MSc Data Science & Information Technologies

Bioinformatics - Biomedical Data Science Specialization

Course: Machine Learning in Computational Biology

Professor: Elias Manolakos

MLCB 2024 Assignment #2 - Nested Cross Validation using Object-Oriented Programming

Author	*Student ID*
Konstantinos Giatras	7115152300005

DEPARTMENT OF INFORMATICS & TELECOMMUNICATIONS

HELLENIC REPUBLIC
National and Kapodistrian
University of Athens
EST. 1837

2023-2024

Contents

Abstract

This technical report details the development and evaluation of a machine-learning pipeline aimed at predicting diabetes through binary classification. The pipeline comprises data preprocessing, repeated nested cross-validation with Bayesian optimization, and final model training and selection, by performing classifier algorithm evaluations using 50 nested cross-validation loops and Bayesian hyperparameter tuning via the Optuna library. We evaluated various classifiers, including Gaussian Naive Bayes, k-Nearest Neighbors, Logistic Regression, Linear Discriminant Analysis, and Support Vector Machines (SVM). Our results demonstrated that SVM with feature selection achieved the best performance, with a median Matthews Correlation Coefficient (MCC) of 0.455, slightly outperforming Logistic Regression, which had a median MCC of 0.442 without feature selection. The SVM model was also more stable, with a lower standard deviation in MCC scores. These findings highlight the potential of advanced machine learning techniques in enhancing early diabetes detection and management, providing a robust predictive tool that can facilitate timely medical interventions and personalized treatment strategies.

Introduction

Diabetes mellitus is an increasingly prevalent global health issue marked by chronic hyperglycemia due to dysfunctional insulin production or action. Estimates predict that by 2045, 693 million adults will be affected by diabetes, underscoring the critical need for understanding and managing this disease (1). Type 2 diabetes, in particular, has seen a notable rise globally, complicating public health efforts due to its related morbidity and mortality (2). The condition results in both macrovascular complications like cardiovascular disease and microvascular issues such as kidney disease, neuropathy, and retinopathy, all of which significantly degrade quality of life and elevate healthcare costs (1).

Early and precise diagnosis of diabetes and its complications is vital for effective management and intervention. Advanced machine learning models offer promising tools for predicting diabetes onset and progression by analyzing large datasets and identifying patterns not discernible by human analysis. Recent studies highlight the complex interplay of genetic factors in diabetes predisposition and complications, which machine learning can effectively investigate (1). Additionally, the high incidence of complications in youth-onset diabetes emphasizes the urgent need for predictive technologies to identify at-risk individuals early, facilitating preemptive medical strategies and lifestyle changes to mitigate long-term negative outcomes (3). By integrating machine learning with comprehensive genetic, lifestyle, and clinical data, researchers can better predict disease trajectories, personalize treatments, and ultimately reduce the prevalence and impact of diabetes-related complications.

In this technical report, we present a machine-learning pipeline designed to predict diabetes, inspired by recent advancements in the field showing promising results. We assess the effectiveness of various classification algorithms, focusing on selecting and fine-tuning the hyperparameters of the most effective model. Hasan et al. (2020) (4) developed an ensemble approach that integrates multiple classifiers and techniques like outlier rejection and feature selection, achieving an AUC of 0.950, a 2.00% improvement over previous models using the Pima Indian Diabetes Dataset. Similarly, Khanam and Foo (2021) (5) evaluated various machine learning algorithms and a neural network model, finding that a neural network with two hidden layers reached an 88.6% accuracy on the same dataset. These studies demonstrate significant improvements in predictive accuracy, setting new standards for early diabetes detection. Our goal is to create a robust predictive tool for early diabetes detection and management, potentially improving on these recent benchmarks.

Materials and Methods

Dataset Description and Exploration

This study involves a dataset of 506 samples, which includes both diabetes patients and healthy controls. Each sample is identified by a patient's ID and includes 8 features. The features and their types are detailed in Table 1.

Feature	Type
Pregnancies	Discrete
Glucose	Discrete
BloodPressure	Discrete
SkinThickness	Discrete
Insulin	Discrete
BMI	Continuous
DiabetesPedigreeFunction	Continuous
Age	Discrete

Table 1: Feature Description of the Diabetes dataset

To get a rudimentary sense of the feature space and class separability, we visualize the features with respect to the class labels in a pair plot and individual feature distributions. A pair plot allows us to visualize pairwise relationships between features and how they relate to the target class. This helps us see if there are any visible separations between the classes (Figure 4). We also create individual distribution plots (histograms) for each feature, colored by the class labels to see how the distributions differ between the classes (Figure 5). Based on those plot, we can deduce that features like Glucose and BMI show the most promise, with higher values more associated with Diabetes patients, demonstrating good separation.

We also performed feature correlation analysis, which involves calculating the Pearson correlation coefficients between all pairs of features to understand how they are related. Correlation coefficients range from -1 to 1, with values close to 1 indicating a strong positive correlation, values close to -1 indicating a strong negative correlation, and values close to 0 indicating little to no linear relationship between the features. Here, we calculated the correlation matrix for the features and visualized it using a heatmap, where the color intensity represents the correlation coefficient values, to identify strongly correlated features. Even though we did not observe any particularly strong correlations between features, we note some notable positive correlations (>0.3) between: 'SkinThickness' and 'BMI' (0.54), 'Pregnancies' and 'Age' (0.53), 'Glucose' and 'Insulin' (0.39), 'BloodPressure' and 'Age' (0.32), 'SkinThickness' and 'Insulin' (0.31) (Figure 6).

Regarding the target labels, the samples are binary classified into healthy controls ('Outcome'=0, negative class) or diabetes patients ('Outcome'=1, positive class). It is important to highlight that, typical of many medical datasets, there is an imbalance in the samples between the labels, with the positive class (diabetes patients) being less prevalent, constituting about 50% of the number of samples of the majority negative class (healthy controls).

During our exploratory data analysis, we noted that the feature range of values for many laboratory results in the dataset vary widely, indicating the need for data scaling during preprocessing prior to the machine learning task. Furthermore, most of these features contain outliers, defined as observations that lie above or below 150% of the interquartile range (IQR) from the first and third quartiles, respectively (Figure 1).

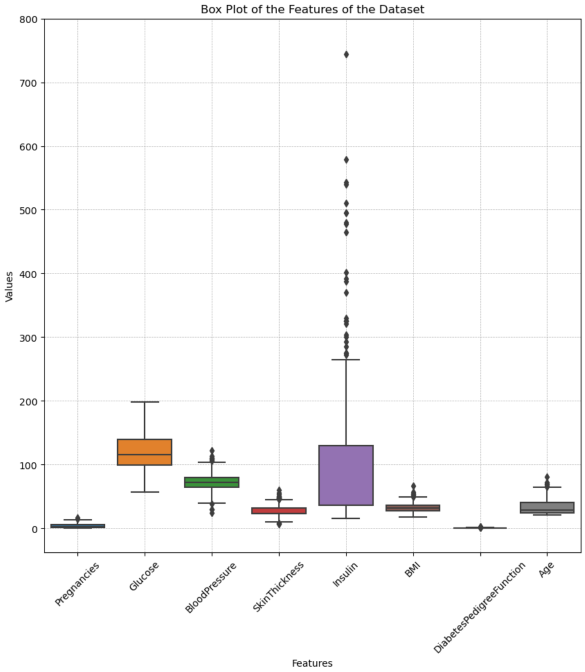


Figure 1: Box plot of the distributions of the features before preprocessing

Pipeline Structure

Machine learning pipelines offer numerous benefits, including reproducibility, scalability, and efficiency. They provide a clear and interpretable framework for implementing machine learning methods. The pipeline we created for diabetes prediction comprises three key steps:

1. Data Preprocessing
2. Repeated Nested Cross Validation using Bayesian Optimization
3. Final Model Training & Selection

Data Preprocessing

Data preprocessing included exploring the removal of outliers by eliminating observations that fall 1.5 times the Interquartile Range (IQR) below or above the first (Q1) and third (Q3) quartiles respectively. However, this approach was found to be inappropriate as it reduced the positive class ratio to 31.4% (135 out of 430 data points) from the original 35% (177 out of 506 data points). This reduction suggests that outliers hold crucial information for the positive class and should be preserved, particularly in medical datasets such as this one, where deviations from the norm often signify pathological conditions.

Preprocessing is also able to deal duplicate row removal and handling missing values. In the case of the existence of null values or zeros in columns where they should not be ('Glucose', 'BloodPressure', 'SkinThickness', 'Insulin' and 'BMI'), these values are replaced with the median value of the respective feature.

Regarding feature scaling, normalization of the data using the MinMax Scaler was selected. This technique transforms features by scaling each feature to a given range, typically between 0 and 1, and it also preserves the relationships between the original data points. It is useful in cases where no particular distribution of the data is assumed and the features have different scales.

Normalization is also a prerequisite for PCA, which is a technique used to reduce the dimensionality of a dataset while retaining most of the variance in the data. It transforms the original features into a new set of features called principal components, which are orthogonal (uncorrelated) and ordered by the amount of variance they explain. Here, we performed PCA by fitting PCA on the normalized features, transforming the features to the new principal components, calculating the explained variance by each principal component, as well as visualizing the data in the new principal component feature space (Figure 7). Notably, the total explained variance of the first three principal components was 0.6711, which is adequate, but might not be enough.

Repeated Nested Cross Validation using Bayesian Optimization

For the main part of the pipeline, a custom class was created, called 'NestedCrossValidation'. This class performs repeated nested cross-validation, involving an outer loop for model evaluation and an inner loop for hyperparameter tuning. It preprocesses data by handling missing values, detecting outliers, scaling features, and optionally selecting features and applying PCA. The class optimizes hyperparameters using Optuna (6; 7), evaluates multiple classifiers with various performance metrics, and stores results to identify the best-performing model. Finally, it saves the trained model with the optimal hyperparameters.

By integrating nested cross-validation with Bayesian hyperparameter tuning, the 'NestedCrossValidation' facilitates a thorough assessment of the model's performance and safeguards against overfitting to any specific train/test split (also taking class imbalance into account). Bayesian hyperparameter tuning is preferred to grid or random search methods, as it provides more insightful and efficient tuning. 'NestedCrossValidation' supports several classifier types, including Gaussian Naive Bayes (GNB), k-Nearest Neighbours (KNN), Logistic Regression (LR), Linear Discriminant Analysis (LDA), and Support Vector Machines (SVM).

The remaining parameters include the number of cross-validation iterations, as well as the number of outer and inner loop cross-validation folds. We have implemented 10 uniquely seeded nCV iterations, with 5 outer and 3 inner loop folds, resulting in a total of 50 nested cross-validation loops per classifier evaluated. In each of these 50 loops, the optimal classifier parameters from the inner loop and various metrics such as Matthews Correlation Coefficient (MCC), Balanced Accuracy, F1, F2, Recall, Specificity, Precision, Average Precision and Negative Predictive Value (NPV) are calculated and stored.

Feature Selection

We would like to investigate whether or not the inclusion of feature selection will improve the performance of our models. To this end, we will run our entire machine learning pipeline twice: once without feature selection and once with feature selection using 'SelectKBest' with the mutual information scoring function, and compare the results. Feature selection with 'SelectKBest' is a method in scikit-learn that selects the top k features (5 in this case) based on a specified scoring function. In this context, the scoring function used is 'mutual_info_classif', which measures the mutual information (MI) between each feature and the target variable. Mutual information quantifies the dependency between two variables, providing a non-negative value that is zero if and only if the variables are independent. Higher MI values indicate a higher dependency. This method is particularly beneficial for capturing any kind of dependency between variables, not just linear relationships, thus helping in identifying the features that provide the most information about the target variable.

Final Model Selection & Tuning

Final Model Selection & Tuning is achieved by selecting the classifier algorithm that attains the highest average Matthews Correlation Coefficient (MCC) across the 50 nested cross-validation loops. The MCC is calculated using the formula:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

The MCC metric is utilized as the selection criterion because it provides high scores only when the classifier effectively predicts both positive and negative cases, making it robust against class imbalance. The MCC value ranges from -1 to +1, where -1 represents perfect misclassification, +1 denotes perfect classification, and 0 indicates random performance by the classifier (8).

After identifying the best-performing classifier algorithm, it is trained on the entire feature set. This process involves Bayesian hyperparameter tuning using a non-nested 5-fold cross-validation setup within the 'NestedCrossValidation' class, employing optuna yet again to fine-tune the optimal hyperparameters.

An overview of the diabetes repeated nested cross-validation with bayesian optimization pipeline scheme can be seen in Figure 8.

Results and Discussion

No Feature Selection Results

In the model evaluation stage depicted in Figure 9, the performance of various classifiers is critically analyzed. The Gaussian Naive Bayes (GNB) classifier exhibits a high mean score across several metrics, yet its broader MCC distribution points to inconsistent results. On the other hand, k-Nearest Neighbors (KNN) show moderate performance, and Linear Discriminant Analysis (LDA) delivers solid results with a tight MCC distribution, suggesting a higher prediction reliability. Support Vector Machines (SVC) maintain a good balance with competitive MCC scores but do not reach the top.

Amidst these, Logistic Regression (LR) outshines the competition, delivering a median MCC score of 0.441 and showcasing relatively low variability in scores as indicated by a standard deviation of 0.084 across 50 cross-validation (CV) loops. This reflects a substantial 7.9% improvement over the GNB baseline. Notably, LR's performance in F1, Balanced Accuracy, and Recall metrics is robust, hinting at its strong capacity to handle binary classification without overfitting. The MCC distribution for LR is symmetric and slightly skewed towards higher scores, suggesting that it often surpasses the median performance, as shown in Figure 2.

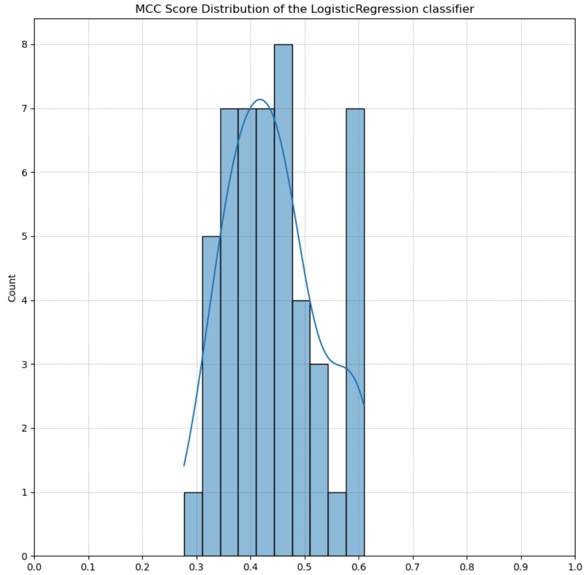


Figure 2: Histogram of 10 bins of the LR MCC score over 50 CV trials without feature selection

Given the totality of evidence, LR is identified as the superior model for the final implementation. It not only performs well in MCC but also demonstrates high consistency across all evaluation metrics, which signifies a well-balanced classification ability. The symmetrical and tight MCC score distribution infers a stable model that promises good generalization to unseen data. This consistent performance across various metrics underscores LR as the most reliable classifier for differentiating between healthy individuals and diabetes patients, confirming its selection as the final baseline model for the no feature selection approach.

Feature Selection Results

In this pipeline, the feature selection process is performed using the 'SelectKBest' method with 'mutual_info_classif' as the scoring function, provided that the 'feature_selection' parameter is set to True when initializing the 'NestedCrossValidation' class. The rest of the pipeline is identical. The performance of the classifiers can be seen in Figure 10, where, through similar observations like before, we conclude that the best classifier is SupportVectorMachine (SVC), which achieved a median MCC score of 0.455 over 50 CV iterations, with a standard deviation of 0.079. Similarly to before, the MCC score distribution appears to be symmetric around the median, with a slight inclination for scores to cluster above the it (Figure 3).

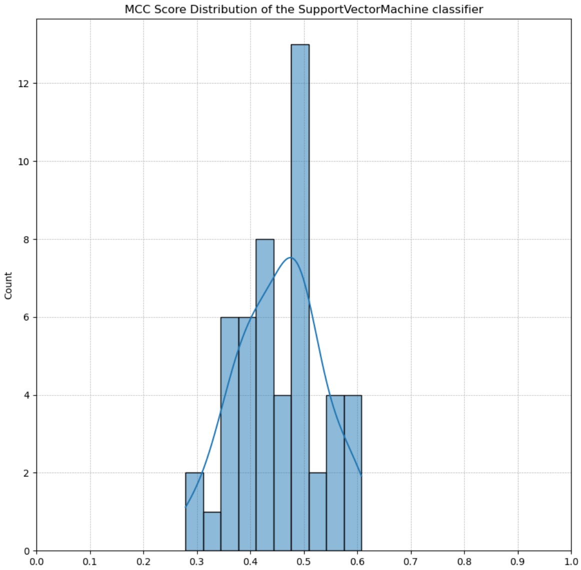


Figure 3: Histogram of 10 bins of the SVC MCC score over 50 CV trials with feature selection

We can see that the median performances of our two models are similar (LogisticRegression: median MCC = 0.442, median std = 0.087; SupportVectorMachine: median MCC = 0.455, median std = 0.079). To determine the better classifier, we used a 95% confidence interval of the medians. Our structured approach involved nested cross-validation and bootstrapping, with functions to calculate confidence intervals for the median MCC scores from the results. The confidence intervals were calculated for scenarios with and without feature selection and compared to identify significant differences.

Since no significant difference was found, we selected and the 'SupportVectorMachine_FS_PCA3_final_model.pkl' model as the better one due to its slightly higher median MCC score and lower median standard deviation. This model is stored in the 'final_model' subdirectory inside the 'models' directory.

Conclusions

In this analysis, a comprehensive machine learning pipeline was developed to predict diabetes using a dataset with multiple features. The pipeline included steps for data preprocessing, repeated nested cross-validation with Bayesian optimization, and final model training and selection. Two approaches for feature selection were compared: one without feature selection and one using the 'SelectKBest' method with mutual information scoring. The Support Vector Machine (SVM) classifier achieved the best performance with a median MCC score of 0.455 in the feature selection scenario, slightly outperforming the Logistic Regression model which had a median MCC of 0.442 without feature selection. The comparison showed no significant differences between the two approaches, but SVM with feature selection had a marginally higher median MCC score and lower standard deviation, suggesting slightly better stability and performance.

However, several limitations were identified in this analysis. The dataset used had an imbalance between the positive and negative classes, which can affect the model's ability to generalize well to unseen data. Additionally, while the feature selection process helped in improving the model's performance marginally, it might not have captured all relevant features due to its reliance on mutual information, which may not fully account for complex relationships among features. Moreover, the nested cross-validation process, though thorough, is computationally expensive and may not be feasible for larger datasets or more complex models without significant computational resources.

Future research should explore alternative feature selection methods that can capture non-linear relationships between features and the target variable, such as those based on deep learning techniques. Additionally, addressing the class imbalance more effectively through techniques such as SMOTE or adaptive resampling strategies could further improve model performance. Investigating ensemble methods that combine multiple classifiers might also yield better predictive accuracy and robustness. Lastly, applying this pipeline to a larger and more diverse dataset could help validate its effectiveness and enhance its generalizability.

LLM Usage

During this assignment, ChatGPT was used for a variety of purposes:

- Explanation of concepts to assist with further understanding and insight
- Assistance in code readability (e.g. help with adding informative comments)
- Suggestions for code error handling and bug fixing
- Task management and organization (e.g. breaking a larger task into more, smaller ones)

References

[1] E. W. Gregg, N. Sattar, and M. K. Ali, "The changing face of diabetes complications," *The Lancet Diabetes & Endocrinology*, vol. 4, no. 6, pp. 537–547, Jun 2016.

[2] J. B. Cole and J. C. Florez, "Genetics of diabetes mellitus and diabetes complications," *Nature Reviews Nephrology*, vol. 16, no. 7, pp. 377–390, May 2020.

[3] T. S. Group, "Long-term complications in youth-onset type 2 diabetes," *New England Journal of Medicine*, vol. 385, no. 5, pp. 416–426, Jul 2021.

[4] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, pp. 76 516–76 531, 2020.

[5] J. J. Khanam and S. Y. Foo, "A comparison of machine learning algorithms for diabetes prediction," *ICT Express*, vol. 7, no. 4, pp. 432–439, Dec 2021.

[6] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, jul 2019.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[8] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, Jan 2020.

Supplementary Material

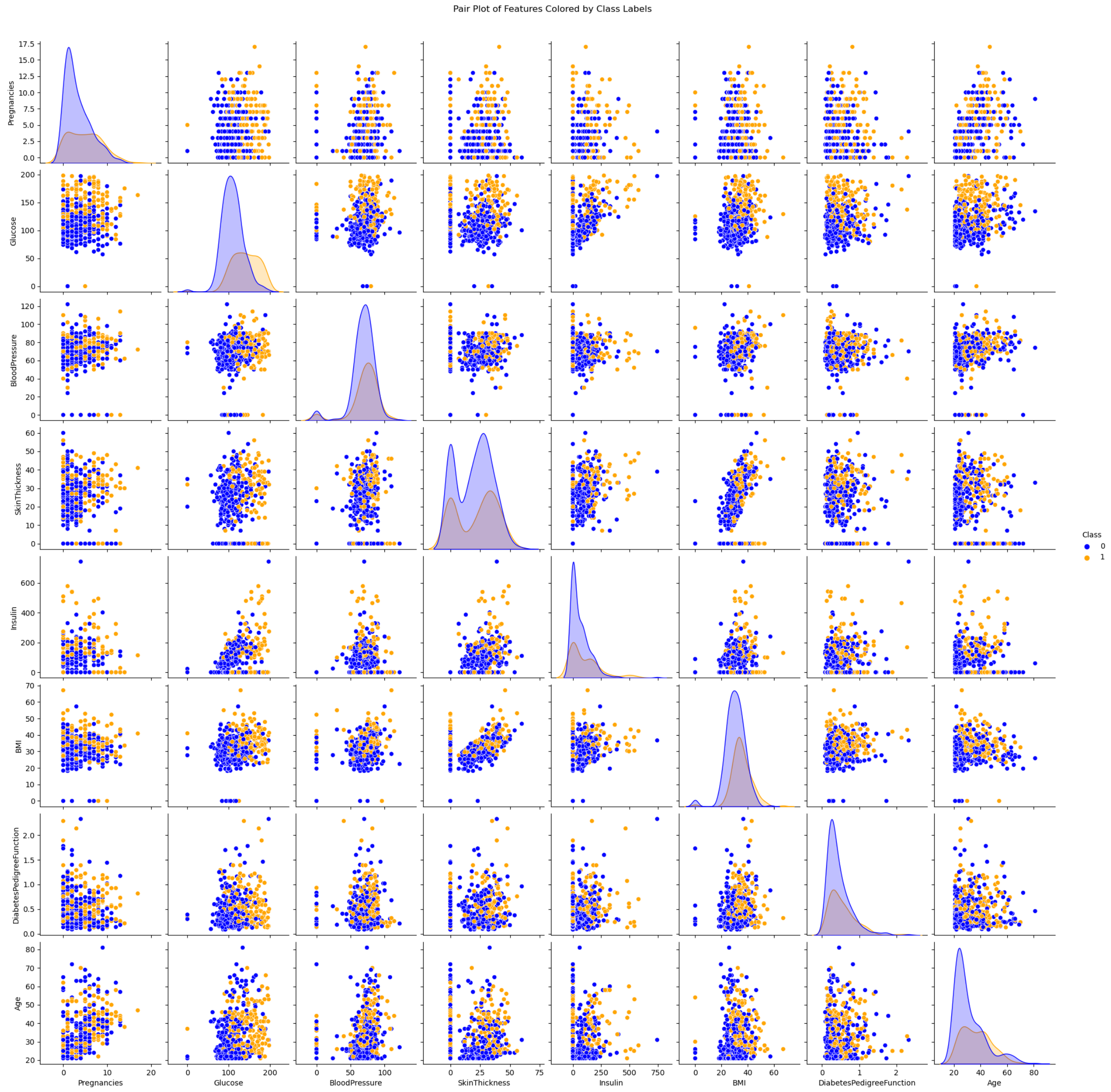


Figure 4: Pair plot of features colored by class labels

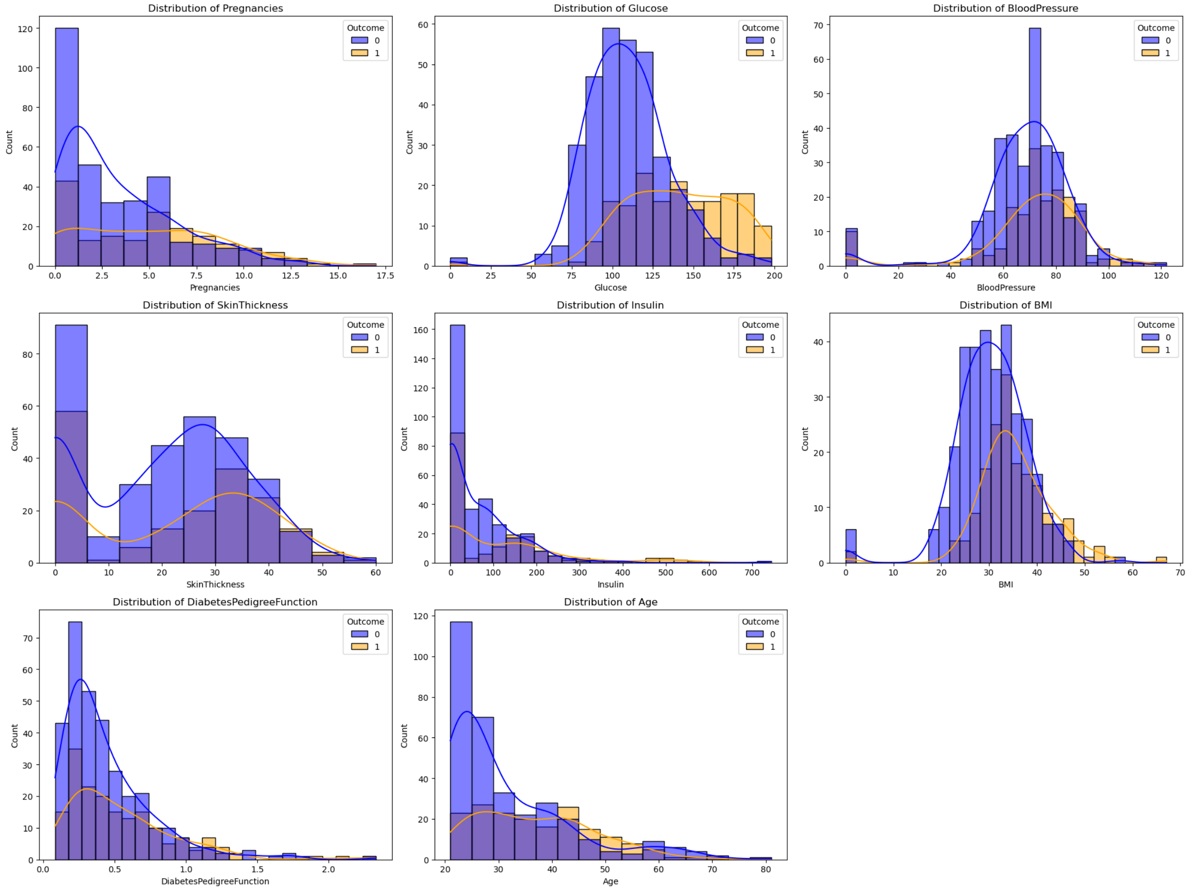


Figure 5: Distributions plots of features colored by class labels

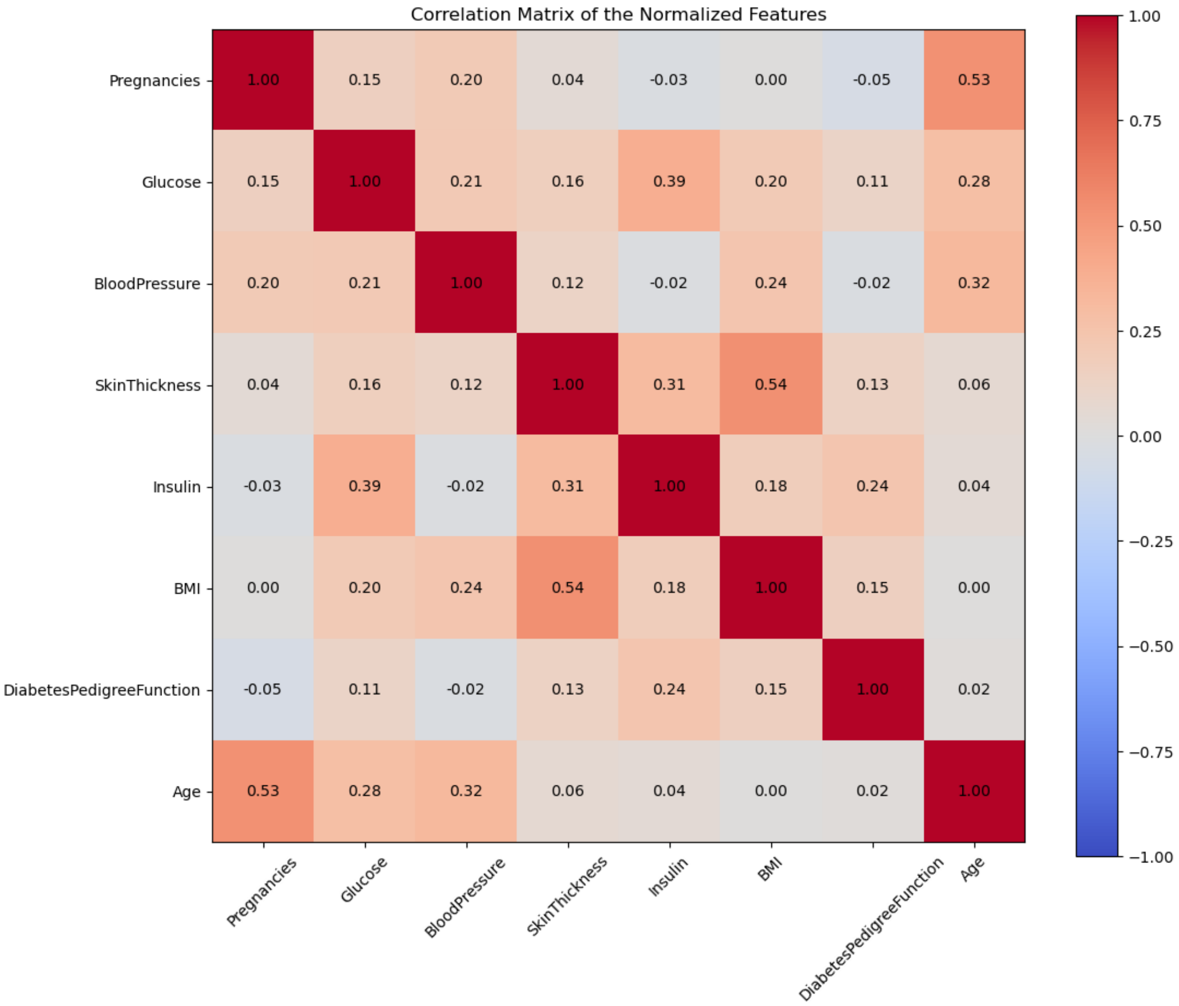


Figure 6: Correlation matrix of the normalized features

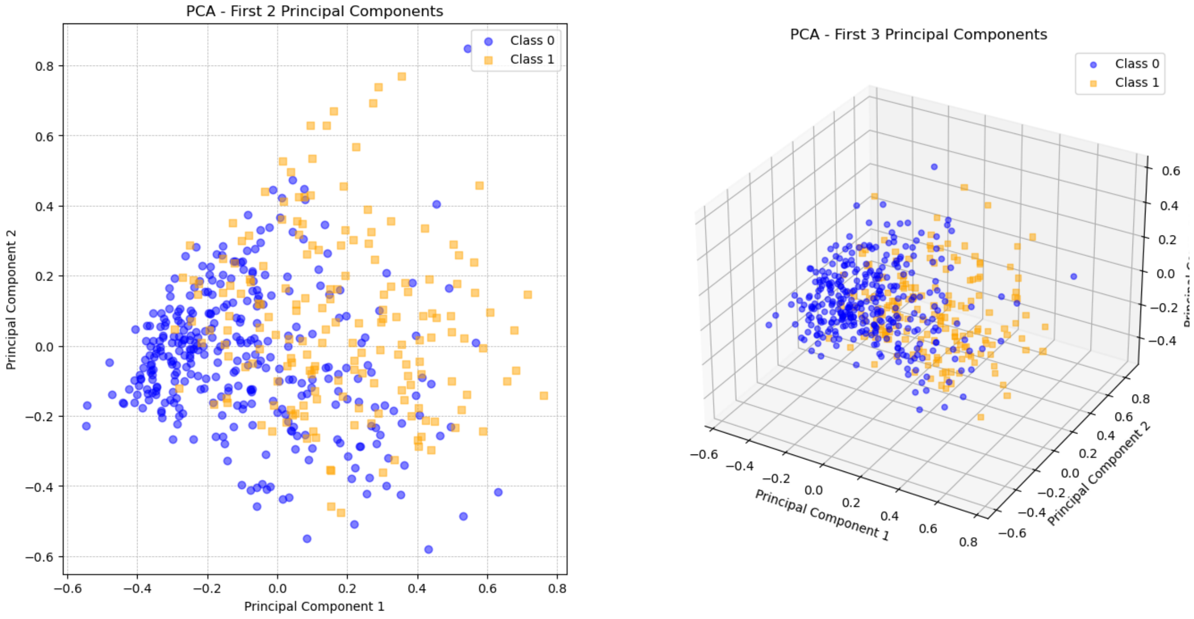


Figure 7: Visualization of the first two and first three principal components after applying PCA to the normalized features

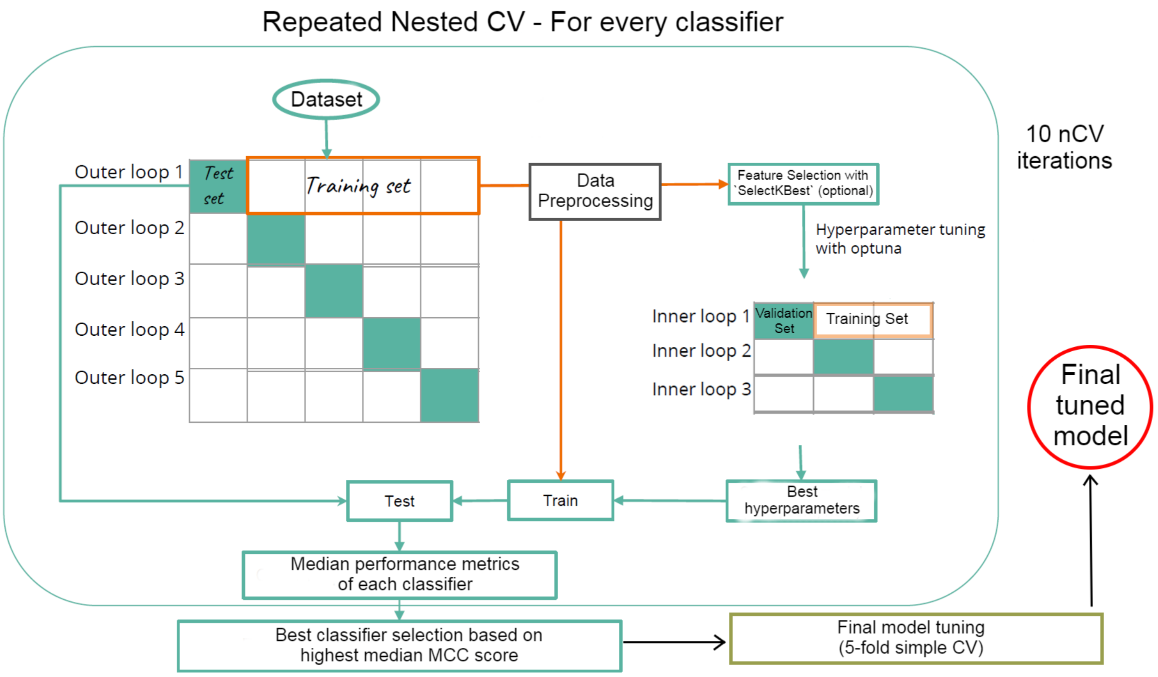
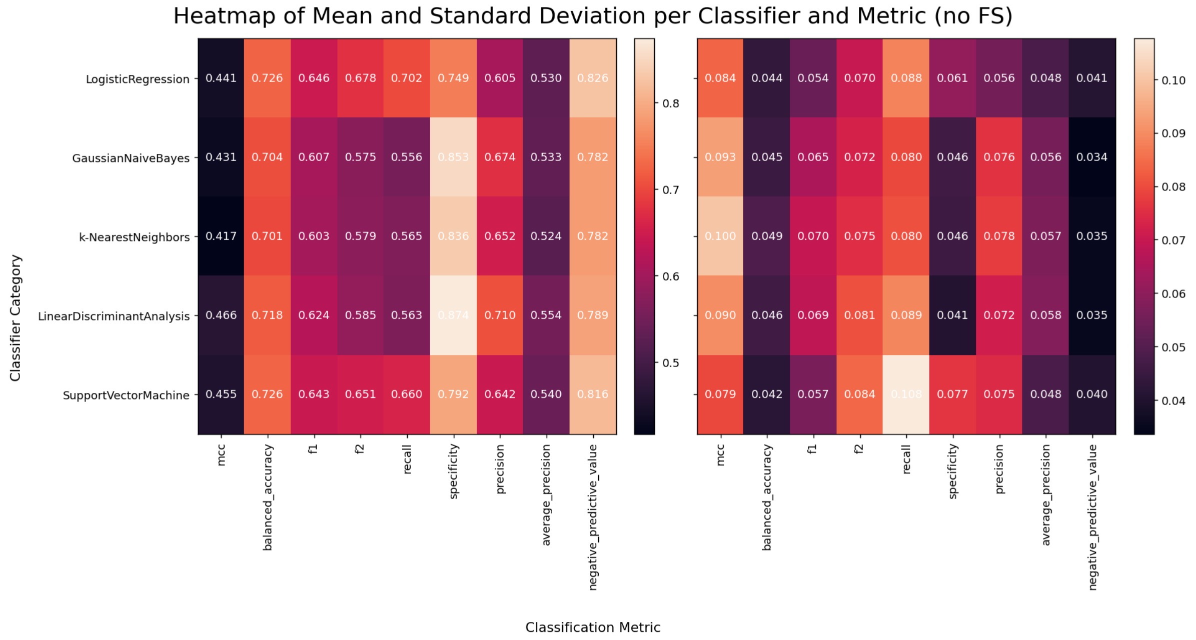
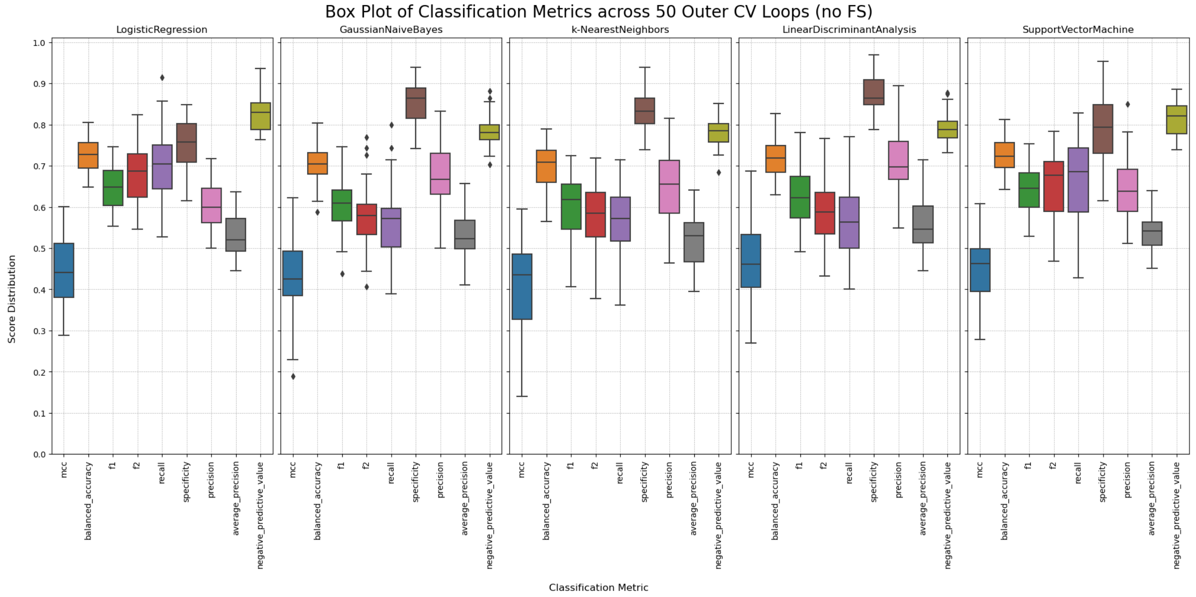

Repeated Nested CV - For every classifier
Dataset
Outer loop 1
Outer loop 2
Outer loop 3
Outer loop 4
Outer loop 5
Test set
Training set
Data Preprocessing
Feature Selection with `SelectKBest` (optional)
Hyperparameter tuning with optuna
Inner loop 1
Inner loop 2
Inner loop 3
Validation Set
Training Set
10 nCV iterations
Final tuned model
Test
Train
Best hyperparameters
Median performance metrics of each classifier
Best classifier selection based on highest median MCC score
Final model tuning (5-fold simple CV)

Figure 8: Diabetes repeated nested cross-validation with bayesian optimization pipeline scheme

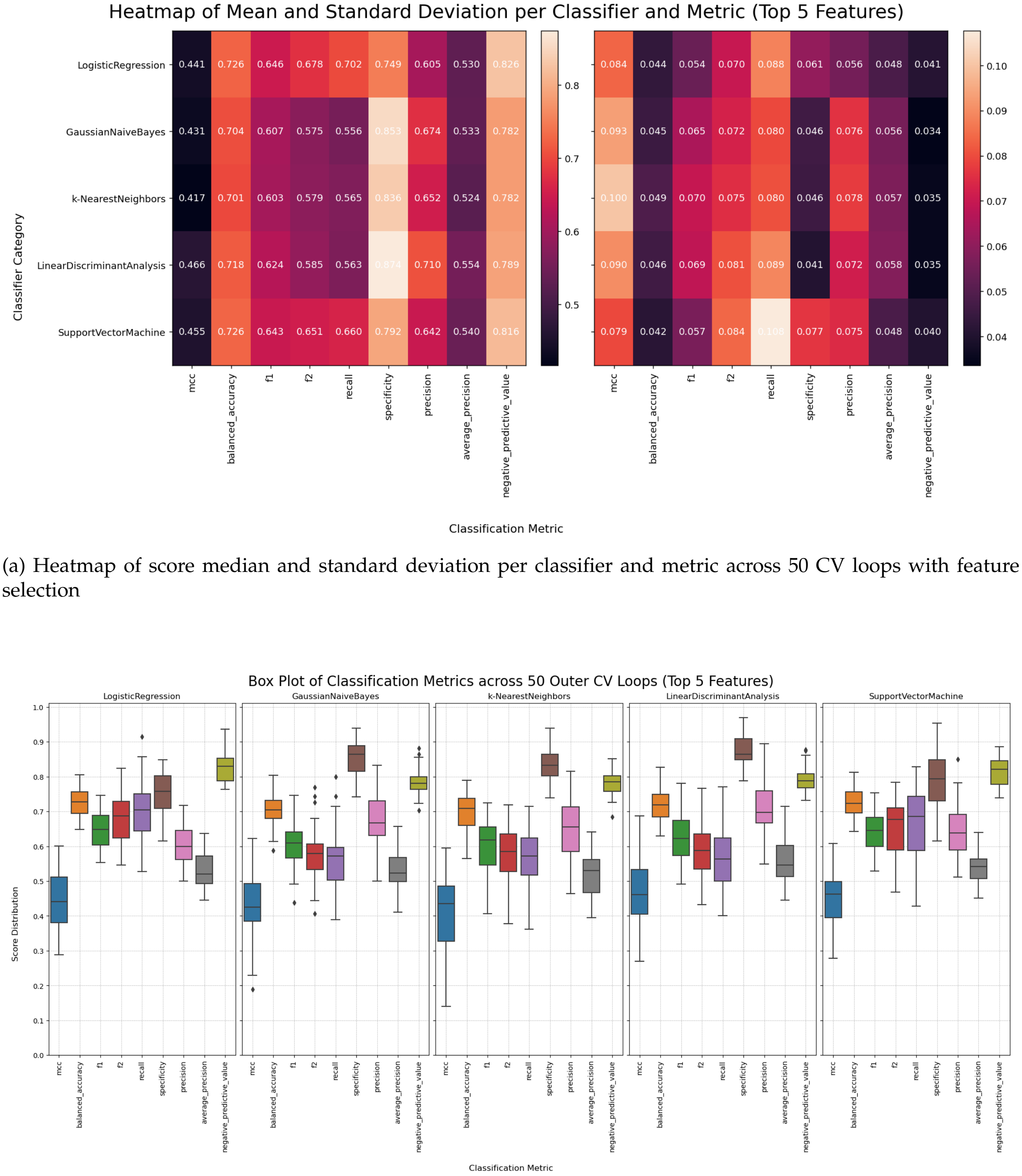


(a) Heatmap of score median and standard deviation per classifier and metric across 50 CV loops without feature selection



(b) Distribution of classification metrics across 50 outer CV loops without feature selection

Figure 9: Results of the pipeline without feature selection



(a) Heatmap of score median and standard deviation per classifier and metric across 50 CV loops with feature selection

(b) Distribution of classification metrics across 50 outer CV loops with feature selection

Figure 10: Results of the pipeline with feature selection