ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO MÔN HỌC
TOÁN ỨNG DỤNG VÀ THỐNG KÊ

Lab 03: Linear Regression

Giảng viên: Nguyễn Đình Thúc
Nguyễn Văn Quang Huy
Ngô Đình Hy
Trần Hà Sơn

Mục lục

I. Thông tin sinh viên

- Họ tên: Võ Nguyễn Hoàng Kim
- Mã số sinh viên: 21127090
- Lớp: 21CLC07

II. Mức độ hoàn thành

Yêu cầu	Mức độ hoàn thành
Yêu cầu 1a	100%
Yêu cầu 1b	100%
Yêu cầu 1c	100%
Yêu cầu 1d	100%

III. Giới thiệu đồ án

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. Sinh viên sẽ được khám phá những thông tin này trong phạm vi đồ án.

Trong đồ án này, sinh viên cần thực hiện: xây dựng mô hình dự đoán mức lương của kỹ sư sử dụng mô hình hồi quy tuyến tính.

IV. Các thư viện và hàm được sử dụng

1. Các thư viện được sử dụng

a. Thư viện pandas

- Thư viện pandas được dùng để hỗ trợ, xử lý và làm việc với các dữ liệu dạng bảng (như DataFrames).
- Nó hỗ trợ các chức năng như đọc, viết, chuyển đổi, phân tích,… dữ liệu.

b. Thư viện numpy

- Numpy được sử dụng để hỗ trợ tính toán và xử lý các dữ liệu đối với mảng đa chiều (hỗ trợ tốt hơn so với các kiểu dữ liệu trong Python).

c. Thư viện matplotlib.pylot

- Thư viện matplotlib.pylot được dùng để hỗ trợ, xử lý và làm việc với các dữ liệu dạng bảng (như DataFrames).
- Nó hỗ trợ các chức năng như đọc, viết, chuyển đổi, phân tích,… dữ liệu.

d. Thư viện seaborn

- Được sử dụng để hỗ trợ tính toán và xử lý các dữ liệu đối với mảng đa chiều
- Đồng thời được sử dụng để trực quan hóa dữ liệu.

e. Thư viện sklearn.model_selection

- Dùng để sử dụng **k-fold Cross Validation**.
- Được sử dụng cho việc lựa chọn các mô hình.

2. Các hàm được sử dụng

a. Trong class OLSLinearRegression

- **fit**(*self, X, y*)
  Hàm được sử dụng để thực hiện quá trình "khớp" – fitting của mô hình hồi quy tuyến tính.
  Nó tính trọng số của mô hình bằng công thức $w = (X^T X)^{-1} X^T y$
  Tham số X và y tương ứng là ma trận (các đặc trưng đầu vào) và vector của biến mục tiêu.
  Hàm trả về các thực thể đã được khớp (fitted instance) của mô hình hồi quy tuyến tính.
- **get_params**(*self*)
  Hàm trả về các trọng số của mô hình sau khi thực hiện khớp (fit) dữ liệu.
- **predict**(*self, X*)
  Hàm được dùng để đưa ra dự đoán bằng cách sử dụng mô hình hồi quy tuyến tính.
  Nó thực hiện nhân trọng số mô hình đã học với ma trận đặc trưng (đầu vào), từ đó đưa ra dự đoán.
  Giá trị trả ra là một mảng các giá trị đã được dự đoán.

b. Các hàm được cài đặt thêm

- **mae**(*y, y_hat*)
  Hàm được cài đặt để ước lượng "độ lỗi tuyệt đối trung bình" (MAE – Mean Absolute Error) giữa biến mục tiêu thực (y) và biến mục tiêu dự đoán (y_hat).
  Hàm trả về độ lỗi MAE đã được tính.
- **preprocess**(*x*)
  Được cài đặt để thêm một cột số 1 vào ma trận đầu vào (thêm vào cột đầu tiên của ma trận).
  Điều này giúp ích cho các tính toán ở bước sau.

V. Giải thích chi tiết cho từng yêu cầu

1. Yêu cầu 1a

Để đọc dữ liệu cho các tập train và test, **lưu ý:** do tham số truyền vào là tên của tập dữ liệu, do đó, dữ liệu cần được nằm cùng thư mục chứa file source code. Nếu không, chương trình sẽ phát sinh lỗi (không đọc được dữ liệu).

Thực hiện đọc dữ liệu (đọc 11 đặc trưng đầu) cho tập train và test:

- Sử dụng hàm **iloc** trên tập train và test ban đầu, ta thu được 4 biến là *X_train_1a, y_train_1a, X_test_1a, y_test_1a.*

Gọi hàm **fit** để thực hiện huấn luyện mô hình trên tập *X_train_1a, y_train_1a*.

Gọi hàm **get_params** để lấy trọng số của mô hình sau khi được huấn luyện.

Khi đó, ta có được công thức hồi quy như sau (trọng số được làm tròn đến số thập phân thứ 3).

Salary = 49248.089 – 23183.329 × *Gende*r + 702.766 × *10percentage* + 1259.019 × *12percentage* – 99570.608 × *CollegeTier* + 18369.962 × *Degree* + 1297.532 × *collegeGPA* – 8836.727 × *CollegeCityTier* + 141.759 × *English* + 145.742 × *Logical* + 114.643 × *Quant* + 34955.750 × *Domain*.

Gọi hàm **mae** (đã được cài đặt) trên tập kiểm tra, ta thu được kết quả MAE cho yêu cầu này là:

MAE ¿ 105 052.529

2. Yêu cầu 1b

Trong yêu cầu này, em sử dụng **KFold** (trong thư viện **sklearn.model_selection**) với numFold = 5 (Dữ liệu đã được xáo trộn).

Với numFold = 5, dữ liệu được chia thành 5 nhóm.

Ta duyệt lần lượt các đặc trưng mà đề bài yêu cầu (5 đặc trưng gồm conscientiousness, agreeableness, extraversion, neuroticism, openness_to_experience), lấy dữ liệu của đặc trưng tương ứng trên tập X_train (*X_train_feature*) và thực hiện tiền xử lý bằng hàm **preprocess**, ta thu được *X_train_feature_pre*.

Gọi hàm **split** cho tập *X_train_feature_pre,* ta thực hiện huấn luyện mô hình tương ứng. Với mỗi lần huấn luyện, ta tính tổng của độ lỗi tuyệt đối trung bình của nó (*sum*)

Tính độ lỗi tuyệt đối trung bình (avgMae) của đặc trưng bằng công thức avgMae = sum / numFold

Thêm đặc trưng cùng độ lỗi tương ứng của nó vào danh sách *listMae*.

Sau khi duyệt qua hết các đặc trưng, ta thực hiện sắp xếp *listMae* theo độ lỗi tăng dần (*sortedMae)*.

Khi đó, đặc trưng tốt nhất sẽ là đặc trưng có độ lỗi bé nhất (phần tử đầu tiên của *sortMae*.

Ta thu được bảng báo cáo sau:

STT	Mô hình với 1 đặc trưng	MAE
1	nueroticism	123473.399787

2	agreeableness	123706.054730
3	extraversion	123809.926200
4	openess_to_experience	123818.333575
5	conscientiousness	124182.563823

➔ Ta có thể kết luận, đặc trưng tốt nhất chính là **nuerotictism**

Thực hiện huấn luyện mô hình *best_personality_feature_model* với đặc trưng tốt nhất trên toàn bộ tập huấn luyện. Khi đó, ta thu được công thức hồi quy sau:

Salary = 304647.552 – 16021.494 × *neuroticism*

Gọi hàm mae (đã cài đặt) trên tập kiểm tra với mô hình *best_personality_feature_model*, ta thu được:

MAE = 119361.917

3. Yêu cầu 1c

Thực hiện tương tự như ở **yêu cầu 1b**, tuy nhiên, tập đặc trưng được xét lúc này sẽ là 3, gồm: *English, Logical, Quant*.

Đặt numFold = 5, thực hiện tương tự như trên, ta thu được bảng báo cáo sau:

STT	Mô hình với 1 đặc trưng	MAE
1	Quant	117353.838
2	Logical	119932.503
3	English	120728.603

➔ Ta có thể kết luận, đặc trưng tốt nhất chính là Quant.

Thực hiện huấn luyện mô hình *best_skill_feature_model* với đặc trưng tốt nhất trên toàn bộ tập huấn luyện. Khi đó, ta thu được công thức hồi quy sau:

Salary = 117759.729 + 368.852 × *Quant*

Gọi hàm mae (đã cài đặt) trên tập kiểm tra với mô hình *best_skill_feature_model*, ta thu được:

MAE = 108814.059

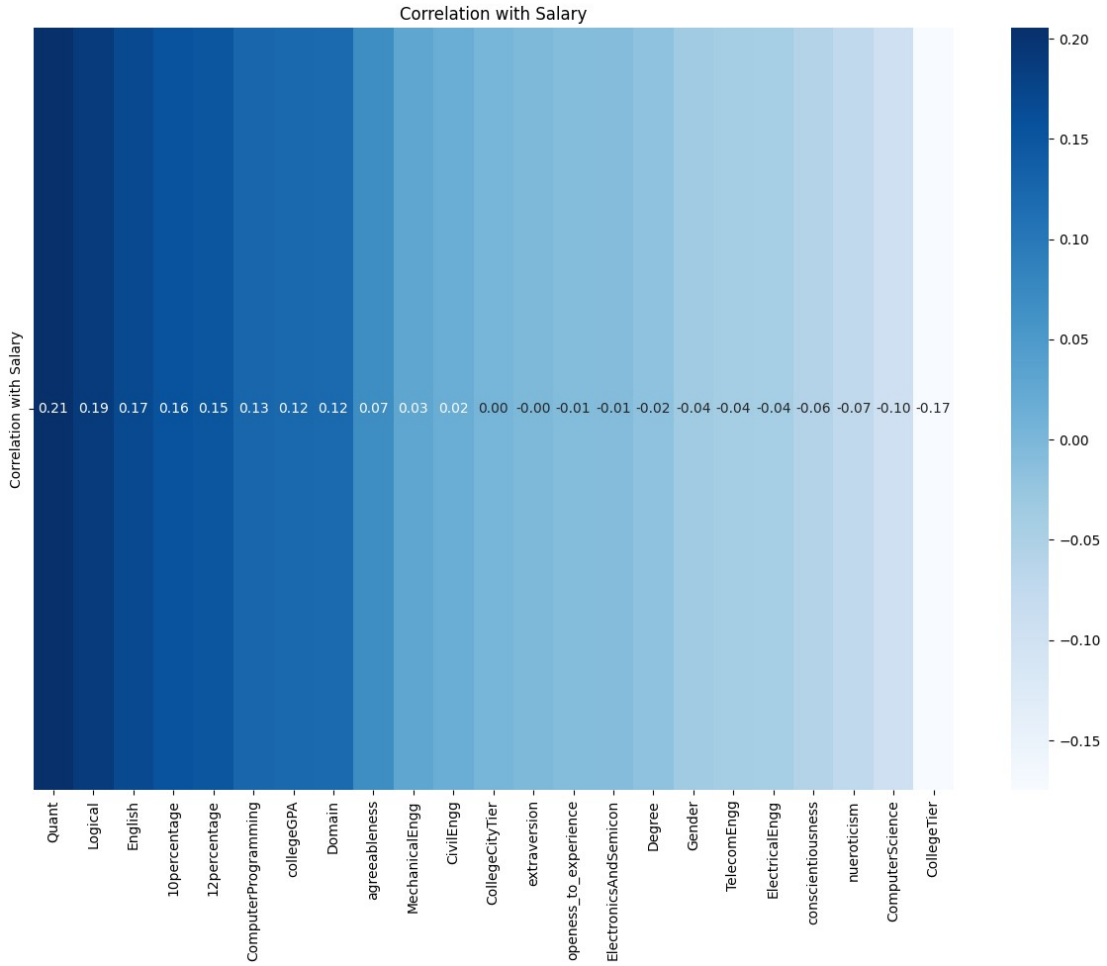
4. Yêu cầu 1d

a. Ma trận tương quan

Để thuận tiện cho việc đưa ra mô hình hợp lý, ta cần quan sát được mối tương quan giữa các đặc trưng với biến mục tiêu trong tập train.

Sử dụng biểu đồ **heatmap**, độ tương quan giảm dần, ta thu được một biểu đồ tương quan như bên dưới.

- Độ tương quan cao: Đây là các đặc trưng có khả năng ảnh hưởng cao đến biến mục tiêu *Salary*. Việc lựa chọn các đặc trưng này có thể cải thiện hiệu suất dự đoán của mô hình.
- Độ tương quan thấp: Là các đặc trưng có độ tương quan gần bằng 0 đối với biến mục tiêu *Salary*. Nó có thể đóng góp (không đáng kể) hoặc không đối với hiệu suất dự đoán của mô hình.
- Độ tương quan âm: Các đặc trưng có chỉ số độ tương quan với biến mục tiêu *Salary* là số âm có thể gây bất lợi trong việc dự đoán của mô hình. Tuy nhiên, để có thể kết luận được điều này hay không, cần xác minh dữ liệu chính xác.



Từ biểu đồ tương quan trên, ta có thể thấy rằng 2 đặc trưng *Quant* và *Logical* có độ tương quan lớn nhất đối với biến mục tiêu là *Salary*.

Dựa vào điểm này, em thực hiện xây dựng các mô hình xung quanh 2 đặc trưng đó và kết hợp với các đặc trưng khác (được trình bày rõ ở phần b).

b. Các mô hình tìm được

❖ Mô hình 1

Sử dụng 11 biến đặc trưng (*10percentage, 12percentage, ..., ComputerProgramming* theo thứ tự từ tập train ban đầu) kết hợp cùng một biến đặc trưng mới có tên là *Quant_Logical*.

Biến *Quant_Logical* là một biến được tạo ra từ phép nhân giữa *Quant* và *Logical.*

Theo em, do 2 biến *Quant* và *Logical* có mối tương quan lớn đối với *Salary*, do đó, việc nhân 2 đặc trưng lại để tạp thành 1 đặc trưng mới sẽ cho ra hiệu suất tốt trong việc dự đoán mô hình.

❖ Mô hình 2

Thu gọn kích thước của biến đặc trưng lại, khi đó danh sách các biến đặc trưng cần xét chỉ còn 8, đây là các biến đặc trưng có độ tương quan cao nhất đối với biến mục tiêu *Salary*. Danh sách biến đặc trưng lúc này là: *10percentage, 12percentage, collegeGPA, English, Logical, Quant, Domain, ComputerProgramming.*

Bên cạnh đó, xây dựng thêm 3 biến mới là:

- *Logical_square*: Biến này có được từ việc bình phương đặc trưng *Logical*.
- *Quant_square*: Biến này có được từ việc bình phương đặc trưng *Quant*.
- *Quant_Logical*: Biến này có được từ việc nhân 2 đặc trưng là *Quant* và *Logical*.

❖ Mô hình 3

Tiếp tục thực hiện thu gọn kích thước của biến đặc trưng lại, lần này, danh sách chỉ còn lại 5 biến đặc trưng có độ tương quan cao nhất đối với biến mục tiêu *Salary*. Danh sách biến đặc trưng lúc này chỉ còn lại: *Quant, Logical, English, 10percentage, 12percentage.*

Bên cạnh đó, xây dựng thêm 4 biến mới là:

- *Logical_cubed*: Biến có được từ lập phương đặc trưng *Logical.*
- *Quant_cubed*: Biến có được từ lập phương đặc trưng *Quant.*
- *Logical_square*: Biến có được từ bình phương đặc trưng *Logical.*
- *Quant_square*: Biến có được từ bình phương đặc trưng *Quant.*

c. Thử nghiệm so sánh các mô hình

Từ 3 mô hình được xây dựng trên, ta thực hiện thử nghiệm để so sánh hiệu suất của các mô hình.

Sử dụng **k-fold Cross Validation** để tiến hành thử nghiệm (tương tự như yêu cầu 1b, 1c). Khi đó, ta thu được bảng báo cáo sau:

STT	Mô hình	MAE
1	Mô hình 1	113274.019
2	Mô hình 2	113301.802
3	Mô hình 3	114880.105

➔ Từ bảng báo cáo trên, ta có thể kết luận rằng mô hình tốt nhất là **mô hình 1** với MAE bé nhất (trong 3 mô hình)

Từ kết quả trên, ta thực hiện huấn luyện mô hình *my_best_model,* ta thu được công thức hồi quy sau:

Salary = 18592.171 + 638.262 × 10percentage + 1017.65412 × percentage + 1275.906 × collegeGPA + 155.410 × English - 1402.431 ×Logical + 494.695 ×Quant + 24961.646 × Domain + 68.846 ×ComputerProgramming

Gọi hàm **mae** (tự cài đặt) trên tập kiểm tra với mô hình *my_best_model*, ta thu được

MAE = 104321.784

d. Kết luận

Từ các mô hình được xây dựng, em có nhận xét rằng:

- Độ tương quan giữa biến đặc trưng và biến mục tiêu có ảnh hưởng lớn đến hiệu suất dự đoán mô hình.
- Do đó, để có thể xây dựng được mô hình tốt, cần cân nhắc lựa chọn, kết hợp, các biến đặc trưng có độ tương quan đáng kể đối với biến mục tiêu

VI. Tài liệu tham khảo

https://trituenhantao.io/kien-thuc/gioi-thieu-ve-k-fold-cross-validation/

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html