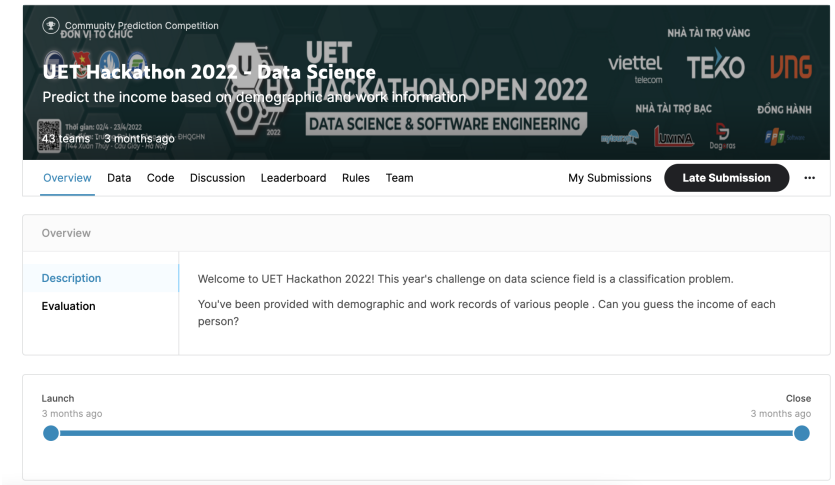
Dự đoán thu nhập

Nhóm 14
Võ Thục Khánh Huyền 20190055
Trịnh Hồng Phượng 20190062

Ngày 18 tháng 7 năm 2022

Giới thiệu



Hình 1: Cuộc thi dự đoán thu nhập trên nền tảng Kaggle

Phân loại một người vào các nhóm thu nhập, dựa vào 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

Mô tả bài toán

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.

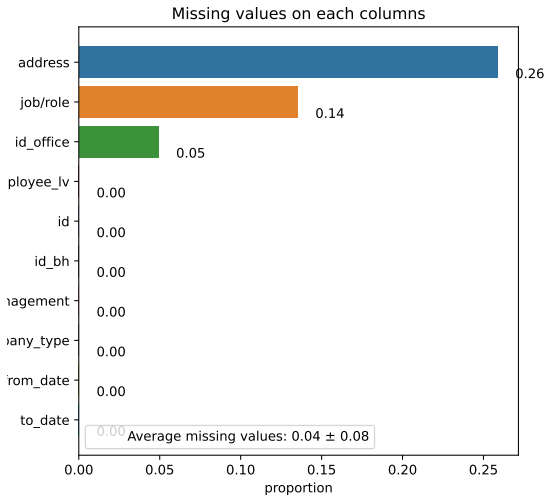
Bảng	Mô tả
info_train, info_test	Thông tin cá nhân (định danh người, giới tính, năm sinh, địa chỉ nhà)
work_train, work_test	Thông tin về kinh nghiệm làm việc (1 người có thể có nhiều công việc, với mỗi công việc sẽ bao gồm các thông tin: định danh bản ghi, định danh người, công việc, ngày bắt đầu, ngày 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

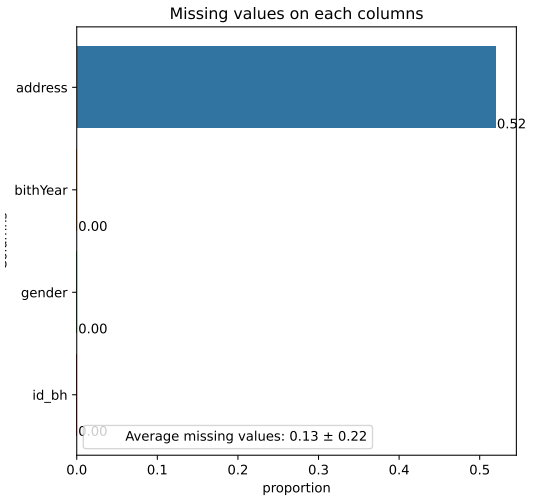
Phân tích dữ liệu

Để tiến hành phân tích các khía cạnh của một bảng dữ liệu, chúng tôi sử dụng công cụ **Pandas profiling**.

Dữ liệu bị thiếu



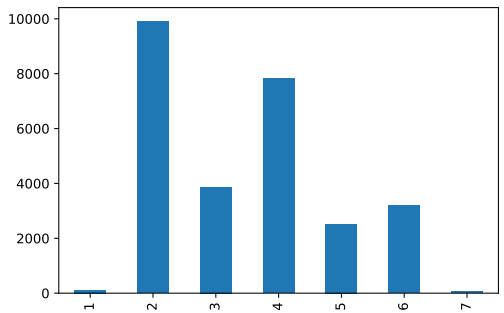
Hình 2: Dữ liệu bị thiếu ở bảng train_info



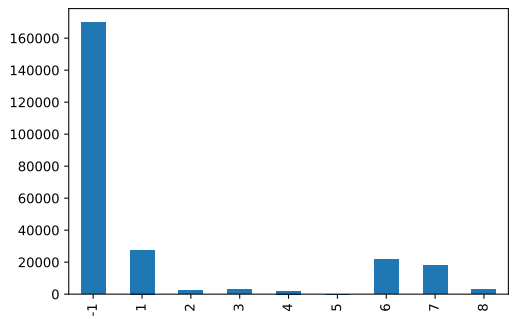
Hình 3: Dữ liệu bị thiếu ở bảng train_info

Phân tích dữ liệu

Phân bố giá trị trong mỗi cột



Hình 4: Phân bố dữ liệu ở cột cấp độ mức thu nhập



Hình 5: Phân bố dữ liệu ở cột loại công ty

1. **Mức thu nhập:** Mức 1, 7 ít, mức 2, 4 nhiều.
2. **Loại công ty:** Loại -1 là giá trị lỗi.
3. **Cấp độ nhân viên:** Giá trị ngoại lai lớn hơn 100, giá trị -1.

Làm sạch dữ liệu

Làm sạch dữ liệu 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.

1. **Chia giá trị dữ liệu thành các nhóm nhỏ** Chúng tôi gom nhóm các giá trị ở cột địa chỉ và công việc lại, nhằm mục đích giảm số lượng lớp trong mỗi thuộc tính categorical.
   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".
2. **Thay thế các giá trị nan và chuẩn hoá các giá trị** Đối với những cột có giá trị nan và giá trị không hợp lệ, chúng tôi sẽ thay thế bằng một giá trị cụ thể.
3. **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.

Tạo ra các thuộc tính mới

Tạo ra các thuộc tính mới là bước giúp làm giàu thông tin của dữ liệu, nhờ đó chúng ta có thể tạo ra các thuộc tính quan trọng, giúp mô hình hoạt động tốt hơn.

1. **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.
2. **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.

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.
- 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:

- Với mô hình SVM và Random Forest, chúng tôi sẽ tiến hành encode trước khi đưa vào mô hình.
- Với mô hình CatBoost, chúng tôi có thể truyền thẳng thuộc tính categorical vào bằng cách định nghĩa.

Ngoài ra, chúng tôi cũng tính mức độ quan trọng của các thuộc tính vừa được tạo.

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.
- Tất cả các mô hình trên, chúng tôi đều sử dụng thư viện **Scikit-learn** Pedregosa et al. (2011) để triển khai.
- Chúng tôi sử dụng thư viện **Optuna** Akiba et al. (2019) để tìm ra các siêu tham số tối ưu cho mô hình của mình.

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.

Đánh giá mô hình

Kết quả của các mô hình thu được ở bảng 3.

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 3: Kết quả của các mô hình

- Bỏ các thuộc tính categorical ra khỏi dữ liệu khi huấn luyện mô hình SVM và Random Forest.
- Mức thu nhập 1, 7 có tỉ lệ dự đoán sai cao, các mức còn lại tỉ lệ dự đoán sai khoảng 10%.

Tổng kết và công việc tương lai

Tổng kết:

1. 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.
2. Chúng tôi vẫn chưa khai thác được các thuộc tính categorical khi thử nghiệm mô hình SVM và Random Forest.

Công việc tương lai:

1. 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.

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.

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.