Machine Learning-Based Prediction of Breast Cancer Using the Wisconsin Data Set

1st Víctor Moreno Borràs
ETSETB - Universitat Politècnica de Catalunya
victor.moreno.borras@estudiantat.upc.edu

2nd Alejandro Lozano Gómez
ETSETB - Universitat Politècnica de Catalunya
alejandro.lozano.gomez@estudiantat.upc.edu

***Abstract*—This document introduces the study of developing a predictive system for classifying breast masses as benign or malignant using machine learning techniques. For training, a specific dataset has been utilized, exploring different classification methods, including Gaussian Naive Bayes, SVM, and Neural Networks. The results obtained are very promising and demonstrate significantly good performance.**

I. Introduction

This document presents a work carried out within the framework of the **Big data and programming in R** [1] course. The objective is explore the application of machine learning techniques for the prediction and classification of breast cancer using the Breast Cancer Wisconsin Data Set. All the code for this project can be found at **github** [7].

For training and evaluating the system, a dataset originating from the Wisconsin region, United States, is utilized. The **dataset** [8] comprises information gathered from digitized images of fine needle aspirates (FNA) of breast masses. Each sample is represented by a feature vector, including 30 real-valued attributes for each cell nucleus present in the image. These attributes describe characteristics such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.

The dataset comprises various attributes related to breast cancer diagnosis. Each entry is characterized by an ID number and includes the following features:

- **Diagnosis:** Malignant (M) or Benign (B)
- **Radius:** Mean of distances from the center to points on the perimeter
- **Texture:** Standard deviation of gray-scale values
- **Perimeter:** Perimeter of the tumor
- **Area:** Area of the tumor
- **Smoothness:** Local variation in radius lengths
- **Compactness:** Computed as $\frac{\text{perimeter}^2}{\text{area}} - 1.0$
- **Concavity:** Severity of concave portions of the contour
- **Concave Points:** Number of concave portions of the contour
- **Symmetry:** Symmetry of the tumor
- **Fractal Dimension:** "Coastline approximation" - 1

The mean, standard error, and "worst" (mean of the three largest values) of these features were computed for each image, resulting in a total of 30 features. For instance, field 3 represents Mean Radius, field 13 represents Radius SE, and field 23 represents Worst Radius. All feature values are recorded with four significant digits. There are no missing attribute values in the dataset. The class distribution consists of 357 benign and 212 malignant cases.

II. Proposed Classification Methods

In this section, the classification methods proposed to address the classification problem are described. After trying various methods seen in class, it has been decided to delve into the study of techniques that have yielded more promising initial results.

A. Naive Bayes

The Naive Bayes algorithm is a probabilistic classifier based on Bayes' theorem. In the context of classification, it leverages Bayes' theorem to calculate the conditional probability of each class C given observed features $X_1, X_2, \ldots, X_n$. Naive Bayes assumes conditional independence among features given the class, simplifying the computations. The classification is performed by selecting the class with the highest conditional probability given the observed features. The formula considers prior class probabilities $P(C = c)$ and conditional probabilities of individual features $P(X_i|C = c)$ for each class c. The classification rule can be expressed as:

$$\hat{C} = \arg\max_c P(C = c) \prod_{i=1}^{n} P(X_i|C = c) \tag{1}$$

In our implementation, we employed the Naive Bayes model with kernel density estimation, made available through the **kernlab library** [4][5].

B. Support Vector Machines (SVM)

The primary concept behind SVM is to discover a hyperplane that maximizes the margin between different class data points in the feature space. For binary classification, such as predicting cancer type, the SVM seeks to create a hyperplane with the maximum margin, considering the nearest data points from each class as support vectors.

Mathematically, the SVM decision function can be expressed as:

$$f(x) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) \tag{2}$$

Here, $\mathbf{w}$ represents the weight vector, $\mathbf{x}$ is the input feature vector, and b is the bias term. The function $\text{sign}(\cdot)$ indicates the predicted class based on the sign of the result.

In our case, using six columns containing mean values as features, the SVM classifies new data points by determining their position relative to the hyperplane. A positive output suggests a benign case (B), while a negative output suggests a malignant case (M). Besides, the SVM model utilized for the predictive task has been provided by the **e1071** [3] library.

C. Neural Network

Neural Networks (NN) are foundational models in machine learning, drawing inspiration from the structure and function of the human brain. Comprising layers of interconnected nodes, NNs include an input layer, one or more hidden layers, and an output layer. The connections, or weights, between neurons determine the strength of information transfer.

Mathematically, the output y_j of a neuron j is calculated using the equation:

$$y_j = \sigma\left(\sum_i w_{ij} \cdot x_i + b_j\right) \tag{3}$$

Here, w_{ij} is the weight, x_i is the input, b_j is the bias, and σ is the activation function introducing non-linearity.

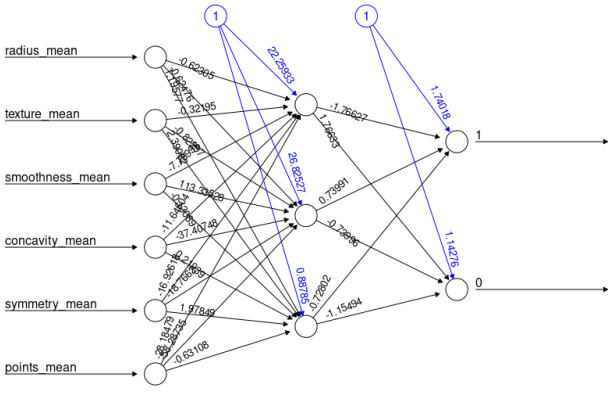

Fig. 1. Classification NN plot

In our specific case, the neural network has 6 inputs, as evident from Figure 1. These five inputs correspond to the values of six columns containing means in the dataset. The network's output is binary, indicating whether the predicted case is benign (B) or malignant (M). Also, the neuronal network model utilized for the predictive task has been provided by the **neuralnet** [6] library.

III. Measures of the Quality of the Solution

In addition to working with the **F1-score** (5), it is important to consider other quality measures. For example, **Recall** (4), also known as sensitivity, is significant. This measure is particularly relevant in the context of clinical data, where it is crucial to adequately detect all cases of disease (diagnosis = M)

$$\textbf{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$\textbf{F-score} = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \tag{5}$$

The Recall measures the proportion of true positive cases (TP) that are correctly detected by the model. In other words, it indicates the system's ability to identify the malignant cases. In this context, having a high Recall is much more important than having a high Precision. If a model has low Recall, it means there is a high number of false negatives, i.e., malignant cases that the model fails to identify. This can have serious consequences for patients, as it leads to missed or delayed opportunities for diagnosis and treatment.

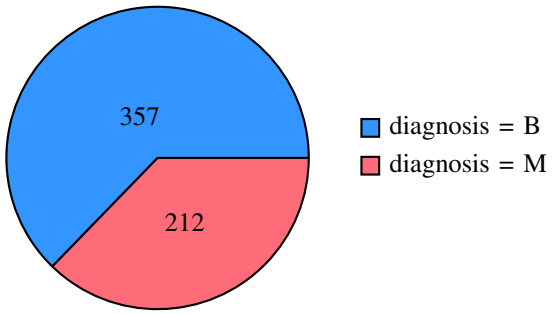

Fig. 2. Dataset class distribution

Additionally, it is important to note that the database used for the predictive system is imbalanced (Figure 2). This imbalance can impact quality measures, as a model may lean towards predicting the majority class to achieve better overall precision. However, this approach may result in a low Recall for the minority class (malignant), which is precisely the class we are interested in identifying correctly.

Therefore, in this task, it is crucial to consider both F1-score and Recall as quality measures. The F1-score provides a balance between Recall and Precision, while Recall gives a direct indication of the system's ability to detect positive cases. Thus, both metrics help evaluate the effectiveness and reliability of the predictive system.

IV. Experiments

A. Data analysis

Prior to model training, an exploratory data analysis was conducted to better understand the relationships between variables. Firstly, the Pearson correlation between variables was calculated. Pearson correlation measures the strength and direction of the linear relationship between two variables. This metric was used to assess the relationship between features in the dataset. Figure 3 displays the Pearson correlation coefficients for pairs of columns.

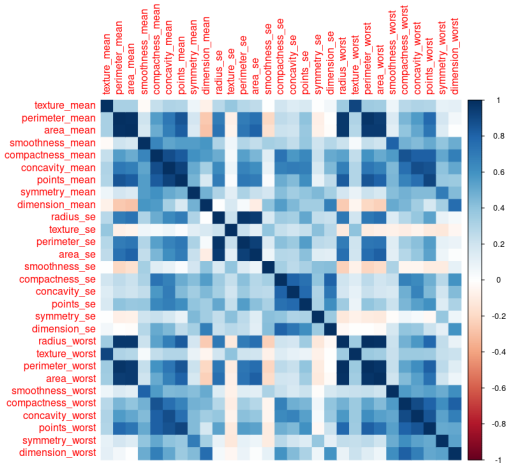

Fig. 3. Pearson correlation between all the features

In addition, detailed visualizations of the correlation analysis can be found in **Appendix B** of this document. Specifically, refer to Figures 5, 6, and 7, where a more granular examination of correlation values between various variables is presented. These figures provide a nuanced exploration of the relationships, and scatter plots are included to facilitate a comprehensive observation of collinearity between variables. Readers are encouraged to refer to the appendix for a more in-depth understanding of the inter-variable associations depicted in the plots.

B. Feature Selection

Based on the insights gained from the data analysis, a variable selection process was undertaken to identify the most representative features within the dataset. Specifically, pairs of variables with a Pearson correlation coefficient exceeding 0.9 were examined. Variables that exhibited higher correlation coefficients with multiple others were deemed to carry more redundant information, prompting the decision to retain them for model training while discarding the remaining variables. For instance, as illustrated in Table 1, "radius mean" displayed a Pearson correlation coefficient exceeding 0.9 with eight other variables. Consequently, "radius mean" was chosen as the representative variable, and the others were excluded from the model training process.

TABLE I

Variable 1	Variable 2	Correlation
perimeter_mean	radius_mean	1.00
area_mean	radius_mean	0.99
radius_worst	radius_mean	0.97
perimeter_worst	radius_mean	0.97
area_worst	radius_mean	0.94

Ultimately, the decision has been made to retain the following six variables as representative features for training the models:

- **radius mean**
- **texture mean**
- **smoothness mean**
- **concavity mean**
- **concave points mean**
- **symmetry mean**

C. Best Models

Prior to the exhaustive study of a few models, a test has been conducted with a broad set of models to identify those that offer better initial results. This test includes models covered in theory and in the laboratory. Using the performance metrics seen in the previous section, the best models have been selected for a more detailed analysis. It is important to note that this initial test does not guarantee the suitability of the selected models, but it provides an initial direction for further study.

D. Partition

For the training and testing of various models, a random partitioning of the dataset was conducted. As illustrated in Figure 4, an 80-20 split was employed, with 80 % of the data allocated for training purposes and the remaining 20 % reserved for testing the models. This approach ensures a robust evaluation of model performance, allowing for an unbiased assessment of their generalization capabilities on unseen data. The figure visually encapsulates the distribution of instances across the training and testing sets.

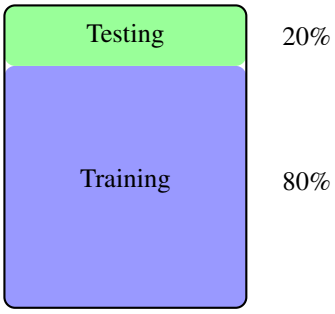

Fig. 4. Database partition

V. Results

In this section, we will first present the results of each model under various hyperparameter configurations. Subsequently, a comparison will be conducted, showcasing the best-performing instances of each model. It is important to note that all these findings are derived from the evaluation on the test dataset.

In the case of Naive Bayes, it has been observed that variations in hyperparameter settings do not significantly impact the model's performance. Naive Bayes classifiers, being relatively simple and robust, often exhibit stable behavior across different configurations. On the other hand, for the Support Vector Machine (SVM) classifier, a more extensive exploration of hyperparameters has been conducted. Specifically, the SVM has been evaluated under three distinct kernel functions: vanilladot, rbfdot, and polydot. The results can be observed in the **Table II**. Also, various configurations of hidden layers in the Neural Network model have been explored to discern the impact on performance. Following a systematic evaluation, it has been determined that the optimal configuration entails a Neural Network with a single hidden layer comprising three neurons.

	Precision	**Recall**	**F1-score**
SVM-vanilladot	0.9649	0.9729	0.9689
SVM-rbfdot	0.9737	0.9594	0.9665
SVM-polydot	0.9649	0.9729	0.9689

TABLE II

	Precision	**Recall**	**F1-score**
Naive Bayes	0.9473	1	0.9729
Support Vector Machine	0.9649	0.9729	0.9665
Neuronal Network	0.956	1	0.9776

TABLE III

The **Table III** provides a comprehensive overview of the performance metrics for the prediction system. In the context of breast cancer detection, Recall is of paramount importance as it measures the system's ability to correctly identify all patients with malignant tumors. A high Recall is crucial to avoid false negatives, ensuring that patients with breast cancer are accurately detected. Turning attention to the table, the results demonstrate strong performance across all three models, with high Precision, Recall, and F1-score values. Notably, the Neuronal Network model achieves a perfect Recall, as well as Naive Bayes model, but having the best Precision value. Based on these results, it can be inferred that the Neural Network emerges as the preferred model, even though all three models, exhibit commendable performance.

VI. Conclusions

The study presented in this document focuses on the development of a predictive system for classifying breast masses as benign or malignant using machine learning techniques.

The results obtained from the evaluation of these models are promising, showcasing significantly good performance. Notably, the Neural Network model emerged as the best choice, achieving high Precision, Recall, and F1-score values. While the Naive Bayes model also exhibited strong performance, the Neural Network demonstrated a slight better Recall and Precision values, highlighting its ability to correctly identify all patients with malignant tumors. The Support Vector Machine model, evaluated under different kernel functions, showed consistent and commendable results as well.

In conclusion, the findings of this study suggest that the application of machine learning techniques, particularly the Neural Network model, holds great potential for accurately classifying breast masses as benign or malignant. The overall performance of the models, with emphasis on the Neural Network, indicates a promising avenue for enhancing diagnostic capabilities in the field of breast cancer detection.

APPENDIX

A. Code

In the following code snippets, the initial steps showcase the data partitioning process. Subsequently, the code proceeds to define and instantiate different machine learning models. For further insights into the code and a more detailed understanding of the implementation, please refer to the GitHub repository linked **here** [7].

```
set.seed(123)
indices_entrenamiento <- sample(1:nrow(breast_cancer_data), 0.8 * nrow(breast_cancer_data))
breast_cancer_train <- breast_cancer_data[indices_entrenamiento, ]
breast_cancer_test <- breast_cancer_data[-indices_entrenamiento, ]
```

Listing 1. Dataset partition

```
nb_model <- naiveBayes(breast_cancer_train_features, breast_cancer_train_labels, laplace = 1)
svm_model <- ksvm(diagnosis~radius_mean+texture_mean+smoothness_mean+concavity_mean+symmetry_mean+points_mean,data=breast_cancer_train,kernel=k)
nn_model<-neuralnet(diagnosis~radius_mean+texture_mean+smoothness_mean+concavity_mean+symmetry_mean+points_mean,data=breast_cancer_train,hidden = 3)
```

Listing 2. Models instantiations

B. Correlation-Scatter Plots

In the code 3, utilizes the chart.Correlation function from the **PerformanceAnalytics package** [2] to create scatter plots and correlation matrices for three distinct subsets of variables in the dataset. The first line generates scatter plots and correlations for variables representing means (columns 1-10). The second line focuses on variables representing standard deviations (columns 11-20). The third line specifically examines a subset of variables (columns 21-30) in a worst-case scenario context. In each case, the scatter plots use a dark grey color for points and include histograms along the diagonal, providing a visual exploration of relationships and correlations between variables.

```
# scatter plot between means
chart.Correlation(X[,c(1:10)], histogram=TRUE, col="grey10", pch=1)

# scatter plot between standard deviations
chart.Correlation(X[,c(11:20)], histogram=TRUE, col="grey10", pch=1)

# Worst-case scatterplot
chart.Correlation(X[,c(21:30)], histogram=TRUE, col="grey10", pch=1)
```

Listing 3. Correlation and Scatter Plots Code

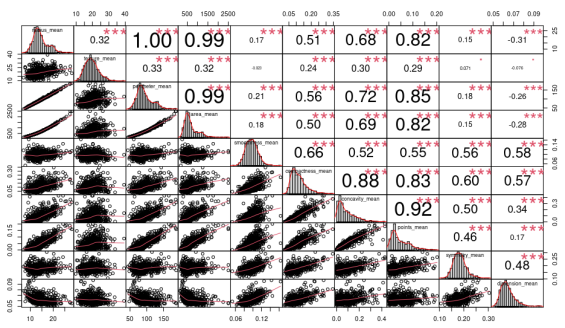

Fig. 5. Pearson correlation between mean features

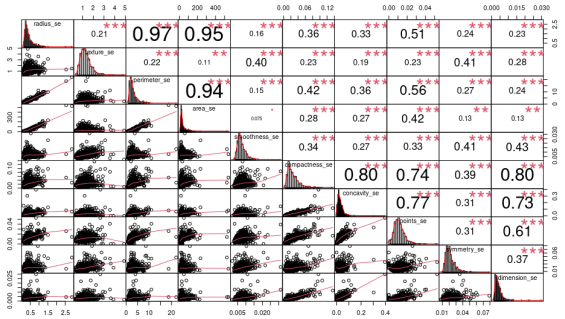

Fig. 6. Pearson correlation between std features

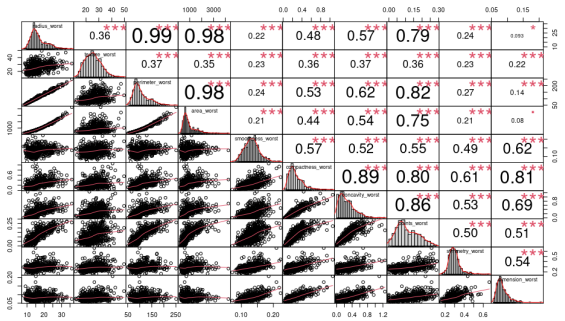

Fig. 7. Pearson correlation between worst features

REFERENCES

[1] Josep Maria ArocaYoutube channel bdr, 2023. BDR YouTube Channel.
[2] Peter Carl Brian G. PetersonPerformanceanalytics: Econometric tools for performance and risk analysis, 2004. R package version 3.5.0.
[3] Kurt Hornik ORCID Andreas Weingessel Friedrich Leisch David Meyer ORCID, Evgenia Dimitriadoue1071: Misc functions of the department of statistics, probability theory group (formerly: E1071), tu wien, 2023. R package version 1.7-14.
[4] Alexandros Karatzoglou, Alex Smola, and Kurt Hornikkernlab: Kernel-based machine learning lab, 2023. R package version 0.9-32.
[5] Alexandros Karatzoglou, Alex Smola, Kurt Hornik, and Achim Zeileiskernlab – an S4 package for kernel methods in R, 2004.
[6] Marvin N. Wright Stefan Fritsch, Frauke Guentherneuralnet: Training of neural networks, 2019. R package version 2.9.0.
[7] Alejandro Lozano Gómez Víctor Moreno BorràsBigdata. https://github.com/victhormoreno/bigdata, 2023.
[8] Mangasarian Olvi Street Nick Wolberg, William and W. StreetBreast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository, 1995. DOI: https://doi.org/10.24432/C5DW2B.