Dự đoán thu nhập

Nhóm 14
Võ Thục Khánh Huyền 20190055
Trịnh Hồng Phượng 20190062

Mục lục

Danh sách hình vẽ

Danh sách bảng

1 Giới thiệu

Trong đồ án của môn học "Nhập môn Trí tuệ Nhân tạo", nhóm chúng tôi sẽ tiến hành bài toán Dự đoán thu nhập. Bài toán này thuộc về một cuộc thi Khoa học dữ liệu, được tiến hành trên

nền tảng Kaggle. Dựa vào thông tin cá nhân và kinh nghiệm làm việc của một người, mục tiêu của bài toán là phân loại người đó vào các nhóm thu nhập trải dài từ cấp độ 1 đến 7.

2 Mô tả bài toán

Về các cấp độ thu nhập, cụ thể hơn chúng ta sẽ phân loại dựa vào các nhóm ở bảng 1.

Cấp độ	Mô tả
7	Rất cao
6	Trung bình cao
5	Cao
4	Trung bình
3	Thấp
2	Trung bình thấp
1	Rất thấp

Bảng 1: Các cấp độ của mức thu nhập

Tiếp theo, chúng ta sẽ đi tìm hiểu phần quan trọng nhất của một bài toán Khoa học dữ liệu, đó chính là dữ liệu. Dữ liệu của bài toán này bao gồm 6 bảng, trong đó 3 bảng thuộc về tập huấn luyện, 3 bảng còn lại thuộc về tập kiểm tra, cụ thể hơn ở bảng 2.

Mục tiêu của bài toán là bằng việc phân tích và sử dụng các mô hình học máy để học bộ dữ liệu huấn luyện, chúng ta có thể dự đoán được mức thu nhập cho bộ dữ liệu kiểm tra.

3 Phương pháp giải quyết bài toán

Để giải quyết một bài toán khoa học dữ liệu, chúng ta cần thực hiện lần lượt các bước như sau: Phân tích dữ liệu, Làm sạch dữ liệu, Tạo thêm các thuộc tính mới, Huấn luyện mô hình, Đánh giá mô hình. Sau đây, chúng ta sẽ đi tìm hiểu từng bước một.

3.1 Phân tích dữ liệu

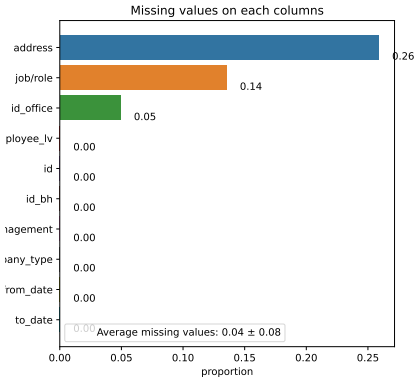
Để tiến hành phân tích các khía cạnh của một bảng dữ liệu, chúng tôi sử dụng công cụ Pandas profiling. Công cụ này biểu diễn tất cả những thông tin cần thiết về một bảng dữ liệu như: số dữ liệu bị thiếu, bị nan, quan hệ giữa các cột, và rất nhiều thông tin khác chúng ta có thể khai thác. Công cụ này hỗ trợ chúng ta nhanh chóng trong việc phân tích, rút ngắn thời gian để đến với bước tiếp theo.

Dữ liệu bị thiếu Có một vài cột trong các bảng dữ liệu bị thiếu thông tin, dựa trên phân tích này, chúng ta sẽ tiến hành phủ các giá trị thiếu ở bước sau. Các hình 1 và 2 mô tả số lượng phần trăm mẫu dữ liệu thiếu đối với từng cột cụ thể. Ở đây ta có thể thấy số lượng phần trăm mẫu dữ liệu thiếu ở bảng huấn luyện và bảng kiểm tra là tương tự nhau.

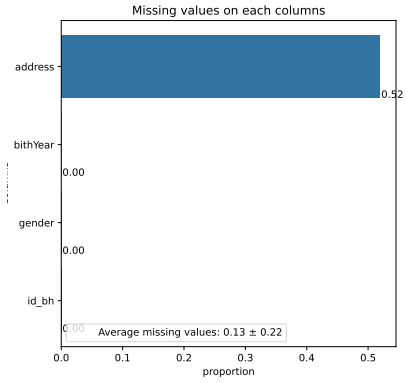
Phân bố giá trị trong mỗi cột Chúng tôi tiến hành tìm hiểu phân bố giá trị trong mỗi cột, từ đó có thể hiểu được xu hướng của dữ liệu tập trung vào đâu, và có thể loại bỏ các giá

Bảng	Mô tả
info_train, info_test	Thông tin cá nhân (định danh người, giới tính, năm sinh, địa chỉ nhà)
work_train, work_test	Thông tin về kinh nghiệm làm việc (1 người có thể có nhiều công việc, với mỗi công việc sẽ bao gồm các thông tin: định danh bản ghi, định danh người, công việc, ngày bắt đầu, ngày kết thúc, cấp độ nhân viên, địa chỉ làm việc, loại công ty, định danh quản lý, định danh văn phòng)
label_train, label_test	Cấp độ của mức thu nhập (định danh người, cấp độ mức thu nhập)

Bảng 2: Thông tin mô tả các bảng trong bộ dữ liệu



Hình 1: Dữ liệu bị thiếu ở bảng train_info



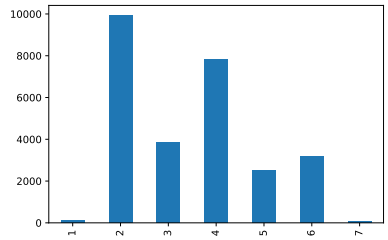
Hình 2: Dữ liệu bị thiếu ở bảng train_info

trị ngoại lai, chuẩn hoá giá trị sao cho phù hợp. Sau đây chúng tôi sẽ trình bày một số thông tin nổi bật cần lưu ý. Đối với cột cấp độ mức thu nhập, ở hình 3, ta thấy cấp độ mức thu nhập tập trung nhiều ở mức 2 và 4, phân bố ít ở mức 1 và 7. Tuy nhiên, hàm đo lường trong bài toán này là F1-weighted, nên số lượng ít hay nhiều của các cấp độ sẽ không ảnh hưởng nhiều nếu như F1-score cho mức độ đó là nhiều hay ít. Ta sẽ tìm hiểu kỹ ở điều này ở phần đánh giá mô hình. Tiếp theo, đối với loại công ty, ở hình 4, ta thấy dữ liệu có giá trị -1 rất nhiều, đây là giá trị lỗi nhằm thay thế cho việc thiếu thông tin, ta sẽ coi các bản ghi có giá trị này thuộc cùng một loại công ty. Cuối cùng, với cột cấp độ nhân viên, ta thấy những giá trị ngoại lai xuất hiện trong cột này là những giá trị lớn hơn 100. Đối với các bản ghi có giá trị bằng -1, đây là giá trị lỗi, ta sẽ chuẩn hoá ở bước sau.

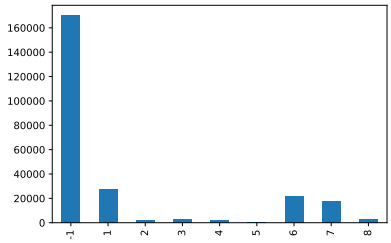
3.2 Làm sạch dữ liệu

Làm sạch dữ liệu là một bước quan trọng trong bài toán dữ liệu. Việc làm này giúp bước học mô hình phía sau được tiến hành dễ dàng hơn, dữ liệu được chuẩn hoá nhằm loại bỏ nhiễu trong dữ liệu gốc.

Chia giá trị dữ liệu thành các nhóm nhỏ Ở các cột địa chỉ và công việc, nếu giữ nguyên giá trị như ở các bảng, số lượng lớp đối với các cột này là rất lớn. Chính vì vậy, chúng tôi gom



Hình 3: Phân bố dữ liệu ở cột cấp độ mức thu nhập



Hình 4: Phân bố dữ liệu ở cột loại công ty

nhóm các giá trị lại, nhằm mục đích giảm số lượng lớp trong mỗi thuộc tính categorical. Lấy ví dụ: Với cột địa chỉ, các giá trị sau thuộc về lớp "nghe an": "nghe an", "do luong", "dien chau", "quynh luu", "q.luu". Trước khi phân lớp nhỏ, chúng tôi loại bỏ tất cả các dấu của dữ liệu chữ để tránh trường hợp viết sai dấu câu, vị trí dấu không đúng.

Thay thế các giá trị nan và chuẩn hoá các giá trị Đối với những cột có giá trị nan, chúng tôi sẽ thay thế bằng một giá trị cụ thể. Đối với những cột có giá trị không hợp lệ, chúng tôi cũng thay thế tương tự như giá trị nan.

Loại bỏ các giá trị ngoại lai Ở cột cấp độ nhân viên, một số giá trị ngoại lai xuất hiện có thể khiến mô hình học sai đi những thứ cần học. Đó là lý do chúng tôi tiến hành loại bỏ các giá trị ngoài lai trong cột này.

3.3 Tạo ra các thuộc tính mới

Đây là bước giúp làm giàu thông tin của dữ liệu. Việc tạo ra các thuộc tính mới chính là việc ứng dụng các tri thức ở bên ngoài vào việc phân tích, nhờ đó chúng ta có thể tạo ra các thuộc tính quan trọng, giúp mô hình hoạt động tốt hơn.

Khai thác từ bảng info Từ các bảng train_info và test_info, chúng tôi tạo thêm các thuộc tính tuổi và lớp tuổi từ cột năm sinh.

Khai thác từ bảng work Từ bảng train_work và test_work, chúng tôi sẽ tạo thêm các thuộc tính mới như: tổng số năm/tháng/ngày làm việc, số năm/tháng/ngày từ lúc bắt đầu làm việc cho đến hiện tại, số năm/tháng/ngày từ lúc kết thúc làm việc cho đến hiện tại.

3.4 Chuẩn bị dữ liệu cho việc huấn luyện

Do dữ liệu ở bảng work có nhiều hơn một bản ghi đối với một người, nên chúng tôi sẽ tạo ra các thuộc tính mới dựa vào quan hệ của các bản ghi đó, cụ thể hơn chúng tôi sử dụng các hàm sum, mean, max, min, std, medium. Ngoài ra chúng tôi tạo ra các thuộc tính liên quan đến công việc đầu tiên, công việc cuối cùng và sự chênh lệch giữa hai công việc này.

Sau khi tạo ra các thuộc tính mới, chúng tôi phân loại các thuộc tính thuộc nhóm categorical hay numeric. Đối với thuộc tính categorical, chúng tôi sẽ tiến hành encode trước khi đưa vào mô hình SVM hay Random Forest, với mô hình CatBoost, chúng tôi có thể truyền thẳng thuộc tính categorical vào bằng cách định nghĩa.

Tiếp theo, chúng tôi sẽ tính mức độ quan trọng của các thuộc tính vừa được tạo. Từ bảng 3, ta có thể thấy những thuộc tính liên quan đến cấp độ nhân viên và công việc cuối cùng có

STT	10 thuộc tính quan trọng nhất	
	Thuộc tính	Độ quan trọng
1	employee_lv_last_work	21315.650852
2	employee_lv_last_work_/10	16643.109649
3	employee_lv_max	6804.497385
4	employee_lv_last_first_work	4764.056627
5	id_office_last_work	3554.918988
6	employee_lv_mean	3228.806061
7	employee_lv_median	2795.160201
8	id_office_2_last_work	2558.972020
9	employee_lv_std	2488.898708
10	employee_lv_first_work	1276.936129
STT	10 thuộc tính kém quan trọng nhất	
	Thuộc tính	Độ quan trọng
1	total_days_distance_first_work	45.555632
2	total_months_distance_first_work	44.416726
3	company_type_nunique	40.044704
4	expire_days_last_work	36.124962
5	expire_months_last_work	34.819623
6	expire_years_last_work	31.354873
7	id_office_1_nunique	16.427460
8	work_address_nunique	14.382244
9	id_office_2_nunique	8.906989
10	id_office_nunique	7.834818

Bảng 3: 10 thuộc tính quan trọng và kém quan trọng nhất

điểm rất cao, đây sẽ là những thuộc tính quan trọng giúp mô hình học tốt hơn.

3.5 Huấn luyện mô hình

Trong bài toán này, chúng tôi sẽ tiến hành thử nghiệm trên ba mô hình, đó là: SVM, Random Forest và CatBoost.

SVM là một mô hình học máy có giám sát, được ứng dụng rộng rãi trong các bài toán phân loại. SVM sử dụng các siêu phẳng để phân tách các lớp, từ đó ta có thể học được nhãn dán tương ứng cho từng bản ghi. Đối với bài toán phân loại nhiều lớp, SVM sẽ tiến hành nhiều bài toán phân loại hai lớp, một lớp so với các lớp còn lại.

Random Forest là một mô hình học máy có giám sát, được ứng dụng rộng rãi trong các bài toán phân loại. Random Forest được bắt nguồn từ mô hình Decision Tree, bằng cách xây dựng nhiều cây quyết định ngẫu nhiên.

CatBoost là một mô hình học máy có giám sát, được bắt nguồn từ các thuật toán cây và sử dụng boosting. CatBoost được sử dụng rộng rãi hiện nay trong các cuộc thi phân tích dữ liệu nhờ độ chính xác rất cao mà nó mang lại. Ngoài ra, CatBoost còn cho phép chúng ta truyền vào thuộc tính categorical mà không cần encode trước.

Tất cả các mô hình trên, chúng tôi đều sử dụng thư viện Scikit-learn [2] để triển khai. Chúng tôi sử dụng thư viện Optuna [1] để tìm ra các siêu tham số tối ưu cho mô hình của mình.

Model	Validation set	Public test set	Private test set
Random Forest	0.83217	0.83831	0.83677
SVM	0.47230	0.52167	0.51461
CatBoost	0.92847	0.93470	0.93283

Bảng 4: Kết quả của các mô hình

3.6 Đánh giá mô hình

Chúng tôi sẽ dùng độ đo F1-weighted để đánh giá mô hình. Với mỗi lớp, độ đo F1-score được tính như sau:

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

Với độ đo F1-weighted, chúng ta sẽ tính trung bình F1-score của các lớp, với số lượng bản ghi trong mỗi lớp là trọng số.

Sau khi tiến hành chạy các mô hình, chúng tôi nhận thấy độ chính xác của hai mô hình SVM và Random Forest thấp đáng kể khi chúng tôi thực hiện encode các thuộc tính categorical. Chính vì vậy, chúng tôi quyết định bỏ các thuộc tính categorical ra khỏi dữ liệu. Độ quan trọng của các thuộc tính categorical cũng không lớn, nên chúng tôi hi vọng độ chính xác của mô hình sẽ không bị ảnh hưởng nhiều khi tiến hành bỏ các thông tin này. Kết quả của các mô hình thu được ở bảng 4.

Ta có thể thấy, CatBoost là thuật toán hiện đại nhất nên kết quả mà nó đem lại cũng vượt trội hơn hai thuật toán kia. Random Forest là một thuật toán cổ điển, tuy nhiên độ chính xác của mô hình này là rất tốt, cao hơn rất nhiều so với SVM. SVM cho kết quả thấp nhất, từ đó ta có thể nhận xét rằng SVM sẽ học không tốt khi mô hình của chúng ta có nhiều lớp.

Ngoài ra, sau khi phân tích vào các bản ghi được dự đoán sai, chúng tôi thấy đa số dự đoán sai thuộc về hai nhóm 1 và 7. Tuy nhiên vì số lượng các bản ghi của hai nhóm này rất ít, nên khi chúng ta tiến hình đánh giá bằng độ đo F1-weighted, kết quả sẽ không bị ảnh hưởng quá nhiều. Nhìn chung, đối với các nhóm còn lại, tỉ lệ dự đoán sai chỉ khoảng 10% đối với mô hình Random Forest.

4 Tổng kết và công việc tương lai

Tổng kết lại, trong bài toán này, chúng tôi đã tiến hành đầy đủ các bước cho một bài toán phân tích dữ liệu. Chúng tôi đã thử nghiệm với ba mô hình học máy là SVM, Random Forest và CatBoost, trong đó CatBoost mang lại kết quả vượt trội hơn so với hai thuật toán còn lại.

Tuy nhiên, chúng tôi vẫn chưa khai thác được các thuộc tính categorical khi thử nghiệm mô hình SVM và Random Forest. Trong tương lai, chúng tôi mong muốn có thể tìm được một thuật toán encode phù hợp, và tiến hành hành thử nghiệm lại trên hai thuật toán này.

Tất cả mã nguồn và hướng dẫn chạy được chúng tôi công khai ở Github.

Tài liệu

[1] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.