MULTI-CLASS CLASSIFICATION OF DRY BEANS USING CV AND MACHINE LEARNING TECHNIQUES

Analysis done by Dheeraj Dhillon

Dataset provided by Murat Koklu, Ilker Ali Ozkan

The primary objective of this study is to provide a method for obtaining uniform seed varieties from crop production, which is in the form of population, so the seeds are not certified as a sole variety. Thus, a computer vision system was developed to distinguish seven different registered varieties of dry beans with similar features in order to obtain uniform seed classification. For the classification model, images of 13,611 grains of 7 different registered dry beans were taken with a high-resolution camera. Bean images obtained by computer vision system (CVS) were subjected to segmentation and feature extraction stages, and a total of 16 features; 12 dimension and 4 shape forms, were obtained from the grains.

To obtain the given dataset, following steps were followed:

1. **Image Acquisition** using a computer vision system.

2. **Processing and segmentation** is done to prepare images prior to feature extraction. The image processing stage consists of removing the dry bean shadows, eliminating background noise and separating each dry bean from the others.

3. **Feature extraction** After the segmentation step, images containing separated dry beans were obtained. A number of features describing each dry bean were extracted. The dry beans do not have a distinctive color feature. Dimensional and shape features were determined by feature analysis. Geometry related features were also obtained. In total, 12 dimensional and 4 shape features were obtained for each dry bean. Values found in all features are in pixel count. The dimensional & shape features in the classification of dry bean seeds are as follows:

a) **Area** (A): The area of a bean zone and the number of pixels within its boundaries.

b) **Perimeter** (P): Bean circumference is defined as the length of its border.

c) **Major axis length**(L): The distance between the ends of the longest line that can be drawn from a bean.

d) **Minor axis length** (l): The longest line that can be drawn from the bean while standing perpendicular to the main axis.

e) **Aspect ratio** (K): Defines the relationship between L and l. K=L/l

f) **Eccentricity** (Ec): Eccentricity of the ellipse having the same moments as the region.

g) **Convex area** (C): Number of pixels in the smallest convex polygon that can contain the area of a bean seed.

h) **Equivalent diameter** (Ed): The diameter of a circle having the same area as a bean seed area. d= sqrt(4A/π)

i) **Extent** (Ex): The ratio of the pixels in the bounding box to the bean area.

j) **Solidity** (S): Also known as convexity. The ratio of the pixels in the convex shell to those found in beans.

k) **Roundness** (R): Calculated with the following formula: R = 4piA/P^2.

l) **Compactness** (CO): Measures the roundness of an object: CO = Ed/L

m) **ShapeFactor1** (SF1) = L/A

n) **ShapeFactor2** (SF 2) = l/A

o) **ShapeFactor3** (SF3) = A/(pi*(L/2)^2)

p) **ShapeFactor4** (SF4) = A/(pi*(L/2)*(l/2))

Seven different types of dry beans were used in this research, taking into account the features such as form, shape, type and structure. They are called Seker, Barbunya, Bombay, Cali, Dermosan, Horoz and Sira. The general features of the specified dry beans are as follows.

1. **Cali:** It is white in color, its seeds are slightly plump and slightly larger than dry beans and in shape of kidney.

2. **Horoz:** Dry beans of this type are long, cylindrical, white in color and generally medium in size.

3. **Dermason:** This type of dry beans, which are fuller flat, is white in color and one end is round and the other ends are round.

4. **Seker:** Large seeds, white in color, physical shape is round.

5. **Bombay:** It is white in color, its seeds are very big and its physical structure is oval and bulging.

6. **Barbunya:** Beige-colored background with red stripes or variegated, speckled color, its seeds are large, physical shape is oval close to the round.

7. **Sira:** Its seeds are small, white in color, physical structure is flat, one end is flat, and the other end is round.

IMPORTING LIBRARIES

Firstly, I imported all the required libraries required in the dataset analysis.

READING THE DATASET

After importing libraries, I read the data from the excel file. I also applied standard scaler on the data and stored it in as separate dataframe. I saved the dataset in both numpy array format and pandas dataframe format. I then split the dataset for model training and testing with a test split of 0.3 for both standard scaled and non-standard scaled data.

FEATURE IMPORTANCE & SELECTION

I started my analysis of the dataset by calculating the feature importances of the sixteen features using the following methods.

1. Decision Trees
2. Random Forest
3. XGBoost

- We started off by constructing a decision tree for the dataset using Gini criterion. We constructed the tree in two ways. One by limiting the depth of the tree to six and in the second one, we had no limitation of depth.
- The decision trees have their root node with the feature MajorAxisLength clearly signifying its high or maybe the highest importance in identification of class of the sample.
- Even in these two models, we observed vast difference in the importance factors of the various features. Not limiting the depth of the decision tree may lead to overfitting of the training data while limiting it to small height may lead to high bias in the data. Although, in both models, the features:
    - Eccentricity, AspectRation, ShapeFactor2, Area, EquivDiameter, ConvexArea, Extent, Solidity

  had negligible importance whereas compactness which the lite tree showed had max Importance, didn't show much importance in the max tree classifier.

- To reach more conclusive result for the feature importances, we used RandomForest classifier and then the XGBoost classifier. Results from these also were not in sync with each other. So, I did hit & trial while dropping columns to find out which model gave the best accuracy. For the same model, I also calculated the drop column importance for

each of column. This means that if you drop that certain column from the training and testing data, the accuracy will decrease depending on how large the drop col feature importance for that particular feature is.

- Though, there is some confusion in the part where we see that ConvexArea has the least dropCol importance (-ve) even though it dominated the feature_importance when calculated using the sklearn library.

DATA VISUALIZATION

For data visualization, I used a few feature transformation techniques to reduce the feature dimensions from sixteen to either three or two.
I used the following methods for this purpose:

1. Principal components analysis

this performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. In practice, the covariance matrix of the data is constructed and the eigenvectors on this matrix are computed. The eigenvectors that correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data.
i. dimensions to two
ii. reducing dimensions to three
The variances retained by the first three principal components are:

Principal component 1: 55.466439 %
Principal component 2: 26.430973 %
Principal component 3: 8.006564 %

So if we just use the first two principal components we would retain a variance of 81.9 %
And if we retain first three components the variance retained would become 89.9 %
I plotted the dataset for both cases, first one in a 2-D plot and second one in a 3-D.
On observing the plots, we clearly observe clusters of data points forming thus ensuring us that our model may work just fine with less no. of features if not better.

2. Autoencoder

I built an autoencoder which consists of an encoder and a decoder. This reduced the dimensionality from sixteen to two using the following framework.
I created four layers in both the encoder and the decoder.
The neural network structure of the autoencoder used is:

```
Autoencoder(
  (e1): Linear(in_features=16, out_features=12, bias=True)
```

```
  (e2): Linear(in_features=12, out_features=8, bias=True)
  (e3): Linear(in_features=8, out_features=4, bias=True)
  (e4): Linear(in_features=4, out_features=2, bias=True)
  (d1): Linear(in_features=2, out_features=4, bias=True)
  (d2): Linear(in_features=4, out_features=8, bias=True)
  (d3): Linear(in_features=8, out_features=12, bias=True)
  (d4): Linear(in_features=12, out_features=16, bias=True)
)
```

Here the e1 to e4 layers are the encoder layers and d1 to d4 layers make up the decoder. After training the model which is done by doing backpropagation on the error between the output of d4 layer and the input at e1. I used ReLU activation function after each step.

3. t-SNE

T-distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear dimensionality reduction technique useful for visualization of high-dimensional datasets.
I used this method to reduce the dimensionality to two.

Using the above three mentioned methods, I created four datasets. One dataset having 3 features while rest three having two features. I then split these datasets with val/test fraction of 0.3 and then trained SVM Classifier with radial kernel with same hyperparameters and then compared the results which are:

method	no. of features retained	Classifier used	accuracy
PCA	2	SVC with RBF kernel	87.37 %
PCA	3	SVC with RBF kernel	88.86 %
Autoencoder	2	SVC with RBF kernel	87.61 %
t-SNE	2	SVC with RBF kernel	90.45 %

PLOTTING DECISION BOUNDARIES

After reducing the no. of features to two, we can now create decision boundaries on our two-feature models to predict the class of the sample data. I have done this using the

K-Nearest Neighbours classifier. For each of the three datasets, I ran a for loop for K from one to forty, calculating the mean error for each iteration and then choosing that value of K which gives minimum error. Using that value I have plotted the decision boundaries for the datasets.

The results of accuracy on the KNN classifiers are:

method used for feature transformation	classifier used on reduced dataset	optimal value of K for max accuracy	accuracy of classification model
PCA	KNN	20	87.44
Autoencoder	KNN	15	86.75
t-SNE	KNN	7	-

DEVELOPING CLASSIFICATION MODEL

Feature transformation or selection maybe favourable when we want to visualize our data but it is not necessary that the dimensionally reduced data may give the best classification model for our multi-class dataset. So, I have created several classifiers both on reduced and non-reduced datasets to find out the best classifier option. Reduced dataset might be obtained by both transforming or selection.

classifier used	no. of features in dataset	approach to reduced dataset	hyperparameters	accuracy	macro F1 %
SVM-linear kernel	16	-	c = 1	92.78	93.98
SVM-cubic polynomial kernel	16	-	degree = 3 c = 1	91.04	92.37
SVM-rbf kernel	16	-	gamma = 1 c = 1 shape = ovo	92.70	93.50
SVM-rbf kernel	13	dropping Eccentricity,	gamma = 1 c = 1 shape = ovo	95.10	95.99

		AspectRation, EquivDiameter			
Feed forward NN	16	-	Layers : 16->12->3->7 activation: ReLU epochs : 100	92.70	93.83
XGBClassifier	16	-	-	92.90	94.12
SVM-rbf kernel	5	Five features are: MajorAxis Length, MinorAxis Length, Perimeter, Compactness,SF1	gamma = 1 c = 1 shape = ovo	91.85	92.89

Thus our best classifier is obtained by using SVM-rbf kernel after dropping Eccentricity, AspectRation and EquivDiameter columns. Decision function shape of this SVM was ‘ovo’ and hyperparameters gamma and c were both equal to one. Validation accuracy obtained for 70-30 split of the dataset was 95.10 %.

KEY OBSERVATIONS

- In all of the classification models that we have built, the prediction for the class BOMBAY had accuracy of 100 % for most of the time. Also in the plots obtained after feature transformation for PCA, Autoencoder, t-SNE all of them showed the BOMBAY class cluster far away from the other classes.
- On taking a look at all the confusion matrices obtained for different classification methods, we see non-trivial overlap between the following pair of classes:
    - DERMASON & SIRA
    - BARBUNYA & CALI

  The following pair of classes also show overlap, but this overlap is less compared to those observed in above classes. These pairs are:
    - SEKER & SIRA
    - SEKER & DERMASON
    - BARBUNYA & SIRA

Actually all these difference are quite apparent when we zoom in the pca2 plot.

- The SVM Model with the three dropped features Eccentricity, AspectRation, EquivDiameter has the best value with a 95.10% of accuracy. In addition, the SVM classification model obtained has the best values for all calculated performance metrics. As a measure of how well the classifier performs, it is significant that the F1-Score value is also high in all the classification models. In addition, it is seen that the Precision value, which is actually positive in all classifiers, determines the ratio of the number of the positive classifieds to the all positives, is proportional to the accuracy.
- When the confusion matrix of the SVM model with the best classification value given in is examined, accuracy rates of Barbunya, Bombay, Cali, Dermason, Horoz, Seker and Sira were 96.26%, 100.00%, 96.81%, 93.99%, 97.57%, 96.41% and 90.92%, respectively. As we already said before, Bombay variety can be fully classified with 100% accuracy. Sira variety has the lowest classification performance of all kinds. In addition, it is seen that the performance of Dermason and Sira varieties is low in terms of differentiation in confusion matrix.

- The DT model divides all the bean varieties into two groups according to the Major Axis Length feature. In addition, the compactness feature is used in the decision tree model to differentiate the Sira and Seker varieties from each other, Horoz and Cali varieties from Barbunya and Bombay varieties. 91.85% accuracy can be achieved by using only the features of the Major Axis Length, Minor Axis length, Perimeter, Compactness and Shape factor 1 from the sixteen features available in the DT model. This gives better results than all the models in which dimensionality reduction is performed.
- Though the Decision Tree classifier itself was not giving much accuracy on models built upon them, they certainly gave us ideas on the importance of certain features which became apparent when we obtained our best classification model with accuracy of 95.10 %.

- Regarding the classification of dry bean seeds, dimension and shape features of bean varieties have no external discriminatory features, which causes this classification process to be complex. Classification of bean seed varieties is important in terms of assuring seed uniformity and quality.
- This shows that the data distribution is regular. High success rates in all metrics show that the models are successful in the classification. It is seen that all classification models have the lowest sorting performance of the Sira variety due to the low distinguishing of the Sira bean variety with the Dermason variety. The fact that the flatness and roundness features of the Dermason and Sira varieties are similar is effective in this result.
  In the study, due to the high number of data in the DT model, a tree was created, which did not include the features of all data. By changing the parameters, there is a high degree of harmony in the different types of tree that use more features and the interpretation of the trees becomes difficult.

- The results show that the proposed classifier based on CVS can be successfully used to automatically classify various types of dry beans in this study. In addition, this developed model structure can also be used for the types of dry beans of different regions. The model can be further improved by the hybrid use of machine learning methods, deep learning and new algorithms.

- In the study, the variables related to the shape and size characteristics of the bean cultivars were taken from two dimensional images. The third dimension of beans was not included in machine learning. This dimension is the suture axis of the bean. If the suture axis of the bean was included, it would be possible to increase the classification success. However, classification machines in the industry, the seed flows through the orifice quickly. So, the analysis of the third dimension is difficult. Also, the seeds are generally analyzed in two dimensions. In the machine learning technique based on two-dimensional images, the differences in the shape of each bean variety could be used as a separate variable (e.g., coefficient of variation for the roundness of Sira cultivar). If the coefficient of variance is also included in the shape and size variables of each cultivar, the classification success of bean cultivars may increase. Beside shape and size features, the texture features and statistical features can improve the classification results further.