**T. C.**

**BOZOK ÜNİVERSİTESİ**

**MÜHENDİSLİK MİMARLIK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

****

**YAPAY ZEKA**

**Makine Öğrenmesi İle Titanik Gemisinden Hayatta Kalma Tahmini**

**HAZIRLAYAN**

16008119072 – TAHSİN SOYAK

**DERSİN ÖĞRETİM ÜYESİ**

**Dr. Öğr. Üyesi MUHAMMET EMİN ŞAHİN**

Mayıs

**YOZGAT 2022**

İçindekiler

[1.Kütühpaneler; 3](#_Toc104080178)

[2.Veri Analizi; 3](#_Toc104080179)

[3.Makine Öğrenmesi İçin Veri Hazırlama 4](#_Toc104080180)

[i. Boş veriler doldurma 5](#_Toc104080181)

[ii. Kategorik verileri sayısal veriye dönüştürme 5](#_Toc104080182)

[4. Train Ve Test Verilerini Ayırmak 5](#_Toc104080183)

[5.Algoritma Seçmek 5](#_Toc104080184)

[Decision Tree: 5](#_Toc104080185)

[Nitelik Seçim Ölçüleri: 6](#_Toc104080186)

[6. Gini Index’i ile Hesaplama 7](#_Toc104080187)

[7. Entropi ile Hesaplama 8](#_Toc104080188)

[8. Karmaşıklık Matrisi 9](#_Toc104080189)

[9. Ağaç Budama 10](#_Toc104080190)

[i. Pre-Pruning: 10](#_Toc104080191)

[ii. Post Pruning: 10](#_Toc104080192)

[10. Raspberry Pi 13](#_Toc104080193)

**Makine Öğrenmesi İle Titanik Gemisinden Hayatta Kalma Tahmini**

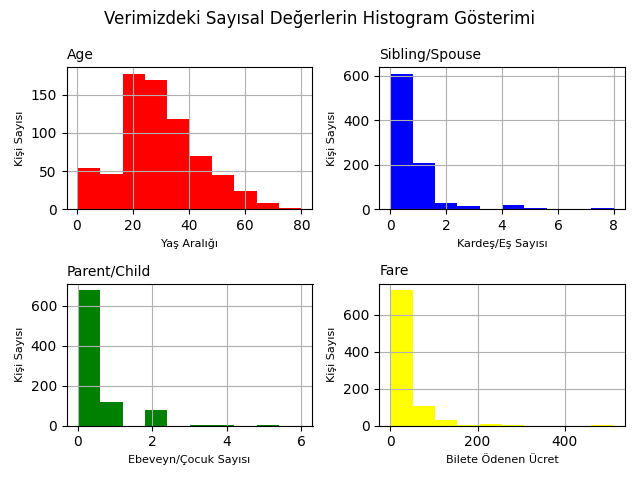
# 1.Kütühpaneler;

* Pandas: Bu kütüphane temel olarak zaman etiketli serileri ve sayısal tabloları işlemek için bir veri yapısı oluşturur ve bu şekilde çeşitli işlemler bu veri yapısı üzerinde gerçekleştirilebilir.
* Numpy: Çok boyutlu dizileri ve matrisleri destekleyen, bu diziler üzerinde çalışacak üst düzey matematiksel işlevler ekleyen bir kitaplıktır.
* Sklearn: Makine öğrenimi kitaplığıdır.
* Matplotlib: Sayısal matematik uzantısı NumPy için bir çizim kitaplığıdır. Tkinter, wxPython, Qt veya GTK gibi genel amaçlı GUI araç takımlarını kullanarak grafikleri uygulamalara gömmek için nesne yönelimli bir API sağlar.
* Seborn: Seaborn, matplotlib tabanlı bir Python veri görselleştirme kütüphanesidir.

# 2.Veri Analizi;

Kütüphaneler tanımladıktan sonra veriler hakkında bilgi edinmek amacı ile verilerin grafiklerini çizdirilmelidir. Grafikleri çizdirirken eğer veriler farklı türlerde ise numerik ve kategorik veriler farklı şekilde matplot yardımıyla gösterilmeli.

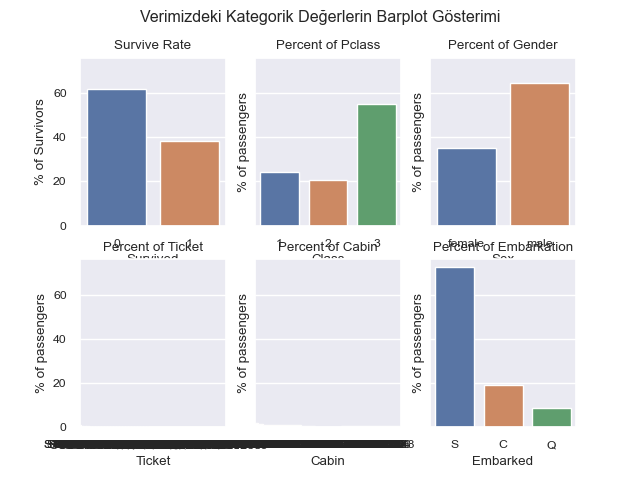
Numerik verilerimiz bir figürde toplanıp yan yana histogram aracılığıyla gösterildi.



Şekil 1'de görüldüğü gibi histogram şeklinde verilerimiz dağılımı gösterilmiştir.

Örneğin; Yaş verisi 20-40 aralığında olanların daha fazla olduğunu gösteriyor. Kardeş eş sayısınında hiç kardeşi eşi olmayanların sayısının 600’ün üstünde olduğunu görürürüz.

Genelde kategorik verilerin gösterimini ve analizini barplot yardımıyla yapılır.



Şekil 2'de görüldüğü gibi barplot şeklinde verilerimiz dağılımı gösterilmiştir.

Örneğin; Kategorik değerlerimizden Survived verimiz 0 ve 1 olarak % olarak gösterilmiştir. %60’tan fazlasının öldüğü görülmektedir. Embarked sütunmuzda büyük bir çoğunluğun S konumundan bindiği görülmektedir.

Siyah şerit şeklinde gösterilen ticket ve kabin barplotları çok fazla kategori olduğundandır. [Gömülü sistem Raspberry Pi uyarlaması için bu barplotlarımızı yorum satırına alıyoruz.]

# 3.Makine Öğrenmesi İçin Veri Hazırlama

Numerik ve Kategorik verilere bakıldığında gereksiz kolonlar olabilir. Örneğin Yolcu Id numarası tahmin için işimize yaramayacaktır. İşlemlerimizde gereksiz dallanma yaratmamaları için bu kolonlar silinmelidir.

Numerik verilerimizden;

* PassengerId
* Sibsp
* Parch

Verilerini sileceğiz.

Kategorik verilerimizden;

* Name
* Ticket
* Cabin
* Embarked

Verilerini sileceğiz.

## i. Boş veriler doldurma

Pandas kütüphanesi yardımıyla kolonlardaki boş verileri kontrol etmemiz gerek. isnull() fonksiyonu ile age kolonlarda boş satırlar kontrol edilmelidir. Bizim kullanacağımız veri setinde Age verimize bir ortalama buluyoruz cinsiyete göre ve sonra lambda expression kullanarak boş satırları dolduruyoruz.

## ii. Kategorik verileri sayısal veriye dönüştürme

Sklearn kütüphanesinden labelencoder kullanarak cinsiyet kolonumuzdaki ‘male’ ve ‘female’ verilerini 1 ve 0 a dönüştürülmelidir. Farklı olarak category encoders kullanılabilir.

# 4. Train Ve Test Verilerini Ayırmak

Sklearn kütüphanesi yardımıyla train ve test verileri ayrılmalıdır. Ancak öncelikle verilerde hedefi belirlemiz gerek. Bizim hedefimizi survived kolonundaki yaşadımı öldümü verisi. Bunun için değişkenlerimizden birine özellikler-features diğerine ise hedefler atanmalıdır.

Ardından train ve test verimizi ayrılır, genel olarak ayırma oranı %30’a %70’tir. %30 test verisi %70 train verisidir. Bizim elimizdeki veriler üzerinden gidecek olacaksak train verisi 623 tane, test verisi ise 268 tanedir.

# 5.Algoritma Seçmek

* Linear regression
* Logistic regression
* Decision tree
* SVM algorithm
* Naive Bayes algorithm
* KNN algorithm
* K-means
* Random forest algorithm
* Dimensionality reduction algorithms
* Gradient boosting algorithm and AdaBoosting algorithm

Algoritmaları arasından Karar Ağaçlarını seçiyoruz. [Desicion Tree]

## Decision Tree:

Karar Ağacı algoritması, denetimli öğrenme algoritmaları ailesine aittir. Diğer denetimli öğrenme algoritmalarından farklı olarak, karar ağacı algoritması regresyon ve sınıflandırma problemlerini çözmek için de kullanılabilir.

Karar Ağacı kullanmanın amacı, önceki verilerden (eğitim verileri) çıkarılan basit karar kurallarını öğrenerek hedef değişkenin sınıfını veya değerini tahmin etmek için kullanılabilecek bir eğitim modeli oluşturmaktır.

Karar Ağaçlarında, bir kayıt için bir sınıf etiketi tahmin etmek için ağacın kökünden başlarız. Kök özniteliğinin değerlerini kaydın özniteliği ile karşılaştırırız. Karşılaştırma temelinde, o değere karşılık gelen dalı takip eder ve bir sonraki düğüme atlarız.



* Root Node: Tüm popülasyonu veya örneği temsil eder ve bu ayrıca iki veya daha fazla homojen kümeye bölünür.
* Splitting: Bir düğümün iki veya daha fazla alt düğüme bölünmesi işlemidir.
* Decision Node: Bir alt düğüm başka alt düğümlere bölündüğünde karar düğümü olarak adlandırılır.
* Leaf / Terminal Node: Bölünmeyen düğümlere Yaprak veya Terminal düğümü denir.
* Pruning: Bir karar düğümünün alt düğümlerini kaldırdığımızda bu işleme budama denir. Bölme işleminin tersini söyleyebilirsiniz.
* Branch / Sub-Tree: Tüm ağacın bir alt bölümüne dal veya alt ağaç denir.
* Parent and Child Node: Alt düğümlere bölünmüş bir düğüm, alt düğümlerin ana düğümü olarak adlandırılırken, alt düğümler bir ana düğümün çocuğudur.

## Nitelik Seçim Ölçüleri:

Eğer veri seti N sayıda öznitelikten oluşuyorsa, hangi özniteliğin köke veya ağacın farklı seviyelerine dahili düğümler olarak yerleştirileceğine karar vermek karmaşık bir adımdır. Kök olarak herhangi bir düğümü rastgele seçmek sorunu çözemez. Rastgele bir yaklaşım izlersek, bize düşük doğrulukla kötü sonuçlar verebilir.

* Entropi,
* Bilgi kazanımı,
* gini indeksi,
* Kazanç Oranı,
* Varyansta Azaltma
* Ki-Kare

Gibi kriterlerle hesaplama yapabiliriz. Bu kriterler, her özellik için değerleri hesaplayacaktır. Değerler sıralanır ve öznitelikler sıra takip edilerek ağaca yerleştirilir, yani yüksek değerli öznitelik (bilgi kazancı olması durumunda) köke yerleştirilir.

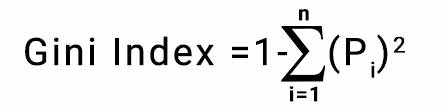
Bilgi Kazanımı bir ölçüt olarak kullanılırken, özniteliklerin kategorik, Gini indeksi için özniteliklerin sürekli olduğu varsayılır.

# 6. Gini Index’i ile Hesaplama

Gini index’i rastgele seçildiğinde yanlış sınıflandırılan belirli bir özelliğin olasılığı miktarını hesaplar. Gini endeksi 0 ve 0.5 değerleri arasında değişir. 0 sınıflandırmanın saflığını ifade eder. Tüm öğeler belirli bir sınıfa aittir veya orada sadece bir sınıf vardır. Gini endeksinin 0.5 değeri, bazı sınıflar üzerinde elemanların eşit dağılımını gösterir.

Tüm öğeler tek bir sınıfla bağlantılıysa, saf olarak adlandırılabilir. Bizim verimiz için ağacımızdaki yaprakta herkes ölüyse gini index 0 çıkar. Başka bir örnek verecek olursak bir düğümümzde verilerimiş 45’e 48 olarak ayrılmış ve gini index’imiz 0.499 olarak gözüküyor.

Gini endeksi, her bir sınıfın olasılıklarının kare toplamının 1’den çıkarılmasıyla belirlenir. Formülü;



Şekil 4 Gini Index Formülü

Karar ağacı algoritmaları bir düğümü bölmek için bilgi kazancını kullanır ve Gini indeksini veya entropi, bilgi kazancını tartmak için geçittir.

Entropi kavramına göre hangi özelliğin sınıflandırma hakkında maksimum bilgi sağladığını ölçmek için bilgi kazancı uygulanır. Genel olarak belirsizlik, bozukluk veya safsızlık boyutunu ölçerek, üstten (kök düğüm) alttan (yaprak düğümleri) başlayan entropi miktarını azaltmak amacıyla.

Gini Index uygulamak için DecisionTreeClassifier fonksiyonumuzda criter olarak ‘gini’ belirtip modelimizi oluşturuyoruz ve fitliyoruz. Sonra elimizdeki %30’luk test verimizle test ediyoruz.

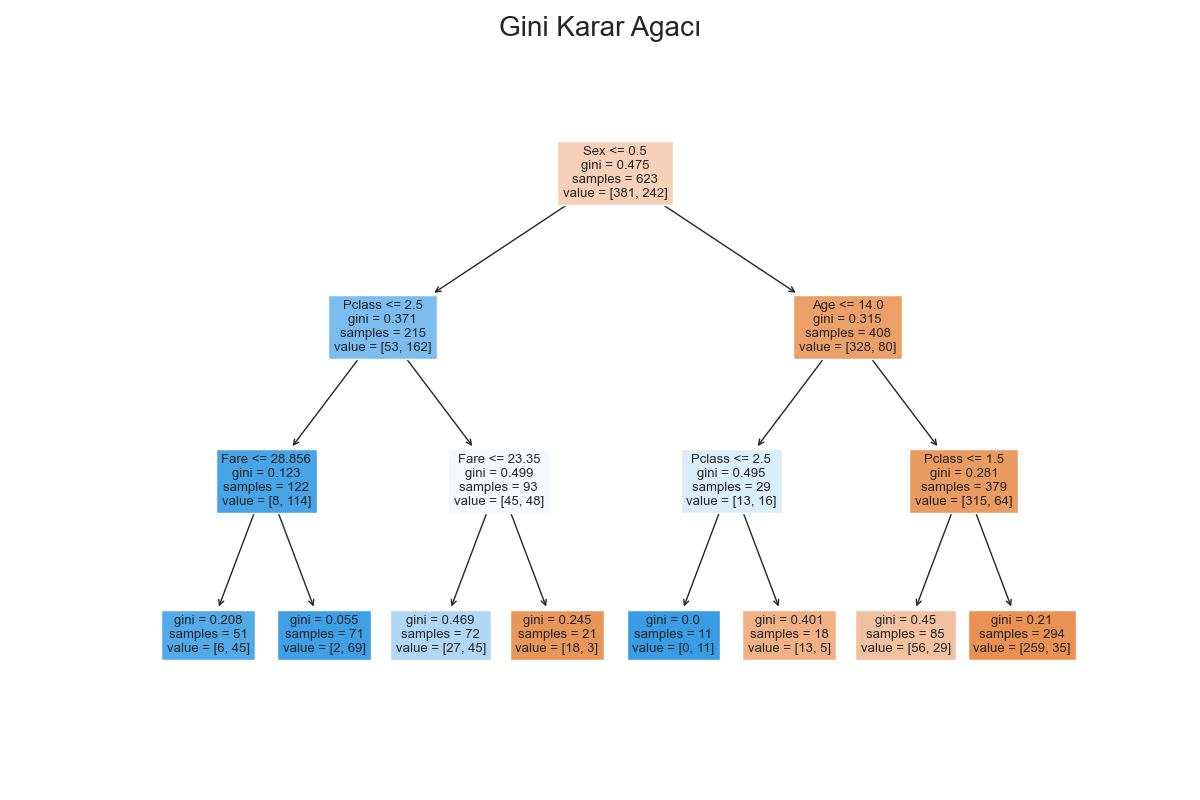
Ardından doğruluk kontrolü için metrics’ten accuracy\_score fonksiyonu import edilmelidir. Ardından train verisi ile eğitilen gini modelinin doğruluğu gösterilmelidir. Sonrasında train verisiyle gini modeliyle tahmin edilmiş veriyi kullanarak train set ile doğruluk oranına bakılmalıdır.

Çıktı olarak;

* Gini ile modelin doğruluğu: 0.81343284
* Train set ile doğruluk oranı: 0.82825040

Doğruluk oranımızı tabiki pre punning yapıp yapmamız etkiliyor. Örnek verirsek bizim gibi ağacımız için derinliği max 3 verdik.

Kendi örneğimiz için ağacımızı tree classını import edip çizdireceğiz.



Şekil 5 Gini kullanarak oluşturduğumuz karar ağacı

Çizilen ağaçta, 623 veriden erkek ve kadınları ayırıyor. Kadın olanlar sola, erkek olanlar sağa ayrılıyor. 396 erkek 227 kadın var. Şimdi ölüm ve yaşam oranı içinde yine ilk düğümde 381 ölü 242 yaşayan var.

# 7. Entropi ile Hesaplama

Entropi temel olarak, veri noktalarındaki safsızlık veya rasgeleliğin ölçümüdür. Entropi her zaman 0 ila 1 arasındadır. Entropi denklemi birçok avantajlı özellik nedeniyle logaritmalar kullanır.Formülü;

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 6 Entropi Formülü

Entropi uygulamak için DecisionTreeClassifier fonksiyonumuzda criter olarak ‘entropy’ belirtip modelimizi oluşturuyoruz ve fitliyoruz. Sonra elimizdeki %30’luk test verimizle test ediyoruz.

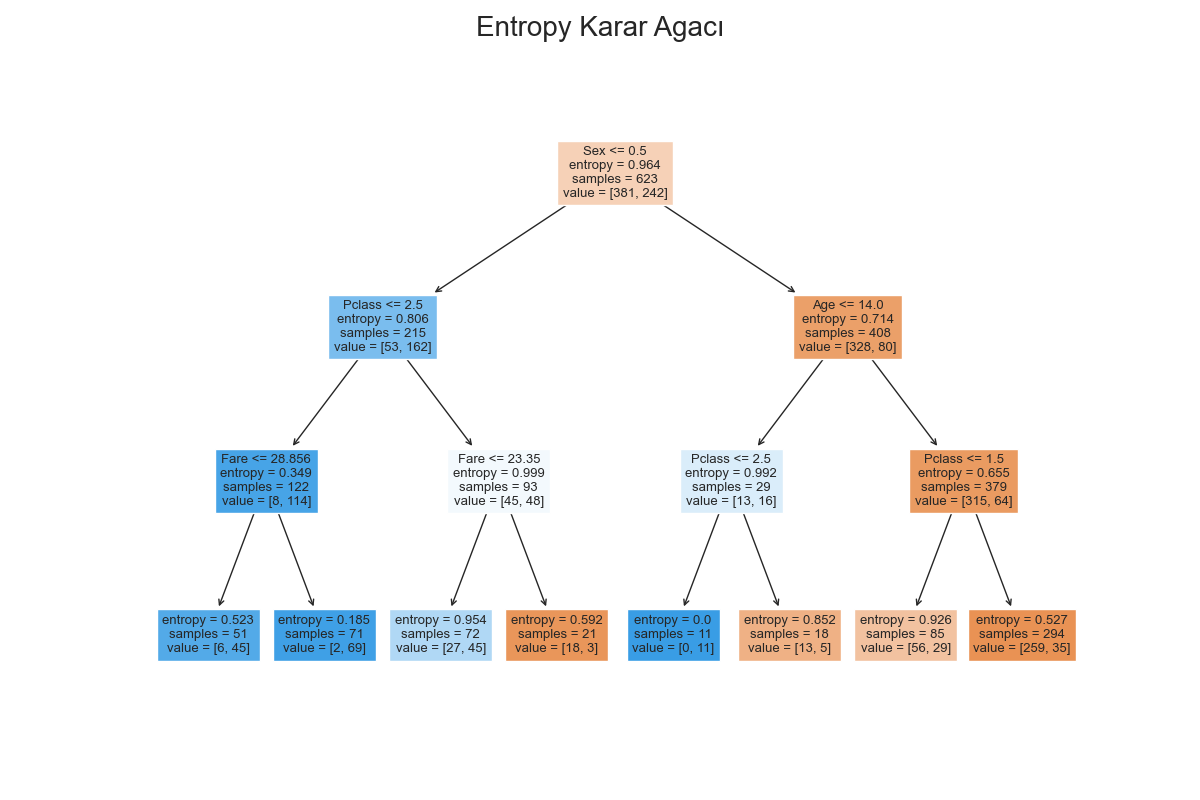
Tekrardan doğruluk kontrolümüz için metrics’ten accuracy\_score fonksiyonunu ile train verimizle eğittiğimiz entropi modelimizin doğruluğunu yazdırıyoruz. Sonrasında train verisiyle entropi modeliyle tahmin edilmiş veriyi kullanarak train set ile doğruluk oranına bakıyoruz.

Çıktı olarak;

* Entropi ile modelin doğruluğu: 0.81343284
* Train set ile doğruluk oranı: 0.8282504

Entropi ile hesaplama yapıldığında doğruluk oranlarının aynısı olduğunu görülür. Aradaki farkların oluşabilmesi için veri sayımız, veri türümüzün önemi vardir. Derinlik, yaprak bakma sayısını aynı kullanmamızın büyük bir etkisi var.

Kendi verimiz için ağacımızı çizdiriyoruz.



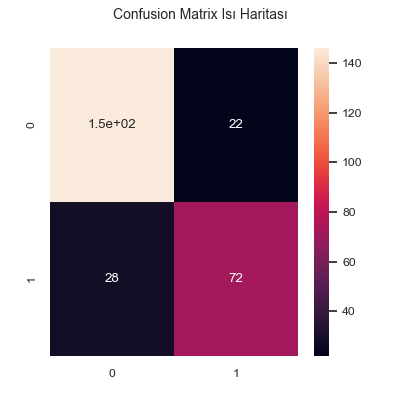
Şekil 7 Entropi kullanarak oluşturduğumuz karar ağacı

# 8. Karmaşıklık Matrisi

Confusion Matrix, hata matrisi olarak da bilinir. Tipik olarak denetimli bir öğrenme algoritması olan bir algoritmanın performansının görselleştirilmesine izin veren özel bir tablo düzenidir.

Kullanmak için metrics class’ından fonksiyonu import edilmelidir. Isı haritasını çıkarmak için matrix’in Seaborn kütüphanesinden heatmap kullanılabilir. Örneğimizdeki matris aşağıdadır;

* [[146  22]
* [ 28  72]]



Şekil 8 Confusion Matrix - Karmaşıklık Matrisi Isı Haritası

Prediction yani tahmin verileri ile observe yani yest kısmındaki bağıntılara bakılır.

# 9. Ağaç Budama

## i. Pre-Pruning:

* Bu teknik, karar ağacının inşasından önce kullanılır.
* Aşırı sığma (overfitting) sorununun üstesinden gelmemize yardımcı olur.
* Hyperparameter ayarlaması kullanılarak yapılabilir.

Projemizde zaten bu teknikleri kullanarak budama yapıldı.

* max\_depth: karar ağacının maksimum derinliği
* min\_sample\_split: Bir dahili düğümü bölmek için gereken minimum örnek sayısı
* min\_samples\_leaf: Bir yaprak düğümde olması gereken minimum örnek sayısı.

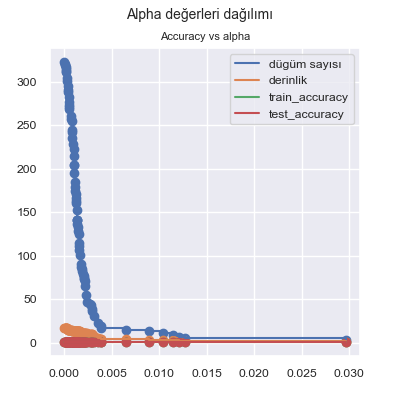
## ii. Post Pruning:

* Bu teknik, karar ağacının inşasından sonra kullanılır.
* Bu teknik, karar ağacının çok büyük bir derinliğe sahip olacağı ve overfitting göstereceği zaman kullanılır.
* Geriye dönük budama olarak da bilinir.
* Bu teknik, sonsuz büyümüş karar ağacımız olduğunda kullanılır.

Maliyet karmaşıklığı budama Alfa için doğru parametreyi bulmakla ilgilidir. Bu ağaç için alfa değerlerini alacağız ve budanan ağaçlarla doğruluğunu kontrol edeceğiz.

cost\_complexity\_pruning\_path fonksiyonunu kullanarak alfa değerimizi alacağız. Ardından her alfa değeri için model oluşturup listeye eklenmelidir. clfs ve alphas içindeki son elemanı çıkarmalıyız. Çünkü 1 düğümlü ağaç.

Alpha değerlerinin dağılımını çizdirip bakıyoruz. x ekseninde alpha değerleri ve y ekseninde düğüm sayısı var.



Şekil 9 Alfa değerlerinin dağılımı

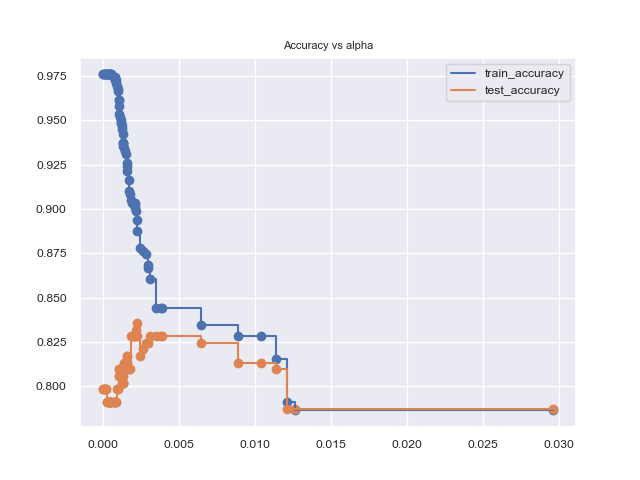
Sapma ve varyans değişimini takip edersek, düşük sapma (düşük eğitim hatası) ve düşük varyans (düşük test hatası) olan Alpha değerlerine bakıp kendi verimiz için tablodan 0.002 değerini seçiyoruz.

Ardından bu alfa değeri ile ağacımızı çizdirip accuracy değerine bakıyoruz. Accuracy;

* Train score: 0.90369181
* Test score: 0.82835821

Accuracy değerinin arttığını görürüz.

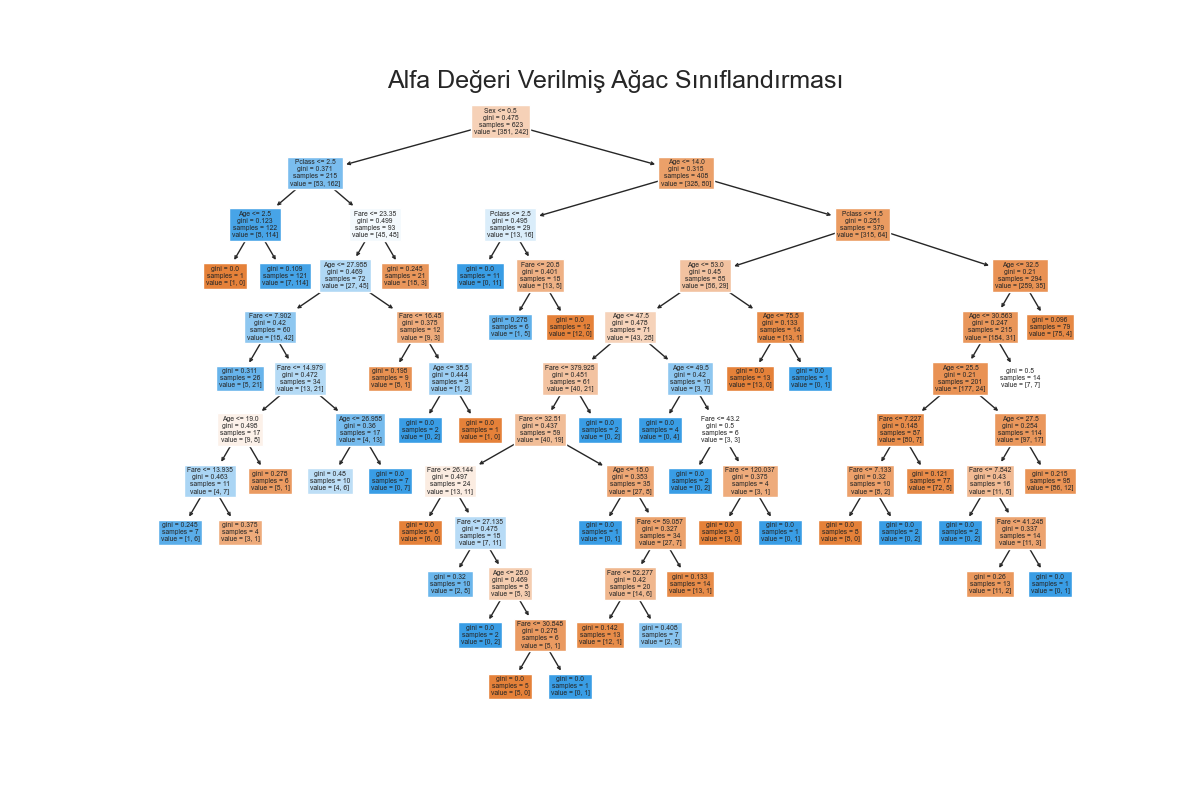
Her biri için accuracy bakalım.



Şekil 10 Doğruluk ile Alfa değerlerinin çizimi

Alfa değeri arttıkça doğruluğun düştüğünü görürüz. Test doğruluğunda tepe nokta için 0.00227 alfa değeri, train verisinde tepe doğruluk için 0.0001 alfa değeri olduğu gözükmektedir.

Şimdi ağacımızı çizdirip bakıyoruz.



Şekil 11 Alpha değeri 0.002 verilmiş ağaç

# 10. Raspberry Pi

Raspberry Pi başlangıçta eğitim amaçlı tasarlanmış çok küçük ve çok ucuz bir bilgisayardır. Gelecek nesil programcı adaylarının becerilerini geliştirmelerine ve bilgisayar donanımının nasıl çalıştığını daha iyi anlamalarına yardımcı olmak için tasarlanmıştır.

elektronik eşyalar, devre içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 12 Raspberry Pi 4 Ürün Görseli

Raspberry Pi,” tek kartlı” bir bilgisayardır. CPU tam orada kartın üzerindedir. Açıkta bulunan bağlantı noktaları ve takılı bilgisayar yongaları içeren geleneksel bir anakarta benzer. Ayrıca eklemek istediğiniz cihazların giriş ve çıkışlarını bağlamak için gereken tüm bileşenlere sahiptir. Ancak, elbette, bir tüketici ürününden bekleyeceğimiz monitör, klavye ve fare gibi birçok şey eksiktir. Ayrıca işletim sistemi veya program yoktur.

Akıllı ev eşyası aletleri, hava durumu merkezleri, oyun konsolları, robotlar, drone destekçisi olarak birçok alanda kullanılmaktadır. Ayrıca radyo istasyonu, medya, web sunucuları, kameralar, çeşitli ağlar, kablosuz baskı sunucuları gibi alanlarda da kendine yer bulmaktadır.

Projemiz için entegrasyon olarak lcd panelde doğruluk oranını yazdırdık.