Büyük Veri Analizi

Ders 7

Ali Mertcan KOSE Ph.D.
amertcankose@ticaret.edu.tr

İstanbul Ticaret Üniversitesi

Giriş

Veriler üzerinden bilgi çıkarma işlemine "Makine Öğrenmesi" denir. Makine Öğrenmesi; İstatistik, Yapay Zeka ve Bilgisayar Bilimlerinin kesiştiği bir alandır. Ayrıca Makine Öğrenmesi, tahmine dayalı analitik ya da istatistiksel öğrenme olarak da ifade edilir.

Makine Öğrenmesinin temel amacı elde edilen veriler doğrultusunda istatistiksel yöntemleri kullanarak veri analizi yapmaktır. Verilerdeki örüntüleri otomatik olarak tespit edebilen ve daha sonra ortaya çıkarılan örüntüleri gelecekteki verileri tahmin etmek için kullanan bir dizi yöntemden oluşur.

Bu yöntemler ile örnek verileri veya geçmiş deneyimleri kullanarak bir performans kriterini optimize etmek için bilgisayarlar programlanmaktadır. Bazı parametrelerle tanımlanmış bir modelde, öğrenme; eğitim verisi ya da geçmiş deneyimler kullanılarak model parametrelerini optimize etmek amacıyla bilgisayar programları ile çalıştırılır

Giriş

Makine Öğreniminde matematiksel modeller oluştururken istatistik teorisi kullanır, çünkü Makine Öğrenmesinin temel görevi elde edilen örneklem üzerinden çıkarım yapmaktır. Burada bilgisayar biliminin iki rolü vardır: Birincisi; eğitim veri setinde optimazyon problemini çözmek ve aynı zamanda genel olarak sahip olduğumuz büyük veriyi depolamak ve işlemek için etkili algoritmaları kullanmaktır. İkincisi; bir model öğrenildikten sonra onun gösteriminin ve çıkarımı için algoritmik çözümünün de verimli olmasını sağlamaktır. Bazı uygulamalarda çıkarım algoritması ya da öğrenmenin etkinliği, uzay ve zaman karmaşıklığı tahmin doğruluğu kadar önemli olabilir

Giriş

Makine öğrenmesi algoritmalarının en başarılı türleri bilinen örneklerden genellemeler yaparak karar verme süreçlerinin makineleştirilmesidir. Denetimli Öğrenme (Supervised Learning) olarak bilinen bu ortamda kullanıcı, algoritmaya girdi (Input) ve istenen çıktı (Output) çiftlerini sağlar ve algoritma, bir girdi verildiğinde istenen çıktıyı üretmenin bir yolunu bulur. Özellikle algoritma, daha önce hiç görmediği bir girdi için, bir insanın yardımı olmadan çıktı oluşturur

Denetimsiz Öğrenmede (Unsupervised Learning) amaç girdilerdeki düzenlilikleri bulmaktır. Denetimli öğrenmede amaç, bir denetçi tarafından doğru değerleri sağlanan girdiden çıktıya eşleşmenin öğrenilmesi iken, denetimsiz öğrenmede böyle bir denetçi yoktur ve elimizde yalnızca girdi verileri vardır. Bazı uygulamalarda, sistemin çıktısı bir dizi eylemden oluşur. Böyle bir durumda, tek eylem önemli değildir. Bu tarzdaki uygulamaların politikası hedefe ulaşmak için doğru eylemleri gerçekleştirmektir. Böyle bir durumda makine öğrenimi programı politikaların iyiliğini değerlendirebilmeli ve bir politika oluşturabilmek için geçmişteki iyi eylem dizilerinden öğrenebilmelidir. Bu tür öğrenme yöntemlerine Takviyeli Öğrenme (Reinforcement Learning) algoritmaları denir.

Giriş

Çalışma kapsamında belirlenen modelimizde Denetimli Öğrenme algoritmalarının kullanılmasından dolayı, bu bölümde sınıflandırma algoritmaları ele alınacaktır. Burada amaç girdilerden elde ettiğimiz sınıf sayısı doğrultusunda, çıktılarla eşleşme durumunu öğrenmektir. Elde edeceğimiz çıktılar iki kategoriden oluşuyorsa ikili (binary) sınıflandırma, iki den fazla kategori olması durumunda çoklu (multi) sınıflandırma olarak adlandırılır. Sınıf etiketleri ayrışıksa, buna çok etiketli sınıflandırma adı verilir, ancak amaç en iyi şekilde birden fazla ilişkili sınıf etkisinin tahmin edilmesidir.

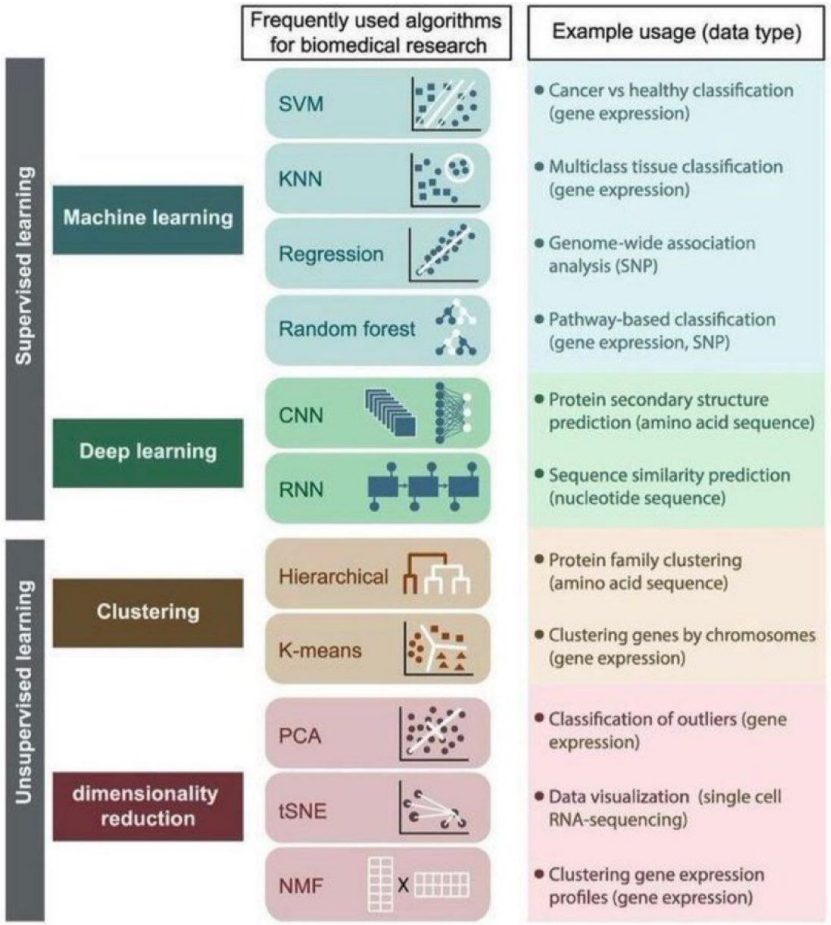

Figure 1: Makine Öğrenmesi.

Lojistik Regresyon

Lojistik regresyon modeli, K sınıflarının sonsal olasılıklarını x'teki doğrusal fonksiyonlar aracılığıyla modelleyen ve aynı zamanda bunların toplamının bir olmasını ve [0,1] aralığında kalmasını sağlayan bir yöntemdir. Lojistik regresyon modeli formu aşağıda verilmiştir.

$$
\begin{aligned}
\log\left(\frac{Pr(G=1 \mid X=x)}{Pr(G=K \mid X=x)}\right) &= \beta_{10} + \beta_1^T x \\
\log\left(\frac{Pr(G=2 \mid X=x)}{Pr(G=K \mid X=x)}\right) &= \beta_{20} + \beta_2^T x \\
&\vdots \\
\log\left(\frac{Pr(G=K-1 \mid X=x)}{Pr(G=K \mid X=x)}\right) &= \beta_{(K-1)0} + \beta_{K-1}^T x
\end{aligned}
\tag{1}
$$

Lojistik Regresyon

Model, K-1 log-odds veya logit dönüşümleri (olasılıklarının bire eşit olduğunu gösterir) cinsinden belirtilir. Model, odds oranlarında payda olarak son sınıfı kullansa da, payda seçimi, tahminlerin bu seçimi altında eş değişkenli olması nedeniyle keyfidir. Bu hesaplama şu şekilde gösterilmektedir.

$$Pr(G = k \mid X = x) = \frac{exp(\beta_{k0} + \beta_k^T x)}{1 + \sum l = 1^{K-1} exp(\beta_{l0} + \beta_l^T x)}, k = 1, ...$$
$$Pr(G = K \mid X = x) = \frac{1}{1 + \sum l = 1^{K-1} exp(\beta_{l0} + \beta_l^T x)}, \quad (2)$$

Lojistik Regresyon

Bunların toplamı açıka 1'e eşittir. Tüm parametre setlerindeki bağımlılığı vurgulamak için $\theta = \beta_1 0, \{\beta_1^T, \ldots, \beta_{(K-1)0}, \beta_{(K-1)}^T\}$, Burada olasılıklar $Pr(G = k \mid X = x) = p_k(x; \theta)$ olarak ifade edilir. Genel olarak lojistik modeller K=2 sınıftan oluşmaktadır. K=3 sınıf olması durumunda multinominal lojistik model olarak adlandırılır.

Doğrusal Diskirminant Analizi

Doğrusal Diskriminant analizi, sınıflandırma problemlerinde boyutsallığın azaltılmasına yönelik denetimli bir yöntemdir. İki sınıfın olduğu durumla başlanır, sonra K>2 sınıflarına genelleştirilir. C_1 ve C_2 olmak üzere iki sınıftan örnekler verildiğinde, w vektörü tarafından tanımlanan yön bulunmak istenir, öyle ki veriler w üzerine yansıtıldığında, iki sınıftan örnekler mümkün olduğunca birirbirinden ayrılır.

$$z = w^T x \tag{3}$$

Burada x'in w üzerine iz düşümü z'dir ve dolayısıyla d'den 1'e kadar bir boyutsallık azalması vardır.

Doğrusal Diskirminant Analizi

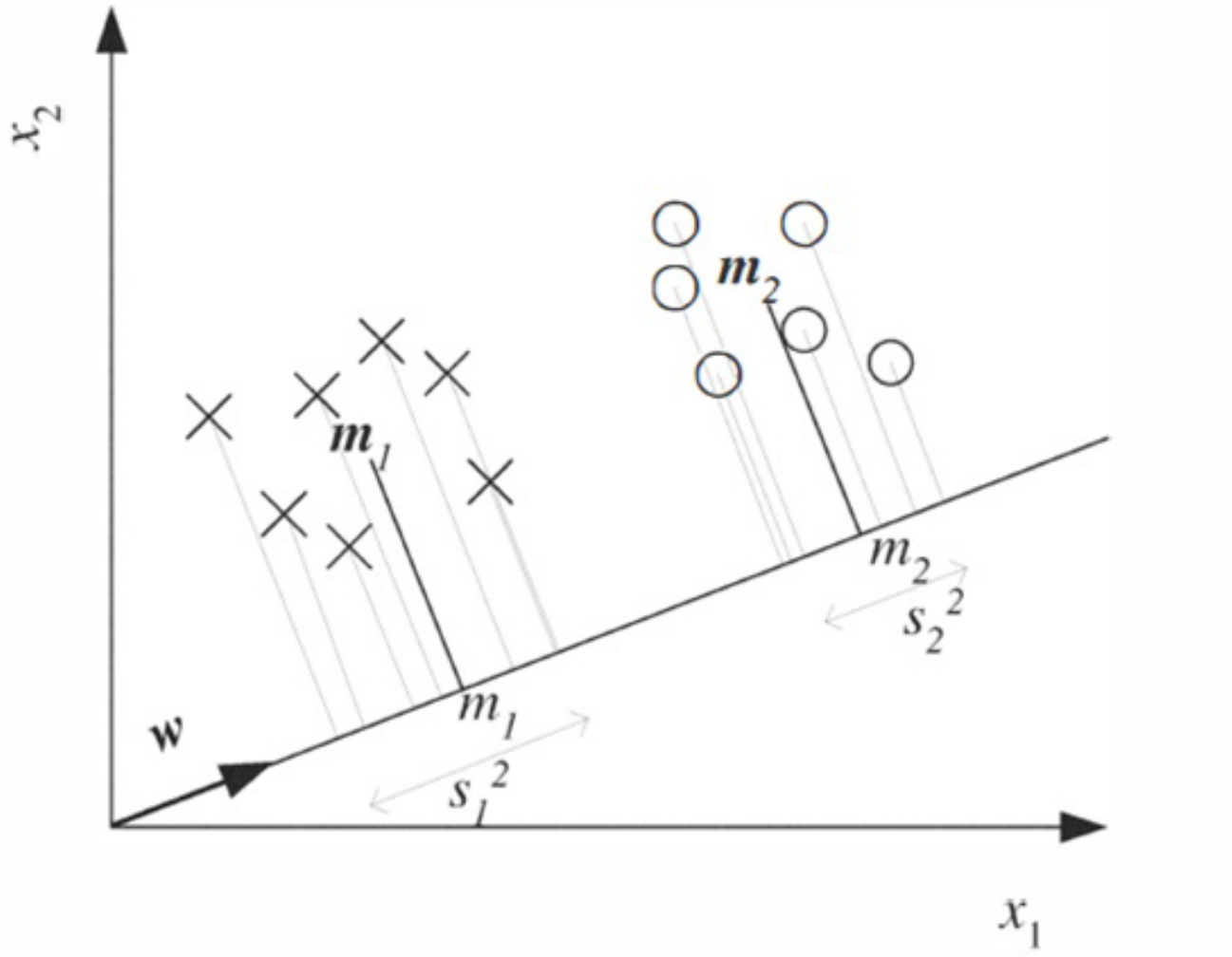

Figure 2: w ye yansıtılmış iki boyutlu, iki sınıflı veriler.

Doğrusal Diskirminant Analizi

Burada $\mathbf{m_1}$ ve m_1 sırasıyla izdüşümü öncesi ve sonrası olmak üzere C_1' den alınan örneklem ortalamalarıdır. $\mathbf{m_1} \in R^d$ ve $m_1 \in$ R. Öyle ki, eğer $\mathbf{x^t} \in C_1$ ve $r^t \in C_2$ ise X= $\{\mathbf{x^t}, r^t\}$ r^t= 1 dir.

C_1 ve C_2 örneklem düzeyinde elde edilen saçılım üzerinden varyans-kovarsyans matrisi üzerinden maksimizasyon yapılarak sınıflama yapılır.

K-Yakın Komşular (KNN)

Parametrik olmayan bir sınıflandırıcının basit bir örneği, K en yakın komşu (KNN) sınıflandırıcısıdır. Bu basitçe eğitim setindeki x test girişine en yakın olan K noktalarına bakar. Bu kümede her sınıfın kaç üyesi olduğunu sayar ve aşağıdaki ampirik kesri tahmin olarak döndürür.

$$p(y = c \mid x, D, K) = \frac{1}{K} \sum i \in N_K(x, D) \parallel (\mathrm{e})(y_i = c) \quad (4)$$

K-Yakın Komşular (KNN)

Burada N_K(x, D) = D'de x'e en yakın K noktaları (indisleri) $\|$ e gösterge fonksiyonu aşağıdaki gibi tanımlanır.

$$\| \text{(e)} = \begin{cases} 1, \textit{eger e dogruysa} \\ 0, \textit{eger e yanlissa} \end{cases} \tag{5}$$

Bu yöntem bellek tabanlı veya örnek tabanlı öğrenmeye bir örnektir. Bu yöntemle kullanılacak en yaygın uzaklık metriği Öklid uzaklığıdır, ancak diğer ölçütlerde kullanılabilir. Girdinin iki boyutlu olması durumunda üç sınıf için K=10 olur. KNN sınıflandırıcısı basit bir yöntem olup, iyi bir uzaklık ölçütü kullanılması koşuluyla iyi sonuçlar verebilir.

Sınıflandırma ve regresyon Ağaçları (CART)

Sınıflandırma ve regresyon ağaçları (CART) girdi alanını yinelemeli olarak bölümlendirerek ve her bölgenin sonuçta yerel bir model olarak tanımlanmasıyla bilinir.

Sınıflandırma ve regresyon Ağaçları (CART)

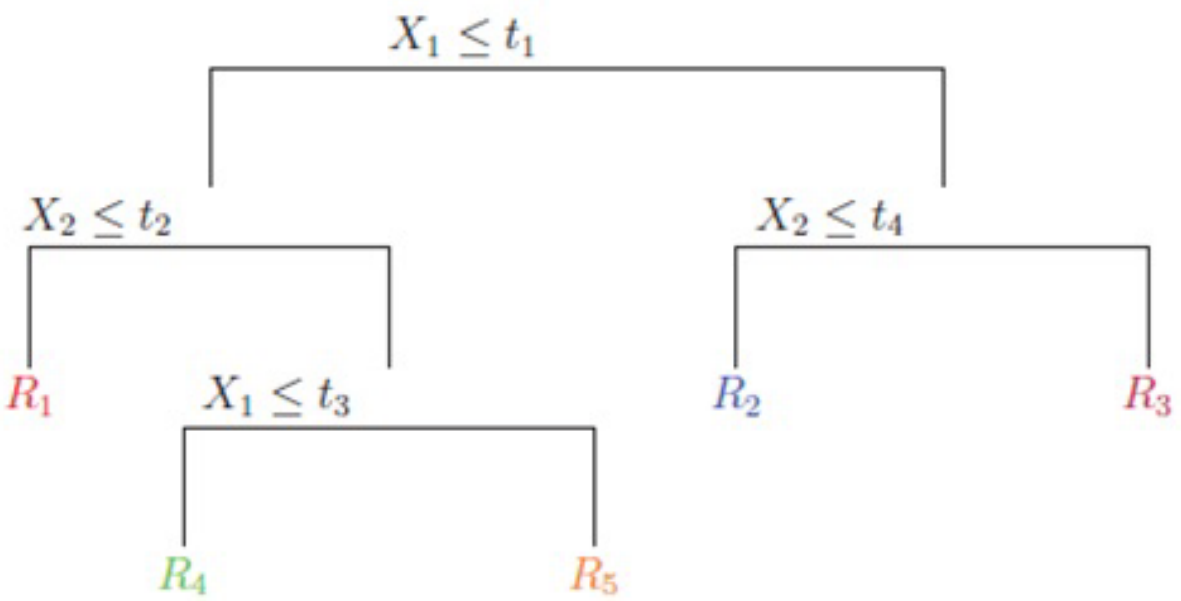

Figure 3: İki girdili basit regresyon ağacı.

Sınıflandırma ve regresyon Ağaçları (CART)

CART modeli aşağıdaki biçimde yazılabilir.

$$f(x) = \mathbb{E}[y \mid x] = \sum m = 1^{M} w_m \parallel (x \in R_m) = \sum m = 1^{M} w_m \phi(x; v_m) \tag{6}$$

Sınıflandırma ve regresyon Ağaçları (CART)

Burada R_m m'inci bölgedir, w_m bu bölgedeki ortalama yanıttır ve vm kökten m'inci yaprağa giden yolda bölünecek değişken seçimini ve eşik değerini kodlar. Bu durumda CART modelinin, temel fonksiyonların bölgedeki yanıt değerini belirleyerek uyarlanabilir bir temel fonksiyon modeli olduğunu açıkça ortaya koymaktadır.

Naive Bayes

Naive Bayes sınıflandırıcıları basit olasılıksal sınıflandırıcılardır. Temeli Bayes teorimine dayanarak, arasında güçlü (naif) bağımsızlık varsayımı özelliği de yer alır. Bayes teorimini matematiksel olarak aşağıdaki formülle ifade edilir.

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)} \tag{7}$$

Naive Bayes

$X = (x_1, x_2, \ldots x_n)$ vektörünün sınıflandırılacak her bir değeri c_j olarak ifade edilen K sınıflarında yer aldığı bir örneği (n bağımsızlık özelliğinde) varsayalım. Bu durumda, Bayes teorimi kullanılarak $P(c_j \mid X)$ sonsal olasılığı, $P(c_j)$, $P(X)$ ve $P(X \mid c_j)$'den hesaplanabilir. Naive Bayes sınıflandırıcısı, bir tahmin edici (x_i) değerinin c_j sınıfı üzerindeki etkisinin diğer tahmin edici değerlerinden bağımsız olduğunu varsayan basitçe (naive) bir varsayım olan sınıf koşullu bağımsızlık sağlar.

$$P(c_k \mid X) > P(c_j \mid X), 1 \leq j \leq K, j \neq k \tag{8}$$

Destek Vektör Makineleri

Makine öğreniminde, destek vektör makineleri (aynı zamanda destek vektör ağları) veri analizi ve sınıflandırma ile regresyon analizi için kullanılan denetimli öğrenme modelleridir.

Destek vektör sınıflandırıcısı durumunda, sınıflar arasında olası doğrusal olmayan sınırlarla ilgili sorunu çözmek için, tahmincilerin karesel, kubik ve hatta daha yüksek dereceli polinom terimlerini kullanarak özellik uzayı genişletilebilir. Bu genişletme, girdi verilerinin doğrusal olmayan ilişkilerini yakalamak için kullanılır ve "kernel trick" olarak adlandırılır. Bu yöntemle, doğrusal olarak ayrılabilir olmayan veri kümeleri üzerinde etkili sınıflandırma yapılabilir.

Destek Vektör Makineleri

Kernel Fonksiyonu $\mathbf{K(x_i,x_j)}$	**Formül**
Linear	$x_i^T x_j$
Gaussian radial basis function	$e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$
Polynomial	$(x_i . x_j + a)^b$
Sigmoidal	$(ax_i . x_j - b)$
Laplacian	$e^{-\frac{\|x_i - x_j\|}{\sigma}}$
Rational quadratic	$1 - \frac{\|x_i - x_j\|^2}{\|x_i - x_j\|^2 + c}$
Power	$\|x_i - x_j\|^d$
Log	$-log\|x_i - x_j\|^2 + 1$
Multiquadratic	$\sqrt{\|x_i - x_j\|^2 + c}$
Wave	$\frac{\theta}{\|x_i - x_j\|} \sin \frac{\|x_i - x_j\|}{\theta}$

Figure 4: Kernel Fonksiyonları ve Formülleri.

Gradyan Arttırma (Xgboost)

Xgboost yüksek oranda ölçeklenebilir olduğu için tasarlanmış gradyan arttırmaya dayalı bir karar ağacı topluluğudur.

$$L_{xgb} = \sum i = 1^{N} L(y_i, F(x_i)) + \sum m = 1^{M} \Omega(h_m) \tag{9}$$

$$\Omega(h_m) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \tag{10}$$

Adaptif Arttırma (Adaboost)

Boosting (Arttırma), çoklu "temel" sınıflandırıcıları birleştirerek, performansı bu temel sınıflandırıcılardan herhangi birinden önemli ölçüde daha iyi olan bir komite oluşturabilen güçlü bir tekniktir.

$$\Omega(J_m) = \sum n = 1^N w_n^{(m)} I(y_m(X_n)) \neq t_n) \tag{11}$$

Burada $I(y_m(x_n) \neq t_n)$ gösterge fonksiyonunu ifade eder ve $y_m(xn)$ $\neq t_n$ olduğunda 1'e eşit olur, aksi halde 0'dır.

Rastgele Orman

Rastgele orman olarak bilinen bu teknik, rastgele seçilmiş bir girdi değişkenleri altkümesine dayalı olarak birçok karar ağacı oluşturarak çalışır. Her bir ağaç, örnekleme ve bölünmeler yapılırken rastgelelik içerir. Bu, her bir ağacın farklı bir şekilde öğrenmesini sağlar ve temel öğreniciler arasındaki korelasyonu azaltır. Sonuç olarak rastgele orman genellikle yüksek tahmin doğrululuğuna sahip güçlü bir modeldir.

$$f(x) = \sum_{m=1}^{M} \frac{1}{M} f_m(x) \tag{12}$$

Burada f_m m'inci ağacı ifade eder. Bu tekniğe "boostrap aggregating" olarak adlandırılan bagging tekniği denir. Basitçe, verinin farklı altkümeleri üzerinden aynı öğrenme algoritmasını yeniden çalıştırmak, sonuç olarak birbiriyle yüksek korelasyonlu tahminciler elde edilmesine neden olabilir.

Yapay Sinir Ağları

Yapay sinir ağlarının arkasındaki fikir, bir sinir hücresinin çıkışının, başka bir sinir hücresinin girdisini oluşturacak bir şekilde birbirine bağlayarak bir sinir ağının oluşturulabilmesidir. Sinir ağları için farklı mimari türleri yer almaktadır. Çok katmanlı algılayıcılar (MLP) yöntemi en eski ve en basit model olarak Rosenblatt tarafından tanıtılmıştır. Özellikle Evrişimli Sinirsel Ağları (CNN) görüntü işleme için geliştirilen yapay sinir ağ mimarisinden biridir.

Yapay Sinir Ağları

Ayrıca, zaman serilerinde veya metinlerde meydana gelen sıralı veriler için kullanılabilen bir başka sinir ağı mimarisi olarak Devirli Sinirsel Ağlarından(RNN) bahsedilebilir. Yapay Sinir Ağlarında bir x girdisi ve y= f(x,θ) çıktısı vardır. Burada θ parametreleri öğrenme örnekleminden tahmin edilmektedir. İstatistiksel öğrenmede olduğu gibi, lokal minimum noktalarında dışbükey olmayan bir fonksiyonu minimize etmek gerekir. Aşağıda yapay sinir ağına ilişkin fonksiyon verilmiştir.

Yapay sinir hücresi, fonksiyonu fj olarak ifade edilir. Bu fonksiyon, vektörü $w_j = (w_j, 1,\ldots,w_j,\mathrm{d})$ bağlantı ağırlıkları vektörü ile ağırlıklandırılmış bir x=$(x_1, \ldots, x_d)$ girdisine sahiptir. Ayrıca nöron sapması b_j ile gösterilir. $\sum_{i=1}^{d} w_{j,i}x_i + b_j$ olarak toplama işlemi ifade edilir. Bunun yanı sıra, belirli bir aktivasyon fonksiyonu ϕ olarak ifade edilerek, $\phi(\sum_{i=1}^{d} w_{j,i}x_i + b_j)$ olarak gösterilir. Diğer taraftan özdeşlik fonksiyonu $\phi(x) = x$ sigmoid fonksiyonları $\phi(x) = \frac{1}{1+e^{-x}}$ aktivasyon fonksiyonu türleri olarak yer almaktadır.

Aktivasyon Fonksiyonu Φ(x)	**Formül**
Identity	X
Sigmoid	$\frac{1}{1+e^{-\beta x}}$
Rectified linear unit (ReLU)	Max (0,x)
Hard threshold	$\mathbf{1}_{x\geq\beta}\begin{cases}0 & eğer\ x<\beta\\1 & eğer\ x\geq\beta\end{cases}$
Hyperbolic tangent(tanh)	$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}=\frac{e^{2x}-1}{e^{2x}+1}$
Piecewise linear	$\begin{cases}0 & eğer\ x\geq x_{min}\\ mx+b & eğer\ x_{max}>x>x_{min}\\1 & eğer\ x\geq x_{min}\end{cases}$
Gaussian	$\frac{1}{\sqrt{2\pi\sigma}}e^{\frac{-(x-\mu)^2}{2\sigma^2}}$

Figure 5: Aktivasyon Fonksiyonları ve Formülleri.

Yapay Sinir Ağları

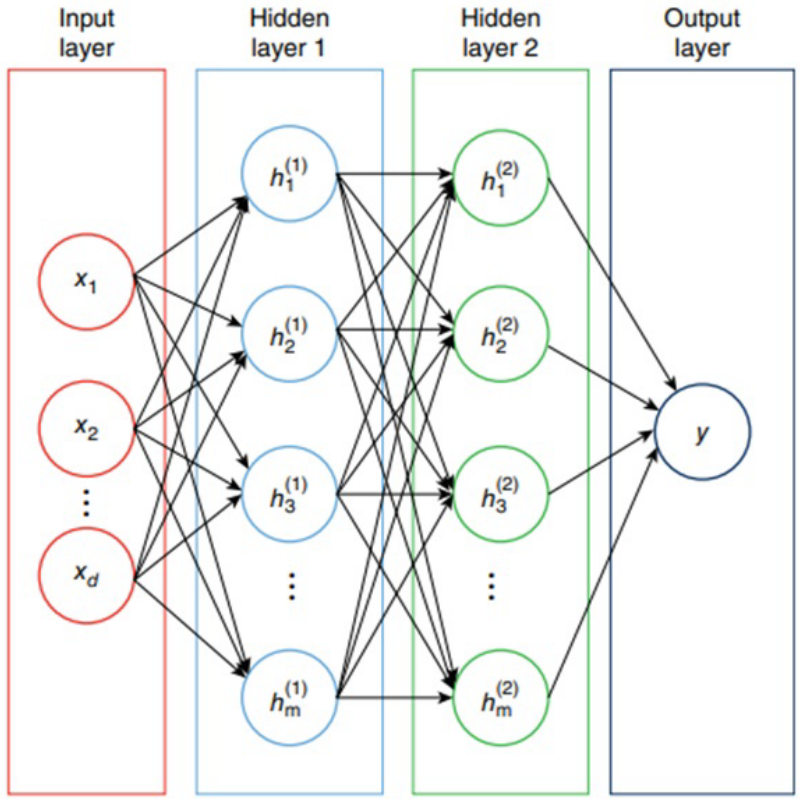

Figure 6: Sinir ağı katmanlarının şematik gösterimi.

Yapay Sinir Ağları

CNN: Genellikle görüntü işlemede kullanılan ve girdi olarak görselleri alan bir derin öğrenme algoritmasıdır. Farklı operasyonlarla görsellerdeki featureları (özellikleri) yakalayan ve onları sınıflandıran bu algoritma farklı katmanlardan oluşmaktadır. Convolutional Layer, Pooling ve Fully Connected olan bu katmanlardan geçen görsel, farklı işlemlere tabii tutularak derin öğrenme modeline girecek kıvama gelir. CNN modelleri oluştururken, unstructural (düzensiz) veri ile uğraştığımızdan klasik makine öğrenmesi algoritmalarına kıyasla veri ön işleme kısmında çok uğraşılmamaktadır.

LSTM: Değerleri rastgele aralıklarla hatırlayan bir RNN mimarisidir. Öğrenilen ilerleme kaydedildiğinde saklanan değerler değiştirilmez. Nöronlar arası ileri ve geri bağlantılara izin verir. LSTM, zaman serilerini sınıflandırmak, işlemek ve öngörmek için oldukça uygundur. LSTM, duygu analizi, metin üretme ve zaman serileri gibi birçok konuda kullanılır

Yapay Sinir Ağları

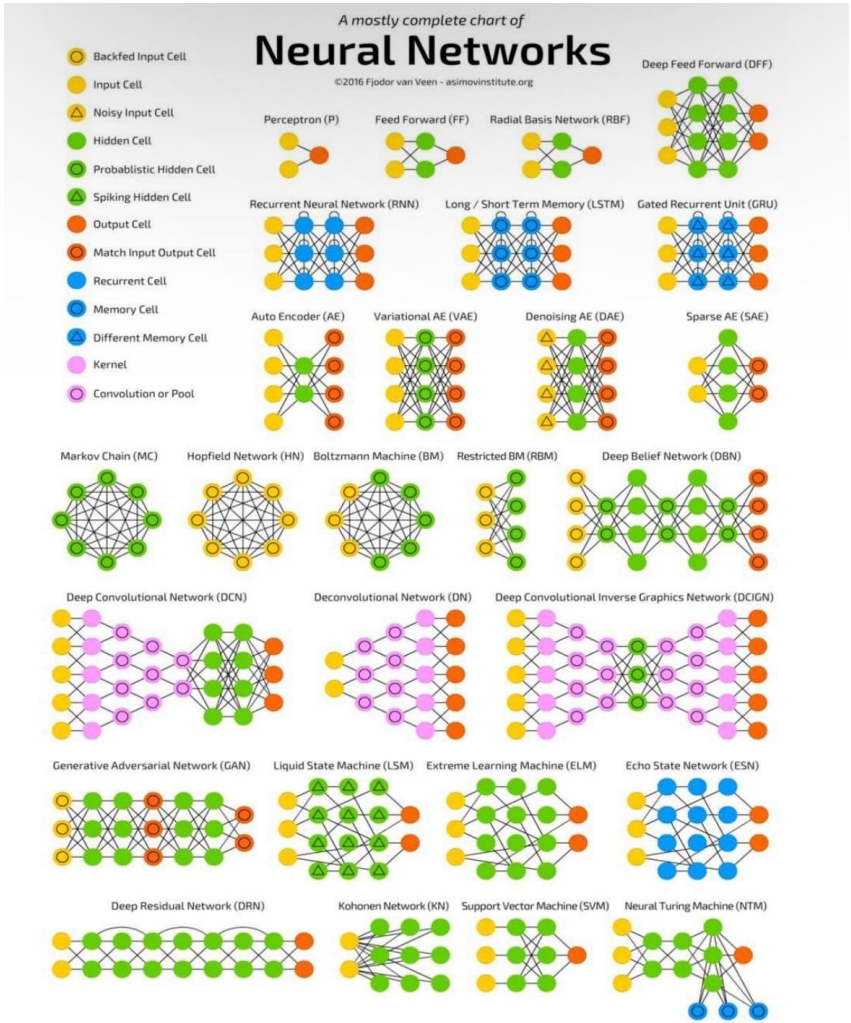

Figure 7: Sinir ağı Türleri.

Makine Öğrenmesinde Uyum

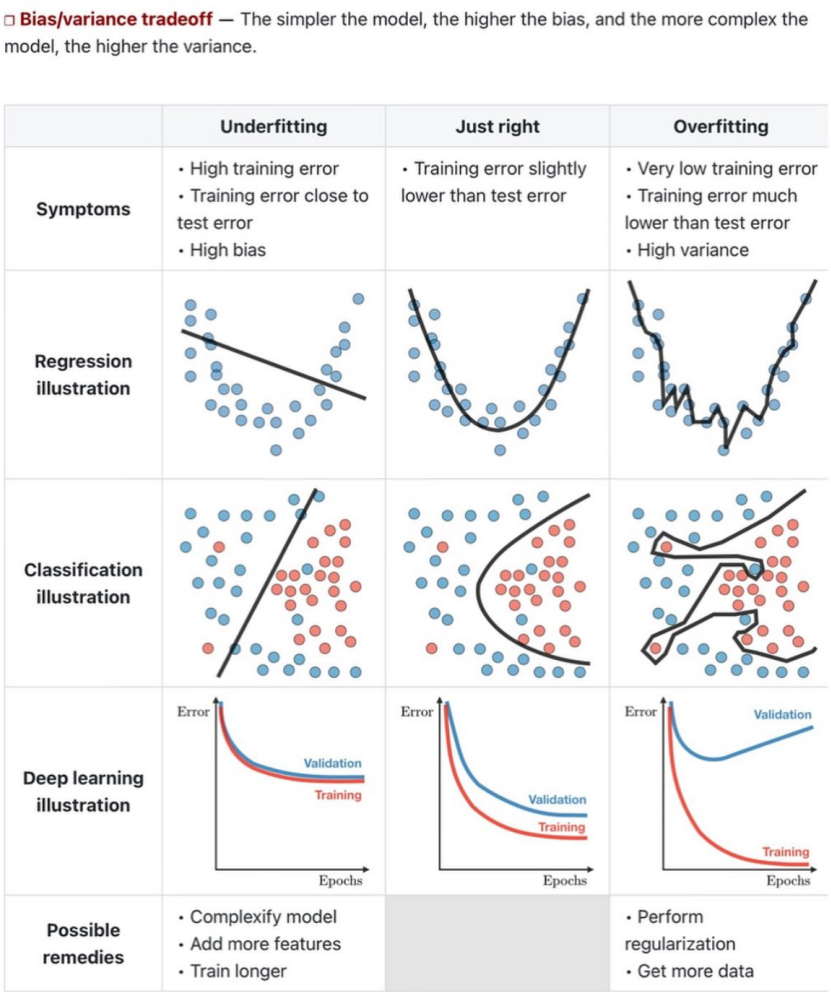

Figure 8: Uyum veUyumsuzluk Durumları.

Makine Öğrenmesinde Uyum

Overfitting: Bir makine öğrenimi modelinin eğitim verilerine aşırı derecede uyum sağlaması ve bu nedenle yeni ve farklı verilere genelleme yapma yeteneğini kaybetmesidir. Düşük Genelleme Yeteneği : Aşırı öğrenen bir model,eğitim verileri üzerinde iyi performans gösterirken,yeni verilerle karşılaşınca beklenenden daha kötü sonuçlar verebilir.

Underfitting: Veri biliminde bir veri modelinin girdi ve çıktı değişkenleri arasındaki ilişkiyi doğru bir şekilde yakalayamadığı ve hem eğitim seti hem de görünmeyen veriler üzerinde yüksek bir hata oranı oluşturduğu bir senaryodur.

Makine Öğrenmesinde Uyum

Oversampling: Dengesiz veri setlerindeki sınıflar arasındaki dengesizliği gidermek için kullanılan bir tekniktir. Oversampling,azınlık sınıfındaki örnek sayısını artırarak veri setinin dengesini sağlar.Bu,azınlık sınıfındaki örnekleri çoğaltarak veya sentetik örnekler oluşturarak gerçekleştirilebilir.

Undersampling: Veri bilimi projelerinde sıkça karşılaşılan bir sorundur. Sınıflandırma algoritmaları genellikle dengeli eğitim setlerini varsayar, ancak gerçek veri setlerinde sınıflar arasında büyük dengesizlikler olabilir. Özellikle azınlık sınıfının doğru tahmin edilmesi kritik öneme sahip uygulamalarda, bu dengesizlikler önemli sorunlar yaratabilir.

Makine Öğrenmesinde Uyum

4 tane Cross validation yöntemi vardır;

- K-fold validation
- Time Series validation
- Holdout Cross validation
- Leave-One Out validation.

Makine Öğrenmesinde Uyum

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$AUC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right)$$

Figure 9: Uyum veUyumsuzluk Durumları.

Makine Öğrenmesinde Uyum

İstatistikte, veriler üzerinde bazı heaplamalar yapmadan önce çeşitli sayısal zorluklardan kaçınmak ve daha iyi sonuçlar elde etmek amacıyla veya numerik veriler arasında farklılığın çok fazla olduğu durumlarda verileri daha dar bir aralığa sıkıştırmak için Standardizasyon ve Normalizasyon teknikleri kullanılır.

Makine Öğrenmesinde Uyum

Standardisation (Standartlaştırma)	Normalisation (Normalleştirme)
$z = \frac{x - \mu}{\sigma}$	$z = \frac{x - min(x)}{[max(x) - min(x)]}$

Figure 10: Standardizasyon ve Normalizasyon.