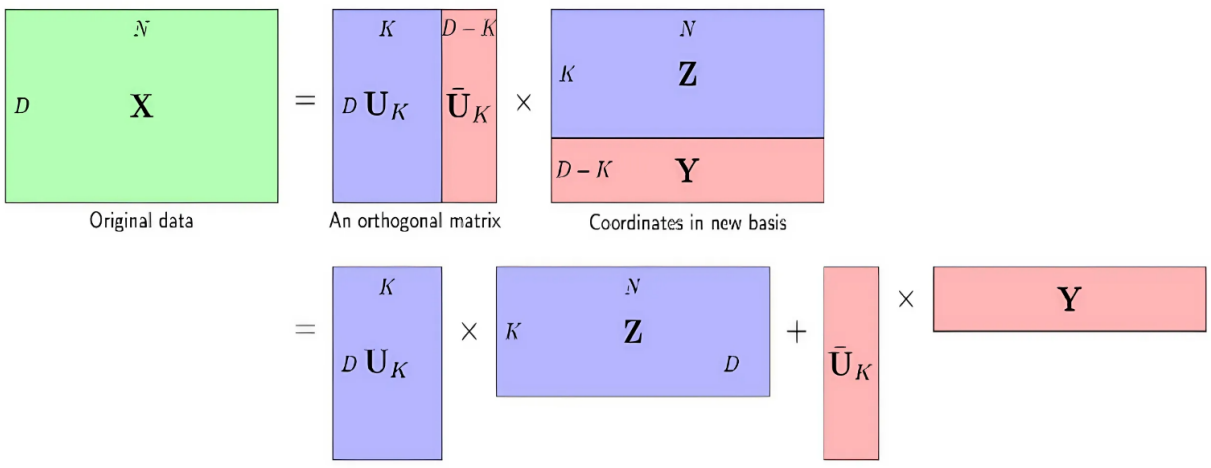
3.1 Phân tích thành phần chính (PCA)

3.1.1 Mục đích

Phân tích thành phần chính (PCA) là phương pháp giảm chiều không giám sát, nhằm loại bỏ dư thừa và giữ lại các thành phần mang nhiều thông tin nhất. PCA biến đổi dữ liệu sang không gian mới với các trục tọa độ ít tương quan, giúp giảm chi phí tính toán, cải thiện hiệu suất mô hình và hỗ trợ trực quan hóa trong các bài toán dữ liệu có số chiều lớn.

3.1.2 Cơ sở lý thuyết

Giả sử hệ cơ sở chuẩn mới là U và chúng ta muốn giữ lại K tọa độ trong hệ cơ sở mới này. Không mất tính tổng quát, giả sử đó là K thành phần đầu tiên.



Trong hình minh họa, hệ cơ sở mới $\mathbf{U} = [\mathbf{U}_k, \bar{\mathbf{U}}_k]$ là hệ trực chuẩn với $\mathbf{U}_k$ là ma trận con tạo bởi k cột đầu tiên của $\mathbf{U}$.

Mục đích của PCA là đi tìm ma trận trực giao $\mathbf{U}$ sao cho phần lớn thông tin được giữ lại ở phần $\mathbf{U}_k\mathbf{Z}$ (phần màu xanh) và có thể lược bỏ phần $\bar{\mathbf{U}}_k\mathbf{Y}$ (phần màu đỏ) và thay bằng một ma trận không phụ thuộc vào điểm dữ liệu.

3.1.3 Các bước thực hiện

1. **Tính vector kỳ vọng của toàn bộ dữ liệu:**

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

   Trong đó N là số lượng mẫu, và $\mathbf{x}_n$ là vector dữ liệu thứ n.

2. **Tính dữ liệu chuẩn hóa $\hat{\mathbf{x}}_n$**

$$\hat{\mathbf{x}}_n = \mathbf{x}_n - \bar{\mathbf{x}}; \quad n = 1, 2, \ldots, N.$$

3. **Tính ma trận hiệp phương sai:**

$$\mathbf{S} = \frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^T$$

   Trong đó $\hat{\mathbf{X}}$ là ma trận dữ liệu đã được chuẩn hóa.

4. **Tính các trị riêng λ_i và vector riêng $\mathbf{u}_i$ của ma trận hiệp phương sai.**

5. **Chọn k giá trị riêng lớn nhất và k vector riêng tương ứng để xây dựng ma trận $\mathbf{U}_k$.**

   Tập hợp các vector này được dùng để tạo thành không gian con mới có số chiều nhỏ hơn.

6. **Các cột của $\{\mathbf{U}_j\}_{j=1}^{k}$ là hệ cơ sở trực chuẩn**.

7. **Chiếu dữ liệu ban đầu đã chuẩn hóa $\hat{\mathbf{X}}$ xuống không gian con mới:**

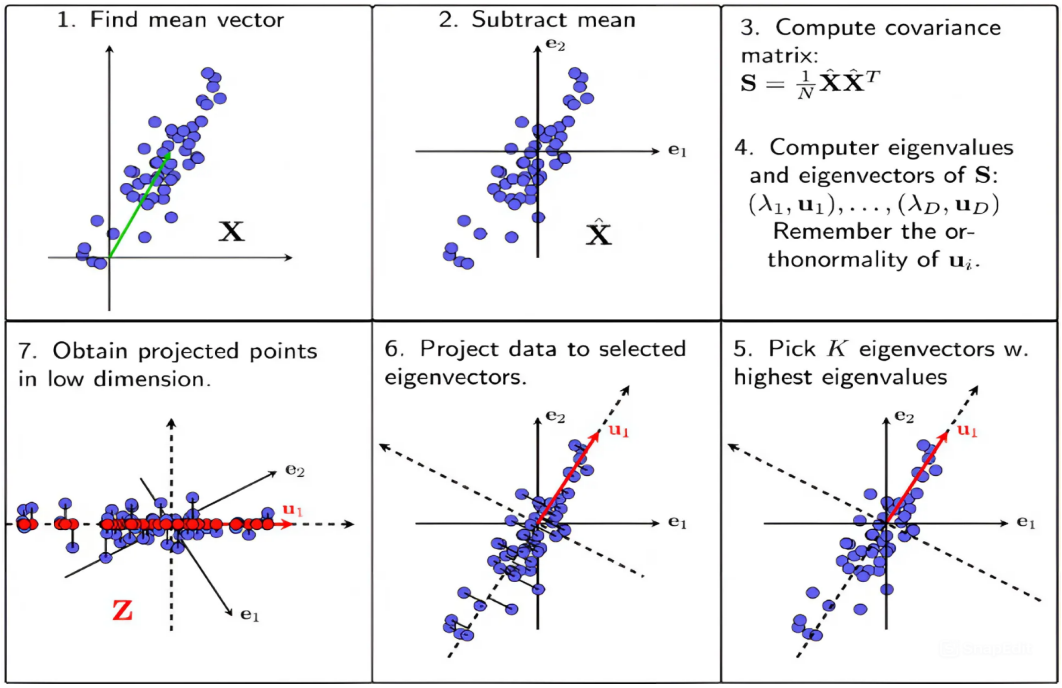
$$\mathbf{Z} = \mathbf{U}_k^T\hat{\mathbf{X}}.$$

   Ma trận $\mathbf{Z}$ chứa các biểu diễn nén của dữ liệu trên không gian có số chiều thấp hơn.

Dữ liệu ban đầu có thể được xấp xỉ lại từ dữ liệu nén như sau:

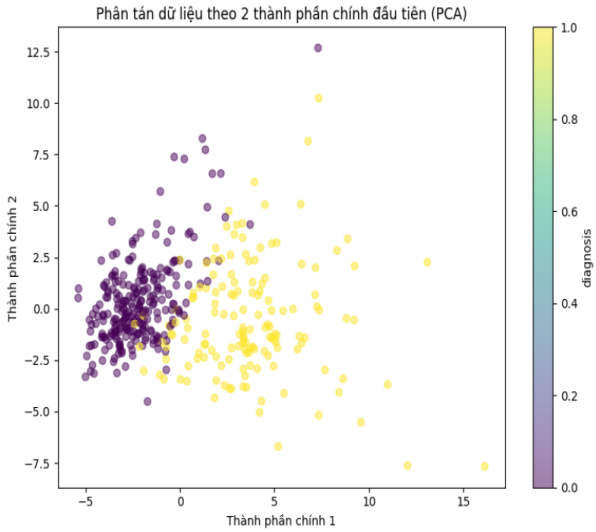
$$\mathbf{x} \approx \mathbf{U}_k\mathbf{Z} + \bar{\mathbf{x}}.$$

Trong đó $\mathbf{U}_k\mathbf{Z}$ là phần thông tin giữ lại và $\bar{\mathbf{x}}$ là trung bình của dữ liệu ban đầu được cộng lại để khôi phục vị trí gốc của dữ liệu.



3.1.4 Trực quan hóa

Bằng thuật toán đã cài đặt lên bộ dữ liệu, ta thu được kết quả sau.

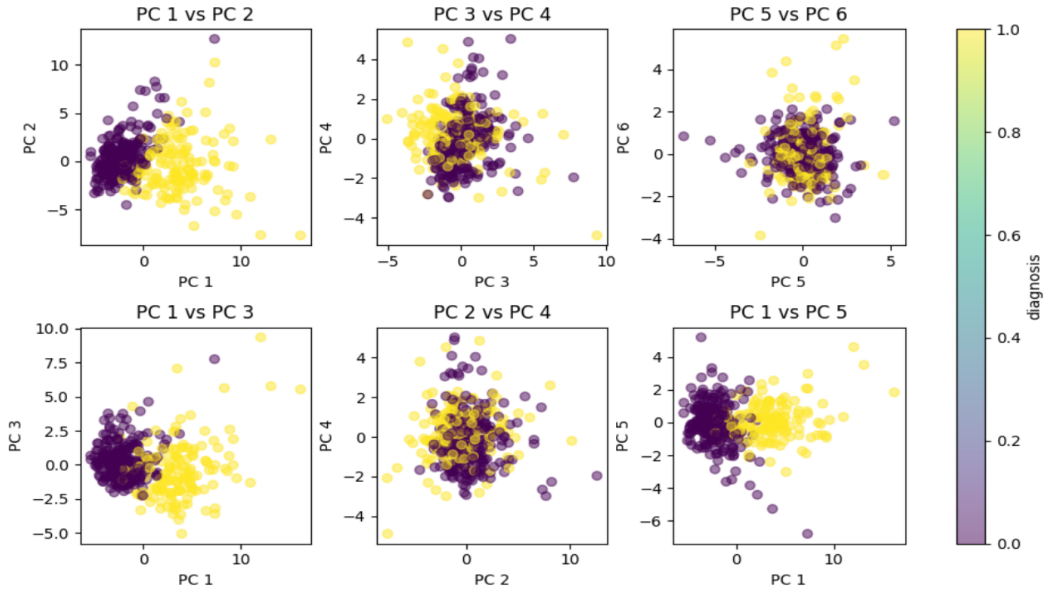


Hình 3.1: Phân tán dữ liệu theo 2 thành phần chính đầu tiên (PCA)

Biểu đồ PCA cho thấy sự phân tách rõ rệt giữa hai nhóm bệnh nhân:

- **Nhóm "B" (Lành tính, gán nhãn "0")**: Tập trung ở phía **trái**, màu **tím**, với giá trị thấp ở cả hai thành phần chính, phản ánh mức độ nghiêm trọng thấp.
- **Nhóm "M" (Ác tính, gán nhãn "1")**: Tập trung ở phía **phải**, màu **vàng**, với giá trị cao, phản ánh mức độ nghiêm trọng cao.

PCA đã giảm chiều dữ liệu hiệu quả, làm nổi bật sự phân tách giữa hai nhóm.



Hình 3.2: Biểu đồ phân tán cho 6 thành phần chính đầu tiên

Nhận xét:

- **PC1 vs PC2**: Cặp thành phần này phân tách rõ rệt giữa nhóm "M" (vàng, PC1 cao) và nhóm "B" (tím, PC1 thấp).
- **PC1 vs PC3 và PC1 vs PC5**: Các cặp này cũng phân tách hai nhóm, nhưng không hoàn hảo như PC1 vs PC2, với một số điểm dữ liệu chồng lấn.
- **PC3 vs PC4 và PC2 vs PC4**: Các cặp này ít phân tách rõ rệt, với sự chồng lấn giữa hai nhóm.

3.2 Phân tích phân biệt tuyến tính (LDA)

3.2.1 Mục đích

Phân tích phân biệt tuyến tính (LDA) là phương pháp giảm chiều dữ liệu, tối đa hóa khả năng phân biệt giữa các nhóm dữ liệu đã gán nhãn. Khác với PCA, LDA sử dụng thông

tin nhãn lớp để xác định hướng chiếu tối ưu xuống không gian tuyến tính. Mục tiêu của LDA là tìm trục chiếu sao cho dữ liệu có sự phân tách rõ ràng giữa các lớp, nâng cao hiệu quả phân loại trong các bài toán như nhận dạng khuôn mặt, giọng nói, hay phân loại văn bản. LDA không chỉ giảm chiều mà còn tăng cường khả năng phân loại cho các hệ thống học giám sát.

3.2.2 Binary labeled data

Giả thiết

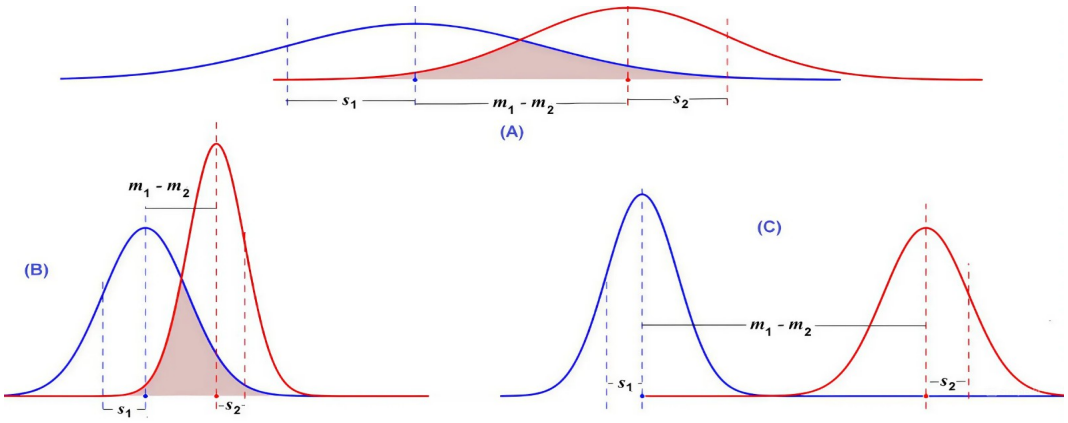
Ta xét bài toán phân loại nhị phân (binary classification) với hai nhãn: $\{01, 02\}$. Đặt:

- X_1: Tập dữ liệu có nhãn 01
- X_2: Tập dữ liệu có nhãn 02

Giả sử khi chiều dữ liệu xuống một chiều bất kỳ, ta có:

- Kỳ vọng (mean): m_1, m_2
- Phương sai (variance): s_1^2, s_2^2

Phân tích độ phân biệt trong các trường hợp của (m_1, m_2) và (s_1^2, s_2^2) như trong hình minh họa sau.



Hình 3.3: Ví dụ

Ta xét ba trường hợp minh họa bằng đồ thị:

- (A): $|m_1 - m_2|$ lớn nhưng s_1, s_2 cũng lớn → dữ liệu phân tán mạnh → vùng chồng lấn lớn → **độ phân biệt thấp**.

- (B): s_1, s_2 nhỏ nhưng $|m_1 - m_2|$ cũng nhỏ → dữ liệu gần nhau → vùng chồng lấn lớn → **độ phân biệt thấp**.
- (C): s_1, s_2 nhỏ và $|m_1 - m_2|$ lớn → vùng chồng lấn ít → **độ phân biệt cao**.

Từ các trường hợp trên, ta rút ra:

- **Phương sai trong lớp** (within-class variances): $N_j s_j^2$ với $N_j = |X_j|$ là số mẫu trong lớp X_j. Nếu nhỏ → dữ liệu tập trung → dễ phân biệt.
- **Phương sai giữa lớp** (between-class variance): $|m_1 - m_2|^2$. Nếu lớn → hai lớp phân bố xa nhau → dễ phân biệt.

Dữ liệu sẽ có khả năng phân biệt cao nếu:

- **Within-class variance nhỏ**: dữ liệu trong từng lớp ít phân tán.
- **Between-class variance lớn**: khoảng cách giữa các lớp lớn.

Các bước thực hiện phân tích Linear Discriminant Analysis (LDA)

Để thực hiện phân tích Linear Discriminant Analysis (LDA), chúng ta làm theo các bước sau:

1. **Bước 1: Tính S_W và S_B**

   Trong bước này, ta tính phương sai trong lớp (within-class variance) S_W và phương sai giữa các lớp (between-class variance) S_B.

   - **Phương sai trong lớp (within-class variance)**:

   $$S_W = \sum_{n \in C_1} (x_n - m_1)(x_n - m_1)^T + \sum_{n \in C_2} (x_n - m_2)(x_n - m_2)^T$$

   S_W thể hiện sự phân tán trong mỗi lớp, tức là sự biến động của các điểm trong lớp so với kỳ vọng của lớp đó.

   - **Phương sai giữa các lớp (between-class variance)**:

   $$S_B = (m_1 - m_2)(m_1 - m_2)^T$$

   S_B thể hiện sự phân tán giữa các lớp, tức là khoảng cách giữa các kỳ vọng m_1 và m_2 của hai lớp.

2. **Bước 2: Tính** $A = S_W^{-1} S_B$

$$A = S_W^{-1} S_B$$

Ma trận A này sẽ giúp ta tìm ra vector phân biệt tối ưu trong không gian dữ liệu.

3. **Bước 3: Tính** $L = \max\{\lambda_i\}$**, với** $i = 1, 2, \ldots, d$

$$L = \max\{\lambda_1, \lambda_2, \ldots, \lambda_d\}$$

.

Chú ý: Với mỗi giá trị riêng λ, sẽ có vô số vector riêng tương ứng.

4. **Bước 4: Chọn** w **sao cho**

Sau khi có giá trị riêng lớn nhất, ta chọn vector phân biệt w sao cho:

$$(m_1 - m_2)^T w = L$$

Vector w được tính như sau:

$$w = S_W^{-1}(m_1 - m_2) \cdot \beta \quad (\text{với } \beta > 0)$$

Trong đó β là hệ số điều chỉnh, giúp tối ưu hóa sự phân biệt giữa các lớp.

Hàm mất mát

Hàm mất mát trong LDA có thể được định nghĩa dưới dạng tỷ lệ giữa phương sai giữa các lớp và phương sai trong lớp.

$$J(w) = \frac{(m_1 - m_2)^T w}{w^T S_W w}$$

Hàm mất mát này đo lường sự phân biệt giữa các lớp trong không gian giảm chiều được tạo ra bởi vector w.

Hàm tối ưu

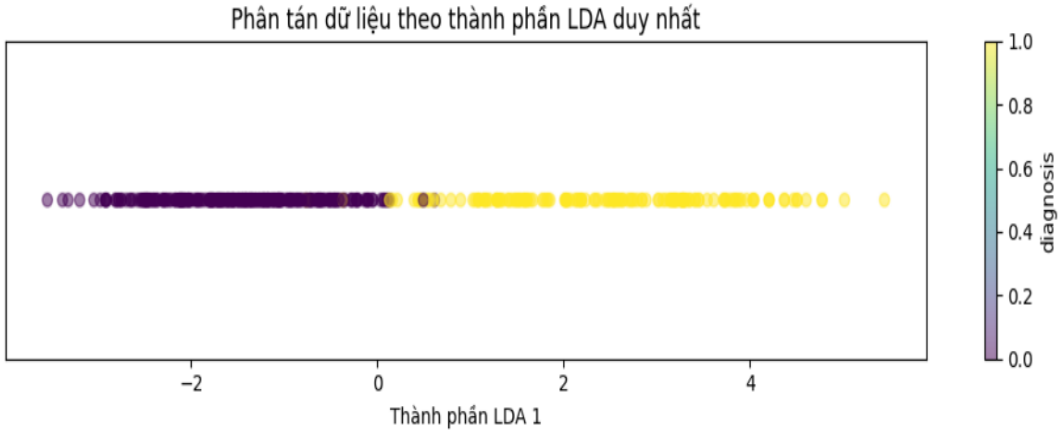
Để tối ưu hóa hàm mất mát $J(w)$, ta cần tìm giá trị của w sao cho hàm mất mát đạt cực đại. Bằng cách tính đạo hàm của $J(w)$ theo w và giải bài toán tối ưu, ta thu được:

$$w = S_W^{-1}(m_1 - m_2)$$

Bằng cách giải bài toán tối ưu này, ta tìm được vector phân biệt w sao cho độ phân biệt giữa các lớp là cao nhất, tức là phương sai giữa các lớp là lớn nhất trong khi phương sai trong lớp là nhỏ nhất.

3.2.3 Trực quan hóa

Bằng thuật toán đã cài đặt lên bộ dữ liệu, ta thu được kết quả sau.



Dữ liệu được phân tán dọc theo trục thành phần LDA 1, với một số chồng lấn nhẹ giữa các nhóm. Điều này có thể chỉ ra rằng LDA đã thực hiện phân tách tốt, nhưng không hoàn toàn rõ ràng giữa hai nhóm.

3.3 So sánh PCA và LDA

Đặc điểm	PCA	LDA
Ưu điểm	Không cần thông tin nhãn lớp, giữ được phần lớn phương sai của dữ liệu, phù hợp với các bài toán không giám sát hoặc khi nhãn lớp không đáng tin cậy.	Tối ưu hóa sự phân tách giữa các lớp, thường mang lại kết quả tốt hơn trong các bài toán phân loại giám sát.
Nhược điểm	Không tối ưu cho phân loại vì không xem xét sự phân tách giữa các lớp, có thể bỏ qua các đặc trưng quan trọng cho việc phân biệt.	Số chiều tối đa bị giới hạn ở $C - 1$, có thể không đủ để biểu diễn dữ liệu phức tạp. LDA cũng giả định dữ liệu tuân theo phân phối chuẩn, điều này có thể không hoàn toàn đúng với dữ liệu ảnh.

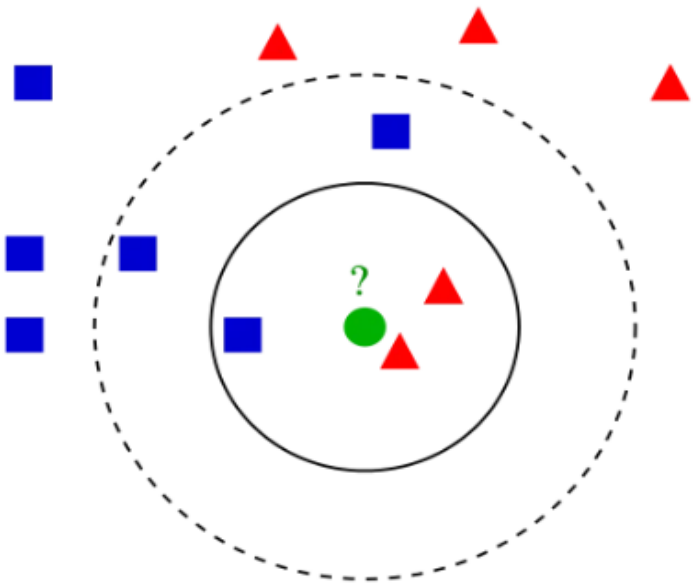
Chương 4

Các mô hình phân loại

4.1 K-nearest neighbors - KNN

4.1.1 Giới thiệu

K-nearest neighbor là một trong những thuật toán supervised-learning đơn giản nhất trong Machine Learning. Khi training, thuật toán này không học một điều gì từ dữ liệu huấn luyện, mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới.K-nearest neighbor có thể áp dụng được vào cả hai loại của bài toán Supervised learning là Classification và Regression.



Hình 4.1: Ví dụ

4.1.2 Cách một thuật toán KNN hoạt động

Trong bài toán phân loại nhị phân, đầu ra y thuộc một trong hai loại: 0 hoặc 1. Các bước thực hiện của phương pháp K-NN được mô tả như sau:

1. Với mỗi điểm x (chưa biết nhãn) trong tập Validation.
2. Tính khoảng cách d_i từ x đến từng điểm x_i trong tập Training $\Rightarrow$ lập thành một mảng các khoảng cách.
3. Sắp xếp các khoảng cách này theo thứ tự tăng dần, đồng thời giữ lại chỉ số ban đầu của các phần tử d_i.
4. Tìm K chỉ số tương ứng với K phần tử x_i trong tập Training có khoảng cách d_i nhỏ nhất.
5. Tính nhãn dự đoán y_{pred} như sau:

$$y_{\text{pred}} = \begin{cases} 1, & \text{nếu } \dfrac{\sum y_i}{K} \geq 0.5 \\ 0, & \text{nếu } \dfrac{\sum y_i}{K} < 0.5 \end{cases}$$

Giải thích bước (v): Do y chỉ nhận giá trị 0 hoặc 1, nên $\frac{\sum y_i}{K}$ chính là tỉ lệ xuất hiện của nhãn 1 trong số K láng giềng gần nhất. Nếu tỉ lệ này lớn hơn hoặc bằng 0.5, ta dự đoán $y_{\text{pred}} = 1$; ngược lại, nếu nhỏ hơn 0.5, ta dự đoán $y_{\text{pred}} = 0$.
Ta sẽ sử dụng lại các phương thức tính khoảng cách và tìm K mẫu gần nhất như sau:

4.1.3 Hàm tính khoảng cách

- **Trường hợp $d = 1$ (dữ liệu một chiều)**:

  Khoảng cách giữa điểm cần dự đoán x và từng điểm huấn luyện x_i được tính theo công thức:

$$d_i = |x - x_i|$$

- **Trường hợp $d > 1$ (dữ liệu nhiều chiều)**:

  Khi dữ liệu là vector nhiều chiều, khoảng cách giữa x và x_i được tính bằng công

thức chuẩn Euclid (L2 norm):

$$d_i = \|x - x_i\|_2 = \sqrt{\sum_{j=1}^{d}(x_j - x_{i,j})^2}$$

Trong đó:

- $x = (x_1, x_2, \ldots, x_d)$ là điểm cần dự đoán.
- $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d})$ là một điểm trong tập huấn luyện.
- d là số chiều của dữ liệu.

4.1.4 Tìm K phần tử gần nhất và dự đoán đầu ra

Sau khi tính được mảng các khoảng cách từ điểm cần dự đoán x đến các điểm trong tập huấn luyện, ta thu được một mảng khoảng cách:

$$\mathbf{D} = (d_1, d_2, \ldots, d_n)$$

Trong đó $d_i = \|x - x_i\|$ là khoảng cách từ điểm x đến mẫu huấn luyện x_i.
Ta sắp xếp các khoảng cách này theo thứ tự tăng dần và lấy chỉ số của K phần tử nhỏ nhất. Tập này tương ứng với K mẫu gần nhất với x trong tập huấn luyện.

Dự đoán đầu ra y cho điểm x được tính bằng cách lấy trung bình các nhãn của K láng giềng gần nhất: Gọi $y_i \in \{0, 1\}$ là nhãn của mẫu huấn luyện thứ i, khi đó:

$$\hat{y} = \frac{1}{K}\sum_{i=1}^{K} y_{\text{i}}$$

Và đầu ra cuối cùng được xác định theo quy tắc:

$$\hat{y} = \begin{cases} 1 & \text{nếu } \dfrac{1}{K}\displaystyle\sum_{i=1}^{K} y_{\text{i}\geq 0.5} \\ 0 & \text{nếu } \dfrac{1}{K}\displaystyle\sum_{i=1}^{K} y_{\text{i}<0.5} \end{cases}$$

Lưu ý khi chọn K

- Một lựa chọn phổ biến cho tham số K là:

$$K = \sqrt{n}$$

trong đó n là số lượng mẫu trong tập huấn luyện.

- Giá trị K **nên là số lẻ** để tránh trường hợp xảy ra tỷ lệ hòa (trung bình bằng 0.5) giữa hai lớp 0 và 1, dẫn đến mô hình không thể đưa ra quyết định phân loại dứt khoát.

- **Ghi chú:**

  – Nếu K quá nhỏ: mô hình dễ bị ảnh hưởng bởi nhiễu (overfitting).

  – Nếu K quá lớn: mô hình trở nên quá tổng quát (underfitting).

4.2 Hồi quy Logistic (Logistic Regression)

Hồi quy logistic là một thuật toán học máy có giám sát được sử dụng phổ biến cho các bài toán **phân loại nhị phân** (binary classification). Mục tiêu của bài toán này là dự đoán một biến đầu ra y chỉ nhận một trong hai giá trị rời rạc, thường được mã hóa là 0 và 1. Tức là, $y \in \{0, 1\}$. Mặc dù tên của phương pháp chứa thuật ngữ "Hồi quy" (Regression), vốn thường liên quan đến việc dự đoán các giá trị liên tục, hồi quy logistic thực chất là một mô hình **phân lớp** (classification). Tên gọi này xuất phát từ việc mô hình cơ bản của nó tương tự như hồi quy tuyến tính, nhưng kết quả đầu ra được biến đổi để phù hợp với bài toán phân loại.

4.2.1 Mô hình hóa Xác suất

Không giống như một số thuật toán phân loại khác chỉ đưa ra dự đoán lớp, hồi quy logistic mô hình hóa **xác suất có điều kiện** (conditional probability) của một điểm dữ liệu thuộc về lớp dương (thường là lớp 1). Cụ thể, mô hình ước lượng xác suất $P(y = 1|\mathbf{x}; \boldsymbol{\theta})$, trong đó:

- $\mathbf{x}$ là vector các đặc trưng đầu vào ($\mathbf{x} \in \mathbb{R}^{n+1}$, với $x_0 = 1$ là hệ số chặn).

- $\boldsymbol{\theta}$ là vector các tham số (trọng số) của mô hình ($\boldsymbol{\theta} \in \mathbb{R}^{n+1}$).

Giá trị đầu ra của mô hình, ký hiệu là $h_{\boldsymbol{\theta}}(\mathbf{x})$, biểu diễn xác suất này:

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = P(y = 1|\mathbf{x}; \boldsymbol{\theta})$$

4.2.2 Hàm Sigmoid (Logistic)

Để đảm bảo rằng giá trị đầu ra $h_{\boldsymbol{\theta}}(\mathbf{x})$ nằm trong khoảng $(0, 1)$, hồi quy logistic sử dụng **hàm sigmoid** (còn gọi là hàm logistic) để biến đổi tổ hợp tuyến tính của các đặc trưng và tham số ($\boldsymbol{\theta}^T\mathbf{x}$). Hàm sigmoid được định nghĩa là:

$$g(z) = \frac{1}{1 + e^{-z}}$$

Hàm này nhận đầu vào là một số thực bất kỳ ($z \in \mathbb{R}$) và trả về một giá trị trong khoảng $(0, 1)$. Do đó, giả thuyết (hypothesis) của mô hình hồi quy logistic được viết là:

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = g(\boldsymbol{\theta}^T\mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T\mathbf{x}}}$$

4.2.3 Quyết định lớp dựa trên xác suất

Sau khi tính được xác suất $h_{\boldsymbol{\theta}}(\mathbf{x})$, ta chuyển đổi nó thành dự đoán lớp cụ thể bằng quy tắc ngưỡng (thường là 0.5):

$$\hat{y} = \begin{cases} 1 & \text{nếu } h_{\boldsymbol{\theta}}(\mathbf{x}) \geq 0.5 \\ 0 & \text{nếu } h_{\boldsymbol{\theta}}(\mathbf{x}) < 0.5 \end{cases}$$

Điều này tương đương với việc kiểm tra $\boldsymbol{\theta}^T\mathbf{x} \geq 0$, vì hàm sigmoid $g(z) \geq 0.5$ khi và chỉ khi $z \geq 0$.

4.2.4 Hàm mất mát (Loss Function)

Để huấn luyện mô hình, ta cần một hàm mất mát đánh giá sai lệch giữa dự đoán và giá trị thực. Hồi quy logistic sử dụng **Log Loss** (hay **Binary Cross-Entropy**):

$$J(\boldsymbol{\theta}) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})) + (1 - y^{(i)})\log(1 - h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}))\right]$$

trong đó:

- m: số lượng mẫu huấn luyện.
- $y^{(i)}$: nhãn thực tế của mẫu thứ i.
- $h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$: xác suất dự đoán cho mẫu thứ i.

Hàm mất mát này khuyến khích mô hình dự đoán xác suất gần 1 cho các mẫu có $y = 1$ và gần 0 cho các mẫu có $y = 0$.

4.2.5 Tối ưu hóa mô hình

Mục tiêu là tìm $\boldsymbol{\theta}$ sao cho $J(\boldsymbol{\theta})$ nhỏ nhất. Phương pháp phổ biến là **Gradient Descent**:

$$\boldsymbol{\theta} := \boldsymbol{\theta} - \alpha \nabla J(\boldsymbol{\theta})$$

với α là tốc độ học (learning rate). Gradient của hàm mất mát được tính như sau:

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

4.2.6 Đặc điểm của mô hình

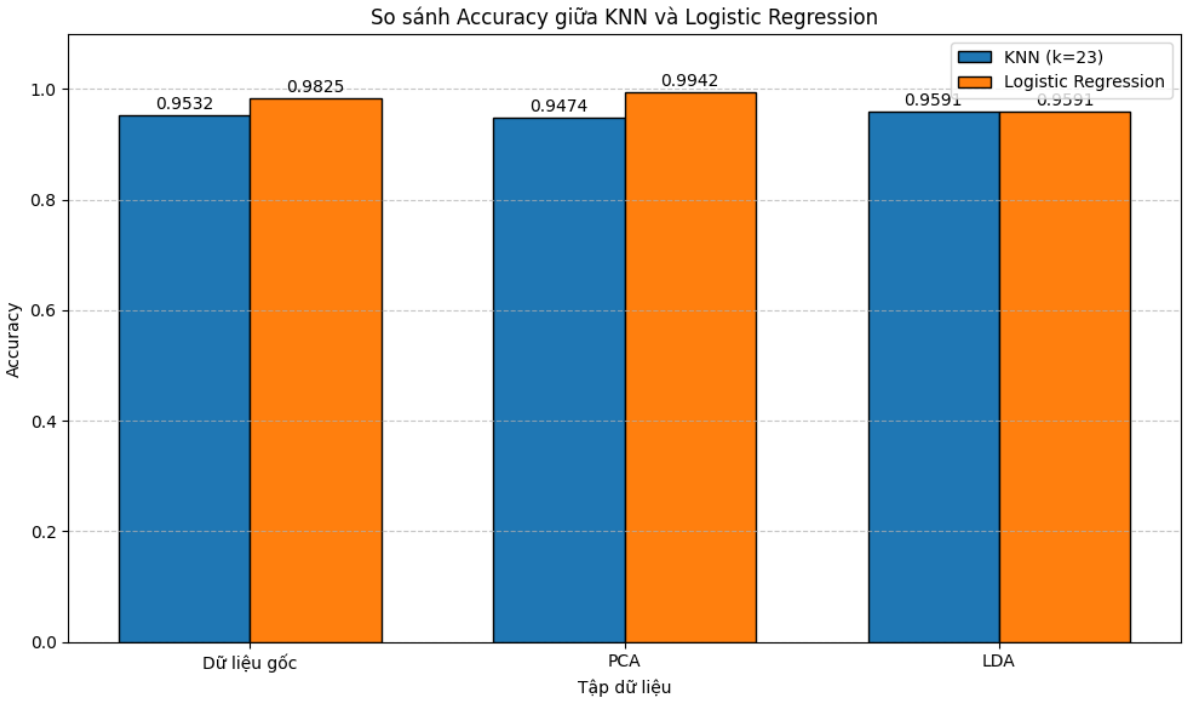
Hồi quy logistic là một **mô hình phân biệt** (discriminative model), học trực tiếp $P(y|\mathbf{x})$ mà không cần mô hình hóa $P(\mathbf{x}|y)$. Một số đặc điểm nổi bật:

- **Dễ diễn giải**: Trọng số $\boldsymbol{\theta}$ cho biết mức độ ảnh hưởng của từng đặc trưng.
- **Hiệu quả**: Phù hợp với dữ liệu phân biệt tuyến tính và dễ tính toán.

Chương 5

Kết quả thực nghiệm và nhận xét

5.1 Kết Quả Thực Nghiệm



Hình 5.1: Biểu đồ cột so sánh accuracy giữa KNN và Logistic Regression.

Dự án đã triển khai hai mô hình học máy là K-Nearest Neighbors (KNN) và Logistic Regression để phân loại ung thư vú (lành tính hoặc ác tính) trên bộ dữ liệu Breast Cancer Wisconsin (Diagnostic). Các mô hình được huấn luyện và đánh giá trên ba tập dữ liệu: dữ liệu gốc (sau chuẩn hóa), dữ liệu giảm chiều bằng PCA (giữ 95% phương sai), và dữ liệu giảm chiều bằng LDA. Các chỉ số hiệu suất chính bao gồm ma trận nhầm lẫn, báo cáo phân loại (precision, recall, F1-score), và accuracy.

Kết quả thực nghiệm cho thấy cả hai mô hình đều đạt độ chính xác (accuracy) tổng thể

trên 90%. Cụ thể:

- **Dữ liệu gốc**: Logistic Regression thường vượt trội hơn KNN về accuracy (98.25% so với 95.32%), nhờ khả năng tận dụng tính tách tuyến tính của dữ liệu.
- **Dữ liệu PCA**: Accuracy của mô hình KNN giảm nhẹ (về còn 94.74%) do mất một phần thông tin khi giảm chiều, nhưng vẫn duy trì hiệu quả cao. Riêng với Logistic Regression, accuracy của mô hình có phần tốt hơn (99.42%) gần như tuyệt đối, cho thấy phương pháp phù hợp nhất với dữ liệu giảm chiều theo PCA.
- **Dữ liệu LDA**: LDA mang lại kết quả tốt nhất đối với KNN, vì kỹ thuật này tối ưu hóa sự tách biệt giữa các lớp.

Biểu đồ cột so sánh accuracy giữa KNN và Logistic Regression cho thấy Logistic Regression có xu hướng hoạt động ổn định hơn trên cả ba tập dữ liệu, trong khi KNN nhạy cảm hơn với việc giảm chiều. Ma trận nhầm lẫn chỉ ra rằng các mô hình có tỷ lệ sai sót thấp, đặc biệt trong việc phân loại các trường hợp ác tính, điều này rất quan trọng trong chẩn đoán y khoa.

5.2 Nhận Xét

Kết quả thực nghiệm khẳng định tính hiệu quả của các mô hình học máy trong bài toán phân loại ung thư vú. Tính tách tuyến tính của bộ dữ liệu Wisconsin cho phép Logistic Regression đạt hiệu suất cao với độ phức tạp tính toán thấp, trong khi KNN cung cấp một cách tiếp cận phi tham số linh hoạt. Việc áp dụng PCA và LDA không chỉ giảm chiều dữ liệu, giúp tăng tốc độ huấn luyện, mà còn duy trì hoặc thậm chí cải thiện hiệu suất trong một số trường hợp (đặc biệt với LDA).

Dự án mang ý nghĩa thực tiễn lớn, vì các mô hình phân loại này có thể hỗ trợ bác sĩ trong việc chẩn đoán sớm ung thư vú, từ đó cải thiện tỷ lệ sống sót của bệnh nhân. Độ chính xác cao với trung bình 96.6% và độ lệch chuẩn thấp (1.82%) cho thấy các mô hình đáng tin cậy trong môi trường thực tế. Ngoài ra, việc so sánh hiệu suất trên các tập dữ liệu khác nhau cung cấp cái nhìn sâu sắc về tác động của giảm chiều đối với hiệu quả phân loại.

5.3 Hạn Chế và Hướng Phát Triển

Mặc dù đạt được kết quả khả quan, dự án vẫn tồn tại một số hạn chế. Thứ nhất, bộ dữ liệu Wisconsin có kích thước tương đối nhỏ (569 mẫu), có thể không phản ánh đầy đủ sự đa dạng của các trường hợp thực tế. Thứ hai, KNN yêu cầu lưu trữ toàn bộ dữ liệu huấn luyện, dẫn đến chi phí tính toán cao khi triển khai trên tập dữ liệu lớn hơn. Thứ ba, các tham số của mô hình (như $k = 23$ cho KNN) được chọn dựa trên kinh nghiệm, có thể chưa tối ưu hoàn toàn.

Trong tương lai, dự án có thể được mở rộng bằng cách:

- Thử nghiệm thêm các mô hình học máy khác (như SVM, Random Forest) hoặc học sâu (như mạng nơ-ron).
- Tối ưu hóa siêu tham số bằng các kỹ thuật như Grid Search hoặc Random Search.
- Sử dụng các bộ dữ liệu lớn hơn hoặc kết hợp nhiều nguồn dữ liệu để tăng tính tổng quát của mô hình.
- Phát triển giao diện người dùng để tích hợp mô hình vào quy trình chẩn đoán y khoa thực tế.

Kết luận, dự án không chỉ chứng minh tiềm năng của học máy trong chẩn đoán y khoa mà còn đặt nền tảng cho các nghiên cứu sâu hơn về ứng dụng trí tuệ nhân tạo trong lĩnh vực y tế.

Tài liệu tham khảo

[1] TS. Cao Văn Chung, *Giáo trình môn học Học máy*. Trường đại học Khoa học tự nhiên. Đại học Quốc gia Hà Nội.

Bộ dữ liệu của dự án được lấy từ nguồn: breast-cancer-wisconsin-data