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

BÁO CÁO ĐỒ ÁN 3
TOÁN ỨNG DỤNG VÀ THỐNG KÊ CHO CÔNG NGHỆ THÔNG TIN

TÊN ĐỒ ÁN:
LINEAR REGRESSION

LỚP: 21CLC05

Thông tin cá nhân:

21127099 Nguyễn Tấn Lộc

Giảng viên hướng dẫn:
Thầy Vũ Quốc Hoàng
Thầy Lê Thanh Tùng
Cô Phan Thị Phương Uyên
Thầy Nguyễn Văn Quang Huy

MỤC LỤC

THÔNG TIN CÁ NHÂN

Họ tên: Nguyễn Tấn Lộc

Mã số sinh viên: 21127099

Lớp: 21CLC5

THÔNG TIN ĐỒ ÁN

Mã học phần: MTH00057

Tên học phần: Toán ứng dụng và thống kê cho công nghệ thông tin

Tên đồ án: Linear Regression

Nội dung: 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.

KẾT QUẢ ĐỒ ÁN

1. Các thư viện sử dụng

- **pandas**: Thư viện dùng để làm việc với dữ liệu dưới dạng bảng (DataFrame) và chuỗi dữ liệu (Series).
- **numpy**: Thư viện dành cho tính toán số học và xử lý mảng nhiều chiều.
- **matplotlib.pyplot**: Thư viện dùng để tạo các biểu đồ và đồ thị trực quan.
- **seaborn**: Thư viện dùng để tạo biểu đồ NHIỆT trực quan dựa trên dữ liệu từ pandas.
- **sklearn.ensemble.RandomForestRegressor**: Là một lớp trong thư viện **scikit-learn** (hay sklearn) được sử dụng để xây dựng mô hình RandomForest cho bài toán hồi quy (dự đoán giá trị số liên tục).

2. Các hàm sử dụng

- Em sử dụng 1 class **OLSLinearRegression** của cô trong bài lab 04: [0]
  - **fit:** huấn luyện mô hình.
  - **get_params:** lấy thông số từng đặc trưng**.**
  - **predict:** dự đoán đầu ra.
  - **mae:** đo độ sai lệch trung bình giữa dự đoán và thực tế.
- **pandas: [1]**
  - **read_csv:** đọc dữ liệu từ file csv. [1.1]
  - **iloc và loc:** truy cập và lấy dữ liệu từ DataFrame. [1.1]
  - **DataFrame**: không phải là hàm, đây là 1 cấu trúc để làm việc với bảng dữ liệu. [1.1]

- **copy:** tạo một bản sao của DataFrame. [1.1]
- **to_numpy:** chuyển đổi DataFrame sang mảng Numpy. [1.1]
- **colums:** là thuộc tính của DataFrame, dùng để truy cập các cột của 1 DataFrame. [1.1]

- **numpy: [2]**
  - **np.linalg.inv:** tính ma trận nghịch đảo. [2.1]
  - **sum:** tính tổng các phần tử trong mảng. [2.1]
  - **ravel:** biến đổi mảng nhiều chiều thành một chiều → *làm phẳng mảng*. [2.1]
  - **mean:** tính trung bình cộng của tập hợp các số. [2.1]
  - **abs:** tính giá trị tuyệt đối của các số trong mảng. [2.1]
  - **permutation:** tạo hoán vị các số với seed tương ứng. [2.1]
  - **concatenate:** nối các mảng thành 1 mảng duy nhất. [2.1]
  - **zeros:** tạo các mảng chứa số 0. [2.1]
  - **argmin:** trả về chỉ số của phần tử có giá trị nhỏ nhất trong một mảng. [2.1]
  - **float64:** không phải là hàm, là một kiểu dữ liệu số thực với độ chính xác kép. [2.1]
  - **array:** tạo một mảng. [2.1]
  - **log:** tính giá trị logarit tự nhiên. [2.1]
  - **corrcoef:** tính ma trận hệ số tương quan. [2.1]
  - **add:** cộng các phần tử trong 2 mảng. [2.1]
- **matplotlib.pyplot: [3]**
  - **figure:** Tạo một figure để chứa các thành phần trực quan. [3.1]
  - **barh:** tạo biểu đồ cột ngang. [3.1]
  - **xlabel:** thêm nhãn cho trục x. [3.1]
  - **ylabel:** thêm nhãn cho trục y. [3.1]
  - **title:** thêm tiêu đề cho biểu đồ. [3.1]
  - **tight_layout:** điều chỉnh khoảng cách giữa các thành phần trên biểu đồ. [3.1]
  - **show:** hiển thị biểu đồ. [3.1]
  - **hist:** tạo biểu đồ lược đồ. [3.1]
  - **legend:** Thêm chú thích vào biểu đồ. [3.1]

- **seaborn: [4]**
  - **heatmap:** tạo biểu đồ heatmap. [4]
- **RandomForestRegressor: [5]** Xây dựng mô hình Random Forest (sklearn).
- **Hàm sử dụng không nằm trong thư viện:**
  - **sort_values:** sắp xếp các hàng/cột.
  - **round:** làm tròn đến chữ số thập phân mong muốn.
  - **format:** định dạng và hiển thị chuỗi**.**

  ➔ **Trong quá trình làm bài và viết báo cáo, em có thể liệt kê không đủ các hàm. Thầy cô lượng thứ bỏ qua cho em.**

3. Các hàm tự cài đặt

- Em giải thích về lớp **OLSLinearRegression**:
- Hàm **fit:**
  - ***Input:***
    - X: ma trận các đặc trưng (thường là X train).
    - y: vector mục tiêu → ở đây là Salary.
  - ***Output:***
    - Vì hàm này chỉ tìm các trọng số w cho mô hình hồi quy tuyến tính bằng phương pháp OLS nên không có giá trị trả về.
  - ***Giải thích hàm:***
    - Hàm fit tính ma trận nghịch đảo của X, sau đó nhân nghịch đảo đó với ma trận X và cuối cùng nhân với vector y để tìm ra các trọng số w. Theo em nghĩ, nếu tính ma trận không vuông thì nên dùng np.linalg.pinv.
- Hàm **get_params:**
  - ***Input:***
    - Không có tham số truyền vào.
  - ***Output:***
    - Trả về trọng số w của mô hình sau khi train.
- Hàm **predict:**
  - ***Input:***
    - X: ma trận các đặc trưng (thường là X test).
  - ***Output:***

        - Trả về vector dự đoán của salary dựa trên w.
    - ***Giải thích hàm:***
        - Hàm predict nhân ma trận X với vector trọng số w, sau đó dùng hàm np.sum với cách nhân phần tử để tính tổng dự đoán của từng mẫu.

- Hàm **mae:**
    - ***Input:***
        - y: vector Salary thực tế (thường là Y test).
        - y_hat: vector Salary dự đoán (là kết quả dự đoán sau khi dùng hàm predict).
    - ***Output:***
        - Trả về giá trị trung bình của độ lỗi tuyệt đối giữa **y** và **y_hat**, là độ đo Mean Absolute Error (MAE) mà em được học trên lớp.
    - ***Giải thích hàm:***
        - Hàm này tính độ lỗi tuyệt đối (tức là np.abs) giữa từng cặp điểm trong **y** và **y_hat**, sau đó tính trung bình (tức là np.mean) của các độ lỗi này để đo lường sai số dự đoán của mô hình.
- Em có làm 2 hàm KFold vì lí do:
    - Ban đầu em nghĩ việc chạy vòng lặp trên từng đặc trưng trước khi chạy vòng lặp trên từng k sẽ giống nhau với việc chạy vòng lặp trên từng k trước khi chạy vòng lặp trên từng đặc trưng vì khi làm câu 1b em thấy k bằng số lượng các đặc trưng nên em đã hiểu nhầm. Sang tới câu 1c thì k khác số lượng đặc trưng nên em đã sửa lại và viết thêm một hàm. Nên hiện tại em có 2 hàm Kfold.
- Hàm **k_fold_1:**
    - ***Input:***
        - X và y: dữ liệu đầu vào là giá trị các dòng của tập train và giá trị Salary tương ứng.
        - k: Số lượng fold trong quá trình Cross Validation.
    - ***Output:***
        - Trả về giá trị trung bình của tất cả các kết quả MAE.
    - ***Giải thích hàm:***
        - Tạo 1 danh sách có size là 2248 sau đó shuffle các chỉ số dựa trên seed là 21127099.

- Lấy fold size = 2248/k
- Trong vòng lặp for, từng fold được xử lý theo k như sau:
  - Em sẽ chia thành 5 tập dữ liệu dùng để huấn luyện và đánh giá dựa trên fold size: (449, 449, 449, 449, 452) → Chia dữ liệu thành fold hiện tại (được chỉ định bởi **start** và **end**).
  - Sau đó em huấn luyện mô hình dựa trên tập dữ liệu đã chia sẵn.
  - Dự đoán giá trị và tính mae giữa thực tế và dự đoán.
  - Sau đó thêm mae đó vào list mae ban đầu là **mae_results**.
- Sau khi chạy hết vòng for thì sẽ trả về output là giá trị trung bình của tất cả các kết quả MAE.

- Hàm **k_fold_2:**
  - ***Input:***
    - X và y: dữ liệu đầu vào là giá trị các dòng của tập train và giá trị Salary tương ứng.
    - k: Số lượng fold trong quá trình Cross Validation.
  - ***Output:***
    - Trả về giá trị trung bình của tất cả các kết quả MAE.
  - ***Giải thích hàm:***
    - Tạo 1 danh sách có size là 2248 sau đó shuffle các chỉ số dựa trên seed là 21127099.
    - Lấy fold size = 2248/k
    - Trong vòng lặp for thứ nhất, từng fold được xử lý theo k như sau:
      - Em sẽ chia thành 5 tập dữ liệu dùng để huấn luyện và đánh giá dựa trên fold size: (449, 449, 449, 449, 452) → Chia dữ liệu thành fold hiện tại (được chỉ định bởi **start** và **end**).
      - Với mỗi đặc trưng, em chỉ giữ lại đặc trưng đó trong dữ liệu huấn luyện và đánh giá và giữ nguyên nhãn của nó.
      - Sau đó sẽ có vòng for thứ 2 được xử lý theo từng đặc trưng riêng lẻ thì em huấn luyện mô hình dựa trên tập dữ liệu đã chia sẵn.
      - Dự đoán giá trị và tính mae giữa thực tế và dự đoán với từng đặc trưng.

- Sau đó thêm mae đó vào list mae ban đầu là **mae_results**.

- Sau khi chạy hết vòng for thì sẽ trả về output là giá trị trung bình của tất cả các kết quả MAE.

4. Giải thích các câu

- Ở đồ án này, em dựa theo quy định đọc dữ liệu của cô là:
  - **X train**: sẽ là dữ liệu tương ứng các dòng của các thuộc tính (đặc trưng) trong bộ train.
  - **X test:** sẽ là dữ liệu tương ứng các dòng của các thuộc tính (đặc trưng) trong bộ test.
  - **Y train:** sẽ là cột *lương* của tất cả các dòng trong bộ train.
  - **Y test:** sẽ là cột *lương* của tất cả các dòng trong bộ test.

4.1. Câu 1a:

- Em lấy các thuộc tính của đề yêu cầu để vào thành 1 mảng.
- Sau đó sẽ lấy X train và X test theo các thuộc tính (nghĩa là chỉ lấy các dòng của các cột có thuộc tính trong mảng):
  - Nếu dùng loc thì syntax sẽ khác với iloc một chút.
- Em tiếp tục lấy cột lương của cả 2 bộ train và test: là Y train và Y test
- Sau khi đã chuẩn bị dữ liệu, em huấn luyện dựa trên mô hình **OLSLinearRegression** bằng hàm **fit → lấy X train và Y train để fit.**
- Để thể hiện công thức hồi quy, em sử dụng hàm **get_params** để lấy các thông số w và in ra công thức hồi quy.
- Dùng mô hình để dự đoán tập test (X test)
- Cuối cùng là tính MAE dựa trên kết quả dự đoán và Y test.

4.2. Câu 1b:

- Đối với cách sử dụng hàm **k_fold_1 (vì đây là hướng suy nghĩ sai nên em đã comment lại code)**:
  - Em cũng tạo một mảng chứa tên các đặc trưng là **['conscientiousness', 'agreeableness', 'extraversion', 'nueroticism', 'openess_to_experience']**.
  - Sau đó vòng lặp sẽ duyệt qua từng đặc trưng trên để thử nghiệm và tính mae trung bình cho từng đặc trưng.
  - Trong vòng lặp for:

- Em cũng lấy X train dựa trên mảng các đặc trưng.
- Em gọi hàm **k_fold_1** để thực hiện k-fold Cross Validation trên mô hình sử dụng chỉ một đặc trưng tính cách và tính toán MAE trung bình.
- Sau khi tính toán thì lưu các giá trị MAE trung bình của mỗi đặc trưng vào mảng **mae_results_p.**
- Tiếp tục, em sử dụng **np.argmin** để tìm ra đặc trưng có MAE thấp nhất trong mảng **mae_results_p.**
- Cuối cùng là in ra console tên đặc trưng tốt nhất và MAE của nó.
- Em có in thêm DataFrame bao gồm kết quả MAE của từng đặc trưng.

- Đối với cách sử dụng hàm **k_fold_2**: Em làm tương tự như cách sử dụng hàm **k_fold_1,** chỉ khác ở việc em không gọi thêm 1 vòng lặp để duyệt qua từng đặc trưng vì vòng lặp này em đã làm trong hàm k_fold_2 (*chạy vòng lặp với k fold trước mới chạy vòng lặp duyệt qua từng đặc trưng*)

4.3. Câu 1c:

- Em làm tương tự như câu 1b, chỉ khác ở mảng các đặc trưng ban đầu của câu c là **['English', 'Logical', 'Quant'].**

4.4. Câu 1d:

- Ở phần 4 này, em chỉ viết các bước làm của từng mô hình. Em sẽ trình bày phần quá trình xây dựng các mô hình trong phần riêng.
- Các bước thực hiện mô hình 1:
  - Bước này em chỉ dùng để đánh giá các MAE của từng đặc trưng, không cần thiết phải làm vì mô hình này em lấy 23 đặc trưng.
    - Em tạo 1 mảng chứa tên 23 đặc trưng của bộ dữ liệu.
    - Em chạy vòng for từng đặc trưng:
      - Lấy ra bộ train và test dựa trên đặc trưng.
      - Huấn luyện mô hình bằng phương pháp OLS.
      - Dự đoán giá trị **y_hat** trên bộ test.
      - Tính MAE của đặc trưng rồi thêm giá trị MAE vào mảng feature_mae_1, thêm cả tên đặc trưng vào mảng feature_1.

- Sau khi chạy hết vòng for thì em dùng sorted để sắp xếp các đặc trưng từ MAE nhỏ nhất đến cao nhất (bước này không cần thiết, em sort lại để em đánh giá dễ hơn).
- Em tạo một DataFrame in ra 23 đặc trưng với MAE tương ứng.
- Sau đó em có thể hiện các MAE lên biểu đồ để đánh giá thông qua thư viện Matplotlib.

- Đến với bước chính của mô hình này như sau:
  - Em lấy ra bộ train và test theo 23 đặc trưng.
  - Cũng như các câu 1a - 1b - 1c, em huấn luyện dữ liệu dựa trên bộ train (kể cả cột Salary).
  - Sau đó sẽ dự đoán giá trị trên bộ test và tính MAE của mô hình.

- Các bước thực hiện mô hình 2:
  - Em nhìn vào mô hình 1 và thấy được MAE chênh lệch rất rõ ràng. Đến với mô hình 2 em làm 1 bước nhỏ để lấy ra 11 đặc trưng có MAE nhỏ nhất (MAE nào nhỏ hơn trung bình của 3 câu 1a – 1b – 1c thì em lấy).
    - Em tạo 1 mảng chứa tên 23 đặc trưng của bộ dữ liệu.
    - Em chạy vòng for từng đặc trưng:
      - Lấy ra bộ train và test dựa trên đặc trưng.
      - Huấn luyện mô hình bằng phương pháp OLS.
      - Dự đoán giá trị **y_hat** trên bộ test.
      - Tính MAE của đặc trưng rồi so sánh với trung bình của 3 câu 1a – 1b – 1c. Nếu MAE nhỏ hơn MAE trung bình thì thêm giá trị MAE vào mảng feature_mae_2, thêm cả tên đặc trưng vào mảng feature_2.
      - Sau khi chạy hết vòng for thì em dùng sorted để sắp xếp các đặc trưng từ MAE nhỏ nhất đến cao nhất (bước này cũng giống như mô hình 1).
      - Em có được 11 đặc trưng có MAE nhỏ nhất.

- Em tạo một DataFrame in ra 11 đặc trưng với MAE tương ứng.
- Sau đó em có thể hiện các MAE lên biểu đồ để đánh giá thông qua thư viện Matplotlib (đến với biểu đồ này thì dễ nhìn hơn nên em có thể đánh giá dễ hơn).

- Đến với bước chính của mô hình này như sau:
  - Em lấy ra bộ train và test theo 6 đặc trưng có MAE nhỏ nhất.
  - Em huấn luyện dữ liệu dựa trên bộ train (kể cả cột Salary).
  - Sau đó sẽ dự đoán giá trị trên bộ test và tính MAE của mô hình.

- Các bước thực hiện mô hình 3:
  - Em lấy ra bộ train và test theo 6 đặc trưng có MAE nhỏ nhất và thêm 2 đặc trưng **ComputerProgramming** và **Domain**.
  - Em bình phương tập dữ liệu train và test.
  - Em huấn luyện dữ liệu dựa trên bộ train (kể cả cột Salary).
  - Sau đó sẽ dự đoán giá trị trên bộ test và tính MAE của mô hình.
- Các bước thực hiện mô hình 4:
  - Em lấy logarithm toàn bộ tập train và test (23 thuộc tính).
  - Em huấn luyện dữ liệu dựa trên bộ train (kể cả cột Salary).
  - Sau đó sẽ dự đoán giá trị trên bộ test và tính MAE của mô hình.
- Các bước thực hiện mô hình 5:
  - Em sử dụng mô hình Random Forest Regressor với số cây là 100, độ sâu là 10 và seed là 21127099.
  - Em huấn luyện dữ liệu dựa trên bộ train (kể cả cột Salary).
  - Sau đó sẽ dự đoán giá trị trên bộ test và tính MAE của mô hình.
- Các bước thực hiện mô hình 6 [7]:
  - Mô hình này em làm tệ nhất vì em chưa hiểu rõ về Phương pháp Ensemble.
  - Em tính kết quả dự đoán bằng cách lấy trung bình từ 7 mô hình:
    - Câu 1a.
    - Câu 1b.
    - Câu 1c.

- Mô hình 1.
- Mô hình 2.
- Mô hình 3.
- Mô hình 5.

▪ Sau đó tính kết quả MAE của mô hình dựa vào dự đoán trên.

- Các bước thực hiện mô hình 7:
  - Em sử dụng hàm **np.corrcoef** để tính toán ma trận tương quan giữa các đặc trưng và Salary.
  - Sau đó, em tiếp tục sử dụng thư viện seaborn để tạo **heatmap** từ ma trận tương quan trên.
  - Em lấy ra bộ train và test theo 11 đặc trưng có giá trị tương quan mạnh.
  - Em huấn luyện dữ liệu dựa trên bộ train (kể cả cột Salary).
  - Sau đó sẽ dự đoán giá trị trên bộ test và tính MAE của mô hình.
  - Mô hình này có 11 thuộc tính giống với câu 1a nên chắc chắn sẽ loại. Em muốn huấn luyện thử để xem MAE của mô hình này nên em viết báo cáo lại.
- Các bước thực hiện mô hình 8:
  - Em lấy bộ train và test của mô hình 7 để lập phương bộ dữ liệu.
  - Em huấn luyện dữ liệu dựa trên bộ train (kể cả cột Salary).
  - Sau đó sẽ dự đoán giá trị trên bộ test và tính MAE của mô hình.
- Các bước thực hiện mô hình 9:
  - Em thể hiện biểu đồ của 23 đặc trưng để đánh giá tính phân phối của bộ dữ liệu.
  - Em lấy bộ train và test của mô hình 13 để lập phương bộ dữ liệu.
  - Em huấn luyện dữ liệu dựa trên bộ train (kể cả cột Salary).
  - Sau đó sẽ dự đoán giá trị trên bộ test và tính MAE của mô hình.

5. Kết quả và nhận xét

- Câu 1a:
  - Kết quả:
    - MAE: 104863.778.
    - Công thức hồi quy:

Salary = - 22756.513 * Gender + 804.503 * 10percentage + 1294.655 * 12percentage - 91781.898 * CollegeTier + 23182.389 * Degree + 1437.549 * collegeGPA - 8570.662 * CollegeCityTier + 147.858 * English + 152.888 * Logical + 117.222 * Quant + 34552.286 * Domain

  - Nhận xét:
    - Theo em thấy, MAE của câu này khá là nhỏ, có nghĩa là các đặc trưng này có mối quan hệ rõ rệt với mức lương.
    - Nhìn vào công thức hồi quy, em thấy có 3 trọng số là âm, em có nhận xét như sau:
      - Đối với Gender, khi giới tính là nữ thì mức lương sẽ giảm đi một lượng 22756 Rupy → Theo em nghĩ đây là một suy nghĩ khá là "trọng nam khinh nữ", đối với thời đại 4.0 hiện nay thì giới tính không ảnh hưởng về mức lương vì có hiện nay thì phụ nữ có rất nhiều cá nhân thành công. Nhưng mà đối với bộ dữ liệu thu thập từ Ấn Độ thì mức lương sẽ giảm nếu giới tính là nữ.
      - Đối với CollegeTier và CollegeCityTier thì em có nhận xét là nếu như trường đại học nằm ở Tier thấp thì chất lượng đào tạo không tốt bằng các trường đại học Tier cao nên sẽ làm giảm lương đi một lượng.
- Câu 1b:
  - *Kết quả*: (5 đặc trưng với 5 MAE)
    - conscientiousness: 306205.604562
    - agreeableness: 300852.198264
    - extraversion: 306974.381510
    - **nueroticism: 299268.075054 → MAE nhỏ nhất**
    - openess_to_experience: 302962.139388

→ MAE của ***nueroticism*** *sau khi train lại****: 291019.693***

- Công thức hồi quy:

Salary = -56546.304 * nueroticism

- *Nhận xét*:
  - Đặc trưng tính cách sẽ có các nhận xét của em như sau:
    - **conscientiousness**: Đặc trưng này có MAE lớn, có thể cho thấy mối quan hệ không phản ánh đầy đủ tác động của *tính tỉ mỉ* đối với mức lương.
    - **agreeableness**: Đặc trưng này có MAE lớn cho thấy không thể hiện chính xác tác động của *tính dễ tính* đối với mức lương.
    - **extraversion**: Đặc trưng này có MAE lớn cho thấy không thể hiện một mối quan hệ rõ ràng với *tính hướng ngoại* trong việc ảnh hưởng đến mức lương.
    - **nueroticism**: Đặc trưng này có MAE tương đối thấp hơn so với các đặc trưng khác cho thấy *tính thần kinh* có tác động tương đối đáng kể đối với mức lương.
    - **openess_to_experience**: Đặc trưng này có MAE ở mức trung bình cho thấy chưa hiện thị đầy đủ tác động của *tính mở lòng* đối với mức lương.
    - Nhìn vào công thức hồi quy, em thấy được trọng số là âm. Em nghĩ đây là một ảnh hưởng đúng, vì khi có áp lực về mặt tinh thần/tâm lý, nhân viên sẽ không làm việc hết năng suất và sẽ bị cắt giảm lương dựa trên mức độ không hoàn thiện công việc.

- Câu 1c:
  - *Kết quả:* (3 đặc trưng với 3 MAE)
    - English: 121907.298027
    - Logical: 120307.385153
    - **Quant: 118096.179105**

➔ MAE của **Quant** sau khi train lại: **106819.578**

- Công thức hồi quy:

Salary = 585.895 * Quant

- *Nhận xét:*
  - Đặc trưng kĩ năng sẽ có các nhận xét của em như sau:
    - **English**: Đặc trưng này có MAE là 121,907.298. Theo em thấy, điểm số tiếng Anh từ bài kiểm tra AMCAT có thể có tác động nhất định đối với mức lương của các kỹ sư. Tuy nhiên, kết quả MAE vẫn khá là lớn nên có thể là do mô hình không thể hiện một cách chính xác tác động của đặc trưng này.
    - **Logical**: Đặc trưng này có MAE lớn có thể không thể hiện đúng mối quan hệ tuyến tính giữa khả năng logic của kỹ sư và mức lương.
    - **Quant**: Đặc trưng này có MAE thấp nhất trong 3 kỹ năng có khả năng tốt hơn trong việc ước tính tác động của khả năng định lượng đối với mức lương.
    - Nhìn vào công thức hồi quy, ta có thể đưa ra kết luận Quant sẽ đóng góp một lượng vào mức lương.

➔ Sau khi thực hiện câu 1a – 1b – 1c, em đưa ra một số nhận xét tổng thể như sau:

- MAE càng thấp thì mối liên quan giữa đặc trưng và mức lương càng cao. Nghĩa là trong câu 1b em thấy được MAE rất cao, cao hơn hẳn so với câu 1a và câu 1c nên theo em thì đặc trưng tính cách không tác động mạnh mẽ vào mức lương, trong khi đó đặc trưng kỹ năng lại tác động rất mạnh mẽ lên mức lương vì các đặc trưng kỹ năng có mối liên quan rõ ràng và trực tiếp đến khả năng của 1 cá nhân của lĩnh vực đó.
- Có thể thấy được, khi một người xin việc hay làm việc, kỹ năng của họ thường đóng vai trò quan trọng trong việc thực hiện công việc đó. Các đặc trưng kỹ năng như khả năng tiếng Anh, khả năng logic và khả năng định lượng đều có thể ảnh hưởng trực tiếp đến hiệu suất làm việc và đóng góp của một người trong công việc. Do đó, các đặc trưng này có khả năng tác động mạnh hơn đến mức lương.
- Ngược lại, đặc trưng tính cách thường là các yếu tố tương đối trừu tượng và phức tạp hơn để đo lường và ảnh hưởng đến hiệu suất làm việc. Hoặc là mô hình hồi quy tuyến tính có thể không thể hiện đầy đủ sự phức tạp của tương quan giữa các đặc trưng tính cách và mức lương.

- Tóm lại, các đặc trưng kỹ năng thường có mối quan hệ rõ ràng hơn và trực tiếp đến khả năng làm việc của từng nhân viên, do đó có thể tác động mạnh hơn đến mức lương trong mô hình.

- Câu 1d:
  - *Kết quả*: (9 mô hình với 9 MAE)
    - Mô hình 1: 101872.211
    - Mô hình 2: 107971.998
    - Mô hình 3: 104174.439
    - Mô hình 4: 300020.0000
    - Mô hình 5: 107043.127
    - Mô hình 6: 101064.851
    - Mô hình 7: 106446.915
    - Mô hình 8: 103288.616
    - Mô hình 9: 106848.357

  ➔ **Mô hình tốt nhất**: 101872.211
    - Công thức hồi quy:

Salary = - 23874.542 * Gender + 898.576 * 10percentage + 1203.496 * 12percentage - 83592.388 * CollegeTier + 11515.431 * Degree + 1626.519 * collegeGPA - 5717.734 * CollegeCityTier + 153.435 * English + 120.511 * Logical + 102.581 * Quant + 27939.640 * Domain + 76.730 * ComputerProgramming - 47.747 * ElectronicsAndSemicon - 177.388 * ComputerScience + 33.933 * MechanicalEngg - 151.471 * ElectricalEngg - 64.198 * TelecomEngg + 145.895 * CivilEngg - 19814.830 * conscientiousness + 15503.267 * agreeableness + 4908.582 * extraversion - 10661.029 * nueroticism - 5815.021 * openess_to_experience

  - *Nhận xét:*
    - Sau khi thực hiện tìm hiểu và tìm ra các mô hình, em có một số nhận xét như sau:
      - Mô hình 1 có MAE nhỏ nhất, nghĩa là mô hình này có khả năng dự đoán mức lương tốt nhất.
      - Mô hình 4 có MAE lớn nhất, nghĩa là mô hình này không tốt trong việc dự đoán mức lương, do việc biến đổi dữ liệu bằng hàm logarithm mà em đã giải thích trong phần 7 (Quá trình xây dựng 9 mô hình).

➔ Sau khi thực hiện đồ án, em có nhận xét tổng thể như sau:

- Theo em, chỉ có thể xây dựng một mô hình dự đoán mức lương đủ tốt chứ không thể xây dựng mô hình hoàn hảo vì có rất nhiều yếu tố ảnh hưởng đến mức lương, theo thời gian thì sẽ có nhiều yếu tố thay đổi. Mô hình tốt nhất của em là mô hình với 23 đặc trưng, có nghĩa là mức lương liên quan đến rất nhiều đặc trưng.
- Em có thử làm một vài mô hình không theo một giả thuyết gì cả thì có mô hình có MAE nhỏ hơn 100000 (nhưng vì em không biết giải thích nên em không ghi vào báo cáo).
- Vì MAE của các mô hình là hơn 1000000 dẫn tới việc so với thực tế thì sai số quá cao. Nên những mô hình của em chưa đủ tốt để dự đoán mức lương. Nếu có thêm thời gian, em sẽ tìm thêm nhiều mô hình tốt hơn.

6. Giả thuyết/Giải thích

- Câu 1b:
  - Sau khi thực hiện câu 1b, em có một số giả thuyết và giải thích:
    - Đặc trưng tốt nhất là **nueroticism.**
      - Tính cách **loạn thần kinh** ảnh hưởng đến khả năng làm việc trong môi trường áp lực:
        - *Giả thuyết*: Đặc trưng **nueroticism** (tức là loạn thần kinh) có thể ảnh hưởng đến cách mà các kỹ sư xử lý và thích nghi trong môi trường làm việc đầy áp lực và căng thẳng.
        - *Giải thích*: Các cá nhân có mức **loạn thần kinh** thấp thường có tính cách ổn định hơn, có khả năng kiểm soát tốt hơn các cảm xúc và tình huống khó khăn. Trong môi trường công việc, theo em thì những người này có thể duy trì sự tỉnh táo và tập trung của họ, thậm chí khi gặp phải áp lực lớn thì họ vẫn có thể hoàn thành tốt công việc đó. Điều này có thể làm cho họ trở thành những thành viên chủ chốt khi làm việc nhóm (thường sẽ làm leader) hiệu quả và có khả năng giải quyết vấn đề một cách hiệu quả hơn.

- Tính cách **loạn thần kinh** liên quan đến sự sẵn sàng thích nghi và đối mặt với thách thức:
  - *Giả thuyết*: Theo em thì mức độ **loạn thần kinh** có thể tương quan với khả năng của cá nhân trong việc đối mặt với sự thay đổi và thách thức.
  - *Giải thích*: Người có mức **loạn thần kinh** thấp thường có xu hướng thích nghi tốt hơn trong môi trường thay đổi nhanh chóng, nhất là thời đại 4.0 này. Họ có khả năng tìm ra cách giải quyết vấn đề một cách sáng tạo và hiệu quả. Khả năng này có thể giúp họ tự tin hơn trong việc giải quyết những tình huống phức tạp trong công việc và đạt được kết quả tích cực.
- Tính cách **loạn thần kinh** tương quan với tinh thần làm việc và động lực cá nhân:
  - *Giả thuyết*: Theo em thì mức độ **loạn thần kinh** cũng có thể liên quan đến tinh thần làm việc và sự động viên của chính mình (tự động viên).
  - *Giải thích*: Những người có mức **loạn thần kinh** thấp thường có tinh thần làm việc cao hơn và sự động viên bản thân mạnh mẽ hơn. Họ có khả năng duy trì tinh thần tích cực trong môi trường làm việc và không bị quá mức ảnh hưởng bởi các tình huống tiêu cực. Điều này có thể dẫn đến khả năng thể hiện hiệu suất làm việc cao hơn và sự cam kết đối với công việc của họ.

➔ Vì em đã đọc một số cuốn sách về tâm lý con người nên em đưa ra những giả thuyết trên.

➔ Các giả thuyết trên sẽ dẫn tới việc mức lương sẽ có ảnh hưởng, nhưng vì đây là đặc trưng tính cách nên sẽ không ảnh hưởng nhiều tới mức lương.

➔ Như vậy, thông qua việc phân tích các giả thuyết trên, em thấy rằng đặc trưng **nueroticism** có thể đóng góp vào hiệu suất làm việc của mỗi người bằng cách ảnh hưởng đến khả năng làm việc trong môi trường áp lực, khả năng thích nghi với thay đổi và thách thức, cũng như tinh thần làm việc và động viên cá nhân. Điều này giải thích tại sao em tìm được đặc trưng này đạt kết quả tốt nhất trong mô hình câu 1b.

- Câu 1c:
    - Sau khi thực hiện câu 1c, em có một số giả thuyết và giải thích:
        - Đặc trưng tốt nhất là **Quant**.
            - Khả năng **định lượng** ảnh hưởng đến khả năng giải quyết vấn đề và tư duy phân tích:
                - *Giả thuyết*: Đặc trưng Quant liên quan đến khả năng định lượng và xử lý dữ liệu số học, có thể ảnh hưởng đến khả năng giải quyết vấn đề và tư duy phân tích của mỗi người kỹ sư.
                - *Giải thích*: Các kỹ sư thường cần phải thực hiện rất nhiều tính toán và xử lý dữ liệu số học trong công việc của họ. Em nghĩ có khả năng Quant cao có thể giúp họ thực hiện các phân tích phức tạp hơn trong khoảng thời gian ngắn và hiểu rõ hơn về dữ liệu đó. Khả năng này cũng có thể ảnh hưởng đến cách họ tiếp cận và giải quyết các vấn đề kỹ thuật trong công việc, từ việc tối ưu hóa hệ thống đến việc phát triển giải pháp sáng tạo.
            - Kỹ năng logic và quyết định trong công việc kỹ thuật:
                - *Giả thuyết*: Đặc trưng Quant có thể phản ánh khả năng logic và quyết định của kỹ sư trong việc giải quyết vấn đề kỹ thuật.
                - *Giải thích*: Kỹ năng logic là quan trọng đối với kỹ sư để phân tích, lập luận, giải thuật và đưa ra quyết định dựa trên dữ liệu và thông tin nào đó (có sẵn hoặc không có sẵn). Khả năng Quant có thể ảnh hưởng đến khả năng họ xác định mô hình hoặc phương pháp tốt nhất để giải quyết một vấn đề cụ thể. Đặc biệt trong các lĩnh vực như phân tích dữ liệu, khoa học máy tính, và thiết kế hệ thống, kỹ năng Quant có thể đóng vai trò quan trọng trong việc xử lý thông tin và đưa ra quyết định có logic. Vì vậy rất quan trọng đối với các kỹ sư, nhất là những người theo ngành máy tính.

- Kỹ năng kỹ thuật và **ứng dụng thực tế** (đây có lẽ là giả thuyết để có thể đưa ra kết quả vì sao MAE lại nhỏ hơn câu 1b):
    - *Giả thuyết*: Đặc trưng Quant có thể liên quan đến khả năng tính toán chính xác trong việc thực hiện các phép tính, phân tích và ứng dụng vào thực tế như thế nào.
    - *Giải thích*: Trong công việc kỹ thuật, kỹ năng Quant có thể giúp kỹ sư thực hiện các tính toán phức tạp một cách chính xác. Khả năng này đặc biệt quan trọng trong việc phát triển sản phẩm, kiểm tra và đảm bảo chất lượng, cũng như trong việc thực hiện các thử nghiệm và phân tích dữ liệu để đưa ra các quyết định kỹ thuật. Vì nếu đưa ra một quyết định sai lầm nhỏ cũng có hậu quả đáng kể.

➔ Để có thể có một mức lương cao, thì thường những kỹ sư có kỹ năng Quant sẽ có lương cao hơn rất nhiều, ví dụ như những kỹ sư làm việc trong các công ty Big Tech lương rất cao nhưng họ rất áp lực về mặt tâm lý. Nên sau khi làm câu 1c ta sẽ thấy sự liên quan giữa 1b và 1c.

➔ Tóm lại, thông qua việc phân tích 3 giả thuyết trên của em, em thấy rằng đặc trưng Quant có thể đóng góp vào hiệu suất làm việc của ứng viên bằng cách ảnh hưởng đến khả năng giải quyết vấn đề, tư duy phân tích, kỹ năng logic và quyết định, cũng như khả năng ứng dụng kỹ thuật vào thực tế đời sống. Điều này giải thích tại sao đặc trưng này đạt kết quả tốt nhất trong mô hình.

- Câu 1d:
    - Trong câu d, em tìm ra được mô hình tốt nhất với 23 đặc trưng, em có một số nhận xét như sau:
        - **Quant** sẽ là đặc trưng đầu tiên ảnh hưởng rất rõ rệt tới mức lương.
        - **Degree** tức là bằng cấp của ứng viên đóng vai trò rất quan trọng trong việc có được mức lương như thế nào. Vì có thể dựa vào bằng cấp để đánh giá thực lực của ứng viên đó khi đi xin việc.
        - **collegeGPA** là GPA lúc tốt nghiệp đại học, nếu đã nói đến bằng cấp thì GPA cũng sẽ là đặc trưng khá quan trọng trong việc đánh giá thực lực thí sinh nhưng nó sẽ không ảnh hưởng rõ rệt như **Quant.**

- **nueroticism** là đặc trưng ảnh hưởng đến cách một ứng viên xử lý công việc khi stress như thế nào.
- Bên cạnh đó còn có những đặc trưng tính cách khác ảnh hưởng tới khả năng làm việc nhóm, tương tác với khách hàng, … cũng sẽ ảnh hưởng tới mức lương.
- Các đặc trưng liên quan đến khả năng kỹ thuật trong các lĩnh vực khác nhau cũng có thể ảnh hưởng đến việc tìm kiếm công việc, và dẫn tới mức lương khác nhau. Không thể giao một người học kinh tế đi làm DEV và ngược lại.

➔ Mỗi yếu tố đều có thể ảnh hưởng tới mức lương riêng lẻ và cả sự tương tác giữa chúng cũng có thể tạo ra sự khác biệt. Các yếu tố này cần được xem xét cùng nhau trong bối cảnh để có cái nhìn toàn diện và chính xác hơn về ảnh hưởng của chúng tới mức lương.

7. Quá trình xây dựng 9 mô hình

- Ban đầu em có đọc bài báo đề cập về việc đặc trưng tính cách ảnh hưởng tới mức lương như thế nào tại Trung Quốc [8], nhưng vì bài báo này đã bị rút lại (có thông báo trên bài báo) và bộ dữ liệu này thu thập ở Ấn Độ. Nên em đã tự đánh giá theo ý mình bằng các cách của em.
- Em suy nghĩ được một số cách có thể xây dựng mô hình:
  - ***Biến đổi đặc trưng***: bình phương, lập phương, lấy logarithm.
  - ***Sử dụng mô hình khác***: để mô phỏng mối quan hệ giữa đặc trưng và lương rõ hơn, em có sử dụng mô hình RandomForest. Ngoài ra còn một số mô hình khác.
  - ***Tạo mô hình kết hợp (Ensemble):*** Kết hợp nhiều mô hình thì em nghĩ có thể cải thiện khả năng dự đoán.
- Với mô hình 1, em nghĩ về việc bây giờ mình train 23 thuộc tính thì sẽ như thế nào, sau khi train thì em thấy được MAE của mô hình này cho đến hiện tại thì nhỏ nhất.
- Em nhìn vào biểu đồ MAE của mô hình 1 và thấy được độ chênh lệch MAE của các đặc trưng khá rõ rệt nên em đã làm mô hình 2, em lấy các đặc trưng có MAE nhỏ hơn MAE trung bình của 3 câu 1a – 1b – 1c in ra 1 biểu đồ mới để có thể đánh giá lại 1 lần nữa. Khi nhìn vào biểu đồ, em thấy được độ chênh lệch của 6 đặc trưng này không quá lớn, bắt đầu sang

tới đặc trưng CollegeTier (đặc trưng top 7) thì độ chênh lệch lớn nên em chỉ lấy 6 đặc trưng như sau để train:

- Quant, 12percentage, 10percentage, collegeGPA, Logical, English.

➔ Mô hình 2 loại vì MAE lớn hơn câu 1a – 1b – 1c (Em chỉ muốn lấy 3 mô hình tốt nhất, có nghĩa là MAE nhỏ hơn cả 3 ý trên).

- Đến với mô hình 3, em vẫn lấy lại 6 đặc trưng của mô hình 2 và em cảm thấy vì em đang học chuyên ngành IT nên kỹ năng và kiến thức chuyên ngành cũng sẽ liên quan đến mức lương sau này của em nên em đã thêm vào 2 đặc trưng **Domain** và **ComputerProgramming**. Sau đó em bình phương bộ dữ liệu vì khi biến đổi như vậy thì em nghĩ có thể làm tăng tương quan giữa các đặc trưng và lương, làm cho mô hình dễ dàng trong việc tìm ra mối quan hệ giữa chúng. Mặt khác, khi biến đổi thì chắc chắn là đã tạo ra đặc trưng mới (các biến của bình phương). Em nghĩ là mô hình sẽ có nhiều thông tin để dự đoán hơn. Khi ra được MAE của mô hình thì MAE nhỏ hơn câu 1a – 1b – 1c nhưng vẫn chưa phải là mô hình có MAE nhỏ nhất. Các đặc trưng như sau:
  - Quant, 12percentage, 10percentage, collegeGPA, Logical, English, Domain, ComputerProgramming.
- Đến với mô hình 4, em lấy logarithm của toàn bộ đặc trưng. Vì hàm logarithm chỉ đúng với các giá trị không âm nên khi em dùng cho mô hình này đã có các thông báo Warning (vì dữ liệu gốc có dữ liệu âm). Khi ra được MAE của mô hình quá lớn, em thấy được việc biến đổi dữ liệu bằng hàm logarithm sẽ loại bỏ một phần của dữ liệu, thêm vào đó thay đổi mối quan hệ giữa các đặc trưng và lương nên điều này làm cho mô hình OLS không phù hợp.

➔ Mô hình này loại vì MAE quá lớn.

- Mô hình 5 em sử dụng mô hình Random Forest, theo em tìm hiểu thì thuật toán hoạt động bằng cách chọn ngẫu nhiên các mẫu từ tập dữ liệu bằng seed cho sẵn (seed em cho là 21127099), sau đó xây dựng cây quyết định cho từng mẫu, và sau đó kết hợp các dự đoán bằng cách bỏ phiếu nhau. Nó sẽ chọn kết quả được dự đoán nhiều nhất → kết quả cuối cùng.

➔ Mô hình này loại vì MAE lớn hơn câu 1a.

- Đối với mô hình 6, đây là mô hình em tệ nhất nhưng vì em đã tìm hiểu nên em ghi vào báo cáo. Theo em tìm hiểu thì phương pháp này là phương pháp Ensemble Learning trong Machine Learning, được chia ra 3 loại

(Bagging – Boosting – Stacking). Ý tưởng của phương pháp này là kết hợp các mô hình khác nhau có khả năng khác nhau, có thể thực hiện tốt nhất các loại công việc khác nhau (subtasks), khi kết hợp các mô hình này với nhau một cách hợp lý thì sẽ tạo thành một mô hình kết hợp (combined model) mạnh có khả năng cải thiện hiệu suất tổng thể (overall performance) so với việc chỉ dùng các mô hình một cách đơn lẻ [9].

- Bagging: xây dựng một lượng lớn các models (thường là cùng loại) trên những subsamples khác nhau từ tập training dataset một cách song song nhằm đưa ra dự đoán tốt hơn. [9]
- Boosting xây dựng một lượng lớn các models (thường là cùng loại). Tuy nhiên quá trình huấn luyện trong phương pháp này diễn ra tuần tự theo chuỗi. Trong chuỗi này mỗi model sau sẽ học cách sửa những errors của model trước (hay nói cách khác là dữ liệu mà model trước dự đoán sai). [9]
- Stacking xây dựng một số models (thường là khác loại) và một mô hình supervisor model, mô hình này sẽ học cách kết hợp kết quả dự báo của một số mô hình một cách tốt nhất. [9]

➔ Vì phương pháp này em chưa hiểu rõ, nên em xin phép cô trong tương lai em sẽ thực hiện tìm hiểu tiếp → Mô hình này loại.

- Mô hình 7 em đã thể hiện một biểu đồ với các giá trị tương quan. Khi nhìn trên biểu đồ heatmap, em nhìn vào dòng/cột Salary và em thấy các đặc trưng có giá trị > 0 thì em tiếp tục nhìn vào giá trị tương quan của chính nó (không phải nằm trên dòng/cột Salary) thì em thấy giá trị gần bằng 1 cho thấy mối tương quan mạnh → Em lấy các đặc trưng như sau:
  - 10percentage, 12percentage, collegeGPA, English, Logical, Quant, Domain, ComputerProgramming, MechanicalEngg, CivilEngg, agreeableness.

➔ Mô hình này loại vì có trùng số lượng đặc trưng với câu 1a.

- Mô hình 8 là lập phương của bộ dữ liệu của mô hình 7.
- Đến với mô hình cuối là mô hình 9, em thể hiện 23 biểu đồ về tính phân phối của bộ dữ liệu. Sau đó em lấy các đặc trưng có tình phân phối đồng đều, nghĩa là không nằm về 1 phía → Em lấy các đặc trưng như sau:
  - 10percentage, 12percentage, collegeGPA, English, Logical, Quant, Domain, ComputerProgramming, conscientiousness, agreeableness, extraversion, nueroticism, openess_to_experience.

8. Điểm yếu bản thân

- Sau khi thực hiện đồ án này, em nhận ra được một số vấn đề mình cần cải thiện:
    - Dù đã thử tìm rất nhiều mô hình (có một số mô hình em tìm được do ngẫu nhiên sử dụng các đặc trưng mà em không biết cách giải thích nên em không báo cáo – kể cả những mô hình có chuẩn hóa dữ liệu) nhưng em vẫn chưa tìm được một mô hình tốt hơn mô hình với 23 đặc trưng.
    - Nếu có thời gian thêm, em có thể tìm ra được các mô hình tốt hơn và giải thích nó

→ Cuối cùng, sau khi hoàn thành đồ án này, em xin chân thành cảm ơn cô Phan Thị Phương Uyên, thầy Vũ Quốc Hoàng, thầy Nguyễn Văn Quang Huy và thầy Lê Thanh Tùng đã dành thời gian và hỗ trợ em trong quá trình làm đồ án này. Nhờ buổi hướng dẫn làm đồ án đã giúp em hiểu được cách tìm ra mô hình và đánh giá mô hình. Đây là đồ án cuối cùng của môn học. Một lần nữa em cảm ơn thầy cô rất nhiều và chúc thầy cô thành công trong sự nghiệp.

9. Tổng kết

- Trong quá trình làm đồ án, em đã tiến hành phân tích và mô hình hóa dữ liệu về mức lương và việc làm của các kỹ sư ngay sau khi tốt nghiệp ở Ấn Độ. Mục tiêu của dự án là tìm hiểu những yếu tố quyết định mức lương và việc làm của họ. Dữ liệu được thu thập từ hơn 6000 cơ sở đào tạo kỹ thuật với hơn 2,9 triệu sinh viên, là nguồn thông tin quý báu để em xây dựng mô hình dự đoán mức lương.
- Em đã tiến hành khám phá dữ liệu, xử lý dữ liệu thiếu và chuẩn bị dữ liệu cho việc xây dựng mô hình.
- Em đã thử nghiệm và so sánh nhiều mô hình dựa trên dữ liệu, từ đó tìm ra mô hình tốt nhất. Em đã phân tích tác động của các đặc trưng tính cách và kỹ năng đến mức lương, và xác định được những đặc trưng có ảnh hưởng mạnh và tác động tích cực đến dự đoán mức lương.
- Dựa trên quá trình thử nghiệm và phân tích, em đã xây dựng một mô hình hồi quy tuyến tính cuối cùng, chọn ra tập hợp các đặc trưng quan trọng nhất để dự đoán mức lương của kỹ sư ngay sau khi tốt nghiệp. Mô hình này có khả năng dự đoán mức lương với độ chính xác *tương đối nhưng chưa đủ tốt.*
- Tóm lại, đồ án 3 này đã giúp em hiểu rõ hơn về những yếu tố quyết định mức lương và việc làm của các kỹ sư ngay sau khi tốt nghiệp.

TÀI LIỆU THAM KHẢO

- *[0]: Tài liệu của giáo viên, cụ thể là lab04*
- *[1]: Pandas User Guide của Pandas: https://pandas.pydata.org/docs/user_guide/dsintro.html#dataframe*
- *[1.1]: Pandas API Reference của Pandas: https://pandas.pydata.org/docs/reference/frame.html*
- *[2]: Numpy User Guide của Numpy: https://numpy.org/doc/stable/user/index.html#user*
- *[2.1]: Numpy API Reference của Numpy: https://numpy.org/doc/stable/reference/routines.html*
- *[3]: Matplotlib User Guide của Matplotlib: https://matplotlib.org/3.5.3/users/index.html*
- *[3.1]: Matplotlib API Reference của Matplotlib: https://matplotlib.org/3.5.3/api/index.html*
- *[4]: Seaborn Heatmap của Seaborn: https://seaborn.pydata.org/generated/seaborn.heatmap.html*
- *[5]: RandomForestRegressor của scikitlearn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html*
- *[6]: Tìm hiểu về Heatmap: https://lptech.asia/kien-thuc/heatmap-la-gi-tim-hieu-ve-heatmap-trong-website-tu-a-den-z*
- *[7]: Mô hình kết hợp Ensemble: https://svcuong.github.io/post/ensemble-learning/*
- *[8]: Longlong Zhao, The effect of personality traits on employees' annual salaries in Chinese startups, Front Psychol, 2022: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9642428/*
- *[9]: Phương pháp Ensemble Learning trong Machine Learning: Boosting, Bagging, Stacking (Sử dụng R code): https://svcuong.github.io/post/ensemble-learning/*

---HẾT---