



RAPOR BAŞLIĞI

HAZIRLAYAN

ADI SOYADI : RABİA ŞENLİK

ÖĞRENCİ NUMARASI : 190303027

TESLİM TARİHİ : 18/12/2022

DERS ADI : MAKİNE ÖĞRENMESİ

DERS YÜRÜTÜCÜSÜ : SAEİD AGAHİAN

KARAR AĞAÇLARI

Karar ağaçları, verileri sınıflandırmak veya tahmin etmek için kullanılan popüler bir makine öğrenimi algoritmasıdır. Karar ağacı, verileri sınıflandırmak için bir seri "karar" noktasından oluşan bir ağaç şeklinde görselleştirilen bir model oluşturur. Her karar noktası, bir özniteliğe dayalı bir "eşik değeri" kullanılarak veri kümesini ikiye böler.

Örneğin, bir karar ağacının bir karar noktası veri kümesinde bir özniteliğin değerlerinin 5'ten küçük olma durumunda bir bölge ve 5'ten büyük olma durumunda başka bir bölge oluşturulacağı anlamına gelir. Bu bölgelerin her birinde ayrı ayrı bir karar ağacı oluşturulur ve bu süreç tekrarlanır. Bu şekilde, veri kümesi bir dizi karar noktaları tarafından parçalanarak, her veri noktasının sınıfına ait bir "kök dal"a ulaşılır. Karar ağacı eğitim sırasında, model veri kümesine uygulanır ve her karar noktası için en iyi eşik değeri seçilir. Eğitim tamamlandıktan sonra, karar ağacı tahmin yapmak için kullanılabilir. Yeni bir veri noktası modelin girişine verilerek, veri noktasının sınıfını tahmin etmek için model kullanılır. Bu, veri noktasının girdi özniteliklerine göre karar ağacının ağacını takip ederek yapılır ve veri noktasının sınıfına ait bir kök dal bulunana kadar devam eder.

Karar ağacı oluşturma sırasında bazı kısıtlamalar bulunabilir. Bunlar şunlardır:

1. **Öznitelik sayısı:** Karar ağacı oluştururken, özniteliklerin sayısı önemlidir. Çok fazla öznitelik kullanılması, modelin doğruluğunu düşürebilir ve ayrıca modelin anlaşılabilirliğini azaltabilir.
- 2.
3. **Eşit dağılım:** Karar ağacı oluştururken, veri kümesinde eşit dağılım olması önemlidir. Örneğin, eğer bir sınıfın örneklerinin çok fazla olması durumunda, model bu sınıfı çok iyi tahmin edebilir ancak diğer sınıfları tahmin etmekte zorlanabilir.
4. **Aşırı öğrenme:** Karar ağacı oluştururken, modelin veri kümesine çok iyi uyum sağlamasına (aşırı öğrenme) dikkat etmek gerekir. Aşırı öğrenme durumunda, model veri kümesine çok iyi uyum sağlar ancak yeni verilere uygulandığında performansı düşebilir.
5. **Eşik değerleri:** Karar ağacı oluştururken, özniteliklerin parçalanması için kullanılan eşik değerleri önemlidir. Eğer eşik değerleri çok fazla olursa, model çok detaylı hale gelebilir ve yeni verilere uygulandığında performansı düşebilir.
6. **Öznitelik önem dereceleri:** Karar ağacı oluştururken, özniteliklerin önem dereceleri de önemlidir. Eğer bazı özniteliklerin önem dereceleri diğerlerine göre çok yüksekse, model bu özniteliklere aşırı önem verir ve diğer özniteliklerin etkisini azaltabilir.

karar ağacı oluştururken 1 bölgede aynı feature'u başka eşik değerle parçalama şansımız var mı?

Evet, karar ağacı oluştururken bir bölgede aynı özniteliği farklı eşik değerleriyle parçalama şansınız vardır. Karar ağacı eğitim sırasında veri kümesi içinde mevcut öznitelikleri kullanarak en iyi bölme noktasını (eşik değerini) seçer. Bu eşik değerleri, veri kümesi içinde mevcut verilerin değerlerine göre değişebilir. Örneğin, veri kümesinde bir özniteliğin değerleri 0 ile 10 arasında değişebilir ve karar ağacı bu özniteliği 5 eşik değerinde parçalayabilir. Bu, veri kümesinde bu özniteliğin değerlerinin 5'ten küçük olma durumunda bir bölge ve 5'ten büyük olma durumunda başka bir bölge oluşturulacağı anlamına gelir. Eğer veri kümesi değişirse, bu özniteliğin parçalanma noktası da değişebilir ve daha farklı bir eşik değeri kullanılabilir.

İkili ağacın arama karmaşıklığı ve train aşamasındaki karmaşıklığı nedir?

İkili ağaçlar, veri kümelerini hiyerarşik olarak sınıflandıran veri yapılarıdır. İkili ağaçlarda, her düğüm bir veri noktasını temsil eder ve her düğümün iki alt düğümü vardır. Bu yapı sayesinde, veri kümesine ulaşılması gereken yerin hızlı bir şekilde bulunmasına yardımcı olur.

İkili ağaçların arama karmaşıklığı, genellikle $O(\log n)$ olarak değerlendirilir. Bu, veri kümesinin büyüklüğüne göre değişebilir, ancak genellikle veri kümesi büyüdükçe arama işlemlerinin hızı azalır. Bu, ikili ağaçların diğer veri yapılarından daha hızlı olduğu anlamına gelir. Örneğin, dizilerde arama işlemleri $O(n)$ karmaşıklığında olur, bu da dizilerin ikili ağaçlardan daha yavaş olduğu anlamına gelir.

İkili ağaçların train aşaması ise, ağacın oluşturulması işlemine denir. Bu aşama, veri kümesinin ikili ağaç yapısına dönüştürülmesi için kullanılır. Train aşamasının karmaşıklığı, veri kümesinin büyüklüğüne ve veri kümesinin nasıl dağıtılmış olduğuna göre değişebilir. Genellikle, veri kümesi daha düzenli bir şekilde dağıtılmışsa, ikili ağacın train aşaması daha az karmaşık olacaktır. Ayrıca, veri kümesinin büyüklüğü arttıkça, train aşaması daha fazla zaman alacaktır.

Özet olarak, ikili ağaçların arama karmaşıklığı $O(\log n)$ olarak değerlendirilir ve train aşaması veri kümesinin büyüklüğüne ve dağılımına göre değişebilir.