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

BÁO CÁO ĐỒ ÁN - LINEAR REGRESSION

TOÁN ỨNG DỤNG VÀ THỐNG KÊ

HỌ VÀ TÊN: BÙI ĐỖ DUY QUÂN

MÃ SỐ SINH VIÊN: 21127141

LỚP: 21CLC02

Giảng viên hướng dẫn:
Phan Thị Phương Uyên

Ngày 20 tháng 8 năm 2023

Mục lục

1 YÊU CẦU CỦA ĐỒ ÁN

- Xây dựng mô hình dự đoán **mức lương** của kỹ sư sử dụng **mô hình hồi quy tuyến tính (linear regression)** với các yêu cầu sau:
  1. Sử dụng 11 đặc trưng gồm: **Gender, 10percentage, 12percentage, CollegeTier, Degree, collegeGPA, CollegeCityTier, English, Logical, Quant, Domain**
  2. Sử dụng 5 đặc trưng tính cách: **conscientiousness, agreeableness, extraversion, nueroticism, openess_to_experience** và sử dụng phương pháp **k-fold cross validation** tìm ra đặc trưng tốt nhất trong các đặc trưng tính cách.
  3. Sử dụng 3 đặc trưng kĩ năng: **English, Logical, Quant** và sử dụng phương pháp **k-fold cross validation** tìm ra đặc trưng tốt nhất.
  4. Sinh viên tự xây dựng các mô hình (tối thiểu 3) và tìm mô hình cho kết quả tốt nhất qua phương pháp **k-fold cross validation**.
- Các thư viện được cho trước: **Numpy**, **pandas**.

2 BỘ DỮ LIỆU

- Bộ dữ liệu **Engineering Graduate Salary** gồm 2998 dòng và 34 cột. Sau quá trình tiền xử lý là loại bỏ các cột có **giá trị chuỗi** và **giá trị liên quan đến định danh và năm** thì còn lại 2998 dòng và 24 cột như sau:
  - Giá trị mục tiêu (y): **Salary**
  - 23 đặc trưng giải thích (X) 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.
- Sinh viên đã được cung cấp 2 bộ dữ liệu: **train.csv** và **test.csv**. Bộ dữ liệu **train.csv** gồm 2248 mẫu để huấn luyện mô hình , và bộ dữ liệu **test.csv** gồm 750 mẫu để kiểm tra mô hình.

3 CÁC THƯ VIỆN SỬ DỤNG THÊM

Ngoài việc sử dụng 2 thư viện được cung cấp là **Numpy** và **pandas**, sinh viên còn sử dụng thêm thư viện **sklearn** và sử dụng module **feature_selection** của

thư viện này. Trong module này sẽ sử dụng 3 lớp:

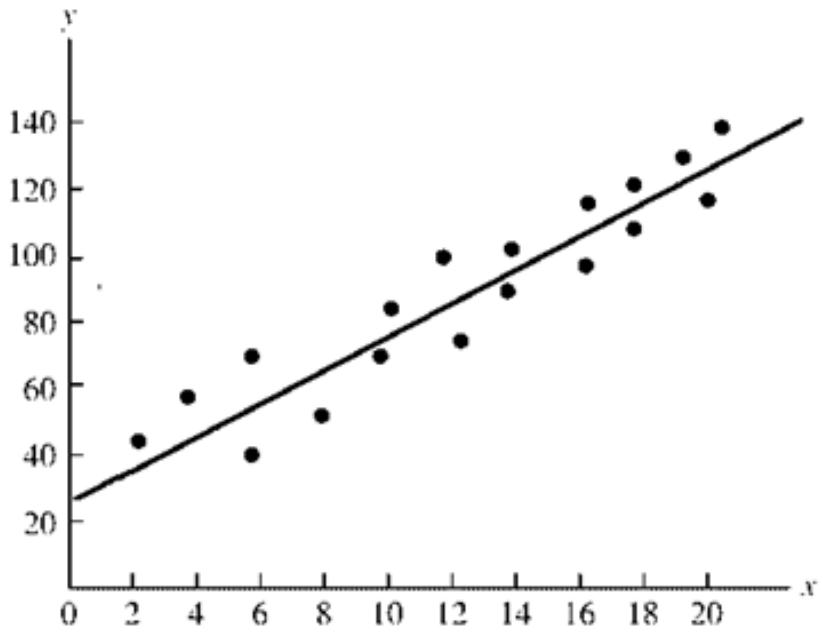
- **VarianceThreshold**: Loại bỏ các đặc trưng có **phương sai** nhỏ hơn ngưỡng được đặt trước.
- **mutual_info_regression**: Tính độ tương quan giữa các đặc trưng và giá trị mục tiêu.
- **SelectPercentile**: Chọn ra nhóm các đặc trưng có độ tương quan cao nhất với giá trị mục tiêu.

4 KIẾN THỨC TÌM HIỂU

4.1 Hồi quy tuyến tính

- **Hồi quy tuyến tính (linear regression)** là phương pháp phân tích quan hệ giữa biến phụ thuộc Y với một hay nhiều biến độc lập X. Phương pháp này sử dụng **hàm tuyến tính (bậc 1)**, và các tham số của mô hình được ước lượng từ dữ liệu. Việc xây dựng **mô hình hồi quy tuyến tính** có thể giúp dự đoán một cách chính xác nhất. Mô hình hồi quy tuyến tính cho mẫu dữ liệu như sau:

$$y = \theta_0 + \theta_1 x \tag{1}$$



Đồ thị minh họa

- Như vậy đối với những dữ liệu có nhiều thuộc tính thì có thể mở rộng mô hình

hồi quy tuyến tính như sau:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n \tag{2}$$

4.2 Huấn luyện và kiểm tra mô hình

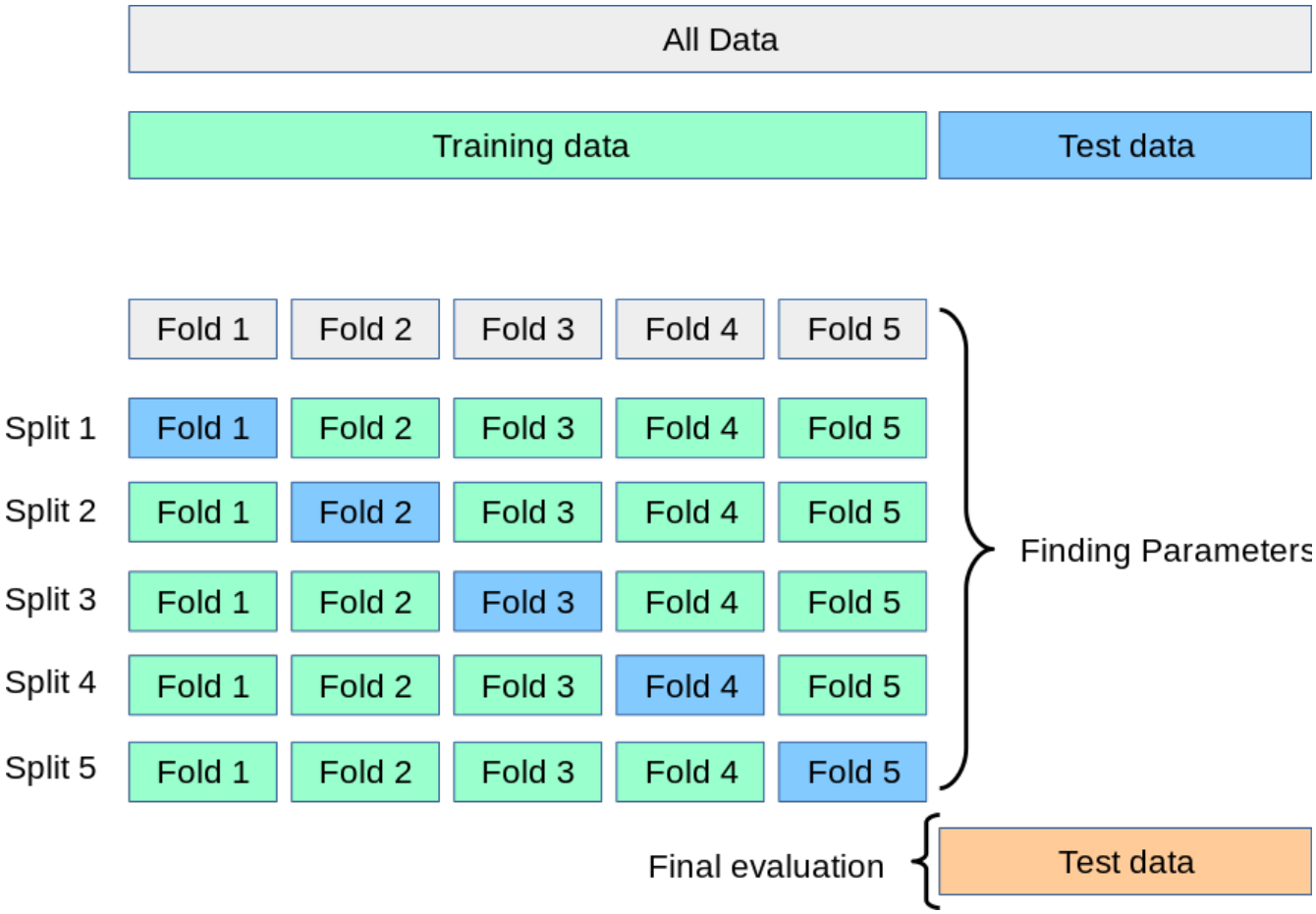
- Để có thể tìm được mô hình phù hợp nhất cho bộ dữ liệu, bộ dữ liệu phải được chia thành 2 phần là **tập huấn luyện** và **tập kiểm tra**. Điều này sẽ giúp mô hình tránh được trường hợp **underfitting** và **overfitting**. Hai trường hợp này lần lượt là những mô hình quá tệ về độ chính xác của dữ liệu dự đoán hay mô hình quá phức tạp, cho kết quả rất tốt trên dữ liệu được cho nhưng lại quá kém so với dữ liệu khác ở thực tế. Tập huấn luyện **trainning set** sẽ được sử dụng để huấn luyện mô hình, còn tập kiểm tra **testing set** sẽ được sử dụng để kiểm tra mô hình đã được huấn luyện có tốt hay không. Tập kiểm tra sẽ được sử dụng để đánh giá mô hình.
- Trong thực tế đã nhiều phương pháp được dùng để tạo ra tập **huấn luyện và kiểm tra**, trong đồ án này sẽ đề cập tới phương pháp **K-fold Cross Validation**.

  K-fold Cross Validation

  – Theo như thông thường, chúng ta sẽ nghĩ tới việc chọn bao nhiêu phần để cho làm phần tập **huấn luyện**, và tập còn lại sẽ là tập **kiểm tra**. Tuy nhiên chúng ta sẽ không biết được 2 phần này nên chứa bao nhiêu dữ liệu trong từng trường hợp và nếu chia sai thì độ chính xác của mô hình sẽ chắc chắn không tốt. Chính vì vậy ý tưởng của phương pháp này chính là sẽ chia đều bộ dữ liệu này thành từng phần **fold** và sẽ thực hiện huấn luyện và kiểm tra từng phần để tìm ra mô hình tốt nhất.

  – Chi tiết các bước làm của phương pháp như sau:

  1. Chia bộ dữ liệu thành **k** phần bằng nhau.
  2. Tại thời điểm xét từng phần dữ liệu, phần dữ liệu đó sẽ được chọn làm tập **kiểm tra**, và **k-1** còn lại sẽ được chọn làm tập **huấn luyện**.
  3. Lưu lại độ chính xác của mô hình tại thời điểm đó.
  4. Lặp lại các bước trên cho đến khi tất cả các phần dữ liệu đều được chọn làm tập **kiểm tra**.
  5. Tính trung bình độ chính xác của các lần huấn luyện và kiểm tra và đây chính là độ chính xác của mô hình.



Hình 1: Minh họa phương pháp K-fold Cross Validation

5 MÔ TẢ CÁC HÀM SỬ DỤNG

5.1 Lớp OLSLinearRegression

Đây là lớp được cô **Phan Thị Phương Uyên** cung cấp, lớp này sẽ giúp sinh viên có thể xây dựng được mô hình hồi quy tuyến tính. Các thuộc tính của lớp này gồm:

- **fit(self, X, y)**: phương thức này sẽ thực hiện huấn luyện mô hình hồi quy tuyến tính với dữ liệu đầu vào là **X (dữ liệu đặc trưng)** và **y (dữ liệu mục tiêu)**. Hàm sẽ thực hiện và trả về trọng số của mô hình tương ứng với các đặc trưng theo công thức:

$$\theta = (X^T X)^{-1} X^T y \tag{3}$$

- **get_params()**: phương thức *getter* sẽ trả về các trọng số của mô hình.
- **predict(self, X)**: phương thức này sẽ thực hiện dự đoán giá trị mục tiêu dựa trên dữ liệu đầu vào là **X (dữ liệu đặc trưng)** và trọng số của mô hình.

Hàm sẽ thực hiện và trả về giá trị dự đoán theo công thức:

$$y = \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n \tag{4}$$

5.2 Hàm mae(y_pred, y_test)

Đây là hàm được cô **Phan Thị Phương Uyên** cung cấp, tham số truyền vào sẽ là **giá trị/tập giá trị dự đoán và giá trị/tập giá trị kiểm tra**. Hàm sẽ **trả về giá trị thể hiện độ chính xác** của giá trị được dự đoán, giá trị càng thấp thì độ chính xác càng cao. Công thức để tính độ chính xác là:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_{pred} - y_{test}| \tag{5}$$

5.3 Hàm getTrain(index, folks)

Đây là hàm hỗ trợ cho việc xác định tập dữ liệu **huấn luyện** trong phương pháp K-fold Cross Validation. Giá trị đầu vào sẽ là **index** của phần dữ liệu đang xét và **folks** là tập dữ liệu đã được chia. Hàm sẽ **trả về tập dữ liệu huấn luyện**.

5.4 Hàm Best_Feature_Personality(Df_np, k_cluster)

Đây là hàm giúp tìm ra đặc trưng tính cách tốt nhất trong 5 tính cách ở yêu cầu **2** trong phần YÊU CẦU ĐỒ ÁN. Giá trị đầu vào sẽ là **Df_np** là tập dữ liệu đã được chuyển thành dạng **numpy array** và **k_cluster** là số lượng cụm cần phải chia. Hàm sẽ thực hiện huấn luyện từng đặc trưng tính cách trên từng phần và sau cùng **trả về số thứ tự của đặc trưng tính cách tốt nhất**.Việc chọn ra đặc trưng tốt nhất sẽ được thực hiện dựa trên so sánh **MAE trung bình** của từng đặc trưng sau khi huấn luyện trên từng **k_cluster** phần dữ liệu.

5.5 Hàm best_personality_feature_model()

Đây là hàm sẽ thực hiện huấn luyện mô hình theo đặc trưng tốt nhất mà được tìm thấy ở hàm **Best_Feature_Personality**. Tham số truyền vào sẽ không có, nhưng giá trị trả về của hàm lần lượt 2 giá trị/tập giá trị **dự đoán** và **kiểm tra**.

5.6 Hàm Best_Feature_Skill(Df_np, k_cluster)

Đây là hàm giúp tìm ra đặc trưng kĩ năng tốt nhất trong 3 kĩ năng ở yêu cầu **3** trong phần YÊU CẦU ĐỒ ÁN. Giá trị đầu vào sẽ là **Df_np** là tập dữ liệu đã được chuyển thành dạng **numpy array** và **k_cluster** là số lượng cụm cần phải chia. Hàm sẽ thực hiện huấn luyện từng đặc trưng kĩ năng trên từng phần và sau

cùng **trả về số thứ tự của đặc trưng kĩ nang9 tốt nhất**.Việc chọn ra đặc trưng tốt nhất sẽ được thực hiện dựa trên so sánh **MAE trung bình** của từng đặc trưng sau khi huấn luyện trên từng **k_cluster** phần dữ liệu.

5.7 Hàm best_skill_feature_model()

Đây là hàm sẽ thực hiện huấn luyện mô hình theo đặc trưng tốt nhất mà được tìm thấy ở hàm **Best_Feature_Skill**. Tham số truyền vào sẽ không có, nhưng giá trị trả về của hàm lần lượt 2 giá trị/tập giá trị **dự đoán** và **kiểm tra**.

5.8 Hàm cross_validation_model(Df_np, k_cluster)

Đây là hàm sẽ thực huấn luyện để tìm ra độ chính xác của mô hình nhờ phương pháp **K-fold Cross Validation**. Về ý tưởng, cách thức cài đặt sẽ khá giống với hàm **Best_Feature_Personality** và **Best_Feature_Skill**, nhưng hàm này sẽ thực hiện huấn luyện trên tất cả các đặc trưng trong mảng **Df_np** truyền đầu vào cùng với số nhóm muốn dữ liệu được chia và **trả về độ chính xác của mô hình**.

5.9 Hàm model_variance_Dropping(X_train)

Đây là hàm sẽ hỗ trợ cho phương pháp xây dưng mô hình trong yêu cầu **3** trong phần YÊU CẦU ĐỒ ÁN, với tên gọi: Dropping Constant Feature, dựa trên giá trị **phương sai** của các đặc trưng. Tham số truyền vào sẽ là **X_train** là tập dữ liệu huấn luyện của các đặc trưng không phụ thuộc. Hàm sẽ thực hiện **tìm ra các đặc trưng có phương sai lớn hơn ngưỡng** và **trả về tên các đặc trưng đó**.

5.10 Hàm model_correlation_Dropping(X_train, threshold)

Đây là hàm sẽ hỗ trợ cho phương pháp xây dưng mô hình trong yêu cầu **4** trong phần YÊU CẦU ĐỒ ÁN,với tên gọi: Dropping high-correlation Feature, dựa trên giá trị **phương sai** của các đặc trưng. Tham số truyền vào sẽ là **X_train** là tập dữ liệu huấn luyện của các đặc trưng không phụ thuộc và mức độ tương quan mong muốn giữa các đặc trưng mong. Hàm sẽ thực hiện **chọn ra các đặc trưng có mức độ tương quan lớn hơn threshold** và **trả về tên các đặc trưng đó**.

5.11 Hàm model_MutualInfor_selection(X_train, Y_train)

Đây là hàm sẽ hỗ trợ cho phương pháp xây dưng mô hình trong yêu cầu **4** trong phần YÊU CẦU ĐỒ ÁN,với tên gọi: Selecting high-MI Feature, dựa trên giá trị **mutual information** của các đặc trưng. Tham số truyền vào sẽ là **X_train** là

tập dữ liệu huấn luyện của các đặc trưng không phụ thuộc và **Y_train** là tập dữ liệu của đặc trưng phụ thuộc. Hàm sẽ thực hiện **chọn ra các đặc trưng có mức độ liên quan tới đặc trưng phụ thuộc cao** và **trả về tên các đặc trưng đó**.

5.12 Hàm my_best_model(models)

Đây là hàm sẽ tìm model nào tốt nhất từ mảng các **models** truyền vào qua phương pháp **K-fold Cross Validation**. Hàm sẽ **trả về số thứ tự của model tốt nhất trong mảng**.

5.13 Hàm train_my_best_model()

Đây là hàm huấn luyện model tốt nhất mà đã tìm được. **Đầu vào** của hàm sẽ không có tham số, nhưng hàm sẽ **trả về giá trị/tập giá trị của dữ liệu phụ thuộc được dự đoán và kiểm tra**.

6 KẾT QUẢ CỦA CÁC YÊU CẦU

6.1 Yêu cầu 1

a Các bước thực hiện

1. Truy xuất dữ liệu của 11 cột đặc trưng được yêu cầu (11 cột đầu tiên) trong tập huấn luyện và tập kiểm tra.
2. Truy xuất cột 'Salary' - là cột mục tiêu trong tập huấn luyện và tập kiểm tra.
3. Tìm trọng số của mô hình hồi quy tuyến tính bằng cách sử dụng phương thức **fit** của lớp **OLSLinearRegression** với dữ liệu đầu vào là tập huấn luyện của 11 đặc trưng yêu cầu và đặc trưng mục tiêu 'Salary'.
4. Sử dụng phương thức **predict** của lớp **OLSLinearRegression** để dự đoán giá trị mục tiêu của tập kiểm tra.
5. Sử dụng hàm **mae** để tính độ chính xác của mô hình.

b Công thức cho mô hình hồi quy (trọng số làm tròn tới 3 chữ số thập phân) và kết quả trên tập kiểm tra

- $$\begin{aligned}\text{Salary} = &(-22756.513) \cdot X_1 + 804.503 \cdot X_2 + 1294.655 \cdot X_3 \\ &+ (-91781.898) \cdot X_4 + 23182.389 \cdot X_5 + 1437.549 \cdot X_6 \\ &+ (-8570.662) \cdot X_7 + 147.858 \cdot X_8 + 152.888 \cdot X_9 \\ &+ 117.222 \cdot X_{10} + 34552.286 \cdot X_{11}\end{aligned} \tag{6}$$

- Độ chính xác của mô hình trên tập kiểm tra là: **104863.77754033315**

6.2 Yêu cầu 2

a Các bước thực hiện

Cài đặt **best_personality_feature_model** sẽ dùng để thực hiện yêu cầu 2 như sau:

1. Khởi tạo mảng chứa tên 5 đặc trưng tính cách vì trong bộ dữ liệu, 5 đặc trưng này đứng liên tiếp nhau.
2. Thực hiện đổi chỗ ngẫu nhiên của cac dòng dữ liệu 1 lần.
3. Gọi hàm **Best_Feature_Personality** để tìm ra đặc trưng tốt nhất theo phương pháp K-fold Cross Validation.
   (a) Thực hiện chia bộ dữ liệu huấn luyện thành 10 nhóm đều nhau (sử dụng hàm **numpy.array_split**).
   (b) Thực hiện huấn luyện trên từng phần:
   - Gọi hàm **getTrain** để lấy tập dữ liệu huấn luyện và tập kiểm tra là phần dữ liệu đang xét.
   - Với mỗi đặc trưng tính cách, thực hiện huấn luyện mô hình và tính độ chính xác của mô hình, lưu lại độ chính xác của mô hình trong mảng **mae_arr**.

   (c) Sau khi xét hết phần dữ liệu, mảng **mae_arr** thu được gồm 10 dòng và 5 cột tương đương với 10 phần dữ liệu được chia và 5 đặc trưng của mỗi dòng.
   (d) Tính độ chính xác trung bình của mỗi đặc trưng bằng cách lấy trung bình cộng của mỗi cột trong mảng **mae_arr** và trả ra giá trị thứ tự của đặc trưng có **mae** nhỏ nhất.
4. Sau khi tìm được đặc trưng tốt nhất, tiến hình truy xuất dữ liệu của đặc trưng đó để huấn luyện mô hình bằng cách sử dụng hàm **fit** của lớp **OLSLinearRegression** và tính độ chính xác của mô hình bằng cách sử dụng hàm **mae**.

b Kết quả tương ứng cho 5 mô hình từ k-fold Cross Validation

STT	Mô hình với 1 đặc trưng	MAE
1	conscientiousness	306267.12993085
2	agreeableness	300894.66699364
3	extraversion	307070.11566216
4	neuroticism	299387.84192331
5	openness_to_experience	303008.96858233

c Kết quả tương ứng cho mô hình tốt nhất

- Mô hình tốt nhất là mô hình có đặc trưng **neuroticism** với độ chính xác trung bình là: **299387.84192331**
- Công thức hồi quy tuyến tính của mô hình tốt nhất:

$$\text{Salary} = (-56546.304) * neuroticism \tag{7}$$

- Kết quả **mae** là: 291019.693226953

d Nhận xét

- Kết quả của **k-fold Cross Validation** đã cho độ chính xác của đặc trưng **neuroticism** là tốt nhất,

7 TÀI LIỆU THAM KHẢO

- Cô Phan Thị Phương Uyên.
- Công thức thay đổi độ sáng và độ tương phản.
- Thay đổi độ tương phản.
- Công thức chuyển thành hình xám.
- Công thức chuyển thànhh hình màu sepia.
- Công thức và ma trận các kernel để làm mờ và rõ nét ảnh.
- Triển khai thuật toán cho làm mờ ảnh.
- Phương trình ellip.
- Công thức của ellip khi xoay 45 độ.

- Tài liệu các hàm trong thư viện numpy.
- Thư viện matplotlib hỗ trợ xuất ảnh và lưu ảnh.
- Thư viện Pillow hỗ trợ đọc ảnh.