

**ĐỀ TÀI: LASSO REGRESSION**

**BÁO CÁO ĐỒ ÁN 2**

**ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

🕮🙞🙜-----

OA ABC

-----🙞🙜🕮🙞🙜-----



**GVHD: TS Trần Nhật Quang**

**SVTH: MSSV**

**Võ Gia Huy 16110092**

**Phan Thị Huỳnh Tú 16110256**

TP.HỒ CHÍ MINH – 05/2019

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Giảng viên hướng dẫn

(ký và ghi họ tên)

…………………….....

# LỜI CẢM ƠN

*Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Nay em xin được phép gửi lời cảm ơn chân thành này đến thầy Trần Nhật Quang, người đã trực tiếp hỗ trợ em trong suốt quá trình định hướng chọn đề tài, hướng dẫn, nhận xét và góp ý cũng như cung cấp tài liệu tham khảo. Nếu không có những lời hướng dẫn, những kinh nghiệm thực tiễn của thầy thì em nghĩ rằng bài thu hoạch này sẽ khó có thể hoàn thiện và hoàn thành đúng thời hạn được. Một lần nữa, em xin cảm ơn thầy.*

*Em cũng xin chân thành cảm ơn các quý thầy cô trong khoa Công Nghệ Thông Tin đã giúp đỡ hỗ trợ kiến thức cũng như giải đáp thắc mắc của em. Cùng với đó, em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp cho em hoàn thiện đề tài hơn.*

*Bài thu hoạch được thực hiện trong khoảng thời gian gần hai tuần. Khoảng thời gian có hạn, cùng với kiến thức còn hạn chế và còn nhiều bỡ ngỡ khác do đó thiếu sót là điều không thể tránh khỏi nên em rất mong nhận được những ý kiến đóng góp quý báo của các quý Thầy Cô để kiến thức của em được hoàn thiện hơn sau này. Em xin chân thành cảm ơn.*

*Thành phố Hồ Chí Minh, ngày 23 tháng 05 năm 2019*

# Mục lục

[LỜI CẢM ƠN 3](#_Toc9543824)

[Mục lục 4](#_Toc9543825)

[Chương 1 Tìm hiểu về đạo văn 5](#_Toc9543826)

[1.1 Đạo văn là gì 5](#_Toc9543827)

[1.2 Nguyên tắc và quy định 5](#_Toc9543828)

[1.3 Lời cam kết 5](#_Toc9543829)

[Chương 2 Ý tưởng thuật toán 6](#_Toc9543830)

[Chương 3 Giới thiệu Lasso regression 7](#_Toc9543831)

[3.1 Regularization technique 7](#_Toc9543832)

[Chương 4 Mô tả Lasso regression[2] 9](#_Toc9543833)

[Chương 5 Generality Gradient decent of Lasso regression 12](#_Toc9543834)

[5.1 Gradient decent cho RSS 12](#_Toc9543835)

[5.2 Subgradient cho L1 13](#_Toc9543836)

[Chương 6 Mô tả dataset 16](#_Toc9543837)

[Chương 7 Source code 17](#_Toc9543838)

[Chương 8 Kết quả 20](#_Toc9543839)

[Chú thích 23](#_Toc9543840)

[Tài liệu tham khảo 23](#_Toc9543841)

# Danh mục các hình

[Hình 1 Hình ảnh biểu thị mức độ over-fitting theo power (degree). 8](#_Toc9543964)

[Hình 2 Hình biểu diễn độ lớn của hệ số so với [3] 10](#_Toc9543965)

[Hình 3 Hình biểu diễn model học dược theo lamda (alpha) 11](#_Toc9543966)

[Hình 4 Hình ảnh miền giá trị của Lasso và RSS 12](#_Toc9543967)

[Hình 5 Hình ảnh thể hiện giá trị của theta trên trục tọa độ 14](#_Toc9543968)

[Hình 6 Hình ảnh miêu tả của lasso regression so với của linear regression 17](#_Toc9543969)

[Hình 7 Dữ liệu đầu vào. 19](#_Toc9543970)

[Hình 8 Hiện tượng over-fitting khi tăng biến degree 23](#_Toc9543971)

[Hình 9 Hình hình ảnh model học được khi tăng giá trị lamda 23](#_Toc9543972)

[Hình 10 Hình số lượng feature bằng không theo độ lớn của lamda 24](#_Toc9543973)

# Chương 1 Tìm hiểu về đạo văn

## Đạo văn là gì

Theo Merriam-Webster Online Dictionary, đạo văn nghĩa là:

* Ăn cắp và hình thành những ý tưởng hay ngôn từ mới khởi nguồn từ ý tưởng của ai đó.
* Sử dụng sản phẩm của một ai đó mà không công bố nguồn.
* Giới thiệu một ý tưởng hay sản phẩm mới được chuyển hóa từ một nguồn đã có từ trước.

## Nguyên tắc và quy định

* Khi mườn từ ngữ của người khác, dùng dấu ngoặc kép và ghi rõ đầy đủ thông thi nguồn (tên tác giả, ngày tháng, số trang).
* Nguồn từ internet của phải được nêu rõ.
* Khi mượn ý tưởng của người khác phải nêu rõ nguồn gốc ý tưởng đó.
* Không diễn giải đoạn văn của người khác và truyền đạt như nó là của chính mình.

## Lời cam kết

Chúng tôi xin cam đoan đồ án này là do chính chúng tôi thực hiện, Chúng tôi không sao chép, sử dụng bất kỳ tài liệu, mã nguồn của người khác mà không ghi rõ nguồn gốc. Chúng tôi xin chịu hoàn toàn trách nhiệm nếu vi phạm.

Ký tên

Võ Gia Huy Phan Thị Huỳnh Tú

# Chương 2 Ý tưởng thuật toán

Học có giám sát (Supervised) được chia thành hai dạng lớn là bài toán hồi quy (regression) và bài toán về phân loại (classification). Trong các thuật toán hồi quy, chúng ta đã quen thuộc với hai mô hình Linear regression và Polynomial regression. Đối với linear regression, điểm yếu của thuật toán này là model học được (đường thẳng, mặt phảng, siêu mặt phẳng) dường như khá đơn giản so với dữ liệu thật dẫn đến tình trạng under-fitting. Vậy làm sao để hạn chế được những nhược điểm của mô hình này, chúng tôi sẽ giới thiệu thuật toán *Lasso Regression*.

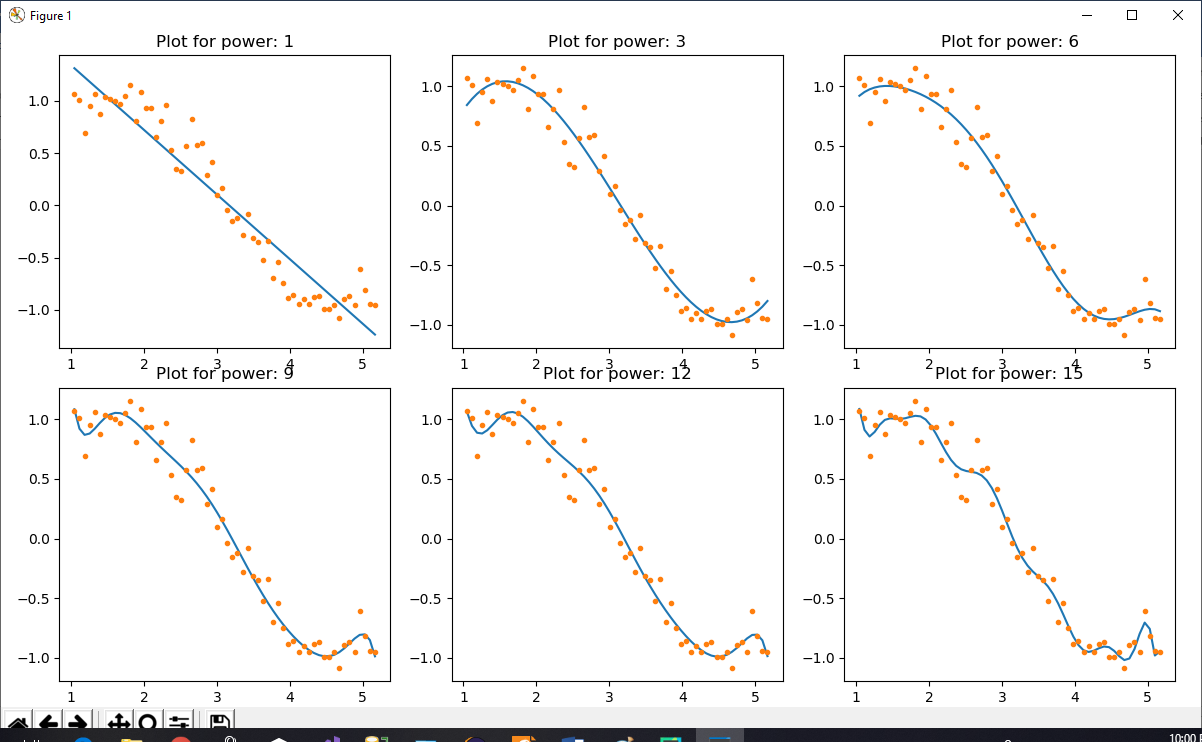
# Chương 3 Giới thiệu Lasso regression

Lasso regression được viết tắt từ *Least Absolute Shrinkage and Selection Operator* là một phương pháp ngăn chặn hiện tượng over-fitting được hình thành từ ý tưởng của mô hình hồi quy tuyến tính. Lasso sử dụng tham số  (lamda)(1) để phạt các (2)(theta) làm nhiễu mô hình để cho ra chính xác khuynh hướng của bộ dữ liệu. Vậy trước khi tìm hiểu Lasso, ta cần tìm hiểu về kỹ thuật chính quy (regularization technique) là gì.

## 3.1 Regularization technique

Regularization trong Machine learning là một khái niệm quan trọng dẫn đến tình trạng over-fitting. Điều quan trọng nhất của regularization là train ra một model tốt. Để giải thích regularization, chúng tôi đưa ra một ví dụ: [1]

Giả sử chúng tôi muốn có một model tính lương theo độ tuổi, rõ ràng linear regression là phù hợp với model này, với biết độc lập x là tuổi, biến phụ thuộc y là lương. Tuy nhiên, nếu xét phương diện thực tế model này là không thành công vì quá đơn giản (too simple). Tiếp theo, ứng với mỗi cá nhân trong dữ liệu, ta thêm các tính năng tuổi, giới tính, trình độ học vấn. Khi đó chúng ta có một ma trận X các tính năng và vecto đầu ra Y. Chưa dừng lại ở đó, khi ta thêm tiếp những tính năng địa chỉ, nghề nghiệp bố mẹ, nền tảng xã hội, số người con, cân nặng, số cuốn sách đã đọc, màu sắc ưa thích, món ăn ưa tích, vv và vv. Khi đó, model sẽ cố học theo những tính năng trên và học luôn cả những dữ liệu nhiễu dẫn đến hiện tượng over-fitting.



Hình 1 Hình ảnh biểu thị mức độ over-fitting theo power (degree).

Vậy regularization sẽ giúp ngăn chặn hiện tưởng này bằng cách xử phạt hệ số của những tính năng làm nhiễu mô hình, làm cho mô hình trở về đúng khuynh hướng của nó. Khi tuning parameter , những tính năng nhiễu hệ số của nó sẽ được đưa về đúng về 0.Để phạt các tính năng nhiễu, đối với Lasso, cần thực hiện regularization L1, ta lấy lamda nhân với tổng trị tuyệt đối các theta. Những feature bị nhiễu trở về 0 sẽ làm mô hình thưa thớt hơn. Sau đó ta dùng phương pháp cross-validation: chia dữ liệu ra thành traning data để fit giá trị của và test data để kiểm tra các đối tượng còn lại, sau đó lặp lại quá trình này trong khi thay đổi. Cuối cùng ta chọn ra lamda có minimize loss function (cost function đủ lớn, số feature bị phạt về 0 chỉ là những feature làm nhiễu model).

# Chương 4 Mô tả Lasso regression[2]

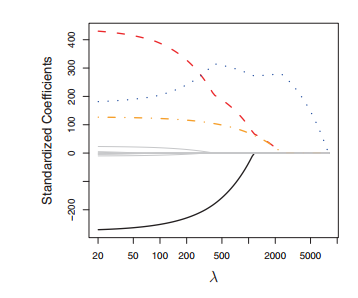
Trong đó

* m: số dòng dữ liệu.
* n: số tính năng (number of feature).
* : tham số cân bằng lượng để tối thiểu hóa RSS so với tối thiểu tổng trị tuyệt đối các hệ số.

hoạt động dựa trên sự đánh đổi giữa việc cân bằng RSS và cường độ của các hệ số. Tức là khi càng nhiều hệ số được phạt về 0, RSS sẽ càng lớn (kéo theo độ lớn của cost function sẽ tăng).

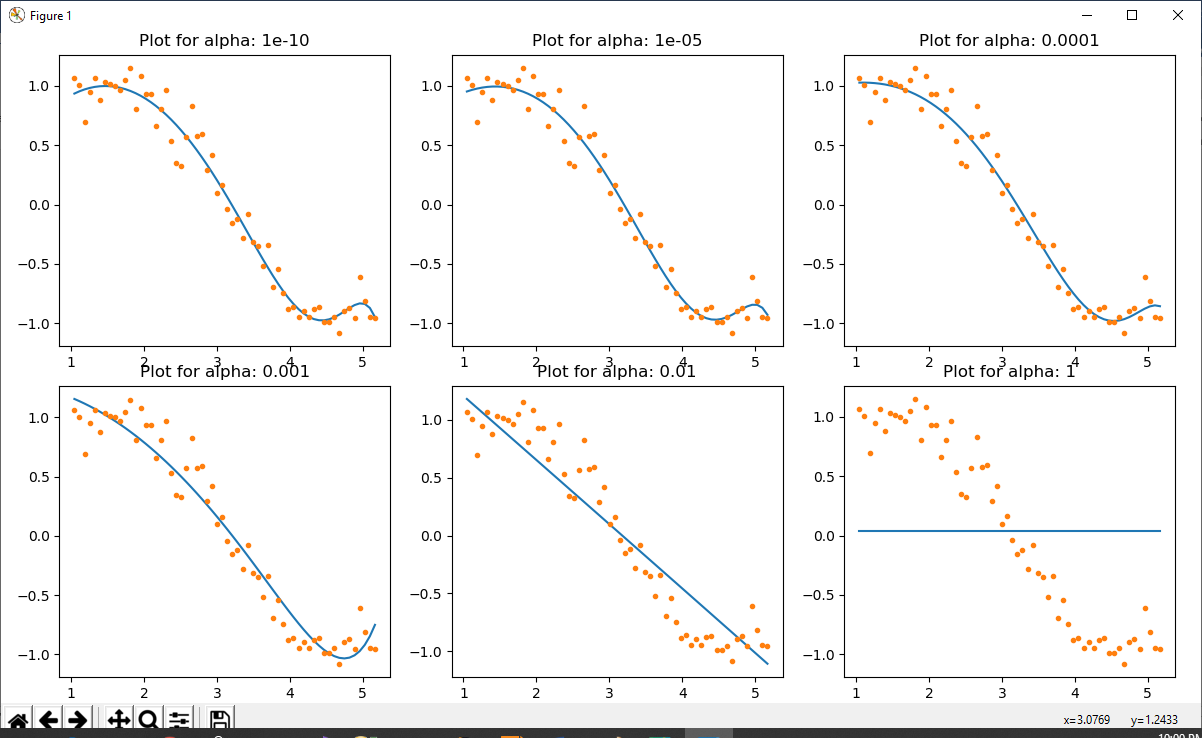
Có ba trường hợp xảy ra với :

* =0, tất cả các hệ số giống y như hồi quy tuyến tính đơn giản
* =, khi đó, hàm Cost() sẽ là vô cùng
* 0<<, hệ số  sẽ dao động từ 0 đến hệ số của mô hình hồi quy tuyến tính



Hình 2 Hình biểu diễn độ lớn của hệ số so với [3]

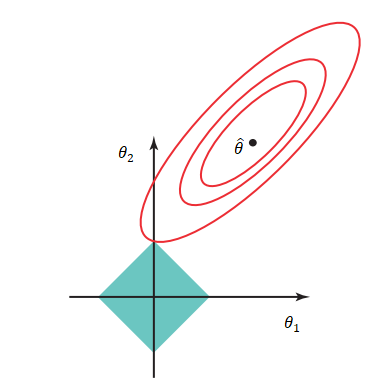
Ta có thể nhận thấy rằng, khi  đủ lớn, một số hệ số đã trở về 0, khi tiến ra vô cùng, tất cả những hệ số đều bằng 0. Lúc này, cost() là vô cùng lớn, model học được không còn được như ý muốn.



Hình 3 Hình biểu diễn model học dược theo lamda (alpha)

Nhận xét: Độ phức tạp của mô hình giảm đi khi ta tăng các giá trị . Ta chú ý hình cuối cùng, khi =1 ta thấy model học được hoàn toàn sai khác so với dữ liệu thật. Các tính năng của dữ liệu bây giờ đã không còn ảnh hưởng đến model nữa, Cost() sẽ rất lớn.

Khi n=2, các theta sẽ có RSS nhỏ nhất và nằm trong hình thoi có phương trình . “s” được gọi là “budget”, là vùng được tính bởi độ lớn của .



Hình 4 Hình ảnh miền giá trị của Lasso và RSS

Trong trường hợp ta có hai tính năng, ứng với mỗi đường elip là giá trị RSS không đổi, càng xa trung tâm đường eclip, các giá trị RSS càng tăng. Như đã nói ở trên, Lasso regression sẽ phạt những hệ số nhiễu, trong hình ảnh trên, ta nhật thấy rằng hình eclip giao với miền giá trị của Lasso tại một điểm trên trục tọa độ. Điều này chứng tỏ sẽ có một tính năng bị triệt tiêu đi. Với số lượng lớn dimensions, sẽ có rất nhiều hệ số trở về 0.

Hình ảnh trên cho chúng ta thấy miền giá trị của Lasso với hai tính năng (feature), vậy với ba tính năng, miền giá trị của Lasso sẽ là một khối đa diện (polyhedron). Khi số tính năng lớn hơn ba (n>3), miền giá trị của Lasso sẽ là khối polytope. Tuy số tính năng thay đổi dẫn tới miền giá trị của Lasso có thể thay đổi hình dạng nhưng ý tưởng của Lasso vẫn không đổi là dùng tham số lamda để xử phạt những feature bị nhiễu.[3]

# Chương 5 Generality Gradient decent of Lasso regression

Đối với Lasso Regression, ta không thể học theta chỉ qua Gradient decent vì L1 có chứa dấu giá trị tuyệt đối. Ta giải quyết bằng cách dùng Gradient decent ở RSS và dùng Subgradient ở L1 để học ra theta.

## 5.1 Gradient decent cho RSS

Cost() = -2

=

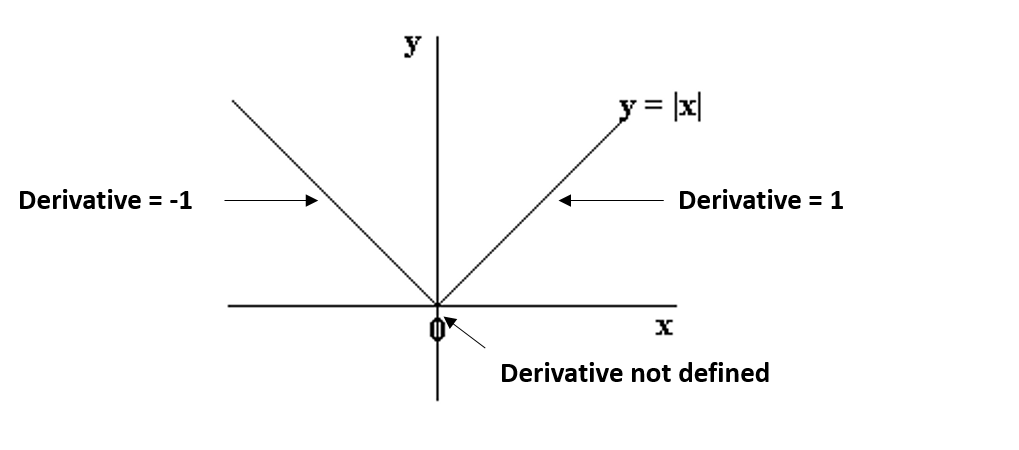
=

Đặt =

= ( luôn dương)

Ta có: Cost() =

## 5.2 Subgradient cho L1



Hình 5 Hình ảnh thể hiện giá trị của theta trên trục tọa độ

* Tại gốc tọa độ (problem point) hàm không xác định.

Sau khi tính ra từng thành phần, ta tổng chúng lại với nhau

Trường hợp 1 (): = 0

=

Ta có:

Trường hợp 2 :

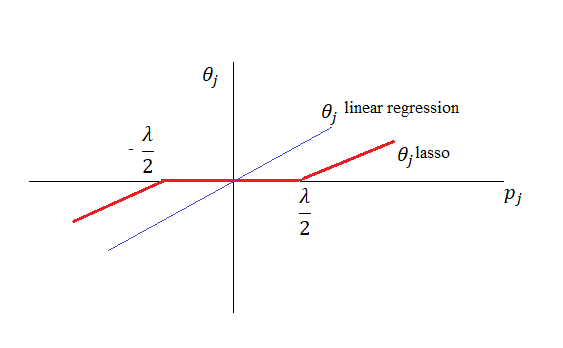
Trường hợp 3 (): = 0

=

Ta có:

Vậy:

thể hiện sự khác biệt giữa dữ liệu thực tế và kết quả dự đoán khi xem xét tất cả các theta ngoại trừ . Nếu giá trị thay đổi này nhỏ chứng tỏ không ảnh hưởng nhiều đến model, thuật toán vẫn có thể dự đoán tốt kể cả khi không có , do đó ta đặt hệ số nó bằng 0.

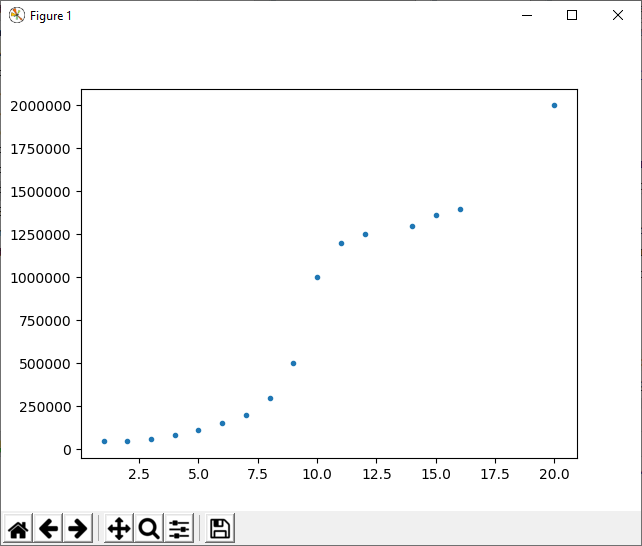


Hình 6 Hình ảnh miêu tả của lasso regression so với của linear regression

Ta nhận thấy rằng trong đoạn từ ; bằng 0. Đây là đoạn các feature bị nhiễu mất đi và chỉ còn lại những feature có ích cho mô hình.

# Chương 6 Mô tả dataset

Dữ liệu mô tả tiền lương theo cấp bậc. Lấy ý tưởng từ polynomial regression, chúng tôi nâng bậc dữ liệu và xem nó như là các feature mới.

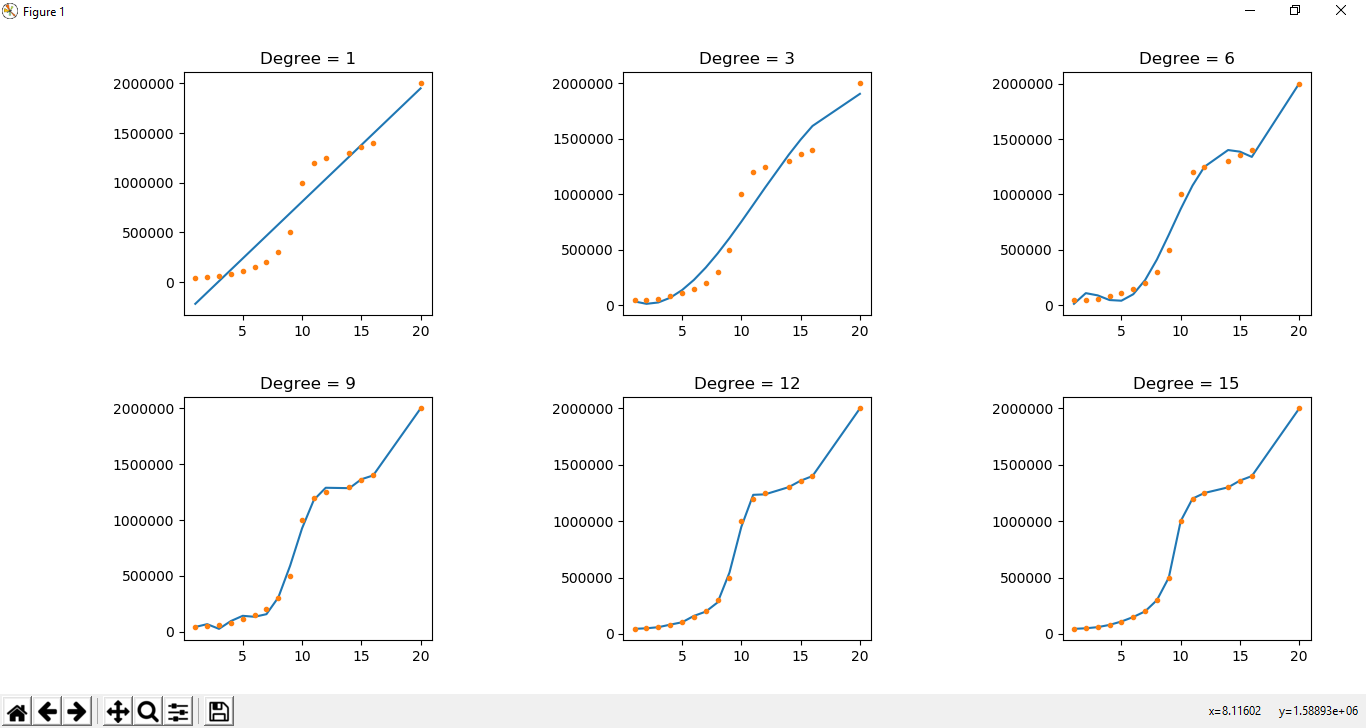


Hình 7 Dữ liệu đầu vào.

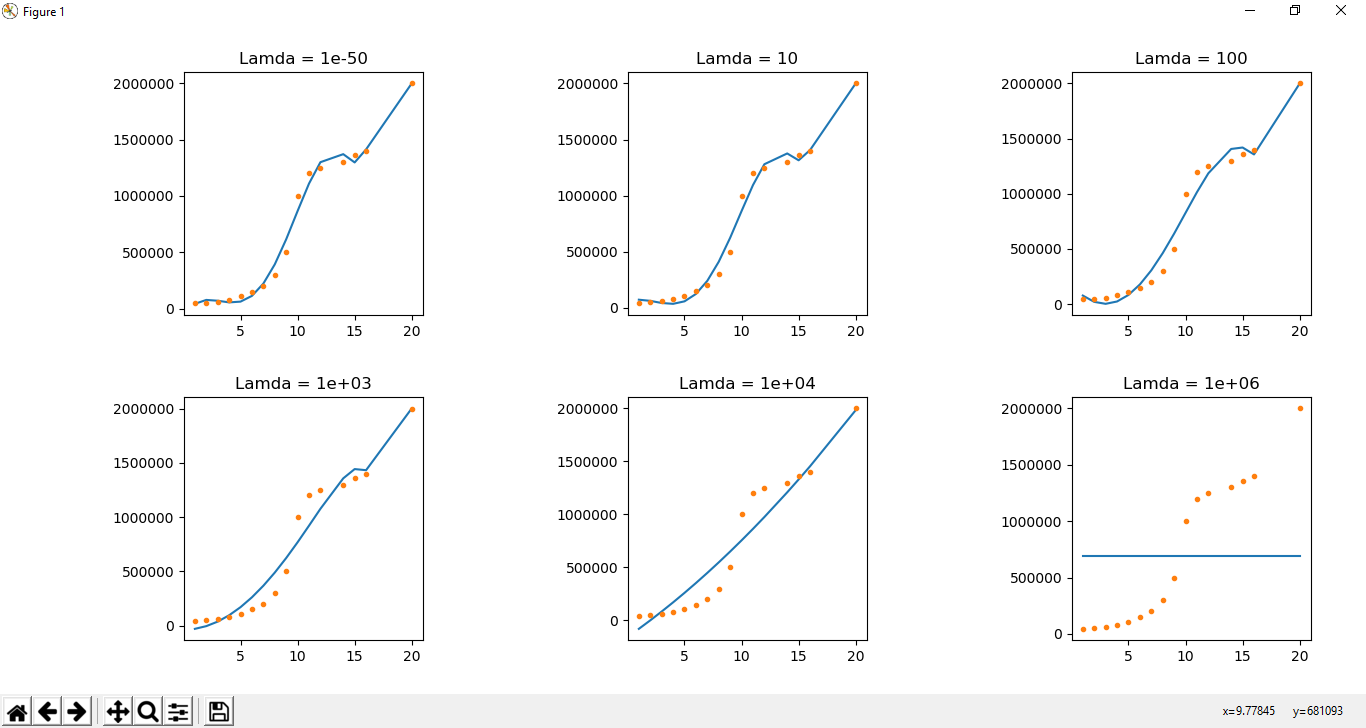
# Chương 7 Source code

*#Importing libraries. The same will be used throughout the article.***import** numpy **as** np  
**import** pandas **as** pd  
**import** random  
**import** matplotlib.pyplot **as** plt  
*#from IPython import get\_ipython  
#ipy = get\_ipython()  
#if ipy is not None:  
 # ipy.run\_line\_magic('matplotlib', 'inline')  
#from matplotlib.pylab import rcParams  
#rcParams['figure.figsize'] = 12, 10  
  
#đọc dữ liệu từ Position\_Salaries.csv*dataset = pd.read\_csv(**'Position\_Salaries.csv'**)  
*#set biến x là dữ liệu vị trí*x = dataset.iloc[:, 0].values  
*#set bien y la tien luong tuong ung voi vi tri*y = dataset.iloc[:, 1].values  
*# vẽ đồ thị các x,y tương ứng*data = pd.DataFrame(np.column\_stack([x,y]),columns=[**'x'**,**'y'**])  
plt.plot(data[**'x'**],data[**'y'**],**'.'**)  
plt.show()  
*#tính các biến x từ bậc 2 tới bậc 15 tương ứng mỗi vòng lặp x tăng một bậc***for** i **in** range(2,16): *#bậc 1 tồn tại* colname = **'x\_%d'**%i *#biến x mới của bậc mới* data[colname] = data[**'x'**]\*\*i  
print(data.head())  
  
  
*#Import Linear Regression model from scikit-learn.***from** sklearn.linear\_model **import** LinearRegression  
**def** linear\_regression(data, power, models\_to\_plot):  
 *#khởi tạo predictors:* predictors=[**'x'**]  
 **if** power>=2:  
 predictors.extend([**'x\_%d'**%i **for** i **in** range(2,power+1)]) *#nếu bậc lớn hơn 2 thì biến x được mở rộng ra(2 feature)  
 #Fit the model* linreg = LinearRegression(normalize=**True**)  
 linreg.fit(data[predictors],data[**'y'**])  
 y\_pred = linreg.predict(data[predictors])  
 *#vẽ sơ đồ polinomial tương ứng* **if** power **in** models\_to\_plot:  
 plt.subplot(models\_to\_plot[power])  
 plt.tight\_layout()  
 plt.plot(data[**'x'**],y\_pred)  
 plt.plot(data[**'x'**],data[**'y'**],**'.'**)  
 plt.title(**'Degree = %d'**%power)  
 *#trả về kết quả của res* rss = ((sum((y\_pred-data[**'y'**])\*\*2))/(2\*len(x))) *#cost function* ret = [rss]  
 ret.extend([linreg.intercept\_])  
 ret.extend(linreg.coef\_)  
 **return** ret  
*#Khởi tạo dataframe để chứa các theta*col = [**'rss'**,**'intercept'**] + [**'coef\_x\_%d'**%i **for** i **in** range(1,16)]  
ind = [**'model\_pow\_%d'**%i **for** i **in** range(1,16)]  
coef\_matrix\_simple = pd.DataFrame(index=ind, columns=col)  
  
*#Khai báo degree cho model*models\_to\_plot = {1:231,3:232,6:233,9:234,12:235,15:236}  
  
*#Lặp lại các bước tính tăng dần theo power(degree) và cho ra kết quả***for** i **in** range(1,16):  
 coef\_matrix\_simple.iloc[i-1,0:i+2] = linear\_regression(data, power=i, models\_to\_plot=models\_to\_plot)  
plt.show()  
  
*#lasso regression***from** sklearn.linear\_model **import** Lasso  
**def** lasso\_regression(data, predictors, alpha, models\_to\_plot={}):  
 *#Fit the model* lassoreg = Lasso(alpha=alpha,normalize=**True**, max\_iter=1e5)  
 model = lassoreg.fit(data[predictors],data[**'y'**])  
 y\_pred = lassoreg.predict(data[predictors])  
 print(model.score(data[predictors],data[**'y'**]))  
 print(**'Bias: \n'**, lassoreg.intercept\_)  
 print(**'Thetas: \n'**, lassoreg.coef\_)  
 *#Vẽ sơ đồ theo lasso* **if** alpha **in** models\_to\_plot:  
 plt.subplot(models\_to\_plot[alpha])  
 plt.tight\_layout()  
 plt.plot(data[**'x'**],y\_pred) *#vẽ model dự đoán* plt.plot(data[**'x'**],data[**'y'**],**'.'**) *# vẽ dữ liệu thật* plt.title(**'Lamda = %.3g'**%alpha) *# tên của model  
  
 #trả về kết quả của ret* rss = ((sum((y\_pred-data[**'y'**])\*\*2))/(2\*len(x))) *#cost function* ret = [rss]  
 ret.extend([lassoreg.intercept\_])  
 ret.extend(lassoreg.coef\_)  
 **return** ret  
*#Khởi tạo dữ liệu dự đoán cho 15 bậc của x*predictors=[**'x'**]  
predictors.extend([**'x\_%d'**%i **for** i **in** range(2,16)])  
  
*#Khai báo alpha(lamda) để test*alpha\_lasso = [1e-50, 1e-10, 1e-8, 1e-5,1, 10, 100, 1000, 10000, 1000000]  
  
*#khởi tạo dataframe để chứa các theta*col = [**'rss'**,**'intercept'**] + [**'coef\_x\_%d'**%i **for** i **in** range(1,16)]  
ind = [**'alpha\_%.2g'**%alpha\_lasso[i] **for** i **in** range(0,10)]  
coef\_matrix\_lasso = pd.DataFrame(index=ind, columns=col)  
  
*#Chọn ra 6 trong số 10 dữ liệu alpha(lamda) để cho vào đồ thị*models\_to\_plot = {1e-50:231, 10:232,100:233, 1000:234, 10000:235, 1000000:236}  
  
*#Lặp lại 10 giá trị của alpha(lamda) để so sánh kết quả.***for** i **in** range(10):  
 coef\_matrix\_lasso.iloc[i,] = lasso\_regression(data, predictors, alpha\_lasso[i], models\_to\_plot)  
plt.show()  
z = []  
z = coef\_matrix\_lasso.apply(**lambda** x: sum(x.values==0),axis=1)  
print(z)  
*#source code from https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-ridge-lasso-regression-python/?fbclid=IwAR2aouDiAbjaW5J3fBHe3fhqpQGCqLGyu-LJ5STYgb-Pxv\_El4yHrssyMiQ*

# Chương 8 Kết quả



Hình 8 Hiện tượng over-fitting khi tăng biến degree



Hình 9 Hình hình ảnh model học được khi tăng giá trị lamda

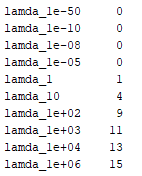
Khi =1x10-50

* Hiệu suất của thuật toán: 0.9897796757388356

* =-72095.99385173805
* Mảng giá trị [ 1.66486621e+05 -5.91947756e+04 6.65069644e+03 -3.10723287e+01 -8.30798740e+00 -2.99732039e-01 -4.93644508e-03 9.47158480e-05 1.20683858e-05 6.51890049e-07 2.75067886e-08 1.01189856e-09 3.35832212e-11 1.00854617e-12 7.72616762e-13]

Khi =1x106

* Hiệu suất của thuật toán: 0.0
* =0.0
* Mảng giá trị [ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. -0.]



Hình 10 Hình số lượng feature bằng không theo độ lớn của lamda

**Bảng phân công công việc**

|  |  |  |
| --- | --- | --- |
| Tên SV | Đánh giá chung phần trăm đóng góp | Mô tả khái quát mảng công việc SV thực hiện trong đồ án. |
| Võ Gia Huy | 50% | Làm slide, tìm hiểu source code của lasso regression, tìm hiểu phương pháp tính theta |
| Phan Thị Huỳnh Tú | 50% | Làm báo cáo, tìm hiểu thuật toán, ý tưởng thuật toán, cách tính theta |

# Chú thích

1. Trong tài liệu này sử dụng thuật ngữ hệ số phạt là (lamda), có thể tồn tại các tài liệu khác có ký hiệu là α (alpha).
2. Trong tài liệu này ký hiệu cho thuật ngữ “coefficient” là , có thể tồn tại các tài liệu khác có ký hiệu là β hoặc

# Tài liệu tham khảo

[1]Will Kailash Ahirwar. (2017, December 6). Retrieved from <https://codeburst.io/what-is-regularization-in-machine-learning-aed5a1c36590>

[2] Aarshay Jain. (2016, January 28). Retrieved from <https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-ridge-lasso-regression-python/?fbclid=IwAR0Uf30nMGtZNS_fFtzWVPwT8AGzDz_5wksXhgjhStP3FkaV4vHbSaS8lW4>

[3] [G James](https://scholar.google.com.vn/citations?user=KUIjZqgAAAAJ&hl=vi&oi=sra), [D Witten](https://scholar.google.com.vn/citations?user=bHZf-c8AAAAJ&hl=vi&oi=sra), [T Hastie](https://scholar.google.com.vn/citations?user=tQVe-fAAAAAJ&hl=vi&oi=sra), [R Tibshirani](https://scholar.google.com.vn/citations?user=ZpG_cJwAAAAJ&hl=vi&oi=sra). (2013).[An introduction to statistical learning](https://link.springer.com/content/pdf/10.1007/978-1-4614-7138-7.pdf): Statistical learning refers to a set of tools for modeling and understanding complex datasets. Springer

(Source code tham khảo đã chú thích trong phần code bài làm)