TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện Công nghệ Thông tin và Truyền thông

BÁO CÁO BÀI TẬP LỚN MÔN HỌC
NHẬP MÔN HỌC MÁY VÀ KHAI PHÁ DỮ LIỆU

ĐỀ TÀI

Thử nghiệm và đánh giá các thuật toán học máy cho bài toán dự đoán rating phim trung bình.

Giáo viên hướng dẫn: PGS.TS Thân Quang Khoát

Sinh viên thực hiện:	Bùi Việt Dũng	20173045
	Lê Văn Linh	20173235
	Bùi Minh Tuấn	20170121
	Nguyễn Thức Quang Hưng	20173162

Hà Nội, tháng 12 năm 2020

Mục lục

1. Mô tả bài toán

Ngày nay, đời sống giải trí của con người ngày càng phong phú với nhu cầu ngày một cao hơn. Trong đó, điện ảnh là một lĩnh vực không thể thiếu và đang mang lại một nguồn doanh thu khổng lồ cho các nhà sản xuất.

Trong lĩnh vực học máy, bài toán dự đoán rating phim là một chủ đề thú vị và được nhiều người quan tâm. Trong nội dung bài tập lớn, nhóm chúng em lựa chọn đề tài thử nghiệm và đánh giá một số thuật toán học máy trong bài toán dự đoán ratings trung bình của người dùng cho một bộ phim.

Bài toán cụ thể như sau:

Đầu vào: Một số thông tin về bộ phim như: tên, đạo diễn, diễn viên, hãng sản xuất, nội dung chính, thể loại, một số từ khoá chính, …

Đầu ra:

- Với bài toán phân lớp (classification): dự đoán lớp tương ứng với rating từ 0 đến 10 của bộ phim (11 lớp).

- Với bài toán hồi quy (regression): một số thực trong đoạn [0, 10] tương ứng với rating của bộ phim.

2. Tập dữ liệu huấn luyện

Dữ liệu được lấy từ bộ tmdb_5000 gồm 2 file là tmdb_5000_credits.csv và tmdb_5000_movies.csv. Sau khi lọc thông tin thừa trích xuất thông tin nhận thấy có các trường dữ liệu quan trọng có thể ảnh hưởng đến điểm đánh giá trung bình của bộ phim là:

- Cast, crew, production_companies, keywords, genres

Các trường dữ liệu này đều đang ở dạng danh sách cần phải chuyển qua dạng số.

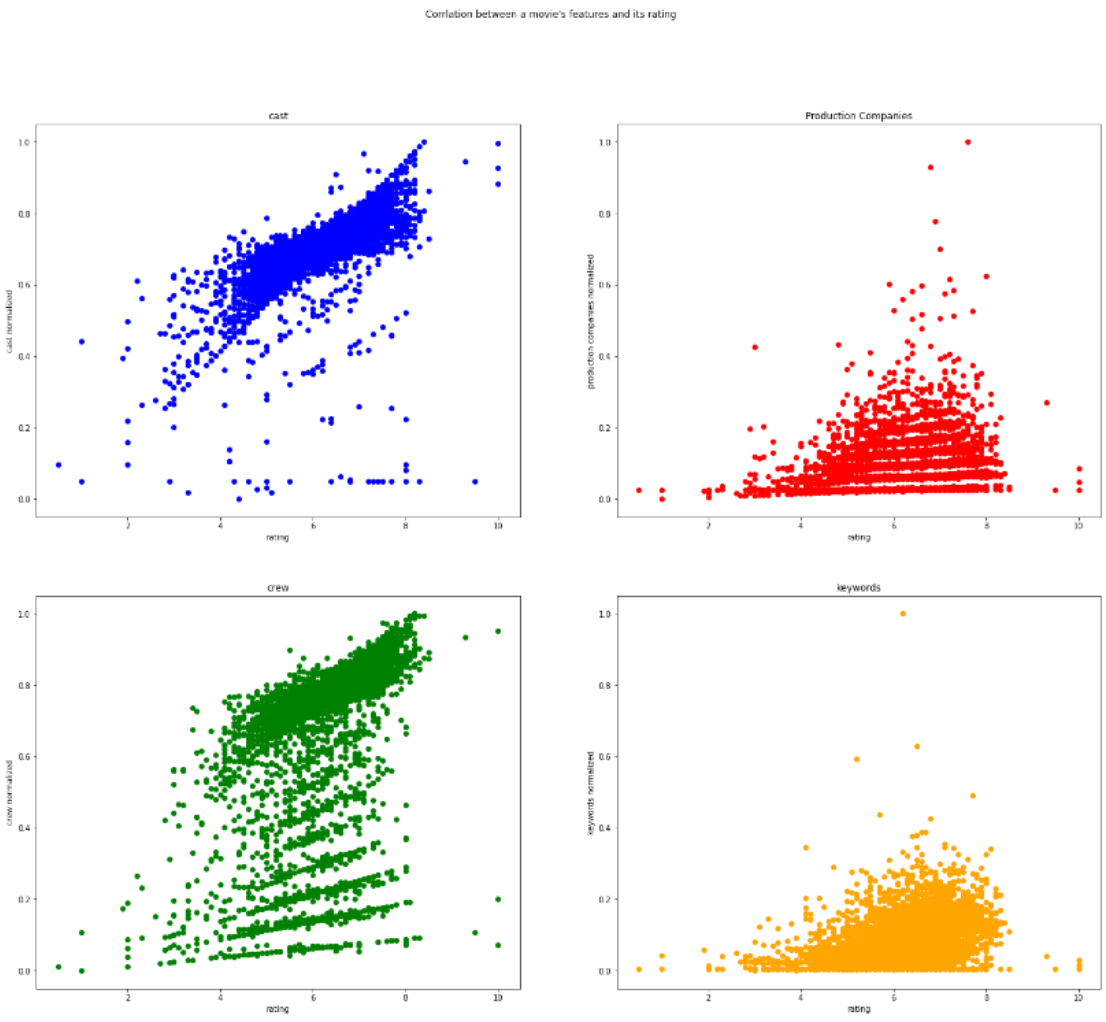
Cách 1:

Liệt kê các phim mà các phần tử trong trường xuất hiện. Giá trị của phần tử sẽ được tính bằng giá trị trung bình của điểm đánh giá của phim mà phần tử đó được xuất hiện. Sau đó xử dụng phương thức chuẩn hóa là MinMaxScaler của thư viện sklearn với công thức:

X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))

X_scaled = X_std * (max - min) + min

Ta thu được phân phối như sau:

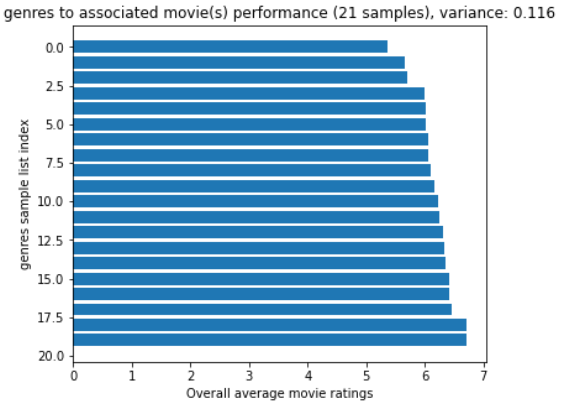


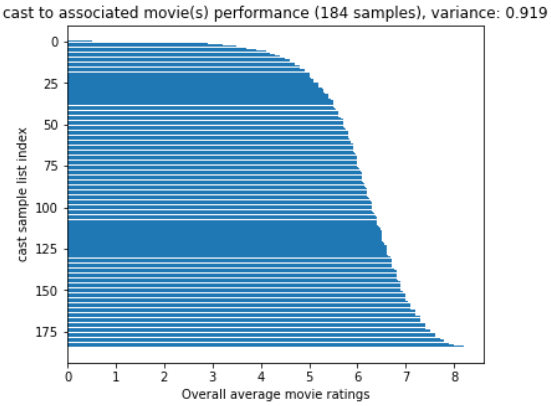
Nhận thấy phân phối ra các trường có sự phân phối không đồng đều. Khi cho vào mô hình BayesianRidge ra được kết quả dự đoán thấp là 0.47.

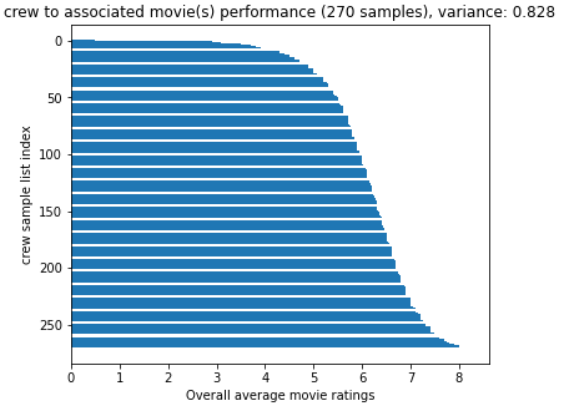
Cách 2:

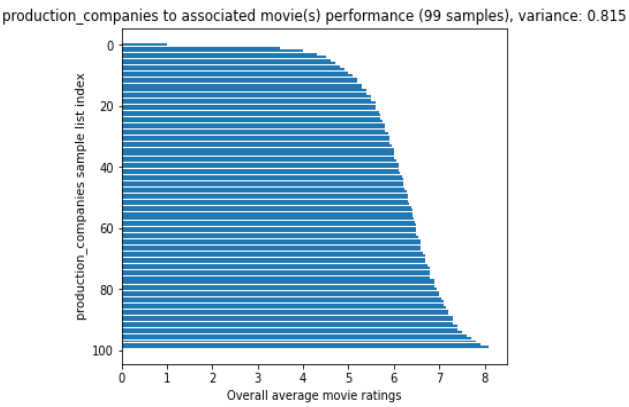
Đánh giá giá trị của 1 thuộc tính bằng trung bình điểm đánh giá của phim. Sắp xếp các thuộc tính đó theo giá tăng dần. Giá trị của điểm đó sẽ bằng giá trị trung bình của các tham số.

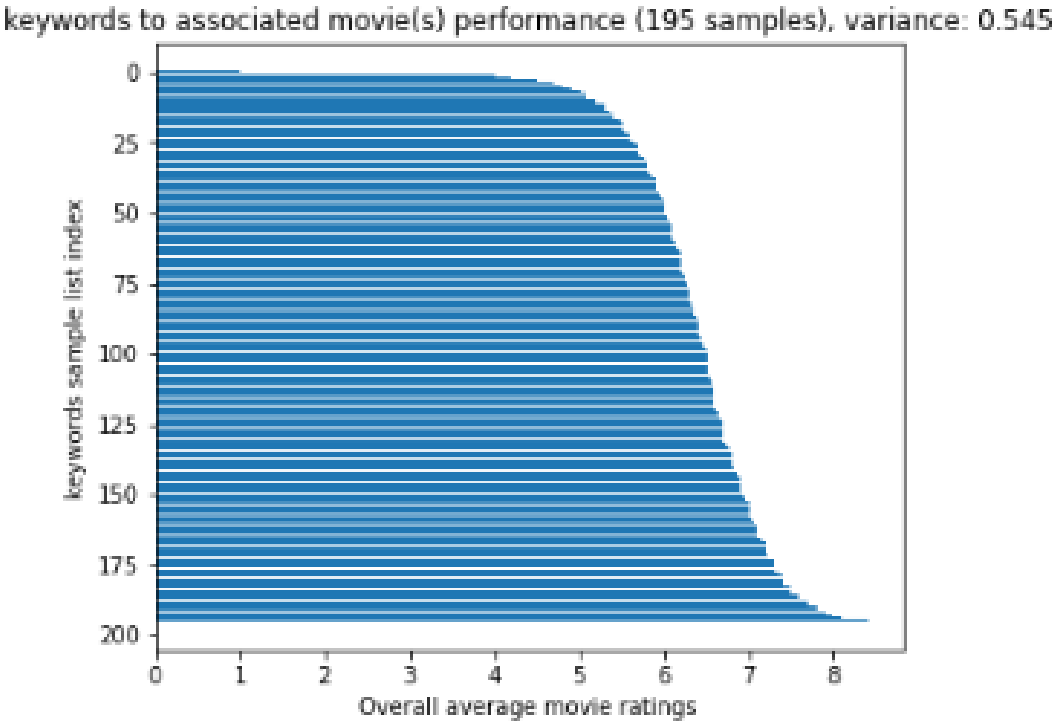
Phân phối của các thuộc tính theo điểm đánh giá trung bình:







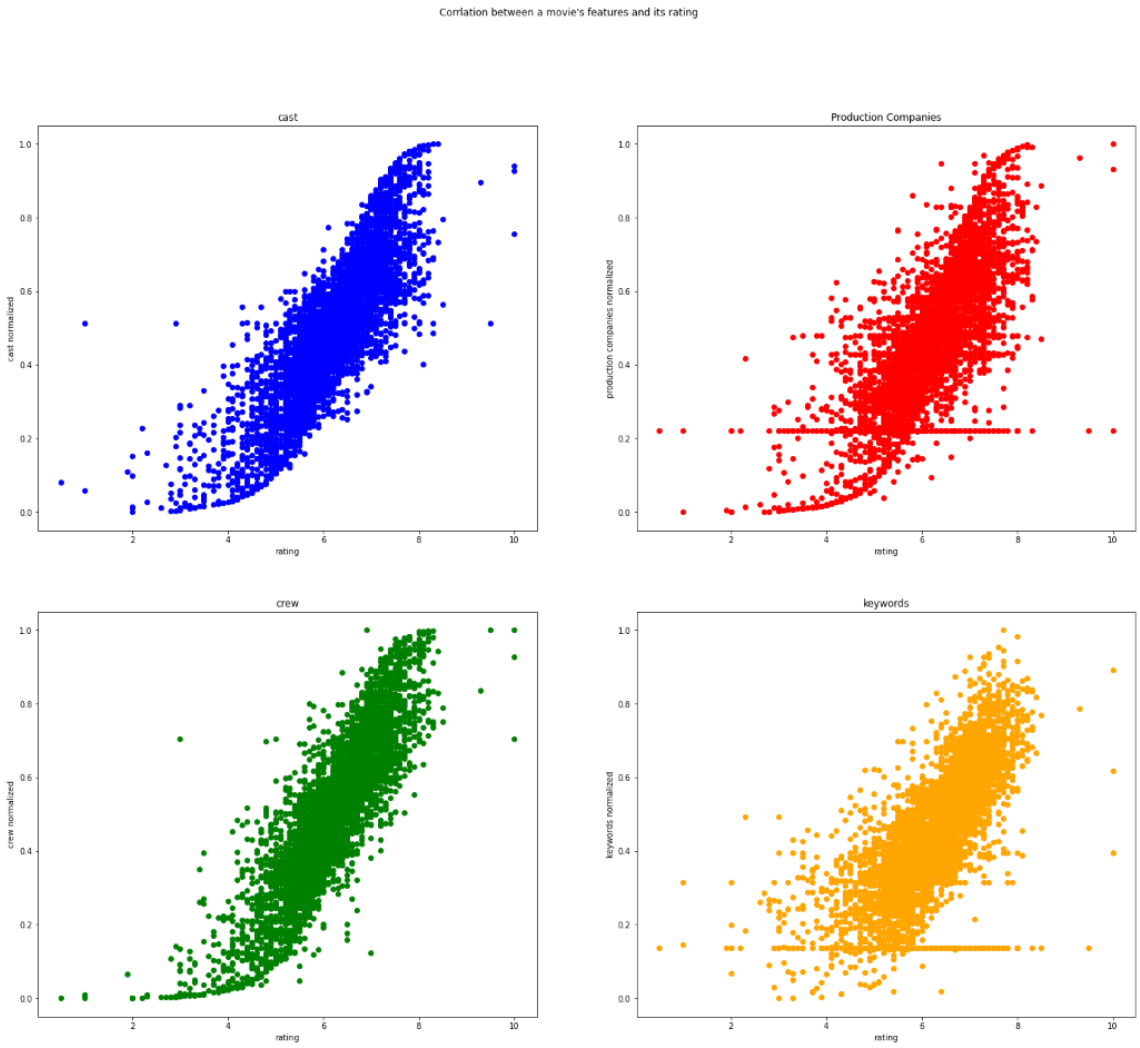




Nhận xét: các ngoại trừ trường genres cho ra phân bổ khá đều nhau ở mỗi mức rating nên không thu được nhiều ý nghĩa thì các trường khác dều cho phân bổ theo hướng tăng dần nên ta loại bỏ trường genres

Sau đó áp dụng MinMaxScaler để chuẩn hóa.

Kết quả ta được phân phối như sau:



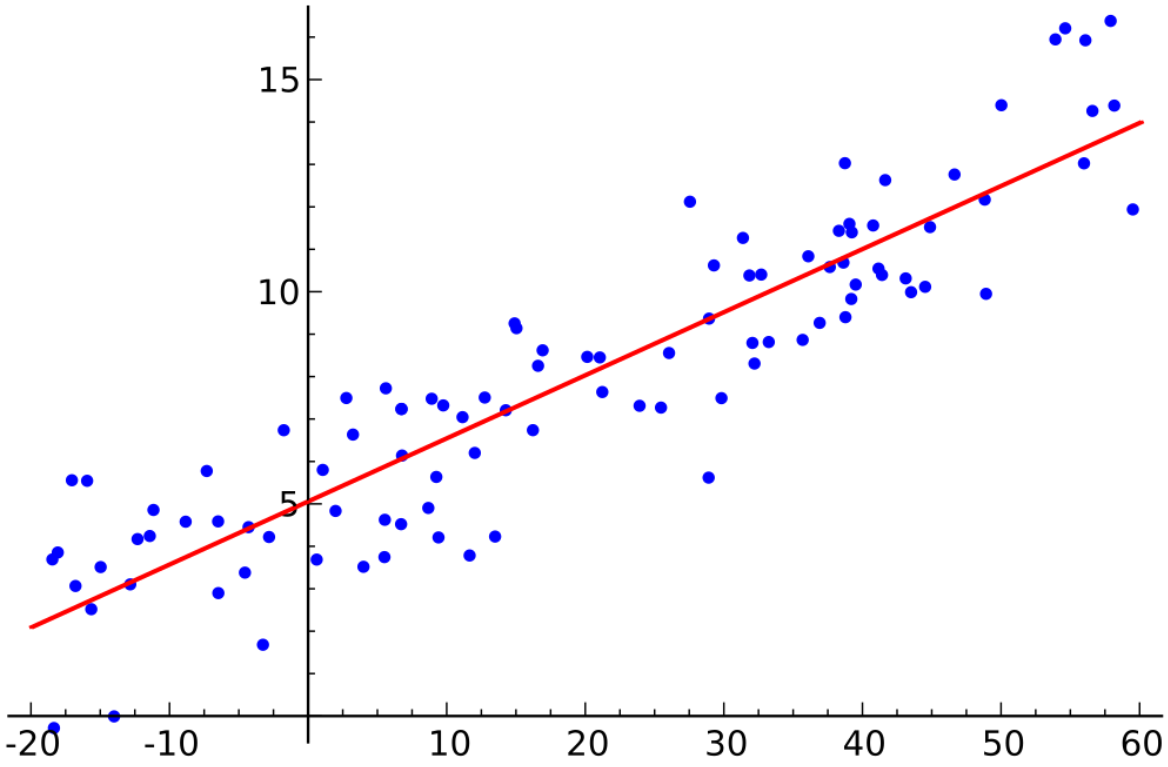
Phân phối các thuộc tính trên theo điểm đánh giá là tuyến tính và các phân phối xấp xỉ nhau nên ta sẽ lựa chọn các phân phối này.

3. Các mô hình học máy thử nghiệm

3.1. Các mô hình hồi quy (Regression)

Bài toán hồi quy là tác vụ yêu cầu máy tính dự đoán một giá trị số thực từ dữ liệu đầu vào. Để giải quyết bài toán này, bài toán học cần học một hàm biểu diễn y = f(x) từ tập dữ liệu D = {(x_1, y_1), (x_2, y_2),…, (x_M, y_M)} sao cho $\boldsymbol{y_i} \cong \boldsymbol{f(x_i)}\ \forall\ \boldsymbol{i}$.

Mỗi điểm dữ liệu x_i được biểu diễn bởi một vector n chiều.



Mô hình hồi quy tuyến tính: hàm f(x) có dạng tuyến tính

$$\boldsymbol{f(x) = w_0 + w_1 x_1 + \cdots + w_n x_n}$$

Trong đó: w_i được gọi là các trọng số của mô hình, w_0 còn được gọi là bias.

- Để đánh giá khả năng học của mô hình, ta có thể sử dụng hàm lỗi thực nghiệm sau:

$$\boldsymbol{RSS(f) = \sum_{i=1}^{M} (y_i - f(x_i))^2}$$

Để tìm được hàm f tối ưu, cần phải cực tiểu hoá hàm RSS trên. Phương pháp này được gọi là cực tiểu hoá bình phương tối thiểu (Ordinary Least Squares, OLS).

Chứng minh được, nghiệm w^* của bài toán trên được tính bởi công thức:

$$\boldsymbol{w^* = (A^T A)^{-1} A^T y}$$

3.1.1. Mô hình hồi quy Ridge (Ridge regression)

Việc sử dụng mô hình OLS có hạn chế là dễ dẫn tới hiện tượng overfitting do quá trình huấn luyện chỉ tập trung tối thiểu hoá lỗi trên tập dữ liệu huấn luyện.

Do người, người ta đề xuất thêm một đại lượng phạt vào hàm loss của mô hình.

$$\boldsymbol{L(w) = \sum_{i=1}^{M}(y_i - f(x_i))^2 + \alpha\,||w||_2^2}$$

α còn được gọi là regularization parameter.

Việc sử dụng α nhằm giảm độ phức tạp của mô hình học được, do đó mô hình học được có tính khái quát hoá cao hơn có thể giảm được overfitting trên tập dữ liệu huấn luyện, tuy lỗi trên tập huấn luyện có thể cao hơn so với OLS.

3.1.2. Mô hình hồi quy Bayes (Bayesian Regression)

3.2. Các mô hình phân lớp (Classification)

Bài toán phân lớp là tác vụ yêu cầu máy tính dự đoán một giá trị số từ dữ liệu đầu vào, trong đó giá trị đầu ra nằm trong tập đã được xác định trước hay còn gọi là nhãn. Đối với bài toán này, mỗi đầu ra *y* chỉ thuộc 01 lớp trong tập {c1, c2, …, c_L} hữu hạn và mỗi quan sát *x* chỉ có 01 nhãn.

Để giải quyết bài toán này có nhiều phương pháp tiếp cận: mô hình k – Nearest neighbor (kNN), mô hình SVM, … Nhóm quyết định sử dụng 2 mô hình để áp dụng cho bộ dữ liệu này là k-NN và SVM.

3.2.1. Mô hình kNN

K-nearest neighbor là một trong những thuật toán supervised-learning khá đơn giản và đôi khi rất hiệu quả trong một vài bộ dữ liệu. Trong bài toán phân lớp này nhóm áp dụng thuật toán kNN với các thông số như sau:

- Khoảng cách: khoảng cách Euclide (mặc định của skicit-learn, là khoảng cách Minkowski với L2-norm)

$$d(x,z) = \sqrt{\sum_{i=1}^{n}(x_i - z_i)^2}$$

- Tham số k: chạy thử và so sánh kết quả giữa các giá trị {11, 13, 15, 17, 19}

- Trọng số của các neighbors: đánh giá trọng số theo khoảng cách, hàng xóm càng gần có trọng số càng cao.

$$d(x,z) = \sqrt{\sum_{i=1}^{n}(x_i - z_i)^2} \longrightarrow d(x,z) = \sqrt{\sum_{i=1}^{n} w_i (x_i - z_i)^2}$$

- Hàm mục tiêu: Accuracy

3.2.2. Mô hình SVC

a. SVM

SVM sẽ tìm một siêu phẳng:

$$\boldsymbol{f(x) = < w.x > + b}$$

Thoả mãn với mỗi x_i: $\boldsymbol{y_i = \begin{cases} 1 \text{ nếu} < w.x > + b \geq 0 \\ -1 \text{ nếu} < w.x > + b < 0 \end{cases}}$

Khi đó, SVM tìm siêu phẳng tách mà có lề lớn nhất (max margin). Với việc tách lề như này, ta sẽ tìm được siêu phẳng có ít lỗi nhất trong các siêu phẳng có thể tồn tại. Bài toán cần tìm w và b sao cho margin lớn nhất:

$$\boldsymbol{(w,b) = arg\max_{w,b}\left\{\min_i \frac{y_i(w^T x_i + b)}{||w||_2}\right\} = arg\max_{w,b}\left\{\frac{1}{||w||_2}\min_i y_i(w^T x_i + b)\right\}} \quad \textbf{(1)}$$

Nếu thay w bởi kw, b bởi kb (k>0) thì khoảng cách từ điểm tới mặt phân chia không đổi. Khi đó, với những điểm gần mặt phân chia nhất, giả sử:

$$\boldsymbol{y_i(w^T x_i + b) = 1}$$

Thì bài toán (1) được biến đổi thành bài toán sau:

$$\boldsymbol{(w, b) = arg\min_{w,b} \frac{1}{2}||w||_2^2} \qquad \boldsymbol{(2)}$$

thoả mãn: $\boldsymbol{1 - y_i(w^T x_i + b) \leq 0, \forall i = 1, 2, \dots, N}$

Dưới góc nhìn hàm Hinge loss kết hợp regularization:

$$\boldsymbol{J(\overline{w}) = \sum_{i=1}^{N} max(0, 1 - y_i \overline{w}^T \overline{x}_i) + \frac{\lambda}{2}||w||_2^2}$$

Với multi-class SVM, ta có hàm loss như sau:

$$\boldsymbol{L(X, y, W) = \frac{1}{N} \sum_{i=1}^{n} \sum_{j \neq y_i} \max\left(0, 1 - w_{y_i}^T x_i + w_j^T x_i\right) + \frac{\lambda}{2}||w||_F^2}$$

b. SVC

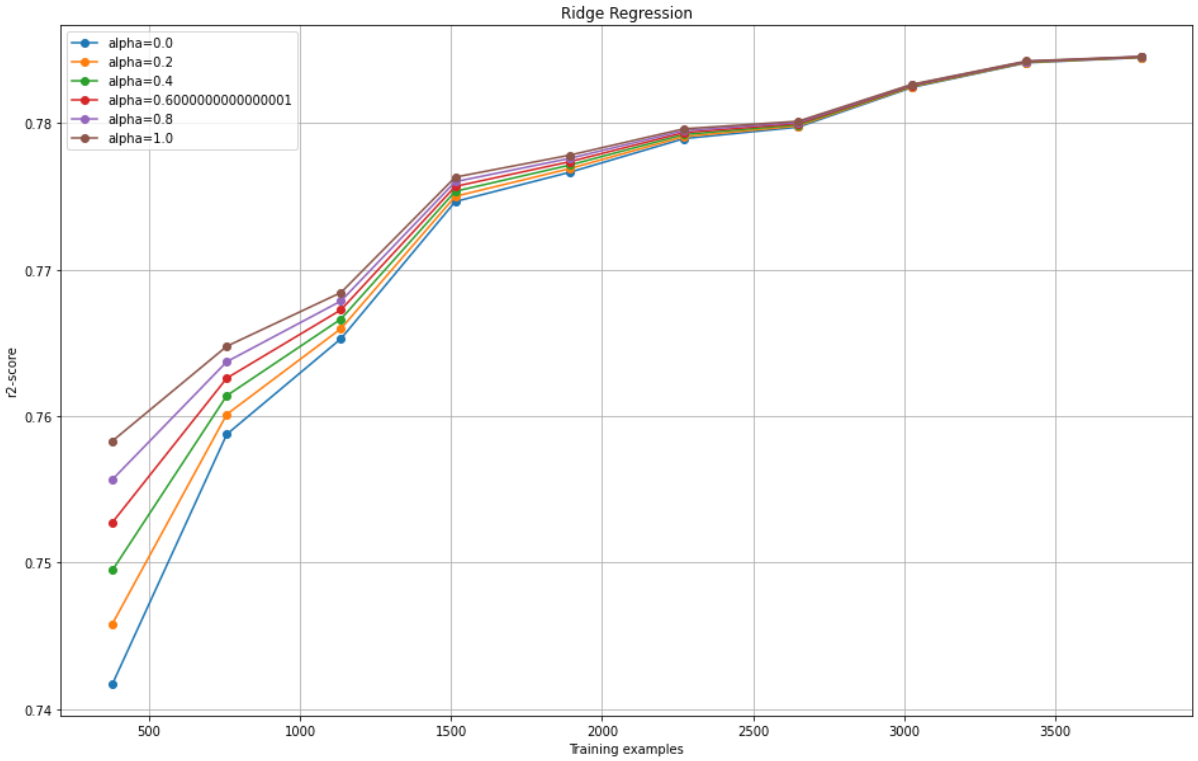
Để giải quyết bài toán multi-class SVM, nhóm sử dụng thư viện scikit-learn với hàm C-Support Vector Classification (SVC). Hàm này cho phép điều chỉnh các tham số kernel và tham số C.

Với kernel, nhóm thực hiện thử với các hàm linear, polynomial, RBF và sigmoid.

Với tham số C, nhóm thực hiện thử với các tham số C ∈ {0.1, 1.0, 2.0, 5.0, 10.0}

4. Đánh giá kết quả thực nghiệm

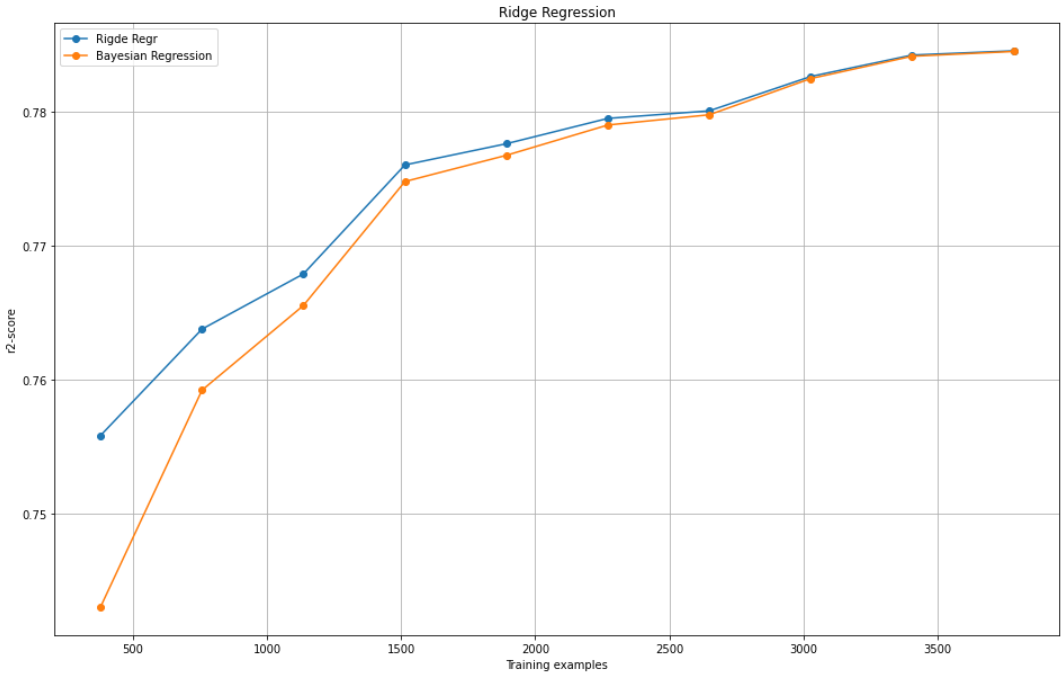
Kết quả của mô hình Ridge Regression với tham số alpha trong khoảng [0, 1] trên tập dữ liệu:



Nhận xét: tập train càng cao thì với các tham số alpha sẽ hội tụ

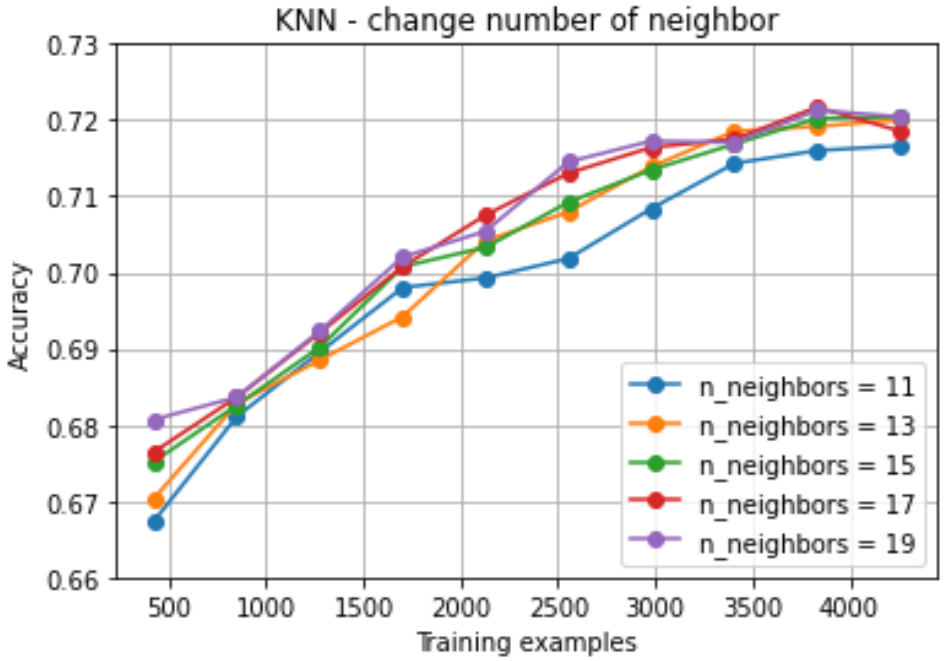
Giá trị alpha tốt nhất là 0.791. Kết quả thu được là 0.797

So sánh kết quả của tốt nhất của mô hình Ridge Regression với mô hình Bayesian Regression huấn luyện dựa trên tham số mặc định của thư viện sklearn



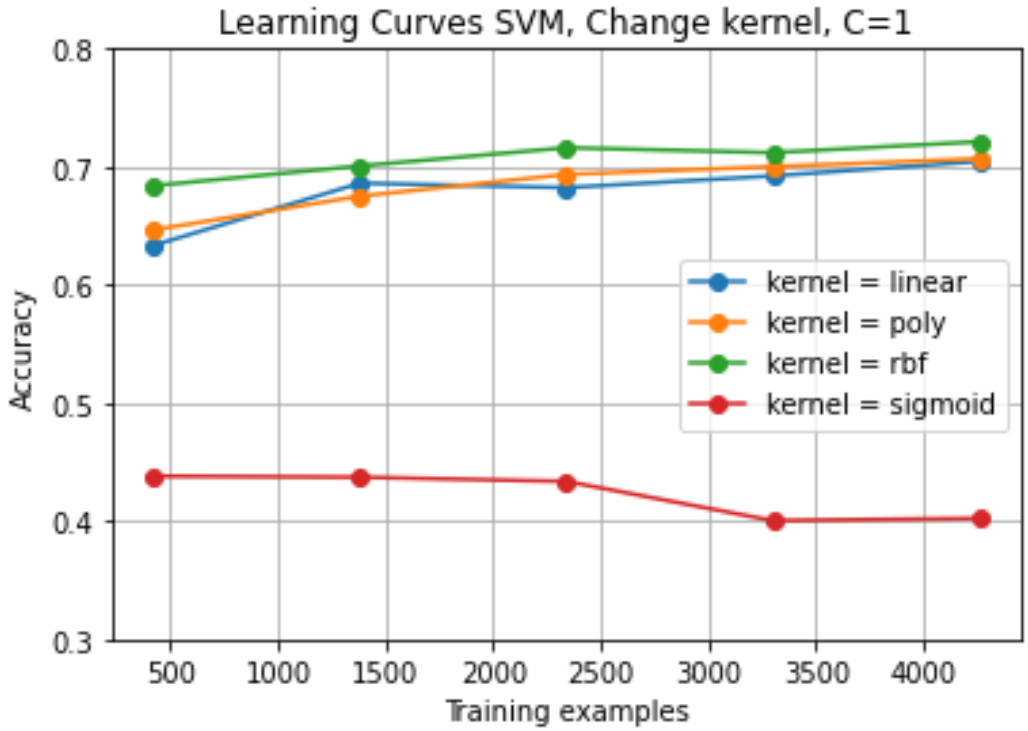
Kết quả thu được là: 0.797

Kết quả của mô hình KNN vơi hàng xóm thay đổi:



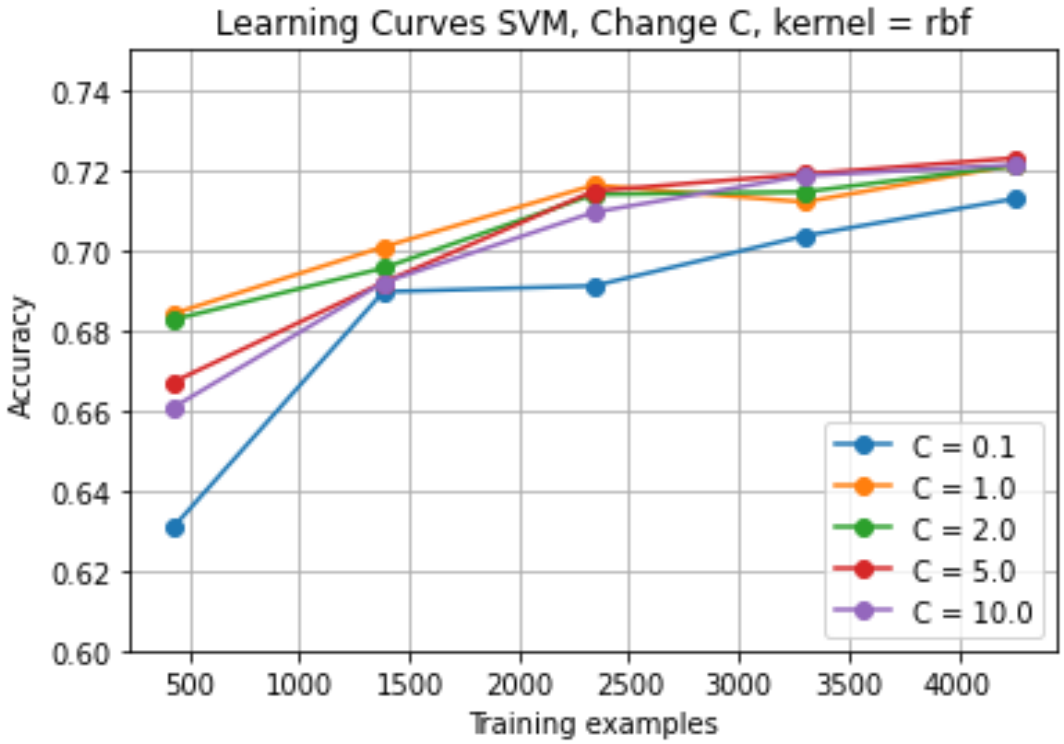
Chọn tham số neighbor = 19 ta được kết quả là: 0.703

Kết quả của SVM khi thay đổi kernel:



Nhân xét với kernel là sigmoid kết quả thấp. Kết quả của rbf là cao nhất. Ta sẽ sử dụng nó để so sánh các tham số C.

Kết quả khi thay đổi với tham số C:



Chọn kernel = rbf và C = 10 ta được kết quả là 0.714

Bảng 1: Kết quả tổng hợp của các mô hình học máy nhóm đã thực hiện

	Ridge Regression	Bayesian Regression	kNN	SVM
Accuracy	0.693	0.692	0.702	0.714

5. Kết luận

Nhóm đã triển khai phân tích dữ liệu và áp dụng thành công các mô hình học máy được học trên lớp lên bộ dữ liệu tmdb_5000. Kết quả thu được là chấp nhận được với bộ dữ liệu chỉ bao gồm 5000 bộ phim. Tuy nhiên độ chính xác của các model là chưa thực sự tốt vì bộ dữ liệu có thể không phù hợp với yêu cầu dự đoán rating của phim, cũng như phần xử lý dữ liệu chưa quá tối ưu.

Để cải thiện kết quả trong tương lai, nhóm dự định sẽ đào sâu hơn về phân tích dữ liệu và thử nghiệm thêm một số mô hình để dự đoán rating phim như là sử dụng cây quyết định, mạng nơ ron, ….

Việc tìm hiểu và nghiên cứu đề tài này đã giúp nhóm có cái nhìn toàn diện hơn trong việc ứng dụng học máy và khai phá dữ liệu vào giải quyết các vấn đề trong thực tế. Do thời gian có hạn nên đề tài không tránh khỏi những sai sót. Rất mong nhận được những lời góp ý từ thầy và các bạn để giúp đề tài của nhóm chúng em được hoàn thiện hơn.

Cuối cùng, nhóm xin gửi lời cám ơn đến PGS.TS Thân Quang Khoát, các anh chị trợ giảng đã đưa đến cho chúng em những kiến thức rất bổ ích và thú vị.

6. Tài liệu tham khảo

[1] Slide bài giảng của lớp Học máy và Khai phá dữ liệu

[2] Bộ dữ liệu tmdb_5000:

https://www.kaggle.com/tmdb/tmdb-movie-metadata

[3] Forum Machine Learning cơ bản:

https://machinelearningcoban.com/