24.01.2022

SVM & Naive Bayes

MUSTAFA KEMAL GÖKÇE 18120205034
MUHAMMED ALİ AL HOUSSINI 18120212006
YUSUF YALÇIN 18120205032

1. Kullanılan Veri Seti

Proje kapsamında kullanılan veri seti 3 farklı verinin birleştirilmesiyle oluşmuştur. Bu 3 veri seti Margin, Shape ve Texture adlı dosyalardan oluşur. Her bir dosya 64 adet feature'a sahiptir. Her bir dosya 1600 örnek(sample)'dan oluşur. Veri seti birleştirildikten sonra 1600 x 193'lük bir matris elde edilir. Bu matrisin ilk satırı bitki isimleri, kalan 192 satır ise bitkilere ait feature'lardır.

```
data1 = pd.read_csv('data_Mar_64.txt', header= None)
data2 = pd.read_csv('data_Sha_64.txt', header= None)
data3 = pd.read_csv('data_Tex_64.txt', header= None)
```

Şekil 1 Veri setinin okunması

Şekil 1'de gösterildiği üzere 3 farklı veri seti pandas kütüphanesi yardımıyla okunmuştur. 3 veri seti birleştirilmeden önce veri setindeki verilerin farklı sırada olmasından dolayı sıralama işlemi uygulanır. Verilerin sıralanması Şekil 2'de gösterilmiştir.

```
data1 = data1.sort_values(by=data1.columns[0]).iloc[:,:]
data2 = data2.sort_values(by=data2.columns[0]).iloc[:,1:]
data3 = data3.sort_values(by=data3.columns[0]).iloc[:,1:]

data2 = data2.reset_index(drop = True)

data3 = data3.reset_index(drop = True)
```

Şekil 2 Verilerin sıralanması

Sıralanan veri setleri birleştirilerek tek bir veri setine dönüştürülür. Bu dönüştürme işlemi Şekil 3'te gösterilmiştir.

```
data = pd.concat([data1, data2, data3], axis=1, ignore_index=True)
```

Şekil 3 Verilerin birleştirilmesi

Modelin eğitilme aşamasından önce X ve y değerleri data veri setinden atanılır. Atama işlemi Şekil 4'te gösterilmiştir.

```
X = data.iloc[:,1:].values
y = data.iloc[:,:1].values
```

Şekil 4 Atama işlemi

Bu atama işleminden sonra alınan y değerleri bitki isimleri yani stringler olacağı için bu değerler LabelEncoder yardımıyla numerik değerlere dönüştürülür. Dönüştürme işlemi Şekil 5'te gösterilmiştir.

```
from sklearn.preprocessing import LabelEncoder

y = LabelEncoder().fit_transform(y)
```

Şekil 5 Numerik veriye çevirme işlemi

2. Kullanılan Modeller

Proje kapsamında 2 farklı model kullanılması gerekmektedir. Modellerden ilki SVM, diğeri ise GaussianNB'dir. Bu 2 model 2 farklı parametreyle çalıştırılmıştır. Modellerin eklenmesi Şekil 6'da gösterilmiştir.

```
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
gaussianModel = GaussianNB(var_smoothing= 1e-9)
svmModel = SVC(kernel= 'rbf')
```

Şekil 6 Modellerin eklenmesi

Modelin accuracy, precision, recall ve f1-score metriklerini hesaplamak için 2 farklı fonksiyon yazılmıştır. Bu fonksiyonlar cross_val_score fonksiyonu tarafından çağırılır ve accuracy, precision, recall ve f1-score değerlerini ekrana yazdırırlar. Fonksiyonlardan my_scorer fonksiyonu getScores fonksiyonundaki değerleri ekrana yazdırmaya ve doğruluk değerlerini kaydetmeye yarar. Bu fonksiyonlar Şekil 7'de gösterilmiştir.

```
def getScores(estimator, x, y):
    yPred = estimator.predict(x)
    return (accuracy_score(y, yPred),
            precision_score(y, yPred,average='macro',zero_division=0),
            recall_score(y, yPred, average='macro',zero_division=0),
            f1_score(y, yPred, average='macro',zero_division=0),)

def my_scorer(estimator, x, y):
    acc, pre, rcl, f1 = getScores(estimator, x, y)
    print ("acc:",acc,"pre:", pre,"recl:", rcl, "f1:", f1)
    return acc
```

Şekil 7 Doğruluk hesaplama fonksiyonları

2.1 SVM

SVM kapsamında 2 farklı kernel yöntemiyle model çağırılmıştır. Bu kernel yöntemlerinden ilki rbf kerneldir. Diğer kullanılan kernel ise poly kerneldir. Bu 2 farklı kernel 2 farklı k-fold değeriyle modele uygulanır.

2.1.1 K-fold 3

Seçilen 3 k-fold değeri ile modelin eğitimi sağlanmış ve sonuçlar ekrana yazılmıştır. K-fold değerinin 3 olarak seçilmesi Şekil 8'de gösterilmiştir.

```
cv = KFold(n_splits=3, shuffle=True, random_state=1)
svmScores = cross_val_score(estimator=svmModel,X= X,y= y, scoring=my_scorer, cv=cv)
print(svmScores.argmax(), ". score svm modelinde en buyuk accuracye sahip", 'Accuray değeri:',svmScores[svmScores.argmax()])
```

Şekil 8 K-fold değerinin 3 olarak kullanılması

İlk olarak rbf kernel ile modelin eğitimi sağlanmıştır. Rbf kernel kullanılması Şekil 9'da gösterilmiştir.

```
svmModel = SVC(kernel= 'rbf')
```

Şekil 9 Rbf kernel implementasyonu

K fold değeri 3 olarak seçildikten sonra rbf kernel üzerinde modelin verdiği doğruluk değerleri Şekil 10'da gösterilmiştir.

```
acc: 0.900749063670412 pre: 0.9136017316017316 recl: 0.9048730158730158 f1: 0.8956936938190807
acc: 0.9080675422138836 pre: 0.9164261664261664 recl: 0.9133357383357383 f1: 0.9050268573441479
acc: 0.9099437148217636 pre: 0.9149563492063492 recl: 0.92209126984127 f1: 0.9083884840649545
2 . score svm modelinde en buyuk accuracye sahip Accuray değeri: 0.9099437148217636
```

Şekil 10 K-fold 3 değeri ve rbf kernel ile doğruluk değerleri

Rbf kernel ile modelin eğitimi tamamlandıktan sonra poly kernel ile modelin eğitimine geçilmiştir. Poly kernel kullanılması Şekil 11'de gösterilmiştir.

```
svmModel = SVC(kernel= 'poly')
```

Şekil 11 Poly kernel implementasyonu

K fold değeri 3 olarak seçildikten sonra poly kernel üzerinde modelin verdiği doğruluk değerleri Şekil 12'da gösterilmiştir.

```
acc: 0.9044943820224719 pre: 0.9102023809523808 recl: 0.9117063492063493 f1: 0.8986573753044342
acc: 0.8761726078799249 pre: 0.9049974996944693 recl: 0.8928290844957513 f1: 0.8793124532429346
acc: 0.900562851782364 pre: 0.9125952380952381 recl: 0.9140793650793653 f1: 0.9016037671478848
0 . score svm modelinde en buyuk accuracye sahip Accuray değeri: 0.9044943820224719
```

Şekil 12 K-fold 3 değeri ve poly kernel ile doğruluk değerleri

2.1.2 K-fold 5

Seçilen 5 k-fold değeri ile modelin eğitimi sağlanmış ve sonuçlar ekrana yazılmıştır. K-fold değerinin 5 olarak seçilmesi Şekil 13'de gösterilmiştir.

```
cv = KFold(n_splits=5, shuffle=True, random_state=1)
svmScores = cross_val_score(estimator=svmModel,X= X,y= y, scoring=my_scorer, cv=cv)
print(svmScores.argmax(), ". score svm modelinde en buyuk accuracye sahip", 'Accuray değeri:',svmScores[svmScores.argmax()])
```

Şekil 13 K-fold değerinin 5 olarak kullanılması

K fold değeri 5 olarak seçildikten sonra rbf kernel üzerinde modelin verdiği doğruluk değerleri Şekil 14'de gösterilmiştir.

```
acc: 0.921875 pre: 0.9139761904761906 recl: 0.9233095238095239 f1: 0.9078861693861694
acc: 0.93125 pre: 0.9306666666666666 recl: 0.9209999999999999 f1: 0.9147099567099568
acc: 0.9375 pre: 0.9446180555555556 recl: 0.9420138888888889 f1: 0.9357285654160655
acc: 0.91875 pre: 0.919533527696793 recl: 0.9220238095238094 f1: 0.9088447833345792
acc: 0.928125 pre: 0.938023088023088 recl: 0.9466329966329966 f1: 0.9302979287827773
2 . score svm modelinde en buyuk accuracye sahip Accuray değeri: 0.9375
```

Şekil 14 K-fold 5 değeri ve rbf kernel ile doğruluk değerleri

K fold değeri 5 olarak seçildikten sonra poly kernel üzerinde modelin verdiği doğruluk değerleri Şekil 15'de gösterilmiştir.

```
acc: 0.91875 pre: 0.9109523809523811 recl: 0.9236190476190477 f1: 0.9013707403707403
acc: 0.934375 pre: 0.9529100529100529 recl: 0.9422558922558922 f1: 0.9377213695395513
acc: 0.9375 pre: 0.9506944444444444 recl: 0.9407986111111111 f1: 0.9373369107744107
acc: 0.91875 pre: 0.9296296296296296 recl: 0.9292929292929293 f1: 0.9149831649831651
acc: 0.928125 pre: 0.936026936026936 recl: 0.9466329966329968 f1: 0.9301645604675909
2 . score svm modelinde en buyuk accuracye sahip Accuray değeri: 0.9375
```

Şekil 15 K-fold 5 değeri ve poly kernel ile doğruluk değerleri

2.2 GaussianNB

GaussianNB kapsamında 2 farklı var_smoothing değeriyle model çağırılmıştır. Bu değerlerden ilki 2e-9'dur. Diğer kullanılan değer ise 1e-4'dur. `Bu 2 farklı değer modele uygulanır.`

2.2.1 K-fold 3

Seçilen 3 k-fold değeri ile modelin eğitimi sağlanmış ve sonuçlar ekrana yazılmıştır. K-fold değerinin 3 olarak seçilmesi Şekil 16'da gösterilmiştir.

```
cv = KFold(n_splits=3, shuffle=True, random_state=1)
gaussianScores = cross_val_score(estimator=gaussianModel,X= X,y= y, scoring=my_scorer, cv=cv)
print(gaussianScores.argmax(), ". score gauss modelinde en buyuk accuracye sahip",'Accuray değeri:', gaussianScores[gaussianScore
```

Şekil 16

K fold değeri 3 olarak seçildikten sonra var_smoothing = 1e-9 üzerinde modelin verdiği doğruluk değerleri Şekil 17'de gösterilmiştir.

```
acc: 0.651685393258427 pre: 0.7661539644565961 recl: 0.6722341269841269 f1: 0.6575420616886292
acc: 0.6529080675422139 pre: 0.7403477142781955 recl: 0.6649831649831651 f1: 0.6510531944155658
acc: 0.6454033771106942 pre: 0.7797967032967031 recl: 0.6657261904761903 f1: 0.6513171653108144
1 . score gauss modelinde en buyuk accuracye sahip Accuray değeri: 0.6529080675422139
```

Şekil 17

K fold değeri 3 olarak seçildikten sonra var_smoothing = 1e-4 üzerinde modelin verdiği doğruluk değerleri Şekil 18'de gösterilmiştir.

```
acc: 0.9382022471910112 pre: 0.9401190476190476 recl: 0.9409007936507936 f1: 0.9348287745965765
acc: 0.9362101313320825 pre: 0.9408114332356757 recl: 0.9328924162257495 f1: 0.9305477215553207
acc: 0.9455909943714822 pre: 0.9541691919191919 recl: 0.9512936507936509 f1: 0.9460975152298681
2 . score gauss modelinde en buyuk accuracye sahip Accuray değeri: 0.9455909943714822
```

Şekil 18

2.2.2 K-fold 5

Seçilen 5 k-fold değeri ile modelin eğitimi sağlanmış ve sonuçlar ekrana yazılmıştır. K-fold değerinin 5 olarak seçilmesi Şekil 19'da gösterilmiştir.

```
cv = KFold(n_splits=5, shuffle=True, random_state=1)
gaussianScores = cross_val_score(estimator=gaussianModel,X= X,y= y, scoring=my_scorer, cv=cv)
print(gaussianScores.argmax(), ". score gauss modelinde en buyuk accuracye sahip",'Accuray değeri:'
```

Şekil 19

K fold değeri 5 olarak seçildikten sonra var_smoothing = 1e-9 üzerinde modelin verdiği doğruluk değerleri Şekil 20'de gösterilmiştir.

```
acc: 0.6875 pre: 0.7207467532467533 recl: 0.6927619047619047 f1: 0.6498077608665844
acc: 0.753125 pre: 0.811976911976912 recl: 0.7892736892736893 f1: 0.7562828865859169
acc: 0.728125 pre: 0.7609572400388728 recl: 0.714698736637512 f1: 0.6994287041279523
acc: 0.728125 pre: 0.7544716787141029 recl: 0.731806156806157 f1: 0.7055528085831116
acc: 0.65625 pre: 0.7383257174923841 recl: 0.6967772967772968 f1: 0.657817751668019
1 . score gauss modelinde en buyuk accuracye sahip Accuray değeri: 0.753125
```

Şekil 20

K fold değeri 5 olarak seçildikten sonra var_smoothing = 1e-4 üzerinde modelin verdiği doğruluk değerleri Şekil 21'de gösterilmiştir.

```
acc: 0.934375 pre: 0.9374458874458875 recl: 0.9433862433862434 f1: 0.9348033560154771
acc: 0.975 pre: 0.9802789802789803 recl: 0.974915824915825 f1: 0.9720699054032388
acc: 0.959375 pre: 0.9640893470790378 recl: 0.9480117820324006 f1: 0.9491125140609677
acc: 0.971875 pre: 0.965993265993266 recl: 0.9654882154882154 f1: 0.9620570787237455
acc: 0.95 pre: 0.955940355940356 recl: 0.9656565656565655 f1: 0.9519210867695715
1 . score gauss modelinde en buyuk accuracye sahip Accuray değeri: 0.975
```

Şekil 21

3. Z score Normalizasyonu

Proje kapsamında en iyi modeller K_fold 5, kernel = 'rbf' seçilince SVM'de en iyi sonuçları vermiştir. K_fold 5, var_smoothing = 1e-4 seçilince Gaussian Naive Bayes'de en iyi sonuçları vermiştir.

Veriyi z score nomalizasyonundan geçirdikten sonra SVM modelin verdiği doğruluk değerleri Şekil 21'de gösterilmiştir.

```
acc: 0.9875 pre: 0.9865319865319866 recl: 0.9898027898027898 f1: 0.9855421794815735
acc: 0.9875 pre: 0.9914141414141415 recl: 0.9861952861952861 f1: 0.9866602533269201
acc: 0.990625 pre: 0.9935763888888888 recl: 0.9904513888888888 f1: 0.9905002405002405
acc: 0.9875 pre: 0.9838435374149659 recl: 0.9894557823129252 f1: 0.9836086815678654
acc: 0.98125 pre: 0.9791245791245792 recl: 0.9890572390572391 f1: 0.9809465506435203
2 . score svm modelinde en buyuk accuracye sahip Accuray değeri: 0.990625
```

Şekil 22

Veriyi z score nomalizasyonundan geçirdikten sonra GaussianNB modelin verdiği doğruluk değerleri Şekil 22'de gösterilmiştir.

```
acc: 0.89375 pre: 0.8903333333333333 recl: 0.9023333333333333 f1: 0.8763215673215673
acc: 0.921875 pre: 0.938047138047138 recl: 0.9365319865319865 f1: 0.9240092118879998
acc: 0.9125 pre: 0.9083498677248677 recl: 0.9093501984126985 f1: 0.8953477367539868
acc: 0.925 pre: 0.9297138047138047 recl: 0.9305435305435306 f1: 0.9184240563028442
acc: 0.909375 pre: 0.915921115921116 recl: 0.9201058201058202 f1: 0.9009304388092267
3 . score gauss modelinde en buyuk accuracye sahip Accuray değeri: 0.925
```

Şekil 23

Şekil 22 ve Şekil 23'te görüldüğü üzere en iyi doğruluk değerini SVM modeli vermiştir.