

ĐẠI HỌC QUỐC GIA TP HCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

Đồ án 3: Linear Regression

Môn học: Toán ứng dụng và thống kê cho Công Nghệ Thông Tin

Sinh viên thực hiện:

Lê Trọng Đức Anh (21127005)

Giáo viên hướng dẫn:

Phan Thị Phương Uyên

Lê Thanh Tùng

Vũ Quốc Hoàng

Ngày 24 tháng 8 năm 2023



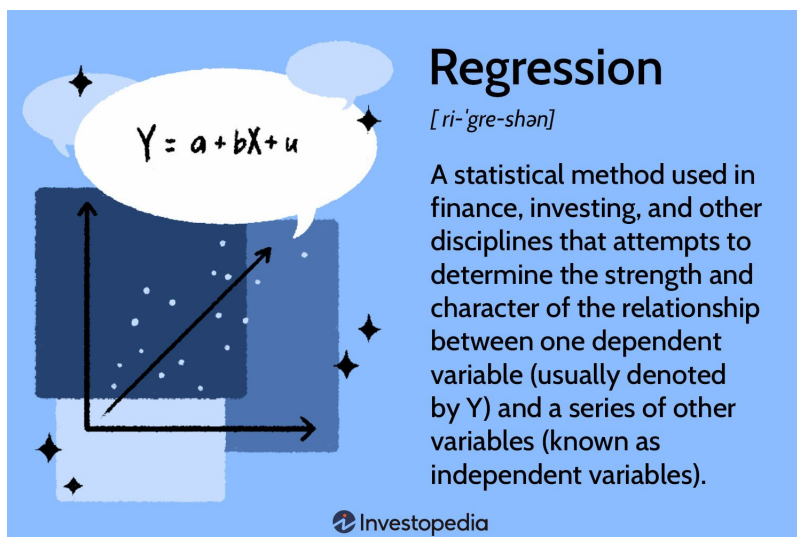
Mục lục

1	Giới thiệu đồ án	2
1.1	Bối cảnh	2
1.2	Hồi quy và hồi quy tuyến tính	2
1.3	Yêu cầu đồ án	3
2	Các thư viện và hàm sử dụng	4
2.1	Thư viện	4
2.1.1	Scikit-learn (sklearn)	4
2.1.2	Numpy	4
2.1.3	Pandas	5
2.1.4	Mathplotlib	5
2.2	Hàm sử dụng	6
3	Kết quả và nhận xét	8
3.1	Câu 1a	8
3.2	Câu 1b	8
3.3	Câu 1c	10
3.4	Câu 1d	11
4	Giả thuyết và giải thích các mô hình	13
4.1	Model 1 : 5 đặc trưng về tính cách	18
4.2	Model 2: 3 đặc trưng về tư duy	19
4.3	Model 3: Sử dụng toàn bộ đặc trưng được cho	19
4.4	Model 4: mô hình phụ thuộc vào đặc tính xã hội của Ấn Độ	20
4.5	Model 5: Áp dụng phương pháp chọn lọc thuộc tính	21
4.6	Model 6: Tối ưu hóa model 5 bằng Huber regression (Huber loss)	21
5	Nhận xét chung	24
6	Tài liệu tham khảo	25

1 Giới thiệu đồ án

1.1 Bối cảnh

Trong ngành Trí Tuệ Nhân Tạo nói riêng và Khoa Học Máy Tính nói chung, Machine Learning (ML) là một thuật ngữ để chỉ việc "dạy" cho máy tính những thuật toán, cách giải quyết vấn đề theo cách tư duy của "con người". Trong ML có phân ra thành nhiều nhánh khác nhau: Supervised Learning (học có giám sát), Unsupervised Learning (học không giám sát), Reinforcement Learning (học tăng cường). Mỗi nhánh được sử dụng trong những tình huống cụ thể đề bài yêu cầu và có những ưu và nhược điểm riêng. Trong đó, Supervised Learning tỏ ra là nhánh dễ tiếp cận nhất đối với người vừa bước chân vào AI khi trong phương pháp học này ta biết được nhãn (label) của dữ liệu. Điều ta cần làm là cố gắng chỉ cho máy cách để tính nhãn đó một cách sát với thực tế nhất. Có rất nhiều mô hình thuật toán để làm điều này như: Decision Tree (cây quyết định), Neural Networks (mạng neural), Regression (hồi quy). Trong số những thuật toán đó, Regression tỏ ra là một công cụ mạnh mẽ để giải quyết những dữ liệu có tính liên tục của hàm số.

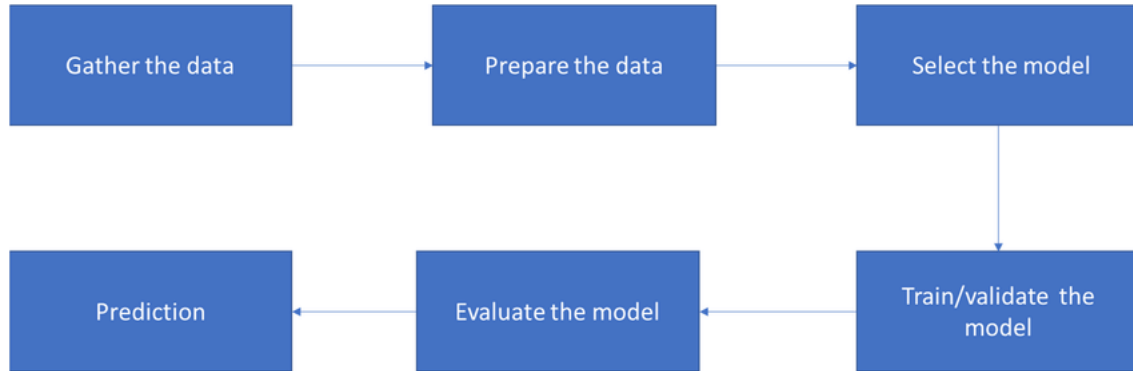


1.2 Hồi quy và hồi quy tuyến tính

Regression là một trong những thuật toán mạnh mẽ nhất khi giải quyết các bài toán thuộc dạng data fitting. Mục tiêu của nó là tìm ra đường "phù hợp" nhất cho tập dữ liệu. Dù đường đó là thẳng, cong hay xiên thì mục tiêu cuối cùng cũng là tìm hệ số của hàm kết quả sao cho "gần" với dữ liệu nhất. Để giải quyết một bài toán bằng Regression ta cũng cần tuân theo những bước sau:

- Thu thập dữ liệu
- Chuẩn bị dữ liệu. Xóa bỏ những dữ liệu thừa và gây nhiễu.
- Chọn model phù hợp cho dữ liệu. Ở đây là chọn kiểu Regression phù hợp với phân phối của dữ liệu.
- Huấn luyện dữ liệu (train/test)
- Đánh giá model

- Dự đoán kết quả (predict)



1.3 Yêu cầu đồ án

Ta sẽ được đưa cho tập dữ liệu là các yếu tố quyết định tới mức lương của các kỹ sư 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 mức lương nhận được hàng tháng của nhân viên. 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. Sau khi chọn lọc thì bộ dữ liệu của chúng ta sẽ gồm 23 thuộc tính tự do và 1 thuộc tính phụ thuộc (Salary). Nhiệm vụ trong đồ án lần này là tìm ra mô hình hồi quy tốt nhất để dự đoán mức lương của các kỹ sư.



2 Các thư viện và hàm sử dụng

2.1 Thư viện

2.1.1 Scikit-learn (sklearn)

Là một thư viện miễn phí cho ngôn ngữ Python dùng cho các bài toán Machine Learning. Sklearn có đa dạng các thuật toán Classification, Regression, Clustering bao gồm K-means, Gradient Boosting, Linear Regression,...



Các hàm và module chúng ta sẽ sử dụng gồm:

- **sklearn.linear_model.LinearRegression**: thực hiện thuật toán hồi quy tuyến tính theo kiểu Ordinary Least Square. Ta sẽ sử dụng những phương thức sau của lớp này.
 - `fit(X,y)`: thực hiện fit model tuyến tính với giá trị biến tự do X và biến phụ thuộc là y
 - `predict(X)`: thực hiện dự đoán kết quả của X sau khi đã fit model
 - `coef_`: trả về hệ số của hàm hồi quy
- **sklearn.linear_model.HuberRegressor**: tương tự như thuật toán Linear Regression nhưng có chú ý đến outliers để tránh gây ra 'ngiên' đường tuyến tính
- **sklearn.metrics.mean_absolute_error**: Tính Mean Absolute Error và 2 kết quả. Trả về một số thực
- **sklearn.feature_selection.SelectKBest**: Chọn k đặc tính tốt nhất dựa trên hàm đánh giá truyền vào (trong đồ án này là `f_regression`)
- **sklearn.feature_selection.f_regression**: Tính mức độ ảnh hưởng của biến độc lập đến biến phụ thuộc

2.1.2 Numpy

Là thư viện dành cho ngôn ngữ Python dùng để tính toán ma trận đơn hay nhiều chiều. Ngoài ra, thư viện này còn bao gồm nhiều hàm tính toán trên ma trận với tốc độ ngang với C và C++, điều mà Python hạn chế.



Các hàm và module chúng ta sẽ sử dụng gồm:

- `numpy.mean`: Tính trung bình của một mảng. Được sử dụng để tính trung bình MAE trong thuật toán K-fold.
- `numpy.ndarray.flatten`: Làm phẳng ma trận cột thành ma trận dòng.

2.1.3 Pandas

Là thư viện miễn phí của ngôn ngữ Python cung cấp khả năng đọc xuất file dữ liệu, thao tác trên dữ liệu và lập các thống kê.



Trong thư viện này chúng ta sẽ chủ yếu đọc xuất file `.csv` và truy xuất các cột dữ liệu

- `pandas.read_csv`: Đọc file `.csv`
- `pandas.DataFrame.iloc`: Lấy những cột dữ liệu theo chỉ số.
- `pandas.DataFrame.loc`: Lấy những cột dữ liệu theo tên cột

2.1.4 Matplotlib

Là thư viện vẽ các biểu đồ phác họa dựa trên thư viện Numpy (2.1.2). Nó hỗ trợ API theo kiểu hướng đối tượng để phù hợp với cả GUI toolkits như Tkinter, wxPython, Qt, hay GTK.



Trong thư viện này chúng ta sẽ vẽ các biểu đồ phân bố những điểm trong tập dữ liệu và đường hồi quy. Ngoài ra nó còn được dùng để vẽ bar chart trong lúc chọn thuộc tính.

- `matplotlib.pyplot.plot`: Plot dữ liệu trên hệ **Oxy**
- `matplotlib.pyplot.bar`: Plot dữ liệu theo kiểu bar chart

2.2 Hàm sử dụng

`choose_model(models,train_org,k=10,display_model=True)`: Hàm chọn model theo K-fold Cross-Validation cho câu **1b,1c,1d**. In ra **MAE** trung bình cho từng model.

- Parameters:
 - **Models: list**
Mảng chứa tập các model. Mỗi model là một tuple chứa kiểu hồi quy sử dụng và một mảng chứa các thuộc tính sử dụng.
 - **train_org: DataFrame**
Tập train từ bên ngoài truyền vào để chạy thuật toán
 - **k: int**
Số lượng tập con khi chia tập train
 - **display_model: Boolean**
Nếu là True thì có in ra model. Nếu không thì chỉ đánh số thứ tự

Thuật toán: Đầu tiên xáo trộn toàn bộ dữ liệu của tập train. Sau đó tiến hành trích xuất từ tập train ra k phần bằng nhau (gọi là A). Ứng với mỗi model, lấy trong A từng bộ train sau đó huấn luyện và tính MAE độc lập. Kế tiếp, tính trung bình các MAE đó và in ra kết quả cho model. Ta sẽ sử dụng giá trị trung bình đó để xác định model nào tốt (MAE càng nhỏ thì càng tốt).

`selectFeatures(X_train,Y_train)`: Hàm chọn và đánh giá thuộc tính tốt nhất để sử dụng cho thuật toán hồi quy. Dùng hàm `SelectKBest` (2.1.1) và hàm đánh giá `f_regression` (2.1.1) để xác định độ tốt của thuộc tính. Điểm đánh giá càng cao thì thuộc tính đó càng có nhiều ảnh hưởng trong thuật toán hồi quy

- Parameters:

- **X_train: array_like**

Chứa tập dữ liệu train của toàn bộ thuộc tính

- **Y_train: array_like**

Chứa thông tin của đặc tính phụ thuộc (ở đây là Salary)

- Returns:

- **fs: object**

Trả về **self** object của SelectKBest

Thuật toán: Sử dụng `f_regression` là hàm đánh giá sau đó `fit(X_train, Y_train)`

3 Kết quả và nhận xét

3.1 Câu 1a

Các bước thực hiện:

- Lấy 11 đặc tính theo đề yêu cầu (11 thuộc tính đầu tiên) trong tập **train** và **test**
- Huấn luyện mô hình hồi quy tuyến tính trên tập dữ liệu vừa lấy sử dụng lớp `LinearRegression` (2.1.1)
- In ra hệ số bằng thuộc tính `coef_` trong lớp `LinearRegression` (2.1.1). Công thức hồi quy cuối cùng là:

$$\begin{aligned} \text{Salary} = & -23183.330 \cdot \text{Gender} + 702.767 \cdot 10\text{percentage} + 1259.019 \cdot 12\text{percentage} \\ & - 99570.608 \cdot \text{CollegeTier} + 18369.962 \cdot \text{Degree} + 1297.532 \cdot \text{collegeGPA} \\ & - 8836.727 \cdot \text{CollegeCityTier} + 141.760 \cdot \text{English} + 145.742 \cdot \text{Logical} \\ & + 114.643 \cdot \text{Quant} + 34955.750 \cdot \text{Domain} \end{aligned}$$

- Tính MAE khi predict trên tập **test** bằng hàm `mean_absolute_error` trong `sklearn` (2.1.1). Kết quả là:

$$MAE = 105052.529$$

Nhận xét: đây là mô hình chung khi giải quyết các bài toán hồi quy. Đầu tiên tách dữ liệu cần dùng. Sau đó fit data vào mô hình. Cuối cùng là tính hệ số và in ra độ lỗi của mô hình. Hệ số trong hàm kết quả cho biết mức độ tương quan của biến độc lập với biến phụ thuộc. Nếu hệ số là dương thì thuộc tính đó với mức lương có tương quan dương (càng tăng thì lương càng tăng). Nếu hệ số là âm thì thuộc tính đó với mức lương có tương quan âm (càng tăng thì lương càng giảm). MAE cho biết mức lương ta dự đoán chênh lệch như thế nào đối với thực tế. Nói cách khác, khi dự đoán mức lương thì mô hình này sai lệch khoảng 105052 (đvt) so với thực tế.

3.2 Câu 1b

Các bước thực hiện:

- Tách 5 đặc tính ra và xem như 5 model khác nhau.
- Gọi hàm `choose_model` để in ra kết quả MAE trung bình của từng thuộc tính khi chạy k-fold Cross-Validation (2.2)

Thuộc tính	MAE
conscientiousness	124656.449
agreeableness	123623.709
extraversion	124192.349
neroticism	123748.285
openness_to_experience	124330.457

Bảng 1: Bảng so sánh các thuộc tính câu 1b

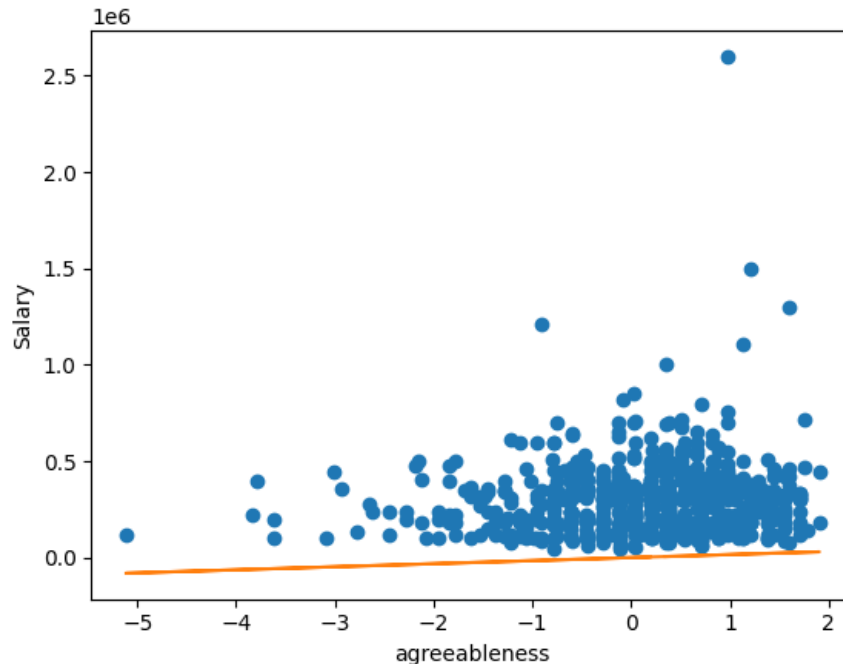
- Chọn ra thuộc tính có MAE nhỏ nhất (ở đây là **agreeableness**)
- Lấy dữ liệu của thuộc tính **agreeableness** trong tập **train** và **test**
- Huấn luyện mô hình hồi quy tuyến tính trên thuộc tính **agreeableness**.
- In ra hệ số bằng thuộc tính `coef_` trong lớp `LinearRegression` (2.1.1). Công thức hồi quy cuối cùng là:

$$Salary = 15834.939 \cdot agreeableness$$

- Tính MAE khi predict trên tập **test** bằng hàm `mean_absolute_error` trong `sklearn` (2.1.1). Kết quả là:

$$MAE = 118153.163$$

- Vẽ đường hồi quy trên tập **test** để xem mức độ ảnh hưởng của hàm.



Nhận xét: câu 1b giống với câu 1a ngoài việc ta phải chọn đặc tính tốt nhất trước khi huấn luyện cho mô hình. Thuật toán K-fold Cross-Validation đã được trình bày chi tiết trong phần 2.2. Nhờ vào kết quả thu được thì ta xác định được rằng **agreeableness** và **neroticism** là 2 thuộc tính tốt nhất trong mô hình (tùy vào dữ liệu được shuffle mà sẽ có trường **neroticism** tốt hơn **agreeableness**). Tiếp đó ta dùng đặc tính này để huấn luyện cho mô hình hồi quy và ra được hệ số và kết quả. Kết quả cho biết, nếu xét theo điểm số trong phần thi tính AMCAT của **agreeableness** thì kết quả dự đoán sẽ sai lệch khoảng 118153 so với mức lương thực tế nhận được. Nhìn vào biểu đồ ta thấy được, mức lương chủ yếu trong tập test nằm trong khoảng 100000 - 500000 và đường hồi quy của chúng ta cũng khá gần so với "cụm" này.

3.3 Câu 1c

Các bước thực hiện:

- Tách 3 đặc tính **English**, **Logical** và **Quant** ra và xem như 3 model khác nhau.
- Gọi hàm `choose_model` để in ra kết quả MAE trung bình của từng thuộc tính khi chạy k-fold Cross-Validation (2.2)

Thuộc tính	MAE
English	120776.275
Logical	120591.901
Quant	117527.168

Bảng 2: Bảng so sánh các thuộc tính câu 1c

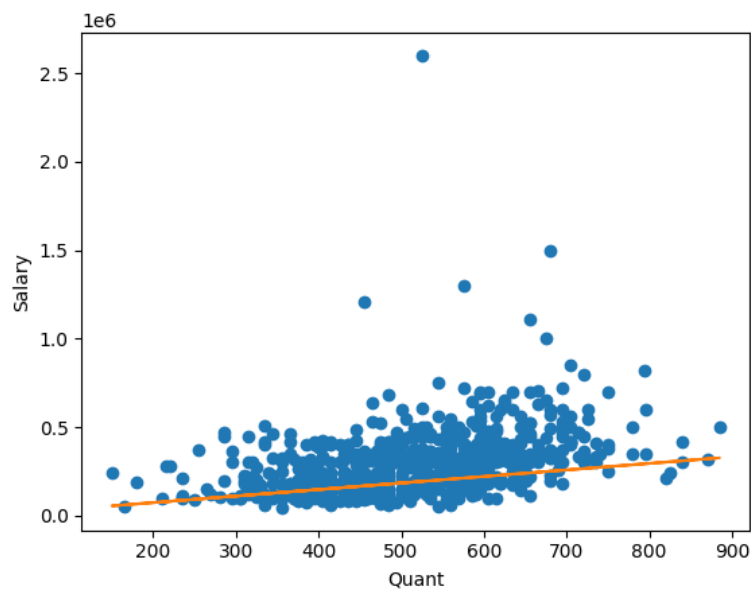
- Chọn ra thuộc tính có MAE nhỏ nhất (ở đây là **Quant**)
- Lấy dữ liệu của thuộc tính **Quant** trong tập **train** và **test**
- Huấn luyện mô hình hồi quy tuyến tính trên thuộc tính **Quant**.
- In ra hệ số bằng thuộc tính `coef_` trong lớp `LinearRegression` (2.1.1). Công thức hồi quy cuối cùng là:

$$Salary = 368.852 \cdot Quant$$

- Tính MAE khi predict trên tập **test** bằng hàm `mean_absolute_error` trong `sklearn` (2.1.1). Kết quả là:

$$MAE = 108814.059$$

- Vẽ đường hồi quy trên tập test để xem mức độ ảnh hưởng của hàm.

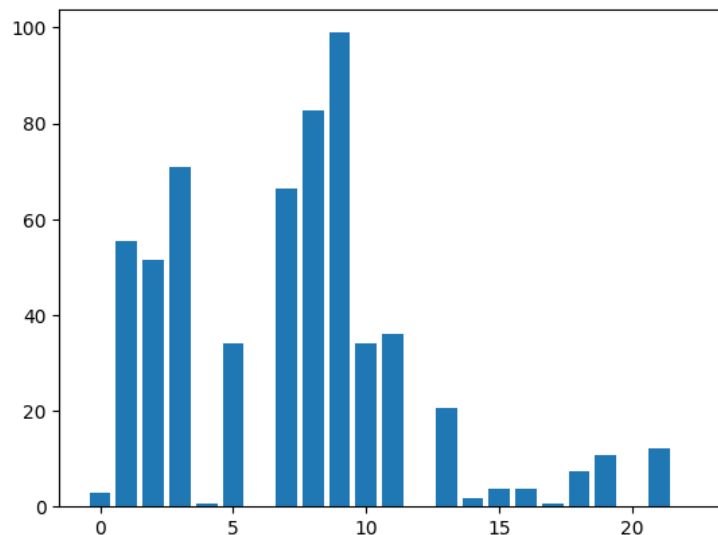


Nhận xét: câu 1b tương tự như câu 1c. Ta cũng chọn đặc tính tốt nhất trong các đặc tính rồi sử dụng đặc tính đó để huấn luyện mô hình thật. Kết quả cho thấy, nếu sử dụng. Kết quả cho biết, nếu xét theo điểm số trong phần thi tính AMCAT của **Quant** thì kết quả dự đoán sẽ sai lệch khoảng 108814 so với mức lương thực tế nhận được và là một quả khá hơn đặc tính **agreeableness** ở câu 1b. Bằng chứng là nhìn vào biểu đồ thì đường hồi quy giờ đây đã nằm giữa tất cả các điểm dữ liệu.

3.4 Câu 1d

Các bước thực hiện:

- Với mô hình đầu tiên, ta đơn giản là chọn ra tất cả các đặc tính được cho.
- Với mô hình thứ 2, ta sẽ dựa vào tình hình thực tế ở Ấn Độ và chọn ra mô hình gồm các đặc tính: **10percentage, collegeGPA, Quant, Domain, ComputerProgramming, ComputerScience**. Phần giải thích lí do sẽ nằm ở phần sau
- Với mô hình 3, ta sẽ làm bước chọn những thuộc tính tốt nhất để sử dụng bằng phương pháp **Correlation Feature Selection**. Cụ thể như sau:
 - Sử dụng hàm **selectFeatures** đã được định nghĩa trong phần 2.2 để tính độ quan trọng của thuộc tính.
 - In ra mức độ quan trọng của từng thuộc tính.
 - Vẽ biểu đồ thể hiện mức độ quan trọng.



- Từ biểu đồ và kết quả, ta sẽ lấy những thuộc tính có mức độ quan trọng từ 10 trở lên: **10percentage,12percentage,CollegeTier,collegeGPA,English,Logical,Quant,Domain,ComputerScience,ComputerProgramming,agreeableness**
- Gọi hàm **choose_model** để in ra kết quả MAE trung bình của từng model đã định nghĩa ở trên khi chạy k-fold Cross-Validation (2.2)

Model	MAE
Model 1	110808.132
Model 2	112323.724
Model 3	111606.024
Model 4	107914.455

Bảng 3: Bảng so sánh các model câu 1d

- Chọn ra model có MAE nhỏ nhất (ở đây là **model 4**)
- Lấy dữ liệu của **model 4** trong tập **train** và **test**
- In ra hệ số bằng thuộc tính `coef_` trong lớp `LinearRegression` (2.1.1). Công thức hồi quy cuối cùng là:

$$\begin{aligned} \text{Salary} = & 813.465 \cdot 10\text{percentage} + 618.183 \cdot 12\text{percentage} - 14887.593 \cdot \text{CollegeTier} \\ & 11.623 \cdot \text{collegeGPA} + 129.314 \cdot \text{English} + 11.924 \cdot \text{Logical} + \\ & 213.761 \cdot \text{Quant} + 7157.488 \cdot \text{Domain} - 129.371 \cdot \text{ComputerScience} + \\ & 98.721 \cdot \text{ComputerScience} + 8727.450 \cdot \text{agreeableness} \end{aligned}$$

- Tính MAE khi predict trên tập **test** bằng hàm `mean_absolute_error` trong `sklearn` (2.1.1). Kết quả là:

$$MAE = 100658.215$$

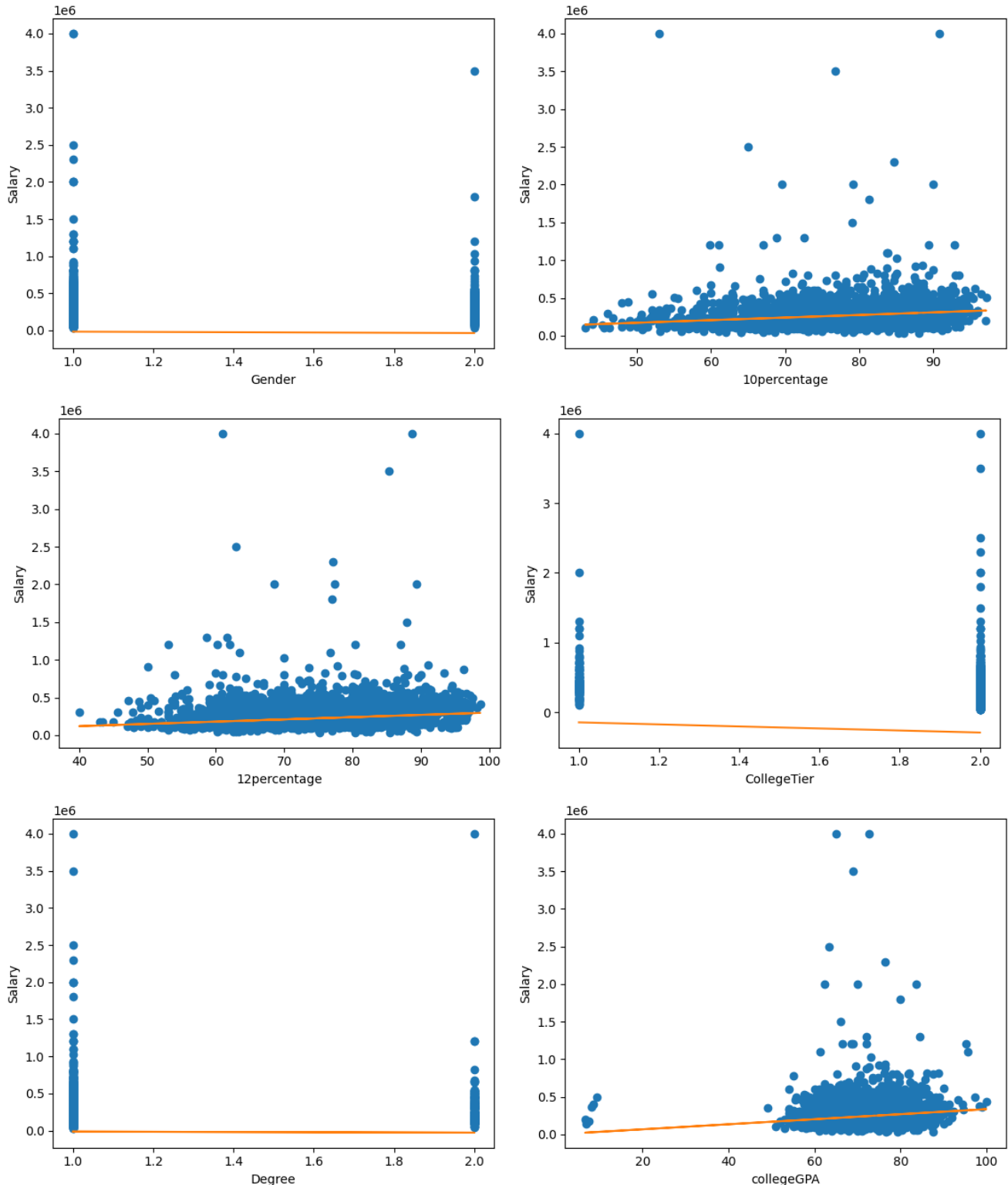
Nhận xét: từng bộ model cho ta từng góc nhìn khác nhau về tập dữ liệu.

- Về model 1: ta xét toàn bộ những thuộc tính được cho do chúng có ít nhiều ảnh hưởng đến kết quả.
- Về model 2: ta dựa vào tính hình kinh tế và chính trị cũng như những đặc điểm về nước Ấn Độ để quyết định những đặc tính này.
- Về model 3: Ta lọc lại những đặc tính nào tốt nhất để lấy và bỏ những đặc tính gây "nhiều". Đây gọi là bước **Feature Selection** trong các bài toán Machine Learning. Bước này giúp hàm kết quả ngắn gọn hơn nhưng không mất tính tổng quát.
- Về model 4: Ta tối ưu hóa Linear Regression ở model 3 bằng Huber loss để model của chúng ta ít bị ảnh hưởng bởi outliers hơn

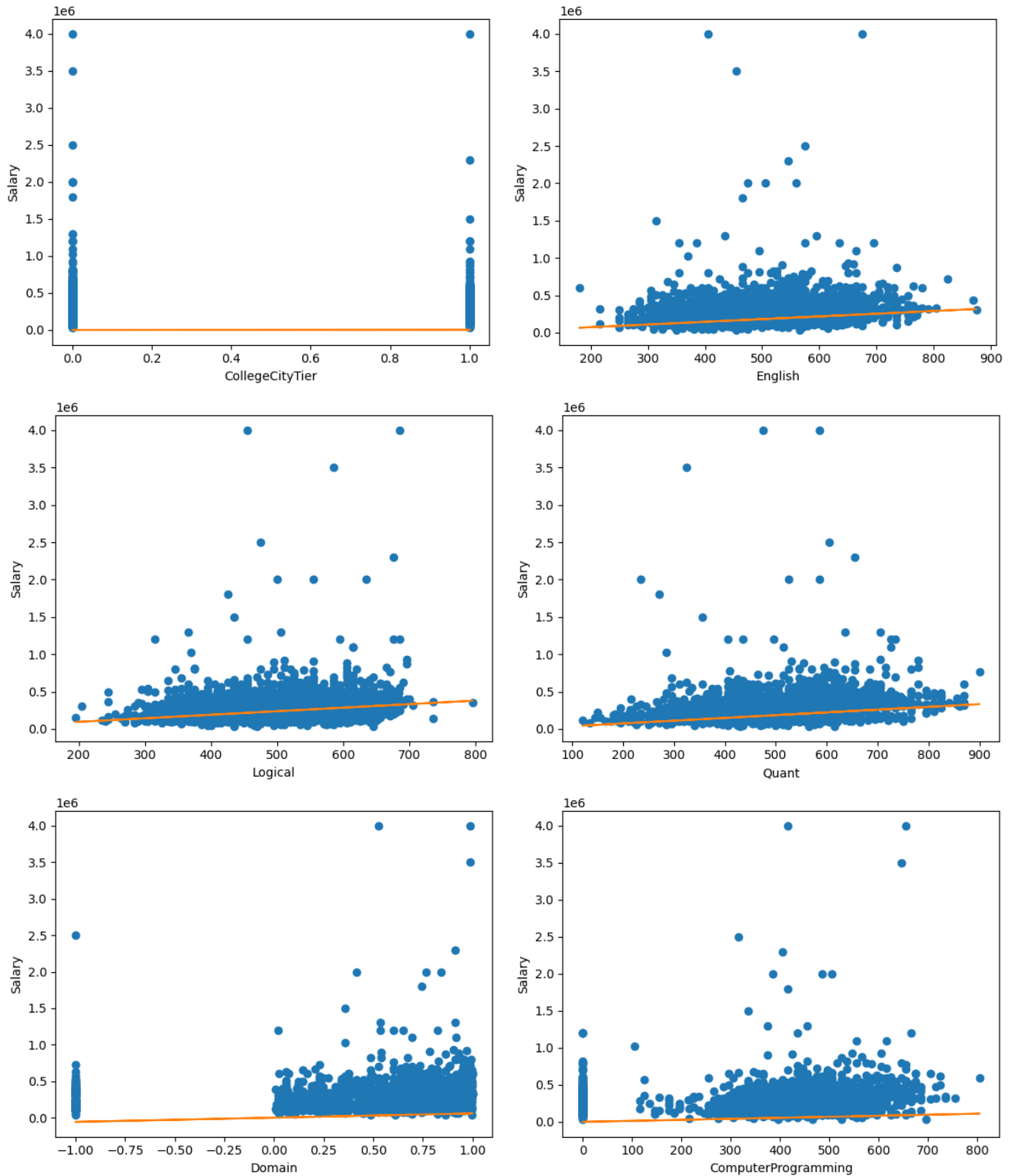
Tất cả những bằng chứng và giải thích cụ thể sẽ nằm ở phần sau. Quá trình tiếp theo cũng tương tự như câu 1b và 1c, ta sử dụng k-fold để lấy ra bộ model tốt nhất và sử dụng để test dữ liệu. Ta thấy model 4 là model tỏ ra tốt nhất trong các model mà chúng ta test. Thật vậy, MAE của model 1 là 107608 tốt nhất trong tất cả những model mà chúng ta đã test. Điều này nói rằng, nếu xét tất cả những đặc tính thì kết quả khi dự đoán bằng hàm hồi quy tuyến tính sai lệch khoảng 107608 (đvtt) so với thực tế.

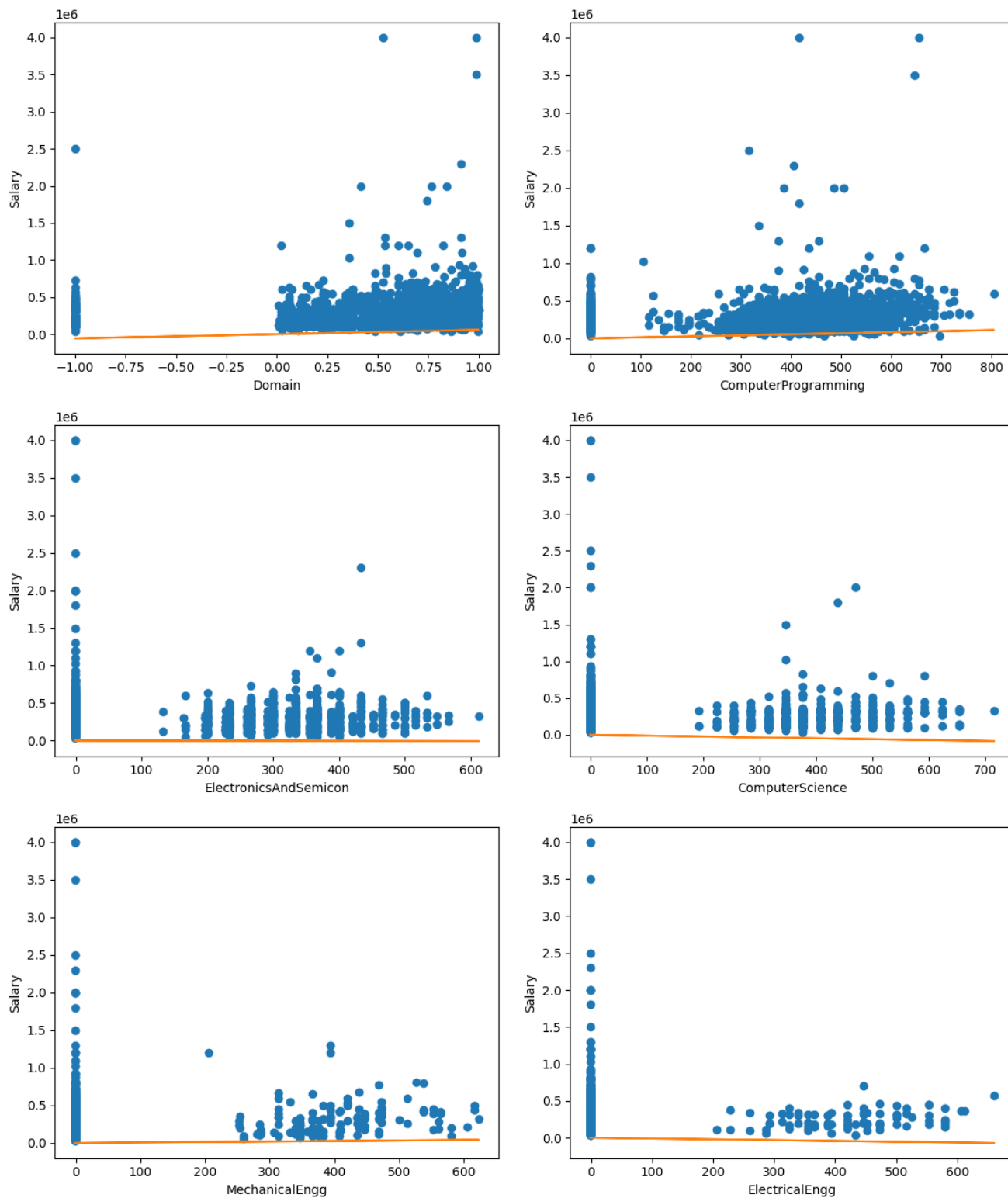
4 Giả thuyết và giải thích các mô hình

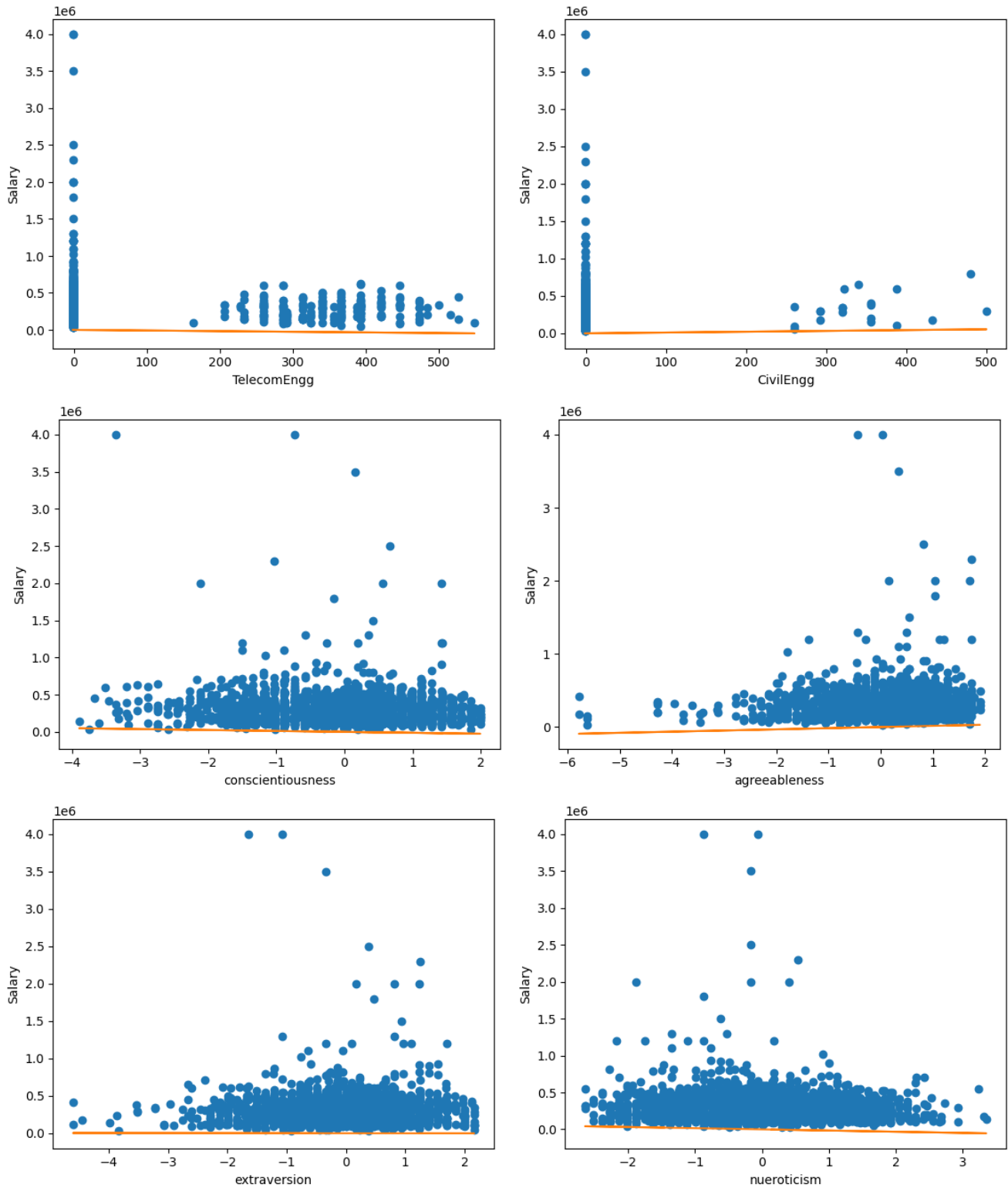
Trước khi giải thích về từng mô hình. Ta sẽ xem xét về phân bố dữ liệu của từng thuộc tính và đường hồi quy của nó (sẽ hơi dài mong thầy/cô sẽ xem hết)

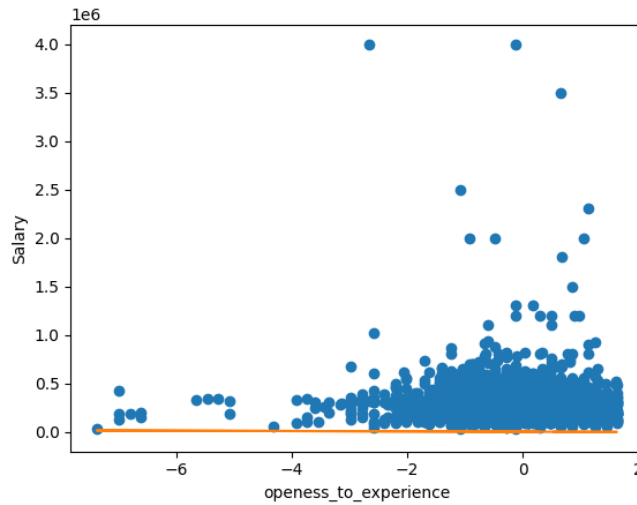


Đồ án 3: Linear Regression









Nhận xét: như biểu đồ cho thấy, ngoài những đặc tính theo kiểu phân loại như **Gender** hay **CollegeTier** thì tất cả đặc tính còn lại phân bố theo kiểu tuyến tính, thành 1 cụm liên tiếp nhau và trải dài trên gặp trục **Ox**. Do đó, mô hình tuyến tính bậc 1 (đường thẳng) hóa ra lại là một trong những mô hình tốt nhất để fit model. Nhiệm vụ của chúng ta bây giờ sẽ là chọn những đặc tính phù hợp nhất sao cho hàm mất mát MAE càng nhỏ càng tốt.

4.1 Model 1 : 5 đặc trưng về tính cách

Như ở phần trước đã đề cập, trong số 5 tính cách thì tính cách **agreeableness** là có mức ảnh hưởng đến lương nhất. Về mặt dữ liệu, ta thấy được **agreeableness** phù hợp với dữ liệu nhất khi chúng có xu hướng "chụm" lại với nhau nhiều hơn.

"Sự đồng tình là một tính cách của con người chỉ khả năng đặt lợi ích của người khác trước bản thân. Những người có xu hướng tán thành ý kiến của người khác thường thấu hiểu và tìm thấy niềm vui khi khi giúp đỡ và làm việc cùng người khác" ([link bài viết](#)). Từ đó, ta có thể hiểu được vì sao người thường tán thành và xem xét ý kiến của người khác thường đạt được mức lương cao hơn do họ có khả năng làm việc nhóm tốt và giúp đỡ người đồng nghiệp nên dễ được lòng mọi người.

Điều này ảnh hưởng rất lớn trong bối cảnh là Ấn Độ, một trong những quốc gia có mật độ dân số đông nhất trên thế giới và việc phân cấp bậc ảnh hưởng sâu sắc trong xã hội. Tuy nhiên, việc quá dễ tán thành ý kiến của người khác đôi khi sẽ khiến cho việc phản biện trở nên khó khăn hơn. Ngoài ra, giúp đỡ người khác không có nghĩa là lúc nào cũng được lòng họ và dễ dàng thăng tiến trong công việc vì chúng ta quá tập trung vào tiến độ của người khác và bị đặt vào tình thế khó phát triển bản thân hơn. Số liệu cũng cho thấy điều này khi trong tất cả những mô hình mà chúng ta huấn luyện thì **agreeableness** là đặc tính có MAE cao nhất.



4.2 Model 2: 3 đặc trưng về tư duy

Tương tự, ta đã biết được rằng trong 3 đặc tính là **English**, **Logical** và **Quant** thì **Quant** là đặc tính ảnh hưởng đến mức lương nhất. Mặt khác, **Quant** lại là đặc tính có mức độ quan trọng nhất trong tất cả 23 đặc tính theo như chúng ta đã thống kê khi lựa chọn đặc tính. Sơ đồ phân phối của **Quant** cũng là "đẹp" nhất trong tất cả các sơ đồ khi đường tuyến tính hầu như nằm ở trung tâm. Định lượng là khả năng thu thập thông tin và phân tích sau đó là đưa ra quyết định. Đây trên thực tế là một trong những kỹ năng quan trọng nhất của con người trong thời buổi kinh tế thị trường hiện nay, việc có thể dự đoán được giá cổ phiếu tăng hay giảm cũng có thể biến một người vô gia cư thành triệu phú. Khả năng này về cơ bản là "dự đoán tương lai" và phản ứng nhanh đối với môi trường. Những người có khả năng định lượng tốt có thể nhìn vào những biểu đồ thống kê hiện tại để đúc trích ra thông tin có ích (có vẻ giống với những gì chúng ta đang làm). Và họ không làm những điều đó một cách cảm tính mà dựa vào những lý thuyết xác suất thống kê và những phương trình toán học để đưa ra những dự đoán về xu hướng. Đặt mình vào trong đất nước Ấn Độ hiện nay khi thị trường trở nên rất nhộn nhịp thì lượng dữ liệu trao đổi hàng ngày rất lớn. Do đó không thể phủ nhận rằng khả năng định lượng là một trong những khả năng ảnh hưởng đến mức lương của lao động Ấn Độ nhất hiện nay.



4.3 Model 3: Sử dụng toàn bộ đặc trưng được cho

Một hướng tiếp cận cơ bản cho bài toán hồi quy đó là: lấy toàn bộ thuộc tính mà đề bài cho. Nhìn theo biểu đồ phân phối thì ta dễ thấy là bậc 1 là bậc tốt nhất cho bài toán hồi quy này. Nghe có vẻ ngớ ngẩn nhưng model này lại là model có MAE nhỏ nhất trong toàn bộ những model mà chúng ta đã train với MAE là 102443.780. Điều này có thể giải thích rằng việc đánh giá mức lương của một cá nhân không thể chỉ dựa vào một hay một vài đặc tính cơ bản mà xác định được do trên thực tế mức lương có thể bị ảnh hưởng bởi rất nhiều yếu tố khác nhau trong đời sống. Điều này tỏ ra đúng đối với môi trường cạnh tranh việc làm ở Ấn Độ khi đến cả giới tính cũng có thể ảnh hưởng ít nhiều đối với mức lương của bạn. Như đã đề cập, dân số của Ấn Độ rất đông và việc phân chia cấp bậc cũng rất sâu sắc trong xã hội ở Ấn Độ. Do đó, số lượng "tham số" để giải quyết bài toán dự đoán mức lương sẽ nhiều hơn so với ở những nước tự do như "Mỹ". Ngoài những đặc tính trong

dữ liệu mà chúng ta thu thập, có một vài đặc tính khác cũng ảnh hưởng đến mức lương như: sếp, số lượng bài báo cáo, số năm kinh nghiệm,... (xem thêm ở [Salary.com](https://www.salary.com))



4.4 Model 4: mô hình phụ thuộc vào đặc tính xã hội của Ấn Độ

Ở mô hình này, ta có những đặc tính sau: **10percentage**, **collegeGPA**, **Quant**, **Domain**, **ComputerProgramming**, **ComputerScience**. Những đặc tính này phụ thuộc hoàn toàn vào tính chất xã hội mà chúng ta đã đề cập. Giờ chúng ta sẽ phân tích từng đặc tính sau:

- **10percentage - tổng điểm trong kì thi lớp 10**: là một trong những kì thi quan trọng nhất ở Ấn Độ. Kì thi này quyết định xem một học sinh có thể tiếp tục theo đuổi con đường học vấn hay không. Và hầu hết cha mẹ ở Ấn Độ quan trọng điểm số của con trong lúc học hơn là những gì con của họ học được. Đó không phải là một vài gia đình mà là nền giáo dục của Ấn Độ có tính phân hóa rất cao giống với sự phân hóa giàu nghèo ở Ấn Độ.
- **collegeGPA - điểm GPA tại thời điểm tốt nghiệp**: tiếp là một điểm số trong học tập nữa của Ấn Độ. Đây là điểm tốt nghiệp của một sinh viên trước khi họ bắt tay để đi "rải CV" xin việc. Với một xã hội phân cấp theo học vấn như Ấn Độ thì không có gì lạ.
- **Quant - điểm trong phần thi định lượng**: không có gì lạ khi đặc tính này nằm ở đây khi, như những gì chúng ta đã trình bày, nó là đặc tính ảnh hưởng nhiều nhất đến mức lương so với những đặc tính khác. Xã hội phân hóa và mật độ dân số đông là chìa khóa giải thích vì sao khả năng định lượng lại quan trọng.
- **ComputerProgramming, ComputerScience - điểm trong phần thi lập trình, khoa học máy tính**: công nghệ máy tính là một trong những ngành phát triển nhất ở Ấn Độ. Nhiều năm trở lại đây, Ấn Độ luôn là đất nước hàng đầu tiên phong trong lĩnh vực công nghệ và khoa học máy tính nhờ sự đầu tư lớn toán và khoa học. Sự phát triển không ngừng nghỉ cả về phần cứng lẫn phần mềm không chỉ nằm ở nguyên nhân về hệ thống giáo dục mà còn là về văn hóa của cả một xã hội. Hệ thống giáo dục ở Ấn Độ kích thích sự cạnh tranh về điểm số như ta đã đề cập. Mặt khác, IT là một trong những ngành có lương "nghìn đô" ở Ấn Độ nên không bất ngờ khi có rất nhiều sinh viên chọn IT cho sự nghiệp của họ.



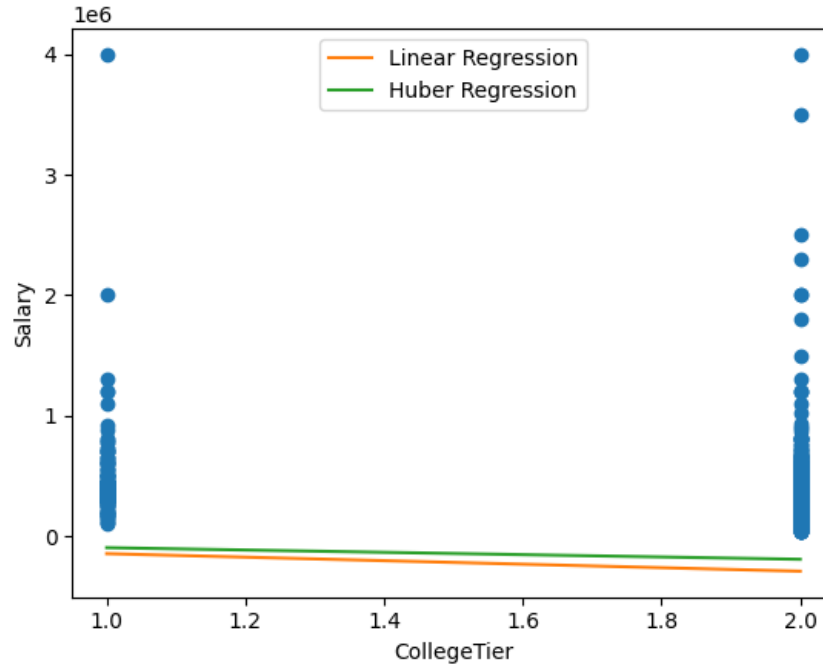
4.5 Model 5: Áp dụng phương pháp chọn lọc thuộc tính

Trong model này, ta chỉ giữ lại những thuộc tính có liên quan tới mức lương thông tính tương quan của các biến. Độ tương quan cho thấy mức độ ảnh hưởng tới nhau của các biến. Độ tương quan càng lớn chứng tỏ các biến càng phụ thuộc vào nhau nhiều hơn. Có nhiều cách để xác định mức độ tương quan như sử dụng tương quan Pearson để đánh giá. Ở đây, ta sử dụng hàm `f_regression` trong thư viện `sklearn` (2.1.1) để đánh giá từng đặc tính so với mức lương. Hàm `f_regression` sử dụng F-statistic và p-values để đánh giá mức độ tương quan và trả về một số điểm cần thiết. Sau đó, ta lấy toàn bộ những đặc tính có số điểm lớn hơn 10 (có thể 20 nếu muốn dùng ít thuộc tính hơn). Từ đó ta trích xuất ra được các thuộc tính như đã trình bày. Và trên thực tế, khi test thì bộ model này cũng tỏ ra rất hiệu quả khi MAE trung bình trong khi tính k-fold Cross-Validation ngang ngửa model 3, là model tốt nhất.



4.6 Model 6: Tối ưu hóa model 5 bằng Huber regression (Huber loss)

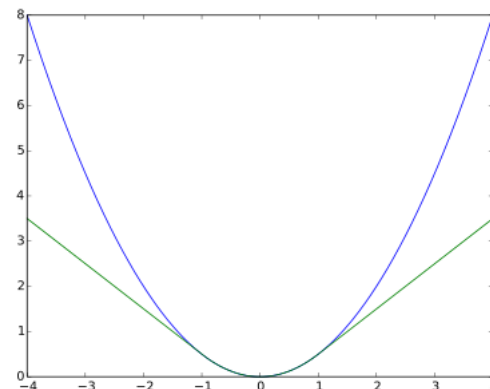
Trước khi đến với Huber Regression, ta cần định nghĩa về outliers. Outliers là những điểm nằm **bất thường** so với các điểm khác. Outliers có thể xảy ra do lỗi trong quá trình đo đạc, mẫu sai lệch hay do những đột biến có trong cộng đồng. Trong khi đó, noisy data là những điểm biến thiên ngoài phân phối chung của dữ liệu. Điểm khác nhau của outliers và noisy data là: noisy data thường không ảnh hưởng quá nhiều đến kết quả còn outliers thì **có ảnh hưởng** đến kết quả. Trong một số trường hợp, đường hồi quy có thể bị lệch do những điểm này. Xem xét tập dữ liệu trên thuộc tính **CollegeTier** sau:



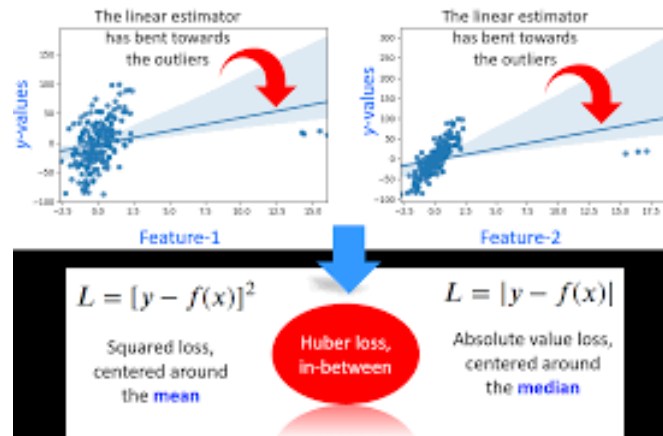
Ta thấy đường màu cam chính là đường hồi quy bình thường của chúng ta còn đường màu xanh là đường hồi quy nếu ta có xem xét các outliers trong tập dữ liệu. Đường màu xanh có xu hướng gần với tập điểm đang xét hơn so với đường màu cam do nó có xem xét một số outliers bất thường như các điểm có mức lương 300000 trở lên. Những điểm này dẫn đến việc đường hồi quy bị bẻ nghiêng đi dẫn tới sai lệch **đáng kể** kết quả cuối cùng. Điều đó dẫn tới sự ra đời của Robust Regression.

Robust Regression là phương pháp được ra đời để cải tiến của các phương pháp Regression thông thường. Một mô hình hồi quy thông thường biểu diễn mối quan hệ của các biến độc lập và biến phụ thuộc. Những kiểu hồi quy thông dụng, ví dụ như **Ordinary Least Square** có những thuộc tính nếu như những gì chúng ta giả định trên tập train là đúng thì sẽ cho ra kết quả tốt, nhưng nếu không thì hoàn toàn ngược lại. Do đó, Robust Regression được tạo ra để hạn chế mức độ ảnh hưởng khi rơi vào trong trường hợp đó. Có rất nhiều kiểu Robust Regression như: Huber Regression, RANSAC Regression, Theil Sen Regression, ... Mỗi phương pháp đều có những cách để xử lý outliers riêng. Ở đây ta sử dụng Huber Regression để đối phó với các điểm outliers này. Huber Regression sử dụng Huber loss thay vì Squared error loss thông thường.

$$L_{\delta}(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta, \\ \delta \cdot (|a| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases}$$

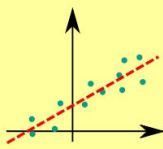
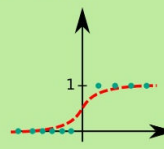
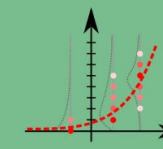


Hàm này đặc biệt ở chỗ, nếu α nhỏ thì sẽ có dạng bậc 2 còn nếu α lớn thì sẽ có dạng tuyến tính. Trong đó, α thường để chỉ phần dư của giá trị quan sát và giá trị dự đoán. Nhờ đó mà Huber loss ít nhạy cảm hơn với các outliers do chúng sẽ tập trung nhiều vào **phần đông** của dữ liệu thay vì xem mọi điểm đều giống nhau như phương pháp Squared error loss thông thường. Cũng nhờ vậy mà MAE của chúng ta cũng đã giảm đáng kể so với ở model 5. Từ *112344.664* giảm xuống còn *107608.961* trong K-fold Cross-Validation.



5 Nhận xét chung

Như đã trình bày ở những phần trước, ta có thể dựa vào đặc thù của dữ liệu hay thực tế xã hội để đoán được mô hình nào cần phải huấn luyện. Việc huấn luyện một hình bằng thuật toán hồi quy tuyến tính về cơ bản là tìm ra hàm tốt nhất sao cho độ lỗi trên dữ liệu càng nhỏ càng tốt. Do tính chất đặc thù của dữ liệu khi vẽ lên, ta thấy được rằng hồi quy tuyến tính với cấp đa thức là 1 trở nên là một trong những mô hình hồi quy tốt nhất có thể sử dụng. Ngoài ra còn nhiều dạng hồi quy khác như: Polynomial Regression cho dữ liệu là đường cong đa thức, Logistic Regression khi dữ liệu phân hóa cao, Ridge Regression, Huber Regression ... Nhìn chung, từ lúc việc nhận dữ liệu sau đó xử lý dữ liệu rồi chọn model, fit data sau đó predict là một quá trình dài để đến được kết quả và cần nhiều kinh nghiệm trong xử lý data và toán học. Do đó AI dần trở thành một trong những ngành khan hiếm lực lượng lao động nhiều nhất.

LINEAR REGRESSION	LOGISTIC REGRESSION	POISSON REGRESSION
<ul style="list-style-type: none"> 1 Econometric modelling 2 Marketing Mix Model 3 Customer Lifetime Value 	<ul style="list-style-type: none"> 1 Customer Choice Model 2 Click-through Rate 3 Conversion Rate 4 Credit Scoring 	<ul style="list-style-type: none"> 1 Number of orders in lifetime 2 Number of visits per user
		
Continuous \Rightarrow Continuous	Continuous \Rightarrow True/False	Continuous \Rightarrow 0,1,2,...
$y = \alpha_0 + \sum_{i=1}^N \alpha_i x_i$	$y = \frac{1}{1 + e^{-z}}$	$y \sim \text{Poisson}(\lambda)$
$\ln(y \sim x1 + x2, \text{data})$	$z = \alpha_0 + \sum_{i=1}^N \alpha_i x_i$	$\ln \lambda = \alpha_0 + \sum_{i=1}^N \alpha_i x_i$
$\text{glm}(y \sim x1 + x2, \text{data})$	$\text{glm}(y \sim x1 + x2, \text{data}, \text{family}=\text{binomial}())$	$\text{glm}(y \sim x1 + x2, \text{data}, \text{family}=\text{poisson}())$
1 unit increase in x increases y by α	1 unit increase in x increases log odds by α	1 unit increase in x multiplies y by e^α

MarketingDistillery.com is a group of practitioners in the area of e-commerce marketing.

Marketing
DISTILLERY

6 Tài liệu tham khảo

- [Wikipedia](#): Xem thông tin về các thuật toán cũng như những thông tin về Ấn Độ
- [machinelearningmastery.com](#): Xem thông tin về các thuật toán Regression và Feature Selection
- [cherylobal.com](#): thông tin về ngành IT ở Ấn Độ
- [Sklearn](#): Xem thông tin về thư viện Sklearn và các thuật toán ML
- [Numpy](#): Xem thông tin về các hàm trong numpy
- [gcu.edu](#): Xem thông tin về độ quan trọng của định lượng trong cuộc sống
- [kaggle.com](#): Xem thông tin về cách implement thuật toán K-Fold Cross-Validation
- [machinelearningcoban.com](#): vấn đề overfitting trong ML
- [thomas.co](#): agreeableness là ảnh hưởng đến công việc
- [machinelearningmastery.com](#): Xem về Robust Regression