Convolutional Neural Network ve Naïve Bayes Algoritmaları ile Wine Veri Kümesini Sınıflandırma

Merve Karagöz   
Kocaeli Üniversitesi *Kocaeli,Türkiye*  
mervekaragoz124@gmail.com

*Özet*—Makine öğrenmesi tekniklerinden olan Naïve Bayes ve Convolutional Neural Network kullanılarak wine veri kümesini sınıflandırmak hedeflenmiştir. Sınıflandırıcı başarıları karşılaştırılarak en iyi doğruluk değerine ulaşılmıştır. Naïve Bayes sınıflandırıcısının türleri olan Gaussian Naïve Bayes, Multinomial Naïve Bayes ve Bernoulli Naïve Bayes kullanılmıştır. Ayrıca Convolutional Neural Network sınıflandırıcısının optimizer ve epoch değerleri değiştirilerek en iyi sonuca ulaşılmaya çalışılmıştır.

Anahtar Kelimeler— Sınıflandırıcı, Convolutional Neural Network, Naïve Bayes, veri kümesi

# GİRİŞ

Makine öğrenmesi, bilgisayarların bir başarım ölçütünü örnek veri veya geçmiş deneyim kullanarak eniyilemesi olarak tanımlanabilir [1]. Bu kapsamda belirli parametrelere sahip bir model tanımlanarak bir bilgisayar programının bir eğitim verisi veya geçmiş deneyimleri kullanarak bu parametreleri iyileştirecek şekilde çalıştırılması da öğrenme sürecini oluşturur.

Makine Öğrenmesi Bilgisayar Bilimleri, Mühendislik ve İstatistik gibi farklı bilim dallarından beslenir ve sınıflandırma, regresyon, kümeleme, yoğunluk kestirimi, örüntü tanıma, uç değer tespiti, sıkıştırma, bilgi çıkarma, vb. gibi farklı görevleri yerine getirmeye çalışır [2]. Yapılan projede sınıflandırma görevi yerine getirilmeye çalışılmıştır.

Makine öğrenmesi tekniklerinden Naïve Bayes ve Convolutional Neural Network kullanılarak wine veri kümesinin sınıflandırılması hedeflenmiştir. Naïve Bayes sınıflandırıcısının türleri olan Gaussian Naïve Bayes, Multinomial Naïve Bayes ve Bernoulli Naïve Bayes kullanılmıştır. Bu 3 Naïve Bayes sınıflandırıcısından hangisinden en iyi sonuç alınabileceği heaplanmışıtr. Convolutional Neural Network sınıflandırıcısı için döngü değerleri ve optimizer (iyileştirme) yöntemleri değiştirilerek en iyi sonuca ulaşılmaya çalışılmıştır. Sınıflandırıcılara ve veri kümesi detaylarına ilerleyen bölümlerde yer verilmiştir.

Bu rapor şu bölümlerden oluşmaktadır: 2. Bölümde yöntemler adı altında bu projede kullanılmış olan makine öğrenmesi teknikleri özetlenmektedir, 3. bölümde bu proje kapsamında sınıflandırılması yapılan wine veri kümesi anlatılmaktadır, 4. bölümde sonuç başlığı altında proje içinde kullanılan makine öğrenmesi tekniklerinin sonuçlarına ve sonuçların karşılaştırılmasına yer verilmiştir. Son bölümde ise yararlanılan kaynaklara yer verilerek rapor tamamlanmıştır.

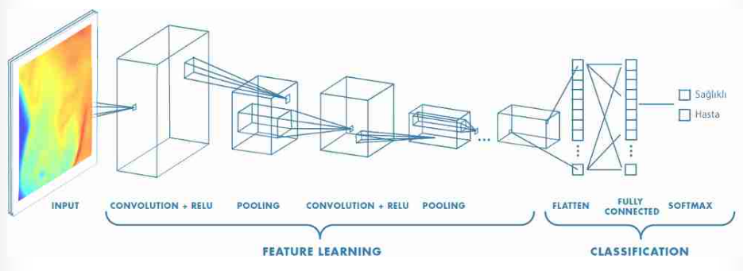
# YÖNTEM

Günümüzde farklı sınıflandırma problemleri için kullanılan makine öğrenmesi yöntemilerinden Convolutional Neural Network ve Naïve Bayes ile wine veri kümesini sınıflandırmak amacıyla yararlanılmıştır. Bu yöntemler farklı farklı özelliklere sahip olsalar da hepsinin ortak özelliği bir sınıflandırma probleminde kullanılabilmeleridir. Projede kullanılan her bir makine öğrenmesi yöntemine ilişkin bilgilere özet şeklinde aşağıda yer verilmiştir.

A. Convolutional Neural Network Sınıflandırıcı

Evrişimli sinir ağları (Konvolüsyonel sinir ağları da denmektedir.) çok katmanlı algılayıcıların bir türüdür. Hayvanların görme merkezinden esinlenilmiştir. Görüntü ve ses işleme, doğal dil işleme ve biyomedikal gibi alanlarda kullanılmakla beraber en iyi sonuçları görüntü işleme alanında vermiştir.

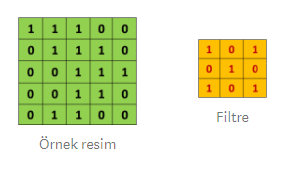
Aşağıdaki şekil (Resim 2.1), bir giriş görüntüsünü işlemek için CNN akışıdır ve nesneleri değerlerine göre sınıflandırır.



Resim 2.1

Convolutional Neural Network (Evrişimli sinir ağları) Convolutional,non-linearity, pooling, flatten ve fully connected katmanlarındn oluşmaktadır.

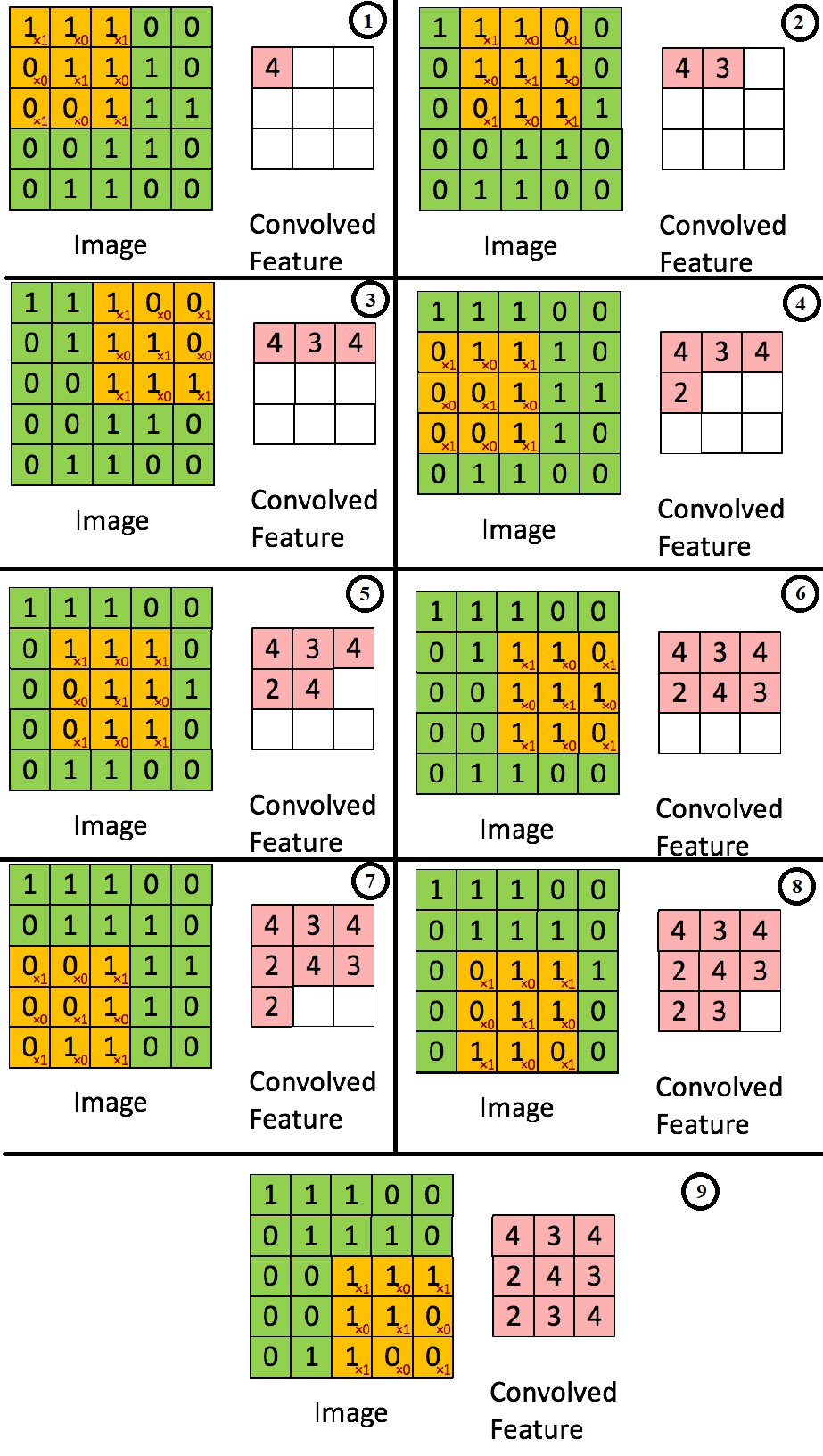
Convolutional katmanı CNN ‘nin ana yapı taşı olarak bilinmektedir. Resmin özelliklerini algılamaktan sorumludur. Bu katman, görüntüdeki düşük ve yüksek seviyeli özellikleri çıkarmak için resme bazı fitreler uygular. Örneğin, bu filtre kenarları algılayacak bir filtre olabilir. Bu filtreler genellikle çok boyutludur ve piksel değerleri içerirler.(5x5x3) 5 matrisin yükseklik ve genişliğini, 3 matrisin derinliğini temsil eder [3].  
Bu filtrenin uygulanışı Resim 2.2 de gösterilmiştir.



Resim 2.2

Örneğin basit olması için sadece 1 kanal işlenecektir.  
Resimin 5×5 boyutunda ve 1 ve 0 ‘lardan oluşan bir resim olduğunu varsayılırarak filtre 3×3 boyutunda oluşturulur.

Filtrenin uygulanışı Resim 2.3 ‘de görüldüğü üzere adım adım gösterilmiştir.



Resim 2.3

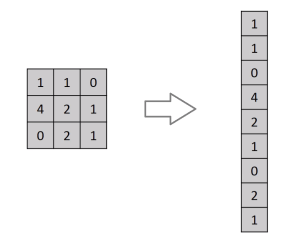
Öncelikle filtre, görüntünün sol üst köşesine konumlandırılır. Burada, iki matris arasında*(resim ve filtre)* indisler birbirisi ile çarpılır ve tüm sonuçlar toplanır, daha sonra sonucu çıktı matrisine depolanır. Ardından, bu filtreyi sağa 1 piksel (“basamak” olarak da bilinir) kadar hareket ettirip işlemi tekrarlanır. 1. satır bittikten sonra 2. satıra geçilir ve işlemler tekrarlanır. Tüm işlemler bittikten sonra çıktı matrisi oluşturulur. Burada çıktı matrisinin 3×3 olmasının nedeni 5×5 matrisinde 3×3 filtresi yatayda ve dikeyde 3 kez hareket etmesinden kaynaklanır.  
Eğer resim 6×4 ve filtre 3×3 boyutunda olsaydı çıkış matrisi 4×2 boyutunda olurdu.  
Çıktı matrisine genellikle ‘Feature Map’ denir. Filtre tarafından temsil edilen özellikte görüntünün bulunduğu yeri gösterir. Kısacası, filtreyi görüntü üzerinden hareket ettirerek ve basit matris çarpımını kullanarak, özellikler tespit edilir. Genellikle, birden çok özelliği tespit etmek için birden fazla filtre kullanlır, yani bir Cnn ağında birden fazla konvolüsyonel (Convolutional) katman bulunur.

Tüm Convolutional katmanlarından sonra genellikle Non-Linearity (doğrusal olmayan) katman gelir. Görüntüdeki doğrusallığın problem olma sebebi, tüm katmanlar doğrusal bir fonksiyon olabildiğinden dolayı Sinir Ağı tek bir perception gibi davranır. Yani sonuç, çıktıların linear kombinasyonu olarak hesaplanabilir.  
Bu katman aktivasyon katmanı (Activation Layer) olarak adlandırılır çünkü [aktivasyon fonksiyonları](https://rubikscode.net/2017/11/20/common-neural-network-activation-functions/" \t "_blank)ndan birini kullanılır. Sinir Ağı eğitiminin hızı konusunda en iyi sonuca ReLu fonksiyonu ile ulaşıldığı için bu fonksiyon aktivasyon fonksiyonu olarak tercih edilmektedir.

**ReLu Fonksiyonu f (x) = max (0, x) ‘dır.**

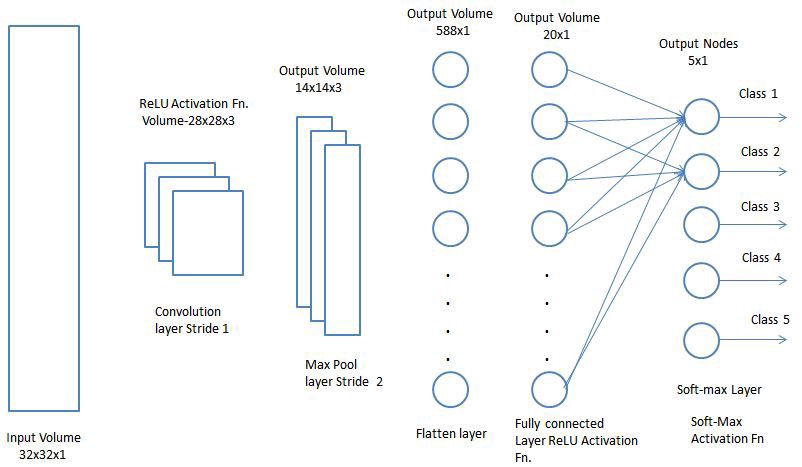
Pooling katmanı, convolutional katmanları arasına sıklıkla eklenen bir katmandır. Bu katmanın görevi, gösterimin kayma boyutunu ve ağ içindeki parametreleri ve hesaplama sayısını azaltmak içindir. Bu sayede ağdaki uyumsuzluk kontrol edilmiş olur. Birçok Pooling işlemleri vardır, fakat en popüleri max pooling’dir.

Flatting layer katmanının görevi, son ve en önemli katman olan Fully Connected Layer’ın girişindeki verileri hazırlamaktır. Genel olarak, sinir ağları, giriş verilerini tek boyutlu bir diziden alır. Bu sinir ağındaki veriler ise Convolutional ve Pooling katmanından gelen matrixlerin tek boyutlu diziye çevrilmiş halidir. (Resim 2.4)



Resim 2.4

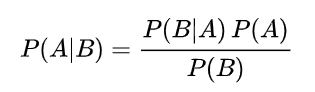
Fully-Connected Layer katmanı CNN’ in son ve en önemli katmanıdır. Veriler Flatting işleminden alınır ve Sinir ağı yoluyla öğrenme işlemini geçekleştirir.Son adım olarak ise Softmax Sınıflandırma tekniğini kullanarak sınıflandırma tamamlanır. Bu işlem Resim 2.5 ile görselleştirilerek anlatılmıştır.



Resim 2.5

B. Naïve Bayes Sınıflandırıcı

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 [4]. Naive Bayes sınıflandırıcısı  [Bayes teoreminin](https://tr.wikipedia.org/wiki/Bayes_teoremi) bağımsızlık önermesiyle basitleştirilmiş halidir. Bayes teoremi aşağıdaki [denklemle](https://tr.wikipedia.org/wiki/Denklem) ifade edilir;

 (1)

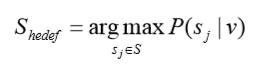
P(*A*|*B*); B olayı gerçekleştiği durumda A olayının meydana gelme olasılığıdır ( [koşullu olasılık](https://tr.wikipedia.org/wiki/Ko%C5%9Fullu_olas%C4%B1l%C4%B1k) )

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

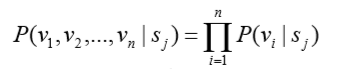
P(A) ve P(B); A ve B olaylarının [önsel olasılıklarıdır](https://tr.wikipedia.org/wiki/%C3%96nsel_olas%C4%B1l%C4%B1k).

Burada önsel olasılık Bayes teoremine [öznellik](https://tr.wikipedia.org/wiki/%C3%96znel) katar. Diğer bir ifadeyle örneğin P(A) henüz elde veri toplanmadan A olayı hakkında sahip olunan bilgidir. Diğer taraftan P(*B*|*A*) [ardıl olasılıktır](https://tr.wikipedia.org/w/index.php?title=Ard%C4%B1l_olas%C4%B1l%C4%B1k&action=edit&redlink=1) çünkü veri toplandıktan sonra, A olayının gerçekleşmiş olduğu durumlarda B olayının gerçekleşme ihtimali hakkında bilgi verir [5].

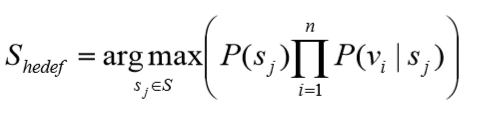
Bayes Teoremi sınıflandırma amaçlı kullanılırken en yüksek olasılıklı durum hedef sınıf olarak seçilir.

 (2)

Fakat girdi vektörlerinin (v) birden çok olduğu durumlarda Bayes formülü farklı bir forma dönüşür. Birçok özelliğin kesişimi görünümündeki veri örneği için hedef sınıf tahmininde tüm özellikler için koşullu olasılıkların çarpımı hesaplanmalıdır.

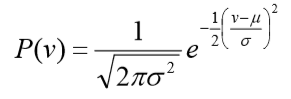
 (3)

Naïve Bayes Sınıflayıcı ile Bayes Teoremi hesaplarında dikkat edilmesi gereken en önemli fark sınıflayıcıların olasılık değerinden ziyade hedef sınıfı bulmaya odaklanmasıdır. Bu yüzden paydada bulunan değer, tüm hedef sınıflara ait olasılık hesaplarında ortak olduğundan ihmal edilebilir. O zaman hedef sınıf bulurken dikkat edilecek formül aşağıda belirtilmiştir.

 (4)

Bu yüzden Bayes Teoreminde bulunan sınıf olasılıkları toplamı 1 olmak zorunda iken Naïve Bayes Sınıflayıcı ile bulunan değerlerin toplamı (ihmal edilen paydadan dolayı) 1 olamaz. Buna göre Bayes Teoremi ile bir durumun olası sonuçlarının olasılıkları bulunurken Naïve Bayes ile normalizasyonsuz olasılıklar üzerinden sınıflandırma yapılabilir.

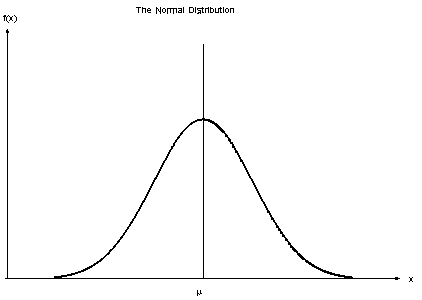
Bayes Teoremi yalnızca kategorik veri özelliklerinde kullanılabilmektedir. Nümerik değerlere sahip özelliklere uygulayabilmek için ilgili özellik uzayında örneklerin Gauss (normal) Dağılımına sahip olduğu varsayılır. Aranan olasılık değeri, özellik vektörünün ortalaması µ ve standart sapması σ değerine bağlı aşağıdaki dağılım formülü ile hesaplanır [6].

 (5)

*a. Bernoulli Naïve Bayes Sınıflandırıcı*

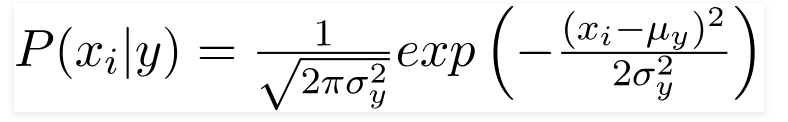
Bu, Multinomial Naive Bayes’e benzer, ancak tahmin ediciler boolean(ikili) değişkenleridir. Sınıf değişkenini tahmin etmek için kullanılan parametreler, örneğin Evet/Hayır, Spam/Spam Değil, 1/0, Var/Yok gibi değerler alır

## b. Gaussian Naïve Bayes Sınıflandırıcı

Tahmin ediciler sürekli bir değer aldıklarında ve ayrık olmadıklarında, bu değerlerin bir gauss dağılımından örneklendiğini varsayılır. Eğer özellikler sürekli değer (continuous value) ise bu değerlerin bir gauss dağılımı veya diğer bir değişle normal dağılımdan örneklendiği varsayılır.

Gauss Dağılımı (Normal Dağılım)

Değerlerin veri kümesinde bulunma şekli değiştiğinden, koşullu olasılık formülü şu şekilde değişir:

 (6)

*c. Multinomial Naïve Bayes Sınıflandırıcı*

Multinomial Naïve Bayesalgoritması, Naive Bayes algoritmasının metin belgeleri için özelleştirilmiş halidir. Naive Bayes belirli kelimelerin olup olmadığını belirlerken, Naive Bayes Multinomial, doküman içerisinde bulunan kelimerin tekrar etme sayılarından frekansını hesaplayarak olasılık kümesi oluşturmaktadır (Adrew McCallum, 1998). Her bir kelimenin tekrar etme sayısı bir başka kelimenin veya kelimelerin tekrar etme sayısına bağlı değildir [7].

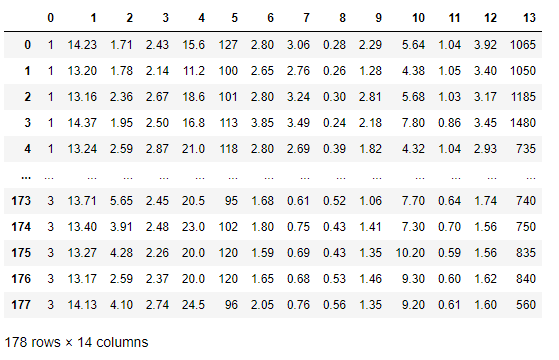
Multinomial Naïve Bayesalgoritmasıbelge sınıflandırma problemi için kullanılır, yani bir belgenin spor, politika, teknoloji vb. kategorisine ait olup olmadığı. Sınıflandırıcının kullandığı özellikler / öngörücüler belgede bulunan kelimelerin sıklığıdır[8].

# VERİ KÜMESİ

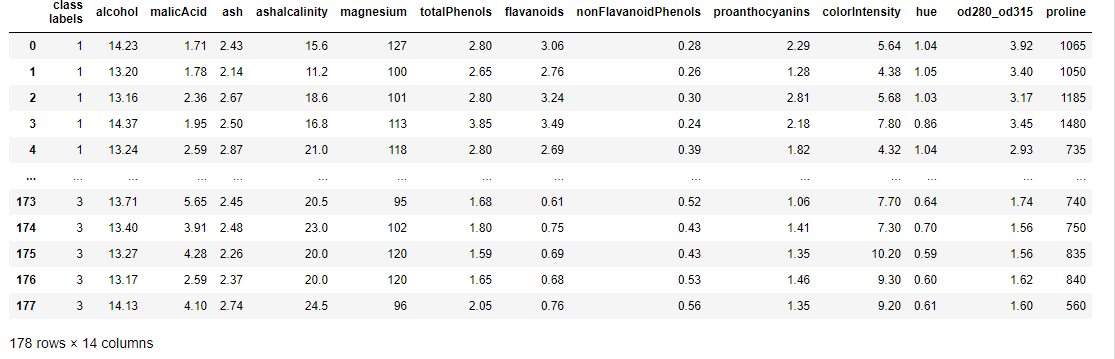
Sınıflandırma amacıyla ‘wine’ veri kümesi kullanılmıştır. Wine veri kümesi 3 farklı sınıf ve 13 farklı değişkene sahiptir. Veri kümesindeki 13 farklı değişkene bakılarak şarapların kalite skalalarına göre sınıflandırılma işlemi gerçekleştirilir[9].

Wine veri kümesi 178 satır ve 14 sütundan oluşmaktadır. Veri kümesinin tanımlanmasının manuel olarak yapılması durumunda 13 farklı değişkenin isimleri yer almaktadır. (Resim 3.1) Tanımlama sonrasında özellikler kod ile tabloya eklenmiştir. (Resim 3.2)

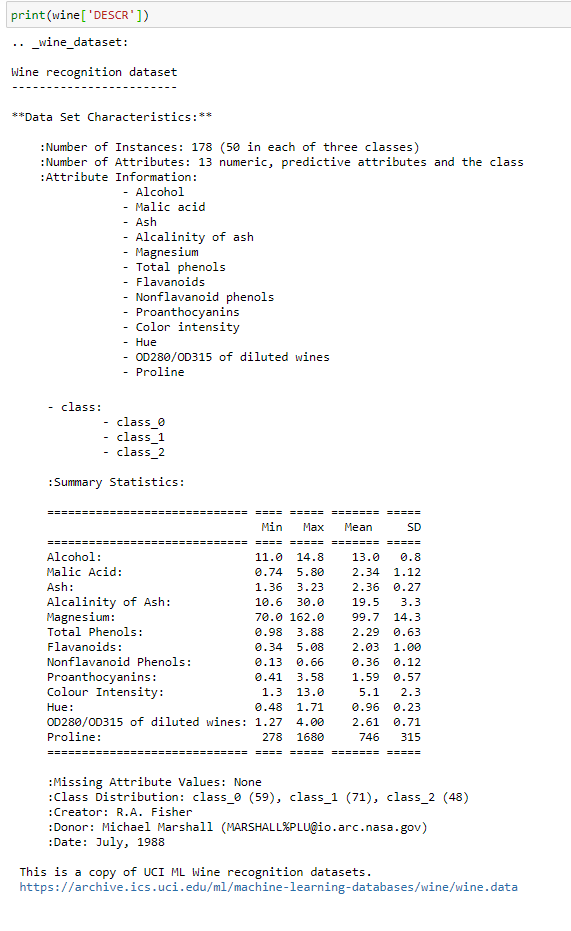
Son olarak Resim 3.3 ‘de veri kümesi hakkında detaylı bilgiye yer verilmiştir.



Resim 3.1



Resim 3.2



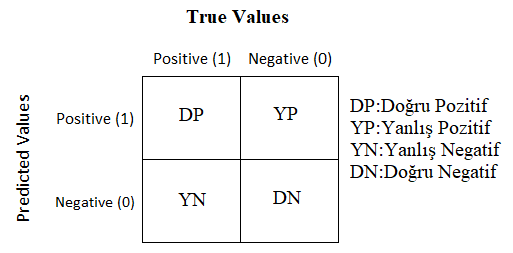
Resim 3.3

# SONUÇLAR

Wine veri kümesi Naïve Bayes ve Convolutional Neural Network ile sınıflandırılmıştır. Sonuçlar başlığı altında farklı eğitim yüzdelerinden edinilen sonuçlara ve farklı döngü değerlerinden elde edilen sonuçlara ‘Bulgular’ alt başlığında, proje içinde kullanılan karşılaştırma ölçütlerine ayrı bir alt başlıkta yer verilmiştir.

A. Karşılaştırma Ölçütleri

Sınıflandırma başarımını ölçmek için öncelikle hata matrisi kullanılmaktadır. Confusion matrix (hata matrisi) sınıf sayısı kadar satır ve sütundan oluşan ve her bir hücrede farklı bir değer tutan bir tablodur (Resim 4.1).



Resim 4.1 İkili sınıflandırmaya ilişkin doğru (1) ve yanlış (0) sınıflandırma sonuç sayılarını gösteren bir hata matrisi.

Hata matrisinden yararlanarak aşağıdaki ölçütler yöntemleri karşılaştırma amacıyla kullanılmaktadır:

* Doğruluk (Accuracy): Doğru olarak sınıflandırılmış girdilerin toplam girdilere oranını veren ölçüttür. Denklem (6) ile hesaplanmaktadır.

 (7)

* Kesinlik (Precision): Doğru olarak sınıflandırılmış pozitif girdilerin toplam pozitif değerlere oranıdır. Denklem (8) ile hesaplanmaktadır.

 (8)

* Geriçağırım (Recall): Doğru olarak sınıflandırılmış pozitif girdilerin gerçek pozitif değerlere oranıdır. Denklem (9) ile hesaplanmaktadır.

 (9)

* F1 ölçütü: Daha iyi bir karşılaştırma sağlamak için kesinlik ve geriçağırımın harmonik ortalaması şeklinde hesaplanan bir ölçüttür (Denklem (10)). (K:Kesinlik, G:Geriçağırım)

 (10)

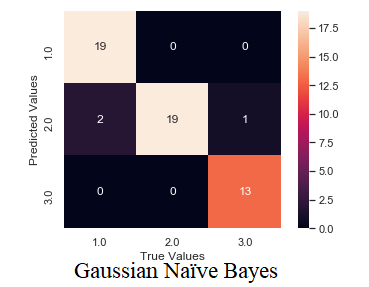
* Support: Veri kümesini sınıflandırırken verilmiş olan sınıfın gerçekleşme sayısıdır.

B. Bulgular

Gaussian Naïve Bayes algoritmasına sınıflandırmak amacıyla verilen test size=0.30 iken elde edilen bulgulara aşağıdaki görsellerde yer verilmiştir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **precision** | **recall** | **f1-score** | **support** |
| **1** | 0.90 | 1.00 | 0.95 | 19 |
| **2** | 1.00 | 0.86 | 0.93 | 22 |
| **3** | 0.93 | 1.00 | 0.96 | 13 |
| **accuracy** |  |  | 0.94 | 54 |
| **macroavg** | 0.94 | 0.95 | 0.95 | 54 |
| **weighted avg** | 0.95 | 0.94 | 0.94 | 54 |

Tablo 1: Test Size = 0.30 iken Gaussian Naïve Bayes ile elde edilen sonuç

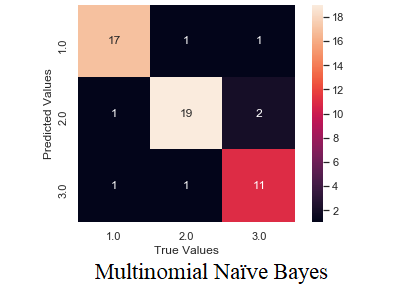


Resim 4.2 Test Size = 0.30 iken Gaussian Naïve Bayes ile elde edilen confusion matrix

Multinomial Naïve Bayes algoritmasına sınıflandırmak amacıyla verilen test size=0.30 iken elde edilen bulgulara aşağıdaki görsellerde yer verilmiştir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **precision** | **recall** | **f1-score** | **support** |
| **1** | 0.89 | 0.89 | 0.89 | 19 |
| **2** | 0.90 | 0.86 | 0.88 | 22 |
| **3** | 0.79 | 0.85 | 0.81 | 13 |
| **accuracy** |  |  | 0.87 | 54 |
| **macroavg** | 0.86 | 0.87 | 0.86 | 54 |
| **weighted avg** | 0.87 | 0.87 | 0.87 | 54 |

Tablo 2: Test Size = 0.30 iken Multinomial Naïve Bayes ile elde edilen sonuç

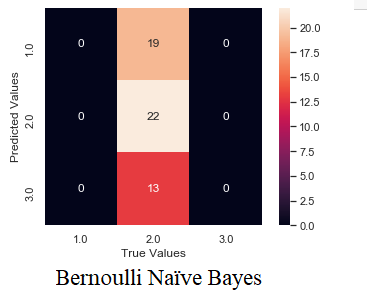


Resim 4.3 Test Size = 0.30 iken Multinomial Naïve Bayes ile elde edilen confusion matrix

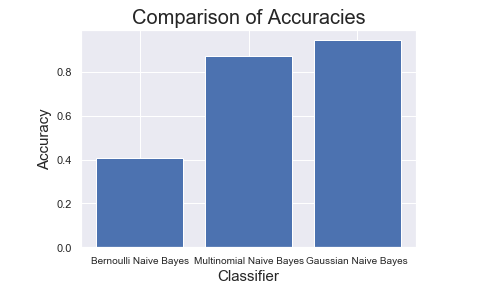
Bernoulli Naïve Bayes algoritmasına sınıflandırmak amacıyla verilen test size=0.30 iken elde edilen bulgulara aşağıdaki görsellerde yer verilmiştir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **precision** | **recall** | **f1-score** | **support** |
| **1** | 0.00 | 0.00 | 0.00 | 19 |
| **2** | 0.41 | 1.00 | 0.58 | 22 |
| **3** | 0.00 | 0.00 | 0.00 | 13 |
| **accuracy** |  |  | 0.41 | 54 |
| **macroavg** | 0.14 | 0.33 | 0.19 | 54 |
| **weighted avg** | 0.17 | 0.41 | 0.24 | 54 |

Tablo 3: Test Size = 0.30 iken Bernoulli Naïve Bayes ile elde edilen sonuç



Resim 4.4 Test Size = 0.30 iken Benoulli Naïve Bayes ile elde edilen confusion matrix



Resim 4.5 Test Size = 0.30 iken accuracy (doğruluk) değerlerinin karşılaştırılması

| Eğitim  Yüzdeleri | Naïve Bayes Türleri ve Eğitim Yüzdeleri Değişikliğine Göre Doğruluk Metriği | | |
| --- | --- | --- | --- |
| Bernoulli Naïve Bayes | Multinomial Naïve Bayes | Gaussian Naïve Bayes |
| **TS70** | 0.40 | 0.87 | 0.94 |
| **TS50** | 0.28 | 0.89 | 0.94 |
| **TS30** | 0.41 | 0.86 | 0.96 |

Tablo 4: Eğitim yüzdeleri değişimlerine ve Naïve Bayes türlerine göre elde edilen sonuç

Convolutional Neural Network algoritmasına sınıflandırmak amacıyla verilen test size=0.30 iken elde edilen bulgulara aşağıdaki görsellerde yer verilmiştir. Eğitim amacıyla kullanılan bilgilere aşağıda yer verilmiştir:

* Batch\_size:Parametre güncellemesinin gerçekleştiği ağa verilen alt örneklerin sayısıdır. Proje de batch\_size 15 olarak belirlenmiştir.
* Loss:keras.losses.categorical\_crossentropy fonksiyonu ile bulunmuştur.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Epoch** | **Optimizer Değişikliğine Göre Doğruluk Metriği** | | | | | | |
| ***Adam*** | ***Adagrad*** | ***Adadelta*** | ***Adamax*** | ***Nadam*** | ***RMSprop*** | ***SGD*** |
| **25** | 0.90 | 0.88 | 0.90 | 0.88 | 0.94 | 0.88 | 0.40 |
| **50** | 0.91 | 0.87 | 0.92 | 0.90 | 0.88 | 0.88 | 0.40 |
| **150** | 0.93 | 0.96 | 0.88 | 0.90 | 0.88 | 0.87 | 0.87 |
| **250** | 0.91 | 0.88 | 0.90 | 0.88 | 0.88 | 0.92 | 0.90 |

Tablo 5: Epoch değerlerinin değişimlerine göre elde edilen sonuç

##### 

Resim 4.6 Test Size = 0.30 ve Epoch=25 iken farklı optimizer türlerinden elde edilen sonuçlar

|  |  |
| --- | --- |
| **Sınıflandırıcı** | **Doğruluk Sonucu** |
| Naïve Bayes | 0.94 |
| Convolutional Neural Network | 0.93 |

Tablo 5: Naïve Bayes ve Convolutional Neural Network algoritmalarından elde edilen ortalama sonuç

KAYNAKLAR

1. Alpaydın E. Yapay Öğrenme. 3. Baskı. İstanbul, Türkiye, Boğaziçi Üniversitesi Yayınevi, 2017.
2. Mitchell T. Machine Learning. New York, USA, McGraw Hill, 1997.
3. <https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad>

CNN algoritması anlatımında kullanılan çoğu resim şu kaynaktan alıntılanmıştır: <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>

1. <https://medium.com/@ekrem.hatipoglu/machine-learning-classification-naive-bayes-part-11-4a10cd3452b4>
2. <https://tr.wikipedia.org/wiki/Naive_Bayes_s%C4%B1n%C4%B1fland%C4%B1r%C4%B1c%C4%B1>
3. Yrd. Doç. Dr. Umut ORHAN ders notları
4. <http://openaccess.maltepe.edu.tr/xmlui/bitstream/handle/20.500.12415/260/10198845.pdf?sequence=1&isAllowed=y>
5. <https://devhunteryz.wordpress.com/2019/12/02/naive-bayes-siniflandirici/>
6. <https://medium.com/datarunner/veri-bilimi-projeleriniz-i%C3%A7in-renkli-veri-setleri-bulabilece%C4%9Finiz-en-iyi-10-site-436383a868b2>
7. Karşılaştırma Ölçütleri Bölümünde yararlanılan kaynak:

<http://oaji.net/articles/2017/1486-1538481206.pdf>

1. Karşılaştırma Ölçütleri Resim 4.1 kaynak:

<https://medium.com/@sengul_krdrl/hata-matrisini-anlamak-7035b7921c0f>

1. Veri kümesi için yararlanılan kaynak:

<https://archive.ics.uci.edu/ml/datasets/wine>