

INTRODUCTION TO ARTIFICIAL INTELLIGENCE DERSİ

DECISION TREE ÖDEVİ

FATİH TALHA TÜMER – 191180081

İÇİNDEKİLER

[İÇİNDEKİLER i](#_Toc124672533)

[ŞEKİLLER LİSTESİ ii](#_Toc124672534)

[1. GİRİŞ 1](#_Toc124672535)

[2 SINIFLANDIRICI 1](#_Toc124672536)

[3 SONUÇ VE KAZANIMLAR 1](#_Toc124672537)

[KAYNAKÇA **Error! Bookmark not defined.**](#_Toc124672538)

# ŞEKİLLER LİSTESİ

[Şekil 1: Rastgele verilerin oluşturulması 1](#_Toc124672127)

[Şekil 2: Kullanıcı tarafından ek verilerin girilmesi 2](#_Toc124672128)

[Şekil 3: Node sınıfı 3](#_Toc124672129)

[Şekil 4: Veri ön hazırlığı ve modelin eğitilmesi 3](#_Toc124672130)

[Şekil 5: Eğitilen modelin Graphviz kütüphanesi aracılığıyla ağaç şeklinde görselleştirilmesi 4](#_Toc124672131)

# GİRİŞ

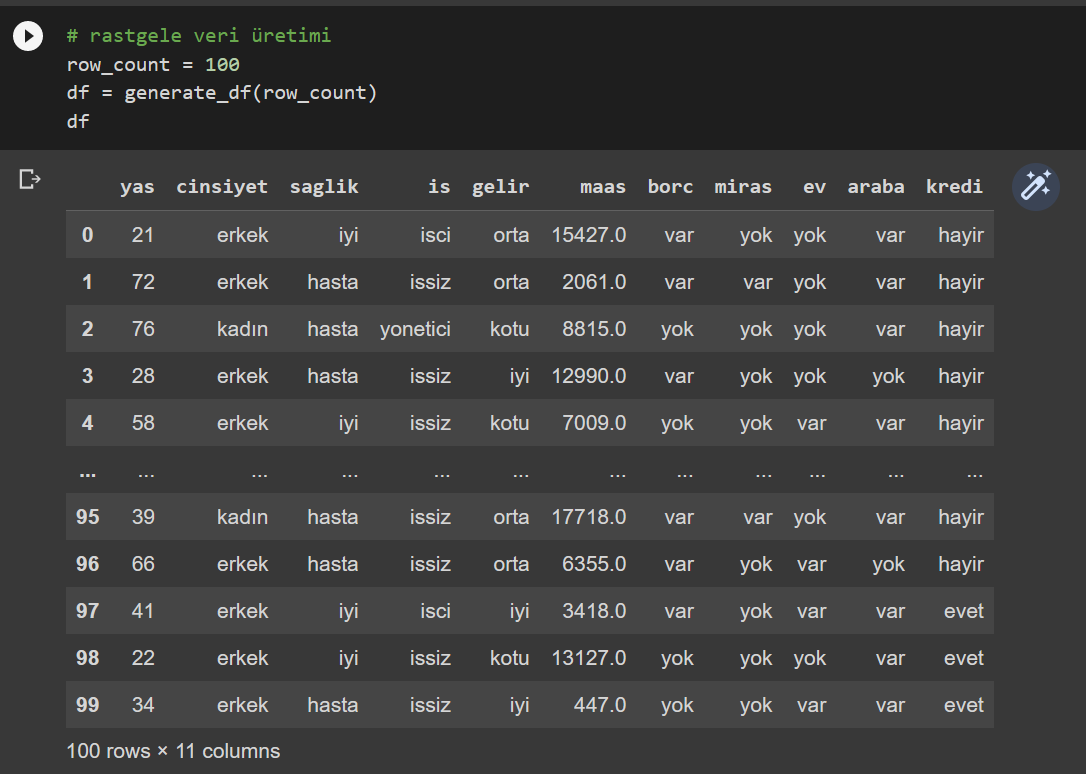
Bu çalışmada bir karar ağacı algoritması, hazır kütüphaneler kullanılmadan geliştirilmiş, veri rastgele olarak kullanıcının istediği kadar yeni özellik girme özelliği eklenmiş bir algoritma geliştirilmiştir. Information gin için gini index ve entropi değerleri kullanılmıştır. Kodlar, açıklamalar ve görselleştirilmiş ağaç çalışmanın diğer bölümlerinde paylaşılmıştır.

# SINIFLANDIRICI

* Ödev, Python, jupyter notebook ve Google colab kullanılarak oluşturulmuştur.
* Karar ağacının gerçekleştirilmesinde herhangi bir hazır fonksiyon kullanılmamıştır.

Ödevin oluşturulma aşamaları:

1. İstenilen tablonun, kullanıcının gireceği satır sayısına göre rastgele oluşturulması:



Şekil 1: Rastgele verilerin oluşturulması

Kullanıcının istediği kadar özelliği tabloya eklemesini sağlayan kısım. Kullanıcı önce başlığı sonra da o başlık için girmek istediği kadar örnek girebiliyor. Daha sonra girilen bu değerler dataframe’e ekleniyor.

Text

Description automatically generated

Şekil 2: Kullanıcı tarafından ek verilerin girilmesi

Node sınıfı oluşturuldu. Eğitilecek olan modelin saklanması için ağaç yapsının temeli olması için node yapısı kullanıldı.

Text

Description automatically generated

Şekil 3: Node sınıfı

DecisionTree oluşturuldu.

\_\_init\_\_, select\_best\_split (en yüksek information gain’e sahip spliti bulma),

split (belirlenen özelliğe göre böle işleminin gerçekleşmesi), fit (verilerin eğitime hazırlanması),

predict (dizi şeklinde alınnan verilerin make\_prediction yardımıyla tahmin edilmesi)

build\_tree (verinin minimum sample count ve maximum depth kurallarına uyması durumunda tekrar tekrar node üretilmesi)

calculate\_leaf\_node (kararlara ulaşılan son yaprak node’ların ulaştıkları kararların atanması)

feature\_type (kategorik ve numerik verilerin ayrı işleler görmesi için ayırma işlemi)

Entropi (entropinin heasplaması)

Gini (gini indexinin hesaplanması), info\_gain (entropi ya da gini index kullanarak information gain’in hesaplanması) fonksiyonları kullanıldı.

Verinin ön hazırlığı ve modelin eğitilmesi, accuracy değerinin tespit edilmesi

A screenshot of a computer

Description automatically generated with medium confidence

Şekil 4: Veri ön hazırlığı ve modelin eğitilmesi

Eğitilen modelin graphviz kütüphanesi kullanılarak görselleştirilmesi (Jupyter notebook’da graphviz kütüphanesinde hata alındığı için çalışma Google colab’de devam ettirilmiştir.) Sonraki sayfada çıktının yükske çözünürlüklü görünütüsü bulunmaktadır. PDF versiyonu da ödev klasöründe mevcuttur.

Text

Description automatically generated

Şekil 5: Eğitilen modelin Graphviz kütüphanesi aracılığıyla ağaç şeklinde görselleştirilmesi

Diagram

Description automatically generated

# SONUÇ VE KAZANIMLAR

Karar ağacı yapıları ve ağaç yapılandırılırken kullanılan purity ölçüm değerlerinin (entropi ve gini index) ne kadar önemli ve faydalı olduğu anlaşıldı. 2 yöntem de başaralı bir şekilde eklendi. Hazır kütüphaneler incelendi ve bu kütüphanelerin geliştirme/yayınlama süreçlerinde zorluklar anlaşıldı. Bu kütüphanelerin sağladığı kolaylıkların farkına varıldı. Kod yazma yeteneğinde ve makine öğrenmesini kavramada gelişmeler oldu.