BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐẠI HỌC QUỐC GIA TP.HCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

-----ଓ📖ଌ -----

TOÁN ỨNG DỤNG VÀ THỐNG KÊ CHO CNTT

Project 3: Linear Regression

Giảng viên hướng dẫn: Vũ Quốc Hoàng, Nguyễn Văn Quang Huy, Ngô Đình Hy, Phan Thị Phương Uyên

Họ tên: Phạm Phú Toàn

MSSV: 21127183

Lớp: 21CLC08

Mục Lục

I. Giới thiệu đồ án

Đây là đồ án cá nhân của môn toán ứng dụng trường Đại học Khoa Học Tự Nhiên TPHCM, khoa công nghệ thông tin.

Mục tiêu của đồ án là tìm hiểu các yếu tố quyết định ảnh hưởng đến mức lương của kỹ sư ngay sau khi tốt nghiệp thông qua xây dựng mô hình OLS Lineargression (hồi quy tuyến tính) trên tập dữ liệu được thu thập tại Ấn Độ từ đó đưa ra các mô hình và nhận định các yếu tố ảnh hưởng đến mức lương của kỹ sư.

II. Checklist

STT	Công việc	Phần trăm
1	1a: Xây dựng mô hình dựa trên 11 đặc trưng	100%
2	1b: Phân tích ảnh hưởng của đặc trưng tính cách	100%
3	1c: Phân tích ảnh hưởng của đặc trưng ngoại ngữ, lô-gic, định lượng	100%
4	1d: Tự xây dựng mô hình (3 mô hình) và đưa ra mô hình tốt nhất	100%
5	Báo cáo	100%
	Tổng	100%

III. Liệt kê các thư viện

- Thư viện:
  - **Scikit-learn**: dùng để thực hiện k-cross validation. [1]
    - KFold
    - Cross_val_score
    - LinearRegression

- **Numpy**
- **Pandas** [2]
- **Matplot** [3]
- **Seabons** [4]

IV. Giải thích các hàm cài đặt – sử dụng

- **Class OLSLinearRegression**: sử dụng lại các hàm đã được giáo viên cài đặt trong lab04 [5]
  - **fit(x,y)**: thực hiện xây dựng mô hình – tìm các hệ số của mô hình dựa trên tập huấn luyện đầu vào.
  - **get_params():** trả về các tập các hệ số của mô hình.
  - **predict(x):** thực hiện dự đoán và trả về các quả y của mô hình đã xây dựng dựa trên tập giá trị đầu vào là x.
- **mae(y, y_hat):** Hàm được giáo viên cài đặt trong lab04 [5]
  - Thực hiện tìm độ lỗi tuyệt đối trung bình của hai tập giá trị y đầu vào
  - Trả về kiểu số thực là sai số của hai tập y trên.
- **KFold(n_splits= 5)**: là hàm/object của thư viện scikit-learn
  - Là K-Folds cross-validator với n_splits là số nhóm mà dữ liệu sẽ được chia ra, ở đây là 5.
- **cross_val_score**: là hàm của thư viện scikit-learn.
  - Input: LinearRegression, tập X, tập Y, cv=Kfold đã được setup trước và các indicators được custom theo từng yêu cầu khác
  - Output: là tập điểm đánh giá k fold
  - Là hàm dùng để thực hiện thuật toán k-cross-validation
- **LinearRegression(fit_intercept=False):**
  - Có vai trò giống class OLSLinearRegression phía trên nhưng là object của thư viện scikit-learn để có thể hoạt động với hàm **cross_val_score**.

V. Trình bày kết quả, giải thích

- **Câu 1a:**
  - Mô hình đã xây dựng:

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

- Độ lỗi MAE (đã làm tròn đến 3 chữ số thập phân):

mae = 104863.778

- Giải thích:

B1: Đầu tiên ta lọc ra 11 đặc trưng từ hai bộ dữ liệu là train và test và lọc ra các cột salary Y từ hai bộ dữ liệu.

B2: ta xây dựng mô hình tuyến tính bằng cách dùng các hàm trong class OLS đã được cài đặt sẵn.

- Gọi hàm fit() để tìm params với input là tập x train và y train.
- Gọi hàm get_params() nếu cần hệ số params.

B3: In ra độ lỗi MAE và mô hình tương ứng.

- Thực hiện:
  - Gọi hàm predict() với đầu vào là tập x test để lấy tập y dự đoán.
  - Gọi hàm mae() với đầu vào là tập y dự đoán và tập y test.
- Đánh giá mô hình:
  - Độ lỗi thấp nhất trong cả bài làm → 11 đặc trưng đầu giúp dự đoán khá chính xác mức lương (phạm vi trong bài làm này).

- **Câu 1b:**
  - Bảng k-cross validation score của 5 đặc trưng tính cách:

	Đặc trưng	MAE
3	neuroticism	299272.456689
1	agreeableness	300741.341545
4	openness_to_experience	302875.302830
0	conscientiousness	306068.256984
2	extraversion	306841.709298

- Đặc trưng tốt nhất: **neuroticism**
- Mô hình với đặc trưng neuroticism:
  - **Salary = neuroticism x -56546.304**
  - Độ lỗi mae = **291019.693** (đã làm tròn)
- **Giải thích:**

B1: Ta trộn bộ dữ liệu bằng câu lệnh train.sample(frac=1)

B2: Ta chạy vòng lặp và thực hiện thuật toán k-cross validation cho 5 đặc trưng:

- Trong mỗi lần lặp em gọi hàm cross_val_sscore để thực hiện thuật toán k-cross validation với k =5 và hàm trả về một mảng gồm điểm của các lần lặp con.
  - Input là các x train và y train của mô hình cùng với các tham số của thư viện yêu cầu.
- Sau đó em tính tổng trung bình và lấy giá trị tuyệt đối của trung bình trên vì hàm cross_val_score có thể trả về các phần tử âm (do thư viện được cài đặt vậy). Đây chính là điểm MAE của từng đặc trưng

B3: Em in bảng điểm MAE với từng đặc trưng tương ứng

B4: Em huấn luyện lại mô hình với đặc trưng có điểm là tốt nhất (ở đây là neuroticism): thực hiện lại các bước 2 và 3 như ở câu 1a.

- Đánh giá mô hình:
  - Từ điểm số MAE trên ta có thể thấy được mức lương không thật sự liên quan đến tính cách và dùng các đặc trưng tính cách để xây dựng mô hình dự đoán mức lương sẽ đưa ra độ lỗi lớn. Đặc trưng tính cách neuroticism sẽ liên quan đến mức lương hơn các đặc trưng tính cách còn lại

- **Câu 1c**:
  - Bảng k-cross validation score của 3 đặc trưng đã sắp xếp:

	Đặc trưng	MAE
2	Quant	118082.273618
1	Logical	120293.539455
0	English	121872.476404

- Đặc trưng tốt nhất: **Quant**
- Mô hình với đặc trưng quant:
  - **Salary = Quant x 585.895**
  - Độ lỗi mae = **106819.578** (đã làm tròn)
- Giải thích**:**

B1: Ta trộn bộ dữ liệu bằng câu lệnh train.sample(frac=1)

B2: Ta chạy vòng lặp và thực hiện thuật toán k-cross validation cho 3 đặc trưng tính cách:

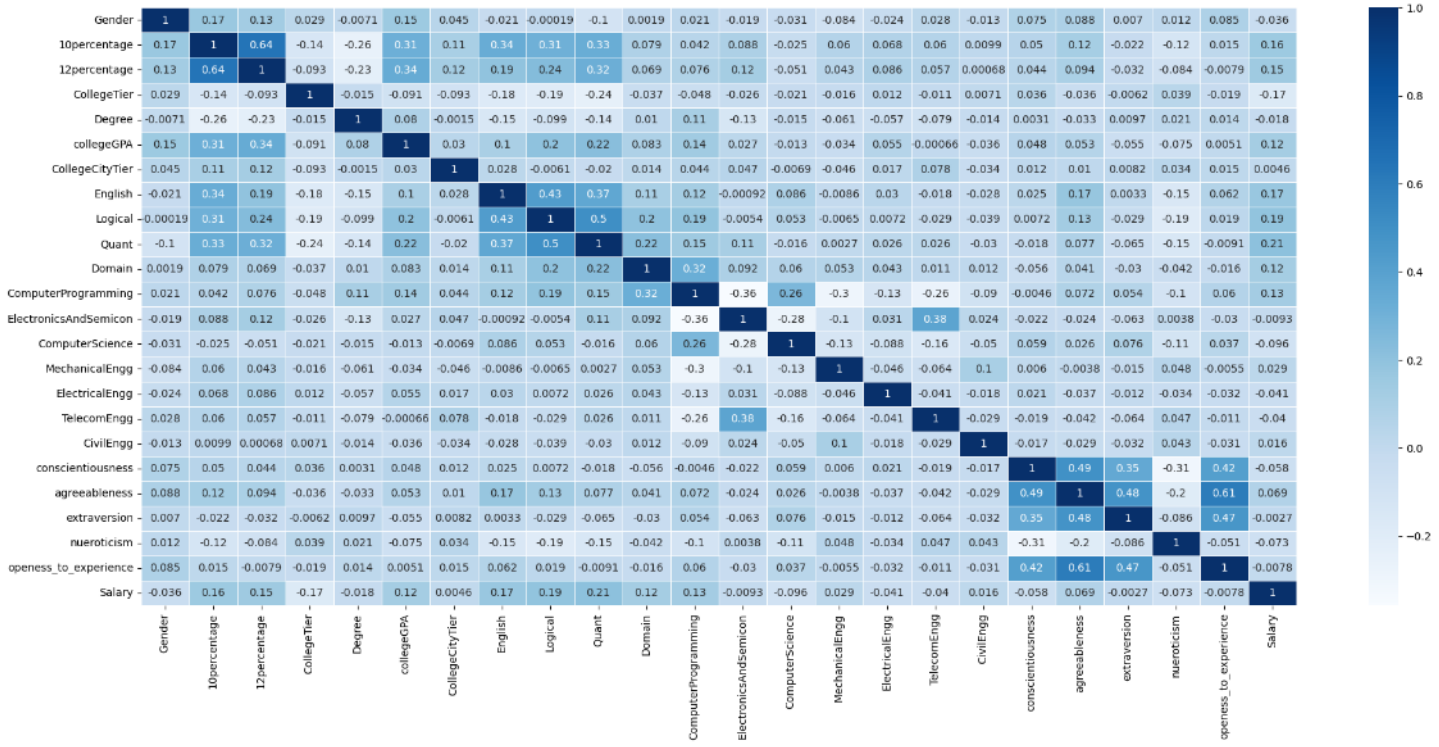
- Trong mỗi lần lặp em gọi hàm cross_val_sscore để thực hiện thuật toán k-cross validation với k =5 và hàm trả về một mảng gồm điểm của các lần lặp con.
  - Input là các x train và y train của mô hình cùng với các tham số của thư viện yêu cầu.
- Sau đó em tính tổng trung bình và lấy giá trị tuyệt đối của trung bình trên vì hàm cross_val_score có thể trả về các phần tử âm (do thư viện được cài đặt vậy). Đây chính là điểm MAE của từng đặc trưng

B3: Em in bảng điểm MAE với từng đặc trưng tương ứng

B4: Em huấn luyện lại mô hình với đặc trưng có điểm là tốt nhất (ở đây là Quant): thực hiện lại các bước 2 và ba như ở câu 1a.

- Đánh giá mô hình**:** các mô hình đánh giá dựa trên đặc trưng quant sẽ đưa ra dự đoán chính xác về mức lương hơn là các mô hình đánh giá dựa trên đặc trưng ngoại ngữ và logic. Cả ba đặc trưng ngoại ngữ, logic và quant nhìn chung sẽ liên quan đến mức lương hơn là các đặc trưng tính cách khi sử dụng để xây dựng mô hình

- **Câu 1d:**
  - Quá trình tìm mô hình:



Em sử dụng thư viện seaborn [6] để biểu diễn tập data đã cho bằng heatmap, thể hiện mối quan hệ giữa các đặc trưng với nhau. Ta chỉ cần quan tâm cột cuối cùng trong bảng vì nó thể hiện mối quan hệ giữa mức lương "Salary" và các đặc trưng còn lại.

Theo em tìm hiểu thì các đặc trưng có quan hệ với nhau là các số có trị tuyệt càng gần 1 thì sẽ càng ảnh hưởng và liên quan mật thiết tới nhau. [7] → Khi đó ta nên sử dụng các đặc trưng có quan hệ với Salary là các số có **trị tuyệt càng gần 1** để xây dựng mô hình.

Model 1: Vậy với model 1 của em là mô hình được xây dựng với các đặc trưng có giá trị ở cột salary là dương, gồm 11 đặc trưng:

‘10percentage','12percentage','collegeGPA','English','Logical','Quant','Domain','ComputerProgramming','MechanicalEngg','CivilEngg','agreeableness'

Model 2: Model 2 của em sẽ được xây dựng với các đặc trưng có giá trị ở cột salary là âm, gồm 12 đặc trưng:

'Gender','CollegeTier','Degree','ElectronicsAndSemicon','ComputerScience','ElectricalEngg','TelecomEngg','conscientiousness','extraversion','nueroticism','openess_to_experience'

Model 3: Sau khi thử nghiệm em thấy các đặc trưng dương là tốt hơn các đặc trưng âm, nên em sẽ xây dựng model 3 là mô hình chỉ gồm các đặc trưng có giá trị dương lớn hơn 0.1, gồm 8 đặc trưng:
'10percentage','12percentage','collegeGPA','English','Logical','Quant','Domain','ComputerProgram ming'

Model 4: Khi thử nghiệm em thấy model 3 và model 1 không khác nhau nhiều, nên em sẽ chọn lọc các đặc trưng dương (>=0.15) và các đặc trưng âm (<= -0.07) để tạo ra model 4, gồm 8 đặc trưng:

'10percentage','12percentage','CollegeTier','English','Logical','Quant','ComputerScience','nueroticis m'Quá trình xây dựng mô hình

- Đánh giá và đưa ra mô hình tốt nhất:

	Model	MAE
3	model 4: trị tuyệt gần 1	112937.692625
2	model_3: rất dương(>0,1)	115043.741852
0	model_1: dương	115371.319150
1	model_2: âm	131626.080875

Nhận xét

- Ta có thể thấy rõ ràng là với **model_2** (chỉ gồm toàn các đặc trưng có số liệu âm trên heatmap) thì độ **lỗi mae là cao nhất**.
- Với **model_1 và model_3** (dương và rất dương) ta thấy được **sự chênh lệch khá nhẹ** và có khi **không rõ ràng** (trong một vài lần chạy điểm MAE của model 3 lại cao hơn model 1)

→ rõ ràng sử dụng các đặc trưng thể hiện trên heap map có quan hệ với salary là các số dương sẽ đưa ra mô hình tốt hơn là sử dụng các đặc trưng là các số âm. Song khi sử dụng các đặc trưng có số liệu là dương lớn hơn lại không đưa ra một mô hình tốt hơn.

- Sử dụng mô hình với các đặc trưng có mối quan hệ với salary là các số có trị tuyệt gần 1 (≥ 0.15 $và$ ≤ 0.07): **model_4** đã cho ra một môt hình **tốt hơn. Model 4 có độ lỗi thấp nhất trong tất các mô hình đề xuất và thấp thứ 2 trong bài làm.**

→ model 4 là mô hình tốt nhất

- Model 4:
  - Độ lỗi MAE = 105385.683
  - Mô hình : Salary = 893.349 x 10percentage + 1415.127 x 12percentage + -69058.575 x CollegeTier + 173.089 x English + 219.112 x Logical + 156.061 x Quant + -130.236 x ComputerScience + -6530.864 x nueroticism

VI. References

[1] "Scikitlearn - API Reference," [Online]. Available: https://scikit-learn.org/stable/modules/classes.html. [Accessed 21 08 2023].

[2] "Pandas - API Reference," [Online]. Available: https://pandas.pydata.org/docs/reference/index.html. [Accessed 21 08 2023].

[3] "Matplot Tutorials," [Online]. Available: https://matplotlib.org/stable/tutorials/index. [Accessed 21 08 2023].

[4] "Seaborn Heatmap," [Online]. Available: https://seaborn.pydata.org/generated/seaborn.heatmap.html. [Accessed 21 08 2023].

[5] Phan Thị Phương Uyên, "Lab 04 - Applied Math".

[6] "Seaborn Heatmap – A comprehensive guide," [Online]. Available: https://www.geeksforgeeks.org/seaborn-heatmap-a-comprehensive-guide/. [Accessed 21 08 2023].

[7] "Giới thiệu về k fold cross validation," [Online]. Available: https://trituenhantao.io/kien-thuc/gioi-thieu-ve-k-fold-cross-validation/. [Accessed 20 08 2023].

[8] "Correlation," [Online]. Available: https://www.surveysystem.com/correlation.htm. [Accessed 21 08 2023].