**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO MACHINE LEARNING**

**NHẬN DIỆN TRÁI CÂY**

**SVTH: VÕ MINH HIẾU 17110136**

**LÊ MINH TIẾN 17110236**

**GVPT: NGUYỄN THIÊN BẢO**

**TP. Hồ Chí Minh, tháng 12 năm 2020**

MỤC LỤC

[LỜI MỞ ĐẦU 1](#_Toc59787735)

[Danh mục các hình 2](#_Toc59787736)

[Danh mục các bảng 3](#_Toc59787737)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 4](#_Toc59787738)

[1.1. Giới thiệu đề tài 4](#_Toc59787739)

[1.2.1. Lý do, mục đích chọn đề tài 4](#_Toc59787740)

[1.2.2. Mục tiêu 4](#_Toc59787741)

[CHƯƠNG 2: NỘI DUNG 5](#_Toc59787742)

[1. Giới thiệu bộ dữ liệu 5](#_Toc59787743)

[2. Thư viện sử dụng 6](#_Toc59787744)

[2. Thuật toán Support Vector Machine (SVM) 7](#_Toc59787745)

[3. Xử lý dữ liệu 8](#_Toc59787746)

[3.1. Import dữ liệu 8](#_Toc59787747)

[3.2. Reshape dữ liệu 8](#_Toc59787748)

[3.3. Feature scaling 8](#_Toc59787749)

[4. Tạo model bằng SVM 8](#_Toc59787750)

[4.1. Trainning data 8](#_Toc59787751)

[4.2. Đánh giá model 9](#_Toc59787752)

[4.2.2. Accuracy score trên tập test set 9](#_Toc59787753)

[4.2.3. Dự đoán thử 11](#_Toc59787754)

[4.3. Phân tích lỗi 11](#_Toc59787755)

[CHƯƠNG 3: KẾT LUẬN 18](#_Toc59787756)

[PHỤ LỤC 19](#_Toc59787757)

[TÀI LIỆU THAM KHẢO 20](#_Toc59787758)

LỜI MỞ ĐẦU

Nhóm xin chân thành cảm ơn sự hướng dẫn tận tình của thầy …, cả về chuyên môn lẫn tinh thần, đã giúp đỡ nhóm trong quá trình thực hiện project.

Với mỗi sinh viên nói chung, việc tích lũy kiến thức qua giáo trình, bài giảng trên lớp là rất quan trọng và cần thiết, tuy nhiên sinh viên lại chưa có nhiều cơ hội để áp dụng các kiến thức đó vào project. Cho nên, việc thực hiện Đồ án 3 chính là cơ hội cho nhóm được thử sức mình tạo ra những project thực tế, được hoạt động theo nhóm một cách chuyên nghiệp dưới sự hướng dẫn của các giảng viên có dày dặn kinh nghiệm trong lĩnh vực trí tuệ nhân tạo, cụ thể là machine learning.

Vì kiến thức và kỹ thuật còn hạn hẹp, nên trong quá trình thực hiện có sảy ra sai sót, rất mong thầy góp ý để nhóm có thể hoàn thiện project tốt hơn.

Được sự hướng dẫn tận tình của thầy, nhóm xin chân thành cảm ơn!

Danh mục các hình

**No table of contents entries found.**

Danh mục các bảng

**No table of contents entries found.**

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu đề tài

1.2.1. Lý do, mục đích chọn đề tài

1.2.2. Mục tiêu

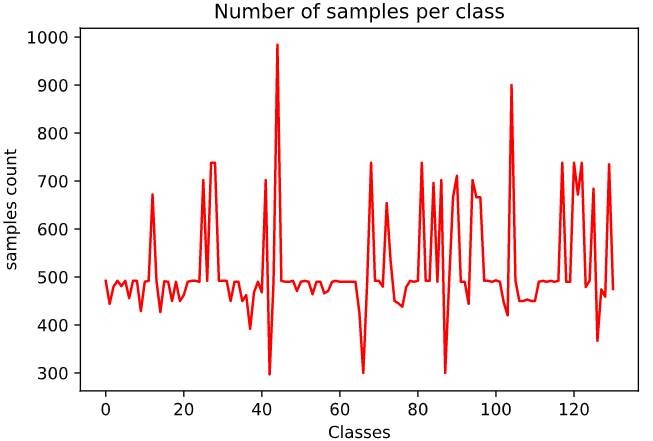
Mục tiêu nhóm đặt ra là nhóm phải biết được quy trình thực hiện một dự án machine learning, hiểu về cách hoạt động của thuật toán, biết được các kỹ thuật xử lý dữ liệu, lựa chọn model, đánh giá model và phân tích model.

Mục tiêu đề ra là model xây dựng được phải có accuracy score trên 0.85.

CHƯƠNG 2: NỘI DUNG

1. Giới thiệu bộ dữ liệu

Biểu đồ phân bố dữ liệu:



2. Thư viện sử dụng

2.1. Thư viện Scikit-learn

Scikit-learn (Sklearn) là thư viện mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling gồm: classification, regression, clustering, và dimensionality reduction. [2]

Project lần này của nhóm thuộc về classification.

2.2.

2.3. Thư viện numpy

Đây là thư viện giúp xử lý tính toán trên mảng có kích thước lớn và nhiều chiều, được sử dụng xuyên suốt project.

2.4. Thư viện matplot

Matplotlib là một thư viện sử dụng để vẽ các đồ thị trong Python, chính vì vậy nó là thư viện cực phổ biến của Python. Nhóm sử dụng matplot để vẽ lên các đồ thị và confusion matrix.

2.5. Thư viện joblib

Đây là thư viện giúp nhóm lưu các model tìm được và các thông số tính toán.

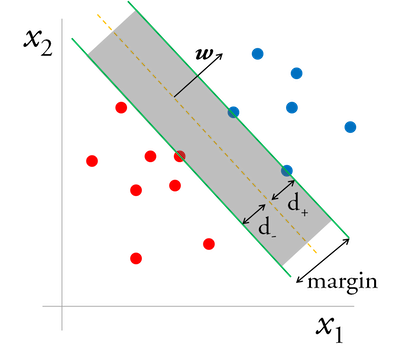
2. Thuật toán Support Vector Machine (SVM)

SVM là một thuật toán thuộc nhóm Supervised Learning (Học có giám sát) dùng để phân chia dữ liệu (Classification) thành các nhóm riêng biệt.

SVM là một thuật toán rất mạnh, chạy rất nhanh, trước khi deep learning ra đời thì nó là một trong những thuật toán tốt nhất. Trong một số trường hợp SVM nó còn tốt hơn deep learning vì deep learnining sinh ra model rất phức tạp và nó cần phải có một lượng lớn dữ liệu nếu không nó chạy không tốt.

2.1. Margin

Xét trong không gian hai chiều, Margin là khoảng cách giữa siêu phẳng đến 2 điểm dữ liệu gần nhất tương ứng với 2 phân lớp.



SVM cố gắng tối ưu thuật toán bằng các tìm cách maximize giá trị margin này, từ đó tìm ra siêu phẳng đẹp nhất để phân 2 lớp dữ liệu.

2.2. Support vectors

Bài toán của chúng ta trở thành tìm ra 2 đường biên của 2 lớp dữ liệu (ở hình bên trên là 2 đường xanh lá cây) sao cho khoảng cách giữa 2 đường này là lớn nhất.

Đường biên của lớp xanh sẽ đi qua một (hoặc một vài) điểm xanh.

Đường biên của lớp đỏ sẽ đi qua một (hoặc một vài) điểm đỏ.

Các điểm xanh, đỏ nằm trên 2 đường biên được gọi là các support vector, vì chúng có nhiệm vụ support để tìm ra siêu phẳng.

Đó cũng là lý do của tên gọi thuật toán Support Vector Machine.

2.3. Soft Margin

Margin được chia thành 2 loại đó là Soft margin và Hard margin.

Để tránh overfitting, nhiều khi để muốn có margin cao, ta chấp nhận việc một vài data có thể không được chia chính xác (ví dụ như 1 bóng xanh bị lọt sang vùng của bóng đỏ). Data này được gọi là nhiễu.

* Margin trong trường hợp này gọi là Soft Margin.

Hard Margin ám chỉ việc tìm được Margin mà không nhiễu (tất cả các data đều thoả mãn sự phân chia). Với các bái toán thực tế, việc tìm được Hard Margin nhiều khi là bất khả thi, vì thế việc chấp nhận sai lệch ở một mức độ chấp nhận được là vô cùng cần thiết.

Trong cài đặt SVM, người ta giới thiệu tham số CC với quy ước:

- C là một số dương vô cùng: đồng nghĩa với việc không cho phép sai lệch, đồng nghĩa với Hard Margin.

- C lớn: cho phép sai lệch nhỏ, thu được margin nhỏ.

- C nhỏ: cho phép sai lệch lớn, thu được margin lớn.

* Tuỳ bài toán cụ thể mà ta cần điểu chỉnh tham số C này để thu được kết quả tốt nhất.

3. Xử lý dữ liệu

3.1. Import dữ liệu

3.2. Reshape dữ liệu

3.3. Feature scaling

4. Tạo model bằng SVM

4.1. Trainning data

|  |
| --- |
|  |

Các parameters:

- decision\_function\_shape: Mặc định SVC sẽ train với option là “ova” tức là one vs all, nhưng nhóm chỉ định option “ovo” để cải thiện tốc độ training. Số lượng class là 131, nên tổng số binary class phải dùng là: n \* (n – 1)/2 = 8515 class. Tuy số lượng class rất nhiều, nhưng đối với one vs all, ta cần phải nạp toàn bộ training data vào Ram mới thực hiện training, còn đối với one vs one, ta chỉ cần nập từng class vào, và do tính độc lập (one vs one) nên có thể thực hiện tính toán song song hiệu quả hơn.

4.2. Đánh giá model

Các điểm đánh giá:

- Accuracy: đây là điểm số hay được sử dụng nhất, nó được tính bằng tỉ lệ giữa số lượng sample dự đoán đúng với tổng số lượng dự đoán trong bộ dữ liệu dùng để kiểm thử, có giá trị trong khoảng từ [0, 1], càng cao càng tốt.

- Precision: được xem như độ chính xác, có giá trị trong khoảng từ [0, 1], càng cao càng tốt, được tính bằng công thức:

|  |
| --- |
|  |

Các ký hiệu được mô tả như sau:

|  |  |  |
| --- | --- | --- |
|  | Dự đoán là Positive | Dự đoán là Negative |
| Actual: Positive | True Positive (TP) | False Newgative (FN) |
| Actual: Negative | False Positive (FP) | True Negative (TN) |

- Recall: được xem như độ bao phủ, có giá trị trong khoảng từ [0, 1], càng cao càng tốt, được tính bằng công thức:

|  |
| --- |
|  |

4.2.1. Accuracy score với cross validation

|  |
| --- |
|  |

Cross validation score với 5 folds cho ra kết quả chưa được tốt cho lắm, 2 folds đầu cho ra tương đối, nhưng 3 folds còn lại cho ra kết quả khá kém.

4.2.2. Accuracy score trên tập test set

|  |
| --- |
|  |

Ta thấy accuracy score trên tập test set cũng khá cao, đã đạt yêu cầu, nhưng vẫn có thể được cải thiện.

4.2.2. Precision và recall và f1 scores

|  |
| --- |
|  |

4.2.3. Dự đoán thử

4.3. Phân tích lỗi

4.3.1. Confusion matrix

5.3.1. Tunning tham số C.

Nhóm thực hiện tunning tham số C sao cho model cho ra kết quả tốt nhất.

Các parameter candidates là: C = {0.1, 1, 10, 17, 25, 37, 50, 75, 100}. Vì thời gian tunning quá lâu, nên nhóm chia từng part để tunning, mỗi lần chỉ tunning 2 giá trị.

- Lần 1:

|  |
| --- |
|  |

Kết quả cho ra:

|  |
| --- |
|  |

- Lần 2:

|  |
| --- |
|  |

Ta thấy với C = 10 thì Best score tăng lên rõ rệt, ta tiếp tục tăng để xem kết quả có còn được cải thiện hay không.

- Lần 3:

|  |
| --- |
|  |

Ta thấy kết quả không tăng lên nhiều, có thể ta đã tăng lên quá nhiều.

- Lần 4:

|  |
| --- |
|  |

- Lần 5:

|  |
| --- |
|  |

=> Với C bằng 17 thì model cho ra score tốt nhất, nên ta sẽ trainning với tham số C = 17 để có được model tốt hơn.

5.3.2. Tunning tham số Gama.

5.4. Trainning lại model với best parameters

|  |
| --- |
|  |

Sau khi tìm được tham số tốt nhất cho model thì nhóm tiến hành trainning lại với 100% dữ liệu training, hi vọng rằng nó sẽ cho ra kết quả tốt.

R2Score và Root Mean Square Error

|  |
| --- |
|  |

Ta thấy kết quả cho ra tốt hơn khá nhiều so với lúc đầu, R2 score tăng lên tới 0.89 so với ban đầu là 0.86.

Cross validation accuracy

|  |
| --- |
|  |

- Kết quả thu được:

|  |
| --- |
|  |

Accuracy trên tập testing data

|  |
| --- |
|  |

- Kết quả thu được:

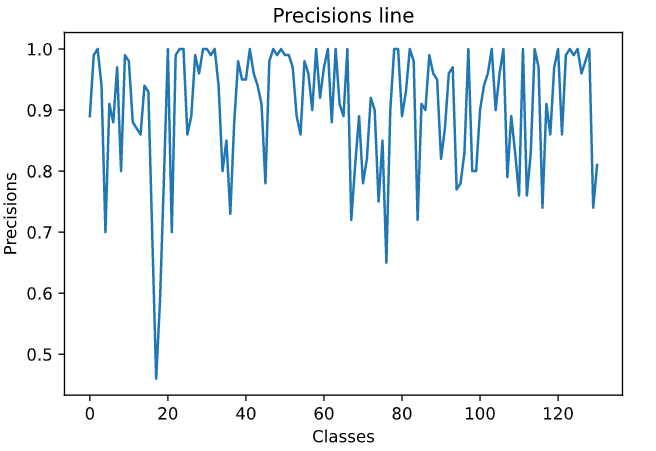
|  |
| --- |
|  |

Precison, recall và f1 score

|  |
| --- |
|  |

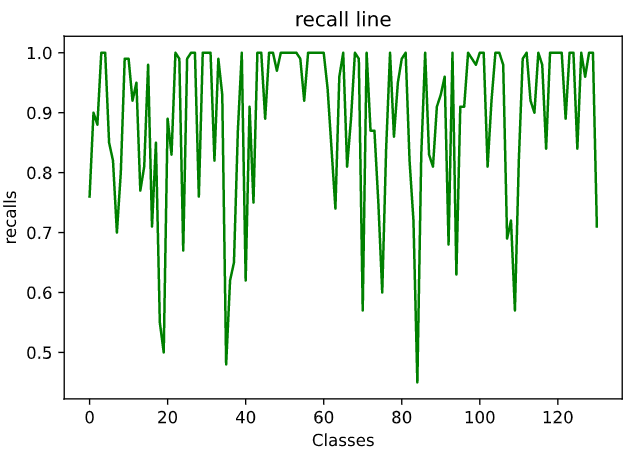
Biểu đồ thể hiện:

Precisons:

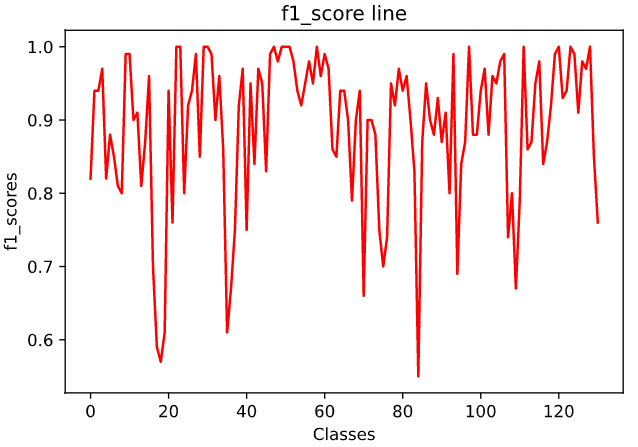


Ta thấy

Recalls:



F1 scores:



CHƯƠNG 3: KẾT LUẬN

Với mục tiêu đề ra từ đầu là xây dựng một ứng dụng dự đoán ký tự sử dụng machine learning, nhóm đã hoàn thành được khoảng 90%. Accuracy của model tìm đạt trên 0.89, đạt được mục tiêu đề ra ban đầu là 0.85.

Sau khi hoàn thành project, nhóm thấy mình đã nắm vững kiến thức về machine learning nhiều hơn, biết cách xử lý dữ liệu sao cho hợp lý, có khả năng đánh giá model qua các điểm đánh giá, biết phân tích lỗi và tối ưu model để hiệu suất của model tăng lên.

PHỤ LỤC

TÀI LIỆU THAM KHẢO