

Makine Öğrenmesine Giriş (Introduction to Machine Learning)

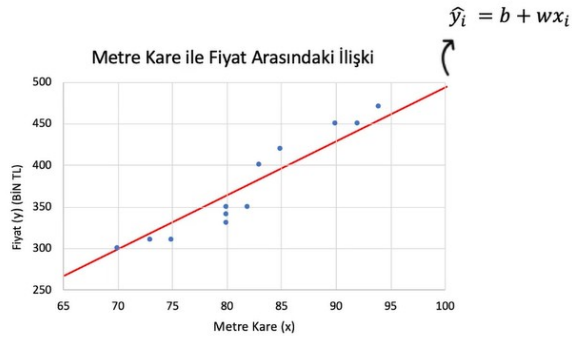
- Bilgisayarların insanlara benzer şekilde öğrenmesini sağlamak maksadıyla çeşitli algoritma ve tekniklerin geliştirilmesi için çalışılan bilimsel bir çalışma alanıdır.
- **Amaç** geçmişteki verileri kullanarak gelecek için tahminlerde bulunmaktır.
- Makineye bir problemi öğretebilmemiz için ilk önce insanlar nasıl düşünür, bir probleme nasıl yaklaşır bunu kavramamız gerekmektedir.
- İnsanlardan bir verinin yorumlanması istendiğinde veriye bakarak, verinin ilişki şeklini anlayıp, çıkarım yapmaya çalışır. Çok büyük bir veri olursa yorumlanması imkansızla yakın olurdu. Yinede bağımsız değişkenlere ilgili ağırlıklar verilerek bir ilişki yakalanabilir ve tahmin yapılabilir. Şimdi **amacımız**, insanların yaptığı gibi öğrenme ve tahmin etme işini makinelerle yapmaya çalışmaktır.

↪ Aşağıda sağda verilen grafik metre kare ile fiyat arasındaki ilişki bir regresyon problemidir. Bağımlı değişken yani hedeflenen değişkeni **sayısal** olan problemlere **regresyon problemleri** denir.

↪ Grafikten görüldüğü üzere metre kare arttıkça fiyatta artmaktadır.

↪ Kırmızı çizgi tahmin edilen değerlerdir. Bir yükseklik almıştır ve bir eğime sahiptir. Yükseklik = Sabit = Bias "**b**" ifadesine, doğru eğimi = ağırlık "**w**"a karşılık gelmektedir.

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



↪ Bağımlı değişkeni yani hedeflenen değişkeni **kategorik** olan problemlere **sınıflandırma problemi** denir.

titanic

Survived	Pclass	Sex	Age
0	3	male	22
1	1	female	38
1	2	female	26
1	3	female	12
1	1	male	35
1	1	male	25
0	2	male	25
0	3	male	15

▪ 45 yaşında, 3.sınıf, erkek yolcu hayatta kalabilir mi?

↪ Titanic verisetinde hedef(target) değişken yolcuların hayatta kalıp kalmamalarıdır. Yani "survived" değişkenidir. 0-1 bir temsildir. Aslında, titanic veriseti bir sınıflandırma problemidir.

↪ Görseldeki sorunun yanıtına sadece yanındaki tabloya bakarak cevap verecek olursak genel hatlarıyla kadınlar ve 1. sınıf, erkekler hayatta kalmış gibi duruyor, 3.sınıf ve erkek olma durumunda hayatta kalan yok gözüküyor. O zaman 45 yaşında 3. sınıf bir erkekte hayatta kalamaz yorumu yapılır.

↪ Peki, 15 yaşında, 1. sınıf, erkek bir yolcu hayatta kalabilir mi ? şeklinde bir soru olsaydı. Tabloda 1.sınıf erkekler hayatta kalmış. Fakat yaş bilgisini yakalayacak bir örüntü bulunamamıştır dolayısıyla ilk soru kadar kesin bir cevap verilemeyip hayatta kalmış **olabilir** denilir...

↪ Peki, 36 yaşında, 1.sınıf, kadın yolcu hayatta kalabilir mi? Tabloya bakarak hiç tereddütsüz doğrudan **evet hayatta kalır** denilir.

- Yukarıdaki çıkarımlar basit bir tablodan yola çıkarak yapıldı, ilk önce tablo analiz edildi, bilgiler öğrenildi sonra öğrenilen veriden çıkarımlar yapıldı. Peki, makine öğrenmesi bunu nasıl yapacak ? İnsan öğrenimine benzer şekilde, verinin yapısını öğrenecek bağımsız değişkenlere önemine göre ilgili ağırlıklar verilecek hedef bilgiyi doğru tahmin etmeye çalışacak.

Değişken Türleri (Variable Types)

- **Sayısal Değişkenler (Quantitative)** : Kesikli ya da sürekli olarak ifade edilen değişkenlerdir.

Kesikli : Çocuk sayısı, ceza sayısı

Sürekli : Yaş, sıcaklık, boy

Not : Başlangıç noktası **sıfır olmayan** sayısal değişkenlerin ölçek türü **aralıktır (interval)**.

Not : Başlangıç noktası **sıfır olan** (yokluk ifade eden sıfır) sayısal değişkenlerin ölçek türü **orandır (ratio)**.

- **Kategorik Değişkenler (Qualitative):**

Nominal :

Binary (ikilik) olanlar,

→ Var / Yok

→ Kadın / Erkek

→ Hasta / Sağlıklı

İkiden çok kategorili, ast/üst ilişkisi olmasın, kategori arasında sıra önemi olmasın,

→ Medeni Durum (Evli / Bekar / Dul / Boşanmış)

→ Şehirler

→ İsimler

Ordinal : Sıralı kategorik verilerdir ve ast/üst ilişkisi olmalıdır,

→ Eğitim düzeyi

→ Askeri rütbe

→ Akademik unvan

→ Sosyoekonomik ölçekler

- **Bağımlı Değişken (target, dependent, output, response) :** İlgilendiğimiz problemdeki hedef değişkene bağımlı değişken denir. Örneğin; kanser olup/olmama, hayatta kaldı/kalmadı gibi değişkenler...
- **Bağımsız Değişken (feature, independent, input, column, predictor, explanatory) :** İlgilenilen problemdeki bağımlı değişken yani targete etki ettiğini varsaydığımız diğer değişkenlerdir.

Örnek 1 : Aşağıdaki tablo için öğrendiğimiz bilgileri yorumlayalım.

advertising

TV	radio	newspaper	sales
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	75.0	7.2
57.5	32.8	23.5	11.8
120.2	19.6	11.6	13.2
8.6	2.1	1.0	4.8
199.8	2.6	21.2	10.6

- **Bağımlı değişken** : Sales değişkeni
- **Bağımsız değişkenler** : TV, radio, newspaper
- **Problem türü** : Regresyon problemi (bağımlı değişken sayısal)

Örnek 2 : Aşağıdaki tablo için öğrendiğimiz bilgileri yorumlayalım.

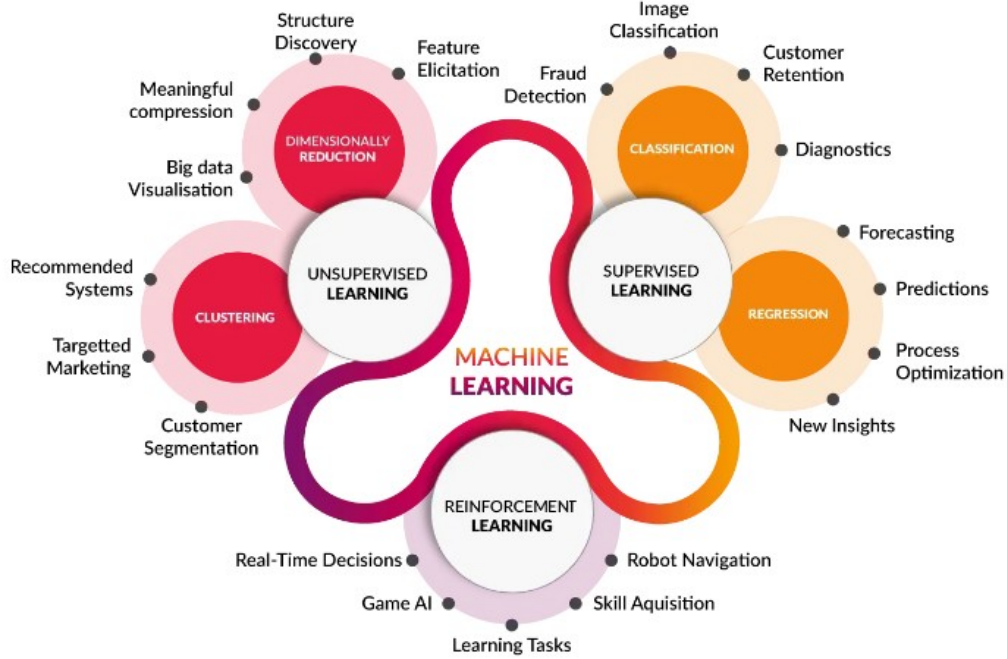
churn

Age	Total_Purchase	Years	Churn
42.0	11066.8	7.22	1
41.0	11916.22	6.5	1
38.0	12884.75	6.67	0
42.0	8010.76	6.71	1
37.0	9191.58	5.56	0
48.0	10356.02	5.12	1
44.0	11331.58	5.23	0
32.0	9885.12	6.92	0
43.0	14062.6	5.46	1
40.0	8066.94	7.11	1

- **Bağımlı değişken** : Churn
- **Bağımsız değişkenler** : Age, Total_Purchase, Years

- **Problem türü :** Sınıflandırma problemi (bağımlı değişken sayısal gibi gözükse de müşterinin terk durumunu (evet/hayır) temsil etmektedir)

Öğrenme Türleri (Learning Types)



Makine öğrenmesi temelinde 3 farklı öğrenme türü ile ele alınmaktadır:

❖ **Denetimli Öğrenme (Supervised Learning) :** Üzerinde çalışılan verilerde bir bağımlı değişken (*target*) *varsa* bu bir gözetimli öğrenme problemidir. Bağımlı ve bağımsız değişkenler arası ilişki öğreniliyor olur.

- **Regresyon Problemleri :** Bağımlı değişkenin **sayısal** olduğu problemlerdir.
- **Sınıflandırma Problemleri :** Bağımlı değişkenin **kategorik** olduğu problemlerdir.

❖ **Denetimsiz Öğrenme (Unsupervised Learning) :** İlgili veri setlerinde bağımlı değişken (*target*) *yoksa* bu bir gözetimsiz öğrenme problemidir.

- **Kümeleme (Clustering)**
- **Boyut İndirgeme (Dimensionality Reduction)**

❖ **Pekiştirmeli Öğrenme (Reinforcement Learning) :** Boş bir oda olduğunu ve içinde robot olduğunu düşünelim, robotun görevi odadan kapıyı kullanarak çıkması, bunun için **deneme-yanılma** yoluyla kapıya ulaşmaya çalışsın. Her yanlış girişimi için **ceza** alıyor olsun. Böylelikle cezalardan öğrenerek kapıya ulaşacaktır.

- Sonuç olarak: deneme yanılma yoluyla, aldığı cezalardan öğrenene öğrenme hedefe ulaşması yöntemine, **pekiştirmeli öğrenme** denir.

Model Başarı Değerlendirme Yöntemleri

Tahminlerim ne kadar başarılı ?

- Modeller tahminlerde bulunduğunda bu modellerde belirli sapmalar bekleriz. Peki bu sapmaları nasıl değerlendiririz ?
- Regresyon ve sınıflandırma problemlerinde kullanılan farklı başarı değerlendirme metrikleri vardır.

Regresyon Modellerinde Başarı Değerlendirme

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

TV	radio	newspaper	sales	predicted_sales
230.1	37.8	69.2	22.1	20
44.5	39.3	45.1	10.4	11
17.2	45.9	69.3	9.3	9
151.5	41.3	58.5	18.5	17
180.8	10.8	58.4	12.9	12
8.7	48.9	75.0	7.2	7
57.5	32.8	23.5	11.8	11
120.2	19.6	11.6	13.2	14
8.6	2.1	1.0	4.8	4.5

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

MSE (Mean Square Error- Hata Kareler Ortalaması): Regresyon problemlerinde bir başarı değerlendirme ölçüsüdür.

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

- Bütün gözlem birimleri gezilerek, gerçek değer (y_i) ile model aracılığıyla tahmin edilen değer (\hat{y}_i) arasındaki farkın kareleri toplamının ortalamasının alınması işlemidir.
- $(y_i - \hat{y}_i)$: hatayı verir.
- Optimizasyon yöntemlerinde kullanılır.
- MSE değeri ne kadar küçükse, sıfıra ne kadar yakınsa o kadar iyidir.

RMSE (Root Mean Square Error- Hata Kareler Ortalamasının Karekökü : MSE'nin karekökünün alınmış halidir.

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

- 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, 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.

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

Sınıflandırma Modellerinde Başarı Değerlendirme

- Başarılı yaptığım işler bölü bütün işler.
- Aşağıdaki sınıflandırma probleminde, gerçek değerler "Churn" ve tahmin edilen değerler "Predicted Churn" verilmiştir. 10 gözlem içinden 7 tanesini doğru tahmin ettiğim için tahmin başarıım, $\frac{7}{10}$ yani %70 dir. Bu hesaplama işlemine "**Accuracy**" denir.
- **Accuracy** = $\frac{DoğruSınıflandırmaSayısı}{ToplamSınıflandırılanGözlemSayısı}$
- Accuracy, sınıflandırma metrikleri için kullanılır ne kadar yüksekse o kadar iyidir.

churn

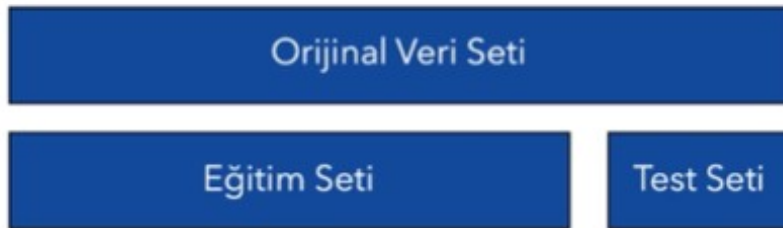
Age	Total_Purchase	Years	Churn	Predicted_Churn	
42.0	11066.8	7.22	1	1	✓
41.0	11916.22	6.5	1	0	✗
38.0	12884.75	6.67	0	0	✓
42.0	8010.76	6.71	1	0	✗
37.0	9191.58	5.56	0	1	✗
48.0	10356.02	5.12	1	1	✓
44.0	11331.58	5.23	0	0	✓
32.0	9885.12	6.92	0	0	✓
43.0	14062.6	5.46	1	1	✓
40.0	8066.94	7.11	1	1	✓

Model Doğrulama (Model Validation) Yöntemleri

- Modellerin başarısını daha doğru değerlendirme, doğrulamaya çalışma çabasıdır.
- İlk zamanlarda modeli, orijinal veri setini kullanarak başarı açısından değerlendiriyorlardı. Burada hem veri seti üzerinden model kurulup hem test etme işlemi yapıldığından, modeli aşırı öğrenme, hataları yanlış değerlendirme gibi problemlere sebep olmuştur. Bunun önüne geçmek için çeşitli yöntemler önerilmiştir.

1- Holdout Yöntemi (Sınama Seti Yöntemi)

- Veriseti, eğitim seti ve test seti olarak ikiye bölünür.
- Eğitim (train) set üzerinden modelleme işlemi gerçekleştirilir. Model burada öğrenir.
- Test setinde de hiç görmediği veriler ile model başarısı doğrulanır.



2- K-Fold Cross Validation (K-Katlı Çapraz Doğrulama Yöntemi)

- Holdout yöntemi varken bu yöntemin çıkış amacını açıklayalım, diyelim ki gözlem sayısı 100 olan bir verimiz var ve eğitim seti -%80- ve test seti -%20- olacak şekilde ayırdık, gözlem az olduğu için test kısmına yani %20 lik kısma acaba modeli doğru

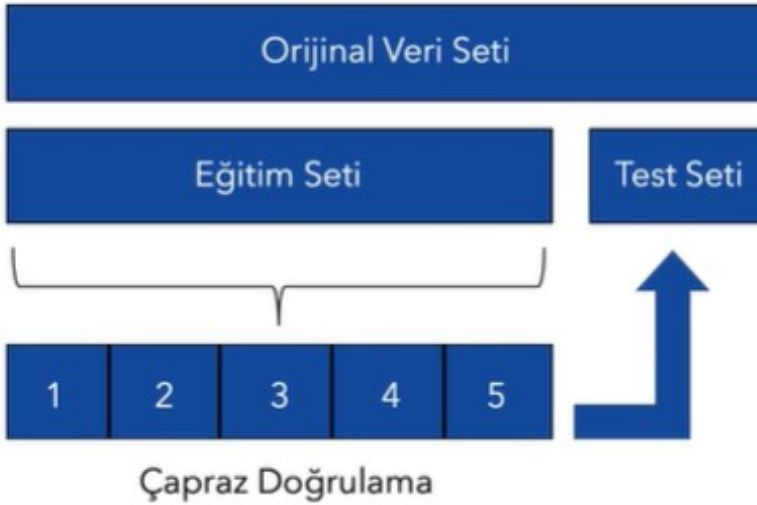
tahmin etmeye yarayacak deęerler düřtü mü? Yani elimizdeki bir sınıflandırma problemi ise ve test seti sadece 0 lardan ya da sadece 1 ler oluşuyorsa bu yanlılıęa sebep olup yanlış bir deęerlendirme yapacaktır.

- Gözlem çoksa, Holdout yöntemi problem olmayacaktır. Kullanılabilir.
- Bu yöntemdeki problem %80-%20 diye ayrılan veri setinin, kalitesini, tutarlılıęını, taşıdıęı bilgiyi, örüntüyü ne şekilde ifade ettięini bilip bilinmemesidir.

K-Katlı Çapraz Doğrulama

→ **1. yol** : Orijinal veri setini eşit 5 parçaya böl. 4 parçayla model kur, 1 parçayla test et. Başka bir 4 parçayla model kur 1 parçayla test et. Bu şekilde devam ederek 4 parçayı kendi içinde sürekli deęiřtirerek hataları hesapla. Bu hataların ortalamasını hesaplayarak Cross-Validation hatasını elde et.

→ **2. yol** : Holdout yöntemi yapılarak train-test olarak veri seti ayrılır. Sonra train seti üzerinden çapraz doğrulama yapılır. 4 parçayla model kur 1 parçayla test et. Sonra başka bir 4 parçayla model kur 1 parçayla test et. Hiperparametre optimizasyonları yap, özellik mühendislięi uygula. Tüm bunları train set üzerinde gerçekleştir. En son verinin hiç görmedięi test set üzerinden başarı deęerlendirmesi yap.



NOT :

- Bol miktarda, yüz binlerce verin varsa Holdout yöntemini kullanabilirsin.
- Bol miktarda, yüz binlerce verin varsa Cross-Validationda 2. yolu tercih edebilirsin.
- Gözlem çok az ise orijinal veriseti üzerinden Cross-Validation yapmayı tercih edebilirsin ki bu daha sağlıklı olacaktır.

Yanlılık - Varyans Deęiř Tokuřu (Bias - Variance Tradeoff)

- **Varyans**, modelin tahmin ettięi verilerin, gerçek verilerin etrafında nasıl (ne kadar) saçıldığını ölçer. Varyans, model eğitim veri setinde iyi performans gösterdięinde,

ancak bir test veri kümesi veya doğrulama veri kümesi gibi, eğitilmemiş bir veri kümesinde iyi performans göstermediğinde ortaya çıkar.

- **Bias**, gerçek değerlerden tahmin edilen değerlerin ne kadar uzak olduğudur. Tahmin edilen değerler gerçek değerlerden uzaksa, bias yüksektir.
- **Bias Variance Tradeoff** : Modelin sapması ve varyansı arasında doğru dengeyi bulmaya, Bias-Varyans takası (Bias-Variance Trade-Off) denir.

★ **Model kurmak** : Bağımlı ve bağımsız değişkenler arasındaki ilişkiyi, özütü çıkarma işlemidir. Model kurmanın amacı veriyi öğrenmek değil **verinin yapısını** öğrenmektir.

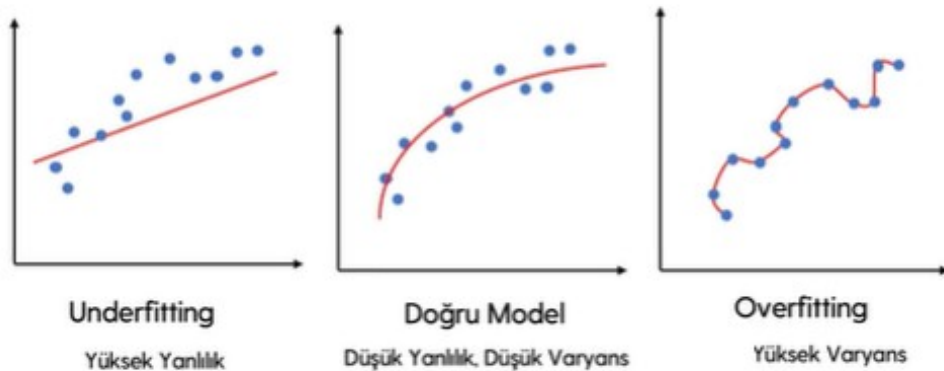
- **Overfitting** :

→ Eğer modelimiz, eğitim için kullandığımız veri setimiz üzerinde gereğinden fazla çalışıp ezber yapmaya başlamışsa ya da eğitim setimiz tek düze ise overfitting olma riski büyük demektir.

→ Eğitim setinde yüksek bir skor aldığımız bu modele, test verimizi gösterdiğimizde muhtemelen çok düşük bir skor elde edeceğiz. Çünkü model eğitim setindeki durumları ezberlemiştir ve test veri setinde bu durumları aramaktadır. En ufak bir değişiklikte ezberlenen durumlar bulunamayacağı için test veri setinde çok kötü tahmin skorları elde ederiz.

→ Overfitting problemi olan modellerde yüksek varyans, düşük bias durumu görülmektedir.

- **Underfitting** : Modelin veriyi öğrenememe problemidir. Ya da az öğrenmesidir. Değişkenlik azdır. Yanlılık yüksektir. (Bazı gözlemlere daha yakın olma, genellenebilirlik kabiliyeti kazanmamış)
- **Doğru Model** : Yanlılığı düşük, varyansı düşük olan modeldir. Verinin yapısını öğrenen modeldir.



! ÇOK ÖNEMLİ !

Modelin Aşırı Öğrenmeye Düşüğünü Overfit Olduğunu Nasıl Anlarsın ?

1 - Test seti ve eğitim seti arasındaki ilişkinin model karmaşıklığı ve tahmin hatası çerçevesinde incelenir.

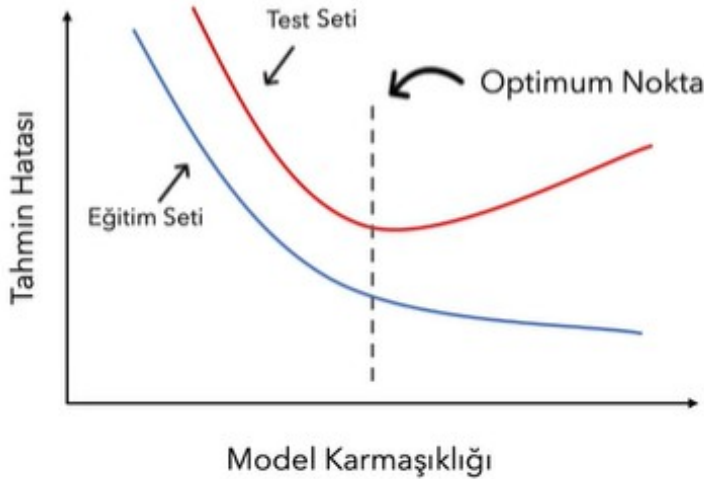
2 - Bu iki setin hatalarının birbirinden ayrılmaya başladığı noktada model overfit olmaya aşırı öğrenmeye başlamıştır.

Modelin Overfit Olduğunu Anladın Önüne Nasıl Geçersin ?

- Hatanın ayrılmaya başladığı noktada "optimum noktada" model eğitimi durdurabilirsin.
- Eğer eğitim seti tek düze ise daha fazla veri ekleyerek boyut arttırılabilir.
- Hiperparametre optimizasyonu yapılabilir.

Gelişmiş Teknikler

- Birbirleriyle yüksek korelasyonlu olan kolonlar silinebilir ya da faktör analizi gibi yöntemlerle bu değişkenlerden tek bir değişken oluşturulabilir.
- Düzenleme (Regularization), modelin karmaşıklığını azaltmak için bir kullanılan tekniktir. Bunu kayıp fonksiyonunu cezalandırarak yapar. Yani modelde ağırlığı yüksek olan değişkenlerin ağırlığını azaltarak bu değişkenlerin etki oranını azaltır. Bu yöntem, aşırı öğrenme probleminin çözülmesine yardımcı olur. Kayıp fonksiyonu, gerçek değer ile öngörülen değer arasındaki farkın karelerinin toplamıdır. Değişkenlerin ağırlığını azaltmak için regularization değerini arttırmak gerekmektedir. En popüler Regularization metotları Lasso ve Ridge teknikleridir.
- Ensembling, birden fazla ayrı modelden tahminleri birleştiren bir yöntemdir.



Model Karmaşıklığı : Modelin hassaslaştırılmasıdır. Daha detaylı tahminler yapabilmesi için özelliklerin kuvvetlendirilmesi çabasıdır. Az öğrenmeden çok öğrenmeye gidiliyordur. Model karmaşıklığı ne demektir ? soruna yanıt, farklı modeller için farklılaşacaktır.

1. Doğrusal Modellere $y = b + wx$, x^2 , x^3 gibi değerler eklenerek model karmaşıklığı artırılabilir.
2. Ağaç Yöntemlerinde ağaç 8 dala mı ayrılacak 18 dala mı ayrılacak bu işlemler modeli karmaşıklatastıracaktır.
3. Optimizasyon Yöntemlerinde mesela LGBM model karmaşıklığı parametresi iterasyon sayısıdır, arttırılarak model karmaşıklığı arttırılabilir.
4. YSA (Yapay Sinir Ağlarında) da katman sayısı, epoch sayısı, learning rate gibi parametreler arttırıldığında, ayarlamalar yapıldığında bir süreye kadar eğitimdeki hata düşecektir fakat test setinde hata artmaya başlayacaktır.