

**TRƯỜNG ĐẠI HỌC HOA SEN**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙠🕮🙢

**BÁO CÁO SEMINAR**

ĐỀ TÀI:

# XÂY DỰNG ỨNG DỤNG DỰ BÁO TRỌNG LƯỢNG CỦA CÁ ÁP DUNG MÔ HÌNH HỒI QUY TUYẾN TÍNH

# XÂY DỰNG ỨNG DỤNG DỰ BÁO ÁP DUNG MÔ HÌNH HỒI QUY TUYẾN TÍNH

# XÂY DỰNG ỨNG DỤNG DỰ BÁO ÁP DUNG MÔ HÌNH HỒI QUY TUYẾN TÍNH

# XÂY DỰNG ỨNG DỤNG DỰ BÁO ÁP DUNG MÔ HÌNH HỒI QUY TUYẾN TÍNH

GV: Nguyễn Thị Thanh

SV Thực hiện: Nguyễn Phương Khoa

MSSV: 2181195

Lớp: TIN340DV01 – 0100-0101

Môn học: Machine Learning

HO CHI MINH CITY, 2021

**MỤC LỤC**

[DANH MỤC HÌNH 2](#_Toc88590110)

[MỞ ĐẦU 3](#_Toc88590111)

[1. Bài Toán 4](#_Toc88590112)

[2. Mô Hình 4](#_Toc88590113)

[3. Lập Trình Mô Hinh 5](#_Toc88590114)

[3.1 Nhập thư viện vào chương trình 5](#_Toc88590115)

[3.2 Import dữ liệu 5](#_Toc88590116)

[3.3 Kiểm tra dữ liệu 6](#_Toc88590117)

[3.4. Tìm hiểu về dữ liệu 8](#_Toc88590118)

[3.5 Xây dựng mô hình 11](#_Toc88590119)

[3.6 Biễu diễn lên đồ thị 13](#_Toc88590120)

[*3.6.1 Đồ thị biểu diễn giá trị thực tế và giá trị dự doán của Y\_test và Y\_predict\_test.* 13](#_Toc88590121)

[*3.6.2 Đồ thị biểu diễn giá trị thực tế và giá trị dự doán của Y\_train và Y\_predict\_train.* 13](#_Toc88590122)

[3.7 Đánh giá mô hình 14](#_Toc88590123)

[TÀI LIỆU THAM KHẢO 18](#_Toc88590124)

# DANH MỤC HÌNH

[Hình 1:Hình mô tả 1 5](#_Toc88588213)

[Hình 2: Hình mô tả 2 6](#_Toc88588214)

[Hình 3: Hình mô tả 3 6](#_Toc88588215)

[Hình 4: Hình mô tả 4 7](#_Toc88588216)

[Hình 5: Hình mô tả 5 7](file:///C:\Users\admin\Desktop\Báo%20Cáo%20&%20TT%20%20(%20Máy%20hoc)\BÁO%20CÁO%20(MÁY%20HỌC).docx#_Toc88588217)

[Hình 6: Hình mô tả 6 8](#_Toc88588218)

[Hình 7: Hình mô tả 7 9](#_Toc88588219)

[Hình 8: Hình mô tả 8 9](#_Toc88588220)

[Hình 9: Hình mô tả 9 10](#_Toc88588221)

[Hình 10: Hình mô tả 10 10](#_Toc88588222)

[Hình 11: Hình mô tả 11 11](#_Toc88588223)

[Hình 12: Hình mô tả 12 13](#_Toc88588224)

[Hình 13: Hình mô tả 13 14](#_Toc88588225)

[Hình 14: Hình mô tả 14 15](#_Toc88588226)

[Hình 15: Hình mô tả 15 16](#_Toc88588227)

[Hình 16: Hình mô tả 16 17](#_Toc88588228)

# MỞ ĐẦU

Chúng ta đang sống trong thời đại của lượng lớn dữ liệu, máy tính mạnh mẽ và trí tuệ nhân tạo. Điều này chỉ là khởi đầu. Khoa học dữ liệu và máy học đang thúc đẩy nhận dạng hình ảnh, phát triển phương tiện tự hành, các quyết định trong lĩnh vực tài chính và năng lượng, tiến bộ trong y học, sự nổi lên của mạng xã hội, v.v. Hồi quy tuyến tính là một phần quan trọng của điều này.Hồi quy tuyến tính là một trong những kỹ thuật thống kê và học máy cơ bản. Cho dù bạn muốn thực hiện thống kê, máy học hay máy tính khoa học, rất có thể bạn sẽ cần nó. Sự kết hợp của một lượng lớn dữ liệu và máy tính mạnh mẽ có thể tính toán nhiều con số hơn trong vài phút so với một người có thể trong suốt cuộc đời của họ, có nghĩa là chúng ta có thể tạo ra các mô hình chính xác và năng động hơn bao giờ hết. Chẳng bao lâu nữa, lĩnh vực học máy sẽ phát triển khi các cá nhân nhận ra rằng họ có thể tạo ra các chương trình máy tính có thể học từ tất cả dữ liệu này. Các mô hình mới được phát triển để xử lý các loại dữ liệu và vấn đề khác nhau, đồng thời có thể tham khảo một kho các kỹ thuật mới để đưa ra các giải pháp mới cho các vấn đề cũ.

# 1. Bài Toán

Chúng ta sẽ tạo ra một mô hình để dự đoán trọng lượng (weight) của 7 loại cá khác nhau được bán phổ biến trong chợ (Bream, Parkki, Perch, Pike, Roach, Smelt, Whitefish) dựa vào các thông số (vertical length, diagnol length, cross length, height, width). Công việc chúng ta làm là liệu biến giải thích có thực hiện tốt công viêc dự đoán của biến phụ thuộc không.

**Mô tả dữ liệu:** Bộ dữ liệu này là một bản ghi về 7 loại cá khác nhau được bán phổ biến trong doanh số bán cá ở chợ. Với bộ dữ liệu này, chúng ta sẽ áp dụng một mô hình dự đoán để có thể ước tính được trọng lượng của cá.

**Tập dữ liệu gồm các thông tin sau:**

* Loại cá: gồm 7 loại cá
* Weight: cân nặng.
* Vertical\_length: chiều dài dọc
* Cross\_length: chiều dài chéo
* Diagnol\_length: diagnol length
* Height: chiều cao
* Width: chiều rộng
* Tập dữ liệu này có 100 hàng và 7 cột.
* Biến phụ thuộc là: weight
* Biến độc lập có 5 biến: vertical length, diagnol length, cross length, height, width.

\*Dữ liệu được đăng bởi Aung Pyae (link: ***https://www.kaggle.com/aungpyaeap/fish-market***)

***Tại sao chọn mô hình hồi quy tuyến tính?***

Lý do chính khiến mô hình hồi quy tuyến tính được chọn cho bài toán này là:

Chúng ta muốn biết liệu một biến đo lường có được liên kết với một biến đo lường khác hay không ?

Chúng ta muốn một phương trình mô tả mối quan hệ và có thể được sử dụng để dự đoán các giá trị chưa biết ?

# 2. Mô Hình

Mô hình tuyến tính sẽ có dạng sau:



Trong đó mỗi thuật ngữ đại diện cho:

* **Y: là biến phụ thuộc**
* **X: là biến độc lập**
* **B0: hằng số**
* **B1: mối quan hệ giữa X và Y**

# 3. Lập Trình Mô Hinh

## *3.1 Nhập thư viện vào chương trình*

Để xây dựng mô hình, chúng ta sẽ cần một số thư viện sau:

* Pandas – Hỗ trợ cho việc lấy và chuyển đổi dữ liệu.
* Matplotlib – Thư viện hỗ trợ việc vẽ đồ thị.
* Numpy (Numeric Python): là một thư viện toán học phổ biến và mạnh mẽ của Python.
* Statsmodels.api - Tạo mô hình thống kê.
* model\_selection.train\_test\_split - chia mảng thành tập hợp con để thử nghiệm và huấn luyện ngẫu nhiên.
* sklearn.linear\_model : Để sử dụng thuật toán Linear Regression đã được cài đặt sẵn trong Scikit-learn, chúng ta thực hiện import
* Sklearn - Chúng ta có thể tận dụng các mô hình học máy tích hợp mà thư viện này có.
* Math: thư viện toán học
* Seaborn: là một thư viện trực quan hóa dữ liệu, Seaborn cho phép tạo đồ họa thống kê.

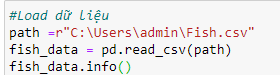
## *3.2 Import dữ liệu*

Khi làm việc với Machine Learning chúng ta phải tiếp xúc với lượng Data lớn, để có thể thực hiện dễ dàng, ta có thể sử dụng các file để lưu trữ data cho Machine Learning.

Ở đây ta sử dụng thư viện **pandas**, thư viện này hỗ trợ trong thao tác dữ liệu.Thư viện này sử dụng một cấu trúc dữ liệu riêng là **Dataframe**. **DataFrame** đem lại sự linh hoạt và hiệu quả trong thao tác dữ liệu và lập chỉ mục.

Dữ liệu ở đây ta có một file csv, thì để đọc tệp csv bằng Python Pandas dưới dạng DataFrame, ta sử dụng phương pháp ***read\_csv().*** Pandas chuyển đổi cấu trúc này thành cấu trúc DataFrame, một cấu trúc giống như bảng.

Chúng ta muốn dữ liệu của mình trong một DataFrame, điều này sẽ cung cấp cho nó cấu trúc thích hợp để phân tích dữ liệu.

****

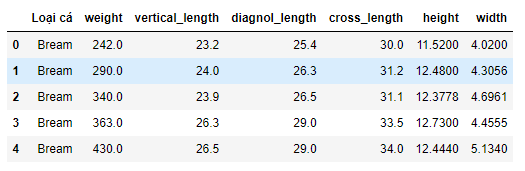
Hình :Hình mô tả 1

Để có thể dễ dàng hiểu rõ hơn về dữ liệu, tốt nhất chúng ta nên đặt lại tên cho nó.

fish\_data.columns = ['species','weight','vertical\_length','diagnol\_length','cross\_length','height','width']

Cuối cùng, tốt nhất là chúng ta nên kiểm tra dữ liệu có chính xác không trước khi tiếp tục, vì vậy chúng ta hãy in ra bản ghi đầu tiên của dataframe sử dụng hàm **head().**

price\_data.head()

****

Hình : Hình mô tả 2

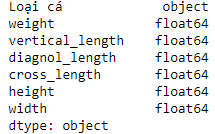
## *3.3 Kiểm tra dữ liệu*

Để có được một tập dữ liệu rõ ràng hoàn hảo đáp ứng tất cả các yêu cầu là không có dữ liệu bị thiếu, điều này làm cho ứng dụng này thực tế hơn, chúng ta sẽ làm sạch dữ liệu. Đây là danh sách kiểm tra khi nói đến việc làm sạch dữ liệu:

1. Kiểm tra các loại dữ liệu để đảm bảo rằng nó thật sự chính xác. Ví dụ: nó thường không có ý nghĩa đối với một số là một chuỗi.
2. Đảm bảo rằng tên các cột là chính xác. Có tên cột chính xác giúp quá trình chọn dữ liệu dễ dàng hơn.
3. Kiểm tra và bỏ / điền các giá trị bị thiếu. Bỏ lỗi giúp kiểm soát lỗi khi chạy.

* **Kiểm tra kiểu dữ liệu.**

Để kiểm tra xem kiểm dữ liệu của tập tin chúng ta sử dụng phương thức **.dtypes.** Phương thức này sẽ trả về kiểu dữ liệu của từng đối tượng trong Dataframe của chúng ta.



Hình : Hình mô tả 3

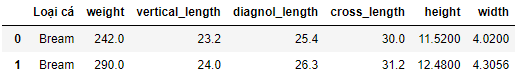
Trong khi xem xét dữ liệu, nếu chúng ta muốn đổi các tên cột trong dữ liệu, chúng ta có thể thay đổi tên bằng cách tạo một đối tượng trong đó tên cũ là khóa và tên mới là giá trị cho khóa đó.

new\_column\_names = {'species':'Loại cá'}

fish\_data = fish\_data.rename(columns = new\_column\_names)

fish\_data.head(100)

Sau khi đặt tên chúng ta có thể gọi phương thức **rename ()** trên **DataFrame** và chuyển qua từ điển **new\_column\_names** thông qua tham số cột.



Hình : Hình mô tả 4

Thiếu giá trị, là một vấn đề có thể tạo ra lỗi khi chạy chương trình. Trước tiên chúng ta cần phải luôn kiểm tra xem có bất kỳ giá trị nào bị thiếu không.

Chúng ta sử dụng phương thức **.isna (). Any ()** trên **DataFrame**, nó sẽ trả về mỗi cột với một boolean,

# kiểm tra giá trị bị thiếu

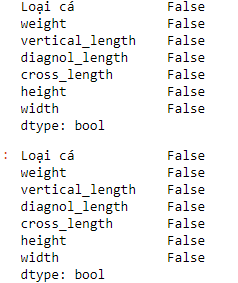
display(fish\_data.isna().any())

# xóa giá trị bị thiếu

fish\_data = fish\_data.dropna()

# Kiểm tra lại dữ liệu

fish\_data.isna().any()



Trong đó:

* **True** có nghĩa là nó bị thiếu giá trị
* **False** có nghĩa là nó không có bất kỳ giá trị nào bị thiếu.

Sau khi biết dữ liệu có giá trị bị thiếu, chúng ta có thể sử dụng phương thức **dropna ()** để loại bỏ bất kỳ hàng nào có giá trị bị thiếu.

Hình : Hình mô tả 5

## *3.4. Tìm hiểu về dữ liệu*

Bây giờ chúng ta đã có một tập dữ liệu hoàn chỉnh. Một trong những bước quan trọng để ta hiểu rõ thêm về dữ liệu như sau:

* Dữ liệu được phân phối như thế nào?
* Có xuất hiện mối quan hệ giữa hai biến không?
* Có bất kỳ ngoại lệ nào không?
* Dữ liệu có bị lệch không?

Đầu tiên ta chia tập dữ liệu X, Y của chúng ta ra. Trong đó X là các dữ liệu về các chỉ số của các loại cá gồm: vertical length, diagnol length, cross length, height, width và Y là weight.

X = fish\_data.drop(['weight','Loại cá'], axis=1).values

Y = fish\_data['weight'].values

***Đo lường mối tương quan.***

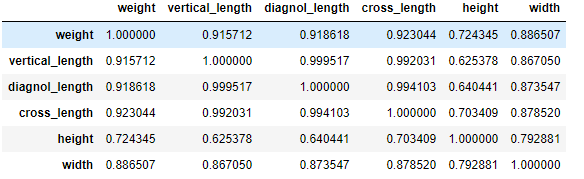
Trong trường hợp này, chúng ta sẽ đo lường mối tương quan giữa hai biến.

Sử dụng DataFrame và gọi phương thức **corr ()** để trả về một DataFrame với các chỉ số.

Sử dụng hàm **.corr()** để tìm mối quan hệ tương quan giữa các cột trong dữ liệu.

# Đo lường tương quan

fish\_data.corr()



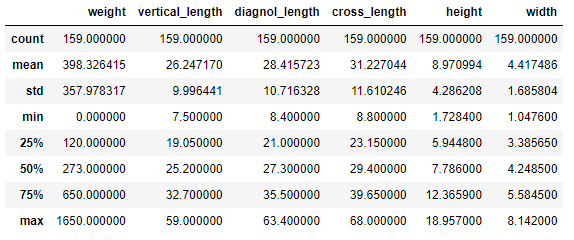
Hình : Hình mô tả 6

***Tạo một bản tóm tắt thống kê***

Tiếp theo là tạo một bản tóm tắt thống kê để giúp mô tả tập dữ liệu. Chúng ta sẽ sử dụng phương thức **describe ()** để xuất một DataFrame với các thống kê cơ bản về dữ liệu:

# chúng ta hãy xem một bản tóm tắt thống kê.

**price\_data.describe ()**

****

Hình : Hình mô tả 7

***Tiếp theo chúng ta sẽ vẽ một biểu đồ cho mỗi cột dữ liệu.***

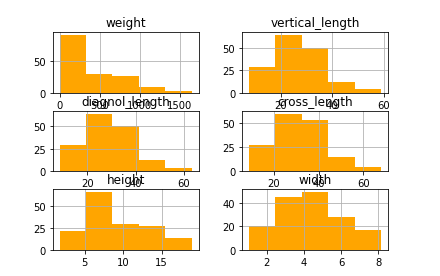
Sử dụng hàm **.hist** để vẽ Histogram thể hiện tần xuất thống kê số lần xuất hiện các dữ liệu trong tệp dữ liệu.

Hàm **Numpy histogram** không vẽ biểu đồ, nhưng nó tính toán các lần xuất hiện của dữ liệu đầu vào trong mỗi thùng, từ đó xác định diện tích (không nhất thiết là chiều cao nếu các thùng không có chiều rộng bằng nhau) của mỗi thanh.

**bins=5**, nó sẽ sử dụng 5 cột có độ rộng bằng nhau giữa giá trị đầu vào tối thiểu và giá trị đầu vào tối đa.

**Grid:**là lưới trong biểu đồ.

fish\_data.hist(grid = True,bins = 5, color = 'orange')

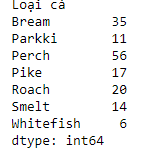


Hình : Hình mô tả 8

Tiếp theo để hiểu rõ hơn số lượng từng loại cá trong dữ liệu của chúng ta. Ta sử dụng phương thức **groupby()** của pandas.

**Pandas groupby():** được sử dụng để nhóm theo các danh mục và áp dụng một chức năng cho các danh mục. Nó cũng giúp tổng hợp dữ liệu một cách hiệu quả.

print(fish\_data.groupby('Loại cá').size())



Hình : Hình mô tả 9

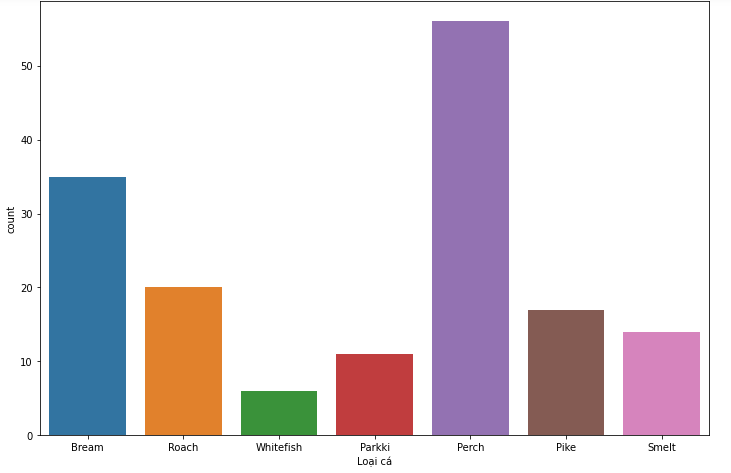
Tiếp theo ta sẽ sử dụng thư viện Seaborn để hiển thị số lượng quan sát trong mỗi loại và hiển thị kết quả dưới dạng biểu đồ thanh. Để làm điều này ta sử dụng phương thức **countplot():**

#Tần suất loại cá

plt.figure(figsize=(12,8))

sns.countplot(fish\_data['Loại cá'])

plt.show()



Hình : Hình mô tả 10

## *3.5 Xây dựng mô hình*

***Tách dữ liệu.***

Điều đầu tiên chúng ta cần làm là tách dữ liệu thành tập huấn luyện và tập kiểm tra.

Tập huấn luyện là những gì chúng ta sẽ huấn luyện mô hình.

Tập kiểm tra là những gì chúng ta sẽ thử nghiệm nó.

**Train\_test\_split()** thực hiện tách và trả về bốn chuỗi theo thứ tự sau:

* **x\_train**: Phần đào tạo của dãy số đầu tiên (x)
* **x\_test**: Phần kiểm tra của chuỗi đầu tiên (x)
* **y\_train**: Phần đào tạo của chuỗi thứ hai (y)
* **y\_test**: Phần kiểm tra của chuỗi thứ hai (y)

**test\_size:** là số xác định kích thước của tập kiểm tra.

**test\_size=** 0.30 có nghĩa là khoảng 30 phần trăm mẫu sẽ được gán cho dữ liệu kiểm tra và 70 phần trăm còn lại sẽ gán cho dữ liệu đào tạo.

Để làm cho các kiểm tra của chúng ta có thể tái tạo, cần một phân tách ngẫu nhiên với cùng một đầu ra cho mỗi lệnh gọi hàm. Chúng ta có thể làm điều dó với random\_state.

**random\_state:** là đối tượng điều khiển sự ngẫu nhiên trong quá trình tách, quyết định việc chia dữ liệu thành các chỉ số đào tào và kiểm tra trong trường hợp của chúng ta.

GIÁ trị **random\_state** không quan trọng – nó thể là bất kỳ số nguyên nào.

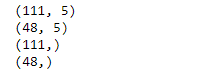
X\_train, X\_test, Y\_train, Y\_test = train\_test\_split( X,Y,test\_size=0.3, random\_state=10)

print(X\_train.shape)

print(X\_test.shape)

print(Y\_train.shape)

print(Y\_test.shape)



Hình : Hình mô tả 11

**Tạo và điều chỉnh mô hình**

Tất cả những gì chúng ta làm là tạo một phiên bản của mô hình hồi quy tuyến tính từ Sklearn và sau đó gọi phương thức **fit ()** để đào tạo mô hình trên dữ liệu đào tạo của chúng ta.

Hàm **fit()** đào tạo hoặc chỉnh sửa mô hình.

# tạo một đối tượng mô hình Hồi quy tuyến tính.

regression\_model = LinearRegression()

# chuyển qua tập dữ liệu X\_train & y\_train.

regression\_model.fit(X\_train, Y\_train)

Với mô hình hồi quy tuyến tính, phù hợp với mô hình có nghĩa là xác định các giá trị chặn **( regression\_model.intercept)** và độ dốc **( regression\_model.coef)** tốt nhất của đường hồi quy.

Ở đây ta có **intercept** của mô hình là: -520.1

print("Intercept cho mô hình là: {:.4}".format(regression\_model.intercept\_))

**Hàm format()** sử dụng để định dạng một giá trị truyền vào thành một định dạng cụ thể.

Coef của mô hình là: **[ 40.93617308 10.06005782 -23.01662758 24.72253785 14.4039206 ]**

print(regression\_model.coef\_)

**Đưa ra nhiều dự đoán trên tập dữ liệu Test**

Để có một mô hình hoạt động tốt, để có thể test dữ liệu trên dữ liệu đã thiết lập.

Ta sẽ gọi phương thức **predict ()** và chuyển qua tập dữ liệu X\_test

tại thời điểm đó, một danh sách các dự đoán sẽ được trả về.

#Đưa ra nhiều dự đoán trên tập test

Y\_predict\_test = regression\_model.predict(X\_test)

# Hiển thị ra 10 dự đoán đầu tiên

Y\_predict\_test[:10]

**Kết quả trả về:** array([-87.84321884, 168.20569933, 207.55330308, 687.70771533,

450.59823258, 229.81721886, 418.22868451, 547.28559303,

536.02016027, 367.92959207])

**Đưa ra nhiều dự đoán trên tập dữ liệu Train**

#Đưa ra nhiều dự đoán trên tập train

Y\_predict\_train = regression\_model.predict(X\_train)

# Hiển thị ra 10 dự đoán đầu tiên

Y\_predict\_train[:10]

**Kết quả trả về:** array([699.71487794, 108.36323164, 190.47343982, 140.74390037,

173.1331627 , 877.38738143, 3.45042974, 82.76013741,

237.40162985, 376.97562926])

***3.6 Biễu diễn lên đồ thị***

Bây giờ chúng ta có mô hình hoàn chỉnh. Tiếp theo chúng ta tạo một biểu đồ.

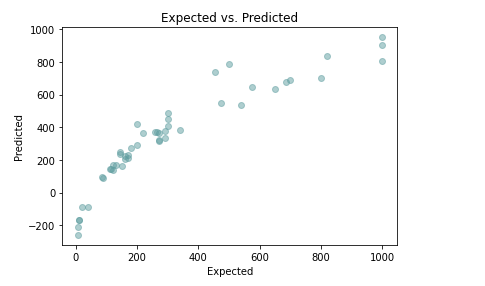
### ***3.6.1 Đồ thị biểu diễn giá trị thực tế và giá trị dự doán của Y\_test và Y\_predict\_test.***

plt.scatter(Y\_test, Y\_predict\_test, color ='cadetblue' ,alpha=0.5)

plt.xlabel("Expected")

plt.ylabel("Predicted")

plt.title("Expected vs. Predicted")



Hình : Hình mô tả 12

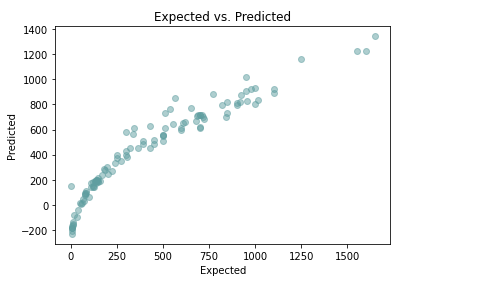
### ***3.6.2 Đồ thị biểu diễn giá trị thực tế và giá trị dự doán của Y\_train và Y\_predict\_train.***

plt.scatter(Y\_train, Y\_predict\_train, color ='cadetblue' ,alpha=0.5)

plt.xlabel("Expected")

plt.ylabel("Predicted")

plt.title("Expected vs. Predicted")



Hình : Hình mô tả 13

## *3.7 Đánh giá mô hình*

Khi chúng ta có mô hình hoạt động mà có thể sử dụng để đưa ra các dự doán, do đó cần phải đánh giá mức độ hữu ích mô hình của chúng ta.

Chúng ta sẽ tạo lại mô hình tương tự của mình bằng cách sử dụng thư viện **statsmodel.api**.

Chúng ta sẽ sử dụng mô-đun **statsmodels** của Python để triển khai phương pháp hồi quy tuyến tính **( Ordinary Least Squares – OLS).**

Lý do tại sao là chúng có nhiều chức năng tích hợp điều đó giúp việc tính toán các chỉ số như khoảng tin cậy và giá trị p trở nên dễ dàng. Đầu ra từ **statsmodel.api** sẽ không giống với thư viện sklearn của chúng ta nhưng nó sẽ xem xem tương tự.

# define our intput

X2 = sm.add\_constant(X)

# create a OLS model.

model = sm.OLS(Y, X2)

# fit the data

est = model.fit()

Ước lượng giá trị P: ta sử dụng phương thức pvalues trên mô hình đánh giá của thư viện **statsmodel.api**.

**est.pvalues**

**Kết quả nhận được:** array([8.44136168e-37, 1.23018636e-01, 8.76005273e-01, 9.64307949e-02, 1.45847750e-03, 2.71689248e-01])

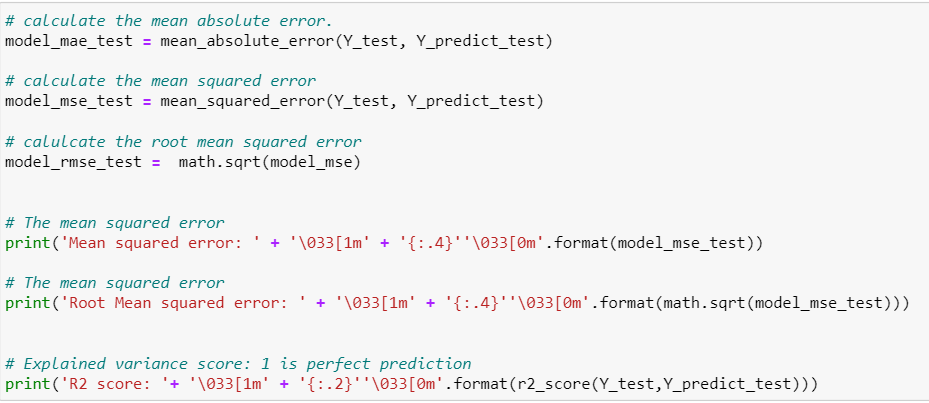
Tiếp theo chúng ta sẽ tính toán một số chỉ số để giúp định lượng mức độ phù hợp của mô hình của chúng ta với dữ liệu. Dưới đây là một số chỉ số phổ biến:

* **Sai số tuyệt đối trung bình (Mean Absolute Error (MAE)):** Là giá trị trung bình của giá trị tuyệt đối của các sai số. Chỉ số này cung cấp ý tưởng về độ lớn nhưng không có ý tưởng về hướng (quá cao hoặc quá thấp).
* **Mean Squared Error (MSE):** Là trung bình của các lỗi bình phương .MSE phổ biến hơn MAE vì MSE “punishes" nhiều lỗi nghiêm trọng hơn.
* **Root Mean Squared Error (RMSE):** Là căn bậc hai của giá trị trung bình của các lỗi bình phương. RMSE thậm chí còn được ưa chuộng hơn vì nó cho phép chúng ta giải thích kết quả đầu ra theo đơn vị y.
* **R-Squared (R-Bình phương)**

Để nhận được giá trị R2 ta sử dụng hàm **.score()**

Chỉ số R-Squared cung cấp cho chúng ta một cách để đo lường dữ liệu của chúng ta phù hợp với mô hình như thế nào. Chỉ số R-Squared càng cao, dữ liệu càng phù hợp với mô hình của chúng tôi.

**Đánh giá trên Y\_test và Y\_predict\_test**

****

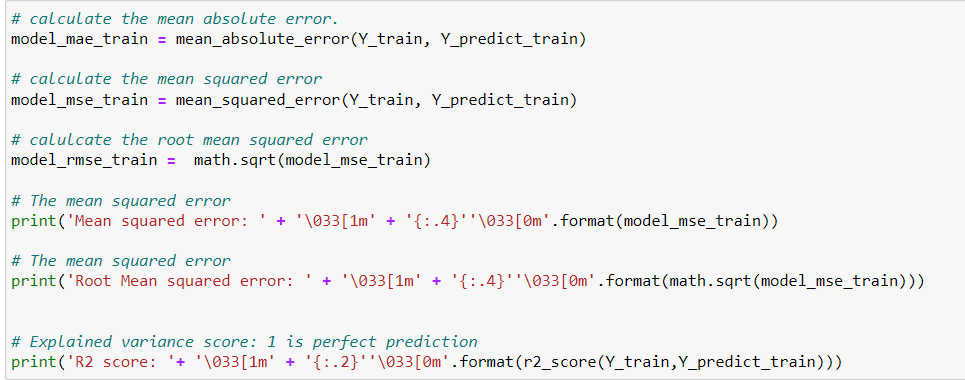
Hình : Hình mô tả 14

Mean Squared Error (MSE) của Y\_test, Y\_predict\_test là 1.421e+04

Root Mean Squared Error (RMSE) của Y\_test, Y\_predict\_test là: 119.2

R2 Score là: 0.81

**Đánh giá trên Y\_train và Y\_predict\_train**



Hình : Hình mô tả 15

Mean Squared Error (MSE) của Y\_train, Y\_predict\_train là 1.526e+04

Root Mean Squared Error (RMSE) của Y\_train, Y\_predict\_train là: 123.5

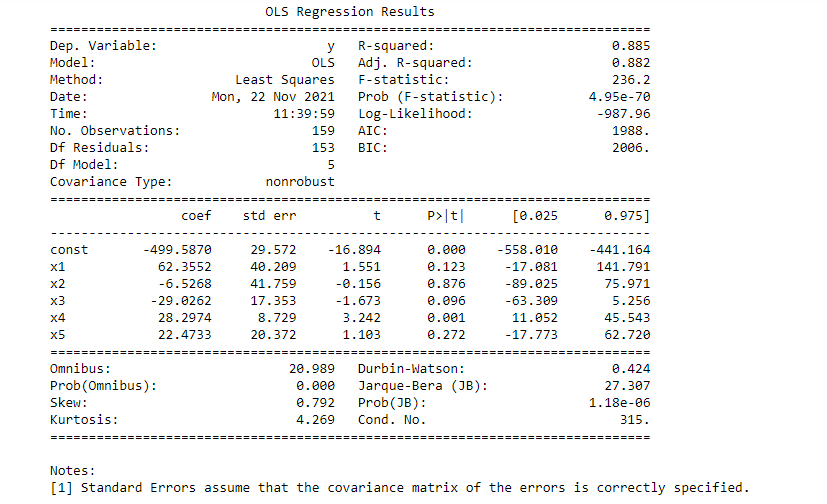
R2 Score là: 0.9

**Tạo tóm tắt đầu ra mô hình**

Ở đây chúng ta sử dụng thư viện statsmodel.api vì chúng ta có thể tạo cùng một mô hình mà chúng ta đã làm ở trên. Một số chỉ số có thể khác nhau một chút, nhưng chúng thường phải giống nhau

Ở đây chúng ta sử dụng phương thức **summary ()** để tạo đầu ra cho chúng ta: **print(est.summary())**

Bây giờ nhìn vào bảng dưới đây, chúng ta có cái nhìn tổng quan về cách mô hình của chúng ta hoạt động:



Hình : Hình mô tả 16

# TÀI LIỆU THAM KHẢO

Aung Pyae (2019), Database of common fish species for fish market  
https://www.kaggle.com/aungpyaeap/fish-market