# CSC14005 – MACHINE LEARNING
# FINAL PROJECT

## Báo cáo: Support Vector Machine (S.V.M)


| Họ tên | — | Mã số sinh viên |
|---|---|---|
| Hồ Thế Phúc | — | 21127670 |
| Hoàng Nhật Quang | — | 21127543 |

# Mục lục

# I. Giới thiệu chung

## Thành viên nhóm:

- Hồ Thế Phúc — 21127670
- Hoàng Nhật Quang — 21127543

## Phân Công:

| Công Việc | Phân Công | Mức Độ Hoàn Thành |
|---|---|---|
| Linear Kernel | ● Hoàng Nhật Quang | 100% |
| Gaussian/RBF kernel | ● Hồ Thế Phúc | 100% |
| Báo Cáo và Nhận Xét về hiệu suất mô hình | ● Hoàng Nhật Quang<br>● Hồ Thế Phúc | 100% |

## Cách lấy giữ liệu từ trên github:

**Bước 1**:Tạo một thư mục **fashion** đặt trong folder **data** để chứa dữ liệu:
**Bước 2**: Tải dữ liệu từ GitHub :
https://github.com/zalandoresearch/fashion-mnist
**Bước 3**: Đảm bảo thư mục data và file final_project_21127670_21127543 nằm chung folder.
**Bước 4**: Sau khi tải dữ liệu trên link github thì bỏ vào trong **data/fashion** thì, bạn có thể sử dụng hàm **load mnist** để đọc dữ liệu từ thư mục **data.**

# II. Huấn luyện SVM

## 1. Linear Kernel

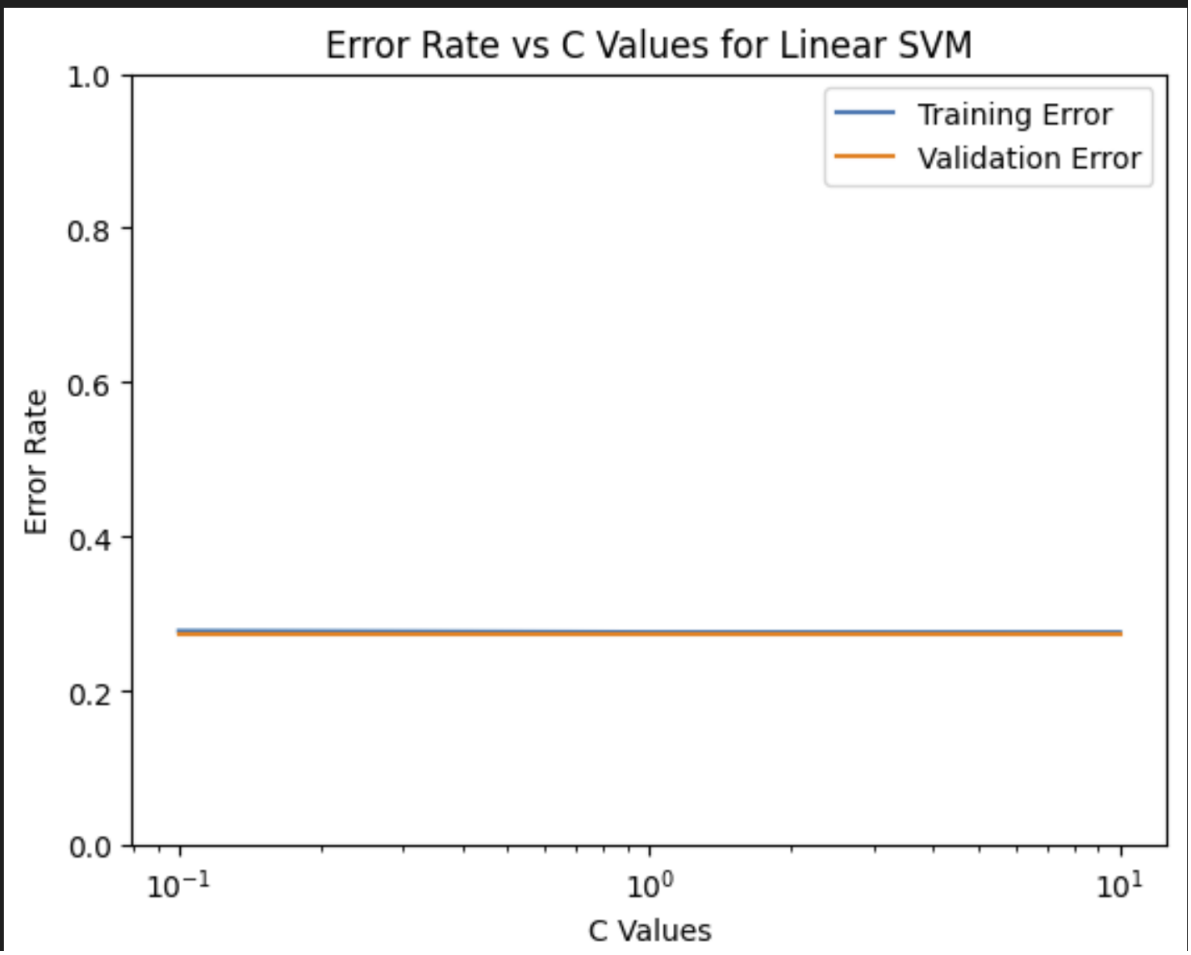
Linear kernel trong SVM là một phương pháp phổ biến để phân loại dữ liệu. Kernel này hoạt động bằng cách đo độ tương đồng tuyến tính giữa các điểm dữ liệu trong không gian feature ban đầu.

Khi sử dụng linear kernel, không cần phải chuyển đổi dữ liệu sang không gian feature cao hơn. Thay vào đó, nó đơn giản là tính toán tích vô hướng (dot product) giữa các điểm dữ liệu mà không cần thực hiện bất kỳ phép biến đổi nào.

Linear kernel phù hợp cho các tập dữ liệu có thể được phân loại một cách tuyến tính trong không gian feature ban đầu. Nó hoạt động tốt khi dữ liệu có ranh giới quyết định đơn giản và tập dữ liệu không có cấu trúc phức tạp. Tuy nhiên, nếu ranh giới giữa các lớp không tuyến tính, các kernel phi tuyến tính (như Gaussian RBF) thường được sử dụng để ánh xạ dữ liệu vào không gian feature cao hơn để tạo ra ranh giới quyết định phức tạp hơn.

## a. 5 đặc trưng

```
For C=0.1:
Training Error: 0.2777
Validation Error: 0.2748
Training Time: 16.66 seconds
----------------------------------
For C=1:
Training Error: 0.2761
Validation Error: 0.2748
Training Time: 26.73 seconds
----------------------------------
For C=10:
Training Error: 0.2760
Validation Error: 0.2748
Training Time: 71.69 seconds
----------------------------------
```



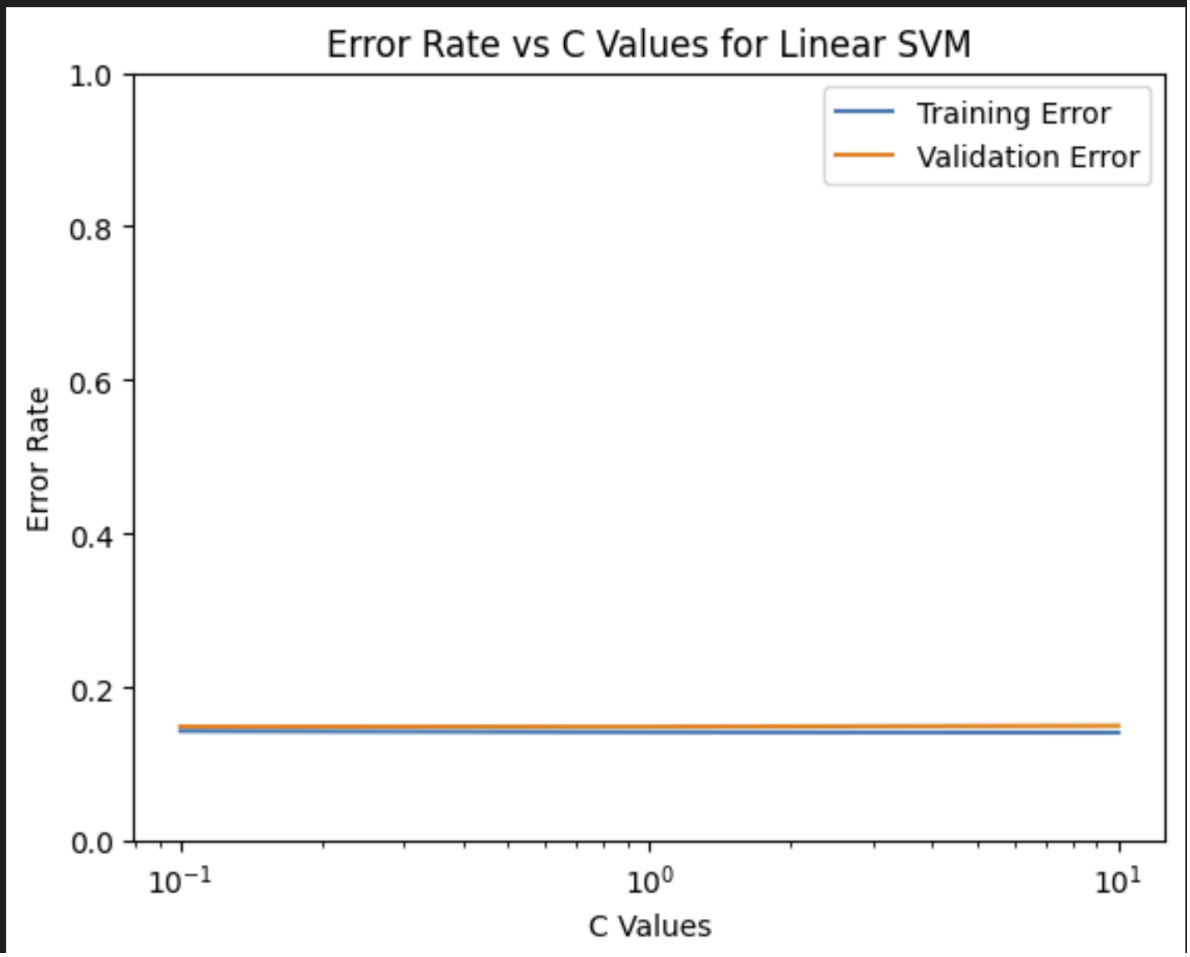

Nhận xét:

- Với 5 đặc trưng thì ta thấy được siêu tham số C ảnh hưởng đến thời gian huấn luyện không quá đáng kể.
- Tuy nhiên độ lỗi tương đối lớn khi lên đến hơn 26, 27% gây thiệt hại đáng kể cho người sử dụng.
- Ta thấy được khi rút từ 784 đặc trưng xuống con số rất nhỏ, ta đã bỏ qua rất nhiều trường, đặc trưng quan trọng.

- Nhưng độ lỗi chênh lệch giữa tập huấn luyện với tập thẩm định không quá lớn nên ta biết được mô hình không có bị quá overfitting tuy được huấn luyện với 60000 mẫu.
- Với siêu tham số C tăng dần thì tương đương với thời gian huấn luyện tăng dần. Nhưng mà giữa C= 0.1 với C=10 thì nó lại không tỷ lệ theo, C=10 tuy gấp 100 lần 0.1 nhưng thời gian huấn luyện lại không lâu như vậy và chỉ gấp 2,5 thời gian huấn luyện của C=1 gấp 10 lần của 0.1.
- So với C=0.1 thì C=1 có thời gian huấn luyện khá sát nhau nhưng độ lỗi huấn luyện nhỏ hơn 1 xíu.
- Nếu đánh giá về độ tốt tuy độ lỗi thẩm định ở cả 3 tập đều ngang nhau nhưng độ lỗi ở tập huấn luyện có giảm nhưng nó không đáng để ý, quan trọng ở độ lỗi khi thẩm định tuy ngang nhau nhưng chắc có thể nếu tập thẩm định lớn hơn nữa hoặc thêm nhiều đặc trưng hơn thì kết quả này có thể thay đổi 1 cách nào đó. Nhưng nếu phải chọn thì ta sẽ chọn ở siêu tham số C=1 vì do độ lỗi tập huấn luyện tuy không thấp hơn so với C=10 nhưng kết quả lại nhỏ hơn đáng kể nó, ta thấy được độ cân bằng giữa tập huấn luyện với tập thẩm định hơn, không bị overfitting.
- Ta thấy được với 5 đặc trưng thì bỉ bỏ sót một số thông tin quan trọng nên ngoại độ lỗi trên lệch không đáng kể mà siêu tham số càng lớn thì lại chẳng được lợi ích được gì nhiều so với siêu tham số nhỏ, tốn ít thời gian mà kết quả lại không quá chênh lệch.

## b. 50 đặc trưng

```
For C=0.1:
Training Error: 0.1431
Validation Error: 0.1490
Training Time: 38.54 seconds
----------------------------------
For C=1:
Training Error: 0.1416
Validation Error: 0.1488
Training Time: 84.19 seconds
----------------------------------
For C=10:
Training Error: 0.1410
Validation Error: 0.1500
Training Time: 383.92 seconds
----------------------------------
```




Nhận xét:

- So với 5 đặc trưng thì với 50 đặc trưng thì độ lỗi lại thấp hơn gần gấp đôi, ta thấy được sự lệch giữa độ lỗi huấn luyện với độ lỗi thẩm định lớn hơn 1 xíu. Cho thấy được với 784 thuộc tính thì 5 thuộc tính chưa chắc bao quát hết được các thuộc tính quan trọng và bị lượt bỏ.
- Đổi lại thì thời gian huấn luyện đã có thay đổi đáng kể.
- Ta thấy được sự chênh lệch khi siêu tham số lớn nhưng kết quả thu được lại không đáng cho thời gian bỏ đã bỏ ra.

- Chênh lệch tận 100 lần thời gian huấn luyện gấp gần 10 lần mà độ lỗi lại giảm chỉ có 1 xíu.
- Nên C = 10 là không tối ưu và nên bị loại bỏ.
- Còn với C = 1 thì thời gian huấn luyện tuy gần gấp 3 nhưng đạt được độ lỗi thẩm định nhỏ nhất trong cả 3 siêu tham số nên ta thấy được không phải siêu tham số càng lớn thì mô hình càng hiệu quả mà phải là con số phù hợp. Ngoài dựa vào biểu đồ vẽ được thì ta thấy được khoảng cách giữa độ lỗi thẩm định với huấn luyện tăng dần bắt đầu có dấu hiệu bị overfit nếu ta cho số C càng lớn hơn nữa.

# 2. Gaussian/RBF kernel

Gaussian SVM (Support Vector Machine) là một biến thể của thuật toán SVM được ứng dụng phổ biến trong học máy. Điểm độc đáo của Gaussian SVM chính là sử dụng kernel Gaussian, còn gọi là kernel RBF (Radial Basis Function), để tạo ra một biên phân loại phi tuyến.

Thuật toán SVM nhắm đến việc tìm ra một siêu phẳng tốt nhất để phân tách các điểm dữ liệu thuộc các lớp khác nhau trong không gian đặc trưng. Kernel Gaussian/RBF được tích hợp vào SVM nhằm ánh xạ dữ liệu từ không gian đặc trưng ban đầu sang một không gian chiều cao hơn.

Ánh xạ này giúp tạo ra biên quyết định phức tạp hơn, đặc biệt là khi đối mặt với dữ liệu không tuyến tính. Kernel Gaussian cho phép SVM xử lý hiệu quả những tình huống phức tạp và tạo ra các biên phân loại mạnh mẽ và linh hoạt. Do đó, Gaussian SVM trở thành một công cụ quan trọng trong việc giải quyết các vấn đề phân loại đặc biệt đối với dữ liệu có độ phức tạp và tính phi tuyến tính.

## a. 5 đặc trưng

```
C=0.1, gamma=0.1, Training Error: 0.2688, Validation Error: 0.2685, Training Time: 33.66 seconds
C=0.1, gamma=0.5, Training Error: 0.2533, Validation Error: 0.2538, Training Time: 36.89 seconds
C=0.1, gamma=1, Training Error: 0.2474, Validation Error: 0.2538, Training Time: 47.91 seconds
C=1, gamma=0.1, Training Error: 0.2513, Validation Error: 0.2508, Training Time: 27.67 seconds
C=1, gamma=0.5, Training Error: 0.2328, Validation Error: 0.2397, Training Time: 31.47 seconds
C=1, gamma=1, Training Error: 0.2171, Validation Error: 0.2418, Training Time: 39.90 seconds
C=10, gamma=0.1, Training Error: 0.2399, Validation Error: 0.2440, Training Time: 30.97 seconds
C=10, gamma=0.5, Training Error: 0.2094, Validation Error: 0.2383, Training Time: 40.55 seconds
C=10, gamma=1, Training Error: 0.1738, Validation Error: 0.2479, Training Time: 50.33 seconds
```

**Nhận xét từng trường hợp của C và gamma:**

1. **C=0.1, gamma=0.1:**
    - Dữ liệu có vẻ không quá phức tạp khiến mô hình không thể đạt được hiệu suất tốt trên cả tập huấn luyện và tập validation. Thời gian huấn luyện là trung bình.
2. **C=0.1, gamma=0.5:**

    - Tăng giá trị gamma giúp giảm training error, nhưng validation error không giảm đáng kể. Mô hình có khả năng không tổng quát hóa tốt trên dữ liệu mới.
3. **C=0.1, gamma=1:**
    - Tăng gamma tiếp tục giảm training error nhưng tăng validation error, cho thấy mô hình trở nên quá phức tạp và overfitting trên dữ liệu huấn luyện.
4. **C=1, gamma=0.1:**
    - Training và validation error giảm đều. Thời gian huấn luyện ngắn. Có thể là một cài đặt tốt với sự cân nhắc giữa hiệu suất và tốc độ.
5. **C=1, gamma=0.5:**
    - Hiệu suất trên cả hai tập dữ liệu tốt hơn so với C=0.1, gamma=0.5. Mô hình trở nên mạnh mẽ hơn mà không quá phức tạp.
6. **C=1, gamma=1:**
    - Hiệu suất tốt nhất trên cả hai tập dữ liệu. Thời gian huấn luyện tăng đáng kể, nhưng có thể chấp nhận được nếu độ chính xác là ưu tiên.
7. **C=10, gamma=0.1:**
    - Hiệu suất tốt trên tập huấn luyện, nhưng validation error bắt đầu tăng lên. Có thể là dấu hiệu của overfitting khi mô hình trở nên quá phức tạp.
8. **C=10, gamma=0.5:**
    - Mô hình có hiệu suất tốt nhất trên cả hai tập dữ liệu, nhưng thời gian huấn luyện đã tăng đáng kể. Có thể là một sự đánh đổi giữa hiệu suất và tốc độ.
9. **C=10, gamma=1:**
    - Overfitting rõ ràng trên tập validation khi training error giảm mạnh, nhưng validation error tăng. Thời gian huấn luyện tăng nhanh chóng.

**Tổng kết:**

- Sự lựa chọn tốt nhất có vẻ là C=1 và gamma=1, tuy nhiên, cần cân nhắc giữa hiệu suất và thời gian huấn luyện.
- Cần theo dõi validation error để tránh overfitting khi mô hình trở nên quá phức tạp.
- Có thể cần thực hiện thêm các kỹ thuật tinh chỉnh siêu tham số hoặc sử dụng các mô hình khác để đạt được hiệu suất tốt hơn.

**Nhận xét về số lượng thuộc tính là 5 :**

1. **Giảm chiều dữ liệu:**
    - Việc giảm số lượng thuộc tính giúp giảm chiều dữ liệu, làm cho mô hình huấn luyện nhanh hơn và giảm khả năng overfitting.
    - Tuy nhiên, cũng có thể dẫn đến mất mát thông tin quan trọng nếu số lượng thuộc tính giảm quá nhiều.
2. **Thời gian huấn luyện:**

- Với số lượng thuộc tính ít hơn, thời gian huấn luyện giảm đi, nhưng điều này còn phụ thuộc vào cách thuật toán xử lý việc giảm chiều dữ liệu.

3. **Hiệu suất mô hình:**
   - Việc giảm số lượng thuộc tính có thể ảnh hưởng đến hiệu suất của mô hình. Trong trường hợp này, các giá trị của training và validation error có thể được ảnh hưởng do sự giảm chiều dữ liệu.
4. **Khả năng tổng quát hóa:**
   - Việc giảm số lượng thuộc tính có thể giúp mô hình tổng quát hóa tốt hơn trên dữ liệu mới nếu mô hình không bị quá phức tạp.
5. **Đặc điểm của thuật toán PCA:**
   - Trong trường hợp này, việc sử dụng PCA để giảm chiều dữ liệu có thể dẫn đến việc mất mát một số thông tin quan trọng, nhưng nó giúp giảm nhiễu và giữ lại các đặc trưng quan trọng.

**Nhận xét:**

- Số lượng thuộc tính ít hơn có thể giảm độ phức tạp của mô hình và giúp tránh overfitting.
- Thời gian huấn luyện giảm đáng kể, có thể là lợi ích quan trọng khi có nhu cầu huấn luyện mô hình nhanh chóng.
- Tuy nhiên, cần cân nhắc giữa việc giảm chiều dữ liệu và giữ lại đủ thông tin để mô hình có thể tổng quát hóa tốt trên dữ liệu mới.
- Cần kiểm tra kỹ lưỡng kết quả trên tập validation để đảm bảo rằng mô hình không bị mất mát quá nhiều thông tin và vẫn tổng quát hóa tốt.

## b. 50 đặc trưng

```
C=0.1, gamma=0.1, Training Error: 0.3104, Validation Error: 0.3575, Training Time: 326.18 seconds
C=0.1, gamma=0.5, Training Error: 0.8990, Validation Error: 0.9034, Training Time: 412.18 seconds
C=0.1, gamma=1, Training Error: 0.8991, Validation Error: 0.9034, Training Time: 451.77 seconds
C=1, gamma=0.1, Training Error: 0.0104, Validation Error: 0.1535, Training Time: 270.03 seconds
C=1, gamma=0.5, Training Error: 0.0000, Validation Error: 0.6148, Training Time: 730.88 seconds
C=1, gamma=1, Training Error: 0.0000, Validation Error: 0.7863, Training Time: 736.67 seconds
C=10, gamma=0.1, Training Error: 0.0000, Validation Error: 0.1494, Training Time: 281.20 seconds
C=10, gamma=0.5, Training Error: 0.0000, Validation Error: 0.5800, Training Time: 806.26 seconds
C=10, gamma=1, Training Error: 0.0000, Validation Error: 0.7629, Training Time: 648.86 seconds
```

**Nhận xét từng trường hợp của C và gamma:**

1. **C=0.1, gamma=0.1:**
   - Training Error: 0.3104, Validation Error: 0.3575
   - Thời gian huấn luyện: 326.18 seconds
   - Mô hình có vẻ bị underfitting vì cả training và validation error đều cao.
2. **C=0.1, gamma=0.5:**
   - Training Error: 0.8990, Validation Error: 0.9034
   - Thời gian huấn luyện: 412.18 seconds
   - Mô hình quá phức tạp và không tổng quát hóa tốt trên dữ liệu mới, có dấu hiệu của overfitting.
3. **C=0.1, gamma=1:**

- Training Error: 0.8991, Validation Error: 0.9034
- Thời gian huấn luyện: 451.77 seconds
- Tình trạng tương tự như trường hợp trước, mô hình quá phức tạp và không tổng quát hóa.

4. **C=1, gamma=0.1:**
   - Training Error: 0.0104, Validation Error: 0.1535
   - Thời gian huấn luyện: 270.03 seconds
   - Có vẻ là một cài đặt tốt với hiệu suất tốt trên cả training và validation set.
5. **C=1, gamma=0.5:**
   - Training Error: 0.0000, Validation Error: 0.6148
   - Thời gian huấn luyện: 730.88 seconds
   - Mô hình hoạt động rất tốt trên training set nhưng có dấu hiệu của overfitting trên tập validation.
6. **C=1, gamma=1:**
   - Training Error: 0.0000, Validation Error: 0.7863
   - Thời gian huấn luyện: 736.67 seconds
   - Overfitting rõ ràng khi mô hình không tổng quát hóa trên tập validation.
7. **C=10, gamma=0.1:**
   - Training Error: 0.0000, Validation Error: 0.1494
   - Thời gian huấn luyện: 281.20 seconds
   - Cài đặt tốt với hiệu suất ổn định trên cả hai tập dữ liệu.
8. **C=10, gamma=0.5:**
   - Training Error: 0.0000, Validation Error: 0.5800
   - Thời gian huấn luyện: 806.26 seconds
   - Mô hình có dấu hiệu overfitting trên tập validation.
9. **C=10, gamma=1:**
   - Training Error: 0.0000, Validation Error: 0.7629
   - Thời gian huấn luyện: 648.86 seconds
   - Overfitting rõ ràng khi mô hình không tổng quát hóa trên tập validation.

**Nhận xét:**

- Trong nhiều trường hợp, mô hình có dấu hiệu overfitting, đặc biệt là khi giá trị của gamma tăng lên.
- Một số trường hợp với giá trị C lớn như 10 và gamma nhỏ như 0.1 có vẻ là tốt nhất, với hiệu suất tốt trên cả hai tập dữ liệu và thời gian huấn luyện hợp lý.
- Cần tiếp tục theo dõi và thực hiện kiểm định cẩn thận để đảm bảo mô hình không bị overfitting và vẫn tổng quát hóa tốt trên dữ liệu mới.

Số lượng thuộc tính lớn trong mô hình Gaussian có thể ảnh hưởng đến hiệu suất và tốc độ huấn luyện của mô hình. Dưới đây là một số ảnh hưởng chính:

1. **Tăng chiều dữ liệu:**

- Số lượng thuộc tính lớn có thể dẫn đến tăng chiều dữ liệu, làm cho không gian đặc trưng trở nên lớn và phức tạp hơn. Điều này có thể làm tăng khả năng overfitting, đặc biệt khi dữ liệu huấn luyện có ít mẫu.

2. **Tăng độ phức tạp của mô hình:**
   - Số lượng thuộc tính lớn có thể tạo ra mô hình Gaussian phức tạp hơn, có thể tận dụng được sự phức tạp của dữ liệu. Tuy nhiên, quá mức phức tạp có thể dẫn đến overfitting, nơi mô hình tốt trên dữ liệu huấn luyện nhưng không tổng quát hóa tốt trên dữ liệu mới.
3. **Tăng thời gian huấn luyện:**
   - Số lượng thuộc tính lớn có thể làm tăng đáng kể thời gian huấn luyện của mô hình. Điều này đặc biệt đúng khi áp dụng PCA để giảm chiều dữ liệu, vì quá trình này có thể đòi hỏi nhiều tài nguyên tính toán.
4. **Cần sự cân nhắc khi số lượng thuộc tính lớn:**
   - Với số lượng thuộc tính lớn, cần cân nhắc kỹ lưỡng trong việc lựa chọn các phương pháp giảm chiều dữ liệu hoặc các phương pháp chọn thuộc tính để giảm độ phức tạp của mô hình và tăng khả năng tổng quát hóa.
5. **Tối ưu hóa hyperparameters:**
   - Số lượng thuộc tính lớn có thể ảnh hưởng đến tối ưu hóa các siêu tham số như C và gamma trong mô hình Gaussian. Quá nhiều thuộc tính có thể làm cho quá trình tối ưu hóa trở nên khó khăn hơn.

**Tóm lại:**

- Số lượng thuộc tính lớn có thể mang lại sự phức tạp và hiệu suất mô hình cao hơn, nhưng đồng thời cũng có nguy cơ overfitting và tăng thời gian huấn luyện.
- Quá trình giảm chiều dữ liệu và chọn thuộc tính có thể là cách tiếp cận để giảm ảnh hưởng của số lượng thuộc tính lớn và cải thiện hiệu suất tổng quát hóa của mô hình.

### 3. Chọn hàm dự đoán cuối cùng

Dựa trên độ lỗi thẩm định nhỏ nhất thì ta chọn mô hình Linear Kernel tập vào 50 thuộc tính với siêu tham số C là 1 cho ra được độ lỗi thấp nhất là 14.88% làm hàm dự đoán cuối cùng

# III. Đánh giá SVM

```
With Linear Kerneal. For C=1:
Test Error: 0.1635
-----------------------------------
```

Đây là độ lỗi tập test thu được với mô hình dự đoán cuối cùng.

## 1. Test Error: 0.1635

- Test Error đạt mức thấp, chỉ 16.35%, là một kết quả tích cực và cho thấy mô hình có khả năng dự đoán tốt trên tập kiểm thử.
- Mức độ chính xác cao, đề xuất rằng mô hình có khả năng tổng quát hóa tốt trên dữ liệu mới.

## 2. So sánh với các thử nghiệm trước đó:

- Mô hình hiển nhiên có hiệu suất tốt hơn so với nhiều cấu hình khác đã được thử nghiệm trước đó, đặc biệt là so với các giá trị Test Error lớn (0.9000).
- Sự chọn lựa của Linear Kernel và C=1 đã giúp cải thiện khả năng dự đoán của mô hình.

```
Test Error 1: 0.2863
Test Error 2: 0.2861
Test Error 3: 0.2864
Test Error 4: 0.1641
Test Error 5: 0.1635
Test Error 6: 0.1633
Test Error 7: 0.2741
Test Error 8: 0.2644
Test Error 9: 0.2632
Test Error 10: 0.2614
Test Error 11: 0.2523
Test Error 12: 0.2520
Test Error 13: 0.2539
Test Error 14: 0.2514
Test Error 15: 0.2584
Test Error 16: 0.3655
Test Error 17: 0.9000
Test Error 18: 0.9000
Test Error 19: 0.1588
Test Error 20: 0.6173
Test Error 21: 0.7852
Test Error 22: 0.1565
Test Error 23: 0.5833
Test Error 24: 0.7628
```

Độ lỗi tập test thu được ở các mô hình huấn luyện trước lần lượt là:
1: Linear 5 thuộc tính C = 0.1
2: Linear 5 thuộc tính C = 1
3: Linear 5 thuộc tính C = 10

4: Linear 50 thuộc tính C = 0.1
5: Linear 50 thuộc tính C = 1 (mô hình được chọn)
6: Linear 50 thuộc tính C = 10

7: RBF 5 thuộc tính C = 0.1 y = 0.1
8: RBF 5 thuộc tính C = 0.1 y = 0.5
9: RBF 5 thuộc tính C = 0.1 y = 1

10: RBF 5 thuộc tính C = 1 y = 0.1
11: RBF 5 thuộc tính C = 1 y = 0.5
12: RBF 5 thuộc tính C = 1 y = 1

13: RBF 5 thuộc tính C = 10 y = 0.1
14: RBF 5 thuộc tính C = 10 y = 0.5
15: RBF 5 thuộc tính C = 10 y = 1

16: RBF 50 thuộc tính C = 0.1 y = 0.1
17: RBF 50 thuộc tính C = 0.1 y = 0.5
18: RBF 50 thuộc tính C = 0.1 y = 1

19: RBF 50 thuộc tính C = 1 y = 0.1
20: RBF 50 thuộc tính C = 1 y = 0.5
21: RBF 50 thuộc tính C = 1 y = 1

22: RBF 50 thuộc tính C = 10 y = 0.1

> 23: RBF 50 thuộc tính C = 10 y = 0.5
> 24: RBF 50 thuộc tính C = 10 y = 1

Nhận xét
- Ta thấy được độ lỗi tập test thấp hơn hầu hết các trường hợp dự đoán thu được.
- Tuy nhiên vẫn có vài trường hợp thu được giá trị nhỏ hơn nữa là 6, 19, 22. Lần lượt là: Linear 50 thuộc tính C = 10, RBF 50 thuộc tính C = 1 y = 0.1, RBF 50 thuộc tính C = 10 y = 0.1.

## 3. Độ biến động trong Test Error:

- Các giá trị Test Error trước đó có sự biến động lớn, và mô hình với Linear Kernel và C=1 giảm điểm Test Error đáng kể so với các trường hợp trước đó.

## 4. Tiềm năng cải thiện và cần kiểm tra thêm:

- Mặc dù kết quả là khả quan, nhưng cần kiểm tra kỹ lưỡng và thử nghiệm để đảm bảo khả năng tổng quát hóa của mô hình.
- Cần kiểm tra sự ổn định của mô hình trên nhiều lượt kiểm thử và nhiều tập dữ liệu để đảm bảo kết quả này không chỉ là do sự may mắn.

## 5. Tổng kết:

- Sự lựa chọn của Linear Kernel và C = 1 có vẻ là một quyết định tốt, mang lại kết quả Test Error thấp và cải thiện hiệu suất so với các thử nghiệm trước đó. Tuy nhiên, tiếp tục kiểm tra và thử nghiệm là quan trọng để đảm bảo sự ổn định và khả năng tổng quát hóa của mô hình.