UNIVERSITY OF MILAN BICOCCA
DEPARTMENT OF INFORMATICS, SYSTEMS AND COMMUNICATION

Machine Learning Project

De Mayda, Francesco[1]
Marinucci, Daria[1]
Suardi, Matteo[1]

[1]Master Degree in Data Science

Abstract

Breast cancer is a prevalent disease where early and accurate diagnosis is critical for improving patient outcomes. Machine learning techniques provide an effective approach to classify breast tumors based on medical imaging and histopathological data. This study evaluates five classification models—Support Vector Machines (SVM), Logistic Regression, Random Forest, Decision Trees, and Naïve Bayes—using the Breast Cancer Wisconsin (Diagnostic) dataset. Feature selection identified key variables such as tumor texture, perimeter, smoothness, concavity and symmetry as the most significant predictors. Class imbalance was addressed using the Synthetic Minority Over-sampling Technique (SMOTE), leading to varying effects on model performance. Logistic Regression and Random Forest showed consistent improvement, while SVM experienced a significant decline in accuracy when trained on the balanced dataset. These findings highlight the importance of feature selection and model-specific strategies for handling imbalanced data. While oversampling techniques improve performance in some models, their effectiveness depends on the classifier's sensitivity to synthetic data. Future work should explore alternative balancing methods and hybrid models to optimize classification in medical applications.

Keywords: Classification Models, Machine Learning, Breast Cancer, SMOTE

Index

1. Research questions 1
2. Dataset overview 2
3. Data loading and preprocessing 3
4. Feature selection 4
5. Class balancing 6
6. Machine learning models 7
7. Cross-validation and Performance evaluation 12
8. Comparison of classification models and performance metrics different from Accuracy 15
9. Conclusions 18
A. Formulas 21
A. References 21

1. Research questions

- What are the most significant variables for defining the optimal classification model for breast cancer?
- Which classification model ensures the best performance in achieving the research objective?

2. Dataset overview

2.1. Introduction

Breast cancer is one of the most prevalent cancers worldwide, where early and accurate diagnosis is crucial for improving patient outcomes. Traditional diagnostic methods, based on medical imaging and histopathology, can be time-consuming and subjective. Machine learning offers a promising alternative by automating classification and enhancing diagnostic precision. This study explores five ML models—Support Vector Machines (SVM), Logistic Regression, Random Forest, Decision Trees, and Naïve Bayes. Model performance is assessed through classification accuracy, the Receiver Operating Characteristic (ROC) curve, and the Area Under the Curve (AUC). The dataset used in this study is the **Breast Cancer Wisconsin (Diagnostic) Dataset**, sourced from the UCI Machine Learning Repository and available on Kaggle[1].

2.2. Structure and characteristics

The dataset consists of a unique identifier (ID), a target variable (**Diagnosis**) indicating whether a tumor is Malignant (M) or Benign (B), and 30 numerical features derived from digitized breast mass images. These features are categorized into mean values (e.g., `radius_mean`, `texture_mean`), standard errors (e.g., `radius_se`, `texture_se`), and worst-case values (e.g., `radius_worst`, `texture_worst`).

The dataset has no missing values to handle, ensuring consistency in preprocessing.

[1] https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

3. Data loading and preprocessing

3.1. Descriptive statistics

Descriptive statistics were computed using the **Statistics** metanode to provide a comprehensive overview of the dataset, summarizing key metrics for each feature. This analysis was essential for understanding data distribution and identifying potential issues such as skewness, outliers, or varying feature scales.

The statistical summary included minimum, maximum, mean, and standard deviation values for each numerical feature, along with variance, skewness, and kurtosis to assess the distribution properties of the variables. Since no missing values were present, consistency in data quality was ensured.

Histogram and box plot visualizations helped to identify data patterns, revealing that some features contained extreme values, while others, such as `radius_mean` and `area_mean`, exhibited a wide range of values, reinforcing the need for feature normalization.

This exploratory analysis justified the application of normalization techniques to enhance machine learning model performance. Before applying transformations, the `ID` column was filtered out as it does not contribute to classification, ensuring that only relevant features were used in subsequent preprocessing and modeling.

3.2. Feature transformation and normalization

3.2.1. Logarithmic and square root transformations

To mitigate the effect of skewed distributions and outliers, we applied specific transformations:

- **Logarithmic transformation:** applied to features with high variance and right-skewed distributions, defined as:

$$x' = \log(x), \quad x > 0 \tag{1}$$

- **Square root transformation:** used for variables containing zero values to avoid undefined logarithmic operations:

$$x' = \sqrt{x}, \quad x \geq 0 \tag{2}$$

The square root transformation was specifically applied to the following features due to the presence of zero values: `concavity_worst`, `concavity_mean`, `concave_points_mean`, `concavity_se`, and `concave_points_se`.

3.2.2. Normalization

After transformations, we applied **Min-Max normalization**, scaling each feature to the range [0,1]:

$$x' = \frac{x - x_{\text{mín}}}{x_{\text{máx}} - x_{\text{mín}}} \tag{3}$$

This method preserved feature distributions while ensuring comparability across different scales.

By combining transformation and Min-Max normalization, we achieved a more stable and well-distributed dataset, essential for optimizing model performance and convergence during training.

4. Feature selection

4.1. Pearson correlation computation

To reduce multicollinearity and improve model interpretability, we computed the **Pearson correlation coefficient**[1] ($r_{X,Y}$) among numerical features.

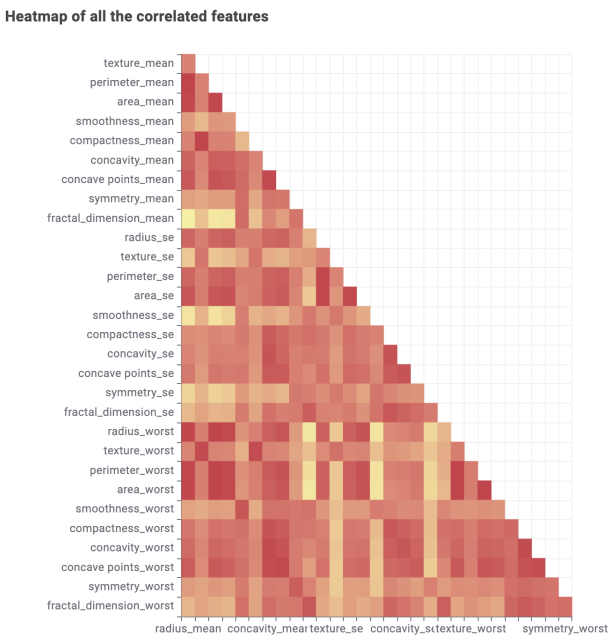

Figura 1: Heatmap visualization of all the correlated numerical features

4.2. Feature filtering

A **correlation filter** with a threshold of 0.8 was applied to eliminate highly correlated features, ensuring that redundant information was removed to enhance model stability. This step was particularly crucial for **logistic regression**, where multicollinearity could lead to unreliable coefficient estimates. By setting the threshold at 0.8, the analysis retained only the most independent features, improving interpretability and reducing the risk of overfitting.

The process was implemented using a structured approach. First, the **Linear Correlation** node was configured to compute Pearson correlation for all numerical features, excluding the categorical variable `diagnosis`. The **Correlation Filter** node was then applied to automatically remove features exceeding the predefined 0.8 threshold, ensuring that only independent variables were retained for model training.

Finally, A **heatmap visualization** was employed to assess correlation patterns, providing insights into feature dependencies.

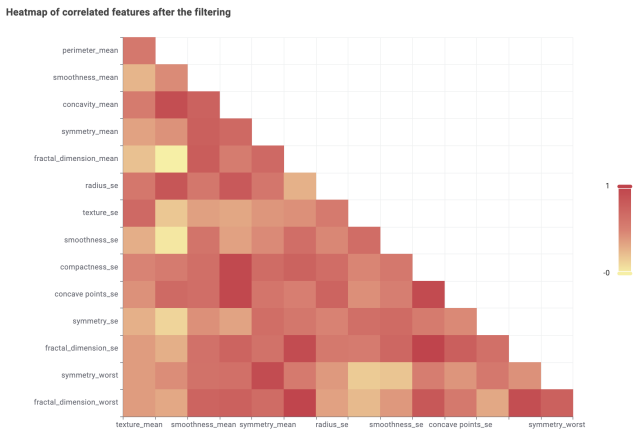

Figura 2: Heatmap visualization of the correlated features after filtering

4.3. Impact of dimensionality reduction on machine learning models

Logistic regression relies on computing feature weights (β) through the **maximum likelihood estimation (MLE)**[2], which determines the optimal parameters that maximize the probability of observing the given data. However, when features exhibit high correlation, the estimation of β becomes unstable, leading to inflated standard errors, unreliable coefficient estimates, and potentially misleading predictions.

To address this issue, **correlation-based feature selection** was applied to enhance model efficiency by reducing dimensionality while preserving the most informative features. By eliminating redundant variables, this approach not only mitigated multicollinearity but also improved computational performance, reduced overfitting risks, and ensured more stable and interpretable model coefficients. This step was particularly crucial for logistic regression, where model stability directly impacts classification accuracy and generalization to unseen data.

Beyond logistic regression, high feature correlation also had a significant impact on **Support Vector Machines (SVMs)**. SVMs rely on maximizing the margin between classes by projecting data into a higher-dimensional space. When redundant features were present, the model's abi-

lity to define an optimal hyperplane was compromised, as correlated variables introduced noise and unnecessary complexity to the decision boundary. This could lead to reduced generalization performance, increased model sensitivity to variations in training data, and longer computational times due to higher-dimensional input spaces. By applying correlation-based feature selection, we ensured that only the most relevant features contributed to the classification process, thereby improving the robustness and efficiency of SVMs.

In **Random Forests**, dimensionality reduction affects both training efficiency and model interpretability. Since random forests construct multiple decision trees based on a random subset of features, the presence of highly correlated or redundant features can lead to unnecessary complexity in tree construction. By reducing dimensionality, we ensured that trees were trained on more informative and independent variables, leading to a more efficient model with reduced training time and potentially improved generalization performance. However, due to the inherent feature selection mechanism in random forests—where only a subset of features is used at each split—the model is generally more robust to high-dimensional input spaces compared to other classifiers.

For **Decision Trees**, high-dimensional input spaces can lead to overfitting, as the model attempts to create increasingly complex splits to accommodate every feature. This results in overly specific decision boundaries that fail to generalize well to unseen data. By applying feature selection, we reduced the likelihood of deep, overly complex trees and promoted simpler, more interpretable models. Additionally, fewer features led to more meaningful splits, improving the model's ability to capture the true underlying structure of the data while preventing fragmentation due to unnecessary variables.

In the case of **Naïve Bayes**, dimensionality reduction had a mixed impact. Since naïve Bayes assumes conditional independence among features, the presence of correlated variables contradicts this assumption, leading to skewed probability estimations. By removing redundant features, we improved the validity of the independence assumption, thereby enhancing the model's probabilistic estimates and classification performance. However, because naïve Bayes is inherently computationally efficient and does not suffer from overfitting in the same way as tree-based or margin-based models, the benefits of dimensionality reduction were primarily theoretical rather than computational.

Overall, dimensionality reduction was a critical preprocessing step that not only streamlined computation but also reinforced model reliability across multiple classifiers. While its necessity varied by algorithm, its application consistently led to improvements in stability, interpretability, and generalization across different machine learning models.

5. Class balancing

We observed a slight imbalance in the absolute frequencies of the target variable `diagnosis`: the dataset contained **212 (37 %)** observations belonging to the **M** class (malignant tumor) and **357 (63 %)** observations belonging to the **B** class (benign tumor), as seen in figure 3:

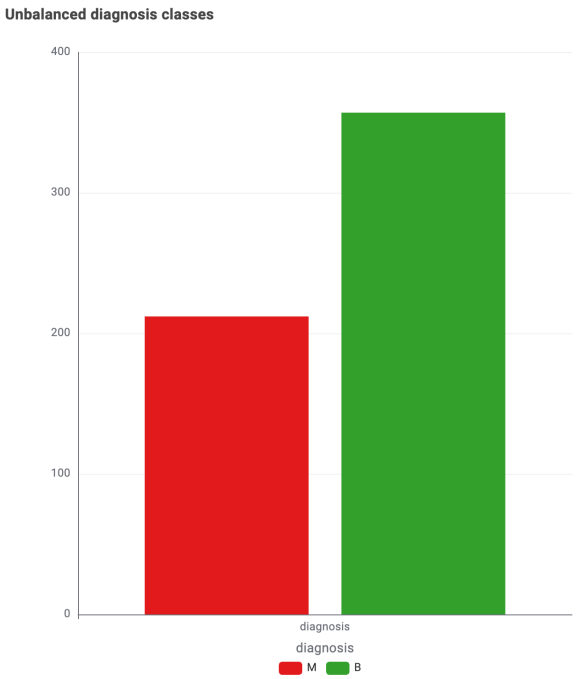

Figura 3: Distribution of unbalanced classes

To address this issue, we adopted a dual approach: the same classification models were trained both on the imbalanced dataset and on a balanced dataset obtained through the **SMOTE (Synthetic Minority Over-sampling Technique)** method.

The SMOTE node was configured with `diagnosis` as the reference class column, applying **oversampling to the minority class** (`diagnosis` = "M"). The synthetic instances were generated by selecting a given instance of the minority class and computing new samples along the vector connecting it to one of its k-nearest neighbors (in this case, $k = 5$). The newly synthesized data points are interpolated between the selected minority class sample and its neighbors rather than simple duplication, thus increasing diversity and mitigating overfitting risks.

By implementing SMOTE, we ensured that the dataset contained an equal representation of the two classes, achieving a **50 %-50 % class distribution** (figure 4). This approach allowed us to systematically compare the classification models' performance under two different training conditions: **with the original imbalanced dataset (63 %-37 %) and with the SMOTE-balanced dataset (50 %-50 %)**.

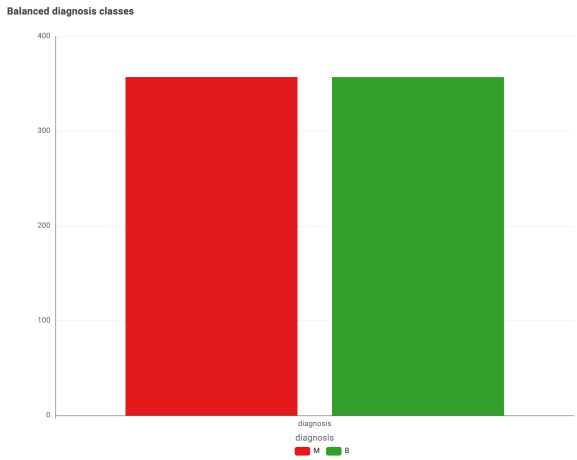

Figura 4: Distribution of classes after applying SMOTE (balanced)

6. Machine learning models

6.1. Data partitioning and cross-validation

The dataset was partitioned into **training and test sets** using the **X-Partitioner** node, configured to implement a robust validation strategy.
To ensure a reliable assessment of model performance, **10-fold cross-validation** was employed. In this approach, the dataset was randomly divided into 10 equally sized subsets (folds): at each iteration, one fold was used as the test set while the remaining nine served as the training set. This process was repeated 10 times, allowing each instance to be used for both training and validation, thereby reducing the variance of performance estimates and mitigating overfitting.
To prevent potential biases introduced by data ordering, **random sampling** was applied when forming the folds. This technique ensured that each subset maintained a representative class distribution, particularly important for a slightly imbalanced dataset such as this one.
For reproducibility, a **fixed random seed** (n=123) was set. This guaranteed that the dataset partitioning remained consistent across multiple runs, enabling a fair comparison between different models.
Following this partitioning strategy, each machine learning model was iteratively trained on the training subset and evaluated on the corresponding test subset. This systematic approach provided a robust framework for comparing their predictive performance while minimizing the risk of overfitting to specific data patterns.

6.2. Models implemented

To classify breast cancer as malignant or benign, we implemented and compared five **supervised** machine learning models, categorized based on their underlying principles:

- **Probabilistic models**
  - **Naïve Bayes**: A probabilistic classifier based on Bayes' theorem, assuming conditional independence among features. It computes the posterior probability of each class given the input features and assigns the most probable label. Despite its simplicity, Naïve Bayes is computationally efficient and performs well in high-dimensional datasets.
- **Heuristic models**
  - **Decision Tree**: A rule-based model that recursively splits the dataset into subgroups based on feature thresholds, creating an interpretable hierarchical structure. The decision tree follows a greedy, top-down approach to minimize impurity at each split.
  - **Random Forest**: An ensemble learning method that constructs multiple decision trees and aggregates their predictions through majority voting. By leveraging bootstrap aggregation (bagging) and random feature selection, random forests enhance robustness, reduce variance, and mitigate overfitting.
- **Regression-based models**
  - **Logistic Regression**: A linear classification model that estimates the probability of a class using the sigmoid activation function[3].
    Logistic regression assumes a linear decision boundary and is highly interpretable, making it a strong baseline model.
- **Kernel-based models**
  - **Support Vector Machine (SVM):** A model that identifies the optimal hyperplane that maximizes the margin between two classes. In cases where the data is not linearly separable, kernel functions are used to project the input space into a higher-dimensional feature space where a linear separation is possible. The decision function is governed by:

    $$f(X) = \sum_{i=1}^{n} \alpha_i y_i K(X_i, X) + b$$

    where $K(X_i, X)$ represents the kernel function and α_i are the learned coefficients.

Each model was trained and evaluated under the same experimental conditions to ensure a fair comparison of their predictive performance.

6.3. Logistic Regression

The **Logistic Regression Learner** node was configured to use the `diagnosis` column as the target variable, setting `M` (malignant) as the reference category.
The **Stochastic Average Gradient (SAG)** optimizer was selected for its efficiency in handling large datasets and ensuring faster convergence. Feature selection involved the exclusion of the `binary_diagnosis` column to prevent redundancy.
To ensure stable learning, hyperparameters were tuned with a maximum of 17,000 epochs to guarantee sufficient iterations for convergence, an epsilon of 10^{-5} as a stopping criterion for optimization, and a fixed learning rate of 0.1 to control weight updates. A Bayesian regularization prior was applied with a variance of 0.1 to mitigate overfitting, ensuring that coefficient estimates remained stable. Regularization followed the principle

$$\beta_i \sim N(0, \sigma^2)$$

where σ^2 controls the regularization strength. The **Logistic Regression Predictor** node was configured to generate predictions with a custom column name `Prediction (diagnosis)`, while also enabling probability output to include predicted class probabilities.
Logistic regression was chosen for its strong interpretability and efficiency in binary classification. The implementation of stratified cross-validation improved model robustness, while the SAG optimizer accelerated convergence. The inclusion of regularization effectively mitigated overfitting, ensuring better generalization across unseen data.

The classification performance of the logistic regression model was evaluated the different conditions described in section 5: using the original imbalanced dataset and a balanced dataset obtained through the Synthetic Minority Oversampling Technique (SMOTE). The accuracy of the model on the imbalanced dataset was found to be **0.968**, whereas balancing the class distribution via SMOTE led to a slight improvement, increasing accuracy to **0.972**.

Additionally, the Area Under the Curve (AUC) metric, which quantifies the model's ability to distinguish between malignant and benign cases across all possible classification thresholds, also exhibited a marginal enhancement. Specifically, the AUC value was **0.995** for the imbalanced dataset and improved to **0.997** when training was performed on the SMOTE-balanced dataset. While this increase is minimal, it suggests a slight gain in the model's discriminative power when class imbalance is mitigated.

These results indicate that balancing the dataset using SMOTE contributes to a modest but measurable improvement in classification performance. The higher accuracy and AUC suggest that the model benefits from a more uniform representation of the classes, potentially reducing bias toward the majority class. However, given the marginal nature of the improvement, the impact of SMOTE in this specific case should be interpreted with caution, and additional evaluation metrics should be considered to ensure that synthetic data generation does not introduce artifacts or overfitting.

6.4. Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees and aggregates their outputs to enhance predictive accuracy and reduce overfitting. The final prediction is determined by majority voting in classification problems. Mathematically, for a given input X, the Random Forest prediction is defined as

$$\hat{Y} = \arg\max_{c} \sum_{t=1}^{T} I(h_t(X) = c)$$

where T represents the total number of trees in the forest, $h_t(X)$ is the prediction made by the t-th decision tree, and $I(\cdot)$ is an indicator function that returns 1 if the tree predicts class c, otherwise 0. The final class $\hat{Y}$ is the one with the highest number of votes across all trees.

The **Random Forest Learner** node was configured with `diagnosis` as the target column, while the `binary_diagnosis` column was excluded to prevent redundancy. The model used the **Information Gain Ratio** as the split criterion, which normalizes entropy-based information gain to improve decision-making and prevent bias toward features with many values. The hyperparameters were set with 100 trees, striking a balance between computational efficiency and accuracy. Tree depth was left unrestricted, allowing trees to grow until reaching pure leaves, while the minimum node size was set to 1, ensuring that each leaf contained

at least one instance. The Information Gain Ratio was computed as

$$IG_{\text{ratio}}(X) = \frac{H(Y) - H(Y|X)}{H(X)}$$

where $H(Y)$ is the entropy of the target variable, $H(Y|X)$ is the entropy of Y given feature X, and $H(X)$ represents the entropy of feature X, acting as a normalization term.

The **Random Forest Predictor** node was configured to generate predictions with a custom column name `Prediction (diagnosis)` while also appending overall prediction confidence and probability scores for both benign and malignant classes. Random Forest was selected due to its capability to handle non-linear relationships, robustness against noise and missing values, and its reduced risk of overfitting compared to single decision trees. The use of 100 trees and the Information Gain Ratio ensured strong predictive performance while maintaining model interpretability.

The classification performance of the model was evaluated under the same conditions as the other models. The accuracy of the Random Forest model on the imbalanced dataset was found to be **0.949**, while applying SMOTE to balance the class distribution led to an improvement, increasing accuracy to **0.972**. This trend is consistent with the results observed for logistic regression, where SMOTE also contributed to a slight increase in accuracy.

Furthermore, the Area Under the Curve (AUC) metric exhibited a minor but notable enhancement. Specifically, the AUC value for the imbalanced dataset was **0.991**, whereas balancing the dataset using SMOTE resulted in an improved AUC of **0.995**, matching the value obtained with logistic regression on the balanced dataset. While the improvement in AUC is marginal, it suggests that balancing the dataset may contribute to a slight enhancement in the model's ability to distinguish between malignant and benign cases.

6.5. Decision Tree

A Decision Tree is a hierarchical model that recursively splits data based on feature values to classify instances. Each internal node represents a decision rule, while each leaf node corresponds to a class label. The model constructs splits to maximize information gain, which is defined as

$$IG(Y, X) = H(Y) - H(Y|X)$$

where $H(Y)$ is the entropy of the target variable and $H(Y|X)$ represents the conditional entropy of Y given feature X. Entropy itself is computed as

$$H(Y) = -\sum_{i=1}^{k} P(y_i) \log_2 P(y_i)$$

where $P(y_i)$ denotes the probability of class y_i in the dataset.

The **Decision Tree Learner** node was trained with `diagnosis` as the target column and the Gain Ratio as the splitting criterion. This approach normalizes information gain, mitigating biases toward attributes with multiple values. Reduced Error Pruning was enabled to simplify the tree and prevent overfitting. The minimum number of records per node was set to 2, ensuring that each node contained at least two instances to avoid excessive branching. Additionally, the model used an average split point strategy, improving threshold selection for continuous variables. The Gain Ratio, which refines information gain by accounting for feature entropy, is given by

$$GainRatio(X) = \frac{IG(Y, X)}{H(X)}$$

where $H(X)$ is the entropy of feature X.

To handle missing values, the model adopted a strategy where the last known prediction was used to prevent information loss. Furthermore, for instances where no clear decision could be made, null predictions were assigned instead of forcing a classification.

The **Decision Tree Predictor** node was configured to output predicted class labels under the column `Prediction (diagnosis)` while appending normalized class probabilities to quantify classification confidence. Decision Trees were selected for their interpretability and their capability to capture non-linear relationships. The use of Gain Ratio as the splitting criterion minimized bias, while pruning techniques helped control overfitting, making the model more robust and generalizable.

The accuracy of the Decision Tree model on the imbalanced dataset was found to be **0.91**, whereas applying SMOTE to balance the class distribution resulted in a minimum performance increase, raising accuracy to **0.913**. This pattern aligns with the improvements observed in logistic regression and random forest, where balancing the dataset

led to moderate but consistent accuracy enhancements.

In addition to accuracy, the Area Under the Curve (AUC) metric also exhibited a noticeable improvement. The AUC for the imbalanced dataset was **0.903**, increasing to **0.944** when training was performed on the SMOTE-balanced dataset. This improvement suggests that the Decision Tree model benefits from synthetic oversampling, although its overall performance remains lower than that of logistic regression and random forest.

These findings emphasize the importance of model selection when dealing with imbalanced datasets. While SMOTE improves performance across all models, its impact varies depending on the model's ability to generalize from synthetic samples. Further investigations, including hyperparameter tuning and pruning techniques, may enhance the Decision Tree's predictive power while mitigating overfitting.

6.6. Naïve Bayes

Naïve Bayes is a probabilistic classifier based on Bayes' theorem, assuming that features are conditionally independent given the class. The model calculates the probability of a sample belonging to class C_k given the feature vector $X = (X_1, X_2, ..., X_n)$ as:

$$P(C_k|X) = \frac{P(C_k)\prod_{i=1}^{n} P(X_i|C_k)}{P(X)}$$

where $P(C_k|X)$ is the posterior probability of class C_k given X, $P(C_k)$ is the prior probability of class C_k, $P(X_i|C_k)$ is the likelihood, i.e., the probability of feature X_i given class C_k, and $P(X)$ is the marginal probability of the feature vector X. Since $P(X)$ is constant across classes, classification is performed by selecting the class with the highest posterior probability.

The **Naïve Bayes Learner** node was configured with the classification column set to `diagnosis`, a default probability of 0.0001 to ensure non-zero probabilities for unseen feature values (Laplace smoothing), a minimum standard deviation of 0.0001 to prevent division by zero in likelihood calculations, and a threshold standard deviation of 0, meaning no additional thresholding was applied. The likelihood $P(X_i|C_k)$ was modeled using a Gaussian distribution:

$$P(X_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(X_i - \mu_k)^2}{2\sigma_k^2}\right)$$

where μ_k and σ_k^2 are the mean and variance of feature X_i for class C_k.

The **Naïve Bayes Predictor** node was configured to predict class labels, storing them in the column `Prediction (diagnosis)`, and to append class probability distributions, providing confidence scores for each classification. Naïve Bayes was chosen for its computational efficiency and suitability for high-dimensional data. Despite its strong independence assumption, it often performs well in medical classification tasks due to the predominance of probabilistic relationships. The use of Laplace smoothing mitigated issues with unseen values, while the Gaussian assumption enabled effective handling of continuous data.

Unlike the previously analyzed models, the application of SMOTE did not lead to an improvement in accuracy for Naïve Bayes. The model achieved an accuracy of **0.931** on the imbalanced dataset, but this value slightly decreased to **0.922** when trained on the SMOTE-balanced dataset.

Also, the AUC decreased from **0.981** in the imbalanced setting to **0.90** in the balanced scenario. This negligible decrease suggests that the SMOTE has a limited impact on the overall discriminative power of the Naïve Bayes classifier, as it does not significantly degrade its ability to differentiate between classes.

When compared to the other models, Naïve Bayes exhibits a distinct behavior. This outcome may be attributed to the fundamental assumption of **feature independence** in Naïve Bayes, which can be disrupted by the synthetic data introduced through SMOTE. Since SMOTE generates new samples by interpolating between existing data points, it may inadvertently alter feature distributions in a way that violates the independence assumption, thereby affecting the model's predictive performance.

These results highlight the importance of considering model-specific characteristics when addressing class imbalance. While SMOTE has proven beneficial for models that rely on decision boundaries (such as logistic regression and random forest), its impact on probabilistic models like Naïve Bayes can be less predictable. Further investigation into alternative balancing techniques, such

as class-weight adjustments or different resampling strategies, may provide better insights into optimizing Naïve Bayes for imbalanced datasets.

6.7. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm that finds the optimal hyperplane to separate classes in a high-dimensional space. Given a training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, SVM solves the following optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N} \xi_i$$

subject to:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

where $\mathbf{w}$ is the weight vector that defines the hyperplane, b is the bias term, ξ_i are the slack variables that allow misclassification, and C is the penalty parameter that balances the width of the margin and classification errors.

The **SVM Learner node** was configured with the class column set to `diagnosis` and a polynomial kernel function defined as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j + \text{bias})^{\text{power}}$$

where power was set at 1, specifying a linear decision boundary, bias was set to 1 to add flexibility to the hyperplane, and gamma was set to 1 to control the influence of individual samples. The overlapping penalty (C parameter) was set to 1.0, ensuring a balance between maximizing the margin and minimizing misclassification.

The **SVM Predictor** node was configured to predict class labels, storing them in the column `Prediction (diagnosis)`, and to append normalized class probabilities, enabling uncertainty estimation in classification. SVM was chosen for its ability to handle high-dimensional data and its robustness in finding an optimal decision boundary. The power 1 polynomial kernel effectively acted as a linear classifier, ensuring computational efficiency while maintaining the flexibility of the model. The penalty parameter C was fine-tuned to balance the bias-variance trade-off.

In contrast to other classifiers, the application of SMOTE had a **profoundly negative impact** on the performance of SVM. The accuracy of the model on the imbalanced dataset was remarkably high at **0.97**, but it dramatically dropped to **0.500** when training on the SMOTE-balanced dataset.

A similar pattern was observed for the Area Under the Curve (AUC). When trained on the imbalanced dataset, SVM achieved an AUC of **0.995**, demonstrating exceptional ability to distinguish between malignant and benign cases. However, when trained on the SMOTE-balanced dataset, the AUC suffered a drastic decline, dropping to **0.797**. This substantial deterioration in both accuracy and AUC suggests that the SVM model is particularly sensitive to the synthetic data generated through SMOTE.

The sharp decline in performance can be attributed to the nature of SVM, which relies on finding an optimal hyperplane that maximizes the margin between classes. SMOTE artificially generates new samples by interpolating between minority class instances, which can distort the original data distribution and introduce synthetic points that do not accurately represent real patterns in the feature space. This can lead to a degradation in the SVM decision boundary, causing severe misclassification. The most extreme consequence is the accuracy drop to **0.500**, which indicates that the model is essentially **making random predictions** in the balanced scenario.

Compared to the other classifiers, SVM exhibited the most severe adverse effect when trained on the SMOTE-balanced dataset. While logistic regression, random forest, and decision trees benefited from SMOTE (with moderate accuracy improvements), and Naïve Bayes experienced only a slight decrease, SVM suffered a catastrophic drop in classification performance. This highlights the limitations of synthetic oversampling for models that rely heavily on well-defined margins and support vectors.

These findings underscore the importance of choosing appropriate class imbalance handling techniques based on the underlying classifier. In the case of SVM, alternative strategies such as cost-sensitive learning, adjusting class weights, or **using different kernel functions** may be more effective in addressing imbalance while preserving the model's generalization ability. Further investigation into the specific structural impact of SMOTE on SVM decision boundaries is warranted to better understand the underlying cause of performance degradation.

7. Cross-validation and Performance evaluation

To assess the performance of the machine learning models and ensure their robustness in predicting breast cancer diagnoses, the cross-validation technique was implemented using `X-Aggregator` nodes, which aggregate the predictions from each model and compare them with the actual values of the target variable. Key performance metrics such as **accuracy** and the **ROC curve** with **AUC** were computed to evaluate model effectiveness. As seen in subsection 6.1, a 10-fold cross-validation approach was adopted to reduce overfitting and obtain a reliable estimate of the models' generalization capability: the dataset was divided into ten subsets, where each subset was used once as a test set while the remaining nine were used for training. The `X-Partitioner` node ensured proper data partitioning for all models. Each `X-Aggregator` node was configured with the target column set to `diagnosis`, the prediction column set to the column containing each model's predictions (e.g., `Prediction (diagnosis)`), and no fold `ID` column to simplify the aggregation of performance across all folds.

7.1. Accuracy calculation

To assess the performance of each model, the accuracy metric is computed as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

where:

- TP (True Positives): Number of correctly classified positive instances.
- TN (True Negatives): Number of correctly classified negative instances.
- FP (False Positives): Number of negative instances misclassified as positive.
- FN (False Negatives): Number of positive instances misclassified as negative.

Each model's predictions are compared against the actual class labels using the **Scorer** node. Only the accuracy value is extracted from the generated metrics through the **Column Filter** node.

Subsequently, the accuracy metric is renamed according to the model using the **Column Renamer** node (e.g., *Accuracy_LogReg*, *Accuracy_RandFor*, etc.), and then aggregated into a single table using the **Column Appender** node.

A boxplot is then generated to compare the accuracy values of different models and assess their classification performance.

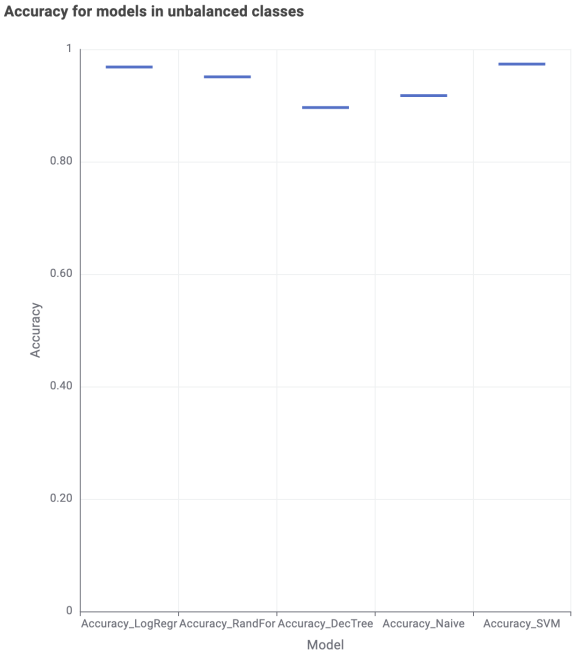

Figura 5: Accuracy for models in unbalanced classes

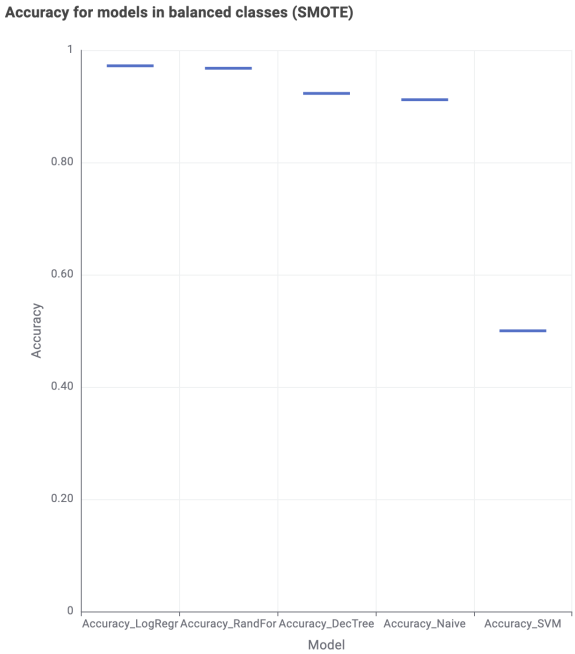

Figura 6: Accuracy for models in balanced classes (SMOTE)

7.2. ROC Curve and AUC calculation

The performance of each classification model was assessed using the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC). The ROC curve is a fundamental tool in machine learning classification, illustrating the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) across various classification thresholds. Formally, these metrics are defined as:

$$\text{TPR} = \frac{TP}{TP + FN}$$

$$\text{FPR} = \frac{FP}{FP + TN}$$

where TP (True Positives) and FN (False Negatives) denote correctly and incorrectly classified positive instances, respectively, while FP (False Positives) and TN (True Negatives) correspond to misclassified and correctly classified negative instances.

The AUC quantifies the classifier's overall ability to distinguish between classes by computing the integral under the ROC curve. AUC values range from 0 to 1, where an AUC of 0.5 represents a random classifier, while an AUC of 1.0 indicates perfect classification performance. A higher AUC implies superior discriminative power, with values closer to 1 reflecting a model's ability to rank positive instances ahead of negative ones.

The ROC curve and AUC values were computed using the *ROC Curve (JavaScript)* node in KNIME. This node generated the ROC curve for each classifier and calculated its corresponding AUC value. To facilitate a structured comparison, the workflow included a series of post-processing steps: the *Row Filter* node extracted only the AUC values, the *Column Renamer* node assigned distinct names to the AUC values corresponding to different models, and multiple *Column Appender* nodes combined the extracted AUC values into a consolidated table.

Given that the dataset exhibited an imbalanced distribution of the target variable, with 212 malignant cases (**37 %**) and 357 benign cases (**63 %**), an additional analysis was conducted to measure model performance under two conditions: training on the original dataset and on a class-balanced dataset generated using the Synthetic Minority Oversampling Technique (SMOTE). The SMOTE algorithm mitigates class imbalance by synthesizing new instances of the minority class along the feature space defined by its k-nearest neighbors, ensuring that the newly created samples preserve structural relationships rather than introducing exact duplicates. This dual evaluation enabled a systematic comparison of the classifiers' performance on both imbalanced and balanced datasets. The graphical representations of both the ROC Curve for balanced and unbalanced classes are shown in fig. 4 and fig. 5.

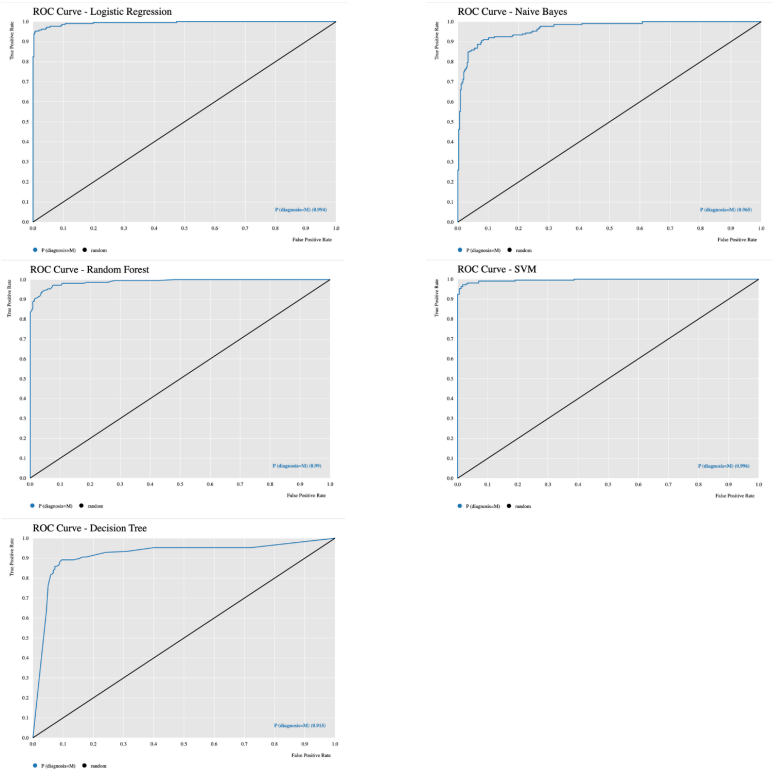

Figura 7: ROC Curve representation for unbalanced classes

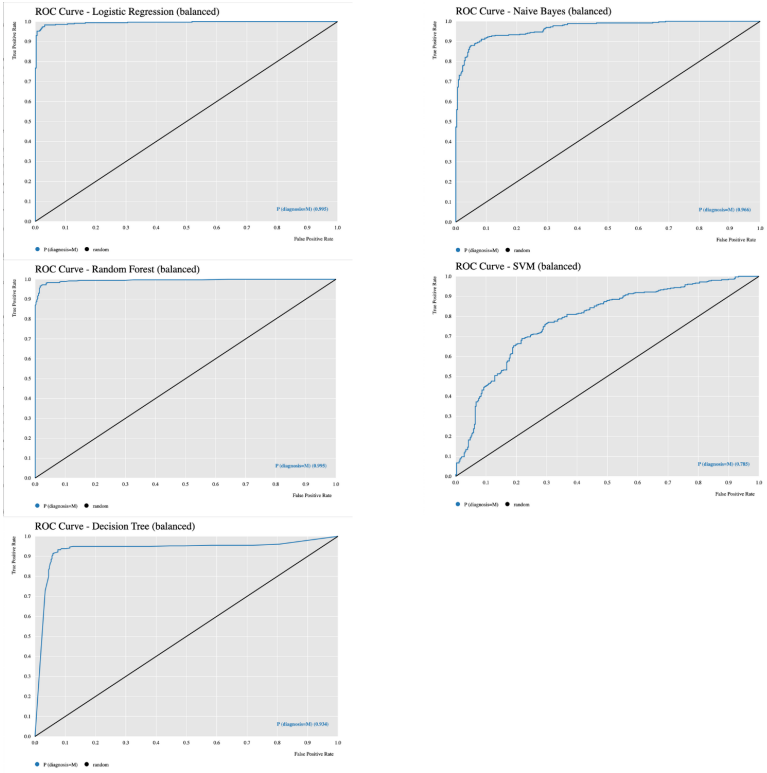

Figura 8: ROC Curve representation after SMOTE balancing operation

8. Comparison of classification models and performance metrics different from Accuracy

This section considers only the class-balanced dataset obtained through SMOTE operation. Particular emphasis was placed on the malignant class (M), given its critical importance in medical decision-making.

8.1. Interpretation of key metrics

Before examining each model, it is helpful to recall the definitions of the fundamental metrics used in this analysis (excluding Accuracy and ROC-related measures, which were already discussed in the previous section):

1. **True Positives (TP), False Positives (FP), True Negatives (TN), False Negatives (FN)**
   These counts describe how the model correctly or incorrectly classifies positive (M) and negative (B) cases. A high TP and a low FN imply strong recognition of malignant cases.

2. **Recall (or Sensitivity)**

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

   This metric indicates the proportion of malignant cases that are correctly identified. A high Recall means fewer false negatives, which is crucial in clinical contexts.

3. **Precision**

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

   Precision measures how often the model is correct when it predicts a case as malignant. A high value implies fewer false positives.

4. **F1-score (F-measure)**

$$\text{F1} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

   This is the harmonic mean of Precision and Recall; higher values indicate a better balance between the two metrics.

5. **Specificity**

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

   Specificity captures the proportion of benign (B) cases correctly classified, thus reflecting the model's ability to avoid false positives.

6. **Cohen's Kappa**

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

   Cohen's Kappa measures agreement between predictions and ground truth, taking into account the possibility of chance agreement. Values close to 1 indicate near-perfect agreement.

8.2. Logistic regression

Focusing on the malignant class (M), Logistic Regression exhibited outstanding Recall, Precision, F1-score, and Specificity, all close to or exceeding 0.95:

- **Recall** ≈ 0.975: identifies about 97.5 % of malignant cases (few false negatives).
- **Precision** ≈ 0.969: rarely misclassifies benign cases as malignant.
- **F1-score** ≈ 0.972: excellent balance between Recall and Precision.
- **Specificity** ≈ 0.969: correctly classifies nearly 97 % of benign cases.
- **Cohen's Kappa**: high value, indicating strong agreement with the true labels.

In conclusion, Logistic Regression on SMOTE-balanced data demonstrated high overall performance, minimizing both false negatives (high Recall) and false positives (high Precision and Specificity).

In clinical terms, this model identifies nearly all malignant tumors while avoiding unnecessary alarms for benign patients. Additional enhancements might include Precision-Recall curves and comparisons with metrics prior to SMOTE balancing.

8.3. Random Forest

Random Forest also achieved strong results, albeit with a slightly different profile than Logistic Regression. For class M:

- **Recall** ≈ 0.915: detects about 91.5 % of malignant cases (lower than Logistic Regression).
- **Precision** ≈ 0.946: maintains a low false positive rate.
- **F1-score** ≈ 0.937: excellent balance between Recall and Precision.
- **Specificity** ≈ 0.987: accurately rules out benign cases (few false alarms).
- **Cohen's Kappa**: around 0.9 or higher, indicating a high level of agreement.

Compared to Logistic Regression:

- Malignant Recall: 0.915 (Random Forest) vs. 0.97 (Logistic Regression).
- Specificity: 0.987 (Random Forest) vs. 0.969 (Logistic Regression).

Hence, if avoiding missed malignant cases is paramount, Logistic Regression may be preferred; if minimizing false positives is more critical, Random Forest is advantageous. Both are high-performing, and the final decision may depend on clinical, cost, or interpretability considerations.

8.4. Decision Tree

Decision Tree models yield decent results but are generally outperformed by Logistic Regression and Random Forest:

- **Recall** $\approx$ 0.877 for malignant cases: lower than the other two.
- **Precision** ≈ 0.882: also slightly lower.
- **F1-score** ≈ 0.879: decent but not excellent.
- **Specificity** ≈ 0.93: classifies 93 % of benign cases correctly.
- **Cohen's Kappa**: indicative of good but not top-tier agreement.

Comparisons:

- Recall: Decision Tree (0.877) <Random Forest (0.915) <Logistic Regression (0.97).
- Specificity: Decision Tree (0.93) <Logistic Regression (0.969) <Random Forest (0.987).

A Decision Tree remains appealing for its *interpretability*, as it is more easily explained to clinicians. However, if the primary goal is to maximize malignant detection and reduce errors, Logistic Regression or Random Forest perform better.

8.5. Naïve Bayes

Naïve Bayes produced reasonably good results but still falls short of Logistic Regression and Random Forest:

- **Recall** $\approx$ 0.90: identifies 90 % of malignant cases.
- **Precision** $\approx$ 0.91: lower than the 0.94–0.97 range of the top models.
- **F1-score** $\approx$ 0.90–0.91: a fair balance.
- **Specificity** $\approx$ 0.95–0.96: decent but not as high as others.
- **Cohen's Kappa**: relatively high but still below the best-performing models.

Naïve Bayes has the advantage of *computational simplicity* and probabilistic interpretability, yet its assumptions of feature independence and sensitivity to distribution shifts (e.g., from SMOTE) may limit its performance. It is still a viable choice for quick, straightforward modeling, though not optimal for maximal sensitivity.

8.6. Support Vector Machine (SVM)

Finally, SVM exhibited *exceptionally high* metrics, in some cases surpassing Logistic Regression and Random Forest:

- **Recall** $\approx$ 0.95–0.96: very few false negatives for malignant cases.
- **Precision** ≈ 0.98: extremely low false positive rate.
- **F1-score** $\approx$ 0.96–0.97: excellent balance between Recall and Precision.
- **Specificity** $\approx$ 0.98–0.99: almost perfect identification of benign cases.
- **Cohen's Kappa**: likely close to 1, indicating near-perfect agreement.

Compared to the other models, SVM tends to lead in Recall and Precision. The main *trade-off* is interpretability: SVMs (particularly with non-linear kernels) are less transparent than linear or tree-based methods. Nonetheless, if maximizing diagnostic accuracy in terms of low false negatives and low false positives is paramount, SVM is a top candidate. Further validation (e.g., larger cross-validation or an external test set) is recommended to confirm robustness.

9. Conclusions

The classification models evaluated in this study demonstrated varying degrees of sensitivity to class imbalance and the application of the Synthetic Minority Over-sampling Technique (SMOTE). While some models exhibited improvements in classification performance when trained on the SMOTE-balanced dataset, others suffered from a degradation in predictive accuracy and discriminative ability. Moreover, feature selection and the identification of the most relevant variables played a crucial role in optimizing model performance, particularly in handling class imbalance and improving model stability.

9.1. Most significant variables for classification

Through Pearson correlation analysis and feature selection techniques, the most significant variables of the Breast Cancer Wisconsin (Diagnostic) Dataset contributing to classification performance were identified as follows:

- *Texture mean* – Measures the variation in pixel intensity of the mass.
- *Perimeter mean* – Captures the boundary length of the tumor.
- *Smoothness mean* – Measures how uniform the mass boundary is.
- *Concavity mean* – Quantifies the severity of concave portions in the tumor contour.
- *Symmetry mean* – Evaluates the symmetry of the tumor shape.

Highly correlated variables were removed using a correlation filter with a threshold of 0.8 to mitigate multicollinearity, particularly for models like logistic regression, where high correlation could distort coefficient estimations. The final set of selected features (as shown in the scatter plot matrix in figure 9) significantly influenced classification performance by ensuring that only the most informative and independent variables were retained.

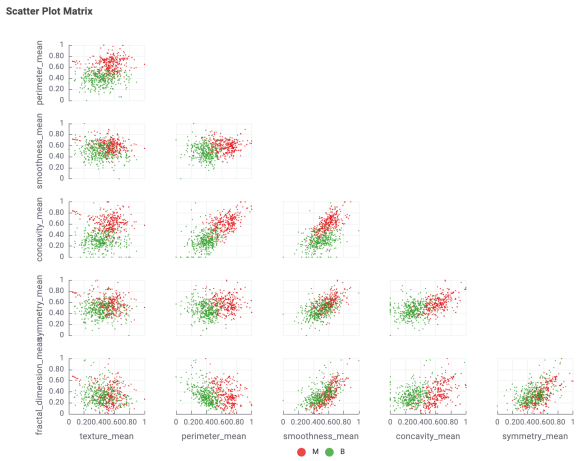

Figura 9: Scatter plot matrix of the most significant variables

9.2. Conclusions

The evaluation of the classifiers revealed that **Logistic Regression** and **SVM** achieved the highest overall performance, as both models exhibited very high recall and precision in detecting malignant cases. Logistic Regression demonstrated a slight improvement in accuracy—from 0.968 to 0.972—and a marginal increase in AUC—from 0.995 to 0.997—when trained on the SMOTE-balanced dataset. In contrast, SVM experienced a dramatic decline, with accuracy dropping from 0.97 to 0.500 and AUC falling from 0.996 to 0.785, an outcome attributable to the disruption of the support vector structure by synthetic data interpolation.

Random Forest also performed competitively, although its recall was slightly lower than that of Logistic Regression; it achieved superior specificity and improved accuracy (from 0.949 to 0.972) and AUC (from 0.991 to 0.995), indicating that class balancing enhanced its robustness without significantly distorting decision boundaries. **Decision Trees**, while offering greater interpretability, underperformed in terms of recall and precision; they benefitted modestly from SMOTE—improving in accuracy from 0.91 to 0.913 and AUC from 0.903 to 0.944—yet remained less effective than ensemble-based models.

Naïve Bayes, recognized for its computational efficiency and simplicity, exhibited a slight decline in performance, with accuracy decreasing from 0.931 to 0.922 and AUC marginally reducing from 0.981 to 0.98. This behavior suggested that the assumption of feature independence may have been compromised by the synthetic data introduced through SMOTE.

In clinical contexts, the choice of the final model depended on the trade-off between recall and specificity, interpretability requirements, and computational constraints. Overall, the metrics considered—namely recall, precision, F1-score, specificity, and Cohen's Kappa—confirmed that Logistic Regression and, despite its sensitivity to SMOTE, SVM achieved superior performance, followed closely by Random Forest, while Decision Trees and Naïve Bayes, although ranking slightly lower, remained viable in scenarios where interpretability or computational simplicity were prioritized.

Unbalanced classes					
	LogReg	**RandFor**	**DecTree**	**NB**	**SVM**
Accuracy	0,968	0,949	0,91	0,931	0,97
AUC	0,995	0,991	0,903	0,981	0,995
Balanced classes (SMOTE)					
	LogReg	**RandFor**	**DecTree**	**NB**	**SVM**
Accuracy	0,972	0,972	0,913	0,922	0,5
AUC	0,997	0,995	0,944	0,98	0,797

Figura 10: Metrics comparison

9.3. Implications of feature selection and class balancing on model selection

The results indicate that while SMOTE is generally beneficial for certain classifiers, its effectiveness is model-dependent. Models that construct flexible decision boundaries, such as random forest and decision trees, demonstrated an improved ability to generalize when trained on the SMOTE-balanced dataset. Conversely, models that rely on strict probabilistic assumptions (e.g., naïve Bayes) or hyperplane optimization (e.g., SVM) exhibited performance degradation, suggesting that synthetic oversampling may introduce inconsistencies in feature distributions.

Furthermore, feature selection played a crucial role in enhancing classification accuracy across all models. Removing redundant variables not only improved interpretability but also enhanced model efficiency by reducing computational complexity and mitigating multicollinearity. This was particularly relevant for logistic regression, where high correlation among variables could lead to unstable coefficient estimates.

These findings underscore the importance of selecting appropriate techniques for handling class imbalance based on the underlying classifier. While SMOTE remains a widely used approach, alternative strategies such as cost-sensitive learning, class-weight adjustments, and ensemble methods may provide more reliable performance enhancements, particularly for models like SVM, where synthetic sample generation disrupts decision boundaries. Additionally, integrating dimensionality reduction techniques such as principal component analysis (PCA) could further optimize feature selection by eliminating irrelevant information while retaining the most predictive attributes.

9.4. Future work

Further investigations are necessary to explore alternative balancing techniques, particularly

for models where SMOTE negatively impacts performance. A promising direction involves the use of **adaptive synthetic sampling (ADASYN)**, which focuses on generating synthetic instances in regions of higher data density, potentially mitigating the adverse effects observed in SVM. Additionally, cost-sensitive learning approaches that penalize misclassification of minority class instances differently may provide a more effective strategy without altering the original data distribution.

Moreover, analyzing the effect of feature selection and dimensionality reduction techniques such as **PCA** could provide deeper insights into the structural changes induced by SMOTE and their impact on different classifiers. Finally, a comparative evaluation of hybrid models, which integrate probabilistic and decision boundary-based approaches, may offer a more robust framework for handling imbalanced medical datasets.

9.5. Final conclusions

This study presented a comprehensive evaluation of five machine learning classifiers applied to the Breast Cancer Wisconsin (Diagnostic) Dataset. The analysis underscored the pivotal role of feature selection and class balancing in enhancing model performance for diagnostic applications.

Key findings indicate that Logistic Regression and Random Forest achieved the highest performance metrics, demonstrating superior recall, precision, and AUC values on both imbalanced and SMOTE-balanced datasets. In contrast, Decision Trees and Naïve Bayes, while offering greater interpretability and computational efficiency, exhibited slightly lower predictive capabilities. Notably, the Support Vector Machine (SVM) classifier proved to be particularly sensitive to synthetic oversampling, with its performance deteriorating markedly under SMOTE—a result that suggests the need for alternative imbalance-handling strategies for kernel-based methods.

These outcomes highlight the necessity of aligning model selection with specific clinical requirements, carefully balancing the trade-offs between diagnostic accuracy, interpretability, and computational cost. Future work should explore alternative resampling techniques, such as ADASYN, and integrate dimensionality reduction methods to further enhance model robustness and clinical applicability. Ultimately, the integration of these strategies is expected to contribute to the development of more reliable and interpretable diagnostic tools, thereby improving patient outcomes in breast cancer detection.

A. Formulas

- [1] **Pearson's correlation coefficient**:

$$r_{X,Y} = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2}\sqrt{\sum(Y_i - \bar{Y})^2}} \quad (5)$$

  where:

  - X_i, Y_i are individual observations of the variables X and Y,
  - $\bar{X}, \bar{Y}$ are the mean values of X and Y.

- [2] **Maximum Likelihood Estimator (MLE)**:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n)}} \quad (6)$$

  where:

  - $P(Y = 1|X)$ is the probability of class 1 given input features,
  - β_0 is the intercept, and $\beta_1, ..., \beta_n$ are the feature weights.

- [3] **Sigmoid activation function**:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n)}}$$

  - where β_i are the feature coefficients learned through maximum likelihood estimation (MLE).

A. References

- Wolberg, William, et al. *Breast Cancer Wisconsin (Diagnostic).* UCI Machine Learning Repository, 1993, https://doi.org/10.24432/C5DW2B.

- Sidey-Gibbons, Jenni A. M. and Chris J. Sidey-Gibbons. *Machine learning in medicine: a practical introduction.* BMC Medical Research Methodology 19 (2019)

- Sharma, Ayush et al. *Machine Learning Approaches for Cancer Detection.* International Journal of Engineering and Manufacturing 8 (2018): 45-55.

- Parandehgheibi, Marzieh. *Probabilistic Classification using Fuzzy Support Vector Machines.* ArXiv abs/1304.3345 (2013): n. pag.