Báo cáo đồ án 3

LINEAR REGRESSION

Trần Minh Hải Uyên – 21127202

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

MỤC LỤC

I. Các thư viện sử dụng

- ***Thư viện pandas:*** hàm read_csv dùng để đọc dữ liệu từ 2 file csv, hàm iloc được sử dụng để truy cập dữ liệu bên trong DataFrame bằng cách sử dụng chỉ số nguyên dựa trên vị trí.

```
# Đọc dữ liệu bằng pandas
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')

# Lấy các đặc trưng X và giá trị mục tiêu y cho các tập huấn luyện (train)
X_train = train.iloc[:, :-1]    # Dataframe (chứa 10 đặc trưng huấn luyện)
y_train = train.iloc[:, -1]     # Series    (chứa 1 giá trị mục tiêu kiểm

X_test = test.iloc[:, :-1]      # Dataframe (chứa 10 đặc trưng kiểm tra)
y_test = test.iloc[:, -1]       # Series    (chứa 1 giá trị mục tiêu kiểm
```

- ***Thư viện sklearn:***
  - sklearn.linear_model.LinearRegression: tạo mô hình hồi quy tuyến tính được sử dụng để tìm mối quan hệ tuyến tính giữa các biến độc lập và biến mục tiêu

```
# Huấn luyện mô hình Linear Regression
model = LinearRegression()
model.fit(X_fold_train_selected, y_fold_train)

# Đưa ra dự đoán trên tập validation
y_pred_val = model.predict(X_fold_val_selected)
```

  - sklearn.metrics.mean_absolute_error: tính giá trị trung bình của sai số tuyệt đối giữa các dự đoán của mô hình và giá trị thực tế

```
# Tính toán mean absolute error
fold_mae = mean_absolute_error(y_fold_val, y_pred_val)
total_mae += fold_mae
```

  - sklearn.model_selection.Kfold: kiểm tra mô hình bằng phương pháp K-fold cross-validation, kiểm tra mô hình trên nhiều tập dữ liệu con khác nhau để đánh giá hiệu suất của mô hình một cách chính xác hơn

```
val_nsplit = 5
kf = KFold(n_splits = val_nsplit,shuffle = True,random_state=21127202)
```

- ***Thư viện IPython.display:***
  - display: Hàm này cho phép hiển thị các đối tượng như hình ảnh, đồ thị, bảng dữ liệu và nhiều nội dung đa phương tiện khác trực tiếp trong môi trường IPython.
  - Math: Hàm này cho phép hiển thị biểu thức toán học được viết bằng cú pháp LaTeX, giúp trình bày các công thức toán học phức tạp trong file ipynb.

II. Các hàm cài đặt

1. Hàm formula_to_latex(formula, features)

```python
def formula_to_latex(formula, features):
    latex_string = f'$$\\text{{Salary}} = {formula[0]:.3f}'
    for i, feature in enumerate(features):
        if formula[i+1] >= 0:
            latex_string += f' + {formula[i+1]:.3f} \\times {feature}'
        else:
            latex_string += f' - {-formula[i+1]:.3f} \\times {feature}'
    latex_string += '$$'
    return latex_string
```

- Hàm này trả về một chuỗi LaTeX biểu diễn công thức tính lương dựa trên các giá trị trong formula và features.
- Chuỗi LaTeX được tạo ra bằng cách nối các phần tử của formula và features với nhau theo một cấu trúc nhất định. Cụ thể, chuỗi bắt đầu bằng $$\\text{{Salary}} = {formula[0]:.3f} để hiển thị giá trị đầu tiên của formula với định dạng làm tròn đến 3 chữ số thập phân.
- Sau đó, vòng lặp for được sử dụng để duyệt qua các phần tử của features và thêm chúng vào chuỗi LaTeX. Nếu giá trị tương ứng trong formula là dương, chuỗi sẽ được nối thêm với dấu cộng và giá trị đó nhân với tên của feature. Nếu giá trị đó là âm, chuỗi sẽ được nối thêm với dấu trừ và giá trị tuyệt đối của nó nhân với tên của feature.
- Cuối cùng, chuỗi kết thúc bằng $$ để đóng cặp dấu $ bắt đầu và trả về chuỗi LaTeX đã tạo.

2. *Hàm kfold_best_feature(features,Ssort = 0,Sprint = 1)*

```
val_nsplit = 5
kf = KFold(n_splits = val_nsplit,shuffle = True,random_state=21127202)
model = LinearRegression()
```

- Trước tiên, để thỏa mãn yêu cầu "khi sử dụng cross-validation, cần xáo trộn dữ liệu 1 lần duy nhất và thực hiện trên toàn bộ đặc trưng", ta khai báo hàm một lớp Kfold để chạy cho tất cả các mô hình bên dưới.
  KFold được khởi tạo với n_splits = 5, tức là dữ liệu sẽ được chia thành 5 phần. Các phần dữ liệu này sẽ được trộn trước khi chia nếu shuffle = True. Giá trị random_state=21127202 đảm bảo rằng mỗi lần chạy, kết quả chia dữ liệu sẽ giống nhau

```
def kfold_best_feature(features,Ssort = 0,Sprint = 1):
    best_score = float('inf')
    x_train = X_train[features]
    scores = []
    for feature in features:
        x_train_feature = x_train[[feature]]
        for train_index, test_index in kf.split(x_train_feature):
            X_train_kf, X_test_kf = x_train_feature.iloc[train_index], x_train_feature.iloc[test_index]
            y_train_kf, y_test_kf = y_train.iloc[train_index], y_train.iloc[test_index]
            model.fit(X_train_kf, y_train_kf)
            y_pred = model.predict(X_test_kf)
            score = mean_absolute_error(y_test_kf, y_pred)
            scores.append(score)
    # In ra các kết quả cross-validation như yêu cầu
    mae = []
    for i in range(len(features)):
            mae.append((features[i],round(scores[i],3)))
    if (Ssort == 1):
        mae.sort(key=lambda x: x[1])
    for i in range(len(features)):
        if mae[i][1] < best_score:
            best_score = mae[i][1]
            best_feature = mae[i][0]
    if (Sprint == 1):
        print('STT  ', 'Mô hình đặc trưng        ', 'MAE     ')
        for i in range(len(features)):
            print(i+1,'   ',mae[i][0],(24-len(mae[i][0]))*' ',mae[i][1])
        print('Đặc trưng tốt nhất: ', best_feature)
    return best_feature
```

- ***Hàm kfold_best_feature*** tìm kiếm và trả về đặc trưng tốt nhất dựa trên kỹ thuật cross-validation (KFold) và đánh giá hiệu suất sử dụng Mean Absolute Error (MAE) cho mô hình học máy.
- Các tham số phụ Ssort, Sprint để nhận biết hàm này có cần sắp xếp/ in ra hay không.
- Xây dựng một danh sách trống để lưu các điểm MAE cho từng đặc trưng.

- Lặp qua danh sách các đặc trưng:
- Tạo một DataFrame x_train chỉ chứa đặc trưng hiện tại.
    - Sử dụng KFold để chia dữ liệu thành các fold và lặp qua các fold:
    - Tạo tập huấn luyện và tập kiểm tra từ dữ liệu theo chỉ số của fold.
    - Đào tạo mô hình trên tập huấn luyện và dự đoán trên tập kiểm tra.
    - Tính toán MAE cho dự đoán và lưu vào danh sách MAE.

- Nếu Ssort được đặt là 1, sắp xếp danh sách MAE theo thứ tự tăng dần.
- Lặp qua danh sách MAE để tìm ra đặc trưng có MAE tốt nhất và lưu trữ MAE tốt nhất và đặc trưng tương ứng.
- Nếu Sprint được đặt là 1, in ra bảng kết quả cross-validation và đặc trưng tốt nhất.
- Trả về đặc trưng tốt nhất. (***best_feature***)

III. Nhận xét kết quả

1. Mô hình 1a

- Sử dụng 11 đặc trưng đầu tiên 'Gender', '10percentage', '12percentage', 'CollegeTier', 'Degree', 'collegeGPA', 'CollegeCityTier', 'English', 'Logical', 'Quant', 'Domain'
- **Mức MAE = 105052.53**
- **Công thức mô hình:**

$$\text{Salary} = 49248.090 - 23183.330 \times Gender + 702.767 \times 10percentage + 1259.019 \times 12percentage - 99570.608 \times CollegeTier + 18369.962 \times Degree + 1297.532 \times collegeGPA - 8836.727 \times CollegeCityTier + 141.760 \times English + 145.742 \times Logical + 114.643 \times Quant + 34955.750 \times Domain$$

- **Nhận xét:**
    - Gender: Hệ số -23183.330 cho đặc trưng Gender cho thấy có một sự phụ thuộc âm mạnh giữa giới tính và mức lương. Giá trị hệ số này cho thấy rằng, sau khi điều chỉnh các đặc trưng khác, mức lương của nam giới sẽ thấp hơn mức lương của nữ giới.

    - 10percentage: Hệ số 702.767 cho đặc trưng 10percentage (tổng điểm kỳ thi lớp 10) cho thấy mối quan hệ dương

mạnh giữa điểm số kỳ thi lớp 10 và mức lương. Giá trị hệ số này cho thấy rằng mức lương tăng khi điểm số kỳ thi lớp 10 tăng.

- 12percentage: Hệ số 1259.019 cho đặc trưng 12percentage (tổng điểm kỳ thi lớp 12) cho thấy mối quan hệ dương mạnh giữa điểm số kỳ thi lớp 12 và mức lương. Tương tự như 10percentage, giá trị hệ số này cho thấy rằng mức lương tăng khi điểm số kỳ thi lớp 12 tăng.

- CollegeTier: Hệ số -99570.608 cho đặc trưng CollegeTier cho thấy sự phụ thuộc âm mạnh giữa CollegeTier và mức lương. Điều này có nghĩa rằng các trường đại học có điểm trung bình AMCAT thấp (được chú thích là 2) có thể có mức lương thấp hơn so với các trường có điểm trung bình cao hơn (được chú thích là 1).

- Degree: Hệ số 18369.962 cho đặc trưng Degree cho thấy mối quan hệ dương giữa loại bằng cấp và mức lương. Có thể hiểu là mức lương tăng khi loại bằng cấp tương ứng tăng.

- collegeGPA: Hệ số 1297.532 cho đặc trưng collegeGPA (GPA tại thời điểm tốt nghiệp) cho thấy mối quan hệ dương giữa GPA và mức lương. Mức lương có thể tăng khi GPA tại thời điểm tốt nghiệp tăng.

- CollegeCityTier: Hệ số -8836.727 cho đặc trưng CollegeCityTier cho thấy sự phụ thuộc âm giữa CollegeCityTier và mức lương. Điều này có thể ám chỉ rằng các trường đại học ở các thành phố có hạng cao hơn (có dân số lớn) có thể có mức lương thấp hơn so với các thành phố có hạng thấp hơn.

- English, Logical, Quant: Các hệ số dương (141.760, 145.742, 114.643) cho các đặc trưng kiểm tra khả năng

tiếng Anh, khả năng lô-gic và khả năng định lượng cho thấy mối quan hệ dương với mức lương. Điều này có nghĩa là mức lương có thể tăng khi điểm số trong các phần kiểm tra này tăng.

- Domain: Hệ số 34955.750 cho đặc trưng Domain cho thấy mối quan hệ dương mạnh giữa điểm số Domain (mô-đun chuyên ngành) và mức lương. Mức lương có thể tăng khi điểm số Domain tăng.

2. 5 mô hình 1b

STT	Mô hình đặc trưng	MAE
1	conscientiousness	123510.521
2	agreeableness	124850.912
3	extraversion	116359.735
4	nueroticism	124353.158
5	openess_to_experience	132187.799

Đặc trưng tốt nhất: extraversion

- Trong 5 mô hình trên, ta thấy rằng đặc trưng tốt nhất được chọn là ***extraversion***, với mức MAE thấp nhất.
- **Mức MAE = 116359.735**
- **Công thức mô hình:**

$$\text{Salary} = 306887.222 - 608.965 \times extraversion$$

- **Nhận xét:**
  - Hệ số -608.965 cho đặc trưng 'extraversion' cho thấy mối quan hệ âm mạnh giữa đặc trưng này và mức lương. Giá trị hệ số này cho thấy rằng mức lương có xu hướng giảm khi đặc trưng 'extraversion' tăng. Công thức trên cho thấy đặc trưng này ảnh hưởng tiêu cực tới mức lương

3. 3 mô hình 1c

STT	Mô hình đặc trưng	MAE
1	English	121477.087
2	Logical	120072.695
3	Quant	114351.776

Đặc trưng tốt nhất: Quant

- Trong 2 mô hình trên, ta thấy rằng đặc trưng tốt nhất được chọn là ***Quant***, với mức MAE thấp nhất.
- **Mức MAE = 114351.776**
- **Công thức mô hình:**

$$\text{Salary} = 117759.729 + 368.852 \times Quant$$

- **Nhận xét:**
  - Hệ số 368.852 cho đặc trưng 'Quant' cho thấy mối quan hệ dương giữa điểm số đặc trưng 'Quant' và mức lương. Giá trị hệ số này cho thấy rằng mức lương có xu hướng tăng khi điểm số 'Quant' tăng. Công thức mô hình này cho thấy đặc trưng 'Quant' có ảnh hưởng tích cực đến mức lương.

4. 3 mô hình tự xây dựng

❖ Mô hình 1:

STT	Mô hình đặc trưng	MAE
1	English	111614.184
2	ElectronicsAndSemicon	111980.392
3	openess_to_experience	115880.773
4	12percentage	116430.138
5	Domain	117528.0
6	collegeGPA	117579.273
7	Logical	117794.226
8	CivilEngg	118745.751
9	conscientiousness	120499.178
10	ComputerScience	121085.707
11	CollegeCityTier	121128.738
12	extraversion	122166.771
13	CollegeTier	122214.594
14	Gender	122363.205
15	ComputerProgramming	122508.962
16	TelecomEngg	122544.739
17	ElectricalEngg	123067.867
18	nueroticism	124956.765
19	10percentage	124958.262
20	Quant	126563.519
21	MechanicalEngg	126914.259
22	agreeableness	131757.321
23	Degree	132649.61

Đặc trưng tốt nhất: English

- ***Ý tưởng:***
  - Xét bảng các giá trị MAE của từng mô hình.
  - Từ kết quả trên, chọn ra các đặc trưng có MAE < 120000.
  - Đây là những đặc trưng có ảnh hưởng nhiều nhất tới mức lương.
  - Ta có mô hình 01 gồm các thuộc tính sau: `English`, `ElectronicsAndSemicon`, `openess_to_experience`, `12percentage`, `Domain`, `collegeGPA`, `Logical`, `CivilEngg`.

```
def Module01():
    features = []
    features = ['English', 'ElectronicsAndSemicon', 'openess_to_experience', '12percentage',
                'Domain', 'collegeGPA', 'Logical', 'CivilEngg']
    return features
```

0.0s

- **Mức MAE = 114894.949**

❖ **Mô hình 2:**

- **Ý tưởng:**
  - Dựa trên các điểm số trong bài thi AMCAT, ta chia thành 3 nhóm như sau:
  - Nhóm phần thi khả năng gồm các đặc trưng `English`, `Logical`, `Quant`
  - Nhóm phần thi về lập trình, máy tính và kỹ thuật gồm các đặc trưng `Domain`, `ComputerProgramming`, `ElectronicsAndSemicon`, `ComputerScience`, `MechanicalEngg`, `ElectricalEngg`, `TelecomEngg`, `CivilEngg`
  - Nhóm phần kiểm tra tính cách `conscientiousness`, `agreeableness`, `extraversion`, `nueroticism`, `openess_to_experience`
  - Từ mỗi nhóm, ta lấy một đặc trưng tốt nhất để tạo thành mô hình 02

```
def Module02():
    features = []
    features.append(kfold_best_feature(['English', 'Logical', 'Quant'],0,0))
    features.append(kfold_best_feature(['Domain', 'ComputerProgramming', 'ElectronicsAndSemicon', 'ComputerScience',
                                        'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg', 'CivilEngg'],0,0))
    features.append(kfold_best_feature(['conscientiousness', 'agreeableness', 'extraversion',
                                        'nueroticism', 'openess_to_experience'],0,0))
    return features
```

- **Mức MAE = 117162.714**
- Sau khi chạy, mô hình 2 gồm các đặc trưng sau:

```
Mô hình thứ  2 ['Quant', 'CivilEngg', 'extraversion']
```

❖ **Mô hình 3:**

- **Ý tưởng:**
  - Kết hợp và chia nhóm 23 đặc trưng, ta có được các nhóm đặc trưng sau:
    - Nhóm đặc trưng giới tính gồm đặc trưng `Gender`
    - Nhóm đặc trưng quá trình trước Đại học gồm các đặc trưng `10percentage`, `12percentage`
    - Nhóm đặc trưng quá trình Đại học gồm các đặc trưng `CollegeTier`, `Degree`, `collegeGPA`, `CollegeCityTier`
    - 3 nhóm đặc trưng từ bài thi AMCAT giống với mô hình 02

- Từ mỗi nhóm, ta lấy một đặc trưng tốt nhất để tạo thành mô hình 03

```
def Module03():
    features = []
    features.append('Gender')
    features.append(kfold_best_feature(['10percentage', '12percentage'],0,0))
    features.append(kfold_best_feature(['CollegeTier', 'Degree','collegeGPA', 'CollegeCityTier'],0,0))
    features.append(kfold_best_feature(['English', 'Logical', 'Quant'],0,0))
    features.append(kfold_best_feature(['Domain', 'ComputerProgramming', 'ElectronicsAndSemicon', 'ComputerScience',
                                        'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg', 'CivilEngg'],0,0))
    features.append(kfold_best_feature(['conscientiousness', 'agreeableness', 'extraversion',
                                        'nueroticism', 'openess_to_experience'],0,0))
    return features
```

- **Mức MAE = 113864.054**
- Sau khi chạy, mô hình 3 gồm có các đặc trưng sau:

```
Mô hình thứ  3 ['Gender', '10percentage', 'collegeGPA', 'Quant', 'CivilEngg', 'extraversion']
```

➔ **Đây là mô hình có MAE thấp nhất trong 3 mô hình. Chọn mô hình này làm mô hình hồi quy tốt nhất.**

STT	Mô hình	MAE
1	Mô hình 1	114894.94915798395
2	Mô hình 2	117162.71400368749
3	Mô hình 3	113864.05401289689

- **Công thức hồi quy:**

$$\text{Salary} = -96889.805 - 24645.431 \times Gender + 2360.409 \times 10percentage + 1291.655 \times collegeGPA + 262.038 \times Quant + 57.235 \times CivilEngg + 4530.107 \times extraversion$$

- **Nhận xét:**
  - Gender: Hệ số -24645.431 cho đặc trưng Gender cho thấy có một sự phụ thuộc âm mạnh giữa giới tính và mức lương. Giá trị hệ số này cho thấy rằng, sau khi điều chỉnh các đặc trưng khác, mức lương của nam giới sẽ thấp hơn mức lương của nữ giới.

  - 10percentage: Hệ số 2360.409 cho đặc trưng 10percentage (tổng điểm kỳ thi lớp 10) cho thấy mối quan hệ dương giữa điểm số kỳ thi lớp 10 và mức lương. Giá trị hệ số này cho thấy rằng mức lương có thể tăng khi điểm số kỳ thi lớp 10 tăng.

  - collegeGPA: Hệ số 1291.655 cho đặc trưng collegeGPA (GPA tại thời điểm tốt nghiệp) cho thấy mối quan hệ dương giữa

GPA và mức lương. Mức lương có thể tăng khi GPA tại thời điểm tốt nghiệp tăng.

- Quant: Hệ số 262.038 cho đặc trưng Quant cho thấy mối quan hệ dương giữa điểm số Quant và mức lương. Mức lương có thể tăng khi điểm số Quant tăng.

- CivilEngg: Hệ số 57.235 cho đặc trưng CivilEngg (Kỹ thuật Xây dựng) cho thấy mối quan hệ dương giữa việc học ngành Kỹ thuật Xây dựng và mức lương. Điều này có thể ám chỉ rằng mức lương có thể tăng khi học ngành này.

- extraversion: Hệ số 4530.107 cho đặc trưng extraversion cho thấy mối quan hệ dương giữa đặc trưng tính cách 'extraversion' và mức lương. Điều này cho thấy mức lương có thể tăng khi tính cách extraversion tăng.

➔ **Tổng kết:**
Sau khi huấn luyện mô hình trên toàn bộ dữ liệu, kết quả thu được Mức MAE = 104911.133, là mức MAE thấp nhất trong các mô hình được liệt kê phía trên. Vậy đây là mô hình tốt nhất tìm được.

--- HẾT ---