

Hazırlayan: Özgür Durak

Öğrenci No: 170101012

Proje Türü: Classfication

Proje Amacı: Bir kişinin yılda 50.000 ’den fazla kazanıp kazanmadığını tahmin eden sınıflama

**1.Proje hakkında**

Proje insanların gelirlerini , eğitimleri ,meslekleri, ülkeleri gibi kriterlerin bulunduğu veri setini kullanarak bir kişinin yıllık kazancının 50.000 ‘in üzerinde olup olmadığı sınıflandırma problemidir.

**2. Makine Öğrenmesi ve Projeye Giriş**

Makine öğrenmesi, insanların öğrenme şekillerini taklit etmek için veri ve algoritmaların kullanımına odaklanıp doğruluğunu kademeli olarak artıran bir yapay zeka(AI) ve bilgisayar bilimi dalıdır.Genel olarak, makine öğrenmesi algoritmaları bir tahmin ya da sınıflandırma yapmak için kullanılır.

Bir makine öğrenmesi metodu tahminde bulunmak için bir çıktı üretir. Bu çıktı kategorik ise **sınıflandırma** (classification) ve eğer nümerik ise **regresyon** (regression) denir. Açıklayıcı bir modelleme olan **kümeleme** (Clustering) ise benzer gözlemleri aynı kümelere atama işlemidir.

**Gözetimli Öğrenme**

Eğitim verileri üzerinden bir fonksiyon üreten bir makine öğrenmesi tekniğidir. Başka bir deyişle, bu öğrenme tekniğinde girdilerle (işaretlenmiş veri – labelled data) ile istenen çıktılar arasında eşleme yapan bir fonksiyon üretir. Eğitim verisi hem girdilerden hem çıktılardan oluşur. Fonksiyon, sınıflandırma (classifiction) veya eğri uydurma (regression) algoritmaları ile belirlenebilir.

Bu yöntemde işaretlenmemiş (unlabelled) veri üzerinden bilinmeyen bir yapıyı tahmin etmek için bir fonksiyon kullanan makine öğrenmesi tekniğidir. Burada girdi verisinin hangi sınıfa ait olduğu belirsizdir.

**Karmaşıklık matrisi**

Bir sınıflama algoritmasının performasını özetlemek için kullanılan bir tekniktir. Karmaşıklık matrisi hesaplamak Sınıflama Modelinizin neleri doğru ve hatalı yaptığını anlamada yardımcı olur.

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, makbuz içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Accuracy = Doğru Tahminlerin Sayısı / Tüm Tahminlerin Sayısı**

**Precision = Pozitif olarak tahmin edilenlerin gerçekte kaçta kaçı doğru.**

**Specifiy = Negatif olarak tahmin edilenlerin gerçekte kaçta kaçı negatif olduğu tahmin ediliyor.**

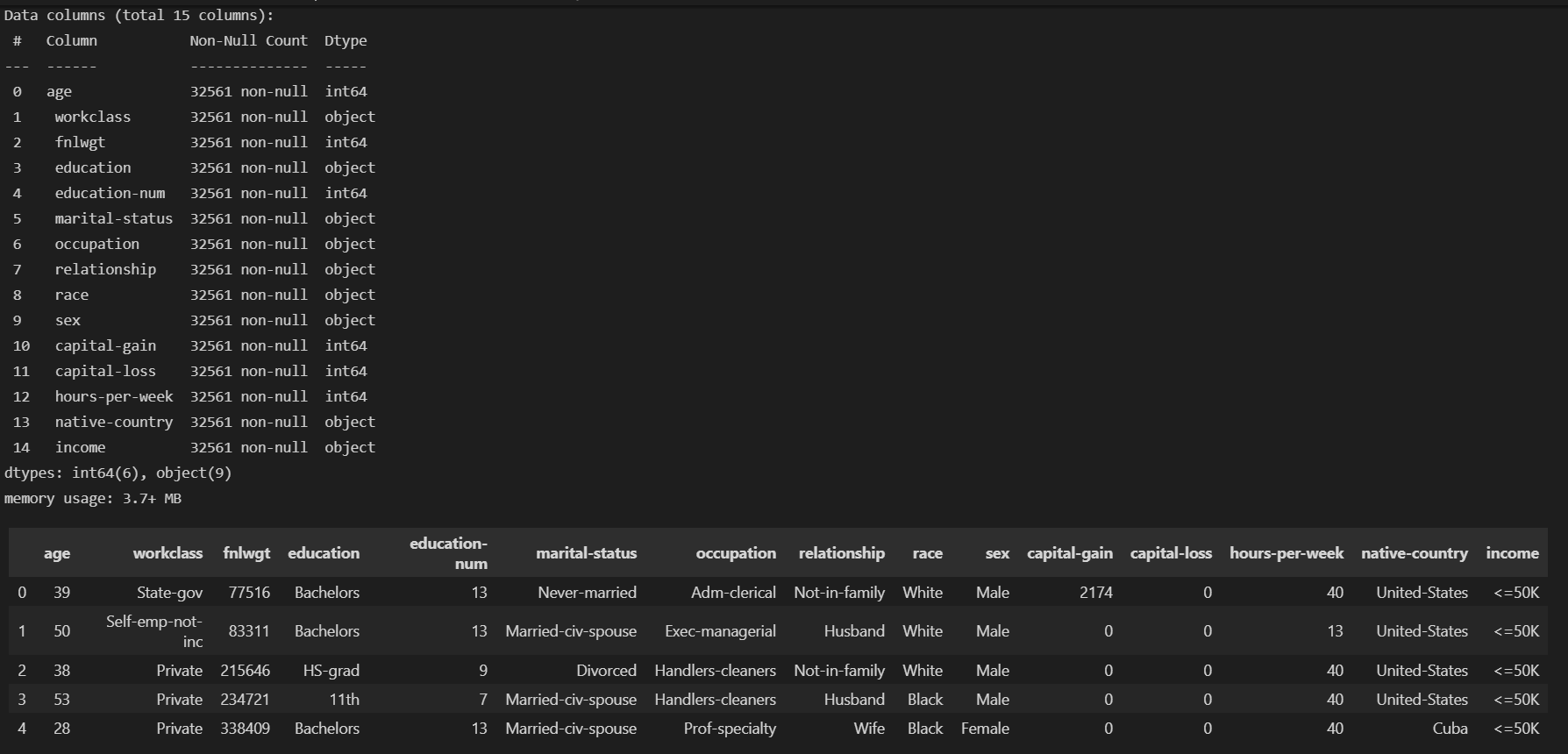
**Recall = Toplam pozitifin yüzde kaçının pozitif olduğu tahmin ediliyor.**

F1 Score,Precision ve Recall değerlerinin ağırlıklı (harmonik) ortalamasıdır.

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**3. Veri Seti**

****

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**4. Veri Seti Ön İşleme(Preprocessing)**

Veri setimizdeki numerik olmayan yani string değerleri Sklearn kütüphanesinden Label Encoder fonksiyonu ile numerik(sayısal)değerleri dönüştürdük. Bu sayede veri setimizdeki bütün değerler integer olarak tutuluyor.

Makine öğrenmesi modelleri kategorik değişkenleri algılayamadığı için ‘object’ tipindeki değişkenleri numeric(int) tipindeki değişkenlere dönüştürdük

**metin içeren bir resim

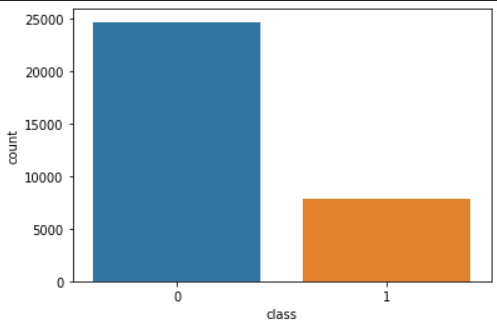
Açıklama otomatik olarak oluşturuldu**

Veri setimizdeki boş(null) olan değerleri kontrol ettim ve toplamını ekrana yazdırdım.

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Hedef değişkenlerinin verisinin görselleştirilmesi .Veri setindeki 50.000 üzeri maaş alanlar için ‘1‘ , alamayanlar için ‘0’ seklinde grafik çıktısı.

****

Veri setinin string değerleri ile işlemlerin ardından verinin daha okunabilir hale gelmesi için normalizyon ile özellik aralığı(feature\_range) (1,0) aralığına sınırlandırılmıştır.

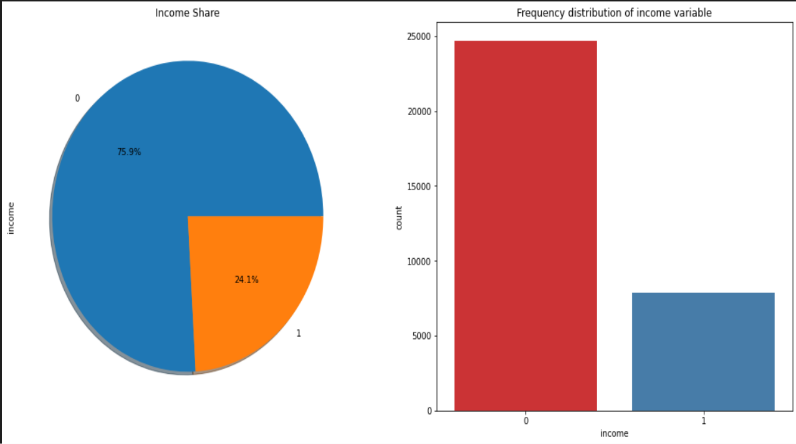
**metin, ekran, siyah, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu**

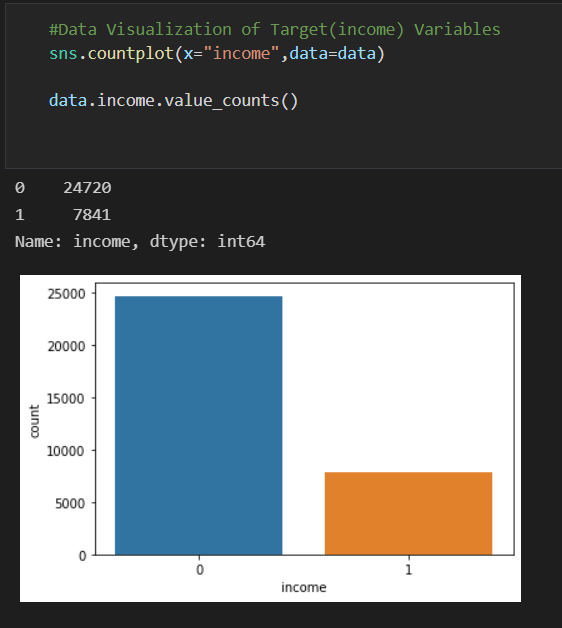
Burada income(gelir) verimizin frekans dağılımını tablolaştırdık.

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

****

İncome 0 ve 1 değeleri için kontrol edilmiştir. bu grafikte görüldüğü gibi iki income classı arasında büyük veri farkı bulunmaktadır. Böyle bir durumda model 0 income yönelimli olacağından bu fark düşürülmektedir. Bundan dolayı verilerin eşit sayıya getirilmesi 24720 adet bulunan 0 income 7841 adete rastgele olarak düşürülmüştür.(random under sampling)



metin içeren bir resim

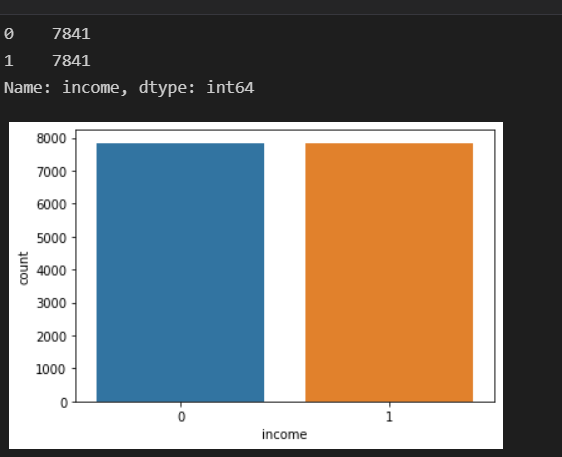
Açıklama otomatik olarak oluşturuldu

Burada shuffle ile verilerimizi karıştırdık

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Under Sampling sonucu



**Train ve Test Verilerinin Hazırlanması**

   Model içerisinde kullanmak amacıyla verinin 2 ayrı katmana bölünmesi gerekmektedir.

**Train:** Veri setindeki bu katman modelin eğitilmesi amacıyla kullanılır.

**Test:** Oluşturulmuş olan modelin performansının tespit edilmesi için ayrılan alandır. Bu bölümde modele verilerek tahmin edilen değerlerin, olması gereken değerler ile karşılaştırılması yapılmaktadır.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu model içerisinde kullanmak amacıyla veri seti %30 test ve %70 train için bölünmüştür.

Train ve test boyutlarının çıktısını yukarıda görüldüğü gibidir.

**4. Makine Öğrenmesi Sınıflama Algoritmaları ve Sonuçlar**

K-NN *( K-Nearest Neighbor)* algoritması en basit ve en çok kullanılan sınıflandırma algoritmasından biridir. K-NN **non-parametric (**parametrik olmayan **)**, **lazy**( tembel ) bir öğrenme algoritmasıdır. *lazy*kavramını anlamaya çalışırsak *eager learning*aksine lazy learning’in bir eğitim aşaması yoktur. Eğitim verilerini öğrenmez, bunun yerine eğitim veri kümesini *“ezberler”*. Bir tahmin yapmak istediğimizde, tüm veri setinde en yakın komşuları arar.

Algoritmanın çalışmasında bir ***K*** değeri belirlenir. Bu K değerinin anlamı bakılacak eleman sayısıdır. Bir değer geldiğinde en yakın K kadar eleman alınarak gelen değer arasındaki uzaklık hesaplanır. Uzaklık hesaplama işleminde genelde **Öklid fonksiyonu**kullanılır. Öklid fonksiyonuna alternatif olarak **Manhattan**, **Minkowski**ve **Hamming**fonksiyonlarıda kullanılabilir. Uzaklık hesaplandıktan sonra sıralanır ve gelen değer uygun olan sınıfa atanır.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

# Naive Bayes Classifier

Naive Bayes sınıflandırıcısı [Bayes teoreminin](https://tr.wikipedia.org/wiki/Bayes_teoremi" \o "Bayes teoremi) bağımsızlık önermesiyle basitleştirilmiş halidir. 1812 yılında [Thomas Bayes](https://en.wikipedia.org/wiki/Thomas_Bayes) tarafından bulunan koşullu olasılık hesaplama formülüdür. **Bayes teoremi**, olasılık kuramı içinde incelenen önemli bir konudur. Bu **teorem** bir rassal değişken için olasılık dağılımı içinde koşullu olasılıklar ile marjinal olasılıklar arasındaki ilişkiyi gösterir.

Bayes teoremi aşağıdaki [denklemle](https://tr.wikipedia.org/wiki/Denklem) ifade edilir;{\displaystyle P(A|B)={\frac {P(B|A)\,P(A)}{P(B)}}}

P(*A*|*B*) ; B olayı gerçekleştiği durumda A olayının meydana gelme olasılığıdır (bakınız [koşullu olasılık](https://tr.wikipedia.org/wiki/Ko%C5%9Fullu_olas%C4%B1l%C4%B1k))

P(*B*|*A*) ; A olayı gerçekleştiği durumda B olayının meydana gelme olasılığıdır

P(A) ve P(B) ; A ve B olaylarının [önsel olasılıklarıdır](https://tr.wikipedia.org/wiki/%C3%96nsel_olas%C4%B1l%C4%B1k).

Naive Bayes sınıflandırıcısının temeli Bayes teoremine dayanır. **lazy**( tembel ) bir öğrenme algoritmasıdır aynı zamanda dengesiz veri kümelerinde de çalışabilir. Algoritmanın çalışma şekli bir eleman için her durumun olasılığını hesaplar ve olasılık değeri en yüksek olana göre sınıflandırır. Az bir eğitim verisiyle çok başarılı işler çıkartabilir. Test kümesindeki bir değerin eğitim kümesinde gözlemlenemeyen bir değeri varsa olasılık değeri olarak 0 verir yani tahmin yapamaz.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Random Forest Classfier

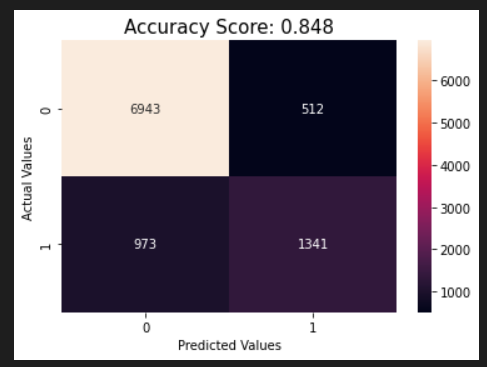
Rastgele Orman algoritması denetimli bir sınıflandırma algoritmasıdır. (Supervised classification algorithm).Fakat geleneksel yöntemlerden biri olan karar ağaçlarının en büyük problemlerinden biri aşırı öğrenme-veriyi ezberlemedir (overfitting). Rassal orman modeli bu problemi çözmek için hem veri setinden hem de öznitelik setinden rassal olarak 10'larca 100'lerce farklı alt-setler seçiyor ve bunları eğitiyor. Bu yöntemle 100'lerce karar ağacı oluşturuluyor ve her bir karar ağacı bireysel olarak tahminde bulunuyor.

Random forest modelinin diğer bir özelliği bize özniteliklerin ne kadar önemli olduğunu vermesi. (Bir özniteliğin önemli olması demek o özniteliğin bağımlı değişkendeki varyansın açıklanmasına ne kadar katkı yaptığıyla alakalı.)

n\_estimators : Bu, maksimum oylama veya tahmin ortalamalarını almadan önce oluşturmak istediğiniz ağaç sayısıdır. Daha fazla sayıda ağaç size daha iyi performans sağlar ancak kodunuzu yavaşlatır.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

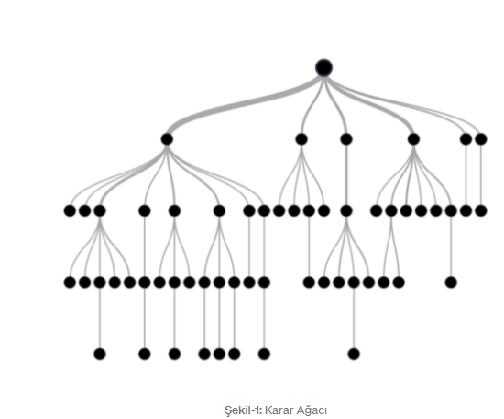


Karar Ağaçları(Desicion Tree)

Ağaç tabanlı öğrenme algoritmaları, en çok kullanılan gözetimli öğrenme algorimalarındandır. Genel itibariyle ele alınan bütün problemlerin (sınıflandırma ve regression) çözümüne uyarlanabilirler. Karar agaçları, tesadüfi orman, gradyen güçlendirme (gradient boosting) gibi yöntemler, her türlü veri bilimi problemlerinde yaygın bir şekilde kullanılmaktadırlar. Bu nedenle veri analistleri için bu algoritmaları öğrenmek ve kullanmak çok önemlidir.

Bir karar ağacı, çok sayıda kayıt içeren bir veri kümesini, bir dizi karar kuralları uygulayarak daha küçük kümelere bölmek için kullanılan bir yapıdır. Yani basit karar verme adımları uygulanarak, büyük miktarlardaki kayıtları, çok küçük kayıt gruplarına bölerek kullanılan bir yapıdır. Karar ağaçları, parametrik olmayan bir yöntem olarak düşünülebilir.

Ezbere öğrenme yaşanabilir (“over-fitting”). Bu problemin çözümü için model parametrelere kısıtlamalar ve budama gibi yöntemler kullanılabilir. Budama işlemi, az sayıda nesneyi barındıran yaprak düğümlerin karar ağacı grafiğinden atılmasını ifade etmektedir

 metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

### Categorical Naïve Bayes

Her bir özellik için kategorik dağılım kabul eden ayrık özniteliklerle sınıflandırmaya uygundur. Özellikler, her kategori benzersiz bir sayı ile eşlenecek şekilde etiket kodlama teknikleri kullanılarak kodlanmalıdır.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

# AdaBoost

Adaptive Boosting'in kısaltması olan AdaBoost algoritması, Machine Learning'de Ensemble Method olarak kullanılan bir Boosting tekniğidir. Ağırlıklar her bir örneğe yeniden atandığından ve yanlış sınıflandırılan örneklere daha yüksek ağırlıklar atandığından buna Uyarlamalı Yükseltme denir. Güçlendirme, denetimli öğrenme için sapmanın yanı sıra varyansı azaltmak için kullanılır. Öğrencilerin sırayla büyümesi ilkesine göre çalışır. İlki dışında, sonraki her öğrenci daha önce yetiştirilen öğrencilerden yetiştirilir. Basit bir deyişle, zayıf öğrenenler güçlü olanlara dönüştürülür. AdaBoost algoritması, küçük bir farkla artırma ile aynı prensipte çalışır.

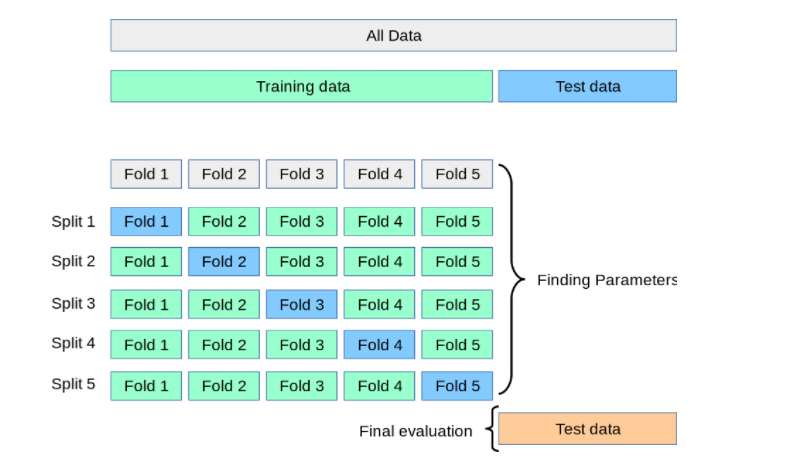
metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

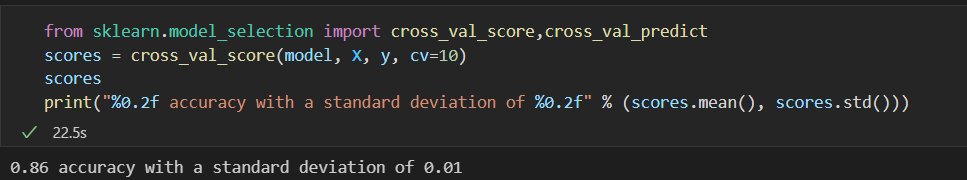
Veri setimize uyguladığımız Makine öğrenmesi algoritmaların en yüksek sonucu AdaBoost algoritması veriyor.

Cross-Validation

Cross-validation, makine öğrenmesi modelinin görmediği veriler üzerindeki performansını mümkün olduğunca objektif ve doğru bir şekilde değerlendirmek için kullanılan istatistiksel bir yeniden örnekleme(resampling) yöntemidir.



AdaBoost modelimizde 10 cross validation sonucunda Accuarcy 0.86 çıkmıştır



Referanslar

<https://www.kaggle.com/lodetomasi1995/income-classification>

<https://inblog.in/Categorical-Naive-Bayes-Classifier-implementation-in-Python-dAVqLWkf7E>

<https://machinelearningmastery.com/confusion-matrix-machine-learning/>

<https://towardsdatascience.com/performance-metrics-confusion-matrix-precision-recall-and-f1-score-a8fe076a2262>

<https://medium.com/data-science-tr/makine-%C3%B6%C4%9Frenmesi-dersleri-5-bagging-ve-random-forest-2f803cf21e07>

<https://medium.com/@ekrem.hatipoglu/machine-learning-classification-naive-bayes-part-11-4a10cd3452b4>

<https://medium.com/@ekrem.hatipoglu/machine-learning-classification-k-nn-k-en-yak%C4%B1n-kom%C5%9Fu-part-9-6f18cd6185d>

<https://www.mygreatlearning.com/blog/adaboost-algorithm/>

<https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-5-karar-a%C4%9Fa%C3%A7lar%C4%B1-c90bd7593010>

<https://medium.com/@sertacozker/boosting-algoritmalar%C4%B1-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-edac1174e971>