# Practical Application 1

Machine Learning
Grupo 20

Cerezo Pomykol, Jan
j.cerezo@alumnos.upm.es

October 2022

## Contents

# 1 Introduction

The goal of this practical application is to study how feature subset selection (FSS) affects different machine learning models. Specifically how they perform with all variables, a univariate filter, a multivatiate filter and a multivariate wrapper. The selected models are the following:

- k-Nearest Neighbours
- Rule Induction
- Support Vector Machines
- Neural Network
- Classification Trees

For this assignment, the dataset chosen [] contains data about instances of dry beans, which was created from images of 13,611 grains of 7 different varieties.

The rest of the document has the following structure. Firstly de dataset is analyzed in the Problem Description, later the metodology is specified in the Metodology section. Subsequently the results are shown in the Results section, and finally they are analyzed in the Discussion and Conclussion sections.

# 2 Problem Description

The datset contains 13,611 instances created from images of dry beans of 7 different varieties. There are a total of 17 attributes (16 plus the class column) and their significance is the following:

1 **Area** (A): The area of a bean zone and the number of pixels within its boundaries.
2 **Perimeter** (P): Length of its border.
3 **Major axis length** (L): Length of the longest line that can be drawn from a bean.
4 **Minor axis length** (l): Length of the longest line that can be drawn from the bean while standing perpendicular to the main axis.
5 **Aspect ratio** (K): Relationship between L and l.
6 **Eccentricity** (Ec): Eccentricity of the ellipse having the same moments as the region.
7 **Convex area** (C): Number of pixels in the smallest convex polygon that can contain the area of a bean seed.
8 **Equivalent diameter** (Ed): The diameter of a circle having the same area as a bean seed area.
9 **Extent** (Ex): The ratio of the pixels in the bounding box to the bean area.
10 **Solidity** (S): Also known as convexity. The ratio of the pixels in the convex shell to those found in beans.
11 **Roundness** (R): Calculated with the following formula: $\frac{4\pi A}{P^2}$
12 **Compactness** (CO): Measures the roundness of an object: $\frac{Ed}{L}$
13 **ShapeFactor1** (SF1)
14 **ShapeFactor2** (SF2)
15 **ShapeFactor3** (SF3)

16 **ShapeFactor4** (SF4)

17 **Class**: Seker, Barbunya, Bombay, Cali, Dermosan, Horoz and Sira.

All atributes are numeric, except the class which is nominal. Additionally there are no *meta-attributes* such as instance or object identifiers, so there is no need to remove any column. Table 1 shows more information about the dataset.

| No. | Name | Min. | Max. |
|---|---|---|---|
| 1 | Area | 20420 | 254616 |
| 2 | Perimeter | 524.736 | 1985.37 |
| 3 | Major axis length | 183.601 | 738.86 |
| 4 | Minor axis length | 122.513 | 460.198 |
| 5 | Aspect ratio | 1.025 | 2.43 |
| 6 | Eccentricity | 0.219 | 0.911 |
| 7 | Convex area | 20684 | 263261 |
| 8 | Equivalent diameter | 161.244 | 569.374 |
| 9 | Extent | 0.555 | 0.866 |
| 10 | Solidity | 0.919 | 0.995 |
| 11 | Roundness | 0.49 | 0.991 |
| 12 | Compactness | 0.641 | 0.987 |
| 13 | ShapeFactor1 | 0.003 | 0.01 |
| 14 | ShapeFactor2 | 0.001 | 0.004 |
| 15 | ShapeFactor3 | 0.41 | 0.975 |
| 16 | ShapeFactor4 | 0.948 | 1 |
| 17 | Class | - | - |

Table 1: Dataset table.

Since there are some variables with high values compared to others (for example Area compared to Extent), normalization is necessary. The amount of instances of each class is shown in Table 2.

| Class | n |
|---|---|
| SEKER | 2027 |
| BARBUNYA | 1322 |
| BOMBAY | 522 |
| CALI | 1630 |
| HOROZ | 1928 |
| SIRA | 2636 |
| DERMASON | 3546 |

Table 2: Numer of instances per class.

# 3 Metodology

## 3.1 Software

For this practical application, Weka [] has been used to perform the traning and evaluation of all models, as well as the normalization of the dataset. Besides from normalization of the dataset, there is no need to perform more preprocessing. There are no missing values and all variables are numerical (except the class).

### 3.2 Evaluation

The evaluation of each model will be conducted with cross validation with 10 folds. Because each class has different weight (different number of instances) the confusion matrix will be discussed as well. Besides the accuarcy, traning and classification time will also be considered.

### 3.3 Classification Algorithms

The following machine learning algorithms have been evaluated. Note that some of them have the option to normalize de input data, which was unset because the data was already normalized.

- **k-Nearest Neighbour**. Since there are 7 classes, the parameter $k$ was set to 7.
- **Rule Induction**. For this classification method the RIPPER algorithm was used.
- **Support Vector Machine**. Non-linear with a polinomial kernel of exponent 2.
- **Neural Network**. With 29 neurons and a single hidden layer.
- **Classification Tree**. C4.5 algorithm. This algorithm is an improved version of the ID3 algorithm that uses a gain ratio to choose attributes. It also penalizes attributes with many values and many uniformly distributed values.

Table 3 shows the Weka function associated to each algorithm.

| Algorithm | Weka Function |
|---|---|
| k-Nearest Neighbours | lazy.IBk |
| Rule Induction (RIPPER) | rules.JRip |
| Support Vector Machine | functions.SMO |
| Neural Network | functions.MultilayerPerceptron |
| Classification Tree (C4.5) | trees.J48 |

Table 3: Weka implementation of each algorithm.

### 3.4 Feature Subset Selection

The selection of features has been performed with the following algorithms. Table 4 shows the Weka function associated to each method.

- **No FSS**. The original dataset was used.
- **Univariant Filter**. Evaluates the worth of each attribute by measuring the information gain with respect to the class. The threshold used for this dataset is 1.2.
- **Multivariant Filter**. Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them.
- **Wrapper Approach**. Evaluates each subset of attributes with the estimated performance of a classifier built with this subset of attributes.

| FSS algorithm | Weka Function |
|---|---|
| No FSS | - |
| Univariant Filter | attributeSelection.InfoGainAttributeEval |
| Multivariant Filter | attributeSelection.CfsSubsetEval |
| Wrapper Approach | attributeSelection.WrapperSubsetEval |

Table 4: Weka implementation of each FSS algorithm.

# 4 Results

This section includes the scores obtained with each classifier (section 3.3) and each Feature Subset Selection approach (section 3.4). Table 5 shows the attributes selected by each method.

| Attribute | No FSS | Univariant | Multivariant | Wrapper (RIPPER) | Wrapper (kNN k=7) | Wrapper (SVM) | Wrapper (MLP) | Wrapper (C4.5) |
|---|---|---|---|---|---|---|---|---|
| Area | • | • | | | | | | • |
| Perimeter | • | • | • | • | • | • | • | |
| MajorAxisLength | • | • | • | | | | | |
| MinorAxisLength | • | • | • | | | | • | |
| AspectRatio | • | | • | | | | • | |
| Eccentricity | • | | | | • | | • | |
| ConvexArea | • | • | • | | | • | | • |
| EquivDiameter | • | • | | • | | • | | • |
| Extent | • | | • | | | • | • | |
| Solidity | • | | | • | • | • | • | • |
| Roundness | • | | • | • | • | • | • | • |
| Compactness | • | | • | • | | • | • | • |
| ShapeFactor1 | • | • | • | • | • | • | • | |
| ShapeFactor2 | • | • | • | | • | | • | • |
| ShapeFactor3 | • | | | • | | | | |
| ShapeFactor4 | • | | • | • | • | • | • | • |
| **N attributes** | 16 | 8 | 11 | 8 | 7 | 9 | 11 | 8 |

Table 5: Attributes selected with each FSS algorithm.

Table 6 shows the score of each classification algorithm with all datasets obtained from each Feature Subset Selection method described, and Table 7 shows the training time of each model. The full version of each result report including the confusion matrices can be found in [].

Note that images of the trees obtained my the C4.5 algorithm, or the neural networks generated are not shown because there are too many nodes to show and the representation is not understandable. All the rules generated from the RIPPER algorithm and the support vectors from the SVMs can be found in the complete result reports. In the same way, a representation of the outputs of the kNN algorithm is not included because it is not possible to visualize a 7 dimenisonal space.

| Dataset | kNN | RIPPER | SVM | MLP | C4.5 |
|---|---|---|---|---|---|
| Original | 91.9477 | 91.4775 | 92.5648 | 92.7852 | 91.4995 |
| Univariate Filter | 90.8897 | 90.2432 | 89.9052 | 91.2718 | 90.3681 |
| Multivariate Filter | 92.1901 | 91.1395 | 92.4987 | 92.5428 | 91.0073 |
| Wrapper (kNN) | 92.4840 | 91.4114 | 92.462 | 92.5428 | 91.3967 |
| Wrapper (RIPPER) | 92.5428 | 91.4261 | 92.4106 | 92.4693 | 91.4628 |
| Wrapper (SVM) | 92.3077 | 91.4187 | 92.5281 | 92.7044 | 91.3452 |
| Wrapper (MLP) | 92.0065 | 91.4334 | 92.5648 | 92.6971 | 91.3085 |
| Wrapper (C4.5) | 92.4326 | 91.4628 | 92.3297 | 92.4546 | 91.4040 |

Table 6: Scores of all classifiers with all obtained datasets (percentage of correctly classified instances).

| Dataset | kNN | RIPPER | SVM | MLP | C4.5 |
|---|---|---|---|---|---|
| Original | <0.1 | 2.38 | 7.98 | 33.37 | 0.3 |
| Univariate Filter | <0.1 | 1.89 | 12.11 | 14.87 | 0.1 |
| Multivariate Filter | <0.1 | 2.43 | 6.16 | 20.67 | 0.18 |
| Wrapper (kNN) | <0.1 | 1.78 | 5.58 | 13.25 | 0.11 |
| Wrapper (RIPPER) | <0.1 | 1.67 | 6.82 | 14.89 | 0.12 |
| Wrapper (SVM) | <0.1 | 1.74 | 6.13 | 16.64 | 0.14 |
| Wrapper (MLP) | <0.1 | 1.91 | 6.71 | 20.56 | 0.17 |
| Wrapper (C4.5) | <0.1 | 1.99 | 6.33 | 15.19 | 0.14 |

Table 7: Training time in seconds.

## 5 Discussion

On the whole, the models trained obtained a score of at least 90% in most cases. However, the results do not differ greatly from each other. The greatest precision obtained is 92.7852%, which was achieved with the original dataset and a Multilayer Perceptron classification algorithm.

Overall the best performing models are the Multilayer Perceptron and the Support Vector Machine, followed closely by the k-Nearest Neighbour. Note that the latter has a training time hundreds times faster compared to the first two.

Regarding the selection of features of this dataset, it is clear that not all variables are needed to obtain models that perform as well as the model generated from all attributes.

In the case of the k-Nearest Neigbours, the reduced datasets obtained a slightly better scores than the original dataset. This is probably due to some moderate overfitting. Generally this model obtained the best results when the training time is considered.

RIPPER (rules)

SVM

MLP

C4.5

Univariate

The rest

Correlation

# 6 Conclusion