ĐẠI HỌC QUỐC GIA – TP. HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN KHOA CÔNG NGHỆ THÔNG TIN

Báo Cáo Đồ án 3: Linear Regression

Môn học: Toán ứng dụng và thống kê cho công nghệ thông tin - MTH00057

Giảng viên hướng dẫn:

- Nguyễn Đình Thúc
- Nguyễn Văn Quang Huy
- Ngô Đình Hy

Tp. Hồ Chí Minh, tháng 8/2023

1. Thông tin sinh viên

Họ và tên: Cao Nguyễn Khánh
MSSV: 21127627
Lớp : 21CLC04
Số trang: 15

2. Các thư viện được sửa dụng

Thư viện	Lý do
import pandas as pd	- Để đọc cũng như in ra dữ liệu - Lọc dữ liệu cần thiết, tìm hệ số tương quan,
from sklearn.model_selection import cross_val_score	- Để thực hiện K-fold Cross Validation để đánh giá hiệu suất của mô hình. Phương thức này cho phép chia dữ liệu thành các phần nhỏ để huấn luyện.
from sklearn.linear_model import LinearRegression	- Để thực hiện huấn luyện mô hình. - LinearRegression sử dụng để tạo mô hình hồi quy tuyến tính.
from sklearn.metrics import mean_absolute_error	- Được sử dụng để tính độ đo MAE, để đánh giá sự sai lệch giữa dự đoán của mô hình và giá trị thực tế.

3. Các hàm được sửa dụng

3.1 Hàm in công thức cho mô hình

```python
def CTMoHinh(model, selected_features):
    # In ra công thức của mô hình hồi quy
    coefficients = model.coef_

    formula = "Salary = {:.2f} + ".format(model.intercept_)  # Hệ số chặn
    for i, coef in enumerate(coefficients):
        feature_name = selected_features[i]
        formula += "{:.3f}*{} + ".format(coef, feature_name)
    formula = formula[:-2]  # Loại bỏ dấu + cuối cùng
    return formula
```

Mô tả:

- Hàm này sẽ nhận vào 1 mô hình hồi quy tuyến tính đã được huấn luyện (**model**), danh sách các đặc trưng(**selected_features**).
- Trích xuất các hệ số (coeficients) của mô hình hồi quy tuyến tính bằng cách sử dụng model.coef_.

- Khởi tạo chuỗi formula để xây dựng công thức của mô hình. Đầu tiên, thêm hệ số chặn (intercept) vào chuỗi formula sử dụng model.intercept_.
- Duyệt qua từng đặc trưng và hệ số tương ứng trong coefficients. Đối với mỗi đặc trưng, thêm vào chuỗi formula một phần tử có định dạng: {hệ số}*{tên đặc trưng} +.
- Loại bỏ dấu + ở cuối cùng của chuỗi formula để hoàn thiện công thức.
- Trả về chuỗi formula đã xây dựng.

3.2 Hàm tìm đặc trưng tốt nhất

```
def Find_Best_Feature(features):
    best_feature = None
    lowest_mae = float('inf')  # Khởi tạo với giá trị dương vô cùng để so sánh

    # Thực hiện K-fold Cross Validation và tìm đặc trưng tốt nhất
    for feature in features:
        X = X_train[[feature]]
        y = y_train

        model = LinearRegression()

        # Tính MAE bằng K-fold Cross Validation
        mae_scores = -cross_val_score(model, X, y, cv=5, scoring='neg_mean_absolute_error')
        avg_mae = mae_scores.mean()
        print(feature , '| MAE =' , avg_mae)

        if avg_mae < lowest_mae:
            lowest_mae = avg_mae
            best_feature = feature

    return best_feature
```

Mô tả:

- Hàm sẽ nhận vào một danh sách các đặc trưng(**features**)
- Khởi tạo biến best_feature và lowest_mae với giá trị ban đầu lần lượt là None và dương vô cùng (float('inf')) để lưu trữ đặc trưng tốt nhất và MAE thấp nhất.
- Duyệt qua từng đặc trưng trong danh sách features:
- Tạo ma trận X chỉ chứa cột của đặc trưng hiện tại từ tập dữ liệu huấn luyện X_train.
- Gán nhãn y bằng tập dữ liệu nhãn huấn luyện y_train.
- Khởi tạo mô hình hồi quy tuyến tính
- Sử dụng phương pháp K-fold Cross Validation (cv=5) để đánh giá mô hình trên từng phần tập dữ liệu và tính các giá trị MAE.
- Tính giá trị trung bình của các giá trị MAE để có avg_mae.
- In ra thông tin về đặc trưng hiện tại và giá trị MAE trung bình.
- So sánh avg_mae với lowest_mae:
- Nếu avg_mae nhỏ hơn lowest_mae, cập nhật lowest_mae và gán best_feature bằng đặc trưng hiện tại.
- Trả về best_feature - đặc trưng được cho là tốt nhất sau khi đã tìm qua tất cả các đặc trưng.

3.3 Hàm tìm mô hình tốt nhất

```
def Find_Best_Model(model, modes):
    best_model = None
    lowest_mae = float('inf')  # Khởi tạo với giá trị dương vô cùng để so sánh
    i=0
    # Thực hiện K-fold Cross Validation và tìm mô hình tốt nhất từ các modes
    for modenew in modes:
        X = X_train[modenew]
        y = y_train

        # Tính MAE bằng K-fold Cross Validation
        mae_scores = -cross_val_score(model, X, y, cv=5, scoring='neg_mean_absolute_error')
        avg_mae = mae_scores.mean()

        i = i+1
        print("Mode", i, ' | Avg MAE =' , avg_mae)

        if avg_mae < lowest_mae:
            lowest_mae = avg_mae
            best_model = modenew

    return best_model
```

Mô tả:

- Hàm sẽ nhận vào một **models** là một ma trận 2 chiều, 1 chiều thể hiện thứ tự các mô hình, chiều còn lại là danh sách các đặc trưng được sử dụng cho mô hình đó.
- Khởi tạo biến best_model và lowest_mae với giá trị ban đầu lần lượt là None và dương vô cùng (float('inf')) để lưu trữ mô hình tốt nhất và MAE thấp nhất.
- Khởi tạo biến i với giá trị 0 để đếm số lượng mô hình đã thử.
- Duyệt qua từng mô hình trong danh sách models:
- Tạo ma trận X chỉ chứa cột của các đặc trưng tương ứng với mô hình hiện tại từ tập dữ liệu huấn luyện X_train.
- Gán nhãn y bằng tập dữ liệu nhãn huấn luyện y_train.
- Khởi tạo mô hình hồi quy tuyến tính
- Sử dụng phương pháp K-fold Cross Validation (cv=5) để đánh giá mô hình trên từng phần tập dữ liệu và tính các giá trị MAE.
- Tính giá trị trung bình của các giá trị MAE để có avg_mae.
- In ra thông tin về mô hình hiện tại và giá trị MAE trung bình.
- Tăng biến đếm i lên 1 và in ra thông tin về mô hình hiện tại và giá trị trung bình của MAE.
- So sánh avg_mae với lowest_mae:
- Nếu avg_mae nhỏ hơn lowest_mae, cập nhật lowest_mae và gán best_model bằng mô hình hiện tại.
- Trả về best_model - mô hình được cho là tốt nhất sau khi đã so sánh qua tất cả các mô hình.

4. Báo cáo và nhận xét kết quả từ toàn bộ các mô hình xây dựng được

4.1 Xây dựng mô hình với 11 đặc trưng đầu tiên

- Để chuẩn bị dữ liệu : ta lấy 11 đặc trưng đầu tiên của x_train và x_test ra

```
# Phần code cho yêu cầu 1a
selected_features = ['Gender', '10percentage', '12percentage', 'CollegeTier', 'Degree',
                     'collegeGPA', 'CollegeCityTier', 'English', 'Logical', 'Quant', 'Domain']

# Tạo DataFrame mới X_train11 với chỉ 11 đặc trưng đầu tiên
X_train11 = X_train[selected_features]
X_test11 = X_test[selected_features]
```

- Sau đó huấn luyện cho mô hình

```
# Xây dựng và huấn luyện mô hình hồi quy tuyến tính trên toàn bộ dữ liệu
model1a = LinearRegression()
model1a.fit(X_train11, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

- Ta có công thức cho mô hình vừa huấn luyện được

```
print(CTMoHinh(model1a, selected_features))
```

```
Salary = 49248.09 + -23183.330*Gender + 702.767*10percentage + 1259.019*12percentage + -99570.608*Col
legeTier + 18369.962*Degree + 1297.532*collegeGPA + -8836.727*CollegeCityTier + 141.760*English + 14
5.742*Logical + 114.643*Quant + 34955.750*Domain
```

- Kiểm tra với x_test11

```
# Dự đoán giá trị mức lương trên toàn bộ dữ liệu
y_pred1a = model1a.predict(X_test11)

y_pred1a
```

```
array([194207.93160491, 340719.58717406, 325416.84861397, 273672.74799754,
       298369.36726444, 352015.74694404, 234779.7619092 , 261961.54605652,
       266112.40481144, 372849.64049772, 301599.21406001, 279946.54486441,
       228413.01407561, 305770.45016173, 403117.84794154, 311514.75295673,
       255771.6406102 , 249511.47496086, 282050.50784367, 342932.68002501,
       252047.78684692, 331783.8543578 , 392592.81848188, 441556.83514415,
       237549.86845078, 406144.5863516 , 325347.97152288, 340287.86083972,
       339589.96608669, 359034.7906052 , 337830.75631687, 388921.75950567,
       307399.94344879, 507243.30753988, 303046.98974608, 288503.42897022,
       488343.13883449, 307550.03410941, 373338.98609063, 281018.74597146,
       499929.31989583, 360613.09095212, 266862.5060087 , 386429.00791345,
       236977.45662334, 391749.09033547, 312501.27630642, 255610.43006437,
       474422.70383532, 300598.05757003, 293416.22841846, 369686.76439215,
       250401.61675547, 297319.646986  , 317699.13103174, 280942.35692969,
```

- Nhận xét:
  - Mô hình được huấn luyện trên 11 đặc trưng nên cho ra kết quả khá tốt trên tập test, dẫn đến MAE khá nhỏ, nên mô hình có hiệu suất tốt.

```
# Gọi hàm MAE (tự cài đặt hoặc từ thư viện) trên tập kiểm tra
# Đánh giá hiệu suất mô hình trên toàn bộ dữ liệu
mae1a = mean_absolute_error(y_test, y_pred1a)
print("Mean Absolute Error:", mae1a)
```

```
Mean Absolute Error: 105052.52978823172
```

- MAE được đánh giá khác nhỏ vì:
  Khoảng giá trị của y_test tương đối lớn

```
print(y_test.min())
print(y_test.max())
```

```
45000
2600000
```

Khoảng giá trị từ 45,000 đến 2,600,000 là khá rộng. Với phạm vi này, giá trị MAE là 105052 chiếm ,khoảng 4.1% của phạm vi tổng giá trị (2,600,000 - 45,000).
4.1% < 5% thì điều này có thể được coi là một hiệu suất tốt.

4.2 Phân tích ảnh hưởng của đặc trưng tính cách dựa trên điểm các bài kiểm tra của AMCAT.

- Để chuẩn bị dữ liệu : ta lấy danh sách tên 5 đặc trưng tính cách : conscientiousness, agreeableness, extraversion, nueroticism, openess_to_experience

```
# Các đặc trưng tính cách
personality_features = ['conscientiousness', 'agreeableness', 'extraversion',
                        'nueroticism', 'openess_to_experience']
```

- Dùng hàm **Find_Best_Feature**(pesonality_features) để tìm ra đặc trưng tốt nhất với k-fold Cross Validation (k tối thiểu là 5), In ra các giá trị MAE của từng mô hình 1 đặc trưng tính cách

```
# Phần code cho yêu cầu 1b
# Tìm ra đặc trưng tốt nhất
# In ra các kết quả cross-validation như yêu cầu

best_feature = Find_Best_Feature(personality_features)

print('\nĐặc trưng tốt nhất là: ', best_feature)
```

```
conscientiousness | MAE = 124444.48696126812
agreeableness | MAE = 123813.28712231014
extraversion | MAE = 123914.50490042963
nueroticism | MAE = 123738.52541404287
openess_to_experience | MAE = 124119.48107191359

Đặc trưng tốt nhất là:  nueroticism
```

Ta thấy đặc trưng tốt nhất là nueroticism, có chỉ số MAE nhỏ nhất.

- Sau đó huấn luyện cho mô hình với đặc trưng tốt nhất được tìm thấy (nueroticism)

```
# Huấn luyện lại mô hình best_personality_feature_model với đặc trưng tốt nhất trên toàn bộ tập huấn lu

# Khởi tạo mô hình hồi quy tuyến tính
best_personality_feature_model = LinearRegression()

# Chọn đặc trưng tốt nhất và huấn luyện mô hình trên toàn bộ dữ liệu
X_best = X_train[[best_feature]]
y_best = y_train
best_personality_feature_model.fit(X_best, y_best)
```

```
▾ LinearRegression
LinearRegression()
```

- Ta có công thức cho mô hình vừa huấn luyện được

```
best_feature1b = [best_feature]

print(CTMoHinh(best_personality_feature_model, best_feature1b))
```

```
Salary = 304647.55 + -16021.494*nueroticism
```

- Kiểm tra với x_test1b

```
X_test1b = X_test[[best_feature]]
y_pred1b = best_personality_feature_model.predict(X_test1b)
y_pred1b
```

```
array([316828.69418332, 296119.31147609, 297530.8050677 , 294185.51719111,
       290122.46639849, 313063.16252799, 273875.06967607, 303649.41349713,
       279174.97977939, 302310.01662701, 334806.41222121, 301766.88799187,
       300278.49123069, 311181.11766754, 298246.96583438, 296217.04258743,
       342930.9116571 , 338869.46301384, 307414.46450765, 306371.46527027,
       311651.66893639, 316828.69418332, 328713.43818163, 298001.83698135,
       277936.51831933, 298246.96583438, 340899.38626079, 316828.69418332,
       299883.8818418 , 298246.96583438, 306371.46527027, 328713.43818163,
       322476.2706991 , 312464.43930985, 316527.49010248, 277936.51831933,
       326681.91278532, 308402.99066658, 314946.16867806, 312371.51464661,
       326681.91278532, 307414.46450765, 298001.83698135, 312464.43930985,
```

- Nhận xét:
  - Mô hình được huấn luyện trên đặc trưng tốt nhất trong 5 đặc trưng nên cho ra kết quả khá tốt trên tập test, dẫn đến MAE khá nhỏ, nên mô hình có hiệu suất tốt.

```
# Gọi hàm MAE (tự cài đặt hoặc từ thư viện) trên tập kiểm tra với mô hình best_personality_feature_mode
mae1b = mean_absolute_error(y_test, y_pred1b)

print("Mean Absolute Error:", mae1b)
```

```
Mean Absolute Error: 119361.91739987816
```

  - MAE được đánh giá khác nhỏ vì:
    Khoảng giá trị của y_test tương đối lớn

```
print(y_test.min())
print(y_test.max())
```

```
45000
2600000
```

    Khoảng giá trị từ 45,000 đến 2,600,000 là khá rộng. Với phạm vi này, giá trị MAE là 105052 chiếm ,khoảng 4.6% của phạm vi tổng giá trị (2,600,000 - 45,000).
    4.6% < 5% thì điều này có thể được coi là một hiệu suất tốt.

- Giả thuyết cho đặc trưng tính cách có mô hình đạt kết quả tốt nhất :
  - Trong 5 đặc trưng được sửa dụng k-fold Cross Validation. Thì MAE của 5 đặc trưng không có sự cách biệt quá lớn.

```
conscientiousness | MAE = 124444.48696126812
agreeableness | MAE = 123813.28712231014
extraversion | MAE = 123914.50490042963
nueroticism | MAE = 123738.52541404287
openess_to_experience | MAE = 124119.48107191359

Đặc trưng tốt nhất là:  nueroticism
```

- Đặc trưng nueroticism là đặc trưng tốt nhất do MAE của nó nhỏ hơn 1 chút so với 4 đặc trưng còn lại.
- Còn mô hình hồi quy tuyến tính cho đặc trưng tốt nhất cho MAE tương đối tốt là vì có thể mô hình này chỉ có 1 đặc trưng, là một mô hình đơn giản nên chỉ số MAE mới nhỏ.

```
# Gọi hàm MAE (tự cài đặt hoặc từ thư viện) trên tập kiểm tra với mô hình best_personality_feature_mode
mae1b = mean_absolute_error(y_test, y_pred1b)

print("Mean Absolute Error:", mae1b)
```

```
Mean Absolute Error: 119361.91739987816
```

4.3 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.

- Để chuẩn bị dữ liệu : ta lấy danh sách tên 3 đặc trưng: ngoại ngữ, lô-gic, định lượng

```
# Các đặc trưng đặc trưng ngoại ngữ, lô-gic, định lượng
skill_features = ['English', 'Logical', 'Quant']
```

- Dùng hàm **Find_Best_Feature**(pesonality_features) để tìm ra đặc trưng tốt nhất với k-fold Cross Validation (k tối thiểu là 5), In ra các giá trị MAE của từng mô hình 1 đặc trưng tính cách

```
# Phần code cho yêu cầu 1c
# Tìm ra đặc trưng tốt nhất
# In ra các kết quả cross-validation như yêu cầu

best_feature = Find_Best_Feature(skill_features)

print('\nĐặc trưng tốt nhất là: ', best_feature)
```

```
English | MAE = 120963.06945762748
Logical | MAE = 120037.71893286356
Quant | MAE = 117461.46396286949

Đặc trưng tốt nhất là:  Quant
```

Ta thấy đặc trưng tốt nhất là Quant, có chỉ số MAE nhỏ nhất.

- Sau đó huấn luyện cho mô hình với đặc trưng tốt nhất được tìm thấy (Quant)

```
# Huấn luyện lại mô hình best_skill_feature_model với đặc trưng tốt nhất trên toàn bộ tập huấn luyện

# Khởi tạo mô hình hồi quy tuyến tính
best_skill_feature_model = LinearRegression()

# Chọn đặc trưng tốt nhất và huấn luyện mô hình trên toàn bộ dữ liệu
X_best = X_train[[best_feature]]
y_best = y_train

best_skill_feature_model.fit(X_best, y_best)
```

```
▾ LinearRegression
LinearRegression()
```

- Ta có công thức cho mô hình vừa huấn luyện được

```
best_feature1c = [best_feature]
print(CTMoHinh(best_skill_feature_model, best_feature1c))
```

```
Salary = 117759.73 + 368.852*Quant
```

- Kiểm tra với x_test1c

```
X_test1c = X_test[[best_feature]]
y_pred1c = best_skill_feature_model.predict(X_test1c)

y_pred1c
```

```
array([197063.00903697, 359358.09312464, 337226.9452945 , 270833.50180409,
       302185.96123012, 300341.69891094, 283743.33803834, 278210.5510808 ,
       315095.79746437, 357513.83080546, 289276.12499587, 285587.60035752,
       296653.17427259, 348292.51920957, 383333.50327396, 333538.42065615,
       289276.12499587, 292964.64963423, 311407.27282601, 351981.04384793,
       267144.97716574, 300341.69891094, 313251.53514519, 329849.89601779,
       248702.35397395, 309563.01050683, 305874.48586848, 333538.42065615,
       318415.46963889, 300341.69891094, 305874.48586848, 355669.56848629,
       292964.64963423, 363046.617763  , 302185.96123012, 302185.96123012,
       394399.07718903, 296653.17427259, 307718.74818765, 322103.99427724,
       335382.68297533, 348292.51920957, 281899.07571916, 313251.53514519,
       211817.10759039, 337226.9452945 , 368579.40472053, 281899.07571916,
       368579.40472053, 355669.56848629, 313251.53514519, 370423.66703971,
       256079.40325067, 318784.32210272, 359358.09312464, 324317.10906026,
```

- Nhận xét:
  - Mô hình được huấn luyện trên đặc trưng tốt nhất trong 5 đặc trưng nên cho ra kết quả khá tốt trên tập test, dẫn đến MAE khá nhỏ, nên mô hình có hiệu suất tốt.

```
# Gọi hàm MAE (tự cài đặt hoặc từ thư viện) trên tập kiểm tra với mô hình best_skill_feature_model
mse1c = mean_absolute_error(y_test, y_pred1c)

print("Mean Absolute Error:", mse1c)
```

```
Mean Absolute Error: 108814.05968837194
```

  - MAE được đánh giá khác nhỏ vì:
    Khoảng giá trị của y_test tương đối lớn

```
print(y_test.min())
print(y_test.max())
```

```
45000
2600000
```

    Khoảng giá trị từ 45,000 đến 2,600,000 là khá rộng. Với phạm vi này, giá trị MAE là 108814 chiếm ,khoảng 4.2% của phạm vi tổng giá trị (2,600,000 - 45,000).
    4.2% < 5% thì điều này có thể được coi là một hiệu suất tốt.
- Giả thuyết cho đặc trưng tính cách có mô hình đạt kết quả tốt nhất :

- Trong 3 đặc trưng được sửa dụng k-fold Cross Validation. Thì MAE của 3 đặc trưng không có sự cách biệt quá lớn.

```
English | MAE = 120963.06945762748
Logical | MAE = 120037.71893286356
Quant | MAE = 117461.46396286949

Đặc trưng tốt nhất là:  Quant
```

- Đặc trung Quant là đặc trưng tốt nhất do MAE của nó nhỏ hơn 1 chút so với 2 đặc trưng còn lại.
- Còn mô hình hồi quy tuyến tính cho đặc trưng tốt nhất cho MAE tương đối tốt là vì có thế mô hình này chỉ có 1 đặc trưng, là một mô hình đơn giản nên chỉ số MAE mới nhỏ.

```
# Gọi hàm MAE (tự cài đặt hoặc từ thư viện) trên tập kiểm tra với mô hình best_skill_feature_model
mse1c = mean_absolute_error(y_test, y_pred1c)

print("Mean Absolute Error:", mse1c)
```

```
Mean Absolute Error: 108814.05968837194
```

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

- Lý do để em quyết định chọn các đặc trưng cho một mô hình là dựa vào hệ số tương quan của một đặc trưng xem nó có tương quan mạnh hay yếu với đặc trưng **Salary** hoặc các đặc trưng có tương quan với nhau như thế nào**.**
  - Để tìm hệ số tương quan của các đặc trưng với nhau :

```
# Trình bày các phần tìm ra mô hình
# Tính hệ số tương quan giữa các thuộc tính
correlations = train.corr()
correlations
```

- Kết quả :

	Gender	10percentage	12percentage	CollegeTier	Degree	collegeGPA	CollegeCityTier	English	Logical	Quant	...	MechanicalEngg
Gender	1.000000	0.165208	0.131372	0.028943	-0.007080	0.153008	0.044938	-0.020830	-0.000189	-0.104069	...	-0.083987
10percentage	0.165208	1.000000	0.644518	-0.135469	-0.255081	0.311057	0.106144	0.335863	0.309735	0.326948	...	0.060161
12percentage	0.131372	0.644518	1.000000	-0.092920	-0.227924	0.340745	0.120529	0.193790	0.240047	0.316088	...	0.04287(
CollegeTier	0.028943	-0.135469	-0.092920	1.000000	-0.014755	-0.091280	-0.093067	-0.182399	-0.189068	-0.240973	...	-0.01585(
Degree	-0.007080	-0.255081	-0.227924	-0.014755	1.000000	0.080067	-0.001511	-0.145472	-0.098722	-0.137183	...	-0.061272
collegeGPA	0.153008	0.311057	0.340745	-0.091280	0.080067	1.000000	0.030261	0.099539	0.200165	0.221253	...	-0.03385(
CollegeCityTier	0.044938	0.106144	0.120529	-0.093067	-0.001511	0.030261	1.000000	0.028303	-0.006065	-0.019965	...	-0.046042
English	-0.020830	0.335863	0.193790	-0.182399	-0.145472	0.099539	0.028303	1.000000	0.431918	0.368248	...	-0.00864(
Logical	-0.000189	0.309735	0.240047	-0.189068	-0.098722	0.200165	-0.006065	0.431918	1.000000	0.502061	...	-0.006461
Quant	-0.104069	0.326948	0.316088	-0.240973	-0.137183	0.221253	-0.019965	0.368248	0.502061	1.000000	...	0.00270(
Domain	0.001947	0.079001	0.069002	-0.037476	0.010125	0.083268	0.013744	0.106778	0.202380	0.224860	...	0.05317(
uterProgramming	0.021369	0.041760	0.076158	-0.047969	0.110226	0.139499	0.043879	0.121789	0.191525	0.149635	...	-0.299781
onicsAndSemicon	-0.019304	0.088068	0.123903	-0.026150	-0.133786	0.026659	0.047274	-0.000923	-0.005432	0.109907	...	-0.101312
ComputerScience	-0.031236	-0.024749	-0.050739	-0.020929	-0.015129	-0.013137	-0.006909	0.086161	0.053090	-0.016059	...	-0.128077
MechanicalEngg	-0.083987	0.060161	0.042870	-0.015850	-0.061272	-0.033850	-0.046042	-0.008649	-0.006461	0.002708	...	1.000000
ElectricalEngg	-0.024408	0.068455	0.085593	0.012326	-0.057312	0.055134	0.016535	0.029723	0.007168	0.026210	...	-0.046272
TelecomEngg	0.028421	0.060164	0.057063	-0.010562	-0.079172	-0.000657	0.077937	-0.018019	-0.028632	0.026092	...	-0.064349
CivilEngg	-0.013109	0.009898	0.000678	0.007080	-0.014273	-0.035964	-0.034213	-0.028461	-0.038780	-0.030404	...	0.10417(
onscientiousness	0.075360	0.050050	0.044066	0.036467	0.003147	0.048044	0.012093	0.024610	0.007225	-0.018171	...	0.00601(
agreeableness	0.087639	0.115473	0.093829	-0.036447	-0.033432	0.053377	0.009984	0.174948	0.130697	0.077396	...	-0.00376(
extraversion	0.006984	-0.022176	-0.031926	-0.006246	0.009707	-0.054623	0.008211	0.003313	-0.028864	-0.065295	...	-0.014642
nueroticism	0.011918	-0.121777	-0.083520	0.038786	0.021054	-0.074752	0.033827	-0.147243	-0.193569	-0.145108	...	0.048024
ss_to_experience	0.084511	0.015292	-0.007928	-0.019414	0.014351	0.005078	0.015314	0.061630	0.018907	-0.009126	...	-0.00546(
Salary	-0.036183	0.155174	0.149531	-0.174824	-0.017602	0.122469	0.004575	0.169293	0.188416	0.205358	...	0.02885(

- Vì các hệ số tương quan ở kết quả là tương đối nhỏ, nên em sẽ chia làm 4 mức, 4 mức trên chỉ phù hợp với dữ liệu kết quả bên trên, r là hệ số tương quan:
    1. |r| <= 0.01 : là mức yếu
    2. 0.01 < |r|<=0.1 : là mức trung bình
    3. 0.1 < |r|<=0.3 : là mức tốt
    4. 0.3 < |r|<=1: là mức rất tốt

- Giới thiệu về 4 mô hình:
    1. Mô hình 1: mô hình lấy những đặc trưng có hệ số tương quan 0.1 < r <=0.3 với đặc trưng **Salary:** gồm 9 đặc trưng được tô màu xanh lá :
    ['10percentage', '12percentage', 'CollegeTier', 'collegeGPA', 'English', 'Logical', 'Quant', 'Domain', 'ComputerProgramming']

ss_to_experience	0,08451	0,01529	-0,00793	-0,01941	0,01435	0,00508	0,01531	0,06163	0,01891	-0,00913	-0,01563	0,05965	-0,02985	0,
Salary	-0,03618	0,15517	0,14953	-0,17482	-0,0176	0,12247	0,00457	0,16929	0,18842	0,20536	0,12202	0,12587	-0,00929	-0,

    Lý do chọn các đặc trưng: muốn kiểm tra xem nếu chúng ta chọn những đặc trưng có hệ số tương quan tốt nhất cho mô hình thì mô hình của ta có tốt nhất hay không
    2. Mô hình 2: mô hình lấy những đặc trưng có hệ số tương quan 0.01 < r <=0.3 với đặc trưng **Salary** (Tức là loại bỏ những đặc trưng có hệ số tương quan yếu)**:** gồm 19 đặc trưng. Loại bỏ những đặc trưng được tô màu xanh dương:
    Các đặc trưng bị loại bỏ: [CollegeCityTier, ElectronicsAndSemicon, extraversion, openess_to_experience] . Lấy 19 đặc trưng còn lại.

3	0,015314	0,06163	0,018907	-0,00913	-0,01563	0,05965	-0,02985	0,037363	-0,00547	-0,03229	-0,01101	-0,03063	0,415984	0,605813	0,468572	-0,05139	1
9	0,004575	0,169293	0,188416	0,205358	0,122022	0,125866	-0,00929	-0,09551	0,028854	-0,04122	-0,04041	0,01615	-0,0577	0,068623	-0,00266	-0,0734	-0,00781

Lý do chọn các đặc trưng: muốn kiểm tra xem nếu như chúng ta loại bỏ những đặc trưng có tương quan kém thì mô hình có đạt kết quả tốt hay không

3. Mô hình 3: mô hình sẽ kết hợp những đặc trưng có hệ số tương quan tốt với nhau, có kiểu dữ liệu giống nhau, và có hệ số tương quan tốt với đặc trưng **Salary.** Ta sẽ tạo ra đặc trưng mới từ những đặc trưng thỏa mãn điều kiện trên bằng cách lấy trung bình cộng.

   - Đặc trưng đầu tiên ('avg_10percentage_12percentage') được tạo ra từ 2 đặc trưng: ['10percentage', '12percentage']

	Gender	)percentag	percentag	Co
Gender	1	0,165208	0,131372	(
10percentage	0,165208	1	0,644518	
12percentage	0,131372	0,644518	1	
CollegeTier	0,028943	-0,13547	-0,09292	

   - Đặc trưng thứ 2 ('avg_English_Logical_Quant') được tạo ra từ 3 đặc trưng: ['English', 'Logical', 'Quant']]

	I	J	K	
yT	English	Logical	Quant	Do
8	-0,02083	-0,00019	-0,10407	0,0
4	0,335863	0,309735	0,326948	0,0
9	0,19379	0,240047	0,316088	0,0
7	-0,1824	-0,18907	-0,24097	-0,
1	-0,14547	-0,09872	-0,13718	0,0
1	0,099539	0,200165	0,221253	0,0
1	0,028303	-0,00607	-0,01997	0,0
3	1	0,431918	0,368248	0,1
7	0,431918	1	0,502061	0,
7	0,368248	0,502061	1	0,
4	0,106778	0,20238	0,22486	
9	0,121789	0,191525	0,149635	0,3
4	-0,00092	-0,00543	0,109907	0,

Lý do chọn các đặc trưng: muốn kiểm tra xem nếu kết hợp 1 số đặc trưng lại thì kết quả mô hình có trở nên tốt hơn.

4. Mô hình 4: Mô hình lấy toàn bộ 23 đặc trưng của X_train

   Lý do chọn các đặc trưng: muốn kiểm tra xem nếu chúng ta dùng toàn bộ các đặc trưng thì mô hình có đạt kết quả tốt nhất hay không

- Để chuẩn bị dữ liệu :

```
# MODE 1
# mô hình có các đặc trưng có hệ số tương quan với đặc trưng Salary từ 0.1 --> 0.3 (thuộc nhóm có hệ số lớn nhất)
features_01_to_03 = ['10percentage', '12percentage', 'CollegeTier', 'collegeGPA', 'English',
                     'Logical', 'Quant', 'Domain', 'ComputerProgramming']

# MODE 2
# mô hình có các đặc trưng có hệ số tương quan với đặc trưng Salary từ 0.01 --> 0.3 (Loại bỏ những đặc trưng có hệ số tương quan
features_001_to_03 = ['Gender', '10percentage', '12percentage', 'CollegeTier', 'Degree', 'collegeGPA', 'English',
                      'Logical', 'Quant', 'Domain', 'ComputerProgramming', 'ComputerScience', 'MechanicalEngg', 'ElectricalEngg',
                      'TelecomEngg', 'CivilEngg', 'conscientiousness', 'agreeableness', 'nueroticism']

# MODE 3
# mô hình có các đặc trưng mới được tạo ra từ cách tính trung bình cộng các đặc trưng cũ (các đặc trưng cũ  phải tương thích và c
# Tạo đặc trưng mới bằng trung bình cộng
X_train['avg_10percentage_12percentage'] = X_train[['10percentage', '12percentage']].mean(axis=1)
X_train['avg_English_Logical_Quant'] = X_train[['English', 'Logical', 'Quant']].mean(axis=1)

features_newfeature = ['avg_10percentage_12percentage', 'avg_English_Logical_Quant']

# MODE 4
# Chọn tất cả 23 đặc trưog đầu tiên của DataFrame X_train
selected_23columns = X_train.columns[:23]

models = [features_01_to_03, features_001_to_03 , features_newfeature, selected_23columns]
```

- Dùng hàm **Find_Best_Model** (models) để tìm ra mô hình tốt nhất với k-fold Cross Validation (k tối thiểu là 5), In ra các giá trị MAE của từng mô hình

```
# Phần code cho yêu cầu 1d
best_model = Find_Best_Model(models)

print('\nMô hình tốt nhất là: ',best_model)

# In ra các kết quả cross-validation như yêu cầu
```

```
Mode 1  | Avg MAE = 113678.27290397554
Mode 2  | Avg MAE = 111453.37341151787
Mode 3  | Avg MAE = 114563.04881623926
Mode 4  | Avg MAE = 111577.78643242033

Mô hình tốt nhất là:  ['Gender', '10percentage', '12percentage', 'CollegeTier', 'Degree', 'collegeGPA', 'English', 'Logical',
'Quant', 'Domain', 'ComputerProgramming', 'ComputerScience', 'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg', 'CivilEngg', 'c
onscientiousness', 'agreeableness', 'nueroticism']
```

Ta thấy đặc trưng tốt nhất là **Mô hình 2(Mode 2)**, có chỉ số MAE nhỏ nhất.

- Sau đó huấn luyện hồi quy tuyến tính cho **Mô hình 2**

```
# Huấn luyện lại mô hình my_best_model trên toàn bộ tập huấn luyện
my_best_model = LinearRegression()

# Chọn đặc mô hình tốt nhất và huấn luyện mô hình trên toàn bộ dữ liệu
X_best = X_train[best_model]
y_best = y_train

my_best_model.fit(X_best, y_best)
```

```
▾ LinearRegression
LinearRegression()
```

- Ta có công thức cho mô hình vừa huấn luyện được

```
print(CTMoHinh(my_best_model, best_model))
```

```
Salary = 91730.85 + -24795.791*Gender + 703.678*10percentage + 1048.662*12percentage + -96755.850*CollegeTier + 3664.376*Degree
+ 1317.729*collegeGPA + 139.267*English + 109.411*Logical + 92.895*Quant + 25007.500*Domain + 91.927*ComputerProgramming + -17
1.319*ComputerScience + 51.782*MechanicalEngg + -143.423*ElectricalEngg + -86.280*TelecomEngg + 141.299*CivilEngg + -20126.168*
conscientiousness + 14787.990*agreeableness + -11570.417*nueroticism
```

- Kiểm tra với x_test1d

```
X_test1d = X_test[best_model]
y_pred1d = my_best_model.predict(X_test1d)
y_pred1d
```

```
array([262202.16533123, 356144.49139128, 322993.76554588, 216902.8491678 ,
       360510.28735511, 301889.20324849, 225722.33643657, 254291.83583178,
       259592.89622843, 321325.74806804, 224430.29229621, 258576.60275675,
       242006.49585217, 307469.59875831, 397965.20083491, 319825.87894547,
       245794.31479371, 167172.73825669, 345314.3233795 , 352427.65837023,
       260880.5692732 , 346540.98763993, 385062.52139422, 491537.91015337,
       164788.94495156, 354692.97272032, 306236.02562349, 358464.12344352,
       368853.2541646 , 309509.84319579, 270200.43904607, 340260.75302092,
       343347.05966043, 430753.21489385, 308390.08340698, 261805.50679135,
       510982.86001653, 300843.37751323, 398136.88961711, 294329.17857934,
       429487.90338902, 356032.54687424, 282860.11400839, 358087.84657668,
       237771.18236966, 404062.40945383, 272294.64929716, 257085.57509431,
       470229.2437303 , 358158.85371704, 218176.15220745, 381403.41808393,
```

- Nhận xét:
  - Mô hình được huấn luyện trên đặc trưng tốt nhất trong 5 đặc trưng nên cho ra kết quả khá tốt trên tập test, dẫn đến MAE khá nhỏ, nên mô hình có hiệu suất tốt.

```
# Gọi hàm MAE (tự cài đặt hoặc từ thư viện) trên tập kiểm tra với mô hình my_best_model
mse1d = mean_absolute_error(y_test, y_pred1d)

print("Mean Absolute Error:", mse1d)
```

```
Mean Absolute Error: 102989.2798338652
```

  - MAE được đánh giá khác nhỏ vì:
    Khoảng giá trị của y_test tương đối lớn

```
print(y_test.min())
print(y_test.max())
```

```
45000
2600000
```

    Khoảng giá trị từ 45,000 đến 2,600,000 là khá rộng. Với phạm vi này, giá trị MAE là 102989 chiếm ,khoảng 4% của phạm vi tổng giá trị (2,600,000 - 45,000).
    4% < 5% thì điều này có thể được coi là một hiệu suất tốt.
- Giả thuyết cho mô hình đạt kết quả tốt nhất :
  - Trong 4 mô hình được sửa dụng k-fold Cross Validation. Thì MAE của 4 mô hình không có sự cách biệt quá lớn.

```
Mode 1  | Avg MAE = 113678.27290397554
Mode 2  | Avg MAE = 111453.37341151787
Mode 3  | Avg MAE = 114563.04881623926
Mode 4  | Avg MAE = 111577.78643242033

Mô hình tốt nhất là:  ['Gender', '10percentage', '12percentage', 'CollegeTier', 'Degree', 'collegeGPA', 'English', 'Logical',
'Quant', 'Domain', 'ComputerProgramming', 'ComputerScience', 'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg', 'CivilEngg', 'c
onscientiousness', 'agreeableness', 'nueroticism']
```

  - **Mô hình 1** là mô hình được tạo nên từ những đặc trưng có hệ số tương quan lớn nhất nhưng không phải là mô hình tốt nhất, giả thuyết:
    - Overfitting: nó có thể tương thích quá mức với dữ liệu huấn luyện cụ thể, bao gồm cả các đặc trưng có tương quan lớn. Tuy nhiên, mô hình này có thể không tổng quát hóa tốt cho dữ liệu mới, dẫn đến hiệu suất kém trên dữ liệu kiểm tra hoặc thực tế
  - **Mô hình 2** là mô hình được tạo nên từ những đặc trưng có hệ số tương quan lớn hơn 0.01 là mô hình tốt nhất, giả thuyết:

    - Mô hình tương đối đầy đủ, các đặc trưng thì có hệ số tương quan tốt đến rất tốt, mô hình không bị Overfitting.
- **Mô hình 3** là mô hình được tạo nên từ việc kết hợp những đặc trưng có hệ số tương quan tốt với nhau, giả thuyết:
    - Overfitting: nó có thể tương thích quá mức với dữ liệu huấn luyện cụ thể, bao gồm cả các đặc trưng có tương quan lớn. Tuy nhiên, mô hình này có thể không tổng quát hóa tốt cho dữ liệu mới, dẫn đến hiệu suất kém trên dữ liệu kiểm tra hoặc thực tế.
    - Mô hình chỉ có 2 đặc trưng nên. Dù có tương quan lớn với biến mục tiêu, 2 đặc trưng có thể không mang đến đủ thông tin hoặc khả năng dự đoán cho mô hình. Có thể có những đặc trưng khác không có tương quan lớn nhưng lại cung cấp thông tin quan trọng để mô hình có thể dự đoán chính xác hơn.
- **Mô hình 4** là mô hình được tạo nên từ toàn bộ 23 đặc trưng, giả thuyết:
    - Một số đặc trưng có thể chứa nhiễu hoặc thông tin không liên quan, và việc bao gồm chúng trong mô hình có thể làm giảm hiệu suất của mô hình. Những đặc trưng này có thể gây ra sự phân tán không cần thiết trong dữ liệu và làm cho mô hình khó khăn trong việc tìm ra mẫu và mối quan hệ chính xác.

5. Nhận xét chung

- Mô hình chỉ sử dụng một đặc trưng để dự đoán biến mục tiêu. Nhưng mô hình quá đơn giản nên chỉ số MAE thấp. Nhưng mô hình quá đơn giản dẫn đến việc nếu dữ liệu kiểm tra quá khác biệt sẽ bị sai số lớn. Điều này có thể dẫn đến hiệu suất dự đoán không cao, vì thông tin từ một đặc trưng có thể không đủ để mô hình hiểu và dự đoán đúng. Điều này cũng có thể dẫn đến sự tồn tại của nhiễu và sự tương quan không chính xác giữa đặc trưng và biến mục tiêu.
- Mô hình dựa vào những đặc trưng có hệ số tương quan lớn nhất với biến mục tiêu. Tuy nhiên, việc chỉ dựa vào tương quan không đảm bảo rằng mô hình sẽ là mô hình tốt nhất. Điều này có thể gây ra hiện tượng overfitting, khi mô hình tương thích quá mức với dữ liệu huấn luyện cụ thể mà không tổng quát hóa tốt cho dữ liệu mới.
- Mô hình sử dụng các đặc trưng có hệ số tương quan lớn hơn 0.01. Điều này có thể đưa đến một mô hình tốt hơn vì các đặc trưng này có khả năng dự đoán tốt đến rất tốt và có thể mang lại kết quả chính xác hơn. Khả năng mô hình không bị overfitting cũng tốt hơn.
- Mô hình này kết hợp các đặc trưng có tương quan tốt với nhau. Tuy nhiên, những vấn đề như overfitting và khả năng tổng quát hóa vẫn có thể xảy ra, tương tự như mô hình 1. Ngoài ra, việc chỉ sử dụng 2 đặc trưng có tương quan lớn không đảm bảo rằng chúng cung cấp đủ thông tin để mô hình dự đoán chính xác.
- Mô hình này sử dụng toàn bộ 23 đặc trưng. Tuy nhiên, việc bao gồm tất cả các đặc trưng có thể dẫn đến hiện tượng nhiễu và làm giảm hiệu suất của mô hình. Một số đặc trưng có thể không mang lại thông tin hữu ích và gây ra sự phân tán không cần thiết trong dữ liệu.

6. Tài liệu tham khảo

- https://github.com/rayleigh420/MTH00051_HCMUS_Project-3_Linear-Regression_Life-Expectancy.git (sử dụng trong việc hiểu về hệ số tương quan ở câu 1d)
- Chat GPT(sử dụng để tham khảo các câu lệnh về mô hình, tính các chỉ số MAE, công thức cho mô hình).