A Repeated Nested Cross-Validation Framework for Breast Cancer Classification

Maria Defteraiou
Machine Learning in Computational Biology

May 9, 2025

Abstract

This report presents a machine learning framework for classifying breast cancer tumors as benign or malignant using measurements derived from fine needle aspirates (FNA) of breast masses. The dataset comprises 512 samples and 30 numerical features, with moderate class imbalance. To ensure robust and unbiased model evaluation, we implemented a repeated nested cross-validation (rnCV) pipeline with 10 repetitions of 5-fold outer and 3-fold inner loops. Six classification algorithms were compared—Logistic Regression, Gaussian Naive Bayes, Linear Discriminant Analysis, Support Vector Machines, Random Forest, and LightGBM—each undergoing hyperparameter tuning within the inner loop. Model performance was evaluated using multiple metrics appropriate for imbalanced data, including Matthews Correlation Coefficient (MCC), AUC, F1 score, and Precision-Recall AUC. Median-based performance was assessed using bootstrap resampling with 95% confidence intervals to support statistically grounded model selection. Logistic Regression with Elastic Net regularization emerged as the winning model, offering the best balance of predictive accuracy, stability, and interpretability. This work highlights the importance of rigorous validation practices and statistically informed comparison in medical ML applications, where reliability and transparency are critical.

1 Introduction

This report addresses the classification of breast cancer tumors as benign or malignant using a dataset derived from digitized images of fine needle aspirates (FNA) of breast masses. The data set contains 512 samples characterized by 30 numerical features that describe the properties of cell nuclei. The primary objective is to identify the best performing classification algorithm for this binary task by implementing a robust object-oriented repeated nested cross-validation pipeline (rnCV).

Unlike simple validation strategies, rnCV provides a more reliable estimate of generalization performance by reducing variance and mitigating bias in model selection and evaluation. The pipeline incorporates multiple algorithms—including Logistic Regression, Gaussian Naive Bayes, Linear Discriminant Analysis, Support Vector Machines, Random Forests, and LightGBM—and evaluates them using a comprehensive set of performance metrics suited for imbalanced data, such as AUC, MCC, and F1-score.

Our contributions include the development of an rnCV class from scratch and a detailed comparative analysis of multiple classifiers. This work not only aids in identifying an optimal diagnostic model, but also provides insight into methodological considerations for developing trustworthy ML systems in biomedical contexts.

2 Materials and Methods

2.1 Dataset Description

The dataset consists of measurements derived from digitized images of fine needle aspirates (FNA) of breast masses. It contains 512 samples, each annotated with a binary target variable, diagnosis, indicating whether a tumor is benign (B) or malignant (M). Each sample is represented by 30 numerical features that describe the physical characteristics of the cell nuclei. These features are based on ten properties—such as radius, texture, perimeter, and concavity—each computed in three ways: mean, standard error, and worst value.

2.2 Data Exploration and Preprocessing

No duplicate entries were found, confirming the validity of the dataset. Although missing values were detected, they will be imputed using median values within each cross-validation fold to avoid data leakage. The dataset is moderately imbalanced, with benign cases comprising approximately 62.5% of the total.

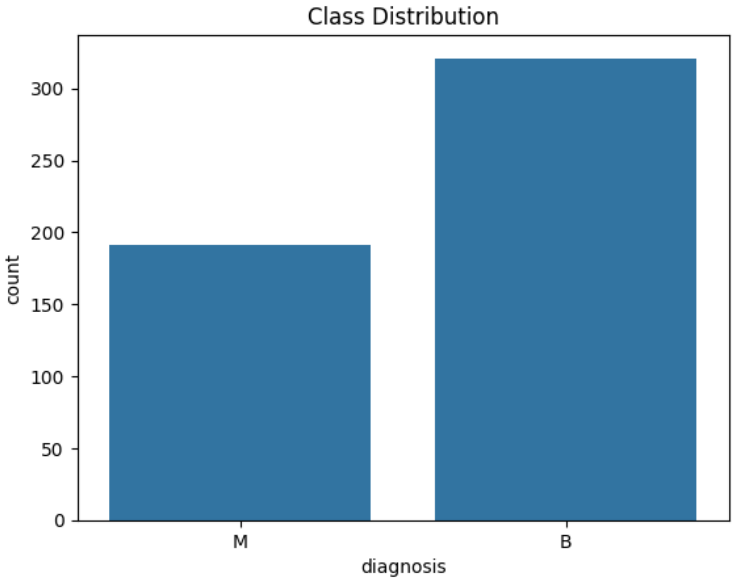

Figure 1: Bar chart showing the class distribution of benign and malignant samples.

2.3 Repeated Nested Cross-Validation Pipeline

Figure 2 illustrates the structure of the repeated nested cross-validation (rnCV) framework implemented in this project. The outer loop, consisting of $k = 5$ folds, is responsible for evaluating the generalization performance of each classifier. For each outer fold, the training set is passed to an inner cross-validation loop ($k = 3$), which performs hyperparameter tuning using grid search.

This entire nested procedure is repeated over 10 rounds with different random splits to ensure performance stability and reduce variance. Each iteration results in a set of performance metrics computed in the outer test fold, providing an unbiased estimate of model performance. Importantly, all preprocessing operations—including imputation and scaling—are performed strictly within each fold to prevent data leakage.

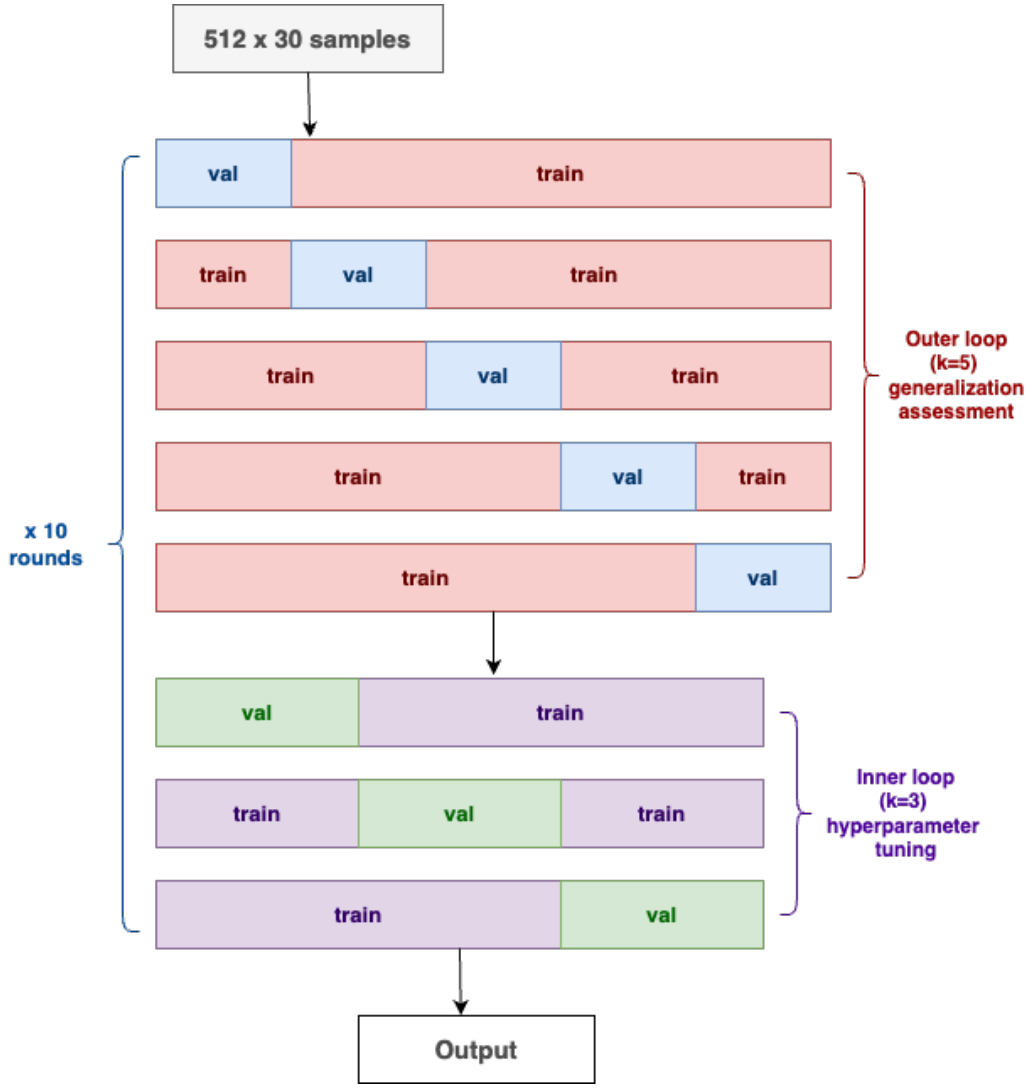

Figure 2: Overview of the repeated nested cross-validation structure used for model evaluation

2.4 Pipeline Stages Description

The classification pipeline was implemented using an object-oriented design, encapsulated in the `nrCV` class, to perform repeated nested cross-validation (rnCV). The pipeline is composed of the following stages, designed to ensure robust model evaluation while preventing data leakage:

1. **Outer Loop (Model Assessment):** The outer loop is a 5-fold cross-validation responsible for estimating the generalization performance of different classifiers. This loop is repeated over 10 rounds to reduce variance and increase robustness.

2. **Inner Loop (Model Selection and Tuning):** For each outer training set, a 3-fold inner cross-validation is performed using `GridSearchCV` to identify the optimal hyperparameters for each estimator based on a chosen metric (e.g., `f1_macro`).

3. **Preprocessing Pipeline:** Within each fold of the inner and outer loops, preprocessing steps are encapsulated in a `Pipeline` to ensure data leakage is avoided:

   - *Imputation:* Missing values are imputed using the median of the training data.
   - *Scaling:* Features are standardized using `StandardScaler` to ensure they are on a comparable scale.
   - *Classification:* The estimator (e.g., SVM, Logistic Regression, Random Forest, etc.) is trained using the best hyperparameters found during tuning.

4. **Metric Computation:** After the final model is trained in the outer fold, predictions are made on the held-out test set. A suite of evaluation metrics is computed to assess performance, including:

   - Area Under the ROC Curve (AUC)
   - Matthews Correlation Coefficient (MCC)
   - Balanced Accuracy
   - F1 and F2 Scores
   - Precision, Recall, and Specificity
   - Negative Predictive Value (NPV)
   - Precision-Recall AUC (PR-AUC)

5. **Results Aggregation:** For each classifier, the outer test fold metrics from all 10 rounds (totaling 50 evaluations) are collected. Means, standard deviations, and full distributions are stored. Bootstrapping is used to estimate 95% confidence intervals around medians for statistical comparison.

6. **Winner Selection:** To identify the best performing classifier, the pipeline first ranks models by the median of a primary metric, the Matthews Correlation Coefficient (MCC), which is particularly suitable for unbalanced data sets. The selection process uses a statistically grounded approach: 95% confidence intervals (CI) around the MCC median are computed

via bootstrap resampling. If the lower bound of the CI of the top model is higher than the upper bound of the second-best model, the winner is declared based on MCC with statistical confidence. If the confidence intervals of the top two models overlap, indicating no clear statistical separation, the process falls back to a secondary metric, the Area Under the ROC Curve (AUC). Among the tied models, the one with the highest AUC median is selected. This two-tier strategy ensures that model selection is based not only on performance but also on statistical rigor, reducing the risk of overfitting to random variation.

This modular and rigorous pipeline ensures that all preprocessing and model selection steps are properly nested, preventing overfitting, and yielding reliable performance estimates for deployment-ready models.

3 Results and Discussion

3.1 Exploratory Data Analysis

To understand the structure and properties of the dataset, in addition to descriptive analytics, visualizations were also provided. To gain insight into the data, several exploratory analyses were performed.

Figure 3 shows boxplots for all 30 features. Features like `area_mean`, `area_worst`, and `perimeter_worst` show a wide range of values and contain numerous outliers, while others (e.g., `fractal_dimension_se`) are tightly distributed. This variation supports the need for feature scaling prior to model training. `StandardScaler` will be applied within each cross-validation fold.

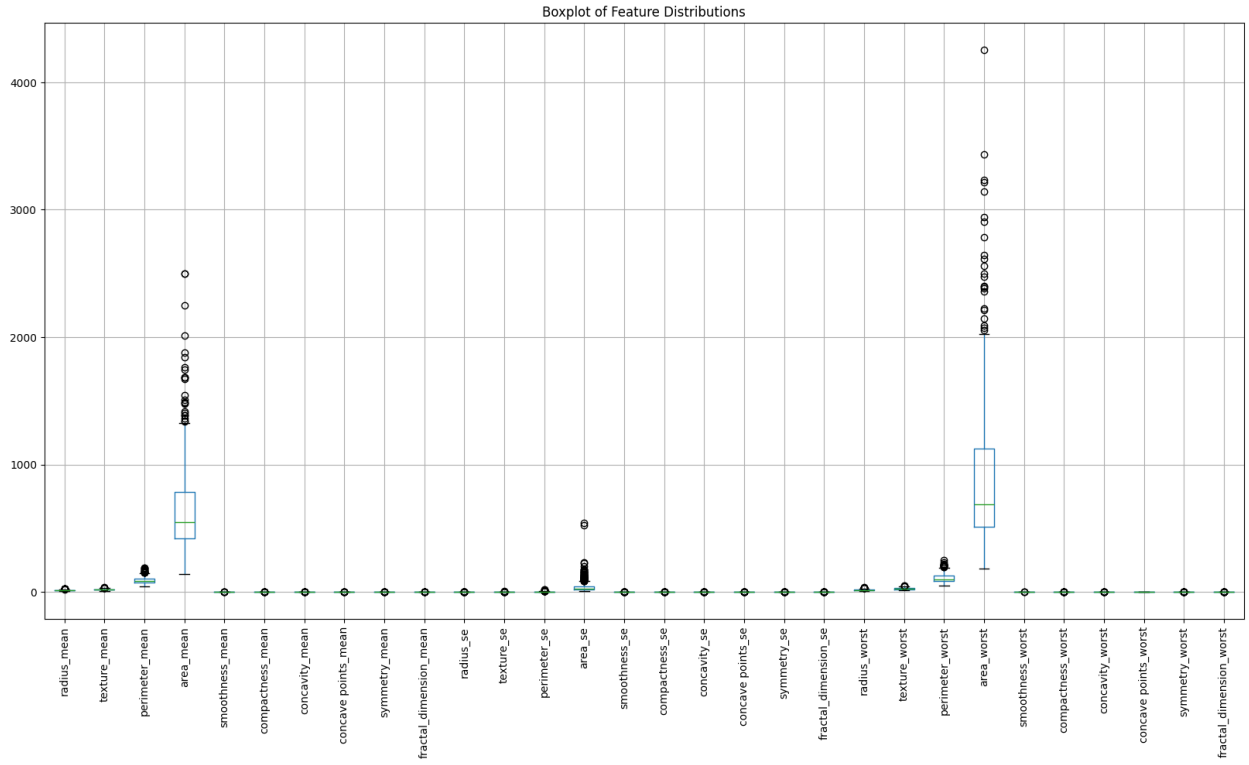

Figure 3: Boxplot of feature distributions. Several features exhibit extreme values and outliers.

Figure 4 presents a Pearson correlation heatmap, revealing strong correlations among several features. For example, `radius_mean`, `perimeter_mean`, and `area_mean` are highly correlated, suggesting possible feature redundancy, which may inform future dimensionality reduction or feature selection.

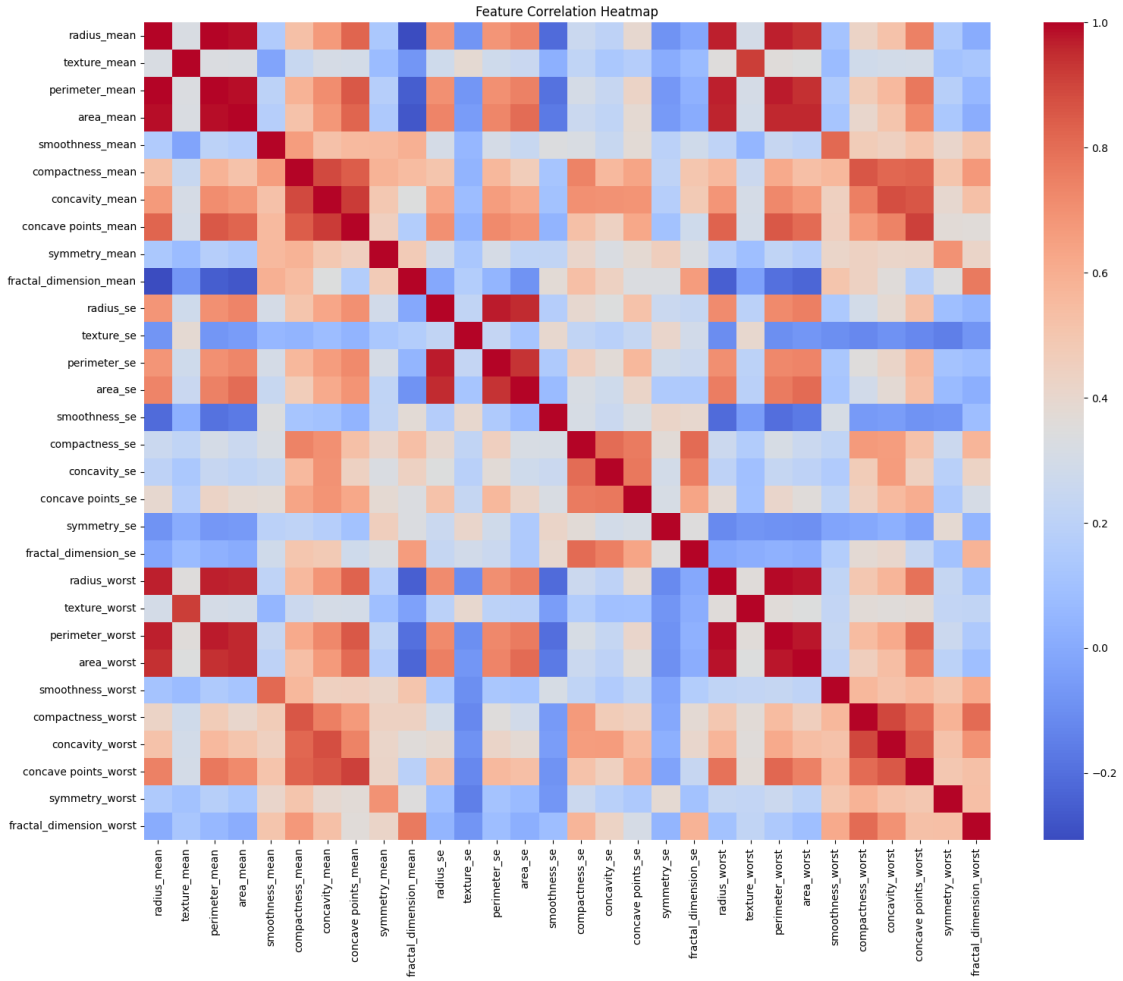

Figure 4: Heatmap of Pearson correlation coefficients among the 30 features.

To visualize the separability of the class, Figure 5 shows the first two principal components obtained by principal component analysis (PCA). Although benign and malignant samples are somewhat separated, overlap remains, highlighting the need for nonlinear or complex classifiers.

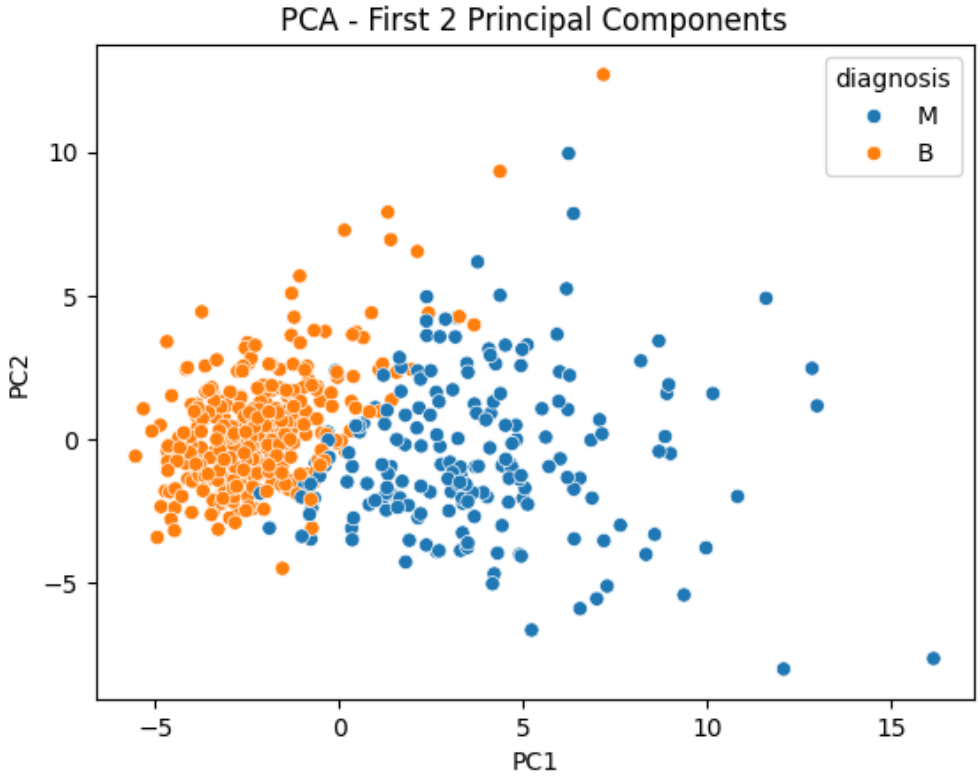

Figure 5: PCA plot of the first two components, colored by diagnosis.

All preprocessing operations, including scaling and imputation, are performed within each fold of the repeated nested cross-validation pipeline (rnCV) to ensure realistic evaluation and prevent data leakage.

3.2 Algorithm Comparison

Six machine learning classifiers were evaluated to determine the most suitable model to classify breast cancer tumors as benign or malignant: Logistic Regression with Elastic Net regularization, Gaussian Naive Bayes, Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), Random Forest, and LightGBM. All models were assessed using the repeated nested cross-validation pipeline (rnCV) described above, with hyperparameter tuning conducted in the inner loop using grid search.

To ensure a reliable comparison, the performance of each model was evaluated in 50 outer test folds (5-fold CV $\times$ 10 rounds). Key classification metrics, including MCC, AUC, F1, recall, and specificity, were calculated for each fold and aggregated using the median. To add statistical integrity, bootstrap resampling was used, where confidence intervals (CI) of 95% was calculated

around each median using 1000 resamples. This allowed for a principled winner selection based on both performance and uncertainty. The results are shown in Table 1 below.

Table 1: Median performance metrics for each model.

Model	**MCC**	**Balanced Accuracy**	**F1**	**F2**	**Recall**	**Precision**	**AUC**
GaussianNB	0.859	0.924	0.928	0.926	0.924	0.931	0.987
LDA	0.901	0.939	0.948	0.942	0.939	0.963	0.994
LightGBM	0.918	0.956	0.959	0.957	0.956	0.958	0.994
LogisticRegression	**0.940**	**0.969**	**0.970**	**0.970**	**0.969**	**0.977**	**0.996**
RandomForest	0.910	0.950	0.955	0.950	0.950	0.954	0.990
SVM	0.937	0.967	0.969	0.968	0.967	0.971	0.995

The final winner was selected using a two-stage decision process. First, the models were ranked by their median MCC, a robust metric for imbalanced datasets. If the lower CI limit of the top model exceeded the upper CI limit of the second best model, it was declared the winner. Otherwise, the decision defaulted to the model with the highest median AUC. This process was implemented via the `winner_model_selection()` function.

As shown in Table 1, Logistic Regression with Elastic Net achieved the highest MCC with non-overlapping confidence intervals compared to the next-best model. It also performed competitively in terms of AUC and Balanced Accuracy, making it a statistically and practically justified winner.

Conceptually, this result is both expected and justified. Logistic regression is known for its stability, interpretability, and strong baseline performance, especially in structured, low-dimensional datasets such as the one used here (30 features). The addition of Elastic Net regularization helped balance between L1 (sparsity) and L2 (ridge-like stability), which likely contributed to generalization without overfitting. Furthermore, its linear decision boundary is often well suited for problems with a degree of class separability, as supported by the PCA analysis.

4 Conclusion

This study addressed the problem of binary classification of breast cancer tumors using machine learning and rigorous evaluation methodology. We developed a repeated nested cross-validation (rnCV) framework that enabled unbiased performance estimation and reliable model selection across six candidate algorithms. By incorporating robust metrics for imbalanced classification tasks and using bootstrap resampling to construct confidence intervals, we ensured that model comparisons were statistically meaningful.

Logistic Regression with Elastic Net regularization was selected as the final model due to its superior median MCC and AUC, combined with non-overlapping confidence intervals against other classifiers. Its simplicity, interpretability, and resistance to overfitting make it a strong candidate for clinical deployment in diagnostic support systems.

However, some limitations should be acknowledged. We did not apply advanced feature selection or data balancing techniques, which might further improve model performance. Additionally, while the rnCV approach offers strong internal validation, external validation on an independent dataset would be necessary before clinical integration.

Future work could explore ensemble models, domain-specific feature engineering, and real-world deployment pipelines. Overall, this project demonstrates that with careful methodology and appropriate statistical tools, even relatively simple models can achieve competitive and trustworthy performance in biomedical classification tasks.

References

[1] scikit-learn, "Plot Nested Cross-Validation Iris," *scikit-learn documentation*, [Online]. Available: `https://scikit-learn.org/stable/auto_examples/model_selection/plot_nested_cross_validation_iris.html`. [Accessed: May 9, 2025].

[2] C. D., "A Guide to Nested Cross-Validation With Code: Step-by-Step," *Medium*, [Online]. Available: `https://medium.com/@cd_24/a-guide-to-nested-cross-validation-with-code-step-by-step-6a8ad06d5af2`. [Accessed: May 9, 2025].

[3] StatQuest with Josh Starmer, "Nested Cross Validation Clearly Explained!!!," *YouTube*, Apr. 27, 2021. [Online]. Available: `https://www.youtube.com/watch?v=X6HeBURppHc`. [Accessed: May 9, 2025].

[4] scikit-learn developers, "sklearn.model_selection.GridSearchCV," *scikit-learn documentation*, [Online]. Available: `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`. [Accessed: May 9, 2025].

Github Repository Link

The complete code, data preprocessing scripts, and the final trained model are available in the GitHub repository: `https://github.com/maria-defteraiou/Assignment-2.git`

AI Assistance Disclosure

I used OpenAI's ChatGPT (GPT-4) as a writing and coding assistant during the development of this assignment. It supported me with:

- Clarifying assignment requirements and interpreting evaluation metrics.
- Structuring the LaTeX document and improving section transitions.
- Writing function summaries and generating some markdown/boilerplate explanations.
- Revising draft text for clarity, grammar, and conciseness.

All machine learning implementation, modeling decisions, result interpretation, and validation logic were developed and executed by me.