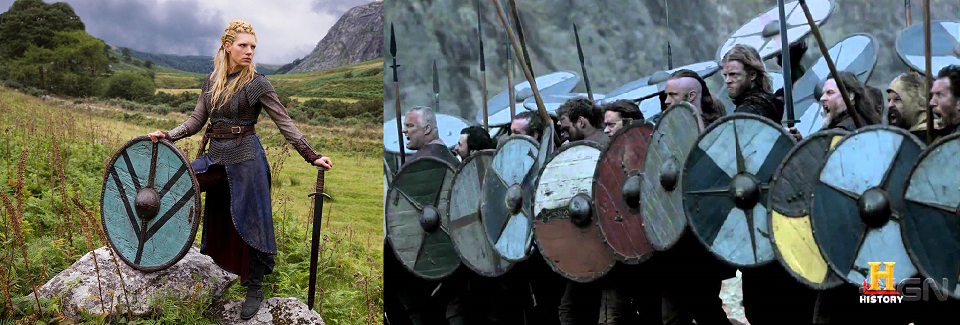
**ML SESSION -3**



**Makine öğrenimi ile istatistik arasındaki fark nedir?**

Makine öğrenimi, veri analizi ve tahminler yapmak için geniş bir teknik yelpazeyi kapsayan geniş bir alandır. Bu tekniklerin bazıları geleneksel istatistiklere kök salmışken, diğerleri daha çok bilgisayar bilimine yakındır. Bununla birlikte, makine öğrenimi ve istatistik derinlemesine bağlantılıdır ve birçok ortak hedef ve yöntem paylaşırlar

Bazı insanlar, makine öğrenimindeki büyüyen ticari ilginin sadece istatistik ve pazarlama kombinasyonu olduğunu düşünebilirler, ancak gerçeklik çok daha karmaşıktır. Makine öğrenimi ve istatistik, farklı geçmişlere ve uygulamalara sahip olsalar da, her ikisi de verideki kalıpları anlamak ve doğru tahminler yapmakla ilgilidir. İstatistik genellikle hipotez testine odaklanırken, makine öğrenimi, bir dizi hipotez arasından en iyi modeli bulma süreci ile daha fazla ilgilenir

Makine öğrenimi sadece modeller oluşturma ve tahminler yapmakla ilgili değildir. Ayrıca veri görselleştirme, veri hazırlama, özellik mühendisliği, model seçimi, model değerlendirmesi ve model yorumlanması gibi diğer alanları da kapsar. Bu nedenle, makine öğrenimi istatistiğin bir alt kümesi olarak değil, kendi özellikleri olan ve veri bilimi alanındaki her ikisi de kendilerine ait farklı alanlara sahip iki farklı alan olarak ele alınmalıdır

**Bias-Variance Trade off:**

Bias, modelin gerçek değerden ne kadar sapma gösterdiğini ifade eder.

**Bias-variance trade off**(*Bias-Varyans Dengelemesi)*, bu iki faktör arasındaki dengeyi bulmayı amaçlıyor. Modelin yüksek bias ve düşük varyansı olması,**aşırı basit**bir modeli ifade ederken, düşük bias ve yüksek varyans**, aşırı karmaşık**bir modeli ifade eder.

Yüksek bias, modelin aşırı basit olduğu ve verilerin karmaşıklığını yeterince yansıtamadığı anlamına gelir. Buna karşılık, düşük bias, modelin verileri iyi yansıttığı anlamına gelir.

Varyans, modelin aynı veri kümesindeki farklı örnekler üzerinde nasıl performans gösterdiğini ifade eder. Yüksek varyans, modelin eğitim verilerine aşırı uyduğu ve farklı veri kümesinde kötü performans gösterebileceği anlamına gelir. Düşük varyans, modelin farklı veri kümesinde iyi performans gösterebileceği anlamına gelir.

Bias-variance trade off, bu iki faktör arasındaki dengeyi bulmayı amaçlar. Modelin yüksek bias ve düşük varyansı olması, aşırı basit bir modeli ifade ederken, düşük bias ve yüksek varyans, aşırı karmaşık bir modeli ifade eder.

**İdeal durumda, modelin hem düşük bias hem de düşük varyansı olmalıdır. Bununla birlikte, bu durum genellikle mümkün değildir ve modeller arasında bir seçim yapmak gerekebilir. Bu nedenle, model seçiminde bias-variance trade off dikkate alınmalıdır.**

**Underfitting problemi:**

Underfitting problemi, bir modelin verilerdeki gerçek ilişkiyi yeterince yakalayamayacak kadar basit olduğu ve bu yüden genelleştirme yapamaması durumlarda ortaya çıkar. Bu sorun genellikle modelin karmaşıklığı ile ilgilidir. Model yeterince karmaşık değilse veya yeterince eğitim verisi yoksa, model eğitim verilerine uygun hale gelmez ve yeni verilerde yanıltıcı sonuçlar üretebilir

**Bu sorunu çözmek için aşağıdaki adımlar atılabilir**:

1. Modelin karmaşıklığını arttırmak: Daha karmaşık bir model, veri üzerindeki daha karmaşık ilişkileri yakalayabilir. Ancak, modelin çok karmaşık olması da aşırı öğrenme (overfitting) sorununa neden olabilir. Bu nedenle, uygun bir denge bulunmalıdır.
2. Modelin karmaşıklığını arttırmak: Daha karmaşık bir model, veri üzerindeki daha karmaşık ilişkileri yakalayabilir. Ancak, modelin çok karmaşık olması da aşırı öğrenme (overfitting) sorununa neden olabilir. Bu nedenle, uygun bir denge bulunmalıdır.
3. Daha fazla veri toplamak: Daha fazla veri toplamak, modelin daha fazla örüntüyü görmesini ve verilerin genel özelliklerini daha iyi kavramasını sağlayabilir. Ancak, bu seçenek her zaman mümkün olmayabilir.
4. Daha iyi özellik seçimi: Daha iyi özellikler, modelin verilerdeki örüntüleri daha iyi kavramasını sağlayabilir. Bu nedenle, modelin daha iyi özellikler seçmesine yardımcı olacak özellik seçimi teknikleri kullanılabilir.
5. Daha düzgün bir eğitim: Modelin daha uzun süre veya daha düzgün bir şekilde eğitilmesi, modelin verilerdeki örüntüleri daha iyi anlamasına yardımcı olabilir.
6. Regülarizasyon: Regülarizasyon teknikleri, modelin aşırı öğrenme sorununu önleyerek daha iyi genelleştirme yapmasını sağlayabilir. Bu teknikler arasında L1 ve L2 regülarizasyonu, dropout, early stopping gibi yöntemler bulunur

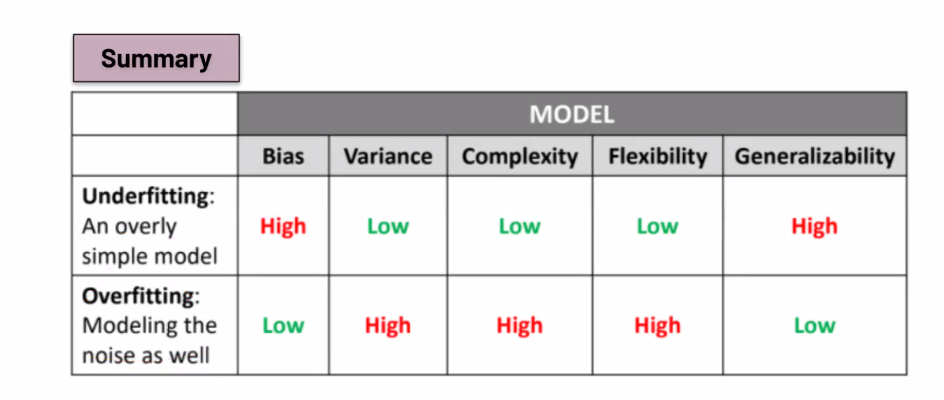
**Overfitting problemi:**

Overfitting, bir modelin eğitim verilerine çok fazla uyum sağlaması ve bu nedenle yeni verilerde performansının düşük olması durumunda ortaya çıkar.

**Bu sorunu çözmek için aşağıdaki adımlar atılabilir:**

1. Daha fazla veri toplamak: Daha fazla veri toplamak, modelin genel veri özelliklerini daha iyi anlamasını ve aşırı uyuma neden olan gürültüyü azaltmasını sağlayabilir.
2. Daha az özellik kullanmak: Daha az özellik, modelin daha az gürültüye maruz kalmasını ve aşırı uyum sorununu önlemesine yardımcı olabilir
3. Regülarizasyon kullanmak: Regülarizasyon teknikleri, modelin aşırı uyum sorununu önleyerek daha iyi genelleştirme yapmasını sağlayabilir. Bu teknikler arasında L1 ve L2 regülarizasyonu, dropout ve early stopping gibi yöntemler bulunur
4. Doğrulama seti kullanmak: Doğrulama seti, modelin eğitim sırasında genelleştirme performansını izlemek için kullanılabilir. Modelin doğrulama setindeki performansı düşükse, aşırı uyum sorunu oluşabilir ve modelde düzeltmeler yapılması gerekebilir
5. Daha basit bir model kullanmak: Daha basit bir model, modelin verilerdeki ilişkileri daha az karmaşık bir şekilde öğrenmesine yardımcı olabilir ve aşırı uyuma sorununu önleyebilir.

**Underfitting ve Overfitting problemleri:**



Aslında tüm mesele, modelin train datası ile eğitiminde öyle bir ayrıntı seviyesinde kalmalı ki eğitim sonunda dışardan gelecek her türlü gerçek datayı da kabul edilebilir optimum bir hata ile tanımlayabilmeli.

total error = bias + variance + irreducible error

formulasyonunda ie değiştirilemezdir, total erorde sabit olduğundan bias artarsa varyans düşer, bias azalırsa varyans  artar.

**Polinomial Regression:**

Polinomiyal regresyon, doğrusal olmayan ilişkileri modellemek için kullanılan bir makine öğrenmesi yöntemidir. Bu yöntemde, veri setindeki bir değişkenin değerine göre, bağımlı değişkenin polinomik bir fonksiyonu kullanılarak tahmin edilir.

Örneğin, bir veri setindeki bir değişkenin karesi ile bağımlı değişken arasında bir ilişki olduğunu varsayalım. Polinomiyal regresyon, bu ilişkiyi yakalamak için bir ikinci dereceden polinom kullanacaktır.

Polinomiyal regresyon, lineer regresyonun bir genişletmesidir ve lineer olmayan verileri modelleyebilir. Bu yöntem, veri setindeki değişkenlerin arasındaki karmaşık ilişkileri keşfetmek için kullanılır.

Polinomiyal regresyon, aynı zamanda aşırı uydurma (overfitting) gibi sorunlarla da karşılaşabilir. Bu nedenle, polinom derecesi doğru bir şekilde seçilmelidir. Polinom derecesi arttıkça, modelin karmaşıklığı artar ve aşırı uydurma riski de artar

Sonuç olarak, polinomiyal regresyon, lineer olmayan verileri modelleme ihtiyacı olan makine öğrenmesi problemlerinde kullanılan bir yöntemdir.

**Gördüğünüz gibi Polinomiyal regresyon, datasetindeki değişkenlerin arasındaki karmaşık ilişkileri keşfetmek için kullanılıyor.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Train datasında 30 değeri olmayabilir. Model farklı independent feature değerleriyle (x-trainlerle) eğitildi ve o değerlerden datayı tanımlayan bir fonksiyon buldu.

İşte o fonksiyona test datasından 30 değerini koyduğumuzda bir predict yapıyor. ve bu predictionın hatası en düşük bir prediction olmasını bekliyoruz. neden? çünkü datayı tanımlayan en doğru polinomial fonksiyonu bulmuştu.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Bayes Teoremi**

**Bayes Teoremi**, bir olayın koşullu olasılığını, olayla ilgili önceden bilinen koşulların göz önünde bulundurularak hesaplayan bir matematiksel formüldür. Temel olarak, Bayes Teoremi, yeni bilgi ve kanıtların elde edilmesiyle birlikte bir olayın olasılığı hakkındaki inançlarımızı güncellememize olanak tanır.

Makine öğrenimi bağlamında, **Bayes Teoremi sıklıkla sınıflandırma problemlerinde kullanılır**. Bu problemlerde, bir veri noktası verildiğinde, verinin hangi sınıfa ait olduğunu belirlememiz gerekmektedir. Örneğin, bir e-postanın spam olup olmadığını belirlemek için, e-postanın içeriğini analiz eden bir makine öğrenimi modeli kullanabiliriz. Bu model, e-postanın belirli özelliklerinin (örneğin, belirli kelimelerin kullanımı) belirli bir sınıfa ait olma olasılığını hesaplayabilir

Bayes Teoremi, bu sınıflandırma problemlerinde kullanılan çeşitli makine öğrenimi algoritmalarının temelinde yer almaktadır. Özellikle, Naive Bayes sınıflandırıcısı adı verilen bir algoritma, Bayes Teoremi'ni kullanarak sınıflandırma problemlerini çözmek için yaygın bir şekilde kullanılmaktadır.

**Naive Bayes algoritması, her özelliğin birbirinden bağımsız olduğunu varsayar**. Bu varsayım sayesinde, algoritma veri kümesindeki özelliklerin birlikte olasılıklarını hesaplamak yerine, her özelliğin sınıf olasılıklarını bağımsız olarak hesaplayabilir. Bu, hesaplamanın daha hızlı ve daha basit olmasını sağlar.

Özetlemek gerekirse, Bayes Teoremi, makine öğrenimi algoritmalarında sınıflandırma problemlerinin çözümünde yaygın olarak kullanılan bir matematiksel formüldür. Naive Bayes algoritması gibi birçok algoritma, Bayes Teoremi'ni temel alır ve özellikle büyük veri kümelerinde **sınıflandırma** problemlerini hızlı ve etkili bir şekilde çözmeye yardımcı olur.

**ML LAB - 1**

**Best fitting line;**

Datasetindeki dağılımı mümkün olduğunca iyi bir şekilde özetlemelidir. Bunu lineer bir fonksiyonunun en uygun parametrelerini hesaplayarak yapar. Bu parametreler, eğim (**slope**) ve **intercept**olarak ifade edilir ve **en küçük kareler yöntemi** ile hesaplanır.

**Gradient descent**yöntemi ise doğrunun en uygun eğim ve intercept'i hesaplamak için **iteratif**bir yaklaşım kullanır. Başlangıçta rastgele değerler atanır ve ardından hatanın azaltılması için **parametreler yavaş yavaş güncellenir**. Bu işlem, **hatanın minimum olduğu noktaya kadar devam eder.**

Gradient descent yöntemi, büyük veri kümeleri için daha hızlı hesaplama yapabilme avantajına sahiptir. Ancak, doğru **hiperparametrelerin**seçimi ve **konverjansın**sağlanması için daha fazla dikkat gerektirir.

**Konverjans**, bir optimizasyon algoritmasının iteratif olarak güncellediği parametrelerin belirli bir değere yaklaşması anlamına gelir. Bu değer, genellikle optimizasyon problemine bağlı olarak hedef değer, minimum veya maksimum noktadır.

Optimizasyon algoritmaları, başlangıçta rastgele değerlerle başlar ve parametreleri her adımda güncelleyerek hedefe yaklaşır. Konverjans, bu güncelleme işleminin yavaşlaması veya durmasıyla gerçekleşir. Algoritmanın konverjans sağladığı noktada, parametreler artık çok az değişiyor veya değişmiyor ve hedef değere yakın bir değere ulaşmıştır.

Optimizasyon problemlerinde, doğru bir konverjans noktasına ulaşmak, algoritmanın istenen sonuca ulaşması için önemlidir. Ancak, bazı durumlarda algoritma, hedefe ulaşamadan önce durabilir veya yanlış minimum veya maksimum noktaya yakınsayabilir. Bu nedenle, optimizasyon algoritmalarının konverjans özellikleri, performans ve sonuçların doğruluğu açısından önemlidir.

***(Konverjans****, bu güncelleme işleminin yavaşlaması veya durmasıyla gerçekleşir. Algoritmanın****konverjans sağladığı noktada, parametreler artık çok az değişiyor veya değişmiyor ve hedef değere yakın bir değere ulaşmıştır.)***

**Coef\_ :**

Lineer regresyon modelinde "**coef\_" (katsayı)** olarak adlandırılan coefficient değeri, modelin öğrendiği **doğrunun eğimini** ifade eder. Bu katsayı, lineer regresyon modelindeki bağımsız değişkenlerin (X) her biri için bir tane olmak üzere hesaplanır.

**Katsayı değerleri, bağımsız değişkenlerin model üzerindeki etkilerini gösterir.** Örneğin, bir lineer regresyon modelinde yüksek bir katsayı değeri, ilgili bağımsız değişkenin yüksek bir etkiye sahip olduğunu gösterirken, düşük bir katsayı değeri düşük bir etki olduğunu gösterir.

**Katsayı değerleri, modelin eğitildiği veri kümesindeki değişkenlerin arasındaki ilişkiyi açıklar**. Katsayı değerleri, bir modelin performansını değerlendirmek ve modelin özelliklerini anlamak için kullanılabilir. Ayrıca, katsayı değerleri, bir modelin yorumlanabilirliği için de önemlidir.

**mae, rmse normalizasyonu:**

MAE (ortalama mutlak hata) ve RMSE (kök ortalama karesel hata) gibi hata metrikleri, bir makine öğrenimi modelinin tahminlerinin ne kadar doğru olduğunu değerlendirmek için kullanılır. Bu metrikler, gerçek değerler ile tahmin edilen değerler arasındaki farkı ölçer.

Bu farkın büyüklüğü, hedef değişkenin ölçeği ile doğrudan ilişkilidir. Farkın ölçeğinin büyük olması, hatanın da büyük olacağı anlamına gelir. Bu nedenle, MAE ve RMSE, hatanın ölçeğinden bağımsız olacak şekilde normalize edilir.

Mae ve rmse normalizasyonu, hatayı ölçeklendirmek için yapılır.

Bu normalizasyon, hedef değişkenin ortalama değeri ile yapılır. MAE ve RMSE'nin ortalama değeri, hedef değişkenin ortalama değeri ile farkı hesaplanarak elde edilir. Bu hesaplama, hatanın büyüklüğünü, hedef değişkenin ortalama değeri ile ilişkilendirir ve hatayı ölçeklendirir

Bunun yanı sıra, hedef değişkenin standart sapması da kullanılabilir. Bu, hedef değişkenin dağılımı hakkında daha fazla bilgi sağlar ve normalizasyonun daha doğru olmasını sağlar. Ancak, genellikle hedef değişkenin ortalama değeri daha yaygın olarak kullanılır çünkü daha basit bir normalizasyon yöntemidir.

