TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN - LINEAR REGRESSION

TOÁN ỨNG DỤNG VÀ THỐNG KÊ

HỌ VÀ TÊN: BÙI ĐỖ DUY QUÂN

MÃ SỐ SINH VIÊN: 21127141

LỚP: 21CLC02

Giảng viên hướng dẫn:
Phan Thị Phương Uyên

Ngày 24 tháng 8 năm 2023

Mục lục

1 YÊU CẦU CỦA ĐỒ ÁN

- Xây dựng mô hình dự đoán **mức lương** của kỹ sư sử dụng **mô hình hồi quy tuyến tính (linear regression)** với các yêu cầu sau:
  1. Sử dụng 11 đặc trưng gồm: **Gender, 10percentage, 12percentage, CollegeTier, Degree, collegeGPA, CollegeCityTier, English, Logical, Quant, Domain**
  2. Sử dụng 5 đặc trưng tính cách: **conscientiousness, agreeableness, extraversion, nueroticism, openess_to_experience** và sử dụng phương pháp **k-fold cross validation** tìm ra đặc trưng tốt nhất trong các đặc trưng tính cách.
  3. Sử dụng 3 đặc trưng kĩ năng: **English, Logical, Quant** và sử dụng phương pháp **k-fold cross validation** tìm ra đặc trưng tốt nhất.
  4. Sinh viên tự xây dựng các mô hình (tối thiểu 3) và tìm mô hình cho kết quả tốt nhất qua phương pháp **k-fold cross validation**.
- Các thư viện được cho trước: **Numpy**, **pandas**.

2 BỘ DỮ LIỆU

- Bộ dữ liệu **Engineering Graduate Salary** gồm 2998 dòng và 34 cột. Sau quá trình tiền xử lý là loại bỏ các cột có **giá trị chuỗi** và **giá trị liên quan đến định danh và năm** thì còn lại 2998 dòng và 24 cột như sau:
  - Giá trị mục tiêu (y): **Salary**
  - 23 đặc trưng giải thích (X) gồm: Gender, 10percentage, 12percentage, CollegeTier, Degree, collegeGPA, CollegeCityTier, English, Logical, Quant, Domain, ComputerProgramming, ElectronicsAndSemicon, ComputerScience, MechanicalEngg, ElectricalEngg, TelecomEngg, CivilEngg, conscientiousness, agreeableness, extraversion, nueroticism, openess_to_experience.
- Sinh viên đã được cung cấp 2 bộ dữ liệu: **train.csv** và **test.csv**. Bộ dữ liệu **train.csv** gồm 2248 mẫu để huấn luyện mô hình , và bộ dữ liệu **test.csv** gồm 750 mẫu để kiểm tra mô hình.

3 CÁC THƯ VIỆN SỬ DỤNG THÊM

Ngoài việc sử dụng 2 thư viện được cung cấp là **Numpy** và **pandas**, sinh viên còn sử dụng thêm thư viện sau:

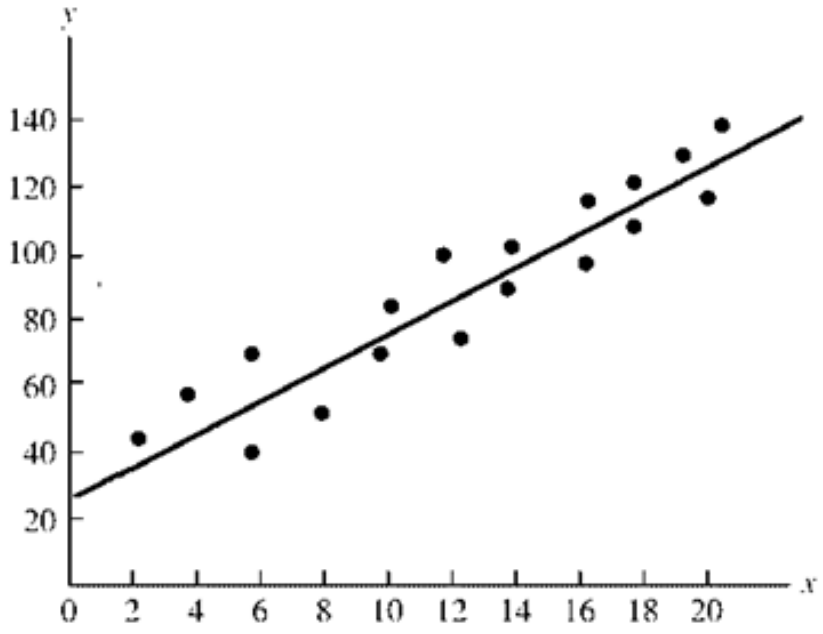
- **sklearn** và sử dụng module **feature_selection** của thư viện này. Trong module này sẽ sử dụng 3 lớp:
  - **VarianceThreshold**: Loại bỏ các đặc trưng có **phương sai** nhỏ hơn ngưỡng được đặt trước.
  - **mutual_info_regression**: Tính độ tương quan giữa các đặc trưng và giá trị mục tiêu.
  - **SelectPercentile**: Chọn ra nhóm các đặc trưng có độ tương quan cao nhất với giá trị mục tiêu.
- **matplotlib**: Thư viện hỗ trợ trực quan thông tin biểu đồ.
- **seaborn**: Thư viện hỗ trợ xây dựng biểu đồ **heatmap**.

4 KIẾN THỨC TÌM HIỂU

4.1 Hồi quy tuyến tính

- **Hồi quy tuyến tính (linear regression)** là phương pháp phân tích quan hệ giữa biến phụ thuộc Y với một hay nhiều biến độc lập X. Phương pháp này sử dụng **hàm tuyến tính (bậc 1)**, và các tham số của mô hình được ước lượng từ dữ liệu. Việc xây dựng **mô hình hồi quy tuyến tính** có thể giúp dự đoán một cách chính xác nhất. Mô hình hồi quy tuyến tính cho mẫu dữ liệu như sau:

$$y = \theta_0 + \theta_1 x \tag{1}$$



Đồ thị minh họa

- Như vậy đối với những dữ liệu có nhiều thuộc tính thì có thể mở rộng mô hình hồi quy tuyến tính như sau:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n \tag{2}$$

4.2 Huấn luyện và kiểm tra mô hình

- Để có thể tìm được mô hình phù hợp nhất cho bộ dữ liệu, bộ dữ liệu phải được chia thành 2 phần là **tập huấn luyện** và **tập kiểm tra**. Điều này sẽ giúp mô hình tránh được trường hợp **underfitting** và **overfitting**. Hai trường hợp này lần lượt là những mô hình quá tệ về độ chính xác của dữ liệu dự đoán hay mô hình quá phức tạp, cho kết quả rất tốt trên dữ liệu được cho nhưng lại quá kém so với dữ liệu khác ở thực tế. Tập huấn luyện **trainning set** sẽ được sử dụng để huấn luyện mô hình, còn tập kiểm tra **testing set** sẽ được sử dụng để kiểm tra mô hình đã được huấn luyện có tốt hay không. Tập kiểm tra sẽ được sử dụng để đánh giá mô hình.
- Trong thực tế đã nhiều phương pháp được dùng để tạo ra tập **huấn luyện và kiểm tra**, trong đồ án này sẽ đề cập tới phương pháp **K-fold Cross Validation**.

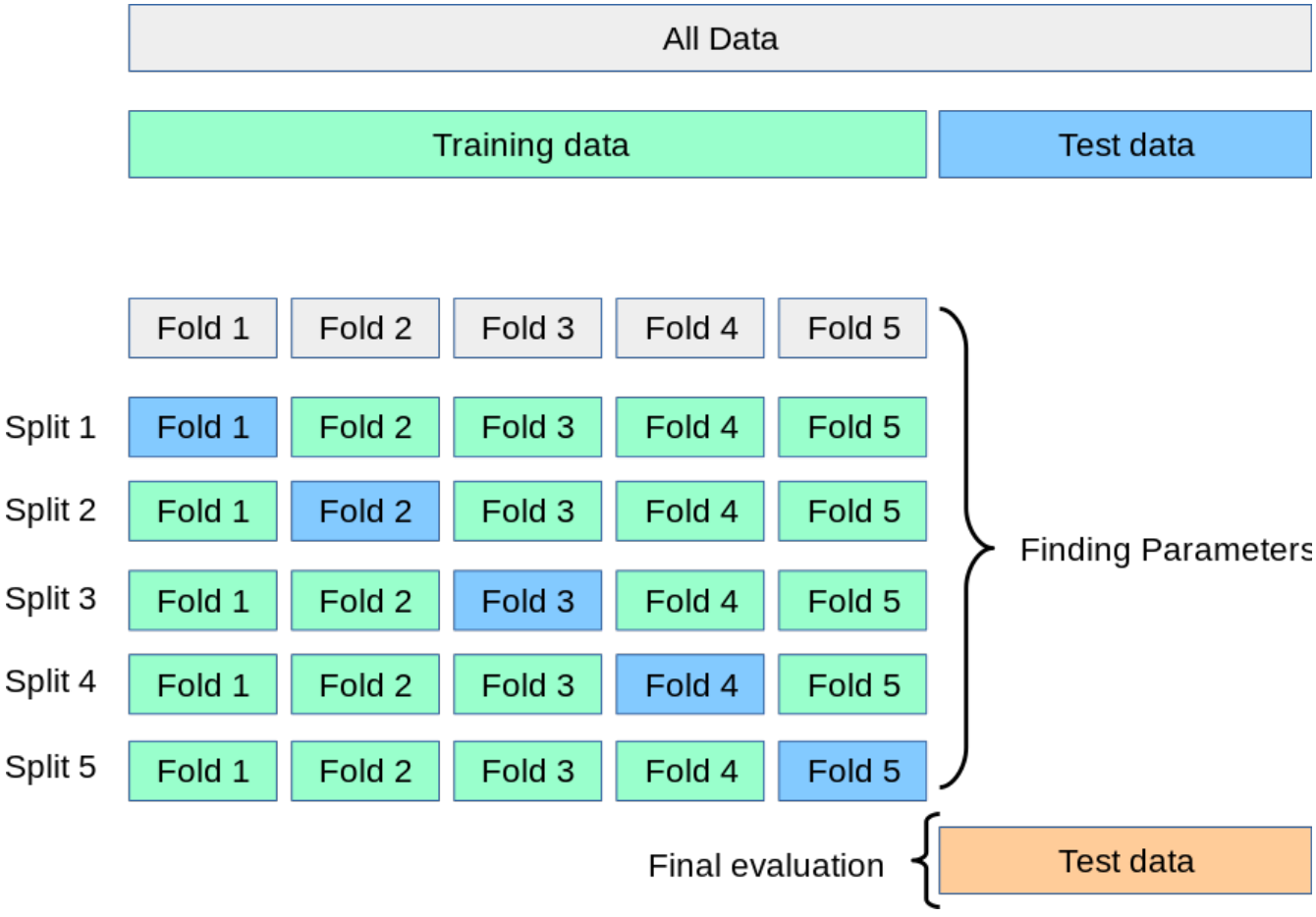
K-fold Cross Validation

  – Theo như thông thường, chúng ta sẽ nghĩ tới việc chọn bao nhiêu phần để cho làm phần tập **huấn luyện**, và tập còn lại sẽ là tập **kiểm tra**. Tuy

nhiên chúng ta sẽ không biết được 2 phần này nên chứa bao nhiêu dữ liệu trong từng trường hợp và nếu chia sai thì độ chính xác của mô hình sẽ chắc chắn không tốt. Chính vì vậy ý tưởng của phương pháp này chính là sẽ chia đều bộ dữ liệu này thành từng phần **fold** và sẽ thực hiện huấn luyện và kiểm tra từng phần để tìm ra mô hình tốt nhất.

– Chi tiết các bước làm của phương pháp như sau:

1. Chia bộ dữ liệu thành **k** phần bằng nhau.
2. Tại thời điểm xét từng phần dữ liệu, phần dữ liệu đó sẽ được chọn làm tập **kiểm tra**, và **k-1** còn lại sẽ được chọn làm tập **huấn luyện**.
3. Lưu lại độ chính xác của mô hình tại thời điểm đó.
4. Lặp lại các bước trên cho đến khi tất cả các phần dữ liệu đều được chọn làm tập **kiểm tra**.
5. Tính trung bình độ chính xác của các lần huấn luyện và kiểm tra và đây chính là độ chính xác của mô hình.



Hình 1: Minh họa phương pháp K-fold Cross Validation

5 MÔ TẢ CÁC HÀM SỬ DỤNG

5.1 Lớp OLSLinearRegression

Đây là lớp được cô **Phan Thị Phương Uyên** cung cấp, lớp này sẽ giúp sinh viên có thể xây dựng được mô hình hồi quy tuyến tính. Các thuộc tính của lớp này gồm:

- **fit(self, X, y)**: phương thức này sẽ thực hiện huấn luyện mô hình hồi quy tuyến tính với dữ liệu đầu vào là **X (dữ liệu đặc trưng)** và **y (dữ liệu mục tiêu)**. Hàm sẽ thực hiện và trả về trọng số của mô hình tương ứng với các đặc trưng theo công thức:

$$\theta = (X^T X)^{-1} X^T y \tag{3}$$

- **get_params()**: phương thức *getter* sẽ trả về các trọng số của mô hình.
- **predict(self, X)**: phương thức này sẽ thực hiện dự đoán giá trị mục tiêu dựa trên dữ liệu đầu vào là **X (dữ liệu đặc trưng)** và trọng số của mô hình. Hàm sẽ thực hiện và trả về giá trị dự đoán theo công thức:

$$y = \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n \tag{4}$$

5.2 Lớp VarianceThreshold

Đây là lớp được cung cấp bởi thư viện **sklearn.feature_selection**, lớp này sẽ giúp có thể loại bỏ các đặc trưng có **phương sai** nhỏ hơn ngưỡng được đặt trước, cụ thể là:

- **VarianceThreshold(threshold)**: phương thức khởi tạo lớp, **tham số truyền vào** sẽ là ngưỡng để loại bỏ các đặc trưng có phương sai nhỏ hơn ngưỡng này.
- **fit(X_train)**: phương thức sẽ thực hiện tính giá trị phương sai của các đặc trưng trong **X_train**. Hàm loại bỏ các đặc trưng có phương sai nhỏ hơn ngưỡng **threshold** được đặt trước.Phương thức sẽ **không trả về giá trị nào** nhưng sẽ loại bỏ các đặc trưng có phương sai nhỏ hỡn ngưỡng của bộ dữ liệu đã được truyền vào.

5.3 Lớp mutual_info_regression

Đây là lớp được cung cấp bởi thư viện **sklearn.feature_selection** được dùng để tính toán độ tương quan giữa các đặc trưng và mục tiêu, trong đó:

- **mutual_info_regression(X, y)**: phương thức khởi tạo lớp, **tham số truyền vào** sẽ là **ma trận các đặc trưng** và **vector mục tiêu**.Giá trị **trả về** của

phương thức là mảng chứa các giá trị **mutual information** tương ứng với các đặc trưng.

5.4 Lớp SelectPercentile

Đây là lớp được cung cấp bởi thư viện **sklearn.feature_selection**, hỗ trợ trong việc chọn ra nhóm các đặc trưng có độ tương quan cao nhất với mục tiêu, trong đó:

- **SelectPercentile(score_function, percentile)**: phương thức khởi tạo lớp, **tham số truyền vào** gồm **hàm tính toán**, trong đây sẽ là hàm dùng để tính toán giá trị liên quan giữa 2 đặc trưng tùy thuộc vào loại tương quan nào của tham số truyền vào, và **phần trăm** của các đặc trưng có độ tương quan cao nhất với mục tiêu. Giá trị **trả về** của phương thức là mảng chứa các đặc trưng có độ tương quan cao nhất với mục tiêu.
- **fit(X_train, y_train)**: phương thức sẽ thực hiện tính toán giá trị liên quan giữa các đặc trưng với mục tiêu dựa trên hàm tính toán được truyền vào và chọn ra các đặc trưng có độ tương quan cao nhất với mục tiêu dựa trên phần trăm được truyền vào. Phương thức sẽ **không trả về giá trị nào** nhưng sẽ chọn ra các đặc trưng có độ tương quan cao nhất với mục tiêu của bộ dữ liệu đã được truyền vào.

5.5 Hàm mae(y_pred, y_test)

Đây là hàm được cô **Phan Thị Phương Uyên** cung cấp, tham số truyền vào sẽ là **giá trị/tập giá trị dự đoán và giá trị/tập giá trị kiểm tra**. Hàm sẽ **trả về giá trị thể hiện độ chính xác** của giá trị được dự đoán, giá trị càng thấp thì độ chính xác càng cao. Công thức để tính độ chính xác là:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_{pred} - y_{test}| \tag{5}$$

5.6 Hàm getTrain(index, folks)

Đây là hàm hỗ trợ cho việc xác định tập dữ liệu **huấn luyện** trong phương pháp K-fold Cross Validation. Giá trị đầu vào sẽ là **index** của phần dữ liệu đang xét và **folks** là tập dữ liệu đã được chia. Hàm sẽ **trả về tập dữ liệu huấn luyện**.

5.7 Hàm Best_Feature_Personality(Df_np, k_cluster)

Đây là hàm giúp tìm ra đặc trưng tính cách tốt nhất trong 5 tính cách ở yêu cầu **2** trong phần YÊU CẦU ĐỒ ÁN. Giá trị đầu vào sẽ là **Df_np** là tập dữ liệu đã

được chuyển thành dạng **numpy array** và **k_cluster** là số lượng cụm cần phải chia. Hàm sẽ thực hiện huấn luyện từng đặc trưng tính cách trên từng phần và sau cùng **trả về số thứ tự của đặc trưng tính cách tốt nhất**.Việc chọn ra đặc trưng tốt nhất sẽ được thực hiện dựa trên so sánh **MAE trung bình** của từng đặc trưng sau khi huấn luyện trên từng **k_cluster** phần dữ liệu.

5.8 Hàm best_personality_feature_model()

Đây là hàm sẽ thực hiện huấn luyện mô hình theo đặc trưng tốt nhất mà được tìm thấy ở hàm **Best_Feature_Personality**. Tham số truyền vào sẽ không có, nhưng giá trị trả về của hàm lần lượt 2 giá trị/tập giá trị **dự đoán** và **kiểm tra**.

5.9 Hàm Best_Feature_Skill(Df_np, k_cluster)

Đây là hàm giúp tìm ra đặc trưng kĩ năng tốt nhất trong 3 kĩ năng ở yêu cầu **3** trong phần YÊU CẦU ĐỒ ÁN. Giá trị đầu vào sẽ là **Df_np** là tập dữ liệu đã được chuyển thành dạng **numpy array** và **k_cluster** là số lượng cụm cần phải chia. Hàm sẽ thực hiện huấn luyện từng đặc trưng kĩ năng trên từng phần và sau cùng **trả về số thứ tự của đặc trưng kĩ nang9 tốt nhất**.Việc chọn ra đặc trưng tốt nhất sẽ được thực hiện dựa trên so sánh **MAE trung bình** của từng đặc trưng sau khi huấn luyện trên từng **k_cluster** phần dữ liệu.

5.10 Hàm best_skill_feature_model()

Đây là hàm sẽ thực hiện huấn luyện mô hình theo đặc trưng tốt nhất mà được tìm thấy ở hàm **Best_Feature_Skill**. Tham số truyền vào sẽ không có, nhưng giá trị trả về của hàm lần lượt 2 giá trị/tập giá trị **dự đoán** và **kiểm tra**.

5.11 Hàm cross_validation_model(Df_np, k_cluster)

Đây là hàm sẽ thực huấn luyện để tìm ra độ chính xác của mô hình nhờ phương pháp **K-fold Cross Validation**. Về ý tưởng, cách thức cài đặt sẽ khá giống với hàm **Best_Feature_Personality** và **Best_Feature_Skill**, nhưng hàm này sẽ thực hiện huấn luyện trên tất cả các đặc trưng trong mảng **Df_np** truyền đầu vào cùng với số nhóm muốn dữ liệu được chia và **trả về độ chính xác của mô hình**.

5.12 Hàm model_variance_Dropping(X_train)

Đây là hàm sẽ hỗ trợ cho phương pháp xây dưng mô hình trong yêu cầu **3** trong phần YÊU CẦU ĐỒ ÁN, với tên gọi: Dropping Constant Feature, dựa trên giá

trị **phương sai** của các đặc trưng. Tham số truyền vào sẽ là **X_train** là tập dữ liệu huấn luyện của các đặc trưng không phụ thuộc. Hàm sẽ thực hiện **tìm ra các đặc trưng có phương sai lớn hơn ngưỡng** và **trả về tên các đặc trưng đó**.

5.13 Hàm model_correlation_Dropping(X_train, threshold)

Đây là hàm sẽ hỗ trợ cho phương pháp xây dưng mô hình trong yêu cầu **4** trong phần YÊU CẦU ĐỒ ÁN,với tên gọi: Dropping high-correlation Feature, dựa trên giá trị **phương sai** của các đặc trưng. Tham số truyền vào sẽ là **X_train** là tập dữ liệu huấn luyện của các đặc trưng không phụ thuộc và mức độ tương quan mong muốn giữa các đặc trưng mong. Hàm sẽ thực hiện **chọn ra các đặc trưng có mức độ tương quan lớn hơn threshold** và **trả về tên các đặc trưng đó**.

5.14 Hàm model_MutualInfor_selection(X_train, Y_train)

Đây là hàm sẽ hỗ trợ cho phương pháp xây dưng mô hình trong yêu cầu **4** trong phần YÊU CẦU ĐỒ ÁN,với tên gọi: Selecting high-MI Feature, dựa trên giá trị **mutual information** của các đặc trưng. Tham số truyền vào sẽ là **X_train** là tập dữ liệu huấn luyện của các đặc trưng không phụ thuộc và **Y_train** là tập dữ liệu của đặc trưng phụ thuộc. Hàm sẽ thực hiện **chọn ra các đặc trưng có mức độ liên quan tới đặc trưng phụ thuộc cao** và **trả về tên các đặc trưng đó**.

5.15 Hàm my_best_model(models)

Đây là hàm sẽ tìm model nào tốt nhất từ mảng các **models** truyền vào qua phương pháp **K-fold Cross Validation**. Hàm sẽ **trả về số thứ tự của model tốt nhất trong mảng**.

5.16 Hàm train_my_best_model()

Đây là hàm huấn luyện model tốt nhất mà đã tìm được. **Đầu vào** của hàm sẽ không có tham số, nhưng hàm sẽ **trả về giá trị/tập giá trị của dữ liệu phụ thuộc được dự đoán và kiểm tra**.

6 KẾT QUẢ CỦA CÁC YÊU CẦU

6.1 Yêu cầu 1

a Các bước thực hiện

1. Truy xuất dữ liệu của 11 cột đặc trưng được yêu cầu (11 cột đầu tiên) trong tập huấn luyện và tập kiểm tra.

2. Truy xuất cột Salary - là cột mục tiêu trong tập huấn luyện và tập kiểm tra.
3. Tìm trọng số của mô hình hồi quy tuyến tính bằng cách sử dụng phương thức **fit** của lớp **OLSLinearRegression** với dữ liệu đầu vào là tập huấn luyện của 11 đặc trưng yêu cầu và đặc trưng mục tiêu Salary.
4. Sử dụng phương thức **predict** của lớp **OLSLinearRegression** để dự đoán giá trị mục tiêu của tập kiểm tra.
5. Sử dụng hàm **mae** để tính độ chính xác của mô hình.

b Công thức cho mô hình hồi quy (trọng số làm tròn tới 3 chữ số thập phân) và kết quả trên tập kiểm tra

- $$\begin{aligned}\text{Salary} = &(-22756.513) \cdot X_1 + 804.503 \cdot X_2 + 1294.655 \cdot X_3 \\ &+ (-91781.898) \cdot X_4 + 23182.389 \cdot X_5 + 1437.549 \cdot X_6 \\ &+ (-8570.662) \cdot X_7 + 147.858 \cdot X_8 + 152.888 \cdot X_9 \\ &+ 117.222 \cdot X_{10} + 34552.286 \cdot X_{11}\end{aligned} \tag{6}$$

- Độ chính xác của mô hình trên tập kiểm tra là: **104863.77754033315**

6.2 Yêu cầu 2

a Các bước thực hiện

Cài đặt **best_personality_feature_model** sẽ dùng để thực hiện yêu cầu 2 như sau:

1. Khởi tạo mảng chứa tên 5 đặc trưng tính cách vì trong bộ dữ liệu, 5 đặc trưng này đứng liên tiếp nhau.
2. Thực hiện đổi chỗ ngẫu nhiên của cac dòng dữ liệu 1 lần.
3. Gọi hàm **Best_Feature_Personality** để tìm ra đặc trưng tốt nhất theo phương pháp K-fold Cross Validation.
   (a) Thực hiện chia bộ dữ liệu huấn luyện thành 10 nhóm đều nhau (sử dụng hàm **numpy.array_split**).
   (b) Thực hiện huấn luyện trên từng phần:
   - Gọi hàm **getTrain** để lấy tập dữ liệu huấn luyện và tập kiểm tra là phần dữ liệu đang xét.

- Với mỗi đặc trưng tính cách, thực hiện huấn luyện mô hình và tính độ chính xác của mô hình, lưu lại độ chính xác của mô hình trong mảng **mae_arr**.

(c) Sau khi xét hết phần dữ liệu, mảng **mae_arr** thu được gồm 10 dòng và 5 cột tương đương với 10 phần dữ liệu được chia và 5 đặc trưng của mỗi dòng.

(d) Tính độ chính xác trung bình của mỗi đặc trưng bằng cách lấy trung bình cộng của mỗi cột trong mảng **mae_arr** và trả ra giá trị thứ tự của đặc trưng có **mae** nhỏ nhất. Một điều lưu ý là giá trị **mae trung bình** của mỗi đặc trưng sẽ thay đổi mỗi khi chạy chương trình vì việc đổi chỗ các dòng dữ liệu là ngẫu nhiên và khác nhau.

4. Sau khi tìm được đặc trưng tốt nhất, tiến hình truy xuất dữ liệu của đặc trưng đó để huấn luyện mô hình bằng cách sử dụng hàm **fit** của lớp **OLSLinearRegression** và tính độ chính xác của mô hình bằng cách sử dụng hàm **mae**.

b Kết quả tương ứng cho 5 mô hình từ k-fold Cross Validation

STT	Mô hình với 1 đặc trưng	MAE
1	conscientiousness	306267.12993085
2	agreeableness	300894.66699364
3	extraversion	307070.11566216
4	neuroticism	299387.84192331
5	openness_to_experience	303008.96858233

c Kết quả tương ứng cho mô hình tốt nhất

- Mô hình tốt nhất là mô hình đặc trưng **neuroticism** với độ chính xác trung bình là: **299387.84192331**
- Công thức hồi quy tuyến tính của mô hình tốt nhất:

$$\text{Salary} = (-56546.304) * neuroticism \tag{7}$$

- Kết quả **mae** là: 291019.693226953

d Nhận xét

- Kết quả của **k-fold Cross Validation** đã cho độ chính xác của đặc trưng **neuroticism** là tốt nhất, và kết quả này cũng đã được xác nhận qua các bài nghiên cứu thực tế. Theo bài nghiên cứu của trường đại học kinh tế **Kyiv** (ở Ukraine), 2 trong 5 tính cách có ảnh hưởng nhiều nhất tới mức lương của mỗi cá nhân đó là đức tính **hài lòng (agreeableness)** và **nhạy cảm trong cảm xúc(neuroticism)**.
- Trong đó, mức độ ảnh hưởng của sự nhạy cảm sẽ nhỉnh hơn một chút so với đức tính hòa thuận. Các dữ liệu được tiến hành nghiên cứu tại các nước như Đức, Anh, và Hà Lan ở đây đã chỉ ra rằng: trong khi mức độ của sự hài lòng làm giảm từ 2-5% mức lương ở Đức, hay 4-6% ở Anh, thì mức độ nhạy cảm trong cảm xúc càng cao thì mức lương giảm khoảng 3% ở Đức và 6% ở Anh.
- Điều này cũng rất hợp lý vì những người có mức độ hài lòng cao sẽ có xu hướng ít phấn đấu để cố gắng đạt mức lương cao hơn, hay người có sự không ổn định nhiều trong cảm xúc sẽ làm cho người đó luôn bị phân tâm, cảm thấy mệt mỏi, tiêu cực và điều này ảnh hưởng rất xấu tới công việc hay sự cố gắng của họ ở thời điểm đó. Chính vì vậy ngay trong bảng kết quả của **k-fold Cross Validation** đã cho thấy đặc trưng **neuroticism** và **agreeableness** có độ chính xác khá bằng nhau và **neuroticism** là tính cách có giá trị thấp hơn.

6.3 Yêu cầu 3

a Các bước thực hiện

Cài đặt **best_skill_feature_model** sẽ dùng để thực hiện yêu cầu 2 như sau:

1. Khởi tạo mảng chứa tên 3 đặc trưng kĩ năng để truy xuất các dữ liệu của 3 đặc trưng này.
2. Thực hiện đổi chỗ ngẫu nhiên của cac dòng dữ liệu 1 lần.
3. Gọi hàm **Best_Feature_Skill** để tìm ra đặc trưng tốt nhất theo phương pháp K-fold Cross Validation.
   (a) Thực hiện chia bộ dữ liệu huấn luyện thành 10 nhóm đều nhau (sử dụng hàm **numpy.array_split**).
   (b) Thực hiện huấn luyện trên từng phần:
   - Gọi hàm **getTrain** để lấy tập dữ liệu huấn luyện và tập kiểm tra là phần dữ liệu đang xét.

- Với mỗi đặc trưng tính cách, thực hiện huấn luyện mô hình và tính độ chính xác của mô hình, lưu lại độ chính xác của mô hình trong mảng **mae_arr**.

(c) Sau khi xét hết phần dữ liệu, mảng **mae_arr** thu được gồm 10 dòng và 3 cột tương đương với 10 phần dữ liệu được chia và 3 đặc trưng của mỗi dòng.

(d) Tính độ chính xác trung bình của mỗi đặc trưng bằng cách lấy trung bình cộng của mỗi cột trong mảng **mae_arr** và trả ra giá trị thứ tự của đặc trưng có **mae** nhỏ nhất. Một điều lưu ý là giá trị **mae trung bình** của mỗi đặc trưng sẽ thay đổi mỗi khi chạy chương trình vì việc đổi chỗ các dòng dữ liệu là ngẫu nhiên và khác nhau.

4. Sau khi tìm được đặc trưng tốt nhất, tiến hình truy xuất dữ liệu của đặc trưng đó để huấn luyện mô hình bằng cách sử dụng hàm **fit** của lớp **OLSLinearRegression** và tính độ chính xác của mô hình bằng cách sử dụng hàm **mae**.

b Kết quả tương ứng cho 3 mô hình từ k-fold Cross Validation

STT	Mô hình với 1 đặc trưng	MAE
1	English	121837.9
2	Logical	120230.589
3	Quant	118051.624

c Kết quả tương ứng cho mô hình tốt nhất

- Mô hình tốt nhất là mô hình đặc trưng **Quant** với độ chính xác trung bình là: **118051.624**
- Công thức hồi quy tuyến tính của mô hình tốt nhất:

$$\text{Salary} = (585.895) * neuroticism \tag{8}$$

- Kết quả **mae** là: 106819.5776198967

d Nhận xét

- Kết quả của đánh giá các mô hình từng kĩ năng đã cho thấy **kĩ năng định lượng (quantitative skill)** là đặc trưng tốt nhất trong 3 đặc trưng. Kĩ năng

định lượng là kĩ năng đưa ra đánh giá, nhận xét một vấn đề nào đó dưới góc nhìn toán học, các thông số dữ liệu khổng lồ thể hiện dạng bảng, biểu đồ, đồ thị. Theo nghiên cứu của nhà xuất bản Đại học Chicago ở đây đã cho rằng các kĩ năng, khả năng toán học là sự tạo ra khác biệt giữa mức độ thu nhập của mỗi người theo mô hình Human Capital (mô hình nghiên cứu về sự tăng trưởng kinh tế). Sự khan hiếm trong nhân lực có trình độ toán học cao sẽ làm cho mức lương của họ tăng lên có thể thấy ở các ngành nghề thực tế, đặc trưng nhất là **Data Science hay Data Analyst** - những ngành nghề đòi hỏi kĩ năng toán học cao.

- Chính vì sự cách biệt lớn giữa mức độ lương được tạo ra bởi kĩ năng định lương cho thấy kĩ năng này sẽ dự đoán được mức lương tốt hơn 2 kĩ năng còn lại là ngoại ngữ (tiếng anh) và kĩ năng suy nghĩ logic. Thêm vào đó, dựa vào hệ số mà mô hình được huấn luyện bởi đặc trưng định lượng là một hệ số dương, cho thấy mức lương sẽ tỉ lệ thuận với mức độ của đặc trưng này.

6.4 Yêu cầu 4

Ở yêu cầu này sinh viên phải tự xây dựng các mô hình và chọn ra mô hình tốt nhất, chính vì thế dưới đây sẽ là phần nêu ra phương pháp và quá trình để chọn ra các mô hình. Tổng cộng có 3 mô hình được lựa chọn từ 3 phương pháp khác nhau, lần lượt là:

1. Loại bỏ các đặc trưng có phương sai thấp.
2. Loại bỏ các đặc trưng có mức độ tương quan cao.
3. Chọn ra các đặc trưng có mức độ liên quan cao với đặc trưng mục tiêu.

a Lựa chọn mô hình

a.1 Mô hình 1: Loại bỏ các đặc trưng có phương sai thấp (dropping constant feature)

Mô hình đầu tiên được thực hiện theo phương pháp loại bỏ các **constant feature**. **Constant feature** là đặc trưng mà ở mỗi trạng thái (các dòng) hầu như đưa ra chỉ duy nhất 1 giá trị mà thôi. Giả sử ta có bộ dữ liệu như sau:
Có thể thấy ở **feature 2**, dù cho trong trường hợp nào thì giá trị của đặc trưng này vẫn sẽ là **1**, lúc này feature 2 được gọi là **constant feature**. Việc giữ lại các đặc trưng này sẽ không có ý nghĩa gì vì chúng không đóng góp vào việc dự đoán mục tiêu. Chính vì vậy, ta sẽ loại bỏ các đặc trưng này đi.

Để có thể loại bỏ các đặc trưng này thì sẽ xét giá trị **phương sai (variance)** của

Target	Feature 1	Feature 2	Feature 3
1	10	1	20
2	19	1	11
3	38	1	47

các đặc trưng. Variance là một đo lường cho mức độ biến đổi của các giá trị trong một tập dữ liệu và giá trị ấy được tính theo công thức:

$$Var(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2 \tag{9}$$

với x_i là 1 trong những giá trị của đặc trưng **X** và $\overline{x}$ là giá trị trung bình của đặc trưng **X**. Khi tất cả các giá trị trong một feature đều giống nhau, không có sự thay đổi giữa chúng, do đó, variance của feature này là 0.

Để có thể xác định các phương sai của các đặc trưng trong bộ dữ liệu, ta sẽ sử dụng hàm **var** của thư viện **pandas**, và giá trị phương sai của các đặc trưng có giá trị như sau:

	Feature	Variance
0	Gender	0.18300
1	10percentage	100.97700
2	12percentage	123.20700
3	CollegeTier	0.07100
4	Degree	0.07300
5	collegeGPA	65.78000
6	CollegeCityTier	0.20800
7	English	10826.04700
8	Logical	7776.37800
9	Quant	15286.54200
10	Domain	0.21900
11	ComputerProgramming	41203.28800
12	ElectronicsAndSemicon	24773.34600
13	ComputerScience	31447.16300
14	MechanicalEngg	9175.99100
15	ElectricalEngg	7193.70800
16	TelecomEngg	10830.10900
17	CivilEngg	1091.92400
18	conscientiousness	1.05300
19	agreeableness	0.92600
20	extraversion	0.94200
21	nueroticism	1.03500
22	openess_to_experience	1.06100

Hình 2: Bảng giá trị phương sai của các đặc trưng

Dựa vào bảng phương sai của các đặc trưng, ta sẽ thực hiện loại bỏ các đặc trưng có giá trị phương sai nhỏ hơn 0.1 (các giá trị này có thể được tính xấp xỉ như bằng 0). Ta sẽ sử dụng lớp **VarianceThreshold** với tham số truyền vào phương thức khởi tạo là khoảng ngưỡng phương sai nhỏ nhất, và sử dụng phương thức **fit** của lớp này để loại bỏ các đặc trưng có phương sai nhỏ hơn ngưỡng này. Tất cả thao tác này được thực hiện trong hàm **model_variance_dropping**.

Sau khi loại bỏ các đặc trưng này, ta sẽ thu được bộ dữ liệu mới với 22 đặc trưng như sau: **Gender, 10percentage, 12percentage, collegeGPA, CollegeCityTier, English, Logical, Quant, Domain, ComputerProgramming, ElectronicsAndSemicon, ComputerScience, MechanicalEngg, CivilEngg , TelecomEngg, ElectricalEngg, conscientiousness, agreeableness, extraversion, nueroticism, openess_to_experience**.

a.2 Mô hình 2: Loại bỏ các đặc trưng có mức độ tương quan cao (dropping high-correlation feature)

Hệ số tương quan (**Correlation coefficient**) là giá trị thể hiện mức độ liên kết giữa 2 đặc trưng. Hệ số tương quan có giá trị nằm trong khoảng [-1, 1], với 1 là tương quan hoàn toàn, -1 là tương quan hoàn toàn nghịch đảo và 0 là không có tương quan. Vậy câu hỏi được đặt ra là: **Vì sao hệ số tương quan lại ảnh hưởng tới quá trình huấn luyện dữ liệu?**

Đối với các mô hình hồi quy tuyến tính, một trong những giả định là các đặc trưng là độc lập tuyến tính với nhau. Nếu có sự tương quan giữa các đặc trưng, thì mô hình sẽ không thể xác định được đặc trưng nào là quan trọng hơn đặc trưng nào. Điều này sẽ làm cho mô hình không thể xác định được đặc trưng nào là quan trọng hơn đặc trưng nào, và dẫn tới việc mô hình sẽ không thể dự đoán được mục tiêu. Hiện tượng có các đặc trưng phụ thuộc vào nhau như vậy thì được gọi là hiện tượng **đa cộng tuyến (Multicollinearity)**.

Thêm vào đó, điều này cũng đã được chứng minh dưới góc nhìn toán học như ở đây. Tóm tắt lại rằng giá trị trọng số được tính theo:

$$W_{LS} = (X^T X)^{-1} X^T Y \tag{10}$$

với X là ma trận các đặc trưng, Y là ma trận mục tiêu. Dưới góc nhìn của phân tích hồi quy, giá trị của biến độc lập y sẽ được phân phối bởi giá trị phương sai σ^2, theo giả định này giá trị phương sai của trọng số sẽ là:

$$Var[W_{LS}] = \sigma^2 * (X^T X)^{-1} \tag{11}$$

Theo bài nghiên cứu đối với mô hình được ổn định thì giá trị phương sai của trọng số phải nhỏ. Nếu giá trị phương sai càng lớn thì có nghĩa là giá trị của trọng số cũng sẽ lớn trong mô hình huấn luyện, điều này sẽ dẫn tới việc mô hình sẽ không thể dự đoán được mục tiêu.

Điều này được chứng minh thông qua **phương pháp phân tích suy biến**. Phương pháp này sẽ phân tích ma trận X thành 3 ma trận U, S, V sao cho:

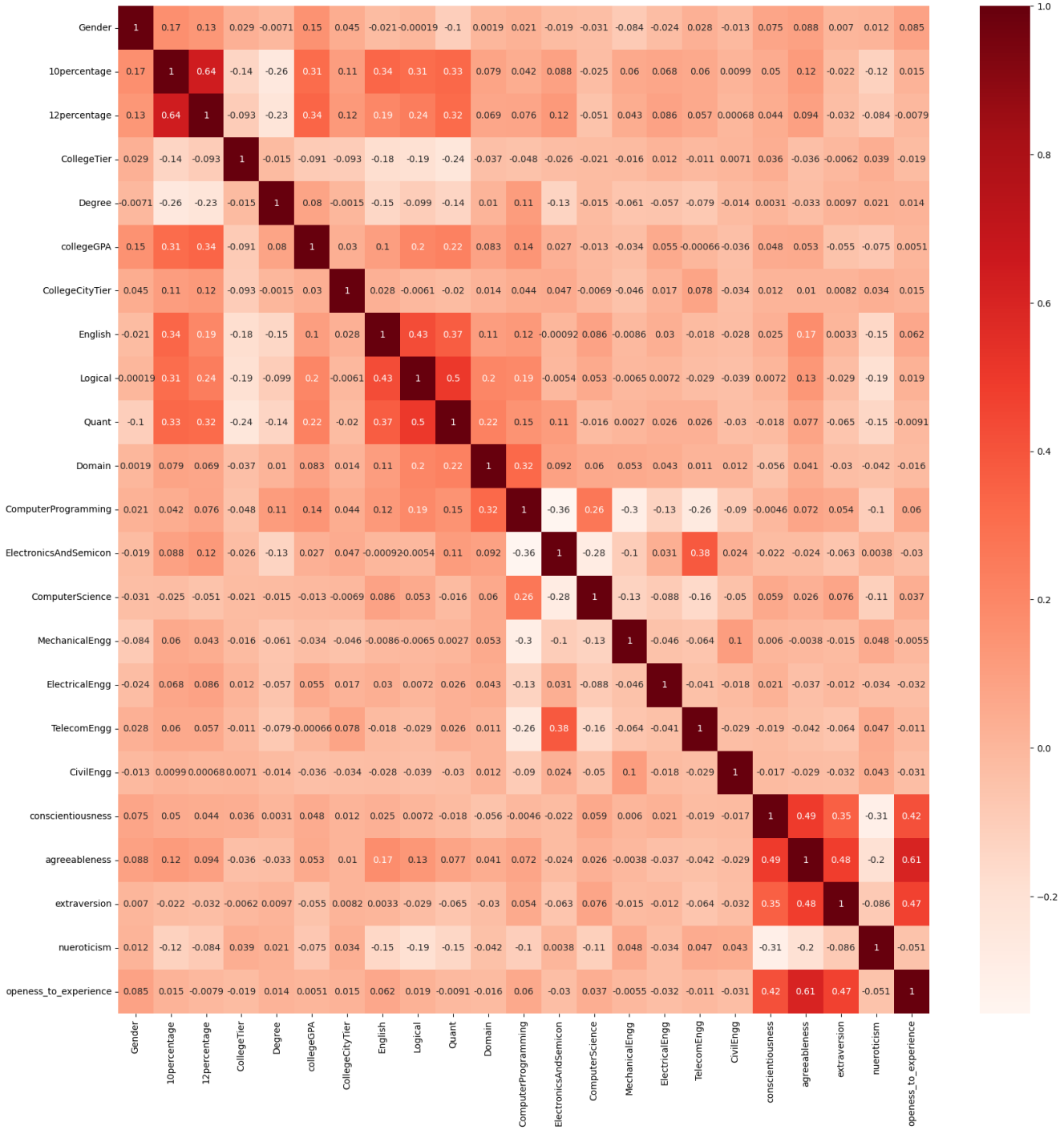
$$X = USV^T \tag{12}$$

Như vậy, phương sai của của trọng số có thể được viết lại như sau:

$$\begin{aligned}\text{Var}[W_{LS}] &= \sigma^2 \cdot (X^T X)^{-1} \\ &= \sigma^2 \cdot (VS^T U^T USV^T)^{-1} \\ &= \sigma^2 \cdot (VS^T SV^T)^{-1} \\ &= \sigma^2 \cdot V(S^T S)^{-1} V^T \\ &= \sigma^2 \cdot VS^{-2} V^T\end{aligned}$$

Vì vậy khi giá trị tương quan cao, giá trị của ma trận "S" sẽ nhỏ, và khi nghịch đảo ma trận S thì giá trị của nó sẽ càng lớn. Điều này sẽ dẫn tới việc giá trị phương sai của trọng số sẽ càng lớn.

Để có thể xác định được các đặc trưng có mức độ tương quan cao với nhau thì ta sẽ sử dụng hàm **corr** của thư viện **pandas**. Bước đầu sẽ truy xuất tất cả cột thông tin của các đặc trưng ngoại trừ cột mục tiêu ("Salary"), tính hệ số tương quan và thể hiện giá trị tương quan của các đặc trưng dưới dạng biểu đồ heatmap như sau:



Hình 3: Biểu đồ heatmap thể hiện mức độ tương quan của các đặc trưng

Theo nội dung của biểu đồ, ta có thể giá trị hệ số cao nhất là 0.64 giữa 2 đặc trưng **10percentage và 12percentage**, ta có thể đánh giá tổng quan rằng các đặc trưng có mức độ tương quan không quá cao, tuy nhiên ta vẫn sẽ tiến hành loại bỏ 1 trong 2 đặc trưng có **giá trị (không quan tâm tới dấu của giá trị)** hệ số tương quan lớn hơn hoặc bằng 0.5.

Sau khi loại bỏ các đặc trưng này, ta sẽ thu được bộ dữ liệu mới với 21 đặc trưng để tạo mô hình 2 như sau: **Gender, 10percentage, CollegeTier, Degree, collegeGPA, CollegeCityTier, English, Logical, Quant, Domain, ComputerProgramming, ElectronicsAndSemicon, ComputerScience, MechanicalEngg, CivilEngg, ElectricalEngg, TelecomEngg, conscientiousness, agreeableness, extraversion, nueroticism**.

a.3 Mô hình 3: Chọn ra các đặc trưng có mức độ liên quan cao với đặc trưng mục tiêu (selecting high-MI feature)

Trong các bỗ dữ liệu thì giữa từng đặc trưng và đặc trưng mục tiêu sẽ có mức độ liên quan với nhau. Mức độ liên quan này sẽ được đo bằng **hệ số tương quan** như đã được trình bày ở trên, và cũng tương tự vậy, giữa các đặc trưng trong bỗ dữ liệu luôn sẽ có mối liên kết với đặc trưng mục tiêu, và mức độ của sự liên quan này chính là giá trị của **mutual information**.

Mutual information là một đo lường cho mức độ liên kết giữa 2 biến ngẫu nhiên. Giá trị của **mutual information** nằm trong khoảng [0, 1], với 0 là không có liên kết và 1 là liên kết hoàn toàn. Để đề cập tới cách thức tính giá trị **mutual information** thì ta sẽ cần đến khái niệm **entropy**.

Entropy là một đo lường cho mức độ bất định của một biến ngẫu nhiên, và công thức để tính entropy của 1 biến ngẫu nhiên **X** là:

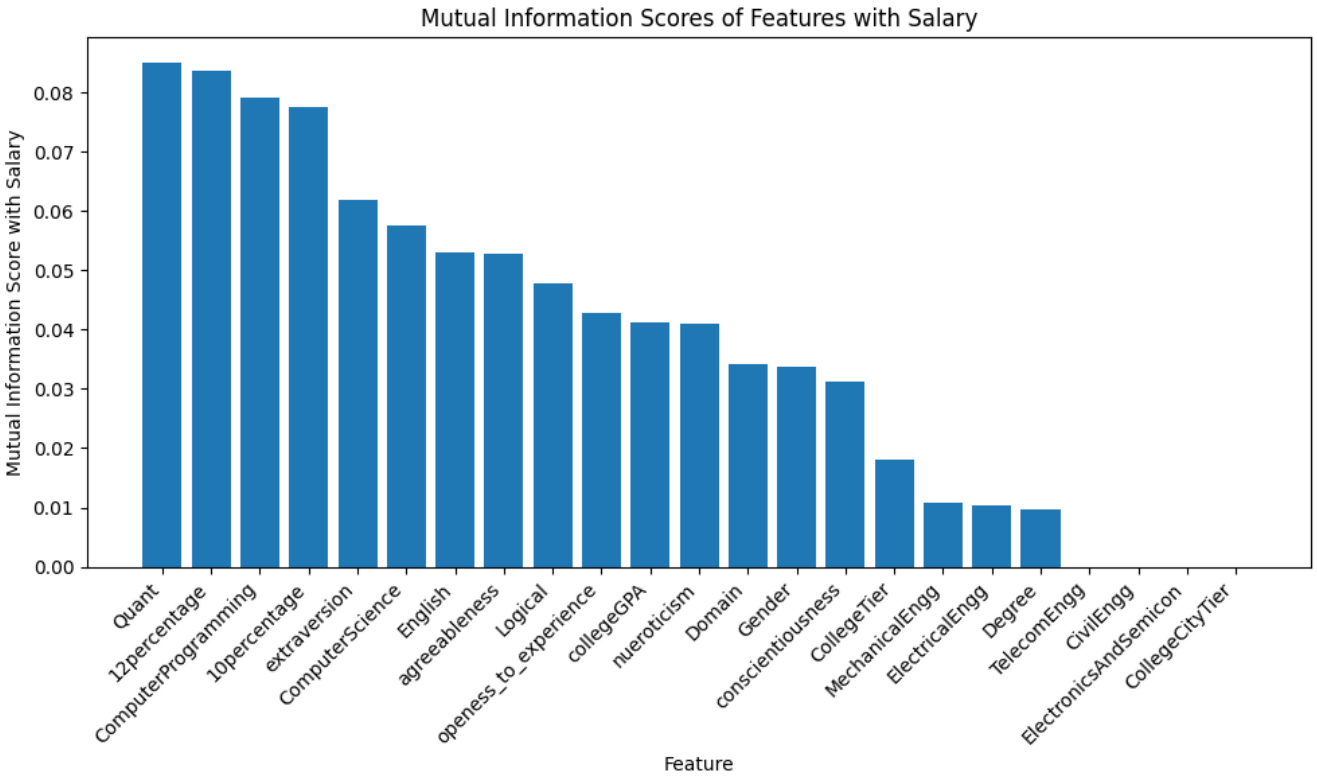
$$H(X) = -\sum_{x \in X} p(x_i) log_2 p(x_i) \tag{13}$$

với $p(x_i)$ là xác suất của biến ngẫu nhiên **X** có giá trị x_i. Thông qua entropy, các khái niệm để đo lường thông tin tương hỗ (lượng thông tin chung của 2 biến ngẫu nhiên) ra đời, và một trong số đó là **entropy có điều kiện** liên quan tới giá trị **mutual information**.

Conditional entropy là một đo lường cho mức độ bất định của một biến ngẫu nhiên **X** khi biết giá trị của một biến ngẫu nhiên **Y**. Bản chất của giá trị **mutual information** chính là tính toán sự khác biệt giữa **entropy** của biến ngẫu nhiên **X** và **conditional entropy** của biến ngẫu nhiên **X** khi biết giá trị của biến ngẫu nhiên **Y**. Công thức để tính giá trị **mutual information** là:

$$I(X; Y) = H(X) - H(X|Y) \tag{14}$$

Như vậy, bước đầu của việc thiết lập mô hình này chính là tách dữ liệu của các đặc trưng và đặc trưng mục tiêu ("Salary"). Đồng thời cũng sẽ tạo ra 1 biến với kiểu dữ liệu là lớp **SelectPercentile** với hàm tính toán truyền vào sẽ là tính **mutual_info_regression**. Sau đó, sử dụng hàm **fit** để tính toán giá trị **mutual information** theo hàm tính toán được truyền vào. Trong các lần lấy thông tin và sắp xếp lại giá trị liên quan giữa các đặc trưng với đặc trưng mục tiêu là như biểu đồ cột dưới đây:



Hình 4: Biểu đồ thể hiện giá trị mutual information của các đặc trưng theo thứ tự giảm dần

Sau quá trình thử chọn ra bao nhiêu phần trăm đặc trưng thì em đã quyết định chọn ra 70% đặc trưng có giá trị **mutual information** cao nhất để tạo mô hình. Sau khi chọn ra các đặc trưng này, ta sẽ thu được bộ dữ liệu mới với 16 đặc trưng để tạo mô hình 3 như sau: **Gender, 10percentage, 12percentage, CollegeTier, collegeGPA, English, Logical, Quant, Domain, ComputerProgramming, ComputerScience, conscientiousness, agreeableness, extraversion, nueroticism, openess_to_experience**.

b Lựa chọn mô hình tốt nhất

Sau khi đã hình thành được 3 mô hình thì sẽ tiến hành lựa chọn mô hình tốt nhất dựa trên phương pháp **k-fold Cross Validation**.Với mỗi mô hình cũng sẽ thực hiện bằng cách chia các dòng dữ liệu thành các phần bằng nhau, và chọn tập huấn luyện và kiểm tra với tất cả đặc trưng của mô hình đó. Sau khi thực hiện tính giá trị **mae** thì sẽ lấy **mae trung bình** đại diện cho tính hiệu quả của mô hình đó. Kết quả cho 3 mô hình từ **k-fold Cross Validation** là:

STT	Mô hình	MAE
1	Sử dụng 22 đặc trưng (Tất cả ngoại trừ CollegeTier, Degree)	112176.27596763
2	Sử dụng 21 đặc trưng (Tất cả ngoại trừ 12%, Quant, openess_to_exp)	111849.95116329
3	Sử dụng 17 đặc trưng (ngoại trừ từ MechanicalEngg đến hết trong bảng biểu đồ cột)	110724.24861457

Và theo như kết quả của bảng, **mô hình có mae nhỏ nhất chính là mô hình 3** với 16 đặc trưng được chọn ra.

c Kết quả tương ứng cho mô hình tốt nhất

Sau khi có được mô hình tốt nhất thì sẽ tiến hành huấn luyện và kiểm tra trên mô hình tốt nhất, và công thức hồi quy tuyến tính của mô hình tốt nhất là:

- $$\begin{aligned}\text{Salary} = {} & (-23438.408) \times \text{Gender} + (887.524) \times \text{10percentage} \\ & + (971.041) \times \text{12percentage} + (-79706.706) \times \text{CollegeTier} \\ & + (1689.069) \times \text{collegeGPA} + (148.766) \times \text{English} \\ & + (132.912) \times \text{Logical} + (95.513) \times \text{Quant} \\ & + (22104.286) \times \text{Domain} + (104.259) \times \text{ComputerProgramming} \\ & + (-164.058) \times \text{ComputerScience} + (-20216.903) \times \text{conscientiousness} \\ & + (16789.581) \times \text{agreeableness} + (5018.386) \times \text{extraversion} \\ & + (-9649.493) \times \text{nueroticism} + (-6317.617) \times \text{openess\_to\_experience}\end{aligned} \tag{15}$$

- Kết quả **mae** là: 102431.57002986001

d Nhận xét các mô hình

- Cả 3 mô hình được xây dựng đã có thể đem lại dự đoán tốt hơn so với **3 yêu cầu** trước vì đã có việc chọn lọc các đặc trưng theo chủ đích, tuy nhiên, mô hình 3 vẫn có thể đem lại kết quả tốt hơn so với 2 mô hình còn lại.

- Mô hình 1 đã loại bỏ được 1 số đặc trưng có phương sai thấp, tuy nhiên, mô hình này vẫn còn đặc trưng có phương sai thấp, và đặc trưng có phương sai thấp này vẫn có thể ảnh hưởng tới mô hình.
- Mô hình 2 đã loại bỏ một số đặc trưng có hệ số tương quan cao, tuy nhiên nhìn tổng thể trong mô hình này, các đặc trưng có hệ số tương quan khá là thấp (cao nhất là 0.64) thì khi loại bỏ đi cũng chưa làm thay đổi quá nhiều về sự hiệu quả của mô hình.
- Đối với mô hình 3, dữ liệu cho thấy có khá nhiều đặc trưng không có giá trị tương hỗ với đặc trưng mục tiêu, vì vậy ta có thể bỏ đi được khá nhiều đặc trưng thừa hay làm cho mô hình huấn luyện không tốt. Từ đó độ chính xác của mô hình này mang lại rõ ràng sẽ khá hiệu quả hơn là 2 mô hình trước đó.

7 TÀI LIỆU THAM KHẢO

- Cô Phan Thị Phương Uyên.
- Khái niệm hồi quy tuyến tính.
- K-fold Cross Validation.
- Lớp VarianceThreshold.
- Lớp mutual_info_regression.
- Lớp SelectPercentile.
- Hàm numpy.array_split.
- Bài nghiên cứu về ảnh hưởng của 5 đặc trưng cảm xúc tới mức độ lương (truy cập: 11/8/2023).
- Bài tìm hiểu về khoảng cách lương giữa cá nhân và kĩ năng toán học (Truy cập: 20/8/2023).
- Lý do phải loại bỏ các constant feature.
- Vì sao có các đặc trưng tương quan cao thì lại không tốt trong mô hình huấn luyện?
- Lý do vì sao có các đặc trưng có tương quan cao thì lại không tốt theo góc nhìn toán học?
- Lý thuyết và công thức tính toán của mutual information.
- Cài đặt loại bỏ các constant feature.
- Loại bỏ các đặc trưng có tương quan cao.

- Chọn nhóm các đặc trưng nằm trong top đầu có mức độ tương hỗ cao với đặc trưng mục tiêu.