Machine learning techniques and applications for breast cancer diagnosis

Abstract

Breast cancer is the most common cancer amongst women in the World and affected over 2.3 million people just in 2020.
Which is the optimal classification algorithm to distinguish between malignant and benign tumours?
Which are the common characteristics of the clinical records among the two different diagnoses?
The research identified, after applying machine learning techniques, the Multilayer Perceptron classifier with feature selection as the optimal solution for the binary classification problem on the target variable *diagnoses,* hence it can forecast the nature of the tumour (malignant or benign), given the patient's parameters of interest. Just four features were used by the optimal classifier with a dimensionality reduction of the 83%. Furthermore, a clustering analysis was applied through the Fuzzy C-means algorithm to extract the main characteristics of the analysed records over a subset of features. It showed the presence of two well defined clusters, one for each of the diagnosis values. It was shown the presence of a third cluster too, identified as located between the two more extreme ones, that improved the overall validity by containing borderline cases that can't be totally handled with just two clusters. That kind of records should need a higher clinical investigation to avoid dangerous misclassifications.

Luca Milazzo, Nicholas Mondella, Marco Pagnini, Stefano Quartuccio

Contents

Introduction

Breast cancer is the most common cancer amongst women in the World. It accounts for 25% of all cancer cases and affected over 2.3 million people in 2020 alone (World Health Organization, 2021). The aims of the research are:

- Identifying the optimal classification algorithm to distinguish between malignant and benign breast tumours, hence to forecast the diagnosis (a binary classification problem);
- Executing a clustering analysis to identify common characteristics of the clinical records among the two different diagnoses.

Data exploration

The two following chapters are dedicated to the exploration and pre-processing of the analysed data frame. The classification chapter try to answer to the first research goal by analysing four different classification algorithms (Logistic regressor, Naïve bayes tree, Multilayer perceptron and J48 decision tree) tested and trained using different approaches (hold-out, cross validation, feature selection and cost sensitive learning). The clustering chapter, related to the second research question, exploits the Fuzzy C-Means algorithm through different approaches by differentiating the number of features and clusters. The conclusion and future extension chapter ends the whole paper by answering to the research questions and suggesting some analysis extensions. The Appendix section contains details about the data set and additional contents that will be described soon. All the pictures are in high resolution format, so it's easy to gather more details by zooming them. The machine learning and data analysis software, called Knime, was used for the entire pipeline *(Wiswedel, 2007)*.

The subject of this research is the breast cancer data frame, downloaded from the web platform Kaggle *(Kaggle, s.d.)*. It contains 30 features and one binary target column called *diagnosis,* which values are M for malignant records and B for benign records (all the features are listed in the Appendix section). Each of the 569 objects contains medical details of interest for a certain patient. The 37% of the records are labelled as malignant tumours, hence the class labels are not well balanced. In total, there are just 10 different medical measures, such as radius and perimeter, that are associated to three different features in the dataset: mean (mean of all the measured samples for a patient), worst (worst measure recorded) and se (standard error of all the measurements). For example, the radius measure is represented by *radius_worst, radius_mean and radius_se*. Furthermore, the data set does not contain any missing values. The analysis of correlation between the features, based on Pearson's product-moment coefficient, suggested the presence of medium and strong relationships, as shown in the following heatmap (blue and red for high positive and negative correlation values):

Classification

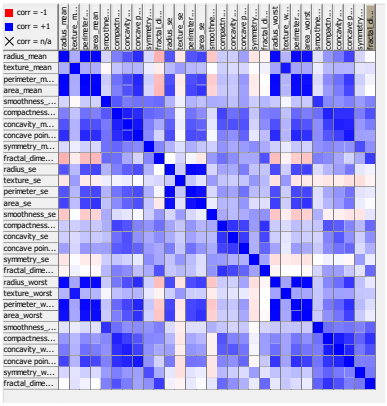

Figure 1-Correlations heatmap

Pre-processing

For this reason the right selection of features in the classification process is relevant.

The min-max normalization, included in the *normalizer* Knime node, in the range $[0,1]$ was applied to avoid influences of features having large domains. Then, the *shuffle* node was used to shuffles the rows of the normalized data set such that they are in random order. From now on, the data set will be intended as the one produced after the pre-processing step.

Different classification algorithms and training models were chosen to forecast the *diagnosis* target variable. In particular, the following Knime nodes were used (Weka 3.7 version):

- *Logistic,* a multinomial logistic regression model with a ridge estimator;
- *NBTree,* a decision tree model with naive Bayes classifiers at its leaves;
- *MultilayerPerceptron,* a neural network based on backpropagation;
- *J48,* a pruned decision tree.

Different training model were used: hold-out, cross validation, feature selection and cost sensitive learning.

Models

Hold-out

The hold-out method was used to train the algorithms on a partition of the original dataset, containing the 70% of records, and then to test them on the remaining data

points. The stratified sampling technique, with fixed seed across the algorithms, was applied to identify both the partitions and to maintain the original proportions of the target variable. The hold-out method could lead, in some cases, to overfitting and low performances, hence the use of the cross-validation technique can help in improving the classification task (Shukla, 2016).

Cross validation

The *k*-fold cross validation technique, with stratified sampling and fixed seed, was used to avoid the overfitting phenomenon that could be caused by the hold-out approach. In particular, k=*5* was identified as best option in relation to the dataset cardinality. Four folds have a cardinality of 114 records and the remaining one has 113 records instead.

Feature selection

The wrapper approach was chosen to identify a subset of features to use for the classification step (Mark A. Hall and Lloyd A. Smith, 1999). More in depth, the Knime node *AttributeSelectedClassifer* (weka 3.7) was exploited on a partition of the dataset containing the 70% of the records which was obtained through the stratified sampling technique. The selection was based on the *wrappersubseteval* evaluator with 4-fold cross validation on the optimization of the F-measure, which represents a trade-off between precision and recall metrics. This evaluation metric is really useful in imbalance class problems as the one analysed in this study. The wrapper cross validation is based on just four folds because the exploited data frame is smaller than the one used in the simple cross validation method with *k=5*. Following the feature filtering, the cross-validation method, as explained in the previous chapter, was applied to train the final model on the dataset containing only the filtered features. The just mentioned dataset is created by merging the partition used by the wrapper and the excluded one too. The wrapping and training phases always shared the same classification algorithm.

Cost sensitive

The aim of all the analysed classification algorithms is to classify patients, hence records, as having a malignant or a benign tumour. It is therefore totally permissible to associate greater cost to erroneously malignant records classified as benignant. Furthermore, the malignant diagnosis represents the minority class level with a relative frequency of 0.37. In this kind of situations, it is difficult to identify the best cost matrix because of the absence of a concrete cost associated with certain misclassifications. For this study, the following cost matrix was used by considering as the positive class the malignant value and as negative class the benignant one:

~~Actual~~ / Classified	*+1(M)*	*-1(B)*
+1(M)	*0*	*10*
-1(B)	*1*	*0*

Table 1 - Cost matrix

As showed, the optimal cost ratio is 1:10 and it was empirically determined. The Knime node *CostSensitiveClassifier* (Weka 3.7), a metaclassifier which makes its base classifier as cost-sensitive, was used to minimize the expected cost of the classification algorithms through a cross validation approach with *k=5*, as already described in the Cross validation paragraph.

Evaluation

The four algorithms were evaluated through five metrics:

- *Accuracy*
  $$acc = \frac{TP+TN}{TP+TN+FP+FN}$$
- *Error*
  $$e = \frac{FP+FN}{TP+TN+FP+FN} = 1 - acc$$
- *F-measure*
  $$F = \frac{2rp}{r+p}$$
  $$r = recall = \frac{TP}{TP+FN}$$
  $$p = precision = \frac{TP}{TP+FP}$$
- *Area Under the Curve (AUC)*
- *Cost*
  $$cost = \frac{(TP*0+TN*0+FP*1+FN*10)}{TP+TN+FP+FN}$$

TP is equal to true positive records (correctly classified malignant records), TN is equal to true negative records (correctly classified benignant records), FP and FN represent the number of misclassified records for their respective classes. The accuracy metrics computes the percentage of correctly classified records, but it shall not be used as unique evaluation measure because of the imbalanced class problem. The confidence interval of the accuracy was computed too by using the Wilson score interval (Wilson, 1927), with a confidence level of 95% ($1 - \alpha$):

$$\boldsymbol{P}\left(acc_{\,true} \in \left(acc + \frac{Z^2_{1-\frac{\alpha}{2}}}{2N} \pm \left(\frac{Z_{1-\frac{\alpha}{2}} * \sqrt{\frac{acc}{N} - \frac{acc^2}{N} + \frac{Z^2_{1-\frac{\alpha}{2}}}{4N^2}}}{1 + \frac{Z^2_{1-\frac{\alpha}{2}}}{N}}\right)\right)\right) = 1 - \alpha$$

In an analogous way, the confidence interval for the difference between errors of cross validation-based methods was computed. In particular, given two different models, M_1 and M_2, tested on the same *K* folds of the original data set, the following confidence interval at level 95% is used:

$$\boldsymbol{P}\left(\bar{d}_{true} \in \left(\bar{d} - t^{K-1}_{1-\frac{\alpha}{2}} * \widehat{\sigma_{\bar{d}^{cv}}}, \bar{d} + t^{K-1}_{1-\frac{\alpha}{2}} * \widehat{\sigma_{\bar{d}^{cv}}}\right)\right) = 1 - \alpha$$

$$\bar{d} = \frac{1}{K}\sum_{i=1}^{K} e_{1i} - e_{2i}$$

$$\widehat{\sigma_{d^{cv}}^2} = \frac{\sum_{i=1}^{K}(e_{1i} - e_{2i} - \bar{d})^2}{K(K-1)}$$

The F-measure is well structured to represent a trade-off between precision and recall, hence it's very useful when dealing with imbalanced class classification problems. The AUC is the area under the ROC curve (Receiver Operating Characteristic curve) which compares the True Positive Rate (TPR or recall) of a classifier with its False Positive Rate (FPR) at distinct size levels of the tested dataset. Finally, the cost of the classification algorithms is computed by using the cost matrix already shown and by normalizing it in $[0; 1]$ to compare different test sizes too. For the cross-validation approaches, each metric is computed on the merged results belonging to the different folds. For a clearer understanding, it is useful to mention that the three approaches based on cross validation share the same folds and their overall metrics are calculated on the results obtained in the 100% of the dataset. While, for the hold-out method, the metrics are obtained just on the 30% of the original dataset.

All the algorithms were evaluated by using four different setups: simple hold-out, simple cross validation, feature selection with cross validation and finally cost sensitive learning with cross validation. The following legend is used for each of the metrics columns:

Best value	*Second best*	*Third best*	*Worst value*

Logistic regression

It is a multinomial logistic regression model with a ridge estimator for computing the regression coefficients. The following table contains its evaluation metrics:

Type	*acc*	*e*	*F-measure*	*AUC*	*Cost*
Hold-out	0.936	0.064	0.915	0.985	0.327
Cross validation	0.954	0.046	0.939	0.985	0.236
Feature selection	0.972	0.028	0.962	0.99	0.202
Cost sensitive	0.951	0.049	0.935	0.951	0.207

Table 2 - logistic classifier

The feature selection classifier with cross validation has the best overall results. Six attributes were identified by the wrapper (number 7, 11, 16, 23, 24 and 26), and despite a dimensionality reduction of the 80%, It owns the best performances. The cost sensitive classifier is near the best cost value, but with lower metrics in the other cases. By excluding the hold-out method, which has the worst performances over a smaller test set, the confidence intervals for the differences between errors were computed (see table 14 in Appendix) for the three approaches based on cross validation. In particular, there is not a significant error difference between the classifiers.

Naïve Bayes Tree

It is a probabilistic classification model which structure is based on a classic decision tree, but with naïve bayes classifiers at its leaves. The following table contains its evaluation metrics:

Type	*acc*	*e*	*F-measure*	*AUC*	*Cost*
Hold-out	0.918	0.082	0.889	0.941	0.503
Cross validation	0.946	0.054	0.927	0.97	0.308
Feature selection	0.944	0.056	0.925	0.973	0.309
Cost sensitive	0.946	0.054	0.928	0.946	0.228

Table 3 - NBTree classifier

As for the logistic algorithm, the hold out method owns the worst results. The cost sensitive approach can be identified as the optimal classifier. In fact, it owns the optimal values except for the AUC measure. Five attributes were identified by the wrapper (number 21, 22, 23, 26 and 29), and despite a dimensionality reduction of the 83%, Its results are similar to the ones associated to the cost sensitive learner which exploits the entire dataset. Neither one of the cross-validation methods have significant error differences (see table 17 in Appendix) as suggested by table 3. Again, the hold-out method was excluded from this last comparison.

Multilayer perceptron

It is a classifier that uses backpropagation to classify instances and it was exploited with $a = \frac{|attributes+classes|}{2}$ number of hidden layers.

Type	*acc*	*e*	*F-measure*	*AUC*	*Cost*
Hold-out	0.947	0.053	0.927	0.988	0.421
Cross validation	0.968	0.032	0.957	0.993	0.221
Feature selection	0.981	0.019	0.974	0.991	0.162
Cost sensitive	0.97	0.03	0.96	0.969	0.141

Table 4 - MLP classifier

The best cost is obtained by the cost sensitive learner, while all the other optimal values, with exception for the AUC, are owned by the feature selection classifier. It identified five key features to use (number 8, 20, 23, 25 and 26) causing a dimensionality reduction of the 83%. In this case, just the difference of the errors between the feature selection method and cost sensitive one was significant (see table 15 in Appendix). It should be noticed that even though the cross validation owns a higher error than the cost sensitive learner, its difference with the feature selection is not significant because of the different distribution of errors across the folds (one of the folds has 113 records).

J48 Decision Tree

The last algorithm to evaluate is a pruned decision tree that terminates non-critical and redundant sections for better accuracy and less complexity. The following table reports its performances:

Type	*acc*	*e*	*F-measure*	*AUC*	*Cost*
Hold-out	0.906	0.094	0.871	0.904	0.62

Cross validation	0.928	0.072	0.904	0.921	0.357
Feature selection	0.94	0.06	0.919	0.945	0.344
Cost sensitive	0.921	0.079	0.897	0.922	0.332

Table 5 - J48 classifier

The hold-out method produces again the worst values, while the feature selection classifier has the best results, with exception for the optimal cost that is owned by the cost sensitive classifier. The wrapper has selected four features (number 6, 13, 25 and 29) with a dimensionality reduction of the 86.6%. Again, just the difference of errors between the feature selection method and cost sensitive one showed a significant difference in favour of the latter (see table 16 in Appendix).

Feature selection and cost sensitive learning

The previous sections showed the great power of the feature selection over the overall metrics. This is explained by the high correlations existing between features and by the semantic of the features themselves. In fact, multiple features represent the same measure, such as *perimeter,* over different views as *mean* of the perimeters or *worst* measured perimeter. For this reason, the dimensionality reduction was applied before the cost sensitive learning too, which is associated with lower costs, in order to combine their power. The following table describes their performances:

Algorithm	***acc***	***e***	***F-measure***	***AUC***	***Cost***
Logistic	0.924	0.076	0.906	0.936	0.139
NBTree	0.942	0.058	0.924	0.944	0.216
MLP	0.968	0.032	0.958	0.97	0.111
J48	0.93	0.07	0.908	0.931	0.292

Table 6 - Feature selection and cost sensitive learning

In this case, the colours are referred to the corresponding tables of each algorithm that were already presented, hence the table is not coloured to compare the four algorithms. The costs are the best ever obtained for all the algorithms, while the other metrics don't reach better values than the approaches already presented. In summary, the feature selection step shall be always used before the cost sensitive learning. In fact, the dimensionality reduction always leaded to smaller costs.

Summary

The feature selection approach is generally the best choice given the applied dimensionality reduction. The cost-sensitive method has a heavy cost impact and is greatly improved by feature selection, which therefore seems to be of high importance. As far as comparisons between algorithms are concerned, the MLP with feature selection achieves the best accuracy, equal to 0.98, and error, hence a good overall score, furthermore it possesses the best F-measure, equal to 0.974, and a good AUC of 0.99, hence a great adaptability to imbalanced classes. Finally, it also owns the smaller cost, equal to 0.11, with the combined use of feature selection and cost sensitive learning. For more details, the confidence intervals of the respective accuracies were computed and visualized using the following boxplots grouped by testing approach:

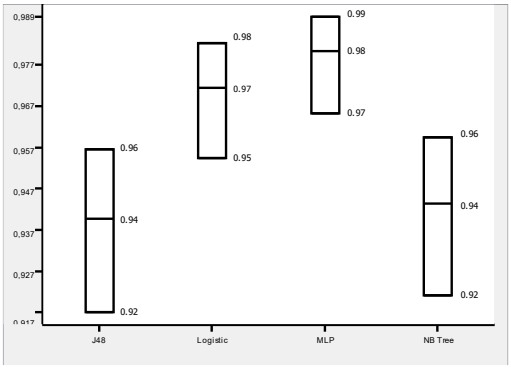

Figure 2 - Simple cross validation

Clustering

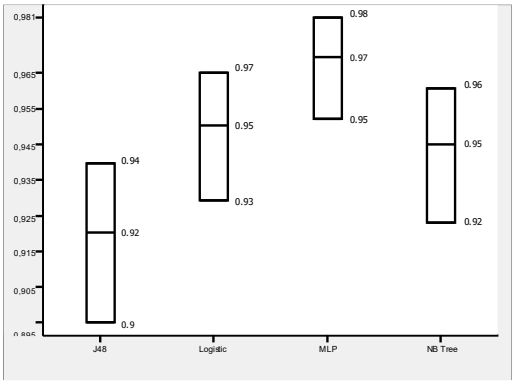

Figure 3 - Cost sensitive learning

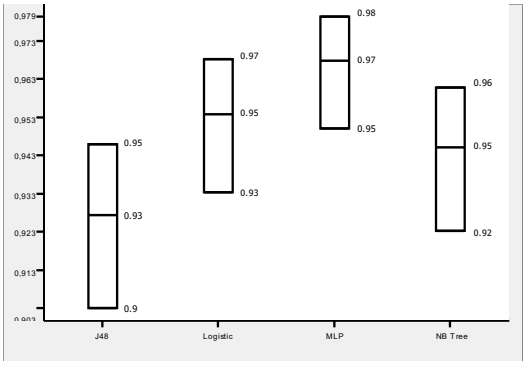

Figure 4 - Feature selection

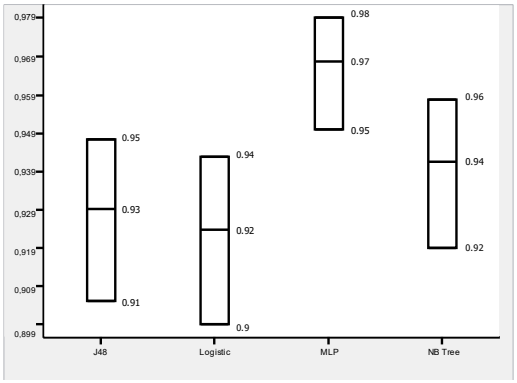

Figure 5 - Feature selection and cost sensitive learning

For the simple cross validation (*Figure 2*), a statistically significant gap between the MLP and the other classifiers is present with exception for the logistic regressor that has a similar accuracy interval. In the case of the feature selection approach (*Figure 4*), there is just a significant difference between the MLP and the J48 decision tree. All the other boxplots are overlayed, but the MLP seems to be settled in a higher interval. In the case of the cost sensitive classifier *(Figure 3),* there's an evident gap between the accuracies of MLP and the ones of the J48 classifier. For what concerns the feature selection with cost sensitive learning *(Figure 5),* the MLP algorithm has a statistically higher accuracy level than the J48 and Logistic algorithms.

As result of the classification analysis, the feature selection process was identified as an important procedure for an optimal classification. Then, a clustering analysis was chosen to identify the behaviour of the data point in relation to the value of the target variable *diagnosis*, hence the differences between patients having a malignant or benign tumour. It was chosen to reduce the original dataset for the clustering task to remove noisy features. In particular, two different data frames were used: D1 with 14 features and D2 with 4 features. The only nominal value, the target variable, called *diagnosis* is not included neither in D1 nor in D2, hence it is not exploited by the clustering algorithm. The 14 features of D1 (number 6,7,8, 11, 13, 16, 20, 21, 22, 23, 25, 26 and 29) were identified by the feature selection process used for the classification task. Only the features that were identified as relevant by at least one algorithm were included in D1. A similar approach was used to identify the attributes in D2. The four features of D2 (number 23, 25, 26 and 29) were chosen as the ones identified as relevant by at least two different algorithms. First, a Principal Component Analysis (PCA) on three components was used to visualize the distribution of records, grouped by the *diagnosis* attribute, belonging to D1 and then the same process was applied for D2 on only two components:

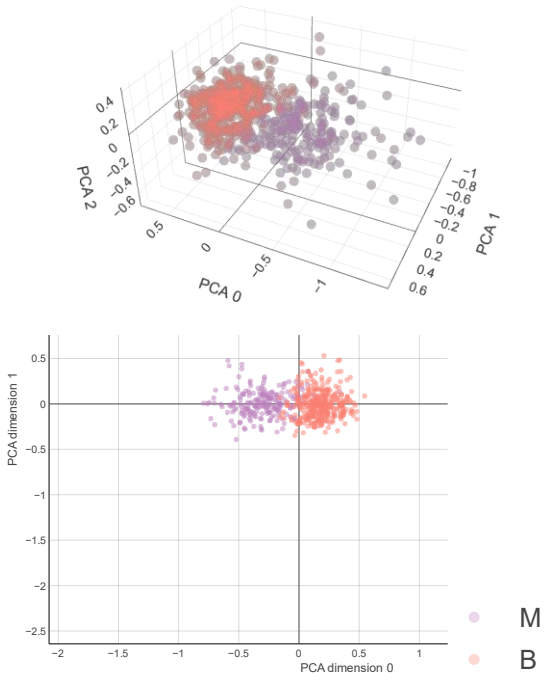

Figure 6 - PCA on dataset

It should be noticed the imbalance presence of records having a benign diagnosis (B) caused by their higher frequency. The *Figure 6* suggested the presence of two spherical clusters with a low inter-cluster distance. For this reason, the Fuzzy C-means algorithm (implemented in the homonymous node in the clustering library of Knime) was chosen. The Fuzzy C-means algorithm belongs to the prototype-based family and it assigns to each record a membership probability that can be useful for further analysis. The "winning" cluster, hence the assigned one, for a certain record, is the cluster with the highest probability. It was decided to use *C = 2*, for the 14-features and 4-features clustering. The prototype approach permits to clearly define the optimal representative records for each cluster, hence its characteristics which are useful to identify differences between records having a malignant or benign diagnosis. Furthermore, the similar K-means behaviour, fits well with the pseudo-spherical shape of the clusters.

Evaluation

It is clear the presence of an external structure to use to evaluate the performances of the clustering algorithm (the already assigned *diagnosis* label). The following external evaluation metrics were used:

- $Rand = \frac{a+d}{M}$
- $Jaccard = \frac{a}{a+b+c}$
- $Fowlkes\&Mallows\ (FM) = \sqrt{\left(\frac{a}{a+b}\right) * \left(\frac{a}{a+c}\right)}$

a = records in the same cluster and equally labelled

b = records in the cluster and not equally labelled

c = records in different clusters and equally labelled

d = records in different clusters and not equally labelled

M = number of all the possible pairs, given n the number of records:

$$M = \frac{n(n-1)}{2}$$

The following table shows the distribution of the labels (M for malignant and B for benign), across the different approaches and clusters, and the three evaluation metrics:

	ID	B%	M%	Rand	Jaccard	FM
14 feat.	0	93.26%	6.74%	0.881	0.801	0.89
	1	5.56%	94.44%			
4 feat.	0	6%	94%	0.881	0.801	0.89
	1	93.5%	6.5%			

Table 7 - Clustering performances 14 or 4 features

The two approaches showed the same performances by rounding to the second decimal number, hence the clustering on D2 shall be identified as the best option because of its reduced dimensions. In the case of the 14-features clustering, the cluster number zero represents well the records having a B diagnosis, while the cluster number one contains most of records with M diagnosis. For the 4-features clustering, the relations are inverted, so cluster zero contains most of the M diagnosis. In general, just by using the four features called *texture_worst, area_worst, smoothness_worst* and *concave_point_worst,* the clustering analysis obtains satisfactory results. It is not a case if the most relevant features, chosen by exploiting the feature selection classifiers, are all related to the *worst* measure of a certain medical dimension. The following table shows the centroids (computed through averages wit exception for *diagnosis*) identified by the 4-features clustering:

	Diagnosis (MODE)	texture_worst	area_worst	smoothness_worst	Concave_points_worst
0	M	0.463	0.463	0.509	0.651
1	B	0.311	0.311	0.347	0.255

Table 8 - Centroids with 4 features

The values of the M records are always the highest (the values are still shown under normalization). It was noticed a region of space, between the two main spherical objects, characterised by a mixed texture of records having M and B diagnoses. For this reason, a clustering based on *C=3* and 4 features was applied on the hypothesis of presence of an additional cluster exactly between the two more "extreme" clusters already identified. By "extreme" is intended the fact that each cluster is well characterised by just one label of the *diagnosis* target feature. The silhouette score was used as internal evaluation measure (the external structure could not be used with three clusters):

$$i-th\ record\ Silhouette = s_i = \frac{b_i - a_i}{\max(b_i, a_i)}$$

$$Silhouette_{overall} = \frac{\sum_{i=1}^{n} s_i}{\mathrm{n}}$$

$a_i = mean\ intra - cluster\ distance$

$b_i = mean\ inter - cluster\ distance\ to\ the\ closest\ cluster$

The following table shows the results of this last approach:

ID	B%	M%	Silhouette
0	0%	100%	0.236
1	77.71%	22.29%	
2	98.22%	1.78%	

Table 9 - Performances with 3 clusters

The "cluster in the middle", number 1, has a higher number of B diagnosis, but is the most balanced that was ever recorded in the entire clustering process. It should be noticed the strong extremization of cluster 0 and 1 to their respective diagnosis trend (100% of M records for cluster 0 and 98.22% of B records for cluster 2). The silhouette score was computed for the clustering with 4 features and two clusters too: 0.425. It is approximately twice higher than the one associated with three clusters because strongly influenced by the low inter-cluster distance, hence it can't prove the goodness of the last presented model. The following table shows its centroids:

	Diagnosis MODE	texture_worst	area_worst	smoothness_worst	Concave_points_worst
0	M	0.471	0.328	0.519	0.680
1	B	0.385	0.123	0.420	0.370
2	B	0.266	0.089	0.304	0.197

Table 10 - Centroids for the 3-cluster approach

Naturally, cluster one has values in the middle between the other two extremal clusters, as it shown in *Figure 7*:

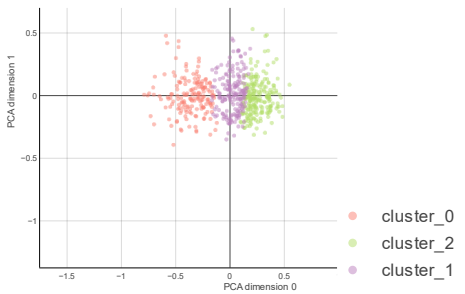

Figure 7 - PCA 3-clusters approach

Despite the negative evaluation obtained from the silhouette score, a second measurement was used to evaluate the model based on three clusters. All the records assigned to the cluster number 1 were removed under the hypothesis that they could introduce a certain kind of noise for the clustering by representing outliers, or borderline cases, of the two main clusters. This time, the external structure was used to compute the following table:

Rand	Jaccard	FM
0.98	0.961	0.98

Table 11 - External measures for the 3-cluster approach

The table contains the best values ever computed among the different approaches, as it could be already noticed by the extremization phenomenon on the structure of M and B diagnoses. In general, the clustering with *C=3* identifies the noise cluster between the two main clusters assigned to malignant and benign records, leading to better performances in terms of external measures. It could be interpreted as containing borderline cases that cannot be totally handled with just two clusters. Furthermore, the silhouette increased too, after the removal of the third cluster, and became the highest one:

Algorithm	Silhouette
2 clusters	0.425
3 clusters	0.236
3 clusters without cluster 1	0.549

Table 12 - Silhouettes

By using the membership probabilities, it was possible to visualise the distribution of probabilities for cluster 0 (M records) and cluster 2 (B records) of objects having cluster 1 as winning cluster (borderline cases). The two distributions were more balanced than in the cases having a winning cluster equal to 0 or 2.

For classifying malignant and benign tumours, the Multilayer perceptron was evaluated as the best overall classification algorithm with highest performances related to the feature selection approach (*acc = 0.981*). The feature selection process, able to shift from 30 to just 4 attributes, is of high importance in order to decrease the number of irrelevant and redundant features. In fact, it leaded to best result amongst all the algorithms. The cost sensitive learning, after the selection of the key features, is capable of well minimizing the cost resulted from misclassification of malignant records as benign ones without leading to too low accuracies or F-measures.

The clustering analysis was strongly connected to the features identified as relevant by the different classification algorithms, as already explained. The Fuzzy C-means clustering with two clusters and just only 4 features represents the best option over the clustering with 14 features. Furthermore, the clustering with 3 clusters successfully identified a third cluster characterised by records neither strongly belonging to the cluster with a M majority nor to the cluster with a B majority, and it was chosen as the optimal solution. Those records are identified as borderline cases that introduced noise in the clustering with just two clusters. In fact, the external and internal measures of the three clusters approach showed its high performances after removing the identified third cluster. The centroids were successfully computed, and the malignant label generally leads to higher value of the used features, hence a good representation of malignant and benign records was obtained. Thanks to the three clusters approach, a third centroid was identified as placed between the two extremes. In general, it could be interpreted as containing borderline cases that can't be totally handled with just two clusters, hence they should be clinically analysed more in depth.

The four features most frequently identified as relevant by the classification algorithms, and then used in the clustering analysis, are always related to the worst measure of *texture, area, smoothness* and *concave_point.*

As a future extension the SMOTE (Synthetic Minority Over-sampling Technique) node could be used to oversample the original data set to enrich the training data, and it might decrease the imbalanced proportion of the *diagnosis* class too. Then, an in-depth analysis of the multilayer perceptron might be useful to identify an optimal number of hidden layers to improve the overall metrics. Finally, distinct types of clustering algorithms, such as density-based or hierarchical algorithms, could be used to improve the cluster analysis process.

Appendix

Conclusion and future extensions

ID	Name
1	diagnosis
2	radius_mean
3	texture_mean
4	perimeter_mean
5	area_mean
6	smoothness_mean
7	compactness_mean
8	concavity_mean
9	concave points_mean
10	symmetry_mean
11	fractal_dimension_mean
12	radius_se
13	texture_se
14	perimeter_se
15	area_se
16	smoothness_se

ID	Name
17	compactness_se
18	concavity_se
19	concave points_se
20	symmetry_se
21	fractal_dimension_se
22	radius_worst
23	texture_worst
24	perimeter_worst
25	area_worst
26	smoothness_worst
27	compactness_worst
28	concavity_worst
29	concave points_worst
30	symmetry_worst
31	fractal_dimension_worst

Table 13 - Dataset features

S Test	D lower_limit	D upper_limit
Feature selection - feauture selection cost sensitive	-0.072	-0.022
Feauture selection - cross validation	-0.046	0.011
Feature selection - cost sensitive	-0.046	0.004
Cross validation - feature selection cost sensitive	-0.07	0.011
Cost sensitive - feature selection cost sensitive	-0.06	0.007
Cross validation - cost sensitive	-0.013	0.006

Table 14 - Logistic error confidence intervals

S Test	D lower_limit	D upper_limit
Feature selection - feauture selection cost sensitive	-0.046	0.022
Feauture selection - cross validation	-0.035	0.01
Feature selection - cost sensitive	-0.02	-0.001
Cross validation - feature selection cost sensitive	-0.017	0.017
Cost sensitive - feature selection cost sensitive	-0.028	0.024
Cross validation - cost sensitive	-0.014	0.018

Table 15- MLP error confidence intervals

Test	lower_limit	upper_limit
Feature selection - feauture selection cost sensitive	-0.04	0.019
Feauture selection - cross validation	-0.034	0.01
Feature selection - cost sensitive	-0.034	-0.005
Cross validation - feature selection cost sensitive	-0.048	0.051
Cost sensitive - feature selection cost sensitive	-0.032	0.049
Cross validation - cost sensitive	-0.017	0.003

Table 16 - J48 error confidence intervals

Test	lower_limit	upper_limit
Feature selection - feauture selection cost sensitive	-0.023	0.019
Feauture selection - cross validation	-0.007	0.011
Feature selection - cost sensitive	-0.007	0.011
Cross validation - feature selection cost sensitive	-0.026	0.019
Cost sensitive - feature selection cost sensitive	-0.029	0.022
Cross validation - cost sensitive	-0.016	0.015

Table 17 - NBTree error confidence intervals

Bibliography

World Health Organization. (2021, 03 26). *World Health Organization*. Retrieved from World Health Organization: https://www.who.int/news-room/fact-sheets/detail/breast-cancer

Kaggle. (n.d.). *Kaggle*. Retrieved from https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset

Mark A. Hall and Lloyd A. Smith, U. o. (1999). Feature Selection For Machine Learning: Comparing a Correlation-based Filter Approach to the Wrapper. *AAAI*.

Shukla, S. Y. (2016). Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. *IEEE 6th International Conference on Advanced Computing (IACC)*.

Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*.

Wiswedel, M. R. (2007). KNIME: The Konstanz Information Miner. *Springer*.