ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

ĐỒ ÁN CUỐI KỲ

Môn: **NHẬP MÔN HỌC MÁY**

Lớp: **CQ2017/21**

Năm học: **2019 - 2020**

Thông tin nhóm

Trần Minh Trí – 1712834

Nguyễn Nhật Trường – 1712852

I. Quá trình làm việc nhóm

Thời gian làm việc	Nội dung thực hiện
11 – 18/8/2020	Tìm hiểu lý thuyết liên quan đến đồ án (SVM, kernel trick, ...)
19 – 20/8/2020	Huấn luyện SVM dùng linear kernel.
21/8/2020 22 – 23/8/2020 24/8/2020	Huấn luyện SVM dùng RBF kernel: - Huấn luyện lần 1. - Huấn luyện lần 2. - Huấn luyện lần 3.
25 – 26/8/2020	Kiểm tra nội dung, kết quả chạy và hoàn thành báo cáo.

II. Nội dung đồ án

1) Huấn luyện SVM

Các model được huấn luyện trên Jupyter Notebook, sử dụng chủ yếu thư viện **sklearn** phiên bản **0.23.2**. Ngoài ra, để tiết kiệm thời gian và hạn chế sự cố, các model sẽ được lưu lại sau khi huấn luyện (Để đọc và load model lại từ file cần sử dụng cùng phiên bản **sklearn** để tránh lỗi, ở đây sử dụng phiên bản **0.23.2)**

Ta sẽ so sánh các mô hình được huấn luyện dựa theo score trên tập validation, với:

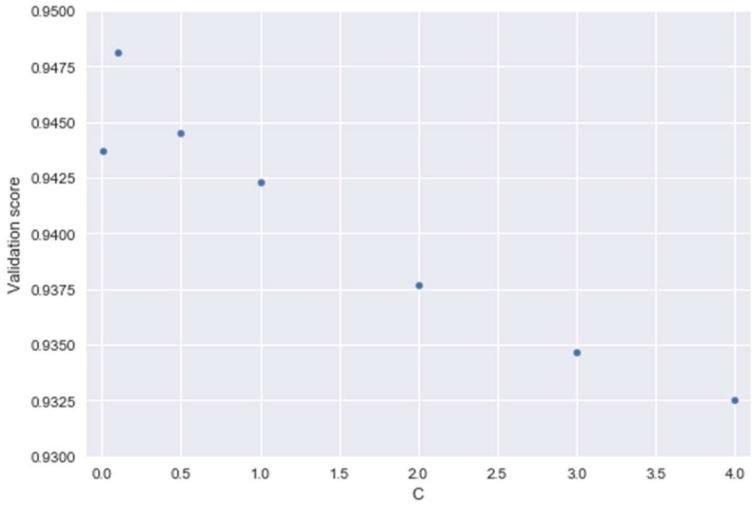
$$\textit{độ lỗi} = 1 - \textit{score}$$

Như vậy, hàm dự đoán cuối cùng có độ lỗi trên tập validation nhỏ nhất ⇨ **hàm dự đoán cuối cùng có score trên tập validation cao nhất**.

a) Linear Kernel

Sau khi huấn luyện mô hình với nhiều giá trị siêu tham số C, kết quả score trên tập validation có giá trị như sau:

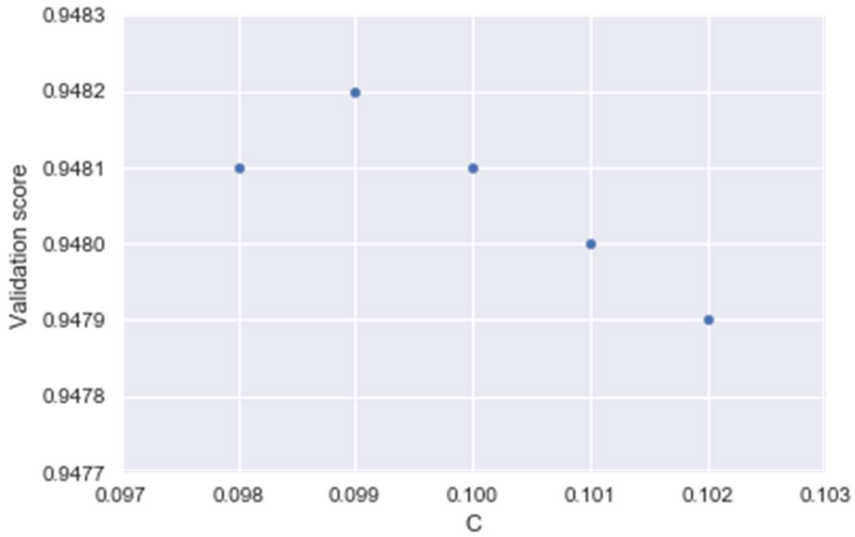
1e-6	1e-4	1e-3	0.001	0.01	0.1	0.5
0.1064	0.6053	0.8971	0.9309	0.9437	0.9481	0.9445
1	2	3	4	5	10	20
0.9423	0.9377	0.9347	0.9325	0.9306	0.9284	0.9264



Mô hình có score cao nhất (**0.9481**) là mô hình với siêu tham số **C = 0.1**. Ta tiếp tục xây dựng các mô hình tương tự với C ở khoảng **0.1** để tìm giá trị tốt nhất.

Kết quả thu được:

0.09	0.098	0.099	0.1	0.101	0.102	0.11
0.948	0.9481	0.9482	0.9481	0.948	0.9479	0.9477



Như vậy, với linear kernel, ta chọn model cuối cùng có **C = 0.099**, với score trên tập validation là **0.9482**.

Bình luận về kết quả:

- Khi tăng siêu tham số C càng cao Eval tăng dần do giảm các điểm sai, margin của mặt siêu phẳng giảm
- Còn khi C giảm (từ 0.099) đến giá trị rất bé, Eval tăng rất nhanh do chấp nhận càng nhiều điểm classified sai

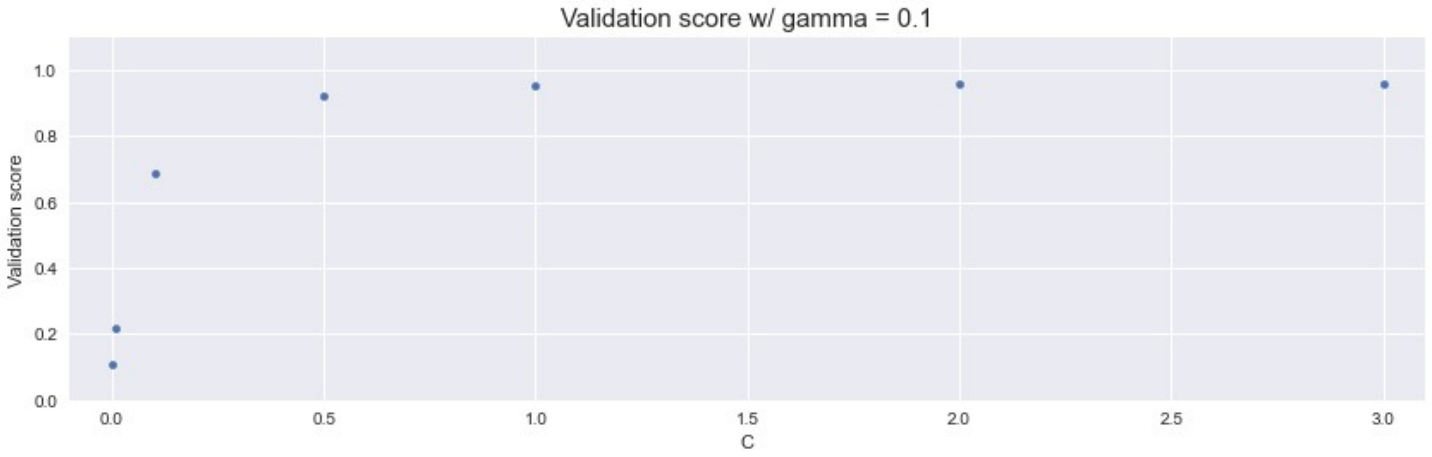
  ➔ Phù hợp với lý thuyết

b) RBF Kernel

Chọn ngẫu nhiên ban đầu giá trị tĩnh $C = 0.1$ và $\gamma = 0.1$, và chạy lần lượt tham số còn lại để tìm kết quả tối ta có kết quả như sau:

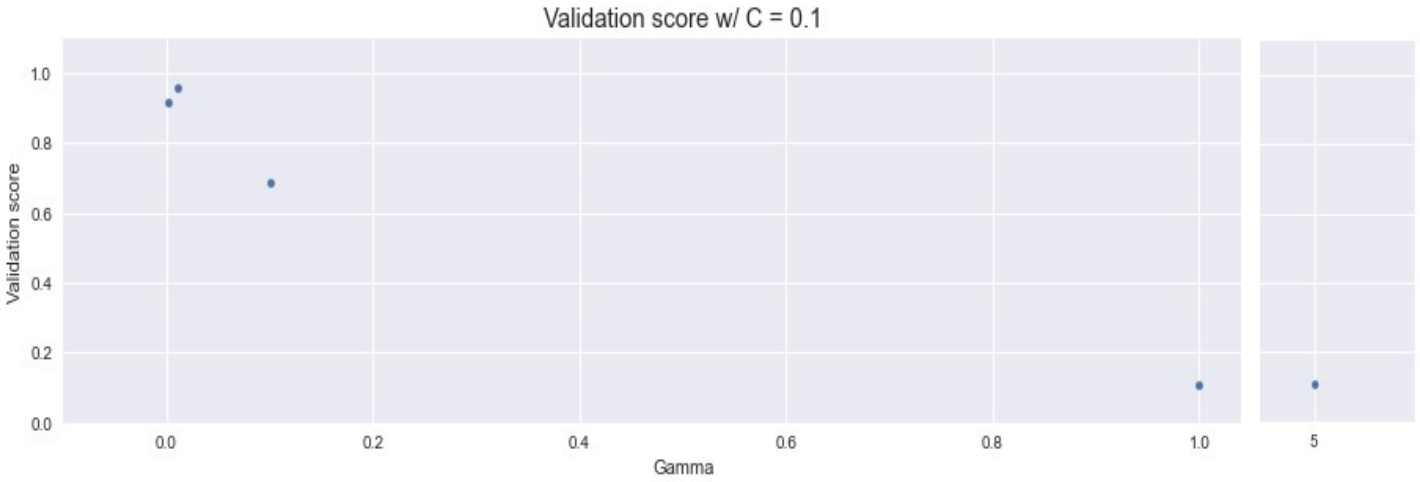
Lần 1: **γ** *tĩnh, C có các giá trị:*

0.001	0.01	0.1	0.5	1	2	3
0.1064	0.2172	0.6875	0.9229	0.9552	0.9566	0.9566



Lần 1: ***C*** *tĩnh, gamma có các giá trị*

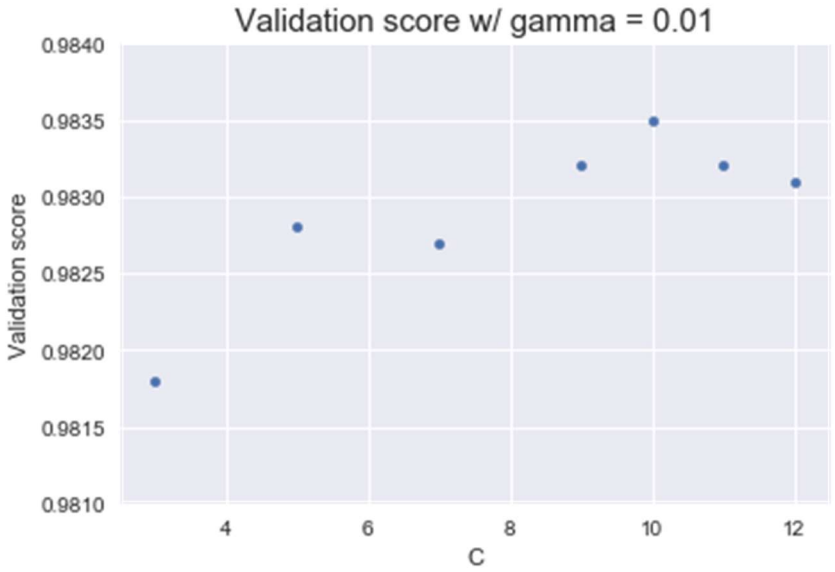
0.001	0.01	0.1	1	5	10
0.9139	0.9578	0.6875	0.1064	0.1064	0.1064



Ta thấy $\gamma \approx 0.01$, $C > 3$, cần tiếp tục tìm C

Ta giữ $\gamma = 0.01$ và thử nghiệm C tăng cao dần như sau:

3	5	7	9	10	11	12
0.9818	0.9828	0.9827	0.9832	0.9835	0.9832	0.9831



Vậy, thấy được $C \approx 10$, $\gamma \approx 0.01$.

Dựa vào kết quả trên,

Lần 2

C \ γ	0.0106	0.0104	***0.0102***	***0.0100***	0.0098	0.0096	0.0094
11	0.9831	0.9831	0.9832	0.9832	0.9832	0.9831	0.9833
10.05	0.9832	0.9832	0.9834	0.9834	0.9832	0.9831	0.9832
10	0.9832	0.9832	0.9835	0.9835	0.9833	0.9830	0.9832
9.95	0.9832	0.9832	0.9835	0.9835	0.9832	0.9829	0.9831
9	0.9835	0.9834	0.9835	0.9832	0.9830	0.9829	0.9830

Lần 3

C \ γ	0.01020	0.01018	0.01014	0.01010	0.01008	0.01004	0.01000
10.01	0.9834	0.9834	0.9833	0.9834	0.9835	0.9835	0.9835
10	0.9835	0.9834	0.9834	0.9833	0.9835	0.9835	0.9835
9.99	0.9835	0.9834	0.9834	0.9835	0.9835	0.9835	0.9835
9.98	0.9834	0.9835	0.9834	0.9835	0.9835	0.9835	0.9835

Chọn **C = 10, γ = 0.01** *(giá trị ít phức tạp)* với score trên tập validation là **0.9835**.

Bình luận về kết quả:

- Ở dữ liệu này, RBF kernel cho Eval tối ưu khi γ nhỏ và C lớn. Vậy hyperplane đơn giản (γ nhỏ), với hyperplane đơn giản không thể hiện hết độ phức tạp dữ liệu nên sẽ có nhiều điểm sai, nên ở đây C lớn

c) Hàm dự đoán cuối cùng

Score trên tập validation ở mô hình sử dụng RBF (**0.9835**) lớn hơn ở mô hình Linear (**0.9482**), cho nên, ta chọn **mô hình RBF**.

2) Đánh giá SVM

- Mô hình được chọn có độ lỗi trên tập test khoảng **1.8%**
- Kém hơn so với mô hình đã thực hiện trước của người khác (1.4%)
- Lý do có thể vì chưa tối ưu tốt nhất siêu tham số. Chia đoạn ở **C = 10, γ = 0.01** chưa đủ nhỏ để có thể tìm thấy tham số tối ưu hơn.

Độ chính xác model đã chọn trên tập test:

```
score = clf_rbf_final.score(test_X, test_Y)
score
```

```
0.982
```

```
print("Test error rate (%): ", (1-score)*100)
```

```
Test error rate (%):  1.8000000000000016
```