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


MÔN : TOÁN ỨNG DỤNG VÀ THỐNG KÊ

CHO CÔNG NGHỆ THÔNG TIN

# BÁO CÁO ĐỒ ÁN 3 :

# LINEAR REGRESSION

## MTH00057 – 21CLC01

**GIẢNG VIÊN:** NGUYỄN ĐÌNH THÚC

NGÔ ĐÌNH HY

NGUYỄN VĂN QUANG HUY

**SINH VIÊN THỰC HIỆN:** NGUYỄN THÁI ĐAN SÂM – 21127414

# Mục lục

# Phần I. Giới thiệu đồ án

## 1. Hồi quy tuyến tính

Hồi quy tuyến tính" là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục. Ví dụ, dự đoán giao thông ở một cửa hàng bán lẻ, dự đoán thời gian người dùng dừng lại một trang nào đó hoặc số trang đã truy cập vào một website nào đó v.v...

Trong đồ án 3 – Linear Regression sẽ trình bày các bước xây dựng một số mô hình hồi quy tuyến tính cho bộ dữ liệu “Mức lương của kỹ sư đã tốt nghiệp đại học” sử dụng một hay nhiều đặc trưng từ bộ dữ liệu

## 2. Ước lượng hồi quy tuyến tính bằng phương pháp bình phương tối thiểu OLS ( Ordinary Least Squares )

Phương pháp bình phương tối thiểu là một phương pháp thống kê được sử dụng để tìm đường thẳng phù hợp nhất theo dạng của phương trình như y = ax + b cho các dữ liệu cho trước. Đường cong của phương trình được gọi là đường hồi quy. Mục tiêu chính của chúng ta trong phương pháp này là giảm tổng bình phương sai số càng nhiều càng tốt. Nghĩa là giá trị sau đây càng nhỏ càng tốt :

$$e^2 = (y - \hat{y})^2 = (y - \bar{x}w)^2$$

Đây là lý do tại sao phương pháp này được gọi là phương pháp bình phương tối thiểu. Phương pháp này thường được sử dụng trong việc khớp dữ liệu với giả định rằng kết quả phù hợp nhất là giảm tổng bình phương sai số được xem là khác biệt giữa các giá trị quan sát và giá trị phù hợp tương ứng.

Điều tương tự xảy ra với tất cả các cặp (x$\bar{i}$,y$i$), $i$=1,2,3,…$n$ với $n$ là số lượng dữ liệu quan sát được. Điều chúng ta muốn, tổng sai số là nhỏ nhất, tương đương với việc tìm $w$ để hàm số sau đạt giá trị nhỏ nhất:

$$\mathcal{L}(\mathrm{w}) = \sum_{i=1}^{n}(y_i - \bar{x}_l w)^2$$

Hàm số $\mathcal{L}(w)$ được gọi là hàm mất mát (loss function) của bài toán Linear Regression. Chúng ta luôn mong muốn rằng sự mất mát (sai số) là nhỏ nhất, điều đó đồng nghĩa với việc tìm vector hệ số $w$ sao cho giá trị của hàm mất mát này càng nhỏ càng tốt. Giá trị của $w$ làm cho hàm mất mát đạt giá trị nhỏ nhất được gọi là điểm tối ưu (optimal point), ký hiệu:

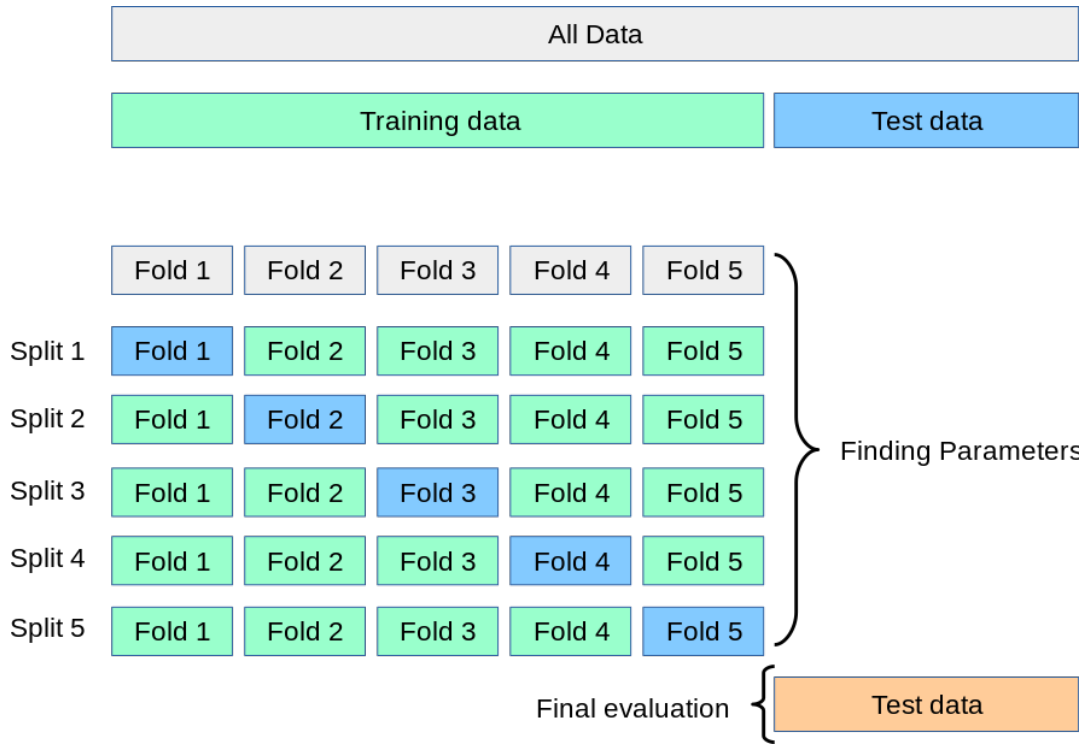
$$w^* = \arg\min_w \mathcal{L}(w)$$

Tìm $w^*$ để cho $|| y - \bar{x}w ||^2$ đạt giá trị nhỏ nhất được gọi là bài toán bình phương tối thiểu (OLS – Ordinary Least Squares).

Ta có được điểm tối ưu của bài toán bình phương tối thiểu là $w^* = (\bar{X}^T\bar{X})^{-1}\bar{X}^T y$, nếu $(\bar{X}^T\bar{X})$ khả nghịch thì $w^* = \bar{X}^{-1}y$ và $(\bar{X}^T\bar{X})^{-1}\bar{X}^T$ được gọi là giả nghịch đảo của $\bar{X}$ .

## 3. K-fold Cross Validation

K-fold cross-validation là một kỹ thuật chia dữ liệu thành k phần được gọi là "folds". Sau đó, mô hình được huấn luyện bằng cách sử dụng k-1 folds được tích hợp vào một tập huấn luyện duy nhất, và nếp gấp cuối cùng được sử dụng như một tập thử nghiệm. Điều này được lặp lại k lần, mỗi lần sử dụng một fold khác nhau làm tập thử nghiệm. Hiệu suất của mô hình sau đó được tính trung bình qua k lần lặp để cung cấp phép đo tổng thể.



Ở đồ án này, ta sẽ tiến hành thực hiện K-fold Cross Validation trên tập dữ liệu được chia sẵn là train và test, thực hiện trên train trước để tính toán xem mô hình nào tối ưu nhất, sau đó áp dụng vào tập dữ liệu test.

## 4. Sai số tuyệt đối trung bình MAE ( Mean Absolute Error )

Trong đồ án này, phương pháp được dùng để tính toán sai số là MAE, với công thức như sau :

$$\boldsymbol{MAE} = \frac{\sum |\boldsymbol{y_i} - \boldsymbol{\hat{y}_i}|}{\boldsymbol{n}}$$

Trong đó:
n là số lượng mẫu quan sát
$\boldsymbol{y_i}$ là giá trị mục tiêu của mẫu thứ i
$\boldsymbol{\hat{y}_i}$ là giá trị mục tiêu của mẫu thứ i được dự đoán từ mô hình hồi quy tuyến tính

# Phần II. Các thư viện hỗ trợ và hàm sử dụng trong đồ án

## 1. Các thư viện hỗ trợ

- `pandas`: Là thư viện được sử dụng để thao tác, phân tích và dọn dẹp dữ liệu. Cung cấp rất nhiều cấu trúc dữ liệu cũng như các phép tính hỗ trợ thao tác dữ liệu số và dữ liệu thời gian(time series). Pandas nhanh, mạnh và hiệu quả.
- `numpy`: Là một thư viện toán học phổ biến và mạnh mẽ của Python. Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng code đơn thuần.
- `matplotlib`: Là một thư viện có khả năng hỗ trợ trực quan hóa dữ liệu. Nó mang đến những công cụ phục vụ việc vẽ biểu đồ đường, cột, hay biểu đồ tròn,… Không những thế, thư viện này sẽ tạo ra nhiều biểu đồ tương ứng với những điều kiện khác nhau như xử lý dữ liệu, hiển thị kết quả cũng như mô phỏng hóa nó.
- `seaborn`: Là một thư viện được sử dụng để hỗ trợ trực quan hóa dữ liệu dựa trên Matplotlib. Seaborn cho phép tạo đồ họa thống kê thông qua nhiều chức năng, giúp tăng tính thẩm mỹ cho các biểu đồ, thống kê,…
- `sklearn`: Là thư viện đóng vai trò quan trọng cho ngôn ngữ lập trình Python. Mục đích chính là xử lý dữ liệu và những thao tác liên quan đến học máy. Nói cách khác, nó hỗ trợ nhiều công cụ cho việc xử lý dữ liệu sơ khởi. Từ đó, giúp ta lựa chọn mô hình học máy cũng như tối ưu những tham số về sau.

## 2. Các hàm trong chương trình

### 2.1 Hàm hỗ trợ

- `def MAE(X, y, model)`

Hàm sử dụng phương pháp dự đoán của mô hình được cung cấp để dự đoán các giá trị đích dựa trên các dữ liệu đầu vào. Sau đó tính toán ra MAE bằng cách so sánh các giá trị được dự đoán với các giá trị thực bằng cách sử dụng hàm mean_absolute_error từ metrics module của thư viện scikit_learnin

MAE tính ra sau đó sẽ được trả về dưới dạng đầu ra của hàm

- **`def pre_Process(X)`**

Hàm có chức năng thêm vào 1 cột toàn giá trị 1 vào ma trận đầu vào với mục đích phục vụ cho việc tính toán với mô hình hồi quy tuyến tính. Kết quả trả về của hàm này là 1 ma trận mới dạng NumPy

- **`def find_best_feature(x_value, y_value, list_features, k=5)`**

Hàm này có chức năng tìm kiếm và xác định đặc trưng tốt nhất từ tập list_features và xác định MAE của đặc trưng đó. Ở mỗi vòng lặp, k_fold_cross_validation sẽ được gọi ra để thực hiện quá trình cross_validation bằng cách sử dụng 'k' fold. Nó sẽ đo lường hiệu suất của mô hình sử dụng đặc trưng trong tập list_features thông qua việc tính toán MAE.

- **`def shuffle_data(X_value, y_value, random_state=42)`**

Hàm này dùng để xáo trộn dữ liệu đầu vào ( tất cả các đặc trưng đầu vào ngoại trừ "Salary" ), mục đích chính là tạo sự ngẫu nhiên trong dữ liệu, đảm bảo rằng mô hình không bị ảnh hưởng bởi thứ tự ban đầu, đồng thời đảm bảo tính khách quan và tin cậy trong việc đánh giá hiệu suất.

### 2.2 `class OLSLinearRegression`

Class này được thiết kế dựa trên ý tưởng class LinearRegression từ thư viện scikit-learn (sklearn). Lớp này tạo ra một mô hình hồi quy tuyến tính dựa trên phương pháp bình phương tối thiểu thông thường và cung cấp các phương thức để huấn luyện, dự đoán và truy xuất thông tin liên quan đến mô hình. Trong class gồm có các hàm:

- **`def fit(self, X, y):`** Hàm tính điểm tối ưu của bài toán bình phương tối thiểu, với công thức là :

$$w^* = (\bar{X}^T \bar{X})^{-1} \bar{X}^T y$$

- **`def get_params(self):`** Hàm trả về điểm tối ưu đã tính.
- **`def predict(self, X):`** Hàm dự đoán các giá trị tương ứng vừa tính được.
- **`def get_value(self):`** Hàm này dung để lấy Hệ số ( Coefficient ) và Sai số ( Intercept ) của mô hình hồi quy tuyến tính.

### 2.3 `def k_fold_cross_validation(x_value, y_value, feature, k=5)`

Hàm này mô tả việc kiểm chứng chéo bằng cách chia tập dữ liệu đầu vào thành k tập dữ liệu con ( gọi là fold và k ở đồ án này được mặc định là 5 ) để huấn luyện và kiểm tra mô hình cần xây dựng. Nó xây dựng mô hình từ dữ liệu của tập train và tính MAE với giá trị dự đoán từ dữ liệu tập test. Số MAE cần tính tương đương với số fold đầu vào ( giá trị k ). Kết quả trả về là giá trị trung bình các MAE tính được trước đó.

# Phần III. Xử lí các yêu cầu của đồ án

## 1.Yêu cầu 1a :

Sử dụng 11 đặc trưng đầu tiên gồm : Gender, 10percentage, 12percentage, CollegeTier, Degree, collegeGPA, CollegeCityTier, English, Logical, Quant và Domain

- Huấn luyện 1 lần duy nhất cho 11 đặc trưng nói trên cho toàn bộ tập huấn luyện ( train.csv )
- Thể hiện công thức cho mô hình hồi quy (tính y theo 11 đặc trưng trên)
- Báo cáo kết quả trên tập kiểm tra ( test.csv ) cho mô hình vừa huấn luyện được

Thực hiện : Dùng iloc để lấy 11 đặc trưng đầu tiên trong tập dữ liệu X_train rồi biến đổi thành ma trận numpy. Cột giá trị mục tiêu y_train cũng được biến đổi thành ma trận numpy để phù hợp với mô hình hồi quy tuyến tính model1a. array trả về là 1 mảng hệ số và giá trị sai số nằm ở cuối như sau :

```
array([-23183.33 ,    702.767,   1259.019, -99570.608,  18369.962,
        1297.532,  -8836.727,    141.76 ,    145.742,    114.643,
       34955.75 ,  49248.09 ])
```

Dùng mô hình trên áp dụng vào tập dữ liệu test và tính MAE. Tương như train, tập dữ liệu test chỉ lấy 11 cột đầu tiên và cũng chuyển về numpy trước khi tính toán. Sau khi tính toán, công thức hồi quy thu được là :

$$Salary = -23183.330 * Gender + 707.767 * 10percentage + 1259.019 * 12percentage - 99570.608 * CollegeTier + 18369.962 * Degree + 1297.532 * collegeGPA - 8836.727 * CollegeCityTier + 141.760 * English + 145.742 * Logical + 114.643 * Quant + 34955.750 * Domain + 49248.090$$

## 2.Yêu cầu 1b :

Phân tích ảnh hưởng của đặc trưng ngoại ngữ, lô-gic, định lượng đến mức lương của các kỹ sư trên điểm các bài kiểm tra của AMCAT

- Thử nghiệm lần lượt trên các đặc trưng tính cách gồm: English, Logical, Quant

- Yêu cầu sử dụng k-fold Cross Validation (k tối thiểu là 5) để tìm ra đặc trưng tốt nhất trong các đặc trưng tính cách
- Báo cáo 5 kết quả tương ứng cho 5 mô hình từ k-fold Cross Validation (lấy trung bình)

Thực hiện :Lấy ra 3 đặc trưng theo yêu cầu từ tập dữ liệu X_train, sử dụng hàm find_best_feature để tính MAE cho từng đặc trưng ( tính bằng k_fold_cross_validation với k = 5). Lưu các giá trị MAE đó vào array tên mae_scores và xác định đặc trưng tốt nhất bằng cách tìm min của array đó và đặc trưng tương ứng với giá trị min đó. Ta tìm được MAE của các đặc trưng như sau :

| Feature | MAE |
|---|---|
| English | 120121.858199 |
| Logical | 119460.779750 |
| Quant | 116865.817199 |

Đặc trưng tốt nhất là Quant với MAE tương ứng là 116865.817199, dùng tập dữ liệu train ( đã lấy ra đặc trưng Quant ) để huấn luyện mô hình model1c và lấy ra Hệ số, Sai số. Hệ số và Sai số thu được lần lượt là 368.852 và 117759.729. Sau đó dùng tập test ( đã lấy ra đặc trưng Quant ) để tính MAE, giá trị MAE thu được là 108814.059 ( đã làm tròn 3 chữ số ). Công thức hồi quy thu được là :

$$Salary = 368.852 * Quant + 117759.729$$

# 3.Yêu cầu 1c :

Phân tích ảnh hưởng của đặc trưng tính cách dựa trên điểm các bài kiểm tra của AMCAT
- Thử nghiệm lần lượt trên các đặc trưng tính cách gồm: conscientiousness, agreeableness, extraversion, nueroticism, openess_to_experience
- Yêu cầu sử dụng k-fold Cross Validation (k tối thiểu là 5) để tìm ra đặc trưng tốt nhất trong các đặc trưng tính cách
- Báo cáo 5 kết quả tương ứng cho 5 mô hình từ k-fold Cross Validation (lấy trung bình)

Thực hiện : Dùng iloc để lấy 5 đặc trưng cuối trong tập dữ liệu X_train, sử dụng hàm find_best_feature để tính MAE cho từng đặc trưng ( tính bằng k_fold_cross_validation

với k = 5). Lưu các giá trị MAE đó vào array tên mae_scores và xác định đặc trưng tốt nhất bằng cách tìm min của array đó và đặc trưng tương ứng với giá trị min đó. Ta tìm được MAE của các đặc trưng như sau :

| Feature | MAE |
|---|---|
| conscientiousness | 124016.187460 |
| agreeableness | 123098.142270 |
| extraversion | 123668.596905 |
| nueroticism | 123067.984574 |
| openess_to_experience | 123687.015610 |

Đặc trưng tốt nhất là neuroticism với MAE tương ứng là 123067.984574, dùng tập dữ liệu train ( đã lấy ra đặc trưng neuroticism ) để huấn luyện mô hình model1b và lấy ra Hệ số, Sai số. Hệ số và Sai số thu được lần lượt là -16021.494 và 304647.553. Sau đó dùng tập test ( đã lấy ra đặc trưng neuroticism ) để tính MAE, giá trị MAE thu được là 119361.917 ( đã làm tròn 3 chữ số ). Công thức hồi quy thu được là :

$$Salary = -16021.494 * neuroticism + 304647.553$$

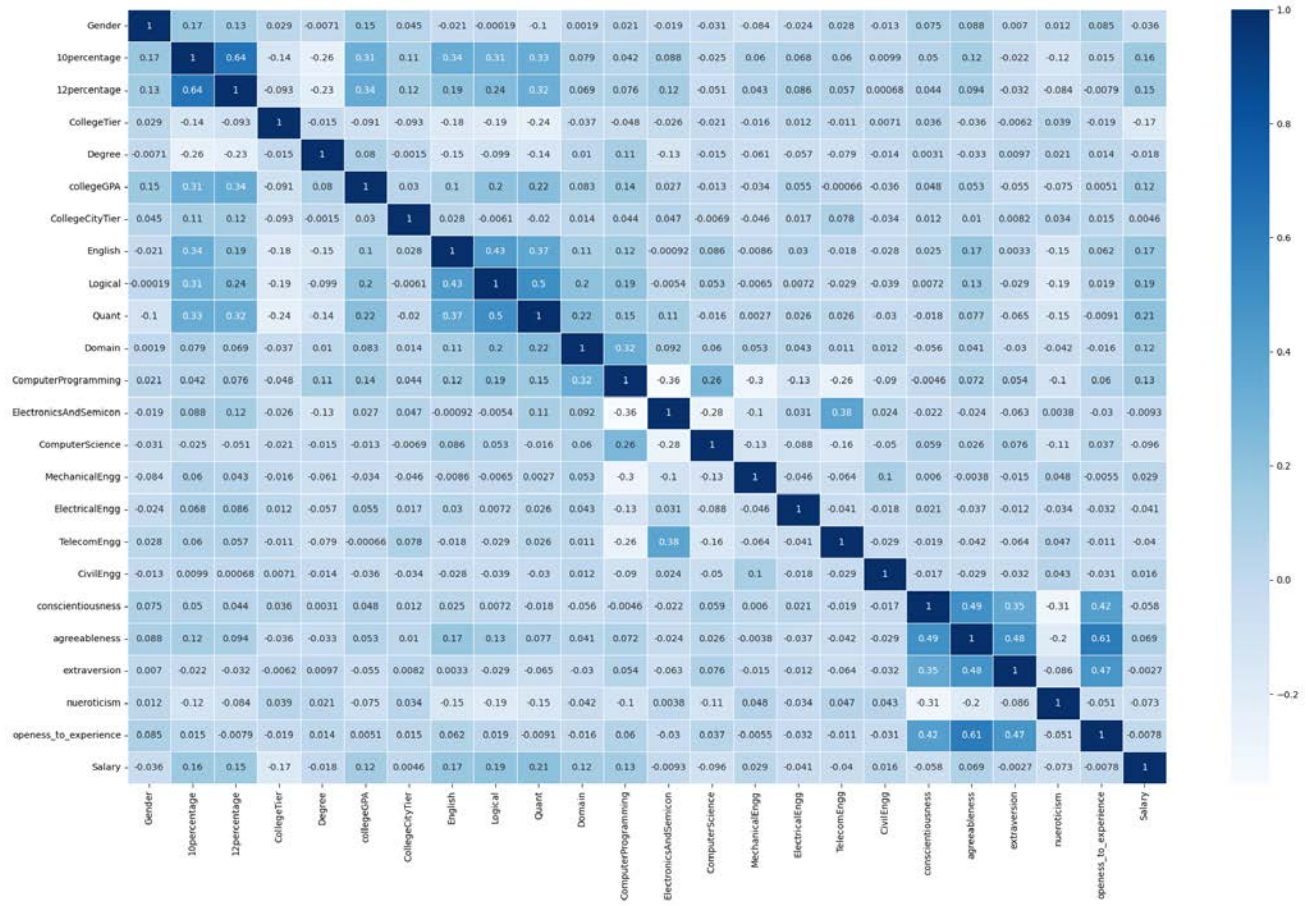
# 4.Yêu cầu 1d :

Sinh viên tự xây dựng mô hình, tìm mô hình cho kết quả tốt nhất

- 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...)

Xét các yếu tố tạo nên một mô hình hồi quy tuyến tính tốt, trước hết các đặc trưng phải có sức ảnh hưởng đến các biến phụ thuộc, tuy nhiên các đặc trưng đó cần tránh phụ thuộc lẫn nhau, hay còn gọi là tính tương quan giữa các đặc trưng. Hệ số tương quan là chỉ số đo lường mức độ tương quan tuyến tính giữa hai biến, dao động từ -1 đến 1. Để quan sát được độ tương quan giữa các đặc trưng, pandas cho phép xây dựng ma trận tương quan từ bộ dữ liệu bất kì. Ta sẽ hiện thị ma trận tương quan giữa các

đặc trưng trong tập train bằng sử dụng biểu đồ nhiệt ( heatmap ) từ thư viện seaborn để trực quan hóa dữ liệu.




Từ ma trận tương quan trên, ta thấy :

- Tất cả 23 đặc trưng đều có mối tương quan với Salary, từ yếu đến rất yếu ( <0.5 )
- 10percentage, 12percentage, collegeGPA, English, Logical, Quant, Domain, ComputerProgramming là những đặc trưng có tính tương quan với Salary mạnh nhất trong số 23 đặc trưng.
- Các đặc trưng tính cách gồm conscientiousness , agreeableness, extraversion, nueroticism, openess_to_experience có tương quan mạnh với nhau.
- 10percentage và 12percentage có tính tương rất mạnh với nhau.
- Các đặc trưng kĩ năng English, Logical, Quant có tính tương quan mạnh với nhau.
- Salary phụ thuộc nhiều nhất vào các đặc trưng 10percentage, 12percentage, CollegeTier, collegeGPA, English, Logical, Quant, Domain và Computer-Programming

**Mô hình 1** : Sử dụng 7 đặc trưng 10percentage, 12percemtage, collegeGPA, Domain, Quant, ComputerProgramming và ComputerScience
Ý tưởng : Lấy ý tưởng từ việc chọn ra các đặc trưng có hệ số tương quan với Salary cao nhất, sau nhiều lần thử nghiệm và xem xét các hệ số tương quan của từng đặc trưng với nhau, mô hình 1 được tạo ra bằng cách giữ lại 6 trong số 8 đặc trưng có hệ số tương quan mạnh nhất với Salary ( bỏ đi English và Logical ) và thêm vào đặc trưng ComputerScience ( do ComputerProgramming và ComputerScience có tính tương quan với nhau nổi bật ). Mô hình 1 cho ra kết quả như sau :
Giá trị MAE trên tập dữ liệu train : **111322.059**
Giá trị MAE trên tập dữ liệu test : **101702.94**

**Mô hình 2** : Sử dụng đặc trưng Domain và ComputerProgramming kết hợp với 3 đặc trưng mới là School, Major và Life
Ý tưởng : Lấy ý tưởng từ việc chọn ra các đặc trưng có hệ số tương quan với Salary cao nhất, sau đó gom nhóm các đặc trưng có liên quan với nhau lại và từ đó hình thành đặc trưng mới :

- **School** : được tạo ra bằng cách nhân 10percentage, 12percentage và collegeGPA lại với nhau.
- **Major**: được tạo ra bằng cách nhân ComputerProgramming và ComputerScience lại với nhau ( do ComputerProgramming và ComputerScience có tính tương quan với nhau nổi bật ).
- **Life** : được tạo ra bằng cách nhân English, Logical và Quant lại với nhau.

Sau khi gom nhóm, còn lại Domain, quyết định để Domain là 1 đặc trưng chung với 3 đặc trưng vừa mới tạo, xét thấy Domain có tính tương quan với ComputerProgramming cao nhất nên chọn thêm ComputerProgramming. Mô hình 2 cho ra kết quả như sau :
Giá trị MAE trên tập dữ liệu train : **111763.327**
Giá trị MAE trên tập dữ liệu test : **104069.534**

**Mô hình 3** : Sử dụng 5 đặc trưng là collegeGPA, English, Quant, ComputerProgramming và ComputerScience kết hợp với đặc trưng mới Learning. Đặc trưng này được tạo ra bằng cách kết hợp trung bình của 5 đặc trưng kia. Chọn 5 đặc trưng kia vì collegeGPA và English có tính tương quan nổi bật với Quant đồng thời ComputerProgramming và ComputerScience có tính tương quan với nhau nổi bật. Mô hình 3 cho ra kết quả như sau :
Giá trị MAE trên tập dữ liệu train : **110356.067**
Giá trị MAE trên tập dữ liệu test : **102608.508**

Bảng tổng kết :

| STT | Mô hình | MAE ( test ) |
|---|---|---|
| 1 | 7 đặc trưng (10percentage, 12percemtage, collegeGPA, Domain, Quant, ComputerProgramming, ComputerScience ) | **101702.94** |
| 2 | 1 đặc trưng Domain và 3 đặc trưng nhân đặc trưng ( Study, IT, Skill ) | 104069.534 |
| 3 | 4 đặc trưng (Logical, Quant, ComputerProgramming, ComputerScience ) và 1 đặc trưng trung bình | 102608.508 |

Kết luận : Mô hình 1 cho kết quả tốt nhất, mô hình này có công thức hồi quy là :

$$Salary = 1457.603 * 10percentage + 657.067 * 12percentage + 1087.317 * CollegeGPA + 241.263 * Quant + 25130.476 * Domain + 120.169 * ComputerProgramming - 152.243 * ComputerScience - 97310.3932$$

# PHẦN V. Tham khảo


[1] Lab04 – OLS Linear Regression, Ngô Đình Hy, Nguyễn Văn Quang Huy

[2] Linear Regression – Machine Learning cơ bản
https://machinelearningcoban.com/2016/12/28/linearregression/

[3] Everything you need to Know about Linear Regression
https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/

[4] Linear Regression in Machine learning – GeeksforGeeks
https://www.geeksforgeeks.org/ml-linear-regression/

[5] seaborn.heatmap — seaborn 0.12.2 documentation - PyData |
https://seaborn.pydata.org/generated/seaborn.heatmap.html

[6] linear-regression · GitHub Topics
https://github.com/topics/linear-regression

[7] The Main Ideas of Fitting a Line to Data (The Main Ideas of Least Squares and Linear Regression.)
https://www.youtube.com/watch?v=PaFPbb66DxQ&ab_channel=StatQuestwithJoshStarmer

[8] Curve Fitting in Python (2022)
https://www.youtube.com/watch?v=peBOquJ3fDo&ab_channel=Mr.PSolver

[9] Data fitting và phương pháp OLS - Trần Lê Hùng Phi
https://www.phitran.asia/kh%C3%B3a-h%E1%BB%8Dc/khoa-h%E1%BB%8Dc-d%E1%BB%AF-li%E1%BB%87u/data-fitting-v%C3%A0-ph%C6%B0%C6%A1ng-ph%C3%A1p-ols

[10] Curve Fitting using Linear and Nonlinear Regression
https://statisticsbyjim.com/regression/curve-fitting-linear-nonlinear-regression/