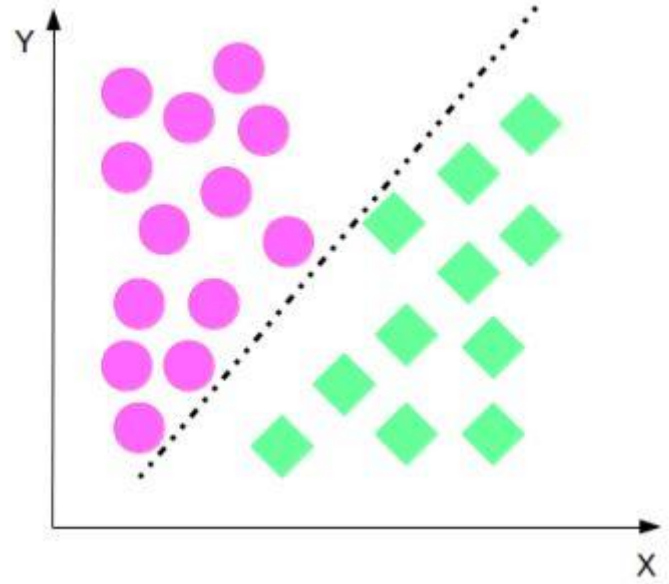
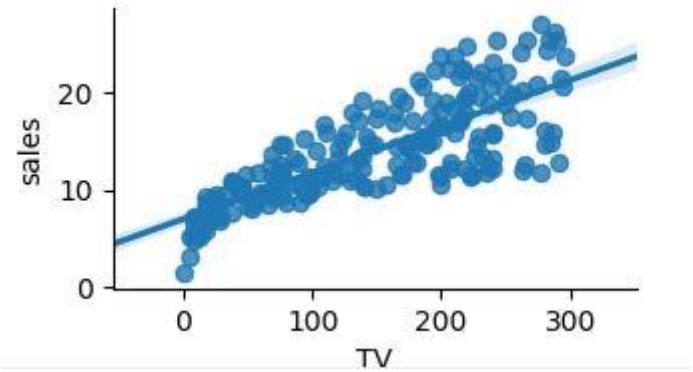
Bir önceki yazımızda gözetimli öğrenme, gözetimsiz öğrenme ve takviyeli öğrenme konularını anlatmış alt başlıklarına değinmiştik. Bu yazımızda bu alt başlıkları açıklayacağız.

1) **Gözetimli öğrenme iki grupta incelenir;**

Sınıflandırma (Classification Method) , veriler belli özelliklerine göre sınıflandırılırlar. Sınıflandırma, yapısal(structure) veya yapısal olamayan(unstructure) veriler üzerinde yapılabilir. Amacı her girdi vektörüne sonlu sayıdaki ayrık kategorilerden birini atama olan durumlar, sınıflandırma problemi olarak adlandırılır. Sınıflandırma problemini çözen öğrenme algoritmasına da sınıflandırıcı denir. Eğer sistem, hangi verinin, hangi koşullarda, hangi sınıfa ait olacağı bilgisi ile sınıflandırarak eğitilirse, yeni veri setindeki veriyi de öğrendiklerine benzer biçimde sınıflandırabilir.



Regresyon analizi (Regression Method), iki yada daha çok değişken arasındaki ilişkiyi ölçmek için kullanılan analiz metodudur. Eğer tek bir değişken kullanılarak analiz yapılıyorsa buna tek değişkenli regresyon, birden çok değişken kullanılıyorsa çok değişkenli regresyon analizi olarak isimlendirilir.

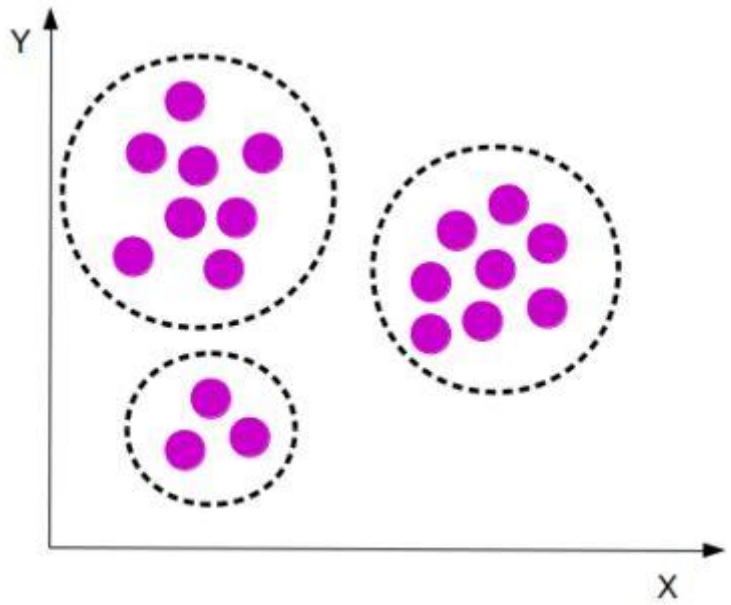


Regresyon problemleri, üretilen çıktının sürekli sayılardan oluştuğu durumlar için kullanılıyor. Örneğin, bir çalışanın işe geldiği gün içinde ürettiği ürün adedine göre ona sayısal bir verimlilik puanı oluşturmak isterseniz regresyon algoritmalarını kullanabiliriz. Regresyon algoritmaları gözetimli öğrenme kategorisinde olmasına rağmen anomaly detection (anormal durum

yakalama) gibi durumlarda hem gözetimli hem gözetimsiz öğremne yöntemlerinde kullanılabilir.

2) **Gözetimsiz öğrenme üç grupta incelenir;**

Kümeleme (Clustering), veri setindeki her bir verinin birbirlerine benzerlik durumlarına göre gruplara ayrılması işlemine denir. Her grup birer küme anlamına gelir. Her kümede birbirine en yakın veriler olmalı ve birbiriyle benzerlik göstermeyen verilerde mümkün olduğunca farklı kümelerde olmalıdır yani kümeler arası benzerliğin az olması gerekmektedir.



Çok fazla küme ile çalışmak birbirine çok benzeyen kümelerin oluşmasına sebep olabilir. Dolayısıyla elimizdeki veriye en uygun küme sayısını belirlemek gerekir. Temel amaç;

- Küme içindeki değerlerin birbirine en çok benzemesi
- Kümelerin ise birbirinden olabildiğince farklı olmasıdır.

Bu yüzden optimal bir küme sayısı belirtmek gerekir. Bu yüzden K-Means algoritması kullanılır. Optimal küme sayısını söyleyebilir.

Birliktelik kuralı (Association Rule Mining), veriseti içindeki geçmiş tarihli hareketlerin örüntülerini analiz eden ve birlikte gerçekleşme durumlarını veri madenciliği yöntemidir. Bu örüntülerden hareketle gelecekteki veriler için tahminleme yapabilir. Birliktelik kuralı ile ilgili en çok kullanılan algoritmalar; Apriori algorithm, Eclat algorithm, FP-growth algoritm. Bu algoritmaların hepsinin amacı veriler arasındaki bağıntıyı ortaya çıkarmaktır. Bu amaç için her biri kendi içerisinde farklı matematiksel işlemleri barındırır.

Not: Kümeleme ve Birliktelik kuralı yöntemleri arasındaki temel fark, kümeleme veri noktaları ile ilgilidir. Birliktelik kuralı ise bu veri noktalarının nitelikleri arasındaki ilişkileri bulmakla ilgilidir.

Boyut Azaltma (Dimensipnality Reduction), önemli bilgilerin hala iletildiğini garanti ederken bir veri kümesinin değişkenlerinin sayısını azaltmak anlamına gelir. Diğer bir değişle

amaç doğrultusunda en iyi sonucu verecek olan öznitelikler(feature) ile çalışmak için kullanışsız, gereksiz olan öznitelikleri çıkarmak, veri boyutunu azaltmak olarak nitelendirilebilir. Boyut azaltma, öznitelik çıkarımı (feature extraction) ve özellik seçimi (feature selection) yöntemleri kullanılarak yapılabilir.

- ❖ Öznitelik(feature): veriye ait herbir özelliğe verilen isimdir. (Örneğin ad, soyad, yas, doğum yeri)

Özellik seçimi, orijinal değişkenlerin bir alt kümesini seçerken öznitelik çıkarımı, yüksek boyutlu bir alandan düşük boyutlu bir alana veri dönüştürme işlemi gerçekleştirir. PCA algoritmasında öznitelik çıkarımı kullanılmıştır. Apriori, K-Means ve PCA, gözetimsiz (denetimsiz) öğrenme örnekleridir.

En çok Kullanılan Algoritmalar

- ✓ **Naive Bayes sınıflandırıcı:** Bir sınıftaki belirli bir özelliğin varlığının, başka herhangi bir özelliğin varlığına bağlı olmadığını varsayar. Verileri olasılık ilkeleri ile hesaplayarak sınıflandıran bir sınıflandırma algoritmasıdır. Örneğin binlerce makalenin hangi alanda yazıldıklarını kategorize etmek istiyoruz. Bunun için belli makalerde geçen belli kelimlerin olasılık değerlerinin, diğerlerine oranla fazla olması durumuna göre o makalenin hangi kategoriye ait olduğubu öğrenmek istersek (yani sağlık kelimesinin çok geçtiği makale tıpla ilgilidir) bu algoritma işe yarayabilir.
- ✓ **K-Nearest Neighbours(K-en yakın komşu):** Veri hangi veriye daha çok yakındır? mantığı ile dallanır. Sınıflandırma sırasında çıkarılan özelliklerden sınıflandırılmak istenen yeni bireyin daha önceki bireylerden k tanesine yakınlığına bakılır. Örneğin k=3 olsun. Yeni bir eleman sınıflandırılmak istenirse eski sınıflandırılmış elemanlardan en yakın üç tanesi alınır. Bu elemanlar hangi sınıfa dahilse yeni elemanlarda o sınıfa dahil edilir. Uzaklık hesapı için "Euclidean" uzaklığı, "Manhattan" uzaklığı, "Minkowski" uzaklığı kullanılabilir. Bu algoritmanın uygulaması kolaydır ve gürültü eğitim verisine (noisy train data) dayanıklıdır. Fakat bunun yanında dezavantajında vadır. Örneğin uzaklık hesabı yaparken bütün durumları sakladığından, büyük veriler için kullanıldığında çok sayıda bellek alanına ihtiyaç duyar.
- ✓ **Decision Tree(Karar Ağacı):** Veriler sınıfları ile birlikte bu algoritmaya verildiğinde algoritma verileri sınıflandırmak için kullanılabilecek bir dizi kural üretir.m karar düğümleri (dimension node) ve yaprak düğümleri (leaf node) olan bir ağaç yapısına sahiptir. Kullanılan ağaç yapıları görselleştirilebilir. Az oranda veri hazırlığına ihtiyaç duyar fakat kayıp değerleri desteklememektedir. Hem sayısal hem kategorik değişkenleri işleyebilir. Hem sınılandırma hem regresyon yönteminde kullanılabilir. Avantajları olduğu gibi dezavabtajları da vardır. Veriyi iyi bir şekilde açılamayan aşırı karmaşık ağaçlar üretebilir. Veriyi ezberleme-aşırı öğrenme (**over-fitting**) yaşanabilir. Bu problemin çözümü için parametrelerde kıstlamalar ve budama gibi yöntemler kullanılabilir.
- ✓ **Random Forest:** Sınıflandırma işlemi sırasında birden fazla karar ağacı kullanılarak sınıflandırma değerinin yükseltilmesini hedefleyen bir algoritmadır. Hiper parametre kestirimi yapılmadan da iyi sonuclar verir. Hem regresyon problemlerinde hemde sınıflandırma problemlerinde kullanılır. Karar ağaçlarının en büyük problemlerinden biri olan aşırı öğrenme problemini çözmek için hem veri setinden hem de öznitelik setinden rassal olarak farklı alt-setler seçer ve bunları eğitir. Bu yöntemle bir çok karar ağacı oluşturur ve her bir karar ağacı bireysel olarak tahminde bulunur. Eğer

problemimiz sınıflandırmaysa tahminler arasında en çok oy alan seçilir fakat probleminiz regresyonsa karar ağaçlarının tahminlerinin ortalaması alınır.

- ✓ **Destek Vector Machine(Destek Vektor Makinesi):** Veri setinde birbirine benzeyen gruplar arasına birbirinden en uzak olan noktalardan sınırlar çizmeye yarayan algoritmadır. Temel olarak iki sınıfa ait verileri birbirinden en uygun şekilde ayırmak için kullanılır. Bunun için karar sınırları yada diğer bir ifadeyle hiper düzlemler belirlenir. Boyut sayısının, gözlem sayısından fazla olduğu durumlarda etkilidir. Karar fonsiyonunda bir takım eğitim noktaları kullanılır. Dolayısıyla bellek verimli şekilde kullanılmış olur. Destek vektor makineleri, veri setinin doğrusal olarak ayrılabilme ve ayrılamama durumuna göre ikiye ayrılmaktadır.
- ✓ **Linear Regression (Doğrusal Regresyon):** Sayısal girdi ve çıktılar arasındaki doğrusal ilişkiyi tespit etmeyi sağlar. Düzlemde yayılmış verinin modelini en iyi biçimde doğrusal olarak çıkartmaya çalışan yöntemdir.
- ✓ **Logistic Regression:** Lojistik regresyon, yalnızca iki değere sahip olabilen bir sonucun olasılığını öngörür (yani, ikiye bölünebilir). Tahmin, bir veya birkaç öngörücünün (sayısal ve kategorik) kullanımına dayanır. Doğrusal regresyon evet/hayır, var/yok gibi binary(ikili) sistemde ifade edilebilecek değerler için uygun değildir. Çünkü,0 ve 1 aralığının dışında değer tahmin edebilir.

  Lojistik regresyon, 0 ile 1 arasındaki değerlerle sınırlı lojistik eğrisi üretir. Lojistik regresyon lineer bir regresyona benzer, ancak eğri olasılık yerine hedef değişkenin olasılıkları' nın doğal logaritması kullanılarak oluşturulur.

- ❖ Doğrusal Regresyon ve Lojistik Regresyon Karşılaştırılması