Regresyon Problemleri:

En Küçük Kareler Yöntemi (Least Square Method) ve Lineer Regresyon

* En küçük kareler yöntemi yaygın olarak denklem sayısı bilinmeyen sayısından fazla olduğu denklem aşırı belirlenmiş denklem sistemlerinin (Overdetermined equation system) en küçük karesel hata ile çözümü için kullanılır.

* Denklem sayının bilinmeyen sayısından fazla olduğu problemler genelde regresyon ve modelleme problemlerinde karşımıza çıkar. Bunun temel nedeni, gerçek uygulamalarda belirlenmesi gereken model parametreleri (ayarlanabilir parametreler) sınırlı sayıda iken sağlanması gereken veri sayısı çok fazla sayıda olur. Her bir sağlanması gereken veri aslında modelin sağlaması gereken bir denklem üretir. Dolayısı ile, çok sayıda veri nedeni ile denklem sayısının ayarlanabilen parametre sayısından büyük olduğu durumda en küçük kareler yöntemi ile toplam karesel hatayı minimum yapan parametre değerleri hesaplanabilir. Bu yöntem veri modellemesi için kullanılan bir matematiksel araca dönüşür.

* En küçük kareler yöntemi, her bir denklem için çözüm hatasının kareleri toplamını minimum yapan çözümü bulur. Bunu, aşağıdaki lineer regresyon probleminde açıklayalım.

Sahadan toplanan veriler $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), .., (x_n, y_n)$ olsun. Bu n adet veriyi lineer bir doğru olan $y = mx + c$ doğrusu ile lineer olarak modellemek isteyelim. Bu noktada modeli elde etmek için verilen $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), .., (x_n, y_n)$ n adet noktayı sağlayan doğrunun parametreleri olan m ve c parametrelerini bulmalıyız. Bu noktaları denklemde kullanarak denklem sistemini elde edelim:

(x_1, y_1) verisi için model sağlatılırsa $y_1 = mx_1 + c$ denklemi

(x_2, y_2) verisi için model sağlatılırsa $y_2 = mx_2 + c$,

....

(x_n, y_n) sağlatılırsa $y_n = mx_n + c$ denklemi elde edilir.

Böylece bütün veri noktalarının sağlanması için

$y_1 = mx_1 + c$

$y_2 = mx_2 + c$

.....

$y_n = mx_n + c$

Burada, iki bilinmeyenli (Bunlar m ve c parametreleridir) n adet denklem sistemi çözülmesi gerekmektedir. Bilinmeyen sayısı 2 adet ancak denklem sayısı $n > 2$ dir. O halde denklem sayısı bilinmeyen sayısından fazladır ve aşırı belirlenmiş (overdetermined) bu problemde bütün denklemleri hatasız sağlayan tek bir çözüm bulunamaz. (ill-posed problemdir) Kesin olmayan (hatalı) bir çok çözüm olabilir. En küçük kareler yöntemi ile toplam karesel hatayı minimize eden optimal bir çözüm bulunabilir.

[Not (Denklem Sistemleri Konusunu Hatırlatma): Denklemler şöyle kategorize edilebilir.

1) Denklem sayısı ile değişken sayısı eşit olursa, denklemi sağlayan çözümler tektir ve hatasız bulunabilir. Örnek,

$$5 = m + c$$
$$1 = m - c$$

İki denklem iki bilinmeyen. Çözüm tektir ve $m = 3$ ve $c = 2$ alınırsa bütün denklemler hatasız olarak sağlanır. Dolayısı çözüm tekdir ve kesin olarak belirlenir. Bu tür denklem sistemlerine tam belirlenmiş sistemler denir.

2) Denklem sayısı değişken sayısından az olduğu duruma eksik belirlenmiş (Underdetermined equation sistem) sistemler denir. Eksik denklem kadar keyfi değişken seçilerek, çözüm keyfi değişkene bağlı ifade edilir. Örneğin aşağıdaki ifade eksik belirlenmiş bir denklem sistemidir.

$$5 = m + c$$

Denklem sayısı 1 ve bilinmeye sayısı 2'dir. Bir değişken keyfi seçilir diğeri ona bağlı ifade edilebilir. Tek bir çözüm yoktur. c keyfi seçilirse. Diğer değişken $m = 5 - c$ olarak belirlenir. c sabitine verilen keyfi değere göre denklemi hatasız belirleyen sonsuz çözüm vardır. Burada çözüm sayısı sonsuzdur ve kesindir (hatasız).

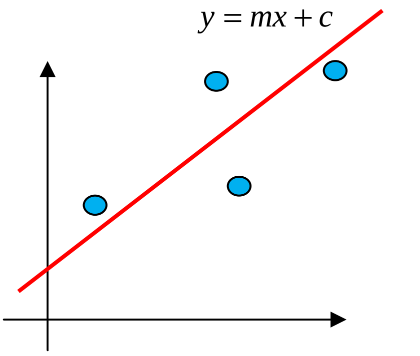
3) Denklem sayısının değişken sayısından fazla olduğu duruma aşırı belirlenmiş denklem sistemleri (Underdetermined equation sistem) denir. Örneğin 3 denklem ve 2 değişken olan aşağıdaki sistem aşırı belirlenmiştir.

$$5 = m + c$$
$$1 = m - c$$
$$2 = 2m - c$$

Bütün denklemleri hatasız sağlayan tek bir sonuç bulumayabilir. Bu durumda denklemi en az hata ile sağlayan bir çözüm aranır. Bunu çözmek için karesel hatayı minimize eden en küçük kareler yöntemi kullanılır.

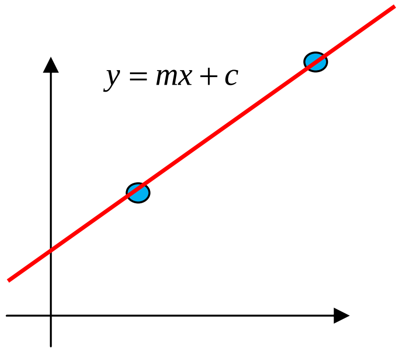
]

3) Aşırı belirlenmiş durum



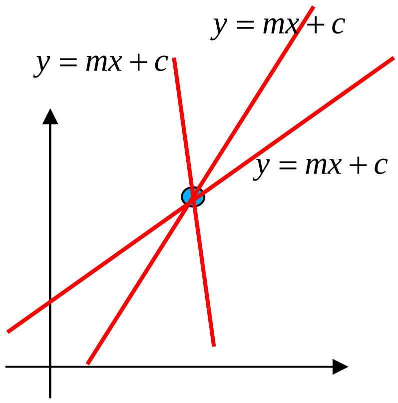
Problem, aşırı belirlenmiş denklem sistemi üretir. (Değişken sayısı 2, denklem sayısı 4) Her nokta sağlanması gereken bir denklem üretir. Dolayısı ile çözüm çok sayıdadır. Optimal çözüm (En aza hatalı) aranır.

1) Tam belirlenmiş durum



Problem, kesin belirlenmiş denklem sistemi üretir. (Değişken sayısı 2, denklem sayısı 2) Çözüm kesin (hatasız) ve tektir.

2) Eksik belirlenmiş durum



Problem, eksik belirlenmiş denklem sistemi üretir. (Değişken sayısı 1, denklem sayısı 2) Keyfi değişkene bağlı sonsuz tane kesin çözüm (hatasız) vardır.

Tam belirlenmiş denklem sistemi çözümü:

Bilinmeyen sayısı ile denklem sayısı eşit olduğu durumda, yani (x_1, y_1) ve (x_2, y_2) iki notayı sağlayan doğru denklemi şöyle bulunabilirdi.

(x_1, y_1) => $y_1 = mx_1 + c$
(x_2, y_2) => $y_2 = mx_2 + c$
Denklemler matris formunda yazılırsa,

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \Rightarrow A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \end{bmatrix}, X = \begin{bmatrix} m \\ c \end{bmatrix}, b = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

denklem sistemi matris formunda $A.X = b$ elde edilir. Burada A matrisi her zaman kare matris çıkar ve tersi alınabilir. Eşitliğin her iki tarafı A^{-1} ile çarpılırsa,

$A^{-1}.A.X = A^{-1}.b$ (burada $A^{-1}.A = I$ olduğunu hatırlayınız)

Çözüm $\boxed{X = A^{-1}.b}$ olur.

Aşırı belirlenmiş denklem sisteminin en küçük kareler yöntemi ile çözümü:

Eğer aşırı belirlenmiş sistem (denklem sayısı değişkenden fazla) varsa

$y_1 = mx_1 + c$
$y_2 = mx_2 + c$
...
$y_n = mx_n + c$

Denklem sistemi matris formunda yazılırsa,

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \cdots & .. \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ .. \\ y_n \end{bmatrix} \Rightarrow A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \cdots & .. \\ x_n & 1 \end{bmatrix}, X = \begin{bmatrix} m \\ c \end{bmatrix}, b = \begin{bmatrix} y_1 \\ y_2 \\ .. \\ y_n \end{bmatrix}$$

denklem sistemi matris formunda $A.X = b$ elde edilir.

Burada A matrisi kare matris olamaz ve tersi alınamaz. Tersinin alınabilmesi için kareselleştirme yapılmalı. $(A^T.A)$ kare matristir. Eşitliğin her iki tarafı A^T ile çarpılırsa,

$(A^T.A).X = A^T.b$ sonra $(A^T.A)^{-1}$ ile çarpılırsa,

$(A^T.A)^{-1}(A^T.A).X = (A^T.A)^{-1}A^T.b$

çözüm $X = (A^T.A)^{-1}A^T.b$ elde edilir.

Aşırı belirlenmiş sistem formundaki lineer problemlerin en küçük kareler çözümü,

$$\boxed{X = (A^T.A)^{-1}A^T.b}$$

ile elde edilir.

Özellik: Bu çözüm aynı zamanda $\boxed{\min_X \|b - AX\|_2^2}$ optimizasyon probleminin çözümüdür.

İspat: $\min_X \|AX - b\|_2^2 \Rightarrow \min_X \|e\|_2^2$ yazılabilir.

(Burada $e = b - AX = \begin{bmatrix} e_1 \\ e_2 \\ .. \\ e_n \end{bmatrix} = \begin{bmatrix} y_1 - (mx_1 + c) \\ y_2 - (mx_2 + c) \\ .. \\ y_n - (mx_n + c) \end{bmatrix}$ hata vektörüdür. Karesel hata $E = \|e\|_2^2$ le ifade edilebilir)

Karesel hatayı yazalım.

$$E=\|e\|_2^2=\left(\sqrt{e_1^2+e_2^2+..+e_p^2}\right)^2=e_1^2+e_2^2+..+e_p^2$$

$$=(y_1-(mx_1+c))^2+(y_2-(mx_2+c))^2+...+(y_p-(mx_p+c))^2=\sum_{i=1}^{p}(y_i-(mx_i+c))^2$$

$\min E(m,c)=(y_1-(mx_1+c))^2+(y_2-(mx_2+c))^2+...+(y_p-(mx_p+c))^2$ çözmek için ektremum noktalarını bulalım.

1. türev sıfıra eşitlenirse

$$\frac{dE}{dm}=2(-x_1)(y_1-(mx_1+c))+2(-x_2)(y_2-(mx_2+c))+...+2(-x_p)(y_p-(mx_p+c))=0$$

$$\frac{dE}{dc}=2(y_1-(mx_1+c))+2(y_2-(mx_2+c))+...+2(y_p-(mx_p+c))=0$$

$x_1, x_2 .. x_p$ sıfırdan farklı değerleri olduğu için bu iki denklemin birlikte sıfıra eşit olmasının bir yolu

$(y_1-(mx_1+c)=0$ => $y_1=mx_1+c$

$(y_2-(mx_2+c)=0$ => $y_2=mx_2+c$

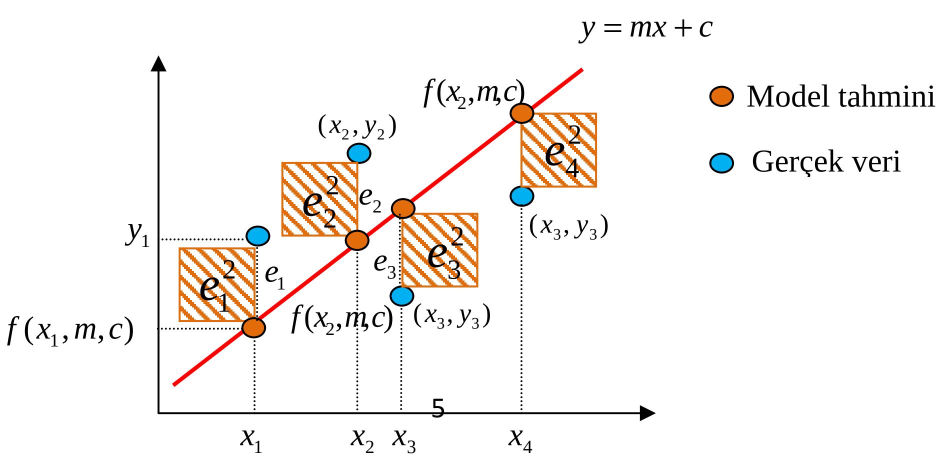
....

$(y_p-(mx_p+c)=0$ => $y_p=mx_p+c$

Bu denklem sisteminin çözümünün matris formunda

$\boxed{X=(A^T.A)^{-1}A^T.b}$ olduğunu görmüştük. O halde $X=(A^T.A)^{-1}A^T.b$ çözümü aslında $\min_X\|b-AX\|_2^2$ probleminin çözümüdür.

Geometrik yorumu: Böylece, aslında minimum kareler yöntemi aşağıda verildiği gibi hata kareleri ile ifade edilen alanları minimum yapan bir $y=f(x,m,c)=mx+c$ eğrisi bulur. Bu eğri verileri en az karesel hata ile temsil eden öğrenme modeli olur.



Örnek: Kayısı üretiminin yağış oranına göre dağılımı şöyle olsun.

Kayısı Üretimi (K)	Yağış Oranı (Y)
5	1
7	2
9	3
11	4

verilerin lineer regresyon modelini $K = mY + c$ elde ediniz. Bu modele göre yağış 3.2 olduğunda kayısı üretimini tahmin ediniz.

Tablodaki veriler $K = mY + c$'de kullanılırsa,

$5 = m + c$,

$7 = 2m + c$

$9 = 3m + c$

$11 = 4m + c$

lineer denklemleri elde edilir. Bu lineer denklemleri sağlayan minimum kareler çözümü,

$X = (A^T.A)^{-1} A^T.b$ ile bulunur. Bunun için önce denklemleri matris formunda yazalım.

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ 9 \\ 11 \end{bmatrix} \Rightarrow A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix}, X = \begin{bmatrix} m \\ c \end{bmatrix}, b = \begin{bmatrix} 5 \\ 7 \\ 9 \\ 11 \end{bmatrix}$$

$(A^T.A) = \begin{bmatrix} 30 & 10 \\ 10 & 4 \end{bmatrix}$ ve $A^T.b = \begin{bmatrix} 90 \\ 32 \end{bmatrix}$ elde edilir. Kare matris tersi alınabilir

$$(A^T.A)^{-1} = \begin{bmatrix} 0.2 & -0.5 \\ -0.5 & 1.5 \end{bmatrix}$$

$$X = \begin{bmatrix} m \\ c \end{bmatrix} \Rightarrow X = (A^T.A)^{-1} A^T.b = \begin{bmatrix} 0.2 & -0.5 \\ -0.5 & 1.5 \end{bmatrix} . \begin{bmatrix} 90 \\ 32 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

dolayısı ile $m = 2$ ve $c = 3$ ve lineer regresyon modeli $K = 2Y + 3$

Yağış tahmini uygulaması:

Bu model göre $Y = 3.2$ yağış miktarı için kayısı üretimini tahmin ediniz. Giriş değişkenimiz yağış miktarı ve çıkış değişkeni kayısı miktarı tahmidir.

$K = 2Y + 3$

$K = 2(3.2) + 3 = 9.4$ olarak tahmin edilir.

Şimdi bu regresyon problemi çözümü için karesel hatayı hesaplayalım.

Yağış Oranı (Y)	Kayısı Üretimi (K) (istenen değer)	Öğrenen model tahmini $K = 2Y + 3$	Öğrenme Hatası $e = K - (2Y + 3)$
1	5	5	0
2	7	7	0
3	9	9	0
4	11	11	0

$E = \sum_{i=1}^{n} e_i^2 = 0 + 0 + 0 + 0 = 0$ karesel hata sıfırdır.

Matlab uygulaması

```
% İnönü Unv. Bilgisayar Mühendisliği
% Makine Öğrenmesi Dersi Güz 2022
% Lineer regresyon uygulama
% X: Giriş verisi
% Y: Çıkışı verisi
% Model Y=m*x+c En küçük kareler yöntemi ile m ve c hesaplanır.
% Çözüm X=[(A*A')^(-1)]*A'*b
clear all;
U=[1 2 3 4];
Y=[5 7 9 11];
% Gurultu eklemek için Gurultu=1
Gurultu=1;
if Gurultu==1
GurultuY=2*(rand(1,length(U))-0.5);
Y=Y+GurultuY
end
U
Y
% A matrisi oluşturuluyor
for i=1:length(U)
    for j=1:2
        if j==1
           A(i,j)=U(i);
        else
           A(i,j)=1;
        end
    end
end

% b stun vektöru
b=Y';
X=[(A'*A)^(-1)]*(A')*b;

% Bulunan katsayılar
fprintf('m = %d \n',X(1));
fprintf('c = %d \n',X(2));

% Tahmin doğrusu
Ux=0.1:0.1:6;
Ymodel=X(1)*Ux+X(2);

figure(1)
plot(U,Y,'o',Ux,Ymodel,'LineWidth',2)
xlabel('U')
ylabel('Y')
legend('Eğitim Verisi','Lineer Regresyon Modeli')
grid
% Öğrenme hataları
Ytahmin=X(1)*U+X(2);
figure(2)
plot(U,Y-Ytahmin,'LineWidth',2)
xlabel('U')
ylabel('Eğitim Hatası e=d-y')
grid
% Toplam Karesel hata
Es=mean((Y-Ytahmin).^2);
Em=mean(abs(Y-Ytahmin));
Ep=mean(abs(Y-Ytahmin)./Y);
Yort=mean(Y);
R2_Score=1-sum((Y-Ytahmin).^2/(Y-Yort).^2)
fprintf('Ortalama Karesel Hata E = %f \n',Es);
fprintf('Ortalama Mutlak Hata E = %f \n',Em);
fprintf('Ortalama Mutlak Yüzde Hata E = %f \n',Ep);
fprintf('R2score = %f \n',R2_Score);

figure(3)
plot(U,(Y-Ytahmin).^2,'LineWidth',2)
xlabel('U')
ylabel('Eğitim Hatası e=(d-y)^2')
grid
```

Örnekte verilen gürültüsüz veri hesaplama (Kodda Gurultu=0 için gürültü eklenmez)

U = 1   2   3   4

Y= 5   7   9   11

m = 2.000000e+00

c = 3.000000e+00

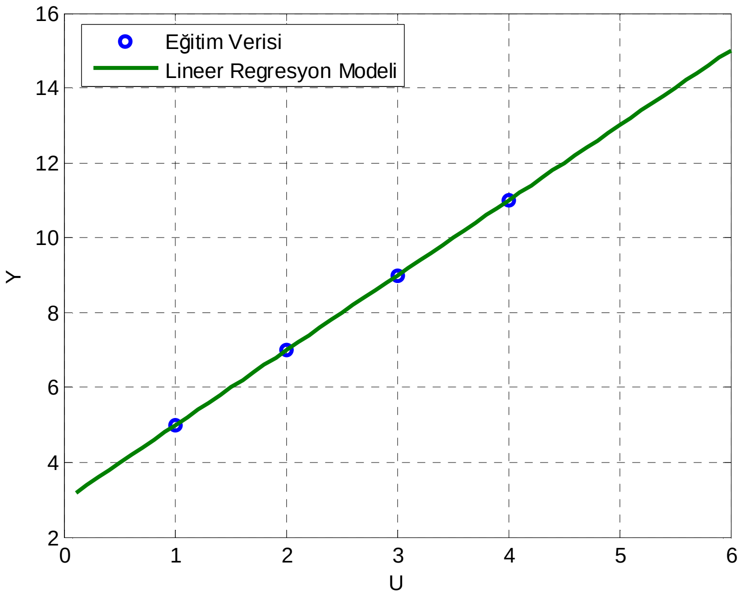
Ortalama Karesel Hata E = 0.000000

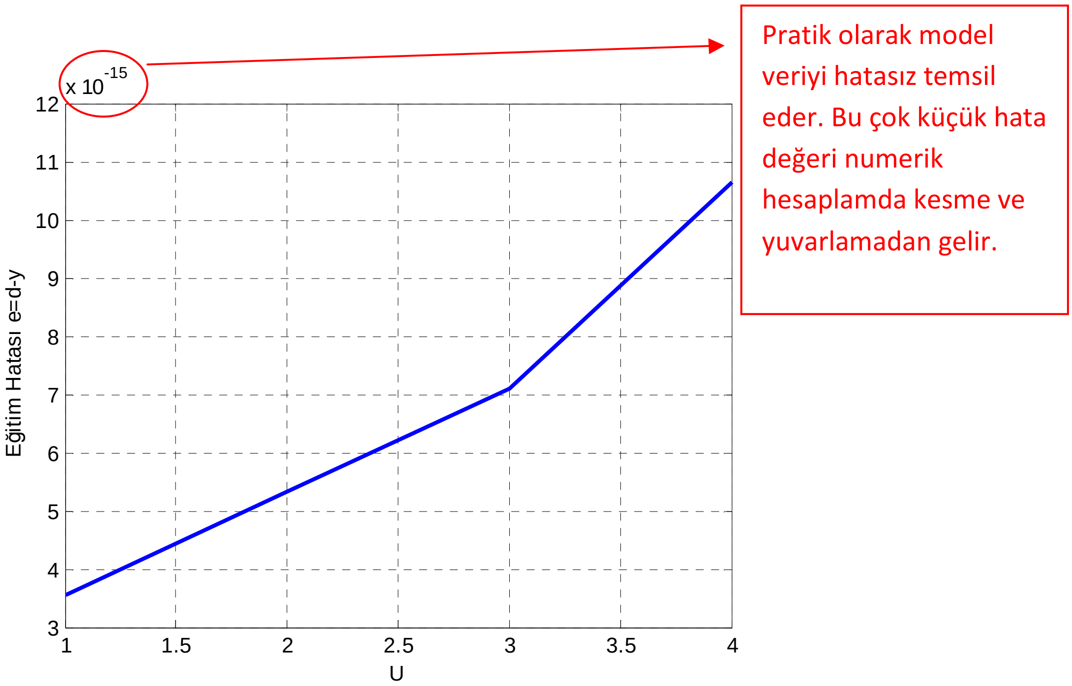
Ortalama Mutlak Hata E = 0.000000

Ortalama Mutlak Yüzde Hata E = 0.000000

R2score = 1.000000

Regresyon modeli veri kümesini hatasız temsil edebiliyor. Veri noktaları modeli temsil eden doğru üzerindedir. Bu hatanın sıfır olmasını ifade eder.





Örnek verisine gürültü eklenirse (Kodda Gurultu=1 gürültü eklenir. Aralığı [-1,1])

U = 1   2   3   4

Y= 4.5521   7.3594   9.3102   10.3252

m = 1.903430e+00

c = 3.621863e+00

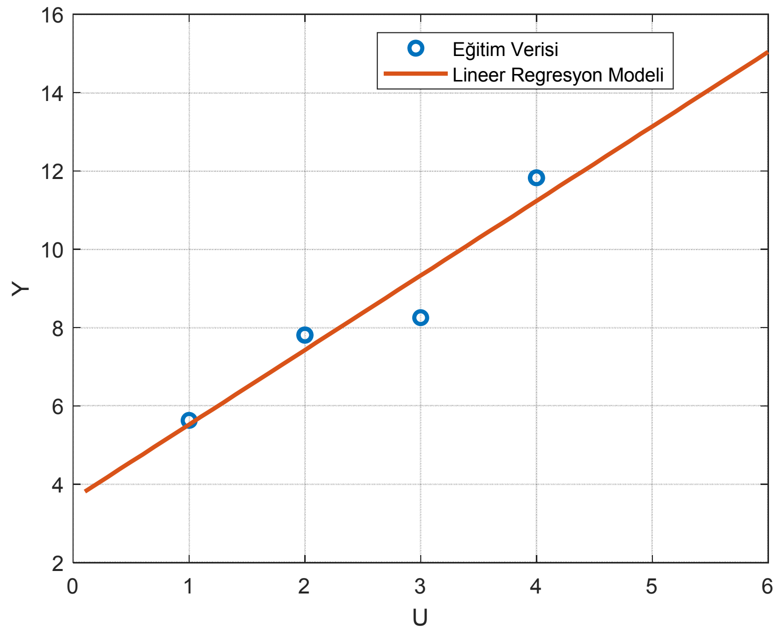
Ortalama Karesel Hata E = 0.417345

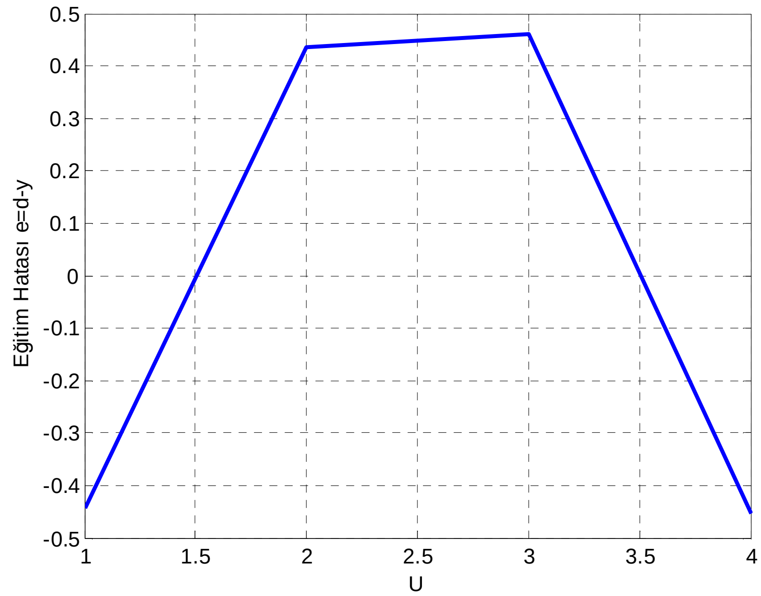
Ortalama Mutlak Hata E = 0.539090

Ortalama Mutlak Yüzde Hata E = 0.062031

R2score = 0.978337

Eklenen gürültü nedeni ile veri noktaları model üzerinde çıkmaz. Hata değerleri gürültü nedeni ile bir miktar artar. Bu nedenle R2-score biraz %100 ün altına düşer ve %97.8 civarına düşür. Burada model verilerden bir genelleme yapmıştır. Veriler matematiksel ilişkiyi temsil edebilmektedir.





Eklenen gürültü nedeni ile veri noktaları model üzerinde çıkmaz. Hata değerleri gürültü nedeni ile artar. Burada model verilerden bir genelleme yapar.

```python
# İnönü Unv. Bilgisayar Mühendisliği
# Makine Öğrenmesi Dersi Bahar 2023
# Lineer regresyon uygulama
# x: Giriş verisi, y: Çıkışı verisi
# Model Y=m*x+c En küçük kareler yöntemi ile m ve c hesaplanır.

import numpy as np
from numpy import random
#Lineer regresyon örneği
#Burada sklearn makine öğrenmesi kütüpanesinden LineerRegresyon kullanılıyor. Son
uçların grafik gösterimi için matplotlib
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

u = np.array([1, 2, 3, 4]).reshape((-1, 1))
y_org = np.array([5, 7, 9, 11])
# p=1 ise [[-1,1] aralığında gürültü (residual) eklendi
p=1
if p==1:
  n = 2*((random.randint(100, size=len(y_org))/100)-0.5);
# Ölçülen işaret modeli.(Gürültülü kanal uygulanıyor)
y=y_org+n;
print ("Giriş:",u)
print ("Sistem Çıkışı:",y_org)
print ("Gürültü:",n)
print ("Ölçülen Çıkış (y_org+n):",y)

model = LinearRegression()

model.fit(u, y)

#r_sq = model.score(x, y)
#print('R^2 score',r_sq)
print(f"c: {model.intercept_}")
print(f"m: {model.coef_}")

x_yeni= np.array([1.5, 2.2, 3.7, 4.2]).reshape(-1,1)
y_pred = model.predict(x_yeni)
print('Tahmin',y_pred)

y_predx = model.predict(u)
plt.plot(u,y,'o g')
plt.plot(u,y_org,'o r')
plt.plot(x_yeni,y_pred ,'o:b')
plt.ylim(ymin=0, ymax=18)
plt.xlim(xmin=0, xmax=6)
plt.xlabel("u")
plt.ylabel ("y")
plt.legend(['Ölçülen', 'orijinal', 'model', 'tahmin'])
plt.grid()
plt.show()

E_org=y_org-y_predx
E=y-y_predx
E_sq=(1/len(E_org))*sum(E_org*E_org)
E_abs=(1/len(E_org))*sum(abs(E_org))
E_mape=(1/len(E_org))*sum(abs(E_org)/abs(y_org))
r_squared = model.score(u,y)
print('****** Regresyon Performas Öçütleri*******')
print('Ortalama karesel hata',E_sq)
print('Ortalama mutlak hata',E_abs)
print('Ortalama mutlak yüzde hata',E_mape)
print('R^2-score',r_squared)

plt.plot(u,E_org,'o-g')
plt.plot(u,E,'o-r')
plt.ylim(ymin=-2, ymax=2)
plt.xlim(xmin=0, xmax=6)
plt.xlabel("x")
plt.ylabel ("E")
plt.legend(['Orijinal veriye göre hata', 'Ölçülen veriye göre hata'])
plt.grid()
plt.show()
```

Giriş: [[1]

[2]

[3]

[4]]

Sistem Çıkışı: [5 7 9 11]

Gürültü: [-0.14 -0.24 0.38 -0.92]

Ölçülen Çıkış (y_org+n): [4.86 6.76 9.38 10.08]

c: 3.2

m: [1.828]

Tahmin [5.942   7.2216 9.9636 10.8776]

****** Regresyon Performas Öçütleri*******

Ortalama karesel hata 0.08987999999999974

Ortalama mutlak hata 0.24399999999999977

Ortalama mutlak yüzde hata 0.026411544011544

R^2-score 0.9593210996532006