ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

# MÔN: PATTERN RECOGNITION AND MACHINE LEARING

# Report: Lab01 – Classification with Linear Models


**22KDL1 – NHÓM 04**

**Họ và Tên**

Lý Vĩnh Thuận

Nguyễn Trần Lê Hoàng

Nguyễn Thuận Phát

Dương Thanh Phong

Phan Bình Phương

Huỳnh Thảo Quỳnh

Nguyễn Ngọc Thanh Thư

Kiều Thị Ngọc Vui


***Thành phố Hồ Chí Minh, ngày 03 tháng 11 năm 2024***

# MỤC LỤC


**1. Giới thiệu và Tổng quan về Dataset ..... 3**

**2. Data Loading & Preprocessing ..... 3**

**3. Logistic Regression & Support Vector Machine ..... 3**

3.1. Logistic Regression ..... 3

3.2. Support Vector Machine ..... 4

3.2.1. Linear kernel SVM ..... 4

3.2.2. RBF kernel SVM ..... 4

3.3. So sánh metrics của RBF kernel SVM với linear kernel SVM và Logistic Regression ..... 4

3.4. Phân tích, so sánh thời gian huấn luyện và đánh giá của SVM RBF và SVM Linear ..... 5

**4. Ảnh hưởng của việc giảm chiều dữ liệu và Cách giải quyết Lời nguyền số chiều (Curse of Dimensionality) ..... 5**

4.1. Đánh giá bộ dữ liệu ..... 5

4.2. Định nghĩa Curse of Dimensionality và ảnh hưởng đến mô hình máy học ..... 5

4.3. Triển khai Principal Component Analysis (PCA) / Linear Discriminant Analysis (LDA) . 6

4.3.1. Principal Component Analysis (PCA) ..... 6

4.3.2. Linear Discriminant Analysis (LDA) ..... 6

4.3.3. Tại sao cần LDA trong khi đã có PCA? ..... 6

4.4. So sánh hiệu suất của các mô hình (LR – Linear SVM – RBF SVM) được PCA/LDA với mô hình trên tập dữ liệu có chiều đầy đủ ..... 6

**5. So sánh và Đánh giá ..... 7**

**6. Kết luận ..... 8**

**7. Tài liệu tham khảo ..... 8**

# 1. Giới thiệu và Tổng quan về Dataset

Bài Lab này nhằm phân loại các mặt hàng từ bộ dữ liệu Fashion-MNIST với **Logistic Regression** (LR) và **Support Vector Machine** (SVM). Fashion-MNIST gồm 70.000 hình ảnh 28x28 pixel thuộc 10 loại thời trang. Ngoài xây dựng mô hình, Lab còn khám phá "**Curse of Dimensionality**" và sử dụng PCA/LDA để giảm số lượng đặc trưng, và tác động của chúng lên các mô hình được phân tích.

# 2. Data Loading & Preprocessing

Sau khi tìm hiểu tổng quan về Dataset, nhóm em đề xuất các bước tiền xử lí dữ liệu như sau:

- Chuẩn hóa dữ liệu từ khoảng ban đầu **[0, 255]** về khoảng **[0, 1]** để đảm bảo rằng các giá trị pixel lớn không ảnh hưởng tiêu cực đến hiệu suất của mô hình.
- Tiếp theo, làm phẳng dữ liệu, chuyển đổi mảng đa chiều thành mảng một chiều để đơn giản hóa, giảm độ phức tạp và đảm bảo tính nhất quán trong định dạng đầu vào.
- Kiểm tra phân bố dữ liệu bằng cách trực quan hóa để xác thực tính cân bằng của tập dữ liệu, đảm bảo hiệu suất và tính công bằng của mô hình khi huấn luyện.

Nhận xét sau khi tiền xử lý dữ liệu:

- Tập dữ liệu đã được cân bằng.
- Một số loại trang phục như áo thun và áo dài tay có nhiều điểm tương đồng, có thể gây khó khăn cho mô hình trong việc phân biệt. Cần lưu ý điều này trong quá trình huấn luyện mô hình.
- Các giá trị pixel đã được chuẩn hóa thành khoảng cố định **[0, 1]**.

Với kích thước mẫu tập train 60,000 mẫu, nhóm em quyết định triển khai phương pháp **Undersampling** và thử nghiệm các kích thước 10,000, 20,000, 30,000 và 40,000, với bước tăng 10,000, sử dụng **cross-validation** để đánh giá sự đánh đổi giữa độ chính xác và thời gian huấn luyện. Từ đó xác định kích thước mẫu tối ưu cho việc tinh chỉnh và phát triển mô hình.

# 3. Logistic Regression & Support Vector Machine

Train cả 2 models trên **full-dimensional data** và đánh giá metrics trên mô hình **chưa tuning.**

**Key metrics cho model:** Dựa vào công thức tính giữa các metrics, nhóm quyết định dùng F1-score và accuracy làm key metrics. Bởi vì F1-score cho cái nhìn tổng quát nhất về precision và recall, còn accuracy cho thấy tỉ lệ dự đoán đúng trên tổng số dự đoán.

## 3.1. Logistic Regression

- *Hiệu suất tổng quan* với accuracy của các lớp Shirt, Pullover, Coat thường thấp hơn mức trung bình, với macro avg là **0.84**, chỉ số F1 cho các lớp này dao động từ **0.59** đến **0.75**, cho thấy mô hình có hiệu suất không ổn định đối với những lớp này.
- *Confusion Matrix:*
  - Nhận dạng tốt với một số lớp: Trouser và Ankle boot có độ chính xác cao, với dự đoán đúng lần lượt là **956** và **949**.
  - Lỗi phổ biến do hình dạng tương tự: T-shirt/top dễ nhầm với Shirt (**109** mẫu),

và Pullover nhầm với Coat (**114** mẫu), do hình dạng tương đồng.

- Lớp ít nhầm lẫn: Sandal và Sneaker ít bị nhầm lẫn, có đặc điểm hình dạng rõ ràng hơn.
- Lỗi nhỏ và hợp lý: Ankle boot nhầm với Sneaker (**36** mẫu), do có một số đặc điểm chung.

- *Các lớp bị nhầm lẫn:* Shirt, Pullover, và Coat - Các lớp này có hình dáng và kiểu dáng tương tự nhau, gây khó khăn cho mô hình trong việc phân loại chính xác. Điều này có thể xuất phát từ sự tương đồng về cả thiết kế lẫn màu sắc của chúng.

## 3.2. Support Vector Machine

### 3.2.1. Linear kernel SVM

- *Hiệu suất tổng quan* với accuracy, recall, F1 score là **0.85**, precision là **0.85** tương đối tốt.
- *Confusion Matrix*:
  - Mô hình phân loại tốt các lớp Trouser, Sandal, Sneaker, Bag, Ankle boot với số mẫu phân loại đúng đều trên 900/1000 mẫu.
  - Đặc biệt lớp Shirt có hiệu suất kém chỉ đạt **0.59** F1-score do sự chồng lấn đặc trưng với các lớp T-shirt/top, Pullover và Coat. Các lớp nêu trên cũng dễ bị nhầm lẫn với nhau vì có đặc điểm tương tự về hình dáng và chất liệu.

### 3.2.2. RBF kernel SVM

- *Hiệu suất tổng quan* với accuracy đạt **0.88** thể hiện hiệu quả phân loại rất tốt trên dữ liệu với 784 chiều. Các chỉ số precision, recall, và F1-score trung bình cũng đều đạt khoảng **0.88**, cho thấy sự cân bằng giữa khả năng phát hiện chính xác các lớp và việc giảm thiểu các phân loại sai.
- *Confusion Matrix*: Quan sát ma trận nhầm lẫn cho thấy các lỗi phân loại chủ yếu xảy ra giữa các lớp có đặc điểm tương tự:
  - Nhầm lẫn nhiều giữa Shirt và Coat, hay giữa Pullover và Coat do tương đồng về hình dạng và màu sắc.
  - Các lớp Trouser và Sandal có rất ít lỗi phân loại, cho thấy mô hình nhận diện các đặc điểm rõ ràng của các lớp này khá tốt.

## 3.3. So sánh metrics của RBF kernel SVM với linear kernel SVM và Logistic Regression

| **Metric** | **Full dimension** | | |
|---|---|---|---|
| | **LR** | **Linear** | **RBF** |
| Accuracy | 0.8435 | 0.85 | 0.88 |
| F1-score | 0.8426 | 0.85 | 0.88 |
| Training time | 241.62 | 517.9 | 402.17 |
| Evaluation time | 0.04 | 122.97 | 244.1933 |

- Nhìn chung, accuracy và F1-score của RBF cao nhất (**0.88**), vượt Linear (**0.84**) và LR (0.84), cho thấy cân bằng tốt giữa độ chính xác và khả năng phân loại tốt.
- Thời gian training: RBF có thời gian huấn luyện trung bình (**402.17**), nhanh hơn so với Linear nhưng chậm hơn khoảng 1.5 lần so với LR. Nhìn chung, sự chênh lệch thời gian huấn luyện giữa các mô hình là không đáng kể.
- Thời gian dự đoán: LR mang lại thời gian phản hồi nhanh nhất trong cả ba mô hình. LR có thể chấp nhận đánh đổi một phần nhỏ trong F1-score để đạt được tốc độ phản hồi đáng kể, gấp 1000 lần so với 2 mô hình còn lại. Do đó, LR trở thành lựa chọn lý tưởng cho các tác vụ yêu cầu thời gian phản hồi theo thời gian thực.

**Kết luận:** Hiệu suất phân loại thấp ở một số lớp của tập dữ liệu Fashion-MNIST chủ yếu xuất phát từ sự tương đồng về hình dáng giữa các sản phẩm. Để cải thiện hiệu suất phân loại, cần áp dụng các giải pháp như tăng cường các đặc trưng phân biệt và cải tiến mô hình để phù hợp hơn với các điểm tương đồng này.

### 3.4. Phân tích, so sánh thời gian huấn luyện và đánh giá của SVM RBF và SVM Linear

Phân tích và giải thích lí do vì sao RBF tuy có thời gian huấn luyện thấp hơn nhưng thời gian dự đoán lại cao hơn Linear:

- Thời gian training: RBF sử dụng một hàm phi tuyến tính và các tối ưu hóa để tìm tham số tốt nhất. Ngược lại, linear cần nhiều vòng lặp để tối ưu trọng số trong không gian đa chiều, dẫn đến thời gian lâu hơn đối với dữ liệu phi tuyến tính.
- Thời gian dự đoán: RBF phải tính hàm RBF cho từng điểm đầu vào so với tất cả điểm trong tập train, còn Linear chỉ cần nhận các đặc trưng đầu vào với trọng số nên có thể thực hiện nhanh chóng.

## 4. Ảnh hưởng của việc giảm chiều dữ liệu và Cách giải quyết Lời nguyền số chiều (Curse of Dimensionality)

### 4.1. Đánh giá bộ dữ liệu

Bộ dữ liệu Fashion-MNIST có 70.000 hình ảnh, mỗi hình 28x28 pixel (784 đặc trưng), gây khó khăn trong tính toán và tăng nguy cơ overfitting. Một số đặc trưng, như các pixel viền thường có giá trị 0, không đóng góp nhiều vào phân loại. Do đó, giảm chiều dữ liệu bằng PCA hoặc LDA là cần thiết để giảm tải và xử lý vấn đề "**Curse of Dimensionality**".

### 4.2. Định nghĩa Curse of Dimensionality và ảnh hưởng đến mô hình máy học

Lời nguyền chiều dữ liệu là một hiện tượng xảy ra khi số lượng chiều (features) của dữ liệu tăng theo cấp số nhân, không gian dữ liệu cũng tăng theo, khiến hiệu quả và hiệu suất của các thuật toán giảm sút, tạo ra nhiều vấn đề về tính toán, độ chính xác và khả năng phân loại.

Tác động đến các mô hình máy học:

- *Sự thưa thớt của dữ liệu:* không gian lớn làm cho dữ liệu "thưa thớt", gây khó khăn cho các thuật toán K-means để nhận diện mẫu
- *Tăng cường tính toán:* số chiều tăng cao làm tăng khối lượng tính toàn và tài nguyên.

- *Overfitting:* mô hình dễ khớp với nhiễu (noise) thay vì mẫu cơ bản. Cách để giảm thiểu là sử dụng PCA hoặc Regularization.
- *Hiện tượng "Concentration of Measuare":* phần lớn các điểm dữ liệu có xu hướng trở nên cách xa trung tâm và tập trung gần biên, gây sai lệch mô hình.
- *Mất ý nghĩa khoảng cách:* khoảng cách giữa các điểm dữ liệu như tương đương do không gian bị phân tán quá mức, làm giảm hiệu quả các thuật toán dựa trên khoảng cách (như KNN, Euclid).
- *Khó visualization:* nhiều chiều làm trực quan hóa phức tạp, hạn chết phát hiện mẫu.

## 4.3. Triển khai Principal Component Analysis (PCA) / Linear Discriminant Analysis (LDA)

### 4.3.1. Principal Component Analysis (PCA)

| Tỷ lệ phương sai (%) | Số lượng thành phần cần thiết |
|---|---|
| 80.0 | 24 |
| 85.0 | 43 |
| 90.0 | 84 |
| 95.0 | 187 |
| 99.0 | 459 |

→ Sử dụng PCA để đảm bảo cân bằng giữa việc giảm chiều dữ liệu và giữ lại phần lớn thông tin. Dựa trên dữ liệu cung cấp ở trên, để giữ được **95%** thông tin thì cần chọn **187 chiều**.

### 4.3.2. Linear Discriminant Analysis (LDA)

LDA hoạt động tốt khi dữ liệu thỏa mãn các giả định sau:

- Có phân phối chuẩn (Guassian)
- Phương sai đồng nhất

→ Nếu không tuân theo các giả định này thì hiệu quả của LDA có thể giảm đi.

LDA tìm tối đa $\boldsymbol{C-1}$ chiều (với $\boldsymbol{C}$ là số lớp) vì mỗi chiều đại diện cho sự khác biệt giữa các lớp. Nó giúp phân tách lớp rõ ràng, giữ đặc trưng riêng, và tối đa hóa thông tin phân loại mà không mất đi các thông tin quan trọng.

### 4.3.3. Tại sao cần LDA trong khi đã có PCA?

- *PCA:* giảm chiều không giám sát, tối đa hóa phương sai dữ liệu, không dùng nhãn lớp. Giữ nhiều thông tin nhưng không đảm bảo phân tách lớp.
- *LDA:* ngược lại, tìm không gian tối ưu để phân biệt các lớp bằng cách sử dụng nhãn lớp.

→ Việc giữ lại thông tin nhiều nhất (như trong PCA) không phải lúc nào cũng là cách tốt nhất để phân loại. Vì vậy, LDA thường được sử dụng trong các bài toán phân loại.

## 4.4. So sánh hiệu suất của các mô hình (LR – Linear SVM – RBF SVM) được PCA/LDA với

**mô hình trên tập dữ liệu có chiều đầy đủ**

| Metric | Full dimension | | | PCA | | | LDA | | |
|---|---|---|---|---|---|---|---|---|---|
| | LR | Linear | RBF | LR | Linear | RBF | LR | Linear | RBF |
| Accuracy | 0.84 | 0.85 | 0.88 | 0.84 | 0.85 | 0.89 | 0.82 | 0.83 | 0.83 |
| F1-score | 0.84 | 0.85 | 0.88 | 0.84 | 0.85 | 0.89 | 0.82 | 0.82 | 0.83 |
| Training time | 241.62 | 517.9 | 402.17 | 52.02 | 210.75 | 146.02 | 3.05 | 43.50 | 28.67 |
| Eval time | 0.04 | 122.97 | 244.19 | 0.011 | 40.37 | 84.93 | 0.0021 | 10.99 | 18.88 |

**Trade-offs giữa giảm chiều dữ liệu và hiệu suất mô hình:**

**Logistic Regression**

- Logistic Regression không giảm chiều đạt hiệu suất cao nhưng tốn thời gian. Nếu ưu tiên thời gian huấn luyện, LDA giúp tiết kiệm thời gian dù hiệu suất có giảm.
- PCA cân bằng giữa độ chính xác và thời gian, là lựa chọn hợp lý khi cần hiệu suất ổn định mà tiết kiệm thời gian.
- Với 9 chiều, LDA duy trì hiệu suất khá tốt, dù kém hơn PCA và full dimension nhưng chi phí tính toán rất thấp.

**Linear kernel – RBF kernel (SVM)**

- SVM với kernel RBF trên full dimension đạt độ chính xác cao nhất (**0.88**), nhưng thời gian huấn luyện khá lâu (**402.17** giây).
- SVM với PCA (kernel Linear và RBF) cân bằng giữa độ chính xác (**0.85** và **0.89**) và thời gian huấn luyện (**210.75** giây và **146.02** giây). PCA là lựa chọn hợp lý nếu cần giảm thời gian huấn luyện mà vẫn giữ hiệu suất tốt.
- LDA giữ độ chính xác ổn định (**0.83** cho cả Linear và RBF) với thời gian huấn luyện thấp (**43.50** giây và **28.67** giây), là lựa chọn khi ưu tiên tốc độ hơn độ chính xác tối đa.

**Kết luận hiệu quả mô hình sau khi giảm chiều dữ liệu:** Thông qua các chỉ số đánh giá (metrics), nhận thấy rằng **mô hình SVM RBF với kỹ thuật giảm chiều PCA** (giảm từ 784 còn 187 chiều) có hiệu suất vượt trội và thời gian huấn luyện hợp lí so với mô hình LR và SVM Linear. Vì mô hình SVM sử RBF kernel có khả năng học sự phức tạp giữa các lớp chồng chéo dễ dàng hơn. Bên cạnh đó, việc giảm chiều còn giúp tăng cường khả năng phân lớp giữa các lớp này khiến mô hình có hiệu suất vượt trội.

## 5. So sánh và Đánh giá

- Logistic Regression và SVM Linear đều có hiệu suất khá tốt trên các lớp có ranh giới phân loại rõ ràng. Nhưng kém hiệu quả đối với các lớp chồng chéo, nhất là dữ liệu phi tuyến.
- SVM RBF cải thiện đáng kể về độ chính xác cũng như hiệu suất, nhờ vào khả năng xử lí dữ liệu không phi tuyến và phức tạp.

- Logistic Regression có thời gian huấn luyện và dự đoán nhanh, nhưng không thể khai thác được các mối quan hệ phi tuyến. Mô hình này phù hợp khi cần kết quả nhanh chóng, nhưng không nên bỏ qua mô hình phức tạp hơn nếu mục tiêu là đạt được độ chính xác cao nhất.
- SVM RBF có độ chính xác cao nhất và xử lí được dữ liệu phi tuyến, tuy nhiên, thời gian dự đoán lại cao hơn đáng kể. Đây là yếu tố cần xem xét trong ứng dụng thực tế, đặc biệt khi thời gian là yếu tố quan trọng trong các ứng dụng yêu cầu phản hồi nhanh.

→ Có thể thấy bộ dữ liệu có tính chất phi tuyến nên sử dụng mô hình SVM RBF Kernel sẽ đạt được độ chính xác cao hơn nếu không bị hạn chế về thời gian và tài nguyên.

→ Tuy nhiên, trong thực tế, nếu yêu cầu về thời gian tính toán, thì có thể cân nhắc 2 mô hình Logistic Regression và SVM Linear Kernel.

## 6. Kết luận

Qua bài Lab, nhóm đã phân tích và so sánh hiệu suất của Logistic Regression và SVM trên bộ dữ liệu Fashion-MNIST, kết hợp với các phương pháp giảm chiều như PCA và LDA. Kết quả cho thấy mô hình SVM RBF kernel vượt trội về độ chính xác nhờ khả năng xử lý dữ liệu phi tuyến. Tuy nhiên, logistic regression và SVM linear vẫn là lựa chọn hợp lý khi cần tốc độ xử lý nhanh. Điều này cho thấy, việc lựa chọn mô hình cần cân nhắc giữa độ chính xác và thời gian xử lý, tùy thuộc vào mục tiêu và hạn chế cụ thể của ứng dụng thực tế. Việc tiếp tục nghiên cứu và tối ưu hóa các đặc trưng sẽ giúp cải thiện khả năng phân loại, đặc biệt là với các lớp có đặc điểm tương tự nhau trong bộ dữ liệu.

## 7. Tài liệu tham khảo


[1] The curse of dimensionality & Dimension reduction

[2] Applying Support Vector Machines and Logistic Regression on the Fashion MNIST dataset | SVM-LR-on-Fashion-MNIST

[3] Dimensionality Reduction and PCA for Fashion MNIST

[4] SVM algorithm

[5] The RBF kernel in SVM: A Complete Guide - PyCodeMates

[6] RBF SVM parameters — scikit-learn 1.5.2 documentation

[7] Logistic Regression

[8] SVC – scikit learn

[9] One-vs-Rest and One-vs-One for Multi-Class Classification