ANKARA
HACI BAYRAM VELİ ÜNİVERSİTESİ
UZAKTAN EĞİTİM UYGULAMA VE ARAŞTIRMA MERKEZİ

YAPAY ZEKÂ TEKNOLOJİLERİ VE YÖNETİMİ

www.hacibayram.edu.tr/uzem

5. Makine Öğrenmesi

5.1. Denetimli Öğrenme Yaklaşımları

5.1.1 Karar Ağaçları

Karar ağaçları, sınıflandırma ve regresyon problemlerinde kullanılan, ağaç benzeri bir yapıdaki modellerdir. Bu modeller, veri setini farklı özellikler kullanarak bir dizi karar kuralı ile bölerek, veri noktalarını sınıflandırır veya sürekli bir değer tahmin eder. Karar ağaçları, her bir düğümde bir özelliği değerlendirir ve bu değerlendirme sonucunda veriyi iki veya daha fazla alt kümeye ayırır. Bu süreç, her bir dalın veri setini daha homojen gruplara ayırana kadar devam eder ve her yaprak düğüm, tahmin edilen sınıf veya değeri temsil eder.

Karar ağaçlarının en büyük avantajlarından biri, modelin anlaşılabilirliğidir. Ağaç yapısı, modelin karar verme sürecini şeffaf hale getirir ve kullanıcıların modelin nasıl çalıştığını ve hangi özelliklerin daha etkili olduğunu kolayca anlamasını sağlar. Ayrıca, karar ağaçları veri ön işleme gereksinimlerini minimize eder ve hem sayısal hem de kategorik verilerle çalışabilir. Bu, onları çeşitli uygulama alanları için uygun hale getirir.

Bununla birlikte, karar ağaçları bazı sınırlamalara da sahiptir. Özellikle, küçük değişikliklere karşı hassas olabilirler ve veri setlerinde aşırı uyuma (overfitting) eğiliminde olabilirler. Aşırı uyum, modelin eğitim verilerine çok iyi uyum sağlaması ancak yeni veriler üzerinde kötü performans göstermesi durumudur. Bu sorunu önlemek için, karar ağaçları genellikle budama (pruning) teknikleri ve ensemble yöntemleri (örneğin, rastgele ormanlar ve gradyan artırma) ile desteklenir. Budama, ağaçların aşırı dallanmasını engellerken, ensemble yöntemleri birden fazla ağaç kullanarak daha sağlam ve genelleştirilmiş sonuçlar elde eder.

Karar ağaçları, çeşitli alanlarda geniş bir uygulama yelpazesi sunar. Örneğin, finansal risk analizi, tıbbi teşhis, müşteri segmentasyonu ve pazarlama stratejileri gibi birçok alanda kullanılır. Her ne kadar basit yapısı ile başlangıçta anlaşılması kolay olsa da, modelin performansını artırmak ve genelleme yeteneğini geliştirmek için daha karmaşık stratejiler ve optimizasyon teknikleri gerektirebilir. Bu nedenle, karar ağaçları

genellikle daha büyük ve daha karmaşık modellerin bir parçası olarak kullanılır.

5.1.2 Rastgele Orman

Rastgele Orman, sınıflandırma ve regresyon problemlerinde kullanılan bir ensemble öğrenme yöntemidir ve birden fazla karar ağacının birleşimiyle oluşturulan güçlü bir modeldir. Rastgele ormanlar, çeşitli karar ağaçlarının birleşimiyle çalışır ve her bir ağacın farklı bir alt veri kümesi ve özellik seti üzerinde eğitilmesini sağlar. Bu yöntem, her bir ağacın kararlarını bir araya getirerek son tahminleri oluşturur, böylece genel modelin doğruluğunu artırır ve overfitting (aşırı uyum) riskini azaltır.

Rastgele ormanların temel avantajlarından biri, yüksek genelleme kapasitesidir. Birden fazla karar ağacının birleşimi, modelin genelleştirme yeteneğini artırır ve veri setindeki rastgelelikleri ve gürültüleri daha iyi yönetir. Her bir ağaç, eğitim veri setinin rastgele bir alt kümesi üzerinde oluşturulur ve her düğümde rasgele seçilen bir özellik alt kümesi kullanılarak bölünür. Bu çeşitlendirme, modelin daha sağlam ve etkili olmasını sağlar. Ayrıca, rastgele ormanlar, hem sayısal hem de kategorik verilerle etkili bir şekilde çalışabilir ve veri ön işleme gereksinimlerini azaltır.

Rastgele ormanların hesaplama maliyeti, özellikle büyük veri setleri ve çok sayıda ağaç kullanıldığında artabilir. Modelin performansını artırmak için ağaç sayısının artırılması, genellikle daha iyi sonuçlar sağlar, ancak bu durum modelin eğitim süresini ve tahmin sürelerini uzatabilir. Bu nedenle, verimli hesaplama ve uygun parametre ayarlamaları yapmak önemlidir. Ayrıca, modelin şeffaflığı karar ağaçlarına kıyasla daha düşüktür; çünkü birden fazla ağaç kullanıldığı için, modelin genel karar verme sürecini anlamak daha zor olabilir.

Rastgele ormanlar, çeşitli uygulama alanlarında başarıyla kullanılmaktadır. Finansal risk analizi, sağlık teşhisleri, müşteri segmentasyonu ve biyoinformatik gibi birçok alanda geniş bir uygulama yelpazesi sunar. Yüksek performansı ve esnekliği sayesinde, veri analizi ve modelleme süreçlerinde sıklıkla tercih edilen bir yöntemdir. Rastgele ormanlar, genellikle büyük veri setleriyle başa çıkabilme yeteneği ve overfitting riskini azaltma avantajları ile öne çıkar.

5.1.3 Naive Bayes

Naive Bayes, sınıflandırma problemlerinde kullanılan basit ama etkili bir olasılık tabanlı makine öğrenmesi algoritmasıdır. Bu

yöntem, Bayes teoremi ve özelliklerin birbirinden bağımsız olduğu varsayımına dayanır. Naive Bayes, verilerin her bir özelliğinin bağımsız olduğunu varsayar ve bu bağımsızlık varsayımı altında sınıflandırma yapar. Örneğin, bir e-postanın spam olup olmadığını belirlemek için, e-postadaki kelimelerin her birinin spam sınıfını tahmin etmeye etkisi ayrı ayrı değerlendirilir. Bu, modelin hesaplama sürecini büyük ölçüde basitleştirir ve hızlandırır.

Naive Bayes algoritması, genellikle Bayes teoremi kullanılarak çalışır. Bayes teoremi, bir sınıfın verilmiş bir özellik seti altında olma olasılığını hesaplar. Özelliklerin bağımsızlığı varsayımı, modelin olasılık hesaplamalarını kolaylaştırır, bu da özellikle büyük veri setlerinde ve yüksek boyutlu özellikler ile çalışırken büyük avantaj sağlar. Bu yöntem, eğitim verisinden her özelliğin sınıf üzerindeki etkisini öğrenir ve yeni veriler üzerinde bu bilgiyi kullanarak tahminlerde bulunur.

Naive Bayes'in avantajlarından biri, hızlı ve düşük hesaplama maliyetli olmasıdır. Özelliklerin bağımsızlık varsayımı, modelin eğitim ve tahmin sürecini oldukça verimli hale getirir. Ayrıca, Naive Bayes, genellikle küçük veri setleriyle veya eksik verilerle çalışabilen ve hızlı bir şekilde sonuçlar üretebilen bir yöntem olarak kabul edilir. Ancak, özellikler arasındaki bağımlılıkları göz ardı etmesi, bazı durumlarda modelin performansını sınırlayabilir. Özelliklerin bağımsız olmadığı durumlarda, Naive Bayes'in tahmin doğruluğu etkilenebilir.

Naive Bayes, metin sınıflandırma, spam filtrasyonu, duygu analizi ve tıbbi teşhis gibi birçok alanda geniş bir uygulama yelpazesi sunar. Özellikle metin sınıflandırma problemlerinde, Naive Bayes'in kelimeler arasındaki bağımsızlık varsayımı iyi sonuçlar verebilir. Modelin basit yapısı ve hızlı hesaplama yeteneği, onu büyük veri setlerinde ve gerçek zamanlı uygulamalarda tercih edilen bir seçenek yapar. Bu özellikleri, Naive Bayes'i veri bilimi ve makine öğrenmesi alanında değerli bir araç haline getirir.

5.2. Karışıklık Matrisi

Karışıklık matrisi (Confusion Matrix), bir sınıflandırma modelinin performansını değerlendirmek için kullanılan temel bir araçtır. Bu matris, modelin tahmin ettiği sınıflarla gerçek sınıflar arasındaki ilişkiyi özetler. Genellikle dört ana bileşenden oluşur: doğru pozitifler (True Positives - TP), doğru negatifler (True Negatives - TN), yanlış pozitifler (False Positives - FP) ve yanlış negatifler (False Negatives - FN). Her bir bileşen, modelin sınıflandırma sonuçlarını değerlendirirken ne kadar doğru veya yanlış tahminde bulunduğunu gösterir.

Doğru pozitifler (TP), modelin doğru bir şekilde pozitif sınıfa ait olduğunu tahmin ettiği veri noktalarını temsil eder. Doğru negatifler (TN), modelin doğru bir şekilde negatif sınıfa ait olduğunu tahmin ettiği veri noktalarını ifade eder. Yanlış pozitifler (FP), modelin yanlış bir şekilde pozitif sınıfa ait olduğunu tahmin ettiği negatif veri noktalarını gösterir. Yanlış negatifler (FN) ise, modelin yanlış bir şekilde negatif sınıfa ait olduğunu tahmin ettiği pozitif veri noktalarını temsil eder. Bu dört bileşen, modelin genel performansını detaylı bir şekilde analiz etmeye olanak sağlar.

Karışıklık matrisi, özellikle dengesiz veri setlerinde modelin performansını daha ayrıntılı bir şekilde analiz etme imkanı sunar. Dengesiz veri setlerinde, bazı sınıfların sayısal olarak diğerlerinden çok daha fazla olması durumunda, doğruluk metriği yanıltıcı olabilir. Karışıklık matrisi, yanlış pozitif ve yanlış negatif oranlarını ayrı ayrı gösterdiği için, modelin her sınıf üzerindeki performansını daha iyi anlamak ve iyileştirmek için bu bilgileri kullanabilirsiniz. Bu nedenle, karışıklık matrisi, modelin güçlük çektiği alanları belirlemeye ve performansı optimize etmeye yardımcı olur.

Karışıklık matrisi, performans değerlendirmesi için çeşitli metrikler hesaplamaya yardımcı olur. Örneğin, doğruluk (accuracy), modelin doğru tahminlerinin toplam tahminlere oranı olarak hesaplanır. Hassasiyet (precision), pozitif olarak tahmin edilen veri noktalarının ne kadarının gerçekten pozitif olduğunu ölçerken, duyarlılık (recall), gerçek pozitiflerin ne kadarının model tarafından doğru tahmin edildiğini gösterir. Ayrıca, F1 skoru gibi harmanlanmış metrikler de, hassasiyet ve duyarlılığın bir kombinasyonu olarak model performansını değerlendirmede kullanılır.

5.2.1 Performans Ölçütleri

Karışıklık matrisi, sınıflandırma modellerinin performansını değerlendirmek için birçok önemli ölçüt sağlar. Bu ölçütler, modelin doğruluğunu, hassasiyetini, duyarlılığını ve genel başarısını anlamak için kullanılır. İşte karışıklık matrisinden türetilen bazı temel performans ölçütleri:

Doğruluk (Accuracy): Modelin tüm tahminlerinin doğruluk oranını ölçer. Hesaplanma şekli, doğru tahminlerin toplam tahminlere oranıdır. Formülü şu şekildedir:

$$\text{Doğruluk} = \frac{\text{Doğru Pozitifler (TP)} + \text{Doğru Negatifler (TN)}}{\text{Toplam Örnek Sayısı}}$$

Doğruluk, genel model performansını özetler, ancak dengesiz veri setlerinde yanıltıcı olabilir.

Hassasiyet (Precision): Pozitif olarak tahmin edilen veri noktalarının ne kadarının gerçekten pozitif olduğunu ölçer. Özellikle yanlış pozitifleri minimize etmek için önemlidir. Formülü şu şekildedir:

$$\text{Hassasiyet} = \frac{\text{Doğru Pozitifler (TP)}}{\text{Doğru Pozitifler (TP)} + \text{Yanlış Pozitifler (FP)}}$$

Hassasiyet, yanlış pozitiflerin etkisini azaltmak için kullanılır ve özellikle kritik uygulamalarda önemlidir.

Duyarlılık (Recall) veya Gerçek Pozitif Oranı: Gerçek pozitiflerin ne kadarının model tarafından doğru tahmin edildiğini ölçer. Formülü şu şekildedir:

$$\text{Duyarlılık} = \frac{\text{Doğru Pozitifler (TP)}}{\text{Doğru Pozitifler (TP)} + \text{Yanlış Negatifler (FN)}}$$

Duyarlılık, modelin pozitif sınıfları kaçırma oranını değerlendirir ve özellikle pozitif sınıfların önemli olduğu durumlarda önemlidir.

F1 Skoru: Hassasiyet ve duyarlılığın harmonik ortalamasıdır ve her iki metriği dengeler. Özellikle sınıflar arasında bir denge sağlamak için kullanılır. Formülü şu şekildedir:

$$\text{F1 Skoru} = 2 \times \frac{\text{Hassasiyet} \times \text{Duyarlılık}}{\text{Hassasiyet} + \text{Duyarlılık}}$$

F1 skoru, özellikle dengesiz veri setlerinde veya her iki metriğin de önemli olduğu durumlarda etkili bir performans ölçütüdür.

5.3. Overfitting (Aşırı Uyum)

Overfitting (aşırı uyum), bir makine öğrenmesi modelinin eğitim verilerine aşırı derecede uyum sağlaması durumunu ifade eder. Bu durumda, model eğitim verileri üzerindeki hataları minimuma indirir, ancak yeni veya bilinmeyen veriler üzerinde kötü performans gösterir. Overfitting, modelin eğitim verilerindeki gürültü ve detayları öğrenmeye çalışarak, genel veri desenlerini öğrenmek yerine, eğitim verilerinin özel örüntülerine odaklanması sonucunda ortaya çıkar. Bu, modelin genelleme yeteneğini azaltır ve gerçek dünyadaki yeni verilerle başa çıkma kapasitesini sınırlar.

Overfitting'i önlemek için çeşitli teknikler kullanılabilir. Düzenleme (regularization) yöntemleri, modelin karmaşıklığını sınırlayarak aşırı uyumu engeller; L1 ve L2 düzenleme gibi yöntemler, modelin ağırlıklarını ceza ekleyerek kontrol eder. Çapraz doğrulama (cross-validation), modelin genel performansını değerlendirmek için veri setini çeşitli alt kümelere ayırarak eğitim ve test sürecini tekrarlar, bu da overfitting riskini azaltır. Ayrıca, erken durdurma (early stopping), modelin eğitim sürecini izleyerek performansın test verilerinde düşüş göstermesi durumunda eğitim sürecini sonlandırarak aşırı uyumu önlemeye yardımcı olabilir. Bu teknikler, modelin hem eğitim hem de test verilerinde iyi performans göstermesini sağlamaya çalışır ve genel modelin doğruluğunu artırır.

5.4. Outlier (Aykırı Değer)

Outlier (aykırı değer), bir veri setinde diğer veri noktalarından önemli ölçüde farklı olan ve genellikle veri dağılımının dışında kalan verilerdir. Aykırı değerler, genellikle veri toplama hataları, ölçüm hataları veya doğal varyasyonlardan kaynaklanabilir. Bu tür değerler, istatistiksel analizlerin ve makine öğrenmesi modellerinin sonuçlarını etkileyebilir ve modelin performansını bozabilir. Örneğin, bir şirketin çalışan maaşları veri setinde çok yüksek bir maaş aykırı değer olarak kabul edilebilir, çünkü bu değer diğer maaşlardan oldukça uzak olabilir.

Aykırı değerlerin etkilerini azaltmak için çeşitli yöntemler uygulanabilir. Görselleştirme teknikleri (örneğin, kutu grafikleri veya scatter plotlar) aykırı değerlerin belirlenmesinde yardımcı olabilir. İstatistiksel yöntemler ise aykırı değerleri tanımlamak için z-skorları veya IQR (Interquartile Range) gibi metrikleri kullanabilir. Ayrıca, veri ön işleme teknikleri arasında aykırı değerlerin çıkarılması, dönüştürülmesi veya düzeltilmesi yer alabilir. Makine öğrenmesi algoritmaları, aykırı değerlerin etkisini minimize etmek için genellikle daha sağlam modeller veya aykırı değer tespiti algoritmaları kullanabilir. Aykırı değerlerin yönetimi, modelin doğruluğunu artırmak ve analizlerin güvenilirliğini sağlamak için kritik öneme sahiptir.

KAYNAKÇA

- Russell, S. J., & Norvig, P. (2016). Artificial intelligence: a modern approach. Pearson.
- Burkov, A. (2019). The hundred-page machine learning book (Vol. 1, p. 32).
- Quebec City, QC, Canada: Andriy Burkov.
- Smola, A. (2008). Introduction to machine learning. Link: https://alex.smola.org/drafts/thebook.pdf

BÖLÜM SORULARI

1) Karar ağacının yapısında her bir iç düğüm, hangi tür bilgiyi temsil eder ve her bir yaprak düğüm neyi temsil eder?

2) Rastgele Orman yönteminde, her bir karar ağacının nasıl oluşturulduğunu ve bu ağaçların nasıl bir araya getirildiğini açıklayınız.

3) Karışıklık matrisinde yer alan dört ana bileşeni tanımlayın ve her bir bileşenin sınıflandırma performans metriklerinin hesaplanmasındaki rolünü açıklayınız.

4) Overfitting nedir ve bir modelin aşırı uyum sağladığı durumlarda ne gibi sorunlar ortaya çıkabilir? Ayrıca, bu sorunu önlemek için hangi stratejiler kullanılabilir açıklayınız.