24.01.2022

Örüntü Tanıma Proje Raporu

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:]
```

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: 0.9063670411985019 pre: 0.9216150793650795 recl: 0.916861111111111 f1: 0.9034983741748448
acc: 0.8968105065666041 pre: 0.9021468253968252 recl: 0.9045277777777778 f1: 0.8899576175458529
acc: 0.9043151969981238 pre: 0.9161825396825397 recl: 0.9126984126984127 f1: 0.9009300062682416
0 . score svm modelinde en buyuk accuracye sahip Accuray değeri: 0.9063670411985019
```

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: 0.8820224719101124 pre: 0.9138809523809525 recl: 0.8969325396825397 f1: 0.8855947206388383
acc: 0.8911819887429644 pre: 0.9016430375180375 recl: 0.8985317460317459 f1: 0.8848145873734109
acc: 0.9043151969981238 pre: 0.9188412698412699 recl: 0.9143690476190477 f1: 0.9035376143464378
2 . score svm modelinde en buyuk accuracye sahip Accuray değeri: 0.9043151969981238
```

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: 0.90625 pre: 0.908970658970659 recl: 0.9272246272246272 f1: 0.9052961068112584
acc: 0.940625 pre: 0.9566859066859066 recl: 0.944011544011544 f1: 0.9412762882459853
acc: 0.95 pre: 0.9510309278350515 recl: 0.946097201767305 f1: 0.9411582689933206
acc: 0.925 pre: 0.9114478114478115 recl: 0.9134559884559884 f1: 0.8990524626888263
acc: 0.903125 pre: 0.8987857142857142 recl: 0.9066666666666667 f1: 0.8904758574758574
2 . score svm modelinde en buyuk accuracye sahip Accuray değeri: 0.95
```

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: 0.890625 pre: 0.9053872053872053 recl: 0.916835016835017 f1: 0.889131575495212
acc: 0.940625 pre: 0.9585858585858587 recl: 0.9449494949494951 f1: 0.9439146151267364
acc: 0.94375 pre: 0.946809032891507 recl: 0.9432744231713305 f1: 0.9336039676245863
acc: 0.9 pre: 0.8989658489658491 recl: 0.8899591149591151 f1: 0.8806963519084731
acc: 0.896875 pre: 0.9040952380952382 recl: 0.905 f1: 0.8864069819069819
2 . score svm modelinde en buyuk accuracye sahip Accuray değeri: 0.94375
```

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: 0.6367041198501873 pre: 0.7560924777836542 recl: 0.6604563492063492 f1: 0.6362182291616939
acc: 0.7110694183864915 pre: 0.81192155908065 recl: 0.7159090909090909 f1: 0.7082045128164215
acc: 0.6604127579737336 pre: 0.7785489365706756 recl: 0.6852261904761905 f1: 0.6737004067483942
1 . score gauss modelinde en buyuk accuracye sahip Accuray değeri: 0.7110694183864915
```

Şekil 17

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

```
acc: 0.9325842696629213 pre: 0.9417301587301589 recl: 0.9400952380952381 f1:
0.930494174126527
acc: 0.9549718574108818 pre: 0.9546626984126984 recl: 0.9464365079365079 f1:
0.9437760246289657
acc: 0.9530956848030019 pre: 0.9593214285714287 recl: 0.9613055555555556 f1:
0.9546675742558096
1 . score gauss modelinde en buyuk accuracye sahip Accuray değeri: 0.9549718574108818
```

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: 0.66875 pre: 0.7211233766233767 recl: 0.7050952380952381 f1: 0.6567215007215007
acc: 0.709375 pre: 0.7582222222222222 recl: 0.7248333333333333 f1: 0.6890852480852482
acc: 0.753125 pre: 0.7907353417557501 recl: 0.7448979591836735 f1: 0.7228268421482549
acc: 0.790625 pre: 0.7955961538461537 recl: 0.7926071428571428 f1: 0.7706290376290376
acc: 0.653125 pre: 0.7835738335738336 recl: 0.690981240981241 f1: 0.6821861079436836
3 . score gauss modelinde en buyuk accuracye sahip Accuray değeri: 0.790625
```

Şekil 20

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

```
acc: 0.928125 pre: 0.9371813371813372 recl: 0.9512265512265513 f1: 0.9322065140246958
acc: 0.965625 pre: 0.9703703703703703 recl: 0.9648148148148148 f1: 0.9618085618085618
acc: 0.95625 pre: 0.94589407191448 recl: 0.9438532555879495 f1: 0.9387998056365403
acc: 0.953125 pre: 0.9444925444925444 recl: 0.9371813371813371 f1: 0.9348565015231682
acc: 0.9625 pre: 0.9592380952380952 recl: 0.9630000000000001 f1: 0.9544356754356754
1 . score gauss modelinde en buyuk accuracye sahip Accuray değeri: 0.965625
```

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: 0.990625 pre: 0.9929292929292929 recl: 0.9926647426647426 f1: 0.9916989856383797
acc: 0.99375 pre: 0.9949494949494949 recl: 0.9924242424242424 f1: 0.9922515195242468
acc: 0.996875 pre: 0.9973958333333334 recl: 0.9979166666666667 f1: 0.9973544973544973
acc: 0.984375 pre: 0.9859572400388725 recl: 0.986904761904762 f1: 0.983471630410406
acc: 0.990625 pre: 0.993939393939394 recl: 0.9915824915824915 f1: 0.9914294459749006
2 . score svm modelinde en buyuk accuracye sahip Accuray değeri: 0.996875
```

Şekil 22

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

```
acc: 0.903125 pre: 0.9215000000000001 recl: 0.9171904761904761 f1: 0.9046637806637807
acc: 0.93125 pre: 0.9441428571428571 recl: 0.9268333333333334 f1: 0.925463924963925
acc: 0.925 pre: 0.9222222222222222 recl: 0.9194444444444444 f1: 0.9046679593554593
acc: 0.915625 pre: 0.9254689754689754 recl: 0.9162097162097161 f1: 0.9114760099608583
acc: 0.9125 pre: 0.927873977873978 recl: 0.9229677729677729 f1: 0.9144903917631192
1 . score gauss modelinde en buyuk accuracye sahip Accuray değeri: 0.93125
```

Şekil 23

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