Practical Application 2

Machine Learning

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

November 22, 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 a dataset with all variables and datasets obtained from a univariate filter, a multivariate filter, and a multivariate wrapper. The selected classification models and *meta-classifiers* are the following:

- Logistic Regression.
- Naive Bayes.
- Tree Augmented Naive Bayes.
- Linear Discriminant Analysis.
- Fusion.
- Stacking.
- Bagging.
- Random Forest.
- Boosting.
- Naive Bayes Tree.
- Logistic Model Trees.

For this assignment, the dataset chosen [1][2] 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 analysed in the Problem Description, later the methodology is specified in the Methodology section. Subsequently the results are shown in the Results section, and finally they are analysed in the Discussion and Conclusion sections.

2 Problem Description

The dataset 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 attributes 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 ranges of values and Table 2 shows the number of instances of each class.

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: Value ranges of each variable.

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

Table 2: Number of instances per class.

Since there are some variables with high values compared to others (for example Area compared to Extent); normalization is necessary.

3 Methodology

3.1 Software

For this practical application, Weka [3] has been used to perform the training 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 pre-processing. 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 in some cases. Besides the accuracy, training time will also be considered.

3.3 Classification Algorithms

The following machine learning algorithms have been evaluated.

- **Logistic Regression**. Default parameters.
- **Naive Bayes**. Default parameters.
- **Tree Augmented Naive Bayes**. With TAN search algorithm.
- **Linear Discriminant Analysis**. Default parameters.
- **Fusion**. With the following classifiers: Naive Bayes, TAN and LDA. The combination rule used is the Average.
- **Stacking**. The *meta-classifier* is set to Naive Bayes. The classifiers are TAN and LDA.
- **Bagging**. With Naive Bayes classifier.
- **Random Forest**. Default parameters.
- **Boosting**. With Naive Bayes classifier.
- **Naive Bayes Tree**. Default parameters.
- **Logistic Model Trees**. Default parameters.

Table 3 shows the Weka function associated to 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.

Algorithm	Weka Function
Logistic Regression	functions.Logistic
Naive Bayes	bayes.NaiveBayes
Tree Augmented Naive Bayes	bayes.BayesNet
Linear Discriminant Analysis	functions.LDA
Fusion	meta.Vote
Stacking	meta.Stacking
Bagging	meta.Bagging
Random Forest	trees.RandomForest
Boosting	meta.AdaBoostM1
Naive Bayes Tree	trees.NBTree
Logistic Model Trees	trees.LMT

Table 3: Weka implementation of each algorithm.

- **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.

Note that the optimization of each classification model and feature selection method is not part of this practical application.

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.

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. Note that full outputs of every model are not listed due to the limited extension of this assignment. The full version of each result report including the confusion matrices can be found in [4].

Attribute	No FSS	Univariant	Multivariant	Wrapper (Logistic)	Wrapper (Naive Bayes)	Wrapper (TAN)	Wrapper (LDA)	Wrapper (Fusion)	Wrapper (Stacking)	Wrapper (Bagging)	Wrapper (Random Forest)	Wrapper (Boosting)	Wrapper (NBTree)	Wrapper (LMT)
Area		•					•			•	•	•		•
Perimeter	•	•	•	•	•	•	•	•	•	•		•	•	•
MajorAxisLength	•	•	•	•					•		•			•
MinorAxisLength	•	•	•	•		•		•					•	
AspectRatio	•		•			•			•					
Eccentricity	•								•					
ConvexArea	•	•	•	•			•		•					•
EquivDiameter	•	•		•					•					•
Extent	•		•	•		•	•	•	•	•	•	•		
Solidity	•			•				•		•	•		•	•
Roundness	•		•	•	•	•		•	•	•	•		•	•
Compactness	•		•		•	•	•	•		•	•		•	
ShapeFactor1	•	•	•	•	•	•				•	•			
ShapeFactor2	•	•	•	•					•	•				•
ShapeFactor3	•										•	•		
ShapeFactor4	•		•	•	•	•	•	•	•	•	•		•	•
N attributes	16	8	11	11	5	8	6	7	10	9	9	4	6	9

Table 5: Attributes selected with each FSS algorithm.

5 Discussion

Generally, 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 92.84%, which was achieved with the dataset obtained from the Wrapper Algorithm (Random Forest), and the Random Forest classifier.

Overall, the best performing models are the Logistic Regression, the Random Forest, and the Logistic Model Tree. The worst being the Linear Discriminant Analysis classifier, and the Bagging and Boosting *meta-classifiers*.

Respecting to training time, the fastest models are the Naive Bayes, Tree Augmented Naive Bayes and Linear Discriminant Analysis classifiers, while the slowest are the Logistic Regression and the Logistic Model Trees.

Regarding the selection of features of this dataset, not all variables are needed to obtain models that perform as well as the model generated from all attributes. Table 5 shows that most Feature Subset Selection methods selected the Perimeter, Extent, Roundness and ShapeFactor4, which resembles the importance of these attributes. Similar results were obtained in the previous assignment, where the most selected features were the Perimeter, Roundness, ShapeFactor1 and ShapeFactor4.

Dataset	Logistic	Naive Bayes	TAN	LDA	Fusion	Stacking	Bagging	Random Forest	Boosting	NBTree	LMT
Original	92.60	89.71	91.47	90.18	91.26	91.28	89.72	92.52	89.71	89.57	92.49
Uni. Filter	92.14	84.09	89.90	89.22	90.00	90.29	84.03	91.04	84.09	87.67	91.94
Mult. Filter	92.57	90.20	91.24	90.05	91.58	91.74	90.31	92.47	90.20	90.63	92.41
Wr. (Logistic)	92.70	89.01	91.47	90.03	91.47	91.53	89.08	92.53	89.01	89.66	89.01
Wr. (N. Bayes)	92.09	91.23	91.54	89.84	89.01	91.77	91.21	92.16	91.23	91.55	92.16
Wr. (TAN)	92.36	90.76	91.60	89.83	91.72	91.62	90.80	92.33	90.76	90.69	92.27
Wr. (LDA)	92.30	88.23	90.42	91.17	91.35	91.56	88.34	91.74	88.23	89.57	92.35
Wr. (Fusion)	92.39	91.05	91.29	90.58	91.91	91.24	91.05	92.68	91.05	90.88	92.44
Wr. (Stacking)	92.55	89.42	91.66	89.86	91.38	92.20	89.45	92.46	89.42	89.97	92.41
Wr. (Bagging)	92.53	90.77	91.44	89.45	91.58	91.78	90.75	92.70	90.77	90.90	92.54
Wr. (R. Forest)	92.38	90.66	91.27	89.72	91.58	91.53	90.65	92.84	90.66	90.33	92.56
Wr. (Boosting)	91.12	80.83	89.60	88.85	89.39	89.69	80.89	91.11	80.83	89.48	91.42
Wr. (NBTree)	92.21	91.22	91.24	90.56	91.89	91.17	91.22	92.46	91.22	91.27	92.27
Wr. (LMT)	92.45	84.75	91.02	90.32	91.22	91.30	84.80	92.28	84.75	89.82	92.52

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

In the case of the Logistic Regression classifier, the scores achieved are amongst the highest, the best one obtained with the dataset from the Wrapper with Logistic Regression classifier. This score is better than the one obtained from the original dataset, which suggests that the most relevant features for this classifier are the Perimeter, MajorAxisLength, MinorAxisLength, ConvexArea, EquivDiameter, Extent, Solidity, Roundness, ShapeFactor1, ShapeFactor2 and ShapeFactor4. Figure 1 displays the output of the coefficients and odds ratios for the dataset obtained with the Wrapper approach (with Naive Bayes). As we can see there are some variables that are more explanatory than others. For instance, in the class BARBUNYA the most relevant attribute is the perimeter.

```
Coefficients...
                             Class
Variable                     SEKER            BARBUNYA              BOMBAY                CALI               HOROZ                SIRA
=======================================================================================================================================
Perimeter                 122.6821             83.2646             722.999             57.7438            147.5876            -96.0632
roundness                   5.1908            -17.4808            358.6025             16.7366             16.8368            -23.7141
Compactness                47.5217             10.6809            195.6578            -36.1733            -13.6638            -38.3752
ShapeFactor1               14.7379            -59.4434            348.0428            -95.3347             47.0252           -106.6375
ShapeFactor4               29.8703              8.9363           -134.3783            -16.5178            -17.6388            -10.7245
Intercept                 -86.6715             11.6021           -642.3413             50.4722            -45.2959            121.7997


Odds Ratios...
                             Class
Variable                     SEKER            BARBUNYA              BOMBAY                CALI               HOROZ                SIRA
=======================================================================================================================================
Perimeter    1.9060785863351933E53 1.4499412945562452E36            Infinity 1.1961829114581766E25 1.2487633551787505E64              0
roundness                 179.6034                   0 5.483927383541303E155       18561985.0514       20518650.4473                   0
Compactness   4.349342686745699E20          43517.6216  9.39975190503262 4E84                   0                   0                   0
ShapeFactor1          2515213.6001                   0 1.422496489365307E151                   0 2.6471799061105616E20                   0
ShapeFactor4  9.386358439569479E12           7602.8993                   0                   0                   0                   0
```

Figure 1: Coefficients and odds ratios of the Logistic Regression model.

The Naive Bayes classifier trained the model significantly faster than the others, while reducing slightly the score. Figure 2 shows a simplified output of the model obtained with the reduced dataset, and the confusion matrix.

In this case, most misclassified instances are from the classes SIRA and DERMASON. This is because both classes have similar mean values in all attributes. From this output we can also deduce what types of beans have higher perimeter (the largest are from the class BOMBAY). Additionally, most classes have high roundness (above 0.7), except BARBUNYA and HOROZ, which suggests that these two varieties have longer

Dataset	Logistic	Naive Bayes	TAN	LDA	Fusion	Stacking	Bagging	Random Forest	Boosting	NBTree	LMT
Original	57.5	0.02	0.14	0.02	0.17	1.67	0.19	3.99	1.17	11.51	14.38
Uni. Filter	2.39	0.01	0.06	0.01	0.07	0.7	0.09	3.1	0.61	5.83	10.58
Mult. Filter	3.89	0.01	0.09	0.01	0.11	1.06	0.15	3.18	0.84	7.17	11.41
Wr. (Logistic)	5.81	0.01	0.09	0.01	0.11	1.06	0.13	3.21	1.14	10.54	12.27
Wr. (N. Bayes)	1.37	0.01	0.03	0.01	0.04	0.45	0.08	2.38	0.46	1.92	8.39
Wr. (TAN)	2.18	0.01	0.06	0.01	0.07	0.74	0.1	3.14	0.51	6.52	9.58
Wr. (LDA)	1.99	0.01	0.04	0.01	0.07	0.53	0.08	2.48	0.39	2.9	13
Wr. (Fusion)	1.97	0.01	0.05	0.01	0.06	0.63	0.09	2.46	0.58	5.23	9.86
Wr. (Stacking)	2.69	0.01	0.07	0.01	0.09	0.92	0.12	3.21	0.99	8.71	10.74
Wr. (Bagging)	2.44	0.01	0.07	0.01	0.09	0.82	0.12	3.17	0.67	5.63	10.57
Wr. (R. Forest)	2.51	0.01	0.07	0.01	0.09	0.82	0.11	3.21	0.87	8.61	10.33
Wr. (Boosting)	0.97	0.01	0.03	0.01	0.03	0.36	0.06	2.18	0.46	1.43	8.26
Wr. (NBTree)	0.96	0.01	0.04	0.01	0.06	0.53	0.09	2.44	0.45	3.96	9.05
Wr. (LMT)	7.95	0.01	0.06	0.01	0.08	0.83	0.11	3.28	0.56	8.26	16.41

Table 7: Training time in seconds.

shapes. Overall the most discriminant attributes are the Perimeter, Compactness and ShapeFactor1.

The Tree Augmented Naive Bayes classifier obtained better scores than the standard Naive Bayes classifier, while slightly increasing the training times. Figure 3 shows the tree obtained from the reduced dataset with the Wrapper approach (Naive Bayes), and tables containing probability distributions of each class for the attributes Class and ShapeFactor1.

The Linear Discriminant Analysis obtained better scores than the standard Naive Bayes, while taking the same amount of time to train the models. Figure 4 shows the mean vectors of some classes of the model obtained from the original dataset. In this case we can see that the least explanatory attributes are the Extent, Solidity and ShapeFactor4.

In the case of the meta-classifiers, the scores are diverse. The best performing models are the Fusion, Stacking, Random Forest and Logistic Model Trees. Note that the last two are significantly slower to train.

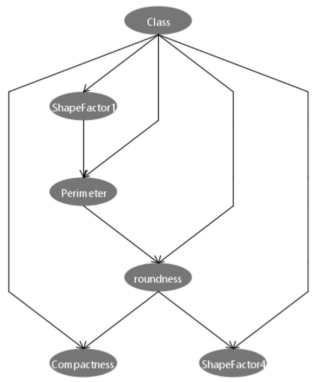
Inspecting Table 6 we can deduce that in most cases the reduced datasets obtain similar or better scores when compared to the original dataset. This is because there are variables that provide no additional information, in other words, there are dependent or irrelevant variables in the dataset. As stated in the Problem Description section, there are some variables that are calculated from others. For instance, the *Roundness* is obtained from the *Area* and the *Perimeter*, and the *Compactness* from the *Equivalent diameter* and the *Major axis length*. Additionally, there are some attributes that are fundamentally related, such as the *Area* and the *Perimeter*.

Overall, all the reduced datasets obtained similar scores compared to the original dataset. The cause of this is the redundancy of some variables. However, the scores are not close to 100%, probably because the dataset contains multiple irrelevant attributes, and the isolated *important* features need more complex models to fit the distribution. This

```
                    Class
Attribute           SEKER  BARBUNYA   BOMBAY     CALI    HOROZ     SIRA  DERMASON
                   (0.15)     (0.1)   (0.04)   (0.12)   (0.14)   (0.19)    (0.26)
==============================================================================
Perimeter
  mean              0.1389   0.3569   0.7263   0.3648   0.2705    0.186   0.0962
roundness
  mean              0.9078   0.6198    0.748   0.7111   0.6083   0.7884   0.8352
Compactness
  mean              0.7391   0.4742   0.4385   0.3349   0.1739   0.4521   0.5149
ShapeFactor1
  mean              0.4635   0.3362   0.0865   0.3494   0.5511   0.5137   0.6486
ShapeFactor4
  mean              0.9741   0.9233   0.8484   0.8242     0.85   0.9165   0.9458

=== Confusion Matrix ===
    a    b    c    d    e    f    g   <-- classified as
 1922   27    0    0    1   55   22 |   a = SEKER
    5 1161    1  106   12   37    0 |   b = BARBUNYA
    0    0  522    0    0    0    0 |   c = BOMBAY
    2   90    0 1488   39   11    0 |   d = CALI
    0    6    0   28 1851   29   14 |   e = HOROZ
   47    7    0   10   67 2307  198 |   f = SIRA
   87    4    0    0   23  265 3167 |   g = DERMASON
```

Figure 2: Model and confusion matrix of the Naive Bayes classifier.



Probability Distribution Table For Class

SEKER	BARBUN...	BOMBAY	CALI	HOROZ	SIRA	DERMAS...
0,149	0,097	0,038	0,12	0,142	0,194	0,26

Probability Distribution Table For ShapeFactor1

Class	'\'(-inf-...	'\'(0.17...	'\'(0.26...	'\'(0.32...	'\'(0.36...	'\'(0.38...	'\'(0.40...	'\'(0.43...	'\'(0.47...	'\'(0.51...	'\'(0.52...	'\'(0.53...	'\'(0.55...	'\'(0.59...	'\'(0.62...	'\'(0.66...	'\'(0.70...	'\'(0.76...
SEKER	0	0	0,001	0,015	0,026	0,064	0,128	0,306	0,347	0,062	0,023	0,021	0,004	0,002	0	0	0	0
BARBUNYA	0,001	0,1	0,329	0,3	0,073	0,074	0,06	0,044	0,012	0	0,003	0	0,001	0	0	0	0	0
BOMBAY	0,984	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001
CALI	0	0,024	0,233	0,407	0,146	0,097	0,058	0,023	0,009	0	0	0	0,001	0	0	0	0	0
HOROZ	0	0	0	0,002	0,001	0,002	0,014	0,081	0,199	0,088	0,067	0,116	0,171	0,128	0,063	0,055	0,012	0
SIRA	0	0	0	0	0,001	0,001	0,012	0,135	0,372	0,128	0,106	0,107	0,091	0,035	0,01	0,001	0	0
DERMASON	0	0	0	0	0	0	0	0	0,018	0,017	0,028	0,059	0,134	0,163	0,16	0,192	0,15	0,077

Figure 3: Tree obtained from the Tree Augmented Naive Bayes classifier.

conclusion is also achieved in the previous assignment.

6 Conclusion

In this practical application, a dataset of dry beans with 13,611 instances and 17 features (Including the class) was analysed with different attribute selection methods and different classifiers with a 10-fold cross-validation test approach. The dataset contains instances of seven different classes. The only pre-processing needed was the normalization of the dataset, since there are no missing values or nominal variables (except the class column).

Different datasets were obtained from the original by extracting features with a Univariate Filter, Multivariate Filter, and different Wrapper approaches. All of which were tested with the following classifiers: Logistic Regression, Naive Bayes, Tree Augmented Naive Bayes and Linear Discriminant Analysis. Additionally, the following *meta-classifiers* were used as well: Fusion, Stacking, Bagging, Random Forest, Boosting, Naive Bayes Tree and Logistic Model Trees.

Estimates for class value SEKER	Estimates for class value BARBUNYA	Estimates for class value BOMBAY
Mean vector:	Mean vector:	Mean vector:
Area: 0.08	Area: 0.21	Area: 0.65
Perimeter: 0.14	Perimeter: 0.36	Perimeter: 0.73
MajorAxisLength: 0.12	MajorAxisLength: 0.34	MajorAxisLength: 0.74
MinorAxisLength: 0.24	MinorAxisLength: 0.35	MinorAxisLength: 0.75
AspectRation: 0.16	AspectRation: 0.37	AspectRation: 0.4
Eccentricity: 0.53	Eccentricity: 0.77	Eccentricity: 0.8
ConvexArea: 0.08	ConvexArea: 0.21	ConvexArea: 0.64
EquivDiameter: 0.16	EquivDiameter: 0.33	EquivDiameter: 0.75
Extent: 0.7	Extent: 0.62	Extent: 0.71
Solidity: 0.94	Solidity: 0.84	Solidity: 0.9
roundness: 0.91	roundness: 0.62	roundness: 0.75
Compactness: 0.74	Compactness: 0.47	Compactness: 0.44
ShapeFactor1: 0.46	ShapeFactor1: 0.34	ShapeFactor1: 0.09
ShapeFactor2: 0.64	ShapeFactor2: 0.27	ShapeFactor2: 0.09
ShapeFactor3: 0.7	ShapeFactor3: 0.42	ShapeFactor3: 0.39
ShapeFactor4: 0.97	ShapeFactor4: 0.92	ShapeFactor4: 0.85

Figure 4: Mean vectors of classes SEKER, BARBUNYA and BOMBAY (LDA model).

Overall, the precision obtained is similar in most cases, the best ones obtained by the Logistic Regression, Random Forest and Logistic Model Trees. For the most part, the reduced datasets obtained the same or better scores than the original dataset. Contrary to the first assignment, were the reduced dataset performed slightly worse, this assignment shows that some models can fit a reduced dataset to obtain better scores. Both Naive Bayes and Linear Discriminant Analysis performed the best when the execution time is taken into consideration. Note that the optimization of each model (finding the best combination of hyperparameters) is not part of this assignment, but it could be an interesting subsequent work.

In summary, the dataset used in this practical application contains variables that are not needed to achieve similar scores to the original dataset. The best precision was obtained by the Random Forest. However, the standard Naive Bayes and the Linear Discriminant Analysis achieved similar scores while spending significantly less computational time to train the model.

References

[1] "Dry Bean Dataset." UCI Machine Learning Repository, 2020.

[2] M. Koklu and I. A. Ozkan, "Multiclass classification of dry beans using computer vision and machine learning techniques," *Computers and Electronics in Agriculture*, vol. 174, p. 105507, 2020.

[3] E. Frank, M. A. Hall, and I. H. Witten, *The WEKA Workbench*. Morgan Kaufmann, fourth edition ed., 2016.

[4] J. Cerezo Pomykol, "Machine learning repository." Available at `https://github.com/ershimen/MachineLearning/tree/main/Assignment2`.