Doğrusal Regresyon (Linear Regresyon)

- **Amaç,** bağımlı ve bağımsız değişken/değişkenler arasındaki ilişkiyi doğrusal olarak modellemektir.
- Tahmin etme işlemini, doğrusal olarak yapılmasını sağlar.

$$\hat{y}_i = b + wx_i$$

$$\hat{y}_i = b + w_1x_1 + w_2x_2 + w_3x_3 + \ ... + w_px_p$$

$\hat{y}_i$ **:** Bağımlı değişkeni temsil eden, bağımlı değişkenin tahmin değeridir.

x_i **:** Bağımsız değişkenlerdir. (i = 1, 2, ...)

b **:** Bias - Beta - Sabit.

w **:** Ağırlıklar, weight - coefficientdır.

- Ağırlıklara göre değişkenlerin etkileri biçimlendirilebiliyor. Peki bu ağırlıklar neye göre bulunur ?

Ağırlıkların Bulunması

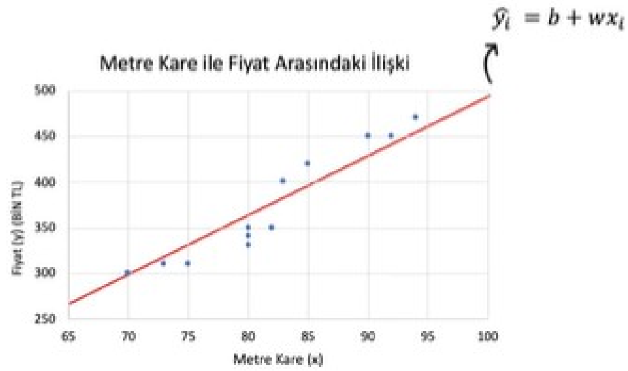
Gerçek değerler ile tahmin edilen değerler arasındaki farkların karelerinin toplamını/ortalamasını minimum yapabilecek b ve w değerlerini bularak.

$$\hat{y}_i = b + w\,x_i$$

Aşağıdaki problem için, evin metre karesi ile fiyatı arasındaki ilişki doğrusal bir şekilde ifade edilebilirse, herhangi bir metre kare bilgisine göre evin fiyatını tahmin edilebilir.

Sağdaki grafikteki kırmızı çizgi modeli temsil etmektedir.

metre_kare (x_i)	fiyat (y_i)
70	300
73	310
75	310
80	330
80	340
80	350
82	350
83	400
85	420
90	450
92	450
94	470



- b ve w'nun bulunması yani kırmızı çizginin nereden geçeceğine karar vermek çok önemlidir. Bunu doğru bir şekilde yapabilmenin yolu Cost fonksiyonunu hesaplamaktan geçmektedir. Cost fonksiyonu MSE' ye karşılık gelmektedir. Ortalamadaki 2 değeri optimizasyondan gelmektedir.
- Her seferinde ağırlıklar ve sabitler iteratif bir şekilde değiştirilerek, hata minimize edilerek optimum doğruya karar verilir.
- MSE iki şekilde değerlendirilir. Birincisi regresyon problemleri için MSE, ikincisi ise optimizasyon problemleri için MSE dir.

$$Cost(b,w) = \frac{1}{2m}\sum_{i=1}^{m}((\underbrace{b + wx_i}_{\hat{y}_i}) - y_i)^2$$

Regresyon Modellerinde Başarı Değerlendirme

Regresyon modellerinde başarı değerlendirme metrikleri MSE, RMSE ve MAE dir.

fiyat (y_i)	fiyat tahmin ($\hat{y}_i$)	hata ($e_i = y_i - \hat{y}_i$)	hata_kare ($e_i = y_i - \hat{y}_i$)^2
300	320	-20	400
310	280	30	900
310	290	20	400
330	329	1	1
340	330	10	100
350	352	-2	4
350	400	-50	2500
400	410	-10	100
420	460	-40	1600
450	420	30	900
450	410	40	1600
470	480	-10	100

$$MSE = \frac{1}{\underbrace{\boxed{n}}_{\text{Gözlem Sayısı}}} \sum_{i=1}^{n} (\overbrace{\boxed{y_i}}^{\text{Gerçek Değerler}} - \overbrace{\boxed{\hat{y}_i}}^{\text{Tahmin Edilen Değerler}})^2$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

- **MSE (Mean Square Error- Hata Kareler Ortalaması):** Regresyon problemlerinde bir başarı değerlendirme ölçüsüdür.
- MSE hesabında gerçek değer ile tahmin edilen değer arasındaki fark sonucu negatiflik/pozitiflik bir **ölçüm problemi** ortaya çıkmaktadır.
- Bunu ortadan kaldırmak için hatanın kareleri alınır.
- Kare alındığı için bu sefer ölçümler yüksek değer çıkmaktadır.

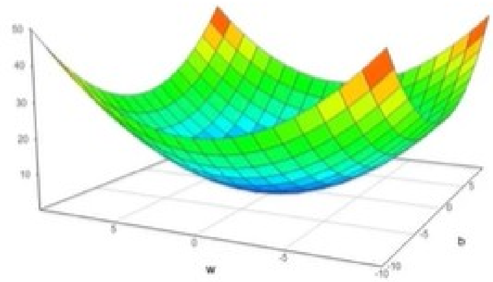
→ Yukarıdaki tablo için MSE hesaplandığında 717,08 çıkmaktadır. Peki bu ne anlama gelir ? Bir tahmin işlemi yapıldığında, yapılan ortalama hatadır. Yukarıdaki problemde bağımlı değişken ortalaması 350 civarındadır fakat hata 717 gibi yüksek bir değer çıkmıştır. Olması gereken hata, bu bağımlı değişkenin ortalaması civarında çıkmalıdır.

- Geri dönüşümün sağlanması için karekök işlemi yapılmalıdır. Bu işleme de **RMSE** denilmiştir.
- **MAE (Mean Absolute Error- Ortalama Mutlak Hata :** Bütün gözlem birimleri gezilerek, gerçek değer (y_i) ile model aracılığıyla tahmin edilen değer ($\hat{y}_i$) arasındaki farkın mutlak değerinin ortalamasının alınması işlemidir.
- Yukarıdaki 3 başarı değerlendirme metriğini de kullanabilirsin.
- MSE, Cost fonksiyonudur, bu fonksiyon optimize edilerek en küçük hatayı veren b ve w bulunmaya çalışılır.

Parametrelerin Tahmin Edilmesi (Ağırlıkların Bulunması)

- En küçük hatayı veren **sabitin** ve **ağırlığın** bulunması işlemidir.
- **Amaç,** belirli b ve w'ya karşılık gelen hataların ne olacağını gözlemleyip, bu b ve w'nun optimum değerlerini bularak, MSE=Cost'un en küçük değerini bulmaya çalışmaktır.

$$Cost(b,w) = \frac{1}{2m}\sum_{i=1}^{m}((b + wx_i) - y_i)^2$$



- Bu ağırlıkların en optimum değerini bulmanın bir yolu **Normal Denklemler Yöntemi** (En Küçük Kareler Yöntemi) dir.
- Bir bağımlı bir bağımsız değişkenin varsa, **Simple Linear Regression** kullanılmalıdır. (SSE = Sum of Square Error)
- Multiple Linear Regression'da $\hat{\beta}$' a karşılık gelen X'ler, bağımsız değişkenler matrisidir ve Y'ler bağımlı değişken matrisidir. Formülde bulunan ters alma işlemi veri çok büyük olduğunda işleri zorlaştırmaktadır. Bu yüzden Gradient Descent Yöntemi uygulanmaktadır.

Analitik Çözüm: Normal Denklemler Yöntemi (En Küçük Kareler Yöntemi)

Simple Linear Regression

$$SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))^2$$

$$b_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

$$b_0 = \overline{y} - \hat{\beta}_1 \overline{x}$$

Multiple Linear Regression

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

$$\hat{\beta} = (X^T.X)^{-1}X^T.Y$$

- Bu ağırlıkların en optimum değerini bulmanın diğer bir yolu Gradient Descent yöntemidir.

Aşağıdaki formülizasyonda,

- $\theta_0 = b_0 = b$
- $\theta_1 = b_1 = w$

karşılık gelmektedir. Bu değerler teorik bir notasyondur.

Optimizasyon Çözümü: Gradient Descent

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}).x^{(i)}$$

Doğrusal Regresyon için Gradient Descent (Gradient Descent for Linear Regression)

- Makine öğrenmesi yönteminden bağımsız olarak gerçekleştirilen bir optimizasyon yöntemidir. Bu optimizasyon yönteminin amacı, bir fonksiyonu minimum yapabilecek parametre değerlerini bulmaktır.

$$\text{Repeat until convergence } \{ \quad \theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \quad \}$$

Gradient Descent Nasıl Çalışır ?

- $J(\theta)$ gibi herhangi türevlenebilir fonksiyonu, minimum yapabilecek parametreleri bulmak için optimize eder.
- İlgili fonksiyonun, ilgili parametreye göre kısmı türevini alır.
- Türev sonucuna göre güncelleme işlemi yapılır.
- Elde edilen türev yani Gradyan değeri ilgili fonksiyonun maksimum artış yönünü verir. Bu yönün tersine doğru α (learning rate) ile yani belirli bir şiddetle giderek, parametrenin eski değerinde değişiklik yaparak her iterasyonda hatanın azalmasını sağlar.

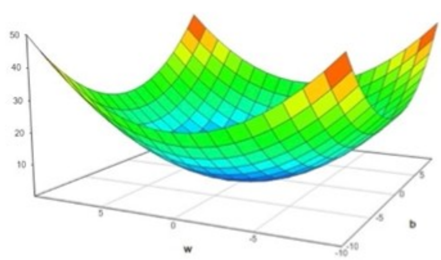
Özetle

- Gradyanın negatifi olarak tanımlanan "en dik iniş" yönünde iteratif olarak parametre değerlerini güncelleyerek ilgili fonksiyonun minimum değerini verebilecek parametreleri bulur.

Makine Öğrenmesi için Gradient Descent

- Gradient Descent bir optimizasyon yöntemidir. Fakat, makine öğrenmesinde cost fonksiyonuna uygulanınca, bu fonksiyonu minimum yapmaya çalışarak, bir parametre tahmin yöntemine dönüşür.

Cost fonksiyonunu minimize edebilecek parametreleri bulmak için kullanılır.

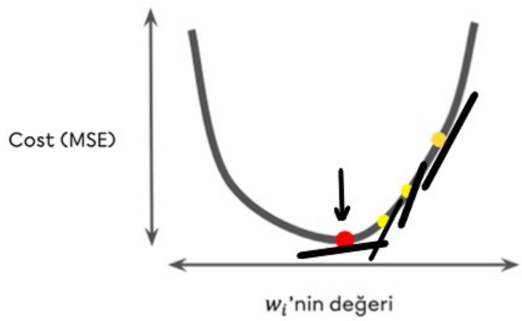


$$Cost(b, w) = \frac{1}{2m}\sum_{i=1}^{m}((b + wx_i) - y_i)^2$$

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\underset{\theta_0, \theta_1}{minimize} J(\theta_0, \theta_1)$$

- Update Rule ilk basamağında, $J(\theta_0, \theta_1)$ fonksiyonunun θ_0'a göre kısmi türevi alınmıştır.
- Update Rule ikinci basamağında, $J(\theta_0, \theta_1)$ fonksiyonunun bu sefer θ_1'e göre kısmi türevi alınmıştır.
- Bu işlemler, iteratif bir şekilde devam ederek, her defasında hatayı azaltarak kırmızı ile gösterdiğimiz lokal minimum noktasına ulaşmayı hedefler.
- Lokal minimum noktasına da belirlenen bir hızla, α (learning rate), öğrenme hızı ile gidilir.



$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\overbrace{\underbrace{h_\theta(x^{(i)})}_{\hat{y}_i}}^{\theta_0 + \theta_1 x_i} - y^{(i)})^2$$

$$\underset{\theta_0, \theta_1}{minimize} J(\theta_0, \theta_1)$$

Update Rule

$$\theta_0 \coloneqq \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

$$\theta_1 \coloneqq \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}).x^{(i)}$$