Assignment No.2 - Supervised Learning

Multiclass Classification of Dry Beans

IART 2020/2021 - T4G04

Breno Accioly up201800170@edu.fe.up.pt
Ivo Saavedra up201707093@edu.fe.up.pt
Rodrigo Reis up201806534@edu.fe.up.pt

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Specification

The main purpose of this project is to use supervised learning methods in order to create a multiclass classification model for a given dataset of dry beans.

The provided dataset consists of seven types of dry beans as well as the following features used to identify them:

- **Area (A):** Area of a bean zone and the number of pixels within its boundaries.
- **Perimeter (P):** Bean circumference is defined as the length of its border.
- **Major axis length (L):** Distance between the ends of the longest line that can be drawn from a bean.
- **Minor axis length (I):** Longest line that can be drawn from the bean while standing perpendicular to the main axis.
- **Aspect ratio (K):** Relationship between **L** and **I**.
- **Eccentricity (Ec):** Eccentricity of the ellipse having the same moments as the region.
- **Convex area (C):** Number of pixels in the smallest convex polygon that can contain the area of a bean seed.
- **Equivalent diameter (Ed):** Diameter of a circle having the same area as a bean seed area.
- **Extent (Ex):** Ratio of the pixels in the bounding box to the bean area.
- **Solidity (S):** Ratio of the pixels in the convex shell to those found in beans (convexity).
- **Roundness (R):** Calculated with the following formula: (4piA)/(P^2)
- **Compactness (CO):** Measures the roundness of an object: Ed/L
- **ShapeFactor(1-4):** Shape features used to classify each bean

References

- **Science Direct -** Multiclass classification of dry beans using computer vision and machine learning techniques
  https://www.sciencedirect.com/science/article/abs/pii/S0168169919311573
- **Scikitlearn** - Supervised Learning
  https://scikit-learn.org/stable/supervised_learning.html#supervised-learning
- **Stack Abuse** - Introduction to Neural Networks with Scikit-Learn
  https://stackabuse.com/introduction-to-neural-networks-with-scikit-learn/
- **Scikitlearn -** Nearest Neighbours
  https://scikit-learn.org/stable/modules/neighbors.html
- **Scikitlearn -** SVM
  https://scikit-learn.org/stable/modules/svm.html
- **Vebuso** - SVM parameter tuning with GridSearchCV
  https://www.vebuso.com/2020/03/svm-hyperparameter-tuning-using-gridsearchcv/

Tools and Algorithms

Language: Python 3

Additional Libraries:

1. numpy
2. matplotlib
3. seaborn
4. pandas
5. sklearn
   a. tree
   b. model_selection.train_test_split
   c. neural_network.MPLClassifier
   d. preprocessing.StandardScaler
   e. metrics.plot_confusion_matrix
   f. metrics.classification_report

Implemented Algorithms:

DST
max_depth: 10
min_samples_leaf: 5

SVM
kernel: RBF - for non-linear problems
C: 10
gamma: 0.001

MLP
solver: ‘adam’ - recommended for large datasets
hidden_layer_sizes: 29

KNN
n_neighbours: 5

Work Carried Out

First we plotted the data provided in order to have a better idea of the feature distributions of each bean class.

In order to remove outliers we had to plot every class separately as the initial plot containing all classes was very bloated.

After this preprocessing the data was divided in two groups, 80% for training and 20% for testing.

The four algorithms were then used to create the classification models. Because some of them were not scale invariant, we first needed to scale the data before applying the algorithms.

For each of the models we evaluated their classification precision for each of the classes and their overall accuracy.

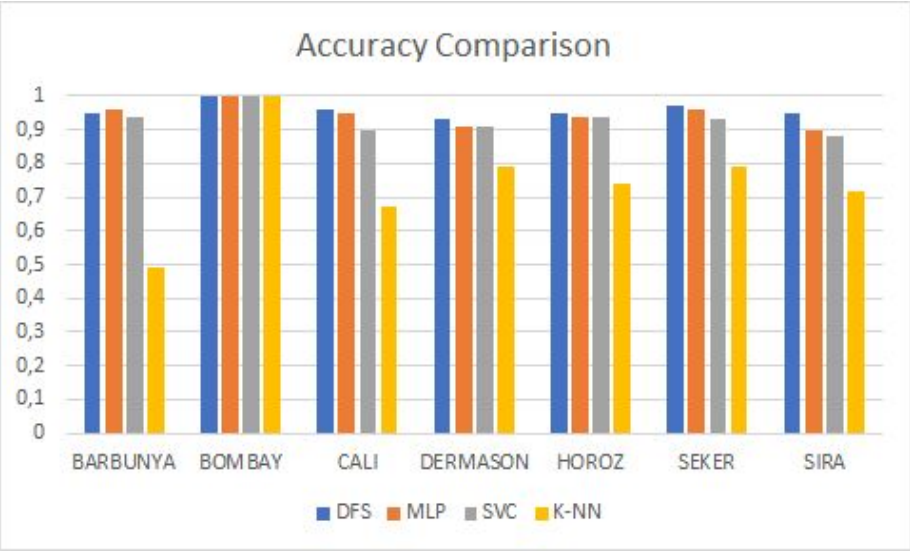

chart 1 - comparison of classification accuracy of each algorithm