

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN
LINEAR REGRESSION

Môn học: Toán ứng dụng và thống kê cho Công Nghệ Thông Tin

Giảng viên hướng dẫn: Phan Thị Phương Uyên

Sinh viên thực hiện: Đoàn Ngọc Mai

Mã số sinh viên: 21127104

Thành phố Hồ Chí Minh, ngày 22 tháng 8 năm 2023

Mục lục

1.	Thông tin sinh viên	3
2.	Các phần đã hoàn thành	3
3.	Liệt kê những thư viện đã sử dụng	3
4.	Ý tưởng thực hiện và mô tả các hàm	4
[1]	Class OLSLinearRegression	4
i.	fit(self, X, y)	4
ii.	get_params(self)	4
iii.	predict(self, X)	4
[2]	Hàm mae(y, y_hat)	4
[3]	Hàm perform_cross_validation(X_train, y_train, personality_features, k=5) 4	
5.	Ý tưởng của các phần.....	5
[1]	Câu 1a.....	5
[2]	Câu 1b.....	6
[3]	Câu 1c.....	9
[4]	Câu 1d.....	12
6.	Nhận xét chung với tất cả các mô hình.....	18
7.	Tài liệu tham khảo.....	19

1. Thông tin sinh viên

- Họ và tên: Đoàn Ngọc Mai
- MSSV: 21127014
- Lớp: 21CLC05

2. Các phần đã hoàn thành

Công việc	Mức độ hoàn thành
Câu 1a	100%
Câu 1b	100%
Câu 1c	100%
Câu 1d	100%

3. Liệt kê những thư viện đã sử dụng

- **import pandas as pd:**
 - Đọc dữ liệu từ tệp CSV với `pd.read_csv()`
 - Tạo dataframe để lưu trữ kết quả cross-validation. Các thông tin về đặc trưng và giá trị MAE (Mean Absolute Error) được lưu trữ trong dataframe này.
- **import numpy as np:** dùng để
 - Tính ma trận nghịch đảo với `np.linalg.inv()`
 - Tính tổng các phần tử trong mảng với `np.sum()`
 - Được sử dụng để tạo một mảng các số liên tiếp với các giá trị nằm trong một khoảng cụ thể với `np.arange()`
 - Chuyển đổi dataframe thành mảng numpy để hiển thị kết quả một cách dễ dàng.
- **from sklearn.model_selection import KFold [1]:** giúp thực hiện phân chia dữ liệu thành các fold (phân đoạn) để thực hiện quá trình cross-validation.
- **import seaborn as sns, import matplotlib.pyplot as plt:** được sử dụng để tạo biểu đồ heatmap để hiển thị ma trận tương quan. Biểu đồ heatmap giúp dễ dàng nhận thấy mẫu tương quan giữa các biến dữ liệu.

4. Ý tưởng thực hiện và mô tả các hàm

[1] *Class OLSLinearRegression*

Đây là một lớp dùng để triển khai mô hình Linear Regression bằng phương pháp OLS (Ordinary Least Squares).

i. fit(self, X, y)

Phương thức này thực hiện việc tìm các trọng số tối ưu cho mô hình Linear Regression bằng phương pháp OLS. Đầu vào X là ma trận đặc trưng, y là vector biến phụ thuộc. Nó tính toán các trọng số w dựa trên công thức OLS và trả về chính đối tượng lớp sau khi huấn luyện.

ii. get_params(self)

Phương thức này trả về các trọng số của mô hình, trong trường hợp này là trọng số w.

iii. predict(self, X)

Phương thức này thực hiện dự đoán đầu ra dựa trên ma trận đặc trưng X và trọng số đã học. Nó tính toán dự đoán bằng cách nhân ma trận đặc trưng với trọng số và trả về một mảng dự đoán.

[2] *Hàm mae(y, y_hat)*

Đây là một hàm tính giá trị Mean Absolute Error (MAE) giữa các giá trị thực tế y và dự đoán y_hat. MAE được tính bằng cách lấy trung bình giá trị tuyệt đối của sự sai lệch giữa y và y_hat.

[3] *Hàm perform_cross_validation(X_train, y_train, personality_features, k=5)*

- **Đầu vào:**

- X_train: Tập dữ liệu đặc trưng để huấn luyện.
- y_train: Vector biến phụ thuộc tương ứng với X_train.
- personality_features: Danh sách các đặc trưng cần kiểm tra trong quá trình cross-validation.
- k: Số lượng folds trong cross-validation (mặc định là 5).

- **Đầu ra:**

- `best_feature`: Tên của đặc trưng tốt nhất dựa trên giá trị MAE trung bình thấp nhất.
- `prediction_df`: DataFrame chứa kết quả cross-validation, bao gồm tên đặc trưng và giá trị MAE trung bình.
- **Công dụng**: Hàm này thực hiện quá trình cross-validation để tìm ra đặc trưng tốt nhất từ danh sách `personality_features`. Nó lặp qua từng đặc trưng, thực hiện k-fold cross-validation, tính MAE trung bình cho mỗi đặc trưng và cuối cùng tìm ra đặc trưng có giá trị MAE thấp nhất. Kết quả được trả về thông qua biến `best_feature` và DataFrame `prediction_df` chứa các giá trị MAE và đặc trưng tương ứng.
- **Ưu điểm**: Hàm này giúp tự động thực hiện quá trình cross-validation và tìm ra đặc trưng tốt nhất dựa trên giá trị MAE, tiết kiệm thời gian và công sức cho người dùng.
- **Nhược điểm**: Hàm này được thiết kế để thực hiện cross-validation cho các đặc trưng cụ thể và phương pháp biến đổi đã được định nghĩa sẵn. Nếu muốn thay đổi phép biến đổi hoặc loại mô hình khác, có thể cần phải sửa đổi hàm này tương ứng.

5. Ý tưởng của các phần

[1] Câu 1a

- **Ý tưởng**:
 - Tạo ra danh sách tên của 11 đặc trưng cần xét để sử dụng trong việc huấn luyện mô hình.
 - Tạo DataFrame `X_train_a` chứa các đặc trưng đã chọn từ tập huấn luyện và Series `y_train_a` chứa giá trị biến phụ thuộc từ tập huấn luyện. `iloc[:, -1]` được sử dụng để chọn cột cuối cùng của DataFrame train, giả định là cột chứa giá trị biến phụ thuộc.
 - Tương tự như trên, tạo DataFrame `X_test_a` chứa các đặc trưng đã chọn từ tập kiểm tra và Series `y_test_a` chứa giá trị biến phụ thuộc từ tập kiểm tra.
 - Tạo một đối tượng của lớp `OLSLinearRegression`, sau đó sử dụng phương thức `fit` để huấn luyện mô hình với các đặc trưng `X_train_a` và giá trị biến phụ thuộc `y_train_a`.
 - Gọi phương thức `get_params` để lấy ra các trọng số tối ưu của mô hình Linear Regression đã huấn luyện.
 - Sử dụng mô hình đã huấn luyện để dự đoán giá trị biến phụ thuộc trên tập kiểm tra `X_test_a` và lưu kết quả dự đoán vào mảng `y_hat`.

- Cuối cùng, in ra giá trị Mean Absolute Error (MAE) giữa giá trị thực tế y_{test_a} và giá trị dự đoán y_{hat} trên tập kiểm tra.
- **Kết luận:**
 - Ở câu a này, công việc là thực hiện việc huấn luyện mô hình Linear Regression với các đặc trưng đã chọn và đo lường độ hiệu quả của mô hình bằng giá trị MAE trên tập kiểm tra. Sau cùng, chúng ta sẽ có được công thức sau:
 - MAE: 104863.77754033149

Công thức hồi quy (phần trọng số làm tròn đến 3 chữ số thập phân, ví dụ 0.012345 → 0.012)

$$\begin{aligned} \text{Salary} = & -22756.513 \times (\text{Gender}) + 804.503 \times (10\text{percentage}) + 1294.655 \times (12\text{percentage}) + (-91781.898) \times (\text{CollegeTier}) + 23182.389 \\ & \times (\text{Degree}) + 1437.549 \times (\text{collegeGPA}) + (-8570.662) \times (\text{CollegeCityTier}) + 147.858 \times (\text{English}) + 152.888 \times (\text{Logical}) + 117.222 \times (\text{Quant}) \\ & + 34552.286 \times (\text{Domain}) \end{aligned}$$

- **Nhận xét:**
 - Kết quả MAE là 104,863.78 có thể được hiểu như sau: trung bình mỗi dự đoán lệch đi khoảng 104,863.78 đơn vị so với giá trị thực tế trên tập kiểm tra. Điều này thể hiện mức độ sai khác trung bình giữa dự đoán của mô hình và thực tế. Giá trị MAE càng thấp, mô hình càng chính xác.
 - Công thức hồi quy cuối cùng được trình bày thể hiện cách mô hình dự đoán lương (Salary) dựa trên các đặc trưng. Các hệ số (coefficients) trong công thức là trọng số tương ứng với từng đặc trưng. Ví dụ: hệ số -22756.513 cho đặc trưng *Gender* cho thấy khi *Gender* tăng lên 1 đơn vị, lương sẽ giảm đi 22756.513 đơn vị (đương nhiên, cần kiểm tra kiểu đặc trưng và thang đo của nó để hiểu rõ hơn).
 - Tuy nhiên, giá trị MAE rất cao (104,863.78) có thể chỉ ra rằng mô hình không hoạt động tốt, và dự đoán của nó có sự sai khác lớn so với thực tế. Có thể có nhiều nguyên nhân dẫn đến sự sai khác lớn này, bao gồm sự phức tạp của mối quan hệ giữa các đặc trưng và lương, việc mô hình không phù hợp hoặc có các giả định không đúng, và nhiễu trong dữ liệu. Để cải thiện hiệu suất của mô hình, có thể cân nhắc đến xem xét chọn lọc đặc trưng tốt hơn.

[2] *Câu 1b*

- **Ý tưởng code:**
 - Mục tiêu của mã code là tìm ra và đánh giá một mô hình Linear Regression dựa trên đặc trưng tính cách tốt nhất từ danh sách đã cho.
 - Quá trình cross-validation được thực hiện để đánh giá hiệu suất của mô hình trên từng đặc trưng.
 - Sau khi tìm ra đặc trưng tốt nhất, mô hình được huấn luyện lại với đặc trưng này và đo lường hiệu suất trên tập kiểm tra.

- **Giải thích code đã hoàn thành:**

- Đầu tiên, danh sách `personality_features` chứa tên của các đặc trưng liên quan đến tính cách.
- Mục tiêu là tìm ra đặc trưng tốt nhất từ danh sách `personality_features` thông qua cross-validation để dự đoán giá trị mục tiêu. Cụ thể:
- Một vòng lặp duyệt qua các đặc trưng tính cách, trong mỗi vòng lặp:
- Tạo ma trận `X_train_feature` chứa đặc trưng hiện tại từ tập huấn luyện và chuẩn bị vector giá trị mục tiêu `y_train_feature` tương ứng.
- Thực hiện k-fold cross-validation (số lượng folds $k=5$) trên ma trận đặc trưng và tính toán MAE trung bình cho mỗi fold.
- Tính trung bình của các giá trị MAE và lưu vào danh sách `list_mae`.
- Sau khi hoàn thành vòng lặp, danh sách `list_mae` chứa thông tin về MAE trung bình cho mỗi đặc trưng.
- Tiếp theo, danh sách `list_mae` được sắp xếp theo giá trị MAE tăng dần và đặc trưng có MAE thấp nhất được xác định là đặc trưng tốt nhất.
- Mô hình Linear Regression mới sẽ được huấn luyện chỉ trên đặc trưng tốt nhất với toàn bộ tập dữ liệu huấn luyện.
- Các thông số của mô hình (trọng số) được lấy ra và in ra màn hình.
- Cuối cùng, mô hình với đặc trưng tốt nhất được sử dụng để dự đoán trên tập kiểm tra và tính giá trị MAE giữa dự đoán và giá trị thực tế.

- **Ưu điểm:**

- Xác định đặc trưng tốt nhất dựa trên cross-validation giúp chọn lọc các đặc trưng quan trọng.
- Cách tiếp cận này tự động và hiệu quả trong việc tìm đặc trưng và đánh giá mô hình.

- **Phân tích:**

- *Với $k < 1000$*
 - Kết quả cuối cùng cho ra đặc trưng tốt nhất là neuroticism.
 - Em có thử nghiệm với k trong khoảng 7 đến 999, nhưng giá trị gần như không thay đổi và thời gian chạy cũng không có quá nhiều sự chênh lệch.

The best feature is: nueroticism

	Feature	MAE
1	conscientiousness	306309.201775
2	agreeableness	300912.677678
3	extraversion	307030.102690
4	nueroticism	299590.049823
5	openess_to_experience	302957.691854

K=5

○ *Với những giá trị $k \geq 1000$*

- Nhưng khi thử với $k=1000$, thời gian chạy rõ ràng đã lâu hơn nhiều, MAE có sự thay đổi, nhưng đặc trưng tốt nhất vẫn là neuroticism.

The best feature is: nueroticism

:

	Feature	MAE
1	conscientiousness	307542.226327
2	agreeableness	301690.721584
3	extraversion	308221.668513
4	nueroticism	300146.748739
5	openess_to_experience	304388.368619

K=1000

● **Kết luận:**

- Ở câu b này, công việc là thực hiện là xây dựng mô hình sử dụng duy nhất 1 đặc trưng tính cách với các đặc trưng tính cách gồm conscientiousness, agreeableness, extraversion, nueroticism, openess_to_experience, tìm mô hình cho kết quả tốt nhất. Sau cùng, chúng ta sẽ có được công thức sau:

The best feature is: nueroticism

	Feature	MAE
1	conscientiousness	306309.201775
2	agreeableness	300912.677678
3	extraversion	307030.102690
4	nueroticism	299590.049823
5	openess_to_experience	302957.691854

MAE: 291019.693226953

Công thức hồi quy (phần trọng số làm tròn đến 3 chữ số thập phân, ví dụ 0.012345 → 0.012)

$$\text{Salary} = -56546.304 \times (\text{nueroticism})$$

- **Nhận xét:**

- Mô hình cuối cùng được xây dựng với đặc trưng tốt nhất là 'nueroticism'. Khi áp dụng mô hình này lên tập kiểm tra, giá trị MAE là 291,019.69. Điều này có nghĩa là trung bình mỗi dự đoán lệch đi khoảng 291,019.69 đơn vị so với giá trị thực tế trong tập kiểm tra. MAE cao cho thấy mô hình vẫn chưa đủ tốt và có sự sai khác lớn giữa dự đoán và thực tế.
- Công thức hồi quy cuối cùng chỉ có một đặc trưng duy nhất là 'nueroticism' và hệ số ứng với đặc trưng này là -56546.304. Điều này có thể được hiểu như sau: mỗi đơn vị tăng lên trong đặc trưng 'nueroticism' sẽ gây giảm 56546.304 đơn vị trong giá trị lương (Salary).
- Mô hình và kết quả này đều cho thấy sự không tốt của mô hình trong việc dự đoán lương dựa trên đặc trưng 'nueroticism'. MAE cao và hệ số rất lớn (-56546.304) đều cho thấy mô hình không thể hiện mối quan hệ tốt giữa 'nueroticism' và lương. Có thể nguyên nhân gốc rễ là đặc trưng này không đủ để giải thích biến đổi lương hoặc có thể mô hình hồi quy tuyến tính không phù hợp cho mối quan hệ này.
- Để cải thiện mô hình, có thể xem xét sử dụng nhiều đặc trưng hơn, thử các mô hình hồi quy khác nhau, tinh chỉnh siêu tham số, và xem xét việc chọn lọc các đặc trưng quan trọng hơn để xây dựng mô hình tốt hơn cho việc dự đoán lương.

[3] Câu 1c

- **Ý tưởng code:**

- Mục tiêu của mã code là tìm ra và đánh giá một mô hình Linear Regression dựa trên đặc trưng tính cách tốt nhất từ danh sách đã cho.
- Quá trình cross-validation được thực hiện để đánh giá hiệu suất của mô hình trên từng đặc trưng.
- Sau khi tìm ra đặc trưng tốt nhất, mô hình được huấn luyện lại với đặc trưng này và đo lường hiệu suất trên tập kiểm tra.
- **Giải thích code đã hoàn thành:**
 - Đầu tiên, danh sách `selected_features` chứa tên của các đặc trưng English, Logical, và Quant.
 - Mục tiêu là tìm ra đặc trưng tốt nhất từ danh sách `selected_features` thông qua cross-validation để dự đoán giá trị mục tiêu. Cụ thể:
 - Một vòng lặp duyệt qua các đặc trưng tính cách, trong mỗi vòng lặp:
 - Tạo ma trận `X_train_feature` chứa đặc trưng hiện tại từ tập huấn luyện và chuẩn bị vector giá trị mục tiêu `y_train_feature` tương ứng.
 - Thực hiện k-fold cross-validation (số lượng folds $k=5$) trên ma trận đặc trưng và tính toán MAE trung bình cho mỗi fold.
 - Tính trung bình của các giá trị MAE và lưu vào danh sách `list_mae`.
 - Sau khi hoàn thành vòng lặp, danh sách `list_mae` chứa thông tin về MAE trung bình cho mỗi đặc trưng.
 - Tiếp theo, danh sách `list_mae` được sắp xếp theo giá trị MAE tăng dần và đặc trưng có MAE thấp nhất được xác định là đặc trưng tốt nhất.
 - Mô hình Linear Regression mới sẽ được huấn luyện chỉ trên đặc trưng tốt nhất với toàn bộ tập dữ liệu huấn luyện.
 - Các thông số của mô hình (trọng số) được lấy ra và in ra màn hình.
 - Cuối cùng, mô hình với đặc trưng tốt nhất được sử dụng để dự đoán trên tập kiểm tra và tính giá trị MAE giữa dự đoán và giá trị thực tế.
- **Phân tích:**
 - Với $k < 1000$
 - Kết quả cuối cùng cho ra đặc trưng tốt nhất là Quant.
 - Em có thử nghiệm với k trong khoảng 7 đến 100, nhưng giá trị gần như không thay đổi và thời gian chạy cũng không có quá nhiều sự chênh lệch.

The skill best feature model is: Quant

Mô hình với 1 đặc trưng		MAE
1	English	121925.884315
2	Logical	120274.777737
3	Quant	118124.524456

K=5

The skill best feature model is: Quant

Mô hình với 1 đặc trưng		MAE
1	English	121968.055178
2	Logical	120373.889636
3	Quant	118033.764567

K=500

- Với những giá trị $k \geq 1000$

- Nhưng khi thử với $k=1000$, thời gian chạy rõ ràng đã lâu hơn nhiều, MAE có sự thay đổi, nhưng đặc trưng tốt nhất vẫn là neuroticism.

The skill best feature model is: Quant

Mô hình với 1 đặc trưng		MAE
1	English	121818.136128
2	Logical	120147.504570
3	Quant	118096.000107

K=1000

- **Kết luận:**

- Ở câu c này, công việc là thực hiện là Xây dựng mô hình sử dụng duy nhất 1 đặc trưng English, Logical, Quant, tìm mô hình cho kết quả tốt nhất. Sau cùng, chúng ta sẽ có được công thức sau:

The skill best feature model is: Quant

Mô hình với 1 đặc trưng		MAE
1	English	121968.055178
2	Logical	120373.889636
3	Quant	118033.764567

Best Feature Model Parameters: 0 585.895381
dtype: float64

MAE: 106819.5776198967

Công thức hồi quy (phần trọng số làm tròn đến 3 chữ số thập phân, ví dụ $0.012345 \rightarrow 0.012$)

$$\text{Salary} = 585.895 \times (\text{Quant})$$

- **Nhận xét:**

- Kết quả output cho biết rằng đặc trưng "Quant" là đặc trưng tốt nhất dựa trên quá trình cross-validation với giá trị MAE thấp nhất.
- Danh sách MAE cho từng đặc trưng được hiển thị trong DataFrame, cho thấy rằng đặc trưng "Quant" có MAE thấp nhất trong danh sách.
- Công thức hồi quy cuối cùng được xác định là: $\text{Salary} = 585.895 \times \text{Quant}$.
- Điều này chỉ ra rằng mô hình hồi quy tốt nhất cho dự đoán lương dựa trên đặc trưng "Quant" sẽ sử dụng hệ số hồi quy là 585.895.
- Giá trị MAE tương đối thấp (106819.577) khi sử dụng mô hình với đặc trưng tốt nhất trên tập kiểm tra, điều này có thể cho thấy mô hình có khả năng dự đoán tương đối tốt trên dữ liệu mới.
- Công thức hồi quy cho thấy mối quan hệ tuyến tính giữa đặc trưng Quant và lương. Công thức này có thể được sử dụng để dự đoán lương dựa trên giá trị của đặc trưng Quant

[4] Câu 1d

- **Ý tưởng:**

- Từ phần gợi ý xây dựng mô hình trong file [project03.ipynp](#) do cô Phan Thị Phương Uyên cung cấp như sau:

- Gợi ý xây dựng mô hình:

- Trực quan hóa các biến và đánh giá tính phân phối, tương quan giữa các biến, và xác định các đặc điểm đáng chú ý của dữ liệu
- Phân tích mối quan hệ giữa biến mục tiêu và các biến dự đoán bằng các biểu đồ phân tán, ma trận tương quan, và biểu đồ histogram
- → lựa chọn đặc trưng phù hợp cho mô hình mới

- Cho nên phương pháp mà em lựa chọn để tìm ra mô hình tốt nhất là dựa vào tính toán ra tương quan giữa các biến [4], [6]. Em xin trích một số đoạn sau từ tài liệu tham khảo.

“Tương quan là phân tích thống kê về mối quan hệ hoặc sự phụ thuộc giữa hai biến. Mối tương quan cho phép chúng ta nghiên cứu cả cường độ và chiều hướng của mối quan hệ giữa hai tập hợp biến.” - (Luna, 2022)

“Dưới đây là một số thông tin quan trọng về hệ số tương quan Pearson:

Hệ số tương quan Pearson có thể nhận bất kỳ giá trị thực nào trong phạm vi $-1 \leq r \leq 1$.

Giá trị tối đa $r = 1$ tương ứng với trường hợp có mối quan hệ tuyến tính dương hoàn hảo giữa x và y . Nói cách khác, giá trị x lớn hơn tương ứng với giá trị y lớn hơn và ngược lại.

Giá trị $r > 0$ biểu thị mối tương quan dương giữa x và y .

Giá trị $r = 0$ tương ứng với trường hợp không có mối quan hệ tuyến tính giữa x và y .

Giá trị $r < 0$ biểu thị mối tương quan nghịch giữa x và y .

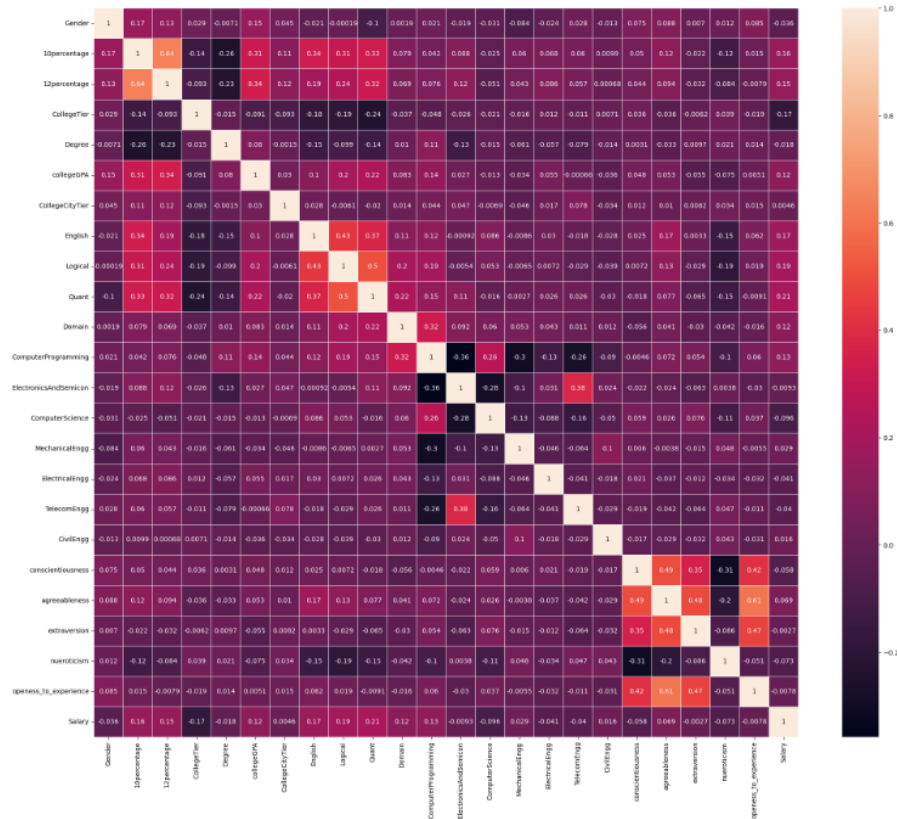
Giá trị tối thiểu $r = -1$ tương ứng với trường hợp có mối quan hệ tuyến tính âm hoàn hảo giữa x và y . Nói cách khác, giá trị x lớn hơn tương ứng với giá trị y nhỏ hơn và ngược lại.

Thực tế trên có thể tóm tắt trong bảng sau:

Giá trị r của Pearson	Tương quan giữa x và y
Bằng 1	Mối quan hệ tuyến tính dương hoàn hảo
Lớn hơn 0	Tương quan tích cực
Bằng 0	Không có mối quan hệ tuyến tính
Tương quan âm nhỏ hơn 0	Mối quan hệ tuyến tính phủ định hoàn toàn

Nói tóm lại, giá trị tuyệt đối lớn hơn của r cho thấy mối tương quan mạnh hơn, gần với hàm tuyến tính hơn. Giá trị tuyệt đối nhỏ hơn của r cho thấy mối tương quan yếu hơn.” - (Stojiljković, 2019)

- Từ những định nghĩa được trích bên trên, em tìm các giá trị tương quan của các đặc trưng thông qua `.corr()`, một phương thức trong thư viện pandas của Python, được sử dụng để tính toán ma trận tương quan giữa các cột (biến) trong một DataFrame, sau đó tạo ma trận tương quan giữa các biến trong tập dữ liệu và tạo một biểu đồ heatmap để trực quan hóa tương quan giữa các biến.



- Nhưng điều mà ta cần quan tâm ở đây là các giá trị tương quan của các đặc trưng với đặc trưng “Salary”, thứ giúp lựa chọn ra những đặc trưng có ảnh hưởng cao đến mục tiêu. Do đó em đã trích ra những giá trị tương quan đó để dễ xem xét.

Gender	-0.036183
10percentage	0.155174
12percentage	0.149531
CollegeTier	-0.174824
Degree	-0.017602
collegeGPA	0.122469
CollegeCityTier	0.004575
English	0.169293
Logical	0.188416
Quant	0.205358
Domain	0.122022
ComputerProgramming	0.125866
ElectronicsAndSemicon	-0.009292
ComputerScience	-0.095507
MechanicalEngg	0.028854
ElectricalEngg	-0.041217
TelecomEngg	-0.040415
CivilEngg	0.016150
conscientiousness	-0.057699
agreeableness	0.068623
extraversion	-0.002661
neroticism	-0.073401
openess_to_experience	-0.007814

Name: Salary, dtype: float64

- Từ những giá trị tương quan trên, ta nhận thấy giá trị lớn nhất là 0.205 là giá trị tương quan giữa Quant và Salary. Thực tế xem xét thì các giá trị tương quan trên đều không quá cao và không quá chênh lệch nên em chia thành 3 ngưỡng giá trị để tạo dựng mô hình, đó là chọn các biến có **tương quan tuyệt đối** lớn hơn hoặc bằng một ngưỡng nào đó (**0.1, 0, 0.05**) với Salary.
- Các biến này sẽ được sử dụng để xây dựng mô hình là:
 - Các đặc trưng được chọn với giá trị tương quan tuyệt đối ≥ 0.1 là 10percentage, 12percentage, CollegeTier, collegeGPA, English, Logical, Quant, Domain, ComputerProgramming.
 - Các đặc trưng được chọn với giá trị tương quan tuyệt đối ≥ 0 là 23 đặc trưng còn lại
 - Các đặc trưng được chọn với giá trị tương quan tuyệt đối ≥ 0.05 là 10percentage, 12percentage, CollegeTier, collegeGPA, English, Logical, Quant, Domain, ComputerProgramming, ComputerScience, conscientiousness, agreeableness, neroticism
- Từ ba mô hình trên và gợi ý từ file project03, ngoài việc thử trên mô hình gốc thì em thử biến đổi các đặc trưng theo hai cách đó là bình

phương và căn bậc ba (ngoài ra là do có cả số âm và số dương khi xem xét nên em đã chọn hai cách biến đổi này)

- Xây dựng **m** mô hình khác nhau (tối thiểu 3), đồng thời khác mô hình ở 1a, 1b và 1c
 - Mô hình có thể là sự kết hợp của 2 hoặc nhiều đặc trưng
 - Mô hình có thể sử dụng đặc trưng đã được chuẩn hóa hoặc biến đổi (bình phương, lập phương...)
 - Mô hình có thể sử dụng đặc trưng được tạo ra từ 2 hoặc nhiều đặc trưng khác nhau (cộng 2 đặc trưng, nhân 2 đặc trưng...)
 - ...

- Từ đó em có được 9 mô hình sau với các MAE trung bình tương ứng

	Mô hình	MAE
0	Mô hình gốc 1	112840.317733
1	Mô hình mũ hai 1	306899.059639
2	Mô hình căn ba 1	306899.059639
3	Mô hình gốc 2	110420.413776
4	Mô hình mũ hai 2	112537.827936
5	Mô hình căn ba 2	111126.449920
6	Mô hình gốc 3	110689.047691
7	Mô hình mũ hai 3	112537.827936
8	Mô hình căn ba 3	111126.449920

- **Nhận xét:**

- hình gốc (không thay đổi): Có MAE thấp nhất trong ba mô hình với giá trị trung bình khoảng 110,983.
- Mô hình mũ hai (bình phương): Cả hai phiên bản mô hình này (1 và 2) có MAE rất cao, khoảng 306718, cho thấy việc áp dụng bình phương lên dữ liệu không đem lại kết quả tốt.
- Mô hình căn bậc ba: Cả hai phiên bản mô hình này (1 và 2) cũng có MAE tương đối cao, khoảng 111126, tương tự như mô hình gốc.
- Dựa trên kết quả cross-validation, mô hình gốc (không biến đổi) có hiệu suất tốt hơn so với các mô hình khác.
- Cả hai mô hình mũ hai và mô hình căn bậc ba không mang lại sự cải thiện về độ chính xác so với mô hình gốc.
- **Mô hình gốc 2 là mô hình tốt nhất** với MAE là 110420.413776, đây là mô hình được tính toán với tất cả các đặc trưng (ngoài Salary). Vì vậy, mô hình gốc 2 được chọn làm mô hình tốt nhất để tiếp tục thực hiện các bước chuẩn hóa hoặc xây dựng.

- **Huấn luyện lại mô hình tốt nhất trên toàn bộ tập huấn luyện**

- Sau khi xác định được mô hình gốc 2 là mô hình tốt nhất, em tiến hành huấn luyện lại mô hình hồi quy tuyến tính `my_best_model` trên tập dữ liệu huấn luyện và trả về các trọng số tương ứng của mô hình.

```
0    -23874.541727
1      898.575621
2    1203.496112
3   -83592.387591
4    11515.430757
5    1626.518605
6   -5717.733852
7     153.434567
8     120.511333
9     102.580853
10   27939.639602
11     76.730246
12    -47.746793
13   -177.387649
14     33.932559
15   -151.471153
16    -64.197706
17    145.894996
18  -19814.830268
19   15503.266941
20   4908.582006
21  -10661.029100
22  -5815.021280
dtype: float64
```

- **MAE và công thức hồi quy sau cùng là:**

MAE: 101872.2105661925

Công thức hồi quy (phần trọng số làm tròn đến 3 chữ số thập phân, ví dụ 0.012345 → 0.012)

$$\begin{aligned} \text{Salary} = & -23874.542 \times \text{Gender} + 898.576 \times 10\text{percentage} + 1203.496 \times 12\text{percentage} - 83592.387 \times \text{CollegeTier} + 11515.430 \times \text{Degree} \\ & + 1626.519 \times \text{collegeGPA} - 5717.734 \times \text{CollegeCityTier} + 153.435 \times \text{English} + 120.511 \times \text{Logical} + 102.581 \times \text{Quant} + 27939.639 \times \text{Domain} \\ & + 76.730 \times \text{ComputerProgramming} - 47.747 \times \text{ElectronicsAndSemicon} - 177.388 \times \text{ComputerScience} + 33.932 \times \text{MechanicalEngg} - 151.471 \\ & \times \text{ElectricalEngg} - 64.197 \times \text{TelecomEngg} + 145.895 \times \text{CivilEngg} - 19814.830 \times \text{conscientiousness} + 15503.267 \times \text{agreeableness} + 4908.582 \\ & \times \text{extraversion} + 10661.029 \times \text{nueroticism} - 5815.021 \times \text{openess_to_experience} \end{aligned}$$

- **Cảm nghĩ cá nhân:**

- Vốn ban đầu em có xem một tài liệu về những yếu tố có thể ảnh hưởng đến lương của kỹ sư [3]. Trong đó yếu tố được cho là ảnh hưởng là Gender, CollageGPA, Degree. Nhưng khi thử áp dụng vào tính toán thử thì kết quả cho ra lại khá cao. Do đó em mới hướng đến tính tương quan giữa các đặc trưng còn lại với Salary, nhờ đó mà cuối cùng mới đạt được kết quả ngoài mong đợi.
- Và đồ án này hoàn thành song song với việc chuẩn bị cho kỳ thi cuối kỳ nên thời gian chuẩn bị khá gấp, và ban đầu em không có tạo hàm riêng (hàm `perform_cross_validation`) mà hoàn thành các câu trước, sau đó

mới tính đến việc tách hàm. Tuy nhiên sau khi hoàn thành các câu và thực hiện tách hàm thì phải chỉnh sửa lại khá nhiều, nên sau khi suy nghĩ em vẫn giữ nguyên đoạn code ban đầu. Vì điều này nên bài làm khá dài, em nhận thấy đây cũng là một điều thiếu sót ạ.


6. Nhận xét chung với tất cả các mô hình

- Ở câu a, việc sử dụng 11 đặc trưng cho ra kết quả MAE sau cùng khá tốt, nó là kết quả MAE nhỏ thứ 2, chỉ sau mô hình ở câu d do chính em xây dựng.
- Ở câu b và c, dù là xây dựng mô hình chỉ với việc sử dụng một đặc trưng tốt nhất nhưng kết quả cho ra của cả hai lại là khá khác biệt, một cái thì cho MAE rất lớn và cho thấy đó là mô hình không tốt với các đặc trưng là *conscientiousness*, *agreeableness*, *extraversion*, *neuroticism*, *openness_to_experience*; một cái lại cho ra kết quả MAE nhỏ ngoài mức mong đợi với các đặc trưng là *English*, *Logic* và *Quant*.
- Ngoài ra, ở câu d, khi xem xét tương quan giữa các đặc trưng khác với *Salary*, *Quant* cũng cho ra một giá trị tương quan lớn nhất, cho thấy rõ đây là một đặc trưng rất có ảnh hưởng đến *Salary*.
- Sau cùng là ở câu d, mô hình khi chuẩn hóa hoặc biến đổi các đặc trưng ở đây lại không cho thấy sự vượt trội hơn so với mô hình có các đặc trưng không chuẩn hóa.

7. Tài liệu tham khảo

- [1] *Sklearn.model_selection.KFold. (n.d.). Scikit-learn.*
Retrieved August 16, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html
- [2] *pandas.DataFrame.to_numpy — pandas 2.0.3 documentation. (n.d.). Retrieved August 17, 2023, from*
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_numpy.html
- [3] *Engineering salaries on the rise. (n.d.). Retrieved August 18, 2023, from* <https://www.asme.org/topics-resources/content/engineering-salaries-on-the-rise>

[4] Luna, J. C. (2022b, February 25). Python details on correlation tutorial. Retrieved August 20, 2023, from https://www.datacamp.com/tutorial/tutorial-datails-on-correlation?utm_source=google&utm_medium=paid_search&utm_campaignid=19589720824&utm_adgroupid=143216588537&utm_device=c&utm_keyword=&utm_matchtype=&utm_network=g&utm_adpostion=&utm_creative=661628555495&utm_targetid=aud-299261629574:dsa-1947282172981&utm_loc_interest_ms=&utm_loc_physical_ms=1028581&utm_content=dsa~page~community-tuto&utm_campaign=230119_1-sea~dsa~tutorials_2-b2c_3-row-p2_4-prc_5-na_6-na_7-le_8-pdsh-go_9-na_10-na_11-na&gclid=Cj0KCQjw3JanBhCPARIsAJpXTx7gpN77gRl7DPp0HfwCBSyTxYqQ_-1b84H6EUQZnmZMCdv-3vBXIOEaAmNpEALw_wcB

- [5] Sujithmandala. (2023, February 19).  Salary prediction of Engineering students. Kaggle. Retrieved August 19, 2023, from <https://www.kaggle.com/code/sujithmandala/salary-prediction-of-engineering-students>
- [6] Python, R. (2023, June 2). NumPy, SciPy, and pandas: Correlation With Python. *realpython.com*. Retrieved August 20, 2023, from <https://realpython.com/numpy-scipy-pandas-correlation-python/>
- [7] Project03.ipynb và lab04 do cô Phan Thị Phương Uyên cung cấp