ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án 3: Linear Regression

Toán ứng dụng và thống kê

< Lab04 >

Huỳnh Sỉ Kha - 21127734
Lớp: 21CLC02

Giảng viên:

Vũ Quốc Hoàng
Nguyễn Văn Quang Huy
Lê Thanh Tùng
Phan Thị Phương Uyên

12th October 2023

Mục lục

1 Giới thiệu thuật toán hồi quy tuyến tính - Linear Regression: 2
2 Một số kỹ thuật được dùng trong đồ án 3
2.1 KFold 3
2.2 XGBoost 4
2.3 Thuật toán di truyền - Genetic 5
3 Các mô hình theo yêu cầu của đồ án 6
3.1 Công cụ và thư viện hỗ trợ 6
3.2 Yêu cầu 1a 6
3.3 Yêu cầu 1b 8
3.4 Yêu cầu 1c 9
3.5 Yêu cầu 1d 11
4 Các hàm chức năng trong chương trình: 14
4.1 Lớp LinearRegression 14
4.2 Hàm tìm mô hình tốt nhất 15
4.3 Hàm tính MAE (Mean Absolute Error) với các đặc trưng 15
5 Nhận xét tổng quát 16

1 Giới thiệu thuật toán hồi quy tuyến tính - Linear Regression:

Linear Regression là một phương pháp trong thống kê và machine learning để mô hình hóa mối quan hệ tuyến tính giữa biến độc lập (hay còn gọi là biến input hoặc biến x) và biến phụ thuộc (hay còn gọi là biến output hoặc biến y).

Mục tiêu của Linear Regression là tìm một đường thẳng (hoặc siêu mặt phẳng trong không gian cao hơn) tốt nhất để tạo ra một dự đoán cho biến phụ thuộc dựa trên giá trị của biến độc lập.

Trong Linear Regression, giả định chính là mối quan hệ giữa các biến có thể được mô tả bằng một đường thẳng, tức là mỗi thay đổi đơn vị trong biến độc lập sẽ dẫn đến một thay đổi cố định trong biến phụ thuộc. Điều này được biểu diễn bởi phương trình đường thẳng:

$$y = mx + b \tag{1}$$

- y là biến phụ thuộc cần dự đoán.
- x là biến độc lập (input).
- m là độ dốc (slope) của đường thẳng, biểu thị mức độ tương quan giữa x và y.
- b là điểm cắt trục y, thể hiện giá trị của y khi x bằng 0.

Tuy nhiên, trong thực tế, dữ liệu thường không thể hoàn toàn phù hợp với một đường thẳng. Do đó, Linear Regression cố gắng tìm một đường thẳng sao cho tổng các sai số bình phương giữa các điểm dữ liệu thực tế và dự đoán trên đường thẳng là nhỏ nhất. Phương pháp phổ biến để đạt được điều này là phương pháp bình phương tối thiểu (Least Squares Method).

Ngoài ra, Linear Regression có thể được mở rộng để xử lý nhiều biến độc lập hơn (Multiple Linear Regression) và cũng có các biến thể như Ridge Regression và Lasso Regression để kiểm soát overfitting và cải thiện hiệu suất dự đoán.

Trong thực tế, Linear Regression được ứng dụng rộng rãi trong nhiều lĩnh vực như kinh tế học, khoa học xã hội, y học, và machine learning để dự đoán và hiểu mối quan hệ giữa các biến.

2 Một số kỹ thuật được dùng trong đồ án

2.1 KFold

Thuật toán K-Fold Cross-Validation là một phương pháp đánh giá hiệu suất của mô hình máy học. Nó được sử dụng để ước tính khả năng tổng quát hóa của một mô hình trên dữ liệu chưa từng thấy. K-Fold Cross-Validation là một phần quan trọng trong quá trình đào tạo và đánh giá mô hình, giúp đảm bảo rằng mô hình hoạt động tốt trên dữ liệu mới mà nó chưa được huấn luyện trước đó.

Cơ bản, K-Fold Cross-Validation hoạt động như sau:

1. Chia tập dữ liệu thành K phần bằng nhau, gọi là "fold".
2. Lặp lại K lần, mỗi lần sử dụng một fold làm tập kiểm tra (test set) và các fold còn lại làm tập huấn luyện (train set).
3. Huấn luyện mô hình trên tập huấn luyện và đánh giá hiệu suất trên tập kiểm tra bằng một metric như mean squared error, accuracy, hay mean absolute error.
4. Lưu giữ kết quả hiệu suất (điểm đánh giá) của mô hình trong mỗi lần lặp.

Cuối cùng, có được K giá trị hiệu suất, một cho mỗi lần lặp. Từ đó, có thể tính giá trị trung bình của các điểm đánh giá này để đánh giá tổng thể hiệu suất của mô hình trên dữ liệu mới.

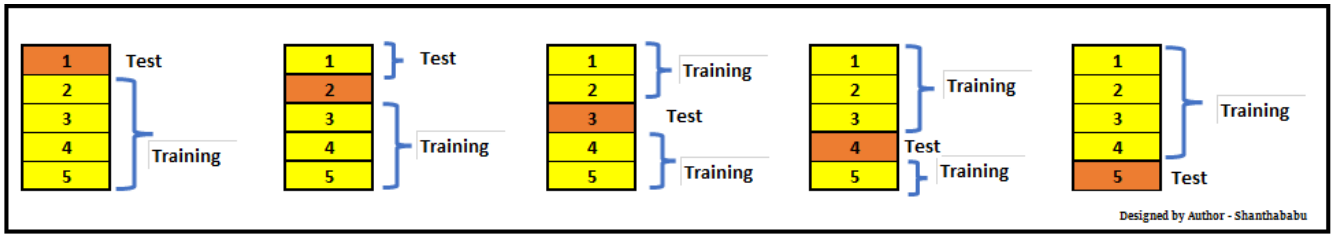

Figure 1: Minh hoạ thuật toán KFold

Nguồn tài liệu tham khảo [11]

2.2 XGBoost

a Giới thiệu:

XGBoost (eXtreme Gradient Boosting) là một thuật toán học máy dựa trên kỹ thuật Gradient Boosting. Nó là một trong những thuật toán mạnh mẽ và phổ biến để giải quyết các vấn đề học có giám sát như dự đoán và phân loại. Một trong những điểm mạnh quan trọng của XGBoost là khả năng xử lý dữ liệu không đồng nhất, cũng như khả năng xử lý các đặc trưng số và đặc trưng rời rạc.

b Cách XGBoost đánh giá điểm số cho các đặc trưng trong mô hình

1. **Boosting Trees:** XGBoost hoạt động bằng cách xây dựng nhiều cây quyết định yếu (weak learners) theo cách lặp. Mỗi cây dự đoán sẽ cố gắng sửa sai số của mô hình trước đó. Cây mới được thêm vào mô hình để cố gắng "chỉnh sửa" những phần mô hình trước đó không thể dự đoán tốt.
2. **Hàm Mất Mát (Loss Function):** XGBoost sử dụng hàm mất mát để đánh giá sự sai khác giữa dự đoán của mô hình và giá trị thực tế. Trong quá trình xây dựng cây mới, XGBoost cố gắng tối thiểu hóa hàm mất mát bằng cách điều chỉnh các trọng số cho các điểm dữ liệu. Điều này đảm bảo rằng cây mới tập trung vào những điểm dữ liệu mà mô hình trước đó dự đoán sai.
3. **Đặc trưng quan trọng:** Khi huấn luyện xong, XGBoost tính toán một thứ tự quan trọng cho các đặc trưng. Quan trọng của mỗi đặc trưng được tính dựa trên cách đặc trưng đó được sử dụng để tạo ra sự cải thiện trong hàm mất mát trong quá trình xây dựng cây. Đặc trưng quan trọng cao hơn đóng góp nhiều hơn vào việc cải thiện mô hình.
4. **Tích hợp đặc trưng quan trọng:** Gọi hàm **xgb.plot_importance(model)** để vẽ biểu đồ về mức độ quan trọng của các đặc trưng. Các đặc trưng quan trọng sẽ được sắp xếp dựa trên độ quan trọng của chúng trong việc cải thiện mô hình.

Nguồn tài liệu tham khảo: [10]

2.3 Thuật toán di truyền - Genetic

a Giới thiệu:

Thuật toán di truyền (Genetic Algorithm) là một phương pháp tối ưu hóa được lấy cảm hứng từ quá trình tiến hóa trong tự nhiên. Nó được sử dụng để tìm ra các giải pháp gần tối ưu cho các vấn đề tối ưu hóa và tối ưu hóa tham số. Trong trường hợp của việc tìm kiếm các đặc trưng tốt nhất cho một mô hình máy học, thuật toán di truyền có thể được áp dụng như sau:

b Cách Genetic tìm ra mô hình:

1. **Khởi tạo quần thể (Population):** Quần thể ban đầu gồm một tập hợp các cá thể (individuals), mỗi cá thể biểu diễn một tập hợp đặc trưng. Các tập hợp này ban đầu được tạo ngẫu nhiên.
2. **Đánh giá (Fitness function):** Để đo lường mức độ tốt của mỗi cá thể. Trong ngữ cảnh tìm kiếm đặc trưng, hàm này có thể được thiết kế để đo lường hiệu suất của mô hình dựa trên các đặc trưng tương ứng với cá thể.
3. **Chọn lọc (Selection):** Các cá thể có hiệu suất tốt hơn trong hàm đánh giá được chọn để tham gia vào quá trình tiếp theo. Quá trình chọn lọc có thể dựa trên nhiều chiến lược như chọn ngẫu nhiên, chọn dựa trên xếp hạng hiệu suất, hoặc sử dụng các kỹ thuật như "roulette wheel selection" hoặc "tournament selection".
4. **Tiến Hóa (Crossover và Mutation):** Quá trình tiến hóa bao gồm hai pha quan trọng là "crossover" và "mutation". Trong pha "crossover", các đặc trưng của hai cá thể được kết hợp để tạo ra cá thể con mới. Điều này giúp thừa hưởng các đặc trưng tốt từ cả hai cha mẹ. Trong pha "mutation", một số đặc trưng trong cá thể con mới được thay đổi ngẫu nhiên để đưa vào sự đa dạng và khám phá.
5. **Thay Thế (Replacement):** Các cá thể con mới sau quá trình tiến hóa thay thế các cá thể gốc trong quần thể. Quá trình này giúp duy trì kích thước quần thể không thay đổi và đưa vào sự cạnh tranh giữa các cá thể.
6. **Tiêu Chuẩn Dừng (Termination Criteria):** Quá trình tiến hóa tiếp tục cho đến khi một số tiêu chuẩn dừng được đáp ứng, chẳng hạn như số lượng thế hệ tối đa hoặc khi giá trị tốt nhất không thay đổi trong một khoảng thời gian dài.
7. **Kết Quả:** Sau khi thuật toán di truyền kết thúc, cá thể có hiệu suất tốt nhất trong quần thể được chọn làm giải pháp cuối cùng. Các đặc trưng tương ứng với cá thể này được sử dụng để xây dựng mô hình máy học cuối cùng.

Nguồn tài liệu tham khảo: [8]

3 Các mô hình theo yêu cầu của đồ án

3.1 Công cụ và thư viện hỗ trợ

a Thư viện hỗ trợ

- numpy: Thực hiện các phép tính trong toán học (tính tổng, trung bình các của các phần tử). [4]
- matplotlib: Minh hoạ số liệu bằng biểu đồ. [14]
- sklearn: Thuật toán KFold. [5]
- xgboost: Đánh giá độ quan trọng của đặc trưng. [6]
- pandas: Đọc dữ liệu từ file csv. [1]
- itertools: Dùng để tạo ra các tổ hợp để thực hiện thuật toán Brute Force. [7]
- Hướng dẫn đồ án 03: [15]

b Công cụ hỗ trợ

- Thiết bị chạy chương trình: Asus ROG Zephyrus G14 (8GB, 512 GB).
- Hệ điều hành: Window 10.

3.2 Yêu cầu 1a

Mô hình gồm đặc trưng:`‘Gender‘, ‘10percentage‘, ‘12percentage‘, ‘CollegeTier‘, ‘Degree‘, ‘collegeGPA‘, ‘CollegeCityTier‘, ‘English‘, ‘Logical‘, ‘Quant‘, ‘Domain‘`

a Mô tả

1. Chọn các đặc trưng theo yêu cầu:
   - Tạo một danh sách **a_column** chứa tên các đặc trưng cần huấn luyện.
2. Chuẩn bị dữ liệu để huấn luyện và kiểm tra:
   - Tách dữ liệu huấn luyện và dữ liệu kiểm tra từ tập dữ liệu gốc bằng cách chọn chỉ các cột đặc trưng quan tâm.
3. Huấn luyện dữ liệu theo mô hình hồi quy tuyến tính:
   - Gọi phương thức **fit()** của đối tượng **'LinearRegression'** để huấn luyện mô hình.
4. Dự đoán kết quả và đánh giá mô hình:
   - Dự đoán kết quả dựa trên tập kiểm tra.
   - Tính toán sai số tuyệt đối trung bình (MAE).

b Kết quả dự đoán

$$MAE = 104863.777$$

$$Salary = -22756.513 * Gender + 804.503 * 10percentage$$
$$+1294.655 * 12percentage - 91781.898 * CollegeTier + 34552.286 * Domain$$
$$+23182.389 * Degree + 1437.549 * collegeGPA - 8570.663 * CollegeCityTier$$
$$+147.858 * English + 152.888 * Logical + 117.222 * Quant$$

c Nhận xét kết quả

1. **Giới tính (Gender):** Đặc trưng này có thể không ảnh hưởng lớn đến dự đoán mức lương, vì mức lương thường phải dựa vào khả năng và kinh nghiệm công việc hơn là giới tính.
2. **Điểm trung bình lớp 10 và lớp 12 (10percentage và 12percentage):** Điểm số trong giai đoạn học trung học có thể phản ánh khả năng học tập của ứng viên. Nhưng cần phải xem xét cẩn thận, vì điểm số này có thể thay đổi theo hệ thống chấm điểm của từng trường học.
3. **Tier của trường đại học (CollegeTier):** Một trường thuộc loại tier cao có thể cung cấp môi trường học tập tốt hơn và cơ hội hơn để phát triển kỹ năng. Điều này có thể ảnh hưởng đến khả năng làm việc và mức lương sau này.
4. **Loại bằng cấp (Degree) và GPA của trường đại học (collegeGPA):** Loại bằng cấp và GPA của trường đại học có thể thể hiện kiến thức chuyên môn và khả năng học tập của ứng viên. Những người có bằng cấp liên quan đến ngành công việc có thể có cơ hội cao hơn để đạt được mức lương tốt hơn.
5. **Tier của thành phố trường đại học (CollegeCityTier):** Nếu trường đại học nằm ở thành phố tier cao, có thể đề xuất rằng môi trường sống và học tập tại đó tốt hơn, ảnh hưởng đến khả năng phát triển và làm việc.
6. **Điểm tiếng Anh, logic và toán (English, Logical, Quant):** Điểm số trong các kỹ năng tiếng Anh, logic và toán có thể ảnh hưởng đến khả năng giải quyết vấn đề và thể hiện khả năng làm việc của ứng viên.
7. **Domain kiến thức (Domain):** Đặc trưng này có thể phản ánh khả năng chuyên môn của ứng viên đối với lĩnh vực công việc cụ thể. Người có kiến thức sâu rộng về lĩnh vực này có thể được trả mức lương cao hơn.

3.3 Yêu cầu 1b

Mô hình gồm đặc trưng:`'conscientiousness', 'agreeableness', 'extraversion', 'nueroticism', 'openess_to_experience'`

a Mô tả

1. Chọn các đặc trưng theo yêu cầu:
   - Tạo một danh sách **b_column** chứa tên các đặc trưng cần huấn luyện.
2. Chuẩn bị dữ liệu để huấn luyện và kiểm tra:
   - Tách dữ liệu huấn luyện và dữ liệu kiểm tra từ tập dữ liệu gốc bằng cách chọn chỉ các cột đặc trưng quan tâm.
3. Huấn luyện dữ liệu theo mô hình hồi quy tuyến tính:
   - Gọi phương thức **fit()** của đối tượng **'LinearRegression'** để huấn luyện mô hình.
4. Dự đoán kết quả và đánh giá mô hình:
   - Dự đoán kết quả dựa trên tập kiểm tra.
   - Tính toán sai số tuyệt đối trung bình (MAE).

b Kết quả dự đoán

$$MAE = 291019.693$$
$$Salary = -56546.308 * nueroticism$$

c Nhận xét kết quả

- Kết quả cross-validation:

STT	Mô hình	MAE
1	conscientiousness	306173.277
2	agreeableness	300862.548
3	extraversion	306854.236
4	nueroticism	299292.731
5	openess_to_experience	303083.905

- Kết quả cross validation cho thấy **nueroticism** cho ra kết quả MAE tốt nhất. Bên cạnh đó thì **agreeableness** cũng cho kết quả khá tốt. Nên có thể kết luận đây là hai tính cách ảnh hưởng nhiều nhất tới mức lương.
- Ở Đức, sự ổn định cảm xúc đóng góp khoảng 4% vào mức lương, trong khi những người có đặc điểm lo âu cao thường mất khoảng 3% tiền lương. Khía

cạnh ổn định cảm xúc có mối liên hệ tích cực với mức lương ở Hà Lan. Nhân viên có đặc điểm lo âu cao tại Vương quốc Anh thường trải qua việc mất khoảng 6% tiền lương. [2]

- Cụ thể hơn, dù các đặc điểm tính cách khác nhau có tác động khác nhau đối với mức lương ở các quốc gia, hai đặc điểm (sẵn lòng và lo âu, trái ngược với sự ổn định cảm xúc) ảnh hưởng tiêu cực đến mức lương ở tất cả các quốc gia nghiên cứu.
- Một số yếu tố tính cách khác có tác động tích cực đối với mức lương, tuy nhiên không thể thấy tác động thống nhất như hai đặc điểm tính cách trên tại tất cả các quốc gia. Tình hình này có thể hiểu được với yếu tố lo âu (những người không thể kiểm soát tâm trạng và tinh thần của họ). Tuy nhiên, với tính sẵn lòng - cái mà được định nghĩa là: "Khuynh hướng hành động một cách hợp tác, không ích kỷ." Đây là một phẩm chất tốt, thể hiện người đó quan tâm đến công ty và nỗ lực trở thành một thành viên đội nhóm xuất sắc, tạo ra giá trị cho công ty. Nhưng hành vi này lại đi kèm với một chi phí. Như một trong những nhà nghiên cứu đã nói, "là người tốt không có nghĩa là bạn được trả tiền" (Heineck, 2007, trang 1). Những người thể hiện phẩm chất như vậy sẽ phải chịu thiệt hại về thu nhập. [9]

3.4 Yêu cầu 1c

Mô hình gồm đặc trưng:`'English', 'Logical', 'Quant'`

a Mô tả

1. Chọn các đặc trưng theo yêu cầu:
   - Tạo một danh sách **c_column** chứa tên các đặc trưng cần huấn luyện.
2. Chuẩn bị dữ liệu để huấn luyện và kiểm tra:
   - Tách dữ liệu huấn luyện và dữ liệu kiểm tra từ tập dữ liệu gốc bằng cách chọn chỉ các cột đặc trưng quan tâm.
3. Huấn luyện dữ liệu theo mô hình hồi quy tuyến tính:
   - Gọi phương thức **fit()** của đối tượng '**LinearRegression**' để huấn luyện mô hình.
4. Dự đoán kết quả và đánh giá mô hình:
   - Dự đoán kết quả dựa trên tập kiểm tra.
   - Tính toán sai số tuyệt đối trung bình (MAE).

b Kết quả dự đoán

$$MAE = 106819.578$$
$$Salary = 585.895 * Quant$$

c Nhận xét kết quả

- Kết quả cross-validation:

STT	Mô hình	MAE
1	English	121873.901
2	Logical	120269.531
3	Quant	118051.217

- Từ kết quả trên, có thể đưa ra kết luận bài kiểm tra về định lượng **Quant** có tác động lớn nhất tới lương của kỹ sư sau tốt nghiệp. Song, không có sự chênh lệch quá lớn về giá trị đối với hai bài kiểm tra về khả năng **Logic** và trình độ tiếng anh **English**. Nói cách khác, cả 3 bài kiểm tra đều có sự tác động lớn đối với khả năng dự đoán lương của kỹ sư sau tốt nghiệp.
- Người lao động là một tài nguyên quan trọng để doanh nghiệp tạo ra giá trị. Trong quá khứ, các quản lý cấp cao là những người đã chấp nhận ủy quyền và đóng vai trò như người quản lý hoạt động hàng ngày của doanh nghiệp. Tuy nhiên, do sự phức tạp ngày càng tăng của cấu trúc doanh nghiệp hiện tại, nhu cầu của các quản lý cấp cao đã trở nên cụ thể hơn.
- Ngoài việc quản lý và sắp xếp hoạt động hàng ngày của doanh nghiệp, họ còn chịu trách nhiệm về việc đưa ra quyết định về các vấn đề quan trọng (Hah và Freeman, 2015). Với vai trò chủ chốt trong việc đưa ra quyết định về đổi mới doanh nghiệp, các quản lý cấp cao chịu trách nhiệm dẫn dắt toàn bộ quá trình hoạt động đổi mới của doanh nghiệp. Nếu những quản lý cấp cao chịu trách nhiệm cho hoạt động đổi mới thiếu sự kỹ năng tư duy (**logic**) hay kỹ năng phân tích (**Quant**), điều này sẽ ảnh hưởng tiêu cực đến sự phát triển của doanh nghiệp, và khả năng đổi mới của họ sẽ không được thể hiện một cách hiệu quả.
- Do đó, thông qua các động viên thích hợp cho nhân viên, có thể tăng cường nhận thức của nhân viên về đầu tư đổi mới và thúc đẩy sự đổi mới công nghệ của doanh nghiệp, từ đó duy trì sự phát triển ổn định dài hạn của doanh nghiệp trong cạnh tranh thị trường (Fang và Shi, 2016; Kong và cộng sự, 2017).

- Tóm lại, việc đưa ra các quyết định dựa trên nền tảng tốt về quan sát cũng như số học, phân tích **logic và quant** sẽ là tiền đề cho sự phát triển của doanh nghiệp, và theo nền kinh tế tư bản thì người chủ sẽ trả nhiều hơn để khiến người nhân viên trở nên trung thành cũng như ngày càng học hỏi, phát triển để có thể cống hiến lâu dài cho doanh nghiệp. [3]

3.5 Yêu cầu 1d

a Mô hình 1: Brute Force

```
  "10percentage", "12percentage", "CollegeTier", "collegeGPA", "CollegeCityTier",
 "Quant", "Domain", "ComputerProgramming", "ElectronicsAndSemicon", "ComputerScience
","MechanicalEngg", "conscientiousness", "agreeableness", "nueroticism", "openess_to_experi
"
```

- Quy trình tìm ra mô hình:
  1. Sử dụng thuật toán BruteForce để tìm ra tổ hợp các đặc trưng cho ra kết quả MAE tốt nhất từ các mô hình:
     + Mô hình từ câu 1a, b, c
     + Mô hình gồm 7 đặc trưng về điểm số trong phần kỹ thuật.
  2. Kết hợp kết quả tốt nhất từ các mô hình thành 1 mô hình kết quả, huấn luyện trên mô hình đó, và chạy lại trên tập kiểm tra.

b Mô hình 2: XGBoost

```
  "conscientiousness", "Logical", "10percentage", "collegeGPA", "agreeableness",
"Quant", "ComputerProgramming", "English", "Domain", "12percentage", "nueroticism",
 "openess_to_experience", "ComputerScience", "extraversion", "Gender", "CollegeTier
"
```

- Quy trình tìm ra mô hình:
  1. Sử dụng thuật toán XGBoost để đánh giá về tầm quan trọng của các đặc trưng trong mô hình. 2.2
  2. Các đặc trưng cao hơn đóng góp nhiều hơn vào khả năng dự đoán mô hình.
  3. Chọn các đặc trưng có mức độ quan trọng cao hơn 15% để đưa vào mô hình.

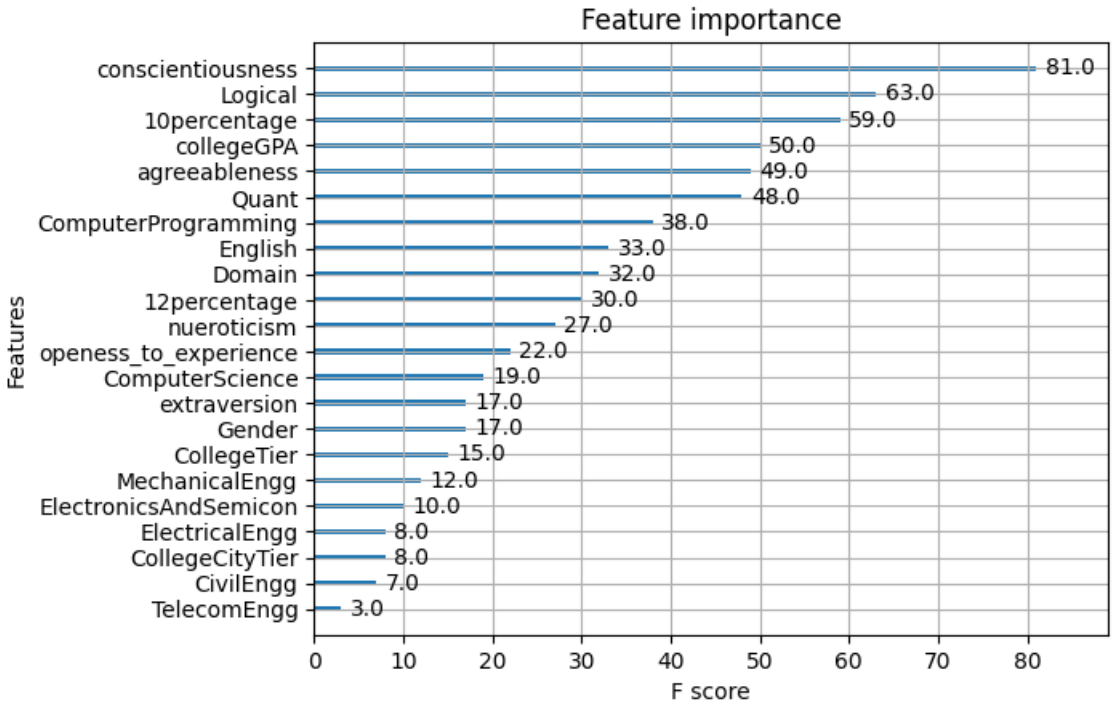

Figure 2: Mức độ quan trọng của 23 đặc trưng

c Mô hình 3: Genetic Algorithm

```
'10percentage', '12percentage', 'collegeGPA', 'English', 'Quant', 'ComputerProgramming
', 'ElectronicsAndSemicon', 'ComputerScience', 'MechanicalEngg', 'conscientiousness
', 'agreeableness', 'extraversion', 'nueroticism', 'openess_to_experience'
```

- Quy trình tìm ra mô hình:
  1. Sử dụng thuật toán Genetic để tạo ra các biến thể các đặc trưng. 2.3
  2. So sánh các kết quả MAE của từng biến thể, từ đó tìm ra biến thể cho kết quả MAE tốt nhất trên tập huấn luyện.

d Kết quả:

- Kết quả cross-validation của các mô hình:

STT	Mô hình	MAE
1	Mẫu 1	110964.179
2	Mẫu 2	110651.346
3	Mẫu 3	112383.245

- Mô hình 2 (XGBoost) cho ra kết quả cross-validation tốt nhất trong 3 mô hình trên.

- Kết quả MAE của mô hình 2:

$$MAE = 102431.57$$

- Công thức hồi quy:

$$\begin{aligned} Salary = &-20216.903 * conscientiousness + 132.912 * Logical \\ &+887.524 * 10percentage + 1689.069 * collegeGPA + 95.513 * Quant \\ &+16789.581 * agreeableness + 104.259 * ComputerProgramming \\ &+148.766 * English + 22104.286 * Domain + 971.041 * 12percentage \\ &-9649.493 * nueroticism - 6317.617 * openess\_to\_experience \\ &-164.058 * ComputerScience + 5018.386 * extraversion \\ &-23438.408 * Gender - 79706.706 * CollegeTier \end{aligned}$$

e Nhận xét kết quả:

1. **Gradient Boosting và Regularization:** XGBoost là một biến thể của Gradient Boosting - kỹ thuật tạo ra một tập hợp các cây quyết định yếu. Quá trình huấn luyện XGBoost tối ưu hóa hàm mất mát bằng cách điều chỉnh các cây theo cách lặp. Sự kết hợp của nhiều cây giúp mô hình tập trung vào các điểm dữ liệu khó dự đoán, cải thiện khả năng tổng quát hóa.
2. **Overfitting:** XGBoost cũng áp dụng các kỹ thuật regularization như hạn chế chiều sâu của cây và tạo ra các nút lá có hệ số giảm cân nhắc (shrinkage coefficient) để ngăn chặn quá khớp (overfitting).
3. **Xử lý Dữ liệu không Đồng Nhất:** XGBoost có khả năng xử lý tốt dữ liệu không đồng nhất, bao gồm cả dữ liệu số và dữ liệu rời rạc. Nó có thể xử lý các dạng dữ liệu khác nhau mà không cần nhiều bước tiền xử lý.
4. **Quan Trọng Đặc Trưng và Tinh chỉnh tham số:** XGBoost cung cấp tính năng đánh giá quan trọng đặc trưng, giúp xác định đặc trưng nào đóng góp nhiều vào dự đoán. Điều này giúp tinh chỉnh mô hình bằng cách tập trung vào những đặc trưng quan trọng nhất.
5. **Tốc Độ Tính Toán và Hiệu Suất:** XGBoost được tối ưu hóa để chạy hiệu quả trên cả dữ liệu lớn. Các kỹ thuật như "column block" và "row block" giúp tăng tốc quá trình tính toán và làm cho XGBoost nhanh hơn so với nhiều thuật toán khác.

Nguồn tài liệu tham khảo: [12] [13]

4 Các hàm chức năng trong chương trình:

4.1 Lớp LinearRegression

a def fit(self, X, y)

+ **Tham số truyền vào:**

  - '**X**': Là ma trận dữ liệu đầu vào, trong đó mỗi hàng là một điểm dữ liệu, và mỗi cột là một đặc trưng.
  - '**y**': Là vector kết quả tương ứng với dữ liệu đầu vào X.

+ **Mô tả hàm:**

  - Thực hiện quá trình huấn luyện mô hình hồi quy tuyến tính dựa trên dữ liệu đầu vào '**X**' và kết quả mong muốn '**y**'.
  - Trong hàm này, ma trận X được biến đổi bằng ma trận giả nghịch đảo MoorePenrose (pseudoinverse) thông qua hàm '**np.linalg.pinv(X)**'. Quá trình này giúp tính toán các trọng số self.w của mô hình hồi quy tuyến tính. Trọng số này được tính bằng cách nhân ma trận giả nghịch đảo với vector y.

b def get_params(self)

Hàm này trả về các tham số của mô hình đã được học, tức là vector trọng số '**self.w**'.

c def predict(self, X)

+ **Tham số truyền vào:**

  - '**X**': Là ma trận dữ liệu đầu vào cần được dự đoán.

+ **Mô tả hàm:**

  - Hàm này thực hiện dự đoán đầu ra dựa trên dữ liệu đầu vào X.
  - Quá trình dự đoán được thực hiện bằng cách tính tổng của tích vô hướng giữa vector trọng số '**self.w**' và từng hàng của ma trận '**X**'. Kết quả dự đoán được trả về là một mảng chứa các giá trị dự đoán tương ứng với từng dòng của ma trận '**X**'.

4.2 Hàm tìm mô hình tốt nhất

```
def find_best_model(model: dict) -> tuple:
```

a Tham số đầu vào

- **model (dict)**: Một từ điển chứa các mô hình và giá trị tương ứng. Mỗi giá trị là một danh sách (list) của các số dùng để tính trung bình.

b Tham số trả về

- **tuple**: Một tuple chứa hai giá trị:
  - **min_mean (float)**: Giá trị trung bình nhỏ nhất được tìm thấy trong danh sách các giá trị của các mô hình.
  - **best_key (str || None)**: Khóa của mô hình có giá trị trung bình nhỏ nhất. Nếu không tìm thấy mô hình phù hợp, giá trị này sẽ là `None`.

c Mô tả hàm

1. Gán **min_mean** giá trị vô cùng lớn (dương vô cùng) để đảm bảo rằng mọi giá trị trung bình thực tế sẽ nhỏ hơn giá trị này.
2. Gán **best_key** giá trị `None` ban đầu. Giá trị này sẽ thay đổi sau mỗi vòng lặp để lưu khóa của mô hình có giá trị trung bình nhỏ nhất.
3. Duyệt qua từng cặp khóa và giá trị trong **model**.
4. Tính giá trị trung bình của danh sách giá trị tương ứng với khóa hiện tại bằng cách sử dụng **np.mean(values)**.
5. So sánh giá trị trung bình vừa tính được với **min_mean**. Nếu nhỏ hơn, cập nhật **min_mean** và **best_key** thành giá trị trung bình hiện tại và khóa tương ứng.
6. Sau khi hoàn thành vòng lặp, trả về tuple chứa **min_mean** và **best_key** như kết quả tìm kiếm mô hình tốt nhất.

4.3 Hàm tính MAE (Mean Absolute Error) với các đặc trưng

```
def calc_MAE_with_features(kFold, features):
```

a Tham số đầu vào

- **kFold**: Đối tượng chứa thông tin về việc chia fold trong cross-validation.
- **features** (**dict**): Một từ điển chứa các đặc trưng. Mỗi khóa là tên đặc trưng, và giá trị tương ứng là danh sách chứa các giá trị MAE cho từng fold.

b Tham số trả về

- **dict**: Trả về từ điển **features** đã được cập nhật với các giá trị MAE mới tính được cho mỗi fold.

c Mô tả hàm

1. Duyệt qua mỗi fold trong cross-validation.
2. Chia dữ liệu mục tiêu (**y_train**) của fold hiện tại thành tập huấn luyện (**y_features_train**) và tập kiểm tra (**y_features_test**).
3. Với mỗi đặc trưng **i** trong **features.keys()**, thực hiện các bước sau:
   a. Chia dữ liệu đặc trưng tương ứng (**X_train[i]**) của fold hiện tại thành tập huấn luyện (**x_features_train**) và tập kiểm tra (**x_features_test**).
   b. Khởi tạo một mô hình hồi quy tuyến tính (**lr**) và huấn luyện nó trên tập huấn luyện của đặc trưng và tập huấn luyện của mục tiêu.
   c. Dự đoán giá trị mục tiêu trên tập kiểm tra của đặc trưng (**y_pred**).
   d. Tính giá trị MAE giữa giá trị thực tế và giá trị dự đoán bằng cách sử dụng hàm **MAE(y_features_test, y_pred)** và thêm giá trị MAE vào danh sách trong từ điển **features** tương ứng với đặc trưng **i**.
4. Trả về từ điển **features** đã được cập nhật với các giá trị MAE mới tính được cho mỗi fold.

5 Nhận xét tổng quát

- Mô hình câu 1a 3.2
  + Kết quả MAE: 104863.777
  + Mô hình này chứa 11 đặc trưng, cho ra kết quả MAE khá nhỏ so với câu 1b và 1c.
  + Nghĩa là, mô hình câu 1a cho ra kết quả dự đoán mức lương khá tốt, do bao gồm các đặc trưng quan trọng, chiếm tỉ lệ cao trong đánh giá lương như: khả năng tích toán, thống kê, đưa ra quyết định hay các kĩ năng, cũng như điều kiện về kinh tế, xã hội.
- Mô hình câu 1b 3.3:
  + Kết quả MAE: 291019.693
  + Từ kết quả MAE trên, chứng tỏ các yếu tố liên quan đến cảm xúc không có tác động quá lớn đối với việc dự đoán mức lương (MAE lớn gần gấp 3

so với câu 1a).

+ Nói cách khác, việc chọn ra chỉ 1 đặc trưng thì chưa đủ để có thể dự đoán mức lương.

- Mô hình câu 1c 3.4:

+ Kết quả MAE: 106819.578

+ Từ kết quả MAE trên, có thể kết luận rằng các kĩ năng liên quan đến logic, toán học, phân tích, định lượng đóng vai trò rất quan trọng trong việc dự đoán mức lương, dù chỉ là 1 đặc trưng (so với câu 1a, 1b).

- Mô hình 1d 3.5:

+ Kết quả MAE: 102431.57

+ Từ kết quả MAE trên, có thể đưa ra kết luận rằng: đây là mô hình cho ra khả năng dự đoán mức lương tốt nhất trong 4 mô hình trên.

+ Kết quả này có được nhờ vào thuật toán của XGBoost để tối ưu cả về hiệu suất, lẫn thời gian, hạn chế các vấn đề liên quan đến dữ liệu như: nhiễu, quá khớp (overfitting), không đồng nhất, ...

Tài liệu tham khảo

[1] Inc. Hosted by OVHcloud AQR Capital Management; NumFOCUS. "Python Data Analysis Library". In: (2008).
https://pandas.pydata.org/.

[2] Braakmann. "The relation between cognitive and noncognitive abilities to economic productivity". In: *Labour Economics, 16(2), 209-216* (2009).
https://kse.ua/kse-research/relationship-between-your-personality-and-your-salary-level/.

[3] J. Molineux C. A. Lengnick-Hall. "The Relationship Between Employee Salary Gap and Enterprise Innovation". In: *Frontiers in Psychology, 11, 1749* (2020).
https://www.frontiersin.org/articles/10.3389/fpsyg.2020.01749/full.

[4] NumPy Developers. "NumPy". In: *Numpy.ndarray* (2008 - 2022).
https://numpy.org/doc/stable/reference/generated/numpy.ndarray.html.

[5] Scikit-learn developers. "Scikit-learn Machine Learning in Python". In: *Model Selection* (2022).
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html.

[6] xgboost developers. "XGBoost Documentation". In: *XGBoost in Machine Learning* (2022).
https://xgboost.readthedocs.io/en/stable/.

[7] Python Software Foundation. "itertools — Functions creating iterators for efficient looping". In: *Function Programming Modules* (2023).
https://docs.python.org/3/library/itertools.html.

[8] Cơ sở trí tuệ nhân tạo GV. Nguyễn Tiến Huy. "Genetic Algorithm". In: *First-Order-Logic* (Second-Semeter, 2023).

[9] Denissen; J. J. A.; Bleidorn; W.; Hennecke. "Intraindividual variability in personality traits: A practical guide to the challenges in data collection and data analysis". In: *Journal of Personality and Social Psychology, 115(4), 487-509* (2018).

[10] PhD Jason Brownlee. "Feature importance and Feature selection with XGBoost in python". In: *Relative Importance of Predictor Variables* (2016).
https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/.

[11] Shanthababu Pandian. "KFold Cross Validation Technique". In: *Data Science Blogathon* (2022).
https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/.

[12] Vishal Morde & Venkat Anurag Setty. "XGBoost Algorithm: Long May She Reign!" In: *AI & Machine Learning* (2019).
https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d.

[13] Simplilearn. "An Introduction to XGBoost Algorithm in Machine Learning". In: *AI & Machine Learning* (2023).
https://www.simplilearn.com/what-is-xgboost-algorithm-in-machine-learning-article.

[14] The Matplotlib development team. "Matplotlib: Visualization with Python". In: (2012 – 2023).
https://matplotlib.org/.

[15] GV. Phan Thị Phương Uyên. "Hướng dẫn đồ án 03". In: *Học kỳ 3 - Năm 2* (2023).