Çoklu Doğrusal Bağlanım

Hesaplamalı Matematik I - Özgür Martin

Doğrusal Bağlanım (Linear Regression)

Güdümlü öğrenmeye basit bir yaklaşım

reklam.xlsx

TV	radyo	gazete	satış
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
120.2	19.6	11.6	13.2

- Reklam bütçesi ile satış değerleri arasında ilişki var mı?
- Reklam bütçesi ile satış değerleri arasında ilişki ne kadar güçlü?
- Satışa hangi medyalar etki ediyor?
- Her bir medyanın satışa etkisini ne kadar hassas tahmin edebiliriz?
- Gelecekteki satışları ne kadar hassas kestirebiliriz?
- Değerler arasındaki ilişki doğrusal mı?

Basit Doğrusal Bağlanım

Hatırlatma: $Y = f(X) + \epsilon \xrightarrow{\text{yaklaşık?}} \hat{Y} = \hat{f}(X)$

$$Y \approx \hat{f}(X)$$

Basit Bağlanım

$$Y \approx \beta_0 + \beta_1 X$$

$$\text{satış} \approx \beta_0 + \beta_1 \text{TV}$$

Çoklu Doğrusal Bağlanım

$$Y = f(X) + \epsilon \xrightarrow{\text{yaklaşık?}} \hat{Y} = \hat{f}(X)$$

$$Y \approx \hat{f}(X)$$

Çoklu Bağlanım (Multiple Regression)

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_1 X_2 + \cdots + \beta_p X_p$$

$$\text{satış} \approx \beta_0 + \beta_1 \text{TV} + \beta_2 \text{radyo} + \cdots + \beta_p \text{gazete}$$

Çoklu Bağlanım

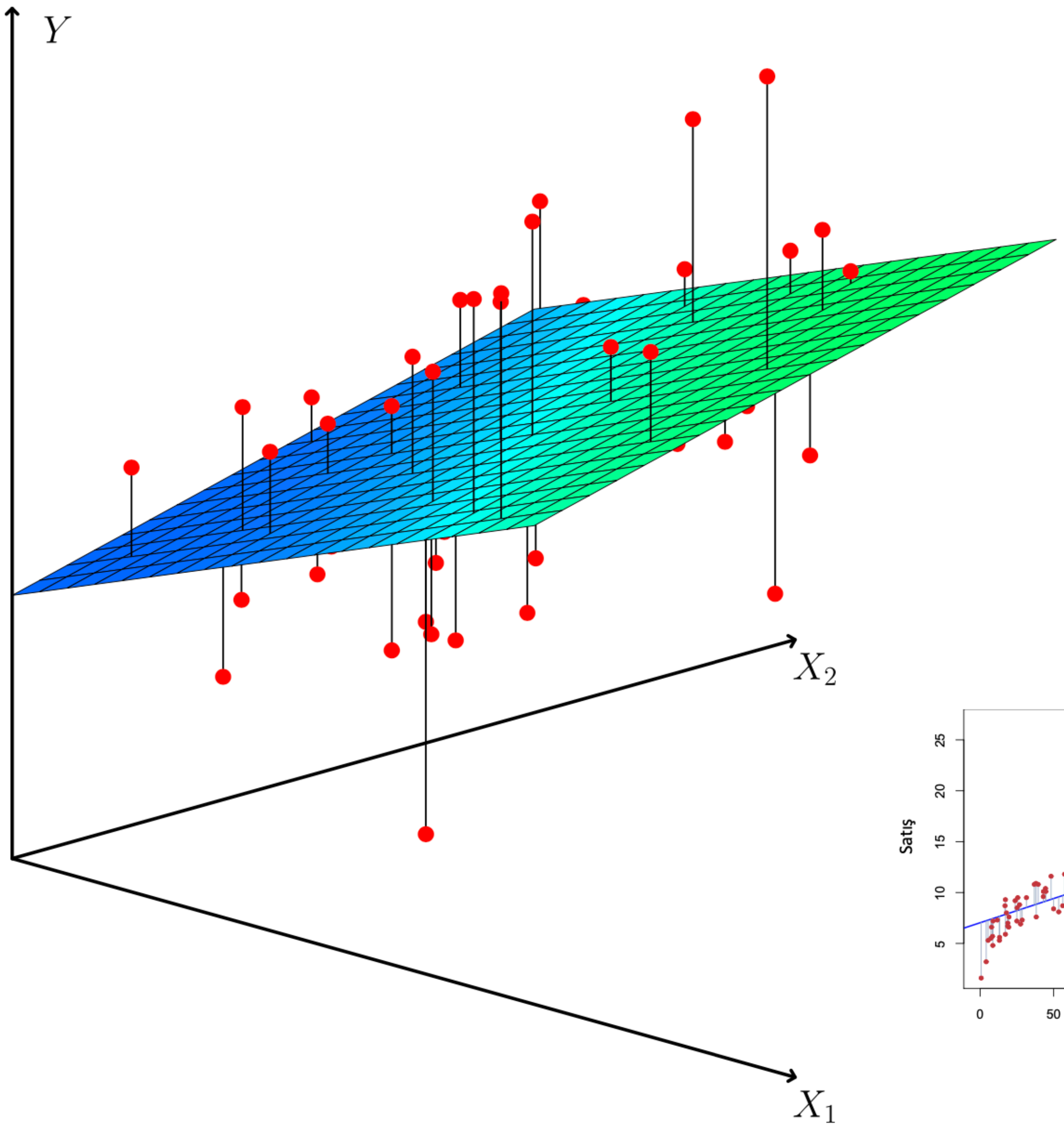
$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

eğitim verisi

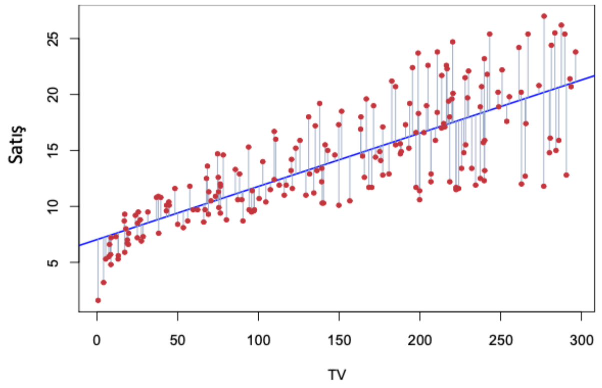
$$\{(x_i, y_i) : 1, \ldots, n\}$$

$$y_i \approx \underbrace{\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_p x_{ip}}_{\hat{y}_i}, \quad i = 1, \ldots, n$$

$$\mathrm{KKT} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$



Y
X_2
X_1



Satış
TV
0
50
100
150
200
250
300
5
10
15
20
25

EKK

$$\min_{\beta_0,\beta_1,\ldots,\beta_p} \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

$\downarrow$

Dışbükey Fonksiyon

$\downarrow$

$$\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$$

$$\boxed{\min_{\beta_0,\beta_1,\ldots,\beta_p} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2}$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^\intercal \\ 1 & x_2^\intercal \\ \vdots & \vdots \\ 1 & x_n^\intercal \end{bmatrix}_{n\times(p+1)} \qquad \mathbf{y}^\intercal = (y_1, \ldots, y_n) \qquad \boldsymbol{\beta}^\intercal = (\beta_0, \beta_1, \ldots, \beta_p)$$

$$\hat{\boldsymbol{\beta}}_{\mathrm{LS}} = \arg\min_{\boldsymbol{\beta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\intercal (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

(tam kerte (full rank) varsayımı ile)

$$\hat{\boldsymbol{\beta}}_{\mathrm{LS}} = (\mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{y}$$

$$\boxed{\hat{y}_0 = (1 \ x_0^\intercal)\hat{\boldsymbol{\beta}}_{\mathrm{LS}}}$$

Çoklu Doğrusal Bağlanım Örneği

TV	radyo	gazete	satış
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
120.2	19.6	11.6	13.2

$$\text{satış} \approx \beta_0 + \beta_1 \text{TV} + \beta_2 \text{radyo} + \beta_3 \text{gazete}$$

Çoklu Doğrusal Bağlanım Örneği

```
[41] x = np.array(reklam_verisi[['TV', 'radyo', 'gazete']])
     y = np.array(reklam_verisi['satış'])

     regr = linear_model.LinearRegression()

     regr.fit(x, y)

     y_tahmin = regr.predict(x)

     print('y =', regr.intercept_, ' + ', regr.coef_[0], '* TV', ' + ', regr.coef_[1], '* radyo', ' + ', regr.coef_[2], '* gazete')
     print('Ortalama kare hata: %.5f' % mean_squared_error(y, y_tahmin))
     print('R-kare: %.5f' % r2_score(y, y_tahmin))

     y = 2.938889369459412  +  0.0457646454553976 * TV  +  0.18853001691820448 * radyo  +  -0.0010374930424763285 * gazete
     Ortalama kare hata: 2.78413
     R-kare: 0.89721
```

Ayrıntılı tartışma Colab dosyasında!

Polinomsal Bağlanım

Bu yalnızca **Çoklu Doğrusal Bağlanım**:

$$Y = \beta_0 + \beta_1 X + \beta_1 X^2 + \cdots + \beta_n X^n + \epsilon,$$

burada $X_1 = X,\ X_2 = X^2, \ldots, X_n = X^n.$

Polinomsal Bağlanım

```
poli_bag_verisi = pd.read_excel('/content/drive/MyDrive/Colab Data/polinomsal_baglanim.xlsx')
poli_bag_verisi.head()
```

	x	y
0	0.610240	0.229201
1	0.844448	0.404077
2	1.201627	1.024286
3	0.406985	-0.136031
4	0.341956	0.202309

```
poli_bag_verisi['x2'] = poli_bag_verisi['x'] ** 2
poli_bag_verisi.head()
```

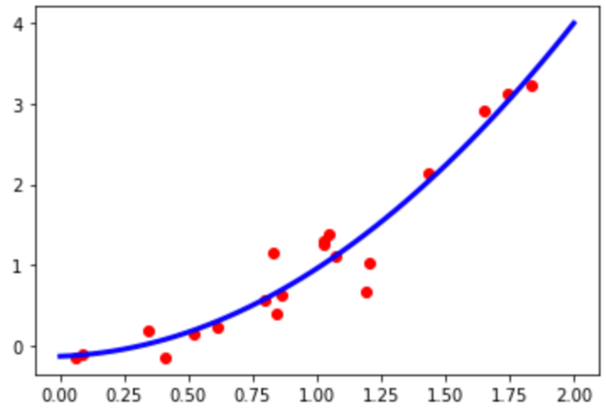
	x	y	x2
0	0.610240	0.229201	0.372393
1	0.844448	0.404077	0.713092
2	1.201627	1.024286	1.443907
3	0.406985	-0.136031	0.165637
4	0.341956	0.202309	0.116934

Polinomsal Bağlanım

```
[53] x = np.array(poli_bag_verisi[['x', 'x2']])
     y = np.array(poli_bag_verisi['y'])

     regr = linear_model.LinearRegression()

     regr.fit(x, y)

     print('y =', regr.intercept_, ' + ', regr.coef_[0], '* x', ' + ', regr.coef_[1], '* x^2')

     y = -0.12345925515434675  +  0.1132762963812204 * x  +  0.9722488772164816 * x^2
```

Polinomsal Bağlanım

```python
def f(x, regr):
  return regr.intercept_ + regr.coef_[0] * x + regr.coef_[1] * x**2

x = np.array(poli_bag_verisi['x'])
y = np.array(poli_bag_verisi['y'])

plt.scatter(x, y,  color='red')

x_ekseni = np.linspace(0,2,50)

y_ekseni = []
for i in range(len(x_ekseni)):
    y_ekseni.append(f(x_ekseni[i], regr))

plt.plot(x_ekseni,y_ekseni, color='blue', linewidth=3)

plt.show()
```



Daha Kolay Bir Yol

```python
plt.plot(x, y, 'o')

a, b, c = np.polyfit(x, y, 2)

x_2 = np.linspace(0,2,50)
y_2 = []
for i in range(len(x_)):
    y_2.append(a * x_2[i]**2 + b * x_2[i] + c)

plt.plot(x_2, y_2)
```

```
[<matplotlib.lines.Line2D at 0x7f7048ba7828>]
```