**C4.5 Algoritması Nedir?**

**· J. Ross Quinlan tarafından geliştirilmiş ve ID3 algoritmasının bir geliştirilmiş versiyonudur. Veri kümesinden anlamlı karar ağaçları oluşturmayı amaçlar.**

**· ID3 algoritmasındaki bilgi kazancı ve entropi hesaplamalarını kullanır.**

***ID3 algoritmasından en büyük farkı ‘sayısal-numerik sütun’ içerebilmesi ve eksik değer ile çalışabilmesidir.***

**Karar Ağacı Nedir?**

**· Veriyi sınıflandırmak veya tahmin yapmak için kullanılan, bir dizi karar kuralına dayalı yapısal bir modeldir. Karar ağacında önemli olan terimler:**

* **Düğümler (Nodes): Karar ağacının herhangi bir seviyesinde yer alan ve bir testin yapıldığı noktadır. Bir ağacın içinde, her iç düğüm (yani kök düğüm hariç) veri kümesini belirli bir özelliğe göre böler.**
* **Dallar (Branches): Bir düğümdeki testin sonucunu temsil eder ve test sonucuna göre yönlendiren bir bağlantıdır.**

**“Yaş> 30 mu?” bir düğümü temsil ederken ‘Evet’ ve ‘Hayır’ gibi cevaplar da dalları temsil eder.**

* **Kök Düğüm (Root Node): Karar ağacının en üst düzeyindeki düğümdür. Burada, veriye ait en önemli özellik veya kriter test edilir.**

**Entropi Nedir?**

**· Bir veri kümesinin karışıklığını veya düzensizliğini ölçen bir kavramdır. Temelde veri kümesinin ne kadar “belirsiz” veya “düzensiz” olduğunu gösterir. Yani verideki rastgeleliğini ölçer. Eğer bir sınıfa ait ise sınıf içindeki düzensizlik azalırsa veri daha düzenli olur ve entropi düşük olur.**

**metin, yazı tipi, beyaz, tipografi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Entropi Hesaplamasının Formülü / (Pi) = Durumun olasılığı**

* **Düşük Entropi: Veri daha düzenlidir ve belirli sınıflara ayrılmak daha kolaydır. Örneğin, veri kümesinde çoğunlukla bir sınıf varsa, o zaman entropi düşer.**
* **Yüksek Entropi: Veri daha dağılmış ve belirsizdir. Veri kümesindeki sınıflar eşit dağıldığında entropi yüksek olur, çünkü sınıflar arasındaki ayrım net değildir.**

**ID3 ile C4.5 Algoritması Arasındaki Farklar**

**1- Sürekli (Numerik) Verilerle Çalışma Yeteneği**

* **ID3 Algoritması, sadece kategorik (ayrık) veri türleriyle çalışabilir.**
* **C4.5 Algoritması, kategorik veri türleriyle beraber sürekli veri türleriyle doğrudan çalışabilir.**

**2- Eksik Verileri İşleyebilme**

* **ID3 Algoritması, Eksik verileri yönetemez ve eksik veriler içeren veri kümeleriyle çalışmak zorlaşır.**
* **C4.5 Algoritması, eksik veri durumlarını işleyebilir. Eksik veri içeren kayıtların ağırlıklarını dikkate alarak bilgi kazancı oranını hesaplar.**

**3-Budama (Pruning) Özelliği**

* **ID3 Algoritmasında karar ağacı oluştururken aşırı dallanmaya yol açabilir ve bu, overfitting (aşırı öğrenme) sorununa neden olabilir.**
* **C4.5 Algoritması karar ağacı oluşturulduktan sonra gereksiz veya fazla dalları kesmek için budama işlemi gerçekleştirir. Bu işlemle karar ağacının genellenebilirliğini artırır ve aşırı öğrenmeyi önler.**

**Bilgi Kazancı ve Bilgi Kazancı Oranı**

**Bilgi Kazancı*= ‘Entropi(Karar Sınıfı) — Entropi(X\_Sütun)’ işlemi ile hesaplanır.***

***Bilgi Kazancı Oranı = (Bilgi Kazancı)/(Bölünme Oranı)*ile hesaplanır.**

***T: Veri kümesi içindeki eleman sayısı.***

***Ti: Seçilen veri sınıfının içindeki eleman sayısı***

****

**[1] SI= Split İnfo — Bölünme Bilgisi**

**! Bölünme Bilgisi(Oranı) Entropi formülünün aynısıdır. Bu yüzden bilgi kazancı oranı için (Bilgi Kazancı)/(Karar Sınıfı Entropi) ile de hesaplanabilir.**

**Eksik Veriler İle Çalışabilme**

**Eksik veri ile yaparken de ağırlıkları ile çarparak o veriyi işleme kazandırmış oluyorsun.**

***F = Değeri bilinen sınıf üyeleri sayısı / Tüm sınıf eleman sayısı***

**Bilgi Kazancı = F x (Entropi(Karar Sınıfı) — Entropi(X\_Sütun))**

**şeklinde hesaplanıyor.**

**! Gözden kaçırmaman gerek: Normal entropi hesaplanırken de toplam eleman sayısı eksik eleman çıkartılınca oluşan eleman sayısıdır.**

**Örnek Uygulama**

**metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**[2] Hava Durumuna Göre Futbol Oynayıp Oynayamama Durumu**

**1- Sayısal Değerler Kategorik Değerlere Çevrilir.**

**Sayısal değerler içeren sütunların medyanları hesaplanır. Tekrar eden elemanları çıkartarak medyan hesabı yapılmalı. Tek sayıda ise ortadaki eleman, çift ise ortadaki iki elemanın ortalaması medyan olacak şekilde hesaplanır.**

**Temperature (Sıcaklık İçin):**

**64, 68, 69, 70, 71, 72, 75, 80, 81, 83, 85**

**Medyan = 72**

**Oluşan medyana göre ‘72 Altında’ veya ‘72 Üzerinde’ olacak şekilde Sıcaklık sütununu sınıflandırılmalı.**

**Humidty (Nem İçin):**

**65, 70, 75, 78, 80, 85, 90, 95, 96**

**Medyan = 80**

**Oluşan medyana göre ‘80 Altında’ veya ‘80 Üzerinde’ olacak şekilde Nem sütunu sınıflandırmalı.**

**metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**[3] Tablomuzun Kategorik Değişkenlere Dönüştürülmüş Hali**

**2- Entropiler ve Bilgi Kazançları Hesaplanır**

**yazı tipi, çizgi, beyaz, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Entropi Play = 0.9402**

**Outlook — Hava Durumu — İçin**

**1-Sunny -Güneşli- : 6 adet veri güneşlidir. Bu verilerin dördü ‘Hayır’, ikisi ‘Evet’ olduğu için buna göre entropi hesabı yapılır.**

**yazı tipi, beyaz, çizgi, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**E(Sunny|Outlook) = 0.9182**

**2-Overcast -Bulutlu-: 4 adet veri bulutludur. Dördü de evet olduğu için entropisi sıfırdır.**

**! Eğer tüm üyeler aynı sınıf üyesi ise Entropi *SIFIR.***

**! Eğer eşit dağılımda sınıf üyeleri var ise Entropi *BİR.***

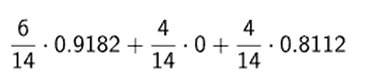
**3-Rainy-Yağmurlu-: 4 adet veri yağmurludur. Bu verilerin biri ‘Hayır’, üçü ‘Evet’ olduğu için buna göre entropi hesabı yapılır.**

**yazı tipi, beyaz, çizgi, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**E(Rainy|Outlook) = 0.8112**

**Tüm entropiler hesaplandığına göre;**

****

**Entropi(Outlook) = 0.6252**

***Bilgi Kazancı = Entropi(Play)- Entropi(Outlook) = 0.9402–0.6252 = 0.315***

**Temperature — Sıcaklık — İçin**

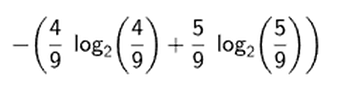
**1- ‘72 Altında’ — ‘Below 72’: 5 adet veri 72 değerin altındadır. Biri ‘Hayır’ dördü ‘Evet’ değerindedir.**

**yazı tipi, çizgi, beyaz, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

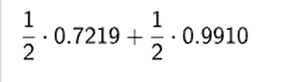
**Entropi(Below72|Temperature) = 0.7219**

**2- ’72 Üzerinde’ — ‘Above 72' :9 adet veri 72 değerin üzerindedir. Dördü ‘Hayır’ Beşi ‘Evet’ karar sınıfı üyeleridir.**

****

**Entropi(Above72|Temperature) =0.9910**

**Tüm entropiler hesaplandığına göre;**

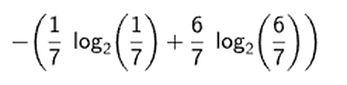
****

**Entropi(Temperature)=0.8564**

***Bilgi Kazancı = Entropi(Play)- Entropi(Temperature) = 0.9402–0.8564= 0.0838***

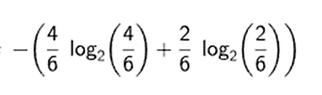
**Humidty — Nem — İçin**

**1- ’80 Altında’ — ‘Below 80’: 8 adet veri 80 değerin altındadır. Biri ‘Hayır’ yedisi ‘Evet’ değerindedir.**

****

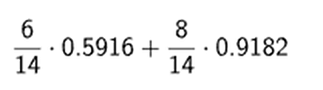
**Entropi(Below80|Humidity) = 0.5916**

**2- ‘80 Üzerinde’ — ‘Above 80’: 6 adet veri 80 değerin üzerindedir. Dördü ‘Hayır’ ikisi ‘Evet’ sınıfına üyedir.**

****

**Entropi(Above80|Humidity) = 0.9182**

**Tüm entropiler hesaplandığına göre;**

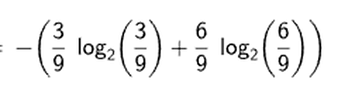
****

**Entropi(Humidity) = 0.7782**

***Bilgi Kazancı = Entropi(Play)- Entropi(Humidty) = 0.9402–0.7782= 0.162***

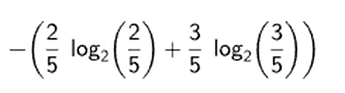
**Windy — Rüzgar Durumu — İçin**

**1- Weak-Zayıf-: 9 adet veri rüzgar hafif durumdadır. Üçü ‘Hayır’, altısı ‘Evet’ kabul sınıfı üyesidir.**

****

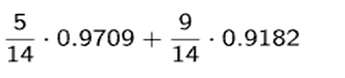
**Entropi(Weak|Windy) = 0.9182**

**2- Strong-Kuvvetli-: 5 adet veri rüzgar kuvvetli durumdadır. İkisi ‘Hayır’, üçü ‘Evet’ sınıfındandır.**

****

**Entropi(Strong|Windy) = 0.9709**

**Tüm entropiler hesaplandığına göre;**

****

**Entropi(Windy) = 0.9370**

***Bilgi Kazancı = Entropi(Play)- Entropi(Windy) = 0.9402–0.9370= 0.0032***

**Hava = 0.315**

**Sıcaklık = 0.0838**

**Nem = 0.162**

**Rüzgâr = 0.0032**

**Bu bilgi kazançlarına göre ilk düğüm ‘Hava’ sütunudur.**

**diyagram, daire, çizgi, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Buna göre oluşan karar ağacının ilk düğüm hava durumudur. Bu çıkan sonuca göre veriler ayrılır.**

**metin, ekran görüntüsü, çizgi, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Overcast — Rainy — Sunny durumlarına göre veriler ayrıştırılır.**

**Eğer oluşan veri kümelerinde karar sınıfı eğer aynı üyeye sahip ise tekrar bir işlem yapılmadan o sonuca göre karar sınıfı üyeliği verilebilir.**

**metin, diyagram, çizgi, daire içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Karar sınıflarında dağılım olan “rainy” ve “sunny” tabloları için önce karar sınıfı entropisi hesaplanır, ardından da tekrar seçilmeyen sütunlar için entropiler hesaplanıp bilgi kazançları bulunur. Bulunan bilgi kazancının en yüksek değerine göre sıradaki düğüm her dal için belirlenir. Farklı olmak zorunda değil aynı düğüme de gidebilirler.**

**KOD ile C4.5**

***Kullanacağımız kütüphaneler***

**import pandas as pd  
import numpy as np  
import math  
from graphviz import Digraph**

**Entropiyi hesaplayacak fonksiyonumuz**

**def calculate\_entropy(data, target\_col):  
  
 values = data[target\_col].value\_counts(normalize=True)  
 entropy = -sum(values \* np.log2(values))  
 return round(entropy, 3)**

***Bilgi Kazancı — Bilgi Kazancı Oranını hesaplayacak fonksiyonumuz***

**def calculate\_information\_gain\_ratio(data, feature\_col, target\_col):  
  
 total\_entropy = calculate\_entropy(data, target\_col)  
 values = data[feature\_col].unique()  
 weighted\_entropy = 0  
  
 for value in values:  
 subset = data[data[feature\_col] == value]  
 weight = len(subset) / len(data)  
 subset\_entropy = calculate\_entropy(subset, target\_col)  
 weighted\_entropy += weight \* subset\_entropy  
  
 info\_gain = total\_entropy - weighted\_entropy  
  
  
 split\_info = -sum(  
 (len(data[data[feature\_col] == value]) / len(data)) \* math.log2(len(data[data[feature\_col] == value]) / len(data))  
 for value in values  
 if len(data[data[feature\_col] == value]) > 0  
 )  
  
  
 gain\_ratio = info\_gain / split\_info if split\_info != 0 else 0  
 return round(gain\_ratio, 3)**

***Medyanları hesaplayan fonksiyonumuz***

**def preprocess\_continuous\_values(data, feature\_col):  
  
 threshold = data[feature\_col].median()  
 data[feature\_col] = data[feature\_col].apply(lambda x: f"below {threshold}" if x <= threshold else f"above {threshold}")**

***Ağacımızı oluşturan fonksiyonumuz***

**def build\_decision\_tree(data, target\_col, features, depth=0):  
  
  
 if len(data[target\_col].unique()) == 1:  
 return data[target\_col].iloc[0]  
  
  
 if not features:  
 return data[target\_col].mode()[0]  
  
  
 gains = {feature: calculate\_information\_gain\_ratio(data, feature, target\_col) for feature in features}  
 best\_feature = max(gains, key=gains.get)   
  
   
 tree = {best\_feature: {}}  
  
 for value in data[best\_feature].unique():  
 subset = data[data[best\_feature] == value]  
 if subset.empty:  
 tree[best\_feature][value] = data[target\_col].mode()[0]  
 else:  
 subtree = build\_decision\_tree(  
 subset,  
 target\_col,  
 [feat for feat in features if feat != best\_feature],  
 depth + 1  
 )  
 tree[best\_feature][value] = subtree  
  
 return tree**

***Kurallarımızı yazdıran fonksiyonumuz***

**def print\_rules(tree, path=""):  
  
 if not isinstance(tree, dict):   
 print(f"{path} => {tree}")  
 return  
  
 for node, branches in tree.items():  
 for value, subtree in branches.items():  
 new\_path = f"{path} {node} = {value}" if path else f"{node} = {value}"  
 print\_rules(subtree, new\_path)**

**Ağacımızı çizen fonksiyonumuz**

**def visualize\_tree(tree, graph=None, parent=None, edge\_label=""):  
  
 if graph is None:  
 graph = Digraph(format='png')  
 graph.attr('node', shape='ellipse')  
  
 if isinstance(tree, dict):   
 for node, branches in tree.items():  
 graph.node(node, label=node)   
 if parent:  
 graph.edge(parent, node, label=edge\_label)  
 for branch\_value, subtree in branches.items():  
 visualize\_tree(subtree, graph, parent=node, edge\_label=str(branch\_value))  
 else:   
 leaf\_label = f"{tree}"   
 leaf\_node = f"{parent}\_{leaf\_label}"  
 graph.node(leaf\_node, label=leaf\_label, shape='box')  
 graph.edge(parent, leaf\_node, label=edge\_label)  
  
 return graph**

***Budama için güven aralığını hesaplayan fonksiyonumuz***

**def calculate\_confidence(data, target\_col, label):  
 total = len(data)  
 count = len(data[data[target\_col] == label])  
 return count / total if total > 0 else 0  
  
def prune\_tree(tree, data, target\_col, confidence\_threshold=0.25):  
  
 if not isinstance(tree, dict):  
 return tree  
  
 pruned\_tree = {}  
 for feature, branches in tree.items():  
 pruned\_tree[feature] = {}  
 for value, subtree in branches.items():  
 subset = data[data[feature] == value]  
  
 if isinstance(subtree, dict):   
 pruned\_tree[feature][value] = prune\_tree(subtree, subset, target\_col, confidence\_threshold)  
 else:  
 confidence = calculate\_confidence(subset, target\_col, subtree)  
 if confidence < confidence\_threshold:   
 pruned\_tree[feature][value] = data[target\_col].mode()[0]   
 else:  
 pruned\_tree[feature][value] = subtree  
  
 return pruned\_tree**

***Diğer Kodlar***

**file\_path = input("Enter the path to your CSV file: ")  
target\_col = input("Enter the target column name: ")  
data = pd.read\_csv(file\_path)  
for feature in data.columns:  
 if feature != target\_col and data[feature].dtype in ['float64', 'int64']:  
 preprocess\_continuous\_values(data, feature)  
  
  
features = [col for col in data.columns if col != target\_col]  
  
  
decision\_tree = build\_decision\_tree(data, target\_col, features)  
  
pruned\_tree = prune\_tree(decision\_tree, data, target\_col, confidence\_threshold=0.25)  
  
dot = visualize\_tree(pruned\_tree)  
pruned\_file\_name = input("Enter the file name to save the pruned tree (without extension): ")  
dot.render(pruned\_file\_name)  
print(f"Pruned decision tree has been saved as {pruned\_file\_name}.png")  
  
# Budanmış ağacın kurallarını yazdır  
print("Pruned Decision Tree Rules:")  
print\_rules(pruned\_tree)**

**metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Kurallar**

**diyagram, metin, çizgi, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Karar Ağacı**

**Kaynakça:**

**[1]-Gümüşçü, A., Taşaltın, R., & Aydilek, İ. B. (2016). C4.5 karar ağaçlarında genetik algoritma ile budama [C4.5 Decision tree pruning using genetic algorithm]. *Harran Üniversitesi, Mühendislik Fakültesi Dergisi*. Geliş: 10 Aralık 2015, Revizyon: 26 Ocak 2016, Kabul: 27 Ocak 2016, Elektronik Yayın: 23 Kasım 2016, Basım: 23 Aralık 2016.**