TAHA ELHARIRI

öğrenci no :  2228142307

Biyomedikal Uygulamaları için Makine Öğrenmesi

final raporu

# Özet

Bu çalışmada, PPG sinyal veri kümesi kullanılarak lasso regresyon tekniği ile şeker seviyesinin tahmin edilmesi ve EEG sinyal veri kümesi kullanılarak Gaussian-based sınıflandırma tekniği ile negatif veya pozitif duygu tahmin edilmesi amaçlandı. Ayrıca, EEG sinyal veri kümesi üzerinde PCA analizi yapıldı.

Lasso regresyon sonuçlarına bakıldığında, R-squared değeri 0.9541 ve MSE değeri 45.220 olarak bulundu. Bu, regresyon modelinin veri kümesi üzerinde oldukça iyi bir performans gösterdiğini göstermektedir.

Gaussian-based sınıflandırma için, elde edilen doğruluk oranı 0.9324 olarak bulundu. Bu, modelin duygu tahmininde oldukça yüksek bir performans gösterdiğini göstermektedir. PCA analizi sonrası, iki kompenantlı veri kümesi üzerinde sınıflandırma doğruluk oranı 0.8604 olarak bulundu.

Sonuç olarak, bu çalışmada lasso regresyon ve Gaussian-based sınıflandırma teknikleri kullanılarak elde edilen sonuçlar, veri kümelerinin analizinde oldukça başarılı olduğunu göstermektedir. Lasso regresyon ile elde edilen R-squared ve MSE değerleri, modelin veri kümesi üzerinde iyi bir performans gösterdiğini gösterirken, Gaussian-based sınıflandırma ile elde edilen doğruluk oranları, modelin duygu tahmininde yüksek bir başarı sağladığını göstermektedir. PCA analizi sonrası elde edilen sınıflandırma doğruluk oranları, iki kompenantlı veri kümesi üzerinde sınıflandırmanın hala başarılı olduğunu göstermektedir. Bu çalışma, lasso regresyon ve Gaussian-based sınıflandırma tekniklerinin veri kümelerinin analizinde etkili olabileceğini göstermektedir.

# Giriş

## Regresyon ve regresyon algoritmaları

Regresyon algoritmaları, veri kümelerinde belirli bir değişkenin belirli bir değere nasıl etkilendiğini veya ne zaman etkilendiğini anlamak için kullanılan yöntemlerdir. Regresyon analizi, bir değişkenin diğer değişkenlerle nasıl ilişkili olduğunu ve bu ilişkiyi nasıl modelleyebileceğimizi anlamamıza yardımcı olur.

Regresyon algoritmaları, veri kümelerindeki bu ilişkileri modellemek için kullanılan yöntemlerdir. En yaygın regresyon algoritmalarından bazıları aşağıda verilmiştir:

* Lineer regresyon: Bu algoritma, bir değişkenin diğer değişkenlere linearly nasıl etkilendiğini anlamak için kullanılır.
* Lojistik regresyon: Bu algoritma, bir değişkenin diğer değişkenlere logistic olarak nasıl etkilendiğini anlamak için kullanılır. Bu algoritma, sınıflandırma problemlerinde de kullanılabilir.
* Lasso ve Ridge regresyon: Bu algoritmalar, lineer regresyondaki hiperparametrelerin seçiminde ve overfitting sorunlarının önlenmesinde kullanılır. Lasso regresyon, daha az önemli olan değişkenleri sıfırlamak için kullanılırken, Ridge regresyon ise hiperparametreleri daha az oynatmak için kullanılır.
* Decision Tree Regression : Bu algoritma, veri kümesindeki değişkenler arasındaki ilişkileri karar ağacı şeklinde göstererek modelleyen bir algoritmadır.
* Random Forest Regression : Bu algoritma, birçok karar ağacı kullanarak modellemekte ve overfitting problemini önlemekte kullanılır.

Bu algoritmalar veri kümelerinde farklı ilişkileri modellemek için kullanılabilir ve her birinin kendi avantajları ve dezavantajları vardır. Örneğin, lineer regresyon veri kümelerinde lineer ilişkileri modellemek için iyi bir seçim olabilirken, decision tree regression daha karmaşık ilişkileri modellemek için daha uygun olabilir. Ayrıca, her bir algoritmanın hiperparametrelerinin doğru olarak seçilmesi ve veri kümesinin iyi bir şekilde temizlenmesi de önemlidir. Bu nedenle, veri kümelerinde regresyon algoritmalarının kullanımı için doğru algoritma ve hiperparametrelerin seçilmesi ve veri kümelerinin iyi bir şekilde temizlenmesi gerekir.

Regresyon analizi ve algoritmaları, birçok farklı alanda kullanılabilir. Aşağıda bazı örnekler verilmiştir:

* Ekonomi: Örneğin, bir ülkenin büyüme hızı ile işsizlik oranı arasındaki ilişki modellemek için regresyon analizi kullanılabilir.
* Finans: Örneğin, bir şirketin kârı ile yatırım yapılan hisse senetleri arasındaki ilişki modellemek için regresyon analizi kullanılabilir.
* Sağlık: Örneğin, bir hastanın yaşı ile kanser riski arasındaki ilişki modellemek için regresyon analizi kullanılabilir.
* Pazarlama: Örneğin, bir ürünün satış miktarı ile reklam harcaması arasındaki ilişki modellemek için regresyon analizi kullanılabilir.
* Meteoroloji: Örneğin, hava sıcaklığı ile yağış miktarı arasındaki ilişki modellemek için regresyon analizi kullanılabilir.
* Bilgisayar mühendisliği ve robotiğinde, hareketli nesnelerin tahmini konumlarını tahmin etmek veya sensörlerin ölçtüğü verileri modellemek için regresyon algoritmaları kullanılabilir.
* Enerji sektöründe, enerji tüketimini tahmin etmek veya enerji üretimini optimize etmek için regresyon analizi kullanılabilir.
* İnsan kaynakları yönetimi ve işletme alanlarında, çalışan performansı ile maaş arasındaki ilişki modellemek için regresyon algoritmaları kullanılabilir.
* İnsan ve sağlık bilimlerinde, hastalıkların risk faktörlerini veya tedavi etkilerini tahmin etmek için regresyon analizi kullanılabilir.

Sonuç olarak, regresyon analizi ve algoritmaları çok geniş bir kullanım alanına sahiptir ve veri kümelerinde farklı ilişkileri modellemek için kullanılabilir. Bu nedenle, veri analizi ve modelleme işlemlerinde regresyon algoritmalarının önemli bir yeri vardır.

## Sınıflandırma algoritmaları

Sınıflandırma algoritmaları, veri kümelerindeki nesnelerin belirli bir sınıfa ait olduğunu tahmin etmek için kullanılan algoritmalardır. Bu algoritmalar, öğrenme kuralları kullanarak veri kümelerinden elde edilen bilgileri öğrenir ve daha sonra bu bilgileri kullanarak yeni veri nesnelerini sınıflandırmak için kullanır.

Sınıflandırma algoritmaları genellikle iki ana kategoriye ayrılır: destek vektör makinesi (SVM) ve ağaç tabanlı algoritmalar.

Destek vektör makinesi (SVM) algoritmaları, veri noktalarının belirli bir sınıfa ait olduğunu tahmin etmek için kullanılan algoritmalardır. Bu algoritmalar, veri noktalarının belirli bir sınıfa ait olduğunu tahmin etmek için öğrenme kuralları kullanır.

Ağaç tabanlı algoritmalar ise, veri kümelerinde belirli bir sınıfa ait olduğu tahmin etmek için kullanılan algoritmalardır. Bu algoritmalar, veri kümelerinde belirli bir sınıfa ait olduğunu tahmin etmek için karar ağaçları veya rasgele ormanları kullanırlar. Bu algoritmalar, veri noktalarının belirli bir sınıfa ait olduğunu tahmin etmek için öğrenme kuralları kullanır ve ayrıca veri kümelerindeki öznitelikler arasındaki ilişkileri de modellemektedir.

Örneğin, dijital görüntüleri veya ses dosyalarını sınıflandırmak için kullanılabilecek sınıflandırma algoritmaları vardır. Ayrıca, ticaret veya finans sektörlerinde müşteri davranışlarını veya ödeme performanslarını tahmin etmek için de sınıflandırma algoritmaları kullanılabilir.

Sınıflandırma algoritmaları, veri kümelerindeki nesnelerin belirli bir sınıfa ait olduğunu tahmin etmek için kullanılır ve çok geniş bir kullanım alanına sahiptir. Bu nedenle, veri analizi ve modelleme işlemlerinde sınıflandırma algoritmalarının önemli bir yeri vardır.

Sınıflandırma algoritmaları arasında en yaygın olarak kullanılanlar arasında logistic regresyon, Naive Bayes, k-en yakın komşu (k-NN), ağaç tabanlı algoritmalar, Random Forest, Gradient Boosting, Neural Network gibi algoritmalar yer alır. Bu algoritmalar arasında en uygun algoritmanın seçimi veri setinin özelliklerine ve amaçlarına göre değişebilir.

Son olarak, sınıflandırma algoritmalarının doğruluğunu arttırmak için veri ön işleme, boyut azaltma, öznitelik seçimi gibi teknikler kullanılabilir. Ayrıca, sınıflandırma algoritmalarının performansını değerlendirmek için kullanılabilecek ölçütler arasında accuracy, precision, recall, f1-score, support gibi metrikler yer alır.

Sınıflandırma algoritmalarının kullanım alanları oldukça geniştir. Örneğin:

* Spam e-postaların filtrelenmesi
* Dokümanların belirli bir kategoriye ayrılması
* Görüntülerin tanımlanması
* Ses sinyallerinin tanımlanması
* Müşteri davranışlarının tahmin edilmesi
* Medikal tanıların yapılması
* Finansal risk analizi
* İnsan yüzü tanıma
* Sentiment analizi
* Tarama formlarının otomatik olarak okunması.

Sınıflandırma algoritmalarının kullanım alanları sürekli olarak gelişmekte ve yeni alanlar da ortaya çıkmaktadır. Özellikle, artan miktarda veri ve işlem gücü ile birlikte sınıflandırma algoritmalarının kullanım alanları daha da genişlemektedir.

# Lasso regresyon ile PPG sinyal data seti kullanılarak kan şeker tahmini

# 

Bu bölümde, PPG sinyal veri seti kullanılarak Lasso regresyon algoritması kullanarak kan şeker tahmini yapılacak ve sonuçların değerlendirilmesi yapılacak.

## Lasso regresyon

Lasso regresyon, bir regresyon algoritmasıdır. Bu algoritma, öznitelik seçimi yapmak için kullanılır. Lasso regresyon, Ridge regresyon algoritmasına benzer bir yöntemdir. Ancak, Lasso regresyon algoritması, Ridge regresyon algoritmasından farklı olarak öznitelikler arasındaki ilişkileri analiz eder ve en önemli öznitelikleri seçer. Lasso regresyon algoritması, öznitelikler arasındaki ilişkileri analiz ederken, öznitelikler arasındaki lineer ilişkileri kullanır.

Lasso regresyon algoritması, öznitelikler arasındaki lineer ilişkileri kullanarak, en önemli öznitelikleri seçer ve diğer öznitelikleri sıfır değerine yakın bir değere getirir. Bu sayede, Lasso regresyon algoritması, veri setinde önemli olan öznitelikleri seçerek daha etkili sonuçlar alınmasını sağlar. Lasso regresyon algoritması, Ridge regresyon algoritmasına göre daha az kompleks olmasına rağmen, daha az öznitelik seçebilir.

Lasso regresyon algoritması, özellikle büyük veri setlerinde öznitelik seçimi yapmak için kullanılır. Örneğin, PPG sinyal data seti kullanılarak kan şeker tahmini yapmak için Lasso regresyon algoritması kullanılabilir. Bu algoritma, veri setinde en önemli öznitelikleri seçerek daha etkili sonuçlar alınmasını sağlar. Bu sayede, doğruluk oranı yüksek bir şekilde kan şeker tahmini yapılabilir. Lasso regresyon algoritması aynı zamanda overfitting problemlerini azaltır ve modeli daha anlamlı hale getirir. Özellikle geniş veri setlerinde öznitelik seçimi yaparken Lasso regresyon algoritması önerilir.

## PPG dataset

Bu data set telefon PPG sensorlarını kullanılarak elde edildi ve daha sonra öznitelikleri HCTSA kütüphanesini kullanılarak çıkarıldı. Bu data setin son sütünü kan şeker seviyesini göstermektedir. Değer sütünler ise sinyalin özentilikleridir. Toplam 1003 satır × 4936 sütünden oluşmaktadır. Kan şeker seviyeleri 61-350 arasındadır.

## Çalışmanın adımları

İlk olarak kullanılacak kütüphaneleri import etmektir bu kütüphaneler:

• Pandas : Pandas, Python programlama dili için geliştirilmiş bir veri analitik kütüphanesidir. Bu kütüphane, veri analizi, temizleme, manipülasyon ve sunum için kullanışlı araclar sunar. Pandas, veri setlerini okuma, yazma, filtreleme, düzenleme ve özetleme gibi işlemleri kolaylaştırır.

• Numpy: Numpy, Python programlama dili için geliştirilmiş bir sayısal hesaplama kütüphanesidir. Bu kütüphane, matematiksel fonksiyonlar, diziler, matrisler ve çok boyutlu veri yapıları gibi işlemleri gerçekleştirmek için kullanılır. Numpy, veri analitikte ve makine öğreniminde sıklıkla kullanılan bir kütüphanedir.

• Matplotlib : Matplotlib, Python programlama dili için geliştirilmiş bir grafikleştirme kütüphanesidir. Bu kütüphane, verileri görselleştirmek, çizimler yapmak ve grafikler oluşturmak için kullanılır. Matplotlib, veri analitikte ve makine öğreniminde özellikle kullanışlıdır. Çeşitli grafikler, histogramlar, scatter plotlar, line plotlar ve diğer çizimler oluşturmak için kullanılabilir. Ayrıca, çeşitli veri setleri üzerinde farklı grafikleri karşılaştırmak ve veri analizine dayalı sonuçları görselleştirmek için kullanılabilir.

• Sklearn : Sklearn, Python programlama dili için geliştirilmiş bir makine öğrenimi kütüphanesidir. Bu kütüphane, veri setleri üzerinde çeşitli makine öğrenimi algoritmalarını uygulamak, model eğitmek, tahmin yapmak ve model performansını değerlendirmek için kullanılır. Sklearn, regresyon, sınıflandırma, kümeleme, boyut indirgeme ve diğer makine öğrenimi teknikleri için kullanışlı araclar sunar.

Daha sonra dataseti csv dosyasından pandas kütüphanesi ile okunmaktadır ve df adılı bir değişkene kayıt etmektir. Ardından datasetin içerikleri görebilmek için ve ön işleme yapmak için çeşitli fonksyonları kullanılarak gösterildi.

1. df.info() methodu, veri setinde bilgi sağlar. Bu method, veri setinde kaç satır ve sütun olduğunu, her sütundaki veri tipini, veri setinde kaç null değer olduğunu ve her sütunda kaç tane değer olduğunu gösterir.
2. df.describe() methodu, veri setindeki sayısal değişkenler için özet istatistikleri sağlar. Bu method, sayısal değişkenler için ortalama, standart sapma, minimum, maksimum, 1. kuartil, 3. kuartil ve ortanca gibi değerleri gösterir.
3. df.duplicated() methodu, veri setindeki özdeş kayıtları bulur. Bu method, her kayıt için True veya False değeri döndürür. Eğer kayıt özdeş ise True, değilse False değerini döndürür.
4. df.isnull().sum() methodu, veri setindeki null değerlerin sayısını verir. Bu method, her sütundaki null değerlerin sayısını gösterir.

Bunları uyguladıktan sonra datasetin içreklerinde boş veri yada data tipi değiştirilmesi gereken bir veri olmadığını görürüz.

Sonraki aşamada aşırı veri analizi yapılacaktır. Bunun için, ilk olarak kan şeker seviyelerin dağılımlarını bakarak 300 ve 350 değerlerin aşırı olduğunu görülmektedir, sebebi ise diğer kan şeker seviyelerini 61 167 arasında olup dağılımların dışında ve arasında boşluk olduğunu görülmektedir.

Ayrıca öznitelik sütünlerindeki aşırı veri bulmak için MAD metodu kullanıldı. Bu şeklide veri setin son hali 603 satır × 4936 sütün olarak güncellendi.

Veri ön işlemenin son aşaması ise özellik seçimidir. Bu aşamada, istenen bir model için en yüksek performans gösteren öznitelikleri seçilecektir. Bunun için sklearn kütüphanesinden SelectFromModel fonksiyonu kullanılacaktır. istenen modeli ve threshold ile fonksiyonu oluşturulmaktadır. ardından X dataları fit edilir. Sonuç olarak yeni X dataler döndürür. Bu şekilde sütün sayısı 4936den 662’e kadar inmiş olacaktı. Ve en son bu X matrisi normalize edilecektir.

İkinci kısımda modeli eğitmek için X ve y train ve test olarak 80% ve 20% olarak ayrılacaktır. Daha sonra lasso modeli alpha,random\_state ve max\_iter değerleri ile oluşturulmaktadır. Daha sonra modelin eğitmesi için X train ve y train dataları ile fit edilmektedir. Modelin başarısını görmek için X test dataların y pred tahminleri yapılacaktır. gelince R-squared ve MSE skorların Sonuçları kabul edilen seviyesine gelinceye kadar alpha ve max\_iter değerlerini değiştirilecektir. Bu çalışmada R-squared: 0.9541307056561931 ve MSE: 45.220112715532274 olarak elde edilmiştir.

R-squared değeri, modelin veriye ne kadar iyi uyduğunu ölçen bir değerdir. R-squared değeri 0 ile 1 arasında olur ve 1'e ne kadar yakınsa, modelin veriye o kadar iyi uymuş olduğu anlamına gelir. Bu nedenle, elde edilen R-squared değeri 0.9541307056561931, modelin veriye oldukça iyi uymuş olduğunu gösterir.

MSE (Mean Squared Error), modelin tahminlerinin gerçek değerlerden ne kadar farklı olduğunu ölçen bir değerdir. MSE değeri, gerçek değerler ile tahmin edilen değerler arasındaki karelerin ortalamasıdır. Bu nedenle, düşük MSE değeri, modelin tahminlerinin gerçek değerlerle iyi eşleştiği anlamına gelir. Bu nedenle, elde edilen MSE değeri 45.220112715532274, modelin tahminlerinin gerçek değerlere oldukça yakın olduğunu gösterir.

Son olarak cross validation yöntemi uygulandı. Cross validation, modelin genel performansını ve tahmin güvenliğini ölçmek için kullanılan bir yöntemdir. Bu yöntem, veri setini belirli bir oranlarda parçalara bölerek, her parçada eğitim ve test işlemlerini gerçekleştirir. Bu sayede, modelin performansının ölçülmesinde veri setinin bütünlüğü korunur ve modelin genel performansı daha doğru bir şekilde ölçülür. Ayrıca, cross validation yöntemi, modelin overfitting (aşırı uyum) durumunu da tespit etmemize olanak tanır. Overfitting, modelin eğitim verileriyle çok iyi uyum sağlamasına rağmen, gerçek verilerde iyi bir performans sergilememesidir. Bu nedenle, cross validation yöntemi, modelin genel performansını ve tahmin güvenliğini ölçerken, aynı zamanda overfitting durumunu da tespit etmemize olanak tanır.

Bu çalışmada 10 fold Cross validation uygulandı ve sonuçları Mean score: 0.8430867972068855 Standard deviation: 0.15003064831126262. Bu sonuçlar, kullanılan modelin cross validation yöntemi ile genel performansının ortalama değerinin 0.84 olduğunu ve standart sapmasının 0.15 olduğunu gösterir. Standart sapmasının düşük olması, modelin genel performansının daha istikrarlı olduğunu ve farklı parçalarda elde edilen skorlar arasındaki farkın düşük olduğunu gösterir. Ancak, ortalama skorun 0.84 olması, modelin genel performansının daha yüksek olması gerektiğini göstermektedir. Bu nedenle, modelin daha iyi performans göstermesi için ek çalışmalar yapılması gerekebilir.

Son olarak tahmin edilen ve gerçek y bazı değerleri bir tablo olarak karşılaştırılabilir .

|  |  |
| --- | --- |
| y\_pred | y\_test |
| 65 | 67 |
| 158 | 156 |
| 121 | 111 |
| 136 | 135 |
| 155 | 156 |
| 114 | 111 |
| 125 | 135 |
| 67 | 61 |

Bu sonuçlar y\_pred ve y\_test değerlerinin karşılaştırmalı olarak gösterildiği bir tablo. Bu tablo, tahmin edilen değerler ile gerçek değerler arasındaki farkı gösterir. Örneğin, 65 olarak tahmin edilen değer 67 gerçek değerdir veya 158 olarak tahmin edilen değer 156 gerçek değerdir. Bu tablo, tahmin edilen değerler ile gerçek değerler arasındaki ne kadar yakın olduğunu gösterir ve modelin performansını değerlendirmek için kullanılabilir.

# Gaussian-based sınıflandırıcı kullanılarak EEG sinyallerinden duygu sınıflandırılması

## Gaussian-based sınıflandırıcı

Gaussian-based sınıflandırıcı, genellikle Gaussian Naive Bayes olarak adlandırılan veya sadece Naive Bayes olarak adlandırılan bir sınıflandırma algoritmasıdır. Bu algoritma, veri noktalarının belirli bir sınıf için olasılıklarını hesaplamak için Gaussian (normal) dağılım fonksiyonlarını kullanır. Algoritma, veri noktalarının her bir özellik için ayrı ayrı Gaussian dağılım fonksiyonlarını kullanarak bu olasılıkları hesaplar. Son olarak, tahmin edilen sınıf, olasılıklar arasındaki en yüksek olasılık değerine sahip olan sınıftır.

Gaussian-based sınıflandırıcı, verileri sınıflandırmak için iki temel adımdan oluşur. İlk olarak, sınıflandırmak istediğimiz verilerin her bir sınıfı için bir Gaussian dağılımı oluşturulur. Bu, verilerin her bir sınıf için ortalama ve varyans değerlerini belirlemek için kullanılır. İkinci olarak, yeni gelen veriler için her bir sınıf için oluşturulan Gaussian dağılımları kullanılarak verilerin sınıflandırılması gerçekleştirilir. Bu adımda, yeni gelen verinin her bir sınıf için olasılık değerleri hesaplanır ve en yüksek olasılık değerine sahip sınıf veriye atanır.

## EEG dataseti

Bu veri seti farklı kişilerden EEG sensorlarını kullanılarak elde edilmiştir. Bu kişilerin duyguları negatif pozitif ve normal olarak etiketlendi. Toplam 2132 satır × 2549 sütünden oluşur.

## Çalışma adımları

Bu çalışmadaki ilk birkaç aşama önceki çalışmadaki aynısıdır. Kütüphaneleri import etmek. Ve sonra data seti okumak ve göstermektir. Ancak bu veri için etiket sütünü 3 farklı sınıf içermektedir. Analizi daha kolay ve hızlı hale getirmek için sadece negatif ve pozitif sınıflarını kullanılacaktır. Ayrıca Negatif ve Pozitif yazılan sınıfları 0 ve 1 olarak değiştirilir. Bu şekilde data seti 1416 satır × 2549 sütün içermektedir.

Önceki çalışmada aşırı veri analiz yöntemini kullanılarak veri setin aşırı veri içeren sütünler silinmiştir. . Bu şekilde data seti sadece 1106 satır × 2549 sütünü kalmıştır. Son olarak X dataları normalize edilmiştir.

Modeli eğitmek için X train ve y train kullanıldı ve modele fit edildi. Daha sonra X test kullanılarak y pred tahmin edildi ve doğruluk oranı hesaplandı. Sonuç acc=0.9324324324324325. doğruluk değeri Gaussian-tabanlı bir sınıflandırıcı için iyi bir performans gösterir. Gaussian-tabanlı bir sınıflandırıcı, veriyi modellemek ve tahmin yapmak için Gaussian dağılımını kullanan bir olasılık tabanlı sınıflandırıcı türüdür. Bu durumda, model, örneklerin %93.24'sünü doğru bir şekilde tahmin etti, bu da yüksek bir doğruluk düzeyine sahip olduğu anlamına gelir. Bu doğruluk, modelin verideki alt düzenlemeleri öğrendiği ve yeni görülmeyen verilerde doğru tahminler yapabileceği için iyi bir göstergedir.

Ancak sadece doğruluk oranı gerçek bir sonuç verilmeyebilir. Bu yüzden daha detaylı bir sonuç raporuna ihtiyaç duyulmaktadır. Sınıflandırma raporun sonuçları aşağıdaki gibidir:

precision recall f1-score support

0 0.92 0.95 0.94 114

1 0.94 0.92 0.93 108

accuracy 0.93 222

macro avg 0.93 0.93 0.93 222

weighted avg 0.93 0.93 0.93 222

Bu tablo, ikili sınıflandırma modelinin değerlendirme metriklerini gösterir. Sütunlar, doğruluk, duyarlılık, f1-skoru ve desteği temsil ederken, satırlar ise iki sınıf etiketini (0 ve 1) temsil eder.

Doğruluk, model tarafından yapılan pozitif tahminlerin doğru olanlarının oranıdır. Sınıf 0 için, modelin doğruluk değeri 0.92'dir, bu modelin yaptığı pozitif tahminlerin %92'sinin doğru olduğu anlamına gelir.

Duyarlılık, gerçek pozitif tahminlerin verideki gerçek pozitif örneklerinin oranıdır. Sınıf 0 için, modelin duyarlılık değeri 0.95'dir, bu verideki gerçek pozitif örneklerin %95'inin model tarafından doğru tahmin edildiği anlamına gelir.

F1-skoru, doğruluk ve duyarlılığın armonik ortalamasıdır ve yüksek değer hem doğruluk hem de duyarlılığın yüksek olduğunu gösterir. Sınıf 0 için, modelin f1-skoru 0.94'dür, bu modelin doğruluk ve duyarlılık arasında iyi bir dengeye sahip olduğu anlamına gelir.

Destek, verideki her sınıf etiketinin örnek sayısıdır. Sınıf 0 için 114 örnek, sınıf 1 için ise 108 örnek vardır.

Modelin genel doğruluğu 0.93'dür, bu modelin örneklerin %93'ünü doğru bir şekilde tahmin ettiği anlamına gelir.

Özet olarak, tablo modelin her iki sınıf etiketi için de iyi bir performansa sahip olduğunu, yüksek doğruluk, duyarlılık ve f1-skoru ile birlikte iyi bir doğruluk oranına sahip olduğunu göstermektedir.

Daha sonra 10 fold cross validation uygulandı ve sonuçları : Mean score: 0.9096314496314497 Standard deviation: 0.017404998752026963.

10-fold cross validation, bir makine öğrenimi modelinin performansını değerlendirmek için kullanılan bir tekniktir. Bu teknik veriyi eşit parçalara ya da "katmanlara" böler ve modeli 9 katmanda eğitirken, geriye kalanını değerlendirir. Bu işlem 10 kez tekrar edilir ve her seferinde farklı bir katman değerlendirme kümesi olarak kullanılır.

Bu durumda, sonuçların ortalama skoru ve standart sapması modelin iyi bir performansa sahip olduğunu gösterir. 0.9096314496314497 ortalama skoru, modelin iyi bir şekilde çalıştığını gösterir, çünkü 1'e yakın olduğu için, maksimum skor olarak kabul edilir. Standart sapması 0.017404998752026963, sonuçların varyasyonunu veya dağılımını gösterir. Düşük bir standart sapması, modelin performansının tüm katmanlar üzerinde tutarlı olduğunu gösterir, bu da modelin sağlamlığı açısından iyi bir göstergedir.

Özet olarak, 10-fold cross validation sonuçları, modelin ortalama skoru 0.9096314496314497 ve düşük standart sapması 0.017404998752026963 ile iyi bir performansa sahip olduğunu gösterir. Bu, modelin performansının tüm katmanlar üzerinde tutarlı olduğunu ve modelin sağlamlığı açısından iyi bir göstergedir.

Ayrıca ROC eğrisi analizi yapılmıştır. AUC değeri 0.94tür. ROC (Receiver Operating Characteristic) eğrisi, ikili sınıflandırıcı sistemin performansının görsel bir temsilidir. Eğri, ayırıcı eşik değeri değiştirildiğinde gerçek pozitif oranı (TPR) ve yanlış pozitif oranı (FPR) arasında çizilir. Eğrinin altındaki alan (AUC) sınıflandırıcının genel performansının yaygın olarak kullanılan bir ölçüsüdür, AUC 1 mükemmel bir sınıflandırıcıyı temsil ederken, AUC 0.5 rastgele tahmin yapmış bir sınıflandırıcıyı temsil eder.

0.94 AUC değeri, bir sınıflandırıcı için çok iyi bir performans gösterir. 0.94 değeri 1'e yakın olup mükemmel bir sınıflandırıcıyı ifade eder. AUC değeri 1'e ne kadar yakın olursa, sınıflandırıcı pozitif ve negatif durumları ayırt etme konusunda o kadar iyidir. Bu, sınıflandırıcının pozitif durumları %94 oranında doğru olarak tanıladığını ve negatif durumları %94 oranında doğru olarak tanıladığını ifade eder.

Son olarak da confusion matrisi çizildi:

[[108 6]

[ 9 99]]

Bu karışıklık matrisinde, üst sol hücredeki 108 rakamı gerçek pozitiflerin sayısını temsil eder, yani "0" sınıfı olarak doğru olarak sınıflandırılmış olan gözlemlerin sayısını.

Alt sağ hücredeki 99 rakamı gerçek negatiflerin sayısını temsil eder, yani "1" sınıfı olarak doğru olarak sınıflandırılmış olan gözlemlerin sayısını.

Üst sağ hücredeki 6 rakamı yanlış pozitiflerin sayısını temsil eder, yani "0" sınıfı olarak yanlış olarak sınıflandırılmış olan gözlemlerin sayısını.

Alt sol hücredeki 9 rakamı yanlış negatiflerin sayısını temsil eder, yani "1" sınıfı olarak yanlış olarak sınıflandırılmış olan gözlemlerin sayısını.

# PCA analizi

PCA (Principal Component Analysis) analizi, veri setlerinde gizli yapıları keşfetmek ve anlamak için kullanılan bir veri madenciliği yöntemidir. Bu yöntem, veri setindeki değişkenler arasındaki korelasyonları kullanarak, veri setinin en önemli yönlerini veya "ana bileşenlerini" belirler. Bu, veri setinin özetlenmesi ve anlaşılması için kullanışlıdır. Özellikle, veri setinde birçok değişken olduğunda veya veri setinin özellikleri arasında karmaşıklık olduğunda, PCA analizi veri setinin anlaşılması için kullanışlıdır.

PCA analizi genellikle veri ön işleme, veri madenciliği veya makine öğrenimi gibi alanlarda kullanılır. Örneğin, görüntü işlemede, ses işlemede veya biyometrik tanıma gibi uygulamalarda PCA analizi kullanılır. Ayrıca, çok sayıda değişken içeren veri setlerinde, PCA analizi, veri setinin daha az değişkenle özetlenmesini ve anlaşılmasını sağlar. Bu, özellik seçimi veya veri azaltma gibi diğer veri ön işleme yöntemlerine göre daha hızlı ve kolaydır.

PCA analizi, veri setindeki değişkenler arasındaki korelasyonları kullanarak, veri setinin en önemli yönlerini belirler. Bu yönler, veri setinin en önemli özelliklerini temsil eder ve veri setinin anlaşılması için kullanışlıdır. PCA analizi ayrıca, veri setinin gürültü veya gereksiz özelliklerden arındırılmasına da olanak tanır.

Sonuç olarak, PCA analizi, veri setindeki gizli yapıları keşfetmek ve anlamak için kullanılan önemli bir veri madenciliği yöntemidir. Bu yöntem, veri setinin özetlenmesi, anlaşılması ve gürültü veya gereksiz özelliklerden arındırılması için kullanılabilir.

## Çalışma aşamaları

Bu çalışmada önceki çalışmadaki EEG dataseti kullanıldı ve aynı Gaussian-based sınıflandırıcı üzerinde test edildi. PCA (Principal Component Analysis) sınıfından bir modeli oluşturulur. Bu modelde, n\_components parametresi 2 olarak ayarlanmıştır. Bu parametre, PCA modelinin kaç ana bileşen üreteceğini belirler. Bu örnekte, 2 ana bileşen üretilecektir.

Sonra, fit() metodu ile veri seti (X) PCA modeline uygulanır. Bu, veri setindeki değişkenler arasındaki korelasyonları kullanarak, veri setinin en önemli yönlerini belirler.

transform() metodu ile veri seti, boyut azaltma uygulanarak (X\_pca) yeniden şekillendirilir. Bu, veri setinin en önemli yönlerini temsil eden ana bileşenleri kullanarak veri setini yeniden şekillendirir.

explained\_variance\_ratio\_ metodu, üretilen ana bileşenlerin veri setindeki varyansı açıkladığı oranı döndürür.

components\_ metodu, üretilen ana bileşenleri döndürür. Bu, veri setinin en önemli yönlerini temsil eden değişkenlerin ağırlıklarını içerir.

PCA uygulanmış veri seti (X\_pca) ile scatter plot oluşturulur. Scatter plot, farklı sınıflar için farklı renklerle verileri görselleştirir.

y==0 olarak filtre edilen verilerin ilk ana bileşen değerleri X\_pca[y==0, 0] ve ikinci ana bileşen değerleri X\_pca[y==0, 1] olarak kullanılır ve mavi renkte görselleştirilir. Benzer şekilde y==1 olarak filtre edilen veriler de kırmızı renkte görselleştirilir.

x ekseni için 'First Principal Component', y ekseni için 'Second Principal Component' etiketleri kullanılır ve başlık olarak 'PCA Plot' kullanılır. Ayrıca grafikte bir açıklama yer alır "Classes"

Son olarak grafik görüntülenir sunulur.

Chart, scatter chart

Description automatically generated

İki sınıf olan N ve P, iki boyutlu bir alana projekte edildiğinde, belirli bir derecede ayrılabilir. Diğer gözlemler, N sınıfının P sınıfı içinde yayılmış. bunun için 2li PCA dogruluğu nisbeten düşük olacaktır. Ancak daha yüksek boyuta çakildiğinde daha iyi hale gelebilir. Bunu görmek için 2 den 10 kadar bir döngü içerisinde PCA codları uygulanacak ve n\_components parametresi her bir döngüde değişecektir ve dogruluk hesplanacaktır. Tüm dogrulukları grafik üzerinden gösterildi.

Chart, line chart

Description automatically generated