

**TRƯ**

**Ờ**

**NG Đ**

**Ạ**

**I H**

**Ọ**

**C KHOA H**

**Ọ**

**C T**

**Ự**

**NHIÊN ĐHQG**

**-**

**HCM**

**\***

**-------**

**\***

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN 3 – Linear regression**

**Môn Học: Toán ứng dụng & thống kê**

**Giảng viên:**

Nguyễn Trọng Hiến

Nguyễn Văn Quang Huy

**Sinh viên thực hiện:**

Nguyễn Tấn Phát 20127588



Nguyễn Đình Thúc

Võ Nam Thục Đoan

## Nội dung đồ án:

* + File "wine.csv" là cơ sở dữ liệu đánh giá chất lượng của 1200 chai rượu vang theo thang điểm 1\_10 dựa trên 11 tính chất khác nhau.

Ảnh có chứa bàn

Mô tả được tạo tự động

* + Xây dựng mô hình đánh giá chất lượng rượu sử dụng phương pháp hồi quy tuyến tính.
  1. Sử dụng toàn bộ 11 đặc trưng đề bài cung cấp,
     + y=theta\_1x\_1+theta\_2x\_2+...+theta\_11x\_11
  2. Sử dụng duy nhất 1 đặc trưng cho kết quả tốt nhất. (Gợi ý: Phương pháp Cross Validation)
     + y=theta\_ix\_i (dùng mô hình lần lượt cho từng đặc trưng).
  3. Xây dựng một mô hình của riêng bạn cho kết quả tốt nhất.

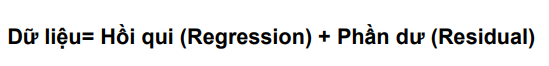
## Môi trường làm việc:

* Ngôn ngữ lập trình: Python
* Text Editer: Visual Studio Code
* Thư viện hỗ trợ: numpy, pandas, matplotlib

## Ý tưởng và các hàm:

### Ý tưởng giải quyết bài toán

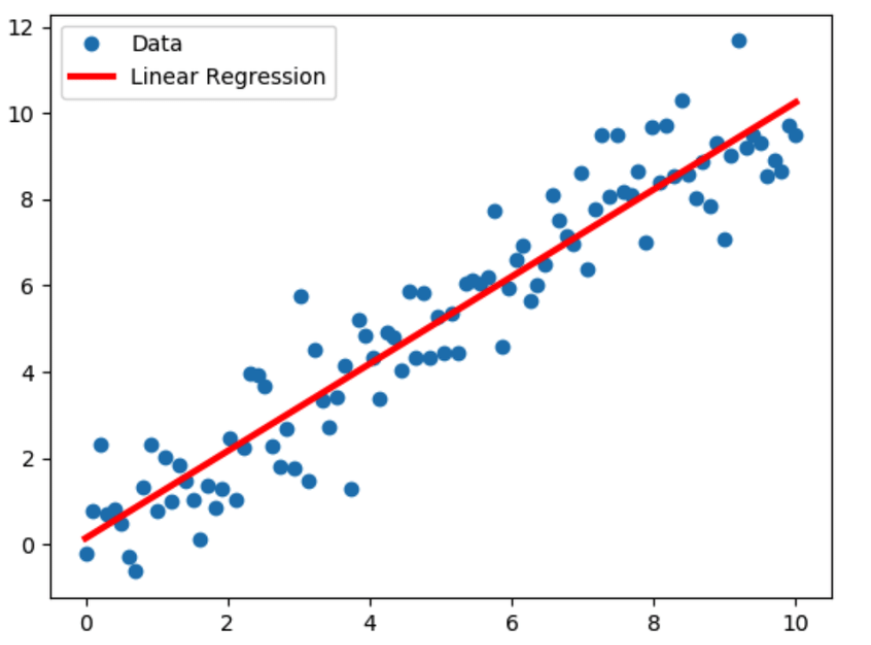
* Xác định mô hình cần phải đưa về:
  + Vd: y = ax + b
  + Hay y = [a, b].T \* [x, 1]
* Phân tích dữ liệu đầu vào thành 2 phần: hồi quy và phần dư:



* Khi đó
  + [a, b] chính là nghiệm của Linear Regresstion cần phải tìm.
  + [x, 1] chính là ma trận hồi quy
  + [y] chính là phần dư
* Đặt A = [x, 1] khi đó:
  + [a, b] =

### Các hàm:

* createMatrix(): tạo ma trận A phía trên
* resultMatrix(): tạo ma trận [y]



## Các hàm hỗ trợ:

### Numpy.linalg.inv()

* *Chức năng: Tính ma trận khả nghịch*
* VD:
  + Input: [A]
  + Output: [A]^-1

### Numpy.transpose()

* *Chức năng: Trả về ma trận nghịch đảo*
* VD:
  + Input: [A]
  + Output: [A].T

### Numpy.dot

* Chức năng: Nhân 2 ma trận với nhau
* VD:
  + Input: [A] [B]
  + Output: [A.B]

## Hồi quy tuyến tính:

* LinearRegressstion(matrix, result): Truyền vào matrix và resultMatrix
  + Áp dụng công thức: [a, b] =
  + Trả về [a, b] trong y = ax + b
  + Hoặc [a, b, c] trong y = ax1 + bx2 + c
* leastSquares(data, title, titleResult): Truyền vào dữ liệu, tên tiêu chí cần tính bình phương tối tiểu và tên tiêu chí kết quả
  + Được tính qua công thức L =
  + Với y là kết quả thực tế.
  + Và f(xi) là hàm hồi quy tuyến tính
* drawLinearRegression: Dùng để vẽ hàm hồi quy tuyến tính để có cái nhìn trực quan hơn.

## Kết quả chạy thử:



Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình

Mô tả được tạo tự động

## Tài liệu tham khảo:

1. Slide bài giảng môn “Toán ứng dụng & thống kê”
2. Slide bài giảng môn “Đại số tuyến tính”

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HẾT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*