Makine Öğrenimi: Basit Doğrusal Regresyon Python

Lineer ne demek?

İlk olarak, Walmart'ta alışveriş yaptığınızı varsayalım. Mal alıp almadığınızdan bağımsız olarak, park etme ücreti olarak 2.00 dolar ödemeniz gerekmektedir. Her elma fiyatı 1.5 dolar ve x adet elma almanız gerekmektedir. Sonra aşağıdaki gibi bir fiyat listesi oluşturabiliriz.

Quantity	Price
1	$ 3.50
2	$ 5.00
3	$ 6.50
4	$ 8.00
5	$ 9.50
...	...
10	$ 17.00
11	$ 18.50
...	...
x	y

Bu örnekte y=2+1.5x denklemi kullanılarak Değer'e (x) dayalı olarak Fiyat'ı tahmin etmek veya hesaplamak oldukça kolaydır. Ve bunun tersini de yapabilirsiniz.

$$y = a + bx$$

a=2
b=1.5

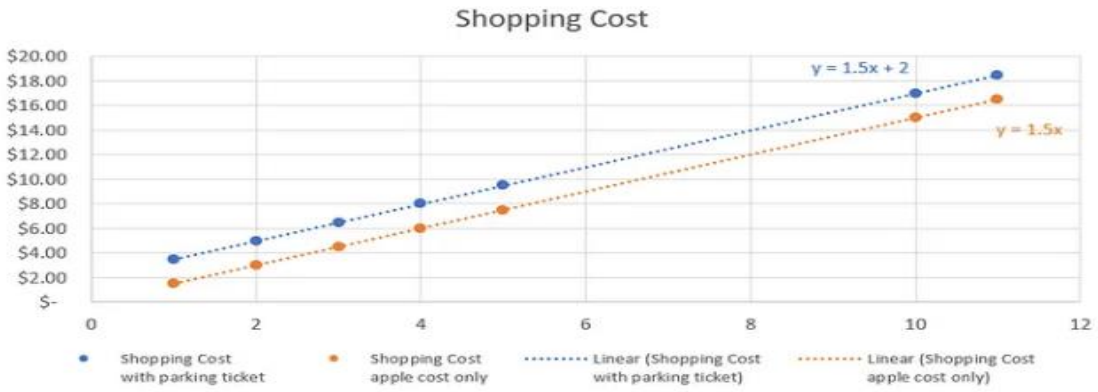
Doğrusal bir fonksiyonun bir bağımsız değişkeni ve bir bağımlı değişkeni vardır. Bağımsız değişken x'tir ve bağımlı değişken y'dir.

a, sabit terim veya y-kesişimidir. x = 0 olduğunda bağımlı değişkenin değeridir. b, bağımsız değişkenin katsayısıdır. Eğim olarak da bilinir ve bağımlı değişkenin değişim hızını verir. Neden ona "doğrusal" diyoruz? Peki, yukarıdaki veri setini görselleştirelim!

Shopping cost (y = 1.5x + 2)

Alışveriş maliyetinin tüm değerlerini çizdikten sonra (mavi çizgi), hepsinin bir çizgi üzerinde olduğunu görebilirsiniz, işte bu yüzden ona doğrusal diyoruz. Doğrusal denklemin (y=a+bx) açıkladığı gibi, a bağımsız bir değişkendir. Eğer a=0 olsa bile (park ücreti ödemeniz gerekmez), Alışveriş Maliyeti çizgisi aşağı kayacak ve hala bir çizgi üzerinde olacak (turuncu çizgi).

Quantity	Shopping Cost with parking ticket	Shopping Cost apple cost only
1	$ 3.50	$ 1.50
2	$ 5.00	$ 3.00
3	$ 6.50	$ 4.50
4	$ 8.00	$ 6.00
5	$ 9.50	$ 7.50
10	$ 17.00	$ 15.00
11	$ 18.50	$ 16.50



Shopping cost (y = 1.5x)

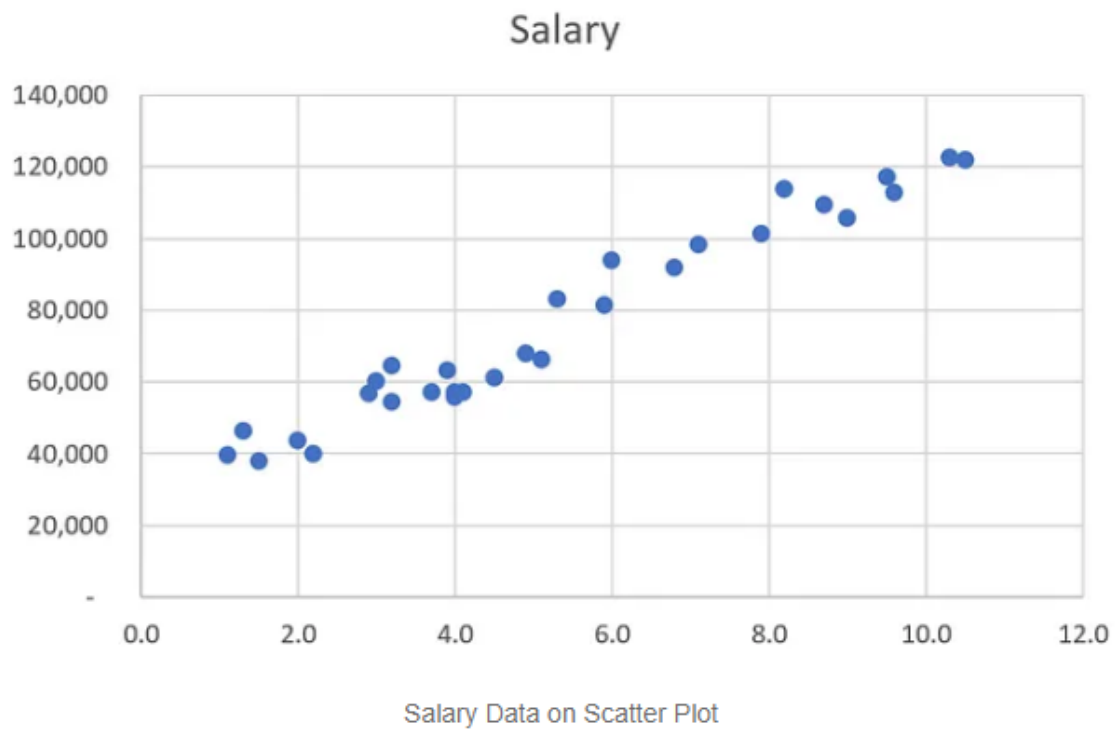
Ancak gerçek hayatta işler o kadar da basit değil! Başka bir örnek verelim, AB Şirketi'nde, Deneyim Yılı'na dayalı olarak aşağıdaki gibi bir maaş dağıtım tablosu bulunmaktadır:

YearsExperience	Salary
1.1	39,343
1.3	46,205
1.5	37,731
2.0	43,525
2.2	39,891
2.9	56,642
3.0	60,150
3.2	54,445
3.2	64,445
3.7	57,189
3.9	63,218
4.0	55,794
4.0	56,957
4.1	57,081
4.5	61,111
4.9	67,938
5.1	66,029
5.3	83,088
5.9	81,363
6.0	93,940
6.8	91,738
7.1	98,273
7.9	101,302
8.2	113,812
8.7	109,431
9.0	105,582
9.5	116,969
9.6	112,635
10.3	122,391
10.5	121,872

Salary based on Years of Experience (salary_data.csv)

"Senaryo şu ki, bir HR görevlisisiniz ve 5 yıllık deneyime sahip bir adayınız var. O zaman ona ne kadarlık bir maaş teklif etmelisiniz?”

Bu soruna derinlemesine inmeden önce, önce veri setini bir grafik üzerine çizelim:



Salary Data on Scatter Plot

Lütfen bu grafiğe dikkatlice bakın. Şimdi kötü bir haberimiz var: tüm gözlemler bir çizgi üzerinde değil. Bu, (y) değerini hesaplamak için denklemi bulamayacağımız anlamına gelir.

Şimdi ne yapmalı? Endişelenmeyin, size iyi bir haberimiz var!

Tekrar Saçılım Grafiğine bakın, aşağı kaydırmadan önce. Görüyor musunuz?

Tüm noktalar bir çizgi üzerinde değil, ANCAK bir çizgi şeklinde! Bu doğrusal!

Salary Data on Scatter Plot with linear shape

Gözlemimize dayanarak, 5 Yıl Deneyime sahip bir aday için maaş aralığının kırmızı aralıkta olması gerektiğini tahmin edebiliriz. Tabii ki, adayımıza kırmızı aralıkta herhangi bir sayı teklif edebiliriz. Ancak en iyi sayıyı nasıl seçeceğiz? Adayımız için en iyi maaşı tahmin etmek için Makine Öğrenimi kullanma zamanı geldi.

Bu bölümde, adayımız için en iyi maaşı bulmak için Python'u Spyder IDE üzerinde kullanacağız. Tamam, hadi yapalım!

Python ile Doğrusal Regresyon

Devam etmeden önce, Makine Öğrenimi'nin 2 temel adımını aşağıdaki gibi özetliyoruz:

1-Eğitim

2-Tahmin

Tamam, veri setiyle çalışmak için numpy ve pandas gibi 4 kütüphaneyi, makine öğrenimi işlevlerini uygulamak için sklearn'i ve görselleştirmelerimizi incelemek için matplotlib'i kullanacağız.

Kod açıklaması:

veri seti: csv dosyamızdaki tüm değerleri içeren tablo

x: Deneyim Yılı dizisini içeren ilk sütun

y: Maaş dizisini içeren son sütun

Sonraki adım olarak, veri setimizi (toplam 30 gözlem) iki sete bölmeliyiz: eğitim seti, eğitim için kullanılan ve test seti, test için kullanılan:

Kod açıklaması:

test_size=1/3: Veri setimizi (30 gözlem) iki parçaya böleceğiz (eğitim seti, test seti) ve test setinin, veri setine oranı 1/3 olacak (10 gözlem test setine yerleştirilecek). Oranı 1/2 veya 0.5 olarak da belirtebilirsiniz, bu aynıdır. Test setini çok büyük yapmamalıyız; eğer çok büyük olursa, eğitmek için yetersiz veriye sahip oluruz. Normalde, yaklaşık %5 ila %30 arasında bir oran seçmeliyiz.

train_size: Eğer zaten test_size kullanıyorsak, geri kalan veri otomatik olarak train_size'a atanacaktır.

random_state: Bu, rastgele sayı üreteci için bir tohumdur. Ayrıca RandomState sınıfından bir örneği de kullanabiliriz. Eğer boş bırakırsak veya 0 girersek, np.random tarafından kullanılan RandomState örneği kullanılacaktır.

Artık eğitim setimiz ve test setimiz var, şimdi Regresyon Modeli'ni oluşturmalıyız.

Kod açıklaması:

regressor = LinearRegression(): Doğrusal Regresyon'u uygulayacak olan eğitim modelimiz.

regressor.fit: Bu satırda, modeli oluşturmak için yıl deneyimi değerlerini içeren **x_train'i** ve belirli bir maaş değerlerini içeren **y_train'i** geçiriyoruz. Bu eğitim sürecidir.

Eğitim modelimizi ve test modelimizi görselleştirelim.

Yukarıdaki kodu çalıştırdıktan sonra, konsol penceresinde 2 grafik göreceksiniz.

Training Set

Test Set

İki grafiği karşılaştırınca, 2 mavi çizginin aynı yönde olduğunu görebiliriz. Modelimiz artık kullanıma hazır.

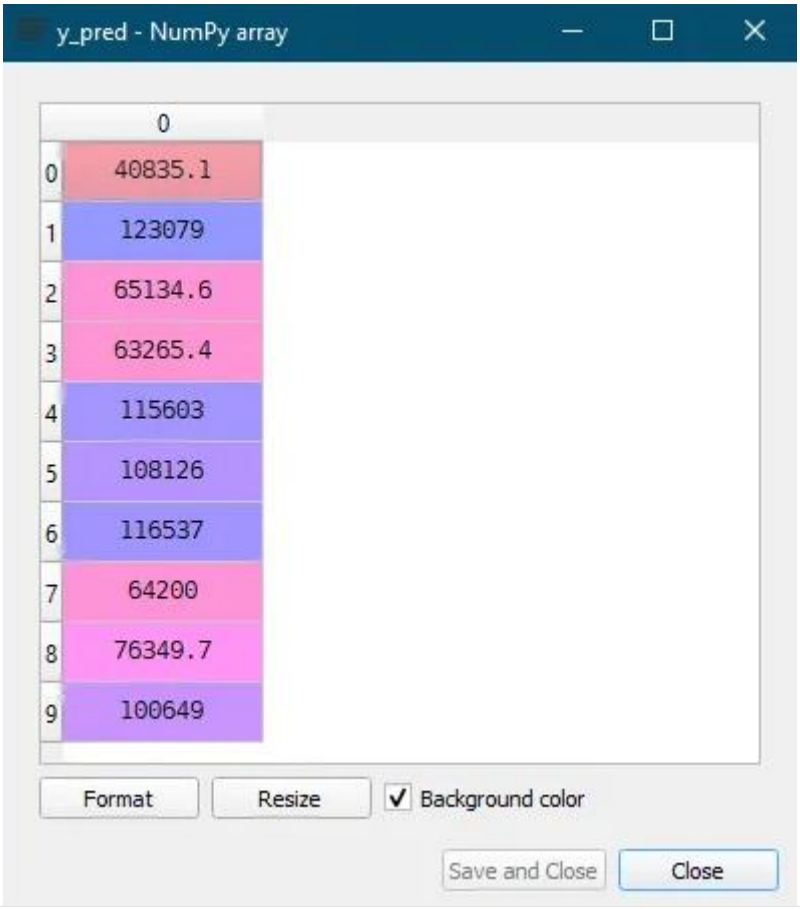
Tamam! Artık modelimiz var, şimdi onu X'e bağlı olarak y'nin herhangi bir değerini veya y'ye bağlı olarak X'in herhangi bir değerini hesaplamak (tahmin etmek) için kullanabiliriz. Bunu yapmanın yolu şöyle:

```
In [2]: y_pred = regressor.predict(5)

In [3]: print(y_pred)
[73545.90445964]

In [4]:
```

Bingo! X = 5 (5 Yıl Deneyim) için y_pred değeri 73545.90'dır.

Adayınıza 73.545,90 dolarlık bir maaş teklif edebilirsiniz ve bu onun için en iyi maaş!

Ayrıca, tek bir X değeri yerine X'in bir dizisini de geçirebiliriz:



Ve y'yi kullanarak X'i tahmin edebiliriz. Kendiniz deneyin!

Basit Doğrusal Regresyon ile aşağıdaki gibi 5 adımı izlememiz gerekiyor:

1. Veri setini içe aktarma.

2. Veri setini eğitim seti ve test setine ayırma (her bir set için X ve y'nin 2 boyutu). Genellikle, test seti veri setinin %5 ila %30'unu oluşturmalıdır.

3. Eğitim setini ve test setini görselleştirerek kontrol etmek (isterseniz bu adımı atlayabilirsiniz).

4. Regresyon modelini başlatma ve eğitim setini kullanarak uyarlama (hem X'i hem de y'yi).

5. Hadi tahmin yapalım!!

Kaynak: https://gist.github.com/panicpotatoe/14d9c7a5a25566a58143e7069813629f