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

ĐỒ ÁN 3: Linear Regression

TOÁN ỨNG DỤNG VÀ THỐNG KÊ

HOÀNG ĐỨC VIỆT - 21127203

Giảng viên hướng dẫn:

Vũ Quốc Hoàng

Nguyễn Văn Quang Huy

Lê Thanh Tùng

Phan Thị Phương Uyên

Ngày 23 tháng 8 năm 2023

Mục lục

1 Thông tin cá nhân

- Họ và tên: Hoàng Đức Việt
- MSSV: 21127203
- Lớp: 21CLC02

2 Các thư viện đã sử dụng trong đồ án

2.1 pandas

- Đọc dữ liệu đầu vào trong file csv dưới dạng các dataframe.

2.2 numpy

- Chuyển các dataframe thành dạng numpy.array để tiện cho việc xử lí.
- Hỗ trợ xử lí, tính toán trên ma trận (chuyển vị, nhân, tìm ma trận khả nghịch).

3 Các hàm đã sử dụng trong đồ án

3.1 Tìm trọng số của mô hình

- **Tên hàm:** find_weight(X, y)
- **Input:**
  - X: Ma trận chứa dữ liệu các đặc trưng để tìm trọng số dựa vào y.
  - y: Ma trận chứa giá trị cần đạt được.
- **Output:**
  - Ma trận chứa trọng số "tốt nhất" có thể đạt được.

- **Mô tả:**
  - Ma trận trọng số được tính dựa vào công thức:

$$(X^T \times X)^{-1} \times X^T \times y$$

3.2 Tìm giá trị dự đoán của y

- **Tên hàm:** calc_y_predict(X, w)
- **Input:**
  - X: Ma trận chứa dữ liệu để tìm giá trị dự đoán của y dựa vào w.

- w: Trọng số của từng đặc trưng có trong mô hình.

- **Output:**
  - Ma trận chứa giá trị dự đoán của y

- **Ý tưởng:**
  - Ma trận X có kích thước $(m \times n)$ với m là số dòng dữ liệu, n là số cột đặc trưng.
  - Ma trận w có kích thước $(n \times 1)$ chứa trọng số của từng đặc trưng.
  - Để có thể nhân **element-wise** của trọng số ở mỗi từng đặc trưng của ma trận w với giá trị tương ứng của chúng ở ma trận X, ta cần chuyển vị ma trận w trước khi thực hiện phép tính - kích thước của w sau khi chuyển vị là $(1 \times n)$. Sau đó, ta cần cộng từng dòng lại để tính giá trị y dự đoán cho từng dòng dữ liệu.

- **Mô tả:**
  - Sử dụng phương thức **matrix.T** để lấy ma trận chuyển vị.
  - Sử dụng **broadcasting** để nhân các phần tử tương ứng với nhau.
  - Sử dụng **numpy.sum** để tính tổng theo từng hàng.

3.3 Tìm độ lỗi MAE

- **Tên hàm:** MAE(y_predict, y_real)
- **Input:**
  - y_predict: Giá trị y dự đoán.
  - y_real: Giá trị y thực tế.
- **Output:**
  - Trả về độ lỗi MAE sau khi tìm được.

- **Ý tưởng:**
  - Cả ma trận y_predict và y_real đều có kích thước $(n \times 1)$ với n là số dòng dữ liệu.
  - Trước hết ta cần tìm độ chênh lệch giữa các giá trị tương ứng trong y_predict và y_real. Sau đó, ta cộng các giá trị vừa tìm được lại.

□ Tìm trung bình cộng của giá trị vừa tìm được. Đó chính là MAE.

- **Mô tả:**

□ Sử dụng **numpy.abs** để lấy độ chênh lệch sau khi trừ các giá trị tương ứng trong y_predict và y_real.

□ Sử dụng **numpy.mean** để tính trung bình cộng của các giá trị vừa tính.

□ Trả về giá trị sau khi tính trung bình cộng (hay **MAE**) cho người dùng.

3.4 Tạo bảng kết quả hiển thị các MAE trung bình

- **Tên hàm:** create_result_table(feature, MAE, column_name)
- **Input:**

□ feature: Tên của từng đặc trưng (model).

□ MAE: Mảng chứa các MAE trung bình (kích thước của feature và MAE phải **giống nhau** - $n \times 1$)

□ column_name: Tiêu đề của 2 cột.

- **Ý tưởng:**

□ Hiển tên của đặc trưng (model) cùng với giá trị MAE trung bình của chúng cho người dùng tiện theo dõi.

- **Mô tả:**

□ Sử dụng **zip** nhằm gom (liên kết) các phần tử tương ứng của 2 mảng feature và MAE lại với nhau.

□ Sử dụng **pandas.DataFrame** nhằm tạo ra DataFrame chứa các thông tin cần hiển thị với tham số columns là column_name.

- **Ví dụ minh họa**:

	Model	MAE
0	Model 1	200841.509391
1	Model 2	121689.494475
2	Model 3	125627.751612

3.5 Tìm mô hình tốt nhất (cho 1b, 1c)

- **Tên hàm:** find_best_model(table, feature, clusters = 10)
- **Input:**
  - table: **DataFrame** của toàn bộ tập train.
  - feature: Các đặc trưng muốn xét (nhằm chọn ra đặc trưng có MAE trung bình nhỏ nhất trong tập).
  - clusters: Số tập con được chia ra từ tập **table** dùng để huấn luyện **(mặc định là 10)**.
- **Output:**
  - Trả về lần lượt **2** giá trị.
  - Giá trị đầu tiên là vị trí của đặc trưng được chọn (MAE trung bình nhỏ nhất trong các đặc trưng được xét).
  - Giá trị kế tiếp là mảng chứa các giá trị MAE trung bình sau quá trình huấn luyện.

- **Ý tưởng:**
  - Chia tập train ban đầu thành **clusters** tập nhỏ.
  - Với các tập nhỏ như trên, lần lượt **chọn từng tập** để huấn luyện. Mỗi tập ấy dùng để huấn luyện qua **tất cả các feature**. Với mỗi feature, sau khi huấn luyện xong, trọng số vừa tìm được qua quá trình huấn luyện sẽ được dùng để **kiểm tra trên clusters - 1 tập còn lại**.
  - Sau mỗi lần kiểm tra, các giá trị MAE tìm được sẽ được cộng vào mảng tổng (cộng vào vị trí tương ứng với vị trí trong mảng feature của chúng).
  - Sau khi hoàn tất quá trình kiểm tra, mảng tổng trên sẽ được dùng để tính MAE trung bình cũng như vị trí của đặc trưng có MAE trung bình nhỏ nhất để trả về cho người dùng.
- **Mô tả:**
  - Đầu tiên, cần thêm **'Salary'** vào mảng feature nhằm coi giá trị lương như một đặc trưng (tránh việc xáo trộn lương không đúng).
  - Sau đó, lấy toàn bộ dữ liệu của các cột có trong feature đồng thời xáo trộn các dòng dữ liệu với nhau.

□ Việc xáo trộn dữ liệu sử dụng **pandas.DataFrame.sample(frac=1)**. Với giá trị frac = 1, kết quả trả về là 1 DataFrame với 100% dữ liệu đã được xáo trộn.

□ Sau khi xáo trộn, ta thực hiện loại bỏ **'Salary'** ra khỏi mảng feature để đảm bảo tính đúng đắn trong quá trình xử lí.

□ Dữ liệu sau khi được xáo trộn sẽ được chia thành **clusters** tập nhỏ bằng **numpy.array_split**.

□ Như ý tưởng đã trình bày phía trên, các tập con sẽ lần lượt được duyệt qua và được chọn làm tập để huấn luyện. Với mỗi tập con như thế, tất cả các feature sẽ lần lượt được huấn luyện qua và đem đi test với **clusters - 1** tập còn lại nhằm lấy giá trị MAE để cộng vào mảng tổng.

□ Trong suốt quá trình trên, các hàm đã được cài đặt như **find_weight**, **calc_y_predict** hay **MAE** sẽ được sử dụng để hỗ trợ.

□ Cuối cùng, nhằm lấy vị trí của đặc trưng có MAE trung bình nhỏ nhất, hàm **numpy.argmin** sẽ được sử dụng.

3.6 Huấn luyện mô hình tốt nhất cho 1b

- **Tên hàm:** best_personality_feature_model(table, best_personality_feature)
- **Input:**

  □ table: **DataFrame** chứa toàn bộ dữ liệu để huấn luyện.

  □ best_personality_feature: Tên của feature tốt nhất trong quá trình huấn luyện câu 1b.
- **Output:**

  □ Trọng số của mô hình sau quá trình huấn luyện.

- **Ý tưởng:**

  □ Tính trọng số của đặc trưng tốt nhất trên toàn bộ tập dữ liệu train.

  □ Trọng số sau khi được trả về sẽ được dùng để tính giá trị y dự đoán của tập test **ở block tiếp theo**.

  □ Từ đó, cũng có thể tính được MAE giữa giá trị y dự đoán và y thực tế.

- **Mô tả:**

  □ Sử dụng hàm **find_weight** đã cài đặt để lấy trọng số của đặc trưng tốt nhất trên tập train (tức **table**).

3.7 Huấn luyện đặc trưng tốt nhất cho 1c

- **Tên hàm:** best_skill_feature_model(table, best_personality_feature)
- **Input:**

  □ table: **DataFrame** chứa toàn bộ dữ liệu để huấn luyện.

  □ best_personality_feature: Tên của feature tốt nhất trong quá trình huấn luyện câu 1c.
- **Output:**

  □ Trọng số của mô hình sau quá trình huấn luyện.

- **Ý tưởng và mô tả:** Tương tự 3.6

3.8 Chuyển các đặc trưng thành mảng mask

- **Tên hàm:** get_masked_array(table, feature)
- **Input:**

  □ table: DataFrame dữ liệu truyền vào nhằm lấy tên các cột.

  □ feature: Các đặc trưng truyền vào để lấy mảng mask.
- **Output:**

  □ Mảng mask chỉ gồm 2 giá trị (True hoặc False) có chiều dài bằng số lượng cột trong **table**.

- **Ý tưởng:**

  □ Tạo 1 mảng có độ dài bằng tổng số lượng cột trong table.

  □ Với mỗi vị trí, kiểm tra xem tên cột đó có tồn tại trong feature không. Nếu có thì đánh dấu bằng True, ngược lại đánh dấu bằng False.
- **Mô tả:**

  □ Lấy tên các cột trong table đem vào list.

  □ Lần lượt kiểm tra tên của các cột theo ý tưởng đã trình bày phía trên.

□ Trả về cho người dùng mảng **numpy.array** có chiều dài bằng số lượng cột.

3.9 Tìm mô hình tốt nhất (cho 1d)

- **Tên hàm:** find_best_model_1d(table, model, clusters = 10, exponent = 2)
- **Input:**

  □ table: **DataFrame** chứa toàn bộ dữ liệu để huấn luyện.

  □ model: Các model được sử dụng để huấn luyện.

  □ clusters: Số lượng tập con được chia ra từ tập **table** được dùng để huấn luyện **(mặc định là 10)**.

  □ exponent: Số mũ của tập dữ liệu (được sử dụng trong quá trình tìm trọng số).

- **Output:**

  □ Trả về lần lượt 2 giá trị.

  □ Giá trị đầu tiên là vị trí của model tốt nhất trong tập model.

  □ Giá trị tiếp theo là mảng MAE trung bình của các model.

- **Ý tưởng:**

  □ Xáo trộn các dòng dữ liệu với nhau.

  □ Chia các dòng dữ liệu ra thành **clusters** tập con.

  □ Lần lượt **chọn từng tập con** làm tập huấn luyện, với mỗi tập ấy ta **huấn luyện cho tất cả các model**. Với mỗi model, sau khi huấn luyện xong, trọng số tìm được qua quá trình huấn kuyện ấy sẽ được dùng để **kiểm tra trên clusters - 1 tập còn lại**.

  □ Sau mỗi lần kiểm tra, các giá trị MAE thu được sẽ được cộng vào mảng tổng (vào vị trí tương ứng của chúng so với vị trí của chúng model được truyền vào).

  □ Khi quá trình kiểm tra kết thúc, mảng tổng trên sẽ được dùng để tính MAE trung bình cũng như vị trí của model có MAE trung bình nhỏ nhất để trả kết quả về người dùng.

- **Mô tả:**

□ Trong quá trình thử nghiệm với các số mũ khác nhau (trong khoảng 1-5), số mũ bằng 2 cho kết quả tương đối tốt. Do đó, chọn số mũ bằng 2 làm mặc đinh.

□ Sử dụng **pandas.DataFrame.sample(frac=1)** như đã trình bày ở phía trên để xáo trộn toàn bộ dữ liệu.

□ Sử dụng **numpy.array_split** để chia mảng ra thành **clusters** tập con.

□ Với cùng một tập dữ liệu sau khi được xáo trộn, cho dù việc sử dụng **numpy.array_split** diễn ra nhiều lần nhưng số tập con cũng như **từng phần tử trong mỗi tập con đều sẽ giống nhau**.

□ Hàm **numpy.array_split** sẽ chia **n // cluster + 1** dòng dữ liệu cho **n % clusters đầu tiên** và **n // clusters** dòng dữ liệu cho các tập còn lại. Do đó, nếu tập ban đầu dùng để chia là không đổi (**trong suốt quá trình huấn luyện chỉ dùng tập đã xáo trộn 1 lần duy nhất)** thì các tập con dù cho được chia bao nhiêu lần nhưng trong tất cả các lần đều dùng cùng 1 clusters đều sẽ cho ra các tập có các phần tử trong tập con là như nhau.

□ Như ý tưởng đã trình bày phía trên, các tập con sẽ lần lượt được duyệt qua và được chọn làm tập để huấn luyện. Với mỗi tập con như thế, tất cả các model sẽ lần lượt được huấn luyện qua và đem đi test với **clusters - 1** tập còn lại nhằm lấy giá trị MAE để cộng vào mảng tổng.

□ Trong suốt quá trình tính toán trên, các hàm đã được cài đặt như **find_weight**, **calc_y_predict** hay **MAE** sẽ được sữ dụng để hỗ trợ.

□ Cuối cùng, nhằm lấy vị trí của model có MAE trung bình nhỏ nhất, hàm **numpy.argmin** sẽ được sử dụng.

4 Đánh giá các mô hình

4.1 Đánh giá mô hình 1a

- **Mô hình:**

$$\begin{aligned}\text{Salary} = &-22756.513 \times Gender &&+804.503 \times 10percentage\\ &+1294.655 \times 12percentage &&+-91781.898 \times CollegeTier\\ &+23182.389 \times Degree &&+1437.549 \times collegeGPA\\ &+-8570.662 \times CollegeCityTier &&+147.858 \times English\\ &+152.888 \times Logical &&+117.222 \times Quant\\ &+34552.286 \times Domain\end{aligned}$$

- **MAE:**

□ Với mô hình 1a, sau khi huấn luyện trên file dữ liệu **train.csv** và chạy trên file dữ liệu **test.csv**, ta thu được MAE tốt nhất là 104863.71390646267 **(với các trọng số sau khi đã làm tròn đến chữ số thứ 3 sau dấu phẩy)**.

4.2 Đánh giá mô hình 1b

- **Minh họa:**

```
                Feature    MAE Average
0      conscientiousness  306519.996380
1          agreeableness  300440.593029
2           extraversion  306979.555265
3            nueroticism  299342.364268
4  openess_to_experience  303460.894237
Best feature: nueroticism
```

□ Phía trên là 1 ví dụ minh họa cho một trong số nhiều lần huấn luyện các đặc trưng.

□ Trong quá trình thực thi, mỗi lần chạy sẽ cho ra mỗi kết quả khác nhau do sự ngẫu nhiên của việc xáo trộn dữ liệu.

- **conscientiousness:**

- Giá trị MAE trung bình cho lần huấn luyện trên là 306519.996380.

- **agreeableness:**
  - Giá trị MAE trung bình cho lần huấn luyện trên là 300440.593029.

- **extraversion:**
  - Giá trị MAE trung bình cho lần huấn luyện trên là 306979.555265.

- **nueroticism:**
  - Giá trị MAE trung bình cho lần huấn luyện trên là 299342.364268.

- **openess_to_experience:**
  - Giá trị MAE trung bình cho lần huấn luyện trên là 303460.894237.

- **Mô hình tốt nhất:**
  - Dựa vào các MAE trung bình trong lần huấn luyện phía trên, giá trị MAE trung bình nhỏ nhất thuộc về **nueroticism**.
  - Do đó, mô hình tốt nhất sẽ được xây dựng dựa trên đặc trưng này.

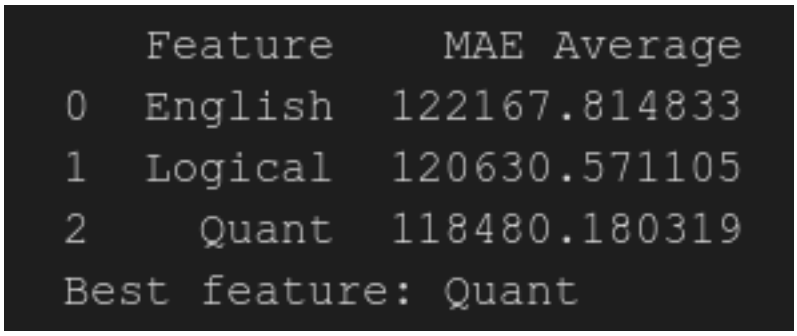
$$\text{Salary} = -56546.304 \times nueroticism$$

- **MAE:**
  - MAE thu được sau khi huấn luyện sau khi huấn luyện trên file dữ liệu **train.csv** và chạy trên file dữ liệu **test.csv** là 291019.6931941854 **(với các trọng số sau khi đã làm tròn đến chữ số thứ 3 sau dấu phẩy)**.

- **Nhận xét và giải thích kết quả:**
  - Dựa trên bài báo sau, những người bị nueroticism (tạm dịch: loạn thần kinh, có vấn đề về thần kinh) sẽ có lương thấp hơn.
  - Việc bị loạn thần kinh hay có vấn đề về thần kinh sẽ gây ảnh hưởng lớn đến hiệu quả làm việc. Những người như việc gây ra những khuynh hướng đáng lo ngại như dễ nóng nảy, khó kiểm soát được cảm xúc, dễ bị tổn thương,... Hay có thể nói cách khác, chỉ số về vấn đề loạn thần kinh càng cao thì mức lương sẽ càng thấp.
  - Những vấn đề này ngoài việc ảnh hưởng đến cá nhân người đó, còn có thể đến những đồng nghiệp xung quanh. Gây nên tình trạng làm giảm hiệu quả làm việc chung.
  - Do đó, đặc trưng nueroticism được cho là đặc trưng tốt nhất trong tập huấn luyện này cũng có thể được coi là hợp lý.

4.3 Đánh giá mô hình 1c

- **Minh họa:**



□ Phía trên là 1 ví dụ minh họa cho một trong số nhiều lần huấn luyện các đặc trưng.

□ Trong quá trình thực thi, mỗi lần chạy sẽ cho ra mỗi kết quả khác nhau do sự ngẫu nhiên của việc xáo trộn dữ liệu.

- **English:**

□ Giá trị MAE trung bình cho lần huấn luyện trên là 122167.814833.

- **Logical:**

□ Giá trị MAE trung bình cho lần huấn luyện trên là 120630.571105.

- **Quant:**

□ Giá trị MAE trung bình cho lần huấn luyện trên là 118480.180319.

- **Mô hình tốt nhất:**

□ Dựa vào các MAE trung bình trong lần huấn luyện phía trên, giá trị MAE trung bình nhỏ nhất thuộc về **Quant**.

□ Do đó, mô hình tốt nhất sẽ được xây dựng dựa trên đặc trưng này.

$$\text{Salary} = 585.895 \times Quant$$

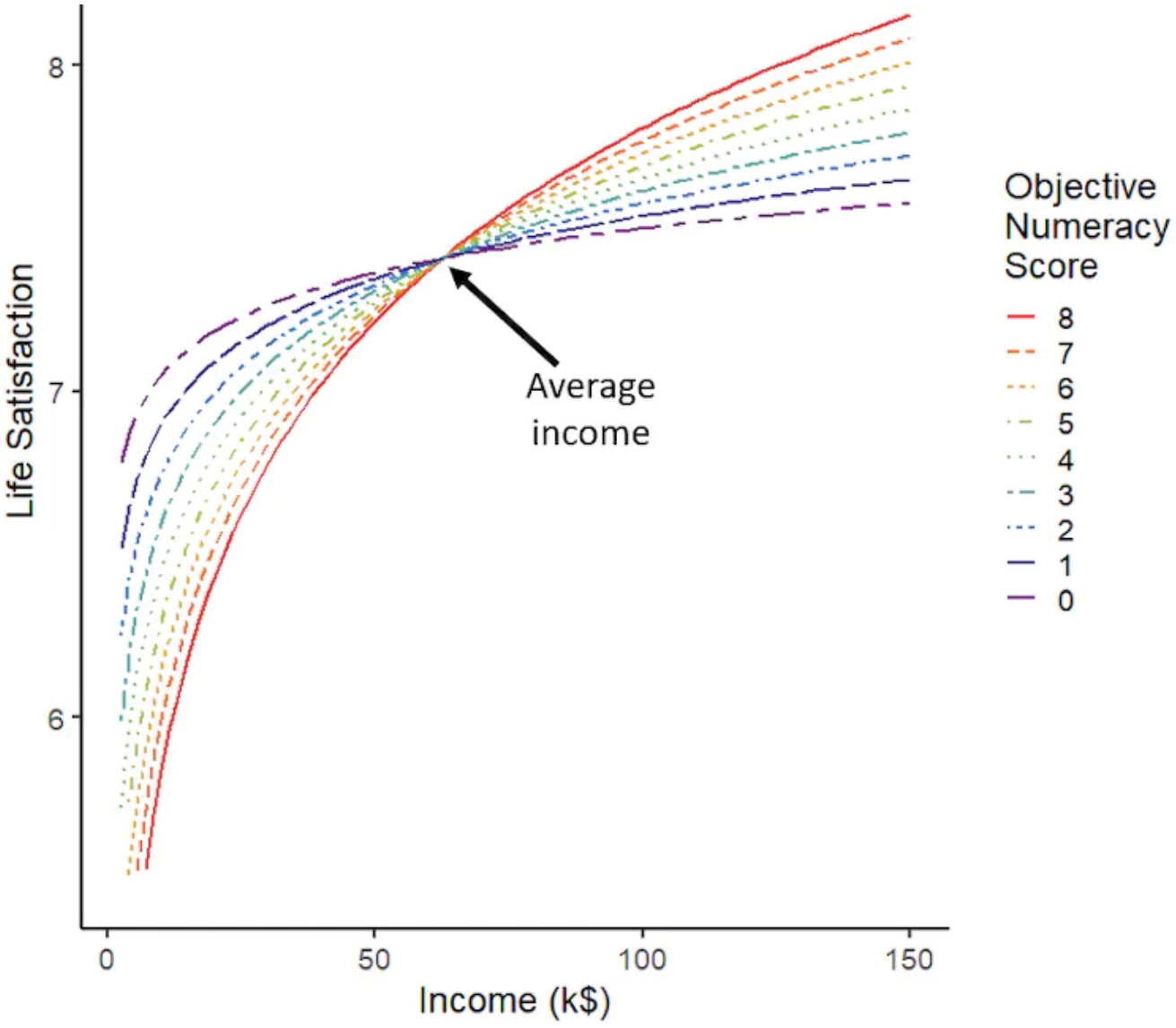
- **MAE:**

□ MAE thu được sau khi huấn luyện sau khi huấn luyện trên file dữ liệu **train.csv** và chạy trên file dữ liệu **test.csv** là 106819.53989333333 **(với các trọng số sau khi đã làm tròn đến chữ số thứ 3 sau dấu phẩy)**.

- **Nhận xét và giải thích kết quả:**

□ Dựa trên bài phân tích sau, ta có thể thấy phần định tính của bài kiểm tra AMCAT thực chất là việc hiểu và giải quyết một hay nhiều vấn đề liên quan đến toán học.

□ Theo nguồn trích dẫn sau, ta có thể thấy rằng điểm của các bài kiểm tra môn toán càng cao thì tiền lương sẽ càng cao như biểu đồ sau đây.



□ Tính từ mức lương trung bình trở lên, ta có thể dễ dàng thấy rằng, mức điểm toán càng cao thì lương sẽ càng cao.

□ Những người học toán tốt đồng thời cũng thường sẽ có khả năng tư duy logic tốt, khả năng ứng biến cũng như tìm ra giải pháp cho vấn đề nhanh. Điều này ảnh hưởng tích cực đến hiệu suất làm việc cũng như tiền lương của họ.

4.4 Đánh giá mô hình 1d:

a Xây dựng mô hình

- **Mô hình 1:**

□ Mô hình được xây dựng dựa trên việc Brute Force hoặc Genetic Algorithm nhằm lấy các MAE nhỏ nhất.

□ Đối với Brute Force cũng như Genetic Algorithm, dùng mảng mask array có độ dài bằng với số đặc trưng. Trong đó, mảng chỉ gồm 2 loại giá trị là **True** hoặc **False** lần lượt đặc trưng cho việc lấy hoặc không lấy đặc trưng đó hay không. Việc sử dụng mask array sẽ giúp cho việc lấy giá trị của các đặc trưng trong khi thực hiện Brute Force cũng như Genetic Algorithm trở nên dễ dàng hơn.

□ Sử dụng Brute Force giúp tìm MAE **nhỏ nhất**. Tuy nhiên, thuật toán này sẽ không đem lại cho người dùng lợi ích về thời gian.

□ Bên cạnh đó, sử dụng Genetic Algorithm chỉ giúp tìm MAE **đủ tốt**. Tuy nhiên, thuật toán này sẽ đem lại lợi ích về thời gian cho người dùng khi so sánh với thuật toán Brute Force.

□ Ngoài ra, trong quá trình thực hiện 2 thực toán trên, ngoài thử nghiệm với số mũ bằng 1, những số mũ khác cũng được sử dụng để có thể tìm ra mô hình tốt nhất về giá trị của MAE.

□ Trong những lần thử nghiệm với 2 thuật toán trên, ta thấy mô hình với các giá trị sau khi được **bình phương** lên (tức wx^2) sẽ cho ra các trọng số đạt MAE đủ tốt.

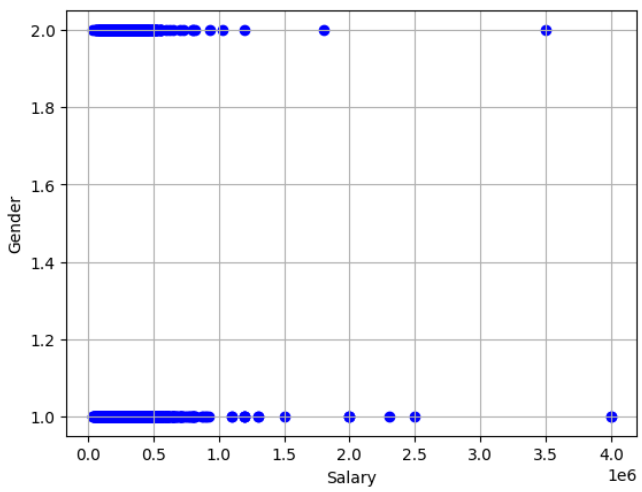
□ Trong khi thực hiện 2 thuật toán trên, các dữ liệu dùng để huấn luyện sẽ không được chia thành nhiều tập con để huấn luyện mà chỉ thực hiện dựa trên mục tiêu đạt MAE nhỏ nhất.

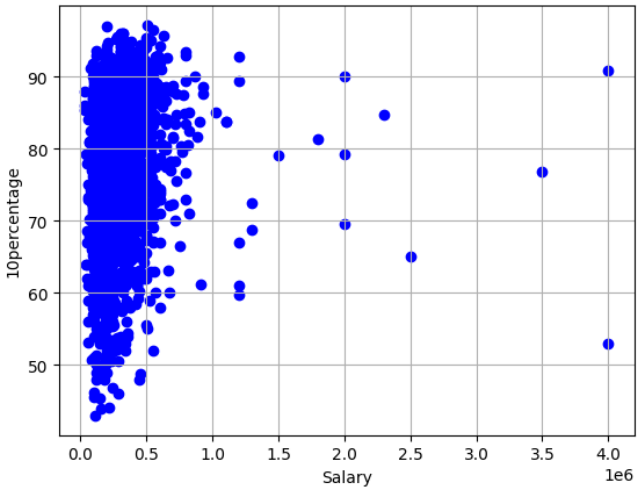
□ File notebook nhằm thực hiện quá trình áp dụng Brute Force để tìm MAE tốt nhất.

□ File notebook nhằm thực hiện quá trình áp dụng Genetic Algorithm để tìm MAE tốt nhất.

□ Từ đó, ta có mô hình 1 gồm **16 đặc trưng** như sau: 10percentage, 12percentage, CollegeTier, collegeGPA, CollegeCityTier, Quant, Domain, ComputerProgramming, ElectronicsAndSemicon, ComputerScience, conscientiousness, ElectricalEngg, CivilEngg, extraversion, nueroticism, openess_to_experience.

- **Mô hình 2:**





□ Dựa vào 2 hình vẽ trên, ta có thể thấy rằng, hình bên phải với đặc trưng là **10percentage** cho ta độ phân hóa cao hơn đặc trưng bên trái **Gender** (tức trải dài trên nhiều miền giá trị của đặc trưng hơn).

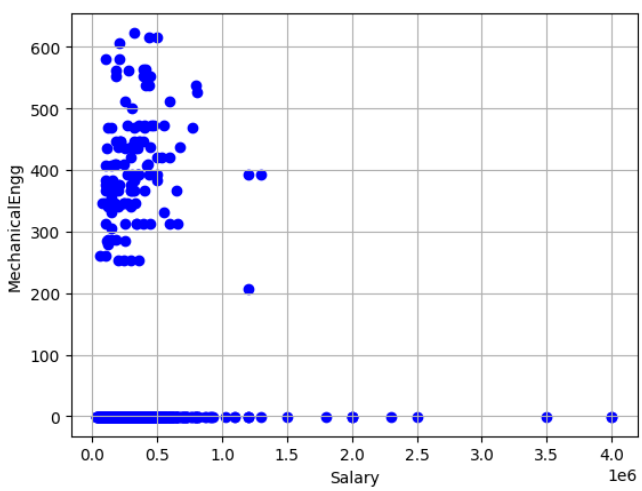
□ Cũng có thể nói theo nghĩa nào đó, sự phân hóa tốt hơn của đặc trưng 12percentage này giúp cho mô hình dự đoán y trở nên chính xác hơn.

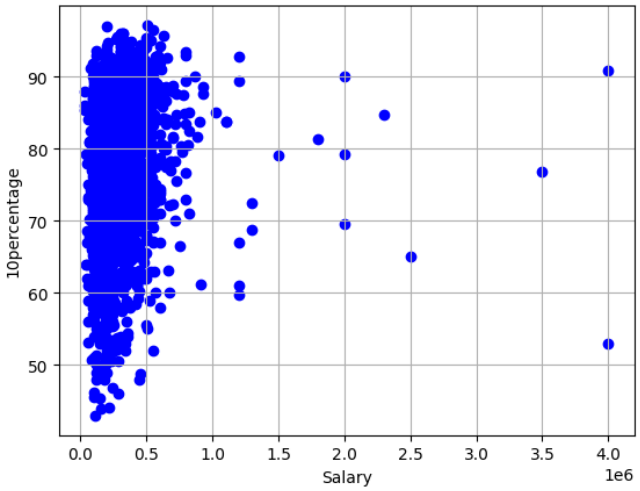
□ Lấy ví dụ, nếu chọn mô hình dựa trên Gender, ta chỉ có 2 sự phân hóa là nam và nữ. Lúc này, về mặt trực quan chúng ta có thể thấy rằng, nếu đưa ra một lựa chọn là nam hoặc nữ, việc dự đoán tiền lương trở nên vô cùng khó khăn hay cũng có thể nói là bất khả thi.

□ Tuy nhiên, nếu lấy 1 đặc trưng có khả năng phân hóa, dàn trải tốt. Về mặt trực quan, nếu đưa chúng ta 1 số điểm, chúng ta có thể khoanh vùng được số tương lương thành một tập hợp nhỏ hơn. Hoặc sẽ có cơ may chúng ta có thể kết luận ngay lập tức số tiền lương (nếu trong nhóm điểm đó chỉ có 1 phần tử).

□ Từ đó, ta có mô hình 2 gồm **19 đặc trưng** như sau: 10percentage, 12percentage, collegeGPA, English, Logical, Quant, Domain, ComputerProgramming, ElectronicsAndSemicon, ComputerScience, MechanicalEngg, ElectricalEngg, TelecomEngg, CivilEngg, conscientiousness, agreeableness, nueroticism, extraversion, openess_to_experience.

- **Mô hình 3:**





□ Mô hình 3 này cũng tương tư như mô hình 2 trên.

□ Tuy nhiên, đối với những tập hợp phân hóa không mạnh như hình bên trái phía trên là biểu đồ của đặc trưng MechanicalEngg so với Salary, chúng ta có thể xem chúng như trường hợp phân hóa không mạnh như ở mô hình và loại chúng ra khỏi mô hình.

□ Theo một cách nào đó, nếu chúng chọn ngẫu nhiên 1 điểm số của tập MechanicalEngg, rất có khả năng cao điểm chúng ta cần tìm sẽ thuộc vào tập hợp có nhiều phần tử hơn (tập hợp nằm ngang ở phía dưới cùng của hình bên trái). Khi đó, sự phân hóa mạnh như cách chúng ta muốn khi chọn mô hình sẽ không còn. Như trên hình bên trái ta có thể chia thành 2 tập hợp là tập có điểm trên 100 và tập có điểm dưới 100.

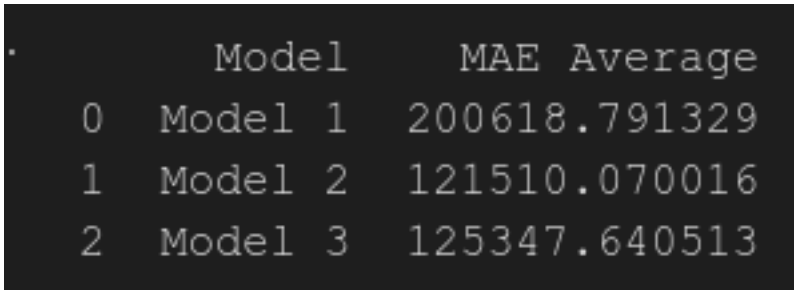
□ Từ những lập luận nêu trên, bỏ các đặc trưng có biểu đồ như hình bên trái cũng là một phương án hợp lý để chọn ra mô hình tối ưu.

□ Từ đó, ta có mô hình 3 gồm **13 đặc trưng** như sau: 10percentage, 12percentage, collegeGPA, English, Logical, Quant, Domain, ComputerProgramming, conscientiousness, agreeableness, extraversion, nueroticism, openess_to_experience.

b Chọn số mũ

- Trong quá trình Brute Force hoặc Genetic Algorithm để tìm MAE nhỏ nhất với các số mũ của bảng dữ liệu thuộc trong khoảng từ 1 đến 5 (tức wx^k), ta có thể thấy rằng chọn số mũ bằng 2 sẽ cho ra kết quả có thể chấp nhận được.
- Ở đây, có thể chấp nhận được được hiểu theo nghĩa là có MAE đủ nhỏ.

c Minh họa



	Model	MAE Average
0	Model 1	200618.791329
1	Model 2	121510.070016
2	Model 3	125347.640513

- **Lưu ý:**

  □ Phía trên là 1 ví dụ minh họa cho một trong số nhiều lần huấn luyện các mô hình.

  □ Trong quá trình thực thi, mỗi lần chạy sẽ cho ra mỗi kết quả khác nhau do sự ngẫu nhiên của việc xáo trộn dữ liệu.

- **Mô hình 1:**

  □ Giá trị MAE trung bình cho lần huấn luyện trên là 200618.791329.

- **Mô hình 2:**

  □ Giá trị MAE trung bình cho lần huấn luyện trên là 121510.070016.

- **Mô hình 3:**

  □ Giá trị MAE trung bình cho lần huấn luyện trên là 125347.640513.

- **Mô hình tốt nhất:**

  □ Dựa vào các MAE trung bình trong lần huấn luyện phía trên, giá trị MAE trung bình nhỏ nhất thuộc về **mô hình 2**.

□ Do đó, mô hình tốt nhất sẽ được xây dựng trên đặc trưng này.

$$
\begin{aligned}
\text{Salary} = & \\
&+6.609 \times 10 percentage^2 + 6.147 \times 12 percentage^2 \\
&+12.691 \times collegeGPA^2 + 0.168 \times English^2 \\
&+0.179 \times Logical^2 + 0.125 \times Quant^2 \\
&+37377.083 \times Domain^2 + 0.225 \times ComputerProgramming^2 \\
&+-0.022 \times ElectronicsAndSemicon^2 + -0.351 \times ComputerScience^2 \\
&+0.178 \times MechanicalEngg^2 + -0.247 \times ElectricalEngg^2 \\
&+-0.123 \times TelecomEngg^2 + 0.516 \times CivilEngg^2 \\
&+11772.488 \times conscientiousness^2 + -5288.306 \times agreeableness^2 \\
&+-2374.622 \times extraversion^2 + -5014.033 \times nueroticism^2 \\
&+745.447 \times openess\_to\_experience^2
\end{aligned}
$$

d MAE

- MAE thu được sau khi huấn luyện trên file dữ liệu **train.csv** và chạy trên file dữ liệu **test.csv** là 101484.43088949646 **(với các trọng số sau khi đã làm tròn đến chữ số thứ 3 sau dấu phẩy)**.

5 References

- Cross-Validation
- Tài liệu về 'numpy.array_split'
- Tài liệu về 'pandas.DataFrame.sample'
- Genetic Algorithm
- Neurotic Personalities Earn Lower Salaries
- AMCAT aptitude questions
- The better you are at math, the more money seems to influence your satisfaction