

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO TIẾN ĐỘ ĐỒ ÁN
Toán ứng dụng và thống kê cho công nghệ thông tin
21CLC04

Đồ án

LINEAR REGRESSION



Giáo viên hướng dẫn

Nguyễn Văn Quang Huy
Ngô Đình Hy
Nguyễn Đình Thúc

Thành viên

Lý Nhật Hào - 21127041

NIÊN KHOÁ 2022 - 2023

LỜI CẢM ƠN

Để hoàn thành được bài báo cáo này, em đã nhận được sự giúp đỡ rất nhiều từ phía thầy cô giảng viên, trợ giảng và bạn bè. Nay em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến giảng viên môn Toán ứng dụng và thống kê cho Công Nghệ Thông Tin lớp 21CLC4, Khoa Công nghệ thông tin :

- Giảng viên **Nguyễn Đình Thúc**
- Giảng viên **Ngô Đình Hy**
- Giảng viên **Nguyễn Văn Quang Huy**

Các thầy đã đồng hành, đã luôn quan tâm, hướng dẫn và truyền đạt, cung cấp kiến thức, tài liệu và các thủ thuật cần thiết để em có thể hoàn thành đồ án !

Trong quá trình thực hiện đồ án không thể tránh khỏi những thiếu sót. Em rất mong nhận được nhiều ý kiến đóng góp từ các giảng viên và bạn bè để đồ án ngày càng hoàn thiện hơn!



THÀNH VIÊN NHÓM BÁO CÁO

X

Lý Nhật Hào

Xin chân thành cảm ơn!



MỤC LỤC

LỜI CẢM ƠN	1
CHƯƠNG I: GIỚI THIỆU ĐỒ ÁN & MỨC ĐỘ HOÀN THIỆN	4
1.1. Thông tin cá nhân sinh viên thực hiện đồ án:	4
1.2. Tổng quát yêu cầu đồ án và mức độ hoàn thiện:	4
CHƯƠNG II: SƠ LƯỢC VỀ LINEAR REGRESSION	7
CHƯƠNG III: GIỚI THIỆU TỔNG QUAN VỀ ĐỒ ÁN LINEAR REGRESSION	8
3.1. Môi trường thực hiện đồ án	8
3.2. Các thư viện đã sử dụng trong đồ án trên	8
3.3. Các hàm đã sử dụng trong đồ án trên	8
• Hàm của thư viện pandas	8
• Hàm của thư viện numpy	9
• Hàm của thư viện sklearn	9
CHƯƠNG IV: CÀI ĐẶT THUẬT TOÁN VÀ HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH	11
4.1. Cài đặt và giải thích thuật toán	11
• Yêu cầu 1A	12
• Yêu cầu 1B	13
• Yêu cầu 1C	15
• Yêu cầu 1D	16
4.2. Hướng Dẫn Sử Dụng, Kết Quả Và Nhận Xét Đáng Chú Ý	19
• Yêu cầu 1A	19
• Yêu cầu 1B	21
• Yêu cầu 1C	23
• Yêu cầu 1D	25
CHƯƠNG V: TỔNG NHẬN XÉT TRÊN 16 MÔ HÌNH	31
CHƯƠNG VI: TÀI LIỆU THAM KHẢO	32
• Tham Khảo cho tổng quan đồ án	32
• Tham Khảo cho câu 1A, 1B, 1C	32
• Tham Khảo cho câu 1D	32

CHƯƠNG I: GIỚI THIỆU ĐỒ ÁN & MỨC ĐỘ HOÀN THIỆN

1.1. Thông tin cá nhân sinh viên thực hiện đồ án:

Họ & Tên	Lý Nhật Hào
MSSV	21127041
EMAIL	lnhao21@clc.fitus.edu.vn

1.2. Tổng quát yêu cầu đồ án và mức độ hoàn thiện:

Bài nộp đã hoàn thành 100% yêu cầu đồ án 3 đưa ra

Nội Dung đồ án 3 như sau:

- Mục tiêu của đồ án là tìm hiểu các yếu tố quyết định mức lương và việc làm của các kỹ sư ngay sau khi tốt nghiệp. Các yếu tố như điểm số ở các cấp/trường đại học, kỹ năng của ứng viên, sự liên kết giữa trường đại học và các khu công nghiệp/công ty công nghệ, bằng cấp của sinh viên và điều kiện thị trường cho các ngành công nghiệp cụ thể sẽ ảnh hưởng đến điều này.
- Bộ dữ liệu được sử dụng trong đồ án này thu thập tại Ấn Độ, nơi có hơn 6000 cơ sở đào tạo kỹ thuật công nghệ với khoảng 2,9 triệu sinh viên đang học tập. Mỗi năm, trung bình có 1,5 triệu sinh viên tốt nghiệp chuyên ngành Công nghệ/Kỹ thuật, tuy nhiên do thiếu kỹ năng cần thiết, ít hơn 20% trong số họ có việc làm phù hợp với chuyên môn của mình. Bộ dữ liệu này không chỉ giúp xây dựng công cụ dự đoán mức lương mà còn cung cấp thông tin về các yếu tố ảnh hưởng đến mức lương và chức danh công việc trên thị trường lao động. Sinh viên sẽ được khám phá những thông tin này trong phạm vi đồ án.

Trong đồ án này, dữ liệu trên đã được thực hiện các bước tiền xử lý sau:

- Loại bỏ các cột có giá trị là chuỗi: `DOB`, `10board`, `12board`, `Specialization`, `CollegeState`
- Loại bỏ các cột liên quan đến định danh và năm: `ID`, `CollegeID`, `CollegeCityID`, `12graduation`, `GraduationYear`

Trong đồ án này, sinh viên được yêu cầu thực hiện:

- **Xây dựng mô hình** *dự đoán mức lương của kỹ sư* **sử dụng mô hình hồi quy tuyến tính** (7 điểm)
- **Yêu cầu 1a:** Sử dụng 11 đặc trưng đầu tiên đề bài cung cấp bao gồm: `Gender`, `10percentage`, `12percentage`, `CollegeTier`, `Degree`, `collegeGPA`, `CollegeCityTier`, `English`, `Logical`, `Quant`, `Domain` (2 điểm)
 - 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 **1 kết quả trên tập kiểm tra (`test.csv`)** cho mô hình vừa huấn luyện được
- **Yêu cầu 1b:** Phân tích ảnh hưởng của **đặc trưng tính cách** dựa trên điểm các bài kiểm tra của AMCAT (1 điểm)
 - 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)
 - Thể hiện công thức cho mô hình hồi quy theo đặc trưng tốt nhất (tính $\$y$ theo đặc trưng tốt nhất tìm được)
 - Báo cáo *****1 kết quả trên tập kiểm tra (`test.csv`)***** cho mô hình với đặc trưng tốt nhất tìm được
- **Yêu cầu 1c:** Phân tích ảnh hưởng của **đặc trưng ngoại ngữ, lô-gic, định lượng** đến mức lương của các kỹ sư dựa trên điểm các bài kiểm tra của AMCAT (1 điểm)
 - Thử nghiệm trên các đặc trưng gồm: `English`, `Logical`, `Quant`
 - Yêu cầu sử dụng *****k-fold Cross Validation***** (**k** tối thiểu là 5) để tìm ra đặc trưng tốt nhất
 - Báo cáo 3 kết quả tương ứng cho 3 mô hình từ k-fold Cross Validation (lấy trung bình)

- Thể hiện công thức cho mô hình hồi quy theo đặc trưng tốt nhất (tính R^2 theo đặc trưng tốt nhất tìm được)
- Báo cáo **1 kết quả trên tập kiểm tra ('test.csv')** cho mô hình với đặc trưng tốt nhất tìm được
- **Yêu cầu 1d:** Sinh viên tự xây dựng mô hình, tìm mô hình cho kết quả tốt nhất (3 điểm)
 - 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...)
 - **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
 - Yêu cầu **sử dụng phương pháp k-fold Cross Validation** (**k** tối thiểu là 5) để tìm ra mô hình tốt nhất trong 'm' mô hình mà sinh viên xây dựng
 - Báo cáo **'m' kết quả tương ứng cho 'm' mô hình** từ k-fold Cross Validation (lấy trung bình)

CHƯƠNG II: SƠ LƯỢC VỀ LINEAR REGRESSION

Hồi quy tuyến tính (Linear Regression) là một phương pháp thống kê phổ biến được sử dụng để dự đoán một biến phụ thuộc liên tục (còn được gọi là biến mục tiêu) dựa trên một hoặc nhiều biến độc lập (còn được gọi là đặc trưng hoặc biến dự đoán). Phương pháp này giả định một mối quan hệ tuyến tính giữa các biến độc lập và biến phụ thuộc.

Mục tiêu của hồi quy tuyến tính là tìm đường thẳng "best-fit" sao cho sai số giữa các giá trị dự đoán và giá trị thực tế của biến phụ thuộc là nhỏ nhất. Đường thẳng "best-fit" này được xác định bằng cách ước lượng các hệ số (trọng số) cho mỗi biến độc lập.

Công thức cho một mô hình hồi quy tuyến tính đơn giản với một biến độc lập có thể được viết dưới dạng:

$$Y = \beta_0 + \beta_1 * X$$

Trong đó:

- Y là biến phụ thuộc (biến mục tiêu) cần dự đoán,
- X là biến độc lập (đặc trưng) được sử dụng để dự đoán Y,
- β_0 là hệ số chặn của đường thẳng hồi quy (intercept),
- β_1 là hệ số góc của đường thẳng hồi quy (slope).

Hồi quy tuyến tính có nhiều ý nghĩa và ứng dụng quan trọng trong các lĩnh vực khác nhau.

- **Phân tích mối quan hệ:** Hồi quy tuyến tính giúp chúng ta hiểu được sự tương quan giữa các biến độc lập và biến phụ thuộc.
- **Dự đoán:** Hồi quy tuyến tính giúp chúng ta dự đoán doanh thu dựa trên số lượng sản phẩm bán ra và chi phí quảng cáo.
- **Kiểm tra giả thuyết:** Chúng ta có thể kiểm tra xem có mối quan hệ tuyến tính giữa tuổi và thu nhập không.
- **Dự báo và kế hoạch:** Hồi quy tuyến tính có thể giúp dự báo và lập kế hoạch trong các lĩnh vực như kinh tế, tài chính, tiếp thị, y tế và hơn thế nữa.
- **Phân tích ảnh hưởng:** Hồi quy tuyến tính cung cấp thông tin về mức độ ảnh hưởng của từng biến độc lập đến biến phụ thuộc. Điều này giúp chúng ta hiểu rõ hơn về tác động của các yếu tố khác nhau đến kết quả mong muốn.

CHƯƠNG III: GIỚI THIỆU TỔNG QUAN VỀ ĐỒ ÁN LINEAR REGRESSION

3.1. Môi trường thực hiện đồ án:

- Toàn bộ đồ án được lập trình trên một file .ipynb duy nhất bao gồm source code và chú thích tổng quan của từng yêu cầu được đề ra.
- Tất cả chú thích, kết quả, phân tích, nhận xét và bàn luận chi tiết của đồ án trên sẽ được triển khai chi tiết trong file .PDF

3.2. Các thư viện đã sử dụng trong đồ án trên:

STT	THƯ VIỆN	CÔNG DỤNG
1	pandas	Để xử lý nhận-trả dữ liệu theo dạng data frame
2	numpy	Thư viện quen thuộc dùng cho ma trận, array,... trong đồ án trên là để append
3	sklearn	Để có thể sử dụng thuật toán linear regression, k fold cross validation và để tính MAE

3.3. Các hàm đã sử dụng trong đồ án trên:

- **Hàm của thư viện pandas:**
 - **pandas.read_csv():** cho phép ta đọc dữ liệu từ một file csv.
 - **input:** tên file.
 - **output:** dữ liệu kiểu dataframe.
 - **pandas.iloc[]:** cho phép ta truy xuất dữ liệu từ của một dataframe theo chỉ mục.
 - **input:** index.
 - **output:** dữ liệu kiểu data frame (dòng hoặc cột ta cần truy xuất).

- **pandas.DataFrame.sample(frac = 1):** Giúp ta xáo trộn các dòng dữ liệu của một dataframe.
 - **input:** frac = 1
 - **output:** data frame đã được xáo trộn
- **pandas.DataFrame(data):** cho phép ta tạo một data frame từ một dữ liệu có sẵn, có thể là list, object,..
 - **input:** data
 - **output:** dữ liệu kiểu data frame in ra các kết quả đẹp hơn.
- **pandas.corr():** cho phép ta tính toán hệ số tương quan giữa các cặp đặc trưng khác nhau trong một dataframe
 - **input:** có thể truyền vào method, có 3 thuật toán tính hệ số tương quan đó là pearson , kendall và spearman .
 - **output:** một ma trận với các phần tử thể hiện hệ số tương quan giữa đặc trưng, hệ số càng cao chứng tỏ 2 đặc trưng đang xét tương quan càng mạnh. Được sử dụng để tính mối tương quan giữa các đặc trưng khác với đặc trưng Salary
- **Hàm của thư viện numpy:**
 - **numpy.mean():** hàm dùng để tính average của một mảng
 - **input:** một mảng các tham số
 - **output:** giá trị trung bình.
 - Hàm được sử dụng để tính trung bình của MAE.
 - **numpy.absolute():** hàm dùng để lấy trị tuyệt đối.
 - **input:** một mảng tham số
 - **output:** giá trị tuyệt đối của tham số đó
 - Hàm được sử dụng để lấy trị tuyệt đối trong MAE.
- **Hàm của thư viện sklearn:**
 - **sklearn.metrics.mean_absolute_error():** là một phương thức cho phép tính MAE.
 - **input:** một mảng các dữ liệu đích và một mảng các dữ liệu dự đoán.
 - **output:** giá trị MAE.

- **LinearRegression.predict(X_test)**: là một method thuộc class LinearRegression giúp ta dự đoán kết quả dựa trên mô hình hồi quy tuyến tính
 - **Input**: một mảng hay ma trận gồm dữ liệu của các đặc trưng cần dự đoán.
 - **Output**: mảng các giá trị y_{pred} mà dự đoán dựa trên dữ liệu truyền vào và mô hình hồi quy tuyến tính đã có.
- **LinearRegression.coef_**: đây là một thuộc tính (property) thuộc class LinearRegression. Phương thức trả ra mảng các hệ số ước lượng cho các mô hình hồi quy tuyến tính của chúng ta.
 - **Input**: không có
 - **Output**: mảng các hệ số tương ứng với đặc trưng
- **sklearn.linear_model.LinearRegression()**: là một class cho phép ta thực hiện thuật toán linear regression để xây dựng mô hình hồi quy tuyến tính.
 - **Input**: hàm có thể truyền vào nhiều tham số tuy nhiên trong đề án này chúng ta không truyền tham số nào vào cả.
 - **Output**: một đối tượng thuộc class LinearRegression.

CHƯƠNG IV: CÀI ĐẶT THUẬT TOÁN VÀ HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH

4.1. Cài đặt và giải thích thuật toán:

- **Đọc dữ liệu từ file .csv:**
 - Đọc dữ liệu bằng pandas
 - Lấy các đặc trưng X và giá trị mục tiêu y cho các tập huấn luyện (train) và kiểm tra (test)

```
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')

x_train = train.iloc[:, :11]
y_train = train.iloc[:, -1]

x_test = test.iloc[:, :11]
y_test = test.iloc[:, -1]

# Sinh viên có thể sử dụng các khác nếu cần
```

- **Cài đặt hàm MAE:**
 - Dùng thư viện sklearn để tính MAE

```
def calculate_mae(test, y_test_pred): # TÍNH MAE
    mae = mean_absolute_error(test.iloc[:, -1], y_test_pred)
    return mae
```

Yêu cầu 1A: Sử dụng toàn bộ 11 đặc trưng đầu tiên:

- Đọc dữ liệu huấn luyện từ tệp 'train.csv' và dữ liệu kiểm tra từ tệp 'test.csv' bằng cách sử dụng hàm `pd.read_csv()`. Dữ liệu huấn luyện được lưu trong biến `training_1`, và dữ liệu kiểm tra được lưu trong biến `testing_1`.
- Tạo một mô hình hồi quy tuyến tính bằng cách sử dụng lớp `LinearRegression()` từ thư viện `scikit-learn`.
- Sử dụng mô hình đã được huấn luyện để dự đoán kết quả trên dữ liệu kiểm tra. Giá trị dự đoán được lưu trong biến `y_test_preds`.
- Tạo một Data Frame có tên **prediction** để so sánh kết quả dự đoán với giá trị thực tế trên dữ liệu kiểm tra. DataFrame prediction được in ra màn hình bằng hàm `print()`, số làm tròn đến 3 chữ số thập phân bằng phương thức `round(3)`.
- Tương tự với **coefficient**. Ta cũng tạo một Data Frame có tên **coefficient** để so sánh kết quả dự đoán với giá trị thực tế trên dữ liệu kiểm tra. DataFrame prediction được in ra màn hình bằng hàm `print()`, số làm tròn đến 3 chữ số thập phân bằng phương thức `round(3)`.

```
# Cài đặt các hàm cần thiết ở đây
# Phần code cho yêu cầu 1a

#đọc 2 cái file
testing_1 = pd.read_csv('test.csv')
training_1 = pd.read_csv('train.csv')

#dùng linearregression và lấy 11 tính cách đầu của bộ dữ liệu
lr1 = LinearRegression(fit_intercept=True).fit(training_1.iloc[:, :11], training_1.iloc[:, -1])

y_test_pred = lr1.predict(testing_1.iloc[:, :11])
predic = pd.DataFrame({'data from file': testing_1.iloc[:, -1], 'Predicted': y_test_pred})
print(predic.round(3))
coef = pd.DataFrame({'Feature': training_1.iloc[:, :11].columns, 'Coefficient': lr1.coef_})
print(coef.round(3))
print()
print("Intercept:", round(lr1.intercept_,3))
```

Gọi hàm MAE (tự cài đặt hoặc từ thư viện) trên tập kiểm tra

```
mae = calculate_mae(test, y_test_pred)
print("MAE: ", mae.round(3))
```

✓ 0.0s

Yêu cầu 1B: Phân tích ảnh hưởng của các đặc trưng tính cách gồm: `conscientiousness`, `agreeableness`, `extraversion`, `nueroticism`, `openess_to_experience` :

Các bước làm gần như tương tự ở câu A, tuy nhiên bộ dữ liệu đặc trưng lúc này là 5 tính cách kể cuối của bộ dữ liệu

- Đọc dữ liệu huấn luyện từ tệp 'train.csv' và lưu vào biến `train_2`. Sau đó, dữ liệu huấn luyện được trộn ngẫu nhiên bằng phương thức `sample(frac=1)`.
- Khởi tạo một đối tượng **LinearRegression** trong biến `reg`.
- Chọn 5 đặc trưng kể đặc trưng cuối cùng trong số 34 cột của dữ liệu huấn luyện và lưu vào danh sách `fea_title`.
- Với mỗi đặc trưng `fea_title`, thực hiện cross-validation bằng cách sử dụng hàm `cross_val_score` để tính toán điểm số sử dụng độ đo `'neg_mean_absolute_error'` trên mô hình `reg` và các đặc trưng tương ứng. Kết quả **cross-validation** được lưu trong danh sách `scores`.
- Tính trung bình giá trị tuyệt đối của các điểm số cross-validation và lưu vào danh sách `MAE`.
- Tìm chỉ số của đặc trưng có giá trị MAE nhỏ nhất bằng hàm `np.argmin(MAE)` và lưu vào biến `best_index`.
- In ra màn hình đặc trưng có giá trị MAE nhỏ nhất và giá trị MAE tương ứng, số làm tròn đến 3 chữ số phần thập phân.

```
training_2 = pd.read_csv('train.csv')
training_2 = training_2.sample(frac=1)
y_df = training_2.iloc[:, -1]

reg = LinearRegression()

fea_title = training_2.columns[-6:-1] # Lấy 5 tính cách kể cuối của 34 cột
feature = [training_2[name].values.reshape(-1, 1) for name in fea_title]

cross = []
scores = []

for i in range(len(feature)):
    scores.append(cross_val_score(reg, feature[i], y_df, scoring="neg_mean_absolute_error", cv=5))

MAE = []
for i in range(len(scores)):
    MAE.append(np.absolute(scores[i]).mean())
    cross.append([fea_title[i], round(MAE[i], 3)]) # Làm tròn số làm 3 chữ số phần thập phân

test = pd.DataFrame(cross, columns=['Feature', 'Average MAE'])
print(test)
print()

best_index = np.argmin(MAE)
print("Minimum of average MAE is:", fea_title[best_index], '-', round(MAE[best_index], 3))
```

- Huấn luyện lại mô hình **best_personality_feature_model** với đặc trưng tốt nhất trên toàn bộ tập huấn luyện và đưa ra dự đoán trên tập kiểm tra để đánh giá hiệu suất của mô hình. Nó cũng cung cấp thông tin về hệ số tương ứng của các đặc trưng trong mô hình:

```
testing_2f = pd.read_csv('test.csv')

training_2f = pd.read_csv('train.csv')

X_k_test = testing_2f[fea_title[best_index]].values.reshape(-1, 1)
y_k_test = testing_2f.iloc[:, -1]

X_k_train = training_2f[fea_title[best_index]].values.reshape(-1, 1)
y_k_train = training_2f.iloc[:, -1]

reg2 = LinearRegression(fit_intercept=True).fit(X_k_train, y_k_train)

y_k_test_pred = reg2.predict(X_k_test)

k_predic = pd.DataFrame({'Data in file': y_k_test, 'Predicted': y_k_test_pred})
print(k_predic.round(3))

k_coef = pd.DataFrame({'Feature': fea_title[best_index], 'Coefficient': reg2.coef_})
print(k_coef.round(3))
print()
print("Intercept:", round(reg2.intercept_,3))
```

- Gọi hàm MAE đã được tự cài đặt ở phía trên cho tập kiểm tra với mô hình **best_personality_feature_model**

```
print("MAE: ", round(mean_absolute_error(y_k_test, y_k_test_pred),3))
```

✓ 0.0s

Yêu cầu 1C: Xây dựng mô hình sử dụng duy nhất 1 đặc trưng English, Logical, Quant, tìm mô hình cho kết quả tốt nhất.

- Các bước làm gần như tương tự ở câu B, tuy nhiên lúc này là các tính cách ở 3 vị trí từ 7 đến 9 của bộ dữ liệu

```
training_2 = pd.read_csv('train.csv')
training_2 = training_2.sample(frac=1)
y_df = training_2.iloc[:, -1]
reg = LinearRegression()
fea_title = training_2.columns[7:10] # Lấy từ 7 đến 9
feature = [training_2[name].values.reshape(-1, 1) for name in fea_title]
cross = []
scores = []
for i in range(len(feature)):
    scores.append(cross_val_score(reg, feature[i], y_df, scoring="neg_mean_absolute_error", cv=5))
# Calculate MAE and append to an array
MAE = []
for i in range(len(scores)):
    MAE.append(np.absolute(scores[i]).mean())
    cross.append([fea_title[i], round(MAE[i], 3)]) # Làm tròn số làm 3 chữ số phần thập phân
test = pd.DataFrame(cross, columns=['Feature', 'Average MAE'])
print(test)
print()
best_index = np.argmin(MAE)
print("Min average MAE is:", fea_title[best_index], "(", round(MAE[best_index], 3), ")") # Làm tròn
```

- Huấn luyện lại mô hình **best_personality_feature_model** với đặc trưng tốt nhất trên toàn bộ tập huấn luyện :

```
training_2f = pd.read_csv('train.csv')
testing_2f = pd.read_csv('test.csv')

X_k_train = training_2f[fea_title[best_index]].values.reshape(-1, 1)
y_k_train = training_2f.iloc[:, -1]

X_k_test = testing_2f[fea_title[best_index]].values.reshape(-1, 1)
y_k_test = testing_2f.iloc[:, -1]

reg2 = LinearRegression(fit_intercept=True).fit(X_k_train, y_k_train)
y_k_test_pred = reg2.predict(X_k_test)

k_predic = pd.DataFrame({'Data in file': y_k_test, 'Predicted': y_k_test_pred})
print(k_predic.round(3))

k_coef = pd.DataFrame({'Feature': fea_title[best_index], 'Coefficient': reg2.coef_})
print(k_coef.round(3))
print()
print("Intercept:", round(reg2.intercept_, 3))
```

- Gọi hàm MAE (tự cài đặt hoặc từ thư viện) trên tập kiểm tra với mô hình **best_personality_feature_model**

```
print("MAE: ", round(mean_absolute_error(y_k_test, y_k_test_pred), 3))
✓ 0.0s
```


Yêu cầu 1D: Sinh viên tự xây dựng mô hình, tìm mô hình cho kết quả tốt nhất.

- Đầu tiên, ta sử dụng hàm `corr()` của pandas để tính hệ số tương quan giữa từng cặp đặc trưng. Ta tính hệ số tương quan của từng cặp đôi một trong 23 đặc trưng để xem độ tương quan như thế nào.

```
# Phần code cho yêu cầu 1d
# Tìm ra mô hình tốt nhất (tự thiết kế bởi sinh viên)
# In ra các kết quả cross-validation như yêu cầu
# Trình bày các phần tìm ra mô hình
train_corr = pd.read_csv('train.csv')
correlations = train_corr.corr()
print(correlations)
```

✓ 0.0s

- Tiếp theo, ta lấy ra hệ số tương quan giữa các đặc trưng với đặc trưng `Salary`

```
correlations_2 = train_corr.corr()['Salary'].drop('Salary')
print(correlations_2)
```

✓ 0.0s

- Thử nghiệm, so sánh các mô hình, ta sẽ lựa chọn giữa 3 mô hình, đó là 23 đặc tính trên toàn tập dữ liệu, 16 đặc tính có độ tương quan tốt nhất, cuối cùng là tập có 10 đặc tính có độ tương quan tốt.
- Giữa 3 mô hình trên, ta sẽ lựa chọn ra một mô hình tốt nhất và thành công nhất để huấn luyện nó trên mô hình tốt nhất trên tập dữ liệu được cung cấp. Từ đó tối ưu Average MAE của mô hình mà ta sắp chọn.

```
train_3 = pd.read_csv('train.csv')
train_3 = train_3.sample(frac=1)

model = ['Use 23 features', 'Use 16 features', 'Use 10 features']
MAE = []
```

- Sử dụng 5-Fold Cross Validation để test với tất cả 23 đặc trưng

```
# 1st model
train_test_1 = train_3.copy()
feature = ['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']
X_test_1 = train_test_1[feature]
y_test_1 = train_test_1.iloc[:, -1]

rg_test_1 = LinearRegression()
scores_test_1 = cross_val_score(rg_test_1, X_test_1, y_test_1, cv=5,
                                scoring='neg_mean_absolute_error')
print("Average MAE: ", round(np.absolute(scores_test_1).mean(),3))
MAAE.append(np.absolute(scores_test_1).mean())
```

✓ 0.0s

- Sử dụng 5-Fold Cross Validation để test với 16 đặc trưng có độ tương quan so với đặc trưng Salary tốt nhất trong bộ dữ liệu

```
train_test_2 = train_3.copy()
feature = ['Gender', '10percentage', '12percentage', 'collegeGPA',
           'English', 'Logical', 'Quant', 'Domain', 'ComputerProgramming',
           'ComputerScience', 'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg',
           'conscientiousness', 'agreeableness', 'nueroticism']
X_test_2 = train_test_2[feature]
y_test_2 = train_test_2.iloc[:, -1]

rg_test_2 = LinearRegression()
scores_test_2 = cross_val_score(rg_test_2, X_test_2, y_test_2, cv=5,
                                scoring='neg_mean_absolute_error')
print("Average MAE: ", round(np.absolute(scores_test_2).mean(),3))
MAAE.append(np.absolute(scores_test_2).mean())
```

- Sử dụng 5-Fold Cross Validation để test với 10 đặc trưng có độ tương quan so với đặc trưng Salary tốt nhất trong bộ dữ liệu

```
train_test_3 = train_3.copy()
feature = ['Gender', '10percentage', '12percentage', 'CollegeTier',
           'collegeGPA', 'English', 'Logical', 'Quant',
           'Domain', 'ComputerProgramming', 'ComputerScience']
X_test_3 = train_test_3[feature]
y_test_3 = train_test_3.iloc[:, -1]

rg_test_3 = LinearRegression()
scores_test_3 = cross_val_score(rg_test_3, X_test_3, y_test_3, cv=5,
                                scoring='neg_mean_absolute_error')
print("Average MAE: ", round(np.absolute(scores_test_3).mean(),3))
MAAE.append(np.absolute(scores_test_3).mean())
```

- So sánh giữa 3 mô hình vừa tạo để chọn ra mô hình tối ưu nhất

```
Average_MAE = pd.DataFrame({'Model': model, 'Average MAE': AMAE})
print(round(Average_MAE,3))
```

- Sau khi đã chọn được mô hình tối ưu nhất trong 3 mô hình đã cài đặt - ở đây là mô hình có 16 đặc trưng, ta bắt đầu dùng mô hình đó để huấn luyện lại mô hình my_best_model trên toàn bộ tập dữ liệu..

```
train3 = pd.read_csv('train.csv')
test3 = pd.read_csv('test.csv')

feature = ['Gender', '10percentage', '12percentage', 'CollegeTier', 'collegeGPA',
           'English', 'Logical', 'Quant', 'Domain', 'ComputerProgramming',
           'ComputerScience', 'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg',
           'conscientiousness', 'agreeableness', 'nueroticism']
X_train_best_model = train3[feature].copy()
X_test_best_model = test3[feature].copy()

reg3 = LinearRegression(fit_intercept=True).fit(X_train_best_model, train.iloc[:, -1])

y_test_best_model_pred = reg3.predict(X_test_best_model)
predic_best_model = pd.DataFrame({'Actual': test3.iloc[:, -1],
                                  'Predicted': y_test_best_model_pred})
print(predic_best_model)





coef_best_model = pd.DataFrame({'Coefficient': X_test_best_model.columns,
                                'Predicted': reg3.coef_})
print(round(coef_best_model,3))
print()
print("Intercept:", round(reg3.intercept_,3))
```

- Gọi hàm MAE (tự cài đặt hoặc từ thư viện) trên tập kiểm tra với mô hình my_best_model

```
print("MAE: ", round(mean_absolute_error(test3.iloc[:, -1], y_test_best_model_pred),3))
```

4.2. Hướng Dẫn Sử Dụng, Kết Quả Và Nhận Xét Đáng Chú Ý:

Bước 1: Đặt file test.csv và train.csv vào cùng một thư mục với file code .ipynb để chương trình có thể đọc được dữ liệu từ tệp file dữ liệu.

 21127041.ipynb	8/18/2023 9:08 PM	Jupyter Source File	45 KB
 project03.ipynb	8/8/2023 5:51 PM	Jupyter Source File	28 KB
 test.csv	8/7/2023 9:28 AM	Microsoft Excel Co...	89 KB
 train.csv	8/7/2023 9:28 AM	Microsoft Excel Co...	264 KB

Bước 2: Chạy chương trình

Bước 3: xem kết quả của các yêu cầu theo trình tự và phân tích sau đây:

○ Yêu cầu 1A:

```

...      data from file      Predicted
0          280000      194207.932
1          520000      340719.587
2          150000      325416.849
3          180000      273672.748
4          300000      298369.367
..          ...          ...
745         330000      283138.706
746         450000      381114.180
747         180000      297490.123
748          90000      242061.854
749         360000      328403.653

[750 rows x 2 columns]
      Feature      Coefficient
0          Gender      -23183.330
1      10percentage         702.767
2      12percentage        1259.019
3      CollegeTier     -99570.608
4          Degree        18369.962
5      collegeGPA         1297.532
6  CollegeCityTier     -8836.727
7          English         141.760
8          Logical         145.742
9           Quant         114.643
10         Domain        34955.750

Intercept: 49248.09

```

- Với kết quả đạt được ở bên trên và dựa vào hệ số của nó, ta có được công thức sau:

Công thức hồi quy (phần trọng số làm tròn đến 3 chữ số thập phân)

```
Salary= 49248.09 + (-23183.330)xGender + 702.767x10percentage + 1259.019x12percentage  
+(-99570.608)xCollegeTier +18369.962xDegree +1297.532xcollegeGPA  
+(-8836.727)xCollegeCityTier + 141.760xEnglish  
+145.742xLogical + 114.643xQuant + 34955.750xDomain
```

- Kết quả của tập đặc trưng câu 1A: **105052.530**

MAE: 105052.53

■ Nhận xét:

- Với ưu thế của mô hình trên khi được huấn luyện từ 11 đặc trưng cho nên khi chạy code trên tập test thì kết quả cho ra rất tốt, MAE khá nhỏ.
- Tuy nhiên, với yêu cầu trên, ta chỉ mới kiểm tra trên một mô hình nên ta chưa thấy được sự khác biệt nào rõ ràng giữa mô hình này và các mô hình khác.

○ **Yêu cầu 1B:**

■ **Mô hình một đặc trưng và mô hình cho kết quả tốt nhất:**

	Feature	Average MAE
0	conscientiousness	124207.763
1	agreeableness	123553.713
2	extraversion	123850.797
3	nueroticism	123495.299
4	openess_to_experience	123818.223

Minimum of average MAE is: nueroticism - 123495.299

...	Data in file	Predicted
0	280000	316828.694
1	520000	296119.311
2	150000	297530.805
3	180000	294185.517
4	300000	290122.466
..
745	330000	328713.438
746	450000	303649.413
747	180000	326681.913
748	90000	322476.271
749	360000	299884.362

[750 rows x 2 columns]

	Feature	Coefficient
0	nueroticism	-16021.494

Intercept: 304647.553

- Với kết quả đạt được ở bên trên và dựa vào hệ số của nó, ta có được công thức sau:

Công thức hồi quy (phần trọng số làm tròn đến 3 chữ số thập phân)

$$\text{Salary} = -16021.494 * \text{nueroticism} + 304647.553$$

- Kết quả của tập đặc trưng câu 1B: **119361.917**

MAE: 119361.917

- **Nhận xét:**

- Dễ dàng thấy được việc sử dụng đặc trưng này cho ra kết quả tốt hơn nhiều so với các kết quả khác sau khi tính toán được các kết quả Average MAE.
- Tuy nhiên khi so sánh với Average MAE của mô hình sử dụng 10 đặc trưng ở câu 1A thì Average MAE của mô hình này vẫn còn khá cao.
- Sử dụng một đặc trưng duy nhất thì cho dù đặc trưng đó có là đặc trưng tốt nhất trong tất cả đi chăng nữa vẫn không đủ để đánh giá dữ liệu, do đó số lượng đặc trưng cần được bổ sung thêm.

○ Yêu cầu 1C:

■ Mô hình một đặc trưng và mô hình cho kết quả tốt nhất:

```

Feature    Average MAE
0 English   120822.210
1 Logical   120084.037
2 Quant     117302.695

```

Min average MAE is: Quant (117302.695)

```

Data in file Predicted
0      280000 197063.009
1      520000 359358.093
2      150000 337226.945
3      180000 270833.502
4      300000 302185.961
..      ...      ...
745     330000 302185.961
746     450000 326161.371
747     180000 245013.829
748      90000 322472.847
749     360000 311407.273

```

[750 rows x 2 columns]

```

Feature Coefficient
0 Quant      368.852

```

Intercept: 117759.729

- Với kết quả đạt được ở bên trên và dựa vào hệ số của nó, ta có được công thức sau:

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

$$\text{Salary} = 368.852 * \text{Quant} + 117759.729$$

- Kết quả của tập đặc trưng câu 1B: **108814.06**

- **Nhận xét:**

- Để tạo ra được một mô hình tốt hơn, đầu tiên ta phải tìm cách để chọn các đặc trưng sao cho phù hợp. Sau đó ta sẽ sử dụng những đặc trưng đó để xây dựng mô hình.
- Chính vì vậy với Yêu cầu của câu 1C ta đã xây dựng một mô hình sử dụng số đặc trưng ít hơn 11 mà lại cho kết quả tốt. Hay thậm chí mô hình trên còn tốt hơn cả mô hình 5 đặc trưng ở câu 1B.

○ Yêu cầu 1D:

- Sinh viên tự xây dựng mô hình, tìm mô hình kết quả tốt nhất:
- Đầu tiên ở yêu cầu trên, ta tính hệ số tương quan giữa của từng cặp đôi một trong **23 đặc trưng** để xem thử độ tương quan như thế nào

	Gender	10percentage	12percentage	CollegeTier	
Gender	1.000000	0.165208	0.131372	0.028943	\
10percentage	0.165208	1.000000	0.644518	-0.135469	
12percentage	0.131372	0.644518	1.000000	-0.092920	
CollegeTier	0.028943	-0.135469	-0.092920	1.000000	
Degree	-0.007080	-0.255081	-0.227924	-0.014755	
collegeGPA	0.153008	0.311057	0.340745	-0.091280	
CollegeCityTier	0.044938	0.106144	0.120529	-0.093067	
English	-0.020830	0.335863	0.193790	-0.182399	
Logical	-0.000189	0.309735	0.240047	-0.189068	
Quant	-0.104069	0.326948	0.316088	-0.240973	
Domain	0.001947	0.079001	0.069002	-0.037476	
ComputerProgramming	0.021369	0.041760	0.076158	-0.047969	
ElectronicsAndSemicon	-0.019304	0.088068	0.123903	-0.026150	
ComputerScience	-0.031236	-0.024749	-0.050739	-0.020929	
MechanicalEngg	-0.083987	0.060161	0.042870	-0.015850	
ElectricalEngg	-0.024408	0.068455	0.085593	0.012326	
TelecomEngg	0.028421	0.060164	0.057063	-0.010562	
CivilEngg	-0.013109	0.009898	0.000678	0.007080	
conscientiousness	0.075360	0.050050	0.044066	0.036467	
agreeableness	0.087639	0.115473	0.093829	-0.036447	
extraversion	0.006984	-0.022176	-0.031926	-0.006246	
neuroticism	0.011918	-0.121777	-0.083520	0.038786	
openness_to_experience	0.084511	0.015292	-0.007928	-0.019414	
Salary	-0.036183	0.155174	0.149531	-0.174824	

	Degree	collegeGPA	CollegeCityTier	English	
Gender	-0.007080	0.153008	0.044938	-0.020830	\
10percentage	-0.255081	0.311057	0.106144	0.335863	
12percentage	-0.227924	0.340745	0.120529	0.193790	
CollegeTier	-0.014755	-0.091280	-0.093067	-0.182399	
Degree	1.000000	0.080067	-0.001511	-0.145472	
collegeGPA	0.080067	1.000000	0.030261	0.099539	
CollegeCityTier	-0.001511	0.030261	1.000000	0.028303	
English	-0.145472	0.099539	0.028303	1.000000	
Logical	-0.098722	0.200165	-0.006065	0.431918	
Quant	-0.137183	0.221253	-0.019965	0.368248	
Domain	0.010125	0.083268	0.013744	0.106778	
ComputerProgramming	0.110226	0.139499	0.043879	0.121789	
ElectronicsAndSemicon	-0.133786	0.026659	0.047274	-0.000923	
ComputerScience	-0.015129	-0.013137	-0.006909	0.086161	
MechanicalEngg	-0.061272	-0.033850	-0.046042	-0.008649	
ElectricalEngg	-0.057312	0.055134	0.016535	0.029723	
TelecomEngg	-0.079172	-0.000657	0.077937	-0.018019	
CivilEngg	-0.014273	-0.035964	-0.034213	-0.028461	
conscientiousness	0.003147	0.048044	0.012093	0.024610	
agreeableness	-0.033432	0.053377	0.009984	0.174948	
extraversion	0.009707	-0.054623	0.008211	0.003313	
neuroticism	0.021054	-0.074752	0.033827	-0.147243	
openness_to_experience	0.014351	0.005078	0.015314	0.061630	
Salary	-0.017602	0.122469	0.004575	0.169293	

	Logical	Quant	...	MechanicalEngg
Gender	-0.000189	-0.104069	...	-0.083987
10percentage	0.309735	0.326948	...	0.060161
12percentage	0.240047	0.316088	...	0.042870
CollegeTier	-0.189068	-0.240973	...	-0.015850
Degree	-0.098722	-0.137183	...	-0.061272
collegeGPA	0.200165	0.221253	...	-0.033850
CollegeCityTier	-0.006065	-0.019965	...	-0.046042
English	0.431918	0.368248	...	-0.008649
Logical	1.000000	0.502061	...	-0.006461
Quant	0.502061	1.000000	...	0.002708
Domain	0.202380	0.224860	...	0.053179
ComputerProgramming	0.191525	0.149635	...	-0.299781
ElectronicsAndSemicon	-0.005432	0.109907	...	-0.101312
ComputerScience	0.053090	-0.016059	...	-0.128077
MechanicalEngg	-0.006461	0.002708	...	1.000000
ElectricalEngg	0.007168	0.026210	...	-0.046272
TelecomEngg	-0.028632	0.026092	...	-0.064349
CivilEngg	-0.038780	-0.030404	...	0.104176
conscientiousness	0.007225	-0.018171	...	0.006010
agreeableness	0.130697	0.077396	...	-0.003763
extraversion	-0.028864	-0.065295	...	-0.014642
neroticism	-0.193569	-0.145108	...	0.048024
openess_to_experience	0.018907	-0.009126	...	-0.005465
Salary	0.188416	0.205358	...	0.028854

	ElectricalEngg	TelecomEngg	CivilEngg
Gender	-0.024408	0.028421	-0.013109
10percentage	0.068455	0.060164	0.009898
12percentage	0.085593	0.057063	0.000678
CollegeTier	0.012326	-0.010562	0.007080
Degree	-0.057312	-0.079172	-0.014273
collegeGPA	0.055134	-0.000657	-0.035964
CollegeCityTier	0.016535	0.077937	-0.034213
English	0.029723	-0.018019	-0.028461
Logical	0.007168	-0.028632	-0.038780
Quant	0.026210	0.026092	-0.030404
Domain	0.042712	0.010813	0.012037
ComputerProgramming	-0.129097	-0.263624	-0.090141
ElectronicsAndSemicon	0.031313	0.382743	0.024320
ComputerScience	-0.087721	-0.157011	-0.050341
MechanicalEngg	-0.046272	-0.064349	0.104176
ElectricalEngg	1.000000	-0.041153	-0.018187
TelecomEngg	-0.041153	1.000000	-0.029350
CivilEngg	-0.018187	-0.029350	1.000000
conscientiousness	0.021002	-0.019231	-0.016809
agreeableness	-0.036537	-0.042123	-0.028675
extraversion	-0.012484	-0.063950	-0.031945
neroticism	-0.033828	0.046946	0.043489
openess_to_experience	-0.032293	-0.011011	-0.030626
Salary	-0.041217	-0.040415	0.016150

	conscientiousness	agreeableness	extraversion
Gender	0.075360	0.087639	0.006984
10percentage	0.050050	0.115473	-0.022176
12percentage	0.044066	0.093829	-0.031926
CollegeTier	0.036467	-0.036447	-0.006246
Degree	0.003147	-0.033432	0.009707
collegeGPA	0.048044	0.053377	-0.054623
CollegeCityTier	0.012093	0.009984	0.008211
English	0.024610	0.174948	0.003313
Logical	0.007225	0.130697	-0.028864
Quant	-0.018171	0.077396	-0.065295
Domain	-0.055640	0.040502	-0.030174
ComputerProgramming	-0.004566	0.071997	0.053795
ElectronicsAndSemicon	-0.022056	-0.023630	-0.062544
ComputerScience	0.059120	0.025604	0.076109
MechanicalEngg	0.006010	-0.003763	-0.014642
ElectricalEngg	0.021002	-0.036537	-0.012484
TelecomEngg	-0.019231	-0.042123	-0.063950
CivilEngg	-0.016809	-0.028675	-0.031945
conscientiousness	1.000000	0.492499	0.354234
agreeableness	0.492499	1.000000	0.480572
extraversion	0.354234	0.480572	1.000000
nueroticism	-0.308616	-0.197990	-0.085622
openess_to_experience	0.415984	0.605813	0.468572
Salary	-0.057699	0.068623	-0.002661

	nueroticism	openess_to_experience	Salary
Gender	0.011918	0.084511	-0.036183
10percentage	-0.121777	0.015292	0.155174
12percentage	-0.083520	-0.007928	0.149531
CollegeTier	0.038786	-0.019414	-0.174824
Degree	0.021054	0.014351	-0.017602
collegeGPA	-0.074752	0.005078	0.122469
CollegeCityTier	0.033827	0.015314	0.004575
English	-0.147243	0.061630	0.169293
Logical	-0.193569	0.018907	0.188416
Quant	-0.145108	-0.009126	0.205358
Domain	-0.041850	-0.015629	0.122022
ComputerProgramming	-0.104625	0.059650	0.125866
ElectronicsAndSemicon	0.003814	-0.029846	-0.009292
ComputerScience	-0.109579	0.037363	-0.095507
MechanicalEngg	0.048024	-0.005465	0.028854
ElectricalEngg	-0.033828	-0.032293	-0.041217
TelecomEngg	0.046946	-0.011011	-0.040415
CivilEngg	0.043489	-0.030626	0.016150
conscientiousness	-0.308616	0.415984	-0.057699
agreeableness	-0.197990	0.605813	0.068623
extraversion	-0.085622	0.468572	-0.002661
nueroticism	1.000000	-0.051393	-0.073401
openess_to_experience	-0.051393	1.000000	-0.007814
Salary	-0.073401	-0.007814	1.000000

- Tiếp theo, ta lấy ra hệ số tương quan giữa các đặc trưng với đặc trưng `Salary`

```
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
nueroticism     -0.073401
openess_to_experience -0.007814
Name: Salary, dtype: float64
```

- Dựa vào các chỉ số tương quan trên ta sẽ lựa chọn cho mình một mô hình mà ta cho rằng đó là mô hình tối ưu nhất cho tập dữ liệu và tốt hơn cả khi ta dùng tất cả 23 đặc trưng của toàn bộ tập dữ liệu.
- Tuy nhiên, để có sự so sánh, và có tính minh bạch, ta sẽ xây dựng 3 mô hình mà ta cho rằng nó sẽ mang lại kết quả tối ưu, bao gồm:
 - Mô hình bao gồm tất cả **23 đặc trưng**.
 - Mô hình bao gồm **16 đặc trưng** phù hợp nhất
 - Mô hình chỉ bao gồm **10 đặc trưng** phù hợp nhất

- Sau đó, ta sẽ so sánh Average MAE của 3 mô hình đã xây dựng và lựa chọn mô hình tốt nhất. Kết quả được minh họa như sau:

...	Model	Average MAE
0	Use 23 features	111706.971
1	Use 16 features	111047.383
2	Use 10 features	111577.122

- Dễ thấy, mô hình 16 đặc trưng là mô hình tối ưu nhất trong số 3 mô hình mà ta lựa chọn, và tốt hơn cả mô hình chứa tất cả đặc trưng vốn đã rất hoàn hảo.
- Ta chọn mô hình 16 đặc trưng để tiếp tục huấn luyện cho mô hình tốt nhất.

[750 rows x 2 columns]		
	Coefficient	Predicted
0	Gender	-24909.837
1	10percentage	702.390
2	12percentage	1039.326
3	CollegeTier	-97057.418
4	collegeGPA	1326.148
5	English	138.267
6	Logical	108.238
7	Quant	91.541
8	Domain	25549.124
9	ComputerProgramming	90.459
10	ComputerScience	-172.478
11	MechanicalEngg	54.516
12	ElectricalEngg	-145.755
13	TelecomEngg	-89.379
14	conscientiousness	-20081.590
15	agreeableness	14665.138
16	neroticism	-11465.937
Intercept:		99137.525

- Với kết quả đạt được ở bên trên và dựa vào hệ số của nó, ta có được công thức sau:

Công thức hồi quy (phần trọng số làm tròn đến 3 chữ số thập phân)

```
Salary= 99137.525+ Gender* -24909.837 + 10percentage* 702.390 + 12percentage * 1039.326  
+CollegeTier * -97057.418 +collegeGPA * 1326.148 + English138.267 +Logical 108.238  
+Quant * 91.541 +Domain * 25549.124 +ComputerProgramming * 90.459+ ComputerScience * -172.478  
+MechanicalEngg * 54.516 + ElectricalEngg* -145.755 +TelecomEngg* -89.379 + conscientiousness* -20081.590  
+agreeableness * 14665.138 + nueroticism * -11465.937
```

- **Nhận xét:**

- Trước hết ta phải kể đến mô hình sử dụng tất cả 23 đặc trưng, thật sự là mô hình sử dụng 23 đặc trưng quá tốt, thật khó để tìm được một mô hình nào vượt qua nó.
 - Một điều may mắn đó là kết quả đặc trưng tối ưu lại thuộc về mô hình dùng 16 đặc trưng để đánh giá mô hình, càng ít đặc trưng thì càng khó để phản ánh đúng mức lương.
 - Tuy nhiên đến kết quả cho ra ở mô hình 16 đặc trưng lại tốt nhất và thành công ngoài dự kiến khi nó vượt qua mô hình chứa tất cả mọi đặc trưng.
 - Mô hình chỉ có 10 đặc trưng cũng cực kỳ tối ưu khi chỉ để thua mô hình 16 đặc trưng rất sát sao và vẫn tốt hơn nhiều so với mô hình chứa tất cả 23 đặc trưng
- Kết quả của tập đặc trưng câu 1D sau khi ta đã huấn luyện lại mô hình tối ưu nhất chứa 16 đặc trưng trên tập dữ liệu cung cấp: **103241.002**
 - Đây là kết quả có MAE thấp nhất trong tất cả mô hình mà ta có trong đề án trên, thể hiện rằng mô hình mà chúng ta đã chọn thật sự có tác dụng tốt và thành công.

CHƯƠNG V: TỔNG NHẬN XÉT TRÊN 16 MÔ HÌNH

1. Với những lý giải và nhận xét, các câu hỏi cũng như lý do lại chọn mô hình này, tại sao phải chọn đặc tính kia cũng như những số liệu so sánh cụ thể đã được phân tích tại phần nhận xét trong phần Kết Quả Đồ Án, cụ thể là ở dưới mỗi kết quả và mỗi mô hình sau khi test xong.
2. Việc sử dụng 11 đặc trưng của Yêu Cầu 1A cho kết quả rất tốt. Vì mô hình này bao gồm hầu hết các yếu tố ảnh hưởng đến mức lương trung bình.
3. Đối với Yêu Cầu Câu 1B và 1C đó là sử dụng 1 đặc trưng riêng lẻ để tạo mô hình thì cũng như con dao hai lưỡi vậy, có mặt tốt và cũng sẽ có mặt xấu đi kèm, mặt lợi đó là mô hình sử dụng 1 và chỉ 1 đặc trưng để xây dựng nên tuy nhiên mặt bất lợi của việc này đó là có những đặc trưng ảnh hưởng cực kỳ vô cùng rất là ít đến cả tập dữ liệu nên việc đó khiến cho mô hình chúng ta không được tốt như mong muốn nữa. Chính vì những lý do đó, việc chỉ sử dụng 1 đặc trưng là không đủ.
4. Việc sử dụng tất cả các đặc trưng mang lại một tập dữ liệu hoàn hảo hỗ trợ cho nhau nhưng sau khi xây dựng mô hình thì đôi khi nó còn cho ra kết quả tệ hơn so với mô hình chỉ có ít đặc trưng nhưng được lựa chọn phù hợp nhờ vào các bảng số liệu như mô hình 16 đặc trưng và mô hình 10 đặc trưng.
5. Cuối cùng, việc loại bỏ một số đặc trưng để xây dựng nên một mô hình có ít đặc trưng là một hướng đi khá tốt tuy nhiên nó lại mang nhiều rủi ro vì đôi lúc lược bỏ quá nhiều dẫn đến không đủ đặc trưng để đánh giá. Do đó phương pháp lược bỏ thật sự là mạo hiểm, tuy nhiên nếu ta lược bỏ theo một số tiêu chuẩn và giữ lại những đặc trưng theo một số tiêu chuẩn nhất định thì kết quả mang lại sẽ khiến cho mô hình của chúng ta tối ưu hơn.



CHƯƠNG VI: TÀI LIỆU THAM KHẢO

- **Tham Khảo cho tổng quan đồ án:**

1. [Everything you need to Know about Linear Regression](#)
2. [Bài 3: Linear Regression](#)

- **Tham Khảo cho câu 1A, 1B, 1C:**

3. [Linear Regression - Hồi quy tuyến tính trong Machine Learning](#)
4. [Thực hiện Linear Regression với Scikit-learn | TopDev](#)
5. [numpy.mean\(\) trong Python](#)

- **Tham Khảo cho câu 1D:**

6. [Tìm nghiệm của bài toán hồi quy tuyến tính bằng tối ưu tham số](#)
7. [pandas.DataFrame.sample — pandas 2.0.3 documentation](#)
8. [Python List append\(\) Method](#)