UNIVERSITY OF SCIENCE

VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY

Báo cáo đồ án
Linear Regression

TOÁN ỨNG DỤNG VÀ THỐNG KÊ CHO CNTT

Giảng viên lý thuyết Vũ Quốc Hoàng
Lê Thanh Tùng

Giảng viên thực hành Phan Thị Phương Uyên
Nguyễn Văn Quang Huy

Sinh viên thực hiện Nguyễn Quốc Huy 21127511

Mục lục

1 LỜI NÓI ĐẦU

- Đây là **đồ án 3 - Linear Regression** môn **Toán Ứng dụng và Thống kê cho Công nghệ Thông tin**, khoa Công nghệ Thông tin, trường Đại học Khoa học Tự nhiên - Đại học Quốc Gia Hồ Chí Minh. Được thực hiện bởi sinh viên **Nguyễn Quốc Huy, MSSV 21127511**.
- Toàn bộ giải thích, ứng dụng, ví dụ, kết quả và tham khảo về đồ án đều được nêu chi tiết và đầy đủ qua báo cáo.
- **Giáo viên giảng dạy:**
  - Vũ Quốc Hoàng
  - Lê Thanh Tùng
  - Phan Thị Phương Uyên
  - Nguyễn Văn Quang Huy

2 MỨC ĐỘ HOÀN THÀNH

Bài toán	Phần trăm	Mô tả kết quả hoàn thành
1a	100%	- Huấn luyện 1 lần duy nhất cho 11 đặc trưng nói trên cho toàn bộ tập huấn luyện (**train.csv**). Thể hiện công thức cho mô hình hồi quy (tính y theo 11 đặc trưng trên) . Báo cáo 1 kết quả trên tập kiểm tra (**test.csv**) cho mô hình vừa huấn luyện được
1b	100%	- Sử dụng **k-fold Cross Validation** (**k** là 10) để tìm ra đặc trưng tốt nhất trong các đặc trưng tính cách. Báo cáo **5 kết quả tương ứng cho 5 mô hình** từ k-fold Cross Validation (lấy trung bình). Thể hiện công thức cho mô hình hồi quy theo đặc trưng tốt nhất (tính y theo đặc trưng tốt nhất tìm được). Báo cáo **1 kết quả trên tập kiểm tra (test.csv)** cho mô hình với đặc trưng tốt nhất tìm được
1c	100%	- Sử dụng **k-fold Cross Validation** (**k** là 10) để tìm ra đặc trưng tốt nhất. Báo cáo **3 kết quả tương ứng cho 3 mô hình** từ k-fold Cross Validation (lấy trung bình). Thể hiện công thức cho mô hình hồi quy theo đặc trưng tốt nhất (tính y theo đặc trưng tốt nhất tìm được). Báo cáo **1 kết quả trên tập kiểm tra (test.csv)** cho mô hình với đặc trưng tốt nhất tìm được
1d	100%	- Sử dụng **k-fold Cross Validation** (**k** là 10) để tìm ra mô hình tốt nhất. Bảo cáo **3 kết quả tương ứng với 3 mô hình** từ k-fold Cross Validation. Nghiên cứu dữ liệu, lựa chọn đặc trưng từ đó có được 3 mô hình. Giải thích ý nghĩa của mô hình, nhận xét sự hiệu quả của mô hình. Báo cáo **1 kết quả trên tập kiểm tra test.csv** cho mô hình với đặc trưng tốt nhất tìm được.

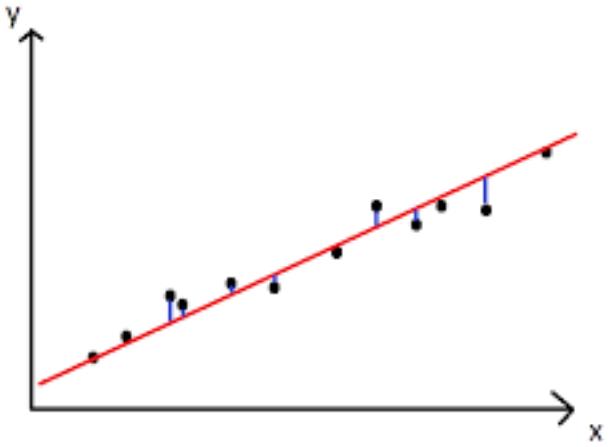
3 GIỚI THIỆU VỀ BÀI TOÁN

3.1 Hồi quy tuyến tính

- **"Hồi quy tuyến tính"** [4] là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Nói cách khác **"Hồi quy tuyến tính"** là một phương pháp để dự đoán biến phụ thuộc (**Y**) dựa trên giá trị của biến độc lập (**X**). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục. Ví

dụ, dự đoán giao thông ở một cửa hàng bán lẻ, dự đoán thời gian người dùng dừng lại một trang nào đó hoặc số trang đã truy cập vào một website nào đó v.v...

- Trong khi sử dụng **"Hồi quy tuyến tính"** [4], mục tiêu của chúng ta là để làm sao một đường thẳng có thể tạo được sự phân bố gần nhất với hầu hết các điểm. Do đó làm giảm khoảng cách (sai số) của các điểm dữ liệu cho đến đường đó.



- Ví dụ, ở các điểm ở hình trên biểu diễn các điểm dữ liệu khác nhau và đường thẳng đại diện cho một đường gần đúng có thể giải thích mối quan hệ giữa các trục x & y. Thông qua, **"Hồi quy tuyến tính"** chúng ta cố gắng tìm ra một đường như vậy.
- Đường hồi quy luôn luôn đi qua trung bình của biến độc lập (**x**) cũng như trung bình của biến phụ thuộc (**y**).
- Đường hồi qui tối thiểu hóa tổng của "Diện tích các sai số". Đó là lý do tại sao phương pháp **"Hồi quy tuyến tính"** được gọi là **"Ordinary Least Square (OLS)"**.
- Ví dụ, nếu chúng ta có một biến phụ thuộc Y và một biến độc lập X - mối quan hệ giữa X và Y có thể được biểu diễn dưới dạng phương trình sau:

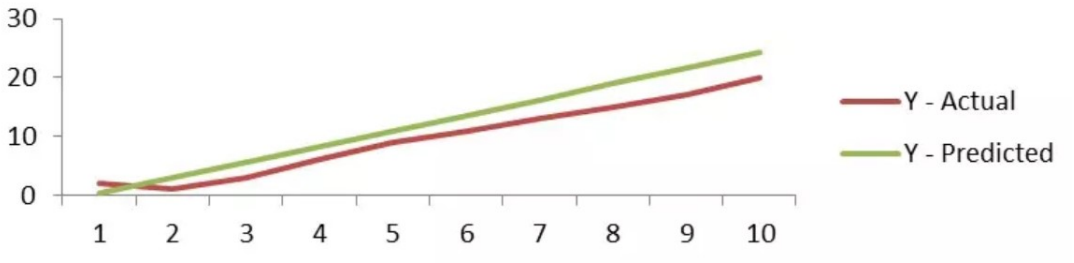
```
Y = B0 + B1*X
```

- Trong đó:
  - Y là biến phụ thuộc
  - X là biến độc lập

  - B0 là hằng số
  - B1 là hệ số mối tương quan giữa X và Y

- Một khi xây dựng mô hình, câu hỏi tiếp theo đến trong đầu là để biết liệu mô hình đó có đủ để dự đoán trong tương lai hoặc là mối quan hệ đã xây dựng giữa các biến phụ thuộc và độc lập là đủ hay không.

- Để đánh giá độ chính xác của mô hình hồi quy, chúng ta cần sử dụng các chỉ số đánh giá như **R-squared**, Root Mean Square Error **(RMSE)**, Mean Absolute Error **(MAE)**, và Mean Absolute Percentage Error **(MAPE)**. Chỉ số R-squared thể hiện tỷ lệ phương sai của biến phụ thuộc được giải thích bởi các biến độc lập. Trong khi đó, các chỉ số **RMSE**, **MAE** và **MAPE** đánh giá sự khác biệt giữa giá trị dự đoán và giá trị thực tế.



- Chỉ số cơ bản đầu tiên chúng ta cần biết để đánh giá một mô hình đó là **MSE**. MSE được gọi nôm na là giá trị sai số bình phương trung bình hoặc là lỗi bình phương trung bình. Vấn đề khi nói về sai số trung bình của một mô hình thống kê nhất định là rất khó xác định mức độ lỗi là do mô hình và mức độ là do ngẫu nhiên. Lỗi bình phương trung bình (MSE) cung cấp một thống kê cho phép các nhà nghiên cứu đưa ra tuyên bố như vậy. MSE chỉ đơn giản đề cập đến giá trị trung bình của chênh lệch bình phương giữa tham số dự đoán và tham số quan sát được. Ta có công thức như sau:

$$MSE = \frac{\sum(f_i - y_i)^2}{N}$$

- Tiếp theo, chúng ta có RMSE. Lỗi trung bình bình phương (RMSE) là độ lệch chuẩn của phần dư (lỗi dự đoán). Phần dư là thước đo khoảng cách từ các điểm dữ liệu đường hồi quy; RMSE là thước đo mức độ lan truyền của những phần dư này. Nói cách khác, nó cho biết mức độ tập trung của dữ liệu xung quanh dòng phù hợp nhất . Lỗi bình phương trung bình thường được sử dụng trong khí hậu học, dự báo và

phân tích hồi quy để xác minh kết quả thí nghiệm. Lỗi trung bình bình phương gốc (RMSE) là thước đo mức độ hiệu quả của mô hình. Nó thực hiện điều này bằng cách đo sự khác biệt giữa các giá trị dự đoán và giá trị thực tế . RMSE càng nhỏ tức là sai số càng bé thì mức độ ước lượng cho thấy độ tin cậy của mô hình có thể đạt cao nhất. Công thức như sau:

$$RMSE = \sqrt{\frac{\sum_i^n (f_i - y_i)^2}{n}}$$

- Chúng ta sang chỉ tiêu cần thiết để áp dụng cho môn học, MAE. MAE cũng đo lường sai số trung bình của mô hình so với dữ liệu thực tế, tuy nhiên MAE tính toán trung bình giá trị tuyệt đối của sai số. MAE có ưu điểm là đơn vị tính của nó tương tự với đơn vị của biến phụ thuộc, giúp dễ dàng so sánh giữa các mô hình và giữa các biến phụ thuộc khác nhau. Có công thức như sau:

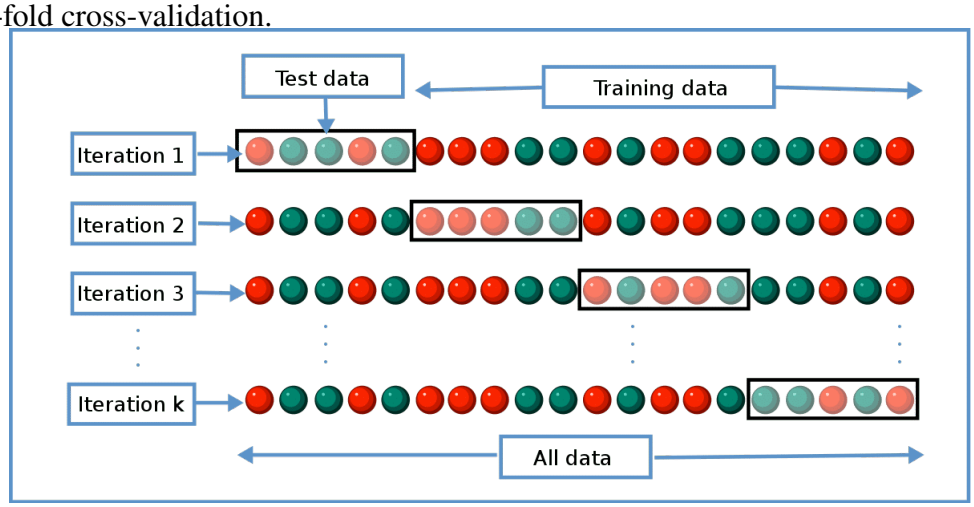
$$MAE = \frac{\sum abs(f_i - y_i)}{n}$$

- Trong đó:
  - n: số lượng quan sát trong mẫu.
  - fi: giá trị thực tế của biến phụ thuộc của quan sát i.
  - yi: giá trị dự đoán của biến phụ thuộc của quan sát i.

3.2 K-fold Cross-Validation

- **Cross validation** [10] là một phương pháp thống kê được sử dụng để ước lượng hiệu quả của các mô hình học máy. Nó thường được sử dụng để so sánh và chọn ra mô hình tốt nhất cho một bài toán. Kỹ thuật này dễ hiểu, dễ thực hiện và cho ra các ước lượng tin cậy hơn so với các phương pháp khác.
- **Cross validation** [11] là một kỹ thuật lấy mẫu để đánh giá mô hình học máy trong trường hợp dữ liệu không được dồi dào cho lắm. Tham số quan trọng trong kỹ thuật này là k, đại diện cho số nhóm mà dữ liệu sẽ được chia ra. Vì lý do đó, nó được mang tên k-fold cross-validation. Khi giá trị của k được lựa chọn, người ta sử dụng trực tiếp giá trị đó trong tên của phương pháp đánh giá. Ví dụ với k=10, phương pháp sẽ mang

tên 10-fold cross-validation.



- Kỹ thuật này thường bao gồm các bước như sau:
  - Xáo trộn dataset một cách ngẫu nhiên
  - Chia dataset thành k nhóm, với mỗi nhóm:
    - Sử dụng nhóm hiện tại để đánh giá hiệu quả mô hình
    - Các nhóm còn lại được sử dụng để huấn luyện mô hình
    - Huấn luyện mô hình
    - Đánh giá và sau đó hủy mô hình
  - Tổng hợp hiệu quả của mô hình dựa từ các số liệu đánh giá
- Một lưu ý quan trọng là mỗi mẫu chỉ được gán cho duy nhất một nhóm và phải ở nguyên trong nhóm đó cho đến hết quá trình. Các tiền xử lý dữ liệu như xây dựng vocabulary chỉ được thực hiện trên tập huấn luyện đã được chia chứ không được thực hiện trên toàn bộ dataset.
- Kết quả tổng hợp thường là trung bình của các lần đánh giá. Ngoài ra việc bổ sung thông tin về phương sai và độ lệch chuẩn vào kết quả tổng hợp cũng được sử dụng trong thực tế.

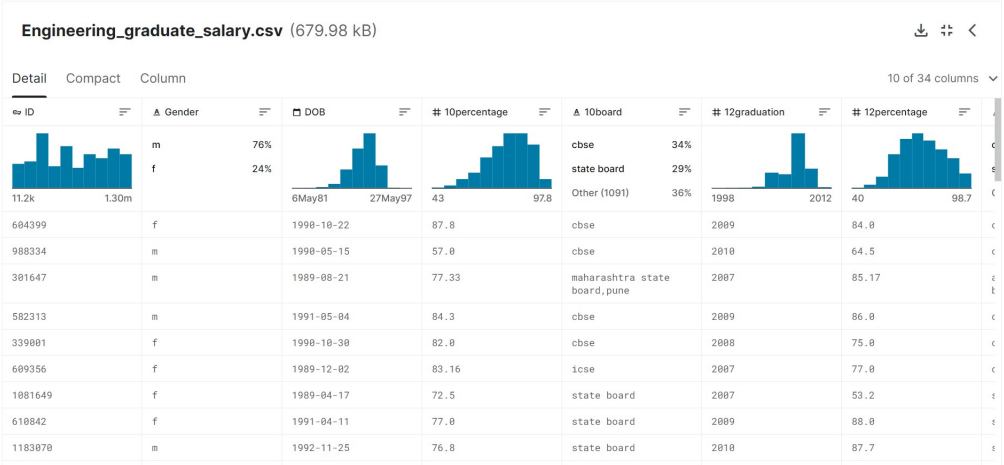
3.2.1 Cấu hình k

- Như đã nêu, k là thông số quan trọng để có thể đánh giá chính xác một mô hình. Nên việc lựa chọn k sao cho hợp lý với mô hình cũng vô cùng quan trọng. [11]
- Có các cách thường sử dụng để lựa chọn thông số k như sau:

- 1/ Giá trị của k được lựa chọn sao cho mỗi tập train, mỗi tập test đủ lớn, để ta có thể thống kê cho những dataset chứa nó.
- 2/ Giá trị của k được gán cố định bằng n, với n là kích thước của dataset, và như vậy mỗi mẫu sẽ được sử dụng để đánh giá mô hình một lần. Cách tiếp cận này có tên là **leave-one-out cross-validation**.
- 3/ Giá trị của k được gán cố định bằng 10, một giá trị thường được sử dụng và được chứng minh là cho sai số nhỏ, phương sai thấp (thông qua thực nghiệm). Giá trị k = 10 là một giá trị rất phổ biến, và trong đồ án này, ta sẽ áp dụng phương pháp chọn K này.

3.3 Dữ liệu Mức lương kỹ sư tốt nghiệp đại học

- Chúng ta sẽ làm quen với dữ liệu dạng bản. [3]
- 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
- Bộ dữ liệu có 2988 dòng và 38 cột, sau khi trải qua các bước tiền xử lý, dữ liệu mới có 2998 dòng dữ liệu và 24 cột dữ liệu gồm:
  - 1 giá trị mục tiêu (y): **Salary**
  - 23 đặc trưng gồm: **Gender, 10percentage, 12percentage, CollegeTier, Degree, collegeGPA, CollegeCityTier, English, Logical, Quant, Domain, ComputerProgramming, ElectronicsAndSemicon, ComputerScience, MechanicalEngg, ElectricalEngg, TelecomEngg, CivilEngg, conscientiousness, agreeableness, extraversion, nueroticism, openess_to_experience**

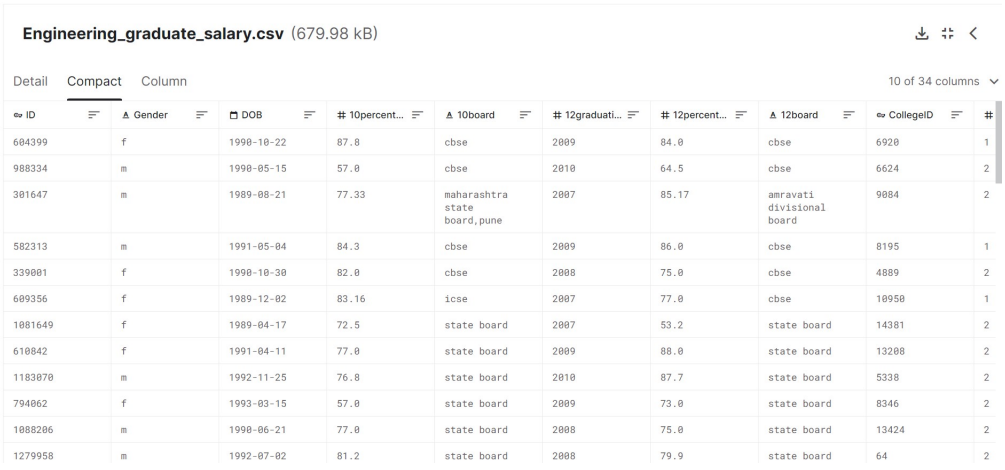


Engineering_graduate_salary.csv (679.98 kB)

Detail Compact Column 10 of 34 columns

ID	Gender	DOB	10percentage	10board	12graduation	12percentage	
11.2k – 1.30m	m 76% / f 24%	6May81 – 27May97	43 – 97.8	cbse 34% / state board 29% / Other (1091) 36%	1998 – 2012	40 – 98.7	
604399	f	1990-10-22	87.8	cbse	2009	84.0	
988334	m	1990-05-15	57.0	cbse	2010	64.5	
301647	m	1989-08-21	77.33	maharashtra state board,pune	2007	85.17	
582313	m	1991-05-04	84.3	cbse	2009	86.0	
339001	f	1990-10-30	82.0	cbse	2008	75.0	
609356	f	1989-12-02	83.16	icse	2007	77.0	
1081649	f	1989-04-17	72.5	state board	2007	53.2	
610842	f	1991-04-11	77.0	state board	2009	88.0	
1183070	m	1992-11-25	76.8	state board	2010	87.7	

- Trong đồ án cung cấp 2 tập tin tương ứng với 2 bộ dữ liệu:
  - **train.csv**: Chứa 2248 mẫu dùng để huấn luyện mô hình
  - **test.csv**: Chứa 750 mẫu dùng để kiểm tra mô hình



Engineering_graduate_salary.csv (679.98 kB)

Detail Compact Column 10 of 34 columns

ID	Gender	DOB	10percent...	10board	12graduati...	12percent...	12board	CollegeID	
604399	f	1990-10-22	87.8	cbse	2009	84.0	cbse	6920	1
988334	m	1990-05-15	57.0	cbse	2010	64.5	cbse	6624	2
301647	m	1989-08-21	77.33	maharashtra state board,pune	2007	85.17	amravati divisional board	9084	2
582313	m	1991-05-04	84.3	cbse	2009	86.0	cbse	8195	1
339001	f	1990-10-30	82.0	cbse	2008	75.0	cbse	4889	2
609356	f	1989-12-02	83.16	icse	2007	77.0	cbse	10950	1
1081649	f	1989-04-17	72.5	state board	2007	53.2	state board	14381	2
610842	f	1991-04-11	77.0	state board	2009	88.0	state board	13208	2
1183070	m	1992-11-25	76.8	state board	2010	87.7	state board	5338	2
794062	f	1993-03-15	57.0	state board	2009	73.0	state board	8346	2
1088206	m	1990-06-21	77.0	state board	2008	75.0	state board	13424	2
1279958	m	1992-07-02	81.2	state board	2008	79.9	state board	64	2

- 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.

4 MÔI TRƯỜNG TIẾN HÀNH

- Chương trình được thực hiện trên **Jupyter Notebook**.
- Các thư viện đã được sử dụng trong bài:
  - import numpy as np
  - import pandas as pd
  - import matplotlib.pyplot as plt
  - import seaborn as sns

5 CÁC HÀM CHỨC NĂNG

5.1 Lớp OLSLinearRegression

- Đây là lớp được cài đặt sẵn trong tiết Lab của cô Uyên. [12]
- Lớp OLSLinearRegression có nhiệm vụ huấn luyện một mô hình hồi quy tuyến tính sử dụng phương pháp bình phương tối tiểu. Mục tiêu là tìm các hệ số (weights) sao cho tổng bình phương sai biệt giữa các giá trị dự đoán và giá trị thực tế là nhỏ nhất.
- **Input:** Các phương thức của lớp này yêu cầu các biến đầu vào**X** và **y**. **X** là ma trận đầu vào chứa các giá trị của các biến độc lập, trong đó mỗi hàng của ma trận đại diện cho một quan sát và mỗi cột đại diện cho một biến độc lập. **y** là một vector chứa các giá trị của biến phụ thuộc tương ứng với các quan sát trong **X**.
- **Output:** Các phương thức của lớp này trả về một mô hình hồi quy tuyến tính đã được train (fit). Hàm **get_params()** trả về vector hệ số hồi quy (weights), và **hàm predict(X)** trả về các giá trị dự đoán của biến phụ thuộc dựa trên mô hình đã học.
- Phương thức **fit(self, X, y):**
  - **X_pinv = np.linalg.inv(X.T @ X) @ X.T**: Tính ma trận giả nghịch đảo của ma trận **X.T @ X**, sau đó nhân với ma trận **X.T** để tính giả nghịch đảo **Penrose-Moore (pseudo-inverse)** của ma trận **X**. Thay vì sử dụng **np.linalg.inv**, ta còn có thể sử dụng **np.linalg.pinv(X)** để tính giả nghịch đảo bảo toàn cả trường hợp nghịch đảo không tồn tại hoặc không ổn định.
  - **self.w = X_pinv @ y**: Tính vector hệ số hồi quy bằng cách nhân giả nghịch đảo của **X** với vector **y**.

- Phương thức **get_params(self):**
  - **return self.w**: Trả về vector hệ số hồi quy đã được học.
- Phương thức **predict(self, X):**
  - **return np.sum(self.w.ravel() * X, axis=1)**: Tính toán giá trị dự đoán bằng cách nhân từng giá trị trong vector hệ số hồi quy với tương ứng các giá trị trong ma trận đầu vào **X**. Hàm **ravel()** dùng để biến đổi ma trận hệ số hồi quy thành vector 1D. Hàm **np.sum** sau đó tính tổng các tích của các phần tử tương ứng trong hai vector để có kết quả dự đoán.

5.2 Hàm mae

5.2.1 Mô tả ý tưởng hàm

- Đây là hàm được cài đặt sẵn trong tiết Lab của cô Uyên. [**?**]
- **Input:** Hàm này yêu cầu hai tham số đầu vào.
  - **y**: Là vector chứa các giá trị thực tế (ground truth).
  - **y_hat**: Là vector chứa các giá trị dự đoán tương ứng.
- **Output:** Hàm trả về một giá trị là sai số trung bình tuyệt đối giữa **y** và **y_hat**. [5]
- **Mục đích:** tính **sai số trung bình tuyệt đối (MAE)** giữa các giá trị dự đoán và giá trị thực tế. Như đã nêu ở phần trên, **MAE** là một phép đo sai số thường được sử dụng trong các bài toán dự đoán để đo lường độ lệch trung bình giữa các giá trị dự đoán và giá trị thực tế. Giá trị **MAE** càng nhỏ thì mô hình dự đoán càng chính xác.

5.2.2 Mô tả thuật toán

- Ta sử dụng **np.mean()** để tính giá trị trung bình của vector sai số tuyệt đối.
- Ta cần biết được mức độ sai lệch giữa dự đoán và thực tế, vì vậy cần tính giá trị tuyệt đối của sự sai khác giữa giá trị thực tế và giá trị dự đoán. Hay còn có thể hiểu là

```
1        np.abs(y.ravel() - y_hat.ravel())
```

5.3 Hàm custom_kfold

5.3.1 Mô tả ý tưởng hàm

- **Input:**
  - **num_splits**: Số lượng fold mà muốn chia dữ liệu thành.
  - **data_length**: Số lượng điểm dữ liệu trong tập dữ liệu ban đầu.
- **Output:** Hàm sẽ tạo ra các chỉ mục (indices) cho tập huấn luyện và tập validation dựa trên số lượng fold và tổng số lượng điểm dữ liệu. [10] [11]
- **Mục đích:** Hàm này liên quan đến kiến thức về phương pháp **cross-validation** trong machine learning đã nêu ở phần trên, đặc biệt là phân chia dữ liệu thành các fold để đào tạo và đánh giá mô hình một cách công bằng. Cách thức triển khai của hàm dựa trên ý tưởng cơ bản của **cross-validation** và sử dụng mảng indices để chia dữ liệu một cách ngẫu nhiên và đảm bảo không có sự trùng lặp trong việc chọn chỉ mục cho các fold.

5.3.2 Mô tả thuật toán

- Đầu tiên, ta sẽ tạo ra một mảng indices từ 0 đến kích thước tập dữ liệu -1. Mỗi phần tử trong mảng đại diện cho một điểm dữ liệu trong tập dữ liệu ban đầu. Để có được điều này, ta có cú pháp như sau:

```
1        indices = np.arange( data_length )
```

- Như đã đề cập ở phần trên về các bước thực hiện **k fold cross validation**, bước tiếp theo ta sẽ tiến hành xáo trộn ngẫu nhiên các chỉ mục trong mảng **indices**. Việc xáo trộn này nhằm đảm bảo tính ngẫu nhiên và cân bằng trong việc chia dữ liệu vào các fold.
- Bước tiếp theo, ta sẽ tính kích thước mỗi fold dựa trên tổng số lượng điểm dữ liệu chia cho số lượng fold mong muốn.
- Và sau cùng, ta sẽ tiến hành chia tập dữ liệu ban đầu thành các fold (phân đoạn) khác nhau để thực hiện phương pháp **cross-validation**. Mỗi lần lặp của vòng for tạo ra một cặp chỉ mục cho tập huấn luyện và tập **validation** cho một fold cụ thể. Số lần lặp của vòng for là bằng **num_splits** (là số lượng fold muốn tạo).
- Trong hàm này, có sử dụng **yield**. **yield** tạo một generator, cho phép lặp qua các cặp chỉ mục một cách lần lượt mà không cần lưu trữ tất cả cặp chỉ mục trong bộ nhớ.

5.4 Hàm custom_cross_validation_mae

5.4.1 Mô tả ý tưởng hàm

- **Input:**
  - **datalength:** Số lượng điểm dữ liệu trong tập dữ liệu.
  - **x:** Ma trận chứa các giá trị của các biến độc lập.
  - **y:** Vector chứa các giá trị của biến phụ thuộc.
  - **k_fold:** Số lượng fold trong cross-validation (mặc định là 10).
- **Output:** Hàm trả về giá trị **MAE** tổng cộng trên tất cả các fold.

5.4.2 Mô tả thuật toán

- Hàm này ta áp dụng các cài đặt từ hàm **custom_kfold** và lớp **OLSLinearRegression** để tiến hành thực hiện phương pháp **cross-validation** để huấn luyện và đánh giá mô hình hồi quy tuyến tính trên cá fold khác nhau. [10] [11]
- Các bước thực hiện như sau:
- Trích xuất dữ liệu tương ứng với các chỉ mục của tập huấn luyện và tập validation.
- Đào tạo mô hình hồi quy tuyến tính trên tập huấn luyện của fold hiện tại.
- Dự đoán giá trị biến phụ thuộc trên tập validation bằng mô hình đã đào tạo.
- Tính **MAE** giữa giá trị thực tế và giá trị dự đoán trên tập validation của fold hiện tại.

Cộng giá trị **MAE** của fold hiện tại vào tổng **MAE**.

5.5 Hàm custom_cross_validation_model

5.5.1 Mô tả ý tưởng hàm

- Hàm này ta có ý tưởng triển khai tương tự như hàm **custom_cross_validation_mae**.
- Điểm khác biệt là hàm này sẽ thực hiện phương pháp **cross-validation** giữa các model với nhau từ đó cho ra kết quả của từng model xem model nào là tốt nhất.

5.5.2 Mô tả thuật toán

- Ta triển khai và cài đặt tương tự hàm **custom_cross_validation_mae**. Ngoài ra, vì giữa các model nên ta sẽ có một list toàn cục **models** lưu danh sách tất cả các mô hình cần được so sánh. [10] [11]

5.6 Hàm detect_outliers

5.6.1 Mô tả ý tưởng hàm

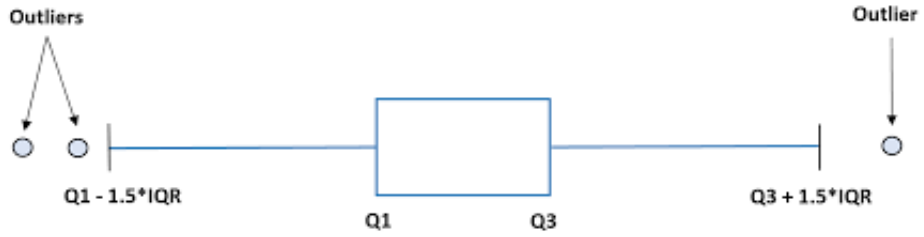
- Hàm này liên quan đến kiến thức về phát hiện và xử lý cá điểm ngoại lai trong dữ liệu. Áp dụng kiến thức xác suất thống kê **khoảng tứ phân vị (IQR)** và ngưỡng trên/dưới để xác định các giá trị ngoại lai. Kiến thức về thống kê và xử lý dữ liệu là quan trọng khi làm việc với dữ liệu thực tế trong lĩnh vực khoa học dữ liệu và machine learning. [13] [14]
- **Input:** Đầu vào là DataFrame chứa dữ liệu cần kiểm tra các điểm ngoại lai.
- **Ouput:** Hàm trả về một DataFrame chứa các thông tin về phần trăm các điểm ngoại lai xuất hiện tương ứng với từng cột, từng đặc trưng trong dữ liệu.
- Để dễ nhận biết hơn, ta sẽ sắp xếp phần trăm các thuộc tính có điểm ngoại lai theo thứ tự giảm dần.
- Đây là hàm được tham khảo qua nghiên cứu từ một học viên trên Kaggle.

5.6.2 Mô tả thuật toán

- Đầu tiên, ta cần phân biệt đâu là kiểu thuộc tính có kiểu dữ liệu dạng số, do đó ta sẽ kiểm tra xem kiểu dữ liệu của cột có phải là dạng số hay không. Nếu đúng, tiến hành kiểm tra ngoại lệ, nếu sai (kiểu dữ liệu là object), bỏ qua bước kiểm tra ngoại lệ với cột đó. Việc phân biệt này theo cú pháp như sau:

```
1         if data[column].dtype != object
```

- Dựa vào kiến thức về xác suất thống kê để tính khoảng tứ phân vị **IQR**, ta lần lượt tính Q1 và Q3 bằng việc sử dụng **np.quantile()**. Từ đó, ta tính được khoảng biến thiên - nghĩa là khoảng xuất hiện thường xuyên nhất của các thuộc tính.



- Và ta suy ra được cận trên, cận dưới, đồng nghĩa ta đã tìm được khoảng tin cậy cho dữ liệu.
- Việc tiếp theo chính là tính phần trăm các điểm ngoại lai.
- Và trả về DataFrame chứa thông tin phần trăm ngoại lệ cho mỗi cột, được sắp xếp theo thứ tự giảm dần.

6 CHI TIẾT ĐỒ ÁN

6.1 Câu 1a

6.1.1 Yêu cầu bài toán

- Sử dụng 11 đặc trưng đầu tiên đề bài cung cấp bao gồm: **Gender, 10percentage, 12percentage, CollegeTier, Degree, collegeGPA, CollegeCityTier, English, Logical, Quant, Domain**
- Huấn luyện 1 lần duy nhất cho 11 đặc trưng nói trên cho toàn bộ tập huấn luyện **train.csv**
- Thể hiện công thức cho mô hình hồi quy
- Báo cáo 1 kết quả trên tập kiểm tra **test.csv** cho mô hình vừa huấn luyện được.

6.1.2 Mô tả thuật toán

- Đầu tiên, ta sẽ tạo một đối tượng là mô hình hồi quy tuyến tính bằng cách áp dụng lớp **OLSLinearRegression**. Đồng thời gọi phương thức **fit()** để huấn luyện mô hình trên tập train **X_train_a** và **y_train**.

- Tiếp theo, dự đoán giá trị trên tập kiểm tra **X_test_a** bằng cách gọi phương thức **predict()** từ mô hình đã huấn luyện. Lưu kết quả tìm được vào một biến dự đoán.
- Theo yêu cầu, ta sẽ hiển thị các trọng số hồi quy (**w**) bằng cách gọi phương thức **get_params**.
- Và ta tiến hành tìm **MAE** trên tập kiểm tra từ mô hình đã được huấn luyện, gọi hàm **MAE** và truyền vào **y_test** và **y_test_predict**

6.1.3 Kết quả & nhận xét

- Ta có kết quả trọng số của các đặc trưng trên tập huấn luyện như sau:

STT	Đặc trưng	Trọng số
1	Gender	-22756.513
2	10percentage	804.503
3	12percentage	1294.655
4	CollegeTier	-91781.898
5	Degree	23182.389
6	collegeGPA	1437.549
7	CollegeCityTier	-8570.662
8	English	147.858
9	Logical	152.888
10	Quant	117.222
11	Domain	34552.286

- Kết quả **MAE** tìm được trên tập kiểm tra như sau:

- Từ kết quả trọng số, ta suy ra được công thức tính **Salary** của mô hình:

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

- Nhận xét:

- Đầu tiên ta nhận thấy có 3 đặc trưng có trọng số âm là **Gender, CollegeTier, CollegeCityTier**. Ta sẽ tiến hành phân tích 3 thuộc tính này.
- **Gender** là đặc trưng biểu thị giới tính của ứng viên, đề bài có quy định ứng viên nam có giá trị là 1, ứng viên nữ giá trị là 2. Từ đó cho thấy, giới tính của ứng viển có ảnh hưởng lớn và chênh lệch rõ rệt đến mức lương **Salary**. Ứng viên giới tính nam có lương vượt trội hơn so với lương của ứng viên nữ.
- **CollegeTier** là chỉ số mức điểm trung bình AMCAT có điểm vượt qua ngưỡng cho trước. Các trường đại học có điểm trung bình vượt qua ngưỡng cho trước được đánh dấu là 1 và các trường khác là 2. Do đó, lương của các ứng viên bị ảnh hưởng lớn đến mức điểm trung bình đó. Nếu ứng viên có trường đại học vượt qua ngưỡng AMCAT thì mức lương sẽ vượt trội hơn so với các ứng viên có trường chưa vượt ngưỡng này.

- **CollegeCityTier** là hạng của thành phố có trường đại học mà ứng viên theo học. Quy định dựa trên dân số của thành phố. Nghĩa là nếu ứng viên đang theo học trường nào có thứ hạng càng cao (Top 1, top 2, ...) thì mức lương sẽ cao hơn các ứng viên học tại các trường không nổi trội (Top thấp hơn).
- Ta sẽ tiếp tục phân tích các đặc trưng có trọng số dương.
- Các đặc trưng **10percentage, 12percentage, Degree, collegeGPA** có điểm chung đó là các đặc trưng về điểm số và chứng chỉ, việc này cũng dễ hiểu vì khi ứng viên có điểm càng cao, chứng chỉ càng tốt thì mức lương sẽ vượt trội hơn mức lương của các ứng viên thấp hơn.
- Các đặc trưng **10percentage, 12percentage** đều là tổng điểm đạt thi trong kỳ thi, nhưng trọng số của **12percentage** lại lớn hơn, cho thấy nếu ứng viên đạt điểm cao hơn trong kỳ thi lớp 12 thì vẫn có cơ hội nhận lương cao hơn ứng viên khác cho dù trong kỳ thi lớp 10 có điểm thấp hơn.
- Đặc trưng **Degree** cho thấy bằng cấp mà ứng viên đã đạt được ở đại học, nếu ở đại học, ứng viên đạt 'Bachelor' thì sẽ có mức lương thấp hơn ứng viên đạt 'Master', điều này là tất nhiên.
- Các đặc trưng **English, Logical, Quant, Domain** đều có điểm chung là xuất phát từ thang điểm AMCAT, vậy AMCAT là gì?

  - AMCAT [1] là bài kiểm tra năng lực và là một hình thức đánh giá tâm lý, được sử dụng để đo điểm mạnh tự nhiên của một người nào đó trong một lĩnh vực nhất định. AMCAT khác với một bài kiểm tra kiến thức chuyên môn ở chỗ nó không yêu cầu sự quen thuộc với một chủ đề cụ thể.
  - Thay vào đó, AMCAT sẽ xem xét các kỹ năng vốn có của bạn và khả năng áp dụng chúng trong các tình huống bất ngờ. Aptitude test thường thuộc một trong hai loại: bài kiểm tra khả năng và bài kiểm tra hành vi.
  - Mục đích chính của AMCAT là xác định năng lực của bạn. Thay vì nhìn vào những gì bạn biết, AMCAT sẽ hỗ trợ nhà tuyển dụng xem xét năng lực học tập của bạn và khả năng làm việc với các thông tin mới có hiệu quả hay không. Điều này cho thấy các khả năng của bạn hoạt động tốt như thế nào trong một tình huống cụ thể.

- Từ đó cho thấy ngoài kiến thức chuyên môn, các ứng viên còn phải đáp ứng được một thái độ tốt, một tinh thần tích cực cũng như là có khả năng xử lý tình huống tốt thì sẽ tỷ lệ thuận với mức lương nhận được.

6.2 Câu 1b

6.2.1 Yêu cầu bài toán

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

- Thử nghiệm lần lượt trên các đặc trưng tính cách gồm: **conscientiousness, agreeableness, extraversion, nueroticism, openess_to_experience**.
- Yêu cầu sử dụng **k-fold Cross Validation** (k tối thiểu là 5) để tìm ra đặc trưng tốt nhất trong các đặc trưng tính cách.
- Báo cáo 5 kết quả tương ứng cho 5 mô hình.
- Thể hiện công thực cho mô hình hồi quy theo đặc trưng tốt nhất.
- Báo cáo 1 kết quả trên tập kiểm tra **test.csv** cho mô hình với đặc trưng tốt nhất tìm được.

6.2.2 Mô tả thuật toán

- Như đã đề cập về việc chọn k, ta sẽ mặc định k = 10
- Đầu tiên, chạy **cross-validation** để đo lường hiệu suất của mỗi đặc trưng dựa trên **MAE**.
- Chọn đặc trưng tốt nhất dựa trên trung bình **MAE** thấp nhất.
- Huấn luyện lại mô hình **best_personality_feature_model** với đặc trưng tốt nhất trên toàn bộ tập huấn luyện.
- Dự đoán giá trị trên tập kiểm tra và tính **MAE** của mô hình tốt nhất trên tập kiểm tra.

- Trong thuật toán có sử dụng **list comprehension** để tính trung bình **MAE** cho mỗi mô hình trong **models_train** bằng cách sử dụng hàm **custom_cross_validation_mae**.
- **custom_cross_validation_mae** đảm bảo việc thực hiện **cross-validation** và tính trung bình **MAE** qua các fold. Kết quả được thêm vào danh sách **average_maes**.
- Ta tìm chỉ mục của đặc trưng có trung bình MAE nhỏ nhất (tốt nhất) qua cú pháp:

```
best_model_index = np.argmin(average_maes)
```

- Ngoài ra, khi huấn luyện lại mô hình với đặc trưng tốt nhất. Tạo mô hình **best_personality_feature_model** bằng cách sử dụng **OLSLinearRegression().fit** với đặc trưng tốt nhất **best_feature** và **y_train_1b**.

6.2.3 Kết quả & nhận xét

- Ta có 5 kết quả tương ứng với 5 mô hình từ **k-fold Cross validation** như sau:

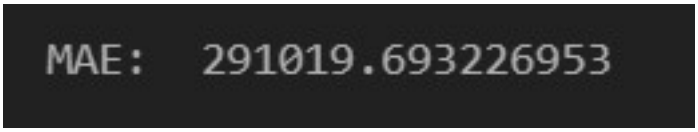
STT	Mô hình với 1 đặc trưng	MAE
1	conscientiousness	3.062816e+06
2	agreeableness	3.007685e+06
3	extraversion	3.070333e+06
4	nueroticism	2.992336e+06
5	openness_to_experience	3.028105e+06

- Từ đó cho thấy, đặc trưng tính cách **nueroticism** có MAE nhỏ nhất, chứng tỏ đây là đặc trưng tốt nhất, ta tiến hành tìm

lại trọng số của **nueroticism** trên toàn bộ tập huấn luyện, ta được kết quả như sau:

STT	Đặc trưng	Trọng số
1	nueroticism	-56546.304

- Kết quả **MAE** tìm được trên tập kiểm tra như sau:



- Từ kết quả trọng số, ta suy ra được công thức tính **Salary** của mô hình:

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

- Nhận xét:

– Các đặc trưng tính cách đều có MAE khá lớn, điều này cho thấy các đặc trưng tính cách là vô cùng quan trọng đối với một ứng viên.

– Tuy **nueroticism** là đặc trưng tốt nhất do MAE nhỏ nhất, nhưng các đặc trưng khác cũng đang ở mức MAE tiệm cận với đặc trưng này.

– Ta có thể giải thích như sau, một kỹ sư luôn có công việc vô cùng áp lực và nặng nề, hiệu quả của công việc đôi lúc cũng bị ảnh hưởng nhiều do tâm trang, do thái

độ làm việc, do tiếp xúc với môi trường xung quanh, điều này sinh ra căng thẳng và stress trong công việc của các kỹ sư. Chính vì vậy, sự nhạy cảm (nueroticism) là vô cùng quan trọng trong mức lương của các kỹ sư.

- Khi kỹ sư có tâm trạng, tinh thần càng nhạy cảm sẽ càng ảnh hưởng đến hiệu suất công việc, từ đó ảnh hưởng đến các đầu sản phẩm kéo theo sự suy giảm trong mức lương, chính vì vậy nên **nueroticism** có trọng số âm.
- Ngoài ra, các đặc trưng khác cũng cần được lưu tâm do có hệ số MAE khá lớn, ví dụ như một kỹ sư có tinh thần cởi mở (openess_to_experience) thì sẽ có cách làm việc và hiệu suất làm việc khác, hay một kỹ sư dễ tính, hòa đồng với mọi người (agreeableness) sẽ có cách làm việc và cách hưởng lương khác hơn.

6.3 Câu 1c

6.3.1 Yêu cầu bài toán

- 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
- Thử nghiệm trên các đặc trưng gồm: **English, Logical, Quant**.
- Yêu cầu sử dụng **k-fold Cross Validation** (k tối thiểu là 5) để tìm ra đặc trưng tốt nhất.

- Báo cáo 3 kết qura tương ứng cho 3 mô hình từ **k-fold Cross validation**
- Thể hiện công thức cho mô hình hồi quy theo đặc trưng tốt nhất
- Báo cáo 1 kết quả trên tập kiểm tra **test.csv** cho mô hình với đặc trưng tốt nhất tìm được.

6.3.2 Mô tả thuật toán

- Như đã đề cập về việc chọn k, ta sẽ mặc định k = 10, và ta cũng sẽ thực hiện ý tưởng tương tự như **Bài 1b** .
- Đầu tiên, chạy **cross-validation** để đo lường hiệu suất của mỗi đặc trưng dựa trên **MAE**.
- Chọn đặc trưng tốt nhất dựa trên trung bình **MAE** thấp nhất.
- Huấn luyện lại mô hình **best_skill_feature_model** với đặc trưng tốt nhất trên toàn bộ tập huấn luyện.
- Dự đoán giá trị trên tập kiểm tra và tính **MAE** của mô hình tốt nhất trên tập kiểm tra.
- Trong thuật toán có sử dụng **list comprehension** để tính trung bình **MAE** cho mỗi mô hình trong **models_train** bằng cách sử dụng hàm **custom_cross_validation_mae**.

- **custom_cross_validation_mae** đảm bảo việc thực hiện **cross-validation** và tính trung bình **MAE** qua các fold. Kết quả được thêm vào danh sách **average_maes**.
- Ta tìm chỉ mục của đặc trưng có trung bình MAE nhỏ nhất (tốt nhất) qua cú pháp:

```
best_model_index = np.argmin(average_maes)
```

- Ngoài ra, khi huấn luyện lại mô hình với đặc trưng tốt nhất. Tạo mô hình **best_skill_feature_model** bằng cách sử dụng **OLSLinearRegression().fit** với đặc trưng tốt nhất **best_feature** và **y_train_1c**.

6.3.3 Kết quả & nhận xét

- Ta có 3 kết quả tương ứng với 3 mô hình từ **k-fold Cross validation** như sau:

STT	Mô hình với 1 đặc trưng	MAE
1	English	1.219811e+06
2	Logical	1.203527e+06
3	Quant	1.181117e+06

- Từ đó cho thấy, đặc trưng tính cách **Quant** có MAE nhỏ nhất, chứng tỏ đây là đặc trưng tốt nhất, ta tiến hành tìm lại trọng số của **Quant** trên toàn bộ tập huấn luyện, ta được kết quả như sau:

STT	Đặc trưng	Trọng số
1	Quant	585.895

- Kết quả **MAE** tìm được trên tập kiểm tra như sau:

```
MAE:  106819.57761989674
```

- Từ kết quả trọng số, ta suy ra được công thức tính **Salary** của mô hình:

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

- Nhận xét:

– Từ kết quả ta thấy, đặc trưng **Quant** là đặc trưng tốt nhất trong 3 khả năng và giá trị cũng gần giống kết quả MAE ở **bài 1a**, điều này cho thấy khả năng định lượng **Quant** là vô cùng cần thiết đối với một kỹ sư.

– Trong môi trường làm việc của các kỹ sư, khả năng tính toán, làm việc với các con số diễn ra thường xuyên, do đó các khả năng về giải quyết dữ liệu, xem xét các con số là vô cùng quan trọng.

– Các đặc trưng **English và Logical** cũng cho thấy sự quan trọng cho một kỹ sư về khả năng ngoại ngữ và khả năng logic. Nhưng với một môi trường nhiều số liệu như một kỹ sư thì **Quant** dĩ nhiên là quan trọng hơn.

– Thông số về trọng số cũng cho thấy điều này khi trọng số là một con số dương. Đồng nghĩa với việc khi khả năng tính toán, giải quyết các con số của một kỹ sư càng tốt thì hiệu suất làm việc và chất lượng công việc cũng tăng lên, điều này tác động mạnh đến mức lương **Salary** của kỹ sư.

7 QUÁ TRÌNH TÌM KIẾM MÔ HÌNH CHO PHẦN 1D

7.1 Yêu cầu bài toán

- Sinh viên tự xây dựng mô hình, tìm mô hình cho kết quả tốt nhất
- Xây dựng **m** mô hình khác nhau (tối thiểu 3), đồng thời khác mô hình ở 1a, 1b và 1c.
- Yêu cầu sử dụng phương pháp **k-fold Cross Validation** (k tối thiểu là 5) để tìm ra mô hình tốt nhất trong ‘m‘ mô hình mà sinh viên xây dựng
- Báo cáo **m** kết quả tương ứng cho **m** mô hình từ **k-fold Cross Validation** (lấy trung bình)

7.2 Mô tả ý tưởng

- Hệ số tương quan (correlation) [6], [9] là một thước đo đo lường mức độ liên quan tuyến tính giữa hai biến. Trong ngữ

cảnh này, việc tính hệ số tương quan giữa mỗi cột trong tập dữ liệu và cột cuối cùng (có thể là cột biến mục tiêu) có thể giúp xác định mức độ tương quan của từng biến với biến mục tiêu. Điều này có thể hữu ích trong việc chọn các đặc trưng quan trọng để sử dụng trong mô hình học máy hoặc trong quá trình hiểu rõ mối quan hệ giữa các biến trong dữ liệu.

- Ta cần tìm ra hệ số tương quan giữa **Salary** và các thuộc tính đặc trưng trong tập dữ liệu train
- Sau đó, nhờ các hệ số tương quan, ta sẽ quyết định các đặc trưng cần thiết cho mô hình. Nếu hệ số tương quan từ 0 đến 1, ta có thể kết luận thuộc tính đó tương quan thuận với Salary, nghĩa là nếu thuộc tính đó tăng thì Salary cũng tăng, và ngược lại. [2], [8]
- Bằng cách này, ta có thể suy ra 3 mô hình tùy thuộc vào 3 hệ số tương quan tối thiểu mà chúng ta muốn nhận, ở đây, em sẽ có 3 hệ số tối thiểu tương ứng với 3 mô hình là 0.15, 0.1 và 0.
- Ta có thể bao quát hơn các đặc trưng của mô hình bằng việc bổ sung thêm các đặc trưng nào có nhiều điểm ngoại lai (Outliers), nếu một đặc trưng có nhiều điểm ngoại lai so với biểu đồ tổng quát, ta cũng có thể thêm đặc trưng đó vào mô hình cần tìm, vì càng nhiều điểm ngoại lai, càng ảnh hưởng đến kết quả chung của giá trị Salary.

- Mục tiêu của mô hình cần có:
  - Có được MAE thấp nhất so với các mô hình đã làm
  - Các đặc trưng không cần thiết hoặc gây mất công bằng (Gender, ...) không nên xuất hiên trong mô hình

7.3 Mô tả thuật toán

- Ta sẽ tiến hành đi tìm hệ số tương quan giữa các đặc trưng và **Salary**
- Để tính toán hệ số tương quan (correlation) giữa mỗi cột trong DataFrame **train** và cột cuối cùng của nó, ta áp dụng cú pháp sau [7].

```
correlation = train.corr().iloc[:, -1]00
```

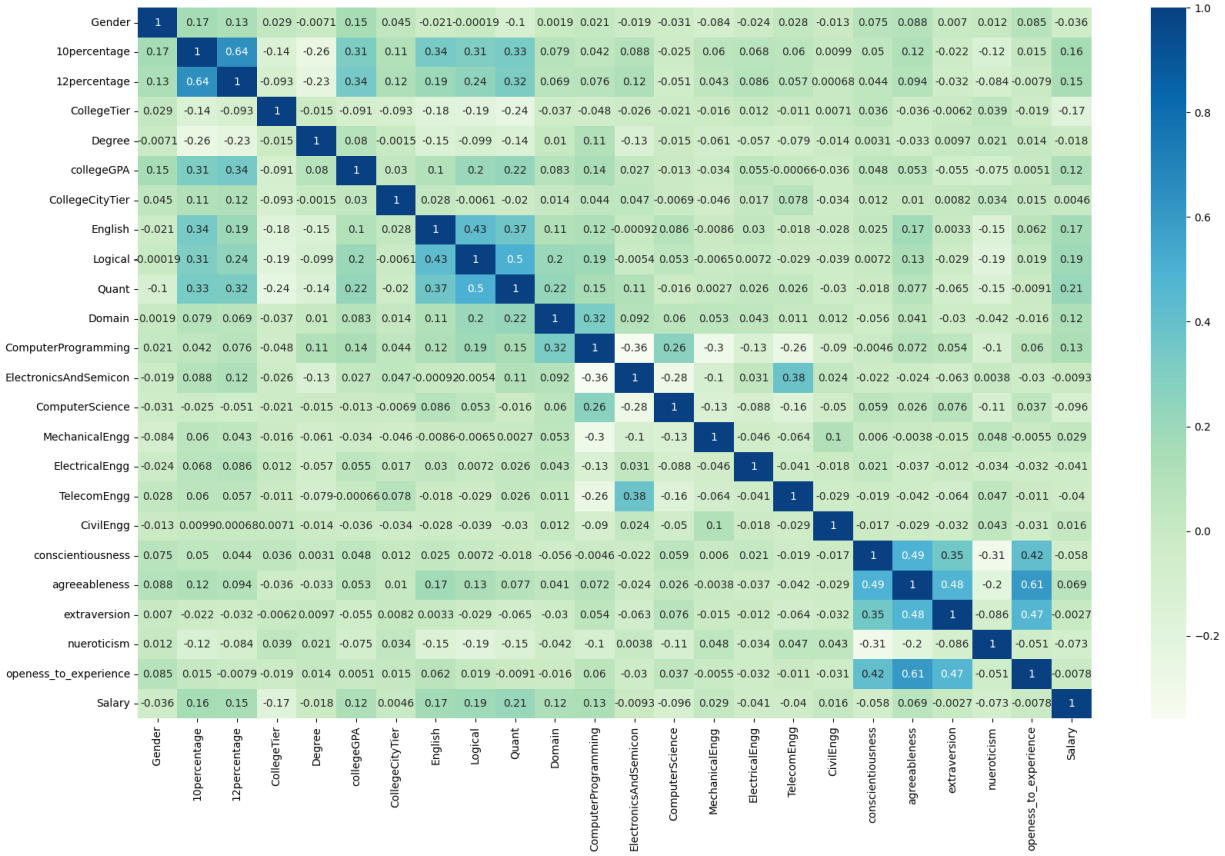
- Trong đó:
  - **train.corr()**: Sử dụng phương thức **.corr()** trên DataFrame train để tính ma trận tương quan giữa tất cả các cột trong tập dữ liệu. Kết quả là một ma trận chứa các giá trị tương quan giữa các cặp cột.
  - **iloc[:, -1]**: Sử dụng **.iloc** để truy cập dữ liệu trong ma trận tương quan. Ở đây, : đại diện cho tất cả các hàng, còn -1 đại diện cho cột cuối cùng của ma trận tương quan, có thể tương ứng với biến mục tiêu.
  - **correlation = train.corr().iloc[:, -1]**: Gán kết quả trên (cột cuối cùng của ma trận tương quan) vào biến **corre-**

lation, mà thường sẽ chứa các giá trị tương quan giữa mỗi biến trong tập dữ liệu và biến mục tiêu.

- Từ đó ta có được hệ số tương quan giữa các đặc trưng và **Salary** như hình:

	Correlation
Salary	1.000000
Quant	0.205358
Logical	0.188416
English	0.169293
10percentage	0.155174
12percentage	0.149531
ComputerProgramming	0.125866
collegeGPA	0.122469
Domain	0.122022
agreeableness	0.068623
MechanicalEngg	0.028854
CivilEngg	0.016150
CollegeCityTier	0.004575
extraversion	-0.002661
openess_to_experience	-0.007814
ElectronicsAndSemicon	-0.009292
Degree	-0.017602
Gender	-0.036183
TelecomEngg	-0.040415
ElectricalEngg	-0.041217
conscientiousness	-0.057699
nueroticism	-0.073401
ComputerScience	-0.095507
CollegeTier	-0.174824

- Ngoài cách biểu diễn hệ số tương quan dạng bảng số liệu, ta có thể biểu diễn dưới dang một ma trận, một bản đồ nhiệt, từ đó cho thấy rõ những đặc trưng liên quan với nhau hơn. Ta cần sử dụng thư viện **seaborn** để có được một biểu đồ nhiệt như sau:



- Như đã đề cập, ngoài việc áp dụng hệ số tương quan, ta cần dựa vào các chỉ số ngoại lai của một bộ dữ liệu để loại bỏ cũng như chọn lọc lại những đặc trưng đáng tin cậy.
- Ta sử dụng hàm đã cài đặt ở phần trên **detect_outliers** và có được thống kê như sau:

	Outlier_percentage
Gender	24.065836
ComputerScience	23.754448
ComputerProgramming	20.951957
TelecomEngg	9.430605
Degree	7.918149
CollegeTier	7.651246
Domain	6.138790
MechanicalEngg	5.560498
ElectricalEngg	3.870107
Salary	2.580071
openess_to_experience	2.402135
agreeableness	1.601423
extraversion	1.156584
conscientiousness	1.112100
collegeGPA	1.067616
CivilEngg	0.889680
Quant	0.756228
10percentage	0.667260
nueroticism	0.400356
English	0.355872
Logical	0.177936
12percentage	0.044484
ElectronicsAndSemicon	0.044484
CollegeCityTier	0.000000

CHỌN LỌC MÔ HÌNH

- Từ số liệu trong bảng hệ số tương quan với **Salary**, ta thấy các đặc trưng **Quant, Logical, English, 10percentage, 12percentage, ComputerProgramming, collegeGPA, Domain, agreeableness, MechanicalEngg, CivilEngg, CollegeCityTier** là những đặc trưng mà hệ số tương quan lớn hơn 0. Đồng nghĩa với việc nếu một trong cá đặc trưng này tăng thì **Salary** cũng sẽ tăng và ngược lại.
- Do đó, ta sẽ chia các đặc trưng này thành 3 mô hình khác nhau để kiểm tra.
- Mô hình đầu tiên sẽ lấy những đặc trưng có correlation lớn

hơn **0.15**.

- Mô hình đầu tiên sẽ lấy những đặc trưng có correlation lớn hơn **0.1**.
- Mô hình đầu tiên sẽ lấy những đặc trưng có correlation lớn hơn **0**.
- Bên cạnh đó, đến lượt bảng thống kê các giá trị ngoại lai, ta thấy **Salary** đang ở khoảng 2.58. Những đặc trưng nào có giá trị ngoại lai vượt trội hoặc xa hơn **Salary** nhiều lần sẽ ảnh hưởng nhiều đến tổng thể bảng số liệu. Do đó ta cần chọn những thuộc tính đó vào mô hình.
- Loại bỏ các đặc trưng ta đã có từ việc chọn tương quan, ta còn **ComputerScience, TelecomEngg, Degree và CollegeTier** là phù hợp nhất do giá trị đủ lớn để ảnh hưởng.
- Ta sẽ thêm các đặc trưng vừa tìm được đó vào cả 3 mô hình.
- Cách chọn mô hình tốt nhất cũng sẽ giống như các **bài 1a**, tìm MAE tốt nhất trên tập huấn luyện sau đó tìm **my_best_model** và tính MAE kiểm tra.

7.4 Kết quả & nhận xét

- Ta có 3 kết quả tương ứng với 3 mô hình từ **k-fold Cross validation** như sau:

STT	Mô hình với 1 đặc trưng	MAE
1	Model 1	277562.489
2	Model 2	277805.003
3	Model 3	277896.525

- Từ đó cho thấy, model 1 là model MAE tốt nhất và là mô hình phù hợp nhất trong số 3 mô hình
- Model 1 gồm có các đặc trưng: **Quant, Logical, English, 10percentage, 12percentage, ComputerProgramming, collegeGPA, Domain, agreeableness, MechanicalEngg, CivilEngg, CollegeCityTier, ComputerScience, TelecomEngg, Degree và CollegeTier**
- Ta có kết quả trọng số của các đặc trưng trên tập huấn luyện như sau:

STT	Đặc trưng	Trọng số
1	Quant	119.728
2	Logical	139.523
3	English	162.418
4	10percentage	730.822
5	12percentage	999.712
6	ComputerProgramming	111.115
7	collegeGPA	1236.523
8	Domain	22859.617
9	agreeableness	5179.526
10	MechanicalEngg	75.531
11	CivilEngg	151.137
12	CollegeCityTier	-8382.822
13	ComputerScience	-164.440
14	TelecomEngg	-69.950
15	Degree	12696.401
16	CollegeTier	-89370.213

- Kết quả **MAE** tìm được trên tập kiểm tra với mô hình **my_best_model** như sau:

- Từ kết quả trọng số, ta suy ra được công thức tính **Salary** của mô hình:

$$\begin{aligned}\text{Salary} = 119.728 * Quant + 139.523 * Logical \\ +162.418 * English + 730.822 * 10percentage \\ +999.712 * 12percentage + 111.115 * ComputerProgramming \\ +1236.523 * collegeGPA + 22859.617 * Domain \\ +5179.526 * agreeableness + 75.531 * MechanicalEngg \\ +151.137 * CivilEngg - 8382.822 * CollegeCityTier \\ -164.440 * ComputerScience - 69.950 * TelecomEngg \\ +12696.401 * Degree - 89370.213 * CollegeTier\end{aligned}$$

- Nhận xét:

– Đây là mô hình tốt nhất trong quá trình tìm được, tốt hơn mô hình ở câu 1a về giá trị MAE cũng như về công thức **Salary**.

– Và đây chỉ là mô hình ta tìm được dựa trên các số liệu thống kê thu nhận được, vậy vì sao mô hình này lại tốt như vậy, bây giờ ta sẽ đi vào giải thích các đặc trưng của mô hình.

– Các đặc trưng về tính cách và thái độ trong AMCAT như đã nêu ở phần trên **Quant, English, Logical, Domain, agreeableness** là vô cùng quan trọng và có trọng

số dương ảnh hưởng lớn đến mức lương của kỹ sư thông qua các tính cách làm việc của mình.

- Các đặc trưng về học thuật như **10percentage, 12percentage, collegeGPA, Degree** cũng vô cùng cùng cần thiết khi điểm số, thành tích và chứng chỉ luôn là tiêu chí ưu tiên trong các mức lương của kỹ sư.
- Các đặc trưng có trọng số âm như **CollegeCityTier, CollegeTier** để chỉ thứ hạng của một trường đại học hay một thành phố càng gần Top đầu thì mức lương mà nhân viên nhận sẽ càng cao.
- **TelecomEngg, ComputerScience** là 2 đặc trưng gây bất ngờ khi kỹ sư không nhất thiết phải được điểm cao trong cá kỳ thi Khoa học máy tính và Kỹ thuật Viễn Thông, thay vào đó là **ComputerProgramming** - ngành Lập trình máy tính, như vậy xu hướng trong bộ dữ liệu này có xu hướng thiên về kỹ sư lập trình và kỹ thuật trong việc lập trình là cần thiết hơn.

Tài liệu

[1] Amcat là gì. https://glints.com/vn/blog/aptitude-test-la-gi/. Truy cập vào 20 tháng 8 năm 2023.

[2] Các chỉ tiêu đánh giá mô hình. https://chaydinhluong.com/9-chi-tieu-danh-gia-do-chinh-xac-mo-hinh-hoi-q Truy cập vào 14 tháng 8 năm 2023.

[3] Dataframe trong python, trong pandas. https://www.w3schools.com/python/pandas/pandas_dataframes.asp. Truy cập vào 16 tháng 8 năm 2023.

[4] Hồi quy tuyến tính trong machine learning. https://www.geeksforgeeks.org/ml-linear-regression/. Truy cập vào 13 tháng 8 năm 2023.

[5] Mae trong hồi quy tuyến tính. https://solieu.vip/cach-tinh-gia-tri-cac-chi-so-aic-bic-mae-mape- Truy cập vào 15 tháng 8 năm 2023.

[6] Thiết lập các đặc trưng nhờ hệ số tương quan. https://www.kaggle.com/code/nitinchoudhary012/engineering-graduate-salary-prediction. Truy cập vào 16 tháng 8 năm 2023.

[7] Thư viện numpy. https://numpy.org/doc/stable/reference/index.html. Truy cập vào 20 tháng 8 năm 2023.

[8] Tìm hiểu về dữ liệu dạng bảng. https://machinelearningcoban.com/tabml_book/ch_intro/properties.html#. Truy cập vào 11 tháng 8 năm 2023.

[9] Tìm hiểu về hệ số tương quan (correlations). https://www.careerlink.vn/cam-nang-viec-lam/kien-thuc-kinh-te/he-so-tuong-quan-correlation-coefficient-la-gi
~:text=H%E1%BB%87%20s%E1%BB%91%20t%C6%B0%C6%A1ng%20quan%20l%C3%A0,trong%20ph%C3%A9p%20%C4%91o%20t%C6%B0%C6%A1ng%20quan. Truy cập vào 20 tháng 8 năm 2023.

[10] Tìm hiểu về k-fold cross-validation (1). https://trituenhantao.io/kien-thuc/gioi-thieu-ve-k-fold-cross-validation/. Truy cập vào 14 tháng 8 năm 2023.

[11] Tìm hiểu về k-fold cross-validation (2). https://web888.vn/k-fold-validation-danh-gia-model/. Truy cập vào 14 tháng 8 năm 2023.

[12] Thầy Huy Cô Uyên. Sự hướng dẫn trong tiết thực hành.

[13] Lê Phong Nguyễn Đình Thức, Đặng Hải Vân. *Giáo trình Thống kê Máy tính*. NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT, 2010.

[14] Nguyễn Thị Mộng Ngọc. *Giáo trình Xác suất thống kê*. NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH, 1 edition, 2022.

Hết!