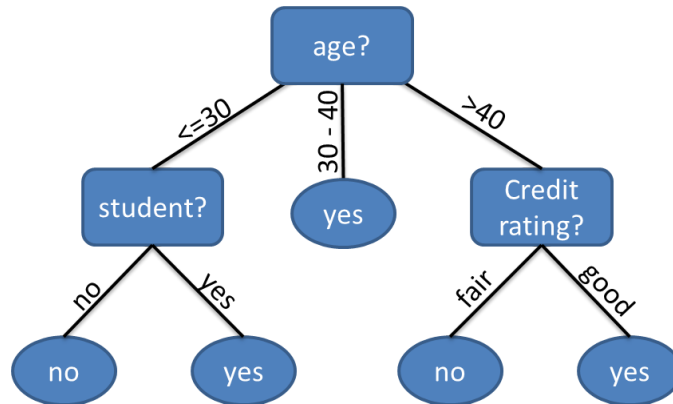




Karar ağacı (Decision Tree)

Karar ağaçları, Sınıflandırma ve Regresyon problemlerinde kullanılan denetimli makine öğrenmesi kategorisinde, ağaç tabanlı algoritmadan biridir. Karmaşık veri setlerinde kullanılabilir. Aynı zamanda bir Topluluk öğrenmesi (Ensemble learning) yöntemi olan Rassal ormanında temelini oluştururlar.

Karar ağaçları nasıl çalışır? (How do decision trees work?)



Karar ağaçları düğümlerinde if-else yapısına benzer bir yapı bulundurarak yani koşulları göze alarak sonuca ulaşmayı ve yapraklara inerek sınıflandırma yapar.

Karar ağaçlarının sınıflandırma için kullanımı daha uygundur. Regresyon için kullanılan karar ağacında da sonuç sınıflandırma yapmaya varır. Makine kendi sınıflarını ortalamalara göre oluşturur.

Bu konunun anlatımında sıklıkla kullanılan bir veri setini kullanacağız.

Veri setinin tanınması

day	outlook	temp	humidity	wind	play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Veri setimiz bir günün hava durumuna göre tenis oynanıp oynanamayacağını gösteren bir veri setidir.

Verilerde görüldüğü üzere bütün veriler kategorik olarak verilmiş. **Verilerin kategorik olması karar ağaçları için herhangi bir sorun teşkil etmez.** Ancak kullanmış olduğumuz kütüphane (sci-kit learn) kategorik veri kullanmaz, yalnızca sayısal veriler üzerinden algoritmalar üretir. Bu yüzden verilerimizi encode ederek eğitmemiz gerekir. Bağımlı değişkenimiz ise play kolonu olacaktır.

Karar Ağaçları nasıl oluşturulur?

Karar ağaçları oluşturmak için birbirinden farklı yöntemler mevcuttur. Burada işlenecek olanlar Entropi ve Gini olacaktır.

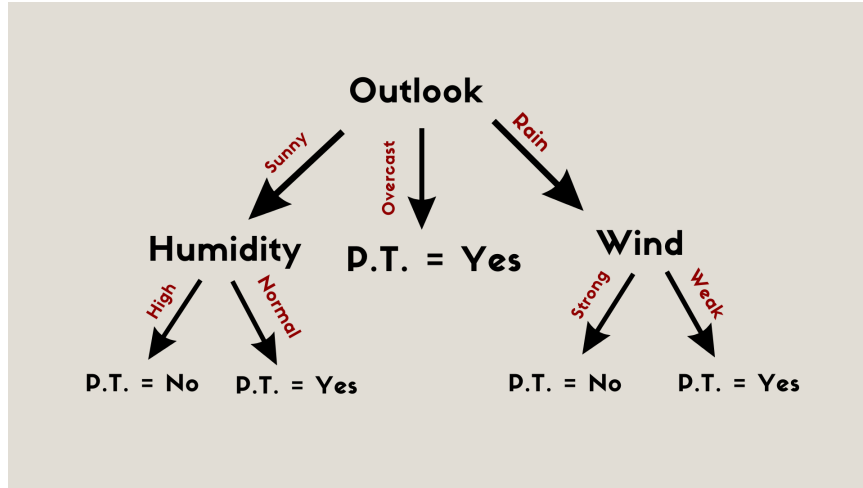
Bu seçimi pythonda nasıl yapacağınızı kod kısmında paylaşacağız.

1.Adım - Teorik anlatım

Karar ağaçlarında ilk soru **Kök** diğer sorular **Düğüm** ve sınıflandırmanın yapıldığı uçlara ise **Yaprak** adı verilir.

Makine Karar ağacını oluştururken en yüksek şekilde düğüm oluşturmayı amaçlar.

Örneğin ilk soru olarak **Outlook** kolonu ele alınmış.



Outlook kolonu 3 adet düğüm oluşturmuş. Havanın overcast (Bulutlu) olduğu durumlarda makine direkt olarak tenis oynanabilir yorumunu yapar. Çünkü verimizde hava bulutlu iken diğer değerler değişmesine rağmen daima tenis oynanabilir.

2.Adım - Matematiksel Anlatım

Karar ağaçları kullanılırken belli başlı bölme yöntemleri ele alınır. Bunlardan bizim ele alacağımız yöntem **Entropy** yöntemi olacak. Bide gini yöntemi var tabi. **Entropy verileri içersindeki mevcut olan belirsizlik miktarıdır.**

Şimdi matematiksel hesaplamaya geçelim.

Entropi hesabı

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Entropi hesabı için yukarıda belirttiğimiz formül kullanılır. Burada p_{+} ifadesi verilerin pozitif olma olasılığını (örneğin 9 Yes 4 No varsa 9/13) p_{-} ise verilerin negatif olma olasılığını belirtir.

Enformasyon Kazanımı hesabı

$$Information\ Gain(S, A) = H(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} H(S_v)$$

Information Gain (Enformasyon kazanımı) değeri her bir kolon için ayrıca hesaplanır. En büyük değeri veren kolon köke yerleştirilebilir. Diğer kolonlar sırasıyla düğümlere yerleştirilir ve karar ağacı tamamlanır.

Pythonda Decision Tree

```
#Öncelikle gerekli kütüphane dahil edilir
from sklearn.svm import DecisionTreeClassifier
#Ardından DecisionTreeClassifier sınıfından bir nesne oluşturulur
Dtc = DecisionTreeClassifier(criterion = "entropy",penalty = "l2" , c = 10.0)
#Eğitim işlemi yapılır
Dtc.fit(xtrain,ytrain)
#Eğitim yapılan veri seti üzerinden tahmin işlemi yapılır
tahmin = Dtc.predict(xtest)
#Karşılaştırma için karmaşıklık matrisi kullanılabilir
from sklearn.metrics import confusion_matrix
print(confusion_matrix(tahmin,ytest))
#Oluşturduğumuz karar ağacını tablo olarak görselleştirmek için
from sklearn.tree import plot_tree
plot_tree(dtc)
#Oluşturduğumuz karar ağacını yazı olarak görselleştirmek için
from sklearn.tree import export_text
export_text(dtc)
```

Mert Aydoğan - Gazi ME