



ft\_linear\_regression

Makine öğrenimine giriş

*Özet: Bu projede, ilk makine öğrenimi algoritmanızı uygulayacaksınız.*

*Versiyon: 4*

# İçindekiler

<b>I</b>	<b>Önsöz</b>	<b>2</b>
<b>II</b>	<b>Giriş</b>	<b>3</b>
<b>III</b>	<b>Amaç</b>	<b>4</b>
<b>IV</b>	<b>Genel talimatlar</b>	<b>5</b>
<b>V</b>	<b>Zorunlu kısım</b>	<b>6</b>
<b>VI</b>	<b>Bonus bölüm</b>	<b>7</b>
<b>VII</b>	<b>Sunum ve akran değerlendirmesi</b>	<b>8</b>

# Bölüm I Önsöz

Makine öğrenimi için en iyi tanımın ne olduğunu düşünüyorum:

"Bir bilgisayar programının, T'deki görevlerde **P** ile ölçülen performansı E deneyimi ile geliyorsa, bazı T görev sınıfı ve P performans ölçütü açısından E deneyiminden öğrendiği söylenir."

Tom M. Mitchell

## Bölüm II Giriş

Makine öğrenimi, biraz karmaşık ve sadece matematikçilere ayrılmış gibi görünebilen, bilgisayar biliminin büyüyen bir alanıdır. Sinir ağlarını veya k-ortalamlar kümelemesini duymuş olabilirsiniz ancak nasıl çalıştıklarını veya bu tür algoritmaların nasıl kodlanacağını bilmiyor olabilirsiniz...

Ancak endişelenmeyin, aslında basit, temel bir makine öğrenimi algoritmasıyla başlayacağız.

## Bölüm III Amaç

Bu projenin amacı sizi makine öğreniminin arkasındaki temel kavramla tanıştırmaktır. Bu proje için, [gradyan iniş algoritması](#) ile [doğrusal bir fonksiyon](#) treni kullanarak bir arabanın fiyatını tahmin eden bir program oluşturmanız gerekecektir.

Proje için kesin bir örnek üzerinde çalışacağız, ancak işiniz bittiğinde algoritmayı başka herhangi bir veri kümesiyle kullanabileceksiniz.

## Bölüm IV

### Genel talimatlar

Bu projede istediğiniz dili kullanmakta özgürsünüz.

Ayrıca, tüm işi sizin için yapmadıkları sürece istediğiniz kütüphaneleri kullanmakta özgürsünüz. Örneğin, python'un `numpy.polyfit` kullanımı hile olarak kabul edilir.



Verilerinizi kolayca görselleştirmenize olanak tanıyan bir dil kullanmalısınız: hata ayıklama için çok yardımcı olacaktır.

# Bölüm V

## Zorunlu kısım

Tek bir özellik ile basit bir doğrusal regresyon uygulayacaksınız - bu durumda, arabanın kilometresi.

Bunu yapmak için iki program oluşturmanız gerekir :

- İlk program, belirli bir kilometre için bir arabanın fiyatını tahmin etmek için kullanılacaktır. Programı başlattığınızda, sizden bir kilometre istemeli ve ardından bu kilometre için tahmini fiyatı size geri vermelidir. Program fiyatı tahmin etmek için aşağıdaki hipotezi kullanacaktır:

$$estimatePrice(mileage) = \theta_0 + (\theta_1 * mileage)$$

Eğitim programının çalıştırılmasından önce, teta0 ve teta1 0 olarak ayarlanacaktır.

- İkinci program modelinizi eğitmek için kullanılacaktır. Veri kümesi dosyanızı okuyacak ve veriler üzerinde doğrusal bir regresyon gerçekleştirecektir. Doğrusal regresyon tamamlandığında, theta0 ve theta1 değişkenlerini ilk programda kullanmak üzere kaydedeceksiniz. Aşağıdaki formülleri kullanacaksınız:

$$tmp\theta_0 = learningRate * \frac{1}{m} \sum_{i=0}^{m-1} (estimatePrice(mileage[i]) - price[i])$$

$$tmp\theta_1 = learningRate * \frac{1}{m} \sum_{i=0}^{m-1} (estimatePrice(mileage[i]) - price[i]) * kilometre[i]$$

M'nin ne olduğunu tahmin etmenize izin veriyorum :)

estimatePrice'in ilk programımızdakiyle aynı olduğuna dikkat edin, ancak burada geçici, son olarak hesaplanan theta0 ve theta1'i kullanır.

Ayrıca, teta0 ve teta1'i aynı anda güncellemeyi unutmayın.

# Bölüm VI Bonus

## kısmı

İşte çok faydalı olabilecek bazı bonuslar:

- Dağılımlarını görmek için verilerin bir grafiğe çizilmesi.
- Sıkı çalışmanızın sonucunu görmek için doğrusal regresyonunuzdan elde edilen çizgiyi aynı grafiğe çizin!
- Algoritmanızın hassasiyetini hesaplayan bir program.



Bonus kısım sadece zorunlu kısım MÜKEMMEL ise değerlendirilecektir. Mükemmel, zorunlu kısmın bütünsel olarak yapıldığı ve hatasız çalıştığı anlamına gelir. TÜM zorunlu gereklilikleri geçmediyseniz, bonus bölümünüz hiç değerlendirilmeyecektir.



## **Bölüm VII**

### **Sunum ve akran değerlendirme**

Ödevinizi her zamanki gibi Git deponuzda teslim edin. Savunma sırasında yalnızca deponuzdaki çalışmalar değerlendirilecektir. Doğru olduklarından emin olmak için klasörlerinizin ve dosyalarınızın adlarını iki kez kontrol etmekten çekinmeyin.

İşte akran-düzeltilmenizin kontrol etmesi gereken noktalar:

- İşi sizin için yapan kütüphanelerin yokluğu
- Belirtilen hipotezin kullanımı
- Belirtilen eğitim fonksiyonunun kullanımı